

Unidad 1: Fundamentos de programación para Inteligencia Artificial

Módulo: “Programación de Inteligencia Artificial”

Curso de Especialización en Inteligencia Artificial y Big Data

IES Serra Perenxisa

Curso 2024/2025

Contenidos

1.- Perfil del programador de inteligencia artificial.	4
1.1.- Conocimientos técnicos.	5
1.2.- Conocimientos matemáticos.	10
2.- Ecosistema en la inteligencia artificial.	13
2.1.- Comunidad de desarrolladores.	13
2.2.- Competiciones y eventos.	14
2.3.- Empleo.	16
3.- Lenguajes de programación.	18
3.1.- Octave.	18
3.2.- Python.	19
3.3.- R.	20
3.4.- Java.	21
3.5.- Lenguajes de marcado.	22
3.6.- Otros.	23

Introducción a la programación de Inteligencia Artificial

En esta unidad, vamos a ver cuáles son los elementos básicos que entran en juego a la hora de utilizar la inteligencia artificial en un problema real, en un negocio u organización. En el mercado laboral actual, los perfiles profesionales implicados en las tecnologías emergentes necesitan estar en un proceso de formación continua, y mantenerse al día a través de comunidades, eventos y proyectos colaborativos. La inteligencia artificial es un campo de estudio que evoluciona muy deprisa en comparación con otras disciplinas, y eso exige estar alerta. Pero, por otro lado, la barrera de entrada va siendo cada vez menor, y hay una oferta muy amplia de entornos de programación, lenguajes y plataformas para desplegar soluciones de este tipo.



[Ministerio de Educación y Formación Profesional](#) (Dominio público)

**Adaptación de los materiales formativos de FP propiedad del
Ministerio de Educación y Formación Profesional.**

[Aviso Legal](#)

1.- Perfil del programador de inteligencia artificial.

Dentro del campo de estudio de la inteligencia artificial, en realidad, **no hay un perfil único**. Al ser un ámbito con una evolución muy rápida, las carreras profesionales tradicionales, no garantizan los conocimientos y habilidades necesarios para llegar a buenos resultados. Por otro lado, algunas tareas a realizar dentro de un equipo en un proyecto de inteligencia artificial, no requieren de conocimientos profundos, aunque sí es recomendable entender bien el paradigma de programación en el que estamos y tener unas nociones de cómo se trabaja con los datos a nivel matemático.

En el escenario actual de la inteligencia artificial, es bastante común la integración de la parte algorítmica con las tareas relativas a los datos, y se ha simplificado la nomenclatura de todo el campo de estudio al término "Data Science". Es decir, **al hablar de "Ciencia de Datos", en realidad, nos estamos refiriendo a una parte de lo que comprende el área de "Big Data", vinculado a las técnicas de trabajo en la que interviene el "Machine Learning"**.

Carmen Bartolomé ([CC BY-SA](#))

Los **"científicos de datos"**, son los que se dedican especialmente al ámbito de la investigación de nuevos algoritmos y soluciones alternativas a los modelos ya existentes, o estudian cómo mejorar, optimizar o ampliar el rendimiento de los datos o simuladores que se usan para el entrenamiento de los modelos.

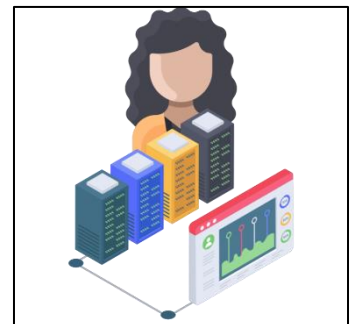


Al tener que analizar diferentes aproximaciones a un problema, o tener que descubrir nuevas formas de aplicar algoritmos de "machine learning", este perfil sí que suele necesitar contar con conocimientos profundos de áreas matemáticas como el álgebra lineal, cálculo numérico y estadística. Es un área en la que hay mucho trabajo por hacer, pero suele estar en manos de equipos de doctorandos en universidades o de ingenieros en grandes tecnológicas como Google o Amazon.

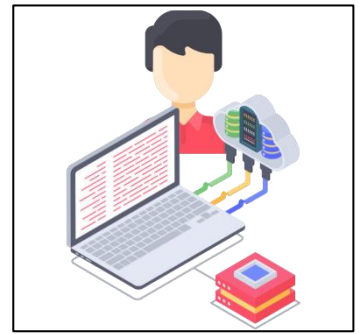
No obstante, hay profesionales y pequeñas empresas que consiguen hacer desarrollos innovadores, y gracias a que lo comparten públicamente, además de ayudar a toda la comunidad, también logran reconocimiento y una mejora importante en su propio proyecto de estudio con las aportaciones de la comunidad.

Los **"ingenieros de datos"**, son los que deben garantizar un buen estado de los datos que entran en el modelo, ya sea de cara al entrenamiento, como después, en producción. Toda la parte de adquisición y gobernanza del dato, se ve en la parte de "Big Data", y en este módulo de Programación de Inteligencia Artificial", supondremos que tenemos los datos y será necesario un cierto tratamiento sencillo para adaptarlos al modelo con el que vayamos a trabajar.

Carmen Bartolomé ([CC BY-SA](#))



El perfil en el que nos vamos a centrar en este módulo, será el de **"analista de datos"**, coloquialmente conocido como **"MLops"**. También se define este perfil como "experto en inteligencia artificial" o "ingeniero de machine learning", que, a pesar de contar con la palabra "ingeniero", no precisa de una titulación universitaria. Este perfil, tiene los conocimientos necesarios para aplicar los desarrollos y herramientas existentes a un problema o situación determinados, para lograr una información de valor que servirá para la toma de decisiones, ya sea dentro de una tarea de negocio, o para la estrategia que es necesario diseñar a futuro en la empresa. Es importante tener en cuenta que este perfil debe conocer bien el negocio o el contexto en el que va a tener que desplegar la solución, y que va a necesitar contar con otros recursos, como un servidor, conexión o acceso a los datos y una interfaz para la presentación de resultados, que puede ser un software, una web, una app móvil,... En realidad, lo normal es que este perfil esté integrado en un equipo multidisciplinar o con otros perfiles técnicos responsables de la parte de negocio o de UI (user interface).



1.1.- Conocimientos técnicos.

Para el desarrollo de una solución de inteligencia artificial, es necesario saber aplicar cálculo computacional para resolver problemas basados en diferentes técnicas y funciones matemáticas. Los científicos ya han planteado y descrito los principales algoritmos que permiten resolver problemas de todo tipo: reconocimiento de imagen, procesamiento del lenguaje natural, modelos analíticos predictivos, sistemas generativos de imágenes, texto, sonido, etc.

Un técnico de inteligencia artificial debe conocer las principales técnicas y modelos existentes y desarrollados por los investigadores, y debe saber adaptarlos al problema concreto que debe resolver, así como saber interpretar los resultados y corregir los errores.

En concreto, debe poder moverse con soltura con los siguientes aspectos técnicos:

Saber programar

Aunque cada vez hay más aplicaciones y plataformas que permiten crear productos y servicios basados en inteligencia artificial sin necesidad de programación ni código en un lenguaje de programación, todavía es bastante necesario contar con habilidades de desarrollo en ciertos lenguajes de programación para a la hora de crear soluciones muy específicas o contar con un verdadero control sobre el desarrollo del producto. Saber programar, además, aporta más habilidades necesarias en este ámbito.

