

Investigación y Gestión de Proyectos en Inteligencia
Artificial

Tema 5. Gestión de proyectos de inteligencia artificial. Enfoque metodológico

Índice

Esquema

Ideas clave

5.1. Introducción y objetivos

5.2. Definición de proyectos de inteligencia artificial empresarial

5.3. Tipos de proyectos de inteligencia artificial empresarial

5.4. Tipos de despliegue de proyectos de inteligencia artificial empresarial

5.5. Ciclo de vida de un proyecto de inteligencia artificial empresarial

5.6. Metodologías y marcos de trabajo para la gestión de proyectos de inteligencia artificial

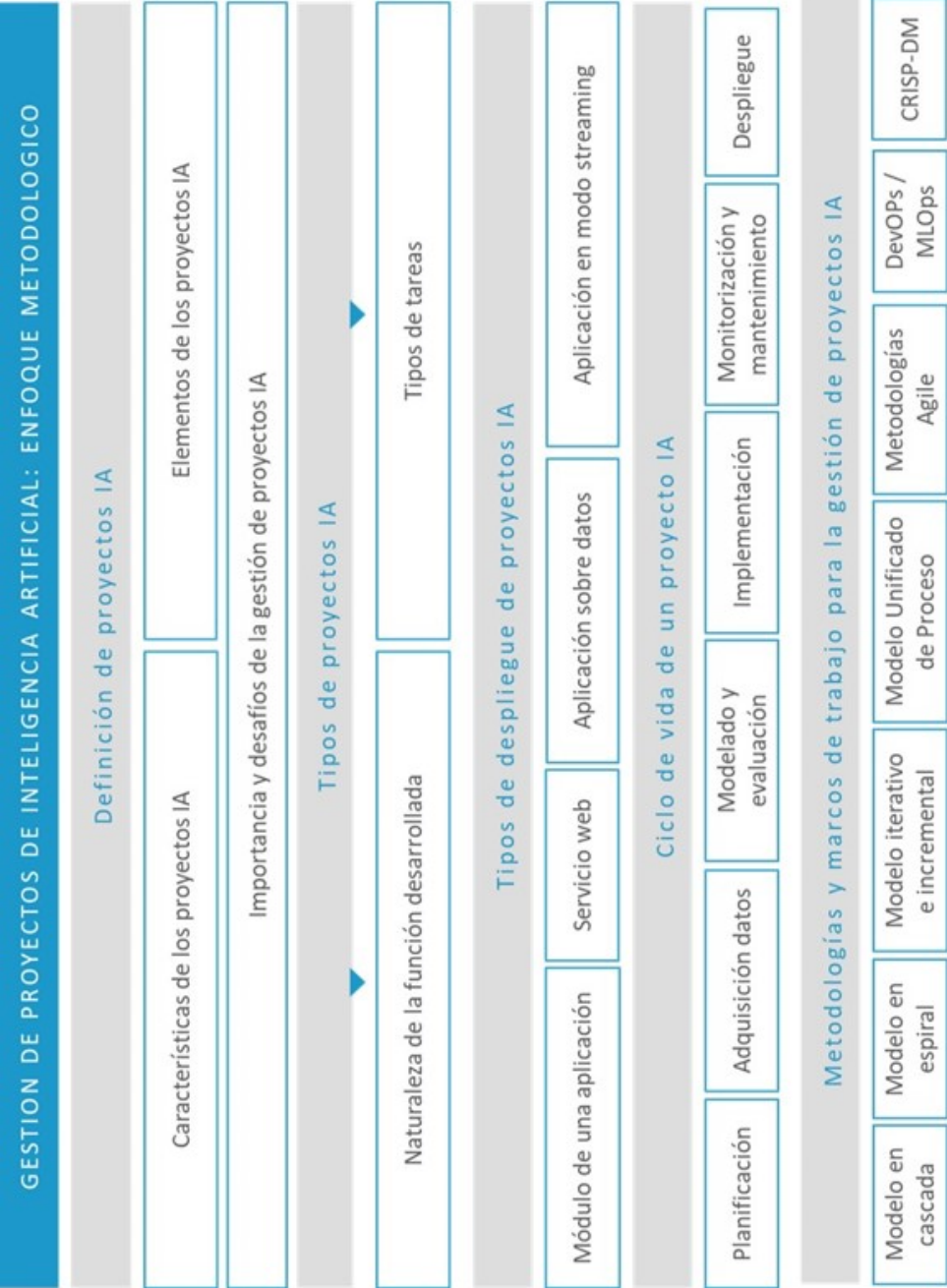
5.7. Referencias bibliográficas

A fondo

Code.org: la importancia de la programación

WIP: why limiting work in progress makes sense (Kanban)

Test



5.1. Introducción y objetivos

La **gestión de proyectos de inteligencia artificial** (IA) presenta algunas particularidades concretas. Sin embargo, todas las **metodologías** que se aplican al desarrollo ágil de *software* se pueden incorporar a esta disciplina aportando un valor notable. En esta unidad nos planteamos los **objetivos** siguientes:

- ▶ Comprender el **ciclo de vida de un proyecto de inteligencia artificial**.
- ▶ Saber diferenciar entre un **entorno de análisis** y un **entorno de producción**.
- ▶ Ser capaz de reconocer a grandes rasgos qué implica la **gestión de un proyecto** de inteligencia artificial.

5.2. Definición de proyectos de inteligencia artificial empresarial

La **inteligencia artificial (IA)** engloba una serie de **tecnologías** que han experimentado un **rápido avance** en las últimas décadas y que se espera que tengan un impacto aún más significativo tanto empresarial como social en los próximos años. En el **ámbito empresarial**, la IA puede proporcionar soluciones a problemas complejos y mejorar la eficiencia y eficacia de los procesos empresariales. En este sentido, es relativamente habitual encontrar abundantes artículos y literatura sobre qué es la IA, para qué sirve y qué beneficios puede aportar. Sin embargo, no es tan común poder encontrar literatura sobre cómo se abordan los proyectos de IA.

Un proyecto de inteligencia artificial no deja de ser un **proyecto de software**, aunque presenta particularidades derivadas de su propio apellido «inteligencia artificial». Un **proyecto de inteligencia artificial empresarial** se define como un **conjunto de actividades planificadas y coordinadas para desarrollar e implementar una solución de inteligencia artificial en una organización**. Es importante remarcar que un proyecto IA se caracteriza por el despliegue de una capacidad IA (funcionalidad concreta) y, por lo tanto, uno de los aspectos más relevantes es la **operacionalización de los modelos** (*machine learning, deep learning...*) para que puedan ser integrados en funciones y procesos de negocios.

Una capacidad IA podría ser «reconocer una imagen» o «proponer el mejor contenido en una plataforma de video en *streaming* para un determinado perfil de usuario».

Los proyectos IA heredan la gestión del ciclo de *software* habitual y también las metodologías de desarrollo e implantación (*waterfall, agile...*). Pero también conectan con la ingeniería de *software* para, por ejemplo, el desarrollo de microservicios que permitan construir APIs con funcionalidades de clasificación, predicción, *scoring...* o

con la ingeniería de datos y arquitecturas de procesamiento masivo de datos como Big Data.

Para entender mejor los proyectos de inteligencia artificial empresarial, es necesario analizar sus **principales características y elementos**. En este sentido, se pueden distinguir los siguientes subapartados:

Características de los proyectos de inteligencia artificial empresarial

Los proyectos de inteligencia artificial empresarial tienen algunas **características específicas** que los diferencian de otros tipos de proyectos empresariales. Entre las principales características se encuentran las siguientes:

- ▶ **Uso de datos.** La IA se basa en el análisis de grandes cantidades de datos para identificar patrones y relaciones. Por lo tanto, los proyectos de inteligencia artificial empresarial se centran en la gestión y procesamiento de datos.
- ▶ **Interdisciplinariedad.** Los proyectos de inteligencia artificial empresarial requieren una combinación de habilidades técnicas y empresariales para su gestión y éxito.
- ▶ **Iteratividad.** Los proyectos de inteligencia artificial empresarial implican un proceso iterativo de prueba y error en el que se ajusta el modelo de IA para lograr el mejor rendimiento posible.
- ▶ **Automatización.** La IA permite la automatización de tareas complejas y repetitivas, lo que puede mejorar la eficiencia y reducir los costos.

