

GITHUB

¿Qué es?

GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc

Fuente: <https://es.wikipedia.org/wiki/GitHub>

¿Para qué sirve?

Su principal función es permitir a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso.

Además, es un servicio que protege tu código y lo mantiene fuera del peligro de las redes y del internet

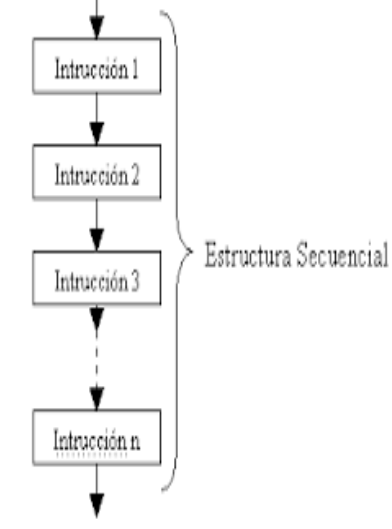
Fuente: <https://www.hostinger.co/tutoriales/que-es-github> ;
<https://www.deustoformacion.com/blog/programacion-tic/que-es-para-que-sirve-github>

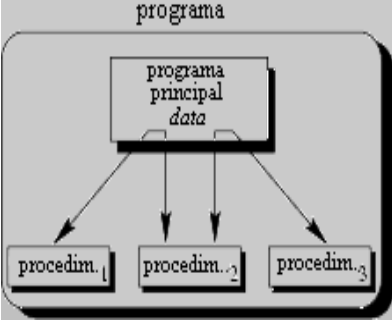
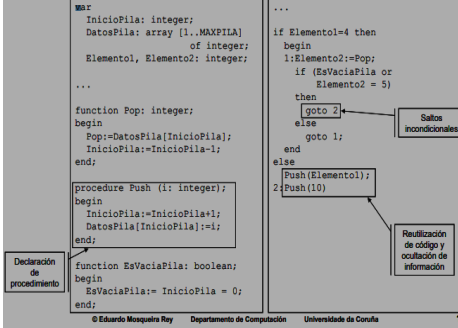
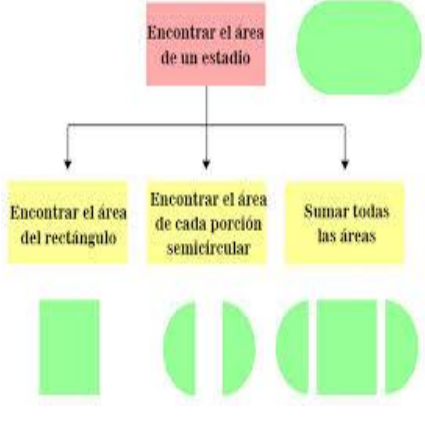
¿Qué es internet de las cosas (IOT)?

El Internet of Things (IoT) describe la red de objetos físicos (cosas) que llevan sensores integrados, software y otras tecnologías con el fin de conectar e intercambiar datos con otros dispositivos y sistemas a través de Internet.

Es uno de los avances mas importantes del siglo xxi ya que gracias a el podemos conectar objetos cotidianos al internet y hacer mas sencillo su uso, poder usarlo a distancia, obtener mejores funciones, entre otras muchas cosas, como lo son por ejemplo, los hornos inteligentes, los autos inteligentes, neveras, audífonos, entre muchas otras cosas.

Fuente: <https://www.oracle.com/mx/internet-of-things/what-is-iot/>

paradigma	descripción	ejemplo	Cómo se usa
1.Programación imperativa	<p>Considerado como el paradigma clásico ya que fue uno de los primeros lenguajes de programación y por extensión también los primeros programas informáticos, se basaban completamente en este enfoque, que prevé una secuencia regularizada de órdenes o instrucciones determinadas. Este paradigma de programación es la base, por ejemplo, de los veteranos lenguajes como pascal y C así como de los lenguajes ensambladores.</p> <p>Fuente:</p>	<p>Fortran. Java. Pascal. ALGOL. C. C# C++ Ensambladores</p>	<p>unprograma se describe en términos de instrucciones, condiciones y pasos que modifican el estado de un programa al permitir la mutación de variables, todo esto con el objetivo de llegar a un resultado.</p> <p>Fuente: https://n9.cl/diuhg</p>
2.Programación estructurada	<p>En pocas palabras es una forma simplificada de la programación imperativa. La principal modificación del principio básico radica en que, en lugar de instrucciones de salto absolutas (instrucciones que provocan que el procesamiento no continúe con la siguiente instrucción, sino en otro lugar) este paradigma de programación de software prevé el uso de bucles y estructuras de control.</p>	<p>.BASIC.</p>  <pre> graph TD Start(()) --> I1[Instrucción 1] I1 --> I2[Instrucción 2] I2 --> I3[Instrucción 3] I3 -.-> In[Instrucción n] In --> End(()) subgraph "Estructura Secuencial" I1 I2 I3 In end </pre>	<p>Este paradigma se fundamenta en el teorema correspondiente, que establece que toda función computable puede ser implementada en un lenguaje de programación que combine solo estas 3 estructuras lógicas.</p> <p>Para las estructuras condicionales o de selección, Python dispone de la sentencia if, que puede combinarse con sentencias elif y/o else.</p> <p>Para los bucles o iteraciones existen las estructuras while y for.</p>

