

# Make III y Repaso del Curso

Coding for No Coders

# UAX

Ivan Eguiguren

# PLAN PARA EL ÚLTIMO DÍA DE CLASE

## REPASO Y ENTREGABLES

- 1) Repasar conceptos para el examen final
- 2) Repasar **QUÉ 5 ENTREGABLES** hay que enviar por la plataforma como **ENTREGABLE FINAL.**

## ESCENARIOS DE MAKE

- 3) Ver los dos últimos escenarios de Make

# Repaso para el examen

Coding for No Coders

 UAX

Ivan Eguiguren

# El Mapa del Ecosistema IA

PENSAR

## LLMs Conversacionales

ChatGPT, Claude, Gemini...

Herramientas diseñadas para el razonamiento, la ideación y el procesamiento de lenguaje natural.

CONECTAR

## Automatización de Flujos

Make, n8n, Zapier...

El "músculo" que orquesta procesos y une diferentes plataformas sin código.

CONSTRUIR

## Vibe Coding

Bolt, Lovable, v0...

Creación acelerada de aplicaciones y productos funcionales mediante lenguaje natural.

⚠ El error más caro: Confundir el terreno de cada una.

**ESTRATEGIA**

Cómo "Hablar" con una IA

**El prompt es una interfaz, no una conversación casual.**

NIVEL  
1

Prompt Básico  
"Resume esto"

NIVEL  
2

Estructurado  
Rol + Contexto + Tarea +  
Formato + Ejemplos

NIVEL  
3

Meta-prompting  
Usar la IA para mejorar tus  
propios prompts.

**Lo que NO funciona:** "Ser cortés", "hablar en inglés porque es mejor", "repetir muchas veces".

# Automatización vs Cognición

Dos mundos diferentes:



## Mundo determinista (flujos)

- Si pasa X, entonces Y
- Datos estructurados, reglas claras
- Ejecución predecible 100 de 100 veces



## Mundo cognitivo (LLMs)

- Interpretar lenguaje natural
- Generar, reescribir, clasificar
- Criterio y matiz

La pregunta correcta siempre es: ¿esto requiere REGLAS o requiere CRITERIO?

## ESTRATEGIA

# Las señales de cada tipo de tarea

Pistas para identificarlas en el lenguaje del cliente



"cada vez que...", "si el cliente...",  
"mover de A a B"

→ **Workflow / Automatización**



"resumir", "analizar", "reescribir",  
"entender"

→ **LLM**



"navegar por", "hacer clicks en",  
"rellenar formularios"

→ **Agente**



"quiero una app", "un dashboard",  
"una interfaz"

→ **Vibe Coding**

# Agentes: el salto más reciente

¿Qué los diferencia de un LLM normal?



## Un LLM normal responde

- Procesa y genera texto de forma pasiva.
- Requiere que el usuario sea el "motor" de ejecución.



## Un agente actúa

- Lee pantalla, hace clicks, navega.
- Toma decisiones autónomas.
- Scraping, research y tareas de navegación.

**Nuevos retos:** Los agentes son más lentos, más caros y pueden romperse al navegar entornos dinámicos.

NO-CODE

# No-Code NO es lo que crees

Aclaremos malentendidos:



No significa "sin tecnología"



No significa "sin pensar"



No significa "sin lógica"

Significa: aplicar lógica de programación (condiciones, bucles, variables) sin escribir código.

**Significa siendo programación. Sigue requiriendo pensamiento estructurado.**

## ESTRATEGIA

## El ciclo de construir algo con IA

1

**Definir el problema**

qué duele hoy, a quién

2

**Especificar requisitos**

qué debe hacer, qué NO

3

**Documentar comportamiento**

casos de uso, ejemplos

4

**Construir**

solo ahora toca la herramienta

Este documento tiene un nombre: **PRD (Product Requirements Document)**. Es estándar en la industria.

Sin él: construyes algo, el cliente dice 'no es esto', y rehaces tres veces.

# LLM vs Vibe Coding

¿Cuándo utilizar cada uno?



## LLM (Razonamiento)

- Procesar lenguaje natural.
- Generar, reescribir y clasificar.
- Tareas que requieren criterio y matiz.



