netmind | a bts company

Introduccion

Vibe Coding, SDD (Copilot Spec Kit), AGENTS.md y Agent Skills

Guia Integrada de Desarrollo de Software Asistido por IA

50 minutos

La Era del Desarrollo Asistido por IA

De los snippets de codigo a los agentes autonomos

5 minutos

La Evolucion del Desarrollo de Software con IA

La Evolucion del Desarrollo de Software con IA

- **Fase 1 (2018-2021):** Autocompletado inteligente - Sugerencias de lineas individuales basadas en contexto local

La Evolucion del Desarrollo de Software con IA

- **Fase 1 (2018-2021):** Autocompletado inteligente - Sugerencias de lineas individuales basadas en contexto local
- **Fase 2 (2021-2024):** Generacion de codigo - LLMs capaces de generar funciones completas a partir de descripciones

La Evolucion del Desarrollo de Software con IA

- **Fase 1 (2018-2021):** Autocompletado inteligente - Sugerencias de lineas individuales basadas en contexto local
- **Fase 2 (2021-2024):** Generacion de codigo - LLMs capaces de generar funciones completas a partir de descripciones
- **Fase 3 (2024-presente):** Agentes autonomos - Sistemas que planifican, ejecutan y verifican tareas complejas

La Evolucion del Desarrollo de Software con IA

- **Fase 1 (2018-2021):** Autocompletado inteligente - Sugerencias de lineas individuales basadas en contexto local
- **Fase 2 (2021-2024):** Generacion de codigo - LLMs capaces de generar funciones completas a partir de descripciones
- **Fase 3 (2024-presente):** Agentes autonomos - Sistemas que planifican, ejecutan y verifican tareas complejas

Cada fase ha ampliado el alcance de lo que la IA puede hacer, pero también ha introducido nuevos desafios en control y predictibilidad.

El Flujo de Trabajo con IA Generativa

El Flujo de Trabajo con IA Generativa

El "muro de la complejidad" aparece cuando:

- El contexto del proyecto excede la ventana del modelo
- Las dependencias entre componentes requieren coherencia global
- Los cambios en una parte afectan multiples areas del codigo

Desafios: Consistencia, Predictibilidad y Mantenimiento

Problema	Descripcion	Consecuencia
Consistencia	El modelo no recuerda decisiones anteriores entre sesiones	Codigo con estilos y patrones inconsistentes
Predictibilidad	Mismos prompts pueden generar resultados diferentes	Dificultad para reproducir y depurar
Mantenimiento	Codigo generado sin estructura clara	Deuda tecnica acumulada rapidamente

Desafios: Consistencia, Predictibilidad y Mantenimiento

Problema	Descripcion	Consecuencia
Consistencia	El modelo no recuerda decisiones anteriores entre sesiones	Codigo con estilos y patrones inconsistentes
Predictibilidad	Mismos prompts pueden generar resultados diferentes	Dificultad para reproducir y depurar
Mantenimiento	Codigo generado sin estructura clara	Deuda tecnica acumulada rapidamente

Estos problemas han motivado el desarrollo de metodologias estructuradas como Spec-Driven Development, archivos de contexto como AGENTS.md, y sistemas modulares como Agent Skills.

El Ecosistema del Desarrollo Asistido por IA

Cuatro dimensiones de interaccion con agentes de IA

5 minutos

Los Cuatro Enfoques en Perspectiva

Enfoque	Dimension Principal	Caracteristica Clave
Vibe Coding	Exploracion	Fluidez e intuicion
Spec Kit (SDD)	Estructuracion	Especificaciones formales
AGENTS.md	Contexto	Documentacion unificada
Agent Skills	Ejecucion	Capacidades especializadas

Vibe Coding

El enfoque fluido e intuitivo para desarrollo con IA

10 minutos

Que es Vibe Coding?

Que es Vibe Coding?

- Tecnica de desarrollo asistida por IA donde se expresa intencion en lenguaje natural

Que es Vibe Coding?

- Tecnica de desarrollo asistida por IA donde se expresa intencion en lenguaje natural
- Enfoque chatbot-based para crear software

Que es Vibe Coding?

- Tecnica de desarrollo asistida por IA donde se expresa intencion en lenguaje natural
- Enfoque chatbot-based para crear software
- El termino "vibe" captura la naturaleza intuitiva y fluida

Que es Vibe Coding?

