



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE
TELECOMUNICACIÓN

GRADO EN INGENIERÍA EN SISTEMAS AUDIOVISUALES
Y MULTIMEDIA

TRABAJO FIN DE GRADO

***Gamificación de una plataforma web de robótica
educativa***

Autora : Marta Quintana Portales

Tutor : Dr. José María Cañas Plaza

Curso Académico 2020/2021

Agradecimientos

A mi familia, amigas, tutor, compañeros y a mí.

Quiero empezar este trabajo dando las gracias a todas las personas que me han acompañado a lo largo de esta aventura. Especialmente a mi familia, no puedo estar más orgullosa, si lo he conseguido ha sido gracias a ellos, sobretodo a mi madre, a mi padre y a mi hermana que siempre me apoyan y sacan lo mejor de mí.

Gracias a Jose María por las oportunidades y la confianza en mí para la realización de este TFG. También a todo el equipo de *Kibotics* en especial a Pablo, David, Roberto y Sergio y a la asociación *JdeRobot*, gracias por hacerlo un poco más fácil a pesar de la situación que estamos viviendo por la pandemia.

También quiero agradecer a mis amigas por estar ahí siempre, a ese grupo que tenemos 'No me da la vida' que refleja perfectamente lo que hemos vivido estos años en la universidad, gracias por apoyarme y colaborar en la realización de este trabajo y no me puedo olvidar a mis compañeros y compañeras de la carrera que han hecho que estos años sean más llevaderos y llenos de momentos para recordar.

Muchas gracias a todos por formar parte de mi vida.

No ha sido fácil pero nada que valga la pena lo será.

Resumen

Este Trabajo de Fin de Grado describe el desarrollo llevado a cabo para ...

Palabras clave: Gamificación, Tecnologías Web, A-Frame, JavaScript...

Índice general

Índice de figuras	4
1. Introducción	1
1.1. Robótica	1
1.2. Tecnologías Web	5
1.2.1. Tecnologías Web lado cliente	7
1.2.2. Tecnologías Web lado servidor	8
1.3. Robótica educativa	9
1.3.1. Importancia de los juegos y multimedia en el aprendizaje	12
1.3.2. Gamificación en el aprendizaje	13
1.3.3. Campeonatos de robótica educativa existentes	13
1.4. Estructura del documento	15
2. Objetivos y Metodología del Trabajo	16
2.1. Objetivos	16
2.2. Requisitos	17
2.3. Metodología	17
2.4. Plan de Trabajo	18
3. Infraestructura utilizada	20
3.1. Lenguajes de programación y de documentos	20
3.1.1. HTML5	20
3.1.2. JavaScript	21
3.1.3. Python y Scratch	22
3.1.4. JSON	24

3.2. Herramientas	25
3.2.1. TensorFlowJS, Web Audio API y Teachable Machine	25
3.2.2. A-Frame	26
3.2.3. Blender	28
3.2.4. Plataforma Kibotics	29
3.2.5. Navegadores web utilizados	30
4. Ejercicio teleoperador acústico y banda sonora	31
5. Ejercicio aspiradora robótica atrapa confeti	32
5.1. Enunciado	32
5.2. Desarrollo del ejercicio	32
5.3. Solución de referencia	36
6. Ejercicio juego del pañuelo	37
6.1. Enunciado	37
6.2. Desarrollo del ejercicio	38
6.3. Solución de referencia	39
7. Conclusiones y trabajos futuros	40
Bibliografía	41

Índice de figuras

1.1.	Sensor ultrasonidos	2
1.2.	Procesador	2
1.3.	Motor	2
1.4.	Pantalla	2
1.5.	Ejemplos de Robots	3
1.6.	Ejemplos de Robots educativos	4
1.7.	Ejemplos aplicaciones web	5
1.8.	Comunicación cliente/servidor a través de HTTP	6
1.9.	Ejemplo petición HTTP Cliente a Servidor	6
1.10.	Ejemplo respuesta HTTP Servidor a Cliente	6
1.11.	Bases de Datos	8
1.12.	Scratch	9
1.13.	Open Roberta	10
1.14.	LEGO EDUCATION	11
1.15.	MBlock IDE	11
1.16.	Plataforma Kibotics.	12
1.17.	Campeonatos de robótica.	14
2.1.	Modelo iterativo	17
2.2.	Página web de este TFG	18
3.1.	Ejemplo de página web con HTML5	21
3.2.	Ejemplo de página web interactiva con HTML5, CSS3 y JavaScript	22
3.3.	Ejemplo de código en Kibotics	23
3.4.	Parte de un fichero config.json de un ejercicio en Kibotics	25

3.5. Herramientas reconocimiento de audio	26
3.6. Ejemplo escena en A-Frame	27
3.7. Inspector en A-Frame	28
3.8. Blender	29
3.9. Ejercicios Kibotics	30

Capítulo 1

Introducción

Este Trabajo Fin de Grado se enmarca en el ámbito educativo, en concreto en la *Gamificación* de una plataforma web de robótica educativa. Según avanza la ciencia y la tecnología la sociedad ha cambiado y las formas de enseñar también. Este proyecto tiene como objetivo crear nuevos juegos y ejercicios que permitan enseñar a programar a los más jóvenes de una forma más divertida para fomentar el aprendizaje y la motivación, así como mejorar sus habilidades. Este proyecto engloba numerosas tecnologías web y herramientas que han servido para crear nuevos ejercicios para la plataforma de robótica educativa *Kibotics* [1].

En este capítulo se hace una breve introducción del uso actual de la Robótica y las tecnologías web, conceptos clave para ponernos en contexto y enfatizar en el tema principal de este trabajo: la “*Gamificación*”.

1.1. Robótica

La Robótica es una rama de la ingeniería que combina las matemáticas, la electrónica, la mecánica, la informática y la física. Gracias a la unión de estas ramas de conocimiento se han podido desarrollar sistemas electromecánicos llamados robots.

Un robot es una máquina autónoma o semi-autónoma que es capaz de percibir su entorno, realizar cálculos para tomar decisiones, actuar en el mundo real de acuerdo con esas decisiones y comunicarse con otras máquinas o con humanos. Un robot se compone principalmente de 4 elementos:

- Sensores: para recibir información de su entorno (láser, lídar, ultrasonidos (Figura 1.1),

1.1. ROBÓTICA

infrarrojos, cámaras, radio frecuencia, sensores térmicos...).

- Actuadores: permiten la locomoción del robot por su entorno y manipulación de objetos (motores eléctricos, de combustión (Figura 1.2), amortiguadores, hélices, patas, ruedas, cadenas...).
- Controladores: Procesador (Figura 1.3) y algoritmos, para el cálculo y toma de decisiones.
- Leds, pantallas (Figura 1.4), antenas, altavoces...: para comunicarse con otros robots o con humanos.



Figura 1.1: Sensor ultrasonidos



Figura 1.2: Procesador



Figura 1.3: Motor



Figura 1.4: Pantalla

Los robots se clasifican según su entorno de aplicación en robots industriales y robots de servicio. También se pueden diferenciar según su forma en androides y zoomórficos, según su capacidad de movimiento en fijos o móviles, o según el medio en el que trabajan en terrestres, acuáticos o áereos. Estas son algunas de sus aplicaciones:

- Robots industriales: brazos y pinzas robóticas para ensamblado de piezas, envasado de alimentos, industria automovilística y gestión de almacenes.

1.1. ROBÓTICA

- Robots de servicio: destinados a limpieza del hogar, asistencia, coches autónomos, entretenimiento, usos militares, limpieza de centrales nucleares, investigación en terrenos hostiles o fines médicos.

En la Figura 1.5 se muestran algunos ejemplos de robots que son de gran utilidad en la sociedad.



(a) Dron



(b) Perseverance Mars



(c) Nao



(d) Brazo biónico



(e) Da Vinci [4]



(f) Roomba

Figura 1.5: Ejemplos de Robots

La Figura 1.5(a) muestra un dron, estos pequeños vehículos no tripulados han sido toda una revolución en la grabación de eventos y películas. El entretenimiento no es su única aplicación, también se utilizan para la búsqueda de personas y vigilancia.

El Perseverance Mars que vemos en la Figura 1.5(b) actualmente esta en Marte haciendo investigaciones del terreno. Este robot busca signos de vida y guarda muestras para un futuro regreso a la Tierra.

Los robots de asistencia como el Nao Figura 1.5 (c), son muy importantes para que los más pequeños y las personas mayores puedan socializar de una forma divertida y a su vez es una

1.1. ROBÓTICA

herramienta de apoyo en procesos de rehabilitación, post-operatorios y terapias ocupacionales.

Los brazos biónicos Figura 1.5(d) permiten recuperar las funciones y el tacto a amputados. El robot Da Vinci Figura 1.5(e) ayuda a los cirujanos a operar con mayor precisión y seguridad. Estos robots han sido un gran avance en el campo de la medicina.

El robot aspirador o más conocido como *Roomba* Figura 1.5(f) es capaz de detectar obstáculos y residuos en el suelo, esto nos ayuda a que la casa esté limpia y nos ahorra mucho tiempo.

Los robots que se han mencionado son solo algunos ejemplos. La robótica es una tecnología en auge, en los últimos años con el avance de la tecnología la realidad ha superado a la ficción, estamos rodeados de robots. Los robots han salido de los laboratorios y han surgido miles de aplicaciones que nos hacen la vida un poco mejor.

Recientemente la robótica se está incorporando a las aulas para que los más pequeños adquieran conocimientos y habilidades importantes para su futuro. En la siguiente Figura 1.6 se muestran unos ejemplos de robots diseñados para el ámbito educativo. [5]



(a) Bee-bot



(b) Makeblock Mbot [6]



(c) LEGO MINDSTORMS EV3

Figura 1.6: Ejemplos de Robots educativos

1.2. Tecnologías Web

Las tecnologías web juegan un papel muy importante en el mundo moderno gracias a Internet. Esta plataforma WWW¹[7] ha ido evolucionando y ha posibilitado potentes aplicaciones con un modelo cliente/servidor. En la Figura 1.7 podemos ver algunos ejemplos de aplicaciones web.

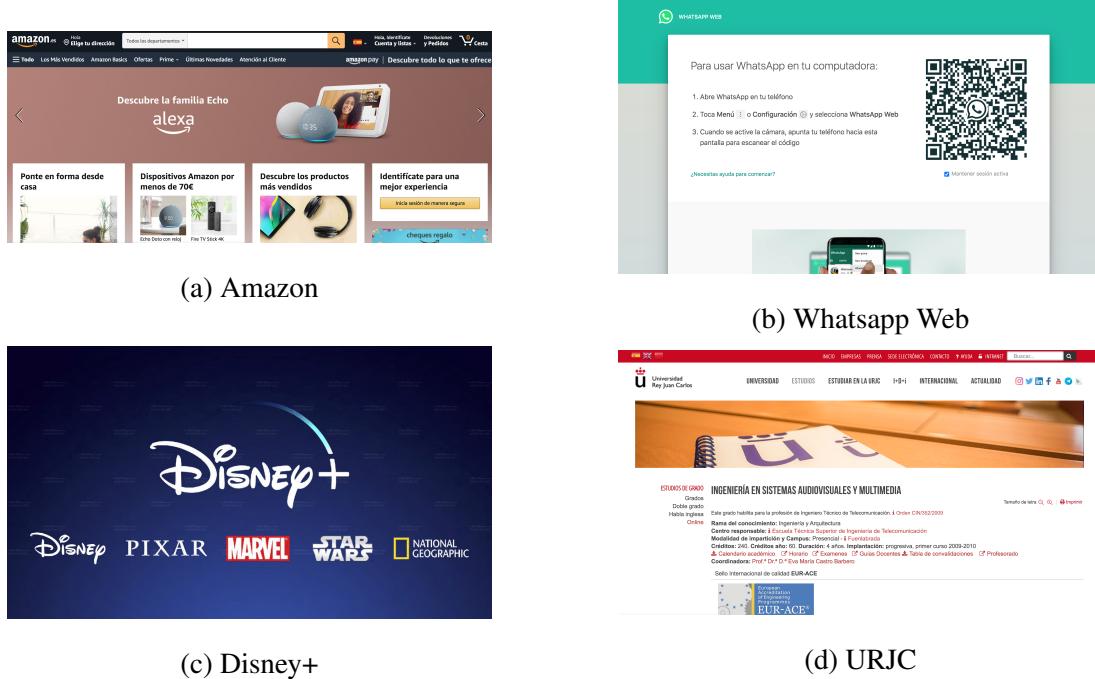


Figura 1.7: Ejemplos aplicaciones web

La web es una colección de documentos enlazados a través de hiperenlaces, cada recurso queda definido por su URL². Cuando accedemos a la web a través del navegador, tenemos que introducir la dirección URL del sitio web al que nos queremos dirigir. El navegador enviará una solicitud al servidor con el protocolo HTTP³. El servidor le enviará a nuestro navegador un fichero html que quedará almacenado en nuestra máquina. Una vez el navegador obtiene el fichero html mostrará al usuario la página web principal de la URL que ha introducido. Si el navegador detecta que hay imágenes, videos u otros ficheros, volverá a mandar peticiones HTTP al servidor para que este le envie toda la información necesaria. En la Figura 1.8 podemos ver una representación de cómo es la comunicación entre cliente y servidor.

