# Apéndice A: Plan de Proyecto Software

## A.1. Introducción.

La planificación del proyecto es una parte fundamental del ciclo de vida de este. En este caso se han usado metodologías ágiles adaptadas a las necesidades y características particulares requeridas por la naturaleza del TFG, estas se verán con mayor profundidad en los siguientes apartados.

En el primer apartado de este apéndice se comentará la organización del proyecto usando como referencia los Sprints creados en ZenHub, herramienta utilizada para realizar un seguimiento de tareas.

En el segundo apartado se profundizará en la viabilidad económica…

## A.2. Planificación temporal

Dado que el equipo responsable del proyecto es significantemente más pequeño de lo que sería habitual se han tenido que hacer algunos ajustes en las metodologías usadas, pero se ha mantenido su filosofía:

* Se fijó la duración de los Sprints en dos semanas. Se planteó en un inicio Sprints con una semana de duración, pero rápidamente se vio que se necesitaba más tiempo para poder profundizar y explorar la complejidad de las diferentes librerías que se han usado.
* Cada vez que un Sprint finalizase se programó una reunión con el objetivo de comentar una breve evaluación del estado del proyecto hasta ese momento y planificar el siguiente Sprint.
* La monitorización del proyecto se hizo con las herramientas que facilita la extensión Zenhub. Siguiendo la filosofía de Kanban se crearon tres pipelines en las que se han ido etiquetando todas las issues:
  + Backlog, el equivalente a la lista “por hacer”, donde se han ido colocando todas las issues que iban a ser tratadas en ese Spring.
  + En progreso, donde se han ubicado las issues en las que se estaba trabajando en ese momento concreto.
  + Review. Pipeline añadida para hacer una revisión de las issues antes de cerrarlas.
  + Closed, similar a “finalizadas”. Tras ser cerradas, todas las issues se colocan automáticamente en este pipeline.

Se estimó una duración máxima de cada issue basándose en story points. Cada story point se ha estimado como días activos de trabajo. De esta forma se estima la finalización de un issue con 2 story points en 2 días de trabajo que pueden o no ser consecutivos.

Al no crear una correspondencia fija entre los story points y la duración de las issues, se permite crear un espacio de trabajo mucho más flexible donde las prioridades de las diferentes tareas pueden ser modificadas en todo momento.

ZenHub ofrece una distribución predeterminada de story points siguiendo la secuencia Fibonacci, por lo que no siempre se ha podido elegir un story point adecuado. Para evitar infravalorar el tiempo requerido para finalizar una tarea y hacer hueco para imprevistos, en el caso de dudar entre dos valores, siempre se ha elegido el número superior de la secuencia.

A continuación, se expondrá un estudio detallado de cada Sprint.

### Sprint 0.

[Link.](https://github.com/albarrom/GII_O_MA_21.05/issues#workspaces/tfg-6229527890a395001341cd6b/reports/burndown?milestoneId=Z2lkOi8vcmFwdG9yL1NwcmludC8zMDIzMTQ)

**Fecha inicio y fin:** 11 marzo - 17 marzo

**Objetivo:** Preparación básica: Formación e instalación de herramientas básicas necesarias.

**Story Points:** 9 (no completados).

En ese primer Sprint se realizó una labor de investigación sobre las diferentes herramientas y software que iban a utilizarse. Aún no se controlaba bien la metodología agile y, como se puede ver en la Ilustración 1 o en el enlace al propio Spring, había problemas para asignar tareas o estimar tiempos. De hecho, a pesar de haber sido completadas antes de finalizar el Sprint, las issues creadas no fueron cerradas antes de que terminase.

**Issues:**

* 001- Familiarizarse con conceptos teóricos de metodología agile básicos.
* 002- Instalación y primera toma de contacto con varias herramientas software.
* 003- Familiarización con normativa y especificaciones del TFG.
* 004- Primera toma de contacto con librerías Python.

