



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE
TELECOMUNICACIÓN

GRADO EN INGENIERÍA EN SISTEMAS AUDIOVISUALES
Y MULTIMEDIA

TRABAJO FIN DE GRADO

Gamificación de una plataforma 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. Gamificación	9
1.3.1. Importancia de los juegos y multimedia en el aprendizaje	9
1.3.2. Robótica educativa	10
1.3.3. Campeonatos de robótica educativa existentes	12
2. Objetivos y Metodología del Trabajo	13
2.1. Objetivos	13
2.2. Requisitos	14
2.3. Metodología	14
2.4. Plan de Trabajo	15
2.5. Estructura del documento	16
3. Infraestructura utilizada	17
3.1. Lenguajes de programación y documentos	17
3.1.1. HTML5	17
3.1.2. JavaScript	17
3.1.3. Python y Scratch	18
3.1.4. JSON	18

3.2. Herramientas	18
3.2.1. TensorFlowJS, Web Audio API y Teachable Machine	18
3.2.2. Blender	19
3.2.3. A-Frame	19
3.2.4. Plataforma Kibotics	19
3.2.5. Websim	19
Bibliografía	20

Índice de figuras

1.1.	Ejemplos de Robots	2
1.2.	Sensor ultrasonidos	3
1.3.	Motor	3
1.4.	Procesador	4
1.5.	Pantalla	4
1.6.	Ejemplos de Robots educativos	4
1.7.	Ejemplos aplicaciones web	5
1.8.	Ejemplo de una 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.	Scratch	10
1.12.	Plataforma Kibotics.	11
1.13.	Ejercicios en campeonatos de robótica.	12
2.1.	Modelo iterativo	14
2.2.	Página web de este TFG	15

Capítulo 1

Introducción

Este Trabajo Fin de Grado se enmarca en el ámbito educativo, en concreto en la "Gamificación de una plataforma 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 cuantiosas tecnologías web y herramientas que han servido para crear nuevos ejercicios para la plataforma de Robótica educativa *Kibotics* [1] de la Asociación de Robótica e Inteligencia Artificial *JdeRobot*.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 diversas áreas de conocimiento como son las matemáticas, la electrónica, mecánica, informática y física.

A lo largo de la historia el ser humano siempre ha tenido interés en crear herramientas o máquinas para facilitarse la vida. En antiguas civilizaciones se hablaba de "seres mecánicos con vida", eran mecanismos basados en poleas y bombas hidráulicas. Se desarrollaron un gran número de figuras dotadas con partes móviles pero no fue hasta finales del siglo XVIII, con

¹<https://jderobot.github.io/>

1.1. ROBÓTICA

la Revolución Industrial, cuando se impulsó el crecimiento de diferentes áreas científicas y tecnológicas. Estos adelantos científico-técnicos impulsaron el desarrollo la robótica que conocemos hoy. [2]

El término *robot* proviene del vocablo checo *roboťa*, que significa servidumbre o trabajo esclavizador. Fue usado por primera vez por el dramaturgo checoslovaco Karel Čapek(1890-1938) en su obra de teatro: Rossum´s Universal Robots, en 1920, una obra teatral basada en la ciencia ficción donde humanos artificiales realizan trabajos forzados. [3]

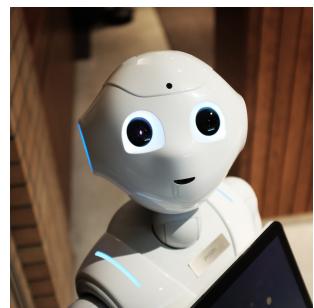
No fue hasta los años 50 cuando se desarrollaron los primeros robots. Actualmente con el rápido avance en tecnología la realidad ha superado a la ficción, estamos rodeados de robots. La Figura 1.1 muestra algunos ejemplos de robots. A primera vista parecen muy diferentes, pero vamos a ver que tienen muchas cosas en común.



