

# Capítulo 2

## Objetivos

Una vez presentado el marco en el que se realiza este trabajo, en este capítulo se exponen los objetivos propuestos y los requisitos de este proyecto, además de la metodología de trabajo que se ha llevado a cabo para su resolución.

### 2.1. Objetivos del TFG

El objetivo principal de este trabajo es conseguir integrar nuevos robots reales en la plataforma Kibotics para poder programarlos desde ella.

En concreto, en este proyecto se pretende dar soporte a tres robots diferentes. Cada uno de estos robots presenta una serie de características propias que le hace ser interesante y distinto, motivo por el cual supone un gran enriquecimiento para esta plataforma.

- *Mbot*. Se ha elegido por ser uno de los más populares en el mundo de la Robótica Educativa, ya que tiene un diseño muy atractivo para los niños. Además, está basado en la placa *ArduinoUno*, una de las placas más famosas de código abierto y de la que hay mucha documentación y ejemplos de ejercicios que ayudan en el aprendizaje.
- *Tello*. Este dron ofrece muy buenas cualidades para aprender, fácil y económicamente, a programar este tipo de robots, los cuales han adquirido una gran popularidad en los últimos años, tanto a nivel profesional como para el entretenimiento.
- *GoPiGo3*. Este robot, cuya placa controladora es una Raspberry Pi, se ha convertido en un robot muy inteligente y versátil. Soporta visión; se le puede conectar una cámara vía USB o mediante un puerto dedicado especialmente a la *Pi Camera*. Esto le otorga una amplia gama de posibilidades para realizar programas relacionados con la visión artificial: detección de objetos, análisis de imagen, etc.

Gracias a estos robots, los alumnos no solo aprenden temas relacionados con la robótica, sino que también van a introducirse en el mundo de la informática y la electrónica.

Hasta el momento, Kibotics incluye algunos robots reales para poder programarlos, entre ellos

el ya mencionado Mbot. Sin embargo, este robot solo puede programarse utilizando un ordenador con Linux como sistema operativo. Además, el proceso de enviarle el programa no es directo, pues cuando el usuario quiere enviar el programa realizado al robot, es necesario descargar un ejecutable.

## 2.2. Requisitos

La integración en Kibotics de los citados robots se realizará, además, siguiendo los siguientes objetivos específicos.

- *Multiplataforma.* La programación de estos robots será soportada por los principales sistemas operativos (Linux, Windows y MacOS).
- *Sin instalación.* El proceso de envío del programa al robot debe ser lo más sencillo posible para evitar que el usuario tenga que hacer instalaciones o configuraciones adicionales. De esta manera conseguimos que las personas con pocos conocimientos en informática también puedan hacer uso de la plataforma.

Con todo lo anterior, se pretende dar continuidad al objetivo principal de Kibotics: acercar la robótica al público más general y a los más pequeños.

## 2.3. Metodología

Para el desarrollo de este trabajo se ha seguido el modelo de diseño iterativo (Figura 2.1), muy utilizado en el desarrollo de software. Se basa en la idea de entregar al cliente algo tangible cuanto antes para que pueda validarlo y, si es necesario, puedan hacerse los cambios de manera rápida sin tener que modificar todo el proyecto por completo. Además, también se deben hacer pequeñas iteraciones para evaluar las funcionalidades en los siguientes ciclos. Este diseño es una buena opción porque aporta cierta flexibilidad, algo muy útil en los procesos de desarrollo del software, los cuales suelen experimentar cambios frecuentes.

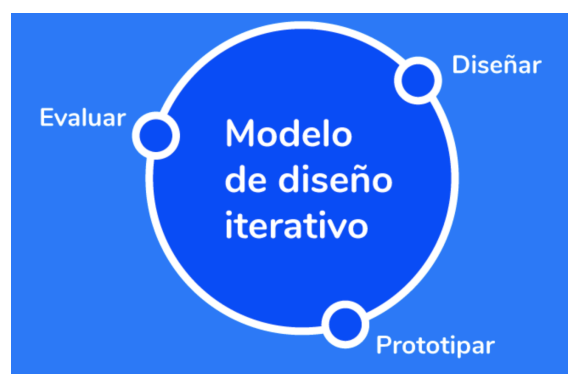


Figura 2.1: Modelo iterativo

La ventaja de usar esta metodología es que los tutores no tienen que esperar mucho tiempo hasta

ver resultados del proyecto, por lo que pueden revisar e ir dando sus opiniones y realimentaciones de forma rápida. De este modo, al hacer varias iteraciones, el riesgo ante un problema es pequeño y fácil de solucionar. Todo esto hace que el conocimiento sobre el producto sea progresivo y creciente.

