

Máster en Ingeniería de Software en la Era de la IA

Arquitectura moderna · Sistemas completos · IA gobernada · Spec-Driven Development

Formación técnica avanzada para programadores e ingenieros de software



Model Context Protocol

Dedicación

10-15h semanales

Titulación

Universitaria Europea

Duración

6 Meses

Equipo Docente

Ingenieros en Activo con Experiencia Real

Claudio Briceño – Director del Máster & CEO de **CodeCrypto**



Ingeniero de Telecomunicación con Máster en Ingeniería de Software y Máster en Blockchain y Criptoconomía. Más de 15 años diseñando, desarrollando e implantando soluciones software siempre muy cerca del negocio: producto digital, arquitecturas distribuidas, APIs y sistemas híbridos Web2 + IA. Es el fundador y CEO de CodeCrypto y el responsable de la visión, el diseño académico y la dirección del máster.

José Viejo – Director Académico & Arquitectura de **Sistemas**



Ingeniero Informático con más de 45 años de experiencia en desarrollo de software y liderazgo técnico. Ha ocupado prácticamente todos los roles posibles en ingeniería: desde desarrollador y arquitecto hasta responsable de grandes sistemas en entornos corporativos. Diseña y lidera los bloques de fundamentos, Spec-Driven Development y los proyectos de dominio del máster.

Andrés León – Lead Técnico & Spec-Driven Development

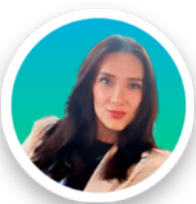


Ingeniero Informático con más de 25 años de experiencia en desarrollo y dirección tecnológica. Especializado en arquitecturas modernas, IA aplicada y Spec-Driven Development. Es la figura clave del Bloque II y de los proyectos de dominio, llevando el método SDD a casos reales de Telco, Banca, Seguros y SaaS.

Equipo Docente

Ingenieros en Activo con Experiencia Real

Lindsey Walker – Plan de Carrera & Posicionamiento Profesional



Psicóloga especializada en desarrollo profesional con más de 15 años de experiencia acompañando a personas en procesos de cambio y crecimiento. Lidera el área de Plan de Carrera, ayudando a los alumnos a traducir su avance técnico en oportunidades reales: LinkedIn, GitHub, entrevistas y roadmap profesional.

María José Rabellino – Soporte Técnico & Acompañamiento del Alumno



Ingeniera Informática con más de 15 años en desarrollo de software y exalumna del máster. Es el punto de soporte más cercano del día a día: resolución de dudas, seguimiento del progreso y dinamización de la comunidad para que la experiencia sea exigente en lo técnico y humana en el acompañamiento.

Conviértete en el ingeniero de software que sabe diseñar, construir y defender sistemas en la era de la IA.

De programador que ejecuta features a ingeniero que diseña sistemas completos.

La IA no va a sustituir a los ingenieros de software. Va a sustituir a los que no actualicen su forma de trabajar.

Este máster te prepara para trabajar con el estándar profesional que exige 2026: no más código por intuición, sino sistemas diseñados, verificables y defendibles.

Competencias que adquirirás:

- Diseñar sistemas desde especificaciones verificables, no desde la intuición
- Derivar arquitectura desde restricciones reales y documentarla con ADRs
- Construir con testing contractual, CI/CD y observabilidad desde el inicio
- Integrar IA como componente técnico bajo control, no como caja negra
- Aplicar el mismo método en distintos dominios: Telco, Banca, Seguros, SaaS
- Defender decisiones técnicas con criterio profesional y datos reales

¿Para quién es este máster?

Este programa está diseñado para perfiles técnicos que ya trabajan en desarrollo de software y quieren adaptarse a la nueva realidad de la IA sin perder criterio.

No es un máster de iniciación. No es para aprender a programar desde cero. Es para programadores e ingenieros que ya tienen base técnica y quieren subir de nivel.

Está pensado para:

- Desarrolladores que ya trabajan con código en su día a día
- Ingenieros de software que quieren actualizar su forma de trabajar
- Perfiles técnicos que quieren diseñar sistemas, no solo implementarlos
- Profesionales que no quieren quedarse en tareas automatizables

No necesitas saber más herramientas. Necesitas trabajar con otro estándar.