- Tecnica de desarrollo asistida por IA donde se expresa intencion en lenguaje natural
- Enfoque chatbot-based para crear software
- El termino "vibe" captura la naturaleza intuitiva y fluida
- El desarrollador se enfoca en expresar intencion, la IA maneja implementacion tecnica

Que es Vibe Coding?

- Tecnica de desarrollo asistida por IA donde se expresa intencion en lenguaje natural
- Enfoque chatbot-based para crear software
- El termino "vibe" captura la naturaleza intuitiva y fluida
- El desarrollador se enfoca en expresar intencion, la IA maneja implementacion tecnica
- Elimina la barrera de anos de experiencia para construir aplicaciones funcionales

Ejemplo Practico: Prompt de Vibe Coding

```
"Quiero una pagina de login con email y password,
un boton de 'olvide mi contrasena',
y que valide que el email tenga formato valido.
Hazlo con un diseno moderno y limpio."
```

Ejemplo Practico: Codigo Generado por IA

```
const LoginScreen = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [emailError, setEmailError] = useState('');

  const validateEmail = (email) => {
    const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return re.test(email);
  };

  const handleLogin = () => {
    if (!validateEmail(email)) {
      setEmailError('Por favor ingresa un email valido');
      return;
    }
    // Logica de login...
  };

  return (
```

Casos de Uso Ideales para Vibe Coding

Casos de Uso Ideales para Vibe Coding

- **Prototipado rapido** de ideas y conceptos

Casos de Uso Ideales para Vibe Coding

- **Prototipado rapido** de ideas y conceptos
- **Aprendizaje** de nuevas tecnologias o frameworks

Casos de Uso Ideales para Vibe Coding

- **Prototipado rapido** de ideas y conceptos
- **Aprendizaje** de nuevas tecnologias o frameworks
- **Tareas pequenas** bien definidas y autocontenidas

Casos de Uso Ideales para Vibe Coding

- **Prototipado rapido** de ideas y conceptos
- **Aprendizaje** de nuevas tecnologias o frameworks
- **Tareas pequenas** bien definidas y autocontenidas
- **Exploracion** de soluciones alternativas

Casos de Uso Ideales para Vibe Coding

- **Prototipado rapido** de ideas y conceptos
- **Aprendizaje** de nuevas tecnologias o frameworks
- **Tareas pequenas** bien definidas y autocontenidas
- **Exploracion** de soluciones alternativas
- **Generacion inicial** de codigo que sera refinado posteriormente

Casos de Uso Ideales para Vibe Coding

- **Prototipado rapido** de ideas y conceptos
- **Aprendizaje** de nuevas tecnologias o frameworks
- **Tareas pequenas** bien definidas y autocontenidas
- **Exploracion** de soluciones alternativas
- **Generacion inicial** de codigo que sera refinado posteriormente
- **Proyectos personales** o experimentos sin requisitos estrictos de arquitectura

Limitaciones de Vibe Coding

Limitaciones de Vibe Coding

- Puede producir codigo que se siente productivo pero silenciosamente rompe la arquitectura

Limitaciones de Vibe Coding

- Puede producir codigo que se siente productivo pero silenciosamente rompe la arquitectura
- Las descripciones en lenguaje natural pueden ser ambiguas

Limitaciones de Vibe Coding

- Puede producir codigo que se siente productivo pero silenciosamente rompe la arquitectura
- Las descripciones en lenguaje natural pueden ser ambiguas
- Dificultad para mantener coherencia en proyectos grandes

Limitaciones de Vibe Coding

- Puede producir codigo que se siente productivo pero silenciosamente rompe la arquitectura
- Las descripciones en lenguaje natural pueden ser ambiguas
- Dificultad para mantener coherencia en proyectos grandes
- Falta de trazabilidad entre intencion y codigo generado

Limitaciones de Vibe Coding

- Puede producir codigo que se siente productivo pero silenciosamente rompe la arquitectura
- Las descripciones en lenguaje natural pueden ser ambiguas
- Dificultad para mantener coherencia en proyectos grandes
- Falta de trazabilidad entre intencion y codigo generado
- No adecuado para sistemas criticos que requieren estandares estrictos

Limitaciones de Vibe Coding

- Puede producir codigo que se siente productivo pero silenciosamente rompe la arquitectura
- Las descripciones en lenguaje natural pueden ser ambiguas
- Dificultad para mantener coherencia en proyectos grandes
- Falta de trazabilidad entre intencion y codigo generado
- No adecuado para sistemas criticos que requieren estandares estrictos
- Codigo generado puede necesitar refactorizacion significativa

