

Carrera INGENIERIA EN INFORMATICA	
Asignatura (03668) INTELIGENCIA ARTIFICIAL APLICADA	
Trayecto: Desarrollo de Software	
Plan 2023	
Jefe de cátedra Damián Montefiori	
Carga horaria semanal 4	Carga horaria total 64
Modalidad: E	
Correlativas anteriores	Correlativas posteriores
<ul style="list-style-type: none"> • Inteligencia Artificial • Estadística Aplicada 	

EQUIPO DOCENTE	
Nombre	Correo electrónico institucional
Damián Montefiori	dmontefiori@unlam.edu.ar

Descripción de la asignatura

La asignatura aborda técnicas contemporáneas de IA con foco en NLP y procesamiento lingüístico, computación afectiva, procesamiento e inteligencia basada en imágenes, Sistemas Autónomos de Robots y modelos generativos. Integra fundamentos de lingüística, aprendizaje automático y profundo con práctica en Python, DNN, transfer learning, y entrenamiento/validación de modelos para texto e imagen, conexión con APIs de modelos generativos en la nube (GPT, Gemini, Sonnet, Azure OpenAI) y despliegue on-premise (Ollama). Se explica la arquitectura Transformer y el funcionamiento de los modelos autoregresivos, articulando teoría y práctica en clase.

Se enfatiza el ciclo de vida de desarrollo de software asistido por IA: captura de requerimientos y user stories en Azure DevOps/GitHub con apoyo de LLMs y MCP para conectar tecnologías y automatizar tareas; generación, corrección y revisión de código, sugerencias de patrones de diseño y mejoras de calidad, eficiencia y eficacia; testeo automatizado con TypeScript y Playwright orquestado vía MCP; y despliegue/CI-CD automatizado asistido por IA, considerando seguridad, costo, latencia y privacidad.

Se incluye línea de tiempo de la evolución de la IA, de enfoques simbólicos y estadísticos a deep learning, Transformers y modelos multimodales.

Al finalizar, el estudiante conocerá los fundamentos de la IA y estará capacitado para diseñar y construir aplicaciones inteligentes basadas en modelos generativos de última generación, con asistencia de IA a lo largo del ciclo de desarrollo. Además, podrá desplegar agentes de IA para la productividad personal y la optimización de procesos de negocio.

Metodología de enseñanza (Modalidad E – Sincrónica por Microsoft Teams)

La cursada es **100% en vivo por Microsoft Teams**, utilizando **equipos con canales** para trabajo por grupos, **diapositivas con screen sharing** y **ejemplos en vivo**. El enfoque es “aprender haciendo” con microclases teóricas, demostraciones y práctica guiada.

Actividades:

- **Teoría + discusión guiada:** conceptos clave (Transformer, modelos autoregresivos, NLP/visión, afectiva, SAR), decisiones de diseño, ética y seguridad.

- **Laboratorio en vivo:** ejercicios en Python/TypeScript; entrenamiento/validación de modelos; uso de APIs (GPT, Gemini, Sonnet, Azure OpenAI) y despliegue on-premise (Ollama); automatización con MCP.
- **Módulo focal de SDLC asistido por IA y aplicaciones inteligentes:** en este tramo se demuestra la construcción de una aplicación *end-to-end* como caso ilustrativo para requisitos, desarrollo, pruebas y despliegue asistidos por IA.
- **Proyecto por equipos:** cada grupo gestiona un **backlog** en **Azure DevOps o GitHub (Projects/Copilot)**, define **roles**, planifica **sprints** y asume compromisos. El trabajo se apoya en **agentes generativos** que cumplen roles de **PO, Tester y Developers** para asistir en *user stories*, generación/revisión de código, pruebas y *release*. **Articulación:** la estructura regular de clase se basa en **teoría + ejemplos en vivo + práctica**. En el **módulo de SDLC asistido por IA**, el docente realiza una **demonstración end-to-end** de una aplicación funcional inteligente **construida desde cero**, evidenciando cómo la asistencia de **modelos generativos** (y su orquestación con **MCP**) aporta en cada etapa: **levantamiento de requerimientos y user stories, diseño arquitectónico, generación/revisión/corrección de código, pruebas automatizadas** (TypeScript/Playwright), y **despliegue/CI-CD**. Cada unidad produce **artefactos verificables** que documentan y demuestran el logro de los objetivos de aprendizaje.

