Subsecretaria de Educación Superior

Dirección General de Educación Superior Tecnológica

Instituto Tecnológico de Orizaba

**TECNOLÓGICO NACIONAL DE MÉXICO**

**INSTITUTO TECNOLÓGICO DE ORIZABA**

RESIDENCIAS PROFESIONALES

**PROYECTO:**

“Kubeet IOT – Robot Autónomo”

**AUTORES:**

BENITEZ PEREZ MAURICIO

**ASESOR:**

M.C. David Bertani Hernández

**FECHA DE REVISIÓN:**

00/00/2017

**INGENIERÍA ELECTRÓNICA**

Avenida Oriente 9 Núm. 852, Colonia Emiliano Zapata

C.P. 94320 Orizaba, Veracruz, México

Teléfonos (272) 7 24 40 96, (272) 7 24 40 16, (272) 7 24 45 79

Fax: (272) 7 25 17 28

E-mail: orizaba@itorizaba.edu.mx www.itorizaba.edu.mx



# CAPÍTULO 1: PRELIMINARES.

# **1.1 AGRADECIMIENTOS.**

## **1.2 RESUMEN.**

Un robot autónomo es un [robot](https://es.wikipedia.org/wiki/Robot) que puede operar con un alto grado de autonomía, lo que es particularmente deseable en campos como la exploración espacial, tratamiento de aguas residuales y tareas que puedan resultar pesadas o tediosas para las personas, como limpieza de suelos, cortar el césped, etc.

De esta manera se programa un robot autónomo, utilizando como base de armado el robot de Lego Mindstorm EV3 el cual tiene como inicio un sistema operativo de linux, programable con un software propio de Lego/Labview que es una forma interativa de bloque programables paresido al Labview normal, de igual forma Lawview tiene una librería especial para la programacion de el EV3.

Inicialmente se empezo con una capacitacion aprendiendo a moverse en el sistema Linux (creando una maquina virtual para este punto), asi como el uso de la plataforma GitHub y su forma de usarlo en el sistema Linux, tambien el aprendizaje de los lenguajes de programacion que se pueden utilizar en el robot siendo estos Python y C++.

Introduciéndose en el software de lego para la programacion que cuenta con varios tiempos de “bloques” de programacion del EV3, siendo estos divididos en 6 categorias siendo bloques de accion, flujo, de sensores, de datos, bloques avanzados y Mis bloques (el cual tiene como funcion guardar los segmentos que se repitan contantemente en los programas/proyectos).

Como se va avanzando en el proyecto se necesita cambiar el sistema interno de el EV3 por otro, siendo la eleccion de EV3Dev que es un Debian/Linux basado en el sistema operativo que se ejecuta em varias plataformas, incluyendo LEGO MINDSTORMS EV3 y Raspberry Pi Powered BrickPi. Este sistema tiene como objetivo el uso de varios lenguajes de programacion de los cuales se destacan el lenguaje C++, Python, etc.

El proyecto culminará con la comunicación de robot con la plataforma kubeet IoT, la cual mostrará cada dato que el robot le envíe y también guardando los mismo en la plataforma para chequeo ó monitoreo de cada sensor del robot.

**1.3 ÍNDICE.**

# CAPÍTULO 2: GENERALIDADES DEL PROYECTO.

## 2.1 INTRODUCCIÓN.

El internet de las cosas (IoT) actualmente esta creciendo en varias plataformas que se especializan en compartir información interactuando conlas personas que no tienen tiempo de esta vigilado ciertos apectos de su vida sea el monitoreo de dispositivo electrónico, sistemas mecanicos, medios que tengan que cuentan con una dirección IP, tales motivos son los que se formas plataformas que logren adtarce ala necesidades de las personas.

La interacción entre humano-maquina llega hacer uno de los principales motivos que el IoT tiene en cuenta llaque se requiere para algunas maquinas tales como brazos roboticos que estan eleborando una tarea que se requiere una supervición aun que no se este en contacto con ellos, formas llevar un control sobre el crecimiento de las plantas en un invernadero, capasidad de comunicarse con drones que son capases de deterninar su ubicación por GPS y enviar imágenes cantadspor camaras, o bien el monitoreo de un robot autonomo que sea capas de medir las varibles del ambiente sea temperatura, humeda, la incidencia de luz ambiental o poder mandar imágenes de su entorno todo en tiempo real.

Los aspectos del IoT van evolucionando, mejorando día a día para poder fasilitar la comunicación de las personas con la información que deceen.

## 2.2 Descripción de la empresa u organización y del puesto o área del trabajo del estudiante.

KUBEET S. DE R.L DE C.V. / Calle Sur 2, No. 225, interior 3, entre Poniente 5 y 7, CP. 94300, Centro, Orizaba, Veracruz.

Es una empresa enfocada en el área de desarrollo de la plataforma de internet de las cosas (Kubeet IoT).

área de Desarrollo Electrónico, enfocada en la implementación del conocimiento eletrónico para el desarrollo de diversos sistemas y aplicaciones que resuelvan determinados problemas.

## 2.3 PROBLEMA A RESOLVER.

En la actualidad se requiere que las personas, equipos electrónicos, robots, drones, equipo médico, artículos de inventario, animales domésticos, ganado, maquinaria, autos, casas, edificios, entre muchas otras estén conectados a internet e informen su situación actual, como ubicación GPS, temperatura, estado en que se encuentra, avisos de mantenimientos preventivos, emergencias entre otros.