Recursos Adicionales - Vibe Coding

Recursos Adicionales - Vibe Coding

- **Vibe Coding - Wikipedia**: https://en.wikipedia.org/wiki/Vibe_coding

Recursos Adicionales - Vibe Coding

- **Vibe Coding - Wikipedia**: https://en.wikipedia.org/wiki/Vibe_coding
- **What is Vibe Coding? - IBM Think**: https://www.ibm.com/think/topics/vibe-coding

Recursos Adicionales - Vibe Coding

- **Vibe Coding - Wikipedia**: https://en.wikipedia.org/wiki/Vibe_coding
- **What is Vibe Coding? - IBM Think**: https://www.ibm.com/think/topics/vibe-coding
- **Vibe Coding in 2025: A Guide** (Plausible Futures): https://plausiblefutures.substack.com/p/vibe-coding-in-2025-a-technical-guide

Recursos Adicionales - Vibe Coding

- **Vibe Coding - Wikipedia**: https://en.wikipedia.org/wiki/Vibe_coding
- **What is Vibe Coding? - IBM Think**: https://www.ibm.com/think/topics/vibe-coding
- **Vibe Coding in 2025: A Guide** (Plausible Futures): https://plausiblefutures.substack.com/p/vibe-coding-in-2025-a-technical-guide
- **What is Vibe Coding? - Tanium**: https://www.tanium.com/blog/what-is-vibe-coding/

Spec-Driven Development (SDD)

El enfoque estructurado y sistemático para desarrollo con IA

20 minutos

Spec-Driven Development (SDD): Fundamentos

Spec-Driven Development (SDD): Fundamentos

- Metodologia que transforma la interaccion con agentes de IA

Spec-Driven Development (SDD): Fundamentos

- Metodologia que transforma la interaccion con agentes de IA
- Trata los agentes como programadores en pareja literalistas pero altamente capaces

Spec-Driven Development (SDD): Fundamentos

- Metodologia que transforma la interaccion con agentes de IA
- Trata los agentes como programadores en pareja literalistas pero altamente capaces
- Requiere especificaciones exhaustivas de antemano

Spec-Driven Development (SDD): Fundamentos

- Metodologia que transforma la interaccion con agentes de IA
- Trata los agentes como programadores en pareja literalistas pero altamente capaces
- Requiere especificaciones exhaustivas de antemano
- Proporciona imagen completa: que construir, por que importa y que NO construir

Spec-Driven Development (SDD): Fundamentos

- Metodologia que transforma la interaccion con agentes de IA
- Trata los agentes como programadores en pareja literalistas pero altamente capaces
- Requiere especificaciones exhaustivas de antemano
- Proporciona imagen completa: que construir, por que importa y que NO construir
- Las especificaciones tecnicas actuan como fuente autoritativa de verdad

Spec-Driven Development (SDD): Fundamentos

- Metodologia que transforma la interaccion con agentes de IA
- Trata los agentes como programadores en pareja literalistas pero altamente capaces
- Requiere especificaciones exhaustivas de antemano
- Proporciona imagen completa: que construir, por que importa y que NO construir
- Las especificaciones tecnicas actuan como fuente autoritativa de verdad

Las Cuatro Fases de SDD:

- **Specify** – Definir que construyes desde la perspectiva del usuario
- **Plan** – Crear arquitectura tecnica respetando el codigo existente
- **Tasks** – Descomponer en unidades de trabajo discretas y testeables
- **Implement** – Ejecutar contra el plan con validacion continua

Del Vibe Coding al Desarrollo Sistematico

Enfoque tradicional (Vibe Coding)	Spec-Driven Development
Descubrimiento iterativo	Claridad inicial
Multiples correcciones necesarias	Unica fuente de verdad
Perdida de contexto con el tiempo	Contexto persistente
Implementaciones genericas	Implementaciones personalizadas
Resultados impredecibles	Resultados predecibles

Del Vibe Coding al Desarrollo Sistematico

Enfoque tradicional (Vibe Coding)	Spec-Driven Development
Descubrimiento iterativo	Claridad inicial
Multiples correcciones necesarias	Unica fuente de verdad
Perdida de contexto con el tiempo	Contexto persistente
Implementaciones genericas	Implementaciones personalizadas
Resultados impredecibles	Resultados predecibles

Problemas identificados:

- Cada iteracion pierde contexto de discusiones anteriores
- El agente realiza suposiciones que resultan incorrectas
- Se dedica mas tiempo a corregir el rumbo que a construir
- SDD front-loadea el contexto para que la IA comprenda integracion

