

DIPLOMADO EN IA Y AUTOMATIZACIÓN | TOP LEARNING

# MÓDULO 5

## Vibe Coding Básico y Análisis de Datos

MANUAL DE EJEMPLOS, DIAGRAMAS Y PROMPTS LISTOS

Nivel 1: IA Generativa y Productividad Automatizada

6 Temas | Dashboards | Apps sin código | Datos con IA | Límites y cuando escalar

*Windsurf \* v0 \* Lovable \* Bolt \* Supabase \* Firebase \* Vercel \* GitHub \* n8n \* Databricks*

## Introducción al Módulo 5: Cuando el lenguaje natural se convierte en software

El Vibe Coding es el nuevo paradigma de desarrollo de software: en lugar de aprender un lenguaje de programación, describes en lenguaje natural lo que quieres construir y la IA genera el código. No es un sustituto del desarrollo profesional para proyectos críticos y complejos, pero sí es una revolución para todos los que necesitan construir herramientas, dashboards y aplicaciones sin depender de un equipo de desarrollo.

Este módulo combina dos habilidades que hasta hace poco estaban separadas en mundos distintos: la capacidad de construir software y la capacidad de analizar datos. Con Vibe Coding puedes crear la herramienta que visualiza tus datos. Con el análisis conversacional puedes extraer insights de esos datos usando lenguaje natural. Juntas, estas habilidades te dan un poder que antes solo tenían los equipos de data science.

### Al terminar este modulo podras...

- \* Crear dashboards con graficas reales de tus datos en menos de 30 minutos
- \* Generar apps y herramientas internas describiendo lo que necesitas en texto
- \* Conectar Google Sheets, APIs y bases de datos a visualizaciones en vivo
- \* Analizar archivos de Excel y CSV con preguntas en lenguaje natural
- \* Identificar tendencias, patrones y anomalías en tus datos sin fórmulas complejas
- \* Saber exactamente cuando el Vibe Coding alcanza su límite y cuando necesitas desarrollo profesional

### Herramientas principales del módulo:

- \* Windsurf: editor de código con IA agéntica de Codeium
- \* v0 de Vercel: componentes React para interfaces profesionales
- \* Lovable y Bolt.new: apps completas desde descripción de texto
- \* Supabase y Firebase: bases de datos en tiempo real para tus apps
- \* Vercel y GitHub: publicación y control de versiones
- \* Databricks: análisis de datos a escala empresarial
- \* ChatGPT y Claude: análisis conversacional de archivos CSV/Excel

## Herramientas del Módulo 5: El Ecosistema de Vibe Coding y Datos

Las herramientas de este módulo cubren el ciclo completo: construir la interfaz, conectar los datos, publicar la aplicación y analizar los resultados. Todas tienen acceso gratuito o periodo de prueba suficiente para los ejercicios.

### Categoría 1: Editores de código con IA agéntica

Herramienta	Empresa	Que es	Diferencial	Costo	URL
Windsurf	Codeium - EE.UU.	Editor de código completo con IA agéntica que puede leer, planificar y ejecutar cambios en múltiples archivos de forma autónoma	Cascade: el agente que actúa en todo el proyecto, no solo en una línea	Gratis (plan Free); Pro: \$15 USD/mes	windsurf.com
Cursor	Anysphere - EE.UU.	Editor con IA integrada (visto en Modulo 4); incluido aqui por su uso en Vibe Coding avanzado	Composer agéntico para cambios multi-archivo	Gratis; Pro: \$20 USD/mes	cursor.com

### Categoría 2: Generadores de apps y componentes (Vibe Coding no-code)

Herramienta	Empresa	Que genera	Stack tecnológico	Costo	URL
v0	Vercel - EE.UU.	Componentes de interfaz React/Next.js de alta calidad desde descripción en texto	React, Next.js, Tailwind CSS, shadcn/ui	Gratis con límites; Pro: \$20/mes	v0.dev
Lovable	Lovable - Suecia	Apps web completas con frontend + backend + base de datos desde descripción	React, TypeScript, Supabase, Tailwind	Gratis (5 msg/dia); Starter: \$20/mes	lovable.dev
Bolt.new	StackBlitz - EE.UU.	Apps completas con editor de código en vivo y exportación del proyecto	React, Vue, Svelte, Angular (a elegir)	Gratis con límites; Pro: \$20/mes	bolt.new

### Categoría 3: Bases de datos y backend en tiempo real

Herramienta	Empresa	Que ofrece	Mejor para...	Costo	URL
Supabase	Supabase Inc. - EE.UU.	Base de datos PostgreSQL gestionada con API REST automática, autenticación y almacenamiento. Open source.	Apps de Lovable, proyectos con datos estructurados, autenticación de usuarios	Gratis (2 proyectos); Pro: \$25/mes	supabase.com
Firebase	Google - EE.UU.	Base de datos NoSQL en tiempo real (Firestore) + autenticación + hosting. Ecosistema Google completo.	Apps que necesitan sincronización en tiempo real, proyectos ya en ecosistema Google	Gratis (plan Spark); Blaze: pago por uso	firebase.google.com

### Categoría 4: Publicación, control de versiones y CI/CD

Herramienta	Empresa	Que hace	Uso en Vibe Coding	Costo	URL
Vercel	Vercel - EE.UU.	Plataforma de despliegue con un clic para apps Next.js y React. Preview automático en cada cambio.	Publicar cualquier app generada con v0 o Bolt.new en segundos	Gratis (hobby); Pro: \$20/mes	vercel.com
GitHub	Microsoft - EE.UU.	Control de versiones + repositorios + GitHub Copilot + Pages para publicar sitios estáticos	Guardar el código de tus apps generadas, colaborar con desarrolladores	Gratis; Team: \$4/mes	github.com

### Categoría 5: Análisis de datos con IA

Herramienta	Empresa	Que hace	Mejor para...	Costo	URL
ChatGPT (Code Interpreter)	OpenAI - EE.UU.	Analiza archivos CSV/Excel, genera gráficas, detecta patrones y responde preguntas sobre los datos en lenguaje natural	Análisis exploratorio rápido, cualquier profesional con datos	Plan Plus: \$20/mes (incluido)	chat.openai.com
Claude (análisis de archivos)	Anthropic - EE.UU.	Lee y analiza CSV, Excel, PDFs y documentos extensos. Ideal para análisis con mucho contexto y datos complejos.	Documentos largos, análisis que requieren razonamiento profundo	Gratis con límites; Pro: \$20/mes	claude.ai
n8n (pipelines de datos)	n8n GmbH - Alemania	Conecta fuentes de datos, transforma y enruta la información	Automatizar el flujo de datos desde múltiples fuentes	Self-hosted: gratis; Cloud:	n8n.io

Herramienta	Empresa	Que hace	Mejor para...	Costo	URL
		hacia visualizaciones o destinos específicos		desde \$20/mes	
Databricks	Databricks - EE.UU.	Plataforma de análisis de datos a escala empresarial con notebooks de IA, SQL y ML sobre grandes volúmenes de datos	Empresas con grandes volúmenes de datos, equipos de data science	Pago por uso (free trial disponible)	databricks.com

**CONSEJO PRACTICO:** Para este módulo comienza con las herramientas gratuitas: v0 o Bolt.new para construir interfaces, ChatGPT Plus para análisis de datos (si ya tienes el plan), y GitHub para guardar tu código. Supabase, Firebase y Vercel tienen planes gratuitos generosos que son suficientes para todos los ejercicios.

## TEMA 1 DE 6

# Qué es vibe coding: el nuevo paradigma de programar describiendo lo que quieres en lenguaje natural

Del código al lenguaje natural \* Como funciona por dentro \* El rol del no-programador \* Windsurf y el agente Cascade

Vibe Coding es el término que describe el nuevo paradigma en el que los humanos describen en lenguaje natural lo que quieren construir y la IA genera el código. El término fue popularizado por Andrej Karpathy (cofundador de OpenAI) en 2025 para describir un estilo de programación donde el desarrollador describe la intención y el agente de IA genera, prueba y corrige el código de forma autónoma.

No es que desaparezca la necesidad de entender la lógica del software. Es que la barrera de entrada se derrumba: ya no necesitas saber la sintaxis de Python, React o SQL para construir cosas que funcionan. Necesitas saber describir con precisión lo que quieres, entender si el resultado es correcto y saber cómo iterarlo.



*El Vibe Coding no elimina la necesidad de pensar. Elimina la necesidad de recordar la sintaxis. Tu aportas el criterio; la IA aporta el código.*

## 5.1.1 Cómo funciona el Vibe Coding por dentro

Para usar el Vibe Coding con criterio necesitas entender el proceso que ocurre entre tu instrucción en lenguaje natural y el código que aparece en pantalla. Es un proceso de 4 etapas que ocurre en segundos:

### ETAPA 1: Tu descripción se convierte en contexto estructurado

Tu instrucción en lenguaje natural (Crea un dashboard de ventas con una gráfica de barras por mes) se combina con el contexto del proyecto: los archivos existentes, el stack tecnológico, las dependencias instaladas y el historial de cambios. Este contexto completo se envía al modelo de lenguaje. Clave práctica: cuanto más contexto das en tu descripción (stack, estructura de datos, estilo visual), mejor es el resultado. El modelo no puede adivinar lo que no le dices.



