

Tema 2. Características de los lenguajes de programación para IA

- ▶ Despliegue
 - Aplicaciones de escritorio
 - Sistemas Enterprise
 - Sistemas embebidos

Un **lenguaje de programación** que sea útil en las anteriores áreas es una buena elección para la resolución de problemas de inteligencia artificial.

Existen cientos de lenguajes de programación, pero solo algunos contribuyen significativamente en la resolución de problemas de inteligencia artificial.

Con el exponiente crecimiento de las tecnologías de **Big Data** e **Inteligencia Artificial** es común utilizar más de un lenguaje de programación en una misma arquitectura software.

Se utilizan entonces diferentes lenguajes de programación y herramientas para cada una de las fases requeridas en los proyectos de Inteligencia Artificial.

Desde la captura de los datos hasta el procesamiento de estos y obtención de conclusiones, se desarrollan **flujos de información** donde cada herramienta se encarga de un proceso diferente.

Tema 3. Principales lenguajes de programación para IA

3.1. Introducción y objetivos

Para estudiar este tema lee las Ideas Clave que se presentan a continuación y revisa los contenidos adicionales para mejorar la comprensión y aumentar los conocimientos sobre las materias tratadas.

Este tema supone un recorrido por los diferentes lenguajes de programación vinculados a la Inteligencia Artificial. Objetivos más concretos son:

- ▶ Ser capaz de identificar la existencia de diferentes **lenguajes de programación**.
- ▶ Conocer los distintos lenguajes de programación y las **características** que los componen.
- ▶ Saber distinguir las **fortalezas** de los diferentes lenguajes de programación.
- ▶ Poder decidir el **uso de un lenguaje de programación** u otro en función del problema a tratar.

Tema 3. Principales lenguajes de programación para IA

3.2. Lenguajes de programación para IA

Existen numerosos

- ▶ Lisp
- ▶ Prolog
- ▶ R
- ▶ Python
- ▶ Haskell
- ▶ Java
- ▶ JavaScript
- ▶ C++
- ▶ Julia

Tema 3. Principales lenguajes de programación para IA

3.3. Python

Python es un lenguaje de propósito general que en los últimos años se ha ido haciendo cada vez más popular en áreas como la ciencia de datos o la inteligencia artificial (IA).

Historia

Python fue creado en 1989 por Guido van Rossum. Guido empezó a implementar este lenguaje como pasatiempo con el objetivo de que fuera fácil de usar y aprender, pero, a la vez, que fuese un lenguaje potente.



Figura 1. Guido van Rossum, creador del lenguaje de programación Python. Fuente: <http://perlcgi-book.com/tag/guido-van-rossum/>

Tema 3. Principales lenguajes de programación para IA

Van Rossum le dio el nombre de «Python» en homenaje al grupo Monty Python del que era fan. Aunque su desarrollo empezó en 1989, la primera versión no se publicaría hasta principios de 1991. Como hemos dicho, el objetivo principal de este lenguaje era que fuese fácil de entender, es decir, que el código que se escribiese con Python fuese más legible que con otros lenguajes de la época como C++ o Java.

Para lograr este objetivo, el desarrollador Tim Peters creó el Zen de Python, que define un conjunto de reglas que representan la filosofía de Python. Todos los usuarios pueden acceder a este conjunto de reglas a través de un easter egg que se introdujo en Python. Para verlo, solo tenemos que ejecutar la instrucción `import this` en una terminal con Python.

Siguiendo la filosofía propuesta en el Zen de Python, se creó una guía de estilo que se encuentra descrita en el Python Enhancement Proposal, abreviado como PEP, versión 8. Esta guía contiene las reglas de estilo que se aplicaron a la hora de desarrollar Python para que se sigan en el desarrollo de nuevas aplicaciones.

PEP 8 -- Style Guide for Python Code

PEP:	8
Title:	Style Guide for Python Code
Author:	Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>
Status:	Active
Type:	Process
Created:	05-Jul-2001
Post-History:	05-Jul-2001, 01-Aug-2013

Figura 2. Información de la guía de estilos de Python PEP8. Fuente: <https://www.python.org/dev/peps/pep-0008/>

Tema 3. Principales lenguajes de programación para IA

Ventajas de Python

Python tiene varias propiedades que lo han convertido en un lenguaje muy potente y fácil de aprender. Estas propiedades son las siguientes:

- ▶ **Tipado dinámico:** Python no necesita que definamos el tipo de las variables cuando las inicializamos como pasa, por ejemplo, en Java o C. Cuando inicializamos una variable Python le asigna el tipo del valor que le estamos asignando. Incluso, durante la ejecución, una misma variable podría contener valores con distintos tipos. Esta propiedad hace que sea más sencillo aprender a usarlo, aunque hace que sea más difícil detectar errores asociados con los tipos de datos.
- ▶ **Lenguaje multiparadigma:** Python permite aplicar diferentes paradigmas de programación como son la programación orientada a objetos, como Java o C++, programación imperativa, como C, o programación funcional, como Haskell.
- ▶ **Interpretado/scripts:** otra ventaja es que podemos ejecutar Python de forma interpretada o usando scripts. Es decir, puedo abrir una consola de Python y escribir y ejecutar las instrucciones una a una o, por otro lado, puedo crear un fichero que almacene todo el programa.
- ▶ **Extensible:** por último, Python cuenta con una gran cantidad de módulos y librerías que podemos instalar para incluir nuevas funcionalidades. Sin embargo, como Python está implementado usando C++, podemos crear nuevos módulos en C++ e incluirlo en Python haciendo que el lenguaje sea extensible a nuevos módulos.

Versiones en Python

Uno de los problemas que tenía una persona que se iniciaba en Python era saber qué versión tenía que instalar, ya que hasta finales de diciembre de 2019 existían dos versiones activas.

Tema 3. Principales lenguajes de programación para IA

Versiónes de Python

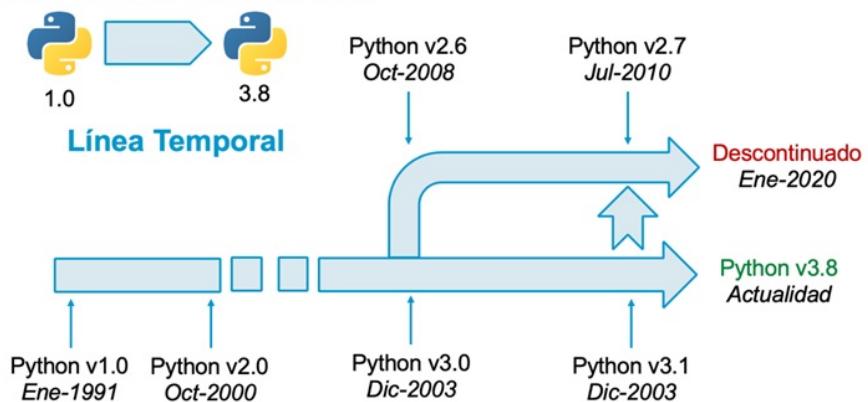


Figura 3. Línea temporal de las versiones de Python.

La primera versión de Python se publicó en 1991, desde entonces se han publicado varias versiones que han seguido siendo más o menos retrocompatibles. Es decir, podía usar esas versiones en programas que había creado en versiones anteriores.

Sin embargo, a finales de 2008 se iba a publicar la versión 3.0. Esta nueva versión era un cambio radical con respecto las predecesoras y esto hacía que no fuera compatible con las versiones anteriores de Python. Por este motivo, se publicó la versión 2.6 como soporte para los desarrollos de la versión 2. En la versión 2.6 se incluyeron nuevas funcionalidades de la versión 3, pero adaptadas a la versión 2. Desde ese momento Python contaba con 2 versiones y las novedades las tenían que duplicar en ambas. Por ejemplo, en 2010 publicaron la versión 3.1 y la 2.7 que incluía las nuevas funcionalidades, pero adaptadas a la versión 2.

Sin embargo, desde el equipo de Python siempre han explicado que las versiones 2.6 o 2.7 eran un parche y que no iban a ser versiones funcionales en un futuro. Este hecho se hizo oficial en enero de 2020 cuando se decidió que la versión 2.7 quedaba descontinuada y que a partir de entonces solo se publicarían nuevas funcionalidades para la versión 3.

Tema 3. Principales lenguajes de programación para IA

3.4. R

R es uno de los lenguajes de programación más populares para el modelado estadístico y el análisis de datos.



Figura 4: Logo de lenguaje de programación R. Fuente: dominio público.

R fue conceptualizado en los Laboratorios Bell por John Chambers en 1976. Este lenguaje fue desarrollado como una extensión, así como una implementación del lenguaje de programación S.

El proyecto R fue desarrollado por Ross Ihaka y Robert Gentleman y lanzado en 1992, su primera versión en 1995 y una versión beta estable en el año 2000.

R es un conjunto integrado de herramientas de software para manipulación de datos, cálculo y gráficos monitor. Algunos de los aspectos clave de R son:

- ▶ Conjunto de operadores para cálculos en matrices.
- ▶ Colección amplia, coherente e integrada de herramientas intermedias para el análisis de datos.

Tema 3. Principales lenguajes de programación para IA

- ▶ Facilidades gráficas para el análisis y visualización de datos
- ▶ Lenguaje de programación bien desarrollado, simple y eficaz (llamado "S") que incluye condicionales, bucles, funciones recursivas definidas por el usuario y facilidades de entrada y salida. (La mayoría de las funciones proporcionadas por el sistema están escritas en el lenguaje S.)