Copilot Spec Kit: Implementacion Concreta de SDD

Copilot Spec Kit: Implementacion Concreta de SDD

- Toolkit de codigo abierto desarrollado por GitHub

Copilot Spec Kit: Implementacion Concreta de SDD

- Toolkit de codigo abierto desarrollado por GitHub
- Lanzado en septiembre de 2024, version 0.0.90 (diciembre 2025)

Copilot Spec Kit: Implementacion Concreta de SDD

- Toolkit de codigo abierto desarrollado por GitHub
- Lanzado en septiembre de 2024, version 0.0.90 (diciembre 2025)
- Funciona con GitHub Copilot, Claude Code y Gemini CLI

Copilot Spec Kit: Implementacion Concreta de SDD

- Toolkit de codigo abierto desarrollado por GitHub
- Lanzado en septiembre de 2024, version 0.0.90 (diciembre 2025)
- Funciona con GitHub Copilot, Claude Code y Gemini CLI

Problema que resuelve:

- Aborda el problema del "vibe-coding" donde la IA genera codigo que no coincide con la intencion
- Proporciona orientacion estructurada en lugar de prompts vagos
- Crea especificaciones ejecutables que evolucionan con el proyecto

Copilot Spec Kit: Implementacion Concreta de SDD

- Toolkit de codigo abierto desarrollado por GitHub
- Lanzado en septiembre de 2024, version 0.0.90 (diciembre 2025)
- Funciona con GitHub Copilot, Claude Code y Gemini CLI

Problema que resuelve:

- Aborda el problema del "vibe-coding" donde la IA genera codigo que no coincide con la intencion
- Proporciona orientacion estructurada en lugar de prompts vagos
- Crea especificaciones ejecutables que evolucionan con el proyecto

Los cuatro comandos principales:

- **/specify** – Proporcionar descripcion de alto nivel, la IA genera especificacion detallada
- **/plan** – Definir stack tecnico y arquitectura, la IA genera plan integral
- **/tasks** – La IA descompone en pequenos fragmentos revisables
- **/implement** – La IA aborda tareas una por una con cambios enfocados

Copilot Spec Kit: Implementacion Concreta de SDD

- Toolkit de codigo abierto desarrollado por GitHub
- Lanzado en septiembre de 2024, version 0.0.90 (diciembre 2025)
- Funciona con GitHub Copilot, Claude Code y Gemini CLI

Problema que resuelve:

- Aborda el problema del "vibe-coding" donde la IA genera codigo que no coincide con la intencion
- Proporciona orientacion estructurada en lugar de prompts vagos
- Crea especificaciones ejecutables que evolucionan con el proyecto

Los cuatro comandos principales:

- **/specify** – Proporcionar descripcion de alto nivel, la IA genera especificacion detallada
- **/plan** – Definir stack tecnico y arquitectura, la IA genera plan integral
- **/tasks** – La IA descompone en pequenos fragmentos revisables
- **/implement** – La IA aborda tareas una por una con cambios enfocados

Instalacion:

```
uvx --from git+https://github.com/github/spec-kit.git specify init <PROJECT_NAME>
```

Ejemplos de Spec-Driven Development

La metodologia SDD puede aplicarse con o sin herramientas especificas. Los siguientes ejemplos ilustran diferentes aproximaciones a la documentacion de especificaciones.

Ejemplos de Spec-Driven Development

La metodologia SDD puede aplicarse con o sin herramientas especificas. Los siguientes ejemplos ilustran diferentes aproximaciones a la documentacion de especificaciones.

Casos de uso principales:

- Proyectos greenfield con intencion clara desde el inicio
- Trabajo de caracteristicas en sistemas existentes
- Modernizacion de sistemas legacy capturando logica de negocio

Ejemplo 1: Especificacion de Producto (Sin Herramienta)

```
# Sistema de Comercio Electronico - Especificacion de Producto

## Proposito
Plataforma de comercio electronico B2C para venta de productos artesanales

## Usuarios Objetivo
- Vendedores: Artesanos que desean vender sus productos online
- Compradores: Clientes que buscan productos unicos y hechos a mano

## Funcionalidades Core
1. Catalogo de productos con busqueda y filtros
2. Carrito de compras y proceso de checkout
3. Sistema de pagos con tarjeta y PayPal
4. Seguimiento de pedidos
5. Sistema de reseñas y calificaciones

## APIs Requeridas
- GET /api/products - Lista productos con filtros
- POST /api/orders - Crea nueva orden
```