### ETAPA 2: El LLM planifica la solución y genera el código

El modelo de lenguaje analiza el contexto y genera el código necesario. Los agentes avanzados como Windsurf Cascade o Cursor Composer no generan un solo archivo: planifican que archivos crear o modificar, en que orden y como conectarlos entre sí. Lo que ocurre en esta etapa:- El modelo identifica que componentes necesita crear- Genera el código para cada componente- Conecta los componentes entre si- Agrega las dependencias necesarias al package.json o requirements.txt



### ETAPA 3: El agente ejecuta los cambios y los prueba

Los agentes avanzados (Cascade en Windsurf, Composer en Cursor) no solo generan el código: lo ejecutan en el entorno del proyecto, detectan errores de compilación y los corrigen automáticamente en la misma sesión. Este ciclo de generar-ejecutar-correr puede repetirse varias veces sin intervención del usuario. Esto es lo que distingue a un IDE con IA agéntica de un simple chatbot de código: el agente actúa en el entorno real, no solo genera texto.



### ETAPA 4: Tu revisas, validas y das la siguiente instrucción

Tu rol en el Vibe Coding es el del revisor crítico y director creativo. Verificas que el resultado es lo que querías, identificas que no está bien y das la siguiente instrucción de ajuste. No necesitas entender cada línea de código, pero si necesitas saber si el comportamiento visual y funcional es el correcto. Regla de oro: nunca aceptes un resultado sin probarlo. El agente puede generar código que compila pero que no hace lo que pediste.

## 5.1.2 Windsurf — El editor con el agente Cascade

Windsurf de Codeium es un editor de código completo (fork de VS Code) con un agente de IA agéntica llamado Cascade. Lo que hace diferente a Cascade de un simple autocomplete es que puede actuar de forma autónoma: lee todos los archivos del proyecto, planifica cambios multi-archivo, ejecuta comandos de terminal y corrige errores sin que tengas que indicarle cada paso.

Característica	Autocomplete básico (GitHub Copilot)	Chat de IA (Cursor Chat)	Agente agéntico (Cascade / Cursor Composer)
Alcance de contexto	Solo el archivo actual	El archivo actual + contexto que le compartes	TODO el proyecto: todos los archivos, dependencias, historial
Tipo de acción	Sugiere completar la línea actual	Responde preguntas y sugiere cambios	Planifica, ejecuta cambios en múltiples archivos y corrige errores
Ejecución de comandos	No	No	Si: puede ejecutar npm install, correr pruebas, leer errores
Intervención humana requerida	En cada sugerencia	En cada instrucción	Solo al inicio (objetivo) y al final (revision)
Mejor para...	Acelerar escritura de código	Entender y modificar código	Construir features completas desde descripción

### Como instalar y usar Windsurf por primera vez:

1. Ve a windsurf.com y descarga el instalador para tu sistema operativo (Mac/Windows/Linux)
2. Instala Windsurf; se verá idéntico a VS Code porque está basado en el mismo motor
3. Crea una cuenta gratuita en windsurf.com con tu email o cuenta de Google
4. Abre el panel de Cascade con el atajo Ctrl+L (Windows) o Cmd+L (Mac)
5. Describe tu proyecto o tarea en lenguaje natural y observa cómo Cascade planifica y ejecuta los cambios

**CONSEJO PRÁCTICO:** Si ya usas VS Code, Windsurf será inmediatamente familiar. Puedes importar todas tus extensiones y configuraciones de VS Code con un solo clic desde el menú de configuración de Windsurf. No tienes que empezar de cero.

### 5.1.3 El nuevo rol del no-programador en el Vibe Coding

El Vibe Coding no convierte a cualquiera en programador senior. Lo que sí hace es cambiar dramáticamente el tipo de habilidades que necesitas para producir software funcional. Estas son las habilidades que importan ahora:

#### Habilidades que **IMPORTAN** en el Vibe Coding:

- \* Describir con precisión lo que quieres (el resultado, no el proceso)
- \* Estructurar el problema en partes más pequeñas y manejables
- \* Evaluar si el resultado visual y funcional es el correcto
- \* Identificar cuando el agente interpretó mal la instrucción
- \* Iterar con instrucciones de ajuste específicas y concretas
- \* Entender la arquitectura general del sistema que estás construyendo

#### Habilidades que ya **NO** son barrera de entrada:

- \* Recordar la sintaxis exacta de un lenguaje de programación
- \* Saber configurar entornos de desarrollo desde cero
- \* Conocer todas las librerías y sus APIs
- \* Saber escribir cada función y componente manualmente
- \* Depurar errores de compilación y de runtime manualmente

### La escala de habilidades en el Vibe Coding según nivel:

Nivel	Perfil	Lo que puede construir con Vibe Coding	Herramientas ideales
Principiante	Sin experiencia técnica, primer contacto con código	Landing pages, formularios, dashboards simples, herramientas de una página	Bolt.new, Lovable, v0 (todas sin instalar nada)

Nivel	Perfil	Lo que puede construir con Vibe Coding	Herramientas ideales
Intermedio	Ha visto algo de código, entiende conceptos básicos de web	Apps multi-página, CRUD completo con base de datos, integraciones de API simples	Lovable + Supabase, Bolt.new + Firebase, v0 + Vercel
Avanzado-u suario	Trabaja con código regularmente aunque no sea su especialidad	Sistemas con autenticación, dashboards con datos en vivo, integraciones con APIs de terceros	Windsurf Cascade, Cursor Composer, GitHub + Vercel para CI/CD
Pre-profesio nal	Quiere transición a desarrollo o tiene bases sólidas	Apps complejas, microservicios, pipelines de datos, n8n + backend propio	Todo el stack: Windsurf + Supabase + n8n + Vercel + Databricks

## TEMA 2 DE 6

# Crear dashboards básicos con prompts: visualiza tus datos de ventas, clientes o métricas sin saber Excel avanzado ni programación

Tipos de dashboard \* Prompts por tipo de gráfica \* v0 + datos reales \* Bolt.new para dashboards funcionales

Un dashboard bien diseñado convierte datos crudos en decisiones. Antes, crearlos requería conocimientos de Excel avanzado, Power BI, Tableau o desarrollo frontend. Hoy, con Vibe Coding, puedes describir el dashboard que necesitas y tenerlo funcionando con tus datos reales en menos de 30 minutos.

La clave no está en la herramienta: está en saber que tipo de visualización comunica mejor cada tipo de dato. Un dashboard mal diseñado con la herramienta más cara sigue siendo un dashboard inutilizable.

## 5.2.1 Qué tipo de gráfica usar según el tipo de dato

Antes de escribir el prompt para tu dashboard, define que tipo de gráfica es la correcta para cada dato. Esta decisión es tuya, no de la IA:

Tipo de gráfica	Cuando usarla	Ejemplo de dato que visualiza	Cuando NO usarla
Barras verticales (Bar Chart)	Comparar valores entre categorías distintas en un momento específico	Ventas por producto en el mes de octubre   Clientes por ciudad   Tickets resueltos por agente	Cuando tienes más de 8-10 categorías (se vuelve ilegible)
Líneas (Line Chart)	Mostrar la evolución de un valor a lo largo del tiempo	Ventas semana a semana   Usuarios activos por mes   Tasa de conversión mensual	Cuando los datos no tienen un orden temporal natural
Area (Area Chart)	Mostrar tendencia temporal + magnitud acumulada del valor	Ingresos acumulados del año   Volumen de transacciones por semana	Cuando hay múltiples series que se superponen y confunden
Pie / Dona (Pie/Donut Chart)	Mostrar la proporción de partes dentro de un todo	Distribución de ventas por categoría   Porcentaje de clientes por tipo de plan	Cuando hay más de 5-6 categorías o cuando las diferencias son pequeñas
Tabla de datos (Data Table)	Mostrar datos exactos que el usuario necesita consultar y exportar	Lista de clientes con sus métricas   Historial de transacciones   Comparativa detallada	Cuando el objetivo es comunicar una tendencia o patrón (usa gráfica en su lugar)

Tipo de gráfica	Cuando usarla	Ejemplo de dato que visualiza	Cuando NO usarla
Número grande (KPI Card)	Resaltar una métrica clave con su valor actual y comparación con periodo anterior	Ventas totales del mes: \$450,000 (+12% vs mes anterior)   Nuevos clientes: 47 esta semana	Cuando no hay un valor de referencia para contextualizar el número

### 5.2.2 Crear un dashboard de ventas con v0 de Vercel

v0 genera componentes de interfaz React de alta calidad. Para un dashboard, genera los componentes visuales (gráficas, tablas, KPIs) que luego puedes integrar en tu proyecto o desplegar directamente en Vercel. Es la opción más rápida para prototipar y mostrar a un cliente como se verá el dashboard final.

