

Herramientas de programación aplicables a los procesos empresariales y de producción en Colombia

Programming tools applicable to business and production processes in Colombia

William Niebles-Nuñez¹
Hugo Hernández-Palma²
Lorena Hoyos-Babilonia³

Cómo citar/ How to cite: Niebles, W., Hernández, H. & Hoyos, L. (2020). Herramientas de programación aplicables a los procesos empresariales y de producción en Colombia. *Revista Saber, Ciencia y Libertad*, 15(2), 91 – 105.
<https://doi.org/10.18041/2382-3240/saber.2020v15n2.6718>

Resumen

El auge tecnológico que se observa en el mundo actual se ha fundamentado en la capacidad del ser humano para diseñar dispositivos que automaticen sus tareas diarias, creando así una tendencia exponencial que ha traído innumerables beneficios en términos de innovación y solución a problemas cotidianos. Es así como a continuación, se plantea una investigación cualitativa que busca analizar cómo las herramientas de programación pueden contribuir con las exigencias globales que afronta la sociedad actual, en el terreno empresarial y tecnológico. Las reflexiones finales permiten visualizar las ventajas y desventajas de varias herramientas ya conocidas, al tiempo que facilitan proyectar su uso e inserción en actividades diversas en el ámbito colombiano.

Palabras clave

Programación; internet de las cosas; empresa; globalización; Colombia.

Abstract

The technological boom observed in today's world has been based on the ability of humans to design devices that automate their daily tasks, thus creating an exponential trend that has brought innumerable benefits in terms of innovation and solution to everyday problems. This is how, next, qualitative research is proposed that seeks to analyze how programming tools can contribute to the global demands faced by today's society, in the field of business and technology. The final reflections make it possible to visualize

Fecha de recepción: 28 de marzo de 2020
Fecha de evaluación: 20 de abril de 2020
Fecha de aceptación: 28 de mayo de 2020

Este es un artículo Open Access bajo la licencia BY-NC-SA
(<http://creativecommons.org/licenses/by-nc-sa/4.0/>)
Published by Universidad Libre



1 Doctor en Ciencias Gerenciales, Magíster en Dirección Estratégica, Administrador de Empresas. Docente de la Universidad de Sucre - Colombia. Correo electrónico: william.niebles@unisuc.edu.co. <https://orcid.org/0000-0001-9411-4583>

2 Magíster en Sistema de Gestión, Especialista en Estudios Pedagógicos, Especialista en Diseño y Evaluación de proyectos, Ingeniero Industrial. Docente Programa de Administración de Empresas Universidad del Atlántico - Colombia. Correo electrónico: hugohernandezpalma@gmail.com. ORCID iD: <https://orcid.org/0000-0002-3873-0530>

3 Magíster en Prevención de Riesgo Laborales, Administradora de Empresas. Coordinadora Maestría en Administración de Empresas. Universidad de Sucre - Colombia. Correo electrónico: lohob81@hotmail.com ORCID iD: <https://orcid.org/0000-0003-1991-6054>.

the advantages and disadvantages of several already known tools, while at the same time making it easier to project their use and insertion in diverse activities in the Colombian sphere.

Keywords

Programming; internet of things; company; globalization; Colombia.

Introducción

El ser humano se caracteriza por su comportamiento enfocado hacia la innovación y el desarrollo. Esto ha traído como resultado que, en los últimos años, se han dado importantes saltos relacionados con la entrada de nuevas tecnologías y la globalización, dicho proceso que ha permitido crear un complicado entramado comunicacional que permite movilizar información como nunca antes se había registrado (Pitre et al, 2018; Berrio, Redondo y Hernández, 2018).

La entrada de la sociedad de la información ha colocado a los datos en el centro de atención, la innumerable cantidad de documentos y cifras que se generan día a día, colocan al usuario frente a retos que antes desconocía, entre estos, seguridad, privacidad, comprensión, etc. Asimismo, las grandes dificultades que afronta el planeta en todas sus dimensiones, obligan a que estos logros tecnológicos se coloquen al servicio de la humanidad y la sustentabilidad (Martin, 2017) (Sachs y Verini, 2015). Así las cosas, es posible identificar cómo van surgiendo dispositivos que apoyan aspectos cotidianos de la vida de forma eficiente y utilizando menos recursos. En este sentido, se cuenta actualmente con nuevas oportunidades, tales como el *cloud computing*, la apertura del campo electrónico a una comunidad de desarrolladores ávida por crear sus propios componentes apoyado por lenguajes de programación más ágiles y aplicables para la puesta en marcha de diversidad de proyectos (Botta, De Donato, Persico y Pescapé, 2016).

De esta manera, las organizaciones se han encaminado hacia la creación e inserción de

sistemas integrados para facilitar la gestión y el control de la información (Pitre et al, 2018; Marin et al, 2017)

Sin embargo, según Niebles et al (2018), la amplia gama de productos existentes en el mercado de la tecnología hace necesaria una revisión profunda de las ventajas y desventajas, las cuales varían según el tipo de solución que se requiera. Por tal motivo, el presente trabajo analiza desde una perspectiva estratégica las fortalezas, debilidades, oportunidades y amenazas de herramientas de cloud computing (Azure, OwnCloud) placas de hardware (Arduino, NodeMCU), con miras a determinar cuáles son las más adecuadas para la aplicación a procesos empresariales, productivos o de servicios a nivel general en Colombia.

Metodología

Para lograr un acercamiento oportuno y estructurado del tema, se propone a continuación una revisión bibliográfica que permitió comparar información de los últimos años relacionada con los avances en temas del internet de las cosas, aplicado especialmente a los renglones productivos. Esta revisión se justifica ya que en la medida que se accede a resultados de investigaciones realizadas por actores involucrados con el tema objeto de estudio (Gómez, et al. 2014), se hace posible mayor conocimiento de los procesos de análisis y resultados publicados en fuentes confiables de interés científico y validada por entes a nivel nacional e internacional.