El término “entorno” o “environment” hace que R sea un sistema totalmente integrado, en lugar de una acumulación de herramientas muy específicas e inflexibles, como suele ser el caso con otro software de análisis de datos.

R es en gran medida un vehículo para nuevos métodos de análisis de datos interactivos en desarrollo.

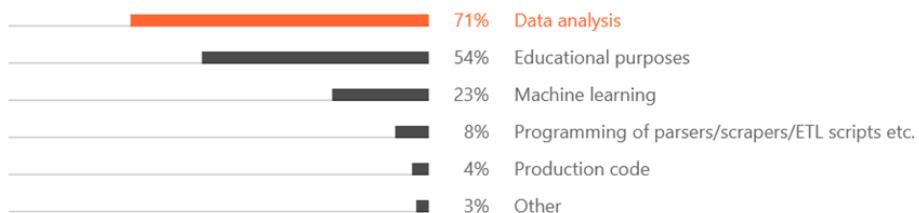


Figura 5: Usos del lenguaje de programación R en la actualidad. Fuente: [JetBrains Devecosystem 2021](#).

Tema 3. Principales lenguajes de programación para IA

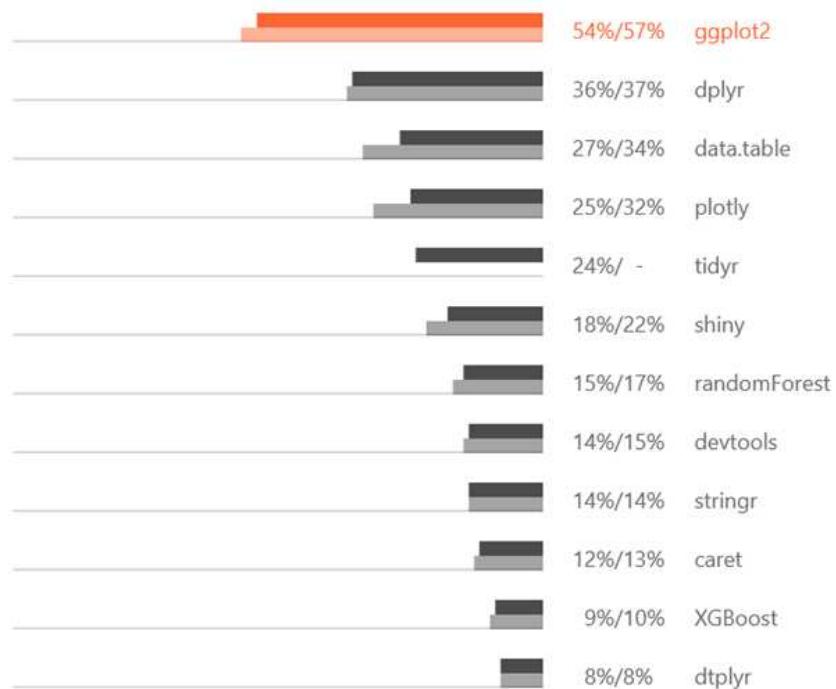


Figura 6: Paquetes más populares para el lenguaje de programación R. Fuente: [JetBrains Devecosystem 2021](#).

Ventajas

R tiene una serie de **ventajas** que lo hacen idóneo para el entorno de tratamiento y análisis de datos:

- ▶ Open Source
- ▶ Buen soporte para la manipulación de datos
- ▶ Amplio ecosistema de paquetes: <https://cran.r-project.org/>
- ▶ Capacidad para visualización de datos
- ▶ Compatibilidad
- ▶ Generación de reportes

Tema 3. Principales lenguajes de programación para IA

- ▶ Portabilidad
- ▶ Útil para proyectos de estadística
- ▶ Ideal para proyectos de aprendizaje automático o Machine Learning
- ▶ Recibe actualizaciones de manera constante

Tema 3. Principales lenguajes de programación para IA

3.5. Java

La tecnología **Java** consiste en un lenguaje de programación y también en una plataforma para el desarrollo de software portable entre sistemas operativos.



Figura 7: Logo de lenguaje de programación Java. Fuente: dominio público.

Surge a comienzos de los años 90 como una iniciativa de la empresa Sun Microsystems para cubrir la necesidad de creación de software para dispositivos inteligentes. En 2010 Oracle toma el relevo de su desarrollo adquiriendo la empresa Sun.

Tema 3. Principales lenguajes de programación para IA



Figura 8: Logo la empresa Sun Microsystems, empresa original en la que surgió el lenguaje de programación Java. Fuente: dominio público.



Figura 9: Logo de Oracle, empresa que desarrolla el lenguaje de programación Java en la actualidad. Fuente: dominio público.

Java es un lenguaje de programación de alto nivel y multiparadigma, es decir, soporta múltiples paradigmas de programación: estructurada, orientada a objetos, funcional.