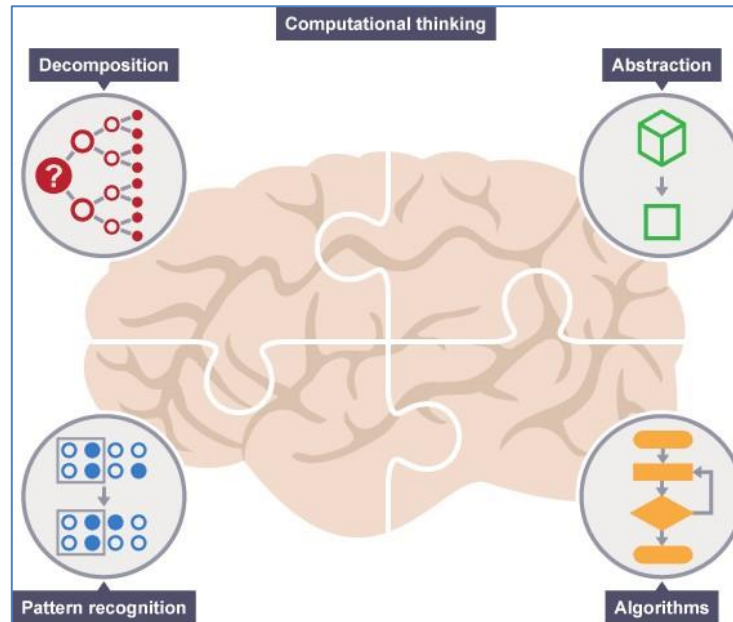
En primer lugar, haber pasado por la experiencia de desarrollo de software, aporta una visión "digital" y un entendimiento del mundo técnico muy necesarios.

Por otro lado, la precisión y robustez que exigen las soluciones de software, entrenan a los que trabajan en ello en habilidades orientadas a optimizar el proceso, simplificarlo al máximo y prever todo tipo de situaciones de fallo o de ambigüedad, que se darán, muy probablemente, también, en otro tipo de entornos o plataformas, aunque no se trabaje directamente con código.

Y, finalmente, es bastante importante haber desarrollado las cuatro habilidades propias del pensamiento computacional que permiten idear y adaptar cualquier problema al ámbito

computacional, donde podremos resolverlo de forma eficiente y minimizando errores.

- ✓ Dividir un problema complejo en partes más simples.
- ✓ Abstracción.
- ✓ Reconocimiento de patrones.
- ✓ Diseño de algoritmos.



[BBC - Introduction to computational thinking \(Licencia BBC\)](#)

Y estas habilidades se adquieren, principalmente, programando. Por ello, es muy recomendable aprender a programar y colaborar en proyectos de código siempre que sea posible, o trabajar en proyectos propios personales para mantener el pensamiento computacional en forma.

Hay muchos técnicos de IA que no necesitan programar en su día a día, pero sí tienen que saber crear y entender cómo se estructura un proyecto de software, o cómo se comportan determinadas arquitecturas de software para saber detectar puntos débiles, errores, etc. A la hora de resolver problemas, sean del tipo que sean, se ha demostrado que las habilidades propias del pensamiento computacional son fundamentales para salir airoso. Y, definitivamente, si se está trabajando en un proyecto de código, será necesario conocer la sintaxis del lenguaje, saber utilizar el entorno de programación que se haya elegido, y tener facilidad para consultar la documentación relativa.

Conocer el entorno Cloud

Casi todas las organizaciones y empresas, hoy en día, tienen su infraestructura en "la nube". Desde hace ya más de una década, se ha ido demostrando la conveniencia de contar con un servicio de servidor en la nube para los datos y aplicaciones que usa la organización. El auge del SaaS (Software as a Service) ha contribuido a que la mayoría de servicios informáticos que las empresas antes debían mantener de forma costosa y poco segura, ahora se puedan utilizar en formato "online", estando siempre actualizados y disponibles.

La nube, o el "cloud computing" permite que la organización se centre en su estrategia y

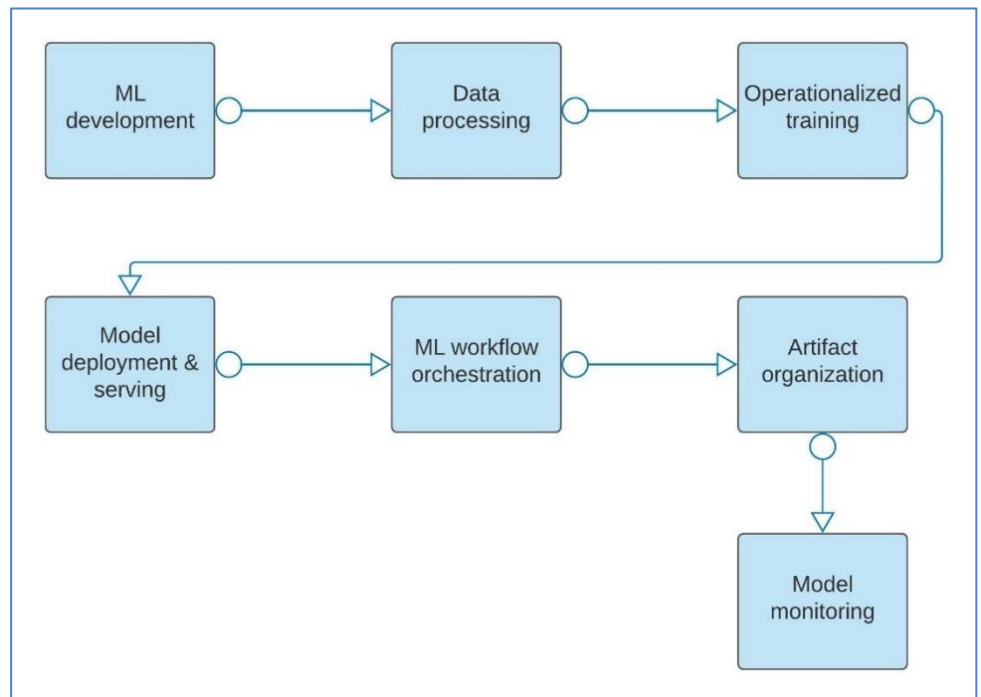
parámetros de negocio, mientras un proveedor de infraestructura se preocupa de que los datos, aplicaciones y desarrollos de ésta siempre estén protegidos y a un coste sostenible, en modo "pago por uso". Los principales proveedores de computación en la nube, son Google Cloud Platform, AWS, IBM Cloud y Azure. Todos ellos te permiten crear una cuenta de desarrollador y probar sus servicios sin coste para un determinado nivel de uso.

Google Cloud Platform

Entre sus servicios, está la herramienta BigQuery, que está orientada a analítica de datos, pero también permite entrar en la experiencia del machine learning. Su servicio específico en el área que nos ocupa, es Vertex AI, una plataforma unificada de aprendizaje automático que contiene la herramienta AutoML.

