



GRADO EN INGENIERÍA TELEMÁTICA

Curso Académico 2020/2021

Trabajo Fin de Grado

# Integración del Robot Lego Ev3 a la plataforma de Kibotics

Autor : Daniel Pulido Millanes

Tutor : Dr. José María Cañas Plaza



# Índice general

<b>Lista de figuras</b>	<b>5</b>
<b>Lista de tablas</b>	<b>7</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Robótica . . . . .	1
1.1.1. Aplicaciones robóticas . . . . .	3
1.1.2. Software en robótica . . . . .	6
1.2. Robótica educativa . . . . .	7
<b>2. Objetivos</b>	<b>11</b>
2.1. Objetivos del TFG . . . . .	11
2.2. Metodología . . . . .	12
2.3. Requisitos . . . . .	13
2.4. Plan de trabajo . . . . .	14



# Índice de figuras

1.1. Imagen clásica de un robot . . . . .	2
1.2. Aspiradora robótica <i>Roomba</i> . . . . .	4
1.3. Robot médico <i>Da Vinci</i> . . . . .	4
1.4. Robot militar <i>Big Dog</i> creado por <i>Boston Dynamics</i> . . . . .	5
1.5. Vehículo <i>Waymo</i> de <i>Google</i> . . . . .	6
1.6. Robot <i>Perseverance</i> de la <i>NASA</i> . . . . .	6
1.7. Interfaz gráfica de Scratch . . . . .	8
1.8. Interfaz de LEGO WeDo . . . . .	9
1.9. Interfaz gráfica de LEGO Ev3 . . . . .	9
1.10. Kit de piezas y sensores de LEGO Ev3 . . . . .	10
2.1. Metodología de modelo iterativo . . . . .	13



# **Índice de cuadros**





# Capítulo 1

## Introducción

En este capítulo se introducen los conceptos básicos en robótica, de como esta nos ayuda en nuestro día a día, y cual es su estado actual. Y como puede ser un gran recurso en la educación. En lo que se basa este proyecto

### 1.1. Robótica

La robótica es una rama de las ingenierías y de las ciencias de la computación que se encarga del diseño, construcción, operación, estructura, manufactura y aplicación de los robots. El término *robot* se popularizó con el éxito de la obra R.U.R. (*Robots Universales Rossum*), escrita por Karel Čapek en 1920. En la traducción al inglés de dicha obra la palabra checa *robota*, que significa trabajos forzados o trabajador, fue traducida al inglés como robot. Un robot es una entidad virtual o mecánica artificial. Están diseñados con un proposito propio. La independencia creada en sus movimientos hace que sus acciones sean la razón de un estudio razonable y profundo en el área de la ciencia y tecnología. La palabra robot puede referirse tanto a mecanismos físicos como a sistemas virtuales de software, aunque suele aludirse a los segundos con el término de bots.



Figura 1.1: Imagen clásica de un robot

No hay un consenso sobre qué máquinas pueden ser consideradas robots, dentro de este proyecto tomaremos como definición que un robot es un sistema autónomo programable capaz de realizar tareas complejas. Además, todos los robots se componen de tres partes esenciales se componen de sensores, controladores y actuadores.

- **Sensores:** Son los sentidos del robot, con ellos ve, escucha y sabe lo que hay en el entorno. Recogen la información necesaria para que el robot realice la tarea En este grupo se encuentran láseres, cámaras, ultrasonidos u odómetros..

- **Controladores:** El equivalente al cerebro humano, utiliza los datos recogidos por los sensores para elaborar una respuesta para que la lleve a cabo los actuadores.
  
- **Actuadores:** Equivalen a los músculos humanos, son los que se encargan de interactuar con el entorno para llevar a cabo su tarea. Son brazos mecánicos, motores, etcétera...

### 1.1.1. Aplicaciones robóticas

Ahora que tenemos las bases de lo que es un robot asentadas podemos hablar de cuales son los principales propósitos de los robots hoy en día, aunque la mayor parte de ellos son utilizados por empresas en labores industriales. Aunque hay otros que podemos encontrar en nuestra vida cotidiana, en casas, hospitales, almacenes de tiendas... Esto es debido a la precisión de algunos trabajos, la eficiencia en el trabajo, la reducción de costes que supone o que pueden realizar acciones de alto riesgo para las personas. Los ejemplos mas famosos en estos campos son los siguientes:

- **Robots Domésticos:** Creados para realizar las tareas del hogar. Los mas famosos y destacados en el mercado son los Robots *Roomba*, aspiradores autónomos, y también el primer robot que se ha comercializado para todos los públicos y de manera global. Un gran paso para la robótica



Figura 1.2: Aspiradora robótica *Roomba*

- Robots médicos: Son robots diseñados para el uso en medicina para realizar tareas que requieren mucha precisión como en el caso de una cirugía, con el robot *Da Vinci* o robots diminutos que son capaces de navegar por las venas hasta llegar al corazón y allí realizar la cirugía necesaria.