#### Proceso para crear un dashboard ejecutivo en v0:

1	<p><b>Ve a v0.dev y crea una cuenta</b></p> <p>Regístrate con tu cuenta de GitHub o Google en v0.dev. El plan gratuito incluye un número limitado de generaciones al día, suficiente para practicar.</p>
2	<p><b>Describe el dashboard con todos los detalles</b></p> <p>La calidad del resultado depende de la calidad de tu descripción. Usa el prompt estructurado de la sección 2.3 de este tema.</p>
3	<p><b>Revisa el resultado y pide ajustes específicos</b></p> <p>v0 genera el componente con datos ficticios de ejemplo. Revisa: las gráficas son del tipo correcto? ¿Los colores son los de tu marca? La disposición es lógica y legible?</p>
4	<p><b>Exporta o despliega directamente</b></p> <p>v0 tiene un botón Open in Vercel que despliega el componente como app en vivo con un clic. O copia el código y pegalo en tu proyecto existente de React/Next.js.</p>
5	<p><b>Conecta tus datos reales</b></p> <p>El código generado tiene datos de ejemplo hardcoded. En la sección 2.4 aprenderemos a reemplazarlos con datos reales de Google Sheets o una API.</p>

#### PROMPT: Prompt maestro para dashboard de ventas en v0:

Crea un dashboard ejecutivo de ventas mensual con estas especificaciones exactas:  
**LAYOUT:** Diseño de una sola página, responsive (funciona en desktop y móvil). Header con título 'Dashboard de Ventas' y el mes/año actual. Grid de 4 KPI cards en la parte superior.  
**KPI CARDS (4 tarjetas):**  
 1. Ventas totales del mes: \$847,500 MXN (+12% vs mes anterior)  
 2. Nuevos clientes: 47 (+8% vs mes anterior)  
 3. Ticket promedio: \$18,032 MXN (-3% vs mes anterior)  
 4. Tasa de conversión: 23% (+2pp vs mes anterior)  
**GRÁFICA 1 (debajo de los KPIs, ancho completo):** Grafica de barras verticales con los datos de ventas de los últimos 6 meses:  
 May: 620000, Jun:

710000, Jul: 680000, Ago: 795000, Sep: 810000, Oct: 847500 Eje Y en miles (MXN), eje X con nombres de mes abreviado GRÁFICA 2 (mitad izquierda):Gráfica de dona con distribución de ventas por categoría de producto:Servicios: 45%, Productos: 32%, Licencias: 23%TABLA (mitad derecha):Tabla de los 5 productos más vendidos este mes con columnas: Producto, Unidades, Ingresos, % del total DISEÑO: Colores morado oscuro (#5B2D8E) para encabezados y elementos principales, dorado (#D4A017) para valores de KPI positivos, rojo suave para negativos. Fondo gris claro. Tipografía limpia sans-serif. Estilo corporativo profesional, no colorido ni infantil.NOTA: Usa la librería Re Charts para las gráficas (está disponible en el ecosistema v0/shadcn).

### 5.2.3 Crear un dashboard funcional con Bolt.new

Mientras v0 genera componentes de interfaz, Bolt.new genera una aplicacion completa con logica de negocio y estado. Un dashboard en Bolt.new puede tener filtros funcionales, datos que cambian al interactuar y exportación a CSV, no solo la imagen del dashboard.

**PROMPT: Prompt para dashboard interactivo en Bolt.new:**

Crea una aplicación web de dashboard ejecutivo con React. El dashboard debe ser completamente interactivo y funcional. SECCIÓN DE FILTROS (barra superior):- Selector de rango de fechas: Este mes, Trimestre actual, Este ano, Personalizado- Selector de categoría: Todos, Servicios, Productos, Licencias- Botón de exportar datos a CSVKPI CARDS (4 tarjetas, primera fila):[Mismas 4 métricas que el ejemplo anterior con valores de muestra]Las tarjetas deben recalcular sus valores cuando el usuario cambia los filtros.GRAFICAS (segunda y tercera fila):- Grafica de líneas: tendencia de ventas de los últimos 12 meses- Grafica de barras agrupadas: comparativa este mes vs mes anterior por categoría- Tabla interactiva: lista de transacciones con búsqueda, ordenamiento por columna y paginación (10 filas por pagina)DATO IMPORTANTE: Usa datos de muestra generados programáticamente (no hardcodeados) para que el dashboard sea reutilizable. Los datos deben ser realistas para una empresa mexicana: montos en pesos MXN, nombres mexicanos.ESTILO: Tailwind CSS, colores morado (#5B2D8E) y dorado (#D4A017), fondo blanco, gráficas con Recharts. FUNCIONALIDAD ADICIONAL: Cuando el usuario cambia el filtro de fecha, todas las gráficas y KPIs se actualizan automáticamente con los datos del periodo seleccionado.

### Los 6 tipos de dashboard más útiles para profesionales de negocio:

Tipo de dashboard	Métricas clave que incluye	Para quien es mas util	Herramienta recomendada
Dashboard de ventas	Ventas totales, ticket promedio, nuevos clientes, tasa de conversión, ventas por producto y por vendedor	Directores comerciales, gerentes de ventas, dueños de negocio	Bolt.new (interactivo) o v0 (prototipo rapido)

Tipo de dashboard	Métricas clave que incluye	Para quien es mas util	Herramienta recomendada
Dashboard de marketing	Visitas web, leads generados, costo por lead, tasa de conversión funnel, rendimiento por canal	Directores y coordinadores de marketing digital	v0 o Bolt.new + integracion con Google Analytics via API
Dashboard de operaciones	Tareas completadas, tiempo de ciclo, cuellos de botella, SLA cumplidos, capacidad utilizada	Directores de operaciones, jefes de producción	Bolt.new con datos de Google Sheets o Notion
Dashboard de atencion al cliente	Tickets abiertos/cerrados, tiempo de respuesta, NPS, satisfacción por agente, tipo de consulta más frecuente	Gerentes de servicio, líderes de soporte	Bolt.new + Supabase para datos en tiempo real
Dashboard financiero	Flujo de caja, ingresos vs gastos, rentabilidad por producto, proyección del mes, cuentas por cobrar	CFO, directores financieros, contadores gerenciales	Bolt.new con datos de Google Sheets actualizado diariamente
Dashboard de RH	Headcount, rotación, ausencias, tiempo de contratación, distribución por departamento	Directores de RH, gerentes de personas	v0 para el diseno + Google Sheets como fuente de datos

## TEMA 3 DE 6

# Generar aplicaciones simples con Lovable, v0 y Bolt: desde landing pages hasta herramientas internas básicas para tu negocio

El proceso de Vibe Coding completo \* Lovable con Supabase \* Firebase como backend \* Vercel y GitHub para publicar

En este tema profundizamos en el flujo completo de Vibe Coding: desde la descripción inicial hasta la aplicación publicada en internet con datos reales. La diferencia entre generar un componente bonito y tener una app funcional está en conectar el frontend (la interfaz) con el backend (los datos y la lógica).

Las tres herramientas de este tema tienen niveles distintos de complejidad y capacidad. Elige la que corresponda a tu nivel actual y al tipo de app que necesitas construir.

## 5.3.1 El flujo completo de Vibe Coding: de idea a app publicada

### FASE 1: Definir el problema con precisión antes de abrir cualquier herramienta

Responde estas preguntas antes de escribir el primer prompt: 1. ¿Qué problema resuelve esta app exactamente? 2. ¿Quién la va a usar y con qué frecuencia? 3. ¿Cuántos usuarios simultáneos tendrá? 4. ¿Qué datos necesita guardar o mostrar? 5. Necesita autenticación de usuarios? Estas respuestas determinan si necesitas Lovable (con base de datos real), v0 (solo interfaz) o Bolt.new (código exportable). Elegir mal la herramienta al inicio es la causa número uno de tiempo perdido en Vibe Coding.



### FASE 2: Construir el prototipo visual con la herramienta elegida

Con la herramienta seleccionada, genera el prototipo usando el prompt estructurado. Clave para buenos resultados: - Describe el problema que resuelve, no como implementarlo - Especifica el tipo de usuario: quien la usa y que necesita ver - Define la estructura de datos si la app guarda información - Indica el estilo visual: colores de marca, si ya tienes uno - Espera el resultado, prueba la funcionalidad básica y da instrucciones de ajuste específicas.