La metodología seleccionada fue del orden cualitativo, de tal manera que se hizo posible realizar un abordaje de la información para

evaluar el tema de estudio mediante técnicas de observación, percepción e interpretación tal como se recomienda para este tipo de fenómenos (Bengtsson, 2016); lo anterior con soporte en un enfoque descriptivo, utilizado con miras a lograr una representación del objetivo central definido, sin necesariamente incurrir en una determinación de sus causas o consecuencias, por el contrario solo observar y reflexionar (Håkansson, 2013). Por lo anterior, se muestran seguidamente los apartados más estratégicos sobre el tema y mediante una técnica de comparación se aprecian de forma organizada ventajas y desventajas de cada herramienta.

Resultados

Análisis FODA: Microsoft Azure

Con el auge del *cloud computing* son muchas las soluciones de TI que compañías como Microsoft están ofreciendo a sus clientes corporativos o particulares. Inicialmente, Azure consiste en un conjunto de servicios en la nube que están orientados a dar respuesta a las necesidades comerciales y gestión de información de un modo sencillo y ágil. En términos concretos, se puede afirmar que es la plataforma de aplicaciones en la nube de Microsoft (Chappell, 2009; Bjørner & Jayaraman, 2015). Seguidamente, se presenta un análisis de sus fortalezas, oportunidades, debilidades y amenazas más relevantes (FODA).

Tabla 1. Análisis FODA de Microsoft Azure

Fortalezas	Debilidades
<ul style="list-style-type: none"> • Ofrece IaaS, SaaS o PaaS. • Quita la carga al usuario asociada con la infraestructura tecnológica (Almacenamiento, sistema operativo, escalabilidad, redes, etc.). • El diseño de la aplicación, desarrollo y pruebas e implementación. • Archivos SDK procesados con visual studio para pruebas en el computador escritorio antes de publicarla en la web. • El usuario paga por lo que consume. • Plataforma interoperable (Linux, Microsoft Windows, Php, open source, MySQL, etc.). 	<ul style="list-style-type: none"> • Costos de capacitación para el desarrollo y gestión de las apps. • Curva de aprendizaje. • Migración de servicios, y reestructuración de arquitectura • Poca captación de programadores y desarrolladores alejados del ecosistema Microsoft. • Aún carece de importantes avances en PaaS respecto a sus competidores.
Oportunidades	Amenazas
<ul style="list-style-type: none"> • Aumento del consumo de servicios de cómputo en la nube. • Cloud Computing para servicios móviles. • Pequeñas y medianas empresas pueden crear infraestructura (accesibilidad). • Cambio del enfoque de productos hacia servicios a empresas. 	<ul style="list-style-type: none"> • Amazon Web Service (AWS). • Google Cloud (GCP). • IBM. • Mercado altamente competitivo en precios. • Variabilidad en la estabilidad de elementos esenciales como ancho de banda, seguridad según países. • Resistencia al cambio por parte de las empresas.

Fuente: elaboración propia, 2019 (con base en Bjørner & Jayaraman, 2015).

Una de las principales fortalezas de Microsoft Azure está asociada con la apuesta que la compañía ha realizado para ofrecer servicios en los tres niveles IaaS, SaaS o PaaS, lo que le ha permitido llegar a un mayor número de usuarios con diversidad de necesidades y demandas (Philip & Shah, 2019). Por otra parte, como lo apoyan Niebles et al (2018) la tecnología en la nube ha disminuido la carga en infraestructura que el desarrollador debía asegurar antes de colocar una aplicación en la web, en este caso, factores como el almacenamiento, la escalabilidad, redes y otros componentes relacionados con la infraestructura ya no es un motivo importante de preocupación. Finalmente, el respaldo de Microsoft hace que este nuevo producto (Azure) cuente con gran aceptación en el mercado, además, la integración con sus herramientas tradicionales (Office 365, Hotmail, etc.), reducen las barreras de entrada de forma considerable para cualquier tipo de organización, indistintamente de su objeto social. Aspectos como su accesibilidad se hace esencial para potenciar el avance a nivel de TI en todo tipo de empresas (Hernández, Cardona & Del Rio, 2017).

La curva de aprendizaje para los usuarios a nivel empresarial puede ser elevada al momento de implementar soluciones como Microsoft Azure. En este sentido, aspectos como la capacitación son factores clave para el éxito de estas herramientas, por ello, es necesario contar con asesoría experta personalizada que planifique un proyecto de estas características (Marín, Inciarte, Hernández & Pitre, 2017). Por otro lado, la gerencia debe tomar en cuenta los costos relacionados con la reestructuración de la arquitectura tecnológica que se posee antes de optar por Azure.

En el mercado del *cloud computing* aún existen potencialidades por explorar, entre estas, la computación en la nube para servicios móviles. Asimismo, el aumento vertiginoso en el consumo de aplicaciones gestionadas bajo este esquema permite que poco a poco los usuarios se vayan familiarizando cada día más con estas soluciones. Adicionalmente, Microsoft Azure

ofrece la oportunidad de crear una infraestructura sólida de TI a pequeñas y medianas empresas sin grandes necesidades de inversión (Yaqoob et al., 2017).

A pesar que Microsoft es una compañía altamente reconocida en el mercado, la competencia cercana como Amazon Web Service (AWS) o Google Cloud, hace que la dinámica competitiva sea compleja. En comparación, servicios IaaS han estado dominados por AWS durante algún tiempo, dejando en desventaja a Azure como una opción. En adición, las organizaciones que ya han iniciado procesos de *cloud computing* con otras soluciones, probablemente no estén dispuestas a cambiar nuevamente de plataforma, dado que los costos de tiempo, inversión y capacitación pueden impactar negativamente las operaciones.

Análisis FODA: OwnCloud

Una de las dificultades para cualquier usuario que utilice la tecnología como medio para satisfacer sus necesidades, sean estas laborales, profesionales, educativas e incluso personales, está relacionada con la capacidad de alojamiento de los numerosos archivos que puede utilizar en su actividad diaria. En respuesta a esta falencia, surge OwnCloud la cual es una aplicación *opensource* creada en el 2010 por un ingeniero con el objeto de proveer un ambiente seguro y privado a las personas que requieren almacenar datos con total autonomía y control (Xu, Groves & Kwok, 2015). Por su importancia, a continuación, se analizan las fortalezas, oportunidades, amenazas y debilidades del proyecto.

