

Implementación y pruebas de IoT

Breve descripción:

Con este componente, se profundizará en los fundamentos necesarios para conocer los diferentes lenguajes de programación, a través del estudio de conceptos y aplicaciones relacionadas con el internet de las cosas, se pretende diseñar y buscar alternativas de transformación digital con componentes del IoT.

Julio 2024

Tabla de contenido

Introducción	3
1. Lenguaje de programación orientado a objetos	5
2. Conceptos asociados al lenguaje	13
2.1. Sintaxis	15
2.2. Estructura general del lenguaje	15
2.3. Editores de código	17
2.4. Conexiones a bases de datos	18
3. Aplicaciones para programar dispositivos IoT	20
3.1. Entornos de desarrollo y Frameworks	21
3.2. Estándares de codificación y plantillas	21
3.3. Implementación de IoT	23
Síntesis	30
Material complementario	31
Glosario	32
Referencias bibliográficas	33
Créditos	35

Introducción

En este componente formativo se realizará un recorrido por los diferentes recursos para la aplicación y gestión del Internet de las cosas. En el siguiente video se introduce el tema.

Video 1. Implementación y pruebas de IoT



[Enlace de reproducción del video](#)

Síntesis del video: Implementación y pruebas de IoT

La POO (Programación Orientada a Objetos) se emplea para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

La programación orientada a objetos se basa en el concepto de crear un modelo del problema de destino en sus programas, disminuye los errores y promueve la reutilización del código.

La Internet de las cosas (IoT) describe la red de objetos físicos ("cosas") que llevan incorporados sensores, software y otras tecnologías con el fin de conectarse e intercambiar datos con otros dispositivos y sistemas a través de Internet.

Las ventajas de la implementación de sistemas IoT, especialmente en su uso industrial (IIoT), son la mejora en la eficiencia por la inmediatez de la información, ahorro de costes y optimización de procesos.

1. Lenguaje de programación orientado a objetos

La programación orientada a objetos se conoce como un paradigma o un modelo de programación, en vez de tratarse de un lenguaje específico o una tecnología, se trata de una forma específica de programar, similar a la manera en la que nos expresamos.

Lenguajes modernos como Java, C++ y Python, utilizan paradigmas para definir los programas, siendo las más populares dentro de la POO, esta aprueba que el código sea reutilizable, organizado y fácil de conservar. Alcanza el principio de desarrollo de software manejado por varios programadores (Roman, 2013).

El primer y más importante concepto que debemos tener en cuenta de la POO es la distinción entre clase y objeto.

Una clase

Es una lista que define genéricamente las características de los objetos de un determinado tipo.

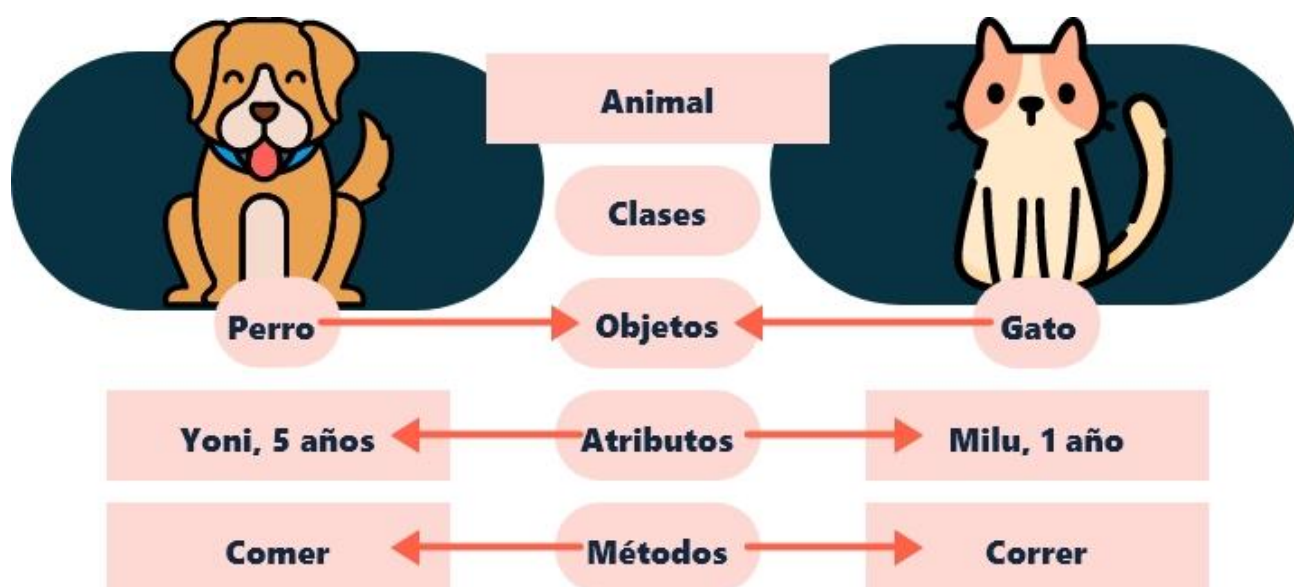
Por ejemplo, una clase para personificar a animales puede llamarse 'animal' y poseer una serie de atributos, como 'nombre' o 'clase de animal' (que regularmente son propiedades) y una sucesión con las actuaciones que estos pueden tener, como comer o correr, y que a su vez se efectúan métodos de la clase siendo (funciones).