### FASE 3: Conectar la base de datos y el backend

Si la app necesita guardar datos, conecta la base de datos: - Lovable: conecta Supabase automáticamente; solo acepta la configuración cuando te la propone - Bolt.new: puedes integrar Supabase o Firebase con una instrucción en el chat - v0: exporta el código y agrega la lógica de datos en tu editor (Windsurf o Cursor) Definir bien el esquema de datos (qué tablas necesitas, qué columnas tiene cada una) ANTES de conectar la base de datos evita rehacer trabajo más adelante.



### FASE 4: Publicar en Vercel o GitHub Pages

Una vez que la app funciona localmente o en el preview de la herramienta:- Desde Bolt.new: conecta tu repositorio de GitHub desde la interfaz > Vercel lo detecta automáticamente y despliega- Desde Lovable: el botón Share te da una URL pública inmediata. Para dominio propio, conecta Vercel.- Desde v0: el botón Open in Vercel despliega el componente en producción en segundos.Cada vez que haces un cambio y lo subes a GitHub, Vercel despliega automáticamente la nueva versión (esto se llama CI/CD continuo).

### 5.3.2 Lovable con Supabase: apps completas con base de datos real

Lovable es la herramienta de Vibe Coding más completa para alguien sin experiencia técnica. Genera el frontend, configura la base de datos en Supabase, escribe las funciones del backend y conecta todo automáticamente. Tú solo describes lo que necesitas.

#### Qué es Supabase y por qué lo usa Lovable:

Supabase es una plataforma de base de datos PostgreSQL gestionada. Es el backend de las apps que genera Lovable: guarda los datos de los usuarios, los registros que crea tu app, los archivos que suben tus usuarios. Su ventaja para el Vibe Coding: tiene una API REST automática que Lovable configura sin que tengas que escribir código de servidor.

Componente de Supabase	Que hace en tu app	Como lo usa Lovable
Database (PostgreSQL)	Guarda todos los datos estructurados de tu app en tablas relacionales	Lovable crea las tablas automáticamente según la descripción de tu app
Authentication	Maneja el registro, login y sesión de los usuarios de tu app	Lovable conecta los formularios de login con el sistema de auth de Supabase en automático
Storage	Almacena archivos: imágenes, documentos, exportaciones CSV	Lovable lo conecta cuando la app necesita subir o descargar archivos
Row Level Security (RLS)	Controla qué datos puede ver o modificar cada usuario según su rol	Lovable configura las reglas básicas de seguridad; ajustes avanzados requieren conocimiento de SQL

**PROMPT: Prompt para app de gestión de proyectos con Lovable + Supabase:**

Crea una aplicación web de gestión de proyectos para un equipo de trabajo pequeño (5-15 personas).**FUNCIONES PRINCIPALES:**1. **PROYECTOS:** Crear proyectos con nombre, descripción, fecha de inicio, fecha de entrega y responsable. Ver lista de todos los proyectos con su estatus (En planeación / En progreso / Completado / Pausado).2. **TAREAS:** Dentro de cada proyecto, crear tareas con nombre, descripción, asignado a, fecha límite y prioridad (Alta/Media/Baja). Las tareas tienen estatus: Pendiente, En progreso, Completada.3. **DASHBOARD:** Pantalla principal con: total de proyectos activos, tareas pendientes hoy, tareas vencidas, y un gráfico de progreso de los 5 proyectos más recientes.4. **AUTENTICACIÓN:** Login con email y contraseña. Solo usuarios registrados pueden acceder.**ESTRUCTURA DE BASE DE DATOS:-** Tabla projects: id, name, description, start\_date, end\_date, owner\_id, status, created\_at- Tabla tasks: id, project\_id, name, description, assigned\_to, due\_date, priority, status, created\_at- Tabla profiles: id (mismo que auth.users), full

name, email, role DISEÑO: Clean y profesional. Sidebar de navegación en el lado izquierdo con: Dashboard, Proyectos, Mis Tareas, Equipo. Colores: morado (#5B2D8E) para la sidebar, fondo blanco para el contenido, dorado (#D4A017) para acciones primarias. Tipografía Inter o similar.FUNCIONALIDAD EXTRA: Cuando una tarea pasa su fecha límite sin completarse, resaltar en rojo en la lista.

### 5.3.3 Firebase como backend: sincronización en tiempo real

Firebase de Google es ideal cuando necesitas datos que se actualizan en tiempo real para todos los usuarios simultáneamente: un dashboard que muestra las métricas del momento, una app de chat, un tablero kanban colaborativo. Su base de datos Firestore sincroniza los cambios al instante en todos los dispositivos conectados.

#### Quando elegir SUPABASE (con Lovable):

- \* Tus datos tienen relaciones entre tablas (clientes + pedidos + productos)
- \* Necesitas consultas SQL complejas o reportes con filtros avanzados
- \* La app necesita autenticación con permisos por rol
- \* Prefieres que Lovable lo configure todo automáticamente

#### Quando elegir FIREBASE:

- \* Necesitas sincronización en tiempo real para múltiples usuarios
- \* La app es del ecosistema Google (Google Workspace, Gmail, etc.)
- \* Los datos son más simples y no requieren relaciones complejas entre tablas
- \* Quieres usar Firebase Hosting junto con la base de datos

### 5.3.4 Vercel y GitHub: publicar tu app y mantenerla actualizada

Una app que solo vive en tu computadora no le sirve a nadie. Vercel y GitHub son las herramientas que te permiten publicar tu app en internet en segundos y mantenerla actualizada de forma automática cada vez que haces un cambio.

#### Flujo de publicación con GitHub + Vercel:

1

#### Crea un repositorio en GitHub

Ve a github.com > New Repository. Ponle un nombre descriptivo (ej: dashboard-ventas-empresa). Dejaló en público para el plan gratuito o privado si tienes plan de pago.

2

#### Sube el código de tu app al repositorio

En Bolt.new: el botón Connect to GitHub sube el proyecto completo automáticamente. Desde tu computadora: usa el terminal con los comandos git init, git add., git commit -m 'primer commit', git push origin main.

3

#### Conecta el repositorio en Vercel

4	Ve a <a href="https://vercel.com">vercel.com</a> > Add New Project > selecciona tu repositorio de GitHub > Vercel detecta automáticamente el framework (React, Next.js, etc.) y configura el despliegue.
	<p><b>Vercel despliega automáticamente</b></p> <p>En 1-3 minutos tu app tiene una URL pública de <a href="https://vercel.app">vercel.app</a>. Cada vez que subes un cambio a GitHub, Vercel despliega la nueva versión automáticamente. Si el despliegue falla, Vercel mantiene la versión anterior activa.</p>
5	<p><b>Configura tu dominio personalizado (opcional)</b></p> <p>En el dashboard de Vercel: Settings &gt; Domains &gt; Add Domain. Apunta tu dominio a los servidores de Vercel siguiendo las instrucciones. Vercel configura el HTTPS automáticamente.</p>

### Galería de apps que puedes construir con Vibe Coding en menos de 60 minutos:

App	Que hace	Herramienta ideal	Tiempo estimado
Calculadora de cotizaciones	Formulario con campos de servicio y cantidad; calcula el total con descuentos y genera un resumen descargable en PDF	Bolt.new	20-30 minutos
Registro de asistencia a talleres	Formulario de registro, lista de asistentes por evento, exportación a CSV	Lovable + Supabase	25-35 minutos
Tracker de hábitos	Listado de hábitos con check diario, racha de días consecutivos, gráfica de progreso semanal	Bolt.new	20-30 minutos
Directorio de contactos del equipo	Lista de empleados con foto, cargo, área y contacto; búsqueda y filtros	Lovable + Supabase (con upload de fotos)	30-45 minutos
Seguimiento de gastos	Formulario de gasto con categoría, monto y fecha; dashboard con gastos del mes por categoría	Bolt.new o Lovable	25-40 minutos
Landing page de curso o servicio	Sección hero, beneficios, testimonios, precios y formulario de registro con confirmación por email	Bolt.new o v0	20-30 minutos
Tablero Kanban simple	Columnas: Por hacer, En progreso, Hecho; arrastrar tarjetas entre columnas	Bolt.new	25-35 minutos
Encuesta con resultados en vivo	Formulario de preguntas múltiple opción; dashboard que muestra los resultados conforme llegan respuestas	Lovable + Supabase (tiempo real)	30-45 minutos

## TEMA 4 DE 6

## Conectar fuentes de datos y visualizarlas: importa información de Google Sheets, bases de datos o APIs y convierte en gráficas útiles

Google Sheets como fuente de datos \* APIs REST \* n8n para pipelines \* Databricks para escala empresarial

Un dashboard con datos ficticios de ejemplo es solo un prototipo. El valor real llega cuando el dashboard muestra tus datos reales, actualizados automáticamente, sin que tengas que exportar ni copiar nada manualmente. En este tema aprenderás las 4 formas principales de conectar fuentes de datos a tus visualizaciones.