¹World Wide Web

²Uniform Resource Locator

³Hyper Text Transfer Protocol

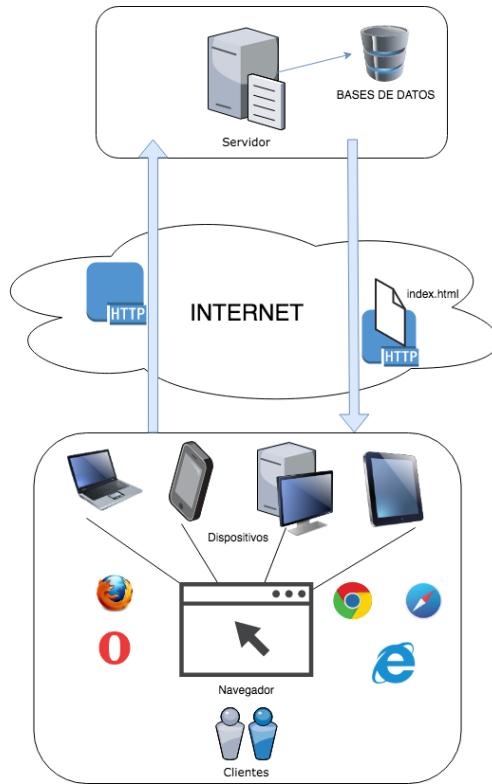


Figura 1.8: Comunicación cliente/servidor a través de HTTP

HTTP es un protocolo entre navegadores y servidores web para transferir documentos de hipertexto. El cliente envía mensajes de solicitud y el servidor manda mensajes de respuesta, ambos mensajes son del mismo formato (ver Figura 1.9 y Figura 1.10.) Los tipos de mensaje más comunes de este protocolo son GET, POST, PUT, DELETE y HEAD. Este protocolo utiliza códigos de estado, los más conocidos son: 200 OK, que significa resultado exitoso, 500 Server Error, cuando hay un error en el lado servidor y el más conocido 404 Not Found, cuando hay un error en la parte cliente. [8]

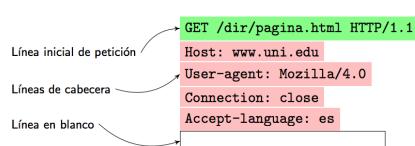


Figura 1.9: Ejemplo petición HTTP
Cliente a Servidor

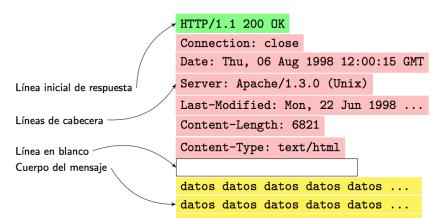


Figura 1.10: Ejemplo respuesta HTTP
Servidor a Cliente

1.2. TECNOLOGÍAS WEB

Las dos partes que forman una aplicación web son independientes entre sí. La tecnologías del lado cliente (*frontend*) se encargan de interactuar con el usuario, visualizar el contenido y establecer la comunicación con el servidor. Se ejecutan en el navegador, que actúa como intérprete. Por otro lado, las tecnologías del lado servidor (*backend*) se encargan de la administración del sitio web, usando bases de datos y gestores de contenidos.

Una de las principales ventajas de usar tecnologías web es que las aplicaciones creadas son multiplataforma y multidispositivo, funcionan tanto en ordenadores, móviles, tabletas, así como en distintos sistemas operativos. Otra ventaja es que no tenemos que instalar nada solo necesitamos el navegador y además la actualización del contenido es inmediata. El principal inconveniente es su dependencia de Internet pero con los últimos avances tecnológicos el Wifi, la fibra óptica y el 5G han permitido que la mayoría de personas del mundo podamos acceder desde cualquier lugar y este no sea un gran inconveniente.

1.2.1. Tecnologías Web lado cliente

Las tecnologías web del lado cliente permiten la interacción del usuario con la página web que corre en el navegador del usuario. Para ello, se usan principalmente estas tres tecnologías [9]:

- HTML5⁴: es un lenguaje de marcado de los contenidos de un sitio web, se usa para asignar la función de cada elemento. Es el esqueleto de la web.
- JavaScript: es un lenguaje de programación interpretado que se encarga del comportamiento de una página web y de su interactividad con el usuario.
- CSS3⁵: es un lenguaje de hojas de estilo creado para controlar la presentación de la página: colores, tipo de letra, tamaños, animaciones, colocación de los elementos...

Entraremos en más detalle en estas tecnologías en el capítulo 3 donde se habla de la Infraestructura utilizada.

⁴HyperText Markup Language

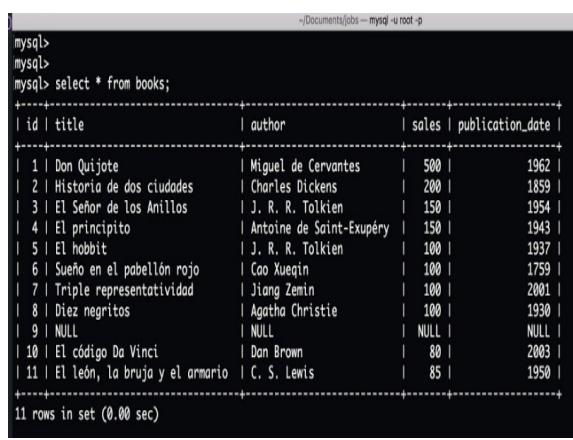
⁵Cascading Style Sheets

1.2.2. Tecnologías Web lado servidor

Las tecnologías web del lado servidor son las que permiten gestionar y servir las páginas web y acceder a bases de datos. En este caso las tecnologías son más flexibles y vamos a nombrar tres de las más utilizadas:

- Django: es un entorno para crear servidores web de alto nivel, que fomenta el desarrollo rápido con un diseño limpio y práctico, destaca por su arquitectura basada en modelo-vista-controlador y el uso de plantillas. De esta forma puedes centrarte en crear tu aplicación web sin grandes complicaciones. Es gratis y de código abierto[10]. Un ejemplo de aplicación web que utiliza Django es Instagram [11].
- Node.js: es un entorno de ejecución para JavaScript orientado a eventos síncronos, construido con el motor de JavaScript V8 de Chrome. Diseñado para aplicaciones web escalables. De esta forma el cliente y el servidor pueden crearse con el mismo lenguaje de programación[12]. Netflix, Paypal o LinkedIn usan esta tecnología para sus servidores[13].
- PHP⁶: es un lenguaje de scripting de uso general popular que es especialmente adecuado para el desarrollo web[14]. Rápido, flexible y pragmático, gracias a su capacidad de creación de webs dinámicas, desde blogs hasta sitios web como Facebook o Wikipedia[15].

En el lado servidor se utilizan bases de datos. Una base de datos es una colección de datos estructurados. Entre ellas podemos destacar mySQL(Figura 1.11a) y MongoDB(Figura 1.11 b).



```
mysql>
mysql>
mysql> select * from books;
+----+-----+-----+-----+
| id | title          | author        | sales | publication_date |
+----+-----+-----+-----+
| 1  | Don Quijote    | Miguel de Cervantes | 500  | 1962 |
| 2  | Historia de dos ciudades | Charles Dickens | 200  | 1859 |
| 3  | El Señor de los Anillos | J. R. R. Tolkien | 150  | 1954 |
| 4  | El principito   | Antoine de Saint-Exupéry | 150  | 1943 |
| 5  | El hobbit       | J. R. R. Tolkien | 100  | 1937 |
| 6  | Sueño en el pabellón rojo | Cao Xueqin | 100  | 1759 |
| 7  | Triple representatividad | Jiang Zemin | 100  | 2001 |
| 8  | Diez negritos   | Agatha Christie | 100  | 1930 |
| 9  | NULL            | NULL           | NULL  | NULL  |
| 10 | El código Da Vinci | Dan Brown | 80   | 2003 |
| 11 | El león, la bruja y el armario | C. S. Lewis | 85   | 1950 |
+----+-----+-----+-----+
11 rows in set (0.00 sec)
```

(a) mySQL
(b) MongoDB

```
{
  title: "Java in action",
  author: "author1",
  language: "English",
  publisher: {
    name: "My publications",
    founded: 1990,
    location: "SF"
  }
}

{
  title: "Hibernate in action",
  author: "author2",
  language: "English",
  publisher: {
    name: "My publications",
    founded: 1990,
    location: "SF"
  }
}
```

Figura 1.11: Bases de Datos

⁶Hypertext Preprocessor

1.3. Robótica educativa

La Robótica educativa es una disciplina de aprendizaje multidisciplinar. Ayuda a desarrollar competencias y habilidades como: la innovación y espíritu emprendedor, la resolución de problemas y lógica, la toma de decisiones, conocimientos de herramientas relacionadas con las tecnologías digitales, el pensamiento crítico, creatividad, el trabajo colaborativo y cooperativo, la flexibilidad y adaptabilidad al trabajo [21].

Ante la falta de estudiantes en carreras técnicas en la actualidad, la robótica educativa puede ofrecer una gran motivación a los alumnos de las primeras etapas de educación: Primaria, ESO y Bachillerato, para fomentar la creatividad y la curiosidad al mostrar la ciencia y la tecnología de una forma diferente, incrementan sus habilidades a la vez que sus conocimientos desde los fundamentos STEM (*Science, Technology, Engineering and Mathematics*).

Gracias a las tecnologías web son muchas las aplicaciones que ofrecen cursos de robótica para todos los niveles educativos, muchos ayuntamientos están comprando cursos y materiales para facilitar a los más pequeños su introducción al mundo de la robótica a través de clases extraescolares. En secundaria, se está introduciendo poco a poco en las asignaturas de tecnología el uso de lenguajes de programación, en el que destaca el lenguaje Scratch. En la Figura 1.12 podemos ver la interfaz de Scratch para programar desde el navegador.

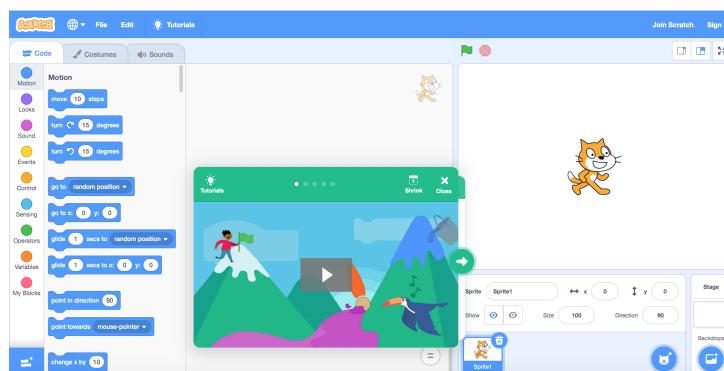


Figura 1.12: Scratch

Scratch es un lenguaje de programación visual basado en bloques, creado y mantenido por Lifelong Kindergarten group en el MIT Media Lab. Scratch además es una comunidad en línea donde los niños pueden programar y compartir medios interactivos como historias, juegos y animaciones con gente de todo el mundo. Los más pequeños pueden aprender a pensar creativamente, trabajar en colaboración y razonar sistemáticamente[22]. Scratch posee un lenguaje de

1.3. ROBÓTICA EDUCATIVA

iniciación llamado Scratch Jr pensando para niños de 5 a 7 años siendo aún más sencillo, aunque Scratch está pensado para todas las edades. Actualmente se puede utilizar desde cualquier dispositivo al utilizar tecnologías web.

