

# Prototipado Rápido de Proyectos IoT sin Programación

Luciano Corniglione Minig, Gabriela Alejandra Beltramen Ramos

**Abstract**—Actualmente estamos viviendo la revolución de IoT debido a la gran cantidad de dispositivos que se encuentran interconectados. Se espera que para 2020, más de 20 billones de dispositivos interconecten a las personas alrededor del mundo [1]. Según un estudio realizado por VisionMobile [2], el 70% de las personas que realizan trabajos relacionados con IoT son personas autodidactas y entusiastas de la tecnología que les apasiona construir sistemas tecnológicos sin tener un conocimiento profundo en la materia. Proponemos una herramienta que abstraiga el conocimiento técnico necesario, con el objetivo de reducir la brecha de conocimiento entre expertos y makers, y ayudar a que más gente construya sistemas IoT y así soportar los desafíos tecnológicos futuros. Bricks se focaliza en incrementar la productividad y reducir los tiempos de prototipado mediante componentes visuales y sin conocimientos previos de programación. De esta manera se podrá validar el sistema y los conceptos de manera rápida y fácil. Además, es posible simular sistemas IoT sin necesidad de tener el hardware necesario físicamente. A continuación se describirá la herramienta en profundidad, se ejemplificará un caso de uso y se hablará acerca de la tecnología con la que fue desarrollada. Finalmente, se hará una breve comparación con otras herramientas.

**Index Terms**—IoT, prototipo, hardware, software, maker, modelado, reglas, despliegue, simulación

## 1 INTRODUCCIÓN

IoT representa una evolución tecnológica que permite la interacción y comunicación entre “cosas”, objetos, dispositivos a través del internet. El desarrollo de soluciones IoT se basa en la convergencia de tres tecnologías principales: hardware, software y comunicaciones. Es necesaria la conjunción y sinergia entre los mismos para poder desarrollarla. Las “cosas” son objetos embebidos (hardware) que generarán datos y, gracias a estar conectados a internet, permitirán transmitirlos como así también recibirlos de otros dispositivos (comunicación). La forma en que los datos son almacenados, interpretados, procesados y comunicados se realiza mediante el uso de un software.

La realización de un proyecto IoT se compone de una serie de etapas que comienzan con un modelado de las placas y los sensores, se especifica que tipo de interacciones va a haber entre ellos y luego, se espera poder validar la salida mediante la observación de los valores sensados.

Actualmente existen varias herramientas que facilitan algunas de las etapas del desarrollo de un proyecto IoT nombradas anteriormente.

Elas son: AutoDesk 123D Circuits [3], Blynk [4], Cayenne [5], Node-RED [6], Artik Cloud [7], MIT App Inventor [8], entre otras. Para que el mismo sea completo, se deben integrar todas estas herramientas, lo que incurre en una pérdida de tiempo y esfuerzo.

Además, desarrollar proyectos IoT requiere una serie de conocimientos, tanto electrónicos como de software tales como entender especificaciones técnicas de las placas, de los sensores, cómo éstos transmiten la información, cómo funciona la tecnología cloud, lógica de programación, hardware, etc.

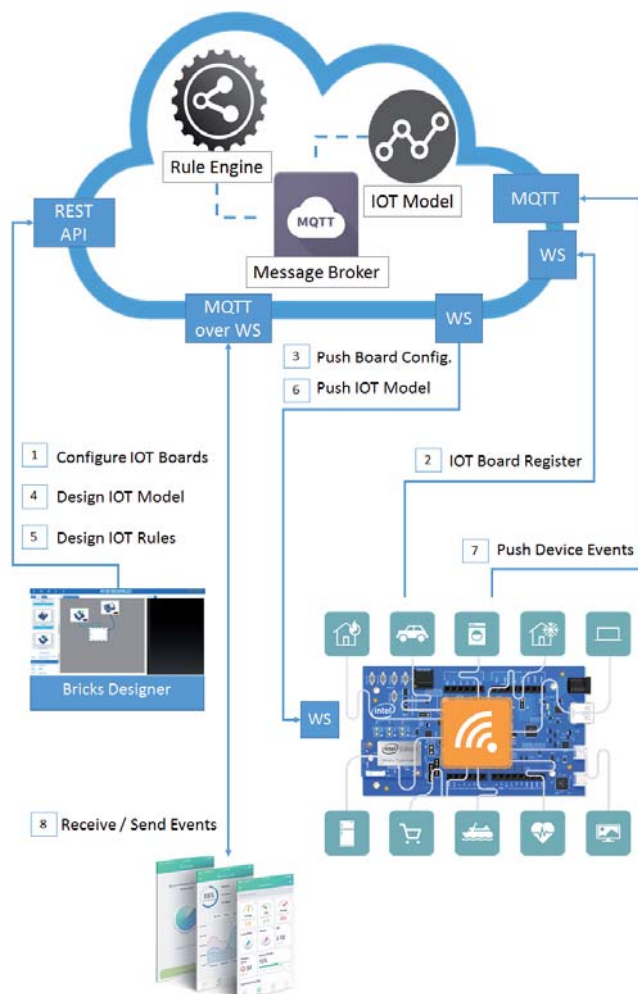
Surge como problemática la inexistencia de una herramienta integral con un alto nivel de abstracción que nos brinde soporte general y completo a todas las etapas del desarrollo de proyectos IoT antes mencionadas.