Elementos de los proyectos de inteligencia artificial empresarial

Los proyectos de inteligencia artificial empresarial se componen de varios elementos que deben ser gestionados adecuadamente para asegurar su éxito. Algunos de los

elementos más importantes son:

- ▶ **Objetivos del proyecto.** Los objetivos del proyecto deben estar claramente definidos y ser coherentes con la estrategia de la organización.
- ▶ **Equipo de trabajo.** Los proyectos de inteligencia artificial empresarial requieren de un equipo de trabajo interdisciplinario que incluya expertos en IA, programación, estadística, negocios y diseño.
- ▶ **Datos.** Los datos son el combustible de la IA, por lo que es esencial tener acceso a datos de calidad y suficientes para entrenar los modelos.
- ▶ **Modelos de IA.** Los modelos de IA son la base del proyecto y deben ser seleccionados y diseñados cuidadosamente para alcanzar los objetivos del proyecto.
- ▶ **Herramientas y tecnologías.** Existen muchas herramientas y tecnologías disponibles para implementar soluciones de IA, por lo que es importante seleccionar las más adecuadas para el proyecto.
- ▶ **Evaluación y validación.** Los modelos de IA deben ser evaluados y validados adecuadamente antes de ser implementados en producción.

Importancia de la gestión de proyectos de inteligencia artificial

La **capacidad de las máquinas para aprender y automatizar tareas** ha generado cambios significativos en prácticamente todos los sectores y ha llevado a la necesidad de incorporar esta tecnología en multitud de proyectos empresariales. Sin embargo, implementar proyectos de inteligencia artificial no es tarea fácil y una mala gestión de estos proyectos puede llevar a resultados deficientes o incluso fracasos.

La gestión de proyectos de inteligencia artificial (IA) es una disciplina que se enfoca

en la **planificación, organización y coordinación** de los recursos y actividades necesarias para desarrollar soluciones de IA. Esta disciplina implica la identificación de los requisitos del proyecto, la definición de los objetivos, la planificación del cronograma, la asignación de recursos y la gestión de los riesgos y problemas que surjan durante el proyecto.

La complejidad de los proyectos de IA requiere una **gestión adecuada** para garantizar la entrega de **soluciones eficientes y efectivas** que cumplan con los objetivos del proyecto. La gestión de proyectos de IA también es importante para **garantizar la calidad y el mantenimiento de las soluciones de IA** a lo largo del tiempo, ya suelen ser iterativas y evolutivas.

Existen varios **métodos y herramientas disponibles para la gestión de proyectos de IA**. Algunas de las metodologías populares incluyen **Agile y Waterfall**, mientras que algunas herramientas de gestión de proyectos populares **incluyen JIRA, Trello y Asana**. Además, existen herramientas específicas para la gestión de proyectos de IA, como **DataRobot y H2O.ai**, que pueden ayudar a acelerar el desarrollo de soluciones de IA.

Desafíos de la gestión de proyectos de inteligencia artificial

La gestión de proyectos de IA enfrenta **varios desafíos** únicos, como la necesidad de trabajar con datos y modelos complejos, la falta de recursos humanos capacitados y la necesidad de realizar pruebas y validaciones rigurosas. Además, la naturaleza iterativa y evolutiva de los proyectos de IA puede presentar desafíos adicionales para la gestión de proyectos tradicionales.

La gestión de proyectos de inteligencia artificial es fundamental para garantizar el éxito de los proyectos de esta índole. Se trata de un **proceso complejo** que implica numerosos factores a considerar, desde la elección de las herramientas y algoritmos adecuados, hasta el aseguramiento de la calidad y la gestión de los riesgos.

La planificación y gestión de proyectos de inteligencia artificial es un área en **constante evolución**, ya que las tecnologías y herramientas continúan desarrollándose a un ritmo acelerado. Por lo tanto, es esencial que los profesionales de la inteligencia artificial y la gestión de proyectos estén **actualizados** en todo momento y puedan adaptarse rápidamente a los cambios en el mercado.

La gestión de proyectos de IA también implica **consideraciones éticas y sociales importantes**. Los proyectos de IA pueden tener impactos significativos en la sociedad, incluyendo el sesgo y la discriminación en los sistemas de IA y **la privacidad y seguridad de los datos**. Por lo tanto, la gestión de proyectos de IA debe considerar estos impactos y trabajar para minimizar los riesgos asociados.

En resumen, la gestión de proyectos de inteligencia artificial es un área crítica que puede marcar la diferencia en el éxito o fracaso de los proyectos de inteligencia artificial. Al adoptar una metodología rigurosa y basada en datos, los equipos de proyecto pueden maximizar las posibilidades de éxito y asegurar que los proyectos de inteligencia artificial generen valor real para las organizaciones.

5.3. Tipos de proyectos de inteligencia artificial empresarial

Existen múltiples formas de clasificar los proyectos de inteligencia artificial. Las más comunes atienden o bien a la **naturaleza de la capacidad que se desarrolla** o bien a **l tipo de tareas necesarias para su construcción** (desarrollo, despliegue y operacionalización de la capacidad IA).

Atendiendo a la **naturaleza de la capacidad desarrollada**, existen **diferentes tipos de proyectos de inteligencia artificial empresarial** y su elección dependerá de los objetivos de la organización, los recursos disponibles y las necesidades del mercado.

Algunos de los **tipos más comunes** en función de la **naturaleza de la función** son:

- ▶ **Proyectos de clasificación:** se trata de proyectos que se enfocan en la categorización de datos según determinados criterios. Por ejemplo, un proyecto de clasificación podría analizar datos de transacciones de clientes para clasificarlos en diferentes segmentos según su comportamiento de compra.
- ▶ **Proyectos de regresión:** estos proyectos se enfocan en predecir un valor numérico con base en un conjunto de variables de entrada. Por ejemplo, un proyecto de regresión podría analizar datos de clima y tráfico para predecir la cantidad de bicicletas que serán alquiladas en una ciudad en un día determinado.
- ▶ **Proyectos de detección:** estos proyectos buscan identificar patrones o anomalías en un conjunto de datos. Por ejemplo, un proyecto de detección podría analizar registros de acceso a un sistema para identificar intentos de intrusión o actividades sospechosas.
- ▶ **Proyectos de recomendación:** este tipo de proyectos se enfocan en recomendar productos, servicios o contenidos a los usuarios con base en sus preferencias y comportamientos previos. Por ejemplo, un proyecto de recomendación podría

analizar las compras y búsquedas previas de un usuario para recomendarle productos que podrían interesarle.

- ▶ **Proyectos de procesamiento de lenguaje natural:** estos proyectos se enfocan en el análisis y generación de texto y lenguaje natural. Por ejemplo, un proyecto de procesamiento de lenguaje natural podría analizar críticas de clientes para identificar las áreas de mejora de un producto o servicio.
- ▶ **Proyectos de visión por computadora:** estos proyectos se enfocan en el análisis y procesamiento de imágenes y vídeos. Por ejemplo, un proyecto de visión por computadora podría analizar imágenes de cámaras de seguridad para identificar posibles riesgos o situaciones peligrosas.

Cada tipo de proyecto tiene sus propias características y requerimientos, y es importante elegir el tipo adecuado según los objetivos y necesidades de la organización.