Junto con Scratch cada vez hay más plataformas y entornos STEM que se han dedicado al desarrollo de herramientas de aprendizaje enfocadas a los más pequeños. Destacan aplicaciones web como:

- *OpenRoberta*⁷: es una plataforma web creada por un instituto alemán perteneciente a la Fraunhofer Society. Tiene como objetivo simplificar conceptos de programación y facilitar a niños y profesores la codificación mediante el uso de robots como Lego Mindstorms y otros sistemas de hardware programables. [23].

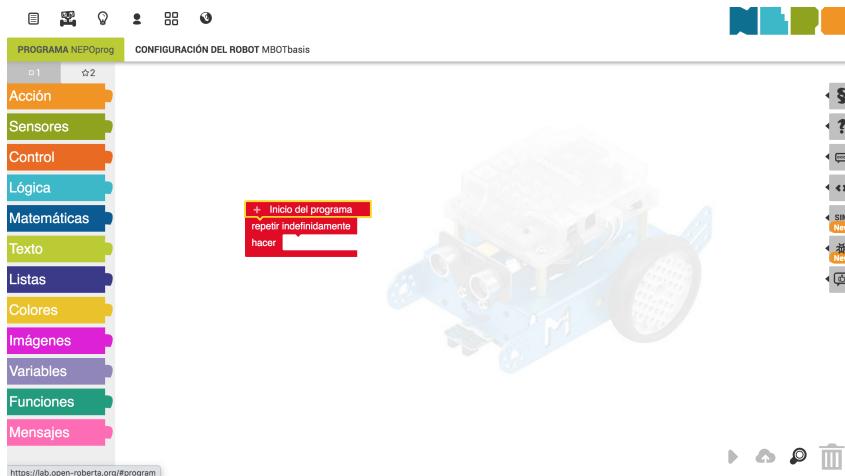
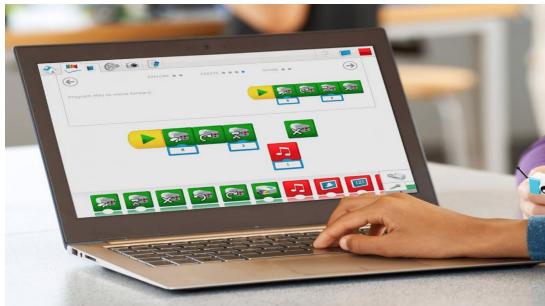


Figura 1.13: Open Roberta

- *LEGO Education*: la plataforma LEGO ofrece una amplia variedad de robots y packs para uso escolar. Sus kits de robótica educativa permiten a los más pequeños construir y programar robots mediante el uso de motores, sensores, engranajes, ruedas, ejes y otros componentes técnicos, además del uso de su propio software basado en bloques. Destacan modelos como MINDSTORMS Education EV3 y LEGO Education WeDo 2.0.[24] [25]

⁷<https://lab.open-roberta.org/>

1.3. ROBÓTICA EDUCATIVA



(a) LEGO



(b) WeDo 2.0

Figura 1.14: LEGO EDUCATION

- **MBlock IDE**⁸: esta aplicación web está diseñada para la educación en ciencia, tecnología, ingeniería, artes y matemáticas (STEAM). Está inspirada en Scratch 3.0, es compatible con lenguajes de programación tanto gráficos (Scratch) como textuales (Python). Se pueden diseñar historias, juegos, animaciones y programar dispositivos como robots Makeblock y microbit. [26]

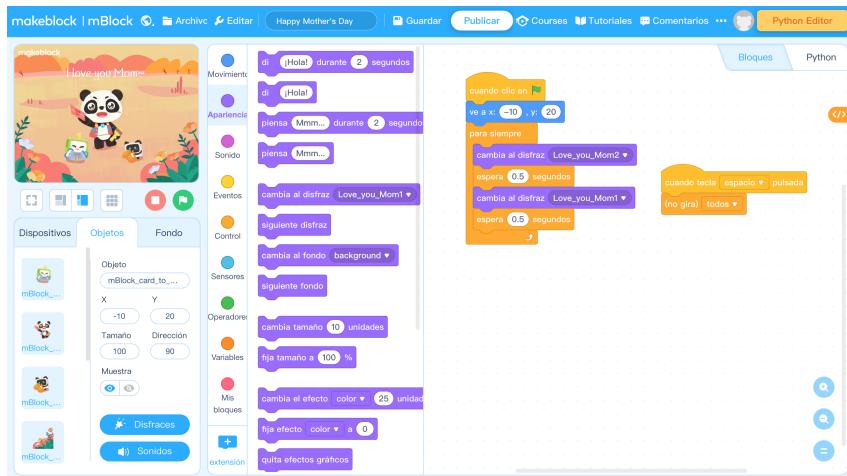


Figura 1.15: MBlock IDE

- Vex⁹, AppInventor¹⁰, Arduino Web Editor¹¹, Kodu¹² o Snap!¹³ son otras plataformas web de programación en robótica educativa.

⁸<https://ide.mblock.cc/>

⁹<https://www.vexrobotics.com/>

¹⁰<https://appinventor.mit.edu/>

¹¹<https://store.arduino.cc/digital/create>

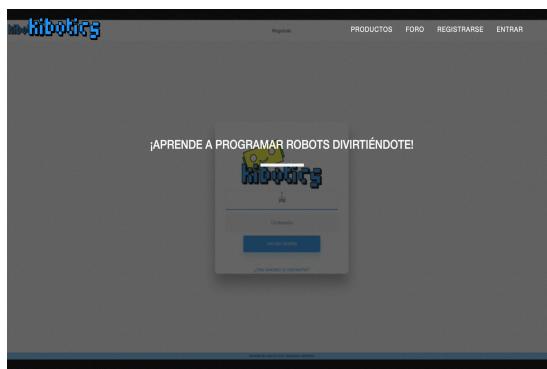
¹²<http://www.kodugamelab.com/>

¹³<https://snap.berkeley.edu/>

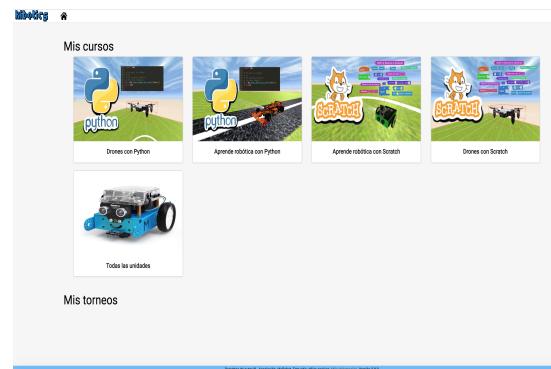
1.3. ROBÓTICA EDUCATIVA

La plataforma de robótica educativa que vamos a utilizar en este TFG es Kibotics. Esta plataforma basada en tecnologías web, enseña de manera atractiva conceptos básicos de tecnología e inicia a niños y adolescentes en robótica y programación de robots. Sigue un enfoque práctico, fomentando el pensamiento y la organización para resolver un problema. Además de formar el espíritu analítico de los alumnos [1].

Actualmente ofrecen cursos de robótica en Scratch y Python (Figura 1.16). En el capítulo 3 se ampliará información sobre Kibotics y veremos cómo se introduce la *Gamificación* en la plataforma a lo largo de este trabajo.



(a) kibotics.org



(b) Cursos disponibles en Kibotics.

Figura 1.16: Plataforma Kibotics.

1.3.1. Importancia de los juegos y multimedia en el aprendizaje

El planteamiento actual del sistema educativo tiene carencias. Es por eso que la innovación en la educación cobra cada vez más importancia. Es por ello, que los docentes buscan actividades enfocadas en conseguir evitar la recurrente pasividad de los alumnos[16].

Según el cono del aprendizaje de Edgar Dale y otros estudios concluyen que aprendemos un 10 % de lo que leemos, 20 % de lo que escuchamos, 75 % de lo que vemos y oímos y 90 % de lo que hacemos [18][17]. Estos porcentajes indican que, por lo tanto, la introducción del vídeo y los juegos interactivos en la clase puede producir modificaciones primordiales en el ámbito educativo.

La multimedia es un valioso recurso en la enseñanza por su naturaleza interactiva. Estos materiales deben ser adecuados y facilitar el aprendizaje. Para que eso deben ser de fácil instalación y uso, versátiles, mostrar información de calidad e interactivos para motivar a los alumnos. Ca-

1.3. ROBÓTICA EDUCATIVA

da material se ajusta a los usuarios para que trabajen a su ritmo [19].

Las nuevas tecnologías están cada vez más interiorizadas en las aulas gracias al uso de pantallas, proyectores y ordenadores. La pandemia vivida en este último año ha impulsado el uso de aplicaciones web para dar clase como las plataformas de videoconferencia. Muchos profesores han optado por estas potentes plataformas, otros han grabado y compartido sus propios videos. De esta forma los estudiantes pueden pararlo y verlo las veces necesarias e interiorizar mejor los conocimientos.

En los últimos años se están incorporando juegos en las aulas como la aplicación Kahoot! [20] en la que hacen juegos de encuestas, además se ha fomentado aprendizaje a través de videos y diapositivas más animadas, foros, así como el uso de plataformas educativas e interactivas como Moodle.

1.3.2. Gamificación en el aprendizaje

El término “*Gamificación*” o ludificación se emplea para referirse al aprendizaje a través de juegos en el entorno educativo y profesional. Los juegos se utilizan para fomentar el aprendizaje de programación, mejorando los conocimientos y habilidades de los alumnos de una forma más dinámica y divertida. La Gamificación facilita la interiorización de los conocimientos, generando una respuesta positiva al usuario por cumplir con un objetivo.

Enseñar a los más jóvenes como funcionan las últimas tecnologías y además les parezca entretenido e interesante es lo que nos ha motivado a realizar este trabajo.

1.3.3. Campeonatos de robótica educativa existentes

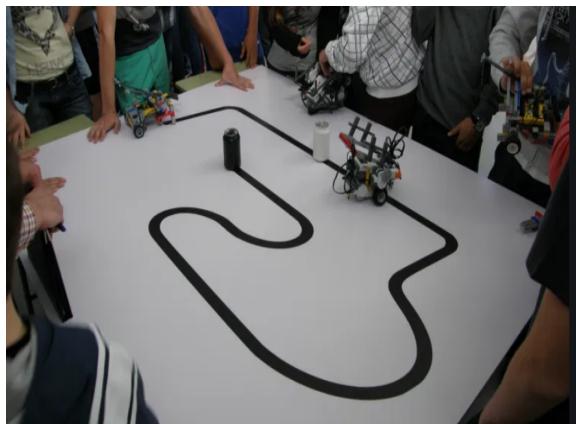
El aprendizaje de la robótica educativa ha ido más allá. Actualmente existen competiciones a nivel Nacional e Internacional que utilizan juegos y ejercicios competitivos [27]. Estas son algunas de ellas:

- RoboCup Junior ¹⁴: es una iniciativa educativa orientada a proyectos que patrocina eventos robóticos locales, regionales e internacionales para jóvenes estudiantes. Destaca la Liga de Rescate, Liga de Futbol (Figura 1.13(b)) y Liga ONSTAGE [28].

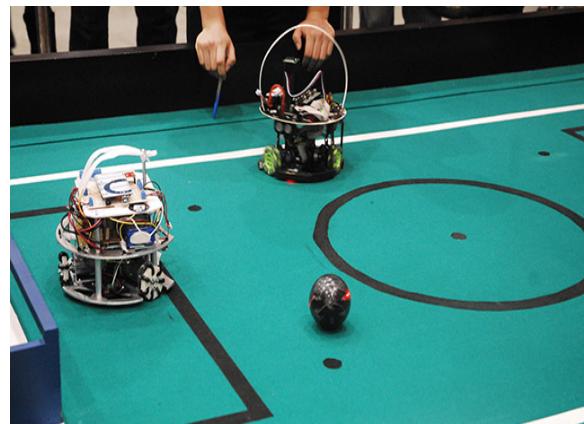
¹⁴<https://junior.robocup.org/>

1.3. ROBÓTICA EDUCATIVA

- Eurobot Junior¹⁵: es una competición europea de robots para estudiantes de primaria, secundaria o clubs de robótica. El grupo de jóvenes debe diseñar, construir y programar un robot telecontrolado por cable. Además pueden tener un robot secundario autónomo[29].
- First Lego league ¹⁶: es un programa internacional para jóvenes de 4 a 16 años a través de la resolución de problemas reales. Se adapta a cada edad con sus cursos Discover, Explore y Challenge[30].
- Torneo Nacional VEX Robotics IQ¹⁷: destinado a equipos de 4 a 8 miembros de secundaria junto un adulto. Ofrecen distintos retos y torneos con premios[31].
- RoboCampeones ¹⁸: Creado en el RoboticsLabURJC de la Universidad Rey Juan Carlos. Los alumnos de instituto compiten en pruebas como sumo con robots LEGO y Arduino. Se celebra en Fuenlabrada y en los últimos años contó con más de 2000 participantes (Figura 1.13(a)) [32].



