**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

| PROGRAMA DE FORMACIÓN | Desarrollo e implementación de soluciones para la transformación digital |
| --- | --- |

| COMPETENCIA | 220501106 - Configurar dispositivos de cómputo de acuerdo con especificaciones del diseño y protocolos técnicos. | RESULTADOS DE APRENDIZAJE | 220501106-02- Programar las plataformas y aplicaciones de acuerdo con la solución IoT y manuales.  220501106-03- Verificar la operación de la red IoT de acuerdo con el diseño de la solución. |
| --- | --- | --- | --- |

| NÚMERO DEL COMPONENTE FORMATIVO | CF09 |
| --- | --- |
| NOMBRE DEL COMPONENTE FORMATIVO | Implementación y pruebas de IoT |
| BREVE DESCRIPCIÓN | Con este componente, se profundizará en los fundamentos necesarios para conocer los diferentes lenguajes de programación diseñar y buscar alternativas de transformación digital con componentes del IoT |
| PALABRAS CLAVE | Código Fuente, Ejecutable, Programación. |

| ÁREA OCUPACIONAL | 6 - VENTAS Y SERVICIOS |
| --- | --- |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS:**

**Introducción**

**1. Lenguaje de programación orientado a objetos**

**2. Conceptos asociados al lenguaje**

2.1 Sintaxis

2.2 Estructura general del lenguaje

2.3 Editores de Código

2.4 Conexiones a bases de datos

**3. Aplicaciones para programar dispositivos IoT**

3.1 Entornos de desarrollo y Frameworks

3.2 Estándares de codificación y plantillas

3.3 Implementación de IoT

1. **INTRODUCCIÓN**

En este componente formativo, tendrá la capacidad de crear aplicaciones que gestionen recursos IoT a través de sistemas de información basado en tecnología de punta, así mismo, dispondrá de la apropiación de conocimiento para realizar pruebas del código desarrollado o de otras soluciones que analicen y midan la eficacia y eficiencia de un aplicativo.

La tecnología IoT son dispositivos que están conectados a una red de datos sea Internet o local, a nivel de desarrollo de software muchas aplicaciones se deben crear para la interacción con estos dispositivos. Veamos este tipo de programación:



1. **DESARROLLO DE CONTENIDOS:**

**1. Lenguaje de programación orientado a objetos**

La programación orientada a objetos se conoce como un paradigma o un modelo de programación, no es un lenguaje específico, o una tecnología sino una forma específica de programar, más cercana a cómo expresaremos las cosas en la vida real que otros lenguajes de programación.

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 genéricamente de cómo van a ser los objetos de un determinado tipo.

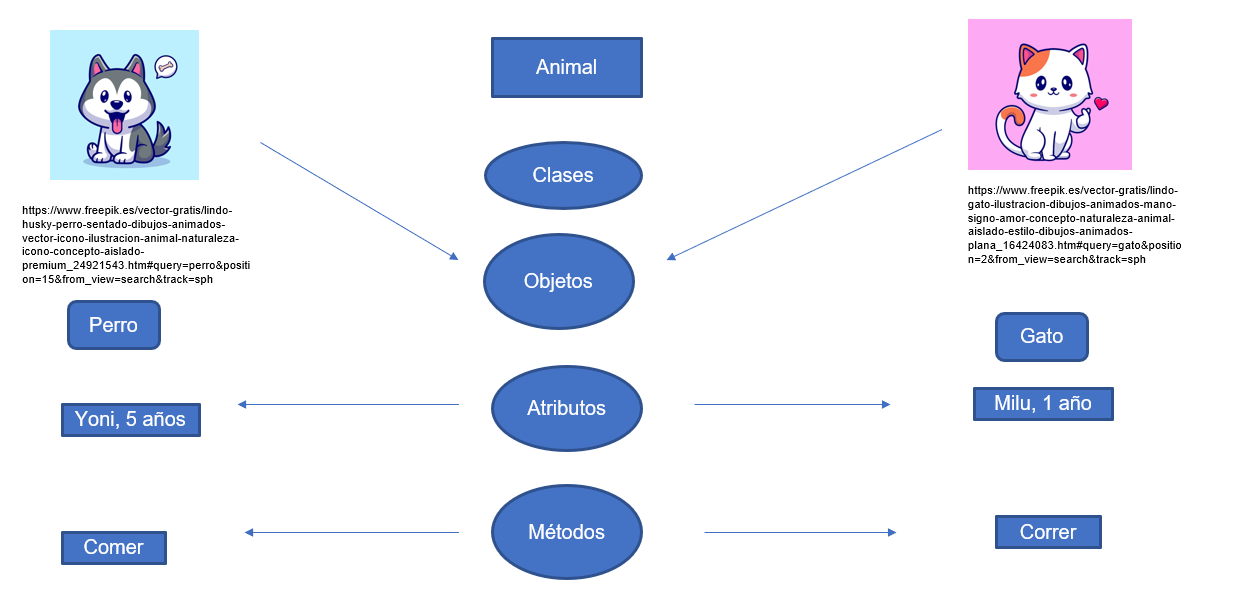
Por ejemplo, una clase para personificar a animales puede llamarse ‘animal’ y poseer una serie de atributos, como ‘nombre’ o ‘clase de animal’ (que regularmente son propiedades) y una sucesión con las actuaciones que estos pueden tener, como comer o correr, y que a su vez se efectúan métodos de la clase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

