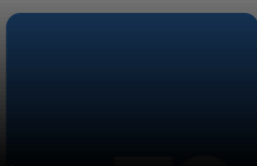


Máster en Ingeniería de Software en la Era de la IA

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

Formación técnica avanzada para programadores e ingenieros de software



Dedicación

10-15h semanales

Titulación

Universitaria Europea

Duración

4 Meses

Manifiesto

De programador ejecutor a ingeniero que diseña sistemas en la era de la IA.

La inteligencia artificial no está eliminando el desarrollo de software.

Está cambiando qué significa hacerlo bien.

Hoy puedes generar código en minutos.

Lo difícil ya no es escribir líneas.

Lo difícil es:

- decidir arquitectura con criterio
- definir contratos claros y verificables
- controlar el comportamiento de la IA
- diseñar sistemas que se puedan operar
- asumir responsabilidad técnica sobre lo construido

El valor ya no está en producir más código.

Está en construir sistemas que tengan sentido.

Que se puedan verificar y que se puedan defender.

La IA no sustituye al ingeniero.

Eleva el nivel al que hay que trabajar.

Este máster nace para responder a ese cambio.

No enseñamos herramientas aisladas.

Entrenamos una forma de pensar y construir software.

Un nuevo estándar profesional.

**La IA no sustituye al ingeniero.
Sustituye al que no sabe diseñar.**

El problema no es aprender IA

El problema es trabajar sin método en un entorno donde la IA multiplica la velocidad.

Hoy muchos desarrolladores están usando IA para producir más rápido.

Pero no necesariamente para construir mejor.

El patrón se repite:

- código generado sin especificación clara
- arquitectura decidida por intuición
- testing insuficiente o inexistente
- poca trazabilidad sobre lo que ocurre en el sistema
- sistemas difíciles de desplegar y operar
- dependencia excesiva del modelo

La IA permite avanzar más rápido.

Pero también amplifica los errores.

Sin un marco de trabajo claro, el resultado no es mejor software.

Es más complejidad.

Más deuda técnica.

Más sistemas que no escalan.

El problema no es la herramienta.

Es la ausencia de método.

Eso no es ingeniería.

**Sin método, la IA no multiplica talento.
Multiplica errores más rápido.**

Qué está cambiando en ingeniería de software

El cuello de botella ya no es escribir código.

Durante años, el valor estaba en la ejecución.

Quién escribía más código más rápido, era mejor.

En la era de la IA, ese modelo deja de ser diferencial.

El valor se desplaza:

- del código → a la especificación
- de la ejecución → a la decisión
- de la sintaxis → a la arquitectura
- de la velocidad → al control
- del “funciona” → al “es verificable, operable y defendible”

La diferencia ya no la marca quién escribe más líneas.

La marca quién sabe diseñar sistemas completos con criterio.

Porque en este nuevo contexto **el rol cambia.**

**La IA acelera la ejecución,
pero no sustituye el criterio.**

¿Para quién es?

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.

Porque el cambio no está en las herramientas.

Está en cómo se construyen los sistemas.

Este programa no te enseña a usar IA.

Te enseña a trabajar en un entorno donde la IA es parte del sistema.

¿Qué serás capaz de hacer?

Al finalizar el programa podrás trabajar como ingeniero de software en un entorno donde la IA forma parte del sistema.

No desde la intuición.

No desde la improvisación.

Desde método.

Serás capaz de:

- diseñar sistemas partiendo de especificaciones verificables
- derivar arquitectura desde restricciones reales
- construir con testing y CI como sistema de control
- integrar IA bajo gobernanza técnica
- diseñar observabilidad y operación desde el inicio
- tomar decisiones técnicas con criterio y poder defenderlas
- aplicar el mismo método en distintos dominios de negocio

No sales con teoría.

Sales con **una forma estructurada de trabajar que te permite:**

- entender lo que estás construyendo
- controlar cómo se comporta el sistema
- mantenerlo en el tiempo
- y defender cada decisión técnica

Porque en la era de la IA no es suficiente con que algo funcione. Tiene que poder explicarse, verificarse y operarse.

