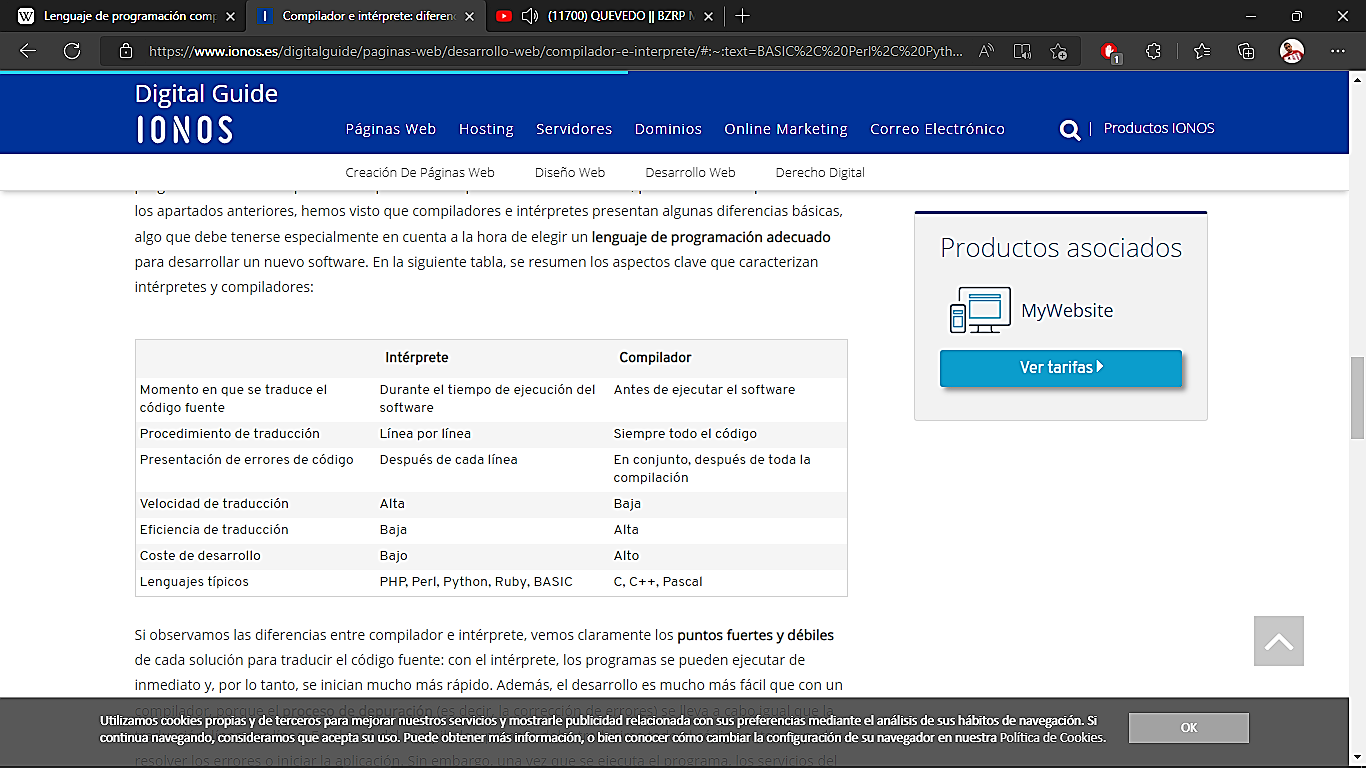
**LENGUAJES INTERPRETES Y COMPILADOS**

**¿Qué es un compilador?**

Un compilador es un programa informático que traduce todo el código fuente de un proyecto de software a código máquina antes de ejecutarlo. Solo entonces el procesador ejecuta el software, obteniendo todas las instrucciones en código máquina antes de comenzar. De esta manera, el procesador cuenta con todos los componentes necesarios para ejecutar el software, procesar las entradas y generar los resultados. No obstante, en muchos casos, durante el proceso de compilación tiene lugar un paso intermedio fundamental: antes de generar la traducción final en código máquina, la mayoría de los compiladores suelen convertir el código fuente en un código intermedio (también llamado código objeto) que, a menudo, es compatible con diversas plataformas y que, además, también puede ser utilizado por un intérprete.