En cuanto a las fortalezas, OwnCloud logró posicionarse como una alternativa no comercial que ofrecía a sus clientes mayor control, seguridad y privacidad en los datos. En este sentido, obtuvo acogida por un segmento de usuarios que buscaba opciones fundamentadas en código abierto. Por otra parte, las múltiples funcionalidades que provee, lo identifican como una solución en la nube atractiva (Martini y Choo, 2013).

Tabla 2. Análisis FODA OwnCloud

Fortalezas	Debilidades
<ul style="list-style-type: none"> • Mayor control, seguridad y privacidad de los datos. • Software libre. • Múltiples funcionalidades estándar (compartir archivos, calendarios, editores de texto, gestión de imágenes, etc.). • Multiplataforma. 	<ul style="list-style-type: none"> • Fue reemplazado por el proyecto Nextcloud en el 2016 en Estados Unidos. • Poco apoyo de la comunidad de desarrolladores. • No posee cifrado de extremo a extremo en la versión estándar. • No posee cortafuego de los archivos en la versión estándar.
Oportunidades	Amenazas
<ul style="list-style-type: none"> • Expansión a mercado de usuarios móviles. • Es una alternativa libre y gratuita en la mayoría de las funcionalidades. • Su estructura modular permite un rápido crecimiento a través de aplicaciones fundamentadas en <i>cloud computing</i>. • Al requerir pocos requisitos de hardware y software su uso puede ser expandido 	<ul style="list-style-type: none"> • Grandes competidores como Dropbox o Google Drive • Potencialidad para ataques de ciberdelincuentes • Proyectos como Nextcloud dentro de la misma compañía (Fork o bifurcación en el ámbito del desarrollo de software)

Fuente: elaboración propia, 2019 (con base en Xu, Groves & Kwok, 2015).

La estrategia *fork* (bifurcación a partir del mismo proyecto) empleada por su creador, quitó fuerza en el mercado, dado que el equipo de desarrolladores se enfocó a la opción Nextcloud, surgiendo así lo que se conoce como *canibalismo* en el argot del marketing. Asimismo, en las versiones iniciales existían vulnerabilidades en cuanto a la suplantación de identidad, el cifrado de extremo a extremo y el uso de cortafuegos para archivos, lo cual dejó en la mente de los usuarios una visión débil del producto (Fernández, 2014).

Con relación a las oportunidades, la expansión de la solución de almacenamiento a mercados de usuarios móviles representa una potencial opción para el crecimiento. Igualmente, al ser una alternativa basada en *opensource*, puede abarcar con mayor facilidad segmentos que sus competidores comerciales de pago no pueden alcanzar. Finalmente, los bajos requisitos técnicos (hardware y software) para su uso, constituye una excelente coyuntura para crecer. Cabe destacar que dicha herramienta ha sido utilizada

incluso por entidades gubernamentales como el Ministerio de Comercio, Industria y Turismo (Martini y Choo, 2013).

La presencia de gigantes como Dropbox y Google Drive, hace que la dinámica competitiva sea compleja. Por otro lado, las falencias en la seguridad abren paso a posibles ataques de ciberdelincuentes, impactando en el control de los datos del usuario, propuesta de valor central que ofrece la compañía. Por último, el mismo proyecto *Nextcloud* ha relegado la inversión en desarrollo para Owncloud, representando así una de sus principales amenazas (Martini y Choo, 2013).

Análisis DOFA Arduino

Al igual que existe la tendencia el software libre, Arduino constituye una iniciativa de desarrollo para crear hardware (placas electrónicas) de código abierto nacida en el año 2003; permitiendo a los usuarios configurar multiplicidad de microordenadores para utilizarlos en proce-

tos sencillo como el control automático de dispositivos. En este sentido, las especificaciones y diagramas del hardware libre son de acceso público, facilitando que cualquier persona con los conocimientos y habilidades pueda diseñar

proyectos funcionales bajo sistemas operativos como Windows, MacOS y GNU/Linux (Goilav & Geoffrey, 2016). A continuación, se realiza un análisis de las principales fortalezas, debilidades, oportunidades y amenazas de esta iniciativa.

Tabla 3. Análisis DOFA Arduino

Fortalezas	Debilidades
<ul style="list-style-type: none"> • Especificaciones y código abierto • Flexibilidad para diseñar a la medida • Integración de componentes como <i>Shields</i> para añadir funciones como GPS, reloj, pantallas táctiles, etc. 	<ul style="list-style-type: none"> • Un elevado número de usuarios utiliza el lenguaje Arduino, el cual resulta limitado. • Está dirigido a la comunidad con poca especialización en programación electrónica, limitando su expansión.
Oportunidades	Amenazas
<ul style="list-style-type: none"> • Implementación en diversidad de dispositivos asociados con el internet de las cosas (IoT). • Con la mejora del lenguaje es posible expandirse a otras comunidades. • Mejorar sus prestaciones puede incrementar la participación en el ámbito profesional. 	<ul style="list-style-type: none"> • Existen opciones más sólidas en el mercado para iniciarse como Raspberry o Picaxe. • Poca acogida en el mundo de profesionales de la electrónica. • El acceso a las placas es limitado en algunos países (precio, disponibilidad).

Fuente: elaboración propia, 2019 (con base en Goilav & Geoffrey, 2016).

Al ser una alternativa de hardware abierto, ofrece la posibilidad a gran cantidad de desarrolladores para modelar sus necesidades de manera flexible. Por otra parte, la integración de componentes adicionales como el denominado *Shield* en el campo de la electrónica, permite que un dispositivo sencillo cuente con funcionalidades adicionales que son muy valoradas por los usuarios. Entre tanto, la facilidad para crear a la medida diversas soluciones, la convierten en una alternativa demandada dentro del mundo de la programación de placas (Nayyar y Puri, 2016).

En cuanto a las debilidades, se puede detallar que la herramienta es útil para aquellos que inician un proceso de aprendizaje dentro de la programación de placas electrónicas, no obstante, para aplicaciones en el ámbito profesional esta prestación puede resultar limitada, lo que a su vez impide su crecimiento. Con base en lo anterior, la comunidad de desarro-

lladores manifiesta en ocasiones que el lenguaje de programación que utilizan las placas es limitado para funciones mucho más avanzadas (Casco, 2014).

