



LENGUAJE	 Common Lisp	 Clojure	 Scala	 Rust	 Go
ABSTRACCIÓN	Alto nivel (se puede usar a bajo nivel)	Alto nivel	Alto nivel	Alto nivel (se puede usar a bajo nivel)	Alto nivel
EJECUCIÓN	Compilado	Compilado (puede usarse con la máquina virtual de Java, máquina virtual de .NET así como compilarse a Javascript)	Compilado (la versión actual se ejecuta en la máquina virtual de Java y es compatible con las aplicaciones Java existentes)	Compilado	Compilado
PARADIGMA	Funcional	Funcional	Funcional y orientado a objetos	Multiparadigma	Imperativo y concurrente, admite programación orientada a objetos pero no dispone de herencia de tipos
TIPADO	Dinámico	Dinámico, fuerte	Estático, fuerte	Estático, fuerte, inferenciado	Fuerte, estático
AUTOR	Comité ANSI X3J13	Richard Hickey	Martin Odersky	Graydon Hoare	Robert Griesemer, Rob Pike y Ken Thompson
AÑO	1984	2007	2003	2010	2009
EXTENSIONES	.lisp, .lsp, .l, .cl, .fasl	clj, cljs, cljc, edn y cljr	.scala, .sc	.rs, .rlib	.go
DOCUMENTACIÓN	https://lisp-lang.org/learn/	https://clojure.org/guides/getting-started	https://docs.scala-lang.org/	https://www.rust-lang.org/learn	https://go.dev/doc/
HELLO WORLD	<pre>(defun hello-world () (format t "Hello World ~%")) (hello-world)</pre>	<pre>(defn -main "I don't do a whole lot ... yet" [& args] (println "Hello world!"))</pre>	<pre>package com.test object HelloWorld extends App { println("Hello World!!") }</pre>	<pre>fn main(){ println("Hello World!!"); }</pre>	<pre>package main import ("fmt") func main(){ fmt.Println("Hello World!\n") }</pre>