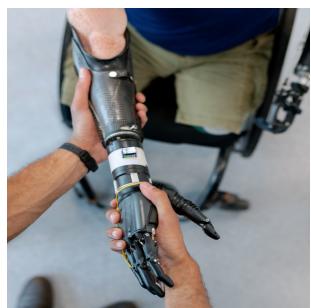
(a) Drone



(b) Perseverance Mars



(c) Nao



(d) Brazo biónico



(e) Da Vinci [4]



(f) Roomba

Figura 1.1: Ejemplos de Robots

1.1. ROBÓTICA

Bajo el nombre robot agrupamos máquinas muy distintas desde robots de limpieza o cocina, robots industriales, incluso robots que salvan vidas. Es por esto que para diferenciarlos se dividen según el entorno de aplicación: robots industriales (brazos y pinzas para ensamblado de piezas) y robots de servicio (destinados a limpieza, coches autónomos, investigación en terrenos hostiles o fines medicos). También se pueden clasificar por su forma (los Androides y zoomórficos), si son fijos o móviles, terrestres, acuáticos o áereos.

Un robot es una máquina autónoma o semi-autónoma que es capaz de percibir su entorno , realizar ciertos cálculos para tomar decisiones, actuar en el mundo real de acuerdo con esas decisiones y comunicarse con otras máquinas o con humanos. Es una combinación de la electrónica, mecánica y software. Para todo ello utilizan:

- Sensores: para recibir información de su entorno (láser, láser ultrasonidos (Figura 1.2), infrarrojos, cámaras, radio frecuencia, sensores térmicos...).



Figura 1.2: Sensor ultrasonidos

- Actuadores: permiten la locomoción del robot por su entorno y manipulación de objetos (Motores eléctricos, de combustión (Figura 1.3), amortiguadores, hélices, patas, ruedas, cadenas...).



Figura 1.3: Motor

1.1. ROBÓTICA

- Controladores: Procesador (Figura 1.4) y algoritmos, para el cálculo y toma de decisiones.



Figura 1.4: Procesador

- Leds, Pantallas (Figura 1.5), Antenas, Altavoces...: para comunicarse con otros robots o con humanos.



Figura 1.5: Pantalla

Recientemente la robótica se está imporporando a las aulas para que los más pequeños adquieran conocimientos y habilidades muy 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 WWW²[7] ha ido evolucionado hasta poder implementar potentes aplicaciones con un modelo cliente/servidor. En la siguiente Figura 1.7 podemos ver algunos ejemplos de aplicaciones web.

