



GUÍA DOCENTE

INTELIGENCIA ARTIFICIAL

DOBLE GRADO EN FÍSICA COMPUTACIONAL E INGENIERÍA DEL SOFTWARE

MODALIDAD: PRESENCIAL

CURSO ACADÉMICO: 2026-2027

Denominación de la asignatura:	Inteligencia Artificial
Titulación:	Doble grado en Física Computacional e Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Programación
Curso:	4º
Cuatrimestre:	1
Carácter:	OB
Créditos ECTS:	6
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	Carl McBride Ellis/ carl.mcbride@u-tad.com
Página Web:	http://www.u-tad.com/

DESCRIPCIÓN DE LA ASIGNATURA

Descripción de la materia

Esta asignatura pertenece a la materia de programación. Esta materia se dedica al estudio de las técnicas y los lenguajes de programación en los que se fundamentarán los estudios del grado de ingeniería del software.

Descripción de la asignatura

Esta asignatura se dedica al estudio de ese concepto tan extenso, multidisciplinar y en algún caso incluso polémico como es la Inteligencia Artificial (IA). La IA se puede definir como el conjunto de capacidades que puedan asimilarse a las humanas pero desarrolladas por máquinas (por ahora, solo “computadoras” tal como las conocemos). Se trata por tanto de desarrollar sistemas que realicen funciones “parecidas” a las humanas como aprender, solucionar problemas, comunicarse con lenguaje natural, reconocer e identificar personas u otros componentes de una imagen y una cada vez más extensa lista de aplicaciones. Algunas existentes, otras imaginadas, algunas de las cuales probablemente posibles en un corto plazo de tiempo, porque si hay algo que resaltar sobre este campo tecnológico es la enorme velocidad de sus desarrollos.

El enorme desarrollo actual se debe, sin duda, a la consolidación y popularización del paradigma de “aprendizaje profundo” (Deep Learning), ayudado por la disponibilidad de hardware específico para el cálculo

de modelos enormes y que está revolucionando áreas tradicionales de computación como son la visión artificial, el procesamiento de lenguaje natural e, incluso, la creación artística

El principal esfuerzo de la asignatura será proporcionar una introducción tan completa como sea posible para el aprovechamiento de estas tecnologías en cualquier ámbito del desarrollo software, así como conocer algunos de sus puntos débiles (que por supuesto tienen) y precauciones importantes sobre la puesta en marcha de este tipo de sistemas en contextos concretos.

Estudiaremos las bases teóricas y tecnológicas que han posibilitado el enorme desarrollo de este campo sobre todo en los últimos 20 años, las aplicaciones tanto existentes como realistas en el corto plazo, en sectores como el reconocimiento de imágenes, procesamiento del lenguaje natural (NLP), juegos, y un largo etcétera.

Con este objetivo, nos concentraremos desde los primeros momentos en los últimos desarrollos de lo que se conoce como “Deep Learning”, explorando una amplia variedad (dentro de las posibilidades del curso) de enfoques dentro de este campo y siempre con casos concretos de aplicación.

El enfoque del curso será por tanto principalmente práctico, basado en el desarrollo exhaustivo de ejemplos sobre el entorno de desarrollo más habitual en estos momentos, que es Python y sus librerías especializadas.

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE DE LA MATERIA

Competencias (genéricas, específicas y transversales)

COMPETENCIAS BÁSICAS Y GENERALES:

CG1 - Capacidad para entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas

CG3 - Conocimiento de los fundamentos científicos aplicables a la resolución de problemas informáticos

CG4 - Capacidad para simplificar y optimizar los sistemas informáticos atendiendo a la comprensión de su complejidad

CG9 - Capacidad para aprender, modificar y producir nuevas tecnologías informáticas

CG10 - Uso de técnicas creativas para la realización de proyectos informáticos

CB1 - Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio

CB2 - Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio

CB3 - Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

CB4 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado

CB5 - Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores

con un alto grado de autonomía

COMPETENCIAS ESPECÍFICAS

CE1 - Conocimiento de la estructura de las computadoras, de los conceptos de codificación, manipulación, tratamiento de la información y uso de lenguajes de bajo nivel