Este programa no te enseña a usar IA. Te enseña a trabajar en un entorno donde la IA es parte del sistema.

Estructura del programa · 6 meses · 4 bloques progresivos

El máster está diseñado como una transición estructurada, donde cada bloque sube el nivel de exigencia.

BLOQUE I – BASE

Construir la línea técnica común y trabajar con entorno profesional real.

BLOQUE II – MÉTODO

Aprender a diseñar sistemas con Spec-Driven Development.

BLOQUE III – APLICACIÓN

Llevar el método a 4 dominios reales distintos.

BLOQUE IV – DEMOSTRACIÓN

Construir un sistema completo y defenderlo técnicamente.

Regla del programa (no negociable):

Sin especificación validada → no se implementa código.

No es aprender herramientas. Es aprender a trabajar como ingeniero en la era de la IA.

Bloque 1

Fundamentos de Ingeniería de Software Profesional en la Era de la IA

Este bloque establece la línea base profesional común sobre la que se construye todo el máster.

No es un repaso de conceptos básicos ni aprender lenguajes.

Es el ajuste necesario para empezar a trabajar con estándar profesional. Con la IA acelerando todo, el problema ya no es saber más herramientas. El problema es trabajar sin método.

Aquí se alinean tres elementos clave: **Mentalidad de Ingeniería · Base Técnica Sólida · Criterio para usar IA**

MÓDULO 1.1 - Fundamentos de entorno y control técnico

El entorno deja de ser algo secundario y pasa a ser parte del sistema. Se trabaja:

- Uso profesional de terminal y tooling
- Git como sistema de trazabilidad real (no solo commits sueltos)
- Flujos de trabajo colaborativos
- Estructura de repositorios mantenibles

Control real sobre el código y el proceso de trabajo.

MÓDULO 1.2 - Lenguajes como herramientas (no como identidad)

Se abandona el enfoque de "especializarse en un lenguaje". Se entrena el criterio para usar cada herramienta según contexto:

- Python → rapidez y experimentación
- Java → sistemas estructurados y robustez empresarial
- TypeScript → backend moderno tipado
- C & Rust → control de rendimiento y memoria

Entender fortalezas, límites y uso real de cada stack.

MÓDULO 1.3 – IA práctica para programadores

La IA se introduce como componente técnico dentro del sistema.

Se trabaja:

- Modelo mental correcto (no determinismo, contratos estructurales)
- Prompts como artefactos versionables
- Patrones de uso: generación, auditoría y asistencia
- Validación estructural y guardrails
- Introducción a MCP, skills y agentes

Uso de IA con control, trazabilidad y criterio técnico.

MÓDULO 1.4 – Mentalidad de ingeniería profesional

Se produce el primer cambio real de enfoque: pasar de programar por tareas a diseñar sistemas.

Se trabaja:

- Separar dominio, infraestructura y operación
- Entender el sistema completo más allá del código
- Dejar de programar de forma reactiva

Base mental para trabajar con Spec-Driven Development.

Contenidos transversales en todo el bloque:

Debugging sistemático · Testing base · Uso responsable de IA

Al finalizar este bloque, el alumno:

- Trabaja con herramientas y entorno profesional real
- Entiende los lenguajes como instrumentos, no como etiquetas
- Utiliza la IA con criterio técnico, manteniendo control sobre el sistema
- Está preparado para adoptar Spec-Driven Development sin fricción

Aquí se fija el estándar sobre el que se construye todo el máster.

Bloque 2

Ingeniería de Software y Sistemas Guiada por Especificaciones (SDD)

El núcleo del máster. Aquí ocurre el cambio de modelo mental.

De programar y ajustar → a diseñar, especificar y verificar.

Este bloque enseña el método que permite que la IA sea una ventaja real. No por generar más código. Sino por construir sistemas que pueden auditarse, verificarse y operarse.

Regla del bloque: Sin especificación validada → no se implementa código.

MÓDULO 2.1 – Requisitos y especificación

- Requisitos funcionales, no funcionales, regulatorios y operativos
- Quality attributes convertidos en contratos medibles
- Criterios de aceptación (BDD / EARS)
- Trazabilidad: conectar cada decisión con su requisito
- Eliminación de ambigüedad

Resultado: requisitos claros, verificables y listos para derivar arquitectura.