Estructura general

4 meses · 4 bloques progresivos

El máster está diseñado como una transición estructurada:



Cada bloque sube el nivel de exigencia.

No es consumo de contenido. Es cambio de estándar.

Empiezas alineando base y mentalidad.

Terminas siendo capaz de diseñar, construir y defender sistemas completos.

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 es 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

Qué se trabaja en este bloque

En este bloque se construye la base para trabajar sistemas con método.

Se trabaja sobre cuatro ejes:

- entorno profesional real: terminal, Git y flujos de trabajo
- lenguajes como herramientas: Python · Java · TypeScript · C/Rust
- IA aplicada a programación con control técnico
- mentalidad de ingeniería de sistemas

Aquí no se trata de acumular tecnología, se trata de entender cómo construir.

De tener control sobre el proceso y cómo se integra la IA dentro del sistema.



**Este bloque fija el estándar
y sobre él se construye todo lo demás.**

Bloque 1

Módulos del Bloque 1

Módulo 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
- flujos de trabajo colaborativos
- estructura de repositorios mantenibles

Control real sobre el código y el proceso de trabajo.

Módulo 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 para rapidez y experimentación
- Java para sistemas estructurados
- TypeScript para backend moderno tipado
- C & Rust para control de rendimiento y memoria

Entender fortalezas, límites y uso real de cada stack.

Bloque 1

Módulos del Bloque 1

Módulo 3: IA práctica para programadores

La IA se introduce como componente técnico dentro del sistema.

Se trabaja:

- modelo mental correcto (no determinismo, contratos)
- 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 4: Mentalidad de ingeniería profesional

Se produce el primer cambio real de enfoque:

- pasar de programar por tareas a diseñar sistemas
- separar dominio, infraestructura y operación
- entender el sistema completo más allá del código

Base mental para trabajar con Spec-Driven Development.

Módulo 5: Contenidos transversales

Durante todo el bloque **se trabaja de forma continua:**

- debugging sistemático
- testing base
- uso responsable de IA

Este bloque construye base técnica y empieza a cambiar la forma de pensar.

Bloque 1

Resultado del Bloque 1

Aquí el alumno deja de trabajar de forma improvisada y empieza a operar con un estándar profesional.

Este bloque marca el punto de partida del máster.

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
- estructura su trabajo con método y mantiene trazabilidad sobre lo que construye
- está preparado para adoptar Spec-Driven Development sin fricción
- tiene la base técnica necesaria para afrontar proyectos exigentes

A partir de aquí, el alumno ya no depende de la intuición.

Empieza a trabajar con orden, control y criterio.

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.

A partir de aquí, el desarrollo deja de ser improvisación.

Y pasa a ser ingeniería.

Regla del bloque:

**Sin especificación
validada** → **no se implementa
código**

Bloque 2

Qué cambia en este bloque

A partir de este punto, **la forma de trabajar cambia.**

El desarrollo deja de ser una secuencia de tareas.

Pasa a ser un proceso estructurado.

Ya no se empieza por el código.

Se empieza por entender qué hay que construir.

Este cambio se refleja en todo el proceso:

- la especificación se convierte en la base
- la arquitectura se deriva de restricciones reales
- el código se escribe como verificación
- la IA se integra como componente bajo control
- la operación se diseña desde el inicio
- los costes se consideran parte del sistema

Aquí dejas de programar por intuición.

Empiezas a trabajar con un sistema completo:

donde cada decisión tiene un motivo,

cada parte se puede verificar,

y todo el conjunto se puede operar.

**Porque en este nivel, no basta con que algo funcione.
Tiene que poder explicarse, verificarse y operarse.**

Bloque 2

Cómo se construye un sistema en este bloque

A partir de este punto, el desarrollo deja de ser una secuencia de tareas.