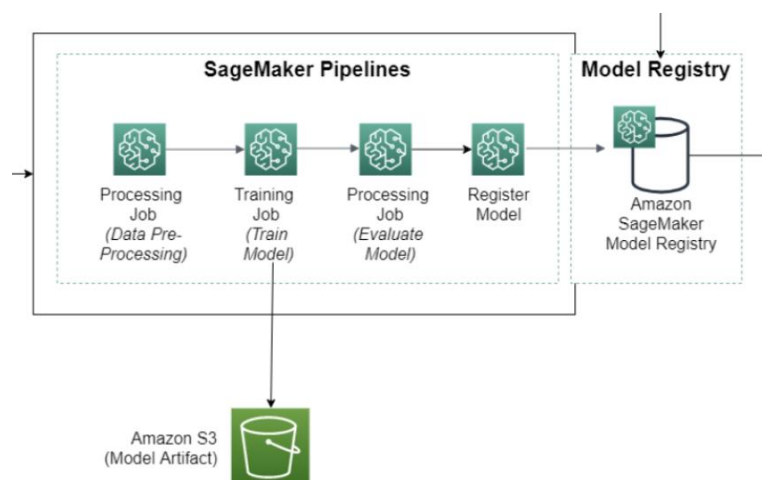
También cuenta con servicios más enfocados en tareas concretas, como por ejemplo Dialog Flow para aplicación de asistentes conversacionales.

[GCP ML pipeline example](#)
(Licencia de uso GCP)



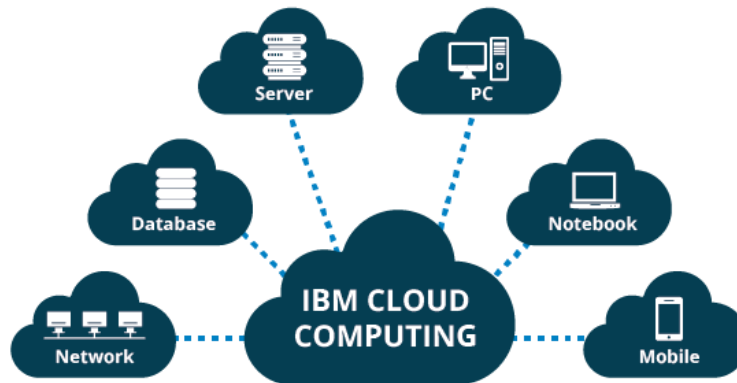
AWS (Amazon Web Services)

En el catálogo de aplicaciones y servicios de Amazon Web Services, las herramientas de IA se segmentan en función de la posibilidad de personalizar y entrenar tú mismo un modelo. La herramienta base, es Amazon SageMaker, que provee de todo lo necesario para crear y entrenar modelos de machine learning con una cierta capa de servicio mayor que si se hace en un IDE con Python, por ejemplo. Pero una propuesta muy interesante, son sus servicios Rekognition para el reconocimiento de imágenes y Comprehend para el análisis de textos y procesamiento de lenguaje natural.



IBM Cloud

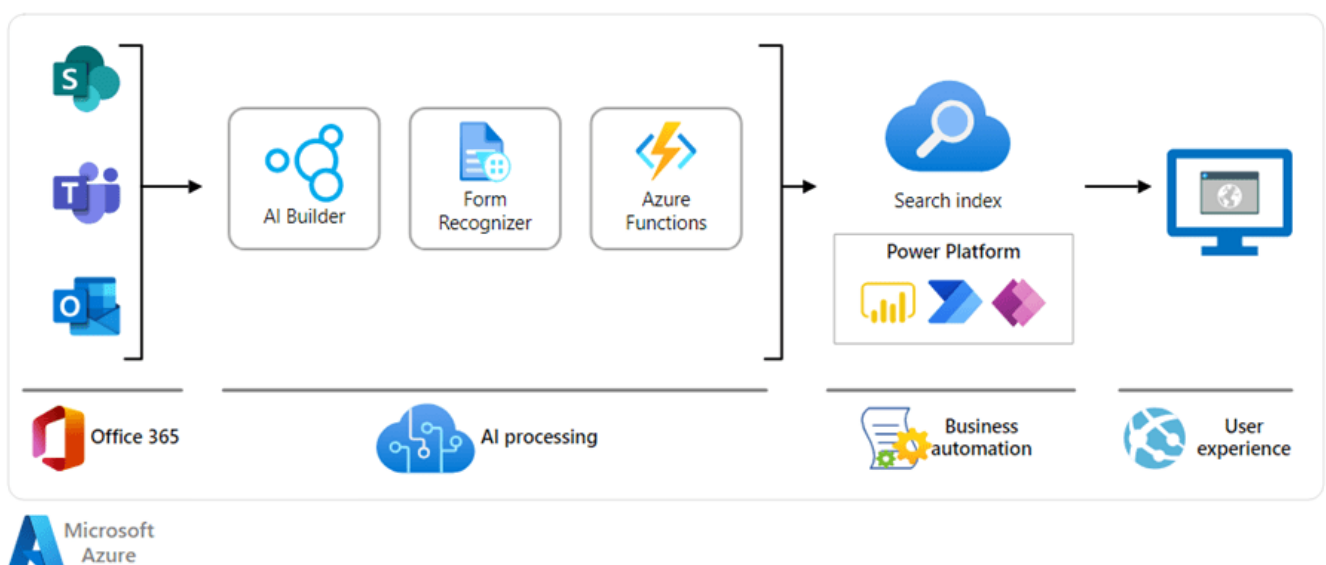
El producto estrella de IBM en el campo de la IA es su sistema cognitivo Watson, que fue el precursor de la era actual de la inteligencia artificial. Aunque en sus inicios estaba enfocado a convertirse en un asistente conversacional, IBM ha ido ampliando su catálogo de servicios incluyendo todo tipo de soluciones.



[IBM Cloud \(Licencia imagen IBM\)](#)

Azure AI

Los servicios en la nube proporcionados por el gigante Microsoft, se concentran en su plataforma Azure. Su catálogo es muy amplio, pero destaca el grupo dentro de su servicio "Cognitive Services" y sus API listas para usar de forma rápida y ágil. Cuentan con modelos ya pre-entrenados que se pueden personalizar después y adaptar a la aplicación de negocio que interese. También cuenta con una serie de soluciones de IA de uso directo basadas en procesos muy habituales en las empresas, y, por otro lado, también se puede recurrir a Azure ML para aplicar técnicas de machine learning y controlar cada paso del proceso.

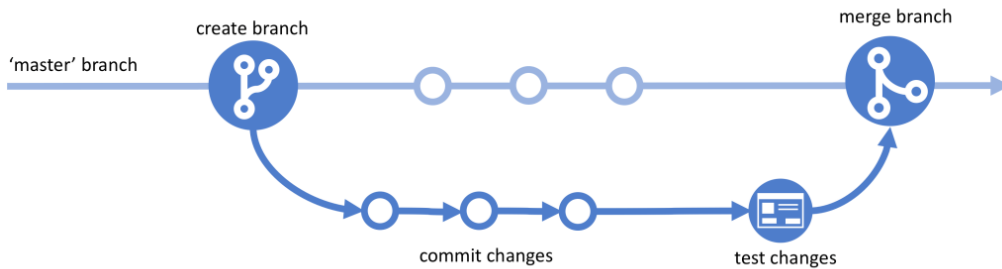


[Azure AI \(Condiciones de uso imagen Microsoft Azure\)](#)

Saber utilizar software de control de versiones. Git

Cuando un equipo debe trabajar de forma colaborativa en un proyecto de código, para que no se pierda ninguna actualización y se pueda ir modificando el código con orden, es muy recomendable utilizar una metodología y/o plataforma en la que vaya quedando registrado el trabajo de cada uno, sugerencias de cambio y la posibilidad de ir incorporando los cambios de forma cómoda, quedando un registro de todas las ramas que se han ido generando y fusionando con el proyecto principal. El software más utilizado es Git, a través de plataformas online como Github o Gitlab.

