

Máster Título Propio Inteligencia Artificial en la Programación



tech universidad
FUNDEPOS

Máster Título Propio Inteligencia Artificial en la Programación

- » Modalidad: online
- » Duración: 12 meses
- » Titulación: TECH Universidad FUNDEPOS
- » Horario: a tu ritmo
- » Exámenes: online

Acceso web: www.techtitute.com/inteligencia-artificial/master/master-inteligencia-artificial-programacion

Índice

01

Presentación

pág. 4

02

Objetivos

pág. 8

03

Competencias

pág. 16

04

Dirección del curso

pág. 20

05

Estructura y contenido

pág. 24

06

Metodología

pág. 42

07

Titulación

pág. 50

01

Presentación

En una era caracterizada por los avances tecnológicos, la Inteligencia Artificial (IA) ha emergido como un instrumento primordial para los expertos en Programación. Su importancia radica en su habilidad para mecanizar actividades arduas, tomar decisiones fundamentadas en datos precisos y aprender de patrones. Así el Aprendizaje Automático ofrece valiosas técnicas para que los informáticos diseñen sistemas más inteligentes. Estas comprenden desde algoritmos para dotar a los programas de una mayor minuciosidad hasta el desarrollo de sistemas autónomos, útiles para cambiar la manera en la que se ejecutan los códigos. Por este motivo, TECH Universidad FUNDEPOS lanza una capacitación académica que otorgará al alumnado los últimos avances de este ámbito. Todo bajo una metodología 100% online, adaptada a la agenda de profesionales ocupados.





“

Serás capaz de diseñar experiencias de usuario personalizadas e intuitivas mediante esta titulación universitaria 100% online”

La Inteligencia Computacional sirve a las instituciones para mejorar la productividad en desarrollo de *software*. Sus herramientas cuentan con la capacidad para manejar datos no estructurados, aprender de experiencias pasadas y adaptarse a cambios en entornos dinámicos. Además, la IA puede predecir problemas potenciales de las aplicaciones antes de que sucedan, lo que permite a los profesionales tomar medidas preventivas para evitar problemas costosos de cara al futuro. En este contexto, las empresas informáticas más prestigiosas a nivel internacional buscan incorporar de forma activa a especialistas en la Arquitectura del Software para QA *Testing*.

Por ello, TECH Universidad FUNDEPOS implementa un innovador programa para que los programadores saquen el máximo provecho a la optimización y gestión del rendimiento en herramientas con IA. Diseñado por expertos de primer nivel, el plan de estudios ahondará en los algoritmos de programación para elaborar productos con sistemas inteligentes. Asimismo, el temario profundizará en las extensiones imprescindibles para el Visual Studio Code, el editor de código fuente más empleado en la actualidad. Por otra parte, los materiales didácticos abordarán la integración de IA en gestión con bases de datos para detectar posibles fallos y crear *test* unitarios. Se trata de una titulación universitaria que posee una diversidad de contenidos audiovisuales en múltiples formatos y un entramado de simulaciones reales para acercar el desarrollo del programa a la realidad de la praxis informática.

Para conseguir los objetivos de aprendizaje propuestos, este programa se imparte mediante una metodología de enseñanza online. De esta manera, los profesionales podrán compaginar a la perfección su trabajo con los estudios. Además, disfrutará de un cuadro docente de primer nivel y de unos materiales académicos multimedia de gran rigor pedagógico como clases magistrales, resúmenes interactivos o ejercicios prácticos. El único requisito para ingresar al Campus Virtual es que el alumnado disponga de un dispositivo electrónico con acceso a Internet, pudiendo emplear hasta su móvil.

Este **Máster Título Propio en Inteligencia Artificial en la Programación** contiene el programa educativo más completo y actualizado del mercado. Sus características más destacadas son:

- ♦ El desarrollo de casos prácticos presentados por expertos en Inteligencia Artificial en la Programación
- ♦ Los contenidos gráficos, esquemáticos y eminentemente prácticos con los que está concebido recogen una información científica y práctica sobre aquellas disciplinas indispensables para el ejercicio profesional
- ♦ Los ejercicios prácticos donde realizar el proceso de autoevaluación para mejorar el aprendizaje
- ♦ Su especial hincapié en metodologías innovadoras
- ♦ Las lecciones teóricas, preguntas al experto, foros de discusión de temas controvertidos y trabajos de reflexión individual
- ♦ La disponibilidad de acceso a los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet



Adquirirás una perspectiva holística sobre cómo el Aprendizaje Automático impacta y mejora cada etapa del desarrollo de software”

“

¿Buscas aplicar a tu praxis los Modelos Transformers para el procesamiento del lenguaje natural? Lógralo gracias a este innovador programa”

El programa incluye en su cuadro docente a profesionales del sector que vierten en esta capacitación la experiencia de su trabajo, además de reconocidos especialistas de sociedades de referencia y universidades de prestigio.

Su contenido multimedia, elaborado con la última tecnología educativa, permitirá al profesional un aprendizaje situado y contextual, es decir, un entorno simulado que proporcionará una capacitación inmersiva programada para entrenarse ante situaciones reales.

El diseño de este programa se centra en el Aprendizaje Basado en Problemas, mediante el cual el profesional deberá tratar de resolver las distintas situaciones de práctica profesional que se le planteen a lo largo del curso académico. Para ello, contará con la ayuda de un novedoso sistema de vídeo interactivo realizado por reconocidos expertos.

Profundizarás en el ciclo de vida del Testing, desde la creación de test cases hasta la detección de bugs.

El Relearning te permitirá aprender con menos esfuerzo y más rendimiento, implicándote más en tu especialización profesional.



02 Objetivos

Esta capacitación convertirá a los informáticos en expertos de la IA aplicada a la Programación. Los egresados adquirirán una visión integral, que combina los conocimientos más actualizados con destrezas prácticas que mejorarán su toma de decisiones. A su vez, los profesionales dominarán las herramientas más modernas para el desarrollo de *softwares* potenciados por el Aprendizaje Automático. De esta forma, los alumnos diseñarán propuestas tanto para las webs como aplicaciones móviles con adaptabilidad. Así estará altamente especializados para satisfacer las demandas actuales de la industria.





“

¿Buscas especializarte en Inteligencia Artificial? Con este programa dominarás la optimización del proceso de despliegue y la integración de Inteligencia Artificial en la computación en la nube”



Objetivos generales

- ♦ Desarrollar habilidades para configurar y gestionar entornos de desarrollo eficientes, asegurando una base sólida para la implementación de proyectos con IA
- ♦ Adquirir habilidades en la planificación, ejecución y automatización de pruebas de calidad, incorporando herramientas de IA para la detección y corrección de *bugs*
- ♦ Comprender y aplicar principios de rendimiento, escalabilidad y mantenibilidad en el diseño de sistemas informáticos a gran escala
- ♦ Familiarizarse con los patrones de diseño más importantes y aplicarlos de manera efectiva en la arquitectura del software



Objetivos específicos

Módulo 1. Fundamentos de la Inteligencia Artificial

- ♦ Analizar la evolución histórica de la Inteligencia Artificial, desde sus inicios hasta su estado actual, identificando hitos y desarrollos clave
- ♦ Comprender el funcionamiento de las redes de neuronas y su aplicación en modelos de aprendizaje en la Inteligencia Artificial
- ♦ Estudiar los principios y aplicaciones de los algoritmos genéticos, analizando su utilidad en la resolución de problemas complejos
- ♦ Analizar la importancia de los tesauros, vocabularios y taxonomías en la estructuración y procesamiento de datos para sistemas de IA
- ♦ Explorar el concepto de la web semántica y su influencia en la organización y comprensión de la información en entornos digitales

Módulo 2. Tipos y ciclo de vida del dato

- ♦ Comprender los conceptos fundamentales de la estadística y su aplicación en el análisis de datos
- ♦ Identificar y clasificar los distintos tipos de datos estadísticos, desde los cuantitativos hasta cualitativos
- ♦ Analizar el ciclo de vida de los datos, desde su generación hasta su eliminación, identificando las etapas clave
- ♦ Explorar las etapas iniciales del ciclo de vida de los datos, destacando la importancia de la planificación y la estructura de los datos
- ♦ Estudiar los procesos de recolección de datos, incluyendo la metodología, las herramientas y los canales de recolección
- ♦ Explorar el concepto de *Datawarehouse* (Almacén de Datos), haciendo hincapié en los elementos que lo integran y en su diseño
- ♦ Analizar los aspectos normativos relacionados con la gestión de datos, cumpliendo con regulaciones de privacidad y seguridad, así como de buenas prácticas

Módulo 3. El dato en la Inteligencia Artificial

- ♦ Dominar los fundamentos de la ciencia de datos, abarcando herramientas, tipos y fuentes para el análisis de información
- ♦ Explorar el proceso de transformación de datos en información utilizando técnicas de extracción y visualización de datos
- ♦ Estudiar la estructura y características de los *datasets*, comprendiendo su importancia en la preparación y utilización de datos para modelos de Inteligencia Artificial
- ♦ Analizar los modelos supervisados y no supervisados, incluyendo los métodos y la clasificación
- ♦ Utilizar herramientas específicas y buenas prácticas en el manejo y procesamiento de datos, asegurando la eficiencia y calidad en la implementación de la Inteligencia Artificial

Módulo 4. Minería de datos. Selección, preprocesamiento y transformación

- ♦ Dominar las técnicas de inferencia estadística para comprender y aplicar métodos estadísticos en la minería de datos
- ♦ Realizar un análisis exploratorio detallado de conjuntos de datos para identificar patrones, anomalías y tendencias relevantes
- ♦ Desarrollar habilidades para la preparación de datos, incluyendo su limpieza, integración y formateo para su uso en minería de datos
- ♦ Implementar estrategias efectivas para manejar valores perdidos en conjuntos de datos, aplicando métodos de imputación o eliminación según el contexto
- ♦ Identificar y mitigar el ruido presente en los datos, utilizando técnicas de filtrado y suavización para mejorar la calidad del conjunto de datos
- ♦ Abordar el preprocesamiento de datos en entornos *Big Data*

Módulo 5. Algoritmia y complejidad en Inteligencia Artificial

- ♦ Introducir estrategias de diseño de algoritmos, proporcionando una comprensión sólida de los enfoques fundamentales para la resolución de problemas
- ♦ Analizar la eficiencia y complejidad de los algoritmos, aplicando técnicas de análisis para evaluar el rendimiento en términos de tiempo y espacio
- ♦ Estudiar y aplicar algoritmos de ordenación, comprendiendo su funcionamiento y comparando su eficiencia en diferentes contextos
- ♦ Explorar algoritmos basados en árboles, comprendiendo su estructura y aplicaciones
- ♦ Investigar algoritmos con *Heaps*, analizando su implementación y utilidad en la manipulación eficiente de datos
- ♦ Analizar algoritmos basados en grafos, explorando su aplicación en la representación y solución de problemas que involucran relaciones complejas
- ♦ Estudiar algoritmos *Greedy*, entendiendo su lógica y aplicaciones en la resolución de problemas de optimización
- ♦ Investigar y aplicar la técnica de *backtracking* para la resolución sistemática de problemas, analizando su eficacia en diversos escenarios

Módulo 6. Sistemas inteligentes

- ♦ Explorar la teoría de agentes, comprendiendo los conceptos fundamentales de su funcionamiento y su aplicación en Inteligencia Artificial e ingeniería de Software
- ♦ Estudiar la representación del conocimiento, incluyendo el análisis de ontologías y su aplicación en la organización de información estructurada
- ♦ Analizar el concepto de la web semántica y su impacto en la organización y recuperación de información en entornos digitales

- ♦ Evaluar y comparar distintas representaciones del conocimiento, integrando estas para mejorar la eficacia y precisión de los sistemas inteligentes
- ♦ Estudiar razonadores semánticos, sistemas basados en conocimiento y sistemas expertos, comprendiendo su funcionalidad y aplicaciones en la toma de decisiones inteligentes

Módulo 7: Aprendizaje automático y minería de datos

- ♦ Introducir los procesos de descubrimiento del conocimiento y los conceptos fundamentales del aprendizaje automático
- ♦ Estudiar árboles de decisión como modelos de aprendizaje supervisado, comprendiendo su estructura y aplicaciones
- ♦ Evaluar clasificadores utilizando técnicas específicas para medir su rendimiento y precisión en la clasificación de datos
- ♦ Estudiar redes neuronales, comprendiendo su funcionamiento y arquitectura para resolver problemas complejos de aprendizaje automático
- ♦ Explorar métodos bayesianos y su aplicación en el aprendizaje automático, incluyendo redes bayesianas y clasificadores bayesianos
- ♦ Analizar modelos de regresión y de respuesta continua para la predicción de valores numéricos a partir de datos
- ♦ Estudiar técnicas de *clustering* para identificar patrones y estructuras en conjuntos de datos no etiquetados
- ♦ Explorar la minería de textos y el procesamiento del lenguaje natural (NLP), comprendiendo cómo se aplican técnicas de aprendizaje automático para analizar y comprender el texto

Módulo 8. Las redes neuronales, base de Deep Learning

- ♦ Dominar los fundamentos del Aprendizaje Profundo, comprendiendo su papel esencial en el *Deep Learning*
- ♦ Explorar las operaciones fundamentales en redes neuronales y comprender su aplicación en la construcción de modelos
- ♦ Analizar las diferentes capas utilizadas en redes neuronales y aprender a seleccionarlas adecuadamente

- ♦ Comprender la unión efectiva de capas y operaciones para diseñar arquitecturas de redes neuronales complejas y eficientes
- ♦ Utilizar entrenadores y optimizadores para ajustar y mejorar el rendimiento de las redes neuronales
- ♦ Explorar la conexión entre neuronas biológicas y artificiales para una comprensión más profunda del diseño de modelos
- ♦ Ajustar hiperparámetros para el *Fine Tuning* de redes neuronales, optimizando su rendimiento en tareas específicas