Es rápido, seguro y fiable, permitiendo a los desarrolladores crear diferentes tipos de aplicaciones software, entre los que destacan:

- ▶ **Aplicaciones standalone:** software que puede ser ejecutado sin ser instalado y que puede trabajar offline.
- ▶ **Aplicaciones web:** software cliente-servidor accedido comúnmente desde un navegador.
- ▶ **Aplicaciones móviles y de internet de las cosas (IoT):** aplicaciones para dispositivos inteligentes.

Tema 3. Principales lenguajes de programación para IA

Principales características del lenguaje Java:

- ▶ **Fácil de aprender:** con respecto a otros lenguajes como C y C++.
- ▶ **Robusto:** tipado fuerte y mecanismos para gestión de excepciones.
- ▶ **Arquitectura neutral:** gracias a la Máquina Virtual Java se puede ejecutar sobre cualquier Plataforma.
- ▶ **Capacidad multihilo:** el propio lenguaje ofrece la capacidad para crear múltiples hilos para ejecutar diferentes tareas simultáneamente.

El código se escribe en ficheros con **extensión .java**, que después son compilados a código intermedio en ficheros con **extensión .class** por el compilador javac.

El código intermedio se conoce como **bytecode** y es interpretado y compilado a código nativo del sistema operativo por la Máquina Virtual de Java.

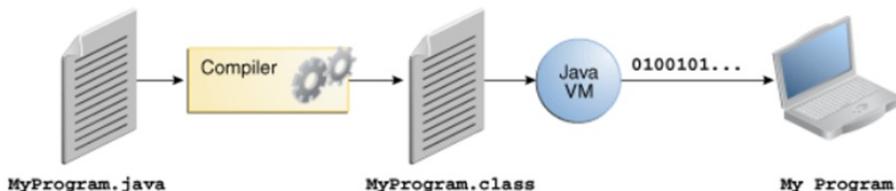


Figura 10: Proceso de compilación y ejecución código Java. Fuente: [Oracle](#).

Tema 3. Principales lenguajes de programación para IA

La plataforma Java es un conjunto de librerías y herramientas para el desarrollo de software.

Se compone de los siguientes elementos:

Bibliotecas estándar o Java Application Programming Interface (API): un conjunto de interfaces y clases con funcionalidades comunes ya desarrolladas para ser utilizadas.

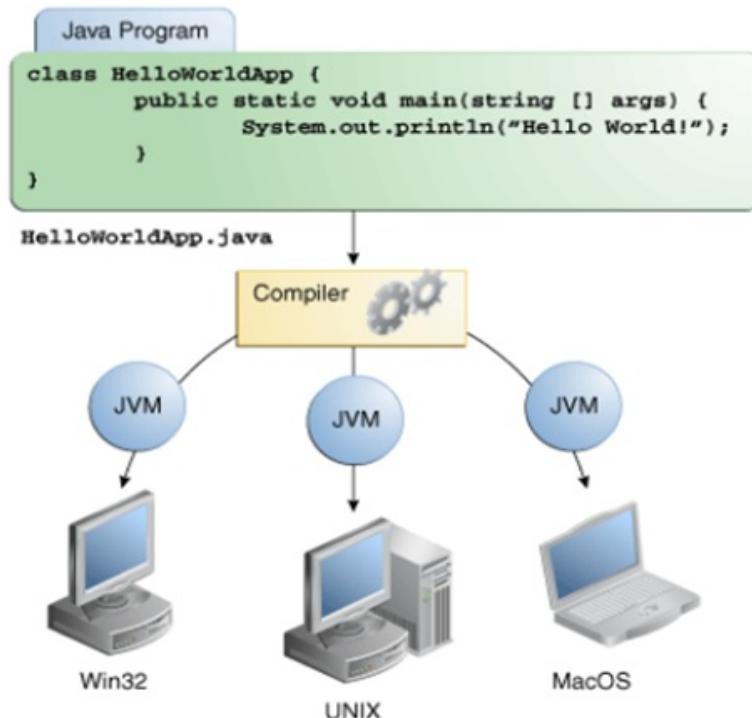
Máquina Virtual Java (JVM): compuesta por los registros, la pila, el recolector de basura, etc. Permite interpretar el *bytecode* creado por el compilador de Java.

La máquina Virtual Java

La **Máquina Virtual Java** es el componente principal de la plataforma Java.

Una de las filosofías principales de Java es la portabilidad entre sistemas. Gracias a la Máquina Virtual Java es posible escribir un código una vez y ser ejecutado en cualquier sistema operativo, transformándose al código nativo correspondiente.

Tema 3. Principales lenguajes de programación para IA



Ediciones de la plataforma Java

Existen múltiples ediciones de la plataforma Java según el tipo dispositivo final sobre el que se ejecutará el software desarrollado, siendo las más populares en la actualidad:

- ▶ **Java SE o Java Standard Edition**: dirigido a estaciones de trabajo y ordenadores de propósito general.
- ▶ **Java EE o Java Enterprise Edition**: ahora conocida como *Jakarta Enterprise Edition*, es un conjunto de librerías añadidas sobre Java SE para incorporar funcionalidades dirigidas a servidores empresariales y entornos distribuidos en Internet, acceso a bases de datos, creación de servicios web, etc.