Mecanismos de seguimiento y supervisión de actividades

Seguimiento del estudiante (sincrónico remoto)

- **Participación en clase (Teams):** registro de asistencia y colaboraciones orales/escritas en reuniones y **canales de equipo**; el docente recorre los canales y *breakout rooms* durante el trabajo grupal para observar y retroalimentar.
- **Proyecto integrador en dos instancias:** seguimiento del avance hacia **Entrega Parcial – MVP** y **Entrega Final** (alcance funcional, funcionalidad “inteligente” y uso de IA en SDLC), con **presentación y defensa técnica** en cada hito.
- **Evaluación individual (coloquio):** entrevista técnica por instancia para validar dominio personal (problema, contribuciones, decisiones de diseño, rol de la IA por etapa del SDLC y **comprensión de los contenidos troncales: procesamiento de lenguaje natural, visión por computadora, computación afectiva y procesamiento de imágenes**).

Supervisión de actividades

- **Coordinación del TP y calidad del producto:** verificación de que el TP cumpla el objetivo (aplicación inteligente + uso de IA en SDLC) y de la contribución efectiva de la IA en diseño, código, pruebas, documentación y despliegue.
- **Integración con APIs de modelos generativos:** control de **instrumentación correcta de APIs/SDK** (endpoints, versiones, errores, *rate limiting*, *retries/backoff*), **gestión segura de credenciales** (variables de entorno, *secret managers*, rotación), **parámetros** (uso eficiente de **tokens, temperatura, top-p**, penalizaciones, *seeds*),

construcción de contexto (prompts, plantillas, versionado, memoria externa),
orquestración con MCP (herramientas, *schemas*, *timeouts*).

- **Presentaciones con defensa y Q&A:** cada grupo expone problema, parte “inteligente”, modelo/API y rol de la IA por etapa; el docente dirige preguntas a cualquier integrante.
- **Verificación de criterios de evaluación:** alineación con los criterios del TP (calidad del producto, nivel de integración de IA, claridad de la exposición, participación y dominio individual).

Objetivos de aprendizaje

- Explicar los fundamentos de NLP y procesamiento lingüístico.
- Describir la arquitectura Transformer y el funcionamiento de modelos autoregresivos y multimodales.
- Entrenar, ajustar y evaluar modelos para texto e imagen (DNN, transfer learning, métricas y validación).
- Construir aplicaciones de IA con herramientas y orquestración mediante MCP, integrando LLMs y APIs (GPT, Gemini, Sonnet, Azure OpenAI).
- Implementar soluciones on-premise con Ollama y comparar con despliegues en la nube (costo, latencia, privacidad).
- Aplicar computación afectiva para análisis de sentimiento/emoción en texto/voz/imagen.
- Diseñar y validar sistemas basados en imágenes (clasificación/detección/segmentación) con buenas prácticas de dataset.
- Elaborar requerimientos y *user stories* asistidos por IA (Azure DevOps/GitHub), y generar/corregir/revisar código con soporte de LLMs.
- Automatizar pruebas end-to-end con TypeScript y Playwright orquestrado vía MCP; integrar CI/CD asistido por IA.
- Justificar decisiones técnicas considerando ética, seguridad, sesgos, robustez y normativas aplicables.

