Capítulo 3

Herramientas

En este capítulo se describen las herramientas que han sido utilizadas para el desarrollo de este trabajo de fin de grado.

3.1. Python

`Python` es un lenguaje de programación diseñado por Guido Van Rossum y publicado en el año 1991 (Challenger, Díaz y Becerra, 2014). Es un proyecto de código abierto gestionado por Python Software Foundation, una sociedad sin ánimo de lucro. Actualmente se encuentra en su versión 3.8 (Zaforas, 2018).

Presenta algunas características que implican una serie de ventajas para los usuarios y que son las responsables de su gran popularidad. En primer lugar, se trata de un lenguaje multiparadigma, es decir, permite utilizar diferentes paradigmas de programación, como son la orientación a objetos, funcional o imperativa. Además, al ser un lenguaje interpretado, no necesita ser compilado ni enlazado. Posee un tipado dinámico (aunque desde la versión 3.5 puede hacerse uso del tipado estático), y es multiplataforma, lo que significa que puede ser ejecutado en diferentes sistemas operativos (Zaforas, 2018). La sintaxis de `Python` es más sencilla que la de otros lenguajes de programación, algo muy importante para la educación. Además, cuenta con una de las librerías más completas, comparable a la de `Java` y `.NET` (Challenger, Díaz y Becerra, 2014). Por todo esto es uno de los lenguajes de programación más utilizados en la actualidad.

3.2. HTML

`HTML` (*HyperText Markup Language*) fue desarrollado en 1991 por Tim Berners-Lee mientras trabajaba en la Organización Europea para la Investigación Nuclear (CERN), y popularizado por el navegador Mosaic desarrollado en NCSA (Raggett, Le Hors y Jacobs, 1999).

No es un lenguaje de programación, es decir, no tiene capacidad de crear una funcionalidad dinámica, sino que consiste en un lenguaje de marcado de hipertexto, basado en el metalenguaje SGML (*Standard Generalized Markup Language*). Este lenguaje da formato a los documentos de la WWW

(*World Wide Web*) (Lamarca, 2018). Se podría considerar un lenguaje entendido universalmente por todos los ordenadores que permite que la información publicada tenga una distribución global (Raggett, Le Hors y Jacobs, 1999).