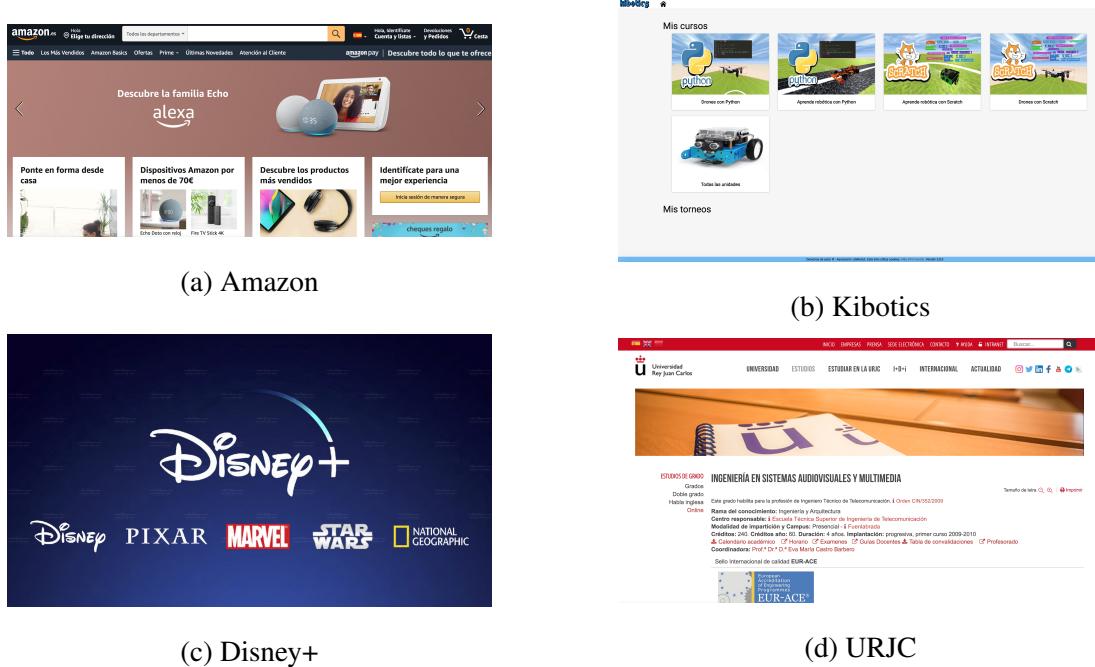


Figura 1.7: Ejemplos aplicaciones web

La web es una colección de documentos enlazados a través de hiperenlaces, cada recurso queda definido por su URL³. Cuando accedemos a la web a través del navegador, tenemos que introducir la URL del sitio web al que nos queremos dirigir. El navegador enviará una solicitud al servidor con el protocolo HTTP⁴. El servidor le enviará a nuestro navegador un fichero index.html que quedará almacenado en nuestra máquina. Una vez el navegador obtiene el fichero index.html y 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 como es la comunicación entre cliente y servidor.

²World Wide Web

³Uniform Resource Locator

⁴Hyper Text Transfer Protocol

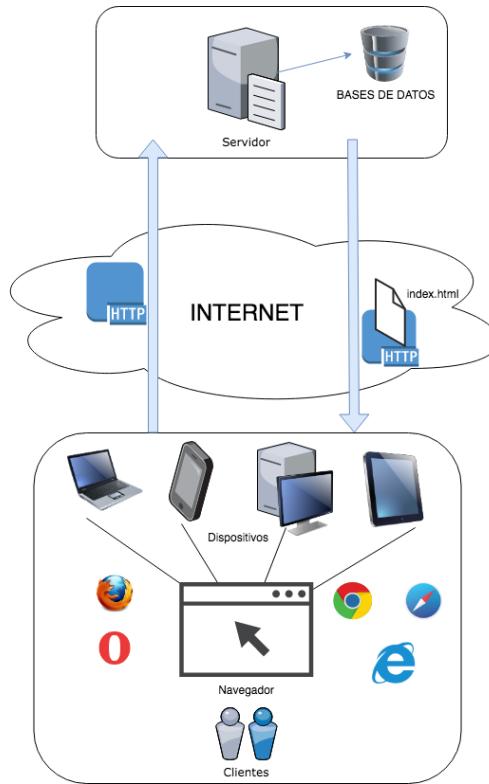


Figura 1.8: Ejemplo de una comunicación cliente/servidor a través de HTTP

HTTP es un protocolo entre navegadores y servidores web para transferir documentos hipertexto. El cliente envia menajes de solicitud y el servidor manda mensajes de respuesta, ambos mensajes son del mismo formato ver Figura 1.9 y Figura 1.10. Los métodos 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 por todos el famoso 404 Not Found cuando hay un error en la parte cliente. [8]

Línea inicial de petición
GET /dir/pagina.html HTTP/1.1
Host: www.uni.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: es

Línea inicial de respuesta
HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 ...
Content-Length: 6821
Content-Type: text/html
Líneas de cabecera
Línea en blanco
Cuerpo del mensaje
datos datos datos datos datos ...
datos datos datos datos datos ...
datos datos datos datos datos ...

Figura 1.9: Ejemplo petición HTTP
Cliente a Servidor

Figura 1.10: Ejemplo respuesta HTTP
Servidor a Cliente

1.2. TECNOLOGÍAS WEB

Las dos capas 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 donde el navegador actúa como interprete. 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, tablets, 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 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 o frontend permiten la interacción del usuario con la página web que corre en el navegador del usuario. Para ello se usan estas tres tecnologías [9]:

- **HTML5⁵**: 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.
- **CSS3⁶**: 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 el capítulo de Herramientas.

