# 4. Técnicas y herramientas

Este apartado de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto.

## 4.1 Metodología

### 4.1.1 Scrum.

*Scrum* es un marco de desarrollo enfocado en crear software funcional en una estructura cíclica hasta finalizar el proyecto:

* Se selecciona uno o varios problemas que deban ser tratados y se dividen en tareas más asequibles que puedan ser realizadas en un marco temporal muy concreto, también denominados sprint.
* El equipo se reúne al finalizar un sprint para llevar un seguimiento de las tareas realizadas, ajustar expectativas y programar el siguiente *sprint*.

El principal objetivo de los ciclos es abordar problemas complejos, pudiendo tener así un mayor control sobre la versión final, teniendo la oportunidad de adaptar o cambiar el desarrollo de forma continua.

### 4.1.2 Kanban.

Es una metodología de desarrollo basado en observar de forma dinámica los cambios en el proyecto. *Kanban* es una palabra de origen japonés que significa “letrero”, término que resume muy bien su filosofía.

Las tareas se colocan en una pizarra o panel siguiendo 3 criterios: pendientes, en progreso o finalizadas. Los miembros del equipo son asignados diferentes tareas y es su responsabilidad cambiar la tarjeta de lugar conforme al estado de la tarea en particular.

En el proyecto se ha intentado seguir las mismas directrices, añadiendo una columna de revisión para aquellas *issues* que necesitaban pasar por un proceso previo antes de ser cerradas.

### 4.1.3 Técnica pomodoro.

Durante todo el desarrollo del proyecto se ha usado una interfaz web[[1]](#footnote-1) para seguir la técnica pomodoro. Su objetivo es mejorar la concentración y tener una administración del tiempo óptima. Para ello divide la carga de trabajo en periodos de 25 minutos de trabajo intenso - denominados *pomodoros* - seguidos por 5 minutos de descanso. Tras 4 pomodoros el tiempo de descanso será de 30 minutos.

## 4.2. Herramientas

### 4.2.1 Git- GitHub.

GitHub es un software de gestión de versiones. Cuenta con una aplicación de escritorio (GitHub Desktop) que simplifica considerablemente la gestión del código. En este proyecto se ha usado como método principal de gestión de repositorio.

Haber tenido contacto previo en otra asignatura del grado, su popularidad y la facilidad para encontrar documentación online hizo que no se consideran otras alternativas.

### 4.2.2 Zenhub.

Zenhub es un plugin de GitHub. Ayuda a gestionar el proyecto aplicando la metodología Kanban.

### 4.2.3 Python

Lenguaje de alto nivel. Se eligió Python por su versatilidad a la hora de tratar con grandes conjuntos de datos.

### 4.2.4 Jupyter Notebooks.

Jupyter es un IDE (Integrated Development Environment) basado en web que permite organizar el código en notebooks. Soporta Python y su característica más importante es su diseño basado en celdas que permite outputs de código, HTML, gráficos, imágenes o incluso LaTeX. El diseño modular hace posible crear flujos de trabajo con una alta personalización.

La flexibilidad que otorga Jupyter Notebooks y el hecho de tener amplia experiencia derivada de otras asignaturas en el grado fue lo que lo hizo destacar frente a otras alternativas como por ejemplo Google Colab. Este último está basado en un concepto muy similar al de Jupyter Notebooks.

Se descartó porque Jupyter tiene acceso local directo en lugar de almacenar el proyecto en Google Drive y tiene el beneficio de usar el hardware del equipo. Dado que esto va a ser un proyecto desarrollado por una sola persona, se prefirió Jupyter.

### 4.2.5 PyCharm.

IDE usado especialmente para Python. Requiere una licencia, pero usando la cuenta de la UBU se puede solicitar una licencia de estudiante[[2]](#footnote-2). Se incorporó en las fases más tardías del desarrollo para suplir las carencias de Jupyter Notebooks.

Se valoraron otras opciones como Visual Studio Code o, incluso, Sublime Text 3 que no es un IDE sino un simple editor de código. Sin embargo, PyCharm destacó por encima de cualquier otra opción por su compatibilidad con Jupyter Notebooks.

## 4.3 Librerías

### 4.3.1 Plotly

Librería gratuita y de código abierto caracterizada por su accesibilidad. Se ha hecho uso de multitud de herramientas que permiten visualizar grandes conjuntos de datos de forma visualmente atractiva para crear gráficos de forma sencilla.

### 4.3.2 Dash

Dash es un framework opensource de Python.

Una de las grandes ventajas de Dash es que se puede renderizar directamente en el navegador. Su aplicación original es crear aplicaciones web, en este proyecto se ha usado Bootstrap para poder generar el HTML.

Se barajó la posibilidad de usar Panel o Streamlit que fueron descartadas, principalmente porque ofrecían mucha menos personalización y documentación disponible que Dash.

### 4.3.3. Dash Bootstrap Components

Librería de components Boostrap para Dash que facilita enormemente la construcción de la página web.

### 4.4 Despliegue.

### 4.4.1 Heroku

Heroku permite hacer deploys de la web en la nube. A pesar de tener una versión gratuita, está muy limitada.

Su mayor ventaja frente a otras plataformas en la nube es su extensa documentación, simplicidad y logs de error. En etapas tempranas de desarrollo, especialmente en el primer deploy, estos mensajes ayudaron enormemente a encontrar errores.

Otras alternativas que fueron brevemente consideradas como Elastic Beanstalk (AWS) o Dokku antes de ser descartadas por la gran curva de aprendizaje que presentaban respecto a Heroku.

### 4.5 Documentación.

### 4.5.1 Zotero.

Gestor de referencias bibliográficas gratuito y de código abierto. Con un solo click permite almacenar cualquier referencia digital para su posterior uso usando un estilo de citado.

La versión de escritorio y pluggins son compatibles con la mayoría de sistemas operativos –Linux, Windows y MacOS– y navegadores web –Chrome, Firefox y Safari.

Mendeley es una alternativa muy similar a Zootero que se terminó descartando por su pobre importación de metadatos de videos de YouTube.

### 4.5.2. Microsoft Word.

Popular procesador de texto. LaTeX fue considerado como principal alternativa, pero se eligió Word por su rapidez de trabajo y sencillez.

1. ‘Pomofocus’ <https://pomofocus.io> [accessed 27 April 2022]. [↑](#footnote-ref-1)
2. ‘Free Educational Licenses - Community Support’, *JetBrains* <https://www.jetbrains.com/community/education/> [accessed 14 May 2022]. [↑](#footnote-ref-2)