Contenidos mínimos

Procesamiento de Lenguaje Natural (NLP) y procesamiento lingüístico

- Definición y alcance del NLP; objetivos de comprender, interpretar y generar lenguaje humano desde una perspectiva interdisciplinaria (lingüística computacional, *machine learning* y *deep learning*).
- Evolución histórica: de enfoques basados en reglas y estadísticos a arquitecturas neuronales modernas y LLMs.
- Generación autoregresiva: noción práctica de generación palabra a palabra y parámetros de muestreo (temperatura, *top-p*).
-

Sistemas basados en textos

- Construcción de componentes que emplean LLMs para generación/continuación de texto y análisis; configuración y control de la generación (temperatura, *top-p*) como parte del diseño del sistema.

- Integración de modelos mediante APIs para funcionalidades textuales (p. ej., modelos de lenguaje, análisis de sentimientos).

Computación afectiva

- Definición y propósito: sistemas que reconocen, interpretan, procesan y simulan emociones humanas.
- Procesos clave: preprocesamiento, extracción de características y clasificación con SVM, redes neuronales y *transformers* (texto) o CNNs (imagen).
- Importancia del contexto e integración multimodal (texto, voz, imagen); consideraciones éticas y de privacidad.
- Canales de detección y respuesta emocional en HCI (texto, voz, imagen/video; adaptación de la interacción).

Procesamiento de imágenes y visión por computadora

- Uso de *deep learning* (CNN) para tareas de clasificación/detección/segmentación y validación del desempeño.
- Aplicaciones en productos inteligentes que consumen APIs de reconocimiento de imágenes.

Sistemas inteligentes basados en imágenes

- Integración de modelos de visión en aplicaciones inteligentes.

Sistemas Autónomos de Robots (SAR)

- Definición y características: autonomía, adaptabilidad, interacción física y percepción multimodal (cámaras, LiDAR, ultrasonido, IMU).
- Paradigma **Sense–Think–Act** y ciclo continuo de realimentación.
- Componentes: percepción (RGB-D, LiDAR, radar, GPS/IMU), control/decisión y actuación (motores/servos; ruedas diferenciales/omnidireccionales; manipuladores).
- Arquitecturas de control: reactivas, deliberativas e híbridas en capas.
- **SLAM** (localización y mapeo simultáneos): noción, enfoque probabilístico/optimización.
- Planificación y navegación: A*, Dijkstra, RRT (planificación global) y DWA (control local en entornos dinámicos).
- Desafíos y consideraciones: robustez en condiciones adversas, seguridad humana, tiempo real, energía; aspectos éticos/legales.

Competencias a desarrollar

Genéricas (*alineadas al cuadernillo de CONFEDI*)

- Comunicación oral y escrita efectiva en ámbitos técnicos (presentación y defensa de soluciones).
- Trabajo en equipo y colaboración interdisciplinaria en entornos virtuales (Teams, repositorios compartidos).
- Pensamiento crítico y resolución de problemas complejos con base en evidencia.
- Aprendizaje autónomo y actualización permanente ante nuevas tecnologías de IA.

- Gestión del tiempo, planificación y organización del trabajo por sprints y *backlogs*.
- Compromiso ético, responsabilidad social y respeto por la privacidad y la seguridad de la información.

Específicas (alineadas al cuadernillo de CONFEDI)

- Modelado, diseño e implementación de sistemas inteligentes para texto e imagen (NLP, visión, afectiva).
- Integración de modelos generativos mediante APIs/SDK y orquestación (MCP), incluyendo *prompting* y manejo de contexto.
- Programación y validación de soluciones con *deep learning* (DNN/CNN/Transformers), *transfer learning* y evaluación con métricas adecuadas.
- Construcción de agentes y sistemas basados en textos (incl. RAG), y su despliegue *cloud* u *on-premise* (Ollama).
- Aplicación del ciclo de vida del desarrollo de producto asistido por IA: requisitos, diseño, generación/revisión de código, pruebas automatizadas (TypeScript/Playwright) y CI/CD.
- Gestión segura de credenciales y configuración de parámetros de modelos (tokens, temperatura, *top-p*), con análisis de costo, latencia, robustez y sesgos.
- Experimentación reproducible, trazabilidad requisito→modelo→prueba, documentación técnica y defensa de decisiones de diseño.