⁵HyperText Markup Language

⁶Cascading Style Sheets

1.2.2. Tecnologías Web lado servidor

Las tecnologías web del lado servidor o backend 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 cuatro de las más utilizadas:

- Python Django: es una estructura base 'framework' 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 (templates). De esta forma puedes centrarte en crear tu aplicación web sin grandes complicaciones. Es gratis y de código abierto.[10] Ejemplos de aplicaciones web que utilizan Django son Instagram [11] y Kibotics [1]
- 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].
- webRTC⁷: es un framework que permite las comunicaciones en tiempo real en el navegador. Incluye componentes fundamentales para las comunicaciones de alta calidad en la web. El servidor puede acceder a los componentes de audio y video a través de una API de JavaScript y permite una fácil implementación de la aplicación chat de voz o video. [14] Podemos destacar aplicaciones como Google Meets o Discord.
- PHP⁸: es un lenguaje de scripting de uso general popular que es especialmente adecuado para el desarrollo web.[15] Rápido, flexible y pragmático, gracias a su capacidad de creación de webs dinámicas desde blogs hasta los sitios web como Facebook o Wikipedia. [16]

En el lado servidor se utilizan bases de datos, entre ellas destacan MySQL y MongoDB.

⁷Web Real-Time Communication

⁸Hypertext Preprocessor

1.3. Gamificación

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 y mejorar 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.

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. Movimientos enfocados a conseguir evitar la recurrente pasividad de los alumnos.[17]

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. [19][18]

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 eficaces y facilitar el aprendizaje. Para que eso ocurra estos materiales tienen que ser de fácil instalación y uso, versátiles, mostrar información de calidad, que sean dinámicos e interactivos para motivar a los alumnos. Cada material se adapta a los usuarios para que trabajen a su ritmo.[20]

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

1.3. GAMIFICACIÓN

En los últimos años se están incorporando juegos en las aulas como la aplicación Kahoot! [21] 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. 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. [22]

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 más dinámica y divertida, incrementan sus habilidades a la vez que sus conocimientos desde los fundamentos STEM (Science, Technology, Engineering and Mathematics).

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

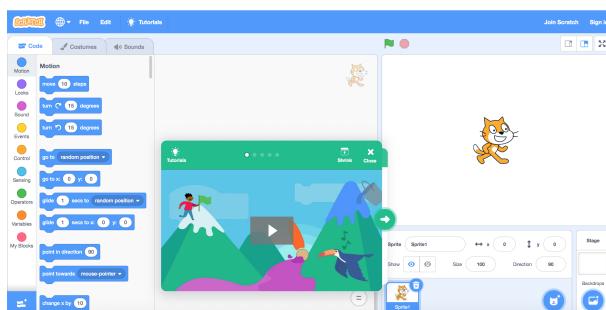


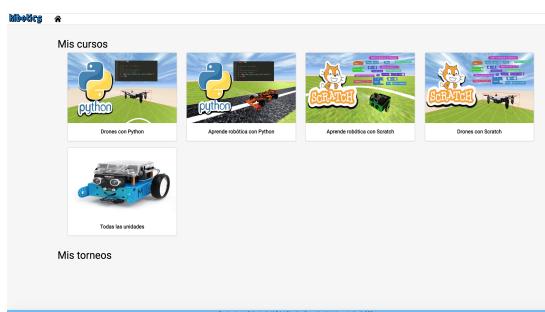
Figura 1.11: Scratch

1.3. GAMIFICACIÓN

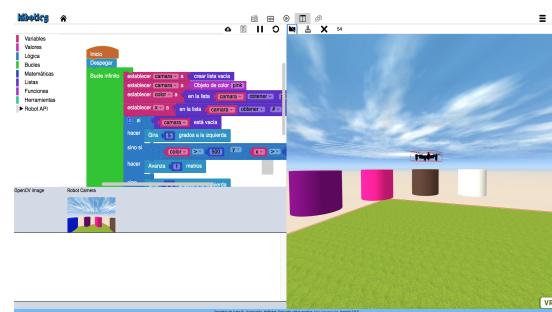
Scratch es un lenguaje de programación visual basado en bloques, creado y mantenido por Lifelong Kindergarten group en el MIT Media Lab. Scratch además es una comunidad en línea donde los niños pueden programar y compartir medios interactivos como historias, juegos y animaciones con gente de todo el mundo. Los más pequeños pueden aprender a pensar creativamente, trabajar en colaboración y razonar sistemáticamente.[23] Scratch posee un lenguaje de 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.