Un objeto

Es un tipo de entidad que cuenta con propiedades (datos) y comportamientos (métodos).

En la siguiente figura se presenta un gráfico donde las entidades: Perro, Gato, Ardilla están asociadas a una clase: “Animal”. Esta clase posee unas características propias que diferencian a las entidades entre sí: edad, nombre, raza, entre otros. A su vez, la clase “Animal” tiene acciones como: Comer, Dormir, y Correr.

Figura 1. Ejemplo POO



En estos términos, la programación orientada a objetos ofrece una forma de abstraer entidades del mundo real, a partir de cuatro (4) principios a nombrar:

- 1. La encapsulación.** Un objeto oculta sus datos y solo muestra (interfaz) u ofrece un servicio al universo de las clases. De esta forma, “el gato Milu corre”, una clase que representa una veterinaria podría operar sobre ella, sabiendo que “Milu” corre, pero desconociendo cuál es su proceso funcional.

2. **La abstracción.** Es la formalización del objeto en la clase. ¿Qué es un animal?, ¿cómo identificar un animal?, ¿Qué acciones hace un animal? Con las respuestas a estas preguntas se crea la clase, los atributos y sus diferentes métodos, comportamientos o funciones.
3. **La herencia.** Clases que dependen del concepto de “es hijo de”, “es relacionado con”, “compartes datos con”. Hace referencia al concepto semántico de “herencia”, es decir, entidades que tienen los atributos de sus padres. A partir del ejemplo, se podría crear una clase “mamífero” a partir de la clase “animal”, ya que todos los mamíferos son animales y comparten los mismos atributos.
4. **El polimorfismo.** Capacidad que tienen las clases y/u operaciones para hacer la misma función en diferentes contextos o dominios. En el caso particular, no es lo mismo “comer para un perro” que “comer para un gato”, los dos objetos tendrán el mismo método, pero diferente acción.

C++

Es un lenguaje de programación orientada a objetos que evolucionó de la extensión de lenguaje informático C. C++ se utiliza para realizar programación estructurada de alto nivel y rendimiento, como sistemas operativos, videojuegos y aplicaciones en la nube. C++ también se emplea para desarrollar software de bajo nivel, como controladores de dispositivos.

- “En lenguaje como C++ donde se diseña dentro de la Programación orientada a Objetos, si dentro del programa a realizar todos los miembros

de una clase son públicos y estáticos, en aquel momento equivalen a variables y funciones globales con el prefijo «NombreDeClase:»” (Gervais, 2016).

- “Definición de clases: los datos de una clase se pueden guardar en una estructura, de hecho, en C++ un struct no es otra cosa que una clase con todos los miembros públicos. Como C no permite métodos, usamos funciones globales. El equivalente de C++.” (Sierra, 2015).

```
1 class MiClase
2 {
3 public:
4     int miDato;
5     void hazAlgo(void)
6     {
7         miDato = 5;
8     }
9 };
```

JAVA (SE,EE)

Oracle Java es la plataforma de lenguaje de programación y desarrollo, que reduce costos, acorta los plazos de desarrollo, impulsa la innovación y mejora los servicios de las aplicaciones, conozcamos más:

- **Plataforma informática Java.** Java es un lenguaje de programación que opera por medio de una plataforma informática Java, hay cuatro ediciones

principales de Java: Java Standard Edition (Java SE) Java Enterprise Edition (Java EE) Java Micro Edition (Java ME).

- **Evoluciones.** Ha tenido constantes evoluciones desde sus inicios, Java nace en 1991 con el nombre “OAK”, posteriormente cambiado por Green por problemas legales, y finalmente con la denominación actual JAVA. El objetivo de Java era crear un lenguaje de programación parecido a C++ en estructura y sintaxis, fuertemente orientado a objetos, pero con una máquina virtual propia.
- **Aplicaciones Java modernas.** Concientizan el tiempo de ejecución y la aplicación de Java, para su correcto funcionamiento se debe instalar Java para escritorio. Encontramos muchas aplicaciones del lenguaje Java en casi todos los campos, ya sea una aplicación o desarrollo web, big data, Inteligencia Artificial, desarrollo móvil, etc.
- **Sitio web Java.com.** Oracle Java está disponible en el Acuerdo de licencia de Oracle Technology Network para Oracle Java SE. Esta licencia permite utilizar Java con fines personales y de desarrollo de forma gratuita.
Java 18 (Oracle JDK 18) ofrece miles de mejoras en cuanto a rendimiento, estabilidad y seguridad, incluyendo nueve propuestas de mejora de la plataforma, impulsando así aún más la productividad de los desarrolladores.
- **Java EE.** Es escalable para los proyectos, es decir, permite acoplar más tecnologías según vayan dándose las necesidades del software. Y Java SE es Edición Estándar.