Con relación a las oportunidades, se observa que el modelo de hardware de código abierto amplía la aplicación de estos recursos sin restricciones comerciales. En este sentido, una clara vía para incrementar las potencialidades de esta herramienta radica en orientarse más hacia las necesidades y demandas de los usuarios avanzados (profesionales) creando categorías de productos para principiante diferenciadas de las más complejas (Lara, Buelna, Quintero y Diaz-Ramírez, 2014).

Analizando las amenazas, estas se ven potenciadas por las debilidades, ya que tiene pocas ventajas frente a opciones más profesionales. Por otro lado, competencia cercana como Raspberry o Picaxe compañías que ofrecen presta-

ciones técnicas más flexibles y adaptadas a las necesidades de los usuarios (D'Ausilio, 2012).

Análisis estratégico: RaspBerry

En el contexto del hardware, RaspBerry resulta una alternativa de bajo coste que consiste en una placa para computadora desarrollado por la Fundación que lleva el mismo nombre en la Universidad de Cambridge en el año 2011. En este sentido, surge como respuesta a la necesi-

dad latente en los usuarios que deseaban programar sus propios dispositivos (*gadgets*) con funcionalidad, eficiencia y precios bajos (Upton y Halfacree, 2014). En este orden de ideas, se presenta como una herramienta clasificada dentro de los microprocesadores, puesto que su arquitectura se asemeja a un miniordenador, por ello, es importante realizar un análisis de sus fortalezas, oportunidades, debilidades y amenazas más destacadas.

Tabla 4. Análisis DOFA RaspBerry.

Fortalezas	Debilidades
<ul style="list-style-type: none"> • Programación en lenguajes de alto nivel como C++, Python o Java. • Potencia de procesamiento, bajo costo. • Puede correr un sistema operativo. • Cuenta con puertos Ethernet y puede ser usado en estándares como NTCS, PAL, HDTV o TV. 	<ul style="list-style-type: none"> • Solo posee el sistema operativo Linux • Carece de opciones para interfaz con sensores externos. • Tiene limitaciones para proyectos que requieran orientación a hardware para dispositivos de uso cotidiano.
Oportunidades	Amenazas
<ul style="list-style-type: none"> • Su filosofía va dirigida al aprendizaje. • Abre la posibilidad para que estudiantes del área de computación pueden interactuar con dispositivos electrónicos. • Su placa presenta compatibilidad con Shields de Arduino. 	<ul style="list-style-type: none"> • Dispositivos como BeagleBone Black cuentan con mayor velocidad de procesamiento (1 Ghz vs 700 MHz de RaspBerry). • Requiere un poco más de conocimiento de electrónica y computación que opciones como BeagleBone Black.

Fuente: elaboración propia, 2019 (con base en Upton y Halfacree, 2014).

Uno de los principales puntos fuertes de RaspBerry es su capacidad de procesamiento en función del costo beneficio que representa. Por otra parte, al surgir como una alternativa para procesos de enseñanza y aprendizaje, su configuración, desarrollo, programación y manejo general es relativamente sencillo en comparación con otras opciones en el mercado. La adaptabilidad para escribir programas en lenguajes como C++ o Python, dan cuenta de su versatilidad y flexibilidad en el campo informático (Salcedo y Cendrós, 2016).

Sus limitaciones en cuanto a las interfaces con sensores externos, hace que sea difícil aplicarlo cuando se trata de proyectos electrónicos que involucren un diseño complejo, sobre todo en dispositivos para uso cotidiano. Por otra parte, microordenadores como BeagleBone Black presentan una mayor velocidad de procesamiento y superan la limitante mencionada anteriormente (Casco, 2014).

Constituye una opción viable para proyectos de aprendizaje, dado que sus características permiten que los estudiantes interactúen con el

hardware de un modo distinto que otras alternativas de microordenadores no poseen. La integración de la placa con Shields de Arduino incrementa considerablemente su potencialidad y uso para otros tipos de proyectos (Casco, 2014).

Sin dudas, existen en el mercado placas con mejor rendimiento tales como la BeagleBone Black, asimismo, a pesar de ser un hardware orientado a la enseñanza aún requiere que el usuario cuente con conocimientos de programación y electrónica un poco más profundo. Al no poseer compatibilidad con sensores externos, se aplicación en proyectos que brinden solución a necesidades cotidianas se reduce considerablemente (Salcedo y Cendrós, 2016).

Análisis estratégico: NodeMcu

De acuerdo con los avances de la tecnología han sido creadas herramientas de gran utilidad que proporcionan mayor velocidad de conexión y generan mejores condiciones para facilitar el desarrollo de las aplicaciones que requieren conexión Wifi. El kit NodeMcu cumple con todas las exigencias para permitir la conexión entre dispositivos que usar Internet of Things (IoT) con la capacidad de recopilar e intercambiar datos (Kashyap, Sharma y Gupta, 2018). Por ello, se realiza un análisis de las fortalezas, debilidades, oportunidades y amenazas de esta forma de conexión.

Tabla 5. Análisis DOFA NodeMCU.

Fortalezas	Debilidades
<ul style="list-style-type: none"> Lenguaje de programación LUA, sencillo a diferencia de C++ para Arduino. Su chip ESP8266 es potente. Código abierto. El bajo costo hace accesible el producto para proyectos de IoT en la comunidad no profesional. Se ejecuta con base en scripts de programas, lo que permite modificar el comportamiento del dispositivo. 	<ul style="list-style-type: none"> Cuenta con 40 módulos distintos. Usa muchos proyectos de código abierto. Se requiere aprender un idioma nuevo e IDE (Entorno de desarrollo integrado).
Oportunidades	Amenazas
<ul style="list-style-type: none"> Diseño abierto. Placa de desarrollo para proyectos de IoT. Alto potencial de crecimiento (exponencial) en la comunidad de desarrollo. 	<ul style="list-style-type: none"> Necesita codificar el lenguaje de programación según la aplicación. Arduino aun representa una serie de ventajas que lo hacen superior.