### 5.4.1 Google Sheets como fuente de datos para dashboards

Google Sheets es la fuente de datos más accesible para la mayoría de los profesionales. Casi todos ya tienen sus datos ahí: ventas, clientes, inventario, métricas. Con la API de Google Sheets, cualquier app generada con Vibe Coding puede leer esos datos en tiempo real sin que tengas que exportar nada.

#### Tres formas de conectar Google Sheets a tu dashboard:

Método	Complejidad	Como funciona	Cuando usarlo
Publicar como CSV público	Muy baja - sin código	En Google Sheets: Archivo > Publicar en la web > CSV. Obtienes una URL que devuelve los datos en CSV cada vez que la visitas.	Para prototipos rápidos con datos que no son sensibles. No requiere autenticación.
Google Sheets API con clave	Media - requiere configuración inicial	Crear proyecto en Google Cloud > activar Sheets API > obtener API Key > usar esa key en el fetch de tu app.	Para apps en producción que necesitan leer datos. Más seguro que el CSV público.
n8n como intermediario	Media - configuración de un flujo	n8n lee Google Sheets periódicamente, transforma los datos y los guarda en tu base de datos (Supabase o Firebase). Tu app lee desde la base de datos, no desde Sheets directamente.	Cuando necesitas transformar o limpiar los datos antes de mostrarlos, o cuando tienes múltiples fuentes de datos que consolidar.

**PROMPT: Prompt para conectar Google Sheets a un dashboard en Bolt.new:**

```
Tengo un dashboard de ventas en React. Quiero que la grafica de ventas mensuales use datos reales de un Google Sheet publico en lugar de los datos hardcoded.URL del Google Sheet como CSV:https://docs.google.com/spreadsheets/d/[ID_DEL_SHEET]/export?format=csv&gid=0E estructura del CSV (primera fila es encabezado):Mes,Ventas,Clientes,TicketPromedioEnero,620000,41,15122Febrero,710000,47,15106[etc]NECESITO:1. Un hook de React (useEffect) que fetch el CSV al cargar la pagina2. Parsear el CSV a un array de objetos JavaScript3. Conectar ese array a los datos de la grafica de Recharts existente4. Mostrar un indicador de carga mientras se obtienen los datos5. Mostrar un mensaje de error si el fetch falla6. Que los datos se refresquen automaticamente cada 5 minutos sin recargar la paginaMantener el resto del codigo del dashboard exactamente igual.
```

### 5.4.2 Conectar APIs REST a tu dashboard

La mayoría de las herramientas que ya usas en tu negocio tienen APIs REST: Google Analytics, HubSpot, Shopify, Stripe, Notion, Airtable y decenas más. Una API REST es simplemente una URL a la que puedes hacer una petición y recibir los datos en formato JSON.

Herramienta	URL base de su API	Que datos puedes obtener	Costo de uso
Google Analytics	reporting.googleapis.com/v4	Sesiones, usuarios, fuentes de tráfico, conversiones, comportamiento por página	Gratis (incluido en GA4)
HubSpot CRM	api.hubapi.com	Contactos, deals, actividades, estado del pipeline de ventas	Gratis con API key
Stripe (pagos)	api.stripe.com	Transacciones, ingresos del periodo, clientes, suscripciones activas	Gratis (incluido en cuenta Stripe)
Shopify	[tienda].myshopify.com/admin/api	Pedidos, productos, inventario, clientes, métricas de la tienda	Requiere plan Shopify activo
Notion	api.notion.com	Contenido de bases de datos de Notion: cualquier tabla, galeria o lista	Gratis con API key
Airtable	api.airtable.com	Registros de cualquier base de Airtable, filtrados y ordenados	Gratis con API key (plan básico)

**PROMPT: Prompt para conectar una API de Stripe al dashboard:**

```

Agrega una sección de métricas de pagos a mi dashboard usando la API de Stripe.DATA QUE NECESITO DE STRIPE:1. Ingresos del mes actual (sum de charges exitosos del mes en curso)2. Número de transacciones del mes 3. Transacciones de los últimos 30 días día por día (para grafica de lineas)CONFIGURACIÓN:- La API key de Stripe se guardará en una variable de entorno llamada NEXT_PUBLIC_STRIPE_KEY- No hardcodees la API key en el código- Usar la versión 2023-10-16 de la API de StripeVISUALIZACION:- KPI card: Ingresos del mes en pesos MXN (convertir de centavos a pesos)- KPI card: Total de transacciones del mes- Gráfica de líneas: ingresos día por día del último mes
NOTA: Manejar el caso donde la API no responde
    
```

```
o devuelve error: mostrar el último dato conocido con un indicador de que los
datos pueden estar desactualizados.
```

### 5.4.3 n8n como pipeline de datos: conecta, transforma y entrega

n8n es el intermediario ideal cuando tus datos están en múltiples fuentes (Google Sheets + CRM + sistema de facturación), necesitan transformarse o limpiarse antes de mostrarse y deben actualizarse en un horario específico. n8n actúa como el ETL (Extract, Transform, Load) de tu dashboard.

<p><b>EXTRAE</b> Google Sheets HubSpot StripeSQL</p>	<p><b>TRANSFORMA</b> Limpia datos Calcula métricas Agrega periodos Merge fuentes</p>	<p><b>GUARDA</b> Supabase Firebase Postgres JSON cache</p>	<p><b>VISUALIZA</b> Dashboard React(Bolt/Lovable/v0)</p>
--	--	--	--

#### Ejemplo de pipeline de datos con n8n para un dashboard de ventas:

<b>1</b>	<p><b>TRIGGER: Schedule cada día a las 6am</b> n8n se activa automáticamente todos los días a las 6am para actualizar el dashboard con los datos del día anterior.</p>
<b>2</b>	<p><b>NODE 1: Leer ventas del mes desde Google Sheets</b> n8n lee la hoja de Google Sheets con el registro de ventas. Filtra solo las filas del mes actual.</p>
<b>3</b>	<p><b>NODE 2: Leer clientes desde HubSpot CRM</b> n8n consulta la API de HubSpot y obtiene los deals cerrados en el mes, con su monto y fecha.</p>
<b>4</b>	<p><b>NODE 3: Calcular métricas consolidadas</b> n8n suma el total de ventas de Sheets + deals de HubSpot, calcula el ticket promedio, cuenta nuevos clientes y genera un objeto JSON con todas las métricas del día.</p>
<b>5</b>	<p><b>NODE 4: Guardar en Supabase</b> n8n hace un upsert (insertar o actualizar) en la tabla daily metrics de Supabase con el JSON de métricas del día. Si ya existe el registro del día, lo actualiza.</p>
<b>6</b>	<p><b>El dashboard lee desde Supabase, no desde las fuentes originales</b> El dashboard de React ya conectado a Supabase siempre tiene datos frescos. No necesita consultar las APIs de Google Sheets y HubSpot directamente (mas rapido y mas seguro).</p>

### 5.4.4 Databricks para análisis de datos a escala empresarial

Databricks es la plataforma para cuando los datos ya no caben en Excel ni en Google Sheets: millones de filas, datos de múltiples sistemas legacy, modelos de machine learning y análisis predictivo. Es la herramienta de data science que ahora tiene IA conversacional integrada.

#### Databricks ES la herramienta correcta cuando:

- \* Tienes millones de registros que Excel o Sheets no pueden procesar
- \* Necesitas conectar data warehouses, datalakes o sistemas ERP empresariales
- \* Tu empresa necesita modelos de ML o predicciones sobre los datos
- \* Tienes un equipo de datos (data engineers, data scientists) que necesitan colaborar
- \* Cumplimiento de seguridad y gobernanza de datos es crítico para tu empresa

#### Databricks NO es para ti todavía si:

- \* Tus datos caben en Google Sheets o en un CSV de menos de 100,000 filas
- \* Eres el único usuario de los datos de tu empresa
- \* No tienes presupuesto asignado para infraestructura de datos
- \* Eres nuevo en el ecosistema de datos y analytics
- \* Tu caso de uso se resuelve con ChatGPT + un CSV (ver Tema 5 de este módulo)

#### Databricks SQL con IA: consultar datos con lenguaje natural

Databricks tiene una función llamada AI/BI que permite hacer preguntas sobre los datos en lenguaje natural y obtener gráficas y dashboards sin escribir SQL. Esta es la parte de Databricks más relevante para profesionales no-técnicos que trabajan en empresas que ya tienen Databricks implementado.

- \* **GENIE:** el asistente de Databricks que responde preguntas sobre los datos en lenguaje natural. Escribe: ¿cuáles fueron los 5 productos con mayor margen el trimestre pasado? y Genie genera el SQL, lo ejecuta y muestra la gráfica.
- \* **DASHBOARDS DE IA:** Databricks puede generar dashboards completos desde una descripción de lo que quieres analizar, sin que el usuario tenga que escribir ninguna consulta SQL.
- \* **ACCESO DESDE EXCEL Y SHEETS:** Si tu empresa tiene Databricks, puedes conectar Google Sheets directamente a las tablas de Databricks para tener datos actualizados en tus hojas de cálculo sin exportar manualmente.

