

SYLLABUS DEL CURSO

De la Idea al Producto con AI

Cómo llevar una idea de negocio hasta una aplicación real, lanzada y funcionando, usando inteligencia artificial como copiloto de desarrollo.

Nivel: introductorio, sin experiencia previa en programación. **Para:** estudiantes de 5.º y 6.º año.

Formato: sesiones de 2 horas, dos veces por semana. **Duración estimada:** unas 16 sesiones (8 semanas).

Sobre el instructor

Mi nombre es Eyal Cumartesi. Soy fundador y director de tecnología (CTO) de una empresa de software, y llegué a ese puesto siendo autodidacta: aprendí a programar por mi cuenta y pasé directamente a liderar la arquitectura técnica de una plataforma completa que se usa en producción todos los días.

La plataforma que co-fundé y construyo, **Spexs AI**, es un sistema de inteligencia de negocios para empresas: conecta datos de distintos sistemas administrativos y contables, los procesa y limpia de forma automática, arma reportes y análisis financieros, y usa AI para interpretarlos. Es un producto que atiende a varias empresas a la vez sobre una misma plataforma, así que en el día a día trabajo justo con los temas que verán en este curso: bases de datos, procesamiento de datos, infraestructura en la nube, monitoreo y automatización de agentes de AI.

Además de la parte técnica, tengo el lado del negocio: estudié economía y ciencia de datos. Esa mezcla de fundador, dueño de negocio y CTO es justo la mirada que quiero pasarles: no solo cómo se programa algo, sino por qué se construye, para quién y cómo se sostiene como negocio.

Lo que van a aprender aquí no es un tutorial suelto. Es el flujo de trabajo real de alguien que todos los días decide cómo construir, cómo lanzar y cómo mantener una aplicación viva, incluyendo las decisiones aburridas pero importantes que hacen que un producto no se caiga cuando de verdad lo usa gente.

De qué trata el curso

Es un curso completamente práctico. Cada quien va a partir de una idea de negocio propia y, al terminar, va a tener una aplicación funcionando de verdad, publicada en internet y con herramientas para saber cómo la usa la gente.

La meta no es solamente "hacer una app". Es entender el camino completo: pensar bien la idea, construirla con buenas bases, sacarla al mundo, mantenerla y mejorarla con el tiempo. Ese recorrido, de principio a fin, es lo que casi nunca se enseña y es lo que de verdad importa.

Lo que vas a saber hacer al terminar

- Convertir una idea en un plan sencillo y validarla con un pequeño estudio de mercado.
- Dejar tu computadora lista para desarrollar, desde cero.
- Trabajar con AI de forma efectiva, sin desperdiciar tu tiempo ni tus mensajes.
- Usar Git y GitHub, y conectarlos con el AI para que trabaje sobre tu proyecto.
- Darle una interfaz limpia y usable a tu app, sin ser diseñador.
- Tomar decisiones de arquitectura con criterio: qué base de datos, cuánta gente la va a usar, qué pasa si crece.
- Cuidar lo básico de seguridad: contraseñas, secretos y quién puede ver qué datos.
- Probar que tu app funciona con pruebas automáticas, para no romper lo que ya servía.
- Publicar tu aplicación en internet y mantenerla viva con monitoreo y analítica.
- Automatizar tareas repetitivas con agentes de AI.

o. Antes de empezar

Estas son las cosas que conviene tener listas antes de la primera clase.

0.1 Una cuenta de AI para programar. Recomendando Claude, que es lo que usaremos en clase.

Sobre el plan de AI. El plan básico (Pro) sirve para arrancar. Sin embargo, es preferible el plan Max si tienes cómo, porque da bastante más margen de uso y evita quedarte sin mensajes justo en medio de un momento de trabajo intenso. No es obligatorio: nuestras sesiones son de 2 horas y no vamos a estar programando sin parar todo el tiempo, así que con Pro se puede llevar el curso. En clase además aprenderemos a cuidar el uso (punto 2.3) para que rinda.

0.2 Una cuenta de GitHub. Es gratis y es donde va a vivir tu código. (la crearemos en el curso si no la tienes)

0.3 Una computadora en la que puedas instalar programas

0.4 Un editor de código instalado. Usaremos Cursor, que es gratis.

0.5 Ganas de aprender. No necesitas saber programar de antes, solo manejar la computadora a nivel básico.

Contenido del curso

1. La idea y cómo saber si sirve

1.1 Cómo se le ocurren buenas ideas de negocio y cómo descartar las que no valen la pena.

1.2 Propuesta de valor: qué problema resuelves y para quién exactamente.

1.3 Un plan de negocio sencillo, sin complicaciones.

1.4 Estudio de mercado y mirar a la competencia.

1.5 Definir el MVP: qué construir primero y qué dejar para después.

1.6 Usar el AI para investigar y ordenar tus ideas en esta etapa.

2. Aprender a trabajar con el AI

2.1 Cómo pedirle bien las cosas al AI: darle contexto, ser claro, dar ejemplos.

2.2 El ida y vuelta con el AI: pedir, revisar lo que hizo, corregir.

2.3 Cuidar el uso: cuándo empezar una conversación nueva, por qué los mensajes pesados gastan más, cómo no quedarte sin cupo.