**¿Qué es un intérprete?**

Un intérprete es un programa informático que procesa el código fuente de un proyecto de software durante su tiempo de ejecución, es decir, mientras el software se está ejecutando, y actúa como una interfaz entre ese proyecto y el procesador. Un intérprete siempre procesa el código línea por línea, de modo que lee, analiza y prepara cada secuencia de forma consecutiva para el procesador. Este principio también se aplica a las secuencias recurrentes, que se ejecutan de nuevo cada vez que vuelven a aparecer en el código. Para procesar el código fuente del software, el intérprete recurre a sus propias bibliotecas internas: en cuanto una línea de código fuente se ha traducido a los correspondientes comandos legibles por máquina, esta se envía directamente al procesador

****

Lista de lenguajes compilados

Algunos lenguajes que comúnmente se consideran compilados:

Ada

ALGOL

Algol 60

Algol 68

SMALL

BASIC

C

C++

Objective-C

C# (a bytecode)

D

CLEO

COBOL

Cobra

Common Lisp

Delphi

Eiffel

Sather

Ubercode

Factor (las últimas versiones)

Forth

Fortran

Go

Haskell

Haxe (to bytecode)

IBM RPG

Java (a bytecode)

JOVIAL

G

Lisp

Lush

Mercury

ML

Standard ML

Alice

OCaml

Modula-2

Modula-3

Open-URQ

PureBasic

Pascal

PL/I

Rust

Scala

Seed7

Swift

Visual Basic

Visual Foxpro

Visual Prolog

Lista de lenguajes interpretados

* [APL](https://es.wikipedia.org/wiki/APL) Un lenguaje orientado a vectores que usa un conjunto de caracteres inusual
  + [J](https://es.wikipedia.org/wiki/J) Una variante de en el que la definición tácita ofrece algunos de los beneficios de la compilación
* [BASIC](https://es.wikipedia.org/wiki/BASIC) (aunque la versión original, era compilada, así como muchos modernos)
  + [thinBasic](https://es.wikipedia.org/w/index.php?title=ThinBasic&action=edit&redlink=1)
* [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript)
  + [ActionScript](https://es.wikipedia.org/wiki/ActionScript) (versión 3.0 no es interpretado, por lo que se eliminó la función eval())
  + [E4X](https://es.wikipedia.org/w/index.php?title=E4X&action=edit&redlink=1) (ECMAScript for XML)
  + [JavaScript](https://es.wikipedia.org/wiki/JavaScript) (primero se llamó **Mocha**, y luego **LiveScript**)
  + [JScript](https://es.wikipedia.org/wiki/JScript)
* Manipulación de ecuaciones y sistemas de resolución
  + [GNU Octave](https://es.wikipedia.org/wiki/GNU_Octave)
  + [IDL](https://es.wikipedia.org/wiki/IDL) (Interactive Data Language)
  + [TK Solver](https://es.wikipedia.org/w/index.php?title=TK_Solver&action=edit&redlink=1)
  + [Mathematica](https://es.wikipedia.org/wiki/Mathematica)
  + [MATLAB](https://es.wikipedia.org/wiki/MATLAB)
* [EUPHORIA](https://es.wikipedia.org/wiki/EUPHORIA_(lenguaje_de_programaci%C3%B3n)) interpretado o compilado
* [Forth](https://es.wikipedia.org/wiki/Forth)
* [Game Maker Language](https://es.wikipedia.org/wiki/Game_Maker_Language)
* [Java (lenguaje de programación)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
* Madness Script
* [Perl](https://es.wikipedia.org/wiki/Perl)
* [PHP](https://es.wikipedia.org/wiki/PHP)
* [PostScript](https://es.wikipedia.org/wiki/PostScript)