Existen plataformas para el internet de las cosas (IoT, Internet of Things) como Amazon IoT, Azure IoT, Intel, por mencionar las más importantes, que ya permiten conectar y comunicar cualquier dispositivo a la nube pública por un costo muy bajo. Sin embargo la información de las “cosas” debe ser subida a nubes públicas y debido a la latencia de la conectividad de la red no se garantiza el tiempo real que muchas aplicaciones requieren.

Para la plataforma de kubeet IoT, un robot tiene que ser capas de enviar datos de cada uno de sus sensores, la distacia que recore, asi como señalar si da vuelta ó otros tipos de datos que sean requeridos, todo en un tiempo real.

## 2.4 OBJETIVOS.

**2.4.1 OBJETIVO GENERAL.**

Diseñar un robot autónomo con base al Lego Mindstorm EV3, con conectividad a la plataforma de Internet de las cosas Kubeet IoT.

### **2.4.2 OBJETIVOS ESPECÍFICOS.**

1. Conocer el concepto de internet de las cosas, sus aplicaciones reales y su futuro.
2. Conocer a moverse en el entorno Linux y GitHub.
3. Conocer los lenguajes de programación Python y C++.
4. Analizar la conectividad que brinda la plataforma Kubeet IoT.
5. Uso de un lenguaje de programación adecuado para la programación del robot.
6. Enviar los datos recolectados por el robot a la plataforma de kubeet IoT, para que puedan ser monitoreados remotamente por un usuario.

## 2.5 JUSTIFICACIÓN.

El internet de las cosas es la conexión de objetos de uso habitual o cotidiano a internet. La finalidad es la interconexión de diferentes objetos que utilizamos con cierta frecuencia, con el objetivo de hacernos la vida más fácil, por ejemplo, en la administración eficiente de la energía o en potenciar nuestra propia seguridad.

Las empresas públicas o privadas, hospitales, manufactureras, centros de investigación y universidades requieren una plataforma privada para controlar sus dispositivos que le garantice tiempo real en el Bigdata que generan los sensores.

El internet de las cosas se basa en sensores, en redes de comunicaciones y en una inteligencia que maneja todo el proceso y los datos que se generan. Los sensores son los sentidos del sistema y, para que puedan ser empleados de forma masiva, deben tener bajo consumo y costo, un reducido tamaño y una gran flexibilidad para su uso en todo tipo de circunstancias.

Un robot autónomo es un [robot](https://es.wikipedia.org/wiki/Robot) que pueden operar con un alto grado de autonomía, lo que es particularmente deseable en campos como la exploración espacial, tratamiento de aguas residuales y tareas que puedan resultar pesadas o tediosas para las personas, como limpieza de suelos, cortar el césped, etc.

Algunos modernos [robots industriales](https://es.wikipedia.org/wiki/Robot_industrial) son "autónomos" dentro de los límites estrictos de su entorno directo. Puede que no existan todos los grados de libertad en su entorno, pero el lugar de trabajo del robot industrial es difícil y, a menudo puede contener variables caóticas, impredecibles. Deberá determinarse la orientación exacta y la posición del siguiente objeto de trabajo y (en las fábricas más avanzadas) incluso el tipo de objeto y la tarea requerida. Esto puede variar impredecible (por lo menos desde el punto de vista del robot).

Los robots autónomos todavía requieren un mantenimiento regular, al igual que otras máquinas, algunos de ellos pueden auto-recargarse.

# CAPÍTULO 3: MARCO TEÓRICO.

**3.1 El internet de las cosas (IoT, Internet of Things).**

Kevin Ashton, cofundador y director ejecutivo del Centro de Auto-identificación del MIT, mencionó por primera vez Internet de las Cosas en una presentación que hizo a Procter & Gamble en 1999. He aquí cómo Ashton explica el potencial de Internet de las Cosas:

"Hoy en día las computadoras y, por lo tanto, Internet dependen casi totalmente de los seres humanos para obtener información. Casi todos los aproximadamente 50 petabytes (un petabyte es de 1.024 terabytes) de datos disponibles en Internet fueron capturados y creados por seres humanos escribiendo, pulsando un botón de grabación, tomando una imagen digital o escaneando un código de barras.

El problema es que la gente tiene tiempo limitado, atención y precisión, lo que significa que no son muy buenos para capturar datos sobre cosas en el mundo real. Si tuviéramos computadoras que supieran todo lo que hay que saber acerca de las cosas - utilizando datos que recopilaron sin ninguna ayuda de nosotros - seríamos capaces de rastrear y contar todo y reducir en gran medida la pérdida, la pérdida y el costo. Sabríamos cuando las cosas necesitaban reemplazar, reparar o recordar y si estaban frescas o pasadas lo mejor. "

Internet de las Cosas (IoT, por sus siglas en inglés) es un sistema de dispositivos informáticos interrelacionados, máquinas mecánicas y digitales, objetos, animales o personas que tienen identificadores únicos y la capacidad de transferir datos a través de una red sin requerir de una interacción humanos-humanos o humanos-computadora.