**Un objeto**

* Se puede ver en el gŕafico 1. Una entidad “perro” ,”gato”, “ardilla” tiene una clase asociada “animal”. Está clase animal posee unas características propias que lo distinguen unos de otros, por ejemplo, edad , nombre, raza , entre otros. A su vez, el animal tiene acciones como “comer”, “dormir” y “correr” .

**Figura 1**

*Ejemplo POO*



A partir de está clase se pueden crear las instancias, es decir, cuando la clase cuenta con memoria en el sistema operativo que va a operar sobre el software.

De está forma, se pueden crear las instancias para un Perro de nombre=”Yoni” con una edad de “5” años o bien la representación de un gato de nombre=”Milu” y de edad=”1” año. Los dos están a partir de la clase animal, pero son dos instancias diferentes. Por lo que al llamar a sus métodos logra poseer resultados diferentes. Los dos comparten la lógica de sus requerimientos funcionales, pero cada uno de forma independiente tiene su estado.

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:

| Referencia de imagen:  <https://as2.ftcdn.net/v2/jpg/04/34/86/55/1000_F_434865564_zYlEORqVS8dQmMtTkCZNgeFSjbm8TkwJ.jpg> | La encapsulación: un objeto oculta sus datos y solo muestra (interfaz) u ofrece un servicio al universo de las clases. De está 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. |
| --- | --- |
| Referencia de imagen:  <https://as2.ftcdn.net/v2/jpg/02/25/82/97/1000_F_225829756_t3zKb8mU9NeJaWO7LjniYGSzjGbq9BYs.jpg> | La abstracción: es la formalización del objeto en la clase. ¿Qué es un animal?, ¿cómo identificar un animal?, ¿Qué acciones hace un animal?. Con estas respuestas , creó la clase, atributos y sus correspondientes atributos. |
| Referencia de imagen:  <https://as1.ftcdn.net/v2/jpg/01/30/18/50/1000_F_130185012_TTHYtNDvHwIJ93mQNnOiHyV8KFZ4GOmZ.jpg> | 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. |
| Referencia de imagen:  <https://as1.ftcdn.net/v2/jpg/04/22/28/56/1000_F_422285641_r0IfiGKUHXykHRKS9uXh5Z4cuPVFMqz3.jpg> | 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 sobre |

**C++**

Es un lenguaje de programación orientado a objetos. Los cuatro principios básicos de la programación orientada a objetos son: Abstracción: modelar los atributos e interacciones pertinentes de las entidades como clases para definir una representación abstracta de un sistema.