* [Python](https://es.wikipedia.org/wiki/Python)
* [Lisp](https://es.wikipedia.org/wiki/Lisp)
* [Logo](https://es.wikipedia.org/wiki/Logo_(lenguaje_de_programaci%C3%B3n))
* [Scheme](https://es.wikipedia.org/wiki/Scheme)
* [MUMPS](https://es.wikipedia.org/w/index.php?title=MUMPS&action=edit&redlink=1) (tradicionalmente interpretado, compilado en las versiones modernas)
* [REXX](https://es.wikipedia.org/wiki/REXX)
* [Ruby](https://es.wikipedia.org/wiki/Ruby)
  + [JRuby](https://es.wikipedia.org/wiki/JRuby) (una implementación Java de Ruby)
* [Seed7](https://es.wikipedia.org/w/index.php?title=Seed7&action=edit&redlink=1) interpretado o compilado
* [Smalltalk](https://es.wikipedia.org/wiki/Smalltalk)
  + [Bistro](https://es.wikipedia.org/wiki/Bistro)
  + [Dolphin Smalltalk](https://es.wikipedia.org/w/index.php?title=Dolphin_Smalltalk&action=edit&redlink=1)
  + [F-Script](https://es.wikipedia.org/w/index.php?title=F-Script&action=edit&redlink=1)
  + [Little Smalltalk](https://es.wikipedia.org/w/index.php?title=Little_Smalltalk&action=edit&redlink=1)
  + [Squeak](https://es.wikipedia.org/wiki/Squeak)
  + [VisualAge](https://es.wikipedia.org/w/index.php?title=VisualAge&action=edit&redlink=1)
  + [VisualWorks](https://es.wikipedia.org/wiki/VisualWorks)
* Lenguajes de *scripting*
  + [WebDNA](https://es.wikipedia.org/w/index.php?title=WebDNA&action=edit&redlink=1)
* [Hojas de cálculo](https://es.wikipedia.org/wiki/Hojas_de_c%C3%A1lculo)
  + [Excel](https://es.wikipedia.org/wiki/Excel) almacena fórmulas, y las interpreta desde un formato simbólico
* [S](https://es.wikipedia.org/w/index.php?title=S_(lenguaje_de_programaci%C3%B3n)&action=edit&redlink=1)
  + [R](https://es.wikipedia.org/wiki/R_(lenguaje_de_programaci%C3%B3n))
* [Tcl](https://es.wikipedia.org/wiki/Tcl)
  + [XOTcl](https://es.wikipedia.org/w/index.php?title=XOTcl&action=edit&redlink=1)
* [VBScript](https://es.wikipedia.org/wiki/VBScript)
* [PowerShell](https://es.wikipedia.org/wiki/Windows_PowerShell)
* [XMLmosaic](http://xmlmosaic.codeplex.com/) Un lenguaje de programación interpretado parecido a C# escrito en Visual Basic .NET

Solución híbrida de intérprete y compilador: compilación en tiempo de ejecución

Para compensar los puntos débiles de ambas soluciones, también existe el llamado modelo de compilación en tiempo de ejecución (en inglés, ***just-in-time-compiler***, o “compilador justo a tiempo”). Este tipo de compilador, que a veces también se conoce por el término inglés ***compreter*** (acrónimo de ***com****piler* e *inter****preter***), rompe con el modelo habitual de compilación y traduce el código del programa durante el tiempo de ejecución, al igual que el intérprete. De esta forma, la **alta velocidad de ejecución** típica de los compiladores se complementa con la **simplificación del proceso de desarrollo**.

**Java** es uno de los ejemplos más conocidos de lenguaje basado en compilación en tiempo de ejecución: el compilador JIT, que figura entre los componentes del **Java Runtime Environment (JRE)**, mejora el **rendimiento de las aplicaciones Java** traduciendo el código de bytes en código máquina de manera dinámica.