El mejor ejemplo de una plataforma educativa es Kibotics, plataforma base de este trabajo, ver Figura 1.13. Kibotics es un entorno web para docencia en robótica y programación (STEM). Enseña de manera atractiva conceptos básicos de tecnología (informática, electrónica, ...) e inicia a niños y adolescentes en robótica y programación de robots. Sigue un enfoque práctico, de aprender haciendo. Ayuda a estructurar el pensamiento, a organizar las acciones en pasos para resolver un problema y a formar espíritu analítico. [1]

Esta plataforma se basa en tecnologías web como Django para la parte servidor y el simulador Websim apoyado en A-Frame para representar los escenarios de los ejercicios en el navegador del usuario. En el capítulo de Herramientas se matizará un poco más en estas tecnologías.



(a) Cursos disponibles en Kibotics.



(b) Ejercicio busca objeto en Scratch.

Figura 1.12: Plataforma Kibotics.

Actualmente ofrece cursos de robótica en Scratch y Python. Estos cursos cuentan con ejercicios que fomentan el aprendizaje como el cuadrado, busca objeto, choca-gira y sigue líneas, así como distintos tipos de robots como Drones, F1, Pibot, Mbot, etc. A lo largo de este trabajo veremos como se introduce la Gamificación en esta plataforma.

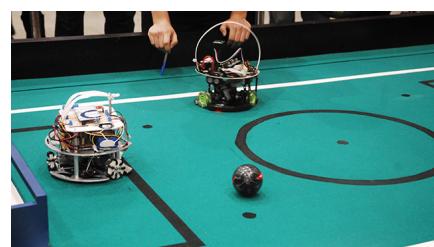
1.3.3. Campeonatos de robótica educativa existentes

El aprendizaje de la robótica educativa ha ido más allá y actualmente existen competiciones a nivel Nacional e Internacional estas son algunas de ellas [24]:

- RoboCup Junior⁹: 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 y Liga ONSTAGE. [25]
- Eurobot Junior¹⁰: 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, y pueden tener un robot secundario autónomo.[26]
- First Lego league¹¹: 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.[27]
- Torneo Nacional VEX Robotics IQ¹²: destinado a equipos de 4 a 8 miembros de secundaria junto un adulto. Ofrecen distintos retos y torneos con premios.[28]
- RoboCampeones¹³: 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.[29]



(a) Juego Pañuelo Robocampeones



(b) Fútbol en RoboCup Junior.

Figura 1.13: Ejercicios en campeonatos de robótica.

⁹<https://junior.robocup.org/>

¹⁰<http://www.eurobot.es/>

¹¹<https://www.firstlegoleague.es/>

¹²<https://vexspain.com/>

¹³<http://robocampeones.org/>

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:

- Desarrollar nuevos juegos que sirvan de herramienta de aprendizaje a niños y jóvenes dentro de la plataforma Kibotics.
- Introducir sonidos a la plataforma para mejorar la experiencia del usuario.
- Desarrollar un Teleoperador acústico.
- Ampliar el catálogo de robots existentes en Kibotics.
- Ampliar el API de los robots para que puedan absorber elementos del simulador y coger objetos con pinzas.

2.2. Requisitos

Para cumplir con los objetivos citados anteriormente debemos tener en cuenta 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). [38] Este modelo consiste en entregar a los usuarios y al equipo de administración 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. Para llevar a cabo 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 Slack¹.

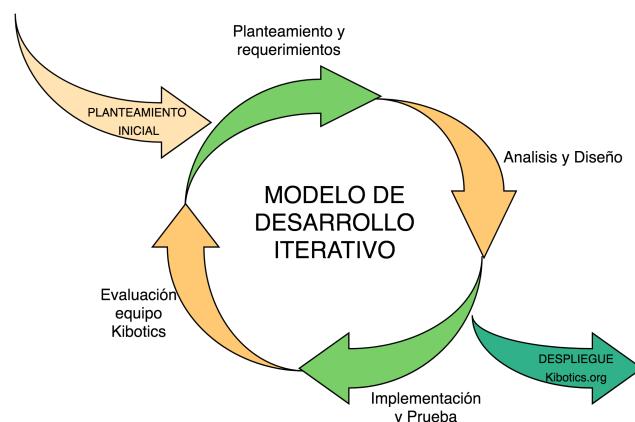


Figura 2.1: Modelo iterativo

¹<https://slack.com/>

2.4. PLAN DE TRABAJO

El código desarrollado y las mejoras se han integrado progresivamente en el código fuente oficial de Kibotics en GitHub² mediante su flujo de trabajo inciencia (issue), rama (branch) y parche (pullrequest). De esta forma la plataforma oficial esta siempre actualizada con los últimos cambios añadidos y problemas solventados y verificados por el equipo de Kibotics.