Módulo 9. Entrenamiento de redes neuronales profundas

- ♦ Resolver problemas relacionados con los gradientes en el entrenamiento de redes neuronales profundas
- ♦ Explorar y aplicar distintos optimizadores para mejorar la eficiencia y convergencia de los modelos
- ♦ Programar la tasa de aprendizaje para ajustar dinámicamente la velocidad de convergencia del modelo
- ♦ Comprender y abordar el sobreajuste mediante estrategias específicas durante el entrenamiento
- ♦ Aplicar directrices prácticas para garantizar un entrenamiento eficiente y efectivo de redes neuronales profundas
- ♦ Implementar *Transfer Learning* como una técnica avanzada para mejorar el rendimiento del modelo en tareas específicas
- ♦ Explorar y aplicar técnicas de *Data Augmentation* para enriquecer conjuntos de datos y mejorar la generalización del modelo
- ♦ Desarrollar aplicaciones prácticas utilizando *Transfer Learning* para resolver problemas del mundo real
- ♦ Comprender y aplicar técnicas de regularización para mejorar la generalización y evitar el sobreajuste en redes neuronales profundas

Módulo 10. Personalización de modelos y entrenamiento con *TensorFlow*

- ♦ Dominar los fundamentos de *TensorFlow* y su integración con NumPy para un manejo eficiente de datos y cálculos
- ♦ Personalizar modelos y algoritmos de entrenamiento utilizando las capacidades avanzadas de *TensorFlow*
- ♦ Explorar la API tfdata para gestionar y manipular conjuntos de datos de manera eficaz
- ♦ Implementar el formato TFRecord para almacenar y acceder a grandes conjuntos de datos en *TensorFlow*
- ♦ Utilizar capas de preprocesamiento de Keras para facilitar la construcción de modelos personalizados
- ♦ Explorar el proyecto *TensorFlow Datasets* para acceder a conjuntos de datos predefinidos y mejorar la eficiencia en el desarrollo
- ♦ Desarrollar una aplicación de *Deep Learning* con *TensorFlow*, integrando los conocimientos adquiridos en el módulo
- ♦ Aplicar de manera práctica todos los conceptos aprendidos en la construcción y entrenamiento de modelos personalizados con *TensorFlow* en situaciones del mundo real

Módulo 11. *Deep Computer Vision* con Redes Neuronales Convolucionales

- ♦ Comprender la arquitectura del córtex visual y su relevancia en *Deep Computer Vision*
- ♦ Explorar y aplicar capas convolucionales para extraer características clave de imágenes
- ♦ Implementar capas de agrupación y su utilización en modelos de *Deep Computer Vision* con Keras
- ♦ Analizar diversas arquitecturas de Redes Neuronales Convolucionales (CNN) y su aplicabilidad en diferentes contextos
- ♦ Desarrollar e implementar una CNN ResNet utilizando la biblioteca Keras para mejorar la eficiencia y rendimiento del modelo
- ♦ Utilizar modelos preentrenados de Keras para aprovechar el aprendizaje por transferencia en tareas específicas

- ♦ Aplicar técnicas de clasificación y localización en entornos de *Deep Computer Vision*
- ♦ Explorar estrategias de detección de objetos y seguimiento de objetos utilizando Redes Neuronales Convolucionales
- ♦ Implementar técnicas de segmentación semántica para comprender y clasificar objetos en imágenes de manera detallada

Módulo 12. Procesamiento del lenguaje natural (NLP) con Redes Naturales Recurrentes (RNN) y Atención

- ♦ Desarrollar habilidades en generación de texto utilizando Redes Neuronales Recurrentes (RNN)
- ♦ Aplicar RNN en la clasificación de opiniones para análisis de sentimientos en textos
- ♦ Comprender y aplicar los mecanismos de atención en modelos de procesamiento del lenguaje natural
- ♦ Analizar y utilizar modelos *Transformers* en tareas específicas de NLP
- ♦ Explorar la aplicación de modelos *Transformers* en el contexto de procesamiento de imágenes y visión computacional
- ♦ Familiarizarse con la librería de *Transformers* de *Hugging Face* para la implementación eficiente de modelos avanzados
- ♦ Comparar diferentes librerías de *Transformers* para evaluar su idoneidad en tareas específicas
- ♦ Desarrollar una aplicación práctica de NLP que integre RNN y mecanismos de atención para resolver problemas del mundo real

Módulo 13. *Autoencoders*, GANs, y modelos de difusión

- ♦ Desarrollar representaciones eficientes de datos mediante *Autoencoders*, GANs y Modelos de Difusión
- ♦ Realizar PCA utilizando un codificador automático lineal incompleto para optimizar la representación de datos
- ♦ Implementar y comprender el funcionamiento de codificadores automáticos apilados
- ♦ Explorar y aplicar autocodificadores convolucionales para representaciones eficientes de datos visuales

- ♦ Analizar y aplicar la eficacia de codificadores automáticos dispersos en la representación de datos
- ♦ Generar imágenes de moda del conjunto de datos MNIST utilizando *Autoencoders*
- ♦ Comprender el concepto de Redes Adversarias Generativas (GANs) y Modelos de Difusión
- ♦ Implementar y comparar el rendimiento de Modelos de Difusión y GANs en la generación de datos

Módulo 14. Computación bioinspirada

- ♦ Introducir los conceptos fundamentales de la computación bioinspirada
- ♦ Explorar algoritmos de adaptación social como enfoque clave en la computación bioinspirada
- ♦ Analizar estrategias de exploración-explotación del espacio en algoritmos genéticos
- ♦ Examinar modelos de computación evolutiva en el contexto de la optimización
- ♦ Continuar el análisis detallado de modelos de computación evolutiva
- ♦ Aplicar programación evolutiva a problemas específicos de aprendizaje
- ♦ Abordar la complejidad de problemas multiobjetivo en el marco de la computación bioinspirada
- ♦ Explorar la aplicación de redes neuronales en el ámbito de la computación bioinspirada
- ♦ Profundizar en la implementación y utilidad de redes neuronales en la computación bioinspirada

Módulo 15. Inteligencia Artificial: Estrategias y aplicaciones

- ♦ Desarrollar estrategias de implementación de inteligencia artificial en servicios financieros
- ♦ Analizar las implicaciones de la inteligencia artificial en la prestación de servicios sanitarios
- ♦ Identificar y evaluar los riesgos asociados al uso de la IA en el ámbito de la salud
- ♦ Evaluar los riesgos potenciales vinculados al uso de IA en la industria
- ♦ Aplicar técnicas de inteligencia artificial en industria para mejorar la productividad
- ♦ Diseñar soluciones de inteligencia artificial para optimizar procesos en la administración pública

- ♦ Evaluar la implementación de tecnologías de IA en el sector educativo
- ♦ Aplicar técnicas de inteligencia artificial en silvicultura y agricultura para mejorar la productividad
- ♦ Optimizar procesos de recursos humanos mediante el uso estratégico de la inteligencia artificial

Módulo 16. Mejora de la productividad en desarrollo de software con IA

- ♦ Ahondar en la implementación de extensiones imprescindibles de IA en Visual Studio Code para mejorar la productividad y facilitar el desarrollo de *software*
- ♦ Obtener una comprensión sólida de los conceptos básicos de la IA y su aplicación en el desarrollo de *software*, incluyendo algoritmos de aprendizaje automático, procesamiento del lenguaje natural, redes neuronales, etc.
- ♦ Dominar la configuración de entornos de desarrollo optimizados, asegurando que los estudiantes puedan crear ambientes propicios para proyectos de IA
- ♦ Aplicar técnicas específicas utilizando ChatGPT para la identificación y corrección automática de posibles mejoras en el código, fomentando prácticas de programación más eficientes
- ♦ Promover la colaboración entre profesionales de diferentes programadores (desde programadores hasta ingenieros de datos o diseñadores de experiencia de usuarios) para desarrollar soluciones de *software* con IA efectivas y éticas

Módulo 17. Arquitectura del software para QA Testing

- ♦ Desarrollar habilidades para diseñar planes de pruebas sólidos, cubriendo diferentes tipos de *testing* y garantizando la calidad del software
- ♦ Reconocer y analizar diferentes tipos de estructuras de *software*, como la monolítica, de microservicios u orientadas a servicios
- ♦ Obtener una visión integral sobre los principios y técnicas para diseñar sistemas informáticos que sean escalables y capaces de manejar grandes volúmenes de datos

- ♦ Aplicar conocimientos avanzados en la implementación de estructuras de datos potenciadas por IA para optimizar el rendimiento y la eficiencia del *software*
- ♦ Desarrollar prácticas de desarrollo seguro, centrándose en evitar vulnerabilidades para garantizar la seguridad del *software* a nivel arquitectónico

Módulo 18. Proyectos web con IA

- ♦ Desarrollar habilidades integrales para la implementación de proyectos web, desde el diseño del *frontend* hasta la optimización del *backend*, con la inclusión de elementos de IA
- ♦ Optimizar el proceso de despliegue de sitios web, incorporando técnicas y herramientas para mejorar la velocidad y la eficiencia
- ♦ Integrar la IA en la computación en la nube, permitiendo al alumnado crear proyectos web altamente escalables y eficientes
- ♦ Adquirir la capacidad de identificar problemas y oportunidades específicos en proyectos web donde la IA puede ser aplicada de manera efectiva, como en el procesamiento de texto, personalización, recomendación de contenido, etc.
- ♦ Fomentar que los estudiantes se mantengan al tanto de las últimas tendencias y avances en IA para su correcta aplicación en proyectos web

Módulo 19. Aplicaciones móviles con IA

- ♦ Aplicar conceptos avanzados de *clean architecture*, *datasources* y *repositories* para garantizar una estructura robusta y modular en aplicaciones móviles con IA
- ♦ Desarrollar habilidades para diseñar pantallas interactivas, iconos y recursos gráficos utilizando IA para mejorar la experiencia del usuario en las aplicaciones móviles
- ♦ Profundizar en la configuración del entorno de trabajo para aplicaciones móviles y emplear *GitHub Copilot* para agilizar el proceso de desarrollo
- ♦ Optimizar las aplicaciones móviles con IA para un rendimiento eficiente, teniendo en cuenta la administración de recursos y el uso de datos
- ♦ Realizar de pruebas de calidad en aplicaciones móviles con IA, que permitan al alumnado identificar problemas y depurar errores

Módulo 20. IA para QA Testing

- ♦ Dominar principios y técnicas para diseñar sistemas informáticos que sean escalables y capaces de manejar grandes volúmenes de datos
- ♦ Aplicar conocimientos avanzados en la implementación de estructuras de datos potenciadas por IA para optimizar el rendimiento y la eficiencia del *software*
- ♦ Comprender y aplicar prácticas de desarrollo seguro, centrándose en evitar vulnerabilidades como la inyección, para garantizar la seguridad del *software* a nivel arquitectónico
- ♦ Generar pruebas automatizadas, especialmente en entornos web y móviles, integrando herramientas de IA para mejorar así la eficacia del proceso
- ♦ Utilizar las herramientas avanzadas de QA potenciadas por IA para una detección más eficientes de *bugs* y una mejora continua del *software*

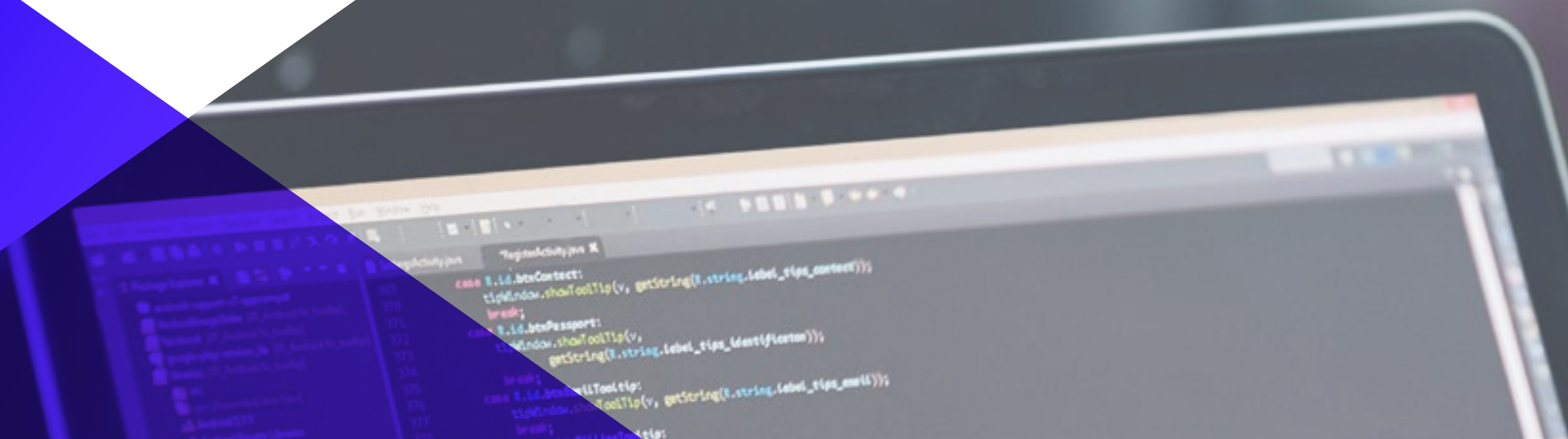


Profundizarás en la integración de elementos del Visual Studio Code y la optimización de código con ChatGPT, a través de un programa académico exhaustivo”

03

Competencias

A través de 12 meses de aprendizaje, los egresados contarán con una serie de competencias que elevarán sus horizontes profesionales en la industria informática. Así pues, los expertos podrán crear e implementar softwares avanzados y diseñar propuestas de IA a gran escala. En relación con esto, los profesionales abordarán la productividad en el desarrollo, al mismo tiempo que realizan las mejores prácticas de QA Testing. Esta titulación universitaria garantiza que los informáticos estén capacitados para superar con éxito los desafíos del mundo real y destaquen considerablemente en un área que avanza a pasos agigantados.