La organización W3C (*World Wide Web Consortium*) es quien lleva a cabo las especificaciones relativas a este lenguaje. Está definido por "etiquetas" que el navegador interpreta y da forma en la pantalla. Estas etiquetas encierran un contenido y, todo en conjunto, forman lo que se denomina "elemento" (Lamarca, 2018).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mi pagina de prueba</title>
  </head>
  <body>
    <img src="images/firefox-icon.png" alt="Mi imagen de prueba">
  </body>
</html>
```

Fragmento 3.1: Ejemplo de código HTML

3.3. JavaScript

`JavaScript` es un lenguaje de programación desarrollado por Brendan Eich a finales de los años 90. Se considera un lenguaje interpretado, de tipado débil y dinámico, y se utiliza, principalmente, para crear páginas web dinámicas en el lado del cliente, es decir, aquellas que incluyen efectos, animaciones, ventanas emergentes, etc. (Eguíluz, 2009). Sin embargo, actualmente también existe la posibilidad de ejecutar `JavaScript` en todo tipo de desarrollo de aplicaciones (NodeJs).

Su nombre se debe a que los programas o aplicaciones creados con este lenguaje de programación se denominan "script" (Eguíluz, 2009). Además, su sintaxis es muy similar a la de otros lenguajes (`Java` y `C`)

En este trabajo `JavaScript` ha tomado un papel importante. Ha permitido, en el lado del cliente, dotar de dinamismo a las páginas web y también ha proporcionado una vía de comunicación, desde el navegador, con el servidor de Kibotics, el servidor utilizado en el GoPiGo3 y el robot Mbot.

3.4. Bootstrap

`Bootstrap` (Figura 3.1) fue diseñado por Mark Otto y Jacob Thornton para crear mejores herramientas internas en Twitter, aunque en agosto de 2011 pasó a ser de código abierto, lo que favoreció su desarrollo (Cumplido, 2018).

Consiste en un entorno utilizado para el desarrollo web basado en `CSS` (*Cascading Style Sheets*), un lenguaje que permite estructurar y componer páginas web, y `JQuery`, una librería de `JavaScript` utilizada para dotar de interactividad a una web. Utiliza un sistema de rejillas (cuadrículas) para el diseño de la web. Este entorno facilita la tarea de maquetar una página web, creando así una interfaz muy limpia y adaptable al tamaño de la pantalla del dispositivo. Es compatible con los principales navegadores (Firefox, Google Chrome, Safari), lo que supone una gran ventaja. Además, existe una gran cantidad de documentación disponible, lo que facilita su uso y comprensión (Guevara, s.f.).

Figura 3.1: Bootstrap

Este entorno ha sido utilizado para la maquetación de las páginas web de las unidades de los robots. Se han utilizado algunos de los elementos que Bootstrap ofrece como son los botones, paneles, etc.

3.5. Django

`Django` es un entorno de código abierto escrito en `Python`, creado en 2005, el cual permite desarrollar un entorno web complejo de una forma rápida y estructurada. Actualmente es mantenido por Django Software Foundation y se encuentra en la versión 3.0 (Novapros, 2020).

Sigue una arquitectura MVC (Modelo – Vista - Controlador), como se observa en la figura 3.2 (Holovaty y Kaplan-Moss, 2007) y se describe a continuación:

- *Modelo.* Los modelos de datos creados están mapeados directamente a las tablas de la base de datos, permitiendo aislar el código de la aplicación de la base de datos.
- *Vista.* Corresponde con la capa de presentación, y está basada en plantillas HTML.
- *Controlador (en `Django` llamado "views").* Responsable de seleccionar la plantilla a mostrar. Atiende a una petición y, según el mapeo de la URL, (*Uniform Resource Locator*) redirige a una vista u otra.

Ofrece una serie de características bastante interesantes como es su excelente capa de seguridad (p ej. permite una protección contra los ataques maliciosos "*Cross-site request forgery*"), dispone de un sistema de administrador "por defecto," sin necesidad de realizar ningún tipo de configuración. También proporciona una interfaz para el acceso a la base de datos, facilitando las consultas (Novapros, 2020).

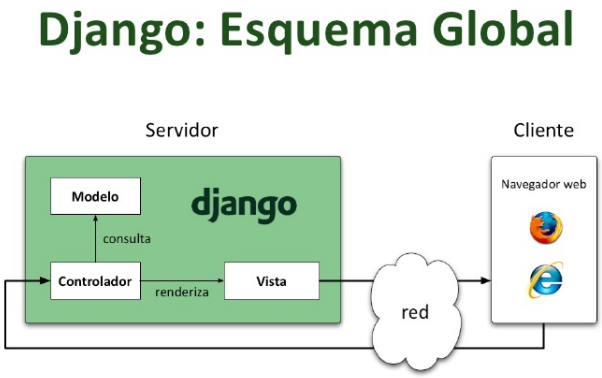

Figura 3.2: Arquitectura Django

3.6. Flask

`Flask` (Figura 3.3), lanzado en abril de 2010 es, junto con `Django`, uno de los entornos webs más famosos escritos en `Python`.

Está enfocado en proporcionar lo mínimo necesario para poner en funcionamiento una aplicación, por eso se le considera un entorno "minimalista". Por otro lado, no requiere de otras dependencias para realizar acciones básicas, de manera que la curva de aprendizaje para su comprensión es muy baja. Debido a su sencillez en la estructura, posee una velocidad mayor que `Django` y es bastante útil para iniciarse en el aprendizaje de desarrollo web, permitiendo crear servidores de una forma rápida. Debido a ello, `Flask` está orientado al sector servicios, los cuales implican muchas visitas y una carga grande de peticiones, y también para proyectos personalizados o sencillos (Rodríguez, 2019).

Figura 3.3: Arquitectura Django

En el robot GoPiGo3 se tendrá montado un servidor `Flask` que permitirá una comunicación HTTP

(*Hypertext Transfer Protocol*) con otros dispositivos.

3.7. Web Serial API

`Web Serial` es un API (*Application Programming Interface*) `JavaScript`, especificada por WICG (*Web Platform Incubator Community Group*), que permite interactuar con dispositivos conectados al puerto serie del ordenador desde un navegador web. Actualmente este API es soportada por los navegadores de escritorio basados en `Chromium` (Google Chrome, Opera, Edge), pero para poder utilizarla es necesario habilitarla.

Muchos dispositivos periféricos que se conectan al ordenador requieren de una descarga software para controlarlo. Un claro ejemplo de esto se ve en la robótica; en la mayoría de los robots en los que la carga del programa se realiza por USB (*Universal Serial Bus*), es necesario instalar previamente en el ordenador un software que permita hacerlo. La ventaja de `Web Serial` es que proporciona una vía capaz de comunicarse con estos dispositivos sin necesidad de instalar nada en el ordenador (W3C Community Group, 2020).

Gracias a las características que presenta el Mbot, puede hacerse uso de `Web Serial API` para abrir una comunicación entre el navegador y el robot conectado al USB de la computadora.

3.8. PyInstaller

`PyInstaller` es un módulo lanzado por la línea de comandos que permite agrupar aplicaciones `Python` en un único paquete junto con todas las dependencias necesarias y el intérprete de `Python`, permitiendo crear un ejecutable para ser distribuido. Es compatible con las versiones de `Python2.7` y para las versiones superiores a `Python3.3`. Puede ser utilizado para `Linux`, `Windows` y `MacOS`, aunque no permite una compilación cruzada, es decir, la aplicación empaquetada solo puede utilizarse en el mismo sistema operativo en el que fue empaquetado, por lo que una aplicación empaquetada en `Linux` no podrá ser utiliza en una computadora con `Windows` o `MacOS` (SO Documentation, s.f.).