**Gráfico Burndown:**

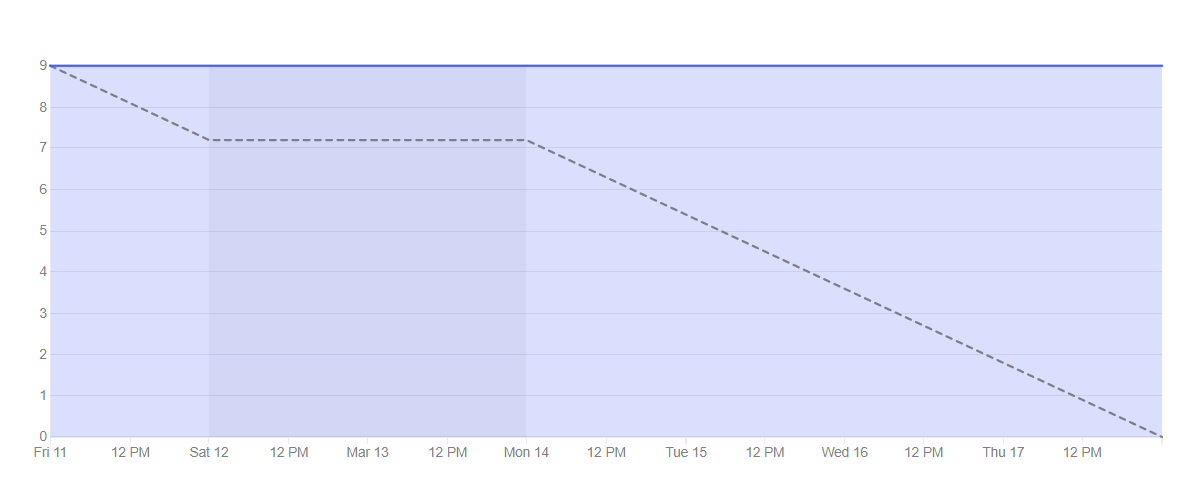


Ilustración 1. Tabla Burndown Sprint 0

### Sprint 1.

[Link](https://github.com/albarrom/GII_O_MA_21.05/issues#workspaces/tfg-6229527890a395001341cd6b/reports/burndown?milestoneId=Z2lkOi8vcmFwdG9yL1NwcmludC8zMDIzMTU)

**Fecha inicio y fin**: 18 marzo - 31 marzo

**Objetivo:** Crear un notebook que pueda mostrar un gráfico.

**Story Points:** 35.

Este Sprint se enfocó en familiarizarse con la librería SQLite de Python y crear una base de datos, con la idea de poder enriquecer el dashboard usando encuestas de otros años en futuras fases del proyecto.

Hubo también un cambio de planificación: se decidió crear sprints de 15 días en lugar de 7, que era la idea original.

Las tareas:

* Algunos campos numéricos en el csv contienen texto.
* No se puede abrir el HTML en localhost a través de jupyter.
* Error visualizando los datos en el gráfico.

fueron creadas debido a un error de concepto con la etiqueta “problema”. Esto ha sido corregido y en futuros Sprints no se crearán issues con estas mismas características.

Las milestones cumplen la misma función que los Sprints[[1]](#footnote-1), debido a un error de concepto se estaban usando ambos. Se cesará el uso de milestones a partir de este Sprint.

**Issues:**

* Cargar csv + usar librería SQLite en notebook
* Organizar todos los elementos necesarios para empezar a escribir la documentación.
* Algunos campos numéricos en el csv contienen texto.
* Crear un notebook que muestre el csv de alguna forma.
* Profundizar en Dash.
* No se puede abrir el HTML en localhost a través de jupyter.
* Error visualizando los datos en el gráfico.