Fuente: elaboración propia, 2019 (con base en Kashyap, Sharma y Gupta, 2018).

Debido a que es uno de los económicos en el mercado se ha convertido en el más accesible. Además, es muy sencillo de implementar, inteligente e interactivo. El lenguaje LUA se basa en la programación C, siendo fácil de integrar. El chip Wifi ESP8266 es de bajo costo y tiene la capacidad de microcontrolador con pila y protocolo TCP/IP (Protocolo de Control de Transmisión) (Kashyap, Sharma y Gupta, 2018).

Gracias a la colaboración de grupos independientes fueron incluidos más de 40 módulos, razón por la cual cada usuario debe elegir el que se adecúe a su proyecto, creando un firmware que se ajuste a sus requerimientos. Asimismo, se necesita manejar nuevos lenguajes de programación para facilitar el desarrollo del software (Saputra y Lukito, 2017).

Por ser una plataforma de código abierto está diseñado para editar, modificar o construir, convirtiéndolo en una herramienta de mucha utilidad para configurar el IoT. Utilizando las ventajas que ofrece ayuda a mejorar el funcionamiento de ciertas aplicaciones que son vitales para estar ajustados a las tendencias tecnológicas actuales (Bento, 2018).

Para lograr implementar NodeMcu es necesaria la correcta selección del lenguaje de programación y contar con suficientes habilidades para la instalación, pues existe una amplia variedad de opciones de acuerdo con el proyecto que haya seleccionado el usuario (Saputra y Lukito, 2017). Por otra parte, la funcionalidad que ofrece Arduino en algunos componentes hace que sea complejo para NodeMcu posicionarse adecuadamente en la comunidad de desarrolladores.

Análisis estratégico: JAVA

En el ámbito informático, JAVA está catalogado como un lenguaje de programación y plataforma al mismo tiempo, inicialmente creada por la compañía Sun Microsystems en el año 1995 y tuvo el propósito de que los desarrolladores escribieran el código a ser ejecutado en diversidad de dispositivos, dando así rapidez y seguridad (Savitch, 2017). Existen grandes prestaciones de esta herramienta sobre la cual reposa la funcionalidad de múltiples aplicaciones y sitios web en todo el mundo; actualmente, la versión que se muestra en la página oficial del lenguaje es la 8.0. A continuación, se presenta un análisis de sus principales fortalezas, debilidades, oportunidades y amenazas.

Tabla 6. Análisis DOFA JAVA.

Fortalezas	Debilidades
<ul style="list-style-type: none"> • Existe un amplio soporte documental para resolver dudas y problemas. • Librerías nativas que contribuyen a la construcción de diferentes aplicaciones. • Los archivos JVM (Java Virtual Machine) garantizan un alto nivel de seguridad. • No es un lenguaje compilado, es orientado a objetos. • Las aplicaciones son distribuidas. 	<ul style="list-style-type: none"> • El lenguaje para ser orientado a objetos es más complicado que otras opciones como Python. • Es poco recomendable para programadores que están iniciando en el desarrollo. • Al ser un lenguaje interpretado (orientado a objetos) presenta un rendimiento menor con otros lenguajes.
Oportunidades	Amenazas
<ul style="list-style-type: none"> • Expansión a múltiples aplicativos web o móviles por su versatilidad. • Al ser un lenguaje multiplataforma, puede expandirse a otros sistemas operativos (Windows, Linux, Mac, etc.). 	<ul style="list-style-type: none"> • Alta dependencia de la máquina virtual JVM • Al ser un lenguaje interpretado requiere mayores prestaciones de los dispositivos (equipos) cuando se producen varios programas basados en JAVA.

Fuente: elaboración propia, 2019 (con base en Savitch, 2017).

Debido al lenguaje orientado a objetos utilizado por JAVA, se puede hacer un programa que sea similar a la manera en la que piensan los seres humanos, en cuanto a su sintaxis. Por otra parte, las librerías estándar sirven de apoyo para la creación de operaciones tradicionales

para realizar las programaciones, presentando tres componentes fundamentales concentrados en el Java API (Bernard, 2017).

A pesar que es un lenguaje para nada complicado para usuarios expertos, los nuevos desa-

rolladores que están ingresando a la tecnología no deberían utilizarlo, puesto que está basado en formas de programación dirigidas a objetos que requieren la implementación y aplicación de otras técnicas diferentes. Con respecto a otros lenguajes de programación, resulta más complicado para construir el código, por lo que en ocasiones es preferible utilizar otras alternativas (Nasrawt y Lam, 2019).

Como opera de manera independiente no es necesario saber el sistema operativo del ordenador en el que se trabaja. El programa puede implementarse con Linux, Windows, en un celular inteligente o en cualquier servidor. Adicionalmente, la versatilidad que ofrece para ejecutarse en múltiples plataformas hace que sea una herramienta popular en el mundo de la programación (Savitch, 2017).

Es imprescindible un simulador JVM, pues si no se cuenta con esta máquina virtual es imposible el desarrollo de los programas que se

codifiquen en Java. El aspecto del rendimiento no es tan satisfactorio, debido al tipo de lenguaje interpretado y, por lo tanto, depende de los dispositivos. En este sentido, requiere mayor capacidad computacional de los equipos, dado que al ejecutar varios programas con JAVA las características del lenguaje interpretado necesitan recursos adicionales para garantizar la velocidad de las aplicaciones (Bernard, 2017).

Análisis estratégico: Python

En la categoría de lenguajes de programación interpretado se encuentra Python como una de las herramientas de mayor auge en el campo del desarrollo y analítica de datos. Además, resulta ser compatible con múltiples plataformas y su código abierto bajo la denominada Licencia Pública General (GNU) permite que los usuarios puedan acceder a una amplia cantidad de recursos, funciones, librerías y apoyo de la comunidad para resolver situaciones complejas (Monsálvez, 2017).