“

Estarás elevadamente cualificado para implementar algoritmos del Aprendizaje Automático a tus proyectos web y aplicaciones móviles”

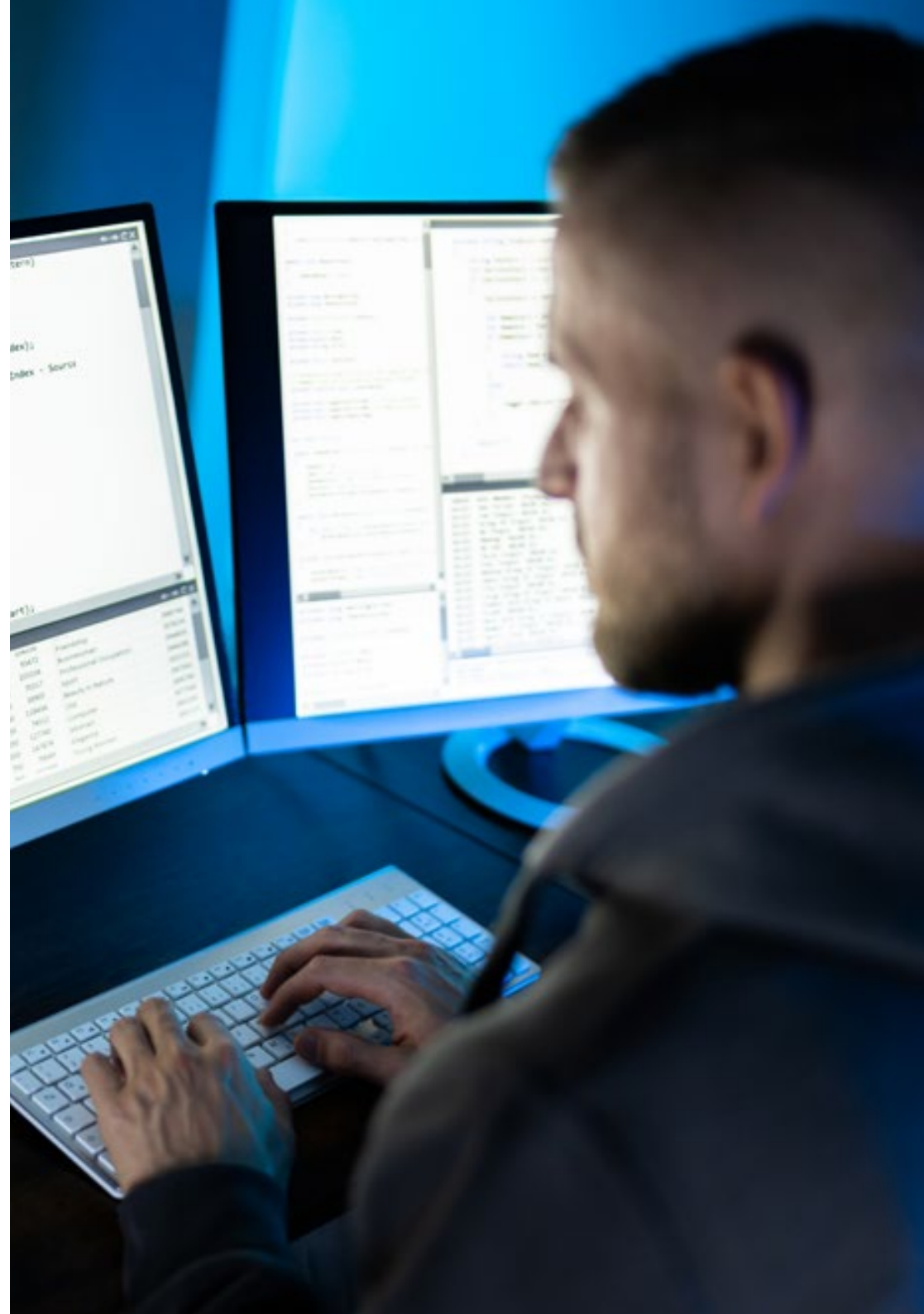


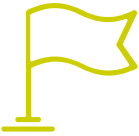
Competencias generales

- ♦ Aplicar extensiones de IA en Visual Studio Code y técnicas de diseño *no-code* para aumentar la eficiencia en el desarrollo de software
- ♦ Utilizar ChatGPT para optimizar y mejorar la calidad del código, aplicando prácticas avanzadas de programación
- ♦ Implementar proyectos web, desde la creación de *workspaces* hasta el despliegue, integrando IA, tanto en el *frontend* como en el *backend*
- ♦ Desarrollar aplicaciones móviles potenciadas por IA, desde la configuración del entorno hasta la creación de funciones avanzadas y la gestión de recursos gráficos
- ♦ Aplicar conceptos avanzados de almacenamiento y estructuras de datos potenciadas por IA para mejorar la eficiencia y escalabilidad de los sistemas
- ♦ Incluir prácticas de desarrollo seguro, evitando vulnerabilidades como la inyección, para garantizar la integridad y seguridad del software desarrollado



Dominarás las tecnologías del futuro con esta exclusiva titulación universitaria 100% online. ¡Solo con TECH Universidad FUNDEPOS!





Competencias específicas

- ♦ Aplicar técnicas y estrategias de IA para mejorar la eficiencia en el sector *retail*
- ♦ Implementar técnicas de eliminación de ruido utilizando codificadores automáticos
- ♦ Crear de manera efectiva conjuntos de datos de entrenamiento para tareas de procesamiento del lenguaje natural (NLP)
- ♦ Ejecutar capas de agrupación y su utilización en modelos de *Deep Computer Vision* con Keras
- ♦ Utilizar funciones y gráficos de *TensorFlow* para optimizar el rendimiento de los modelos personalizados
- ♦ Optimizar el desarrollo y aplicación de *chatbots* y asistentes virtuales, comprendiendo su funcionamiento y potenciales aplicaciones
- ♦ Dominar la reutilización de capas preentrenadas para optimizar y acelerar el proceso de entrenamiento
- ♦ Construir la primera red neuronal, aplicando los conceptos aprendidos en la práctica
- ♦ Activar Perceptrón Multicapa (MLP) utilizando la biblioteca Keras
- ♦ Aplicar técnicas de exploración y preprocesamiento de datos, identificando y preparando datos para su uso efectivo en modelos de aprendizaje automático
- ♦ Indagar en lenguajes y software para la creación de ontologías, utilizando herramientas específicas para el desarrollo de modelos semánticos
- ♦ Desarrollar las técnicas de limpieza de datos para garantizar la calidad y precisión de la información utilizada en análisis posteriores
- ♦ Dominar la configuración de entornos de desarrollo optimizados, asegurando que los estudiantes puedan crear ambientes propicios para proyectos de IA
- ♦ Aplicar técnicas específicas, utilizando ChatGPT para la identificación y corrección automática de posibles mejoras en el código, fomentando prácticas de programación más eficientes
- ♦ Crear pruebas automatizadas, especialmente en entornos web y móviles, integrando herramientas de IA para mejorar la eficacia del proceso.
- ♦ Utilizar herramientas avanzadas de QA potenciadas por IA para una detección más eficiente de bugs y una mejora continua del software
- ♦ Integrar la IA en la computación en la nube, permitiendo a los estudiantes crear proyectos web altamente escalables y eficientes
- ♦ Configurar el entorno de trabajo para aplicaciones móviles y utilizar Github Copilot para agilizar el proceso de desarrollo

04

Dirección del curso

Leal su compromiso de ofrecer la máxima calidad educativa, TECH Universidad FUNDEPOS dispone de un claustro docente de primerísimo nivel. Por eso, los expertos que conforman esta titulación universitaria destacan por un profundo conocimiento sobre la IA en Programación, a la vez que atesoran años de experiencia profesional a sus espaldas. Gracias a estos aspectos, este itinerario académico brinda al alumnado las mejores herramientas tecnológicas y estrategias no solo para ampliar su entendimiento, sino para que obtengan habilidades para perfeccionar sus procedimientos informáticos. Así pues, los estudiantes tienen las garantías que precisan para especializarse en un sector digital que ofrece numerosas oportunidades.



“

Los principales expertos del Aprendizaje Automático en la Programación se han unido para compartir contigo todos sus conocimientos en este sector”

Dirección



Dr. Peralta Martín-Palomino, Arturo

- ♦ CEO y CTO en Prometheus Global Solutions
- ♦ CTO en Korporate Technologies
- ♦ CTO en AI Shepherds GmbH
- ♦ Consultor y Asesor Estratégico Empresarial en Alliance Medical
- ♦ Director de Diseño y Desarrollo en DocPath
- ♦ Doctor en Ingeniería Informática por la Universidad de Castilla-La Mancha
- ♦ Doctor en Economía, Empresas y Finanzas por la Universidad Camilo José Cela
- ♦ Doctor en Psicología por la Universidad de Castilla-La Mancha
- ♦ Máster en Executive MBA por la Universidad Isabel I
- ♦ Máster en Dirección Comercial y Marketing por la Universidad Isabel I
- ♦ Máster Experto en Big Data por Formación Hadoop
- ♦ Máster en Tecnologías Informáticas Avanzadas por la Universidad de Castilla-La Mancha
- ♦ Miembro de: Grupo de Investigación SMILE



D. Castellanos Herreros, Ricardo

- ♦ Especialista en Ingeniería Informática de Sistemas
- ♦ *Chief Technology Officer* en OWQLO
- ♦ Consultor Técnico *Freelance*
- ♦ Desarrollador de Aplicaciones Móviles para eDreams, Fnac, Air Europa, Bankia, Cetelem, Banco Santander, Santillana, Groupón y Grupo Planeta
- ♦ Desarrollador de Páginas Web para Openbank y Banco Santander
- ♦ Curso de *Machine Learning Engineer* en Udacity
- ♦ Ingeniero Técnico en Informática de Sistemas por la Universidad de Castilla la Mancha

05

Estructura y contenido

Este Máster Título Propio proporcionará a los egresados un enfoque holístico, que les supondrá una ventaja significativa en el desarrollo informático equiparlos con destrezas específicas. Para conseguirlo, la capacitación abarcará desde la preparación del entorno de desarrollo hasta la optimización del *software* y la implementación de IA en proyectos reales. El temario profundizará en aspectos tales como el diseño de *no-code* de interfaces, uso de ChatGPT para optimizar los códigos o la aplicación del Aprendizaje Automática en el *QA Testing*. De esta forma, los egresados implementarán soluciones innovadoras de un modo efectivo en diversas aplicaciones como proyectos web y móviles.



“

Actualiza tus conocimientos sobre Inteligencia Artificial en la Programación través de un contenido multimedia innovador”

Módulo 1. Fundamentos de la Inteligencia Artificial

- 1.1. Historia de la Inteligencia artificial
 - 1.1.1. ¿Cuándo se empieza a hablar de inteligencia artificial?
 - 1.1.2. Referentes en el cine
 - 1.1.3. Importancia de la inteligencia artificial
 - 1.1.4. Tecnologías que habilitan y dan soporte a la inteligencia artificial
- 1.2. La Inteligencia Artificial en juegos
 - 1.2.1. Teoría de Juegos
 - 1.2.2. *Minimax* y poda Alfa-Beta
 - 1.2.3. Simulación: Monte Carlo
- 1.3. Redes de neuronas
 - 1.3.1. Fundamentos biológicos
 - 1.3.2. Modelo computacional
 - 1.3.3. Redes de neuronas supervisadas y no supervisadas
 - 1.3.4. Perceptrón simple
 - 1.3.5. Perceptrón multicapa
- 1.4. Algoritmos genéticos
 - 1.4.1. Historia
 - 1.4.2. Base biológica
 - 1.4.3. Codificación de problemas
 - 1.4.4. Generación de la población inicial
 - 1.4.5. Algoritmo principal y operadores genéticos
 - 1.4.6. Evaluación de individuos: Fitness
- 1.5. Tesoros, vocabularios, taxonomías
 - 1.5.1. Vocabularios
 - 1.5.2. Taxonomías
 - 1.5.3. Tesoros
 - 1.5.4. Ontologías
 - 1.5.5. Representación del conocimiento: web semántica
- 1.6. Web semántica
 - 1.6.1. Especificaciones: RDF, RDFS y OWL
 - 1.6.2. Inferencia/razonamiento
 - 1.6.3. *Linked Data*

- 1.7. Sistemas expertos y DSS
 - 1.7.1. Sistemas expertos
 - 1.7.2. Sistemas de soporte a la decisión
- 1.8. *Chatbots* y Asistentes Virtuales
 - 1.8.1. Tipos de asistentes: asistentes por voz y por texto
 - 1.8.2. Partes fundamentales para el desarrollo de un asistente: *Intents*, entidades y flujo de diálogo
 - 1.8.3. Integraciones: web, *Slack*, Whatsapp, Facebook
 - 1.8.4. Herramientas de desarrollo de asistentes: *Dialog Flow*, *Watson Assistant*
- 1.9. Estrategia de implantación de IA
- 1.10. Futuro de la inteligencia artificial
 - 1.10.1. Entendemos cómo detectar emociones mediante algoritmos
 - 1.10.2. Creación de una personalidad: lenguaje, expresiones y contenido
 - 1.10.3. Tendencias de la inteligencia artificial
 - 1.10.4. Reflexiones

Módulo 2. Tipos y Ciclo de Vida del Dato

- 2.1. La Estadística
 - 2.1.1. Estadística: estadística descriptiva, estadística inferencias
 - 2.1.2. Población, muestra, individuo
 - 2.1.3. Variables: definición, escalas de medida
- 2.2. Tipos de datos estadísticos
 - 2.2.1. Según tipo
 - 2.2.1.1. Cuantitativos: datos continuos y datos discretos
 - 2.2.1.2. Cualitativos: datos binomiales, datos nominales y datos ordinales
 - 2.2.2. Según su forma
 - 2.2.2.1. Numérico
 - 2.2.2.2. Texto
 - 2.2.2.3. Lógico
 - 2.2.3. Según su fuente
 - 2.2.3.1. Primarios
 - 2.2.3.2. Secundarios