2.4 Diferencia entre pedirle que te explique algo y pedirle que construya algo por ti.

3. Preparar tu computadora para programar

3.1 Instalar lo necesario: Node.js, la terminal y el editor.

3.2 Cómo se organiza un proyecto en carpetas.

3.3 Comandos básicos de la terminal, sin miedo.

3.4 Comprobar que todo quedó bien instalado.

4. Guardar tu trabajo: Git y GitHub

- 4.1 Qué es Git y por qué todo programador lo usa.
- 4.2 Guardar cambios (commits), y trabajar en ramas sin romper lo que ya funciona.
- 4.3 Los pull requests: proponer cambios y revisarlos.
- 4.4 Buenas costumbres al describir tus cambios.

5. Conectar GitHub con el AI

- 5.1 Cómo darle al AI acceso a tu repositorio usando skills (más rápido y simple que el MCP).
- 5.2 Cómo el AI lee tu código y hace cambios directamente.
- 5.3 El flujo completo: tú pides, el AI trabaja, GitHub lo guarda.

6. Poner buenas bases al proyecto

- 6.1 Reglas del proyecto: instrucciones para que el AI trabaje como tú quieres.
- 6.2 Skills propios del proyecto para tareas que se repiten.
- 6.3 Elegir el lenguaje. En este curso trabajaremos con TypeScript.
- 6.4 Protección por tipado: cómo TypeScript te avisa de errores antes de que ocurran.
- 6.5 Herramientas que revisan y ordenan tu código solas (linters y formatters), para que no tengas que pensar en eso.
- 6.6 Dejar todo configurado de una vez para que el AI respete las reglas sin que se lo recuerdes.

7. Decisiones importantes antes de construir

- 7.1 Qué preguntarte antes de empezar, sin complicarte de más.
- 7.2 Cuánta gente va a usar tu app al mismo tiempo y por qué eso cambia las cosas.
- 7.3 Tipos de base de datos: SQL y NoSQL, y cuándo usar cada una.
- 7.4 Ideas básicas para que tu app no se caiga cuando la usa mucha gente.
- 7.5 Entender que casi toda decisión es un intercambio: ganas algo, cedes algo.
- 7.6 Cuidar lo básico de seguridad desde el inicio: variables de entorno, contraseñas y secretos que nunca se suben a GitHub, y reglas de quién puede ver qué datos.

8. Darle cara a tu app: la interfaz

- 8.1 Cómo se ve una app y por qué el diseño importa aunque no seas diseñador.
- 8.2 Estilos rápidos con Tailwind, sin pelear con el CSS.
- 8.3 Componentes listos y bonitos con librerías como shadcn/ui.
- 8.4 Usar el AI para armar pantallas y ajustar el diseño a lo que imaginas.

9. Construir, probar y mejorar en tu computadora

9.1 Correr tu app en local y verla funcionar.

9.2 El ciclo de mejora: construir, probar, ajustar, repetir.

9.3 Encontrar y arreglar errores con ayuda del AI.

9.4 Pruebas automáticas: pequeños chequeos que revisan solos que tu app siga funcionando cuando haces cambios.

9.5 Cómo darte cuenta de que ya está lista para publicarse.

10. Publicar tu app en internet

10.1 Qué significa publicar (hacer deploy) y qué pasa por detrás.

10.2 Herramientas modernas que lo hacen fácil, como Vercel, y bases de datos gestionadas como Supabase.

10.3 Los grandes proveedores de nube: AWS y Google Cloud, sus diferencias y cuándo tiene sentido cada uno (a nivel general, sin entrar en lo técnico profundo).

10.4 Una máquina para ti solo, o una máquina compartida entre muchos clientes: qué significa cada opción.

10.5 Cómo agregar cuentas de usuario (login) y, si aplica, cobros a tu aplicación.

10.6 Publicación automática: que cada cambio que apruebas en GitHub salga solo a internet (CI/CD).

10.7 Pensar en el costo desde temprano para no llevarte sorpresas.

11. Mantener la app viva y saber qué pasa con ella

11.1 Detectar errores en tiempo real con Sentry.

11.2 Entender cómo la usa la gente con PostHog (qué botones tocan, dónde se traban).

11.3 Cuidar tu base de datos: respaldos y por qué conviene que alguien más la gestione por ti.

11.4 Leer estas señales para decidir qué mejorar después.

12. Automatizar lo repetitivo con agentes

12.1 Agentes que abren solos los pull requests para mantener el proyecto actualizado.

12.2 Automatizar tareas de mantenimiento que se repiten cada cierto tiempo.

12.3 Cuándo vale la pena automatizar y cuándo es mejor hacerlo a mano.

13. Cierre y presentación final

13.1 Cada quien presenta su aplicación funcionando frente al grupo.

13.2 Una lista de verificación de "lista para el mundo real".

13.3 Hacia dónde seguir y qué estudiar por tu cuenta después.

Cómo serán las clases

Las clases están pensadas sobre todo para aprender los conceptos y tomar decisiones juntos: entender el porqué de cada cosa, ver ejemplos y resolver dudas en el momento. Cada sesión tiene una parte de explicación, que casi no consume mensajes del AI, y una parte de trabajar sobre tu propio proyecto.

Para avanzar más rápido, buena parte del trabajo de construir se lleva como tarea a la casa. En clase aprendes el concepto y lo empiezas; en casa lo terminas y lo practicas con calma. Así aprovechamos el tiempo juntos para lo que de verdad necesita explicación y acompañamiento, y no para esperar a que algo termine de instalarse o de correr.

Y no te quedas solo con las tareas: me pueden escribir cuando lo necesiten para hacer preguntas. Si te trabas en casa, mandas tu duda y la vemos, sin tener que esperar hasta la próxima clase.

Lo más importante: vas a trabajar sobre tu idea durante todo el curso. Cada tema que veamos se aplica de inmediato a algo tuyo, no a un ejercicio de mentira.