Tema 3. Principales lenguajes de programación para IA

Versiones Java

Desde el lanzamiento de su versión 9, Java ha adoptado un ciclo de release en el que cada **6 meses** se libera una nueva versión.

Aquellas versiones establecidas como **LTS** o *Long Term Support* son las más recomendadas por ofrecer mantenimiento y actualizaciones a largo plazo.

Early-Access Releases
JDK 18
Loom
Metropolis
Panama
Valhalla

Reference Implementations
Java SE 17
Java SE 16
Java SE 15
Java SE 14
Java SE 13
Java SE 12
Java SE 11
Java SE 10
Java SE 9
Java SE 8
Java SE 7

Figura 12: versiones de Java disponibles. Fuente: [Open JDK](#).

Tema 3. Principales lenguajes de programación para IA

Popularidad de Java

Desde su **aparición en 1995** Java se ha convertido en una de las tecnologías más populares para el desarrollo de software, manteniéndose entre los primeros puestos año tras año.

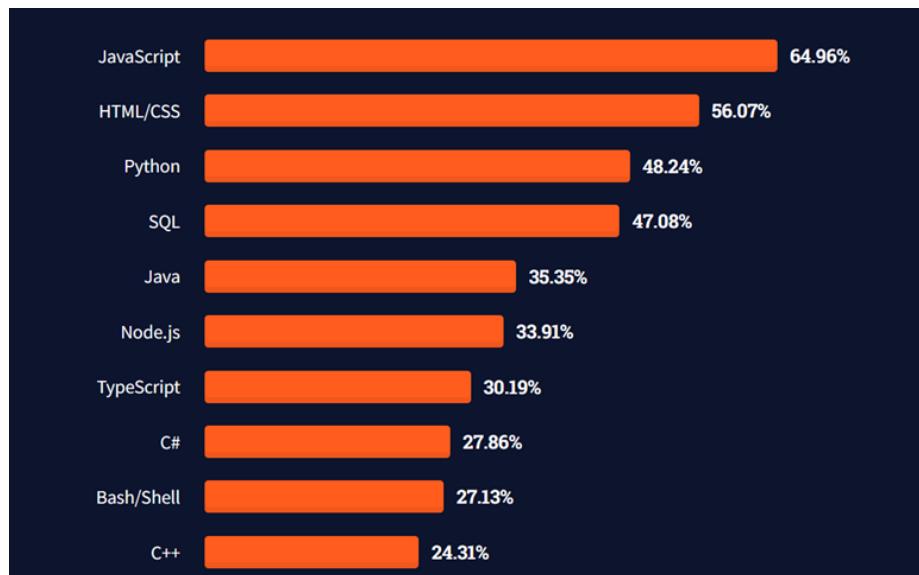


Figura 13: 10 tecnologías más populares según la encuesta [Stack Overflow Developer Survey 2021](#).

Nov 2021	Nov 2020	Change	Programming Language	Ratings	Change
1	2	▲	Python	11.77%	-0.35%
2	1	▼	C	10.72%	-5.49%
3	3		Java	10.72%	-0.96%

Figura 14: los 3 lenguajes de programación más populares. Fuente: Índice [TIOBE](#).

Tema 3. Principales lenguajes de programación para IA

En la actualidad Java se emplea mayoritariamente para el desarrollo de software empresarial, sitios web y herramientas de utilidad. Entre los numerosos usos de Java se encuentra la ciencia de datos y el aprendizaje automático:



Figura 15: Principales usos de Java en la actualidad. Fuente: [JetBrains Devecosystem 2021](#).

Tema 3. Principales lenguajes de programación para IA

3.6. Lisp



Figura 16: Logo de lenguaje de programación Lisp. Fuente: dominio público.

Lisp es un lenguaje de programación bajo el paradigma de la programación funcional, de propósito general, se basa en la aplicación de funciones a los datos y se apoya en la utilización de funciones matemáticas para el control de estos.

El elemento fundamental en Lisp es la lista, una estructura de datos. Cada función de Lisp y cada programa que genera viene dado en forma de lista.

Lisp es un lenguaje de programación creado en los años 50 por John McCarthy con el objetivo de desarrollar un lenguaje que mostrara la información estructurada en listas en las que permitieran gestionarla. De ahí el nombre que le dio a este lenguaje, LISP (List-Processing), porque fue creado principalmente para el procesamiento de listas.

Con el transcurso del tiempo Lisp dio origen a varios dialectos. Entre los más importantes podemos citar a: Scheme, Common Lisp, Clojure, Emacs Lisp, Racket. De todos ellos el Common Lisp se está consolidando como el estándar.