Pasa a ser un **proceso estructurado, donde todo tiene propósito y orden.**

Ya no se construye empezando por el código.

Se construye entendiendo el problema, definiendo qué hay que hacer y tomando decisiones con criterio desde el inicio.

Cada parte del sistema responde a una necesidad concreta.

Cada decisión tiene una justificación.

Y todo el proceso sigue una lógica clara.

El sistema se construye siguiendo un flujo claro:

Del problema a una arquitectura justificable

Todo empieza en la especificación. No en el código. Se trabaja:

- definición precisa de requisitos
- eliminación de ambigüedad
- trazabilidad entre decisiones y necesidades
- derivación de arquitectura desde restricciones reales

La arquitectura deja de ser opinión. Pasa a ser consecuencia.

Aquí empieza el sistema.

En la siguiente fase, se construye, se valida y se opera.

Bloque 2

Cómo se construye un sistema en este bloque

Construcción verificable, operación real e infraestructura reproducible

El sistema no se construye para que funcione.

Se construye para poder verificarse, desplegarse y operarse. Se trabaja:

- slices verticales y planificación con criterio
- testing como verificación de contratos
- CI/CD como auditor automático
- observabilidad: logs, métricas y trazas
- despliegues controlados
- Docker, cloud y Terraform

El sistema se protege solo. No depende de la intuición.

Coste, riesgo y defensa técnica

El sistema no solo tiene que funcionar. Tiene que poder justificarse. Se trabaja:

- performance y costes como parte del sistema
- evaluación de riesgos
- defensa de decisiones técnicas
- responsabilidad profesional en IA

No basta con construir algo que funcione.

Hay que poder defender por qué está diseñado así.

**A partir de aquí, cada decisión tiene una base.
Y cada parte del sistema responde a un criterio.**

Bloque 2

Módulos del Bloque 2

Este bloque no se aprende como contenido aislado.

Se entrena como un sistema completo.

Los módulos se organizan siguiendo el flujo de construcción que has visto:

Del problema a una arquitectura justificable

- **Módulo 1: Requisitos y especificación**
 - requisitos funcionales y no funcionales
 - criterios de aceptación medibles
 - eliminación de ambigüedad
 - trazabilidad entre decisiones y necesidades
- **Módulo 2: Arquitectura derivada de la especificación**
 - decisiones arquitectónicas justificadas
 - monolito vs servicios según contexto
 - patrones como Clean, Hexagonal y DDD
 - documentación mediante ADRs

Aquí se define qué hay que construir y cómo se estructura el sistema desde restricciones reales.

La base sobre la que se toman todas las decisiones.

Bloque 2

Módulos del Bloque 2

Construcción verificable, operación real e infraestructura reproducible

- **Módulo 3: Construcción y verificación**
 - planificación por slices
 - Clean Code y SOLID en la práctica
 - testing como verificación de contratos
 - CI/CD como sistema de control automático
- **Módulo 4: Observabilidad y operación**
 - métricas derivadas de la especificación
 - SLOs, SLIs y gestión de errores
 - logs, métricas y trazas
 - despliegue controlado
- **Módulo 5: Plataforma e infraestructuran**
 - Docker y contenedores
 - cloud con criterio técnico
 - Terraform como infraestructura versionada
 - gestión de entornos y despliegues

Aquí el sistema se implementa, se valida y se prepara para operar en condiciones reales.

No se construye para que funcione, se construye para poder controlarlo.

Coste, riesgo y defensa técnica

- **Módulo 6: Economía, ética y defensa técnica**
 - costes y performance como parte del sistema
 - evaluación de riesgos
 - defensa de decisiones técnicas
 - responsabilidad en el uso de IA

Aquí el sistema pasa a ser responsabilidad del ingeniero.

Cada decisión tiene un coste, un riesgo y una justificación.

Bloque 2

Resultado del Bloque 2

Este bloque marca el cambio real en la forma de trabajar.