- 2.3. Ciclo de vida de los datos
 - 2.3.1. Etapas del ciclo
 - 2.3.2. Hitos del ciclo
 - 2.3.3. Principios FAIR
- 2.4. Etapas iniciales del ciclo
 - 2.4.1. Definición de metas
 - 2.4.2. Determinación de recursos necesarios
 - 2.4.3. Diagrama de Gantt
 - 2.4.4. Estructura de los datos
- 2.5. Recolección de datos
 - 2.5.1. Metodología de recolección
 - 2.5.2. Herramientas de recolección
 - 2.5.3. Canales de recolección
- 2.6. Limpieza del dato
 - 2.6.1. Fases de la limpieza de datos
 - 2.6.2. Calidad del dato
 - 2.6.3. Manipulación de datos (con R)
- 2.7. Análisis de datos, interpretación y valoración de resultados
 - 2.7.1. Medidas estadísticas
 - 2.7.2. Índices de relación
 - 2.7.3. Minería de datos
- 2.8. Almacén del dato (*Datawarehouse*)
 - 2.8.1. Elementos que lo integran
 - 2.8.2. Diseño
 - 2.8.3. Aspectos a considerar
- 2.9. Disponibilidad del dato
 - 2.9.1. Acceso
 - 2.9.2. Utilidad
 - 2.9.3. Seguridad
- 2.10. Aspectos Normativos
 - 2.10.1. Ley de protección de datos
 - 2.10.2. Buenas prácticas
 - 2.10.3. Otros aspectos normativos

Módulo 3. El dato en la Inteligencia Artificial

- 3.1. Ciencia de datos
 - 3.1.1. La ciencia de datos
 - 3.1.2. Herramientas avanzadas para el científico de datos
- 3.2. Datos, información y conocimiento
 - 3.2.1. Datos, información y conocimiento
 - 3.2.2. Tipos de datos
 - 3.2.3. Fuentes de datos
- 3.3. De los datos a la información
 - 3.3.1. Análisis de Datos
 - 3.3.2. Tipos de análisis
 - 3.3.3. Extracción de Información de un *Dataset*
- 3.4. Extracción de información mediante visualización
 - 3.4.1. La visualización como herramienta de análisis
 - 3.4.2. Métodos de visualización
 - 3.4.3. Visualización de un conjunto de datos
- 3.5. Calidad de los datos
 - 3.5.1. Datos de calidad
 - 3.5.2. Limpieza de datos
 - 3.5.3. Preprocesamiento básico de datos
- 3.6. *Dataset*
 - 3.6.1. Enriquecimiento del *Dataset*
 - 3.6.2. La maldición de la dimensionalidad
 - 3.6.3. Modificación de nuestro conjunto de datos
- 3.7. Desbalanceo
 - 3.7.1. Desbalanceo de clases
 - 3.7.2. Técnicas de mitigación del desbalanceo
 - 3.7.3. Balanceo de un *Dataset*
- 3.8. Modelos no supervisados
 - 3.8.1. Modelo no supervisado
 - 3.8.2. Métodos
 - 3.8.3. Clasificación con modelos no supervisados

- 3.9. Modelos supervisados
 - 3.9.1. Modelo supervisado
 - 3.9.2. Métodos
 - 3.9.3. Clasificación con modelos supervisados
- 3.10. Herramientas y buenas prácticas
 - 3.10.1. Buenas prácticas para un científico de datos
 - 3.10.2. El mejor modelo
 - 3.10.3. Herramientas útiles

Módulo 4. Minería de Datos. Selección, preprocesamiento y transformación

- 4.1. La inferencia estadística
 - 4.1.1. Estadística descriptiva vs. Inferencia estadística
 - 4.1.2. Procedimientos paramétricos
 - 4.1.3. Procedimientos no paramétricos
- 4.2. Análisis exploratorio
 - 4.2.1. Análisis descriptivo
 - 4.2.2. Visualización
 - 4.2.3. Preparación de datos
- 4.3. Preparación de datos
 - 4.3.1. Integración y limpieza de datos
 - 4.3.2. Normalización de datos
 - 4.3.3. Transformando atributos
- 4.4. Los valores perdidos
 - 4.4.1. Tratamiento de valores perdidos
 - 4.4.2. Métodos de imputación de máxima verosimilitud
 - 4.4.3. Imputación de valores perdidos usando aprendizaje automático
- 4.5. El ruido en los datos
 - 4.5.1. Clases de ruido y atributos
 - 4.5.2. Filtrado de ruido
 - 4.5.3. El efecto del ruido
- 4.6. La maldición de la dimensionalidad
 - 4.6.1. *Oversampling*
 - 4.6.2. *Undersampling*
 - 4.6.3. Reducción de datos multidimensionales

- 4.7. De atributos continuos a discretos
 - 4.7.1. Datos continuos versus discretos
 - 4.7.2. Proceso de discretización
- 4.8. Los datos
 - 4.8.1. Selección de datos
 - 4.8.2. Perspectivas y criterios de selección
 - 4.8.3. Métodos de selección
- 4.9. Selección de instancias
 - 4.9.1. Métodos para la selección de instancias
 - 4.9.2. Selección de prototipos
 - 4.9.3. Métodos avanzados para la selección de instancias
- 4.10. Preprocesamiento de datos en entornos Big Data

Módulo 5. Algoritmia y complejidad en Inteligencia Artificial

- 5.1. Introducción a las estrategias de diseño de algoritmos
 - 5.1.1. Recursividad
 - 5.1.2. Divide y conquista
 - 5.1.3. Otras estrategias
- 5.2. Eficiencia y análisis de los algoritmos
 - 5.2.1. Medidas de eficiencia
 - 5.2.2. Medir el tamaño de la entrada
 - 5.2.3. Medir el tiempo de ejecución
 - 5.2.4. Caso peor, mejor y medio
 - 5.2.5. Notación asintótica
 - 5.2.6. Criterios de Análisis matemático de algoritmos no recursivos
 - 5.2.7. Análisis matemático de algoritmos recursivos
 - 5.2.8. Análisis empírico de algoritmos
- 5.3. Algoritmos de ordenación
 - 5.3.1. Concepto de ordenación
 - 5.3.2. Ordenación de la burbuja
 - 5.3.3. Ordenación por selección
 - 5.3.4. Ordenación por inserción
 - 5.3.5. Ordenación por mezcla (*Merge_Sort*)
 - 5.3.6. Ordenación rápida (*Quick_Sort*)

- 5.4. Algoritmos con árboles
 - 5.4.1. Concepto de árbol
 - 5.4.2. Árboles binarios
 - 5.4.3. Recorridos de árbol
 - 5.4.4. Representar expresiones
 - 5.4.5. Árboles binarios ordenados
 - 5.4.6. Árboles binarios balanceados
- 5.5. Algoritmos con *Heaps*
 - 5.5.1. Los *Heaps*
 - 5.5.2. El algoritmo *Heapsort*
 - 5.5.3. Las colas de prioridad
- 5.6. Algoritmos con grafos
 - 5.6.1. Representación
 - 5.6.2. Recorrido en anchura
 - 5.6.3. Recorrido en profundidad
 - 5.6.4. Ordenación topológica
- 5.7. Algoritmos *Greedy*
 - 5.7.1. La estrategia *Greedy*
 - 5.7.2. Elementos de la estrategia *Greedy*
 - 5.7.3. Cambio de monedas
 - 5.7.4. Problema del viajante
 - 5.7.5. Problema de la mochila
- 5.8. Búsqueda de caminos mínimos
 - 5.8.1. El problema del camino mínimo
 - 5.8.2. Arcos negativos y ciclos
 - 5.8.3. Algoritmo de Dijkstra
- 5.9. Algoritmos *Greedy* sobre grafos
 - 5.9.1. El árbol de recubrimiento mínimo
 - 5.9.2. El algoritmo de Prim
 - 5.9.3. El algoritmo de Kruskal
 - 5.9.4. Análisis de complejidad
- 5.10. *Backtracking*
 - 5.10.1. El *Backtracking*
 - 5.10.2. Técnicas alternativas

Módulo 6. Sistemas inteligentes

- 6.1. Teoría de agentes
 - 6.1.1. Historia del concepto
 - 6.1.2. Definición de agente
 - 6.1.3. Agentes en Inteligencia Artificial
 - 6.1.4. Agentes en ingeniería de Software
- 6.2. Arquitecturas de agentes
 - 6.2.1. El proceso de razonamiento de un agente
 - 6.2.2. Agentes reactivos
 - 6.2.3. Agentes deductivos
 - 6.2.4. Agentes híbridos
 - 6.2.5. Comparativa
- 6.3. Información y conocimiento
 - 6.3.1. Distinción entre datos, información y conocimiento
 - 6.3.2. Evaluación de la calidad de los datos
 - 6.3.3. Métodos de captura de datos
 - 6.3.4. Métodos de adquisición de información
 - 6.3.5. Métodos de adquisición de conocimiento
- 6.4. Representación del conocimiento
 - 6.4.1. La importancia de la representación del conocimiento
 - 6.4.2. Definición de representación del conocimiento a través de sus roles
 - 6.4.3. Características de una representación del conocimiento
- 6.5. Ontologías
 - 6.5.1. Introducción a los metadatos
 - 6.5.2. Concepto filosófico de ontología
 - 6.5.3. Concepto informático de ontología
 - 6.5.4. Ontologías de dominio y ontologías de nivel superior
 - 6.5.5. ¿Cómo construir una ontología?
- 6.6. Lenguajes para ontologías y Software para la creación de ontologías
 - 6.6.1. Tripletas RDF, *Turtle* y N
 - 6.6.2. RDF *Schema*
 - 6.6.3. OWL

- 6.6.4. SPARQL
- 6.6.5. Introducción a las diferentes herramientas para la creación de ontologías
- 6.6.6. Instalación y uso de *Protégé*
- 6.7. La web semántica
 - 6.7.1. El estado actual y futuro de la web semántica
 - 6.7.2. Aplicaciones de la web semántica
- 6.8. Otros modelos de representación del conocimiento
 - 6.8.1. Vocabularios
 - 6.8.2. Visión global
 - 6.8.3. Taxonomías
 - 6.8.4. Tesoros
 - 6.8.5. Folksonomías
 - 6.8.6. Comparativa
 - 6.8.7. Mapas mentales
- 6.9. Evaluación e integración de representaciones del conocimiento
 - 6.9.1. Lógica de orden cero
 - 6.9.2. Lógica de primer orden
 - 6.9.3. Lógica descriptiva
 - 6.9.4. Relación entre diferentes tipos de lógica
 - 6.9.5. *Prolog*: programación basada en lógica de primer orden
- 6.10. Razonadores semánticos, sistemas basados en conocimiento y Sistemas Expertos
 - 6.10.1. Concepto de razonador
 - 6.10.2. Aplicaciones de un razonador
 - 6.10.3. Sistemas basados en el conocimiento
 - 6.10.4. MYCIN, historia de los Sistemas Expertos
 - 6.10.5. Elementos y Arquitectura de Sistemas Expertos
 - 6.10.6. Creación de Sistemas Expertos

Módulo 7. Aprendizaje automático y minería de datos

- 7.1. Introducción a los procesos de descubrimiento del conocimiento y conceptos básicos de aprendizaje automático
 - 7.1.1. Conceptos clave de los procesos de descubrimiento del conocimiento
 - 7.1.2. Perspectiva histórica de los procesos de descubrimiento del conocimiento
 - 7.1.3. Etapas de los procesos de descubrimiento del conocimiento

- 7.1.4. Técnicas utilizadas en los procesos de descubrimiento del conocimiento
- 7.1.5. Características de los buenos modelos de aprendizaje automático
- 7.1.6. Tipos de información de aprendizaje automático
- 7.1.7. Conceptos básicos de aprendizaje
- 7.1.8. Conceptos básicos de aprendizaje no supervisado
- 7.2. Exploración y preprocesamiento de datos
 - 7.2.1. Tratamiento de datos
 - 7.2.2. Tratamiento de datos en el flujo de análisis de datos
 - 7.2.3. Tipos de datos
 - 7.2.4. Transformaciones de datos
 - 7.2.5. Visualización y exploración de variables continuas
 - 7.2.6. Visualización y exploración de variables categóricas
 - 7.2.7. Medidas de correlación
 - 7.2.8. Representaciones gráficas más habituales
 - 7.2.9. Introducción al análisis multivariante y a la reducción de dimensiones
- 7.3. Árboles de decisión
 - 7.3.1. Algoritmo ID
 - 7.3.2. Algoritmo C
 - 7.3.3. Sobreentrenamiento y poda
 - 7.3.4. Análisis de resultados
- 7.4. Evaluación de clasificadores
 - 7.4.1. Matrices de confusión
 - 7.4.2. Matrices de evaluación numérica
 - 7.4.3. Estadístico de Kappa
 - 7.4.4. La curva ROC
- 7.5. Reglas de clasificación
 - 7.5.1. Medidas de evaluación de reglas
 - 7.5.2. Introducción a la representación gráfica
 - 7.5.3. Algoritmo de recubrimiento secuencial
- 7.6. Redes neuronales
 - 7.6.1. Conceptos básicos
 - 7.6.2. Redes de neuronas simples
 - 7.6.3. Algoritmo de *Backpropagation*
 - 7.6.4. Introducción a las redes neuronales recurrentes

- 7.7. Métodos bayesianos
 - 7.7.1. Conceptos básicos de probabilidad
 - 7.7.2. Teorema de Bayes
 - 7.7.3. Naive Bayes
 - 7.7.4. Introducción a las redes bayesianas
- 7.8. Modelos de regresión y de respuesta continua
 - 7.8.1. Regresión lineal simple
 - 7.8.2. Regresión lineal múltiple
 - 7.8.3. Regresión logística
 - 7.8.4. Árboles de regresión
 - 7.8.5. Introducción a las máquinas de soporte vectorial (SVM)
 - 7.8.6. Medidas de bondad de ajuste
- 7.9. *Clustering*
 - 7.9.1. Conceptos básicos
 - 7.9.2. *Clustering* jerárquico
 - 7.9.3. Métodos probabilistas
 - 7.9.4. Algoritmo EM
 - 7.9.5. Método *B-Cubed*
 - 7.9.6. Métodos implícitos
- 7.10. Minería de textos y procesamiento de lenguaje natural (NLP)
 - 7.10.1. Conceptos básicos
 - 7.10.2. Creación del corpus
 - 7.10.3. Análisis descriptivo
 - 7.10.4. Introducción al análisis de sentimientos