Figura 1.3: Robot médico *Da Vinci*

- Robots militares: Son robots orientados a tareas militares, como reconocimientos de zo-

nas conflictivas o rescate de personas, desactivación de bombas. En los últimos años también se han desarrollado mucho los drones en combate.



Figura 1.4: Robot militar *Big Dog* creado por *Boston Dynamics*

- Vehículos autónomos: Es el campo de la robótica que más en auge está ahora mismo. El objetivo de estos robots es usar la información que proporcionan sus sensores internos, como cámaras, sensores infrarrojos *Lidar*, y sensores externos como el GPS para llevar de un punto a otro un vehículo.



Figura 1.5: Vehículo *Waymo* de *Google*

- Robots Espaciales: Los famosos *Rover* de la *NASA* son robots diseñados para entornos donde el ser humano no puede llegar. Se centran en reconocimiento del terreno y análisis de las muestras que recogen.

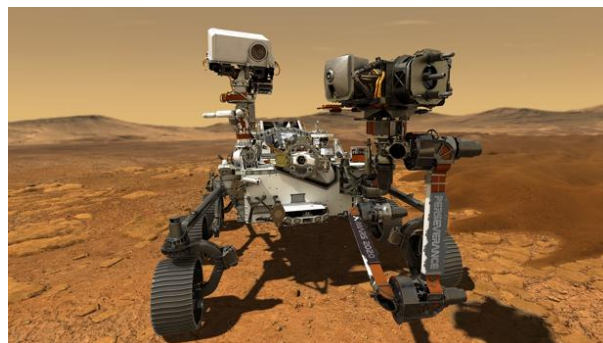


Figura 1.6: Robot *Perseverance* de la *NASA*

### 1.1.2. Software en robótica

Para dotar de esta inteligencia a los robots se necesitan herramientas que transformen los datos recibidos de los sensores en algo que puedan aplicar en los actuadores. Hace años, cada máquina tenía un software específico con sensores y actuadores únicos para ese robot y esa tarea a desarrollar. Esto hacía, que aunque hubieras implementado el software para otros robots anteriormente, tuvieras que repetir el proceso con cada nuevo robot. Con los años se desarrollaron plataformas de software que permiten desarrollar de manera genérica para todos los robots, y actuando de mediador entre el robot y el software del creador, estos son los llamados *middleware*

que hacen que te puedas abstraer de los *drivers* característicos de cada robot. Los *middleware* mas importantes a día de hoy son:

- **Robot Operating System (ROS)**[?]. Plataforma de *software* libre para el desarrollo de *software* de robots. Provee servicios estándar de un sistema operativo como la abstracción de *hardware*, control de dispositivos de bajo nivel, mecanismos de intercambio de mensajes entre procesos y mas herramientas vitales para el desarrollo del robot. Es el mas utilizado a día de hoy porque fue especialmente desarrollado para *UNIX* y luego se implemento para el resto de sistemas operativos
- **ORCA**[?]. Plataforma de *software* libre diseñado para crear aplicaciones mas complejas, ya que esta orientado a las componentes por separado
- **OROCOS**[?] Proyecto de *software* libre también orientado a componentes y basado en C++
- **JdeRobot**[?] Plataforma de desarrollo robótico, en la que se basa este proyecto. Tiene varios nodos programados con varios lenguajes de programación, con compatibilidad con otros *middleware*.

## 1.2. Robótica educativa

La robotica educativa ha ido tomando mas importancia con los años, ya que cada vez es mas importante que estudiantes de cualquier nivel estén familiarizados con la tecnología, tiene valores positivos como la implementación de pensamiento lógico, resolución de problemas y trabajo en equipo en las actividades académicas, que son ramas del conocimiento que se desarrollan poco en edades tempranas , con una componente en conocimiento matemático y físicos y ademas añade un atractivo que no tienen las asignaturas convencionales. Muchos estudios han demostrado que el uso de kits de robótica en la educación favorece a la capacidad de reflexión de los estudiantes. Cada año se crean mas cursos de robótica, y en 2015 la comunidad de Madrid introdujo la asignatura de robótica en los planes docentes de Enseñanza Secundaria con la asignatura “Tecnología, Programación y Robótica”[?] y en el curso 2020-2021 se empezará a implantar en Educación Primaria la asignatura “Programación y Robótica”[?].