## Vibe Coding (Construcción)

- Crear apps y webs funcionales.
- Describir lo que quieres en lenguaje natural.
- Prototipado rápido sin saber programar.

El **LLM** es para pensar y procesar; el **Vibe Coding** es para materializar productos digitales.  
Usa LLM si necesitas **criterio**, usa Vibe Coding si necesitas **una aplicación**.

**ESTRATEGIA**

# Arquitectura de soluciones reales

**✗ Trampa del principiante**

**"Tengo un problema → voy a resolverlo con [mi herramienta favorita]"**

**✓ Enfoque profesional**

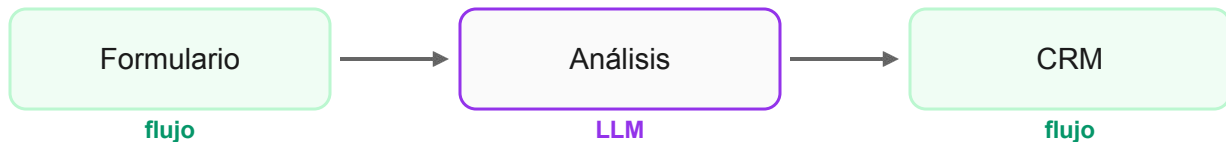
**"Descomponer el problema en pasos → asignar la herramienta correcta a cada paso"**

Un proyecto real rara vez usa una sola tecnología. Usa **flujo + LLM + UI + base de datos**, cada pieza donde encaja.

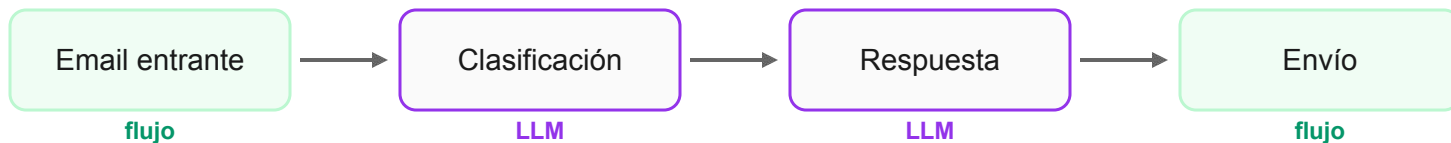
## ESTRATEGIA

# Ejemplos de orquestación: patrones reales

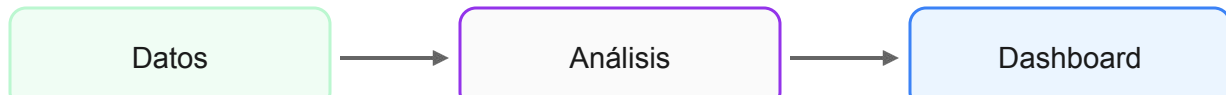
## Lead scoring



## Atención al cliente



## Reporte automático



Cada paso usa la herramienta adecuada.

## Qué llevarse de este curso

Tres ideas. Todo lo demás es detalle.

01

**La herramienta correcta para la  
tarea correcta**

No hay una bala de plata.

02

**Define antes de construir**  
PRD, aunque sea mental.

03

**Tu valor es el criterio**  
No la ejecución técnica.

# Presentación de los retos

Coding for No Coders

 UAX

Ivan Eguiguren

## ESTRATEGIA

## Una sola entrega, un solo documento



Todos los miembros del grupo suben el portfolio



Es el MISMO documento (idéntico) para todos



Un único archivo (no ficheros sueltos)

No es "cada uno sube lo suyo". Es el **mismo archivo compartido** por todo el grupo.

# Qué debe contener la portada del documento

## Datos obligatorios

- Título del Proyecto
- Listado completo de nombres de integrantes
- Número de grupo y Reto correspondiente

TÍTULO DEL PROYECTO

---

Nombre Apellido 1  
Nombre Apellido 2  
Nombre Apellido 3  
Nombre Apellido 4

**Grupo 1 - Reto 1**  
blablabla

 Si falta un nombre, esa persona no tiene nota.

# Repaso del curso

Coding for No Coders

 UAX

Ivan Eguiguren

REPASO

## El mapa del curso

S1

### Oportunidades No-Code & IA

Tipos de automatización  
y el concepto Agente

S2

### LLMs y Prompts

Arquitectura del prompt,  
Claude Projects

S3-4

### Vibe Coding

HTML/CSS, webs  
estáticas, Single Page  
Apps

S5-7

### Make y Workflows

Triggers, módulos, lógica  
condicional, IA

¿Dudas?