**Gráfico Burndown:**

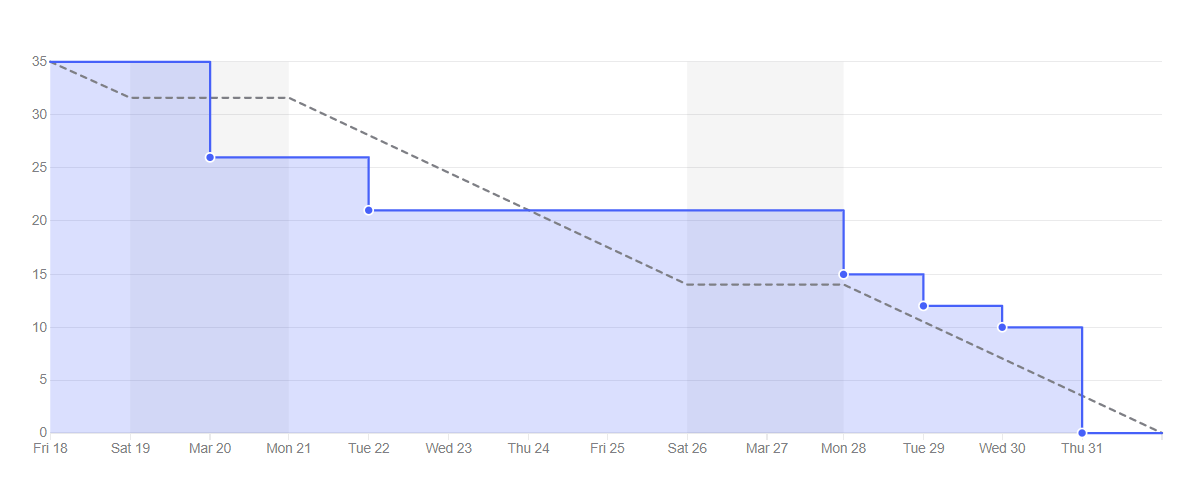


Ilustración 2. Tabla Burndown Sprint 1

### Sprint 2.

[Link.](https://app.zenhub.com/workspaces/tfg-6229527890a395001341cd6b/reports/burndown?entity=sprints&milestoneId=Z2lkOi8vcmFwdG9yL1NwcmludC8zMDIzMTY&invite=true)

**Fecha inicio y fin:** 1 abril - 14 abril

**Objetivo:** Familiarizarse con la creación de varios gráficos y charts.

**Story Points:** 39

El Sprint se ha basado principalmente en la familiarización con la librería plotly express. Para ello se han creado diferentes gráficos y se ha investigado en profundidad cómo funcionan: añadir diferentes colores en la leyenda, usar pestañas, explorar nuevas formas de que el usuario introduzca input e introducir porcentajes.

**Issues:**

* Añadir información en bibliografía. Apartado Técnicas y herramientas
* Elegir gráficos
* Preprocesar datos
* Generar gráficos.
* Documentar el Sprint 18/Mar - 31/Mar.
* Generar un notebook que genere una página con más de un gráfico.