Simplified Git Flow



Copyright © 2018 Build Azure LLC

<http://buildazure.com>

1.2.- Conocimientos matemáticos.

Las técnicas utilizadas actualmente para crear productos de inteligencia artificial, tienen una base matemática amplia y profunda. Aunque se pueden aplicar sin contar con una formación académica avanzada, siempre es conveniente tener nociones sobre los campos de estudio que han originado las funciones que se utilizan en los algoritmos, así como las técnicas de optimización y cálculo numérico que permiten resolver las tareas propias del Aprendizaje Automático.

Lógica Matemática

El área de las matemáticas que comprende las teorías de modelos, conjuntos, teorías de la demostración y la teoría de la computabilidad, es el de la lógica. Aunque es muy anterior a los ordenadores, el campo de las Ciencias de la Computación, se beneficia de una parte importante de esta disciplina. La teoría de la computabilidad pone los cimientos de la computación en términos lógicos y aritméticos.

Operador	Descripción	Java	Python	C++	JavaScript
<code>==</code>	Igual a	<code>==</code>	<code>==</code>	<code>==</code>	<code>==</code>
<code>!=</code>	Diferente de	<code>!=</code>	<code>!=</code>	<code>!=</code>	<code>!=</code>
<code>&&</code> / and	Lógico AND	<code>&&</code>	<code>and</code>	<code>&&</code>	<code>&&</code>
<code> </code> / or	Lógico OR	<code> </code>	<code>or</code>	<code> </code>	<code> </code>
<code>!</code> / not	Lógico NOT	<code>!</code>	<code>not</code>	<code>!</code>	<code>!</code>

La lógica matemática estudia las definiciones de nociones y objetos matemáticos básicos como conjuntos, números, demostraciones y algoritmos, pero también proporciona las reglas y técnicas para determinar si es o no válido un argumento dado, lo cual es fundamental en cualquier programa computacional.

También, una de sus aplicaciones más directas, es el uso de los operadores lógicos y las sentencias de control. Las variables de tipo "booleano" (True/False) descansan en cierta medida en las bases de la lógica matemática.

Álgebra

El álgebra es la disciplina que trabaja con estructuras numéricas en distintas dimensiones. Se dice que es la matemática de los datos. Es fácil distinguirla por la presencia de elementos apilados en forma de vectores, matrices, tensores, etc.

Estudia las características de estas estructuras de datos y las operaciones y relaciones entre ellas. En cualquier problema con un sistema de múltiples variables de entrada y de salida, es necesario plantear relaciones o ecuaciones entre ellas. Para la resolución de dichas

ecuaciones, el álgebra propone estas estructuras de datos y una serie de reglas para poder trabajar con ellos.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

En el ámbito de la inteligencia artificial, son elementos fundamentales, porque facilitan la infinidad de operaciones que hay que realizar con las técnicas actuales de "machine learning" o aprendizaje automático.

Y dentro del álgebra destaca el subcampo del álgebra lineal.

El **álgebra lineal** es una de las ramas matemáticas más importantes para la inteligencia artificial, especialmente en el aprendizaje automático y el aprendizaje profundo. Muchos de los modelos y algoritmos de IA operan con matrices y vectores, que son objetos matemáticos estudiados en esta disciplina. Algunos conceptos clave incluyen:

- **Vectores y matrices:** Los datos de entrada, pesos y parámetros de muchos modelos se representan en forma de vectores y matrices.
- **Operaciones matriciales:** Sumas, productos y transposición de matrices son fundamentales para cálculos eficientes.
- **Descomposición de matrices:** Técnicas como la descomposición en valores singulares (SVD) son utilizadas en algoritmos como el análisis de componentes principales (PCA).

Ejemplo práctico: en redes neuronales, los pesos entre neuronas y los datos de entrada se representan como matrices, y las operaciones matriciales son esenciales para la propagación hacia adelante y hacia atrás en el entrenamiento de la red.

Cálculo

El cálculo diferencial e integral también es crucial, especialmente en el aprendizaje profundo, donde se utiliza para optimizar los modelos ajustando sus parámetros. El cálculo proporciona las herramientas necesarias para minimizar funciones de pérdida y ajustar los modelos a los datos de entrenamiento.

- **Derivadas:** Las derivadas parciales se utilizan en el proceso de backpropagation en redes neuronales para actualizar los pesos.
- **Gradiente descendente:** Es un algoritmo de optimización que se basa en derivadas para minimizar la función de coste en muchos algoritmos de aprendizaje.

- **Funciones multivariables:** En el entrenamiento de modelos, las funciones objetivo a minimizar suelen ser de múltiples variables, lo que requiere un buen entendimiento del cálculo multivariable.

Ejemplo práctico: El algoritmo de gradiente descendente utiliza derivadas para ajustar los pesos en los modelos de redes neuronales, con el fin de reducir el error entre las predicciones y los valores reales.

Estadística y probabilidad

En realidad, la verdadera disciplina de los datos es la estadística. Aquí la encuadramos dentro de los conocimientos matemáticos, aunque no es del todo una rama de las matemáticas, sino una ciencia a parte que utiliza las matemáticas. Se dedica a estudiar colecciones de datos, su organización y análisis. Cuenta con teorías y ecuaciones que permiten interpretar la relación que hay entre los datos, a través de indicadores que recogen el comportamiento del conjunto como tal.



Se utiliza en multitud de ciencias y contextos, pero en el campo de la inteligencia artificial, ha tenido un papel clave a la hora de introducir las principales técnicas de análisis y exploración de los datos. En muchos casos, simplemente con un adecuado análisis estadístico, podemos extraer bastante información de valor de los datos, sin necesidad de aplicar técnicas más avanzadas. En la gran mayoría de los casos, utilizaremos el análisis estadístico para obtener una "foto" inicial de los datos y orientar mejor las técnicas de "machine learning" o aprendizaje automático.

La probabilidad y la estadística proporcionan las herramientas para modelar el comportamiento de los sistemas de IA y evaluar su rendimiento. Entre los conceptos más importantes se incluyen:

- **Distribuciones de probabilidad:** Modelan eventos inciertos, como por ejemplo las distribuciones: normal, binomial y de Poisson.
- **Inferencia estadística:** Los métodos estadísticos, como la estimación y la prueba de hipótesis, permiten evaluar la confianza de los resultados de los modelos.
- **Teorema de Bayes:** Es fundamental en técnicas como el aprendizaje bayesiano y los modelos generativos.
- **Desviación estándar y varianza:** Miden la dispersión en los datos, lo que es crucial para entender la incertidumbre en las predicciones.

Ejemplo práctico: El algoritmo de **Naive Bayes** utiliza el teorema de Bayes para realizar clasificaciones basadas en probabilidades condicionales.

2.- Ecosistema en la inteligencia artificial.

La inteligencia artificial es un campo de estudio bastante reciente. Tiene su lugar en algunos ámbitos académicos, y ya hay multitud de bibliografía y documentos disponibles para quien quiere especializarse en este campo, pero debido a su rápida evolución, no es fácil mantenerse debidamente actualizado en las técnicas más eficientes a la hora de aplicarla. En todas las disciplinas es importante estar en continuo aprendizaje y seguir los avances y nuevos descubrimientos o desarrollos que se van produciendo, pero en el caso de la inteligencia artificial, es indispensable estar al día para generar verdadero valor en las organizaciones.