Una de las mayores partes de la robótica tiene que ver con la programación, que además de ser una habilidad muy importante para la sociedad actual, es algo complejo. Por lo que se utilizan lenguajes de programación visual, estos se tratan de lenguajes que abstraen en bloques las funciones o métodos de cualquier lenguaje de programación. Dentro de este tipo de lenguajes, los más destacables son :

- **Scratch**[?]: proyecto liderado por el Grupo *Lifelong Kindergarten* del MIT, es utilizado por estudiantes para programar animaciones, juegos e interacciones. Su atractivo reside en lo fácil que es de entender el pensamiento computacional debido a su sencilla interfaz gráfica y la implementación de sus bloques.



Figura 1.7: Interfaz gráfica de Scratch

- **LEGO**[?]: Es el robot base de este proyecto, dispone de una amplia gama de robots programables y cada uno de ellos tiene un sistema gráfico, que es similar entre ellos pero también ligado a la edad del estudiante para el que está diseñado el software.



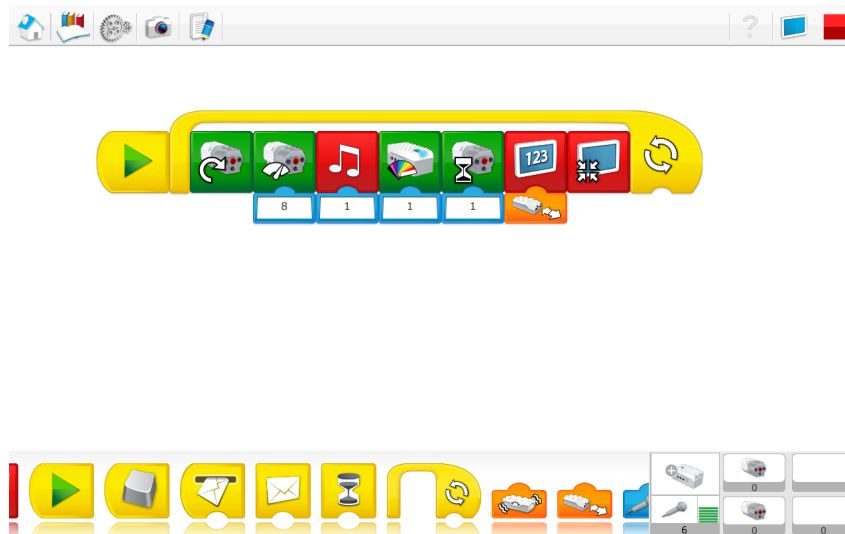


Figura 1.8: Interfaz de LEGO WeDo

Por ejemplo en la figura 1.8 se puede ver que la interfaz en este caso, es con colores vivos, los cuales representan distintas funcionalidades dentro del robot, es decir, el amarillo representa las acciones propias de programación, como: inicio de programa, fin de programa, bucles, esperar, etcétera. El color rojo representa los sensores del robot, todo lo que recoja datos. Y el color verde representa los motores que equivalen a los actuadores en este robot. Como se puede observar es una abstracción muy simple para estudiantes de mas corta edad.