Tema 3. Principales lenguajes de programación para IA

Lisp es considerado en la actualidad como un modelo de referencia para la programación funcional.

Características

Lisp al ser un lenguaje funcional, se basa en la utilización de funciones matemáticas para el control de los datos. Lisp diferencia dos tipos de elementos básicos:

- ▶ El **átomo**, datos elementales de varios tipos como números, símbolos, caracteres y cadenas de caracteres.
- ▶ Las **listas**: dentro destaca la lista “nil”, que es una lista nula que no tiene ningún elemento.

Lisp trata a los elementos o parámetros en las funciones de manera no destructiva, de forma que la mayoría de las funciones devuelven una lista que es el resultado de alguna transformación de otra que recibió, pero sin modificar los parámetros recibidos.

Cabe destacar que Lisp fue uno de los primeros lenguajes en manejar las excepciones con los comandos catch y throw. La **gestión de excepciones** es un pilar fundamental en la creación de programas robustos que puedan gestionar errores de forma correcta sin producir fallos que impidan su utilización.

Los datos en Lisp se organizan:

- ▶ Átomos literales (símbolos)
- ▶ Átomos numéricos
- ▶ La estructura de datos básica es la lista. Incluye primitivas para su manipulación.

Tema 3. Principales lenguajes de programación para IA

- ▶ Los comentarios comienzan por ‘;’
- ▶ Los parámetros de función van todos por valor o por referencia según la clasificación de la función.

Una de las razones por las que Lisp está especialmente vinculado a la programación en inteligencia artificial (IA), es precisamente porque su código y todos los datos tienen la misma estructura, en forma de lista.

Lisp se utilizó como lenguaje de programación en el sistema de inteligencia artificial SHRDLU, un programa informático de comprensión del lenguaje natural (CLN).

Tema 3. Principales lenguajes de programación para IA

3.7. JavaScript

JavaScript (abreviado JS) es un lenguaje de programación de alto nivel y multiparadigma creado para dotar de dinamismo a las páginas web, evitando así una apariencia estática.

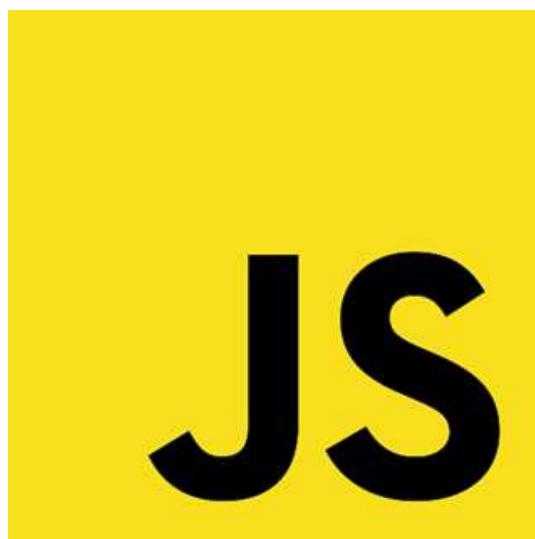


Figura 17: Logo de lenguaje de programación JavaScript. Fuente: dominio público.

Los programas desarrollados en JavaScript se llaman scripts y se pueden escribir directamente sobre documentos HTML o en archivos externos con extensión .js, siendo ejecutados por el navegador cuando se carga la página web.

Actualmente JavaScript es soportado por todos los navegadores web, lo que lo convierte en el lenguaje de programación idóneo para ser ejecutado junto a HTML y CSS para crear páginas web dinámicas, permitiendo mejoras en la interfaz de usuario.

JavaScript es un lenguaje de programación multiparadigma en cuya sintaxis podemos encontrar: variables, operadores, funciones, estructuras de datos, estructuras de control condicional, estructuras de control repetitivo...

Tema 3. Principales lenguajes de programación para IA

En los orígenes se llamaba LiveScript, pero se renombró a JavaScript debido a que por aquella época el lenguaje Java era muy popular, pero conviene destacar que Java y JavaScript son lenguajes completamente diferentes.

```
/**  
 * Get CSS computed property of the given element  
 * @method  
 * @memberof Popper.Utils  
 * @argument {Element} element  
 * @argument {String} property  
 */  
function getStyleComputedProperty(element, property) {  
    if (element.nodeType !== 1) {  
        return [];  
    }  
    // NOTE: 1 DOM access here  
    var window = element.ownerDocument.defaultView;  
    var css = window.getComputedStyle(element, null);  
    return property ? css[property] : css;  
}  
  
/**  
 * Returns the parentNode or the host of the element  
 * @method  
 * @memberof Popper.Utils  
 * @argument {Element} element  
 * @returns {Element} parent  
 */  
function getParentNode(element) {  
    if (element.nodeName === 'HTML') {  
        return element;  
    }  
    return element.parentNode || element.host;  
}
```

Figura 18: Ejemplo de código JavaScript.