La práctica del desarrollo de software, desde la aparición del paradigma "Open Source", se nutre en gran medida de la colaboración y el trabajo en equipo. Colectivamente, podemos explorar nuevas formas de hacer las cosas y de resolver los nuevos problemas que se van planteando mucho mejor que de forma individual. En el desarrollo y aplicación de la IA, es verdaderamente muy recomendable interactuar con la comunidad y no dejar de participar en iniciativas y eventos.

2.1.- Comunidad de desarrolladores.

El sector del desarrollo lleva varios años en transformación hacia un modelo de trabajo en remoto, con equipos distribuidos por todo el mundo que mantienen una dinámica de trabajo consolidada a través de metodologías como Scrum. Sin embargo, los profesionales técnicos, son de los que más crean y colaboran en eventos e iniciativas que fomentan la interacción y la participación, la mayoría de las veces, de forma presencial.

La comunidad que se ha ido formando en torno a la ciencia de datos y la inteligencia artificial, se sostiene y mantiene gracias a varios factores, casi todos ellos iniciativas libres de personas que se van poniendo de acuerdo para lanzarlas, aunque también hay una gran tracción por parte de grandes empresas tecnológicas que utilizan estas comunidades como cantera de profesionales, así como fuente de retroalimentación frente a sus productos y servicios.

La mejor manera de formar parte y mantenerse en contacto con la comunidad de profesionales y entusiastas de la inteligencia artificial es participar en algunas de las plataformas, foros o actividades que existen.

Entrar a formar parte de la comunidad te permitirá, principalmente:

- Aprender cosas nuevas.
- Tener un espacio de referencia para preguntar dudas o encontrar proyectos que te interesen.
- Compartir con otros profesionales de tu mismo perfil tus inquietudes, pasión y logros.
- Debatir sobre diferentes aproximaciones o nuevas tecnologías.
- No permanecer aislado e incluso hacer nuevos amigos.
- Conseguir apoyo y ayuda para proyectos o tests.

- Tener visibilidad desde el punto de vista laboral y aumentar posibilidades de encontrar o mejorar trabajo.

Pero antes de pasar a ver todas las opciones a tu disposición, una advertencia importante: se trata de una relación bi-direccional. Es necesario tanto dar como recibir. Debes tener una actitud positiva ante lo que los demás comparten u ofrecen de la misma manera que a ti te gusta que se respete y valore tu esfuerzo y trabajo. Y no es bueno recurrir a la comunidad solo para recibir: aprovecha ese espacio para exponerte, para presentar tus ideas y experimentos, porque la comunidad te aportará mucho para mejorarlo y ganarás visibilidad.

2.2.- Competiciones y eventos.

Una de las formas más directas y eficientes de mantenerse en contacto con la comunidad, hacer contactos de valor y estar al día en las novedades de la industria de la inteligencia artificial es asistir a eventos del sector. También es muy recomendable participar en competiciones abiertas, porque se conoce y colabora con otros profesionales y amantes de la inteligencia artificial, se aprende mucho, y se gana visibilidad de cara empresas que convocan estas competiciones u otros participantes que también están interesados en conocer gente con este perfil.

Eventos

La mayoría de las grandes tecnológicas cuentan con un calendario de eventos presenciales y online. Te recomendamos que entres en sus páginas web y busques la sección de eventos. La mayoría son gratuitos, pero suele ser necesario registrarse. No siempre habrá eventos convocados en tu ciudad, pero la opción de seguir el evento de forma online ya está muy generalizada. Incluso en el formato online, hay bastantes opciones de interacción a través de un chat simultáneo y moderación de preguntas para los ponentes.

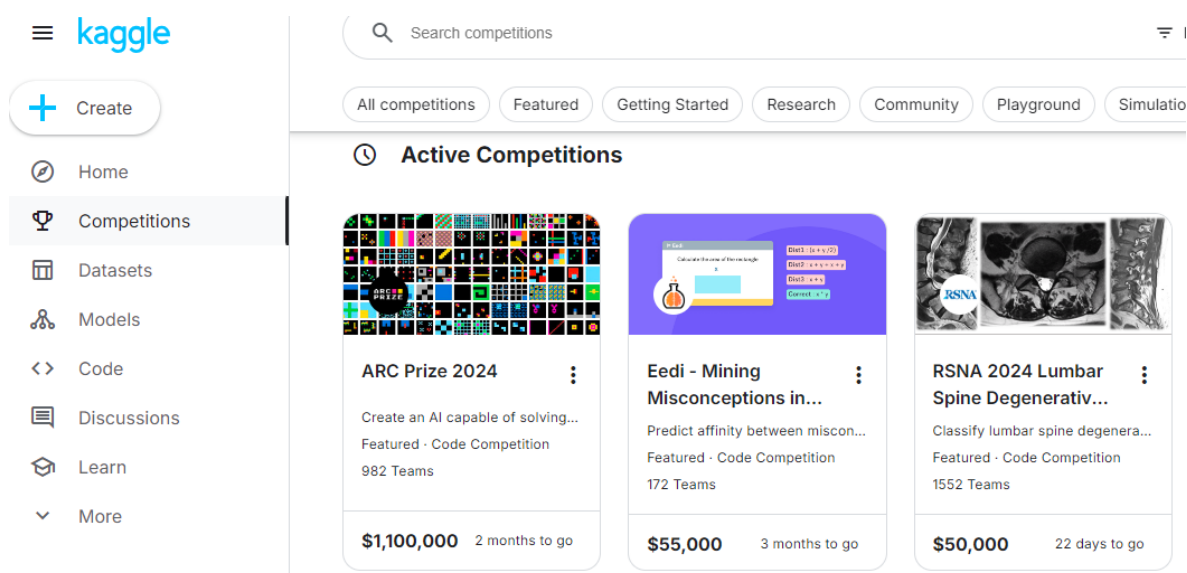
Otra opción interesante para conocer eventos en tu localidad, es utilizar plataformas como meetup, en las que se pueden publicar grupos con calendario de eventos y puedes suscribirte a una de estas comunidades. Un ejemplo, sería el meetup "Machine Learning Spain". Solo tienes que entrar en la plataforma y hacer una búsqueda por localidad y tema del evento, y aparecerán todas las comunidades activas en un cierto radio, y dentro de cada una, su calendario de eventos pasados y futuros. Suelen ser eventos con poco aforo, por lo que se facilita mucho una interacción más cercana y frecuente. También puedes plantearte colaborar con la organización de alguno de estos eventos en tu ciudad, pues es una forma de aprender y adquirir también experiencia en otras habilidades.

Por otro lado, las grandes plataformas tecnológicas también suelen convocar eventos formativos sobre sus servicios y productos o sobre el sector en general que pueden ser muy útiles. En las webs para desarrolladores de cada una de ellas, hay opciones para suscribirse a alertas de noticias sobre novedades y eventos, a través del email. Aunque, en estos casos, el objetivo principal del evento será promocionar sus productos, se suelen impartir ponencias

y talleres que ayudan a mejorar las habilidades de los desarrolladores que deben trabajar sobre dichas plataformas.