Atendiendo al **tipo de tareas** necesarias para la construcción de nuevas capacidades IA y uso de las funcionalidades generadas, se pueden dar las siguientes tipologías de proyecto:

- ▶ **Proyectos de desarrollo de capacidades IA:** pueden necesitar el desarrollo de todas las actividades alineadas con la idea común de lo que se entiende por un «proyecto IA»: diseño de algoritmos, entrenamiento de modelos, validación de su precisión, etc. Estas tareas también incluyen aquellos aspectos relativos a la obtención, exploración, limpieza y preparación de los datos.
- ▶ **Proyectos de ensamblado (uso) y despliegue de las capacidades IA:** los modelos ya existen y están preentrenados y el proyecto consiste en la automatización de la captura y disponibilidad de datos y el uso eficiente de las capacidades IA existentes. Estas capacidades las pueden proporcionar proveedores de servicios tipo Google, Amazon o Microsoft, servicios suministrados por aplicaciones de mercado, o modelos propietarios desarrollados por nosotros mismos

o por proveedores externos.

5.4. Tipos de despliegue de proyectos de inteligencia artificial empresarial

El despliegue de un proyecto IA impacta en el desarrollo de este pero, sobre todo, en cómo se traslada para su operación posterior. Es decir, cómo se utilizará el modelo en el día a día por parte de la empresa.

Un **modelo preentrenado** es un archivo binario que contiene los **parámetros** obtenidos durante el proceso de entrenamiento y otra información relevante para identificar el **algoritmo** que se utiliza para aplicar el modelo. Generalmente, estos algoritmos se encuentran agrupados en **bibliotecas** que conforman un **marco de programación específico** para la IA, como *scikit-learn* o *tensorflow*. Además, estos marcos cumplen diversas **funciones**, tales como la persistencia de modelos entrenados en archivos y el suministro de funciones para interpretarlos y aplicarlos a nuevos datos.

Hay varias formas de utilizar modelos preentrenados, y depende principalmente del **tipo de integración** que se desee entre las capacidades de IA y las aplicaciones que los usarán.

Módulo de una aplicación

Este método de implementación implica la **integración del framework** de ejecución del modelo y el modelo entrenado dentro del código de la aplicación. Cuando sea necesario, la función IA se invocará desde la aplicación. De esta manera, la capacidad de IA se integra en la aplicación y se obtiene una aplicación autocontenida que se puede distribuir en dispositivos móviles o en aplicaciones web.

- **Ventajas.** La aplicación es compacta y no tiene dependencias externas. Puede funcionar sin conexión a Internet porque tiene todos los elementos necesarios para

su ejecución autónoma.

- ▶ **Desventajas.** Actualizar el modelo de IA implica actualizar toda la aplicación. El *framework* debe ser compatible con la plataforma de ejecución de la aplicación.

Servicio web

Este método de implementación supone **exponer el modelo IA como un servicio web**. Este servicio podrá utilizarse desde diversas aplicaciones y es el modelo habitual empleado en los servicios proporcionados por las grandes tecnológicas como Google, Amazon o Microsoft.

- ▶ **Ventajas.** La aplicación que utiliza el servicio es independiente de este, por lo que se asegura su independencia. Además, la aplicación no requiere conocer cómo está implementado el servicio. Solo debe invocarlo y pasarle los parámetros necesarios a la función.

Asimismo, la actualización de los servicios no impacta en la aplicación y el proveedor puede monitorizar fácilmente su rendimiento.

- ▶ **Desventajas.** El servicio se aloja en una plataforma que necesita tener correctamente dimensionados los recursos asignados a este, atendiendo a parámetros como capacidad de computación requerida, memoria, concurrencia de usuarios, etc. Asimismo, la aplicación necesita de un acceso a Internet para utilizar los servicios.

Aplicación en modo *batch* (lotes) sobre los datos

En este caso el modelo IA **se aplica a un conjunto de datos masivo**. Se podrían utilizar los servicios *webm*, pero sería necesario compartir los datos registro a registro lo que sería poco eficiente. Por ello, la opción más habitual es cargar todos los datos en un entorno **Big Data** si son datos masivos y ejecutar el **modelo en modo batch**. Ello permite el procesado de los datos **en lotes** en lugar de procesar cada registro de forma individual.

- ▶ **Ventajas.** En el aprendizaje automático, el modo *batch* se utiliza para entrenar modelos con un gran volumen de datos, lo que reduce los tiempos de entrenamiento y mejora la eficiencia del modelo.
- ▶ **Desventajas.** Este modelo requiere de una arquitectura de computación adecuada a la ejecución de los modelos y el tratamiento masivo de datos.

Aplicación en modo *streaming* de datos

A diferencia del tipo de despliegue anterior, en este caso los datos no están almacenados en bloques estáticos sino que su origen es un **flujo de datos** cuya **transferencia se produce de forma constante y en tiempo real**. Este tipo de datos es característico en procesos de **monitorización continua** de la actividad en redes sociales, en el análisis de transacciones financieras o en la monitorización de sensores IoT (*Internet of Things*).

En este escenario los modelos se despliegan en arquitecturas que están especializadas para dicho fin y se pueden utilizar *frameworks* como Apache Spark, capaz de procesar datos en tiempo real con el módulo de procesamiento Spark Streaming, o Tibco StreamBase, una plataforma capaz de procesar datos de múltiples fuentes en tiempo real. StreamBase también cuenta con herramientas de

análisis en tiempo real que permiten a los usuarios detectar patrones y tendencias en los datos en tiempo real, y tomar decisiones en consecuencia.

Ventajas. Este modelo permite trabajar con datos que requieren una mínima latencia en su tratamiento. La latencia es el tiempo que tarda el sistema en responder a una solicitud o en completar una operación.

Desventajas. Este modelo requiere de una arquitectura de computación adecuada a la ejecución de los modelos en tiempo real y su capacidad en el procesamiento de los datos necesita de *frameworks* especializados (Spark Streaming).

5.5. Ciclo de vida de un proyecto de inteligencia artificial empresarial

Un proyecto de inteligencia artificial sigue un ciclo de vida que se compone de una serie de fases que van desde la concepción del proyecto hasta la implantación del sistema de IA. La gestión adecuada del ciclo de vida es esencial para el éxito de un proyecto de IA. En esta sección, se describirán las diferentes fases del ciclo de vida de un proyecto de inteligencia artificial empresarial.

Fase de planificación

La **fase de planificación** es la **primera etapa del ciclo de vida de un proyecto de inteligencia artificial empresarial**. En esta etapa, se identifican los **objetivos** del proyecto, se define el alcance del proyecto, se establecen los recursos necesarios, se determinan los riesgos y se elabora un **plan de proyecto** detallado. La planificación es una etapa crucial, ya que la mayoría de los problemas que surgen durante el proyecto pueden ser evitados mediante una buena planificación.

Fase de adquisición de datos

En esta fase, **se recopilan los datos necesarios** para entrenar y validar el modelo de IA. Esta etapa puede ser larga y costosa, ya que la **calidad de los datos** es un factor crítico en el éxito del proyecto de IA. Por lo tanto, es necesario prestar atención a la calidad de los datos y garantizar su adecuación para el uso previsto.

Fase de modelado y evaluación

En esta fase, **se construyen y se prueban varios modelos** de inteligencia artificial. La elección del modelo adecuado dependerá del problema que se está tratando de resolver y de los datos disponibles. Es importante tener en cuenta que los modelos de IA no son infalibles y pueden tener sesgos y errores. Por lo tanto, es importante evaluar el rendimiento del modelo y garantizar que sea justo e imparcial.

Fase de implementación

Una vez que se haya seleccionado el modelo adecuado, se procede a su **implementación**. En esta fase, **se desarrolla y se integra** el modelo de IA en el sistema existente. Es importante garantizar que el modelo funcione correctamente y se integre de manera efectiva con el sistema existente.

Fase de monitorización y mantenimiento

La **monitorización y el mantenimiento** son una parte esencial del ciclo de vida de un proyecto de inteligencia artificial empresarial. En esta fase, se monitoriza el rendimiento del modelo y se lleva a cabo el mantenimiento necesario. Es importante realizar ajustes y mejoras en el modelo para garantizar su rendimiento y eficacia a largo plazo.