## TEMA 5 DE 6

## Análisis de datos conversacional: sube archivos a ChatGPT o Claude y hazles preguntas sobre tendencias, patrones y oportunidades

Code Interpreter de ChatGPT \* Claude para documentos extensos \* Prompts de análisis \* Del archivo a la decisión

El análisis de datos conversacional es la habilidad de hablarle a tus datos como si fueran una persona experta que los conoce perfectamente. Subes un archivo CSV o Excel, le haces preguntas en lenguaje natural y obtienes gráficas, tablas, resúmenes e insights sin necesidad de saber formulas de Excel, SQL ni Python.

Esta es posiblemente la habilidad de mayor impacto inmediato del módulo: cualquier profesional que tenga datos tabulares (ventas, clientes, inventario, encuestas, métricas de operación) puede empezar a extraer valor de ellos hoy mismo con solo subir el archivo y hacer la pregunta correcta.



*Los datos que tienes en Excel pueden responderte preguntas que llevan meses sin respuesta. El único prerequisite es saber cómo formularlas.*

### 5.5.1 ChatGPT con Code Interpreter (Advanced Data Analysis)

Code Interpreter es la función de ChatGPT que convierte al modelo en un analista de datos. Cuando subes un archivo CSV o Excel, ChatGPT puede leerlo, procesarlo con Python en un entorno sandbox, generar gráficas, hacer cálculos estadísticos y responder preguntas sobre los datos. Todo esto ocurre sin que veas una sola línea de código: solo la pregunta y el resultado.

#### Como activar y usar Code Interpreter:

6. Abre ChatGPT con plan Plus o superior (chat.openai.com)
7. Selecciona el modelo GPT-4o en el selector de la parte superior
8. Clic en el icono de clip para adjuntar archivo
9. Selecciona tu archivo CSV, XLSX o cualquier formato de datos
10. Espera que ChatGPT procese el archivo (5-15 segundos)
11. Escribe tu primera pregunta. ChatGPT confirmara que entendió la estructura del archivo antes de responder

## La secuencia de análisis recomendada: de lo general a lo específico

El error más común es empezar con preguntas muy específicas antes de entender la estructura del archivo. Sigue esta secuencia para obtener el máximo valor de cualquier dataset:

### PASO 1: Entender la estructura del archivo (2 minutos)

Primera pregunta siempre: 'Describe la estructura de este archivo. Cuantas filas tiene, cuales son las columnas, que tipo de dato tiene cada columna y hay valores faltantes o anomalías evidentes que deba conocer antes de analizar?' Esto te da el mapa del territorio antes de explorar. ChatGPT identificará: columnas de fecha que puede usar para análisis temporal, columnas numéricas para cálculos, columnas categóricas para agrupaciones, y cualquier problema de calidad de datos que podría afectar el análisis.



### PASO 2: Análisis panorámico — el resumen ejecutivo (5 minutos)

Segunda pregunta: 'Genera un resumen ejecutivo de los datos con los 5 hallazgos más importantes que un director general debería conocer de esta información. Para cada hallazgo, incluye el dato específico que lo soporta.' Este paso identifica patrones, anomalías y tendencias principales sin que tengas que saber dónde buscar. A veces el hallazgo más importante es uno que no sabías que existía.



### PASO 3: Gráficas de los patrones principales (5-10 minutos)

Tercera pregunta: 'Genera las 3 gráficas más informativas para comunicar los patrones principales de estos datos a un equipo directivo. Para cada gráfica, explica que muestra y por qué es relevante para la toma de decisiones.' ChatGPT genera gráficas reales en Python (matplotlib o plotly) que puedes descargar directamente. Son imágenes listas para usar en presentaciones.



### PASO 4: Preguntas específicas orientadas a decisión (10-20 minutos)

A partir de aquí, haz las preguntas específicas que más te importan para tu contexto: - Cuáles son los 10 clientes con mayor valor de por vida y que tienen en común? - En qué mes del año las ventas caen más y hay alguna correlación con otro factor del dataset? - Hay algún producto o categoría que tenga tendencia negativa consistente en los últimos 6 meses? Estas preguntas orientadas a decisión son las que producen insights accionables.



### PASO 5: Predicciones y proyecciones simples (5 minutos)

Última etapa (opcional): 'Basandote en la tendencia de los últimos 6 meses, proyecta los próximos 3 meses para [la métrica clave]. Indica el nivel de confianza de la proyección y los supuestos que estás asumiendo.' ChatGPT puede hacer proyecciones simples con regresión lineal u otros modelos estadísticos básicos. Estas proyecciones son indicativas, no definitivas, pero dan un punto de partida para la planificación.

## 5.5.2 Prompts de análisis listos para usar según tipo de archivo

### Para archivos de ventas (CSV con fecha, producto, monto, vendedor):

#### PROMPT: Secuencia completa de análisis de ventas:

PROMPT 1 (estructura): Describe la estructura de este archivo de ventas. Identifica: el rango de fechas cubierto, el número de transacciones, los productos o servicios que aparecen, si hay información de vendedor y el rango de montos. Advierte si hay datos faltantes o inconsistencias en fechas o montos.'

PROMPT 2 (panorama ejecutivo): 'Genera un análisis ejecutivo de ventas con: total del periodo, mes de mayor venta, mes de menor venta, producto más vendido por volumen, producto más vendido por monto, y si hay información de vendedor, los 3 mejores vendedores. Presenta los resultados en una tabla limpia.'

PROMPT 3 (tendencia temporal): 'Crea una gráfica de barras con las ventas totales por mes. Incluye una línea de tendencia. Calcula si la tendencia general es positiva, negativa o plana e indica el porcentaje de cambio de la primera mitad a la segunda mitad del periodo.'

PROMPT 4 (composición): 'Muestra la distribución de ventas por producto/categoría/vendedor]. Usa una gráfica de dona o de barras horizontales. Identifica si existe concentración de riesgo: algún producto o vendedor que represente más del 40% de las ventas totales.'

PROMPT 5 (accionable): 'Basandote en los datos, identifica: (1) el patrón temporal más relevante para planificar el siguiente trimestre, (2) el segmento o producto con mayor potencial de crecimiento según la tendencia, (3) un riesgo específico que debería atenderse con urgencia.'

### Para archivos de clientes (CRM exportado: nombre, industria, fecha adquisición, valor):

#### PROMPT: Análisis de base de clientes:

PROMPT 1 (segmentación): 'Segmenta la base de clientes en grupos significativos. Puedes usar criterios como: antigüedad (nuevos, activos, veteranos), valor (alto, medio, bajo ticket), industria o cualquier otro criterio que encuentres relevante en los datos. Muestra cuántos clientes hay en cada segmento y el porcentaje del total.'

PROMPT 2 (valor y retención): 'Calcula para cada cliente o segmento: tiempo como cliente, valor total generado y valor promedio por periodo. Identifica los 20% de clientes que generan el 80% del valor (principio de Pareto). Muestra los resultados en una tabla ordenada de mayor a menor valor.'

PROMPT 3 (patrones de comportamiento): 'Hay patrones en la fecha de adquisición de clientes? Hay temporadas del año donde se adquieren más clientes? La distribución por industria o segmento ha cambiado en el tiempo? Gráfica los hallazgos más importantes.'

PROMPT 4 (riesgo de churn): 'Identifica clientes que podrían estar en riesgo de abandono basándote en: tiempo sin actividad reciente si hay fechas de última compra, clientes con tendencia decreciente en su valor, o cualquier otro patrón que pueda indicar desenganche. Lista los 10 casos más urgentes.'

### Para encuestas y datos de satisfacción (NPS, CSAT, formularios):

#### PROMPT: Análisis de datos de satisfacción y encuestas:

PROMPT 1 (distribución de respuestas): 'Muestra la distribución de frecuencia para cada pregunta de la encuesta. Para preguntas numéricas (como NPS del 1 al 10), calcula el promedio, la mediana y el porcentaje de detractores, neutrales y

promotores. Para preguntas categóricas, muestra el porcentaje de cada opción de respuesta.'

PROMPT 2 (temas en comentarios abiertos): 'En las preguntas de texto abierto o comentarios, identifica los 10 temas o palabras más frecuentes. Agrupa los comentarios en categorías: positivos, negativos y sugerencias. Para cada categoría, muestra 3 ejemplos representativos (los comentarios más típicos o frecuentes).'

PROMPT 3 (correlaciones): 'Hay correlación entre la puntuación general de satisfacción y alguna variable específica del encuestado (área, cargo, antigüedad, plan contratado)? ¿Qué grupo de encuestados da las puntuaciones más bajas y que grupo de las más altas? ¿Qué debería priorizar el equipo basándose en esto?'