En lenguaje como C++ donde los mismos están diseñados 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  2  3  4  5  6  7  8  9 | class MiClase  {  public:      int miDato;      void hazAlgo(void)      {          miDato = 5;      }  }; |
| --- | --- |

**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:



**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 su contenido cambia, debemos tener en cuenta que las páginas webs 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: <https://replit.com/languages/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:



**2. Conceptos asociados al lenguaje**

Pudimos analizar que en cada lenguaje de programación la lógica es la misma pero los conceptos de cada uno pueden cambiar. Los lenguajes de programación se podrían definir como el proceso donde se crea, depura y compila creando programas de computadora, la misma se realiza por código o instrucciones que el desarrollador realiza en un lenguaje de computadora se cuenta con lenguajes de programación fuertes, de bajo medio o alto nivel, compilados, y declarativos u orientado a objetos.

Para iniciar a programar debemos tener los conceptos básicos de programación, como:



**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 los puntos de conexión :

| Imagen de referencia:  <https://as2.ftcdn.net/v2/jpg/03/17/70/03/1000_F_317700340_L2N619hykKTaeMowfOP74xBctbKnPaPa.jpg> | * Describe a la escritura del código * Enfrenta a la [semántica](https://es.wikipedia.org/wiki/Sem%C3%A1ntica_de_lenguajes_de_programaci%C3%B3n) 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](https://es.wikipedia.org/wiki/Concurrencia_(inform%C3%A1tica)). * 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.) |
| --- | --- |

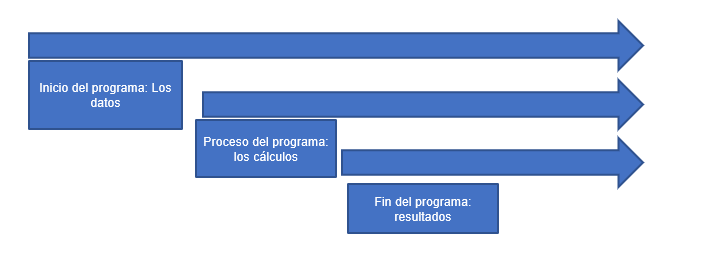
**2.2 Estructura general del lenguaje**

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, creando 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**

A medida de los grandes avances de la tecnología en plena cuarta revolución industrial los programas de computadora que nos ayuda a realizar código fuentes de los diferentes proyectos de programación y se pueden alternar varios lenguajes o editores en un mismo proyecto, ya que en un proyecto de comercio electrónico se puede combinar lenguajes como Python, JavaScript, php, HTML, etc. Los mismos evolucionan constantemente, muchos usuarios crean página web sin tener conocimientos en programación, pero sí requiere paginas webs profesionales, existen editores de código que funcionan perfectamente en 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:

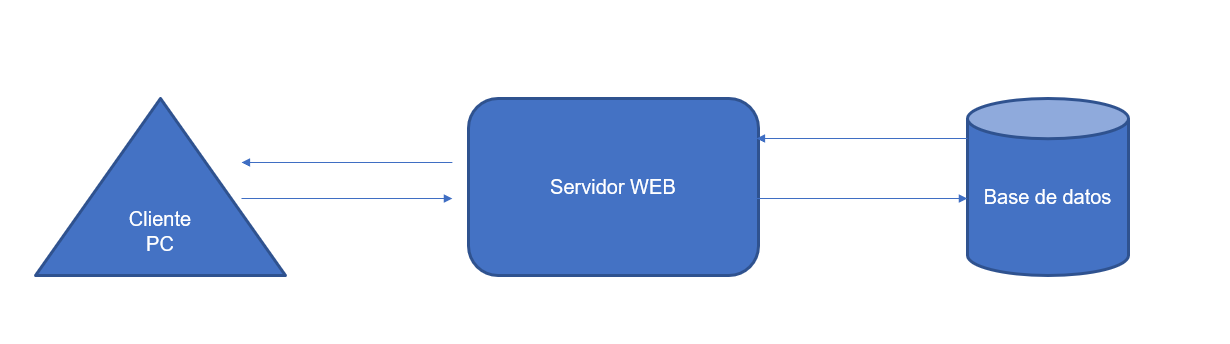
* **Coda:** es una aplicación web que funciona como editor de código para MacOS.
* **Notepad++:** es un editor versátil y soporta lenguajes como HTML, JavaScript y php.
* **Visual Studio code:** es desarrollado por Microsoft se usa con diferentes lenguajes de programación, incluidos C (C, C ++, C #), 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 y 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 a partir del gŕafico 3, donde existe un client que invoca una petición a la aplicación web que está alojado en un servidor web y desde ahí, el backend de la aplicación interactúa con la base de datos respectiva, generando todos los accesos y dan resultados de nuevo al core de la webapp 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

Los componentes de una base de datos

**3. Aplicaciones para programar dispositivos IoT**

Para programar en IoT se puede realizar en Java, lenguaje C, Python, JavaScript son algunos de los lenguajes más importantes que se utilizan en el desarrollo de IoT.