La Internet de las cosas se refiere a la creciente red de objetos físicos que cuentan con una dirección IP para la conectividad a Internet y la comunicación que se produce entre estos objetos y otros dispositivos y sistemas habilitados para Internet [1].

Las implicaciones para la Internet de las Cosas sobre la producción y el consumo son tremendas, y transformarán la forma en que la gente tienda, almacena y comparte los productos. El código de barras analógico que durante tanto tiempo ha sido una referencia cifrada muda a un sistema de inventario de tiendas, será reemplazado por una plataforma abierta en la que cada objeto fabricado podrá ser rastreado desde la cuna hasta la tumba, pasando por el fabricante hasta el distribuidor, hasta potencialmente cada persona que entra en contacto con ella después de su compra. Además, cada objeto que se acerca a otro objeto, y está dentro del alcance de un lector, también podría ser registrado en una base de datos y utilizado para encontrar correlaciones entre los propietarios, las condiciones ambientales y las aplicaciones [2].

Figura 1. Arquitectura del Internet de las cosas.

**3.1.1 GitHub.**

GitHub es una [forja](https://es.wikipedia.org/wiki/Forja_(software)) (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de [control de versiones](https://es.wikipedia.org/wiki/Control_de_versiones) [Git](https://es.wikipedia.org/wiki/Git" \o "Git). Utiliza el [framework](https://es.wikipedia.org/wiki/Framework" \o "Framework) [Ruby on Rails](https://es.wikipedia.org/wiki/Ruby_on_Rails) por GitHub, Inc. (anteriormente conocida como Logical Awesome). Desde enero de [2010](https://es.wikipedia.org/wiki/2010), GitHub opera bajo el nombre de GitHub, Inc. El código se almacena de forma [pública](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto), aunque también se puede hacer de forma privada, creando una cuenta de pago [3].

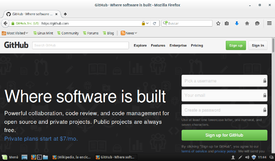
[](https://commons.wikimedia.org/wiki/File:P%C3%A1gina_principal_de_GitHub.png)

Figura 2. GitHub web.

**Características.**

* [Wiki](https://es.wikipedia.org/wiki/Wiki) para cada proyecto.
* [Página web](https://es.wikipedia.org/wiki/P%C3%A1gina_web) para cada proyecto.
* Gráfico para ver cómo los desarrolladores trabajan en sus repositorios y [bifurcaciones](https://es.wikipedia.org/wiki/Bifurcaci%C3%B3n_(desarrollo_de_software)) del proyecto.
* Funcionalidades como si se tratase de una [red social](https://es.wikipedia.org/wiki/Red_social), como por ejemplo: seguidores;
* Herramienta para [trabajo colaborativo](https://es.wikipedia.org/wiki/Trabajo_colaborativo) entre programadores.
* Gestor de proyectos de estilo Kanban.

**3.1.2 kubeet iot.**

La idea de la creación y desarrollo de la plataforma Kubeet IoT, surgió a partir del reciente auge del Internet de las Cosas, donde las “cosas” interactúan con otras “cosas” creando modelos de negocio, productos y compañías, ocasionando una demanda de herramientas que satisfagan esas interacciones, además de representar un nuevo negocio mundial.

El desarrollo de la plataforma Kubeet IoT se da por medio de la metodología ágil SCRUM; En SCRUM la preferencia por tener documentación en todo momento es poco estricta, ya que en esta metodología se encuentra más necesario el mantener la comunicación directa en el equipo de trabajo, por eso su herramienta principal es el Backlog [4].

La plataforma se divide en 3 partes:

* FrontEnd: Contiene toda la parte que interactúa con el usuario, es decir todas aquellas tecnologías que corren del lado del navegador web.
* BackEnd llamado WEBTOKEN: Donde se procesan los datos de entrada desde el FrontEnd, como la creación de usuarios, creación y verificación del Token y el CRUD de los dispositivos que podía crear el usuario.
* BackEnd de dispositivos públicos llamado RESTAPI [5].

**3.2 Lego Mindstorms.**

Lego Mindstorms es una línea de juguetes de [robótica](https://es.wikipedia.org/wiki/Rob%C3%B3tica) para niños fabricado por la empresa [LEGO](https://es.wikipedia.org/wiki/LEGO), que posee elementos básicos de las teorías robóticas, como la unión de piezas y la [programación](https://es.wikipedia.org/wiki/Programaci%C3%B3n) de acciones en forma [interactiva](https://es.wikipedia.org/wiki/Interactividad). Este robot fue comercializado por primera vez en septiembre de [1998](https://es.wikipedia.org/wiki/1998).

Lego Mindstorms puede ser usado para construir un modelo de sistema integrado con partes electromecánicas controladas por computador. Prácticamente todo puede ser representado con las piezas tal como en la vida real, como un elevador o robots industriales [6].

Hasta 2015 ha habido tres generaciones de Lego Mindstorms: el bloque RCX, el bloque NXT y el EV3.

Lego Mindstorms fue uno de los resultados de la fructífera colaboración entre Lego y el MIT. Esta asociación se emplea como ejemplo de relación entre la industria y la investigación académica que resulta muy beneficiosa para ambos socios [7].

