

Instituto Tecnológico de Colima
Programación lógica y funcional

Lenguajes de programación funcionales							
Lenguaje de programación	Fecha de creación	Quién y dónde se diseño o creó	Principales características	Para que se utiliza	Ventajas	Desventajas	Como adquirirlo
<div></div> <div>Haskell</div>	Durante en el año 1990 .	Creado en un comité liderado por Simon Peyton Jones , investigador de Microsoft26, y Paul Hudak , Lennart Augustsson , John Hughes , Erik Meijer y Philip Wadler en la Universidad de Yale y la Universidad de Glasgow	<ul style="list-style-type: none">• PURO: En Haskell toda operación computacional se contempla como operaciones matemáticas.• DECLARATIVO: Como su elemento central son las funciones, este se centra en determinar.• POLIMÓRFICO: Un valor es polimorfo si puede tener más de un tipo.• ESTÁTICAMENTE TIPADO: Se pueden encontrar errores antes de que se ejecute el programa ya que los tipos se verifican en tiempo de compilación.• PEREZOSO: Una expresión no es evaluada cuando se le asocian valore a sus variables.• PROGRAMAS CONCISOS: Tiene pocas palabras clave. Usa indentación como medio de estructuración del código• SISTEMA DE TIPOS: Tiene un sistema de tipo que requiere un poco de información del programador con los cuales puede detectar una gran variedad de errores de incompatibilidad.• EFFECTOS MONÁDICOS: Para una función dada una misma entrada, siempre se producirá la misma salida, independientemente del contexto de las variables.• CONCURRENTE: Permite que, durante un periodo de tiempo, más de un proceso se ejecute.	Haskell es a menudo la elección clara en los campos que requieren un mapeo muy preciso de los algoritmos matemáticos, entre los que se incluyen las aplicaciones para la seguridad de red, los marcos de especificaciones para sistemas integrados o los programas para modelar cálculos matemáticos complejos.	<ul style="list-style-type: none">• La productividad de los desarrolladores puede aumentar considerablemente.• El código del software de Haskell es breve, claro y fácil de mantener.• Las aplicaciones de Haskell son menos propensas a errores y ofrecen una gran fiabilidad.• La brecha "semántica" entre el programador y el lenguaje es mínima.	<ul style="list-style-type: none">• Es muy difícil de aprender sin una formación adecuada en ciencias de la computación.• Las opciones del IDE no son tan buenas como las de otros lenguajes de programación.• Haskell no se puede utilizar para el desarrollo de Android o IOS.• Algunas bibliotecas carecen de personas de apoyo que las mantengan y las mejoren.• El uso ingenuo de la evaluación perezosa en situaciones no triviales puede generar cuellos de botella innecesarios en el desempeño, y comprender lo que sucede debajo del capó no es exactamente sencillo.• No puede razonar fácilmente sobre el orden de ejecución.	Siguiendo los pasos de instalación en la página oficial https://www.haskell.org/downloads/
<div></div> <div>Scala</div>	Se desarrolló en el año 2001 y debutó en 2004 .	Por un equipo de desarrolladores dirigidos por Martin Odersky en la universidad EPFL de Suiza . fue creado como evolución del lenguaje java, por ende, la similitud entre ambos lenguajes de programación.	<ul style="list-style-type: none">• Sintaxis ligera para definir funciones anónimas.• Soporta funciones de primer orden.• Las clases caso de Scala y las construcciones incorporadas al lenguaje para reconocimiento de patrones.• Scala estáticamente tipado.• Clases genéricas.• Anotaciones variables.• Límites de tipado superiores e inferiores.