- **Requisitos para programar Java.** Debe tener una sólida comprensión de la programación orientada a objetos. Estar familiarizado con varios diseños y patrones arquitectónicos. Habilidad para escribir bibliotecas en lenguaje Java reutilizables. Conocimiento de los patrones de concurrencia en Java.
- **Aplicación de uso personal.** Si quiere descargar la aplicación de uso personal es gratuita donde se contará con todos los kits de desarrollo y herramientas útiles y compatibles. Para instalar java puede consultar el siguiente tutorial en línea:

https://www.java.com/es/download/help/windows_manual_download.html

En la plataforma Link: <https://www.jdoodle.com/online-java-compiler/>

PHP

Es un lenguaje de código abierto muy reconocido sobre todo para el desarrollo web ya que el mismo puede ser fácilmente incrustado en HTML, una de las grandes ventajas es su simplicidad para aquellos desarrolladores principiantes, por medio de este lenguaje se generan páginas web dinámicas que frecuentemente cambian su contenido, se debe tener en cuenta que las páginas web estáticas son las que sus contenidos permanecen siempre igual.

El esquema de php funciona mediante una petición al servidor, el mismo recibe la información necesaria realizando las consultas a bases de datos, otros servidores u otras páginas web, el servidor responde si es una página web normal estática, pero donde el diseño y creación del mismo ha sido dinámico.

En el editor link: <https://paiza.io/es/projects/new>

Se puede realizar un pequeño saludo:

```
<?php
```

```
//Your code here!
```

```
print "hello";
```

```
?>
```

Ruby

Ruby es un lenguaje de programación interpretado, abstraído y orientado a objetos, su licencia es distribuida bajo software libre, todos los tipos de datos son con objeto, clases, métodos, es descrito como un lenguaje multiparadigma, permitiendo programación procedimental.

“Se utiliza en su mayoría en el desarrollo de aplicaciones web, pero también se utiliza para otros tipos de desarrollo en streaming. Ruby maneja estrechamente el paradigma de POO, así que estas variables son privadas y no logran verse desde fuera de la clase, toda comunicación que se realiza entre objetos se hace con los métodos. Se caracteriza por un lenguaje dinámico, ya que los programas en Ruby no se compilan, todo el código de clases, módulos y métodos se realiza por medio de un intérprete.” (Cupul, 2021).

Editor en línea. Enlace web. Code, create, and learn together with Ruby

Programa: puts 'Hello, world!'

Python

“Python es un lenguaje de programación considerablemente utilizado en la mayor parte de las aplicaciones web, al realizar programas empresariales, la ciencia de datos y el machine learning (ML). Los programadores utilizan Python porque es eficientemente muy fácil de aprender, además de que se puede ejecutar en diferentes plataformas. El software Python es gratuito, se integra bien a todos las características de sistemas y aumenta la velocidad del desarrollo.” (S & G, 2018).

Las empresas requieren programadores expertos en Python, esto debido que es más fácil con tecnologías emergentes como:

Video 2. Python - Lenguaje de programación



[Enlace de reproducción del video](#)

Síntesis del video: Python - Lenguaje de programación

El Big Data y análisis de datos es una de las tendencias actuales por su simplicidad y número de bibliotecas para el procesamiento de datos, analizando y gestionando en tiempo real.

La Minería de datos o data mining puede predecir las futuras tendencias, y con Python puede limpiar y organizar los datos mediante usos de algoritmos de aprendizaje automático simplificando el análisis de datos.

Los científicos de datos trabajan con este lenguaje de programación con un gran número de datos, debido a las grandes bibliotecas y parámetros requeridos en esta práctica. También, gran parte del avance de las IA se le debe a Python gracias a su robustez y facilidad de escritura.

Python es un lenguaje versátil, positivo y rápido, es muy útil formando cadenas de bloques, e incluso permite a los programadores crear cadenas de bloques naturales para blockchain en menos de 60 líneas de código, haciendo sencillo algo muy complicado.

El Machine Learning consiste en asimilación de competencias por sí mismas (Inteligencia Artificial). A medida que se van procesando datos, python juega un papel fundamental en el tratamiento de datos.

2. Conceptos asociados al lenguaje

Se concluye que cada lenguaje de programación utiliza una lógica similar, pero los conceptos pueden cambiar. Los lenguajes de programación se definen como procesos donde se crea, se depura y se compila información para crear programas, utilizando códigos o instrucciones que el desarrollador convierte en lenguaje de programación fuerte, de bajo, medio o alto nivel, compilados y declarativos u orientados a objetos.

Para iniciar a programar, se deben tener en cuenta los siguientes conceptos básicos de programación:

Algoritmo. Es la base para la programación, al dominar los algoritmos es posible encontrar soluciones en cualquier lenguaje de programación.

Variable. Son unidades de almacenamientos de datos que se utilizan para programar y se reservan en espacios en la memoria RAM.