La idea principal detrás de la creación de este lenguaje fue sobreponer la limitación que había en aquella época de que las webs fueran estáticas una vez se cargaban en el navegador.

Tema 3. Principales lenguajes de programación para IA

Versiones

JavaScript fue desarrollado en 1995 por Netscape, empresa de software famosa por el navegador popular Netscape navigator.

En 1996 ECMA lo empieza a evaluar como una especificación para todos los navegadores y en 1997 se declara estándar (ECMA-262).

A partir de 2016 se empiezan a lanzar versiones cada año, utilizando el número de año como nombre de versión, en lugar de continuar la numeración ES7, ES8, ES9, ES10, etc.

ECMAScript 1 (ES1)	1997
ECMAScript 2 (ES2)	1998
ECMAScript 3 (ES3)	1999
ECMAScript 4 (ES4)	No se publicó.
ECMAScript 5 (ES5)	2009
ECMAScript 6 (ES6)	2015
ECMAScript 2016	2016
ECMAScript 2017	2017
ECMAScript 2018	2018
ECMAScript 2019	2019
ECMAScript 2020	2020

Figura 19: Tabla de versiones JavaScript.

Tema 3. Principales lenguajes de programación para IA

Machine Learning con JavaScript

TensorFlow

TensorFlow es una biblioteca de código abierto que sirve para desarrollar y entrenar modelos de aprendizaje automático. TensorFlow es una plataforma de extremo a extremo para el aprendizaje automático o machine learning.



Figura 20: Logo de la librería TensorFlow.cFuente: <https://www.tensorflow.org/>

TensorFlow es una biblioteca de código abierto que sirve para desarrollar y entrenar modelos de aprendizaje automático. TensorFlow es una plataforma de extremo a extremo para el aprendizaje automático o machine learning.

Gracias a su ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad, TensorFlow ayuda a los investigadores a innovar con el aprendizaje automático y a los desarrolladores a crear e implementar aplicaciones con tecnología de aprendizaje automático de forma fácil.

Dentro del proyecto TensorFlow existe TensorFlow.js, una biblioteca de JavaScript para entrenar e implementar modelos en el navegador y en [Node.js](#).

TensorFlow.js permite utilizar modelos listos para usar de JavaScript, o convertir modelos de TensorFlow para Python.

Privacidad y personalización

Otra razón por la que JavaScript es una opción interesante para Machine Learning es la personalización de modelos. Por ejemplo, desarrollar un modelo de aprendizaje

Tema 3. Principales lenguajes de programación para IA

automático de generación de texto que se adapte a las preferencias de idioma de cada usuario. Una solución sería almacenar un modelo por usuario en el servidor y entrenarlo con los datos del usuario. Esto supondría una carga adicional en los servidores a medida que el número de usuarios de la plataforma crece y también requeriría almacenar datos potencialmente confidenciales en la nube.

Una alternativa sería crear un modelo base o plantilla en el servidor, crear una copia en el dispositivo del usuario y ajustar el modelo con los datos del usuario mediante bibliotecas de aprendizaje automático de JavaScript.

Por un lado, esto mantendría los datos en los dispositivos de los usuarios y evitaría la necesidad de enviarlos al servidor. Por otro lado, liberaría los recursos del servidor al evitar enviar cargas adicionales de inferencia y entrenamiento a la nube. De esta forma, los usuarios podrían utilizar las capacidades de aprendizaje automático incluso cuando estén desconectados de los servidores, es decir, en entornos offline.

Dispositivos móviles

Otro beneficio del aprendizaje automático de JavaScript es la fácil integración con aplicaciones móviles. El soporte de Python en sistemas operativos móviles aún se encuentra en las etapas preliminares. Pero ya existe un amplio conjunto de herramientas de desarrollo de aplicaciones móviles JavaScript multiplataforma, como, por ejemplo:

- ▶ [Apache Cordova](#)
- ▶ [Ionic](#)
- ▶ [Flutter](#)
- ▶ [React Native](#)

Estas herramientas se han vuelto muy populares porque permiten escribir un mismo código base una vez e implementarlo para dispositivos iOS y Android. Para que el

Tema 3. Principales lenguajes de programación para IA

código sea compatible en diferentes sistemas operativos, las herramientas de desarrollo multiplataforma lanzan una "vista web", un objeto de navegador que puede ejecutar código JavaScript y puede integrarse en una aplicación nativa del sistema operativo de destino. Estos objetos de navegador son compatibles con las bibliotecas de aprendizaje automático de JavaScript.

Una excepción es React Native, un popular marco de desarrollo de aplicaciones móviles multiplataforma que no depende de la vista web para ejecutar aplicaciones. Sin embargo, dada la popularidad de las aplicaciones móviles de aprendizaje automático, Google ha lanzado una versión especial de TensorFlow.js para React Native.