Módulo 8. Las redes neuronales, base de *Deep Learning*

- 8.1. Aprendizaje Profundo
 - 8.1.1. Tipos de aprendizaje profundo
 - 8.1.2. Aplicaciones del aprendizaje profundo
 - 8.1.3. Ventajas y desventajas del aprendizaje profundo
- 8.2. Operaciones
 - 8.2.1. Suma
 - 8.2.2. Producto
 - 8.2.3. Traslado
- 8.3. Capas
 - 8.3.1. Capa de entrada
 - 8.3.2. Capa oculta
 - 8.3.3. Capa de salida
- 8.4. Unión de Capas y Operaciones
 - 8.4.1. Diseño de arquitecturas
 - 8.4.2. Conexión entre capas
 - 8.4.3. Propagación hacia adelante
- 8.5. Construcción de la primera red neuronal
 - 8.5.1. Diseño de la red
 - 8.5.2. Establecer los pesos
 - 8.5.3. Entrenamiento de la red
- 8.6. Entrenador y Optimizador
 - 8.6.1. Selección del optimizador
 - 8.6.2. Establecimiento de una función de pérdida
 - 8.6.3. Establecimiento de una métrica
- 8.7. Aplicación de los Principios de las Redes Neuronales
 - 8.7.1. Funciones de activación
 - 8.7.2. Propagación hacia atrás
 - 8.7.3. Ajuste de los parámetros
- 8.8. De las neuronas biológicas a las artificiales
 - 8.8.1. Funcionamiento de una neurona biológica
 - 8.8.2. Transferencia de conocimiento a las neuronas artificiales
 - 8.8.3. Establecer relaciones entre ambas
- 8.9. Implementación de MLP (Perceptrón multicapa) con Keras
 - 8.9.1. Definición de la estructura de la red
 - 8.9.2. Compilación del modelo
 - 8.9.3. Entrenamiento del modelo
- 8.10. Hiperparámetros de *Fine tuning* de Redes Neuronales
 - 8.10.1. Selección de la función de activación
 - 8.10.2. Establecer el *Learning rate*
 - 8.10.3. Ajuste de los pesos

Módulo 9. Entrenamiento de redes neuronales profundas

- 9.1. Problemas de Gradientes
 - 9.1.1. Técnicas de optimización de gradiente
 - 9.1.2. Gradientes Estocásticos
 - 9.1.3. Técnicas de inicialización de pesos
- 9.2. Reutilización de capas preentrenadas
 - 9.2.1. Entrenamiento de transferencia de aprendizaje
 - 9.2.2. Extracción de características
 - 9.2.3. Aprendizaje profundo
- 9.3. Optimizadores
 - 9.3.1. Optimizadores de descenso de gradiente estocástico
 - 9.3.2. Optimizadores Adam y *RMSprop*
 - 9.3.3. Optimizadores de momento
- 9.4. Programación de la tasa de aprendizaje
 - 9.4.1. Control de tasa de aprendizaje automático
 - 9.4.2. Ciclos de aprendizaje
 - 9.4.3. Términos de suavizado
- 9.5. Sobreajuste
 - 9.5.1. Validación cruzada
 - 9.5.2. Regularización
 - 9.5.3. Métricas de evaluación
- 9.6. Directrices Prácticas
 - 9.6.1. Diseño de modelos
 - 9.6.2. Selección de métricas y parámetros de evaluación
 - 9.6.3. Pruebas de hipótesis
- 9.7. *Transfer Learning*
 - 9.7.1. Entrenamiento de transferencia de aprendizaje
 - 9.7.2. Extracción de características
 - 9.7.3. Aprendizaje profundo
- 9.8. *Data Augmentation*
 - 9.8.1. Transformaciones de imagen
 - 9.8.2. Generación de datos sintéticos
 - 9.8.3. Transformación de texto

- 9.9. Aplicación Práctica de *Transfer Learning*
 - 9.9.1. Entrenamiento de transferencia de aprendizaje
 - 9.9.2. Extracción de características
 - 9.9.3. Aprendizaje profundo
- 9.10. Regularización
 - 9.10.1. L y L
 - 9.10.2. Regularización por máxima entropía
 - 9.10.3. *Dropout*

Módulo 10. Personalización de Modelos y entrenamiento con *TensorFlow*

- 10.1. *TensorFlow*
 - 10.1.1. Uso de la biblioteca *TensorFlow*
 - 10.1.2. Entrenamiento de modelos con *TensorFlow*
 - 10.1.3. Operaciones con gráficos en *TensorFlow*
- 10.2. *TensorFlow* y NumPy
 - 10.2.1. Entorno computacional NumPy para *TensorFlow*
 - 10.2.2. Utilización de los arrays NumPy con *TensorFlow*
 - 10.2.3. Operaciones NumPy para los gráficos de *TensorFlow*
- 10.3. Personalización de modelos y algoritmos de entrenamiento
 - 10.3.1. Construcción de modelos personalizados con *TensorFlow*
 - 10.3.2. Gestión de parámetros de entrenamiento
 - 10.3.3. Utilización de técnicas de optimización para el entrenamiento
- 10.4. Funciones y gráficos de *TensorFlow*
 - 10.4.1. Funciones con *TensorFlow*
 - 10.4.2. Utilización de gráficos para el entrenamiento de modelos
 - 10.4.3. Optimización de gráficos con operaciones de *TensorFlow*
- 10.5. Carga y preprocesamiento de datos con *TensorFlow*
 - 10.5.1. Carga de conjuntos de datos con *TensorFlow*
 - 10.5.2. Preprocesamiento de datos con *TensorFlow*
 - 10.5.3. Utilización de herramientas de *TensorFlow* para la manipulación de datos
- 10.6. La API *tfdata*
 - 10.6.1. Utilización de la API *tfdata* para el procesamiento de datos
 - 10.6.2. Construcción de flujos de datos con *tfdata*
 - 10.6.3. Uso de la API *tfdata* para el entrenamiento de modelos

- 10.7. El formato *TFRecord*
 - 10.7.1. Utilización de la API *TFRecord* para la serialización de datos
 - 10.7.2. Carga de archivos *TFRecord* con *TensorFlow*
 - 10.7.3. Utilización de archivos *TFRecord* para el entrenamiento de modelos
 - 10.8. Capas de preprocesamiento de Keras
 - 10.8.1. Utilización de la API de preprocesamiento de Keras
 - 10.8.2. Construcción de *pipelined* de preprocesamiento con Keras
 - 10.8.3. Uso de la API de preprocesamiento de Keras para el entrenamiento de modelos
 - 10.9. El proyecto *TensorFlow Datasets*
 - 10.9.1. Utilización de *TensorFlow Datasets* para la carga de datos
 - 10.9.2. Preprocesamiento de datos con *TensorFlow Datasets*
 - 10.9.3. Uso de *TensorFlow Datasets* para el entrenamiento de modelos
 - 10.10. Construcción de una Aplicación de Deep Learning con *TensorFlow*
 - 10.10.1. Aplicación Práctica
 - 10.10.2. Construcción de una aplicación de Deep Learning con *TensorFlow*
 - 10.10.3. Entrenamiento de un modelo con *TensorFlow*
 - 10.10.4. Utilización de la aplicación para la predicción de resultados
- Módulo 11. Deep Computer Vision con Redes Neuronales Convolucionales**
- 11.1. La Arquitectura *Visual Cortex*
 - 11.1.1. Funciones de la corteza visual
 - 11.1.2. Teorías de la visión computacional
 - 11.1.3. Modelos de procesamiento de imágenes
 - 11.2. Capas convolucionales
 - 11.2.1. Reutilización de pesos en la convolución
 - 11.2.2. Convolución D
 - 11.2.3. Funciones de activación
 - 11.3. Capas de agrupación e implementación de capas de agrupación con Keras
 - 11.3.1. *Pooling* y *Striding*
 - 11.3.2. *Flattening*
 - 11.3.3. Tipos de *Pooling*
 - 11.4. Arquitecturas CNN
 - 11.4.1. Arquitectura VGG
 - 11.4.2. Arquitectura *AlexNet*
 - 11.4.3. Arquitectura *ResNet*
 - 11.5. Implementación de una CNN *ResNet*- usando Keras
 - 11.5.1. Inicialización de pesos
 - 11.5.2. Definición de la capa de entrada
 - 11.5.3. Definición de la salida
 - 11.6. Uso de modelos preentrenados de Keras
 - 11.6.1. Características de los modelos preentrenados
 - 11.6.2. Usos de los modelos preentrenados
 - 11.6.3. Ventajas de los modelos preentrenados
 - 11.7. Modelos preentrenados para el aprendizaje por transferencia
 - 11.7.1. El Aprendizaje por transferencia
 - 11.7.2. Proceso de aprendizaje por transferencia
 - 11.7.3. Ventajas del aprendizaje por transferencia
 - 11.8. Clasificación y Localización en *Deep Computer Vision*
 - 11.8.1. Clasificación de imágenes
 - 11.8.2. Localización de objetos en imágenes
 - 11.8.3. Detección de objetos
 - 11.9. Detección de objetos y seguimiento de objetos
 - 11.9.1. Métodos de detección de objetos
 - 11.9.2. Algoritmos de seguimiento de objetos
 - 11.9.3. Técnicas de rastreo y localización
 - 11.10. Segmentación semántica
 - 11.10.1. Aprendizaje profundo para segmentación semántica
 - 11.10.2. Detección de bordes
 - 11.10.3. Métodos de segmentación basados en reglas

Módulo 12. Procesamiento del lenguaje natural (NLP) con Redes Naturales Recurrentes (RNN) y Atención

- 12.1. Generación de texto utilizando RNN
 - 12.1.1. Entrenamiento de una RNN para generación de texto
 - 12.1.2. Generación de lenguaje natural con RNN
 - 12.1.3. Aplicaciones de generación de texto con RNN
- 12.2. Creación del conjunto de datos de entrenamiento
 - 12.2.1. Preparación de los datos para el entrenamiento de una RNN
 - 12.2.2. Almacenamiento del conjunto de datos de entrenamiento
 - 12.2.3. Limpieza y transformación de los datos
 - 12.2.4. Análisis de Sentimiento
- 12.3. Clasificación de opiniones con RNN
 - 12.3.1. Detección de temas en los comentarios
 - 12.3.2. Análisis de sentimiento con algoritmos de aprendizaje profundo
- 12.4. Red de codificador-decodificador para la traducción automática neuronal
 - 12.4.1. Entrenamiento de una RNN para la traducción automática
 - 12.4.2. Uso de una red *encoder-decoder* para la traducción automática
 - 12.4.3. Mejora de la precisión de la traducción automática con RNN
- 12.5. Mecanismos de atención
 - 12.5.1. Aplicación de mecanismos de atención en RNN
 - 12.5.2. Uso de mecanismos de atención para mejorar la precisión de los modelos
 - 12.5.3. Ventajas de los mecanismos de atención en las redes neuronales
- 12.6. Modelos *Transformers*
 - 12.6.1. Uso de los modelos *Transformers* para procesamiento de lenguaje natural
 - 12.6.2. Aplicación de los modelos *Transformers* para visión
 - 12.6.3. Ventajas de los modelos *Transformers*
- 12.7. *Transformers* para visión
 - 12.7.1. Uso de los modelos *Transformers* para visión
 - 12.7.2. Preprocesamiento de los datos de imagen
 - 12.7.3. Entrenamiento de un modelo *Transformers* para visión
- 12.8. Librería de *Transformers* de *Hugging Face*
 - 12.8.1. Uso de la librería de *Transformers* de *Hugging Face*
 - 12.8.2. Aplicación de la librería de *Transformers* de *Hugging Face*
 - 12.8.3. Ventajas de la librería de *Transformers* de *Hugging Face*

- 12.9. Otras Librerías de *Transformers*. Comparativa
 - 12.9.1. Comparación entre las distintas librerías de *Transformers*
 - 12.9.2. Uso de las demás librerías de *Transformers*
 - 12.9.3. Ventajas de las demás librerías de *Transformers*
- 12.10. Desarrollo de una Aplicación de NLP con RNN y Atención. Aplicación Práctica
 - 12.10.1. Desarrollo de una aplicación de procesamiento de lenguaje natural con RNN y atención
 - 12.10.2. Uso de RNN, mecanismos de atención y modelos *Transformers* en la aplicación
 - 12.10.3. Evaluación de la aplicación práctica

Módulo 13. Autoencoders, GANs, y Modelos de Difusión

- 13.1. Representaciones de datos eficientes
 - 13.1.1. Reducción de dimensionalidad
 - 13.1.2. Aprendizaje profundo
 - 13.1.3. Representaciones compactas
- 13.2. Realización de PCA con un codificador automático lineal incompleto
 - 13.2.1. Proceso de entrenamiento
 - 13.2.2. Implementación en Python
 - 13.2.3. Utilización de datos de prueba
- 13.3. Codificadores automáticos apilados
 - 13.3.1. Redes neuronales profundas
 - 13.3.2. Construcción de arquitecturas de codificación
 - 13.3.3. Uso de la regularización
- 13.4. Autocodificadores convolucionales
 - 13.4.1. Diseño de modelos convolucionales
 - 13.4.2. Entrenamiento de modelos convolucionales
 - 13.4.3. Evaluación de los resultados
- 13.5. Eliminación de ruido de codificadores automáticos
 - 13.5.1. Aplicación de filtros
 - 13.5.2. Diseño de modelos de codificación
 - 13.5.3. Uso de técnicas de regularización
- 13.6. Codificadores automáticos dispersos
 - 13.6.1. Incrementar la eficiencia de la codificación
 - 13.6.2. Minimizando el número de parámetros
 - 13.6.3. Utilización de técnicas de regularización

- 13.7. Codificadores automáticos variacionales
 - 13.7.1. Utilización de optimización variacional
 - 13.7.2. Aprendizaje profundo no supervisado
 - 13.7.3. Representaciones latentes profundas
- 13.8. Generación de imágenes MNIST de moda
 - 13.8.1. Reconocimiento de patrones
 - 13.8.2. Generación de imágenes
 - 13.8.3. Entrenamiento de redes neuronales profundas
- 13.9. Redes adversarias generativas y modelos de difusión
 - 13.9.1. Generación de contenido a partir de imágenes
 - 13.9.2. Modelado de distribuciones de datos
 - 13.9.3. Uso de redes adversarias
- 13.10. Implementación de los Modelos
 - 13.10.1. Aplicación Práctica
 - 13.10.2. Implementación de los modelos
 - 13.10.3. Uso de datos reales
 - 13.10.4. Evaluación de los resultados

