



El futuro digital  
es de todos

MinTIC



# De algoritmos a Programas

---



Universidad de Caldas

# ¡Hola a todos! << ● ● ● ●

En esta revista vamos a ver el proceso que sucede para convertir los algoritmos escritos en lenguaje natural, es decir, en lenguaje que nosotros entendemos a programas que son entendidos por las máquinas, es decir, por los computadores

## Bienvenidos

El título de esta diapositiva es **Lo siento pero no te entiendo** y recordemos cuál es el mecanismo o entendimiento que tenemos entre nosotros y un computador, un programa computacional es una lista de instrucciones, escritas en un lenguaje de programación que el computador entiende, sin embargo, Aquí subrayó este computador entiende porque ahí hay un poquito digamos que no es del todo cierto; como decía René Descartes la duda es el origen de la sabiduría, René Descartes dudaría que el computador entiende Lo que nosotros le escribimos en un lenguaje de programación y resulta que este señor tenía razón, como así profe usted nos lleva diciendo una semana un programa entiende a un computador es decir que si yo escribo aquí imprime “Hola mundo” en un lenguaje de programación como python donde diría así y se la envío a la CPU que realmente es lo que es un computador, un procesador pequeño que está allí en el interior de sus dispositivos computacionales resulta que el va a decir, **lo siento pero no te entiendo** y entonces qué estamos diciendo aquí, pues que lo que realmente entiende un procesador es código de máquina es código que como lo hemos visto en muchas películas sólo tiene ceros y unos eso es lo que entiende el computador.

Entonces el procesador que tienen todos estos dispositivos los portátiles, los computadores de escritorio los dispositivos móviles los grandes servidores de la nube sólo entiende una cosa **código de Máquina** eso ¿que nos plantea? nos plantea que si yo como programador escribo el código que recuerden que lo que van a escribir ustedes se llama código Fuente y el procesador es el que me lo debe entender y el solo entiende código de máquina que en este caso está en hexadecimal se puede convertir a ceros y unos también, como es la electricidad interna o cuando hablamos de esos computadores cuánticos que están de moda pues son ceros y unos.



Si una persona que habla alemán le quiere decir algo a usted, pues qué tiene que hacer uno si no sabe nada de alemán y quiere hablar con un alemán, necesita alguien que le traduzca y eso mismo va a pasar aquí, necesitamos un proceso que nos convierta lo que le estamos escribiendo a lo que entiende el procesador que está en nuestros dispositivos computacionales que es código de máquina para que todos nos entendamos.

Resulta que actualmente Existen dos formas de hacer eso una se llama **compilador** y otro se llama **intérprete** en el fondo sirven para Lo mismo para que los programas que yo escribo en un lenguaje de programación sean entendidos por el procesador que tienen nuestras máquinas Gracias a que el compilador e intérprete los convierte en código de máquina

Y entonces vamos a ver la primera opción que es el compilador **Cómo funciona el compilador**, tenemos el computador del programador, el computador del desarrollador de software y al otro lado tenemos el computador del usuario que es el que va a utilizar el programa que nosotros hacemos, recuerden nosotros escribimos y mal escrito esto es como para que vean que el computador solamente va a entender el print, paréntesis pero no le importa que tengas mala ortografía que no sepamos inglés, está mal escrito ahí hello world y esta opción del compilador lo que hace es que inmediatamente yo lo pongo a funcionar convierte el código fuente a código de máquina ,como el código de máquina ya es entendido por él dispuesto computacional pues lo que tengo que hacer es instalarlo allí, el detalle aquí es interesante que cuando yo compilo un programa si se llama Hola.pi Qué es la forma de darle el nombres a los programas en python el lo va a convertir en un en un programa compilado que se llamará hola.exe quiere decir que es ejecutable quiere decir que como ya está en código de máquina yo lo copió en todos los computadores de los usuarios que quieren ver mi programa que Saluda ya les va a funcionar eso es lo que hace un compilador.

Print ("helo world")