```
$ pyinstaller --onefile hello.py
```

Fragmento 3.2: Ejemplo para generar un ejecutable con PyInstaller

3.9. mCore

`mCore` (Figura 3.4) es una placa basada en `ArduinoUno`, una microcontroladora de código abierto que utiliza un microchip ATmega328P, pero extendida con más componentes y que, además, incorpora la electrónica de potencia para el manejo de los motores. El robot Mbot incorpora esta placa como controladora.

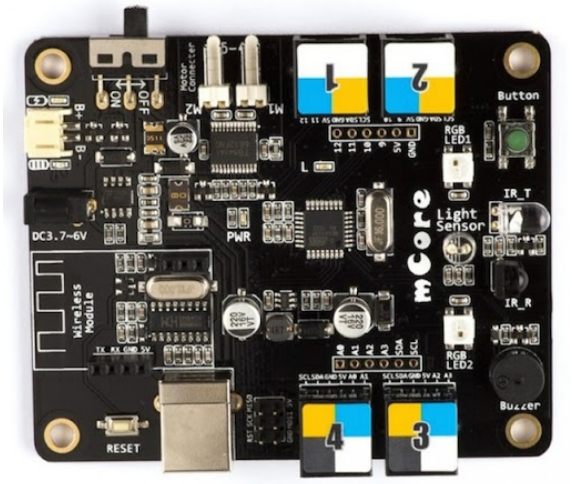

Figura 3.4: Placa mCore

La placa permite ser programada a través del puerto serie gracias al bootloader, un software alojado en la memoria flash. En el arranque de la placa, el bootloader comprueba si se está intentado programarla y, en caso afirmativo, se procede a grabar el programa en la memoria y se reinicia. En el caso contrario, el bootloader ejecuta el último programa que estuviera grabado. Estas placas pueden ser programadas utilizando el lenguaje `Arduino`, un lenguaje de alto nivel que está basado en el lenguaje `C++`, pero adaptado para facilitar la programación de los pines de entrada y salida (Llamas, 2016).

Una vez que el programa ha sido escrito en este lenguaje de alto nivel, es necesario compilarlo para obtener el código ejecutable entendido por la placa para el proceso de carga (Figura 3.5).

3.10. Raspberry Pi

`Raspberry Pi` es un proyecto que nace en 2009 en Londres bajo la asociación caritativa denominada Fundación Raspberry Pi, cuyo objetivo era mejorar la enseñanza de la informática en las aulas.