(a) Juego Pañuelo Robocampeones



(b) Fútbol en RoboCup Junior.

Figura 1.17: Campeonatos de robótica.

¹⁵<http://www.eurobot.es/>

¹⁶<https://www.firstlegoleague.es/>

¹⁷<https://vexspain.com/>

¹⁸<http://robocampeones.org/>

1.4. Estructura del documento

La estructura de este trabajo fin de grado está compuesta por los siguientes capítulos:

- *Capítulo 1 Introducción*: una breve introducción a la robótica y tecnologías web para ponernos en contexto y presentar el tema principal del trabajo.
- *Capítulo 2 Objetivos y Metodología*: Se fijan los objetivos concretos y se explica la metodología y plan de trabajo seguidos a lo largo de este proyecto.
- *Capítulo 3 Infraestructura utilizada*: se describen lenguajes, tecnologías y herramientas empleadas en este trabajo.

Para una mejor exposición del trabajo que se ha realizado, el desarrollo se ha dividido en 3 capítulos, de esta forma nos centraremos en el enunciado, la infraestructura y solución de referencia de cada ejercicio:

- *Capítulo 4 Ejercicio Teleoperador Acústico y Banda Sonora*
- *Capítulo 5 Ejercicio Aspiadora robótica atrapa confeti*
- *Capítulo 6 Ejercicio Juego del pañuelo*
- *Capítulo 7 Conclusiones y trabajos futuros*: Conclusiones del trabajo y futuros proyectos posibles a partir de este.

Capítulo 2

Objetivos y Metodología del Trabajo

En este capítulo se plantean los objetivos a cumplir con este proyecto, así como los requisitos, metodología y el plan de trabajo que se ha seguido para alcanzarlos.

2.1. Objetivos

El objetivo principal de este trabajo es introducir la gamificación en la plataforma Kibotics, para ello, vamos a centrarnos en los siguientes subobjetivos u objetivos específicos a cumplir:

- Diseñar y desarrollar un nuevo juego que analice el audio en tiempo real y explorar la posibilidad de añadir bandas sonoras a los ejercicios actuales.
- Diseñar y desarrollar un nuevo ejercicio sobre una aspiradora robótica que tiene que limpiar una habitación.
- Diseñar y desarrollar un nuevo ejercicio sobre un robot que juega al pañuelo, recorriendo una línea, una lata que ejerce de pañuelo y regresando con ella al lugar de partida.

Para los tres ejercicios se desarrollará la infraestructura necesaria, modelos nuevos de robot, escenarios en el simulador, evaluadores automáticos, soluciones de referencia y se integrarán en la plataforma web de robótica educativa Kibotics.

2.2. Requisitos

Para cumplir con los objetivos citados anteriormente debemos tener en cuenta además los siguientes requisitos:

- Los robots y juegos desarrollados deben ser compatibles con la versión actual v.2.8 o superior de Kibotics.
- No se debe requerir de instalaciones adicionales. Todo debe correr en el navegador web del cliente.
- Uso de software de simulación Websim y A-Frame.

2.3. Metodología

La metodología que se ha seguido para la realización de este trabajo es la basada en el modelo de desarrollo software iterativo y creciente (Figura 2.1) [41]. Este modelo consiste en entregar a los usuarios y al equipo de desarrolladores de Kibotics, una versión usable lo antes posible y en continua actualización. En cada iteración se van solventando pequeños errores y mejoras que convergen en la versión final del proyecto.

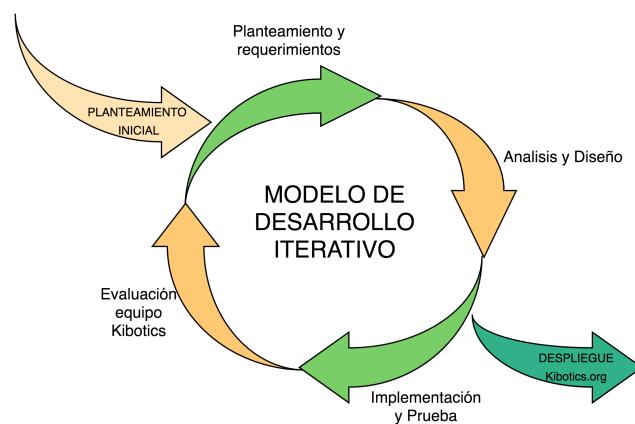


Figura 2.1: Modelo iterativo

El código desarrollado y las mejoras se han integrado progresivamente en el código fuente oficial de Kibotics en GitHub¹ mediante su flujo de trabajo incidencia (*issue*), rama (*branch*)

¹<https://github.com/>

2.4. PLAN DE TRABAJO

y parche (*pullrequest*). De esta forma la plataforma oficial está siempre actualizada con los últimos cambios añadidos y verificados por el equipo de desarrolladores.

Para llevar acabo esta metodología se establecieron reuniones semanales con el tutor para la orientación de este trabajo fin de grado. A lo largo del proyecto se ha mantenido la comunicación con el tutor y el equipo de Kibotics a través de la plataforma Slack².

Además se ha creado una página web tipo blog para llevar un seguimiento de las tareas realizadas y los objetivos semanales.³

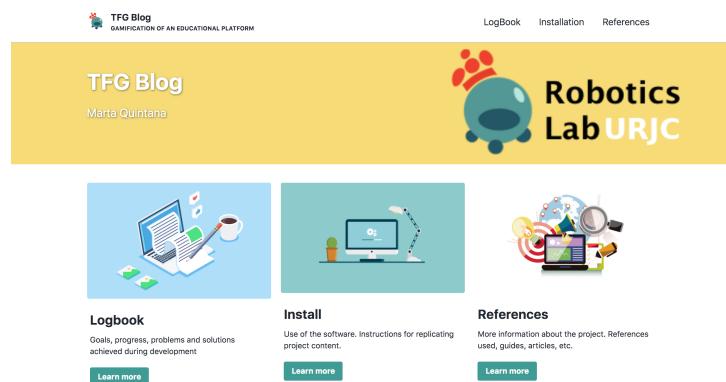


Figura 2.2: Página web de este TFG

2.4. Plan de Trabajo

El plan de trabajo seguido durante este proyecto se puede dividir en las siguientes etapas:

- **Aterrizaje en Kibotics y repaso de tecnologías web:** Toma de contacto con Kibotics y repaso de HTML5, JavaScript, Django y otras tecnologías que se utilizan en la plataforma.
- **Estudio Web Audio API y Tensor FlowJS:** Realización de ejercicios y tutoriales de distintas herramientas para introducir sonido y reconocimiento de audio en Kibotics.
- **Diseño Teleoperador Acústico con Teachable Machine:** Creación de un teleoperador acústico con reconocimiento de audio.
- **Prototipo de aspiradora robótica:** Diseño y creación de modelos del nuevo ejercicio.

²<https://slack.com/>

³<https://roboticslaburjc.github.io/2020-tfg-marta-quintana/>

2.4. PLAN DE TRABAJO

- **Diseño ejercicio aspiradora robótica confeti:** Desarrollo de una aspiradora robótica simulada, un nuevo actuador de absorción y piezas de confeti que puedan ser absorbidas por la aspiradora.
- **Prototipo ejercicio Mbot con Pinza basado en A-Frame:** Creación de un prototipo en A-Frame nativo para el estudio de las físicas y mallas de colisión.
- **Preparación del ejercicio del pañuelo:** Creación e implementación del ejercicio, desarrollo del circuito, una lata y un Mbot con Pinza realizados con Blender y JavaScript, para ofrecernos físicas más realistas.

Capítulo 3

Infraestructura utilizada

En este capítulo vamos a profundizar en las tecnologías y herramientas que se han empleado a lo largo de este trabajo. Por un lado se explican los lenguajes de programación (JavaScript, Python y Scratch) y los lenguajes de documentos (HTML5 y JSON) utilizados. Por otro lado, hablaremos de TensorFlowJS, Web Audio API y Teachable Machine, tecnologías que nos ofrecen procesamiento de audio en aplicaciones web. Otras aplicaciones como Blender y A-Frame han sido fundamentales para el modelado 3D y realidad virtual. Finalmente, hablaremos de la plataforma Kibotics, su estructura y las tecnologías web que la componen.

3.1. Lenguajes de programación y de documentos

3.1.1. HTML5

HTML5 es la última versión de HTML cuyas siglas corresponden a “*HyperText Markup Language*” es el bloque de construcción más básico de la Web[43].

“*HyperText*” significa hipertexto, se refiere a enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web.

“*Markup*” que significa marca o etiqueta, ya que todas las páginas web están construidas en base a etiquetas. Con ellas se anota texto, imágenes y otros contenidos para mostrar en un navegador web. El formato HTML5 incluye etiquetas como: <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, , , <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, ,

3.1. LENGUAJES DE PROGRAMACIÓN Y DE DOCUMENTOS

, y otros. En la Figura 3.1(a) podemos ver como es el uso de las etiquetas para la creación de una página web y en Figura 3.1(b) como se visualizaría en el navegador.

“Language” cuyo significado es lenguaje, porque HTML es un lenguaje, tiene sus normas, tiene su estructura y una serie de convenciones que nos sirven para definir tanto la estructura como el contenido de una web. Esto no quiere decir que sea un lenguaje de programación. HTML no tiene estructuras de lenguaje de programación, como los bucles, las condiciones, las funciones, etcétera. HTML5 es un estándar que sirve para definir la estructura y el contenido de una página Web [44].



```
html5_image.png index.html x
1 <!DOCTYPE html>
2 <html lang="es" dir="ltr">
3   <head>
4     <meta charset="utf-8">
5     <title>Mi pagina web</title>
6   </head>
7   <body>
8     <h1>Esto es una página web</h1>
9     
10   </body>
11 </html>
```



(a) Fuente de la página web

(b) Página web visualización

Figura 3.1: Ejemplo de página web con HTML5

Generalmente junto con HTML se utilizan otras dos tecnologías web: CSS para la apariencia/presentación y JavaScript para la funcionalidad/comportamiento de una página web.

Kibotics utiliza plantillas basadas en HTML5 en las páginas web que sirve. En este TFG se va a utilizar HTML5 para modificar y crear las plantillas de las páginas web que mostrarán los nuevos ejercicios que vamos a introducir en la plataforma.

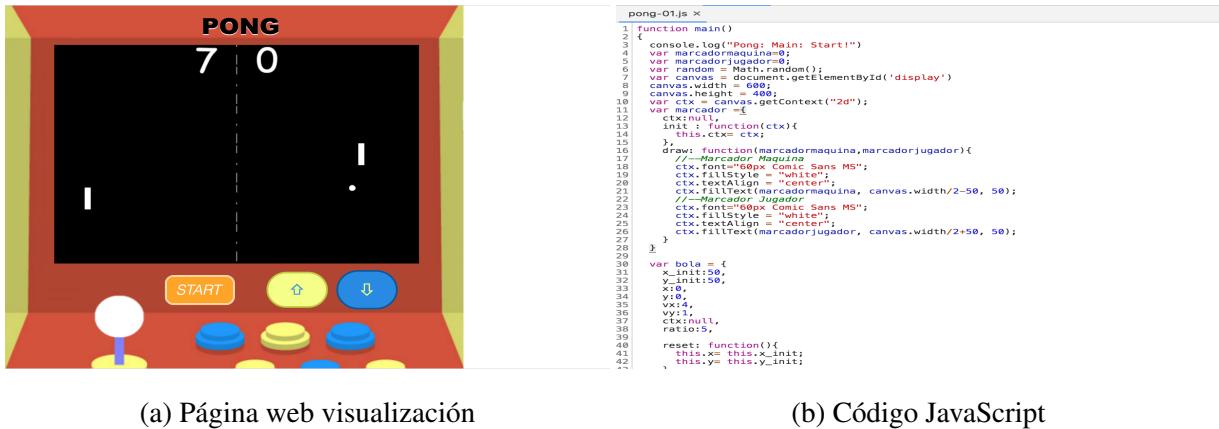
3.1.2. JavaScript

JavaScript es un lenguaje de programación interpretado, no necesita compilar los programas para ejecutarlos, pero sí un intérprete que en nuestro caso es el navegador. Sigue el estándar ECMAScript que se encarga de regir como debe ser interpretado, el buen funcionamiento y operatividad del lenguaje JavaScript. Este lenguaje posee una sintaxis derivada de C y Java pero no tiene nada que ver con ellos.