• Clases internas y tipos abstractos como miembros de objetos• Tipos compuestos• Auto-referencias explícitamente tipadas• Vistas• Métodos polimórficos	<ul style="list-style-type: none">• Desarrolladores, científicos de datos y empresas no sólo utilizan Scala para crear sus nuevos proyectos, sino que también usan otras herramientas como Play Framework, Apache Spark , Akka o Kafka.• Scala es ejecutado sobre la JVM, permitiendo una unión con Java beneficiosa para ambos lenguajes, y para el desarrollador, ya que éste (o los equipos) pueden programar en Java y Scala al mismo tiempo.• Permite la integración de sus características y librerías, pudiendo, entre muchísimas cosas más, heredar clases o implementar interfaces.	<ul style="list-style-type: none">• Cantidad de códigos que se necesita para programar.• Gran diversidad de herramientas, mantenimiento, formación y consultas para empresas.• Confianza y eficiencia.• Combina con programación orientada a objetos.• Sencillo de aprender.	<ul style="list-style-type: none">• Tiene problemas de rendimiento debido a su enfoque de inmutabilidad y programación funcional.• Scala ha descuidado versiones anteriores, lo que ha generados muchos fallos en la compatibilidad.• Este lenguaje de programación puede ser un poco difícil de dominar en ciertos casos.	Se puede descargar en https://www.scala-lang.org/download/
<div></div> <div>Lisp</div>	Se desarrolló en el año 1958 .	Por John McCarthy mientras estaba en el Instituto Tecnológico de Massachusetts (MIT).	<ul style="list-style-type: none">• No posee asignaciones.• Su principal estructura de control es la recursión.• Los programas y los datos son equivalentes.• Su principal estructura de datos es la lista.• La memoria es asignada por demanda.	Es un lenguaje de programación que fue diseñado para una fácil manipulación de cadenas de datos. LISP se convirtió en un lenguaje común para la programación de inteligencia artificial (IA), en parte debido a la confluencia del trabajo de LISP e IA en el MIT y en parte porque los programas de IA capaces de "aprender" podían escribirse en LISP como programas automodificables.	<ul style="list-style-type: none">• Es flexible.• Maneja listas y arboles con cierta facilidad.• Tiene características funcionales.• Se puede razonar matemáticamente.• Compatibilidad con todos los sistemas operativos.• Es multiparadigma.• Es corto y sencillo.	<ul style="list-style-type: none">• Carece de bibliotecas extensas.• Puede tener una sintaxis complicada.	No existe una pagina oficial que nos permita descargarlo, pero existen muchas fuentes gratuitas para obtenerlo, como https://lisp-ed.programas-gratis.net/ . Otra forma de utilizarlo es descargando la librería "Common Lisp" en el editor de código Visual Studio Code, siendo uno de los más famosos en la actualidad.
<div></div> <div>F#</div>	Fue desarrollado en 2005 .	Por investigadores de Microsoft (E.U.A) Research, dirigidos por Don Syme . El lenguaje, esencialmente, era una implementación .Net de OCaml ya que combinaba la sintaxis y el poder del lenguaje funcional con los miles de funciones de biblioteca disponibles con todos los lenguajes .NET.	<ul style="list-style-type: none">• Sintaxis ligera.• Inmutable por defecto.• Inferencia de tipos y generalización automática.• Funciones de primera clase.• Tipos de datos potentes.• Programación asíncrona.• Buena interoperabilidad.	