Al finalizar este bloque, el alumno:

- diseña sistemas antes de escribir código
- convierte ambigüedad en especificaciones claras y verificables
- toma decisiones arquitectónicas a partir de restricciones reales
- construye con verificación continua (testing + CI/CD)
- integra IA como componente bajo control
- diseña sistemas que se pueden desplegar, operar y mantener
- entiende el coste, el riesgo y el impacto de cada decisión

**Aquí es donde el desarrollo deja de ser ejecución.
Y pasa a ser diseño, control y responsabilidad.**

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.

En cada proyecto se trabaja siguiendo el mismo flujo:

- requirements engineering
- spec + invariantes + criterios de aceptación
- arquitectura derivada
- implementación verificable
- pipeline de CI
- observabilidad básica
- IA integrada solo donde tenga sentido, bajo control

El objetivo no es construir un proyecto concreto.

Es aprender a aplicar el mismo método en distintos contextos.

Porque el valor está en la capacidad de transferir el método.

Regla del bloque:

**Sin especificación
validada**



**no se implementa
código**

Bloque 3

Los 4 proyectos del máster

TELCO

Alta de clientes, contratación de servicios, estados e incidencias.

Foco: dominio, reglas y estados.

BANCA

Web admin + web cliente para cuentas, pagos y transferencias.

Foco: contratos, trazabilidad y operaciones.

SEGUROS

Pólizas, coberturas, siniestros y gestión simple de expedientes.

Foco: reglas duras + IA integrada con contratos.

MARKETPLACE / SaaS

Planes, suscripciones, facturación, upgrades, cancelaciones y métricas.

Foco: conectar ingeniería, operación y negocio.

Bloque 3

Qué demuestra este bloque

Aquí es donde el método deja de ser teoría y se convierte en capacidad real.

Al finalizar este bloque, el alumno habrá aplicado el método SDD en **4 contextos** distintos y será capaz de:

- aterrizar un dominio sin improvisar
- derivar arquitectura desde restricciones reales
- construir sistemas verificables y operables
- aplicar el mismo método en distintos dominios
- integrar IA bajo control
- defender decisiones con criterio profesional