Fase de despliegue

La fase de despliegue es la **última fase** del ciclo de vida de un proyecto de inteligencia artificial empresarial. En esta fase, **se pone el sistema en producción y se lanza al mercado**. Es importante garantizar que el sistema se integre con éxito con el sistema existente y que funcione correctamente en condiciones de uso reales.

En conclusión, la **gestión adecuada del ciclo de vida** de un proyecto de inteligencia artificial empresarial es esencial para el éxito del proyecto. Cada fase del ciclo de vida tiene su propia importancia y debe ser gestionada adecuadamente para garantizar el éxito general del proyecto.

5.6. Metodologías y marcos de trabajo para la gestión de proyectos de inteligencia artificial

Un **proyecto** es un **esfuerzo temporal emprendido para crear un único producto o servicio**. Se debe distinguir claramente los proyectos de las operaciones. Hacer una nueva carretera es un proyecto. Mantener una carretera en las condiciones adecuadas para que sea transitable forma parte de las operaciones de mantenimiento. Las operaciones finalizan cuando finaliza la vida del producto o servicio en cuestión. El proyecto finaliza cuando se ha creado el producto o servicio y, normalmente, entregado al cliente.

La **gestión de proyectos** es la aplicación de **conocimientos, habilidades, herramientas y técnicas** para proyectar actividades destinadas a satisfacer las necesidades y expectativas de los **interesados (stakeholders)** de un proyecto. Entre estos interesados podemos encontrar además de los propios clientes o usuarios, a otros actores como: directivos, servicios jurídicos de la compañía, departamento de seguridad, departamento de marketing, entidades reguladoras, legisladores, gobiernos, grupos de presión, etc.

¿Qué **tipo de producto o servicio** se espera de un proyecto de inteligencia artificial? Quizá estemos pensando en un coche autónomo, un robot, etc. Todos esos productos tienen algo en común: **software**. El *software* es el producto más habitual que un proyecto de estas características genera. Por supuesto, en muchas ocasiones el *software* viene acompañado de *hardware* o algún tipo de componente mecánico que ejecuta las órdenes emitidas por el *software*. Dado el ámbito y alcance de esta asignatura, nos centraremos en el **software**, que será el vehículo en el que se expresarán nuestros algoritmos y modelos.

La gestión de proyectos de IA guarda similitudes con los **ámbitos tradicionales de desarrollo de software**. De hecho, el producto final en gran medida es un modelo

basado en el *software*.

Sin embargo, presenta **desafíos particulares** que deben ser abordados adecuadamente para asegurar su éxito, especialmente en lo relativo a la gestión y tratamiento de los datos y la naturaleza de su **iteratividad**.

Para ambos contextos, **desarrollo de *software* y tratamiento de datos**, existen **metodologías y marcos de trabajo** adecuados que ayudan a planificar, ejecutar y controlar los proyectos de IA de manera efectiva y que en definitiva, nos permiten gestionar todo el ciclo de vida de un proyecto IA.

Una **metodología** es una serie de pasos y técnicas para lograr un objetivo, mientras que un **marco de trabajo** es un conjunto de principios, prácticas y directrices para abordar un problema o una tarea de manera general. Las metodologías se enfocan en cómo realizar el proyecto, mientras que los marcos de trabajo proporcionan un entorno general para la gestión del proyecto. En el contexto de la gestión de proyectos de IA, se pueden utilizar tanto metodologías como marcos de trabajo.

Metodologías

Las metodologías para la gestión de proyectos son un **conjunto de procesos, procedimientos y herramientas que permiten la planificación, el control y la supervisión de proyectos *software***. La utilización de metodologías en la gestión de proyectos de IA es fundamental para garantizar su éxito, ya que este tipo de proyectos son complejos y requieren una gestión rigurosa.

Las **metodologías para la gestión de proyectos de IA** más utilizadas son las siguientes:

Modelo en cascada (*waterfall*)

Las metodologías de desarrollo de *software* han evolucionado con el objetivo de **mitigar los riesgos y conseguir una mejor adaptación** del producto final a lo que

realmente se espera de él.

En los inicios, el modelo en cascada (*waterfall*) era la principal opción. Este ciclo de vida determinaba claramente una serie de fases que, de una forma u otra, se han mantenido más o menos comunes en la mayoría de los modelos posteriores.

El **modelo *waterfall*** es un enfoque secuencial y lineal utilizado en proyectos **software**. Esta metodología se divide en **fases** (Figura 1), y cada fase debe completarse antes de que se inicie la siguiente.

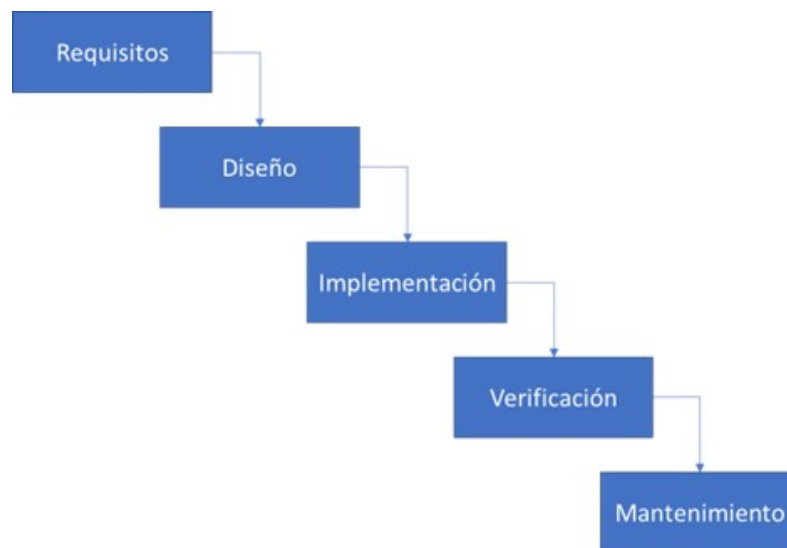


Figura 1. Modelo en cascada.

Tal cual se definió, este modelo presentaba **serios problemas**, ya que la fase de verificación tenía lugar al final del proceso. En caso de detectar errores en la implementación, en el diseño o en la propia toma de requisitos (que determina los objetivos a conseguir) el camino a desandar implica unos costes más que considerables en tiempo y dinero.

La metodología *waterfall* es útil para proyectos de IA donde los requisitos son claros y bien definidos desde el principio. Sin embargo, puede ser difícil adaptar la metodología *waterfall* a proyectos de IA donde los requisitos cambian con frecuencia

Modelo en espiral

El modelo del **ciclo de vida en espiral** (Figura 2) desarrollado por Boehm en 1986 en su trabajo *A Spiral Model of Software Development and Enhancement* introdujo elementos importantes al incorporar una **completa gestión del riesgo y control de la calidad**. El desarrollo iterativo permite avanzar gradualmente hacia el producto final, mitigando los riesgos existentes y adelantando la detección de problemas y sus soluciones lo antes posible.

En cada iteración se tienen en cuenta los **objetivos** (qué necesidad tiene que cubrir el producto), las **alternativas** (formas de conseguir los objetivos, considerando el riesgo asumido en cada alternativa) así como el **desarrollo y verificación del software** (programación y pruebas). Si el entregable de la iteración no es el adecuado, se planificarán los siguientes pasos para incluir las **mejoras** y funcionalidades necesarias, comenzando una nueva iteración.