Función. Hace referencia a los bloques que realizan tareas específicas, el comando es: `function` y se reutiliza este código para realizar la redacción de líneas de manera más rápida.

Tipos de datos. Se refiere a conjuntos de valores que significan lo mismo. Los tipos de datos pueden ser numéricos, booleanos (dos opciones), fecha / Hora, de texto (cadena), variant o de objeto.

2.1. Sintaxis

La sintaxis dentro de todo lenguaje de programación es la combinación de reglas y símbolos que se consideran declaraciones o expresiones estructuradas en un lenguaje de programación.

- Ideas principales y puntos de conexión:
- Describe a la escritura del código.
- Enfrenta a la semántica en sí el significado.
- El método semántico suele ser posterior al sintáctico.
- El método semántico es necesario para completar el análisis sintáctico, y se realizan juntos o al mismo tiempo.
- Ruby es un lenguaje orientado a objetos, su sintaxis y semántica presentan que en su mayoría todos los tipos son objetos.
- Muchos operadores son métodos de los objetos (Aprendiendo Ruby y Rails. Parte 1. Introducción a Ruby, 2015.)

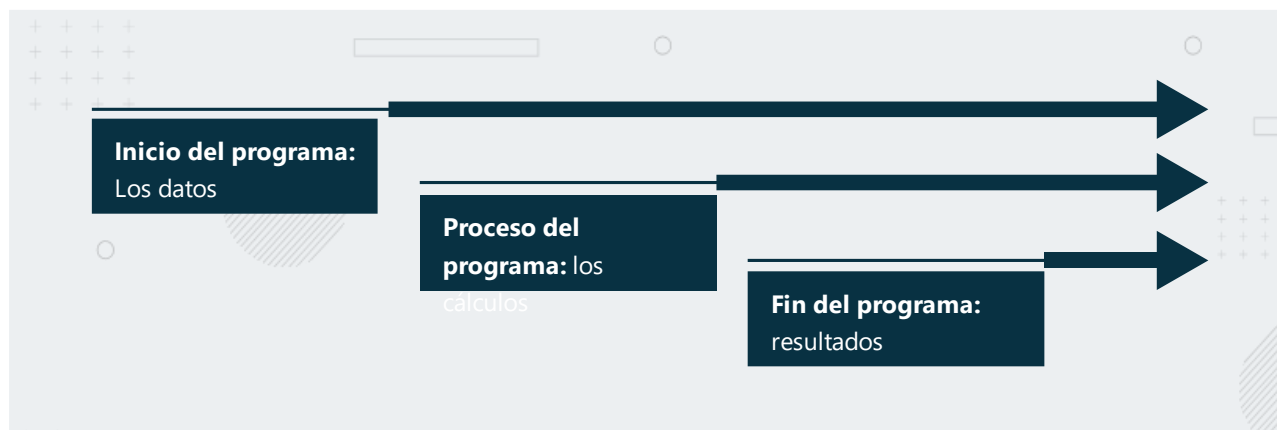
2.2. Estructura general del lenguaje

“Un programa informático es una serie de acciones (instrucciones o comandos) que manejan un conjunto de objetos (datos e información), depende mucho el lenguaje que estés manejando para conformar su estructura. Los bloques que existen son declaraciones e instrucciones.” (Inicio - Introducción a Ruby, 2018.)

Las declaraciones son los objetos que utiliza el programa como variables y archivos (Variables, archivos, etc.) las instrucciones son los resultados que se esperan

por las acciones y operaciones realizadas. En la figura 2, presenta la estructura básica de un programa en Ruby.

Figura 2. Estructura de un programa en Ruby



Entrada de datos: Almacenan en la memoria interna datos procedentes de un dispositivo externo.

Proceso o algoritmo: Instrucciones que modifican los objetos de entrada y en ocasiones y en ocasiones crean otros nuevos.

Salida de resultados: Conjunto de instrucciones que toman los datos finales de la memoria interna y los envían a los dispositivos externos.

La estructura básica de un programa informático se muestra en la tabla 1.

Tabla 1. Especificación de la estructura de un programa en Ruby

Cabecera	Se especifica por medio del nombre de programa, los datos de entrada y datos de salida.
Funciones	Son las diseñadas por el desarrollador y se usa las ocasiones necesarias.
Declaraciones	Las mismas se definen en los tipos de variables, constantes y los nuevos tipos de datos que se den.
Asignaciones	Valores iniciales declarados previamente.
Entradas	Almacenan en memoria algunos valores.
Control	Las mismas pueden ser alternativas y repetitivas
Salidas	Son las que muestran los resultados obtenidos.

2.3. Editores de código

En la medida del avance de la tecnología en plena cuarta revolución industrial, los programas de computadora que nos ayudan a realizar códigos fuente de los diferentes proyectos de programación, evolucionan constantemente y pueden combinar diferentes lenguajes o editores: Python, JavaScript, HTML, etc. Al crear una página Web es importante considerar que existen editores de códigos compatibles con sistemas operativos como Windows, MacOS o Linux.