[](https://commons.wikimedia.org/wiki/File:Many_Lego_Mindstorms_models.jpg)Figura 3. Varios modelos de bloques programables.

De arriba a abajo y de izquierda a derecha: Prototipo de bloque Lego Mindstorms creado por el MIT Media Lab en 1996, Primer bloque Mindstorms RCX comercializado en 1998, Bloque Mindstorms NXT comercializado en 2006, Bloque Mindstorms EV3 comercializado en 2013, 4 variaciones de bloquess Mindstorms comercializados por Lego, 1 bloque PicoCrickets compatible con Mindstorms creado por el MIT Media Lab y comercializado en 2006.

**3.2.1 Tercera Generación.**

[](https://commons.wikimedia.org/wiki/File:Lego-mindstorms-ev3.jpg)Figura 3.1 Lego Mindstorms EV3.

La tercera generación EV3 (evolution 3), comenzó a comercializarse en 2013. La versión normal permite realizar cinco modelos de base. En el sitio web de acompañamiento hay robots adicionales. Hay además otros sitios webs donde se muestran las instrucciones para construir otros modelos y se venden también libros donde se muestran otros modelos adicionales así como guías para desarrollar sus propios modelos.

El cambio más grande del [Lego Mindstorms NXT](https://en.wikipedia.org/wiki/Lego_Mindstorms_NXT) y [NXT 2.0](https://en.wikipedia.org/wiki/Lego_Mindstorms_NXT_2.0) al EV3 es los avances tecnológicos en el ladrillo programable. El procesador principal del NXT era un microcontrolador [ARM7](https://en.wikipedia.org/wiki/ARM7), mientras que el EV3 tiene una CPU [ARM9](https://en.wikipedia.org/wiki/ARM9) más potente con [Linux](https://en.wikipedia.org/wiki/Linux) . Un conector [USB](https://en.wikipedia.org/wiki/USB) y una ranura Micro [SD](https://en.wikipedia.org/wiki/Secure_Digital) (hasta 32GB) son nuevos para el EV3 [8].

El bloque EV3 actúa como centro de control y fuente de potencia del robot. Tiene:

* 4 puertos de entrada rj12 modificada (para conectar los sensores al bloque EV3)
* 4 puertos de salida rj12 modificada (para conectar los motores al bloque EV3)
* Un puerto mini USB para PC (para conectar el bloque EV3 a un ordenador)
* Un puerto de host USB (para agregar un conector [Wi-Fi](https://es.wikipedia.org/wiki/Wi-Fi" \o "Wi-Fi) y establecer conexiones en cadena)
* Un puerto para tarjetas [Micro SD](https://es.wikipedia.org/wiki/Micro_SD) (para ampliar la memoria disponible en el bloque EV3)
* Un [altavoz](https://es.wikipedia.org/wiki/Altavoz) integrado
* Receptor de señales infrarrojas para recibir comandos
* Receptor [Bluetooth](https://es.wikipedia.org/wiki/Bluetooth) y Wi-Fi.

  
Figura 3.2 Bloque programable EV3.

Las conexiones rj12 modificada son cabezales de rj12 normales a excepción de que la pestaña no la tienen en el centro sino desviada hacia un lateral. Esta modificación está hecha para que los cables no sean compatibles con las entradas de teléfono, que funcionan a voltajes más altos que los de seguridad [9].

El conjunto se completa con

* 2 servomotores grandes y un servomotor pequeño
* Sensores de color, infrarojo, de contacto
* Mando a distancia por señales infrarrojas
* Guía de usuario
* las piezas de fijación, de movimiento y flexibles especiales, las fijas compatibles con Lego Technic.

Los sensores de [NXT](https://en.wikipedia.org/wiki/Lego_Mindstorms_NXT) (Segunda Generación) se pueden utilizar con el EV3. Puede arrancar un sistema operativo alternativo desde una tarjeta microSD, lo que hace posible ejecutar [ev3dev](http://www.ev3dev.org/) , un [sistema](http://www.debian.org/) operativo basado en [Debian/Linux](http://www.debian.org/) .

**3.3 Ev3dev.**

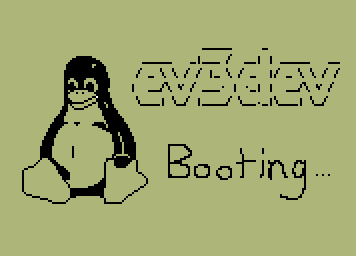
  
Figura 3.3 Logo de ev3dev.

Ev3dev es un sistema operativo basado en Debian/Linux que se ejecuta en varias plataformas compatibles con LEGO® MINDSTORMS, incluido el LEGO® MINDSTORMS EV3 y Raspberry Pi-powered BrickPi.

*“Ev3dev le da el poder para programar la forma en que desea. Se creo un marco de control de bajo nivel para controlar sensores, motores y prácticamente todo lo demás. Es tan fácil como leer y escribir en un archivo”.*

Respaldado por todo el poder de Linux, desde que ev3dev está basado en Debian Linux, hay más de 43,000 paquetes de software libres disponibles para su instalación. Y con el kernel de Linux en su núcleo, muchos dispositivos USB y Bluetooth, como dongles Wi-Fi, teclados y joysticks trabajan bien.