Nuestra propuesta, es una herramienta que permite una fácil interacción del usuario con distintos equipos IoT de manera visual. La idea principal es que personas, sin mucho conocimiento sobre hardware y lenguajes de programación, puedan rápidamente implementar sistemas IoT.

## 2 BRICKS

El funcionamiento de Bricks se compone de cuatro etapas, una de las cuales es opcional.

- 1) Creación del modelo visual
- 2) Aplicación de reglas
- 3) Deployment y/o simulación del proyecto
- 4) Visualización de datos



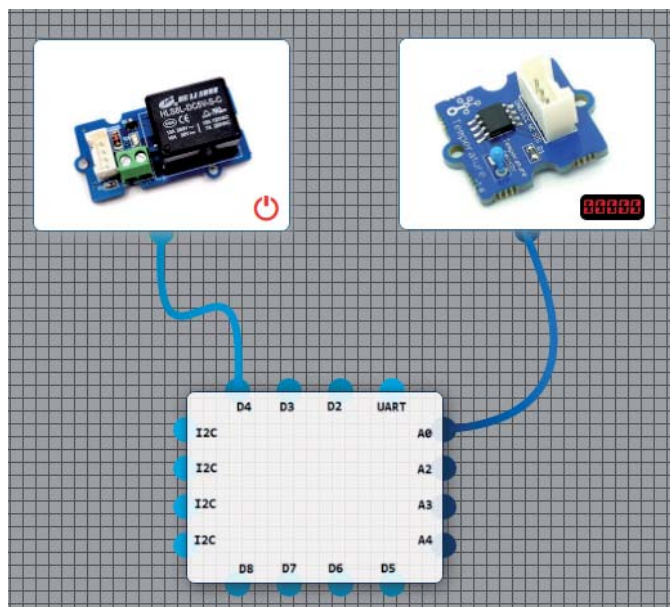
Primeramente, se debe crear un modelo visual que representará las conexiones de los sensores en una placa específica. Luego, se le introducirán reglas al modelo de manera tal de asignarle un comportamiento a nuestro sistema. Esto puede ser interpretado como asignaciones de acciones ante distintos eventos, permitiendo la intercomunicación entre sensores. La siguiente etapa consiste en la ejecución del prototipo creado a través de el deployment del proyecto en la placa física o a través de la simulación en la placa virtual de la herramienta. Por último, podemos acceder a una visualización de los datos, es decir, los

valores que fueron sensados, en un dashboard. Esto nos permite tener un acceso a datos históricos y gráficos que nos brinden mayor información.

A continuación, se irán describiendo con mayor profundidad las etapas antes mencionadas mediante un caso de uso para lograr mayor claridad. El mismo propone realizar un sistema de climatización inteligente que, de manera automática, encienda o apague el aire acondicionado para mantener la temperatura del ambiente.

### 2.1 Creación del modelo visual

Una de las partes del IDE corresponde a la sección donde se modela el proyecto. Esta sección permite crear un proyecto a través de un modelo visual. Posee una barra de herramientas donde podemos encontrar placas Intel y sensores que provienen del Groove Kit. Cada uno de ellos tiene una descripción que relata las principales características. Estos elementos pueden ser arrastrados a un espacio de trabajo llamado LabView. Dentro del espacio de trabajo los elementos pueden ser interconectados, es decir, un sensor puede ser conectado a una placa construyendo el modelo visual antes mencionado. Dentro de las opciones del menú que ofrece el IDE podemos encontrar funciones de edición (borrar un elemento, guardar últimos cambios, etcétera), de deployment a una placa y de comienzo de simulación. Comenzando con nuestro ejemplo, asumamos que tenemos una placa IoT y queremos comenzar con nuestro proyecto. Para eso, deberíamos seleccionar la placa y conectarle un relay y un sensor de temperatura como se muestra en la siguiente imagen:



El sensor irá tomando la temperatura y el relé enviará una orden al aire acondicionado para que éste se encienda cuando se supere una cierta temperatura.