Las características de los editores de código en cada sistema operativo son, capacidad de almacenar proyectos, sintaxis en varios idiomas, funciones de búsqueda y reemplazo, funciones de mostrar y ocultar secciones, edición de macros. Algunos editores son:

- **Coda:** es una aplicación web que funciona como editor de código para MacOS.
- **Notepad++:** es un editor versátil y soporta lenguajes como HTML, JavaScript y php.
- **Visual Studio code:** es desarrollado por Microsoft se usa con diferentes lenguajes de programación, incluidos C (C, C ++, C #), F #, HTML entre otros, incluidos PHP, Java, Ruby y muchos otros. Une un conjunto de funciones que se reforman según el idioma que esté utilizando.
- **Atom:** es un editor de JavaScript ligero y tiene una gran cantidad de extensiones donde se puede personalizar su funcionamiento.
- **Ultraedit:** es uno de los lenguajes donde se puede editar multitud de lenguajes de programación en HTML; JavaScript, ASP.

2.4. Conexiones a bases de datos

Una conexión de base de datos es un registro de configuración en el que se especifican los detalles físicos de una base de datos, como el tipo de base de datos la versión, y los parámetros para habilitar una conexión. Desde una aplicación por ejemplo en entornos web, se puede concebir la interacción con la base de datos, donde existe un cliente que invoca una petición a la aplicación web que está alojada en

un servidor web, desde ahí el backend de la aplicación interactúa con la base de datos respectiva, generando todos los accesos y dando resultados de nuevo al core de la aplicación web y de allí mostrar resultados a las vistas (frontend) del cliente respectivo.

Figura 3. Conexión base de datos



Una base de datos permite acumular gran número de información de una forma constituida para su futura consulta, construcción de búsquedas, nuevo ingreso de datos, etc. Todo esto lo aprueba si se realiza de una forma rápida y simple desde un computador.

En proyectos de IOT en la nube se pueden adquirir servicios de base de datos como Amazon y Dynamo DB.

3. Aplicaciones para programar dispositivos IoT

Los lenguajes más importantes y utilizados en el desarrollo de Internet de las cosas- IoT son Java, lenguaje C, Python y JavaScript.

Los dispositivos IoT cuentan con sensores que registran y transforman lo físico o químico en datos electrónicos que podemos manejar.

Cualquier dispositivo IoT permite conectarse a otros dispositivos y aplicaciones IoT para transmitir información utilizando protocolos estándar de Internet, veamos los siguientes:

Aplicaciones para programar dispositivos IoT

- a. **Phyton.** Es un lenguaje interpretado de alto nivel, y es uno de los más populares para el desarrollo de IoT.
- b. **C.** Es un lenguaje de nivel medio comprende toda la arquitectura de proyectos IoT por las grandes propiedades como la portabilidad y el número de bibliotecas que tiene.
- c. **Lua.** Lenguaje de nivel alto maneja funciones de descripción de datos y para que funcione correctamente se incorpora a un cliente HOST.
- d. **Golang.** Go - aprovecha al máximo el hardware lo que lo hace útil y apropiado en IoT Java: esté código puede ejecutarse en cualquier plataforma para el adecuado desarrollo de proyectos de Internet de las cosas.

3.1. Entornos de desarrollo y Frameworks

Un entorno de desarrollo o IDE es un software que combina en un solo lugar todas las herramientas para desarrollar un proyecto, se puede decir que es la combinación de varios procesos de programación.

Los entornos de desarrollo cuentan con un editor, un compilador, y un desbloqueador de las gestiones de código genérico, las cuales se dan en entornos de los lenguajes de desarrollo Python y Java.

Por medio de los entornos de desarrollo, los programadores pueden testear los diferentes segmentos de código, con esto se logra identificar cualquier error de la aplicación antes de salir al mercado.

Los frameworks pueden ser una de las herramientas más importantes para los programadores en general. Se trata de toda una base de datos en la que se establecen proyectos completos, y que pueden ser utilizados las veces necesarias.

3.2. Estándares de codificación y plantillas

Los estándares de codificación concentran elementos de ingeniería sólidos para la programación en sus diferentes lenguajes y constituyen la base de cualquier orientación preventiva.

Los estándares de calidad más relevantes en la construcción de software, de acuerdo con las normas ISO, son los siguientes:

Estándares de Calidad en la construcción de un software

- **Norma ISO 25000.** La norma consta de 5 áreas enfocadas a la gestión, el modelo, la medición del mismo, los requisitos y la evaluación. Donde se asegura la máxima calidad y confianza en los desarrollos de software.
- **Norma ISO 9126.** Esta norma se compone dentro de los estándares y plantillas en 6 aspectos que se deben tener en cuenta que son la funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad. Donde se puede conocer que los desarrollos de software cumplan con las necesidades de los usuarios.
- **Norma ISO 9001.** Mediante el software de calidad, donde la confianza y reputación de los usuarios son los beneficios de este modelo de gestión.
- **Norma ISO 15504.** Es la norma conocida por la mejora de procesos, contando con los estándares de calidad de software.