No es un firmware, más bien un arranque dual. Ev3dev se ejecuta desde una tarjeta microSD y nunca toca el firmware instalado en el EV3. Para volver a cambiar, simplemente se apaga y retire la tarjeta microSD [10].

  
Figura 3.4 Primer pantalla de inicio.

**3.3.1 Linux.**

**¿Qué es linux?**

**La respuesta rápida:**

Linux es un sistema operativo, una gran pieza de software que gestiona una computadora. Es similar a Microsoft Windows, pero es totalmente gratuito. El nombre exacto es GNU / Linux pero "Linux" se utiliza con más frecuencia.

GNU / Linux no es el producto de una empresa, pero un número de compañías y grupos de personas contribuyen a ello. De hecho, el sistema GNU / Linux es un componente central, que se ramifica en muchos productos diferentes. Se les llama distribuciones.

Las distribuciones cambian completamente el aspecto y la función de GNU / Linux. Ellos van desde los sistemas completos, totalmente soportados (respaldados por las empresas) hasta los más livianos que caben en una memoria USB o funcionan con computadoras viejas (a menudo desarrolladas por voluntarios).

**La respuesta completa:**

Cuando obtienes una distribución de GNU / Linux, también obtienes la libertad de estudiar, copiar, cambiar y redistribuirlo, eso es lo que lo hace verdaderamente software libre.

Muchas empresas desarrollan su propio sistema operativo basado en el software básico de GNU: productos en los que no tienen derechos exclusivos. ¿Cómo gira la rueda?

La mayoría de las empresas consiguen beneficios gracias al *soporte técnico y otros servicios de pago* relacionados con su distribución de GNU/Linux. Los clientes empresariales pagan por la garantía de actualizaciones de seguridad y asistencia, e incluso muchas veces entrenamiento y modificaciones al software para adecuarlo a sus necesidades.

Una enorme comunidad participa en el desarrollo y mejora del software, disminuyendo los costos y mejorando la eficiencia.

Finalmente, las personas suelen obtener el software gratuitamente, mientras que los clientes empresariales suelen estar felices de pagar para un soporte más avanzado [11].

**Usando GNU/Linux:**

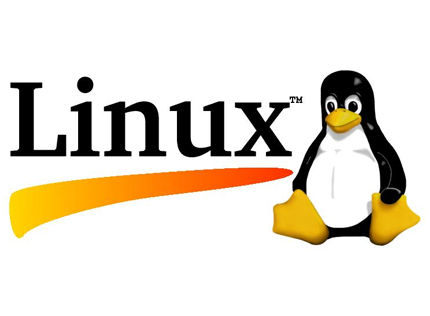
GNU / Linux no es más difícil de usar que Windows, y tiene muchas más capacidades. Sólo tarda una docena de minutos para familiarizarse con una distribución como las que recomendamos para los recién llegados, que vienen con muchos programas instalados.

Si necesita un software de calidad comercial para trabajar con documentos de negocios, Internet / redes, o multimedia y gráficos, está ahí de inmediato. ¿Quieres más que eso? GNU / Linux puede hacer - hay cientos de aplicaciones gratuitas y de alta calidad que puede encontrar, instalar y desinstalar de forma ordenada y fácil.

Con la adopción por numerosas empresas fabricantes, un buen número de [computadoras](https://es.wikipedia.org/wiki/Computadora) se venden con [distribuciones](https://es.wikipedia.org/wiki/Distribuci%C3%B3n_Linux) preinstaladas, y GNU/Linux ha comenzado a tomar su lugar en el vasto mercado de las computadoras de escritorio.

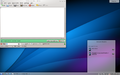
Algunas de las empresas que colaboran en la difusión de este sistema operativo ya sea trabajando en el núcleo Linux, proporcionando soluciones de software o preinstalando el sistema operativo, son: [Intel](https://es.wikipedia.org/wiki/Intel),​ [Google](https://es.wikipedia.org/wiki/Google), [IBM](https://es.wikipedia.org/wiki/IBM), [AMD](https://es.wikipedia.org/wiki/AMD), [Sun Microsystems](https://es.wikipedia.org/wiki/Sun_Microsystems), [Dell](https://es.wikipedia.org/wiki/Dell),​ [Lenovo](https://es.wikipedia.org/wiki/Lenovo), [Asus](https://es.wikipedia.org/wiki/Asus" \o "Asus), [Hewlett-Packard](https://es.wikipedia.org/wiki/Hewlett-Packard) (HP), [Silicon Graphics International](https://es.wikipedia.org/wiki/Silicon_Graphics" \o "Silicon Graphics) (SGI), [Renesas Technology](https://es.wikipedia.org/wiki/Renesas_Technology" \o "Renesas Technology),​ [Fujitsu](https://es.wikipedia.org/wiki/Fujitsu),​ [Analog Devices](https://es.wikipedia.org/wiki/Analog_Devices" \o "Analog Devices),​ [Freescale](https://es.wikipedia.org/wiki/Freescale" \o "Freescale),​ [VIA Technologies](https://es.wikipedia.org/wiki/VIA_Technologies),​ [Oracle](https://es.wikipedia.org/wiki/Oracle),​ [Novell](https://es.wikipedia.org/wiki/Novell) y [RedHat](https://es.wikipedia.org/wiki/RedHat" \o "RedHat),​ entre otras.