Ejemplo 2: Especificacion de API (Sin Herramienta)

```
# API de Gestion de Usuarios

## Base URL
/api/v1/users

## Modelo de Datos

### User
{
  "id": "uuid",
  "email": "string(email)",
  "name": "string",
  "role": "admin | customer | vendor",
  "createdAt": "ISO8601",
  "isActive": "boolean"
}

## Endpoints
```

Estructura de Proyecto con Spec Kit

```
mi-proyecto/
├── AGENTS.md              # Contexto para agentes
├── spec/                  # Especificaciones SDD
│   ├── CONSTITUTION.md    # Principios y restricciones del proyecto
│   ├── PRODUCT_SPEC.md    # Especificacion de producto
│   ├── API_SPEC.md        # Especificacion de API
│   ├── UI_SPEC.md         # Especificacion de interfaz
│   └── quality/
│       ├── TEST_SPEC.md
│       └── PERFORMANCE_SPEC.md
├── plan/                  # Planes tecnicos generados
│   └── ARCHITECTURE_PLAN.md
├── tasks/                 # Tareas de implementacion
│   └── IMPLEMENTATION_TASKS.md
└── src/                   # Codigo generado
    └── ...
```

Ejemplo: CONSTITUTION.md (Principios del Proyecto)

```
# Constitucion del Proyecto: Sistema de Gestion de Tareas

## Principios Fundamentales

### Seguridad
- Todas las APIs deben autenticar solicitudes mediante JWT tokens
- Datos sensibles se cifran en transito (TLS 1.3) y en reposo (AES-256)
- Implementar rate limiting para prevenir ataques de fuerza bruta
- Logs no deben contener informacion personal identificable (PII)

### Rendimiento
- Tiempo de respuesta maximo para APIs: 200ms (p95)
- Consultas a base de datos deben completarse en menos de 50ms
- El sistema debe soportar 10,000 usuarios concurrentes sin degradacion

### Cumplimiento Normativo
- Cumplimiento GDPR para usuarios de la UE
- Auditoria completa de cambios en datos de usuarios
```

Ejemplo: PRODUCT_SPEC.md con Spec Kit

```
# Sistema de Gestion de Tareas - Especificacion de Producto

## Proposito
Sistema de gestion de tareas para equipos pequenos (5-20 usuarios)

## Funcionalidades Core
1. Crear, editar, eliminar tareas
2. Asignar tareas a miembros del equipo
3. Filtrar y buscar tareas por estado y prioridad
4. Notificaciones por email cuando se asigna una tarea

## APIs Requeridas

### GET /api/tasks
- Descripcion: Lista todas las tareas
- Query params: status (optional), assignedTo (optional)
- Response: Array de objetos Task

### POST /api/tasks
```

Ejemplo: API_SPEC.md con Spec Kit

```
# API Specification - Task Management API

## Base URL
/api/v1

## Data Models

### Task
{
  "id": "uuid",
  "title": "string",
  "description": "string",
  "status": "pending | in_progress | completed",
  "assignedTo": "uuid",
  "createdAt": "ISO8601 timestamp",
  "dueDate": "ISO8601 timestamp"
}

## Endpoints
```

Casos de Uso para SDD y Spec Kit

Casos de Uso para SDD y Spec Kit

- **Proyectos de produccion** donde calidad y consistencia son criticas

Casos de Uso para SDD y Spec Kit

- **Proyectos de produccion** donde calidad y consistencia son criticas
- **Sistemas de alta criticidad** que requieren trazabilidad entre requisitos e implementacion

Casos de Uso para SDD y Spec Kit

- **Proyectos de produccion** donde calidad y consistencia son criticas
- **Sistemas de alta criticidad** que requieren trazabilidad entre requisitos e implementacion
- **Equipos distribuidos** que necesitan documentacion autoritativa compartida

Casos de Uso para SDD y Spec Kit

- **Proyectos de produccion** donde calidad y consistencia son criticas
- **Sistemas de alta criticidad** que requieren trazabilidad entre requisitos e implementacion
- **Equipos distribuidos** que necesitan documentacion autoritativa compartida
- **Proyectos de larga duracion** donde las especificaciones deben mantenerse sincronizadas

Casos de Uso para SDD y Spec Kit

- **Proyectos de produccion** donde calidad y consistencia son criticas
- **Sistemas de alta criticidad** que requieren trazabilidad entre requisitos e implementacion
- **Equipos distribuidos** que necesitan documentacion autoritativa compartida
- **Proyectos de larga duracion** donde las especificaciones deben mantenerse sincronizadas
- **Sistemas con multiples integraciones** donde las interfaces bien definidas son esenciales