Módulo 14. Computación bioinspirada

- 14.1. Introducción a la computación bioinspirada
 - 14.1.1. Introducción a la computación bioinspirada
- 14.2. Algoritmos de adaptación social
 - 14.2.1. Computación bioinspirada basada en colonia de hormigas
 - 14.2.2. Variantes de los algoritmos de colonias de hormigas
 - 14.2.3. Computación basada en nubes de partículas
- 14.3. Algoritmos genéticos
 - 14.3.1. Estructura general
 - 14.3.2. Implementaciones de los principales operadores
- 14.4. Estrategias de exploración-explotación del espacio para algoritmos genéticos
 - 14.4.1. Algoritmo CHC
 - 14.4.2. Problemas multimodales

- 14.5. Modelos de computación evolutiva (I)
 - 14.5.1. Estrategias evolutivas
 - 14.5.2. Programación evolutiva
 - 14.5.3. Algoritmos basados en evolución diferencial
- 14.6. Modelos de computación evolutiva (II)
 - 14.6.1. Modelos de evolución basados en estimación de distribuciones (EDA)
 - 14.6.2. Programación genética
- 14.7. Programación evolutiva aplicada a problemas de aprendizaje
 - 14.7.1. Aprendizaje basado en reglas
 - 14.7.2. Métodos evolutivos en problemas de selección de instancias
- 14.8. Problemas multiobjetivo
 - 14.8.1. Concepto de dominancia
 - 14.8.2. Aplicación de algoritmos evolutivos a problemas multiobjetivo
- 14.9. Redes neuronales (I)
 - 14.9.1. Introducción a las redes neuronales
 - 14.9.2. Ejemplo práctico con redes neuronales
- 14.10. Redes neuronales (II)
 - 14.10.1. Casos de uso de las redes neuronales en la investigación médica
 - 14.10.2. Casos de uso de las redes neuronales en la economía
 - 14.10.3. Casos de uso de las redes neuronales en la visión artificial

Módulo 15. Inteligencia Artificial: Estrategias y aplicaciones

- 15.1. Servicios financieros
 - 15.1.1. Las implicaciones de la Inteligencia Artificial (IA) en los servicios financieros. Oportunidades y desafíos
 - 15.1.2. Casos de uso
 - 15.1.3. Riesgos potenciales relacionados con el uso de IA
 - 15.1.4. Potenciales desarrollos/usos futuros de la IA
- 15.2. Implicaciones de la Inteligencia Artificial en el servicio sanitario
 - 15.2.1. Implicaciones de la IA en el sector sanitario. Oportunidades y desafíos
 - 15.2.2. Casos de uso
- 15.3. Riesgos Relacionados con el uso de la IA en el servicio sanitario
 - 15.3.1. Riesgos potenciales relacionados con el uso de IA
 - 15.3.2. Potenciales desarrollos/usos futuros de la IA

- 15.4. Retail
 - 15.4.1. Implicaciones de la IA en Retail. Oportunidades y desafíos
 - 15.4.2. Casos de uso
 - 15.4.3. Riesgos potenciales relacionados con el uso de IA
 - 15.4.4. Potenciales desarrollos/usos futuros de la IA
- 15.5. Industria
 - 15.5.1. Implicaciones de la IA en la Industria. Oportunidades y desafíos
 - 15.5.2. Casos de uso
- 15.6. Riesgos potenciales relacionados con el uso de IA en la Industria
 - 15.6.1. Casos de uso
 - 15.6.2. Riesgos potenciales relacionados con el uso de IA
 - 15.6.3. Potenciales desarrollos/usos futuros de la IA
- 15.7. Administración Pública
 - 15.7.1. Implicaciones de la IA en la Administración Pública. Oportunidades y desafíos
 - 15.7.2. Casos de uso
 - 15.7.3. Riesgos potenciales relacionados con el uso de IA
 - 15.7.4. Potenciales desarrollos/usos futuros de la IA
- 15.8. Educación
 - 15.8.1. Implicaciones de la IA en la educación. Oportunidades y desafíos
 - 15.8.2. Casos de uso
 - 15.8.3. Riesgos potenciales relacionados con el uso de IA
 - 15.8.4. Potenciales desarrollos/usos futuros de la IA
- 15.9. Silvicultura y agricultura
 - 15.9.1. Implicaciones de la IA en la silvicultura y la agricultura. Oportunidades y desafíos
 - 15.9.2. Casos de uso
 - 15.9.3. Riesgos potenciales relacionados con el uso de IA
 - 15.9.4. Potenciales desarrollos/usos futuros de la IA
- 15.10 Recursos Humanos
 - 15.10.1. Implicaciones de la IA en los Recursos Humanos. Oportunidades y desafíos
 - 15.10.2. Casos de uso
 - 15.10.3. Riesgos potenciales relacionados con el uso de IA
 - 15.10.4. Potenciales desarrollos/usos futuros de la IA

Módulo 16. Mejora de la productividad en Desarrollo de software con IA

- 16.1. Preparar un entorno de desarrollo adecuado
 - 16.1.1. Selección de herramientas esenciales para desarrollo con IA
 - 16.1.2. Configuración de las herramientas elegidas
 - 16.1.3. Implementación de pipelines de CI/CD adaptados a proyectos con IA
 - 16.1.4. Gestión eficiente de dependencias y versiones en entornos de desarrollo
- 16.2. Extensiones imprescindibles de IA para Visual Studio Code
 - 16.2.1. Exploración y selección de extensiones de IA para Visual Studio Code
 - 16.2.2. Integración de herramientas de análisis estático y dinámico en el IDE
 - 16.2.3. Automatización de tareas repetitivas con extensiones específicas
 - 16.2.4. Personalización del entorno de desarrollo para mejorar la eficiencia
- 16.3. Diseño No-code de Interfaces de Usuario con Flutterflow
 - 16.3.1. Principios del diseño *No-code* y su aplicación en interfaces de usuario
 - 16.3.2. Incorporación de elementos de IA en el diseño visual de interfaces
 - 16.3.3. Herramientas y plataformas para la creación *No-code* de interfaces inteligentes
 - 16.3.4. Evaluación y mejora continua de interfaces *No-code* con IA
- 16.4. Optimización de código usando ChatGPT
 - 16.4.1. Identificar código duplicado
 - 16.4.2. Refactorizar
 - 16.4.3. Crear códigos legibles
 - 16.4.4. Entender lo que hace un código
 - 16.4.5. Mejora nombre de variables y funciones
 - 16.4.6. Creación de documentación automática
- 16.5. Gestión de repositorios con IA usando ChagGPT
 - 16.5.1. Automatización de procesos de control de versiones con técnicas de IA
 - 16.5.2. Detección de conflictos y resolución automática en entornos colaborativos
 - 16.5.3. Análisis predictivo de cambios y tendencias en repositorios de código
 - 16.5.4. Mejoras en la organización y categorización de repositorios mediante IA
- 16.6. Integración de IA en gestión con bases de datos con AskYourDatabase
 - 16.6.1. Optimización de consultas y rendimiento utilizando técnicas de IA
 - 16.6.2. Análisis predictivo de patrones de acceso a bases de datos
 - 16.6.3. Implementación de sistemas de recomendación para optimizar la estructura de la base de datos
 - 16.6.4. Monitoreo y detección proactiva de posibles problemas en bases de datos

- 16.7. Búsqueda de fallos y creación de test unitarios con IA usando ChatGPT
 - 16.7.1. Generación automática de casos de prueba mediante técnicas de IA
 - 16.7.2. Detección temprana de vulnerabilidades y errores utilizando análisis estático con IA
 - 16.7.3. Mejora de la cobertura de pruebas mediante la identificación de áreas críticas por IA
 - 16.8. *Pair Programming* con GitHub Copilot
 - 16.8.1. Integración y uso efectivo de GitHub Copilot en sesiones de *Pair Programming*
 - 16.8.2. Integración Mejoras en la comunicación y colaboración entre desarrolladores con GitHub Copilot
 - 16.8.3. Integración Estrategias para aprovechar al máximo las sugerencias de código generadas por GitHub Copilot
 - 16.8.4. Integración Casos de estudio y buenas prácticas en *Pair Programming* asistido por IA
 - 16.9. Traducción automática entre lenguajes de programación usando ChatGPT
 - 16.9.1. Herramientas y servicios de traducción automática específicos para lenguajes de programación
 - 16.9.2. Adaptación de algoritmos de traducción automática a contextos de desarrollo
 - 16.9.3. Mejora de la interoperabilidad entre diferentes lenguajes mediante traducción automática
 - 16.9.4. Evaluación y mitigación de posibles desafíos y limitaciones en la traducción automática
 - 16.10. Herramientas de IA recomendadas para mejorar la productividad
 - 16.10.1. Análisis comparativo de herramientas de IA para el desarrollo de software
 - 16.10.2. Integración de herramientas de IA en flujos de trabajo
 - 16.10.3. Automatización de tareas rutinarias con herramientas de IA
 - 16.10.4. Evaluación y selección de herramientas basada en el contexto y los requerimientos del proyecto
- Módulo 17. Arquitectura del software con IA**
- 17.1. Optimización y gestión del rendimiento en herramientas con IA con la ayuda de ChatGPT
 - 17.1.1. Análisis y perfilado de rendimiento en herramientas con IA
 - 17.1.2. Estrategias de optimización de algoritmos y modelos de IA
 - 17.1.3. Implementación de técnicas de *caching* y paralelización para mejorar el rendimiento
 - 17.1.4. Herramientas y metodologías para la monitorización continua del rendimiento en tiempo real
 - 17.2. Escalabilidad en aplicaciones de IA usando ChatGPT
 - 17.2.1. Diseño de arquitecturas escalables para aplicaciones de IA
 - 17.2.2. Implementación de técnicas de particionamiento y distribución de carga
 - 17.2.3. Manejo de flujos de trabajo y carga de trabajo en sistemas escalables
 - 17.2.4. Estrategias para la expansión horizontal y vertical en entornos con demanda variable
 - 17.3. Mantenibilidad de aplicaciones con IA usando ChatGPT
 - 17.3.1. Principios de diseño para facilitar la mantenibilidad en proyectos de IA
 - 17.3.2. Estrategias de documentación específicas para modelos y algoritmos de IA
 - 17.3.3. Implementación de pruebas unitarias y de integración para facilitar el mantenimiento
 - 17.3.4. Métodos para la refactorización y mejora continua en sistemas con componentes de IA
 - 17.4. Diseño de sistemas de gran escala
 - 17.4.1. Principios arquitectónicos para el diseño de sistemas de gran escala
 - 17.4.2. Descomposición de sistemas complejos en microservicios
 - 17.4.3. Implementación de patrones de diseño específicos para sistemas distribuidos
 - 17.4.4. Estrategias para la gestión de la complejidad en arquitecturas de gran escala con componentes de IA
 - 17.5. Almacenamiento de datos de gran escala para herramientas de IA
 - 17.5.1. Selección de tecnologías de almacenamiento de datos escalables
 - 17.5.2. Diseño de esquemas de bases de datos para el manejo eficiente de grandes volúmenes de datos
 - 17.5.3. Estrategias de particionamiento y replicación en entornos de almacenamiento de datos masivos
 - 17.5.4. Implementación de sistemas de gestión de datos para garantizar la integridad y disponibilidad en proyectos con IA
 - 17.6. Estructuras de datos Con IA usando ChatGPT
 - 17.6.1. Adaptación de estructuras de datos clásicas para su uso en algoritmos de IA
 - 17.6.2. Diseño y optimización de estructuras de datos específicas con ChatGPT
 - 17.6.3. Integración de estructuras de datos eficientes en sistemas con procesamiento intensivo de datos
 - 17.6.4. Estrategias para la manipulación y almacenamiento de datos en tiempo real en estructuras de datos con IA

- 17.7. Algoritmos de programación para productos con IA
 - 17.7.1. Desarrollo e implementación de algoritmos específicos para aplicaciones con IA
 - 17.7.2. Estrategias de selección de algoritmos según el tipo de problema y los requisitos del producto
 - 17.7.3. Adaptación de algoritmos clásicos para su integración en sistemas de inteligencia artificial
 - 17.7.4. Evaluación y comparación de rendimiento entre diferentes algoritmos en contextos de desarrollo con IA
- 17.8. Patrones diseño para desarrollo con IA
 - 17.8.1. Identificación y aplicación de patrones de diseño comunes en proyectos con componentes de IA
 - 17.8.2. Desarrollo de patrones específicos para la integración de modelos y algoritmos en sistemas existentes
 - 17.8.3. Estrategias de implementación de patrones para mejorar la reusabilidad y mantenibilidad en proyectos de IA
 - 17.8.4. Casos de estudio y buenas prácticas en la aplicación de patrones de diseño en arquitecturas con IA
- 17.9. Implementación de clean architecture usando ChatGPT
 - 17.9.1. Principios y conceptos fundamentales de *Clean Architecture*
 - 17.9.2. Adaptación de *Clean Architecture* a proyectos con componentes de IA
 - 17.9.3. Implementación de capas y dependencias en sistemas con arquitectura limpia
 - 17.9.4. Beneficios y desafíos de la implementación de *Clean Architecture* en el desarrollo de software con IA
- 17.10. Desarrollo de software seguro en aplicaciones web con DeepCode
 - 17.10.1. Principios de seguridad en el desarrollo de software con componentes de IA
 - 17.10.2. Identificación y mitigación de posibles vulnerabilidades en modelos y algoritmos de IA
 - 17.10.3. Implementación de prácticas de desarrollo seguro en aplicaciones web con funcionalidades de Inteligencia Artificial
 - 17.10.4. Estrategias para la protección de datos sensibles y la prevención de ataques en proyectos con IA