3.Programación procedimental	<p>Es un programa que amplía el enfoque del paradigma imperativo, con la capacidad de desglosar algoritmos en porciones reducidas y manejables denominados como : procedimientos.</p> <p>Su sentido es hacer que el código de su programa sea más fácil y claro y evitar repeticiones innecesarias del código</p>		
4.Programación modular	<p>Su enfoque es similar al de la programación procedimental. En este sentido, el código fuente se divide específicamente en bloques parciales lógicos independientes los unos de los otros para proporcionar más transparencia y facilitar el proceso de debugging (resolución de errores) esto es con el fin de que cada bloque, también llamado módulo se pueda corregir por separado antes de juntarlos</p>		<p>Al aplicar la programación modular, un problema complejo debe ser dividido en varios sub problemas más simples, y estos a su vez en otros sub problemas más simples aún. Esto debe hacerse hasta obtener sub problemas lo suficientemente simples como para poder ser resueltos fácilmente con algún lenguaje de programación.</p>
5. Programación declarativa	<p>El principio fundamental de la programación declarativa radica en la descripción del resultado final que se busca. Por lo tanto, en primera línea se encuentra el “qué” del resultado y no el “cómo” de los pasos que llevan a la solución, como es el caso en la programación imperativa. Esto provoca que el código de la programación declarativa sea más difícil de comprender debido al alto grado de abstracción, aunque resulta muy corto y preciso.</p>	<p>Programación imperativa:</p> <pre>1 \$listaparticipantes = [1 => 'Peter', 2 => 'Hans', 3 => 'Sarah']; 2 \$nombres = []; 3 foreach (\$listaparticipantes as \$id => \$apellido) { 4 \$nombres[] = \$apellido; 5 }</pre> <p>Programación declarativa:</p> <pre>1 \$nombres = array_values(\$listaparticipantes);</pre>	<p>Como la programación declarativa no determina el “cómo”, sino que funciona a un nivel de abstracción muy alto, este paradigma deja margen para la optimización. Si se ha desarrollado un procedimiento de ejecución mejor, el algoritmo integrado lo encuentra y lo aplica. En este sentido, el paradigma está muy preparado para el futuro porque, al escribir el código, no es necesario determinar el procedimiento según el cual se alcanza el resultado.</p>

6. programación lógica	<p>es un programa basado en la lógica matemática, lugar de una sucesión de instrucciones, un software programado según este principio contiene un conjunto de principios que se pueden entender como una recopilación de hechos y suposiciones. Todas las solicitudes al programa se procesan de forma que el intérprete recurre a estos principios y les aplica reglas definidas previamente para alcanzar el resultado deseado.</p>	<h3>Programación Lógica</h3> <h4>Ejemplo de programa lógico</h4> <div> <div> Programa lógico </div> <div> <div> <div> animal(perro). animal(gato). animal(canguro). arbol(palmera). flor(margarita). </div> <div>Hechos</div> </div> <div> <div> vegetal(X) :- arbol(X). vegetal(X) :- flor(X). </div> <div>Reglas</div> </div> </div> </div> <div> <div> Invocación </div> <div> <div> > vegetal(perro) > false > animal(perro) > true > vegetal(palmera) > true </div> </div> </div>	<p>consiste en indicar como resolver un problema mediante sentencias, se trabaja en una forma descriptiva, estableciendo relaciones entre entidades, indicando no como, sino que hacer, Dado un problema S, saber si la afirmación A es solución o no del problema o en que casos lo es. Además queremos que los métodos sean implantados en maquinas de forma que la resolución del problema se haga de forma automática. La programación lógica construye base de conocimientos mediante reglas y hechos.</p>
------------------------	---	--	---

Fuentes: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/paradigmas-de-programacion/>

https://ferestrepoca.github.io/paradigmas-de-programacion/proglogica/logica_teoría/lang.html#:~:text=La%20programaci%C3%B3n%20l%C3%B3gica%20estudia%20el,para%20alcanzar%20la%20soluci%C3%B3n%20autom%C3%A1tica.&text=Se%20puede%20ver%20como%20una,y%20reglas%20para%20representar%20conocimiento.

http://quegrande.org/apuntes/EI/1/EDI/teoria/07-08/tad_introduccion.pdf