El respaldo de compañías de software también está presente, ya que, entre otras aplicaciones, [Nero](https://es.wikipedia.org/wiki/Nero_Linux), [Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java), [Google Earth](https://es.wikipedia.org/wiki/Google_Earth), [Google Desktop](https://es.wikipedia.org/wiki/Google_Desktop), [Adobe Reader](https://es.wikipedia.org/wiki/Adobe_Reader), [Adobe Flash](https://es.wikipedia.org/wiki/Adobe_Flash), [RealPlayer](https://es.wikipedia.org/wiki/RealPlayer" \o "RealPlayer)y [Yahoo! Messenger](https://es.wikipedia.org/wiki/Yahoo!_Messenger) están disponibles para GNU/Linux. Al haber software equivalente, es innecesario instalar los precisamente mencionados con anterioridad.

Figura 3.4 Logo de linux.

**3.3.2 Entorno gráfico.**

GNU/Linux puede funcionar tanto en [entorno gráfico](https://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica) como en [modo consola](https://es.wikipedia.org/wiki/Interfaz_de_l%C3%ADnea_de_comandos). La consola es común en distribuciones para servidores, mientras que la interfaz gráfica está orientada al usuario final tanto de hogar como empresarial. Asimismo, también existen los [entornos de escritorio](https://es.wikipedia.org/wiki/Entorno_de_escritorio), que son un conjunto de programas conformado por ventanas, iconos y muchas aplicaciones que facilitan la utilización del computador. Los escritorios más populares en GNU/Linux son: [GNOME](https://es.wikipedia.org/wiki/GNOME), [KDE](https://es.wikipedia.org/wiki/KDE) SC, [LXDE](https://es.wikipedia.org/wiki/LXDE), [Xfce](https://es.wikipedia.org/wiki/Xfce), [Unity](https://es.wikipedia.org/wiki/Unity_(entorno_de_escritorio)), [MATE](https://es.wikipedia.org/wiki/MATE) y [Cinnamon](https://es.wikipedia.org/wiki/Cinnamon) [12].

[](https://commons.wikimedia.org/wiki/File:KDE_4.png)  
Figura 3.5 Escritorio [KDE](https://es.wikipedia.org/wiki/KDE) SC 4.10.

[](https://commons.wikimedia.org/wiki/File:Gnome.png)  
Figura 3.6 Escritorio [GNOME](https://es.wikipedia.org/wiki/GNOME) 3.0.

[](https://commons.wikimedia.org/wiki/File:Xfce-4.4.png)  
Figura 3.7 Escritorio [XFCE](https://es.wikipedia.org/wiki/XFCE) 4.4.

[](https://commons.wikimedia.org/wiki/File:LXDE_desktop_full.png)  
Figura 3.8 Escritorio [LXDE](https://es.wikipedia.org/wiki/LXDE) 0.5.

**3.3.3 sistema de programación.**

La colección de utilidades para la [programación](https://es.wikipedia.org/wiki/Programaci%C3%B3n) de GNU es con diferencia la familia de [compiladores](https://es.wikipedia.org/wiki/Compilador) más utilizada en este sistema operativo. Tiene capacidad para compilar [C](https://es.wikipedia.org/wiki/C), [C++](https://es.wikipedia.org/wiki/C%2B%2B), [Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java), [Ada](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Ada), [Pascal](https://es.wikipedia.org/wiki/Pascal_(lenguaje_de_programaci%C3%B3n)), [Python](https://es.wikipedia.org/wiki/Python" \o "Python), [Perl](https://es.wikipedia.org/wiki/Perl), entre otros muchos lenguajes. Además soporta diversas arquitecturas mediante la compilación cruzada, lo que hace que sea un entorno adecuado para desarrollos heterogéneos.

Hay varios [entornos de desarrollo integrados](https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) disponibles para GNU/Linux incluyendo, [Anjuta](https://es.wikipedia.org/wiki/Anjuta), [KDevelop](https://es.wikipedia.org/wiki/KDevelop), [Lazarus](https://es.wikipedia.org/wiki/Lazarus_(entorno_de_desarrollo)), [Ultimate++](https://es.wikipedia.org/wiki/Ultimate%2B%2B), [Code::Blocks](https://es.wikipedia.org/wiki/Code::Blocks), [NetBeans IDE](https://es.wikipedia.org/wiki/NetBeans_IDE) y [Eclipse](https://es.wikipedia.org/wiki/Eclipse_(software)). También existen editores extensibles como [Emacs](https://es.wikipedia.org/wiki/Emacs" \o "Emacs) o [Vim](https://es.wikipedia.org/wiki/Vim" \o "Vim). GNU/Linux también dispone de capacidades para lenguajes de guion (script), aparte de los clásicos lenguajes de programación de [shell](https://es.wikipedia.org/wiki/Int%C3%A9rprete_de_comandos" \o "Intérprete de comandos), o el de procesador de textos por patrones y expresiones regulares conocido como [awk](https://es.wikipedia.org/wiki/Awk" \o "Awk), la mayoría de las distribuciones tienen instalado [Python](https://es.wikipedia.org/wiki/Python" \o "Python), [Perl](https://es.wikipedia.org/wiki/Perl), [PHP](https://es.wikipedia.org/wiki/PHP) y [Ruby](https://es.wikipedia.org/wiki/Ruby).