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:



**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 poder desarrollar un proyecto de software, la combinación de varios procesos de programación, cuentan con un editor, compilador, un desbloqueador de las gestiones de código genérico, las mismas se dan en entornos de los lenguajes de desarrollo Python y Java, los sistémicos y programadores pueden testear por medio de estos entornos de desarrollo los diferentes segmentos de código, esto para poder 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 las 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:



Las herramientas de integración aprueban a las organizaciones a conseguir una mayor eficiencia, productividad y agilidad ya que integra diferentes aplicaciones de software y servicios, interfaces de programación de aplicaciones (API) datos y dispositivos para automatizar los procesos empresariales y ofrecer al usuario

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.

**Herramientas:**

| Referencia de imagen:  <https://as2.ftcdn.net/v2/jpg/05/01/56/69/1000_F_501566980_kGPooUSLslZfILvwm0IQ79NfakGvKaAz.jpg> | 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 IoTa 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 dia 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**

Cuando hablamos sobre las pruebas de conectividad las mismas se miden mediante herramientas de diagnóstico en el cual podemos verificar la conectividad entre los puntos de la red. Podemos realizar un análisis de la configuración y poder analizar datos del plano en vivo, las mismas pruebas de conectividad se usan para ayudar a controlar problemas de conexión de red como:

* Opciones incoherentes no deseadas
* Opción 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 (Ver gráfico 4) mediante el valor de shapley ya que es un método de optimización de recursos y podemos aplicarlo a la conectividad o transmisión de paquetes, también mediante el ping podemos tomar los valores de conectividad de red.

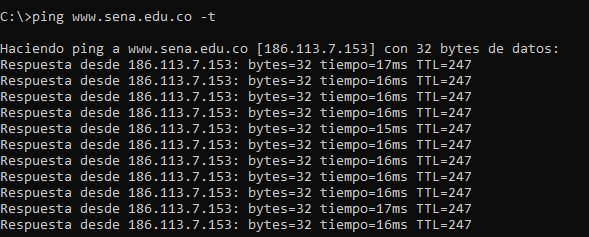


Gráfico 4. Testeo ping con una página web

Pero se toma las direcciones IP de cabecera de red y se procede a llenar la tabla 2.

**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 q 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 del 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 loT, 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.

El enfoque de prueba IoT puede ser diferente según el sistema/arquitectura involucrado, observemos como:



1. **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.



1. **ACTIVIDADES DIDÁCTICAS (Se debe incorporar mínimo 1, máximo 2)**

| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| --- | --- |
| Nombre de la actividad | Autoevaluación de conceptos sobre Configuración, Implementación y pruebas de los dispositivos de IoT de acuerdo con los requerimientos de implementación. |
| Objetivo de la actividad | Identificar los conceptos sobre desarrollo e implementación y pruebas de IoT con el fin de comprender las bases para empezar a programar. |
| Tipo de actividad sugerida | https://lh6.googleusercontent.com/FvABbrwgrnxeHtlHBpt0FdRjJH5VpWBdLbK0Y-NyHF0MxmgecZxbT_QLjV8v2h_0Gr5ETSV7C0ihbVBtf_VBqiBSiEmxWXfjkqpV-5uTXN3HQPFv2D2_nnl_RHrCosO6f6A5mfAWEenp2HD8W1mNoBi4amrW5VkKn7N1ZGfcpGbi1gj0WLQjzi5eb6uE_Sz7u4uqGA |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo\_CF09\_ActividadDidactica\_completar\_espacios |