Se trata de una placa computadora de bajo coste que puede considerarse como un ordenador de tamaño reducido. Su *hardware* se compone, principalmente, de una CPU (*Central Processing Unit*), una GPU (*Graphics Processing Unit*), una memoria RAM (*Random Access Memory*), una ranura SD (*Secure Digital*) utilizada para el almacenamiento, salidas y entradas de audio y vídeo, pines de entrada y salida de propósito general (GPIO), varios buses USB, tarjeta de red y alimentación. En cuanto a su software, existen una gran cantidad de sistemas operativos que pueden usarse, tales como `Raspbian`, `Kali Linux`, `Windows 10 IoT Core`, `Ubuntu Core`, etc. (Escuela Técnica Superior de Ingeniería Informática de la Universidad Politécnica de Valencia, 2013).

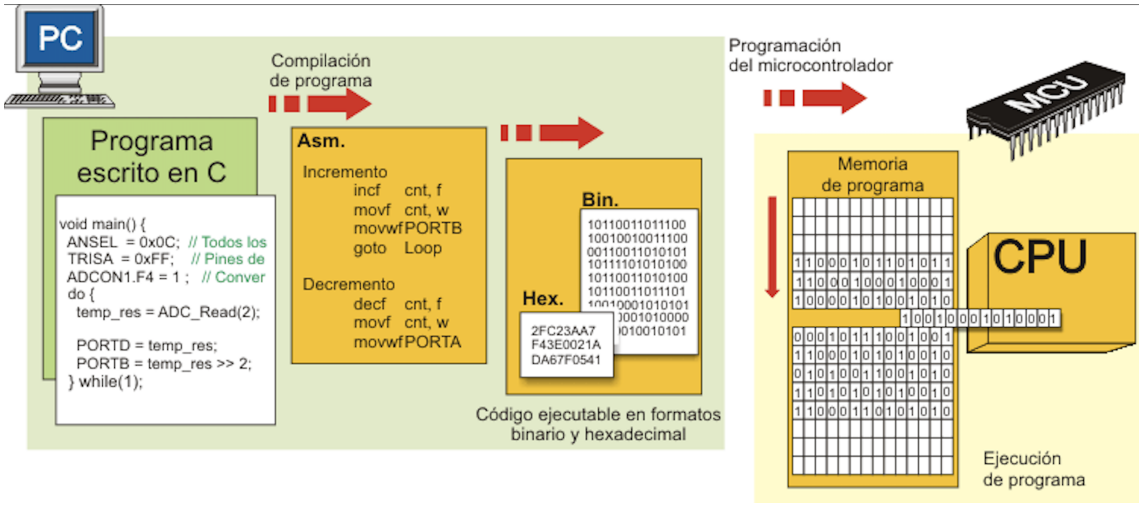

Figura 3.5: Proceso de carga de un programa a la placa mCore

Este ordenador es muy utilizado en la actualidad gracias a la gran variedad de usos que puede ofrecer, ya que puede emplearse como un ordenador de sobremesa, utilizarse en Robótica, como dispositivo controlador de un sistema domótico, etc.

Para este TFG, hemos usado la `Raspberry Pi` 3 modelo B, la cual forma parte de la controladora del robot GoPiGo3. En la figura 3.6 se muestran sus características principales.

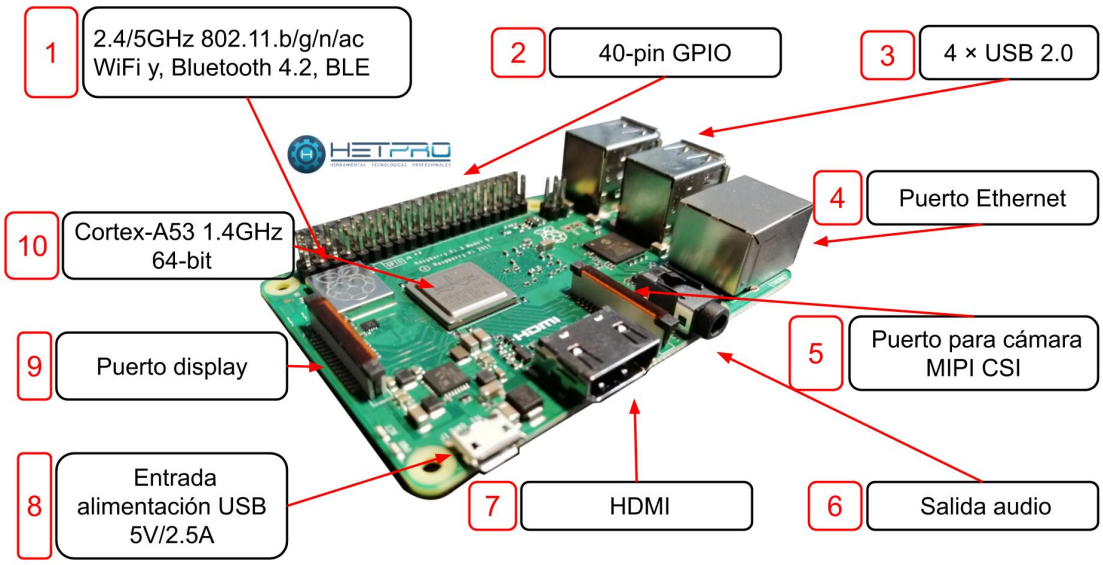

Figura 3.6: Características hardware para la Raspberry Pi modelo 3B