Las herramientas de integración ayudan a las organizaciones a conseguir mayor eficiencia, productividad y agilidad ya que integran diferentes aplicaciones de software y servicios, interfaces de programación de aplicaciones (API), datos y dispositivos para automatizar los procesos empresariales y ofrecer al usuario un producto de calidad.

Las herramientas de gestión de cambios son cualquier cosa que se pueda utilizar para aliviar las tensiones que surgen durante un cambio en el proceso, el personal o el software.

Azure Pipelines

Son servicios en la nube que combina y realiza entrega continua, donde compila y prueba automáticamente el código para finalmente colocar el mismo a disposición de otros usuarios.

Jenkins

Es una herramienta de integración continua, siendo la misma de código abierto y se puede configurar en entornos productivos.

Gitlab

Tiene un servicio en el cual se construye y prueba el software cada vez que el programador actualiza el repositorio.

3.3. Implementación de IoT

Cuando miramos las implementaciones de proyectos en IoT pensamos que el conocimiento de inteligencia empresarial no deja de transformarse y, uno de los últimos negocios relevantes se debe a las aplicaciones IoT.

Se denomina aplicaciones IoT a las soluciones de software diseñadas para realizar tareas, en este caso, en base a la información procedente de dispositivos conectados a una red. Según datos de IDC, para el 2020, la cantidad de objetos conectados habrá alcanzado los 30 miles de millones, ofreciendo una oportunidad de ingresos de 1.7 miles de millones de dólares americanos, para el año 2030 la proyección sería de 125 mil millones conectados.

En la implementación de proyectos IoT se puede conectar hoy en día cualquier objeto como:

- Tráfico: como proyectos en vehículos, semáforos.
- Objetos personales: ropa, zapatos.
- Hogar: timbre, garaje, puertas hogar, electrodomésticos.
- Equipos electrónicos: todos.
- Dispositivos de salud en control
- Dispositivos de deporte

Dentro del metaverso y comercio electrónico los proyectos de IoT serían fundamentales proyectos de esta línea para los diferentes almacenes o empresas en mejoramiento de los procesos de ventas y afinidad con sus clientes.

Pruebas de conectividad

Las pruebas de conectividad se miden mediante herramientas de diagnóstico en el cual se puede verificar la conectividad entre los puntos de la red. Es posible realizar un análisis de la configuración y los datos del plano en vivo. Adicionalmente, las pruebas de conectividad se usan para ayudar a controlar problemas de conexión de red, como lo son:

- Opciones incoherentes no deseadas.
- Opciones obsoletas en la configuración esto se puede dar según las migraciones o cambios que tenga la arquitectura de hardware.

- Error en la variedad de servicio y funciones de la red.

Pruebas de funcionamiento

Dentro de las pruebas de funcionamiento se puede realizar el test de conectividad mediante el valor de shapley, este es un método de optimización de recursos y es posible aplicarlo a la conectividad o transmisión de paquetes, mediante el ping también es posible tomar los valores de conectividad de la red.

Figura 4. Testeo ping con una página web

```
C:\>ping www.sena.edu.co -t

Haciendo ping a www.sena.edu.co [186.113.7.153] con 32 bytes de datos:
Respuesta desde 186.113.7.153: bytes=32 tiempo=17ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=16ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=16ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=16ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=15ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=16ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=16ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=16ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=16ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=17ms TTL=247
Respuesta desde 186.113.7.153: bytes=32 tiempo=16ms TTL=247
```

Tomando los datos obtenidos al realizar el testeo se procede a diligenciar una tabla como la siguiente:

Tabla 2. Plantilla de seguimiento de conexiones punto a punto.

Nodo	Valores de RED tiempo	Excelente	Regular	Malo
1				
2				
3				
4				
5				

En el anterior escenario podemos decir que tenemos $N = 5$ donde sería los nodos que conforman la Red Power Line Communications (PLC), dando valores de red solicitados que se pueden extraer los valores de red tiempo por puntos y realizar la comparación entre nodos dando rangos según la topología de red.

Ejemplo: Excelente rango de 1 – 20 MS.

Regular sería de 1 – 30 MS.

Malo sería rango de 1 – 40 Ms pero con interrupciones y fallas.

Validación de rendimiento

Los puntos de conexión inalámbricos IoT trabajan durante años y logran instalarse fácilmente en el lugar deseado. Con la aparición continua de nuevas aplicaciones esto puede implicar que estas terminales deban usarse en ambientes de RF realmente adversos. Los medidores inteligentes instalados en bodegas o sótanos, por ejemplo, tienen que hacer frente a condiciones complicadas de cobertura. Nuevos dispositivos, como los sensores de estacionamiento, deben ser muy fuertes y resistentes al agua, lo cual es un gran desafío para el diseño de RF global. El rendimiento de las conexiones inalámbricas de los dispositivos IoT es un factor crucial para el éxito de las aplicaciones del IoT y requiere de atención al detalle, así como de un tiempo y esfuerzo considerables.