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

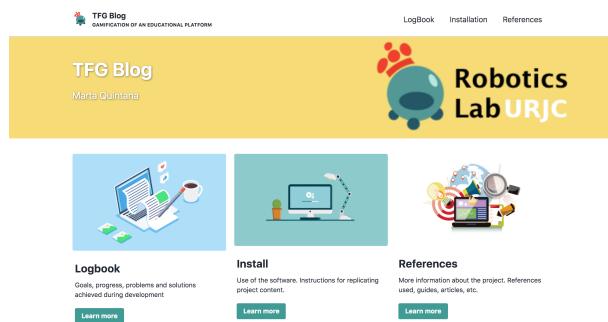


Figura 2.2: Página web de este TFG

2.4. Plan de Trabajo

El plan de trabajo seguido durante este proyecto se puede dividir en las siguientes siete 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:** Se han realizado una serie de ejercicios y tutoriales para introducir Sonido y reconocimiento de audio en Kibotics.
- **Diseño Teleoperador Acústico con Teachable Machine:** creación de un teleoperador acústico con reconocimiento de audio.
- **Planteamiento Gamificación :** Una vez realizado el teleoperador acústico se planteó la gamificación, hicimos una lista de juegos posibles en la plataforma desde ejercicios compartidos de varios jugadores, nuevos robots y juegos distintos a los que ofrecían.

²<https://github.com/>

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

2.5. ESTRUCTURA DEL DOCUMENTO

Finalmente se optó por la realización de dos juegos Roomba atrapa confeti y el juego del pañuelo con Mbot Pinza.

- **Diseño ejercicio Roomba atrapa confetti:** Se planteó y desarrolló un Roomba simulado junto con el actuador de absorción de confeti.
- **Prototipo Mbot Pinza o Gripper con A-Frame.:** Diseño del prototipo de un robot con pinzas capaz de coger objetos y moverlos.
- **Diseño y desarrollo final ejercicio Mbot Pinza:** Se obtó por crear un Mbot Pinza en Kibotics con Blender y JavaScript, que nos ofrecía físicas más realistas.

2.5. 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 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 la infraestructura, enunciado y solución de referencia de cada ejercicio:

- **Capítulo 4 Ejercicio: Teleoperador Acústico y Banda Sonora**
- **Capítulo 5 Ejercicio: Roomba 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.
- **Bibliografía**

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. También presentamos en gran detalle la plataforma Kibotics, su estructura y todas las tecnologías web que utiliza como el simulador Websim.

3.1. Lenguajes de programación y documentos

3.1.1. HTML5

3.1.2. JavaScript

JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Es un lenguaje que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica 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 al

3.2. HERRAMIENTAS

usuario.[39]

3.1.3. Python y Scratch

3.1.4. JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generararlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo. Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias. Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

3.2. Herramientas

3.2.1. TensorFlowJS, Web Audio API y Teachable Machine

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

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 (como paneo) y mucho más.

Teachable Machine es una herramienta basada en la Web que hace posible crear modelos de

3.2. HERRAMIENTAS

aprendizaje automático de manera rápida, sencilla y accesible para todos.

3.2.2. Blender

Blender es un software de creación 3D gratuita y de código abierto. Es compatible con la totalidad de la canalización 3D: modelado, montaje, animación, simulación, renderizado, composición y seguimiento de movimiento, edición de vídeo y canalización de animación 2D.

3.2.3. A-Frame

A-Frame es un marco web para crear experiencias de realidad virtual (VR). A-Frame se basa en HTML, lo que facilita el inicio. Pero A-Frame no es solo un gráfico de escena en 3D o un lenguaje de marcado; el núcleo es un poderoso marco entidad-componente que proporciona una estructura declarativa, extensible y componible a three.js.