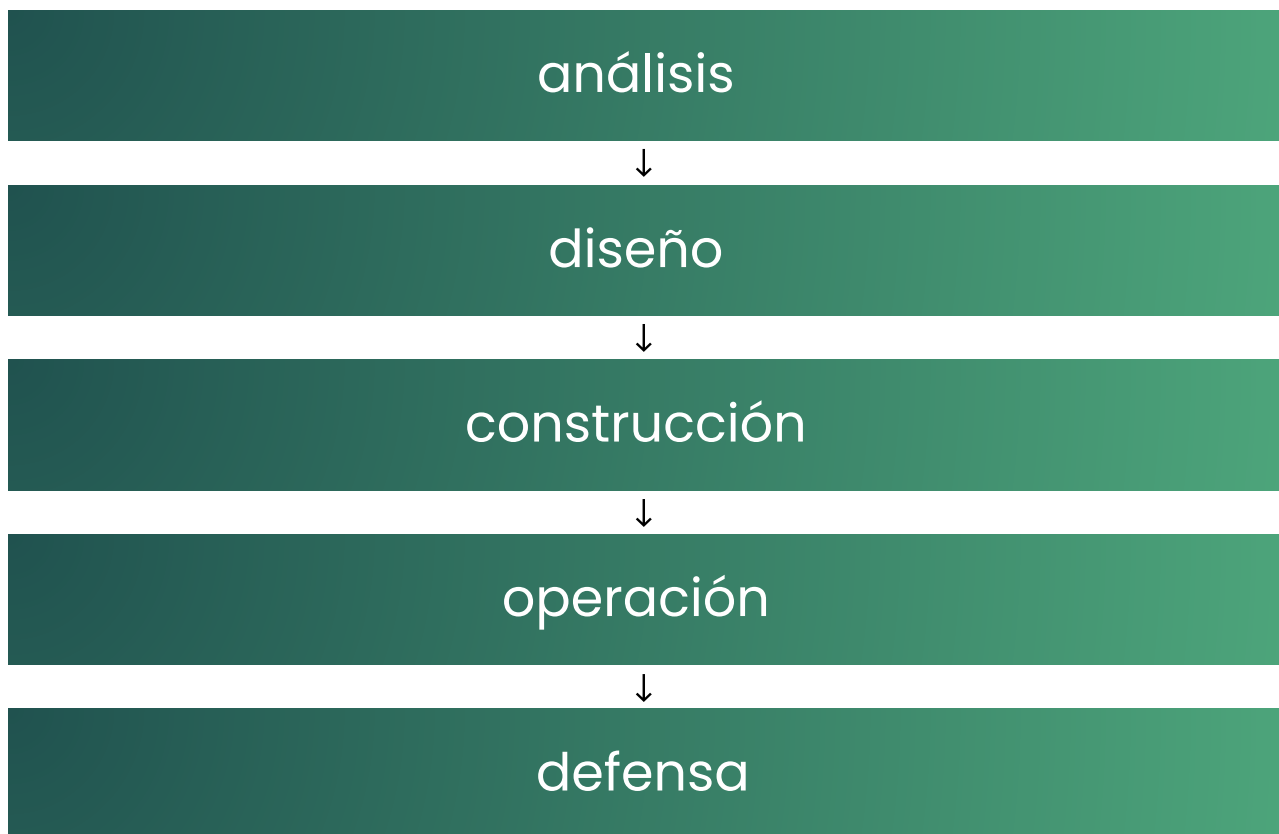
No aprendes recetas
Aprendes método

Bloque 4

Proyecto Final de Máster + Defensa Técnica

Este proyecto no es académico ni guiado paso a paso.

Es una simulación de entorno real:



Se construye un sistema de trazabilidad y gobernanza de Supply Chain donde el foco está en dominio, datos, estados, reglas y auditabilidad.

Bloque 4

Qué incluye el Proyecto Final

El proyecto no empieza con código.

Artefactos

- requirements
- spec
- invariants
- event model
- arquitectura + ADRs
- criterios de aceptación

IA

- detectar anomalías
- clasificar incidencias
- resumir historial para auditoría

Control

- contratos claros de entrada y salida
- validación de outputs
- auditoría humana posible

El sistema se implementa como verificación de lo especificado.

Bloque 4

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:

Lenguajes



Backend



Frontend



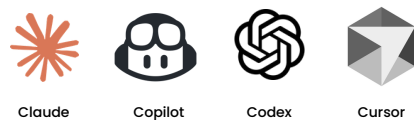
Base de datos



Infraestructura



IA



Testing



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.

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

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

- Actualización continua con tecnologías emergentes.
- Acceso de por vida a actualizaciones
- Tu acceso sigue activo.
- El contenido evoluciona contigo.

Promoción válida solo para la edición 2026 lanzada durante el evento Ingeniería de Software en la Era de la IA

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>

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>

Equipo Docente

Ingenieros en Activo con experiencia real en:



Claudio Briceño – Director del Máster & CEO de CodeCrypto



José Viejo – Director Académico & Arquitectura de Sistemas Web2 + Web3 + IA



Andrés León – Lead Técnico Web3 Avanzado & CodeCrypto Innovation



Lindsey Walker – Plan de Carrera & Posicionamiento Profesional



María Jose Rabellino – Soporte Técnico & Acompañamiento del Alumno

Formación exigente, rigor técnico y acompañamiento real.

Antes y Después

Después de ver el recorrido completo, la diferencia es clara.

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
- 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.

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

<p>Existen opciones de pago fraccionado</p> <p>Reserva tu plaza</p>	<p>Pago Único</p> <hr/> <p>1.299 \$</p> <p>Reserva tu plaza</p>	<p>Existen opciones de pago fraccionado</p> <p>Reserva tu plaza</p>
---	--	---

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-arg/>



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

</codecrypto.academy>