Documentación de pruebas y resultados

Un dispositivo IoT, como su nombre indica, es un dispositivo que se conecta a internet para realizar alguna operación, ya sea almacenar datos, consultar un servicio, etc.

Las pruebas IoT, permite al usuario controlar dispositivos de forma remota a través de una red. El IoT testing es un nuevo enfoque de prueba que involucra un alcance más amplio y complejo de pruebas de software tradicionales.

Video 3. Documentación de pruebas y resultados



[Enlace de reproducción del video](#)

Síntesis del video: Documentación de pruebas y resultados

Pruebas de Integración: Este tipo de prueba avala una interacción constante del software de IoT con los dispositivos inteligentes, se realiza testeos a los diferentes software o aplicaciones.

Pruebas de seguridad: Asegura que el software IoT compile, examine y resuelva datos correctamente, sin filtraciones a través de todos los puntos de conexión, redes y sistemas inteligentes, también se pueden aplicar testeos con owasp top 10 para encontrar fallas y vulnerabilidades en las aplicaciones WEB, donde se busca bajar el impacto de los ataques a las mismas.

Pruebas de usabilidad: Las pruebas de usabilidad de IoT ayudan a escuchar a los usuarios con cuidado, identificando los puntos más sensibles del flujo de trabajo de su aplicación.

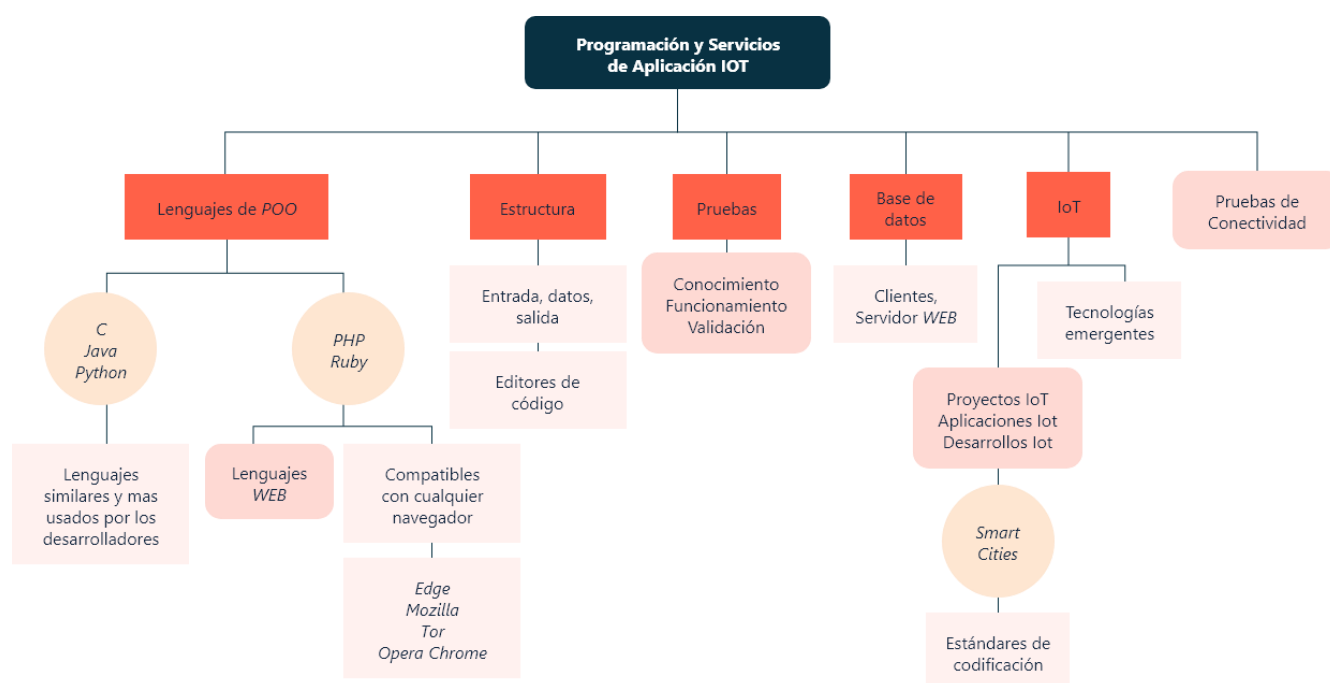
Pruebas de rendimiento: A través de escenarios de prueba de carga, estrés e Internet, se puede simular y probar productos de IoT para aumentar la productividad, la estabilidad de la carga y hacer que el código se ejecute sin problemas dentro de ecosistemas de IoT inusuales.

Pruebas de compatibilidad: Es un paso esencial de las pruebas de IoT que realiza una interacción fluida y sin errores entre el software de IoT y diferentes dispositivos inteligentes, plataformas, capas de red y sistemas operativos.

Pruebas de confiabilidad y escalabilidad: Estos tipos de pruebas ayudan a construir los entornos de IoT correctos, optimización e implementación de nuevas funciones, utilizando simuladores de sensores virtuales y herramientas de prueba progresivas de IoT.