## 2.2 Aplicación de Reglas

Haciendo uso del modelo creado en el software de diseño, la herramienta permite el ingreso de reglas las cuales permitirán definir un comportamiento al sistema a través de interconexiones inteligentes entre sensores. Esta parte del IDE corresponde al Rule Engine, encargado principalmente de disparar acciones correspondientes a condiciones preestablecidas por el usuario.

Provee una interfaz gráfica, que permite crear reglas del tipo [ If ... Then ... ], es decir, a ciertas condiciones le corresponden ciertos eventos. Estas reglas son almacenadas como parte del modelo y luego son inyectadas al rule engine. Las reglas son introducidas de una forma muy sencilla, semejante al lenguaje natural, con el objetivo de que sea comprensible.

Continuando con nuestro ejemplo, una vez que se ha instaurado el modelo de placas y sensores, se define la regla que nos permite encender el aire acondicionado cuando se superen los 30C. Se muestra a continuación cómo queda definida la regla en la herramienta:

Choose device activity to monitor

If

Joule Temperature Sensor greater than 30

Add fields

Send Actions to your devices

Then

Joule Relay 1|

Add fields

Save

## 2.3 Deployment y/o simulación del proyecto

Luego que haya sido modelado el prototipo junto a sus reglas, se puede optar por el despliegue del mismo en una placa física o simular dicha interacción. Suponiendo que se tiene la placa física, se procede entonces a tomar el modelo y establecerlo sobre ella. Para ello, se desarrolló un Runtime, encargado de la comunicación entre el hardware y el software, el cual permitirá implementar físicamente el modelo creado. Luego comenzará a recolectar los valores que arrojen los sensores y a publicarlos en internet para que puedan ser vistos por el usuario.

## 2.4 Visualización de datos

Finalmente, una vez que el proyecto ha sido desplegado o simulado, se pueden visualizar los valores sensados en el software de diseño IoT o a través del uso de un dashboard, donde se disponen gráficos que permiten ver los datos a través del tiempo y el comportamiento del sistema. En nuestro ejemplo, se verá cómo cambia la temperatura y el estado en el que se encuentra el aire acondicionado (on/off).

El rule engine será el encargado automáticamente de ejecutar las reglas cada vez que recibe un dato nuevo y disparar los eventos correspondientes a cada regla. Así, una vez que sean superados los 30° C, el rule engine le enviará una acción al relé que permita encender el aire acondicionado.

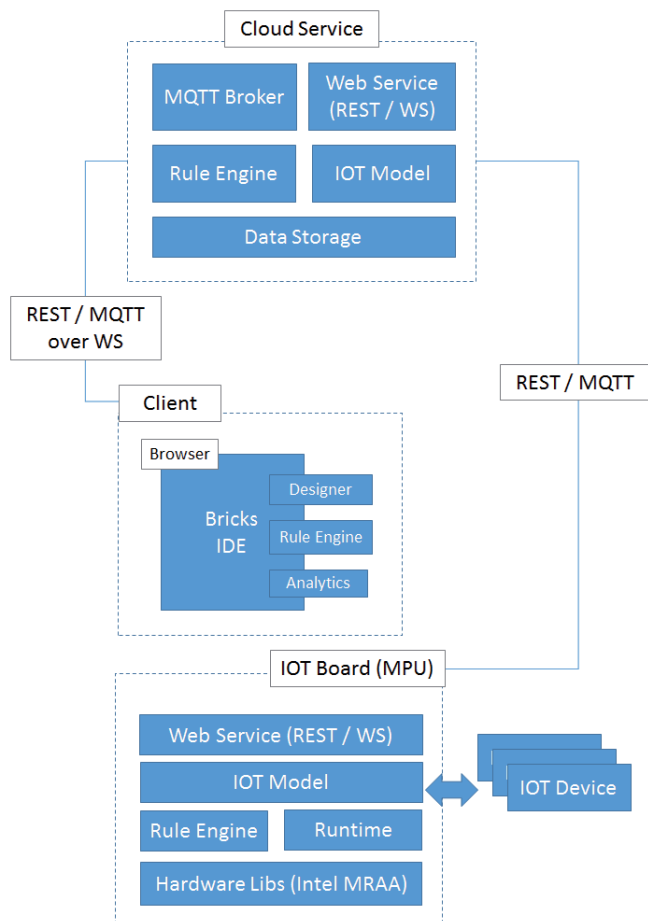
# 3 TECHNICAL DETAILS

## 3.1 Arquitectura

La arquitectura elegida para este proyecto es una arquitectura orientada a servicios, lo que

nos permite tener sistemas más escalables y robustos.