3.1. LENGUAJES DE PROGRAMACIÓN Y DE DOCUMENTOS

Se utiliza principalmente para crear páginas web interactivas. Una página web interactiva es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso [42].

El código JavaScript se ejecuta en el navegador del cliente y permite que este interactúe con la página web. En la siguiente Figura 3.2 podemos ver un ejemplo de página web interactiva.¹



(a) Página web visualización

(b) Código JavaScript

Figura 3.2: Ejemplo de página web interactiva con HTML5, CSS3 y JavaScript

En este trabajo JavaScript ha sido una pieza clave para su desarrollo y todos los ejercicios se han creado usando este lenguaje de programación.

3.1.3. Python y Scratch

Python es un lenguaje de programación orientado a objetos, es un lenguaje interpretado o de script, al igual que JavaScript necesita un programa intermedio que haga de “intérprete”. Es un lenguaje fuertemente tipado y multiplataforma. Su filosofía es que el código sea legible y tenga una sintaxis muy limpia. Sencillo pero potente, lenguaje muy usado en la universidad y trabajo [45].

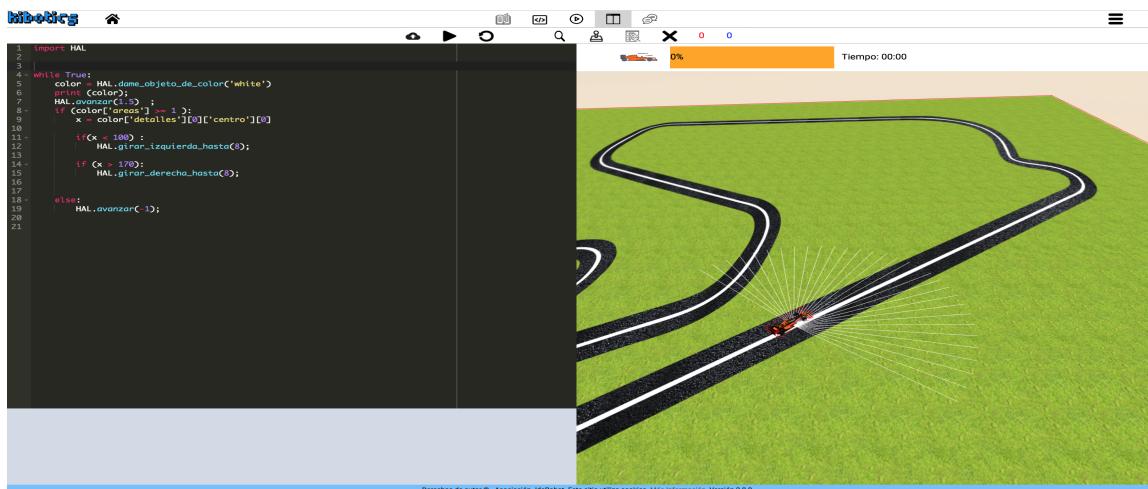
El servidor de Kibotics está programado en Python (usa la tecnología web Django). Los ejercicios disponibles en la plataforma se pueden resolver también con este lenguaje. Para usar Python las páginas web de los ejercicios contienen el editor ACE. ACE es un editor de código incrustable escrito en JavaScript [46], esto permite al usuario escribir el código de la solución del ejercicio en Python e internamente este código se traduce a JavaScript para el correcto funcionamiento de los ejercicios.

¹<https://martaquintana.github.io/2018-19-CSAAI-Pong/pong-01.html>

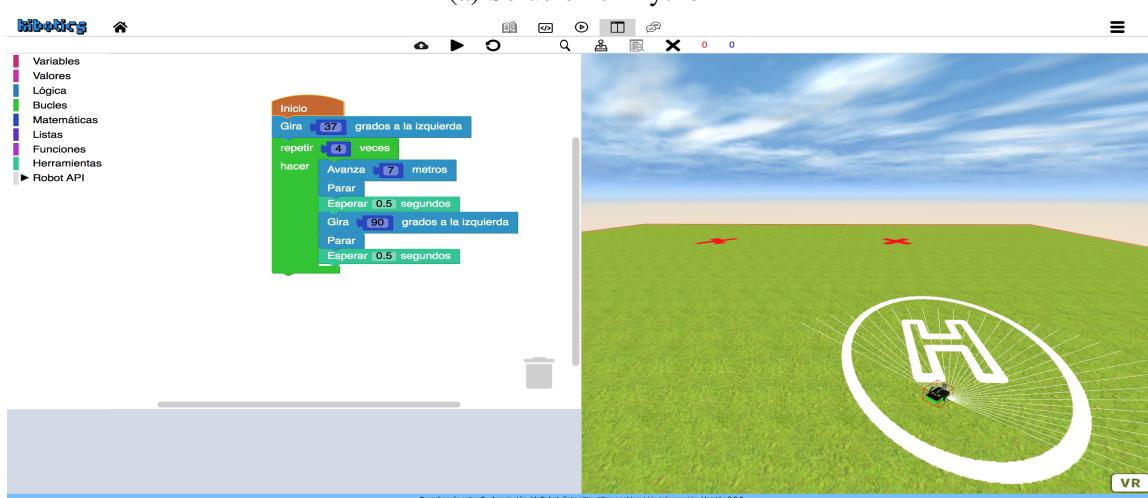
3.1. LENGUAJES DE PROGRAMACIÓN Y DE DOCUMENTOS

Como hemos comentado en la introducción Scratch es un lenguaje de programación visual basado en bloques, creado y mantenido por Lifelong Kindergarten group en el MIT Media Lab. Para usar Scratch en Kibotics se usa Blocky es una librería de JavaScript desarrollado por Google. Permite usar en una página web un editor de código visual. Es compatible con Chrome, Firefox, Safari, Opera y otros navegadores. Corre por el lado cliente y no tiene dependencia con el servidor [47].

Las soluciones de los ejercicios plateados en este TFG están resueltos tanto en Python como en Scratch al ser una plataforma para el ámbito educativo estos dos lenguajes son una buena alternativa para empezar a programar porque son lenguajes sencillos y con una curva de aprendizaje muy buena. En la Figura 3.3 podemos ver unos ejemplos de soluciones.



(a) Solución en Python



(b) Solución en Scratch

Figura 3.3: Ejemplo de código en Kibotics

3.1.4. JSON

JSON cuyas siglas significan JavaScript Object Notation, es un formato de intercambio de datos muy ligero. Para nosotros es fácil de leer y escribir, además la interpretación y generación de ficheros es muy sencilla para las máquinas [48].

JSON es un formato de datos basado en texto estándar para representar datos estructurados con la sintaxis de objetos de JavaScript. Son archivos de texto plano con codificación UTF8, que son compatibles con todos los sistemas. Se utiliza para transmitir datos en aplicaciones web [49]. Este formato puede ser utilizado independientemente de JavaScript, y muchos entornos de programación poseen la capacidad de leer (convertir; **parsear**) y generar ficheros JSON. En nuestro caso vamos a leer ficheros JSON desde JavaScript.

Un fichero JSON está compuesto de dos estructuras:

- **Una colección de pares de nombre/valor:** En otros lenguajes son conocidos como objeto, registro, estructura, diccionario o lista de claves. Ejemplo objeto :

```
1  {
2      "id" : 7,
3      "name" : "Robot",
4      "type" : "Drone"
5 }
```

- **Una lista ordenada de valores:** En la mayoría de los lenguajes, esto se implementa como arreglos “arrays”, vectores o listas. Ejemplo array:

```
1 [ "blue", "yellow", "orange" ]
```

Estas estructuras son universales en todos los lenguajes de programación, es por esto que JSON es muy fácil utilizar para un programador. Todos los lenguajes disponen de funciones para interpretar cadenas JSON y convertir datos en cadenas JSON válidas.

En la siguiente Figura 3.4 podemos ver una pequeña parte de un fichero de configuración de un ejercicio con JSON en Kibotics. En esta plataforma se utiliza JSON para crear los ejercicios con las características específicas que le correspondan y en este TFG se han usado tanto para la creación de los mundos como para fijar las posiciones de los confetis en el ejercicio del Roomba que veremos más adelante.

3.2. HERRAMIENTAS

```
1  {
2    "scene-parent-id": "myIFrame",
3    "scene": {
4      "id": "scene",
5      "gravity": -9.8,
6      "ground": "/static/websim/assets/textures/handkerchief.png",
7      "sky": "/static/websim/assets/textures/sky.png",
8      "background": "color: gray;",
9      "inspector": "url: https://aframe.io/releases/0.4.0/aframe-inspector.min.js",
10     "embedded": true,
11     "physics": "debug; true; friction:0.0002",
12     "renderer": "colorManagement: true"
13   },
14   "robots_config": [
15     {
16       "controller": "user1",
17       "id": "a-pibot"
18     }
19   ],
20   "evaluator_config": {
21     "world_limits": {
22       "x": [-40, 40],
23       "z": [-44, 24]
24     },
25     "idle_time": 30
26   },
27 }
```

Figura 3.4: Parte de un fichero config.json de un ejercicio en Kibotics

3.2. Herramientas

3.2.1. TensorFlowJS, Web Audio API y Teachable Machine

Para procesar el audio en JavaScript se han tenido que estudiar diferentes herramientas o APIs² de procesamiento de audio en la web. De esta forma hemos podido elegir la herramienta que mejor se adapta a nuestro problema. Entre ellas TensorFlowJS, WebAudioAPI y Teachable Machine (Figura 3.5) son las herramientas que se han estudiado para llevar a cabo el Teleoperador Acústico.

TensorFlow.js es una biblioteca de JavaScript para el entrenamiento y la implementación de modelos de aprendizaje automático en navegadores y en Node.js. TensorFlow es una plataforma de código abierto para la creación de modelos de aprendizaje automático

La API de Audio Web provee un sistema poderoso y versatil para controlar audio en la Web, permitiendo a los desarrolladores escoger fuentes de audio, agregar efectos al audio, crear visualizaciones de audios, aplicar efectos espaciales entre otras cosas [51].

Teachable Machine es una herramienta basada en la Web que hace posible crear modelos de aprendizaje automático de manera rápida, sencilla y accesible para todos [52].

²Interfaz de programación de aplicaciones

3.2. HERRAMIENTAS

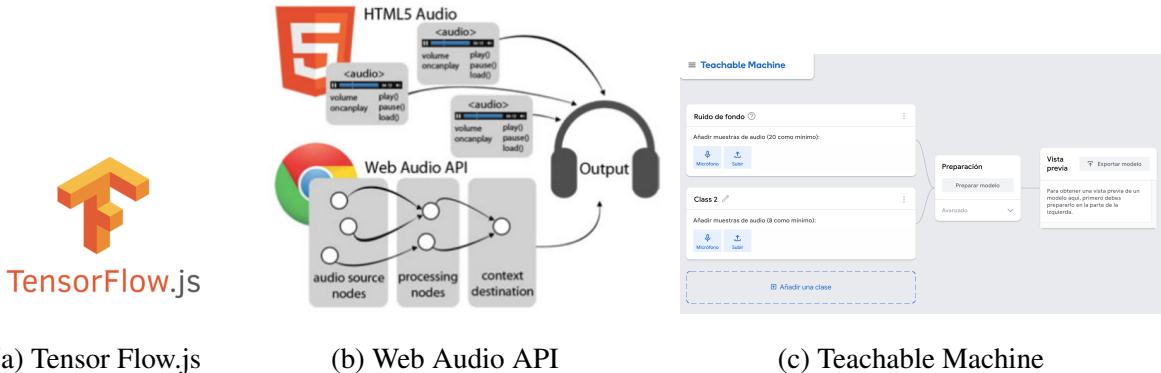


Figura 3.5: Herramientas reconocimiento de audio

En el Capítulo 4 se explicará en profundidad como se han usado estas tecnologías y por qué después de probar con cada una de estas tres herramientas, hemos elegido Teachable Machine para el desarrollo final del Teleoperador Acústico.