MÓDULO 2.2 – Arquitectura derivada de la especificación

- Decisiones arquitectónicas justificadas desde restricciones reales
- Monolito modular vs servicios: cuándo y por qué
- Patrones modernos: Clean / Hexagonal / DDD pragmático
- Documentación de decisiones con ADRs

Resultado: arquitectura justificable, no opinión personal.

MÓDULO 2.3 – Construcción y verificación

- Planificación por slices verticales para reducir riesgo
- Clean Code y SOLID en la práctica
- Testing como verificación de contratos (no "a ver si pasa")
- Contract testing para integraciones
- CI/CD como auditor automático

Resultado: flujo profesional donde el sistema se protege solo.

MÓDULO 2.4 – Observabilidad y operación

- Métricas derivadas de lo que la spec prometió
- SLOs / SLIs y error budgets con criterio
- Telemetría estructurada: logs, métricas y trazas
- Runbooks operativos activados por alertas
- Despliegue controlado: feature flags, canary releases

Resultado: sistemas que se pueden operar de verdad, no solo enseñar.

MÓDULO 2.5 – Plataforma e infraestructura

- Docker y contenedores como base reproducible
- Cloud con criterio técnico según restricciones
- Terraform: infraestructura versionada y repetible
- CI/CD avanzado y gestión segura de secretos
- Entornos de despliegue y promoción de artefactos

Resultado: despliegues replicables, sin "en mi máquina funciona".

MÓDULO 2.6 — Economía, ética y defensa técnica

- Performance y costes como contrato (incluyendo costes de IA)
- Defensa de trade-offs con datos, sin jerga
- Responsabilidad profesional en IA: EU AI Act, NIST AI RMF
- Evaluación de riesgo integrada al proceso

Resultado: criterio profesional + capacidad de defender decisiones.

Al finalizar este bloque, el alumno:

- Diseña sistemas antes de escribir código
- Convierte ambigüedad en especificaciones verificables
- Toma decisiones arquitectónicas desde restricciones reales
- Construye con verificación continua (testing + CI/CD)
- Integra IA como componente bajo control
- Puede defender cada decisión técnica

A partir de aquí, el método se aplica en proyectos reales.

Bloque 3

Proyectos en 4 Dominios Reales

Aquí el método deja de ser teoría y pasa a aplicarse en contextos reales. El alumno demuestra que no memoriza una solución, sino que aprende el método y lo transfiere de un dominio a otro.

Marco común en cada proyecto:

- Requirements Engineering
- Spec + invariantes + criterios de aceptación
- Arquitectura derivada con decisiones justificadas
- Implementación verificable (tests y contratos)
- Pipeline de CI que bloquea cambios incorrectos
- Observabilidad básica y operación mínima defendible
- IA integrada solo donde tenga sentido, bajo control

Proyecto 1 | TELCO Foco: dominio, reglas y estados

Alta de clientes, contratación de servicios (fibra, móvil, packs), gestión de estados e incidencias.

La IA se usa para clasificación automática de incidencias y sugerencia de resolución, siempre bajo validación humana.

Tecnologías: backend con Node.js / Spring, testing contractual, CI/CD, observabilidad básica.

Proyecto 2 | BANCA Foco: contratos, trazabilidad y operaciones

Web admin (banco) + web cliente: altas y bajas de cuentas, pagos y transferencias.

El objetivo es practicar método, contratos y trazabilidad en un dominio con reglas duras, sin convertir el proyecto en "un banco real".

Tecnologías: arquitectura Clean/Hexagonal, contract testing, despliegue controlado.

Proyecto 3 | SEGUROS Foco: reglas duras + IA integrada con contratos

Web admin + web cliente: alta de pólizas, gestión de coberturas, declaración de siniestros mediante conversación asistida por IA, gestión simple de expedientes.

La IA se integra para guiar la declaración del siniestro, con contratos claros de entrada/salida y validación de outputs.

Tecnologías: IA gobernada con MCP, testing de contratos de IA, observabilidad.