3.2.4. Plataforma Kibotics

3.2.5. Websim

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-webtecnologías-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] “Frequent Questions — What is WebRTC.” <https://webrtc.github.io/webrtc-org/faq/what-is-webrtc> (accessed Apr. 18, 2021).
- [15] “¿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-cu00803bccatid=70Itemid=193 (accessed Apr. 18, 2021).
- [16] “Top 10 websites built with PHP technology - Facebook, Yahoo...” <https://cybercraftinc.com/blog/top-10-projects-developed-with-php-technology> (accessed Apr. 18, 2021).
- [17] “La importancia del contenido multimedia en educación.” <https://www.cursosfemxa.es/blog/14089-la-importancia-del-contenido-multimedia-en-educacion> (accessed Apr. 18, 2021).
- [18] E. Andrade and E. Chacón, “Implicaciones teóricas y procedimentales de la clase invertida,” *Pulso*, vol. 41, pp. 251–268, 2018.
- [19] “El Vídeo Educativo como recurso dinamizador del Aprendizaje - EVirtualplus.” <https://www.evirtualplus.com/video-educativo-como-recurso-aprendizaje/> (accessed Apr. 20, 2021).

BIBLIOGRAFÍA

- [20] “La Importancia Del Contenido Multimedia En La Educación.” <https://es.calameo.com/read/005984440640dded50f70> (accessed Apr. 18, 2021).
- [21] “Play Kahoot! - Enter game PIN here!” <https://kahoot.it/> (accessed Apr. 18, 2021).
- [22] “Robótica educativa: ¿qué es y cuáles son sus ventajas?” <https://www.unir.net/educacion/revista/robotica-educativa/> (accessed Apr. 20, 2021).
- [23] “Scratch - Imagine, Program, Share.” <https://scratch.mit.edu/> (accessed Apr. 18, 2021).
- [24] “Competiciones de robótica educativa a los que apuntarse.” <https://blog.juguetronica.com/competiciones-de-robotica-educativa/robocampeones> (accessed Apr. 18, 2021).
- [25] “RoboCupJunior – Creating a learning environment for today, fostering technological advancement for tomorrow.” <https://junior.robocup.org/> (accessed Apr. 20, 2021).
- [26] “EUROBOT JR.” <http://www.eurobot.es/index.php/eurobot-jr> (accessed Apr. 20, 2021).
- [27] “FIRST LEGO League.” <https://www.firstlegoleague.es/> (accessed Apr. 20, 2021).
- [28] “Vex Spain — Torneos.” <https://vexspain.com/vex-iq/torneos/> (accessed Apr. 20, 2021).
- [29] “Robocampeones” <http://robocampeones.org/> (accessed Apr. 20, 2021).
- [30] Imágenes de: “Pexels.” <https://www.pexels.com/es-es/> (accessed Apr. 18, 2021).
- [31] Imágenes de :“ Pixabay.” <https://pixabay.com/es/> (accessed Apr. 18, 2021).
- [32] “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).
- [33] “Tecnologías para el desarrollo web más actuales — proun Madrid - Asturias.” <https://www.proun.es/blog/tecnologias-web-actuales/> (accessed Apr. 18, 2021).
- [34] “Conceptos básicos sobre tecnologías de desarrollo web - ingeniovirtual.com.” <https://www.ingeniovirtual.com/conceptos-basicos-sobre-tecnologias-de-desarrollo-web/> (accessed Apr. 18, 2021).

BIBLIOGRAFÍA

- [35] “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).
- [36] “Así se enseña robótica y programación en las aulas españolas.” <https://www.educaciontrespuntocero.com/noticias/robotica-y-programacion-espana/> (accessed Apr. 18, 2021).
- [37] “historia de robotica - el mundo de la robotica 604.” <https://sites.google.com/site/elmundodelarobotica604/historia-de-robotica> (accessed Apr. 18, 2021).
- [38] “Modelos de desarrollo de software” <https://www.elconspirador.com/2013/08/19/modelos-de-desarrollo-de-software/> (accessed Apr. 25, 2021).
- [39] J. Eguílez Pérez, “Introducción a JavaScript.” Accessed: Apr. 26, 2021. [Online]. Available: www.librosweb.es.