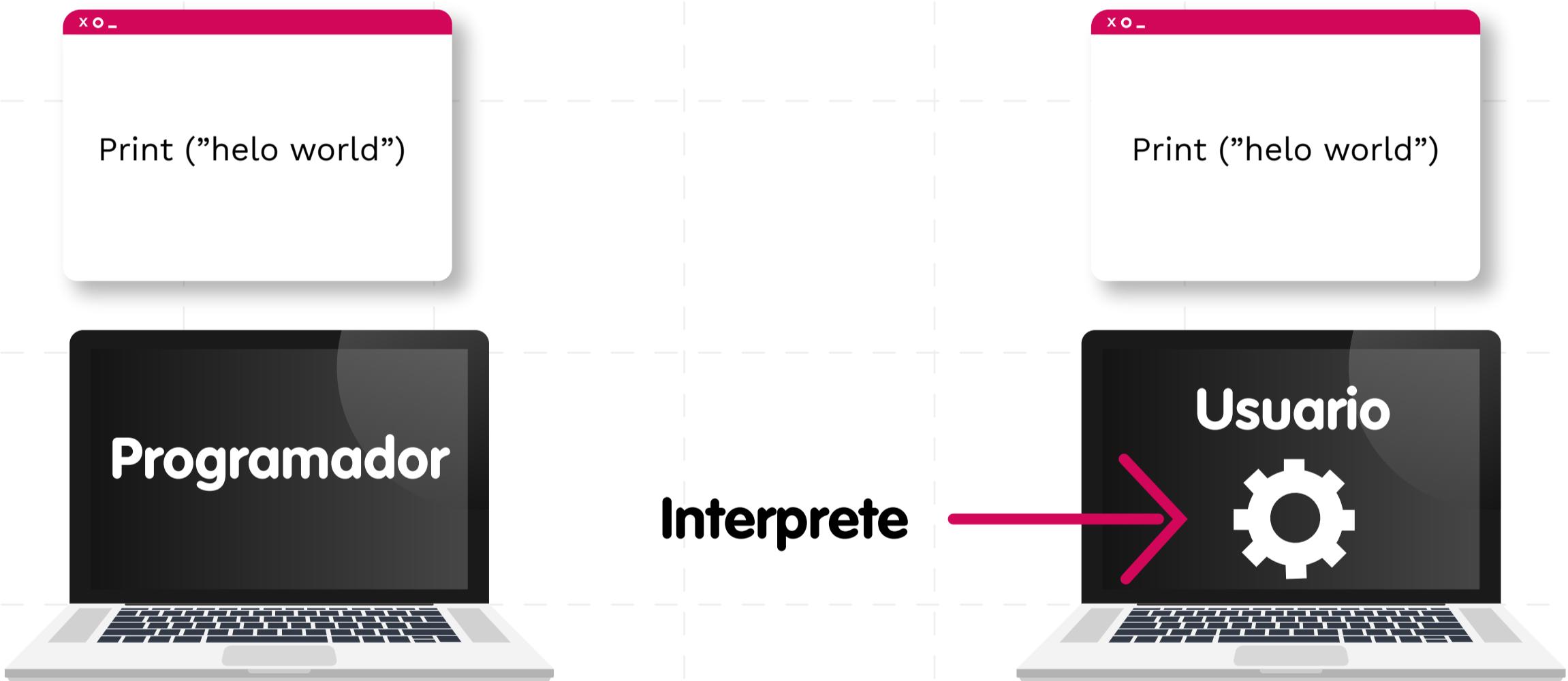
.Exe

**Compilador**

**Programador**

**Usuario**

La otra opción es el intérprete, tengo el mismo escenario tengo el código fuente tengo el computador del usuario y quiero que en el computador del usuario funcione mi programa lo que debo hacer es copiar ni código Fuente al del computador del usuario Sí señor así como lo escucha el código Fuente al del computador del usuario dentro del computador del usuario tiene que estar instalado un intérprete Ahí estás el señor que va a traducir y el intérprete empieza a funcionar y lo convierte a código de máquina no nos debemos preocupar porque tengo que instalar en el intérprete realmente la mayoría de sistemas operativos ya tienen los intérpretes allí instalados y además de no ser así pueden interpretarse con programas que funcionan en la nube que es lo que nosotros vamos a hacer en el curso



Entonces tenemos dos alternativas los lenguajes compilados y los lenguajes interpretados, ya vimos que los lenguajes compilados compila dentro del computador del programador, genera el código ejecutable y yo copio los códigos; Hice un jueguito, lo compilo se llama juego.pi yo lo compilo se llama juego.exe y lo copió el todos mis amigos y ellos juega, en cambio en el interpretado yo hago el juego .pi y para que los otros les funcione les tengo que pasar el juego .pi al computador de mis amigos Pero ellos tienen que tener el intérprete de python para poder jugar o en su defecto tienen que ir a la web y Buscar un intérprete de phyton.

**ventajas y contras** de ambos enfoques obviamente los lenguajes compilados como ya no tiene que ser el proceso de traducción que se hace en el computador del cliente son comúnmente más rápido y lo contrario de los interpretados que necesitan hacer este proceso son más lentos, específicos para la plataforma claro, cada tipo de dispositivo, cada tipo de computador que tenga Android por ejemplo que tenga Windows que tenga el de iOS tiene un sistema diferente y el código de máquinas diferentes entonces se debe complicar para cada máquina mientras que el lenguaje interpretado no yo hago mi código fuente y ya que cada computador tenga su intérprete para el lenguaje; ahora el código fuente No se comparte que para algunos puede ser bueno o mal, en cambio el código fuente en el interpretador se comparten y deben entender porqué el código fuente No se comparten en compilador y si me han entendido hasta ahora el código fuente si se comparte en el interpretado eso puede ser bueno o malo dependiendo de lo que yo quiera hacer compartir el código, hay muchos tendencia código abierto y generar creatividad otros dicen no yo como voy a dar el codigo de mi programa si me trasnoche haciéndolo.

El lenguaje compilado está listo para ejecutarse y el lenguaje interpretado se requiere acceso al interprete bueno malo, nosotros vamos a utilizar un lenguaje interpretado que es phyton, no olviden phyton es interpretado.

Hemos llegado al final de la revista, antes de irnos los dejo con esta lista de lenguajes comunes Compilados, interpretados e híbridos.

## Compilados

C  
C++  
Swift

## Interpretados

JavaScript  
PHP  
Python

## Híbridos

C#  
Java  
ActionScript



Universidad de Caldas