Competiciones

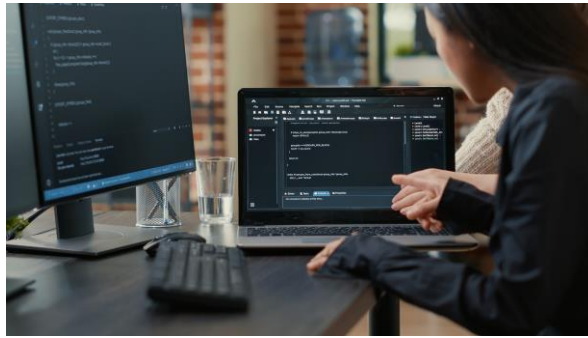
En el sector del desarrollo de software, es tradicional la convocatoria de los denominados "hackathones". Se trata de un evento en el que se plantea un problema o reto y los participantes conforman equipos para ponerse a trabajar sobre un producto de software que lo resuelva. Suele tener una duración acotada de pocos días en los que los equipos trabajan intensivamente. A menudo, los equipos se forman al principio del evento, buscando contar con varios perfiles de forma que cada equipo sea lo más multidisciplinar posible. Suelen ser presenciales, y la organización provee de servicios básicos para que los participantes tengan todo lo necesario para las horas que van a pasar trabajando, pero casi siempre, los participantes suelen traer sus propios ordenadores y equipamiento.



[Kaggle.com](https://www.kaggle.com) (CC BY-NC-SA)

Hay algunas plataformas de competencias que ofrecen posibilidades interesantes para los que no tienen fácil asistir a los hackathones presenciales. Una plataforma que ha crecido mucho en los últimos años y que se ha convertido en referencia para todos los que desarrollan modelos de machine learning, es [Kaggle](https://www.kaggle.com). En esta plataforma se publican retos por parte de empresas y organizaciones en los que es necesario utilizar los datasets que éstas comparten y crear un modelo que, tras ser entrenado con dichos datos, sea útil para la misión de la organización que recurre a la comunidad. Por otro lado, después de varios años de publicación de retos, la plataforma cuenta con un catálogo de datasets muy variado y amplio, por lo que es un medio ideal para todos los que están aprendiendo o que necesitan probar nuevas técnicas con datos reales y actuales en algún caso de uso concreto.

2.3.- Empleo.



[DCStudio](#) (CC BY-SA)

La inteligencia artificial es herramienta y producto para las empresas

Finalmente, en el ecosistema de la IA, el mayor peso lo tiene la industria, las empresas, en definitiva, el negocio. Encontramos dos casos de desarrollo de IA en las empresas:

- ✓ **Productos o servicios internos**, que mejoren la operativa de negocio o su gestión. En este grupo, tenemos toda el área de analítica avanzada de negocio, herramientas de IA para marketing, ventas, monitorización de sistemas, etc.
- ✓ **Productos o servicios para cliente**, que la empresa crea y comercializa. Es el caso de consultoras, empresas de desarrollo de software o startups con aplicaciones de IA.

Búsqueda de profesionales de IA

Dado que todas las organizaciones y empresas necesitan transformar sus procesos para hacerlos más eficientes y la IA es uno de los mejores medios para ello, hay una creciente demanda de este tipo de profesionales. Pero no siempre se consigue gestionar adecuadamente esta búsqueda con los métodos tradicionales de selección de talento.

A la hora de entrar en el mercado laboral relacionado con la IA, hay que tener en cuenta que hablamos de un sector todavía relativamente joven y muy cambiante. Dentro del área técnica, hay varios factores relativos al candidato a un puesto que suelen contribuir a generar confianza en el reclutador o la persona que debe tomar la decisión de contratar:

- ✓ Contar con experiencia laboral en un puesto de trabajo relacionado con el desarrollo dentro del área concreta que se busca.
- ✓ Contar con un portfolio de proyectos o desarrollos relacionados con la tecnología en la que se quiere cubrir el puesto.
- ✓ Tener buenas puntuaciones o reconocimiento en competiciones, plataformas colaborativas, etc.
- ✓ Ser emprendedor con un prototipo de la tecnología de interés para el puesto que se quiere cubrir.

En general, cualquier iniciativa que sirva para darte visibilidad y hacer contactos, será de ayuda, pero también debes mostrar tus habilidades, y en un ámbito que evoluciona tan deprisa como es la inteligencia artificial, implica un proceso continuo de "aprender haciendo".

Emprendimiento en el área de la IA

La disminución de barreras de entrada en el aprendizaje y desarrollo de inteligencia artificial, ha facilitado que surjan proyectos de emprendedores que proponen ideas de productos o aplicaciones que se basan en la IA para resolver problemas o necesidades que se detectan en la sociedad. A menudo, el proceso es el siguiente:

- ✓ El emprendedor tiene una idea. Le da forma y prepara un Plan de Negocio.
- ✓ Busca un socio técnico que le ayude a preparar un prototipo en modo MVP (mínimo producto viable). Este socio es el desarrollador de IA. Se reparten las participaciones de la empresa, pero no suele haber remuneración mensual.
- ✓ Se presenta el Plan de Negocio y el prototipo a inversores. Si hay suerte, los inversores aportan capital y el proyecto se lanza al mercado. En esta etapa, ya hay remuneración para socios y trabajadores de lo que ya es una empresa.
- ✓ Cuando el negocio genera suficientes beneficios, se alcanza lo que se denomina "break even" y los socios (incluidos los inversores) perciben dividendos anuales.
- ✓ Si el negocio es adquirido por otra corporación, los socios perciben la cantidad de capital de la venta correspondiente a la cantidad de participaciones que tienen del negocio. En este punto es cuando los inversores esperan recibir una cantidad entre 5 y 10 veces el capital invertido.

Ser desarrollador de IA para una startup implica lidiar con la incertidumbre, soportar presión y una gran carga de trabajo. Pero permite aprender deprisa y adquirir una experiencia realmente valiosa.

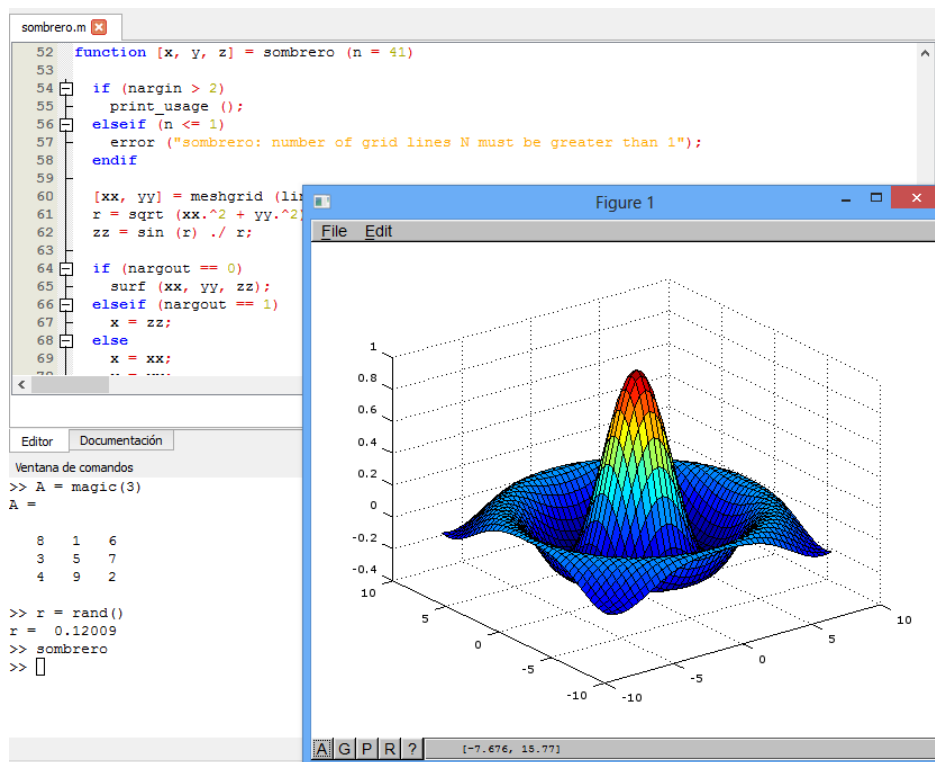
3.- Lenguajes de programación.