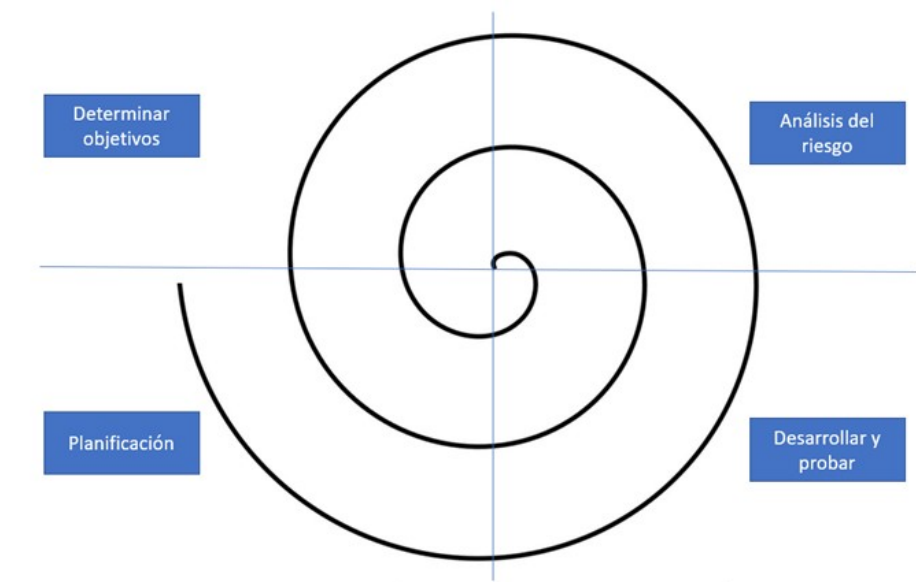


Figura 2. Modelo en espiral.

Este modelo **da participación continua al cliente**, evaluando el producto de forma constante. De esta forma, **los errores se pueden detectar de forma anticipada**.

Las distintas iteraciones permiten **generar diferentes prototipos** del producto final. Por lo que este modelo exige equipos muy preparados y un seguimiento continuo y detallado.

El modelo en espiral está orientado a proyectos grandes y complejos.

Modelo iterativo e incremental

Este modelo es la base de los **modelos más usados** hoy en día. Con este modelo se persigue que el cliente reciba de forma continua algo de valor. Según se avanza en el número de iteraciones el producto final incorpora más funcionalidades y se convierte en un producto más elaborado y complejo. Este modelo permite acometer el producto con base en pequeños entregables que hacen más realizable la tarea.

El modelo iterativo e incremental (Figura 3) es, por lo tanto, un **conjunto de tareas que se agrupan en pequeñas iteraciones que se repiten**. Cada iteración comienza con la etapa de análisis y finaliza con la aprobación del entregable. Y en cada iteración se aborda un requerimiento que debe tener un desarrollo completo en una única iteración.

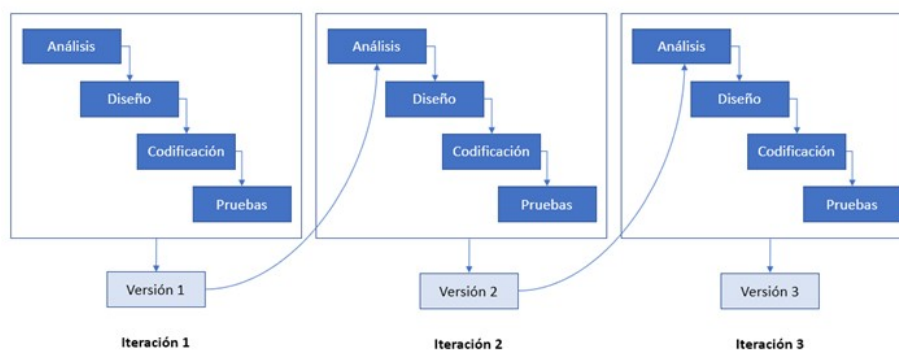


Figura 3. Modelo iterativo e incremental

Este modelo es una parte esencial de los *frameworks* de desarrollo rápido de *software* como **Extreme Programming**.

Modelo unificado de proceso

El principal problema de los modelos iterativos es que **la carga de trabajo no es proporcional** en todas las iteraciones. En las primeras fases de un proyecto, la toma de requerimientos y el análisis suelen tener mayor peso que el desarrollo. E incluso, las pruebas en ocasiones tienen al principio una presencia testimonial. Esta proporción puede llegar a invertirse en iteraciones más avanzadas. Esta situación genera en la práctica que se usen **modelos ligeramente adaptados** para equilibrar las cargas de trabajo en las diferentes iteraciones.

De esta situación surge el denominado **modelo unificado de proceso**. Este es el modelo culmen de la ingeniería del *software* clásica y también se trata de un marco de desarrollo iterativo e incremental. Está compuesto por **cuatro fases: inicio, elaboración, construcción y transición**. Cada una de estas fases a su vez tiene diversas iteraciones y que recuerdan en gran medida a las cuatro fases que formaban cada iteración del modelo iterativo e **incremental (análisis, diseño, codificación y pruebas)**.

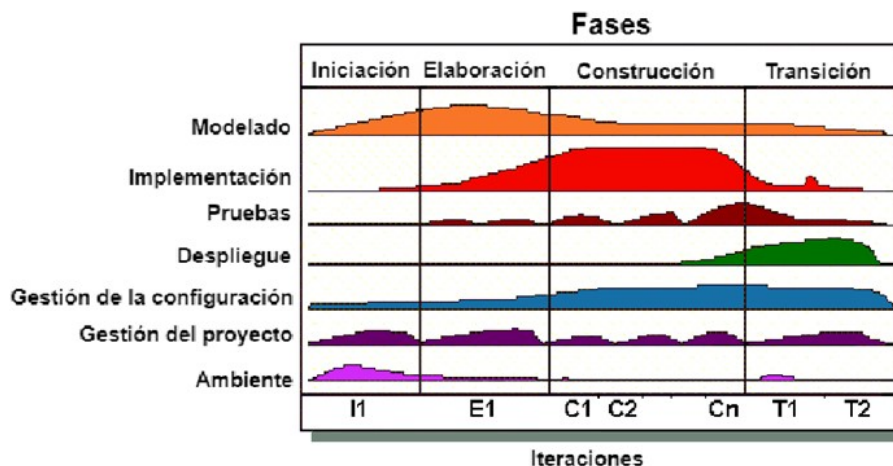


Figura 4. Modelo iterativo e incremental. Fuente: DOI: 10.17993/3ctecno/2020.v9n3e35.17-45

Su **principal ventaja** es que es muy **verificable** y está muy **documentado**, lo que le hace ideal para proyectos de alto riesgo y que necesitan una verificación y un control

de calidad exhaustivo. Su principal **inconveniente** es su **excesiva sobrecarga** para llevarla a cabo. Actualmente este modelo se usa para entornos muy concretos como proyectos de alto riesgo, para las grandes administraciones, para seguridad del estado, sistemas críticos, etc.

Metodología Agile

La **metodología Agile** aplica un **enfoque iterativo y colaborativo**. Esta metodología se divide en **ciclos de trabajo**, y cada ciclo produce un resultado parcial que se puede revisar y ajustar en función de los requisitos. La metodología Agile es útil para proyectos de IA donde los requisitos no son claros desde el principio y cambian con frecuencia. Esta metodología permite una **mayor flexibilidad y adaptabilidad** que la metodología Waterfall.

Las metodologías ágiles, por lo tanto, **permiten adaptar la forma de trabajo a las condiciones del proyecto**, consiguiendo **flexibilidad e inmediatez en la respuesta** para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno. Se persigue de esta forma **reducir costes e incrementar la productividad**, al avisar rápidamente de los errores y problemas encontrados. Este proceso surge como un rebote frente a los modelos tradicionales donde cada vez se tendía a introducir más y más documentación y más y más procesos que dificultaban y alargaban enormemente los proyectos.

Las metodologías ágiles han desplazado a las tradicionales en muchos entornos de desarrollo, pero eso no significa que estas no tengan cabida. Las metodologías tradicionales se siguen utilizando en entornos donde se necesita un control exhaustivo de los procesos y donde el resultado del *software* es crítico, como por ejemplo en *software* para el ejército, para sistemas de control críticos o para grandes proyectos en la administración.

Las metodologías ágiles suelen ser útiles en proyectos **donde los requisitos no**

están claros o son muy cambiantes. En este tipo de proyectos, el prototipado y la investigación continua marcan el camino a seguir. Los modelos ágiles facilitan la salida con un producto con la funcionalidad mínima y que se puede ir mejorando e incrementando.