Tabla 7. Análisis DOFA Python

Fortalezas	Debilidades
<ul style="list-style-type: none"> Al pertenecer a los lenguajes de tipo interpretado su sintaxis es fácil de manejar. Permite generar los códigos bajo patrones comprensibles, flexibles y con mayor rapidez que otras soluciones. Funciona en multiplataformas (Mac, Windows, Linux). Cuenta con una amplia comunidad de apoyo. 	<ul style="list-style-type: none"> En ocasiones su curva de aprendizaje es lenta Las librerías estándar no cumplen con la mayoría de las necesidades, requiriendo que se utilicen las de terceros. Comparado con otros lenguajes de programación no cuenta con una adecuada documentación.
Oportunidades	Amenazas
<ul style="list-style-type: none"> Cuenta con una elevada aplicabilidad en diferentes contextos, situaciones y problemas. Los paquetes desarrollados por terceros amplían considerablemente las funcionalidades de Python. Cuenta con potencialidad para aplicar a proyectos de IoT. 	<ul style="list-style-type: none"> La poca adaptabilidad de servicios en la nube o hosting para soportar Python Según su implementación para una determinada situación, en ocasiones no resulta ser la mejor opción como lenguaje.

Fuente: elaboración propia, 2019 (con base en Monsálvez, 2017).

La característica fundamental de Python es su estructura que permite escribir código de modo interpretativo, lo cual hace que el desarrollador se concentre en ser más productivo en vez de resolver situaciones asociadas con la sintaxis o compilación de los programas. Por otra parte, su flexibilidad para implementarse en diversos contextos (análisis de datos, desarrollo web, visualización, simulación, etc.) hacen que este lenguaje sea uno de los más aceptados por la comunidad actualmente (Monsálvez, 2017).

Para algunos proyectos o soluciones, la curva de aprendizaje de Python puede ser lenta. Asimismo, las librerías estándar de la plataforma no cuentan con las características deseadas, obligando a los desarrolladores a recurrir a fuentes de terceros. Adicionalmente, en comparación con otros lenguajes de programación Python no cuenta con una documentación sólida y robusta que le permita a la comunidad resolver problemas sobre código de modo ágil y sencillo (Chazallet, 2016).

A pesar de su auge en los últimos años, existen servicios y plataformas que no presentan una total integración con este lenguaje de programación, por lo que puede resultar complejo adaptar los programas para que funcionen correctamente, ejemplo de esto se observa en el hosting web cuando las páginas se encuentran desarrolladas bajo Python (Raschka, 2015).

La licencia GNU y la gran comunidad de desarrolladores, permite crecer exponencialmente en el desarrollo de paquetes y librerías que dan solución a diversas situaciones. Esto sin duda

representa una oportunidad de crecimiento importante para el lenguaje, puesto que podrá ser utilizado con mayor fuerza en todo tipo de proyectos, a diferencia de otros que presentan limitaciones en este sentido (Monsálvez, 2017).

Conclusión

En una exploración general, Microsoft Azure es una solución que se está posicionando rápidamente en el mercado, además, representa uno de los cambios de paradigmas más importantes de la compañía, centrandose su atención en los servicios a empresas más que hacia los productos, tal como se ha caracterizado durante las últimas décadas con softwares como Office 365, Windows, etc. No obstante, presenta algunas ventajas importantes relacionadas con la adopción de estas tecnologías a lo interno de la organización, las cuales están relacionadas con las barreras de entrada (capacitación, resistencia al cambio, inversión).

Por otra parte, las placas Arduino, si bien no están dirigidas a un segmento profesional, sirven como muy buena base para la experimentación y el aprendizaje, además, complementado con los Shields y la incorporación de lenguajes de programación de alto nivel como Python, es posible crear soluciones realmente innovadoras a problemas comunes. En la búsqueda de proyectos similares (a nivel empresarial y de producción), se encontró una amplia gama de posibilidades para la adopción de dichas herramientas, por lo tanto, se concluye que son viables para ser insertadas en diversos tipos de empresa por su flexibilidad y costos accesibles.

Referencias

- Barillaro, S., Anand, D., Gopstein, A. M., & Barillaro, J. (2019, October). A Demonstration of Low Power Wide Area Networking for City-Scale Monitoring Applications. In *International Conference on Ad-Hoc Networks and Wireless* (pp. 608-618). Springer, Cham. https://doi.org/10.1007/978-3-030-31831-4_44
- Bengtsson, M. (2016). How to plan and perform a qualitative study using content analysis. *NursingPlus Open*, 2, 8-14. <https://doi.org/10.1016/j.npls.2016.01.001>