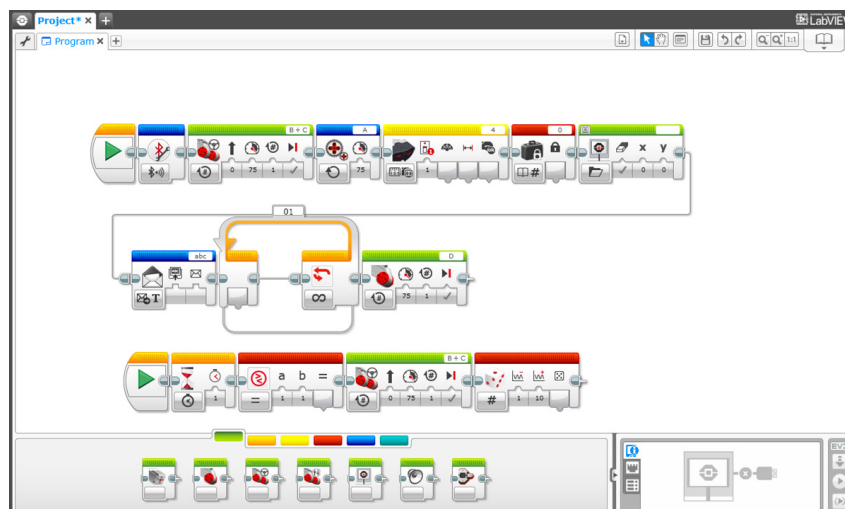


Figura 1.9: Interfaz gráfica de LEGO Ev3

En el caso del software para el **LEGO Ev3**, añade un grado de complejidad, incluyendo

apartados para realizar operaciones matemáticas, envío de archivos entre robots, y además actuadores, como la pantalla que integra el robot, o los altavoces.

En el caso de LEGO y en otros kits incorporan los elementos básicos para la construcción de un robot. En este en particular viene con lo indispensable para construir con piezas de LEGO. También incluye un microprocesador para ser programado, con Linux instalado, sensores (infrarrojos, táctiles y de color) y motores.



Figura 1.10: Kit de piezas y sensores de LEGO Ev3

En el siguiente capítulo, profundizaremos más en lo que se puede hacer con el robot **LEGO Ev3** y explicaremos cuáles van a ser los objetivos y por qué elegir este robot.

# Capítulo 2

## Objetivos

Una vez explicado en el ámbito en el que se realiza este proyecto, en este capítulo explicaremos los objetivos que se han tratado de alcanzar y el método de trabajo que se ha seguido para lograrlo

### 2.1. Objetivos del TFG

El objetivo de este trabajo es dar soporte en la plataforma de *Kibotics*<sup>1</sup> a un robot bastante popular para la robótica educativa como es el *LEGO Ev3*, esto significa integrarlo de forma que se pueda programar dentro de la plataforma, y que también funcione para el robot real.

#### Soporte Simulado

- Añadir una simulación 3D realista sabiendo que hay modelos de robots prediseñados por *LEGO* de varios modelos de robots, al menos uno por cada tipo de sensor que pueda llevar. Y que tenga sentido físico dentro de la simulación.
- Añadir la infraestructura necesaria como *drivers*, y funciones al *Robot API* para que el robot se programe en cualquier lenguaje soportado por la plataforma, y funcione en el robot real
- Crear un conjunto de ejercicios para que haya un temario fácil de seguir por el estudiante, y con una curva de dificultad moderada

---

<sup>1</sup><https://kibotics.org/>

### Soporte Real

- Instalar una imagen de un sistema operativo en el *Lego ev3* en este caso, una distribución basada en *Debian Linux*.
- Instalar un servidor en el robot capaz de recibir mensajes con el código, y lo transforme en un archivo y lo ejecute dentro de la máquina.

## 2.2. Metodología

La metodología para completar el trabajo de fin de grado se puede dividir en diferentes fases que se iban repitiendo cada cierto tiempo, en cada una de ellas, semanalmente, tenía lugar una reunión con el tutor del trabajo para determinar los siguientes objetivos a cumplir y evaluar las tareas propuestas en anteriores sesiones. Esto ayuda mucho en proyectos como este, en constante desarrollo.

Este proyecto se lleva a cabo con un equipo de trabajo, que se ocupa de la plataforma de *Ki-botics*, cada uno con sus labores y ocupaciones. Por lo tanto, es necesaria la comunicación y realimentación con el resto de integrantes. Para ello se utiliza la herramienta *Slack*<sup>2</sup> en la que los desarrolladores están en contacto en todo momento, no solo para comunicar avances, si no también para ayudar en todo momento si surge algún contratiempo en el desarrollo.

Para trabajar en local, con el repositorio original me hice *git clone* de los repositorios que necesitaba, e iba trabajando sobre ellos, guardando los cambios en un repositorio creado solo para el trabajo de fin de grado<sup>3</sup> hasta que la funcionalidad estaba completa, que era cuando se subían al repositorio principal. Esta metodología, es el llamado modelo iterativo de desarrollo de software.

