

Syllabus del Curso de Ingeniería de Software y Programación con Agentes de Inteligencia Artificial (Videoconferencia / 40 horas)

Unidad 1: DE CHATBOTS A CO-INGENIEROS (4 horas)

- GenAI 1.0 vs GenAI 2.0
- Ciclo ORA (Observar, Razonar, Actuar)
- Entornos de escritorio vs CLI: AntiGravity, KiloCode, Claude, Codex, OpenCode
- Minimax como motor de razonamiento
- Archivo AGENTS.md como contexto base
- Rol humano como gestor y no ejecutor

Caso de estudio:

Análisis de la transición de un sistema de atención al cliente basado en reglas a un co-ingeniero de código basado en el ciclo ORA.

Actividad práctica:

Configuración de un entorno AI-first (como Codex) para generar un micro-servicio simple, con el operador humano como gestor.

Unidad 2: DESCUBRIMIENTO Y ALINEACIÓN (4 horas)

- Interrogación inversa (grilling)
- Documentos de requisitos orientados a IA (PRD)
- Definición de *Definition of Done*
- Contratos de prompt (objetivos, restricciones, éxito)
- Revisión de planes antes de ejecución
- Eliminación de ambigüedad en requisitos

Caso de estudio:

Diseño de un PRD orientado a IA para una funcionalidad de resumen automático, incluyendo el *Definition of Done* y métricas de éxito.

Actividad práctica:

Creación de un *contrato de prompt* detallado para un agente, utilizando la técnica de interrogación inversa para refinar los objetivos.

Unidad 3: DISEÑO Y ARQUITECTURA (4 horas)

- Contextos delimitados (bounded contexts)
- Arquitectura hexagonal (puertos y adaptadores)
- Event sourcing como modelo de estado
- Separación entre lógica central y dependencias
- Dirección estratégica humana
- Diseño como fuente de verdad

Caso de estudio:

Evaluación de una aplicación monolítica existente para su refactorización a una arquitectura hexagonal con separación clara de contextos.

Actividad práctica:

Diseño de la arquitectura (bounded contexts, puertos y adaptadores) para un sistema de gestión de pedidos asistido por IA, priorizando la separación de lógica.

Unidad 4: INGENIERÍA DE CONTEXTO (4 horas)

- Presupuesto de atención (smart zone vs dumb zone)
- Divulgación progresiva
- Compactación de contexto
- Prevención de *degradación de contexto*
- Optimización de formatos (Markdown, YAML)
- Control del tamaño del contexto

Caso de estudio:

Optimización del contexto en un agente de soporte técnico para reducir la latencia y evitar la degradación *de contexto* en conversaciones largas.

Actividad práctica:

Experimentación con diferentes formatos de contexto (Markdown vs. YAML) para un conjunto de reglas de negocio, midiendo la eficacia y el control del tamaño.

Unidad 5: BACKEND E INFRAESTRUCTURA (4 horas)

- Servicios modulares
- Ejecución en contenedores (Docker)
- Generación de esquemas de base de datos
- Desarrollo orientado a contratos (API-first)
- Separación definición vs. implementación
- Agente como ejecutor dentro del sistema

Caso de estudio:

Integración de un agente como microservicio ejecutable dentro de un pipeline de CI/CD utilizando contenedores Docker.

Actividad práctica:

Uso de un agente para generar un esquema de base de datos (schema.sql) y la definición de una API (usando OpenAPI) a partir de un PRD.

Unidad 6: FRONTEND Y CAPA VISUAL (4 horas)

- Generación de interfaces desde diseño
- Reglas de estilo como contexto reutilizable
- Desarrollo por *cortes verticales*
- Integración UI + API + Base de datos
- Evaluación de experiencia de usuario
- Iteración rápida de interfaces
- Depuración empleando un servidor MCP para Google Chrome

Caso de estudio:

Implementación de un flujo de *desarrollo por cortes verticales* donde la IA genera la UI y la conecta al backend en una sola iteración.

Actividad práctica:

Uso de herramientas de IA para generar un componente de interfaz de usuario a partir de un diseño conceptual, estabilización y evaluación rápida de su usabilidad.

Unidad 7: ESTANDARIZACIÓN DE FLUJOS (4 horas)

- Habilidades reutilizables (skills)
- Comandos personalizados
- Automatización de tareas repetitivas
- Transformación de SOPs en instrucciones
- Bibliotecas de habilidades
- Reducción de dependencia conversacional

Caso de estudio:

Conversión de un Procedimiento Operativo Estándar (SOP) de recursos humanos en un conjunto de instrucciones y habilidades reutilizables para un agente.

Actividad práctica:

Diseño de una biblioteca de habilidades (skills) que permitan a un agente realizar tres tareas repetitivas diferentes, minimizando la necesidad de conversación paso a paso.

Unidad 8: ORQUESTACIÓN MULTIAGENTE (4 horas)

- Agentes especializados por rol
- Patrón planificador verificador ejecutor
- Ejecución paralela
- Debate entre agentes
- Consenso estadístico
- Separación de responsabilidades

Caso de estudio:

Establecimiento de un flujo multiagente (Requisitos, Arquitecto, Desarrollador) para la creación de una nueva característica de software. Los agentes deben debatir las dependencias y consensuar un plan técnico.

Actividad práctica:

Construcción de un *pipeline* de orquestación donde un agente *Planificador* descompone una *user story* en tareas, un agente *Verificador* valida la factibilidad y un agente *Ejecutor* (Codificador) se encarga de la implementación.

Unidad 9: CALIDAD Y VERIFICACIÓN (4 horas)

- Desarrollo guiado por pruebas (TDD)
- Separación implementador revisor
- Validaciones automáticas
- Ciclos de autocorrección
- Entornos controlados
- Prioridad en confiabilidad

Caso de estudio:

Establecimiento de un ciclo de autocorrección para un agente de codificación, donde un agente implementador genera código y un agente revisor genera pruebas y verifica la ejecución.

Actividad práctica:

Aplicación de la filosofía TDD (Desarrollo guiado por pruebas) utilizando un agente para generar primero las pruebas unitarias antes de escribir la implementación.

Unidad 10: DESPLIEGUE EN PRODUCCIÓN Y AUTOMATIZACIÓN (4 horas)

- Distribución de tareas por complejidad
- Automatización de despliegues
- Mecanismos de seguridad
- Supervisión mínima
- Monitoreo en producción
- Automatización del ciclo completo

Caso de estudio:

Diseño de un sistema de monitoreo en producción para detectar *deriva de modelo* y otros fallos operacionales en un sistema agéntico.

Actividad práctica:

Creación de un script de automatización que distribuya tareas de complejidad variada a diferentes agentes especializados y gestione su despliegue continuo.