Los algoritmos en los que se basa la inteligencia artificial actual son principalmente planteamientos muy matemáticos, y por eso, se suele programar en lenguajes de programación que tengan una buena orientación hacia ese tipo de desarrollos. Hay lenguajes que son muy eficientes en cálculos matemáticos, pero si son de bajo o medio nivel, harán demasiado compleja y tediosa la programación de tareas como las que requiere la inteligencia artificial.

Además, estos lenguajes no son adecuados para trabajar con estructuras grandes o dinámicas de datos, porque utilizan la memoria de forma menos eficiente que los de alto nivel. Por lo tanto, se van a utilizar casi siempre lenguajes de alto nivel, que son compilados o ejecutados por un intérprete.

Por otro lado, las aplicaciones más orientadas a negocio suelen utilizar un tipo de lenguajes, como Python o Java, mientras que en el ámbito de la investigación o de la ingeniería, se utilizan otros como Matlab o su homólogo "open source" Octave. Dentro de una estrategia "Data Driven" o de negocio basado en los datos y la estadística, será común la utilización del lenguaje R, aunque para inteligencia artificial es un poco limitado respecto a otros. Si estamos hablando de ejecución dentro de un dispositivo móvil o en web apps que lo necesitan, también podría utilizarse Javascript o Node.js. Y, aunque no es muy común, para aplicaciones de alto rendimiento, se recurre a C++.

3.1.- Octave.



[GNU Octave \(GNU/GPL\)](#)

GNU Octave es un lenguaje de programación científica con una sintaxis muy orientada al cálculo matemático. Es parte del proyecto GNU, por lo que es de código abierto u "open source", y gratuito. Al mismo tiempo, es compatible con Matlab, por lo que constituye una alternativa muy interesante cuando hay que colaborar en proyectos en los que también se utiliza éste. Está escrito en C++ y tiene un intérprete propio, que se suele ejecutar en su propio IDE (Entorno de desarrollo integrado). Es compatible con otros programas GNU para enriquecer la parte gráfica.

Su potencia en el planteamiento de cálculos matemáticos y sus herramientas de visualización 2D y 3D los han hecho muy adecuado para la investigación en el campo del machine learning y la optimización.

En su propia web están las distintas opciones de descarga según el sistema operativo y puedes consultar cualquier duda en su wiki.

```
1  n = 10;
2  suma = 0;
3
4  % Calcular la suma de los primeros n números enteros
5  for i = 1:n
6      suma = suma + i;
7  end
8
9  fprintf('La suma de 1 a %d es: %d\n', n, suma);
```

3.2.- Python.

Python es un lenguaje de programación de propósito general sencillo y fácil de aprender, pero al mismo tiempo es muy potente y versátil, por lo que se ha extendido bastante en el ámbito académico y científico. Es un lenguaje multiplataforma, de tipado dinámico y adaptable a distintos paradigmas: imperativo, orientado a objetos, funcional...

Una de sus principales ventajas es que, además de soportar objetos también ofrece programación estructurada y funcional, siendo este último un paradigma que va cogiendo fuerza y aceptación con los nuevos desarrollos de la industria actual. Se adapta al sistema operativo en el que se quiera utilizar y cuenta con un extensísimo catálogo de librerías creadas y mantenidas por la comunidad.

Pero su normalización en el ámbito de la inteligencia artificial ha tenido bastante que ver con la liberación del código de la librería TensorFlow por parte de Google. Esta librería estaba escrita en Python, lo que dio una ventaja especial a la comunidad de este lenguaje de programación, que se aceleró después y que ha contado con la contribución de multitud de proyectos de investigación en este campo de estudio desde un marco académico y científico.

El objetivo principal de este lenguaje es su legibilidad, es decir, que sea un lenguaje fácil e intuitivo, que se entienda casi como cualquier texto en inglés.

```
1  def sumar(n1, n2):
2      return n1 + n2
3
4  resultado = sumar(3, 7)
5  print(resultado)
```

Pero, por otro lado, se le intenta dar toda la potencia del resto de lenguajes de su nivel, apto para programar rápidamente tareas sencillas o construir prototipos en poco tiempo. Es un proyecto de código abierto, lo cual ha sido determinante de cara a su adopción y evolución. Su capacidad para ser "incrustado" en otros

programas escritos en C y C++ también lo hacen especialmente útil y aceptado en proyectos grandes.

Se trata del lenguaje más empleado para la programación en inteligencia artificial. En este ámbito, además, existen varias bibliotecas que podemos utilizar para simplificar el desarrollo de aplicaciones con IA, como las siguientes:

- **NumPy**: ideal para manejar colecciones de datos multidimensionales.
- **Pandas**: facilita el análisis y filtrado de datos.
- **Scikit-learn**: para el modelado de machine learning.
- **TensorFlow/Keras/PyTorch**: útiles para construir redes neuronales, simples y complejas.
- **Matplotlib** o Plotly, entre otras, para la visualización gráfica de datos relacionados con un problema.

3.3.- R.

R es un entorno de software libre y lenguaje de programación interpretado, sin necesidad de compilación. Se integra bien con otros lenguajes de programación como C o Fortran, lo que le permite la ejecución de tareas de análisis de datos muy voluminosas o intensivas. No es tanto un lenguaje para un código estático sino más bien unas instrucciones para generar un informe final, que es lo que realmente se busca.

R es un lenguaje cuyas librerías se han centrado más en cálculo estadístico, visualización y flujo de datos en general. Cuenta con un repositorio centralizado de paquetes que facilita bastante la programación de tareas de analítica de datos y lo que se conoce como "Big Data" o "Smart Data".

Más que un lenguaje para programación de inteligencia artificial, podemos decir que se trata de un lenguaje auxiliar o complementario para parte de un proyecto integral en el que tengamos que atender al proceso completo desde la adquisición de datos hasta la obtención de predicciones y su presentación en una interfaz o aplicación, en el que el motor de inteligencia artificial se habrá generado y se ejecutará muy probablemente en otro lenguaje.

```
1  numeros <- c(5, 10, 15, 20, 25)
2
3  # Calcular la media y DESV
4  media <- mean(numeros)
5  desv_std <- sd(numeros)
6
7  # Mostrar los resultados
8  cat("Los números son:", numeros, "\n")
9  cat("La media es:", media, "\n")
10 cat("La desviación estándar es:", desv_std, "\n")
```

Al igual que ocurre con otros lenguajes, existen varios paquetes o librerías que podemos incorporar a este lenguaje relacionadas con la inteligencia artificial:

- **tidyverse**, un conjunto de librerías específicas para ciencia de datos
- **ggplot2**, para representaciones gráficas de conjuntos de datos
- **dplyr**, para manipulación y transformación de datos (incluida en *tidyverse*)
- **Caret**, para modelos predictivos

3.4.- Java.