**Gráfico Burndown:**

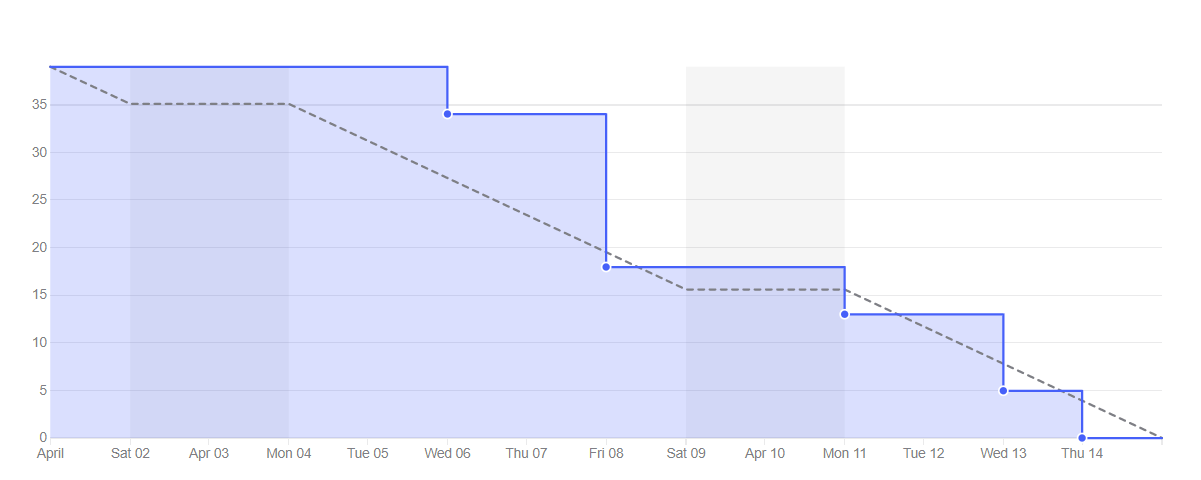


Ilustración 3. Tabla Burndown Sprint 2

### Sprint 3.

[Link.](https://github.com/albarrom/GII_O_MA_21.05/issues#workspaces/tfg-6229527890a395001341cd6b/reports/burndown?milestoneId=Z2lkOi8vcmFwdG9yL1NwcmludC8zMDQ2ODQ)

**Fecha inicio y fin:** 15 abril – 28 abril

**Objetivo:** Deploy app en Heroku.

**Story Points:** 32

Este Sprint se ha basado en continuar profundizando en la librería de plotly express y en realizar un primer deploy de la app en Heroku. Por otro lado, hubo un problema creando las issues “Generar un gráfico que pueda procesar más de un CSV” y “Crear un único archivo con varios csv”.

Ambas se crearon con un objetivo similar: usar encuestas de varios años para generar un único gráfico usando datos de todas las encuestas en el caso de la primera issue y usar encuestas de varios años para generar varios gráficos, cada uno de ellos haciendo referencia a un único año en el caso de la segunda-. Tras una reevaluación de prioridades se decidió simplemente completar la primera issue ya que la segunda no iba a aportar información o conocimientos relevantes.

Se comenzó a conectar las issues con los commits a los que hacen referencia.

**Issues:**

* Usar un path para cargar los csv.
* Familiarizarse con Heroku.
* Crear un histograma.
* Añadir documentación.
* Generar un gráfico que pueda procesar más de un CSV
* Crear un único archivo con varios csv.
* Deploy app con Heroku