Síntesis

El primer objetivo de IoT es montar una interconectividad digitalizada y automatizada entre objetos, creando nexos que permitan una continua comunicación entre las cosas a través de sensores, softwares, Inteligencia Artificial y sistemas computarizados. Para potenciar su aplicación, el IoT trabaja con otras tecnologías punteras como C++, Java (SE, EE), PHP, Ruby. y Python.



Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
Lenguaje de programación orientado a objetos	Norma Iso 25000 https://e-collection-icontec-org.bdigital.sena.edu.co/normavw.aspx?ID=6385	Página Web	https://iso25000.com/index.php/normas-iso-25000
Lenguaje de programación orientado a objetos	Edureka. (2019). Learn IoT In 4 Hours [Video]. YouTube. https://www.youtube.com/watch?v=h0gWfVCSGQQ	Video	https://www.youtube.com/watch?v=h0gWfVCSGQQ
Aplicaciones para programar dispositivos IoT	Edureka. (2019). Learn IoT In 4 Hours. [Video]. YouTube. https://www.youtube.com/watch?v=h0gWfVCSGQQ	Video	https://www.youtube.com/watch?v=h0gWfVCSGQQ

Glosario

Azure: servicios de computación en la nube de Microsoft.

Framework: es un marco de trabajo específico, con instrucciones y estructuras, para la implementación de alguna acción.

IoT: internet de las cosas.

Partner: socio u operador de servicios en la nube (CSN) es una parte que se dedica a respaldar o proporcionar actividades de soporte auxiliar a las actividades del cliente o proveedor de servicios cloud.

SaaS: software como servicio.

Script: se refiere a fragmentos de código de programación que pueden ejecutar una o varias funciones.

Smart cities: ciudades inteligentes.

TI: (IT) Abreviatura de Tecnología de la información.

Referencias bibliográficas

Conza, A. (15 de abril de 2018). Estándares para Diseño e Implementación de Bases De Datos Relacionales.

<http://adrianconza.com/:https://adrianconza.medium.com/estandares-de-bases-de-datos-2bee81b53447>

Gervais, L. (2016). Aprender la programación orientada a objetos con el lenguaje C#. Ediciones ENI

Inicio - Introducción a Ruby. (2018.). <https://makeitrealcamp.gitbook.io/ruby-book/>

ISO/IEC 17789:2014. (2021). <https://ccsp.alukos.com/standards/iso-iec-17789-2014/>

Román, L. L. (2013). Metodología de la programación orientada a objetos. Alfaomega.

S, G. & A, V. (2018). Introduction to Python Programming (Illustrated). CRC Press

SG Buzz, (2015) Aprendiendo Ruby y Rails. Parte 1. Introducción a Ruby. <https://sg.com.mx/content/view/564>

Somasundaram, K. (2014). Introduction to Java Programming. Penguin Random House.

Sotaquirá, M. (15 de junio de 2021). Guía completa para el Manejo de Datos Faltantes. <https://www.codificandobits.com/blog/manejo-datos-faltantes/>

Velthuis, M. P. (2019). Calidad de datos. Bogotá: Ediciones de la U.

<https://www.perlego.com/es/book/2165228/calidad-de-datos-pdf>

Créditos

Nombre	Cargo	Regional y Centro de Formación
Milady Tatiana Villamil Castellanos	Responsable del Ecosistema	Dirección General
Claudia Johanna Gómez Pérez	Responsable de Línea de Producción	Regional Santander - Centro Agroturístico
César Antonio Villamizar Núñez	Experto Temático	Regional Norte de Santander- Centro CIES
Diego E. Acevedo Guevara	Diseñador Instruccional	Regional Santander - Centro Industrial del Diseño y la Manufactura
Carolina Coca Salazar	Asesor Metodológico	Regional Distrito Capital - Centro de Diseño y Metrología
Rafael Neftalí Lizcano Reyes	Responsable Equipo de Desarrollo Curricular	Regional Santander – Centro Industrial del Diseño y la Manufactura
Sandra Patricia Hoyos Sepúlveda	Corrección de estilo	Regional Distrito Capital - Centro de Diseño y Metrología
Sandra Paola Morales Páez	Evaluada Instruccional	Regional Santander - Centro Agroturístico
Marcos Yamid Rubiano Avellaneda	Diseñador de Contenidos Digitales	Regional Santander - Centro Agroturístico
Pedro Alonso Bolivar González	Desarrollador Fullstack	Regional Santander - Centro Agroturístico
Maria Alejandra Vera Briceño	Animadora y Productora Multimedia	Regional Santander - Centro Agroturístico
Lucenith Pinilla Moreno	Actividad Didáctica	Regional Santander - Centro Agroturístico

Nombre	Cargo	Regional y Centro de Formación
Yineth Ibette González Quintero	Validadora de Recursos Educativos Digitales	Regional Santander - Centro Agroturístico
Diana Lizeth Lozada Díaz	Evaluadora para contenidos inclusivos y accesibles	Regional Santander - Centro Agroturístico