Podríamos decir que Java es el lenguaje de programación de la industria. Es el más utilizado en proyectos de negocio, en plataformas y herramientas de gestión, etc. Pero a medida que la programación de inteligencia artificial y modelos de machine learning ha ido evolucionando hacia el paradigma de la programación funcional, Java, que es un lenguaje con una fuerte base en la POO (Programación Orientada a Objetos), va dejando de ser utilizado por una nueva generación de desarrolladores en este campo.

Sin embargo, los desarrolladores que están acostumbrados a programar otros tipos de aplicaciones y que lo conocen muy bien, a la hora de programar inteligencia artificial, prefieren utilizar los recursos disponibles en este framework antes que iniciarse en Python.

Este lenguaje de programación permite un mayor control del hardware, y su JVM (Java Virtual Machine) soporta la ejecución de otros lenguajes adaptados desde sus originales. Por ejemplo, Jython es el lenguaje JVM adaptado desde Python. Otra ventaja es que se puede ejecutar en cualquier arquitectura.

Entre sus principales opciones orientadas al desarrollo de aplicaciones de inteligencia artificial, podemos destacar las siguientes:

- **MLib**, una librería para *machine learning* que funciona sobre Apache Spark, una plataforma para el procesamiento de datos a gran escala
- **DL4J**, una librería para desarrollo de aplicaciones *deep learning*
- **Java-ML**, una librería con distintos algoritmos y estructuras de *machine learning*
- **MOA**, *Massive Online Analysis*, destinada al análisis de grandes cantidades de datos
- **Neuroph**, framework para creación de redes neuronales.

```
import java.util.ArrayList;
import java.util.List;
import org.javaml.classification.LinearRegression;
import org.javaml.core.Dataset;
import org.javaml.core.Instance;

public class JavaMLLinearRegressionExample {
    public static void main(String[] args) {
        List<Instance> instances = new ArrayList<>();

        // Datos de entrenamiento: (x, y) donde y = 2*x + 1
        instances.add(new Instance(new double[]{1.0}, 3.0)); // y = 2*1 + 1
        instances.add(new Instance(new double[]{2.0}, 5.0)); // y = 2*2 + 1
        instances.add(new Instance(new double[]{3.0}, 7.0)); // y = 2*3 + 1
        instances.add(new Instance(new double[]{4.0}, 9.0)); // y = 2*4 + 1
        instances.add(new Instance(new double[]{5.0}, 11.0)); // y = 2*5 + 1

        Dataset dataset = new Dataset(instances);
        LinearRegression model = new LinearRegression();
        model.build(dataset);

        // Predecir un nuevo valor
        double prediction = model.predict(new double[]{6.0});
        System.out.println("Predicción para x = 6: " + prediction);

        // Mostrar coeficientes
        System.out.println("Pendiente: " + model.getWeights()[0]);
        System.out.println("Intersección: " + model.getBias());
    }
}
```

3.5.- Lenguajes de marcado.

Además de los lenguajes de programación que utilizamos para desarrollar aplicaciones de inteligencia artificial, es fundamental considerar otros tipos de lenguajes que nos ayudan a organizar la información con la que trabajamos, nos referimos a los lenguajes de marcado.

En la actualidad, hay tres o cuatro lenguajes que son predominantes en el marcado o etiquetado de datos, y podemos utilizar cualquiera de ellos, así como otros adicionales, para almacenar la información necesaria para nuestros programas. A continuación, revisaremos algunos ejemplos y características clave de: HTML, XML, JSON y CSV.

HTML (HyperText Markup Language) es el lenguaje más utilizado para crear contenido web. Todas las páginas web que visitamos en la actualidad utilizan este lenguaje para etiquetar la información. Mediante sus etiquetas, es posible organizar el contenido en diversas secciones o áreas, y estructurarlo en listas, tablas, párrafos, entre otros formatos.

```
<DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Inteligencia Artificial</title>
</head>
<body>
  <h1>Áreas de la IA</h1>
  <ul>
    <li>Reconocimiento de voz</li>
    <li>Visión por computadora</li>
  </ul>

  <h2>Herramientas</h2>
  <table border="1">
    <tr>
      <th>Herramienta</th>
      <th>Uso</th>
    </tr>
    <tr>
      <td>TensorFlow</td>
      <td>Aprendizaje profundo</td>
    </tr>
  </table>
</body>
</html>
```

XML (eXtended Markup Language) es un lenguaje de marcado para propósito general. Permite definir un conjunto de marcas o etiquetas personalizadas con las que estructurar y clasificar la información que queremos almacenar en un documento. Así, por ejemplo, podríamos almacenar datos de un catálogo de libros en un archivo XML.

```
<dataset>
  <image>
    <filename>manzana.jpg</filename>
    <label>manzana</label>
  </image>
  <image>
    <filename>banana.jpg</filename>
    <label>banana</label>
  </image>
</dataset>
```

JSON: a la hora de almacenar información y poderla consultar o incluso enviar en comunicaciones por red, desde hace varios años se está imponiendo el lenguaje llamado JSON (*JavaScript Object Notation*). Su popularidad se debe a su simplicidad y a lo compacto que resulta en comparación con XML. Se necesitan muchos menos caracteres (y mucha menos memoria) para estructurar la misma información, siendo algo similar a los arrays en JavaScript.

```
{
  "dataset": [
    {
      "filename": "manzana.jpg",
      "label": "manzana"
    },
    {
      "filename": "banana.jpg",
      "label": "banana"
    }
  ]
}
```

CSV (Comma-Separated Values) es un tipo de archivo de texto plano donde los datos se organizan en filas y columnas, separadas por comas o punto y coma. Es ampliamente utilizado en inteligencia artificial debido a su simplicidad y compatibilidad con diversas herramientas y lenguajes de programación. Los datasets en CSV son fáciles de manipular y procesar para tareas de **machine learning** o **deep learning**, ya que permiten almacenar grandes cantidades de datos estructurados de manera eficiente, facilitando el análisis, la limpieza y la alimentación de modelos para su entrenamiento.

```
filename,label,width,height
manzana.jpg,manzana,256,256
banana.jpg,banana,256,256
```

3.6.- Otros.

Hay desarrollos de inteligencia artificial en otros lenguajes de programación. Por ejemplo, aunque Javascript es considerado el lenguaje de programación de la web, existen librerías y herramientas que permiten ejecutar algoritmos de machine learning en aplicaciones, corriendo en Javascript o en sus frameworks más utilizados, como Node.js. Ejemplo: TensorFlow.js

También es posible encontrar desarrollo de inteligencia artificial programados en C y en C++. Este lenguaje, además de ser la base de otros lenguajes de alto nivel que se usan en el campo del machine learning, cuenta con librerías muy potentes para aplicaciones que requieren de rapidez y rendimiento.

Por otra parte tenemos a Scala. Se trata de un lenguaje creado en 2003, y orientado a un paradigma funcional. Este paradigma es especialmente indicado para el análisis de datos, y también se puede aplicar el lenguaje a diferentes tareas relacionadas con el ámbito del *machine learning*. Es un lenguaje multiplataforma y orientado a objetos, con algunas características similares a Java, con quien comparte su entorno de ejecución (*JVM*, máquina virtual de Java).