Proyecto 4 | MARKETPLACE / SaaS Foco: conectar ingeniería, operación y negocio

Web admin + web cliente: planes, suscripciones, facturación, upgrades, downgrades, cancelaciones y métricas de negocio.

El proyecto conecta la ingeniería con la operación y el negocio real: SLOs, métricas de retención y coste por operación.

Tecnologías: arquitectura de suscripciones, métricas derivadas de la spec, CI/CD avanzado.

Al finalizar este bloque, el alumno:

- Aterriza un dominio nuevo sin improvisar
- Deriva arquitectura desde restricciones reales
- Construye sistemas verificables y operables
- Aplica el mismo método en distintos dominios
- Integra IA bajo control
- Defiende decisiones con criterio profesional

No aprendes recetas. Aprendes método.

Bloque 4

Proyecto Final de Máster + Defensa Técnica

Este proyecto no es académico ni guiado paso a paso. Es una simulación de entorno real: análisis → diseño → construcción → operación → defensa.

Qué se construye:

Sistema de trazabilidad y gobernanza de Supply Chain

Un sistema backend que permite:

- Registrar activos y lotes
- Registrar eventos sobre esos activos
- Mantener historial íntegro y reconstruible
- Auditar cambios, responsables y causas
- Detectar inconsistencias o anomalías mediante IA

Artefactos obligatorios (el proyecto no empieza con código):

- Requirements
- Spec + invariantes
- Event model
- Arquitectura + ADRs
- Criterios de aceptación

IA integrada con criterio:

- Detectar anomalías en secuencias de eventos
-
- Clasificar incidencias
-
- Resumir historial para auditoría
-

Siempre con: contratos claros de entrada/salida · validación de outputs · posibilidad de auditoría humana.

Defensa técnica final

El proyecto se considera aprobado cuando el alumno puede defender:

por qué su arquitectura es coherente con la spec	✓
qué trade-offs asumió y por qué	✓
cómo controla la IA	✓
cómo verifica contratos e invariantes	✓
cómo operaría el sistema en producción	✓
cuánto cuesta operarlo y por qué	✓

No se evalúa solo el código.

Se evalúa el criterio del ingeniero.

Stack y principios de trabajo

Trabajarás con un stack real, el mismo que usan los equipos de ingeniería profesionales hoy.

Lenguajes



Python



Java



TypeScript



C



Rust



Go

Backend



Node.js



Express



Spring



Flask

Frontend



React



Astro

NEXT.JS

Next.js



Tailwind

Base de datos



PostgreSQL



MongoDB



MariaDB



Redis

Infraestructura



Docker



AWS



Google
cloud



Azure



Traefik



Terraform

IA



Claude



Copilot



Codex



Cursor



Anthropic SDK



MCP

Principios de trabajo:

Diseño antes que implementación

Especificación antes que código

Verificación antes que despliegue

Responsabilidad antes que velocidad

La IA se integra como herramienta gobernada, no como sustituto del criterio.

Antes y Después

ANTES – Trabajas escribiendo código

- Empiezas por la implementación sin una base clara
- Tomas decisiones por experiencia o intuición
- La arquitectura evoluciona sobre la marcha
- Utilizas IA, pero sin control sobre su integración
- Validar el sistema depende de pruebas puntuales
- La operación se plantea cuando el sistema ya está construido

Resultado: sistemas que funcionan, pero son difíciles de explicar, verificar o mantener.

DESPUÉS — Trabajas diseñando y construyendo sistemas

- Defines qué hay que construir antes de escribir código
- Tomas decisiones a partir de restricciones reales
- La arquitectura responde a una lógica clara y documentada
- Integras la IA como componente bajo control
- Cada parte del sistema se puede verificar
- La operación se diseña desde el inicio

Resultado: sistemas que no solo funcionan. Se entienden, se pueden verificar y se pueden operar.

A partir de aquí, la diferencia no está en lo que sabes. Está en cómo trabajas.

Comunidad Internacional de Perfiles Técnicos

500+ profesionales en más de 20 países (núcleo hispanohablante)

En **CodeCrypto** formas parte de la comunidad de perfiles técnicos nº1 del mundo en habla hispana.

Desarrolladores e ingenieros que comparten el mismo objetivo: mantenerse técnicamente al día y avanzar de forma seria en su carrera.