Entre las metodologías ágiles más aplicadas hoy en día tenemos **Scrum, Kanban, Scrumban y Extreme Programming (XP)**. Se comentará muy brevemente Scrum y Kanban por ser metodologías de uso habitual en los proyectos que nos ocupan.

Metodología Ágile: SCRUM

Scrum (Figura 5) se caracteriza por la planificación en base a iteraciones (*sprints*), con una duración de entre 2 y 4 semanas. Cada *sprint* genera un entregable para el cliente. Las necesidades o requisitos del cliente se anotan en una «pila» de producto de forma priorizada denominada **backlog**.

La lista de los trabajos que debe realizar el equipo durante el *sprint* se añade a la pila del *sprint* y se selecciona una persona o personas responsables de su ejecución. Así el proyecto se descompone en unidades más pequeñas y gestionables.

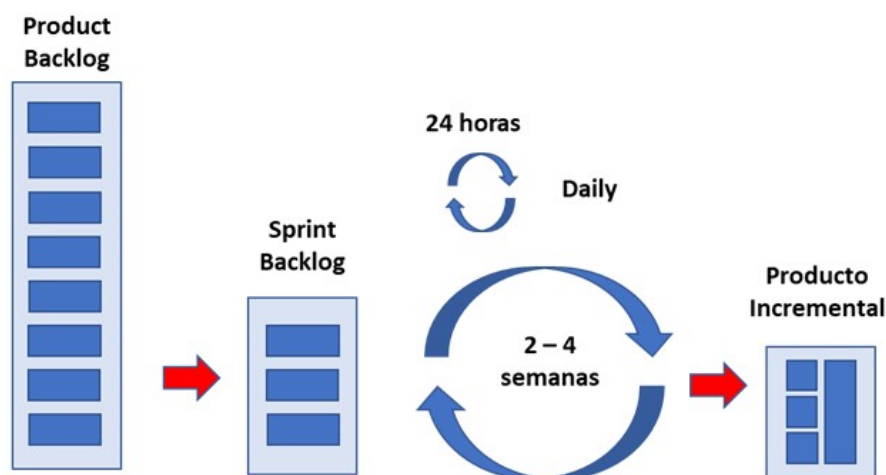


Figura 5. Metodología Scrum.

- ▶ **Product backlog (o traducido al español, pila de producto):** listado completo de funcionalidades ordenadas según el valor de negocio que establece el dueño del producto y que trata de cubrir todas las funcionalidades necesarias.
- ▶ **Sprint backlog:** listado de tareas que el equipo elabora en la reunión de planificación de la iteración como plan para completar los objetivos seleccionados para esa iteración y que se compromete a demostrar al cliente al final de esta. Cabe destacar que la duración de una iteración puede variar de 2 a 4 semanas, aunque, como ya se ha hablado anteriormente, se recomienda que sea lo menos posible.
- ▶ **Daily:** reunión diaria que se hace con el equipo para sincronizar las tareas de las que cada uno es responsable.
- ▶ **Producto incremental:** producto que el equipo se ha comprometido a demostrar al final de la iteración con el fin de comprobar el progreso y las funcionalidades nuevas que se han integrado en este período de tiempo. Este producto se llama incremental porque el número de funcionalidades va aumentando sprint a *sprint*.

A las tareas del **backlog** se les asigna unos **puntos de esfuerzo o puntos de historia**. Estos puntos que se van colocando de forma manual por los propios encargados de realizarlas denotan el esfuerzo que considera el responsable que va a

tener llevarlas a cabo. Este esfuerzo se va recalculando en función del número de tareas que el responsable es capaz de resolver. Por ejemplo, en una semana un ingeniero ha podido resolver tareas por valor de 5 puntos de esfuerzo en media. De forma que si se le asignan tareas que él estima tendrían 10 puntos de esfuerzo debería tardar dos semanas en realizar. Si esto no es así y las realiza en una semana, incrementará el número de puntos de esfuerzo que es capaz de resolver. De esta forma se regula de forma dinámica la gestión del trabajo y el esfuerzo y con unas cuantas iteraciones, el equipo es capaz de planificar mucho mejor cuándo tendrá la próxima iteración y con qué contenido.

La **figura del Scrum Máster** es una figura capital dentro de *scrum*. Sus funciones son la de asegurar que se cumplen las buenas prácticas y valores descritos en el modelo Scrum.

El Scrum Máster tendría una figura similar a la de un *coach*/mentor que acompañará al equipo durante todo el desarrollo del proyecto y asegurará que se cumplan las buenas prácticas, actuando como un facilitador y solucionador de problemas.

Entre otras **funciones**, el Scrum Máster debe:

- ▶ Asesorar y formar a los diferentes miembros para trabajar de forma autoorganizada y con responsabilidad de equipo.
- ▶ Moderar las reuniones.
- ▶ Resolver impedimentos que en el *sprint* pueden entorpecer la ejecución de las tareas.
- ▶ Gestionar las dinámicas de grupo en el equipo.
- ▶ Encargarse de la configuración, diseño y mejora continua de las prácticas de *scrum* en la organización.

Las **características básicas** de los proyectos gestionados con metodologías ágiles son las siguientes:

- ▶ Los equipos son **autoorganizados** y no existen roles especializados. El equipo posee total autonomía y libertad para la toma de decisiones.
- ▶ Las fases de desarrollo no existen como tales, sino que se **desarrollan tareas en función de necesidades** que van cambiando a lo largo de la vida del proyecto.
- ▶ Todas las iteraciones tienen un **período de tiempo fijado**.
- ▶ Como en cualquier proyecto, tienen que existir ciertos **puntos de control** para realizar un seguimiento adecuado. Este momento suele ser el final de cada *sprint*, punto en el que el equipo puede echar la vista hacia atrás y ver todo lo que se ha hecho.
- ▶ La **difusión y la transferencia** del conocimiento tienen que ser algo común, ya que siempre existe la posibilidad de una alta rotación de los miembros del equipo entre diferentes proyectos que la empresa pueda tener, de forma que el equipo tiene que intentar ser lo más interdisciplinar posible.

Metodología Agile: KANBAN

Kanban nace en Japón alrededor del año 1950 como una propuesta para manejar de forma eficiente el flujo de materiales en una línea de producción. Algunos de sus componentes principales son:

- ▶ **La visualización del flujo de trabajo:** dividiendo el trabajo en bloques y escribiendo cada elemento en una tarjeta para después ponerlo en la tabla o pizarra creada para ello.
- ▶ **La limitación del trabajo en curso (*Work In Progress* – *WIP*)** asignando límites concretos del número de elementos que pueden estar en progreso en cada estado del flujo trabajo.
- ▶ **La medición del denominado tiempo de ciclo** que es el tiempo medio para completar un elemento y también ayuda a optimizar el proceso para conseguir que este tiempo sea lo más pequeño y predecible posible.

A continuación, se refleja un ejemplo de tablero Kanban (Figura 6).

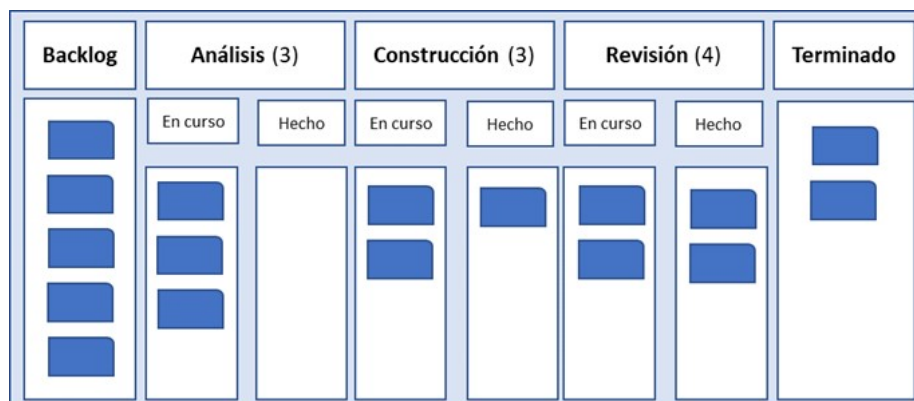


Figura 6. Tablero Kanban.