Programa analítico [Agregar una fila por cada unidad temática]	
Unidad 1	Introducción a la IA Aplicada y encuadre de la cursada Panorama y evolución (simbólico→estadístico→DL→Transformers→multimodal), objetivos y modalidad E (Teams). Ética, seguridad y privacidad.
Unidad 2	Procesamiento lingüístico & Sistemas basados en texto Pipeline de NLP (tokenización, POS, NER, sintaxis, semántica), <i>embeddings</i> y modelos neuronales para texto. Arquitectura Transformer y modelos autoregresivos ; <i>prompt engineering</i> y control de generación (temperatura, <i>top-p</i>). Sistemas basados en textos: búsqueda semántica y RAG ; diseño de <i>pipelines</i> y evaluación.
Unidad 3	Agentes y orquestación con MCP Diseño de agentes con herramientas, <i>tool schemas</i> , memoria y contexto; encadenamiento de tareas y buenas prácticas de seguridad. Integración de agentes en flujos de trabajo y aplicaciones.
Unidad 4	Computación afectiva Modelos de emoción (discretos/dimensionales), análisis de sentimiento y emoción en texto/voz/imagen; integración multimodal y consideraciones éticas.
Unidad 5	Procesamiento de imágenes & Sistemas basados en imágenes Preprocesamiento y CNN . Tareas de clasificación, detección y segmentación; métricas y validación. Diseño de <i>pipelines</i> de visión e integración en aplicaciones inteligentes.

Unidad 6	Integración de modelos generativos vía APIs Servicios en la nube (GPT, Gemini, Sonnet, Azure OpenAI) y despliegue on-premise (Ollama). Gestión de <i>API keys</i> , parámetros (tokens, temperatura, <i>top-p</i>), costos, latencia y privacidad.
Unidad 7	SDLC asistido por IA (unidad integrada) Asistencia con LLMs a lo largo del ciclo de vida: captura de requerimientos y user stories (Azure DevOps/GitHub), diseño arquitectónico, generación/corrección/visión de código y sugerencia de patrones, testing automatizado (TypeScript + Playwright orquestado con MCP), documentación, y CI/CD asistido por IA con telemetría y trazabilidad.
Unidad 8	Sistemas Autónomos de Robots (SAR) – Introducción Sense–Think–Act; percepción (RGB-D/LiDAR/IMU), decisión/control y actuación. Nociones de SLAM, planificación y navegación; desafíos de tiempo real, seguridad y robustez.

Planificación de actividades [15/16 semanas dependiendo del calendario académico universitario – UNLaM]							
Semana	Clase	Actividad	Tipo	Duración estimada	Unidad	Modalidad	Plataforma
Semana 1	1	U1 – Introducción, encuadre y línea de tiempo de la IA; ética/seguridad; setup de repos/repositorios de curso	teoría	4 h	U1	E – Sincrónica	Microsoft Teams, GitHub
Semana 2	2	U2 – NLP & Proc. lingüístico (I): tokenización, POS, NER, sintaxis/semántica; ejercicios guiados	práctica de laboratorio	4 h	U2	E – Sincrónica	Teams, Google Colab / VS Code
Semana 3	3	U2 – NLP & Sist. basados en texto (II): embeddings, Transformer/AR, prompting, RAG/búsqueda semántica	práctica de laboratorio	4 h	U2	E – Sincrónica	Teams, Colab / VS Code
Semana 4	4	U7 – SDLC asistido por IA: user stories (Azure DevOps/GitHub), diseño, demo E2E	práctica	4 h	U7	E – Sincrónica	Teams, Azure DevOps/GitHub, GitHub Copilot