3.2.2. A-Frame

A-Frame es un marco web para crear experiencias de realidad virtual (VR). A-Frame utiliza HTML declarativo, lo que facilita bastante la creación de escenas en 3D y es accesible para todos, desde desarrolladores web, artistas, diseñadores, educadores a estudiantes. Su estructura entidad-componente proporciona una infinidad de posibilidades y ofrece compatibilidad con three.js, una biblioteca escrita en JavaScript para crear y mostrar gráficos animados 3D en un navegador Web.

Sin instalar nada A-Frame permite manejar modelos 3D para crear entornos de realidad virtual solamente usando las etiquetas `<script>` y `<a-scene>`. Es compatible con aplicaciones de realidad virtual como GearVR y Windows Mixed Reality entre otras, además funciona perfectamente en ordenadores y teléfonos inteligentes [53].

Para crear escenas en una página web solo es necesario importar la librería de A-Frame

```
1 <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
```

y poner las etiquetas correspondientes `a-scene` y los objetos que queramos que aparezcan en la escena con sus atributos. A continuación se muestra un ejemplo de código correspondiente con la visualización de la escena en el navegador en la Figura 3.6.

3.2. HERRAMIENTAS

```
1 <html>
2   <head>
3     <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
4   </head>
5   <body>
6     <a-scene>
7       <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
8       <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
9       <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-
10      cylinder>
11      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4">
12        </a-plane>
13      <a-sky color="#ECECEC"></a-sky>
14    </a-scene>
15  </body>
16 </html>
```

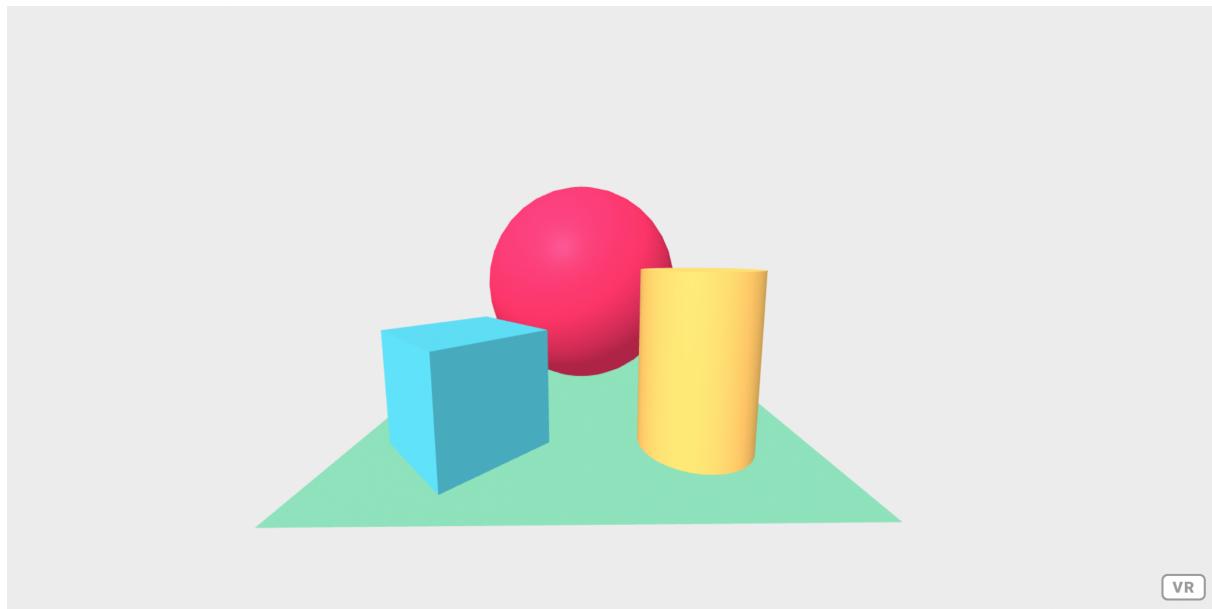


Figura 3.6: Ejemplo escena en A-Frame

A-Frame proporciona inspector visual 3D incorporado. Presionando **ctrl + alt + i**, podemos inspeccionar la escena, esto es muy útil cuando necesitas la posición concreta de un objeto.

3.2. HERRAMIENTAS

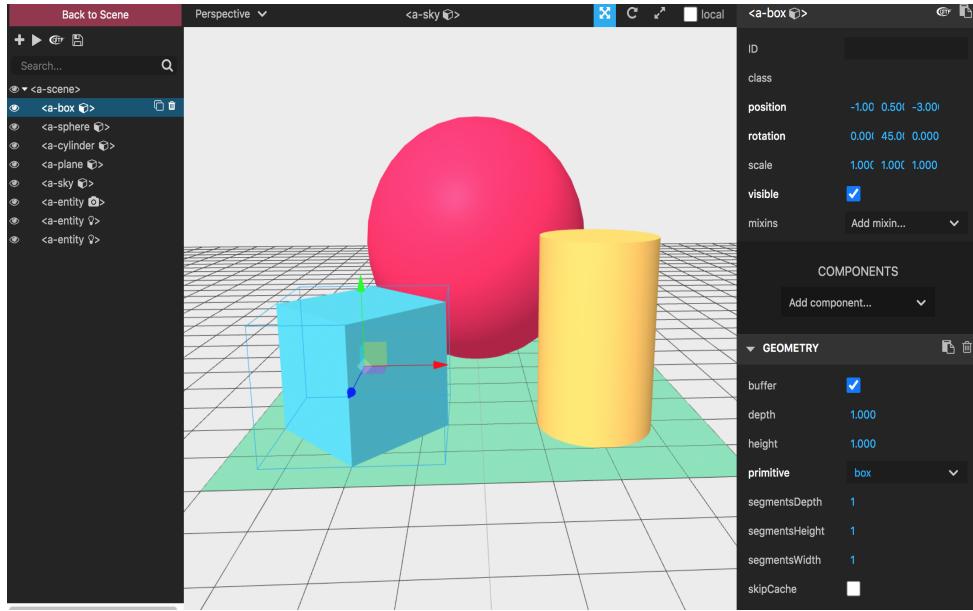


Figura 3.7: Inspector en A-Frame

En nuestro caso a-entity serán los modelos 3D de los robots que se crearán con Blender y se usarán componentes como a-box, a-plane, a-cylinder y a-sphere para crear componentes adicionales a la escena dándoles sus respectivas texturas.

3.2.3. Blender

Blender es un software de creación de modelos 3D gratuito, de código abierto y multiplataforma. Este programa se utiliza para el modelado, montaje, animación, simulación, renderizado 3d, así como para la composición y seguimiento de movimiento, edición de vídeo y animación 2D [54]. Para exportar los modelos de Blender se ha usado el formato glTF (GL Transmission Format). Es un formato de archivo para escenas y modelos 3D basado en JSON. De esta forma se pueden integrar los modelos en A-Frame de forma sencilla y rápida.

Este programa se ha utilizado en este proyecto para crear los nuevos robots, escenarios e introducir animaciones a la plataforma. En la Figura 3.8 se muestra un ejemplo de un cubo animado con Blender.

3.2. HERRAMIENTAS

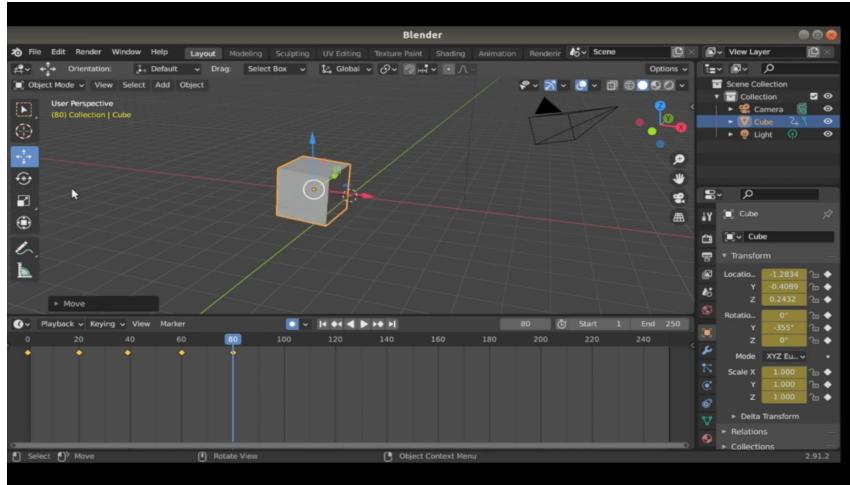


Figura 3.8: Blender

3.2.4. Plataforma Kibotics

Kibotics es una plataforma web para docencia en robótica y programación. Desarrollada por la Asociación JdeRobot³. Esta plataforma se basa en tecnologías web como Django para la parte servidor y utiliza un simulador llamado Websim que se apoya en A-Frame para representar los escenarios de los ejercicios en el navegador del usuario.

Esta plataforma en línea ofrece contenidos educativos para facilitar el aprendizaje en programación a alumnos de primaria, secundaria y bachillerato. Ofrece cursos en lenguajes Scratch y Python. Los ejercicios están disponibles tanto para robots físicos como simulados. Destacan Mbot, Dron Tello y Pibot entre otros. También ofrece iteracción social gracias a un foro.

La simulación de robots reales permite depurar el software al máximo antes de ejecutarlo en un robot físico y así reducir los costes, ya que no necesitas tener un robot para cada alumno y evitar posibles daños de los robots y accidentes.

Para usar esta plataforma no es necesario instalar nada, solo tener acceso a Internet. Al ser una aplicación web tenemos la ventaja de que es multiplataforma y podemos usarla en distintos dispositivos.

Kibotics tiene la filosofía *Learn by doing*, aprender haciendo. Las lecciones de teoría junto ejercicios prácticos hacen que los contenidos educativos se vayan adaptando según la complejidad de cada ejercicio. En la Figura 3.9 podemos ver algunos de los ejercicios que ofrecen.

³<https://jderobot.github.io/>

3.2. HERRAMIENTAS

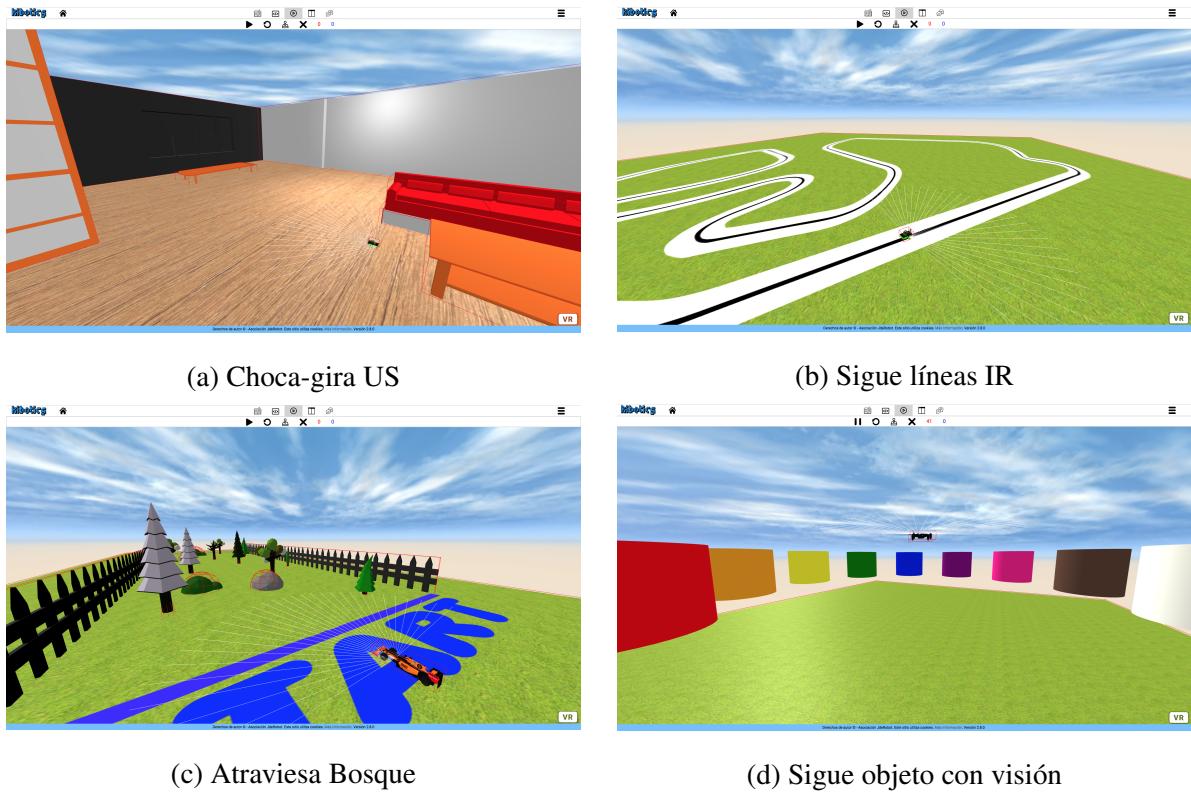


Figura 3.9: Ejercicios Kibotics

Kibotics en los últimos años ha ofrecido cursos para la escuela de pensamiento computacional INTEF, el ayuntamiento de Fuenlabrada, la empresa Logix5, Universidad Rey Juan Carlos, IES Martínez Uribarri (Salamanca) y la Comunidad de Madrid [56].