Se puede utilizar en inteligencia artificial, aprendizaje mecánico y análisis de datos relacionados. Se ha convertido en un lenguaje listo para producción que muchas empresas utilizan en la industria financiera en particular. Prácticamente se puede utilizar para prácticamente cualquier cosa, pero se destaca en lo que llamamos "desarrollo de componentes analíticos".	<ul style="list-style-type: none">• Tipado.• Usa inferencia de tipos.• Los tipos no necesitan ser declarados.• Declaración explícita de tipos de datos.• soporta los objetos y tipos de .NET.• Permite programar de una manera que se asemeja más a como pensamos.• Creación de código asíncrono y paralelo.	<ul style="list-style-type: none">• No tiene una función de sobrecarga para el nombrado de las variables.• Estructuras de datos más complejas.• Herramientas menos avanzadas, (no tiene métodos básicos para la refactorización), (Se invierte más en C# que en F#).	En la página oficial de Microsoft se recomienda instalarlo mediante el editor Visual Studio Code. <ul style="list-style-type: none">• Primero hay que tener instalado git, de no tenerlo se puede descargar de https://git-scm.com/download .• Como segundo paso se debe tener instalado .NET SDK, se puede encontrar en su página oficial https://dotnet.microsoft.com/en-us/download, de igual forma asegurarse de tener Visual Studio Code, sino también se puede encontrar en su página oficial https://code.visualstudio.com/ .• Para finalizar, en el editor Visual Studio Code, se debe descargar la extensión "Ionide-fsharp".
<div></div> <div>Python</div>	Fue desarrollado en el año de 1991 .	Por Guido van Rossum en el centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informatica), en los Países Bajos , como un sucesor del lenguaje de programación ABC, capaz de manejar excepciones e interactuar con el sistema operativo Amoeba.	<ul style="list-style-type: none">• Es un lenguaje interpretado, no compilado, usa tipado dinámico, fuertemente tipado.• Es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.• Es un lenguaje de programación multiparadigma, el cual soporta varios paradigmas de programación como orientación a objetos, estructurada, programación imperativa y, en menor medida, programación funcional.• En Python, el formato del código (p. ej., la indentación) es estructural.• Multiparadigma.• Lenguaje de alto nivel.	<ul style="list-style-type: none">• Data analytics y big data.• Data mining.• Data science.• Inteligencia artificial.• Blockchain.• Machine learning.	<ul style="list-style-type: none">• Python es fácil y sencillo de aprender.• Python hace referencia a su limpieza y legibilidad.• Está desarrollado bajo una licencia de código abierto, por lo que es de libre uso y distribución, incluso para uso comercial.• Este lenguaje de programación permite una diversidad de desarrollos de una manera fácil, ágil y rápida.• Su característica multiplataforma permite que este pueda ser usado en diferentes sistemas operativos.• Cuenta con una amplia comunidad que organiza eventos, conferencias, reuniones y colabora en materia de códigos e información.	<ul style="list-style-type: none">• Python posee una baja velocidad de procesamiento, por lo que puede volverse difícil de manejar para aplicaciones grandes y complejas.• Sus funciones dinámicas pueden causar errores en tiempo de ejecución.• La mayoría de los servidores con servicio hosting no tienen soporte para Python.• Posee un consumo ineficiente de memoria para realizar algunas tareas.• Cuenta con una programación débil para dispositivos móviles.	Para obtener Python debes descargar el instalador en la página oficial e instalarlo en tu computadora https://www.python.org/