**SESIÓN 1**

# Oportunidades No-Code & IA

**RPA**

El pasado: robots que replican acciones manuales en pantalla

**Workflows**

Automatización de tareas predefinidas con datos estructurados

**LLMs**

Modelos que razonan, generan y transforman contenido

**Vibe Coding**

Crear apps funcionales usando lenguaje natural

**Agentes**

IA que no solo responde: ejecuta tareas de forma autónoma

**PoC vs Producción**

Mínimo viable para validar vs producto robusto y escalable

 ¿Dudas?

## SESIÓN 2

# LLMs y Prompts

## Arquitectura del Prompt

**Rol:** Quién es la IA en este contexto

**Contexto:** Qué información necesita

**Tarea:** Qué tiene que hacer exactamente

**Formato:** Cómo estructurar la respuesta

## Conceptos clave

### Meta-prompting

Usar la IA para generar mejores prompts

### Claude Projects

Asistente con tu información entrenada

### Limitaciones

Contexto, alucinaciones, temperatura

### Validación

Siempre valida el resultado: la IA falla

 ¿Dudas?

**SESIONES 3-4**

# Vibe Coding

## Lo que vimos

- HTML: estructura jerárquica de etiquetas
- CSS: diseño visual y estructura
- Javascript: lenguaje de programación de páginas web
- Webs estáticas: landings
- Webs dinámicas: flightwatch, chat

## Lo que construísteis

### Flightwatch

Página visual de seguimiento de aviones, iterando estructura y diseño y usando API

### Single Page App

Aplicación de chat donde utilizamos websockets

### Landing Page

Página que conectamos con make para enviar los datos de un formulario

¿Dudas?

SESIONES 5-7

## Make y Workflows

### 1 Clima → Email

Trigger programado · API del tiempo · envío automático

Módulos:

Schedule

Weather

Email

### 2 Formulario → Airtable

Formulario web dispara el flujo · datos guardados en Airtable

Módulos:

Web Form

Airtable

### 3 Airtable + LLM → Email

Lee registros · filtra por deporte · LLM resume · envía email

Módulos:

Airtable

Filter

LLM

Gmail

### 4 Bot de Telegram

Recibe información del chat, hace cosas, responde

Módulos:

Telegram

Email

LLM

## PERSPECTIVA

## Más allá de Make: el ecosistema

## Zapier

## ✓ A favor

- 7.000+ integraciones
- El más fácil de usar
- Bien documentado

## ✗ En contra

- El más caro
- Menos flexible para lógica compleja

*Velocidad y presupuesto no es problema*

## Make

## ✓ A favor

- Visual e intuitivo
- Buen balance precio/capacidad
- Escenarios complejos

## ✗ En contra

- Coste escala con operaciones
- Datos en servidores externos

*Casos profesionales sin necesidad de autoalojar*

## n8n

## ✓ A favor

- Open source y gratuito
- Self-hosted: control total
- Sin límite operaciones

## ✗ En contra

- Más setup técnico
- Mantenimiento propio

*Control de los datos o escala sin coste*

¿Dudas?

**CIERRE**

## ¿Qué sabéis hacer ahora?



### Usar LLMs

Prompts estructurados y asistentes entrenados a medida



### Crear webs

Landings y apps estáticas sin escribir código



### Automatizar

Flujos con triggers, módulos y lógica condicional



### Integrar IA

LLMs como módulo dentro de un workflow real



### Evaluar herramientas

Elegir entre Make, n8n o Zapier según el caso



### Pensar en sistemas

Identificar oportunidades y diseñar soluciones

 ¿Dudas?

No aprendisteis a programar.

**Aprendisteis algo mejor.**

---

*Aprendisteis a tener la visión para automatizar sistemas*

# Recursos del Curso

Encuentra todas las presentaciones y la plantilla para la entrega de los retos aquí:

<https://ivaneguiguren.me/UAX/>

## La hora de la foto



[ivan.eguiguren@thevalleylive.es](mailto:ivan.eguiguren@thevalleylive.es)