CE7 - Conocimiento de los principales tipos de estructuras de datos y utilización de las librerías y de los técnicas algorítmicas asociadas a dichas estructuras junto con los órdenes de complejidad que caracterizan a dichas técnicas

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación

CE9 - Conocimiento de las estructuras de control, variables, sintaxis de programación y gestión del uso de la memoria de manera eficaz en el desarrollo de una aplicación informática

CE10 - Capacidad para manejar un gestor de versiones de código y generar la documentación de una aplicación de forma automática.

CE15 - Conocimiento de la tolerancia a los fallos, la adaptabilidad, el balance de carga y la predictividad del sistema para el desarrollo de aplicaciones distribuidas

CE17 - Conocimiento de las características de paralelización de tarjetas gráficas y de arquitecturas de altas prestaciones para el desarrollo de aplicaciones.

CE20 - Capacidad para testar el funcionamiento y funcionalidad de una aplicación informática, elaborando planes de pruebas y empleando técnicas de diseño y programación orientado a las pruebas

CE23 - Conocimiento de los principios de la inteligencia artificial y uso de algoritmos de búsqueda deterministas y máquinas de estado

Resultados de aprendizaje

Al acabar la titulación, el graduado o graduada será capaz de:

- Entender y manejar el concepto de memoria dinámica
- Identificar clases de objetos con los datos de un problema.
- Crear clases y objetos y manipularlos.
- Entender y utilizar los mecanismos de herencia, polimorfismo y sobrecarga de operadores.
- Identificar las relaciones entre clases en distintos casos de uso.
- Dominar un lenguaje de programación orientado a objetos.
- Dominar los patrones de programación

- Conocer las distintas formas de resolución de problemas desde el punto de vista de la algoritmia, como, por ejemplo, el esquema divide y vencerás,
- programación dinámica, backtracking o algoritmos genéticos.
- Estudiar la complejidad de un determinado algoritmo, interpretar dicha complejidad y analizar posibles optimizaciones.
- Codificar un programa que sea capaz de encontrar el camino óptimo que une dos nodos de un grafo siguiendo los distintos algoritmos de pathfinding.
- Crear y entrenar redes neuronales que solucionen problemas concretos.

CONTENIDO

Computación amorfa, autoestabilización, grid computing

Representación del conocimiento

Sistemas de producción

Razonamiento con incertidumbre

Algoritmos de Pathfinding

Behaviour Trees

Task planners

TEMARIO

1. ¿Qué es exactamente la IA?

Aprendizaje supervisado, no supervisado y por refuerzo

IA, IGA, y e inteligencia artificial superinteligente (ASI)

2. Introducción a Python, Jupyter Notebooks, Google Colab y Kaggle

3. datos tabulares: ANN

regresión lineal

regresión logística y clasificación

redes neuronales con Keras

- Neurona única: regresión lineal

- Neurona única: regresión logística

4. datos de imagen: CNN

historia: el concurso ImageNet y AlexNet

prácticas: MNIST / fashion MNIST / CIFAR-10

modelos de Keras preentrenados: VGG-16, ResNet y EfficientNet

5.datos de texto: RNN

limpieza con REGEX, scraping con BeautifulSoup

tokenization y vectorization (BoW, TF-IDF,...)

redes neuronales recurrentes: LSTM con Keras

6.aprendizaje profundo no supervisado

autoencoders

GAN

7.aprendizaje por refuerzo

bandido de múltiples brazos

8.Ética, IA y sociedad, RGPD y la Ley de IA de la UE

ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DOCENTES

Actividades formativas

Actividad Formativa	Horas totales	Horas presenciales
<i>Clases teóricas / Expositivas</i>	36	36
<i>Clases Prácticas</i>	19	19
<i>Tutorías</i>	4	2
<i>Estudio independiente y trabajo autónomo del alumno</i>	52	0
<i>Elaboración de trabajos (en grupo o individuales)</i>	34	0
<i>Actividades de Evaluación</i>	6	6