El **control visual del proceso** y la **limitación del número de trabajos** en curso contribuye a un entendimiento más completo del estado del proyecto y permite que las tareas avancen de forma eficiente.

Es muy común ver usar **Kanban y Scrum simultáneamente** en el mismo proyecto.

Existen multitud de herramientas para realizar Kanban y Scrum, pero por citar las

más usadas puedes echar un vistazo a **Trello**, un gestor de tareas de tipo Kanban gratuito, o **Pivotal Tracker** que usa la metodología Scrum y el concepto de historia y los puntos de esfuerzo que es capaz de calcular y actualizar de forma dinámica. Aunque hay otras herramientas como Active Collab, Taiga, Kanbanize, Atlassian Jira, iceScrum , Scrumbler, etc.

DevOps / MLOps

DevOps es una metodología que **combina el desarrollo de software (Dev) con las operaciones de TI (Ops)**. Esta metodología se centra en la **colaboración entre desarrolladores y operadores** para acelerar la entrega de *software* y mejorar la calidad de este. En proyectos de IA, la metodología DevOps se utiliza para asegurar que el modelo de IA **se integre correctamente en los sistemas de producción** y se implemente de manera segura y eficiente.

Basándose en los principios de DevOps tenemos **MLOps (Machine Learning Operations)**. Es una metodología que se enfoca en la implementación y el despliegue de modelos de aprendizaje automático en producción. Proporciona herramientas y técnicas para la gestión de proyectos de inteligencia artificial en producción, incluyendo la monitorización, la evaluación y el mantenimiento de modelos de aprendizaje automático.

En el mundo del aprendizaje automático, es común tener **equipos de científicos de datos** que están a cargo de construir modelos y encontrar patrones en los datos. Sin embargo, una vez que se ha construido un modelo, hay que implementarlo en un entorno de producción para que pueda ser utilizado en la **vida real**.

Es aquí donde entran en juego las **prácticas y herramientas de MLOps**, que ayudan a gestionar todo el ciclo de vida de los modelos de aprendizaje automático, desde el desarrollo hasta la implementación y el mantenimiento continuo.

El objetivo principal de MLOps es **garantizar que los modelos de aprendizaje**

automático sean precisos, escalables y confiables en producción. Para lograrlo, se utilizan varias **técnicas y herramientas**, como:

- ▶ **Automatización:** los procesos manuales pueden ser lentos y propensos a errores. La automatización de tareas como el entrenamiento del modelo, la evaluación de la calidad de los datos y la implementación del modelo en producción puede ayudar a ahorrar tiempo y minimizar errores.
- ▶ **Integración continua y entrega continua (CI/CD):** estas prácticas son comunes en el desarrollo de *software* y se han adaptado para el desarrollo de modelos de aprendizaje automático. CI/CD se utiliza para garantizar que los cambios en el código y los datos se prueben y validen automáticamente antes de ser implementados en producción.
- ▶ **Monitorización:** los modelos de aprendizaje automático pueden degradarse con el tiempo si los datos cambian o si el modelo se utiliza de manera incorrecta. La monitorización continua del modelo puede ayudar a detectar problemas y garantizar que el modelo siga siendo preciso y relevante.
- ▶ **Control de versiones:** los modelos de aprendizaje automático suelen ser complejos y pueden estar compuestos por varios componentes. El control de versiones puede

ayudar a gestionar los cambios en estos componentes y a garantizar que se puedan revertir los cambios si es necesario.

- ▶ **Gestión de la configuración:** la configuración de los modelos de aprendizaje automático puede ser complicada y variar según el entorno. La gestión de la configuración puede ayudar a gestionar la configuración de los modelos de forma consistente y a garantizar que se utilicen las mismas configuraciones en todos los entornos.
- ▶ **Gestión de los datos:** los modelos de aprendizaje automático dependen de los datos de entrenamiento para ser precisos. La gestión de los datos puede ayudar a garantizar que los datos se limpien y procesen correctamente antes de ser utilizados para entrenar el modelo.

Existen varias **herramientas y plataformas** que se utilizan en el ámbito MLOps:

- ▶ **Plataformas de gestión de modelos:** son herramientas que permiten a los equipos de ciencia de datos construir, entrenar, validar y desplegar modelos de aprendizaje automático de manera eficiente. Algunas de las plataformas más populares incluyen: AWS SageMaker, Google Cloud AI Platform, Microsoft Azure Machine Learning, Databricks, Kubeflow, entre otras.

- ▶ **Herramientas de automatización de flujo de trabajo:** son herramientas que permiten automatizar tareas repetitivas y tediosas del ciclo de vida de los modelos de aprendizaje automático. Ejemplos de herramientas populares incluyen: Apache Airflow, Prefect, Dagster, etc.
- ▶ **Herramientas de control de versiones:** son herramientas que permiten gestionar el código fuente, las configuraciones y los datos de los modelos de aprendizaje automático. Algunas de las herramientas de control de versiones más populares incluyen: Git, GitHub, GitLab, Bitbucket, etc.
- ▶ **Herramientas de gestión de configuración:** son herramientas que permiten gestionar la configuración de los modelos de aprendizaje automático, incluyendo las configuraciones de infraestructura, configuraciones de hiper-parámetros, etc. Algunas herramientas populares incluyen: Ansible, Puppet, Chef, Terraform, etc.
- ▶ **Herramientas de monitorización y registro de modelos:** son herramientas que permiten monitorizar el rendimiento de los modelos de aprendizaje automático en producción, así como registrar información y métricas importantes. Ejemplos de herramientas populares incluyen: Prometheus, Grafana, Datadog, etc.

- ▶ **Herramientas de orquestación y gestión de contenedores:** son herramientas que permiten gestionar contenedores y orquestar la infraestructura necesaria para ejecutar modelos de aprendizaje automático de manera escalable y eficiente. Algunas de las herramientas de orquestación más populares incluyen: Kubernetes, Docker Swarm, Amazon ECS, etc.

Marco de trabajo CRISP-DM

Las metodologías y procedimientos explicados en los apartados anteriores sirven para cualquier tipo de proyecto de *software*, desde un entorno virtual hasta un *software* de gestión contable. Sin embargo, la tipología de proyectos que tratamos en esta asignatura tiene una serie de **particularidades** que es preciso reflejar.

En primer lugar, porque la inteligencia artificial como ciencia que es nos obliga a seguir el **método científico**. En segundo lugar, los proyectos IA se caracterizan por un **uso intensivo de los datos**. Además, una diferencia esencial cuando hablamos de modelos IA frente a un proyecto tradicional de *software* es que una vez puesto el modelo a disposición de los usuarios puede ser necesario **evaluar su comportamiento de forma periódica y ajustar los parámetros** de este. Los usuarios del modelo pueden cambiar sus hábitos y la propia solución puede inducir a comportamientos no previstos en un inicio.

A diferencia de las metodologías, los **marcos de trabajo** son enfoques más generales que no proporcionan una guía detallada para la planificación, ejecución y control de proyectos de IA. En cambio, los marcos de trabajo proporcionan un conjunto de **principios y directrices generales** que se pueden adaptar a los proyectos de IA según sea necesario.

El más importante es **CRISP-DM** (*Cross Industry Standard Process for Data Mining*)

reflejado en la Figura 7. CRISP-DM es un marco de trabajo ampliamente utilizado para **proyectos de minería de datos y análisis de datos**. Consta de seis fases que se deben seguir para llevar a cabo un proyecto de minería de datos: comprensión del problema, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue. Aunque CRISP-DM fue diseñado para proyectos de minería de datos, su estructura y metodología se pueden adaptar a proyectos de inteligencia artificial.