		docente, testing y CI/CD asistido					
Semana 5	5	U6 – Integración de modelos generativos vía APIs: llamadas, manejo de API keys, parámetros (tokens, temperatura, top-p)	práctica de laboratorio	4 h	U6	E – Sincrónica	Teams, Azure/Open AI/Google/Anthropic, Postman/Colab
Semana 6	6	U5 – Imagen & sist. basados en imagen (I): preprocesamiento, CNN, transfer learning, clasificación	práctica de laboratorio	4 h	U5	E – Sincrónica	Teams, Colab / VS Code
Semana 7	7	U5 – Imagen & sist. basados en imagen (II): detección/segmentación; métricas y validación	práctica de laboratorio	4 h	U5	E – Sincrónica	Teams, Colab / VS Code
Semana 8	8	Presentación TP Parcial – Grupos A (MVP + Q&A individual)	otra	4 h	TP Parcial	E – Sincrónica	Microsoft Teams
Semana 9	9	Presentación TP Parcial – Grupos B (MVP + Q&A individual)	otra	4 h	TP Parcial	E – Sincrónica	Microsoft Teams
Semana 10	10	U3 – Agentes inteligentes y orquestación con MCP: herramientas, tool schemas, memoria/contexto, flujos multi-paso	práctica de laboratorio	4 h	U3	E – Sincrónica	Teams, VS Code, MCP
Semana 11	11	U8 – SAR (Intro): Sense–Think–Act; percepción (RGB-D/LiDAR/IMU), control/actuación; SLAM/navegación (nociones)	teoría	4 h	U8	E – Sincrónica	Microsoft Teams

Semana 12	12	Presentación TP Integral – Grupos A (defensa técnica + uso de IA en SDLC)	Evaluación	4 h	TP Integral	E – Sincrónica	Microsoft Teams
Semana 13	13	Presentación TP Integral – Grupos B (defensa técnica + uso de IA en SDLC)	Evaluación	4 h	TP Integral	E – Sincrónica	Microsoft Teams
Semana 14	14	U4 – Computación afectiva: modelos discretos/dimensionales; sentimiento/emoción en texto/voz/imagen; práctica guiada	práctica de laboratorio	4 h	U4	E – Sincrónica	Teams, Colab / VS Code
Semana 15	15	Recuperatorios (según normativa)	Evaluación	4 h	—	E – Sincrónica	Microsoft Teams
Semana 16	16	Cierre de notas/actas y devolución final	Evaluación	2–4 h	—	E – Sincrónica	Microsoft Teams

Evaluación

Evaluación

El proceso evaluativo combina **evidencias individuales y de equipo** y contempla tanto la **presentación con demo** como **entregables documentales y verificación técnica**.

Criterios específicos

1. **Dominio conceptual:** NLP/procesamiento lingüístico, visión/procesamiento de imágenes, computación afectiva, fundamentos de Transformers/modelos autoregresivos y nociones de SAR.
2. **Integración técnico-práctica:** uso de IA en el **SDLC** (análisis y construcción de requerimientos, diseño, generación/corrección/revisión de código, pruebas automatizadas y CI/CD), **integración con APIs** de modelos generativos (manejo seguro de *API keys*, parámetros: tokens, temperatura, *top-p*, penalizaciones; construcción de contexto/prompting), y **orquestración con MCP**.
3. **Calidad del producto/software:** claridad de la solución y valor agregado por la IA para construir aplicaciones inteligentes, reproducibilidad de experimentos, métricas, seguridad/privacidad, trazabilidad **requisito→código→pruebas→demo**, y comunicación técnica del equipo e individual.

Instrumentos e evidencias

- **Presentación con demo** (equipo) + **coloquio individual** (Q&A técnico dirigido a cualquier integrante).
- **Repositorio del proyecto:** work items, issues, resultados de pruebas automatizadas, *readme*.
- **Informe del equipo (entrega obligatoria)** que incluye:
 - **Diapositivas** de la presentación compartida.