Módulo 18. Proyectos web con IA

- 18.1. Preparación del Entorno de Trabajo para Desarrollo Web con IA
 - 18.1.1. Configuración de entornos de desarrollo web para proyectos con inteligencia artificial
 - 18.1.2. Selección y preparación de herramientas esenciales para el desarrollo web con IA
 - 18.1.3. Integración de bibliotecas y *frameworks* específicos para proyectos web con inteligencia artificial
 - 18.1.4. Implementación de buenas prácticas en la configuración de entornos de desarrollo colaborativos
- 18.2. Creación de *Workspace* para Proyectos de IA
 - 18.2.1. Diseño y organización efectiva de *workspaces* para proyectos web con componentes de inteligencia artificial
 - 18.2.2. Uso de herramientas de gestión de proyectos y control de versiones en el *workspace*
 - 18.2.3. Estrategias para la colaboración y comunicación eficientes en el equipo de desarrollo
 - 18.2.4. Adaptación del *workspace* a las necesidades específicas de proyectos web con IA
- 18.3. Patrones de Diseño en Productos con GitHub Copilot
 - 18.3.1. Identificación y aplicación de patrones de diseño comunes en interfaces de usuario con elementos de inteligencia artificial
 - 18.3.2. Desarrollo de patrones específicos para mejorar la experiencia de usuario en proyectos web con IA
 - 18.3.3. Integración de patrones de diseño en la arquitectura general de proyectos web con Inteligencia Artificial
 - 18.3.4. Evaluación y selección de patrones de diseño adecuados según el contexto del proyecto
- 18.4. Desarrollo Frontend con GitHub Copilot
 - 18.4.1. Integración de modelos de IA en la capa de presentación de proyectos web
 - 18.4.2. Desarrollo de interfaces de usuario adaptativas con elementos de inteligencia artificial
 - 18.4.3. Implementación de funcionalidades de procesamiento de lenguaje natural (PLN) en el Frontend
 - 18.4.4. Estrategias para la optimización del rendimiento en el desarrollo Frontend con IA

- 18.5. Creación de Base de Datos usando GitHub Copilot
 - 18.5.1. Selección de tecnologías de bases de datos para proyectos web con inteligencia artificial
 - 18.5.2. Diseño de esquemas de bases de datos para almacenar y gestionar datos relacionados con IA
 - 18.5.3. Implementación de sistemas de almacenamiento eficientes para grandes volúmenes de datos generados por modelos de IA
 - 18.5.4. Estrategias para la seguridad y protección de datos sensibles en bases de datos de proyectos web con IA
- 18.6. Desarrollo Backend con GitHub Copilot
 - 18.6.1. Integración de servicios y modelos de IA en la lógica de negocio del Backend
 - 18.6.2. Desarrollo de APIs y endpoints específicos para la comunicación entre el Frontend y los componentes de IA
 - 18.6.3. Implementación de lógica de procesamiento de datos y toma de decisiones en el Backend con Inteligencia Artificial
 - 18.6.4. Estrategias para la escalabilidad y rendimiento en el desarrollo Backend de proyectos web con IA
- 18.7. Optimizar el Proceso de Despliegue de Tu Web
 - 18.7.1. Automatización de procesos de construcción y despliegue de proyectos web con ChatGPT
 - 18.7.2. Implementación de pipelines de CI/CD adaptados a aplicaciones web con GitHub Copilot
 - 18.7.3. Estrategias para la gestión eficiente de versiones y actualizaciones en despliegues continuos
 - 18.7.4. Monitoreo y análisis post-despliegue para la mejora continua del proceso
- 18.8. IA en la Computación en la Nube
 - 18.8.1. Integración de servicios de inteligencia artificial en plataformas de computación en la nube
 - 18.8.2. Desarrollo de soluciones escalables y distribuidas utilizando servicios de nube con capacidades de IA
 - 18.8.3. Estrategias para el manejo eficiente de recursos y costos en entornos de nube con aplicaciones web con IA
 - 18.8.4. Evaluación y comparación de proveedores de servicios en la nube para proyectos web con Inteligencia Artificial

- 18.9. Creación de un Proyecto con IA para Entornos LAMP con la ayuda de ChatGPT
 - 18.9.1. Adaptación de proyectos web basados en la pila LAMP para incluir componentes de Inteligencia Artificial
 - 18.9.2. Integración de bibliotecas y *frameworks* específicos de IA en entornos LAMP
 - 18.9.3. Desarrollo de funcionalidades de IA que complementan la arquitectura LAMP tradicional
 - 18.9.4. Estrategias para la optimización y mantenimiento en proyectos web con IA en entornos LAMP
- 18.10. Creación de un Proyecto con IA para Entornos MEVN usando ChatGPT
 - 18.10.1. Integración de tecnologías y herramientas de la pila MEVN con componentes de Inteligencia Artificial
 - 18.10.2. Desarrollo de aplicaciones web modernas y escalables en entornos MEVN con capacidades de IA
 - 18.10.3. Implementación de funcionalidades de procesamiento de datos y aprendizaje automático en proyectos MEVN
 - 18.10.4. Estrategias para la mejora del rendimiento y la seguridad en aplicaciones web con IA en entornos MEVN

Módulo 19. Aplicaciones móviles con IA

- 19.1. Preparación de Entorno de Trabajo para Desarrollo Móvil con IA
 - 19.1.1. Configuración de entornos de desarrollo móvil para proyectos con Inteligencia Artificial
 - 19.1.2. Selección y preparación de herramientas específicas para el desarrollo de aplicaciones móviles con IA
 - 19.1.3. Integración de bibliotecas y *frameworks* de IA en entornos de desarrollo móvil
 - 19.1.4. Configuración de emuladores y dispositivos reales para pruebas de aplicaciones móviles con componentes de inteligencia artificial
- 19.2. Creación de un *Workspace* con GitHub Copilot
 - 19.2.1. Integración de GitHub Copilot en entornos de desarrollo móvil
 - 19.2.2. Uso efectivo de GitHub Copilot para la generación de código en proyectos con IA
 - 19.2.3. Estrategias para la colaboración entre desarrolladores al utilizar GitHub Copilot en el *workspace*
 - 19.2.4. Buenas prácticas y limitaciones en el uso de GitHub Copilot en el desarrollo de aplicaciones móviles con IA

- 19.3. Configuración de Firebase
 - 19.3.1. Configuración inicial de un proyecto en Firebase para el desarrollo móvil
 - 19.3.2. Integración de Firebase en aplicaciones móviles con funcionalidades de Inteligencia Artificial
 - 19.3.3. Uso de servicios de Firebase como base de datos, autenticación y notificaciones en proyectos con IA
 - 19.3.4. Estrategias para la gestión de datos y eventos en tiempo real en aplicaciones móviles con Firebase
- 19.4. Conceptos de Clean Architecture, DataSources, Repositories
 - 19.4.1. Principios fundamentales de Clean Architecture en el desarrollo móvil con IA
 - 19.4.2. Implementación de capas de DataSources y Repositories con GitHub Copilot
 - 19.4.3. Diseño y estructuración de componentes en proyectos móvil con GitHub Copilot
 - 19.4.4. Beneficios y desafíos de la implementación de *Clean Architecture* en aplicaciones móviles con IA
- 19.5. Creación de Pantalla de Autenticación con GitHub Copilot
 - 19.5.1. Diseño y desarrollo de interfaces de usuario para pantallas de autenticación en aplicaciones móviles con IA
 - 19.5.2. Integración de servicios de autenticación con Firebase en la pantalla de inicio de sesión
 - 19.5.3. Uso de técnicas de seguridad y protección de datos en la pantalla de autenticación
 - 19.5.4. Personalización y adaptación de la experiencia de usuario en la pantalla de autenticación
- 19.6. Creación de Dashboard y Navegación con GitHub Copilot
 - 19.6.1. Diseño y desarrollo de *Dashboards* con elementos de Inteligencia Artificial
 - 19.6.2. Implementación de sistemas de navegación eficientes en aplicaciones móviles con IA
 - 19.6.3. Integración de funcionalidades de IA en el *Dashboard* para mejorar la experiencia del usuario
- 19.7. Creación de Pantalla con Listado usando GitHub Copilot
 - 19.7.1. Desarrollo de interfaces de usuario para pantallas con listados en aplicaciones móviles con IA
 - 19.7.2. Integración de algoritmos de recomendación y filtrado en la pantalla de listado
 - 19.7.3. Uso de patrones de diseño para la presentación efectiva de datos en el listado
 - 19.7.4. Estrategias para la carga eficiente de datos en tiempo real en la pantalla con listado
- 19.8. Creación de Pantalla de Detalle con GitHub Copilot
 - 19.8.1. Diseño y desarrollo de interfaces de usuario detalladas para la presentación de información específica
 - 19.8.2. Integración de funcionalidades de IA para enriquecer la pantalla de detalle
 - 19.8.3. Implementación de interacciones y animaciones en la pantalla de detalle
 - 19.8.4. Estrategias para la optimización del rendimiento en la carga y visualización de detalles en aplicaciones móviles con IA
- 19.9. Creación de Pantalla de Settings con GitHub Copilot
 - 19.9.1. Desarrollo de interfaces de usuario para configuración y ajustes en aplicaciones móviles con IA
 - 19.9.2. Integración de ajustes personalizados relacionados con componentes de inteligencia artificial
 - 19.9.3. Implementación de opciones de personalización y preferencias en la pantalla de configuración
 - 19.9.4. Estrategias para la usabilidad y claridad en la presentación de opciones en la pantalla de *settings*
- 19.10. Crear Iconos, *Splash* y Recursos Gráficos para Tu App con IA
 - 19.10.1. Diseño y creación de iconos atractivos para representar la aplicación móvil con IA
 - 19.10.2. Desarrollo de pantallas de inicio (*splash*) con elementos visuales impactantes
 - 19.10.3. Selección y adaptación de recursos gráficos que mejoren la estética de la aplicación móvil
 - 19.10.4. Estrategias para la consistencia y branding visual en los elementos gráficos de la aplicación con IA

Módulo 20. IA para QA Testing

- 20.1. Ciclo de Vida de *Testing*
 - 20.1.1. Descripción y comprensión del ciclo de vida de *testing* en el desarrollo de software
 - 20.1.2. Fases del ciclo de vida de *testing* y su importancia en el aseguramiento de la calidad
 - 20.1.3. Integración de la inteligencia artificial en diferentes etapas del ciclo de vida de *testing*
 - 20.1.4. Estrategias para la mejora continua del ciclo de vida de *testing* mediante el uso de IA

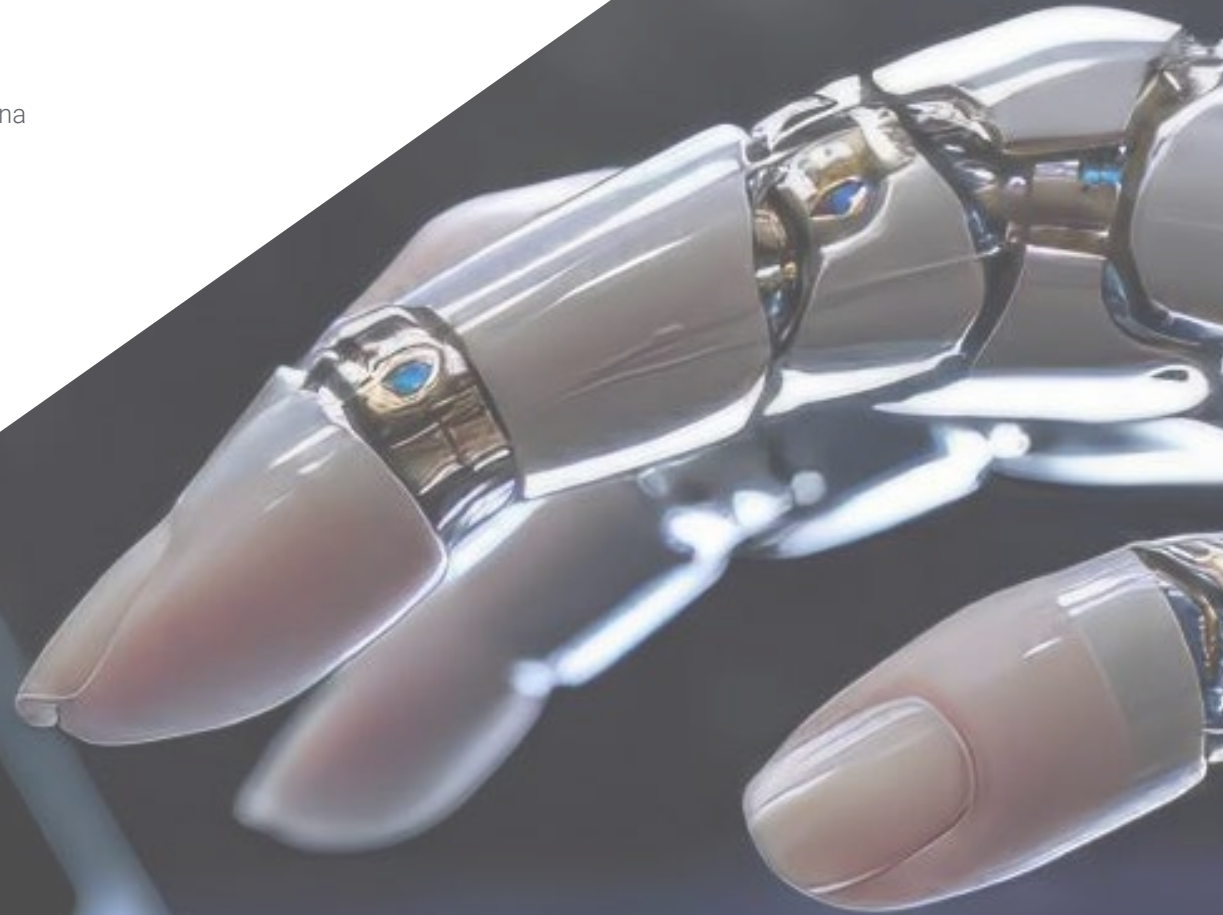
- 20.2. Test Cases y Detección de Bugs con ayuda de ChatGPT
 - 20.2.1. Diseño y escritura efectiva de casos de prueba en el contexto de QA *Testing*
 - 20.2.2. Identificación de bugs y errores durante la ejecución de casos de prueba
 - 20.2.3. Aplicación de técnicas de detección temprana de bugs mediante análisis estático
 - 20.2.4. Uso de herramientas de inteligencia artificial para la identificación automática de bugs en test cases
- 20.3. Tipos de *Testing*
 - 20.3.1. Exploración de diferentes tipos de *testing* en el ámbito de QA
 - 20.3.2. Pruebas unitarias, integración, funcionales, y de aceptación: características y aplicaciones
 - 20.3.3. Estrategias para la selección y combinación adecuada de tipos de testing en proyectos con ChatGPT
 - 20.3.4. Adaptación de tipos de testing convencionales a proyectos con ChatGPT
- 20.4. Crear un Plan de Pruebas
 - 20.4.1. Diseño y estructuración de un plan de pruebas integral
 - 20.4.2. Identificación de requisitos y escenarios de prueba en proyectos con IA
 - 20.4.3. Estrategias para la planificación de pruebas manuales y automatizadas
 - 20.4.4. Evaluación y ajuste continuo del plan de pruebas en función del desarrollo del proyecto
- 20.5. Detección y Reportar *Bugs* con IA
 - 20.5.1. Implementación de técnicas de detección automática de bugs mediante algoritmos de aprendizaje automático
 - 20.5.2. Uso de ChatGPT para el análisis dinámico de código en busca de posibles errores
 - 20.5.3. Estrategias para la generación automática de informes detallados sobre bugs detectados usando ChatGPT
 - 20.5.4. Colaboración efectiva entre equipos de desarrollo y QA en la gestión de bugs identificados por IA
- 20.6. Creación de Pruebas Automatizadas con IA
 - 20.6.1. Desarrollo de scripts de prueba automatizados para proyectos usando ChatGPT
 - 20.6.2. Integración de herramientas de automatización de pruebas basadas en IA
 - 20.6.3. Uso de ChatGPT para la generación dinámica de casos de prueba automatizados
 - 20.6.4. Estrategias para la ejecución eficiente y mantenimiento de pruebas automatizadas en proyectos con IA
- 20.7. *API Testing*
 - 20.7.1. Conceptos fundamentales de *API testing* y su importancia en QA
 - 20.7.2. Desarrollo de pruebas para la verificación de APIs en entornos usando ChatGPT
 - 20.7.3. Estrategias para la validación de datos y resultados en *API testing* con ChatGPT
 - 20.7.4. Uso de herramientas específicas para el *testing* de APIs en proyectos con inteligencia artificial
- 20.8. Herramientas de IA para *Web Testing*
 - 20.8.1. Exploración de herramientas de inteligencia artificial para la automatización de pruebas en entornos web
 - 20.8.2. Integración de tecnologías de reconocimiento de elementos y análisis visual en *web testing*
 - 20.8.3. Estrategias para la detección automática de cambios y problemas de rendimiento en aplicaciones web usando ChatGPT
 - 20.8.4. Evaluación de herramientas específicas para la mejora de la eficiencia en el *web testing* con IA
- 20.9. *Mobile Testing* Mediante IA
 - 20.9.1. Desarrollo de estrategias de *testing* para aplicaciones móviles con componentes de inteligencia artificial
 - 20.9.2. Integración de herramientas de *testing* específicas para plataformas móviles basadas en IA
 - 20.9.3. Uso de ChatGPT para la detección de problemas en el rendimiento de aplicaciones móviles
 - 20.9.4. Estrategias para la validación de interfaces y funciones específicas de aplicaciones móviles mediante IA
- 20.10. Herramientas de QA con IA
 - 20.10.1. Exploración de herramientas y plataformas de QA que incorporan funcionalidades de Inteligencia Artificial
 - 20.10.2. Evaluación de herramientas para la gestión y ejecución eficiente de pruebas en proyectos con IA
 - 20.10.3. Uso de ChatGPT para la generación y optimización de casos de prueba
 - 20.10.4. Estrategias para la selección y adopción efectiva de herramientas de QA con capacidades de IA