- Bento, A. C. (2018). IoT: NodeMCU 12e X Arduino Uno, Results of an experimental and comparative survey. *International Journal*, 6(1). https://www.researchgate.net/profile/Antonio_Bento8/publication/326741893_IoT_NodeMCU_12e_X_Arduino_Uno_Results_of_an_experimental_and_comparative_survey/links/5b61d631458515c4b2590691/IoT-NodeMCU-12e-X-Arduino-Uno-Results-of-an-experimental-and-comparative-survey.pdf
- Bernard, S. (2017). 2: Java Programming Intermediate Concepts For The Fundamentals Of OO Programming (Volume 2). <https://dl.acm.org/doi/book/10.5555/3169240>
- Berrio, S. E. C., Redondo, R. P., & Hernández, H. G. (2018). Impact of ICT on the Generation of New Services Companies. *Contemporary Engineering Sciences*, 11(52), 2591-2599. <http://www.m-hikari.com/ces/ces2018/ces49-52-2018/p/hernandezCES49-52-2018-3.pdf>
- Bjørner, N., & Jayaraman, K. (2015, February). Checking cloud contracts in Microsoft Azure. In *International Conference on Distributed Computing and Internet Technology* (pp. 21-32). Springer, Cham. https://doi.org/10.1007/978-3-319-14977-6_2
- Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. *Future generation computer systems*, 56, 684-700. <https://doi.org/10.1016/j.future.2015.09.021>
- Bustamante, A., Patricio, M. A., & Molina, J. M. (2019). Thinger.io: An Open Source Platform for Deploying Data Fusion Applications in IoT Environments. *Sensors*, 19(5), 1044. <https://doi.org/10.3390/s19051044>
- Casco, S. (2014). Raspberry Pi, Arduino y Beaglebone Black Comparación y Aplicaciones. *vol, 1*, 4-8. <http://jeuazarru.com/wp-content/uploads/2014/10/MiniPCs.pdf>
- Chappell, D. (2009). Introducing windows azure. Microsoft, Dec. https://www.idt-inc.com/wp-content/uploads/sites/4755/2017/05/IntroducingWindowsAzureFinal_5_5_2011_9_55_46_AM.pdf
- Chazallet, S. (2016). *Python 3: Los fundamentos del lenguaje*. Ediciones ENI. https://books.google.com.co/books?hl=es&lr=&id=KRYyvKmZvpwC&oi=fnd&pg=PA31&dq=Python+3:+Los+fundamentos+del+lenguaje&ots=UFc7BAmfM0&sig=IFlfd5sTZiy10-NsR7sb6m-tDs&redir_esc=y#v=onepage&q=Python%203%3A%20Los%20fundamentos%20del%20lenguaje&f=false
- Chiquillo, J. Hernández, H. & Pitre, R. (2018). Business Marketing and Its Transformation with the Insertion of Social Networks. *Indian Journal of Science and Technology*, 11(25). <https://doi.org/10.17485/ijst/2018/v11i25/129095>
- D'Ausilio, A. (2012). Arduino: A low-cost multipurpose lab equipment. *Behavior research methods*, 44(2), 305-313. <https://doi.org/10.3758/s13428-011-0163-z>
- Fernández Romero, S. (2014). *Nube multimedia personal: un caso de estudio con Raspberry Pi, ownCloud y XBMC* (Bachelor's thesis). <http://hdl.handle.net/10016/24943>
- Forsström, S., & Jennchag, U. (2017, June). A performance and cost evaluation of combining OPC-UA and Microsoft Azure IoT Hub into an industrial Internet-of-Things system. In *2017 Global Internet of Things Summit (GIOTS)* (pp. 1-6). IEEE. <https://doi.org/10.1109/GIOTS.2017.8016265>
- Goilav, N., & Geoffrey, L. (2016). *Arduino: Aprender a desarrollar para crear objetos inteligentes*. Ediciones ENI. https://books.google.com.co/books?hl=es&lr=&id=R6RCxQI_H6YC&oi=fnd&pg=PA9&dq=Arduino:+Aprender+a+desarrollar+para+crear+objetos+inteligentes&ots=0bF8seki10&sig=1miVAqhJZ_xETR-WgocCIL04u_4&redir_esc=y#v=onepage&q=Arduino%3A%20Aprender%20a%20desarrollar%20para%20crear%20objetos%20inteligentes&f=false
- Gómez-Luna, E., Fernando-Navas, D., Aponte-Mayor, G., & Betancourt-Buitrago, L. A. (2014). Metodología para la revisión bibliográfica y la gestión de información de temas científicos, a través de su estructuración y sistematización. *Dyna*, 81(184). <https://www.redalyc.org/pdf/496/49630405022.pdf>

- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), 1645-1660. <https://doi.org/10.1016/j.future.2013.01.010>
- Håkansson, A. (2013). Portal of research methods and methodologies for research projects and degree projects. In *The 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing WORLDCOMP 2013*; Las Vegas, Nevada, USA, 22-25 July (pp. 67-73). CSREA Press USA.
<https://www.diva-portal.org/smash/get/diva2:677684/FULLTEXT02.pdf>
- Hernández, H. G., Cardona, D. A., & Del Rio, J. L. (2017). Direccionamiento Estratégico: Proyección de la Innovación Tecnológica y Gestión Administrativa en las Pequeñas Empresas. *Información tecnológica*, 28(5), 15-22.
<http://dx.doi.org/10.4067/S0718-07642017000500003>
- Kaseniades, N., & Paspallis, N. (2019). A Systematic Mapping Study of MMOG Backend Architectures. *Information*, 10(9), 264. <https://doi.org/10.3390/info10090264>
- Kashyap, M., Sharma, V., & Gupta, N. (2018). Taking MQTT and NodeMcu to IOT: Communication in Internet of Things. *Procedia computer science*, 132, 1611-1618. <https://doi.org/10.1016/j.procs.2018.05.126>
- Lara, M., Buelna, A., Quintero, V., & Diaz-Ramírez, A. (2014). Arduino Uno, retos y oportunidades. In *Congreso Nacional de Tecnologías de la Información y Comunicación: CONATIC 2014* (p. 25).
- Li, S., Da Xu, L., & Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, 17(2), 243-259. <https://doi.org/10.1007/s10796-014-9492-7>
- Marín, F. V., Inciarte, A. D. J., Hernández, H. G., & Pitre, R. C. (2017). Estrategias de las Instituciones de Educación Superior para la Integración de las Tecnología de la Información y la Comunicación y de la Innovación en los Procesos de Enseñanza. Un Estudio en el Distrito de Barranquilla, Colombia. *Formación universitaria*, 10(6), 29-38.
<http://dx.doi.org/10.4067/S0718-50062017000600004>
- Martin, W. J. (2017). *The global information society*. Routledge. https://books.google.com.co/books?hl=es&lr=&id=yEQrDwAAQBAJ&oi=fnd&pg=PP1&dq=The+global+information+society&ots=Xlmw9-JgI1&sig=uKVmYyHLnd8s6bKBsGvt3DZ855Y&redir_esc=y#v=onepage&q=The%20global%20information%20society&f=false
- Martini, B., & Choo, K. K. R. (2013). Cloud storage forensics: ownCloud as a case study. *Digital Investigation*, 10(4), 287-299. <https://doi.org/10.1016/j.diin.2013.08.005>
- Monsálvez, J. C. G. (2017). Python como primer lenguaje de programación textual en la Enseñanza Secundaria. *Education in the Knowledge Society*, 18(2), 147-162. <https://www.redalyc.org/pdf/5355/535554766009.pdf>
- Nasrawt, Z. O., & Lam, M. O. (2019). Less-Java, more learning: language design for introductory programming. *Journal of Computing Sciences in Colleges*, 34(3), 64-72. http://files.ccsc-eastern.org/ccsce18/JCSC_34_3.pdf#page=64
- Nayyar, A., & Puri, V. (2016, March). A review of Arduino board's, Lilypad's & Arduino shields. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 1485-1492). IEEE.
<https://ieeexplore.ieee.org/abstract/document/7724514/metrics#metrics>
- Niebles, L., Niebles, W., Rodero, A., Barrios, I., & Hernández, H., (2018). Management Skills and Leadership Styles: A Reflection with a Global Focus. *Contemporary Engineering Sciences*, Vol. 11, 2018, no. 54, 2661 - 2668
- Niebles, W., Rodero, A., Niebles, L., Barrios, I., & Hernández, H., (2018). Entrepreneurship and the Informal Economy: Characterization of Business in Colombia, an Analysis from the Global Entrepreneurship Monitor. *Contemporary Engineering Sciences*, Vol. 11, 2018, no. 54, 2651 - 2659
- OwnCloud. (2019). Página web oficial. Disponible en: www.owncloud.org