---

<sup>2</sup><https://slack.com/>

<sup>3</sup><https://github.com/RoboticsLabURJC/2020-tfg-daniel-pulido>

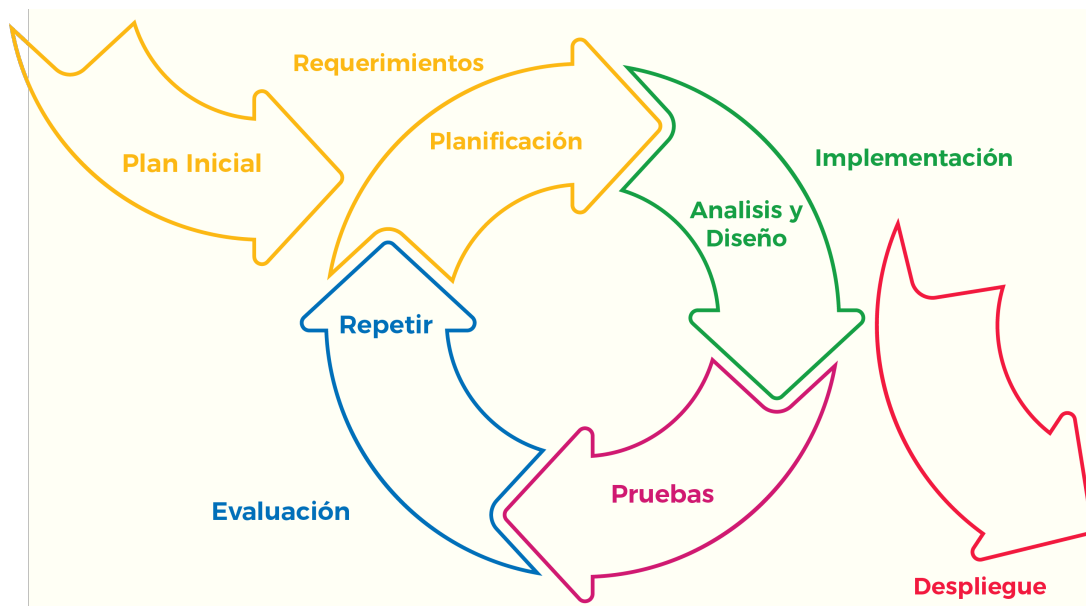


Figura 2.1: Metodología de modelo iterativo

Cuando los cambios eran revisados por el tutor, se seguía una dinámica de *Incidencia y parche* en el repositorio principal. Para ello se creaba una incidencia (*issues*) con el tema que se iba a solucionar o añadir en la funcionalidad de la plataforma, y una vez resuelto en local y para cerrarla, se creaba una rama (*branch*) creando parche (*pull request*) para que un desarrollador principal de *Kibotics*, lo aceptará y fusionará con la rama principal y así arreglar la incidencia. Esto se hace para registrar todos los cambios, y comprobarlos antes de que se trabaje directamente en el repositorio

## 2.3. Requisitos

Para completar la integración del robot **LEGO Ev3** se necesitan ciertos requisitos que cumplir:

- Dentro del **LEGO Ev3** debe correr una distribución de *Linux*, en este caso he optado por la imagen creada por *ev3dev*, que es una distribución basada en el sistema operativo *Debian Linux*, de la cual entrare en mas detalle en el Capitulo 5: *Soporte Lego ev3 físico*.
- El resultado final debe ser lo más sencillo posible, no debe requerir configuraciones adicionales. Este software tiene que estar diseñado para estudiantes.

## 2.4. Plan de trabajo

El plan de trabajo a seguir para conseguir el objetivo se puede dividir en los siguientes pasos:

- *Paso 1:* Aterrizaje en *Kibotics*. Lo primero que hay que hacer es familiarizarse con el entorno con el que se va a trabajar. *Kibotics* es una plataforma web en la que entraremos mas en detalle en el siguiente capitulo
- *Paso 2:* Comienzo de la creación del robot simulado. Comenzaremos por la creación de varios modelos 3D para usarlos en el entorno simulado, son varios diferentes ya que lo original que tiene *LEGO* es poder construir tu robot en base a la tarea que vaya a realizar, con las piezas que incluye el kit .
- *Paso 3:* Desarrollo de las funciones y ejercicios dentro de la aplicación de *Kibotics* para crear una dinámica de trabajo con el nuevo robot.
- *Paso 4:* Dar soporte al robot real para poder programarlo en *python* desde el exterior, y la instalación de un server para que pueda recibir código y ejecutarlo en local.
- *Paso 5:* Creación de los *drivers* que hagan de traductor entre lo que creas en la plataforma, y lo que entiende el robot.