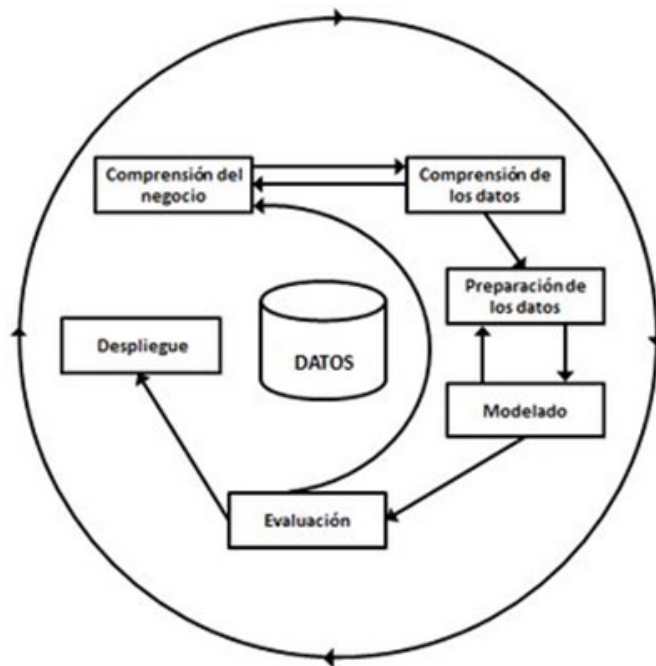


Figura 7. Marco de trabajo CRISP-DM.

Es preciso mencionar que el proceso CRISP-DM comienza con una **etapa inicial** de gran relevancia que es el **entendimiento del negocio**, de forma que podamos detectar **necesidades y oportunidades susceptibles** de ser resueltas con un enfoque innovador.

Profundizando en el **ecosistema de la ciencia de datos**, Rachel Schutt en su libro *Doing Data Science* propone el esquema específico de la Figura 8:

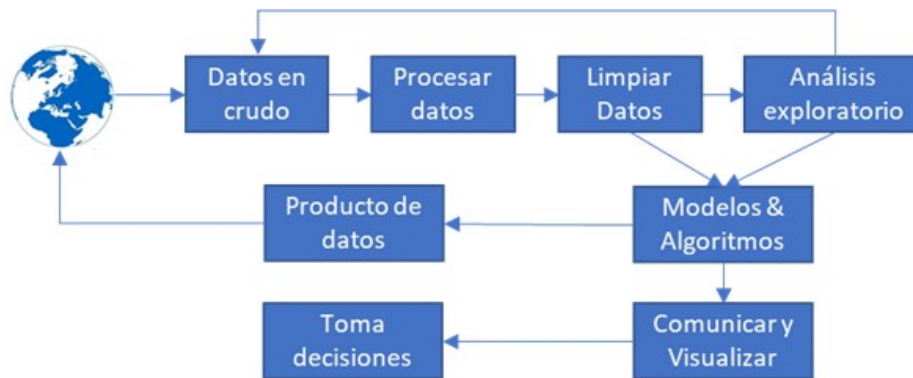


Figura 8. Proceso Data Science.

Dicho esquema detalla la parte de tratamiento de datos tan propia del ecosistema Big Data e incorpora nuevas relaciones entre las distintas fases. La generación de modelos que sean capaces de obtener patrones de los datos y realizar predicciones se convierte en parte central del esquema. Así como la visualización y comunicación de resultados adquiere especial relevancia frente a otros modelos más generalistas.

5.7. Referencias bibliográficas

Boehm, B. (1986). A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes, *ACM*, 11(4):14-24.

O'Neil, C., Schutt, R. (2013). *Doing Data Science: Straight Talk from the Frontline*. O'Reilly.

Solé, J. M. (2020). *Proyectos de Inteligencia Artificial*. Amazon [distribuidor].

Code.org: la importancia de la programación

En este vídeo se explica la importancia de la programación informática y por qué debería convertirse en un conocimiento transversal en la escuela.

Accede al vídeo a través del aula virtual o desde la siguiente dirección web:
<https://www.youtube.com/watch?v=Qhn48RekQeg>

WIP: why limiting work in progress makes sense (Kanban)

Explicación del concepto *Work in progress* (WIP) de la metodología Kanban y sus implicaciones para la gestión del proyecto.

Accede al vídeo a través del aula virtual o desde la siguiente dirección web:
<https://www.youtube.com/watch?v=W92wG-HW8gg>

1. ¿Cuáles de las siguientes frases no refleja una característica de un proyecto?

Señala todas las que procedan:

- A. Es un esfuerzo temporal.
- B. Incluye la fase de mantenimiento del producto.
- C. Pretende crear un único producto o servicio.
- D. Debe tener en cuenta a los interesados del proyecto.

2. El ciclo de vida del *software*:

- A. Abarca la evolución desde el primer prototipo del producto hasta su retirada.
- B. Abarca la evolución desde la concepción del producto hasta su retirada.
- C. Abarca la evolución desde la concepción del producto hasta su puesta en producción.
- D. Abarca la evolución desde que se consiguen los fondos necesarios hasta que no se dispone de presupuesto para continuar con el servicio.

3. El ciclo de vida en cascada:

- A. Permite gestionar de forma eficaz todo tipo de proyectos.
- B. Es la metodología más utilizada en el desarrollo de proyectos IA.
- C. El diseño se hace de forma iterativa.
- D. No permite una detección temprana de errores y fallos en la toma de requisitos.

4. El ciclo de vida iterativo e incremental:
 - A. Permite entregar «algo» de valor al cliente o usuario final de forma iterativa.
 - B. No permite entregar «algo» de valor al cliente o usuario final de forma iterativa.
 - C. Incluye una única fase de análisis y en la primera iteración.
 - D. Incluye una única fase de diseño y en la primera iteración.

5. Una metodología:
 - A. Se limita a especificar cómo se realizará el diseño del producto.
 - B. No debe contemplar elementos de gestión del equipo.
 - C. Define un conjunto de procedimientos que permiten producir y mantener un producto *software*.
 - D. Estándar es la PMP (*Product Management Protocol*).

6. La gestión ágil de proyectos:
 - A. Permite ir más rápido ya que no obliga a documentar.
 - B. Es especialmente útil donde los requisitos son cambiantes y no están definidos con precisión.
 - C. Se basa en poner en una pizarra las tareas pendientes.
 - D. Sirve para todo tipo de proyectos en todos los ámbitos.

7. El Data Science Process ¿en qué se diferencia fundamentalmente de los modelos ágiles?
 - A. Es más corto y con menos fases que los proyectos normales.
 - B. No necesita mantenimiento.
 - C. El modelo una vez puesto en marcha probablemente necesite ser reevaluado.
 - D. Es más lineal debido al poco riesgo en los requisitos de estos proyectos.

8. El despliegue de un módulo IA dentro de una aplicación:
- A. Implica la integración del *framework* de ejecución del modelo y el propio modelo entrenado dentro del código de la aplicación.
 - B. Expone el modelo IA como un servicio web.
 - C. Aplica el modelo a un conjunto de datos procesándolo por lotes.
 - D. Exige la ingesta de datos en el propio modelo.
9. La clasificación de los tipos de proyectos IA puede atender a diferentes criterios. Algunos son los siguientes:
- A. Proyectos de clasificación y anonimización
 - B. Proyectos de desarrollo y validación.
 - C. Proyectos de aplicación, entornos de datos y *streaming*.
 - D. Naturaleza de la capacidad desarrollada y el tipo de tareas necesarias para su construcción.
10. MLOps utiliza diferentes técnicas y herramientas para garantizar que los modelos de aprendizaje automático sean precisos, escalables y confiables en producción. Algunas de ellas son las siguientes:
- A. Integración y entrega continua.
 - B. Control de versiones.
 - C. Gestión de la configuración.
 - D. Todas las respuestas son válidas.