<ul style="list-style-type: none"> ○ Enlace al repositorio de la aplicación inteligente construida con asistencia de IA. ○ Documento técnico con: <ul style="list-style-type: none"> ▪ Descripción del producto software y problema que resuelve; ▪ Identificación de la parte “inteligente” del sistema; ▪ Modelos Generativos/APIs utilizados y justificación; ▪ Aporte de la IA en cada etapa del SDLC (requisitos, diseño, desarrollo, pruebas, despliegue/operación); ▪ Dificultades/problemas encontrados y cómo se abordaron; ▪ Métricas de calidad (modelo, software y proceso) y consideraciones de seguridad/privacidad. • Observación en vivo de participación y colaboración en canales/salas de Teams durante el trabajo grupal. 			
Primera evaluación	Semanas 8–9	Presentación TP Parcial (MVP) + coloquio individual + informe parcial (diapositivas, repo y doc. técnico acotado)	3 hrs, 19 a 22hrs.
Segunda evaluación	Semanas 12–13	Presentación TP Integral + coloquio individual + informe final (diapositivas, repo y doc. técnico completo)	3 hrs, 19 a 22hrs.
Recuperatorio	Semana 15	Instancias de recuperación según normativa vigente	3 hrs, 19 a 22hrs.

Bibliografía obligatoria [Disponibles en la Biblioteca Leopoldo Marechal, o con acceso digital]				
Título	Autor	Editorial	Edición	Año
Speech and Language Processing (3rd ed. draft)	Jurafsky, D.; Martin, J.H.	Prentice Hall (manuscrito en línea)	3ª (borrador)	2024– 2025
Computer Vision: Algorithms and Applications (2nd ed.)	Szeliski, R.	Springer (acceso abierto)	2ª	2022
Deep Learning (online)	Goodfellow, I.; Bengio, Y.; Courville, A.	MIT Press (edición web)	1ª	2016
A Comprehensive Survey on Affective Computing arXiv	Afzal, S. et al.	arXiv		2023

Bibliografía complementaria recomendada

Titulo	Autor	Editorial	Edición	Año
Natural Language Processing with Transformers (Revised Edition)	Tunstall, L.; von Werra, L.; Wolf, T.	O'Reilly	1ª (rev.)	2022
Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (2nd ed.)	Géron, A.	O'Reilly	2ª	2019
Machine Learning with PyTorch and Scikit-Learn	Raschka, S.; Mirjalili, V.; Liu, Y.H.	Packt	1ª	2022
Probabilistic Robotics	Thrun, S.; Burgard, W.; Fox, D.	MIT Press	1ª	2005

Otros recursos obligatorios [Videos, enlaces, otros. Incluir una fila por cada recurso]

- APIs de modelos generativos (documentación oficial):
OpenAI API [OpenAI Platform](#) · Azure OpenAI (Azure AI Foundry) [Microsoft Learn](#) ·
Gemini API (Google) [Google AI for Developers](#) · Anthropic Claude API
docs.anthropic.com
- Model Context Protocol (MCP): especificación y repositorio oficial.
modelcontextprotocol.io
- Ollama: documentación y referencia de API (despliegue *on-premise*).
docs.ollama.com
- Playwright (Testing E2E con TypeScript): introducción e instalación. [Playwright](#)
- OpenCV (visión por computadora): documentación oficial. [OpenCV Docs](#)
- ROS 2 (SAR): tutoriales oficiales. docs.ros.org

Otros recursos complementarios [Videos, enlaces, otros. Incluir una fila por cada recurso]

- Transformers: "Attention Is All You Need" (arXiv/NIPS 2017). [arXiv](#)
- Visión profunda: "Deep Residual Learning for Image Recognition" (ResNet). [arXiv](#)
- Cursos abiertos: Stanford CS224n (NLP) [Stanford University](#) · Stanford CS231n
(Visión) cs231n.github.io · fast.ai – Practical Deep Learning for Coders [Practical Deep Learning for Coders](#) · Hugging Face – LLM/Agents/MCP courses [Hugging Face](#)
- Datasets de referencia (visión): COCO (oficial) cocodataset.org · ImageNet (oficial)
image-net.org

"Declaro que el presente programa de estudios de la asignatura, es el vigente para el ciclo lectivo, guarda consistencia con los contenidos mínimos del Plan de Estudios"



Firma

DAMIAN MONTEFIORI

Aclaración

11/11/2025

Fecha