Las aplicaciones para GNU/Linux se distribuyen principalmente en los formatos “.[deb](https://es.wikipedia.org/wiki/Deb" \o "Deb)” y “[.rpm](https://es.wikipedia.org/wiki/RPM_Package_Manager" \o "RPM Package Manager)”, los cuales fueron creados por los desarrolladores de [Debian](https://es.wikipedia.org/wiki/Debian" \o "Debian) y [Red Hat](https://es.wikipedia.org/wiki/Red_Hat) respectivamente. También existe la posibilidad de instalar aplicaciones a partir de código fuente en todas las distribuciones.

**Uso de los dispositivos con ev3dev desde el código**

La ejecución del ev3dev permite una gama de diversos lenguajes de programación. Con la contribución de los miembros de la comunidad se crean bibliotecas que permiten utilizar características como motores y sensores por medio de código [13]. Los lenguajes son:

**Python**

Figura 4. Logo de python.

Python es un lenguaje de programación fácil de aprender y potente. Tiene estructuras de datos eficientes de alto nivel y un enfoque simple pero efectivo para la programación orientada a objetos. La elegante sintaxis y la tipificación dinámica de Python, junto con su naturaleza interpretada, lo convierten en un lenguaje ideal para el desarrollo de scripts y aplicaciones rápidas en muchas áreas de la mayoría de las plataformas.

El intérprete de Python se amplía fácilmente con nuevas funciones y tipos de datos implementados en C o C ++ (o en otros idiomas llamados desde C). Python también es adecuado como un lenguaje de extensión para aplicaciones personalizables.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en [C](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n)) o [C++](https://es.wikipedia.org/wiki/C%2B%2B). Python puede incluirse en aplicaciones que necesitan una interfaz programable [14].

**JavaScript**

Figura 5. Logo de javascript.

JavaScript es el lenguaje de scripting de la Web. Con un programa llamado Node.js, puede escribir JavaScript que se puede ejecutar localmente como cualquier otro lenguaje de scripting. JavaScript con Node.js es ideal para escribir servidores web y otros programas asíncronos que no necesitan ejecutar bucles ajustados o temporizadores precisos. Tiene una sintaxis similar a la de C y lenguas relacionadas, y es relativamente fácil de aprender [13].

**Java (compatible con LeJOS)**

Figura 6. Logo de lejos.

Java es un lenguaje de programación informática de uso general que es concurrente, basado en clases y orientado a objetos. En Java 8, el lenguaje agregó algunas características funcionales muy interesantes de aprender [13].

**Go**

Figura 7. Logo de go.

Go es un lenguaje compilado y estáticamente escrito en Google. Su objetivo es ser simple y ligero, al mismo tiempo que proporciona características lingüísticas modernas. Si bien es un lenguaje compilado, tiene su propio compilador incorporado, lo que significa que no es necesario pasar tiempo configurando herramientas especiales como lo hace con la mayoría de los otros lenguajes compilados [13].

**C**

Figura 8. Lenguaje c.

C es un lenguaje compilado de bajo nivel que es útil para interactuar con otro código basado en C. Es muy ligero y a menudo el más portátil a través de plataformas.

Las principales características del Lenguaje C son:

1.-Tiene un conjunto completo de instrucciones de control.

2.-Permite la agrupación de instrucciones.

3.-Incluye el concepto de puntero (variable que contiene la dirección de otra variable).

4.-Los argumentos de las funciones se transfieren por su valor.

5.- E/S no forma parte del lenguaje, sino que se proporciona a través de una biblioteca de funciones.

Permite la separación de un programa en módulos que admiten compilación independiente [15].

**C ++**

Figura 9. Logo de c++.

C ++ es un lenguaje de bajo nivel, compilado que es altamente performant mientras que todavía proporciona características modernas del lenguaje. Es mejor para aplicaciones que requieren la ejecución más rápida o la interacción con las bibliotecas existentes de C ++.

En la actualidad, el C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores profesionales le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. El C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original. La evolución de C++ ha continuado con la aparición de Java, un lenguaje creado simplificando algunas cosas de C++ y añadiendo otras, que se utiliza para realizar aplicaciones en Internet [16].

**Vala, Genie y otros lenguajes basados ​​en GObject con ev3devKit**

A través de GObject-introspección , esta biblioteca puede ser utilizada por idiomas como Vala y Genie, entre muchos otros . Esto es ideal para las personas que desean usar una sintaxis de alto nivel mientras siguen produciendo una aplicación de rendimiento, o para las personas que quieren una API menos propensa a errores para C. El Administrador de ladrillo para ev3dev se escribe utilizando esta biblioteca [13].

**3.4 conectividad.**

Conexión a Internet

Se tienen algunas opciones para conectarse a Internet para el Ev3.

**Con un dongle Wi-Fi**

La ruta más sencilla es usar un dongle Wi-Fi conectado a través del puerto USB del EV3. Muchos dongles Wi-Fi USB que soportan Linux funcionarán, algunos soportados por el sistema operativo predeterminado de EV3 [17]:

•NetGear WNA1100 Wireless-N 150 (oficialmente soportado por el software de LEGO) [18]:

  
Figura 10. NetGear WNA1100.