Metodologías docentes

Método expositivo o lección magistral

Aprendizaje de casos

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología Flipped classroom o aula invertida

Gamificación

Just in time Teaching (JITT) o aula a tiempo

Método expositivo o lección magistral

Método del caso

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología flipped classroom o aula invertida

Gamificación

DESARROLLO TEMPORAL

UNIDADES DIDÁCTICAS / TEMAS	PERÍODO TEMPORAL
-----------------------------	------------------

Tema 1: ¿Qué es exactamente la IA? - Semana 1	
---	--

Tema 2: Introducción a Python, Jupyter Notebooks, Google Colab y Kaggle - Semana 2	
--	--

Tema 3: Datos tabulares - Semanas 3, 4 y 5	
--	--

Tema 4: Datos de imagen - Semanas 6, 7 y 8	
--	--

Tema 5: Datos de texto - Semanas 9, 10 y 11	
---	--

Tema 6: Aprendizaje profundo no supervisado - Semanas 12 y 13	
---	--

Tema 7: Aprendizaje por refuerzo - Semana 14	
--	--

Tema 8: Ética - Semana 15	
---------------------------	--

SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	0	30
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	30	80
<i>Prueba Objetiva</i>	10	60

CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	10
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	60	60
<i>Prueba Objetiva</i>	30	30

Consideraciones generales acerca de la evaluación

Durante el semestre habrá tres evaluaciones, cada una con un peso del 20 % en la calificación final del curso:

trabajo parcial 1 (CNN): cuaderno de competición sobre gatos y perros

trabajo parcial 2 (NLP): cuaderno de competición sobre clasificación binaria de textos

proyecto individual final (más o menos) a elección del estudiante

Para aprobar, la puntuación mínima requerida en cada evaluación es de 10 sobre 20.

Todos los trabajos son estrictamente individuales.

La puntuación en la clasificación de las competiciones no es el factor principal; el trabajo se evaluará en función de la calidad de las presentaciones, la funcionalidad y la documentación detallada del código.

Además, habrá una exposición oral de 5 minutos sobre cada uno de los proyectos.

El examen final se realizará sin conexión a Internet (es decir, utilizando SMOWL) y sin apuntes del curso.

La nota mínima para aprobar es del 50 % en el examen final.

Las notas extraordinarias tienen un límite máximo del 80 %.

BIBLIOGRAFÍA / WEBGRAFÍA

Hastie, Tibshirani, and Friedman “The Elements of Statistical Learning” 2nd Edition, Springer (2017)

François Chollet “Deep Learning with Python” 3rd Edition, Manning (2025)

Aurélien Géron “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, 3rd Edition O’Reilly Media (2022) (especially Part II)

Ian Goodfellow, Yoshua Bengio and Aaron Courville “Deep Learning”, The MIT Press (2016)

Christopher M. Bishop, Hugh Bishop “Deep Learning: Foundations and Concepts”, Springer (2024)

Simon J. D. Prince “Understanding Deep Learning”, (2025)

Daniel Jurafsky, James H. Martin “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models” 3rd Edition (2025)

Jay Alammar, Maarten Grootendorst “Hands-On Large Language Models”, O’Reilly Media (2024)

Sutton and Barto “Reinforcement Learning: An Introduction” (2020)

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Tipología del aula

Aula teórica

Equipo de proyección y pizarra

Materiales:

Ordenador personal

Software:

Python:

Última versión de Python: <https://www.python.org/downloads/>

Como mínimo serán necesarias las siguientes librerías:

- virtual environments (virtualenv): <https://virtualenv.pypa.io/en/latest/>

- jupyter notebooks (<https://jupyter.org/>) - numpy (<https://numpy.org/>)
- keras (<https://keras.io/>)

y probablemente algunas de las librerías de Image Processing y NLP que aparecen referenciadas en la primera página enlazada en esta tabla