PROMPT 4 (tendencia si hay datos de múltiples períodos): 'Si el archivo tiene datos de varios meses o trimestres, muestra cómo ha cambiado el NPS o la satisfacción general en el tiempo. Ha mejorado, empeorado o se mantiene estable? ¿Hay algún periodo específico donde hubo un cambio notable?'

### 5.5.3 Claude para análisis de documentos extensos y datos complejos

Claude tiene ventajas específicas sobre ChatGPT para ciertos tipos de análisis: su ventana de contexto extendida (hasta 200,000 tokens) le permite procesar documentos mucho más grandes sin perder el hilo, y su razonamiento es especialmente preciso para análisis que requieren lógica multi-paso o que involucran texto narrativo junto con datos numéricos.

Tipo de análisis	Mejor con ChatGPT Code Interpreter	Mejor con Claude	Por qué
Datos numéricos puros (CSV/Excel)	X		ChatGPT genera gráficas reales con Python. Claude no ejecuta código de gráficas nativas.
Documentos de texto extensos (contratos, informes, reportes de 50+ páginas)		X	Claude procesa hasta 200K tokens sin perder contexto. ChatGPT tiene límites más estrictos.
Análisis mixto: datos + texto narrativo		X	Claude combina mejor el análisis numérico con el razonamiento sobre el contexto narrativo.
Generación de gráficas e imágenes	X		ChatGPT con Code Interpreter genera imágenes descargables. Claude solo describe los datos.
Datasets muy grandes (100,000+ filas)	X		Code Interpreter de ChatGPT puede procesar grandes volúmenes de datos con Python.
Comparar múltiples documentos (3-5 archivos relacionados)		X	Claude carga todos los archivos y los cruza manteniendo el contexto de cada uno.

**PROMPT: Prompt para análisis profundo de datos con Claude:**

He subido [numero] archivos con los siguientes contenidos: Archivo 1: [descripción breve del contenido y estructura] Archivo 2: [descripción breve][etc.] ANÁLISIS QUE NECESITO: 1. CRUCE DE INFORMACIÓN: Identifica conexiones o patrones que aparezcan en más de un archivo. ¿Qué información de un archivo complementa, contradice o enriquece la de otro? 2. VACÍOS E INCONSISTENCIAS: Hay información que falta o que parece contradictoria entre los archivos? Señala el nivel de riesgo o impacto que representa. 3. RESUMEN EJECUTIVO CONSOLIDADO: Un resumen de no más de 5 puntos que capture lo más importante de TODOS los archivos en conjunto, no uno por uno. Orientado a la toma de decisiones, no a la descripción del contenido. 4. RECOMENDACIONES ACCIONABLES: Basándose exclusivamente en lo que está en los archivos, cuales son las 3 acciones más urgentes o importantes que recomendarías? Justifica cada una con datos específicos de los archivos. NOTA: Si algo no está en los archivos, indícalo claramente. No especules ni inventes datos que no existan en el contenido cargado.

TEMA 6 DE 6

# Límites del vibe coding: entiende que puedes lograr con estas herramientas y cuando necesitas desarrollo profesional o el Nivel 2

La matriz de complejidad \* Señales de que necesitas escalar \* Lo que la IA no puede hacer por ti  
 \* El puente al Nivel 2

El Vibe Coding es una herramienta extraordinariamente poderosa para el rango correcto de problemas. Usarlo para lo que no está diseñado produce frustración, deuda técnica y, en el peor caso, sistemas críticos construidos sobre bases frágiles. Este tema te da el criterio para saber exactamente donde termina el Vibe Coding y donde empieza el desarrollo profesional.

La pregunta no es si el Vibe Coding tiene límites. Claro que los tiene. La pregunta es saber dónde están esos límites para que puedas tomar decisiones informadas sobre cuándo usarlo, cuando combinarlo con ayuda profesional y cuando simplemente contratar desarrollo.



*El Vibe Coding bien aplicado no compite con el desarrollo profesional. Lo complementa: hace posibles cosas que antes no existían y libera a los desarrolladores para el trabajo que realmente requiere expertise humano.*

## 6.1 La Matriz de Complejidad del Vibe Coding

Esta matriz clasifica los proyectos según dos ejes: la complejidad técnica del sistema y el riesgo de negocio si el sistema falla. La posición de tu proyecto en esta matriz determina el nivel de supervisión y expertise que requiere:

<p><b>BAJA COMPLEJIDAD + BAJO RIESGO</b>  <b>ZONA VERDE: Vibe Coding puro</b>                  Construye y despliega sin supervisión técnica adicional. Estas herramientas son suficientes.                  Ejemplos: landing pages, formularios, dashboards con datos propios, herramientas internas de uso ocasional, prototipos para validar ideas, calculadoras y utilidades de equipo.</p>	<p><b>ALTA COMPLEJIDAD + BAJO RIESGO</b>  <b>ZONA AMARILLA: Vibe Coding supervisado</b>                  Usa Vibe Coding para avanzar rápido, pero con revisión de un desarrollador antes de producción.                  Ejemplos: apps con autenticación, integraciones con APIs de pago, sistemas con múltiples roles de usuario, apps con lógica de negocio compleja.</p>
<p><b>BAJA COMPLEJIDAD + ALTO RIESGO</b></p>	<p><b>ALTA COMPLEJIDAD + ALTO RIESGO</b></p>

<p><b>ZONA AZUL: Auditar antes de activar</b></p> <p>El sistema parece simple pero maneja datos o procesos críticos que requieren auditoría de seguridad.</p> <p>Ejemplos: formulario que captura datos personales o bancarios, sistema de facturación, proceso que mueve dinero o afecta contratos legales.</p>	<p><b>ZONA ROJA: Desarrollo profesional</b></p> <p>El Vibe Coding puede usarse para prototipo inicial, pero la producción requiere desarrollo profesional completo.</p> <p>Ejemplos: sistemas bancarios, plataformas de salud con datos clínicos, ERP de manufactura, cualquier sistema con regulación sectorial estricta.</p>
--	--

## 5.6.2 Las 8 señales de que el Vibe Coding ya no es suficiente

Estas son las situaciones específicas que indican que es momento de involucrar a un desarrollador profesional o de usar las herramientas avanzadas del Nivel 2 del Diplomado:

Señal	Descripción de la situación	Por que el Vibe Coding no alcanza	Qué hacer en cambio
1. El agente de IA rompe más de lo que arregla	Cada instrucción genera nuevos errores. La app se vuelve progresivamente más inestable con cada iteración.	La deuda técnica acumulada por el código generado sin supervisión supera la capacidad de corrección del agente.	Empezar de cero con mejor arquitectura o contratar un desarrollador para refactorizar el código existente.
2. Necesitas manejar datos sensibles o regulados	La app procesa datos bancarios, datos de salud (expedientes clínicos), datos biométricos o cualquier dato que tenga regulación específica (LFPDPPP, HIPAA, PCI DSS).	El código generado por IA no garantiza cumplimiento regulatorio. Un error de seguridad en datos sensibles puede tener consecuencias legales.	Desarrollo profesional con auditoría de seguridad especializada. El Vibe Coding puede usarse solo para el prototipo de validación.
3. El rendimiento es crítico para el negocio	La app necesita responder a miles de usuarios simultáneos, procesar transacciones en tiempo real con SLA estricto, o manejar grandes volúmenes de datos con baja latencia.	El código generado por Vibe Coding no está optimizado para rendimiento. Funciona para decenas de usuarios, no para miles.	Arquitectura de microservicios con desarrollador senior. Pruebas de carga antes de producción. Infraestructura escalable.
4. Múltiples equipos necesitan mantener el código	El proyecto crece y varias personas necesitan trabajar en el mismo código simultáneamente con control de versiones avanzado.	El código del Vibe Coding tiende a no seguir convenciones ni documentación que faciliten el trabajo en equipo.	Establecer un equipo de desarrollo con estándares de código, revisar y documentar el código existente.
5. Necesitas integraciones con sistemas legacy o ERP	Debes conectar con SAP, Oracle, sistemas de facturación del SAT o cualquier sistema empresarial con protocolos propietarios.	Las APIs de sistemas legacy no están documentadas para el Vibe Coding y requieren conocimiento específico del sistema.	Desarrollador especialista en el sistema legacy específico. n8n puede ayudar como capa intermedia en algunos casos.