**Gráfico Burndown:**

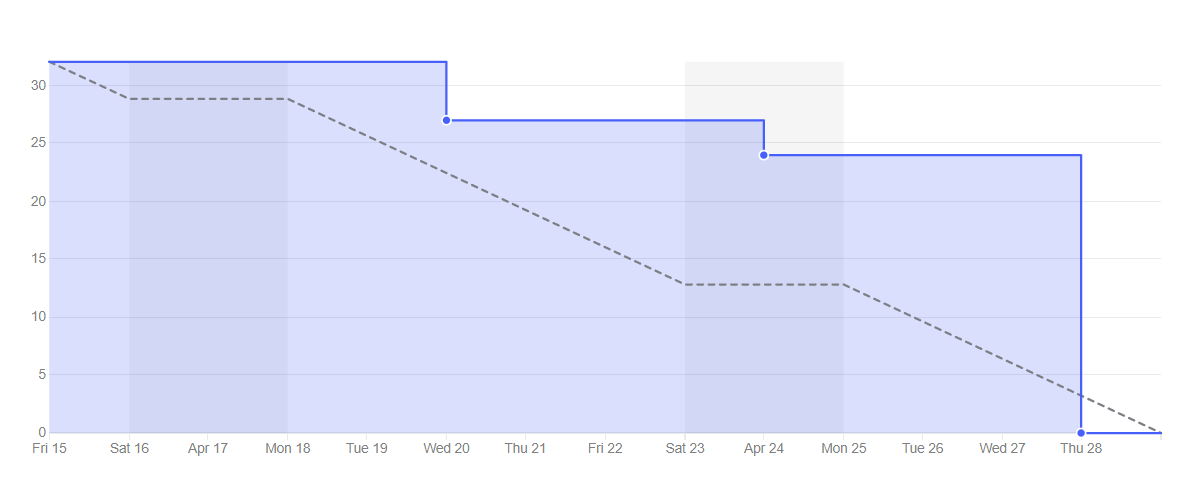


Ilustración 4 Tabla Burndown Sprint 3

### Sprint 4.

[Link.](https://github.com/albarrom/GII_O_MA_21.05/issues#workspaces/tfg-6229527890a395001341cd6b/reports/burndown?milestoneId=Z2lkOi8vcmFwdG9yL1NwcmludC8zMTgxNzQ)

**Fecha inicio y fin:** 29 abril – 12 mayo

**Objetivo:** Crear layout definitivo.

**Story Points:** 35

El foco principal de este Sprint ha sido en continuar familiarizándose con conceptos teóricos, en concreto sobre la exportación HTML interactiva en Python y la librería de componentes Bootstrap para Plotly Dash, con el fin de crear un layout definitivo de la app.

Este Sprint cuenta con una gestión temporal particularmente mala debido a que nunca se ha tratado con conceptos web (HTML, CSS, Bootstrap, etc.). La curva de aprendizaje ha sido mucho más elevada de lo que se estimó en un primer momento.

**Issues:**

* Añadir nueva información a la documentación.
* Familiarizarse con conceptos más avanzados de Dash.
* Mejorar la navegación de la página.
* Hacer otro deploy en Heroku de la página mejorada.
* Elegir gráficos definitivos
* Crear layout definitivo del dashboard

**Gráfico Burndown:**

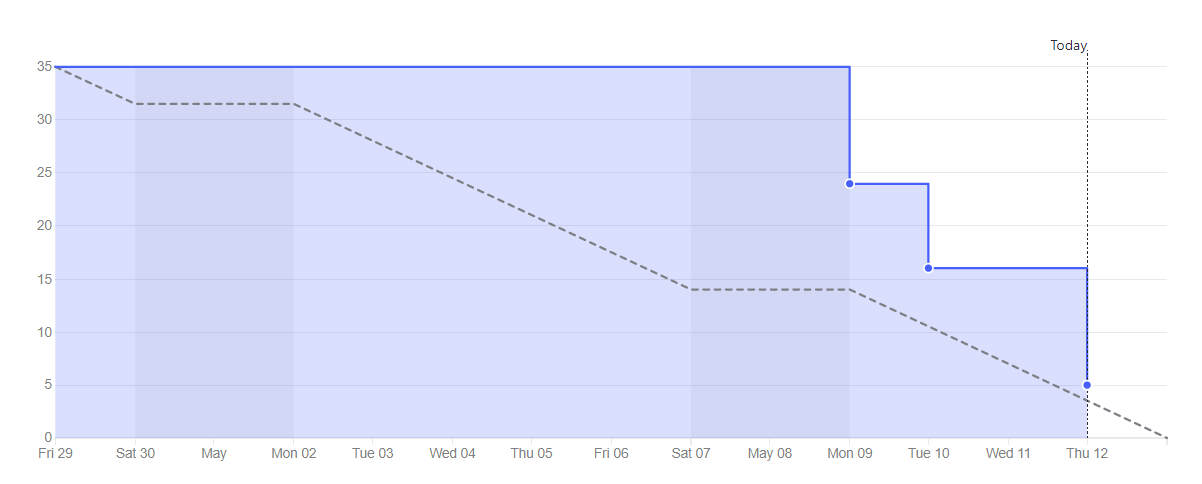


Ilustración 5 Tabla Burndown Sprint 4

### Sprint 5.

[Link.](https://github.com/albarrom/GII_O_MA_21.05/issues#workspaces/tfg-6229527890a395001341cd6b/reports/burndown?milestoneId=Z2lkOi8vcmFwdG9yL1NwcmludC8zMTgxNzQ)

**Fecha inicio y fin:** 13 mayo – 26 mayo

**Objetivo:** Crear gráficos definitivos.

**Story Points:**

**Issues:**

**Gráfico Burndown:**

Tabla Burndown Sprint 5

## A.3. Estudio de viabilidad.

### Viabilidad económica.

En este apartado se realizarán cálculos para estimar el coste real de este proyecto en su estado actual.