**F. MATERIAL COMPLEMENTARIO:**

| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del recurso o  archivo del documento o material |
| --- | --- | --- | --- |
| 1. Lenguaje de programación orientado a objetos | Noma Iso 25000 <https://e-collection-icontec-org.bdigital.sena.edu.co/normavw.aspx?ID=6385> | Página Web | <https://e-collection-icontec-org.bdigital.sena.edu.co/normavw.aspx?ID=6385> |
| 1. Lenguaje de programación orientado a objetos | [Edureka](https://www.youtube.com/hashtag/edureka). (2019). *Learn IoT In 4 Hours* [Video]. YouTube. <https://www.youtube.com/watch?v=h0gWfVCSGQQ> | Video | <https://www.youtube.com/watch?v=h0gWfVCSGQQ> |
| 3. Aplicaciones para programar dispositivos IoT | [Edureka](https://www.youtube.com/hashtag/edureka). (2019). *Learn IoT In 4 Hours.* [Video]. YouTube. <https://www.youtube.com/watch?v=h0gWfVCSGQQ> | Video | <https://www.youtube.com/watch?v=h0gWfVCSGQQ> |

**G. GLOSARIO:**

| TÉRMINO | SIGNIFICADO |
| --- | --- |
| 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. |
| Script | Se refiere a fragmentos de código de programación que pueden ejecutar una o varias funciones. |
| Smart cities | Ciudades inteligentes |
| SaaS | Software como servicio. |
| TI | (IT) Abreviatura de Tecnología de la información. |

**H. REFERENCIAS BIBLIOGRÁFICAS**

SG Buzz, (2015) Aprendiendo Ruby y Rails. Parte 1. Introducción a Ruby

<https://sg.com.mx/content/view/564>

Conza, A. (15 de abril de 2018). ESTÁNDARES PARA DISEÑO E IMPLEMENTACIÓN DE BASES DE DATOS RELACIONALES. Obtenido de http://adrianconza.com/:

<http://adrianconza.com/estandares-de-bases-de-datos/>

Cupul, L. D. N., Ake, G. D. J. C., & Rosado, V. M. B. (2021). Web Page Development in Ruby Programming.

<https://www.codecademy.com/courses/learn-rails/lessons/start/exercises/start-hello-rails>

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

<https://www.llardelllibre.cat/es/libro/aprender-la-progamacion-orientada-a-objetos-con-el-lenguaje-c_850051>

Inicio - Introducción a Ruby. (2018.).

<https://makeitrealcamp.gitbook.io/ruby-book/>

ISO/IEC 17789:2014. (2021).

<https://www.iso.org/standard/60545.html>

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

Sotaquirá, M. (15 de junio de 2021). Guía completa para el Manejo de Datos Faltantes. Obtenido de codificandobits:

<https://www.codificandobits.com/blog/manejo-datos-faltantes/>

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

<https://www.taylorfrancis.com/books/mono/10.1201/9781351013239/introduction-python-programming-gowrishankar-veena>.

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

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

<https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=9094>

**I. CONTROL DEL DOCUMENTO**

| Autor (es) | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| --- | --- | --- | --- | --- |
| César Antonio Villamizar Núñez | Experto Temático | Sena - Norte de Santander Centro CIES | Noviembre de 2022 |
| Diego E. Acevedo Guevara | Diseñador Instruccional | Regional Santander - Centro Industrial del Diseño y la Manufactura | Noviembre de 2022 |
| Carolina Coca Salazar | Asesor Metodológico | Regional Distrito Capital- Centro de Diseño y Metrología | Noviembre de 2022 |
| Rafael Neftali Lizcano Reyes | Responsable Equipo de Desarrollo Curricular. | Regional Santander - Centro Industrial del Diseño y la Manufactura | Noviembre de 2022 |
| Sandra Patricia Hoyos Sepúlveda | Corrección de estilo | Regional Distrito Capital - Centro de Diseño y Metrología | Noviembre de 2022 |

**J. CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  | Nombre | Cargo | Dependencia | Fecha | Razón del cambio |
| --- | --- | --- | --- | --- | --- |
| Autor (es) |  |  |  |  |  |