Nombre de los alumnos
Juan Pablo Anguiano Olmos
Luis Francisco Barragan Flores
Luis Enrique Castillo Martínez

No. Control
18460609
18460194
18460222

Conclusión

La programación funcional sus raíces en el cálculo lambda, un sistema formal desarrollado en los años 1930, este es un paradigma de programación que se base en funciones, recursividad, datos inmutables, funciones lambda, entre otras propiedades. Con la finalidad de resolver problemas matemáticos códigos cortos, legibles y eficientes, además de que crea sistemas altamente seguros. Existen diversos lenguajes de programación funcional, especialmente los puramente funcionales, los cuales priorizan el uso de recursividad y aplicación de funciones de orden superior para resolver problemas que en otros lenguajes se resolverían mediante estructuras de control, entre estos destacan lenguajes de programación funcional como Lisp, Scala, F# y Haskell, que han sido utilizados en aplicaciones comerciales e industriales por muchas organizaciones.

Nombre de los alumnos
Juan Pablo Anguiano Olmos
Luis Francisco Barragan Flores
Luis Enrique Castillo Martínez

No. Control
18460609
18460194
18460222

Referencias Electrónicas (Bibliografía)

- [1] Digital Guide IONOS, «Haskell: el lenguaje de programación funcional,» Digital Guide IONOS, 09 10 2020. [En línea]. Available: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/que-es-haskell/>. [Último acceso: 14 03 2022].
- [2] J. Inca, «Estudio del lenguaje de programación Haskell, ventajas y desventajas con respecto a otros lenguajes de programación,» 1library.co, 12 09 2012. [En línea]. Available: <https://1library.co/document/9yngldiz-estudio-lenguaje-programacion-haskell-ventajas-desventajas-lenguajes-programacion.html>. [Último acceso: 12 03 2022].
- [3] Haskell, «Kaskell un lenguaje de programación puramente funcional,» Kaskell, [En línea]. Available: <http://ferestrepoca.github.io/paradigmas-de-programacion/progfun/tutoriales/Haskell/Haskell.html>. [Último acceso: 12 03 2022].
- [4] J. A., «PFH: Introducción a la programación funcional con Haskell (Parte 1 de 2),» glc.us.es, 09 08 2020. [En línea]. Available: <https://www.glc.us.es/~jalonso/vestigium/pfh-introduccion-a-la-programacion-funcional-con-haskell-parte-1-de-2/>. [Último acceso: 12 03 2022].
- [5] Haskell, «Haskell Downloads,» Haskell.org, [En línea]. Available: <https://www.haskell.org/downloads/>. [Último acceso: 12 03 2022].
- [6] Lógica para Ciencias de la Computación. (2009). *Programación Funcional en LISP*. PDF. <http://www.cs.uns.edu.ar/~grs/Logica/011.Programacion%20Funcional%20en%20LISP.Color.pdf> [Último acceso: 10 03 2022].
- [7] IT, R. (2020, 13 de octubre). ¿Qué es LISP?. Reclu IT. <https://recluit.com/que-es-lisp/#.Yin7VHrMJPY> [Último acceso: 10 03 2022].
- [8] Hemmendinger, D. (s. f.). *LISP | computer language*. Encyclopedia Britannica. <https://www.britannica.com/technology/LISP-computer-language> [Último acceso: 10 03 2022].
- [9] Colaboradores de los proyectos Wikimedia. (2005, 23 de enero). *Common Lisp - Wikipedia, la enciclopedia libre*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Common_Lisp [Último acceso: 10 03 2022].
- [10] Seibel, P. (2003). *Introduction: Why Lisp?* gigamonkeys. <https://gigamonkeys.com/book/introduction-why-lisp.html> [Último acceso: 10 03 2022].

Nombre de los alumnos
Juan Pablo Anguiano Olmos
Luis Francisco Barragan Flores
Luis Enrique Castillo Martínez

No. Control
18460609
18460194
18460222

Instituto Tecnológico de Colima Departamento de Sistemas y Computación
Programación lógica y funcional

