

Syllabus del Curso de Introducción a la Inteligencia Artificial Generativa y Agéntica (Videoconferencia / 40 horas).

Unidad 1: La Inteligencia Artificial como sistema empresarial (4 horas)

- Definición de Inteligencia Artificial en contexto empresarial (qué es y qué no es)
- Definición de sistema de IA: datos, modelo, inferencia, orquestación, control y supervisión
- Definición y diferencias: automatización, sistemas inteligentes, sistemas clasificadores, sistemas generativos y sistemas agénticos
- Diferencias clave entre IA moderna y software tradicional (probabilística vs determinística)
- Modelos de lenguaje de gran tamaño (LLM): concepto, alcance y límites prácticos
- Tokens, embeddings y espacio latente como base de representación para lenguaje y señales
- Entrenamiento, ajuste (fine-tuning) e inferencia: qué cambia y qué cuesta en cada fase
- Calidad y gobernanza de datos como factor dominante del desempeño y del riesgo
- Errores conceptuales frecuentes en adopción de IA: demos, pilotos eternos, prompts sin sistema
- Criterios para decidir cuándo no usar IA y cómo diseñar alternativas seguras

Caso de estudio: evolución de los sistemas de IA (de sistemas expertos a LLMs) y cambio en el modelo operativo empresarial

Actividad práctica: clasificación de iniciativas por nivel: automatización, soporte a decisión, decisión automatizada, agencia

Unidad 2: Ciclos de vida de los sistemas de inteligencia artificial (4 horas)

- Fuentes, captura, almacenamiento, transformación y consumo
- Etapas del ciclo de vida: diseño, entrenamiento, inferencia, despliegue y monitoreo
- Entendiendo más a fondo el aprendizaje automático
- Modelos de lenguaje y modelos de embeddings como componentes funcionales distintos
- Arquitecturas empresariales de IA: pipelines, servicios, APIs y orquestación

- Métricas operativas clave en sistemas de IA: desempeño, confiabilidad, latencia y costo
- Deriva de datos y de modelos como causa principal de degradación en producción

Actividad práctica: análisis de un diagrama de arquitectura de un sistema de IA para identificar riesgos en cada etapa del ciclo de vida

Unidad 03: Ingeniería de prompts con chatbots de propósito general (04 horas)

- Estructura interna de una conversación: prompts, respuestas y *threads*
- Prompt como capa de control cognitivo y contractual en sistemas conversacionales
- Prompts de sistema, prompts de desarrollador y prompts de usuario como capas diferenciadas
- Entendiendo la personalidad del chatbot como el producto de instrucciones internas, políticas de seguridad y mecanismos de alineamiento del sistema
- Componentes de un prompt efectivo: contexto, rol, tarea, restricciones y formato de salida
- Ventanas de contexto: límites, truncamiento y efectos en la calidad de respuesta
- Gestión explícita del contexto dentro de una conversación larga
- Diferencia entre memoria implícita, memoria conversacional y ausencia de memoria persistente
- Configuración de rol y personalidad del chatbot como mecanismo de control
- Diferencias operativas entre ChatGPT, Claude y Gemini en estilo, razonamiento y límites

Actividad práctica: creación de prompts de respuestas personalizadas según contexto y caso de uso en ChatGPT, Claude y Gemini

Unidad 04: Programación con Python asistida por un modelos de lenguaje (04 horas)

- Instalación de paquetes para desarrollo y dependencias
- Instalación de versiones de Python con la herramienta asdf
- Uso de modelos de lenguaje para generar y explicar código Python
- Sintaxis y tipos de datos
- Estructuras de control
- Funciones y módulos
- Programación orientada a objetos
- Uso del intérprete de Python y ejecución de scripts
- Exploración de la biblioteca estándar
- Depuración de código con asistencia de IA

Actividad práctica: construcción de un microservicio para la generación en lotes de contraseñas, identificadores únicos y secretos criptográficos utilizando VSCode.

Unidad 05: Ingeniería de software apoyada por agentes (04 horas)

- Diferencias entre autocompletado, copilotos y agentes de ingeniería de software
- Características principales de un entorno de desarrollo de software con capacidades agénticas
- Integración de agentes de código en entornos IDE con Visual Studio Code
- Desarrollo basado en bucles de razonamiento, acción y verificación
- Introducción a OpenRouter como proveedor de modelos para inferencia
- Configuración y uso de VS Code para desarrollo agéntico con Kilo Code y OpenRouter
- Planificación explícita, descomposición de tareas y validación iterativa
- Uso de agentes para lectura, explicación y refactorización de bases de código existentes
- Configuración y uso de OpenCode con modelos de OpenRouter

Actividad práctica: Corrección de errores, refactorización de código e implementación de nuevas características en una aplicación web ya existente.