Señal	Descripción de la situación	Por que el Vibe Coding no alcanza	Qué hacer en cambio
6. El sistema tiene reglas de negocio muy complejas	La lógica de negocio tiene decenas de excepciones, casos especiales, reglas condicionales encadenadas o cumplimiento de normativa interna compleja.	El agente de IA tiende a simplificar la lógica y omitir casos bordes. Los errores en reglas de negocio complejas son difíciles de detectar.	Modelado formal de la lógica de negocio primero, luego desarrollo guiado. El Vibe Coding puede implementar cada regla por separado una vez definidas claramente.
7. Necesitas garantías contractuales o SLA de disponibilidad	Un cliente o contrato exige disponibilidad del 99.9%, tiempo máximo de respuesta o penalizaciones por fallos.	El Vibe Coding no produce sistemas que cumplan SLAs de nivel empresarial sin infraestructura y monitoreo profesional adicional.	DevOps profesional con monitoreo, alertas, backups automáticos y plan de recuperación ante desastres.
8. El análisis de datos supera lo exploratorio	Necesitas modelos predictivos, segmentación con ML, detección de fraude automática o cualquier análisis que va más allá de estadística descriptiva.	ChatGPT y Claude hacen análisis descriptivo muy bien. Los modelos de ML en producción requieren validación, entrenamiento continuo y MLOps.	Científico de datos o equipo de datos con Databricks, Python y herramientas de MLOps. El Vibe Coding puede construir la interfaz; los datos los maneja el equipo especializado.

### 5.6.3 Lo que la IA no puede hacer por ti en el Vibe Coding

Estas son las habilidades y responsabilidades que siguen siendo enteramente tuyas, independientemente de que tan buena sea la herramienta de Vibe Coding que uses:

<p><b>Lo que SIEMPRE sigue siendo responsabilidad tuya:</b></p> <ul style="list-style-type: none"> <li>* Definir claramente el problema que necesita solución: la IA construye lo que le pides, no lo que realmente necesitas si no sabes articularlo</li> <li>* Verificar que el resultado funciona correctamente en todos los casos de uso reales, no solo en el caso ideal</li> <li>* Proteger los datos de los usuarios y garantizar la privacidad: el código puede funcionar pero ser inseguro</li> <li>* Decidir si el sistema está listo para producción o si necesita más pruebas</li> <li>* Comunicar a los usuarios finales como funciona el sistema y cómo reportar problemas</li> <li>* Planificar el mantenimiento: quien actualiza las dependencias, quien</li> </ul>	<p><b>Lo que la IA hace muy bien en el Vibe Coding:</b></p> <ul style="list-style-type: none"> <li>* Generar el código base funcional mucho más rápido que un desarrollador desde cero</li> <li>* Traducir tu descripción en lenguaje natural a código específico</li> <li>* Corregir errores de sintaxis y lógica simple dentro de su contexto</li> <li>* Proponer múltiples opciones de implementación para el mismo problema</li> <li>* Adaptar el código a un stack tecnológico específico</li> <li>* Explicar el código que genera para que puedas entenderlo</li> </ul>
---	---

corrige errores, quién escala la infraestructura

### 5.6.4 El puente al Nivel 2: cuando el Vibe Coding básico es el inicio

El Vibe Coding básico que aprendiste en este módulo es el punto de entrada. El Nivel 2 del Diplomado lleva estas capacidades a un nivel completamente diferente: agentes autónomos que construyen y modifican software, pipelines de datos complejos con n8n, integraciones con sistemas empresariales y arquitecturas de producción.

Capacidad	Nivel 1 (este modulo)	Nivel 2 del Diplomado
Vibe Coding	Apps simples con Lovable, Bolt, v0. Sin código propio.	Windsurf Cascade y Cursor Composer para sistemas complejos. Revisión y mantenimiento de código propio.
Bases de datos	Supabase o Firebase configurados por Lovable automáticamente	Diseño de esquemas de datos, consultas SQL, migraciones, índices de rendimiento
Integraciones	Google Sheets via CSV, APIs simples con fetch	n8n para pipelines complejos, sistemas legacy, transformación de datos, webhooks bidireccionales
Despliegue	Vercel con un clic desde Bolt o Lovable	CI/CD completo con GitHub Actions, multi-entorno (desarrollo/staging/producción), monitoreo
Análisis de datos	ChatGPT y Claude con archivos CSV/Excel	Databricks para grandes volúmenes, modelos de ML, SQL avanzado, dashboards en vivo con actualización automática
Agentes de IA en apps	Usar GPTs y Gems como asistentes personales	Construir agentes con n8n + LLM que actúen autónomamente dentro de los sistemas de la empresa

## Resumen del Módulo 5 y Checklist de Cierre

### Lo que aprendiste y tu plan de acción de la semana

Tema del módulo	Habilidad central adquirida	Primera acción esta semana
1. Que es Vibe Coding	Entender el paradigma, el flujo del agente y tu rol como director creativo	Instala Windsurf y abre el panel de Cascade con tu primer proyecto de prueba
2. Dashboards con prompts	Elegir el tipo de gráfica correcto y generar dashboards funcionales con tus datos	Crea el dashboard de ventas con Bolt.new o v0 usando el prompt maestro del Tema 2
3. Apps con Lovable, v0 y Bolt	Elegir la herramienta correcta según el tipo de app y el nivel de complejidad	Genera una herramienta interna simple que tu equipo necesita usando el flujo de 4 fases
4. Conectar fuentes de datos	Conectar Google Sheets, APIs y n8n como pipeline a tus visualizaciones	Conecta tu Google Sheets al dashboard generado usando el prompt de fetch del Tema 4
5. Análisis conversacional	Extraer insights de cualquier archivo CSV o Excel con preguntas en lenguaje natural	Sube un archivo de datos real de tu trabajo a ChatGPT o Claude y sigue la secuencia de 5 pasos
6. Límites del Vibe Coding	Aplicar la Matriz de Complejidad para decidir cuándo usar Vibe Coding y cuando escalar	Clasifica tus 3 proyectos digitales pendientes en la matriz y define el nivel de supervisión de cada uno

### Tus 6 tareas de práctica para consolidar el modulo

Tarea	Instrucción específica	Herramienta
1. Dashboard de ventas funcional	Crea un dashboard con al menos 3 gráficas y 4 KPI cards usando datos reales de tu trabajo (exporta un Excel a CSV primero). Publícalo en Vercel y compártelo con alguien de tu equipo para recibir retroalimentación.	Bolt.new + Vercel o v0 + Vercel
2. App interna con base de datos	Construye una herramienta interna que tu equipo necesita hoy: registro de asistencia, seguimiento de tareas, catálogo de clientes o cualquier otra. Debe guardar datos reales y funcionar para más de un usuario.	Lovable + Supabase
3. Pipeline de datos con n8n	Configura un flujo en n8n que lea datos de Google Sheets diariamente y los guarde en Supabase. Conecta ese Supabase al dashboard del ejercicio 1 para que se actualice automáticamente.	n8n + Google Sheets + Supabase
4. Análisis de datos conversacional	Toma el dataset más importante de tu trabajo (puede ser ventas, clientes, operaciones o RH). Siguiendo la secuencia de 5 pasos del Tema 5, genera al menos 3 insights accionables que no conocías antes del análisis.	ChatGPT Plus o Claude con archivo adjunto

Tarea	Instrucción específica	Herramienta
5. Proyecto Vibe Coding completo	Construye un proyecto desde cero usando Windsurf Cascade: crea un directorio del proyecto, describe el objetivo al agente Cascade, deja que planifique y ejecute los cambios, revisa el resultado y pública en GitHub + Vercel.	Windsurf + GitHub + Vercel
6. Evaluación de proyectos pendientes	Usa la Matriz de Complejidad del Tema 6 para evaluar los 3 próximos proyectos digitales de tu empresa o trabajo. Para cada uno: clasifica su posición en la matriz, define que herramientas usaras y si necesitas supervisión profesional adicional.	Google Docs o papel y lápiz

## Checklist de habilidades del Módulo 5

Marca cada item cuando puedas realizarlo con confianza:

- Puedo explicar que es el Vibe Coding y cómo funciona el ciclo de 4 etapas del agente
- Se elegir el tipo de gráfica correcto según el tipo de dato (barras, líneas, dona, KPI cards, tabla)
- Puedo generar un dashboard ejecutivo funcional en Bolt.new o v0 usando el prompt maestro estructurado
- Se elegir entre Lovable, v0 y Bolt.new según el tipo de app, complejidad y stack requerido
- Se la diferencia entre Supabase (SQL relacional) y Firebase (NoSQL tiempo real) y cuando usar cada uno
- Puedo publicar una app en Vercel y conectarla a un repositorio de GitHub para CI/CD automatico
- Se conectar Google Sheets como fuente de datos a un dashboard en React usando fetch
- Entiendo para qué sirve n8n como pipeline de datos (Extract-Transform-Load) entre fuentes y destinos
- Puedo analizar un archivo CSV o Excel con ChatGPT siguiendo la secuencia de 5 pasos del Tema 5
- Puedo aplicar la Matriz de Complejidad para decidir cuando el Vibe Coding es suficiente y cuando escalar