3.2.5. Navegadores web utilizados

En este Trabajo Fin de Grado se han usado los navegadores Chrome y Firefox por ser los navegadores más extendidos y que proporcionan gran rendimiento, estabilidad y seguridad. Se han utilizado para la visualización de la página web de Kibotics y la ejecución y pruebas de los distintos ejercicios.

Capítulo 4

Ejercicio teleoperador acústico y banda sonora

En los siguientes capítulos se explican los ejercicios desarrollados en este trabajo fin de grado. El primero de ellos es el teleoperador acústico. En este capítulo se nombran las herramientas utilizadas y como se ha realizado este ejercicio. También se muestra como se ha añadido la posibilidad de incluir banda sonora a los ejercicios ya existentes.

Capítulo 5

Ejercicio aspiradora robótica atrapa confeti

En este capítulo se explica el desarrollo del ejercicio de una aspiradora robótica que atrapa confeti. Por un lado se expone el enunciado y objetivos del ejercicio, por otra parte cómo se ha implementando en la plataforma Kibotics y finalmente una solución de referencia.

5.1. Enunciado

El objetivo de este ejercicio es hacer una aspiradora robótica que se encuentre en una habitación y sea capaz de recoger piezas de confeti cuando esta pase por encima. Para tener un ejercicio más completo se creará un evaluador automático que cuente el confeti recogido en un tiempo determinado.

El alumno deberá programar en Scratch o en Python un algoritmo de planificación de ruta para que la aspiradora consiga atrapar el mayor número de confetis de la habitación en 5 minutos.

5.2. Desarrollo del ejercicio

Para hacer este ejercicio se utilizaron las herramientas JavaScript, Blender, el simulador Websim de Kibotics que contiene la tecnología A-Frame y los lenguajes HTML5 y JSON.

Lo primero que se realizó para crear este ejercicio fueron varios prototipos en Blender de una aspiradora robótica. En las Figura se pueden ver el modelo que se ha utilizado finalmente para el aspirador..

5.2. DESARROLLO DEL EJERCICIO

figura con roombagrey en blender

A la par se creó el modelo de confeti con la etiqueta a-cylinder de A-Frame para hacer pruebas de absorción. **insertar video absorción** fotos sin absorber absorber confeti rojo ****

El simulador Websim analiza (*parsea*) ficheros JSON para formar mundos con la tecnología A-Frame. De esta forma es sencillo crear los objetos con pares “atributo”:valor. A continuación podemos ver como se define el id robot del nuevo robot, se importa el gltf-model que hemos creado en Blender de la aspiradora, y se le asignan otros atributos como la posición, escala y rotación del robot. El dynamic-body nos facilita el movimiento del robot y el atributo collide asigna una malla de colisión para que la aspiradora pueda chocarse con los demás elementos del mundo usando físicas de A-Frame y todo sea mucho más realista.

```
1   [...]
2       {
3           "tag": "a-robot",
4           "attr": {
5               "id": "a-pibot",
6               "gltf-model": "/static/websim/assets/models/roombajderbotgrey.gltf",
7               "scale": { "x":2, "y":2, "z":2},
8               "position": { "x":0, "y":4, "z":30},
9               "rotation": { "x":0, "y":90, "z":0},
10              "dynamic-body":{ "mass": 10},
11              [...]
12              "collide":{}
13
14      },
15
16  [...]
```

El confeti no se crea desde el fichero de configuración dado que se tedrían que crear uno a uno y esto extendería demasiado el código del fichero de configuración. Lo que se ha hecho es que una vez esta cargado el mundo en el navegador del usuario, usando JavaScript creamos dinámicamente todos los confetis.

Para hacerlo de esta forma, primero se generó un programa para fijar aleatoriamente las posiciones de los confetis en el mundo, así los confetis quedan esparcidos por la habitación uniformemente y todos los alumnos cuentan con el mismo escenario. Estas posiciones x, y, z de los confetis se guardaron en otro fichero JSON llamado data.json. En total el escenario está formado por 100 confetis de colores, el color de cada confeti se elige aleatoriamente cuando se

5.2. DESARROLLO DEL EJERCICIO

crea desde JavaScript .

Para leer data.json se utiliza

```
1
2 <script type="text/javascript" src="#">% static '/data.json' %></script>
```

en la plantilla html del ejercicio.

```
1
2 function getRandomColor() {
3     var letters = '0123456789ABCDEF';
4     var color = '#';
5     for (var i = 0; i < 6; i++) {
6         color += letters[Math.floor(Math.random() * 16)];
7     }
8     return color;
9 }
10
11 document.addEventListener('robot-loaded', (evt)=>{
12     localRobot = evt.detail;
13     console.log(localRobot);
14
15     var sceneEl = document.querySelector('a-scene');
16
17     // CREATE CONFETTI
18     var n = 0;
19     var n_confetis = 99;
20     score = 0;
21     var array = JSON.parse(data);
22
23     for ( n = 0; n <=n_confetis ; n++) {
24         var c = document.createElement('a-cylinder');
25         var num_conf=confeti+ String(n)
26         c.setAttribute('id', num_conf);
27         pos = {x:array[n].x, y:0,z:array[n].z}
28
29         c.setAttribute('position',pos);
30
31         var color = getRandomColor();
32         c.setAttribute('color', color);
33         c.setAttribute('height', "0.25");
34         c.setAttribute('radius', 1);
35         sceneEl.appendChild(c);
36     }
```

En el código anterior podemos ver como se crea el confeti con a-cylinder. El id es confeti más un número n que es un número de 0-99 que se le asigna para crear 100 confetis diferentes

5.2. DESARROLLO DEL EJERCICIO

(Ejemplo del id del confeti número 50 id="confeti50"). Además se le asigna los atributos posición con las posiciones que se leen del data.json, el color aleatorio y como es un cilindro, se define la altura y radio del confeti. Los confetis no cuentan con malla de colisión para que el aspirador pueda pasar por encima de ellos y absorberlos de una forma más natural que si lo hacemos por colisión.

La absorción se implementó en JavaScript. Se utilizó la función setInterval que ejecuta una serie de instrucciones indefinidamente cada un cierto periodo de tiempo. Cada 25ms en este programa se comprueba la distancia entre la aspiradora robótica y cada uno de los confetis. La posición del robot es su centro de masas, por eso se utilizó la distancia euclídea para calcular la distancia entre el centro del robot y el centro del confeti n, si esta distancia d es menor o igual a 2, el confeti n cambia su atributo 'visible' a false.

Dentro de la absorción se estableció el evaluador automático. Cada vez que d es menor o igual a 2 se suma un punto e indica que el confeti n ha sido “absorbido” por el aspirador. La puntuación máxima es de 100 (depende del número de confetis 0-99). También se establece una cuenta atrás de 5:00 minutos y cuando llega a 0:00 se deja de contar el confeti. Aunque la aspiradora siga pasando por encima de otros confetis que siguen visibles en el escenario, estos no serán absorbidos ni se aumentará el contador del evaluador.

```
1 startEvaluator = () => {
2     started = true;
3 }
4 roomba=sceneEl.querySelector('#a-pibot')
5
6 setInterval(function() {
7     //console.log("Roomba",roomba.getAttribute('position').z);
8     // console.log("Confeti",confeti.getAttribute('position'));
9     //console.log("Confeti",confeti.getAttribute('position').z)
10    for ( n = 0; n <=n_confetis ; n++) {
11        d = Math.sqrt(Math.pow((array[n].z-roomba.getAttribute('position').z), 2)+Math.pow(
12            array[n].x-roomba.getAttribute('position').x), 2));
13
14        if ( d <= 2 ){
15            num_conf="#confeti"+ String(n)
16            confeti=sceneEl.querySelector(num_conf)
17            if (confeti.getAttribute('visible') == true) {
18
19                var counter= document.getElementById('time').innerHTML;
20                // Tiempo: 00:00
```

5.3. SOLUCIÓN DE REFERENCIA

```
21     if((counter[8] =='0') && (counter[9]=='0') && (counter[11]=='0') && (counter[12]== '0'))
22     ) {
23     score = score;
24   }else{
25     score+=1;
26   document.getElementById('confeti_recogido').innerHTML = "Confetti recogido: "+ score;
27 }
28   confeti.setAttribute('visible', false);
29
30
31 }
32 }
33
34 }, 25);
35 );
```

Una vez se consiguió que los confetis desaparecieran del mundo cuando la aspiradora pasara por encima, se mejoró el mundo creando nuevos muebles con Blender, en este caso queríamos que el escenario fuera una habitación de una casa.

Fotos muebles Blender y en A-frame antes y despues

Para que el escenario tenga mayor dificultad se introdujeron 2 pelotas con movimiento. Para el movimiento había dos posibilidades, animación desde Blender y Animación desde A-Frame. Se estudiaron ambas opciones. En el video de este enlace *** se puede ver como hacer una animación en Blender. Se optó por la animación en A-Frame desde el fichero de configuración que era el metodo ideal para Websim, de esta forma dejamos toda la creación de las pelotas, asignación de texturas, posición y animación definido en la configuración.

Animaciones en Blender y Aframe poner lo de las pelotas

** Página de teoria

5.3. Solución de referencia

Una de las posibles soluciones de este ejercicio en Python es la que se muestra en la Figura En la Figura ... se puede ver una solución en Scratch para el mismo ejercicio.

*** FOTOS SOLUCIONES REFERENCIA*** Este ejercicio ya está disponible en la plataforma Kibotics. En este TFG también se ha realizado un video promocional para presentar el nuevo ejercicio de la plataforma. **enlace video***

Capítulo 6

Ejercicio juego del pañuelo

En este capítulo se expone como ha sido el desarrollo del ejercicio del juego del pañuelo. Primero hablaremos del enunciado del ejercicio, seguidamente se muestra como ha sido su implementación, que herramientas se han utilizado, prototipos y como finalmente se ha llevado a cabo el ejercicio. Finalmente se ofrecen dos soluciones de referencia que resuelven el ejercicio en Kibotics.

6.1. Enunciado

El propósito de este ejercicio es crear un nuevo ejercicio en Kibotics con un nuevo robot que sea capaz de coger objetos y moverlos por el escenario. El nuevo robot debe tener pinzas móviles para atrapar otros objetos. El juego del pañuelo es muy popular entre los más pequeños y también en el mundo de la robótica. En las competiciones de robots este juego consiste en programar un sigue lineas el robot debe recorrer un circuito y el robot debe ser capaz quitar una o varias latas que obstruyan el camino para poder realizar el circuito. Para que nuestro ejercicio sea más acorde con el juego del pañuelo que conocemos todos, necesitamos hacer un robot con pinzas, un circuito y una lata ajustados a las necesidades del circuito.