06

Metodología

Este programa de capacitación ofrece una forma diferente de aprender. Nuestra metodología se desarrolla a través de un modo de aprendizaje de forma cíclica: **el Relearning**.

Este sistema de enseñanza es utilizado, por ejemplo, en las facultades de medicina más prestigiosas del mundo y se ha considerado uno de los más eficaces por publicaciones de gran relevancia como el ***New England Journal of Medicine***.





“

Descubre el Relearning, un sistema que abandona el aprendizaje lineal convencional para llevarte a través de sistemas cíclicos de enseñanza: una forma de aprender que ha demostrado su enorme eficacia, especialmente en las materias que requieren memorización”

Estudio de Caso para contextualizar todo el contenido

Nuestro programa ofrece un método revolucionario de desarrollo de habilidades y conocimientos. Nuestro objetivo es afianzar competencias en un contexto cambiante, competitivo y de alta exigencia.

“

Con TECH Universidad FUNDEPOS podrás experimentar una forma de aprender que está moviendo los cimientos de las universidades tradicionales de todo el mundo”



Accederás a un sistema de aprendizaje basado en la reiteración, con una enseñanza natural y progresiva a lo largo de todo el temario.



El alumno aprenderá, mediante actividades colaborativas y casos reales, la resolución de situaciones complejas en entornos empresariales reales.

Un método de aprendizaje innovador y diferente

El presente programa de TECH Universidad FUNDEPOS es una enseñanza intensiva, creada desde 0, que propone los retos y decisiones más exigentes en este campo, ya sea en el ámbito nacional o internacional. Gracias a esta metodología se impulsa el crecimiento personal y profesional, dando un paso decisivo para conseguir el éxito. El método del caso, técnica que sienta las bases de este contenido, garantiza que se sigue la realidad económica, social y profesional más vigente.

“ *Nuestro programa te prepara para afrontar nuevos retos en entornos inciertos y lograr el éxito en tu carrera*”

El método del caso ha sido el sistema de aprendizaje más utilizado por las mejores escuelas de Informática del mundo desde que éstas existen. Desarrollado en 1912 para que los estudiantes de Derecho no solo aprendiesen las leyes a base de contenidos teóricos, el método del caso consistió en presentarles situaciones complejas reales para que tomaran decisiones y emitieran juicios de valor fundamentados sobre cómo resolverlas. En 1924 se estableció como método estándar de enseñanza en Harvard.

Ante una determinada situación, ¿qué debería hacer un profesional? Esta es la pregunta a la que te enfrentamos en el método del caso, un método de aprendizaje orientado a la acción. A lo largo del curso, los estudiantes se enfrentarán a múltiples casos reales. Deberán integrar todos sus conocimientos, investigar, argumentar y defender sus ideas y decisiones.

Relearning Methodology

TECH Universidad FUNDEPOS aúna de forma eficaz la metodología del Estudio de Caso con un sistema de aprendizaje 100% online basado en la reiteración, que combina elementos didácticos diferentes en cada lección.

Potenciamos el Estudio de Caso con el mejor método de enseñanza 100% online: el Relearning.

En 2019 obtuvimos los mejores resultados de aprendizaje de todas las universidades online en español en el mundo.

En TECH Universidad FUNDEPOS aprenderás con una metodología vanguardista concebida para capacitar a los directivos del futuro. Este método, a la vanguardia pedagógica mundial, se denomina Relearning.

Nuestra universidad es la única en habla hispana licenciada para emplear este exitoso método. En 2019, conseguimos mejorar los niveles de satisfacción global de nuestros alumnos (calidad docente, calidad de los materiales, estructura del curso, objetivos...) con respecto a los indicadores de la mejor universidad online en español.



En nuestro programa, el aprendizaje no es un proceso lineal, sino que sucede en espiral (aprender, desaprender, olvidar y reaprender). Por eso, se combinan cada uno de estos elementos de forma concéntrica. Con esta metodología se han capacitado más de 650.000 graduados universitarios con un éxito sin precedentes en ámbitos tan distintos como la bioquímica, la genética, la cirugía, el derecho internacional, las habilidades directivas, las ciencias del deporte, la filosofía, el derecho, la ingeniería, el periodismo, la historia o los mercados e instrumentos financieros. Todo ello en un entorno de alta exigencia, con un alumnado universitario de un perfil socioeconómico alto y una media de edad de 43,5 años.

El Relearning te permitirá aprender con menos esfuerzo y más rendimiento, implicándote más en tu capacitación, desarrollando el espíritu crítico, la defensa de argumentos y el contraste de opiniones: una ecuación directa al éxito.

A partir de la última evidencia científica en el ámbito de la neurociencia, no solo sabemos organizar la información, las ideas, las imágenes y los recuerdos, sino que sabemos que el lugar y el contexto donde hemos aprendido algo es fundamental para que seamos capaces de recordarlo y almacenarlo en el hipocampo, para retenerlo en nuestra memoria a largo plazo.

De esta manera, y en lo que se denomina Neurocognitive context-dependent e-learning, los diferentes elementos de nuestro programa están conectados con el contexto donde el participante desarrolla su práctica profesional.



Este programa ofrece los mejores materiales educativos, preparados a conciencia para los profesionales:



Material de estudio

Todos los contenidos didácticos son creados por los especialistas que van a impartir el curso, específicamente para él, de manera que el desarrollo didáctico sea realmente específico y concreto.

Estos contenidos son aplicados después al formato audiovisual, para crear el método de trabajo online de TECH Universidad FUNDEPOS. Todo ello, con las técnicas más novedosas que ofrecen piezas de gran calidad en todos y cada uno los materiales que se ponen a disposición del alumno.



Clases magistrales

Existe evidencia científica sobre la utilidad de la observación de terceros expertos.

El denominado Learning from an Expert afianza el conocimiento y el recuerdo, y genera seguridad en las futuras decisiones difíciles.



Prácticas de habilidades y competencias

Realizarán actividades de desarrollo de competencias y habilidades específicas en cada área temática. Prácticas y dinámicas para adquirir y desarrollar las destrezas y habilidades que un especialista precisa desarrollar en el marco de la globalización que vivimos.



Lecturas complementarias

Artículos recientes, documentos de consenso y guías internacionales, entre otros. En la biblioteca virtual de TECH Universidad FUNDEPOS el estudiante tendrá acceso a todo lo que necesita para completar su capacitación.





Case studies

Completarán una selección de los mejores casos de estudio elegidos expresamente para esta titulación. Casos presentados, analizados y tutorizados por los mejores especialistas del panorama internacional.



Resúmenes interactivos

El equipo de TECH Universidad FUNDEPOS presenta los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audios, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

Este exclusivo sistema educativo para la presentación de contenidos multimedia fue premiado por Microsoft como "Caso de éxito en Europa".



Testing & Retesting

Se evalúan y reevalúan periódicamente los conocimientos del alumno a lo largo del programa, mediante actividades y ejercicios evaluativos y autoevaluativos para que, de esta manera, el estudiante compruebe cómo va consiguiendo sus metas.



07

Titulación

El Máster Título Propio en Inteligencia Artificial en la Programación garantiza, además de la capacitación más rigurosa y actualizada, el acceso a dos diplomas de Máster Propio, uno expedido por TECH Universidad Tecnológica y otro expedido por Universidad FUNDEPOS.



“

Supera con éxito este programa y recibe tu titulación universitaria sin desplazamientos ni farragosos trámites”

El programa del **Máster Título Propio en Inteligencia Artificial en la Programación** es el más completo del panorama académico actual. A su egreso, el estudiante recibirá un diploma universitario emitido por TECH Universidad Tecnológica, y otro por Universidad FUNDEPOS.

Estos títulos de formación permanente y actualización profesional de TECH Universidad Tecnológica y Universidad FUNDEPOS garantizan la adquisición de competencias en el área de conocimiento, otorgando un alto valor curricular al estudiante que supere las evaluaciones y acredite el programa tras cursarlo en su totalidad.

Este doble reconocimiento, de dos destacadas instituciones universitarias, suponen una doble recompensa a una formación integral y de calidad, asegurando que el estudiante obtenga una certificación reconocida tanto a nivel nacional como internacional. Este mérito académico le posicionará como un profesional altamente capacitado y preparado para enfrentar los retos y demandas en su área profesional.

Título: **Máster Título Propio en Inteligencia Artificial en la Programación**

N.º Horas: **2.250 h.**



Máster Título Propio en Inteligencia Artificial en la Programación

Distribución General del Plan de Estudios		Distribución General del Plan de Estudios			
Tipo de materia	Horas	Curso	Materia	Horas	Carácter
Obligatoria (OB)	2.250	1º	Fundamentos de la Inteligencia Artificial	112	OB
Optativa (OP)	0	1º	Tipos y ciclo de vida del dato	112	OB
Prácticas Externas (PE)	0	1º	El dato en la Inteligencia Artificial	112	OB
Trabajo Fin de Máster (TFM)	0	1º	Minería de datos: Selección, preprocesamiento y transformación	112	OB
	Total 2.250	1º	Algoritmia y complejidad en Inteligencia Artificial	112	OB
		1º	Sistemas inteligentes	112	OB
		1º	Aprendizaje automático y minería de datos	112	OB
		1º	Las redes neuronales, base de Deep Learning	112	OB
		1º	Entrenamiento de redes neuronales profundas	112	OB
		1º	Personalización de modelos y entrenamiento con TensorFlow	112	OB
		1º	Deep Computer Vision con Redes Neuronales Convolucionales	113	OB
		1º	Procesamiento del lenguaje natural (NLP) con Redes Naturales	113	OB
		1º	Recurrentes (RNN) y Atención	113	OB
		1º	Autoencoders, GANs, y modelos de difusión	113	OB
		1º	Computación bioinspirada	113	OB
		1º	Inteligencia Artificial: Estrategias y aplicaciones	113	OB
		1º	Mejora de la productividad en desarrollo de software con IA	113	OB
		1º	Arquitectura del software para QA Testing	113	OB
		1º	Proyectos web con IA	113	OB
		1º	Aplicaciones móviles con IA	113	OB
		1º	IA para QA Testing	113	OB

*Apostilla de la Haya. En caso de que el alumno solicite que su diploma de TECH Universidad Tecnológica recabe la Apostilla de La Haya, TECH Universidad FUNDEPOS realizará las gestiones oportunas para su obtención, con un coste adicional.

salud futuro
confianza personas
educación información tutores
garantía acreditación enseñanza
instituciones tecnología aprendizaje
comunidad compromiso
atención personalizada innovación
conocimiento presente calidad
desarrollo web formación
aula virtual idiomas

tech universidad
FUNDEPOS

Máster Título Propio Inteligencia Artificial en la Programación

- » Modalidad: online
- » Duración: 12 meses
- » Titulación: TECH Universidad FUNDEPOS
- » Horario: a tu ritmo
- » Exámenes: online

Máster Título Propio Inteligencia Artificial en la Programación