A continuación, mostraremos la arquitectura general del producto:



### 3.2 Tecnologías Usadas

Generalizando, el IDE fue creado usando tecnologías web HTML5, JS y CSS3. En todo el proyecto se usó TypeScript y ES6 con el objetivo de lograr un código bajo las mejores prácticas que permita un desarrollo más productivo. Se usó React.js [9] en conjunto con Redux [10] y Angular 2 [11].

La aplicación es servida por un servidor Node.js, usando el framework Express. En este proyecto, la aplicación cuenta con la mayoría de la lógica implementada del lado del Cliente. Nuestro sistema se encuentra montado sobre tecnología Cloud, más específicamente sobre Amazon Web Services, lo que nos brinda una alta disponibilidad del sistema, una respuesta rápida a los pedidos de los usuarios y alta

escalabilidad.

El almacenamiento se realizó sobre una base de datos no relacional denominada MongoDB [12].

## 4 TRABAJO RELACIONADO

A continuación, se presentan con detalle algunas herramientas del mercado que han sido investigadas y que sirven para llevar adelante proyectos IoT. Éstas serán comparadas con nuestro producto final.

Dado que se planteó como problemática la necesidad de una herramienta integral, Bricks permite la creación end to end de un proyecto IoT, esto es: diseño del modelo visual de placas y sensores interconectados, aplicación de reglas para estipular su comportamiento, deployment del proyecto en la placa física, posibilidad de simular sin necesidad de hacer un deployment para poder validar el prototipo creado y, finalmente, la visualización gráfica de los valores sensados a través de un dashboard que permita una mejor interpretación de los datos.

A modo de resumen, se muestra un cuadro comparativo de algunas de las herramientas que han sido investigadas y las etapas a las que dan soporte:

	Diseño del Modelo	Comportamiento	Deployment	Simulación	Visualización de Datos
Autodesk 123D Circuits	x		x	x	
Blynk	x				
Cayenne	x	x			x
Node-RED		x			
Ubidots					x
Wylidrin Sensors					x
Artik Cloud		x			x

Asimismo, se vuelve relevante que cada una de estas etapas pueda ser cumplida de una forma sencilla e intuitiva. Para ésto, Bricks implementa funciones visuales del tipo drag & drop. También cuenta con un lenguaje adecuado evitando elevados niveles de tecnicismos y representando las acciones que se pueden realizar de una manera descriptiva.



Parte de los conocimientos técnicos necesarios para el desarrollo de un proyecto IoT tienen que ver con la programación. Existen herramientas que permiten la creación de modelos visuales, tal como Autodesk 123D Circuits, pero la codificación es algo imprescindible para poder lograr su funcionamiento. Es por ello que en Bricks no es necesario tener conocimientos en programación enfocando el esfuerzo en la representación de la solución en sí misma.

Gracias a su sencillez y el grado de abstracción que logra, Bricks permite la creación de proyectos complejos que, en otras herramientas tales como Blynk, MIT App Inventor, entre otras, se debería incurrir en gastar mucho tiempo y esfuerzo. Es también, por esta razón, que Bricks puede ser utilizada como una herramienta educativa donde permita a personas introducirlas en este campo y lograr un avance en su aprendizaje de manera más rápida. Cabe aclarar que una persona que esté interesada en aprender no tendría la necesidad de tener una placa y sensores para poder comenzar en su aprendizaje. Es posible la creación de proyectos y su posterior simulación para poder evidenciar su funcionamiento.

Bricks se fundamenta en la idea del prototipado rápido, lo cual permite que el prototipo pueda ser deployado en una placa física real o simulado, para poder validarlos sin perder demasiado tiempo.

Finalmente, Bricks es una herramienta web permitiendo la creación de los proyectos desde el browser sin la necesidad de la descarga e instalación de programas y, además, permitiendo acceder a los mismos desde cualquier computador.