**Costes de personal.**

Contando el inicio de este proyecto con el mismo día que inició el cuatrimestre (9 de marzo de 2022) y el fin como el día de la entrega (9 de junio de 2022), este proyecto ha tenido una duración de 92 días/3 meses.

El desarrollo del proyecto lo ha llevado a cabo una sola persona, empleada a tiempo completo durante la duración de este.

|  |  |
| --- | --- |
| Concepto | Coste |
| Salario neto mensual | 1.200€ |
| Retención IRPF (10%) | 2.00€ |
| Seguridad social (30%) | 6.00€ |
| Salario bruto mensual | 2.000€ |
| Salario bruto trimestral | 6.000€ |

Tabla 1 Coste de personal

**Costes hardware.**

Todo el proyecto ha sido realizado con un único ordenador portátil. Este ha costado más de 300€, se considera un bien de inversión y puede amortizarse durante 4 años al 25%. Dado que este proyecto tan solo ha tenido 3 meses de duración, estos gastos no se van a añadir al estudio.

**Costes software.**

Su coste es virtualmente cero ya que todo el software usado para la generación de este proyecto tiene licencia de código abierto. Aprovechando los recursos que ofrece la UBU se ha podido acceder gratuitamente a software de pago (i.e. Microsoft Word o PyCharm).

**Costes Indirectos.**

En este apartado se detallarán costes que no están relacionados con el proyecto, pero que son necesarios para su realización. Estos serán gastos relacionados con electricidad e internet.

|  |  |
| --- | --- |
| Concepto | Coste (en euros) |
| Internet | 50€ |
| Electricidad | 15€ |
| Total | 65€ |

Tabla 2 Costes indirectos

**Coste total**

|  |  |
| --- | --- |
| Concepto | Coste (en euros) |
| Coste de personal | 2.000€ |
| Costes Software | 0€ |
| Coste Hardware | 0€ |
| Costes Indirectos | 65€ |
| Total (mensual) | 2.065€ |
| Total (trimestral) | 6.195€ |

Tabla 3 Coste total

La suma total del proyecto ascendería a 6.195€

### Viabilidad legal.

En esta sección se comentará brevemente la situación legal del proyecto, para ello se hará una breve mención de todas las licencias usadas en la realización del proyecto.

|  |  |  |
| --- | --- | --- |
| Nombre del componente | Descripción | Licencia. |
| Pandas | Librería especializada en análisis de datos | [BSD 3-Clause](https://opensource.org/licenses/BSD-3-Clause) |
| Numpy | Librería usada para cálculo científico | BSD 3-Clause |
| Plotly | librería capaz de generar gráficos interactivos | [MIT](https://opensource.org/licenses/MIT) |
| Dash | Framework que ayuda a construir aplicaciones web. | MIT |
| Dash Bootstrap Components | Librería con componentes Bootstrap compatible con Dash. | [Apache License 2.0](https://opensource.org/licenses/Apache-2.0) |
| Gunicorn | Servidor que permite ejecutar aplicaciones python. | MIT |

Tabla 4 Licencias usadas

Todas las licencias usadas son de código abierto y permisivas. Lo único a tener en cuenta es que, si se modifica el código fuente de la librería, se pierden los permisos de distribución.

La licencia usada por este proyecto será la más restrictiva de todas. En este caso, la BSD 3-Clause.

1. *Introducing ZenHub Sprints: Automated Sprint Planning in GitHub*. (2021, febrero 23). Agile Project Management Best Practices & Guides | ZenHub Blog. <https://blog.zenhub.com/introducing-zenhub-sprints/> [↑](#footnote-ref-1)