El alumno deberá programar en Python o en Scratch un algoritmo que permita que el Mbot Pinza avance siguiendo la linea negra del circuito hasta que se encuentre a poca distancia de la lata, una vez se encuentre enfrente de la lata, tu robot debe cerrar las pinzas y dar media vuelta para volver a la casilla de salida y llevar consigo la lata en todo momento. Gracias a un evaluador automático vamos a obtener la puntuación, esta depende del porcentaje de circuito

recorrido y si se lleva o no la lata entre las pinzas.

Juego del pañuelo robotica

6.2. Desarrollo del ejercicio

Para este ejercicio se investigó con un prototipo en A-Frame nativo, (HTML5, JavaScript y la librería de A-Frame) sin Websim. En este primer prototipo se cogía el modelo gltf del Mbot que ya se usaba en Kibotics para algunos ejercicios y lo representaba en la escena, el robot se movía con eventos de teclado desde JavaScript que hacían que este cambiara su posición. Se crearon las pinzas que eran dos octaedros a los lados del robot. Estas pinzas estaban creadas con a-box y eran independientes del robot. Con este prototipo se estudió el movimiento de las pinzas con respecto a la posición del robot. En el código JavaScript era muy complejo y dependía de la actualización continua de la posición y se encontraron muchas dificultades a la hora de realizar giros del robot con las pinzas dado que estas están definidas por el centro de masas y al rotarlas había que hacerlo con funciones senoidales para que se movieran en concordancia con la rotación del robot. Esta dependencia continua de la posición del robot con respecto a las pinzas y la lata con el cierre de las pinzas era muy compleja y muy poco realista en este video se puede ver como era el funcionamiento de este prototipo. *video

La idea principal es que el robot pudiera coger con las pinzas una lata de la forma más natural posible. Se creó un segundo prototipo esta vez había que hacer que las pinzas fueran dependientes del robot para que su movimiento y los giros sean coherentes.

Se estudió la posibilidad de hacer unas pinzas que con mallas de colisión fueran capaces de atrapar una lata.

*mallas colision aframe

Se creó el modelo 3D en Blender de una lata, con GIMP un programa de edición de imágenes gratuito y multiplataforma se diseño la imagen .png del circuito.

**LAta **Pinzas ** implementacion config **implementación HAL API brains/miniproxy-worker.js e interfaces robot.

** mallas colision y elemento static.

6.3. Solución de referencia

Este ejercicio se puede resolver de muchas formas, una de ellas es la que se muestra en la Figura ... para Python y para Scratch una solución posible es la que se indica en la Figura... .

videos promocionales e imágenes soluciones

Capítulo 7

Conclusiones y trabajos futuros

Bibliografía

- [1] “Kibotics.” <https://kibotics.org/> (accessed Apr. 18, 2021).
- [2] R. Barrientos Sotelo, Víctor Ricardo; García Sánchez, José Rafael; Silva Ortigoza, “Robots Móviles: Evolución y Estado del Arte,” Polibits, vol. 19, pp. 228–237, 2007, doi: 10.3233/978-1-60750-530-3-228.
- [3] “¿Quién inventó la palabra robot?” <https://www.muyinteresante.es/cultura/arte-cultura/articulo/i quien-invento-la-palabra-robot> (accessed Apr. 18, 2021).
- [4] “Abex.” <https://www.abexsl.es/es/sistema-robotico-da-vinci/que-es> (accessed Apr. 18, 2021).
- [5] “Los mejores KITS de ROBÓTICA y ROBOTS para NIÑOS por edades en 2020.” <https://revistaderobots.com/robotica-educativa/comprar-kits-de-robotica-y-robots-programables-para-ninos/> (accessed Apr. 18, 2021).
- [6] “mBot Explorer Kit,Makeblock ROBOTIX.” <https://www.robotix.es/es/mbot> (accessed Apr. 18, 2021).
- [7] “¿Qué es la World Wide Web (www) y cómo funciona?” <https://www.fotonostra.com/digital/paginasweb.htm> (accessed Apr. 18, 2021).
- [8] “Introducción a las tecnologías web · myTeachingURJC/2018-19-CSAAI Wiki.” <https://github.com/myTeachingURJC/2018-19-CSAAI/wiki/Introducción-a-las-tecnologías-web#tecnologías-en-el-lado-del-servidor> (accessed Apr. 18, 2021).
- [9] “Tecnologías y herramientas para el desarrollo web.” http://cv.uoc.edu/annotation/a9c35c372dcee6e6b92afad6993cd048/620334/PID_00250214/PID_00250214.html (accessed Apr. 18, 2021).

BIBLIOGRAFÍA

- [10] “The Web framework for perfectionists with deadlines — Django.” <https://www.djangoproject.com/> (accessed Apr. 18, 2021).
- [11] “Web Service Efficiency at Instagram with Python — by Instagram Engineering — Instagram Engineering.” <https://instagram-engineering.com/web-service-efficiency-at-instagram-with-python-4976d078e366> (accessed Apr. 18, 2021).
- [12] “Acerca — Node.js.” <https://nodejs.org/es/about/> (accessed Apr. 18, 2021).
- [13] “Top Companies That Use Node.JS in Production: Netflix, Trello, and Co.” <https://youteam.io/blog/top-companies-that-used-node-js-in-production/> (accessed Apr. 18, 2021).
- [14] “¿Qué es PHP? y ¿Para qué sirve? Un potente lenguaje de programación para crear páginas web. (CU00803B).” https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:que-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193 (accessed Apr. 18, 2021).
- [15] “Top 10 websites built with PHP technology - Facebook, Yahoo...” <https://cybercraftinc.com/blog/top-10-projects-developed-with-php-technology> (accessed Apr. 18, 2021).
- [16] “La importancia del contenido multimedia en educación.” <https://www.cursosfemxa.es/blog/14089-la-importancia-del-contenido-multimedia-en-educacion> (accessed Apr. 18, 2021).
- [17] E. Andrade and E. Chacón, “Implicaciones teóricas y procedimentales de la clase invertida,” *Pulso*, vol. 41, pp. 251–268, 2018.
- [18] “El Vídeo Educativo como recurso dinamizador del Aprendizaje - EVirtualplus.” <https://www.evirtualplus.com/video-educativo-como-recurso-aprendizaje/> (accessed Apr. 20, 2021).
- [19] “La Importancia Del Contenido Multimedia En La Educación.” <https://es.calameo.com/read/005984440640dded50f70> (accessed Apr. 18, 2021).

BIBLIOGRAFÍA

- [20] “Play Kahoot! - Enter game PIN here!” <https://kahoot.it/> (accessed Apr. 18, 2021).
- [21] “Robótica educativa: ¿qué es y cuáles son sus ventajas?” <https://www.unir.net/educacion/revista/robotica-educativa/> (accessed Apr. 20, 2021).
- [22] “Scratch - Imagine, Program, Share.” <https://scratch.mit.edu/> (accessed Apr. 18, 2021).
- [23] E. P. Morales and M. F. G. Muñoz, “Manual Open Roberta.” http://robomatrix.org/wp-content/uploads/2021/Manual_OpenRoberta.pdf (accessed May 09, 2021).
- [24] “LEGO MINDSTORMS Education EV3 — LEGO® Education.” <https://www.robotix.es/es/lego-mindstorms-education-ev3> (accessed May 09, 2021).
- [25] “Recursos, Software y Actividades GRATIS — LEGO Education - ROBOTIX.” <https://www.robotix.es/es/descargar-software-lego-education> (accessed May 09, 2021)
- [26] “What Is mBlock 5?” <https://www.yuque.com/makeblock-help-center-en/mblock-5/overview> (accessed May 09, 2021).
- [27] “Competiciones de robótica educativa a los que apuntarse.” <https://blog.juguetronica.com/competiciones-de-robotica-educativa/#robocampeones> (accessed Apr. 18, 2021).
- [28] “RoboCupJunior – Creating a learning environment for today, fostering technological advancement for tomorrow.” <https://junior.robocup.org/> (accessed Apr. 20, 2021).
- [29] “EUROBOT JR.” <http://www.eurobot.es/index.php/eurobot-jr> (accessed Apr. 20, 2021).
- [30] “FIRST LEGO League.” <https://www.firstlegoleague.es/> (accessed Apr. 20, 2021).
- [31] “Vex Spain — Torneos.” <https://vexspain.com/vex-iq/torneos/> (accessed Apr. 20, 2021).
- [32] “Robocampeones” <http://robocampeones.org/> (accessed Apr. 20, 2021).
- [33] Imágenes de: “Pexels.” <https://www.pexels.com/es-es/> (accessed Apr. 18, 2021).
- [34] Imágenes de :“ Pixabay.” <https://pixabay.com/es/> (accessed Apr. 18, 2021).

BIBLIOGRAFÍA

- [35] “Programación para niños con vídeos de Scratch y Arduino.” <https://revistaderobots.com/robots-y-robotica/lenguajes-de-programacion-para-ninos-y-ninas/> (accessed Apr. 18, 2021).
- [36] “Tecnologías para el desarrollo web más actuales — proun Madrid - Asturias.” <https://www.proun.es/blog/tecnologias-web-actuales/> (accessed Apr. 18, 2021).
- [37] “Conceptos básicos sobre tecnologías de desarrollo web - ingeniovirtual.com.” <https://www.ingeniovirtual.com/conceptos-basicos-sobre-tecnologias-de-desarrollo-web/> (accessed Apr. 18, 2021).
- [38] “Competencia docente de la robótica educativa: ¿una realidad o un nuevo reto para el profesorado? - Equipamiento para centros educativos.” <https://www.interempresas.net/Tecnologia-aulas/Articulos/156527-Competencia-docente-de-robotica-educativa-realidad-o-nuevo-reto-para-profesorado.html> (accessed Apr. 18, 2021).
- [39] “Así se enseña robótica y programación en las aulas españolas.” <https://www.educaciontrespuntocero.com/noticias/robotica-y-programacion-espana/> (accessed Apr. 18, 2021).
- [40] “historia de robotica - el mundo de la robotica 604.” <https://sites.google.com/site/elmundodelarobotica604/historia-de-robotica> (accessed Apr. 18, 2021).
- [41] “Modelos de desarrollo de software” <https://www.elconspirador.com/2013/08/19/modelos-de-desarrollo-de-software/> (accessed Apr. 25, 2021).
- [42] J. Eguílez Pérez, “Introducción a JavaScript.” Accessed: Apr. 26, 2021. [Online]. Available: www.librosweb.es.
- [43] “HTML: lenguaje de marcado de hipertexto — MDN.” <https://developer.mozilla.org/en-US/docs/Web/HTML> (accessed May 02, 2021).
- [44] ”Qué es HTML5: Definición y funcionamiento — OpenWebinars.” <https://openwebinars.net/blog/que-es-html5/> (accessed May 02, 2021).

BIBLIOGRAFÍA

- [45] R. G. Duque, “Python PARA TODOS.” Accessed: May 03, 2021. [Online]. Available: <http://mundogeek.net/tutorial-python/>.
- [46] “Ace - The High Performance Code Editor for the Web.” <https://ace.c9.io/> (accessed May 03, 2021).
- [47] “Blockly — Google Developers.” <https://developers.google.com/blockly> (accessed May 03, 2021).
- [48] “Trabajando con JSON Aprende sobre desarrollo web — MDN.” <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON> (accessed May 03, 2021).
- [49] “JSON.” <https://desarrolloweb.com/home/json> (accessed May 03, 2021).
- [50] “TensorFlow Core — Aprendizaje automático para principiantes y expertos.” <https://www.tensorflow.org/overview?hl=es-419> (accessed May 03, 2021).
- [51] “Web Audio API - Referencia de la API Web — MDN.” https://developer.mozilla.org/es/docs/Web/API/Web_Audio_API (accessed May 03, 2021).
- [52] “Teachable Machine.” <https://teachablemachine.withgoogle.com/> (accessed May 03, 2021).
- [53] “Introduction – A-Frame.” <https://aframe.io/docs/1.2.0/introduction/#getting-started> (accessed May 03, 2021).
- [54] “blender.org - Home of the Blender project - Free and Open 3D Creation Software.” <https://www.blender.org/> (accessed May 03, 2021).
- [55] “Vista de Navegadores web — El Tecnológico.” <https://revistas.utp.ac.pa/index.php/el-tecnologico/article/view/1287/html> (accessed May 03, 2021).
- [56] J. María and C. Plaza, “Robótica del siglo XXI y educación.”