## 5 CONCLUSIÓN Y TRABAJO FUTURO

Bricks es un entorno de desarrollo visual que permite la creación de proyectos IoT de manera fácil y rápida. A través de funciones drag drop, facilita la creación de prototipos arrastrando placas y sensores e interconectándolos. Con el objetivo de crear sistemas más complejos, se le pueden introducir reglas, las cuales serán dispuestas por el usuario. Finalmente, el proyecto

puede ser deployado en una placa o simulado. La herramienta está orientada a personas que están interesadas en la creación de este tipo de proyectos sin la necesidad de que los mismos tengan mucho conocimiento sobre placas y programación, pudiendo probar y crear soluciones IoT sin mucho tiempo ni esfuerzo.

El sistema se compone de 3 pilares importantes. El primero es un IDE en el cual se le permitirá al usuario crear el modelo donde se podrán seleccionar placas y conectarle sensores a ella. Luego, el segundo pilar, es el Rule Engine donde se indican las condiciones. Y por último, un Dashboard para poder visualizar los datos de los sensores a través del uso de distintos widgets.

Tenemos como objetivo a futuro añadir un nuevo pilar, el uso de un webhook como IFTTT [13]. Consideramos que es muy interesante la creación de nuevas experiencias a través de la automatización de distintas acciones brindadas por los applets de IFTTT haciendo uso de los diferentes servicios en línea que se usan comúnmente en el día a día, tales como Facebook, Instagram, Spotify, Google Drive, etc. Creemos que así se podrá facilitar aún más la creación de proyectos y administración de distintas plataformas virtuales ahorrando tiempo y trabajo.

Otro objetivo a futuro sería poder aumentar el catálogo de sensores de la herramienta permitiendo especificar la creación de un nuevo sensor con sus descripciones, medidas, funciones, etc. Además se podría tener un store de sensores donde los proveedores de sensores publican su sensor y dan sus especificaciones, y luego pueda ser simulada en la aplicación. Actualmente, ésta herramienta funciona sobre placas Intel, como las conocidas Galileo, Joule o Edison. A futuro, la idea es generalizar la herramienta, pudiendo generar código en C++ automáticamente para que pueda ser utilizada sobre otras placas como Arduino.

## REFERENCES

- [1] Gartner, 'Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016', 2017. [Online]. Available: <http://www.gartner.com/newsroom/id/3598917> [Accessed: 27- Apr- 2017].

- [2] Stijn. Schuermans, '70% of Smart Home developers are hobbyists', 2015. [Online]. Available: <https://www.visionmobile.com/blog/2015/09/70-of-smart-home-developers-are-hobbyists>. [Accessed: 27- Apr- 2017].
- [3] Autodesk 123D Circuits, 'Design, compile, and simulate your electronic project online – for free.'. [Online]. Available: <http://www.123dapp.com/circuits> [Accessed: 27- Apr- 2017].
- [4] Blynk, 'Getting Started with Blynk'. [Online]. Available: <http://www.blynk.cc/getting-started/> [Accessed: 27- Apr- 2017].
- [5] Cayenne , 'Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016', 2017. [Online]. Available: <http://www.gartner.com/newsroom/id/3598917> [Accessed: 27- Apr- 2017].
- [6] Node-RED , 'Getting Started'. [Online]. Available: <https://nodered.org/docs/getting-started/> [Accessed: 27- Apr- 2017].
- [7] Artik Cloud, 'Make Connections, Not Silos'. [Online]. Available: <https://artik.cloud/> [Accessed: 27- Apr- 2017].
- [8] MIT App Inventor, 'Getting Started with MIT App Inventor 2'. [Online]. Available: <http://appinventor.mit.edu/explore/get-started?> [Accessed: 27- Apr- 2017].
- [9] React.js, 'A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES'. [Online]. Available: <https://facebook.github.io/react/> [Accessed: 27- Apr- 2017].
- [10] Redux, 'Redux'. [Online]. Available: <http://redux.js.org/> [Accessed: 27- Apr- 2017].
- [11] Angular 2, 'ARCHIVED ANGULAR V2 DOCS TYPESCRIPT'. [Online]. Available: <https://v2.angular.io/docs/ts/latest/> [Accessed: 27- Apr- 2017].
- [12] MongoDB, 'What is MongoDB?'. [Online]. Available: <https://www.mongodb.com/what-is-mongodb> [Accessed: 27- Apr- 2017].
- [13] IFTTT , 'Do more with the services you love'. [Online]. Available: <https://ifttt.com/> [Accessed: 27- Apr- 2017].