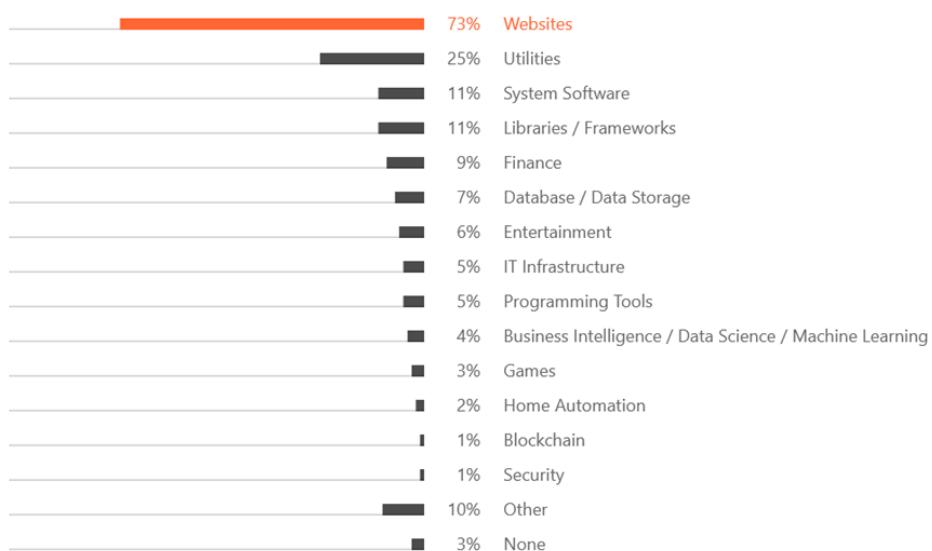


Figura 21: Principales usos de JavaScript en la actualidad. Fuente: [JetBrains Devecosystem 2021](#).

Tema 3. Principales lenguajes de programación para IA

3.8. Prolog



Figura 22: Logo de lenguaje de programación Prolog. Fuente: dominio público.

Prolog es un lenguaje de programación lógico y semi-interpretado. La programación lógica es un paradigma de los lenguajes de programación en el cual los programas se consideran como una serie de aserciones lógicas. La programación lógica es un subparadigma del paradigma declarativo.

Prolog es un lenguaje de programación basado en reglas, desarrollado en la Universidad de Aix-Marseille (Marsella, Francia) por los profesores Alain Colmerauer y Philippe Roussel, como una herramienta práctica para programación lógica.

Nació de un proyecto que no tenía como objetivo la implementación de un lenguaje de programación, sino el procesamiento de lenguajes naturales.

El funcionamiento de Prolog es muy similar a Java. El código fuente se compila a un código objeto o *bytecode* el cuál se interpreta en una máquina virtual denominada Warren Abstract Machine (comúnmente denominada WAM).

Por eso, un entorno de desarrollo Prolog se compone de:

- ▶ Un **compilador**: Transforma el código fuente en código de byte. A diferencia de Java, no existe un Standard al respecto. Por eso, el código de byte generado por un entorno de desarrollo no tiene por qué funcionar en el intérprete de otro entorno.

Tema 3. Principales lenguajes de programación para IA

- ▶ Un **intérprete**: Ejecuta el código de byte. Se trata de una utilidad que permite probar los programas, depurarlos, etc. Su funcionamiento es similar a los interfaces de línea de comando de los sistemas operativos.
- ▶ Una **biblioteca de utilidades**: similar a la biblioteca estándar de Java. Muchos entornos incluyen unas bibliotecas standard-ISO que permiten funcionalidades básicas como manipular cadenas, entrada/salida, etc.

Generalmente, los entornos de desarrollo ofrecen extensiones al lenguaje como pueden ser la programación con restricciones, concurrente, orientada a objetos, etc.

SICStus, CIAO Prolog, y posiblemente otros más, ofrecen entornos integrados generalmente basados en Emacs que resultan muy fáciles de usar. CIAO Prolog además ofrece un auto documentador similar al existente para Java además de un preprocesador de programas. Prácticamente todos ellos son multiplataforma.

Elementos en Prolog

Para construir programas informáticos en Prolog es necesaria una serie de elementos:

- ▶ **Átomos**: es una definición genérica de un objeto del mundo que se quiere representar.
- ▶ **Predicados**: permite especificar características de los objetos del mundo real o las relaciones entre ellos.

Hechos

Es algo que está ocurriendo en el mundo, característica o relación entre objetos. En el lenguaje natural un hecho podría ser por ejemplo que Algoritmos y Estructuras de datos es una asignatura de Ingeniería Informática. Expresan relaciones entre objetos. Suponiendo que se quiera expresar el hecho de que "un coche tiene ruedas". Este hecho, consta de dos objetos, "coche" y "ruedas", y de una relación llamada "tiene".