- Páez, M. Á. L., Pineda, J. R., & Sánchez, E. C. R. (2019). LiFi y su integración con la internet de las cosas. *Revista vínculos*, 16(1). <https://doi.org/10.14483/2322939X.15281>
- Pflanzner, T., & Kertész, A. (2016, May). A survey of IoT cloud providers. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 730-735). IEEE. <https://doi.org/10.1109/MIPRO.2016.7522237>
- Philip, J., & Shah, D. (2019). Implementing Signature Recognition System as SaaS on Microsoft Azure Cloud. In *Data Management, Analytics and Innovation* (pp. 479-488). Springer, Singapore. https://doi.org/10.1007/978-981-13-1402-5_36
- Raschka, S. (2015). *Python machine learning*. Packt Publishing Ltd. https://books.google.com.co/books?hl=es&lr=&id=GOVOCwAAQBAJ&oi=fnd&pg=PP1&dq=Python+machine+learning&ots=NcgYliQW-K&sig=ZfTis_3KVz9ByfvkRE28xeTKoY&redir_esc=y#v=onepage&q=Python%20machine%20learning&f=false
- Ray, P. P. (2016). A survey of IoT cloud platforms. *Future Computing and Informatics Journal*, 1(1-2), 35-46. <https://doi.org/10.1016/j.fcij.2017.02.001>
- Redondo, R. P., Cárdenas, M. J., Daza, B. C. O., & Cataño, C. L. M. (2018). Integrated Management Systems as a Tool for Strengthening and Competitiveness in Higher Education Institutions in Colombia. *Contemporary Engineering Sciences*, 11 (43). <http://www.m-hikari.com/ces/ces2018/ces41-44-2018/p/hernandezCES41-44-2018.pdf>
- Sachs, J. D., & Vernis, R. V. (2015). *La era del desarrollo sostenible*. Deusto. https://planetadelibrosco0.cdnstatics.com/libros_contenido_extra/31/30978_La_era_del_desarrollo_sostenible.pdf
- Salcedo, M., & Cendrés, J. (2016). Uso del minicomputador de bajo costo “Raspberry Pi” en estaciones meteorológicas. *Télématique*, 15(1), 62-84. <http://telematiquejournal.com/index.php/th/article/download/225/228>
- Saputra, L. K. P., & Lukito, Y. (2017, November). Implementation of air conditioning control system using REST protocol based on NodeMCU ESP8266. In *2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)* (pp. 126-130). IEEE. <https://doi.org/10.1109/ICON-SONICS.2017.8267834>
- Savitch, W. (2017). *Java: An Introduction to Problem Solving and Programming, Student Value Edition Plus MyProgrammingLab with Pearson eText-Access Card Package*. Pearson. <https://dl.acm.org/doi/book/10.5555/2683864>
- Soldatos, J., Kefalakis, N., Hauswirth, M., Serrano, M., Calbimonte, J. P., Riahi, M., ... & Skorin-Kapov, L. (2015). Openiot: Open source internet-of-things in the cloud. In *Interoperability and open-source solutions for the internet of things* (pp. 13-25). Springer, Cham. https://doi.org/10.1007/978-3-319-16546-2_3
- Strickland, J. R. (2018). *Raspberry Pi for Arduino Users: Building IoT and Network Applications and Devices*. Apress. https://books.google.com.co/books?hl=es&lr=&id=PuhgDwAAQBAJ&oi=fnd&pg=PR3&dq=Raspberry+Pi+for+Arduino+Users:+Building+IoT+and+Network+Applications+and+Devices&ots=L5g-cWdfVh&sig=QmnBgYTnMz9JeReXu8ERPQfCn3s&redir_esc=y#v=onepage&q=Raspberry%20Pi%20for%20Arduino%20Users%3A%20Building%20IoT%20and%20Network%20Applications%20and%20Devices&f=false
- Trentin, I. F., Berlemont, S., & Barone, D. A. C. (2018, March). Lightweight M2M protocol: Archotyping an IoT device, and deploying an upgrade architecture. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (pp. 403-408). IEEE. <https://doi.org/10.1109/PERCOMW.2018.8480313>
- Upreti, R. (2017). *Evaluation of Internet of Things Solutions which Includes Cloud Analytic Features* (Master's thesis). https://www.duo.uio.no/bitstream/handle/10852/57751/1/Master_Thesis_Ramesh-.pdf
- Upton, E., & Halfacree, G. (2014). *Raspberry Pi user guide*. John Wiley & Sons. https://books.google.com.co/books?hl=es&lr=&id=g-XhBQAAQBAJ&oi=fnd&pg=PA123&dq=Raspberry+Pi+user+guide&ots=ElutWBAWt&sig=EuA3gyaW6gZrNxsZgIRQ11aS6BU&redir_esc=y#v=onepage&q=Raspberry%20Pi%20user%20guide&f=false

Xu, W., Groves, B., & Kwok, W. (2015). Penetration testing on cloud---case study with owncloud. *Global Journal of Information Technology: Emerging Technologies*, 5(2), 87-94. <https://doi.org/10.18844/gjit.v5i2.198>

Yaqoob, I., Ahmed, E., Hashem, I. A. T., Ahmed, A. I. A., Gani, A., Imran, M., & Guizani, M. (2017). Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges. *IEEE wireless communications*, 24(3), 10-16. <https://doi.org/10.1109/MWC.2017.1600421>