* Tecnología WiFi: 802.11n
* Rendimiento WiFi: N150 (150 Mbps), Frecuencia de señal inalámbrica máxima derivada de las especificaciones IEEE 802.11. El rendimiento real de datos y la cobertura inalámbrica variarán. Las condiciones de la red y los factores ambientales, incluido el volumen de tráfico de la red, las interferencias y la construcción de edificios, pueden reducir el rendimiento real de los datos y la cobertura inalámbrica.
* Estándares: IEEE ™ 802.11 b / g / n 2.4 GHz
* Banda WiFi: 2.4GHz
* Especificaciones físicas: Dimensiones: 60,86 x 18,44 x 8,83 mm (2,40 x 0,73 x 0,35 pulgadas), Peso: 74 g (2,6 onzas)

•Adaptador inalámbrico Edimax EW-7811Un 802.11n (oficialmente compatible con el software de LEGO) [19]:

  
Figura 11. Edimax EW-7811Un.

EW-7811Un es un adaptador inalámbrico USB nano que admite el rango máximo y la velocidad. A pesar del tamaño, este pequeño adaptador USB soporta una mayor velocidad de transferencia de datos de hasta 150Mbps cuando se conecta con un dispositivo inalámbrico 802.11n que es 3 veces más rápido que su conexión normal de 11g.

* Cumple con estándares inalámbricos 802.11b / g / n con velocidad de datos de hasta 150Mbps
* Ahorro de energía verde: admite el control inteligente de potencia de transmisión y el ajuste de estado de reposo automático
* Aumenta la cobertura inalámbrica 3 veces más
* Incluye el asistente de configuración de múltiples idiomas EZmax.
* Soporta cifrado WEP, WPA, WPA2 de 64/128-bit y compatible con WPS.
* Soporta QoS-WMM, WMM-Modo de ahorro de energía

•EP-N8508GS 150Mbps Mini adaptador inalámbrico 802.11N USB (no soportado oficialmente por LEGO, pero también funciona) [20]:

  
Figura 12. EP-N8508GS.

* Utiliza el chip REALTEK 8188 y la tecnología de transmisión inalámbrica 802.11n (1T1R), velocidad de transmisión de hasta 150Mbps por 130 m2
* Dimensión: 1.8 x 1.4 x 0.6 cm
* rango de frecuencias: 2.4 a 2.4835 GHz
* canal: 13
* protocolo de red de apoyo: CSMA / CA con acuse de recibo
* modo de transmisión: DSSS
* Sensibilidad: 54-68dbm @ 10%, 11m: -85dBm @ 8%, 6m: -88dBm @ 10%, 1m: -90dBm @ 8%, 256k: -105dbm @ 8% (valor típico)
* características de seguridad: soporte encriptación de 64/128/125 bits de datos WEP, WPA, soporte IEEE 802.1 x, TKIP, AES cifrado.
* sistemas de apoyo: win2k / xp / vista / win7 / mac / linux
* Tipo de interfaz: usb

• Adaptador USB Nano Inalámbrico N 150Mbps TL-WN725N [21]:

  
Figura 13. TL-WN725N.

Adaptador USB TP-LINK 150Mbps inalámbrico N Nano, TL-WN725N permite al usuario conectar un ordenador de sobremesa o portátil a una red inalámbrica en 150Mbps. Este adaptador miniatura está diseñado para ser lo más cómodo posible y una vez conectado al puerto USB de un ordenador, se puede dejar allí, ya sea que viaje o en casa.

* Diseño liso y brillante en miniatura tan pequeño que una vez conectado, puede dejarlo en un puerto USB de la Laptop
* Rápida transmisión de velocidad de hasta 150Mbps ideal para el streaming de video o llamadas por internet
* Seguridad avanzada: Soporta 64/128 WEP, WPA, PA2/WPA-PSK/WPA2-PSK(TKIP/AES)
* Dimensiones (W X D X H): 0.73x0.59x0.28pulgadas. (18.6x15x7.1mm)
* Estándares inalámbricos: IEEE 802.11b, IEEE 802.11g, IEEE 802.11n
* Frecuencia: 2.400-2.4835GHz
* Ambiente: Temperatura de Funcionamiento: 0℃~40 ℃ (32 ℉~104℉), Temperatura de Almacenamiento: -40℃~70 ℃ (-40 ℉~158℉), Humedad de Funcionamiento: 10%~90% sin condensación, Humedad de Almacenamiento: 5%~90% sin condensación
* Interfaz: USB 2.0

**Con USB a través de una PC**

Se puede utilizar un PC para conectarse al EV3 a través de su mini puerto USB y compartir una conexión a Internet [17].

  
Figura 14. Conexión por USB.

**Con Bluetooth a través de un PC**

Si tiene un PC o dispositivo Android con capacidad Bluetooth, puede conectar su EV3 a través de él para acceder a la Web [17].

**Con un adaptador Ethernet USB**

Con un adaptador USB Ethernet compatible con Linux, puede usarlo para acceder a la red desde su EV3. Sólo tiene que conectarlo al ladrillo y conectar el adaptador a un cable Ethernet; No se necesita ninguna configuración [17].