Así, para el seguimiento de este trabajo, se ha utilizado la herramienta de control de versión **Git** y **GitHub**. Además, se planificaban reuniones semanales con los tutores del TFG y se creó un blog <sup>1</sup> donde se registraban las tareas realizadas a lo largo de la semana (Figura 2.2).

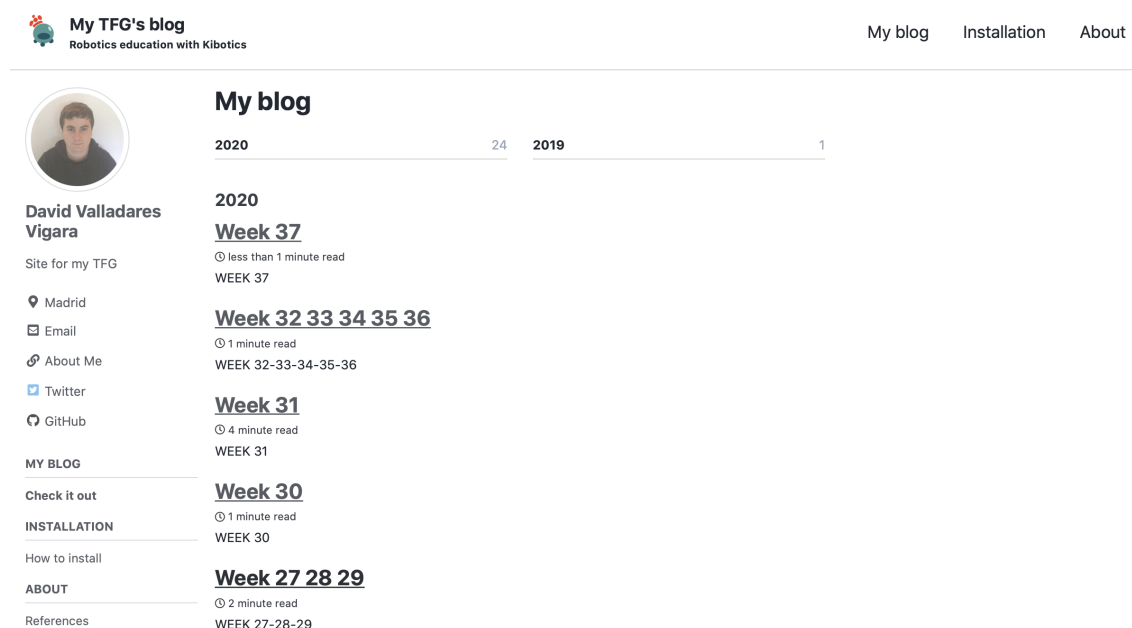


Figura 2.2: Blog para el seguimiento del TFG

## 2.4. Plan de trabajo

El plan de trabajo que se ha seguido durante la elaboración del proyecto puede dividirse en las siguientes etapas:

- **Aterrizaje en la plataforma de Kibotics.** Esta plataforma web está montada sobre un servidor usando la tecnología **Django**, por lo que se tuvo que aterrizar en cómo consistía el funcionamiento de esta aplicación.
- **Diseño para el Mbot.** En el inicio de esta fase fue fundamental entender el API de **Web Serial** y de las características del propio robot. Se hizo un primer prototipo de ejemplo para

<sup>1</sup><https://roboticslaburjc.github.io/2019tfg-david-valladares/>

ver la interacción con una placa **ArduinoUno**. Posteriormente, se realizó la integración en la plataforma y se perfiló la interfaz y la usabilidad del envío del programa al robot.

- ***Diseño para el dron Tello.*** Al igual que en la fase anterior, también se empezó estudiando las características del robot y del uso de **PyInstaller**. La primera integración que se diseñó fue para los ordenadores con **Linux** y, después, se procedió con el diseño de la integración para **MacOS** y **Windows**.
- ***Diseño para el GopiGo3.*** En este caso, además de conocer las características del robot, también fue necesario aprender a manejar la Raspberry Pi y entender la librería **Flask**. Se desarrolló un primer prototipo de ejemplo para encender un led en la placa **Arduino** vía web. Después, se realizó la integración en Kibotics y se mejoró la interfaz y usabilidad del envío del programa.