Unidad 06: Uso y desarrollo de servidores MCP (04 horas)

- Definición del **Model Context Protocol (MCP)** como estándar de integración entre modelos de lenguaje y sistemas externos
- Diferencias entre integraciones directas con APIs de modelos y arquitecturas basadas en MCP
- Componentes fundamentales de una arquitectura MCP: servidor, herramientas, recursos y contexto
- Estructura interna de un servidor MCP y su ciclo de vida
- Definición y exposición de herramientas (*tools*) consumibles por modelos de lenguaje
- Gestión de contexto estructurado y no estructurado mediante MCP
- Control de permisos, alcance y superficie de acción a través del diseño del servidor MCP
- Integración de servidores MCP en chatbots y entornos de codificación agéntica

Actividad práctica: Construcción de un servidor MCP con Python para la gestión de inventarios incluyendo registros de auditoría y permisos por rol.

Unidad 07: Gestión VPS y despliegue de modelos de pesos abiertos (04 horas)

- Conceptos fundamentales de un servidor Linux VPS para cargas de trabajo de IA
- Preparación del entorno: sistema operativo, usuarios, permisos y seguridad básica
- Gestión de recursos en VPS: CPU, memoria, almacenamiento y red
- Introducción a modelos de pesos abiertos y diferencias frente a modelos cerrados
- Impacto de la cuantización en consumo de memoria, rendimiento y calidad
- Criterios para descubrir, evaluar y seleccionar modelos de pesos abiertos disponibles públicamente a través de Internet
- Uso de herramientas para descarga y ejecución local de modelos, incluyendo Ollama

Actividad práctica: configuración de un servidor Linux VPS y despliegue de un modelo open-weights usando Ollama, aplicando criterios de selección y cuantización según los recursos disponibles.

Unidad 08: Automatización y orquestación moderna de flujos con n8n (04 horas)

- Introducción a n8n como plataforma de automatización flexible y auto-hospedable
- Posicionamiento de n8n frente a Zapier y Make en arquitecturas modernas de IA
- Conceptos fundamentales de n8n: workflows, nodos, triggers, acciones e items
- Diseño de flujos reproducibles, versionables y mantenibles
- Gestión segura de credenciales e integración con APIs y servicios externos
- Uso de expresiones, variables y nodos de código para transformación avanzada de datos
- Orquestación de modelos de lenguaje dentro de workflows con OpenRouter y Ollama
- Integración de n8n con servicios auto-hospedados y modelos open-weights
- Manejo de errores, bifurcaciones, reintentos y confiabilidad operacional
- Exposición de workflows como APIs mediante webhooks para consumo externo

Actividad práctica: Construcción de un workflow en n8n para clasificación de severidad de tickets y asignación inteligente de personal según urgencia y experiencia, integrando modelos de lenguaje y reglas de negocio, y exponiendo el flujo como API interna.

Unidad 09: Integración y creación de habilidades complejas (04 horas)

- Definición de *skill* como unidad funcional reusable para sistemas agénticos
- Diferencias entre prompts, herramientas (*tools*), workflows y *skills*
- La importancia del descubrimiento progresivo y su relación con las ventanas de contexto de modelos de lenguaje
- Estructura de una skill al estilo Anthropic: intención, entradas, salidas y restricciones
- Separación entre razonamiento del agente y ejecución encapsulada en skills
- Diseño de contratos claros para skills: esquemas de entrada y salida
- Versionado, documentación y descubrimiento de skills
- Consumo de skills en chatbots y entornos de codificación agéntica

Actividad práctica: diseño e implementación de un conjunto de skills para copywriting y creación de blog posts.

Unidad 10: Agentes conversacionales con OpenClaw y Telegram (04 horas)

- Introducción a OpenClaw como framework para agentes conversacionales auto-hospedados
- Arquitectura general de OpenClaw: núcleo agéntico, herramientas y canales de interacción
- Configuración de un entorno local o VPS para ejecutar OpenClaw
- Integración de OpenClaw con modelos de lenguaje mediante proveedores configurables
- Conceptos básicos de bots en Telegram y su modelo de interacción
- Creación y configuración de un bot de Telegram para consumo de agentes
- Conexión de OpenClaw con Telegram como canal conversacional
- Gestión de contexto, estado y memoria básica en agentes conversacionales
- Diseño de flujos conversacionales útiles y controlados
- Pruebas, ajuste y despliegue inicial de un agente funcional accesible desde Telegram

Actividad práctica: instalación y configuración de un agente OpenClaw integrado con Telegram, capaz de responder a consultas reales y ejecutar acciones simples, dejando al participante con un agente funcional operando en su propio entorno.