Presencia internacional

- **Europa:** España, Portugal, Italia, Francia, Alemania, Reino Unido, Holanda, Bélgica, Irlanda y Suiza



- **LATAM:** Chile, Argentina, Uruguay, Paraguay, Venezuela, Perú, Ecuador, Colombia, El Salvador, Panamá y México



- **Norteamérica:** Estados Unidos.



Más de 500 alumnos y exalumnos conectados por la misma misión: **construir sistemas Web2 + Web3 + IA con criterio de ingeniería.**

Qué te aporta esta comunidad

- **Acompañamiento técnico:** dudas de arquitectura, smart contracts, IA aplicada, revisiones de código
- **Red profesional:** contactos en empresas y proyectos distribuidos por Europa, LATAM y EE. UU.
- **Actualización continua:** sesiones, directos y contenido nuevo alineado con lo que pide hoy la industria

</codecrypto.academy>

Titulación universitaria europea

En colaboración con la **Universidad Europea Miguel de Cervantes (UEMC)**

Al completar el Máster en Ingeniería de Software en la Era de la IA obtendrás una titulación universitaria europea, emitida por la Universidad Europea Miguel de Cervantes (UEMC).

Es un título con reconocimiento académico en el Espacio Europeo de Educación Superior y válido a nivel internacional.

Qué significa para ti

- **Reconocimiento académico europeo**
- **Válido internacionalmente:** el título puede apostillarse para su uso fuera de Europa.
- **Refuerzo en tu perfil profesional:** acredita de forma oficial tu especialización en Blockchain, Web3 e IA.



</codecrypto.academy>

La titulación es otorgada por la Universidad Europea Miguel de Cervantes (UEMC) en el marco del Espacio Europeo de Educación Superior.

</codecrypto.academy>

Addons Exclusivos del Evento · 25 plazas

Estos módulos adicionales están incluidos únicamente para quienes accedan al máster desde el evento Ingeniería del Software en la Era de la IA.

Después de esta edición, no estarán disponibles.

Plan de Carrera

Del punto en el que estás hoy → al rol técnico que quieres.

Acompañamiento profesional completo: portfolio, marca personal, entrevistas, networking y estrategia laboral.

- Marca técnica en GitHub.
- Posicionamiento estratégico en LinkedIn.
- Preparación de entrevistas.
- Roadmap profesional personalizado.



CodeCrypto Innovation – Mantente actualizado al ritmo real del ecosistema.

Acceso continuo a tecnologías, frameworks y casos de uso que están entrando en producción en empresas reales.

Durante el programa trabajas con stacks y patrones que evolucionan: nuevas integraciones de IA, herramientas de testing, despliegue y lo que usan hoy los equipos punteros.

El objetivo: que cuando termines el máster, tu forma de pensar y trabajar esté alineada con la ingeniería de software que se está construyendo ahora.

Estos addons son exclusivos para esta edición del máster. **Una vez cubiertas las 25 plazas**, ningún alumno nuevo tendrá acceso a ellos.

Plazas y Admisión

PLAZAS LIMITADAS & PROCESO DE ADMISIÓN

Esta edición abre únicamente **25 plazas** para garantizar acompañamiento real y rigor técnico.

Proceso de admisión:

- 1 Reserva tu reunión de asesoramiento
- 2 Evaluación de perfil técnico y objetivos
- 3 Confirmación de acceso al programa

Reserva tu plaza y valida si encajas en esta edición

La IA no va a sustituir a los ingenieros de software.

Va a sustituir a los ingenieros que no actualicen su forma de trabajar.

Máster en Ingeniería de Sistemas en la Era de la IA

Existen
opciones de
pago
fraccionado

Reserva tu plaza

Pago Único

1.599 \$

Reserva tu plaza

Existen
opciones de
pago
fraccionado

Reserva tu plaza

Estas son las opciones de pago para que elijas la que mejor te encaje
(posterior a haber superado el Proceso de Selección)

<https://codecrypto.academy/reserva-tu-asesoramiento-miseia-mxc/>



*Si aún no has reservado tu Reunión de Admisiones, aquí puedes buscar hueco

</codecrypto.academy>