- [11] Shahaliyev, I. (2020, 13 de septiembre). *Lisp is Beyond Good and Evil*. Medium. <https://medium.com/analytics-vidhya/lisp-is-beyond-good-and-evil-336f6bb763f0> [Último acceso: 10 03 2022].
- [12] hdeleon.net. (2022, 14 de enero). *Introducción a LISP* [Video]. YouTube. <https://www.youtube.com/watch?v=0mRuRXP8J8> [Último acceso: 10 03 2022].
- [13] Microsoft. (2022, 3 de noviembre). *What is F#*. Developer tools, technical documentation and coding examples | Microsoft Docs. <https://docs.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp> [Último acceso: 10 03 2022].
- [14] EDUCBA. (s. f.). *F# Programming | Features | Advantages | Disadvantages*. <https://www.educba.com/f-sharp-programming/> [Último acceso: 10 03 2022].
- [15] Ward, K. (2015, 14 de agosto). *Nota del Editor - foco de Fa sostenido*. Developer tools, technical documentation and coding examples | Microsoft Docs. <https://docs.microsoft.com/es-es/archive/msdn-magazine/2011/august/editor-s-note-f-sharp-focus> [Último acceso: 10 03 2022].
- [16] Desarrollo Web. (2020, 9 de octubre). *Haskell: el lenguaje de programación funcional*. IONOS Digitalguide. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/que-es-haskell/> [Último acceso: 10 03 2022].
- [17] Microsoft. (2021, 11 de abril). *Install F#*. microsoft. <https://docs.microsoft.com/en-us/dotnet/fsharp/get-started/install-fsharp> [Último acceso: 10 03 2022].
- [18] Lenguajes de programación. (s. f.). ▷ Lenguaje de programación SCALA [como aprender a programar]. Lenguajes de programación. <https://lenguajesdeprogramacion.net/scala/> [Último acceso: 10 03 2022].
- [19] Belda Santonja, J. F. y Puentes, J. C. (s. f.). ¿Cuáles son las ventajas y desventajas de lisp? tusclases. <https://www.tusclases.mx/questions/programacion/cuales-son--ventajas--desventajas--lisp> [Último acceso: 10 03 2022].
- [20] Colaboradores de los proyectos Wikimedia. (2008, 1 de septiembre). *F Sharp - Wikipedia, la enciclopedia libre*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/F_Sharp [Último acceso: 10 03 2022].
- [21] Milena Machado, A. (2020, 22 de diciembre). *Descubre por qué aprender Phytton y cuáles son sus ventajas*. Pragma S.A. <https://www.pragma.com.co/academia/lecciones/descubre-por-que-aprender-phyton-y-cuales-son-sus-ventajas> [Último acceso: 10 03 2022].

Nombre de los alumnos
Juan Pablo Anguiano Olmos
Luis Francisco Barragan Flores
Luis Enrique Castillo Martínez

No. Control
18460609
18460194
18460222

Instituto Tecnológico de Colima Departamento de Sistemas y Computación
Programación lógica y funcional

[22] Colaboradores de los proyectos Wikimedia. (2002, 13 de febrero). *Python - Wikipedia, la enciclopedia libre*. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Python#Historia> [Último acceso: 10 03 2022].

[23] Santander Universidades. (2022, 1 de marzo). *Python: qué es y por qué deberías aprender a utilizarlo*. Becas Santander. <https://www.becas-santander.com/es/blog/python-que-es.html> [Último acceso: 10 03 2022].

[24] COVANTEC. (2019). 1.2. *Características — Materiales del entrenamiento de programación en Python - Nivel básico*. Programación en Python - Nivel básico — Materiales del entrenamiento de programación en Python - Nivel básico. <https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/caracteristicas.html> [Último acceso: 10 03 2022].

[25] Yoan. (2017). *El Lenguaje de Programación Python, Pros y Contras*. LovTechnology. <https://lovtechnology.com/lenguaje-programacion-python-pros-contras/> [Último acceso: 10 03 2022].

[26] Python. (s. f.). Welcome to Python.org. Python.org. <https://www.python.org/> [Último acceso: 10 03 2022].

[27] programador clic , «Introducción al lenguaje scala,» programador clic , [En línea]. Available: <https://programmerclick.com/article/7736214510/>. [Último acceso: 10 03 2022].

[28] lenguajesdeprogramacion.net, «Scala,» lenguajesdeprogramacion.net, [En línea]. Available: <https://lenguajesdeprogramacion.net/scala/>. [Último acceso: 10 03 2022].

[29] Readacción KeepCoding, «Aprender Scale,» KeepCoding, [En línea]. Available: <https://keepcoding.io/blog/10-motivos-por-los-que-debes-aprender-scala/>. [Último acceso: 10 03 2022].

[30] docs.scala, «TOUR OF SCALA Introduccion,» docs.scala, [En línea]. Available: <https://docs.scala-lang.org/es/tour/tour-of-scala.html>. [Último acceso: 10 03 2022].

[31] J. Contreras, «Scribd,» [En línea]. Available: <https://es.scribd.com/document/540452180/Scala>. [Último acceso: 10 03 2022].

Nombre de los alumnos
Juan Pablo Anguiano Olmos
Luis Francisco Barragan Flores
Luis Enrique Castillo Martínez

No. Control
18460609
18460194
18460222