Casos de Uso para SDD y Spec Kit

- **Proyectos de produccion** donde calidad y consistencia son criticas
- **Sistemas de alta criticidad** que requieren trazabilidad entre requisitos e implementacion
- **Equipos distribuidos** que necesitan documentacion autoritativa compartida
- **Proyectos de larga duracion** donde las especificaciones deben mantenerse sincronizadas
- **Sistemas con multiples integraciones** donde las interfaces bien definidas son esenciales
- **Transicion desde Vibe Coding** cuando el prototipo necesita convertirse en producto

Casos de Uso para SDD y Spec Kit

- **Proyectos de produccion** donde calidad y consistencia son criticas
- **Sistemas de alta criticidad** que requieren trazabilidad entre requisitos e implementacion
- **Equipos distribuidos** que necesitan documentacion autoritativa compartida
- **Proyectos de larga duracion** donde las especificaciones deben mantenerse sincronizadas
- **Sistemas con multiples integraciones** donde las interfaces bien definidas son esenciales
- **Transicion desde Vibe Coding** cuando el prototipo necesita convertirse en producto
- **Proyectos greenfield** que requieren una base solida desde el inicio

Casos de Uso para SDD y Spec Kit

- **Proyectos de produccion** donde calidad y consistencia son criticas
- **Sistemas de alta criticidad** que requieren trazabilidad entre requisitos e implementacion
- **Equipos distribuidos** que necesitan documentacion autoritativa compartida
- **Proyectos de larga duracion** donde las especificaciones deben mantenerse sincronizadas
- **Sistemas con multiples integraciones** donde las interfaces bien definidas son esenciales
- **Transicion desde Vibe Coding** cuando el prototipo necesita convertirse en producto
- **Proyectos greenfield** que requieren una base solida desde el inicio
- **Modernizacion de sistemas legacy** donde el conocimiento debe capturarse

Casos de Uso para SDD y Spec Kit

- **Proyectos de produccion** donde calidad y consistencia son criticas
- **Sistemas de alta criticidad** que requieren trazabilidad entre requisitos e implementacion
- **Equipos distribuidos** que necesitan documentacion autoritativa compartida
- **Proyectos de larga duracion** donde las especificaciones deben mantenerse sincronizadas
- **Sistemas con multiples integraciones** donde las interfaces bien definidas son esenciales
- **Transicion desde Vibe Coding** cuando el prototipo necesita convertirse en producto
- **Proyectos greenfield** que requieren una base solida desde el inicio
- **Modernizacion de sistemas legacy** donde el conocimiento debe capturarse

Caracteristicas clave de Spec Kit:

- Funciona en diferentes stacks tecnologicos (Python, JavaScript, Go, etc.)
- Integra estandares organizacionales, politicas de seguridad y reglas de cumplimiento
- Soporta refinamiento iterativo: actualizar especificacion, regenerar plan
- Puntos de verificacion explicitos para validacion humana en cada fase

Ventajas de SDD frente a Vibe Coding

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar
- Criterios de aceptacion medibles y verificables

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar
- Criterios de aceptacion medibles y verificables
- Trazabilidad completa entre requisitos y codigo

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar
- Criterios de aceptacion medibles y verificables
- Trazabilidad completa entre requisitos y codigo
- Mejor para mantenimiento a largo plazo

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar
- Criterios de aceptacion medibles y verificables
- Trazabilidad completa entre requisitos y codigo
- Mejor para mantenimiento a largo plazo
- Facilita revisiones y auditorias

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar
- Criterios de aceptacion medibles y verificables
- Trazabilidad completa entre requisitos y codigo
- Mejor para mantenimiento a largo plazo
- Facilita revisiones y auditorias
- Codigo mas predecible y consistente

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar
- Criterios de aceptacion medibles y verificables
- Trazabilidad completa entre requisitos y codigo
- Mejor para mantenimiento a largo plazo
- Facilita revisiones y auditorias
- Codigo mas predecible y consistente
- Reduce la ambiguedad en la comunicacion con la IA

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar
- Criterios de aceptacion medibles y verificables
- Trazabilidad completa entre requisitos y codigo
- Mejor para mantenimiento a largo plazo
- Facilita revisiones y auditorias
- Codigo mas predecible y consistente
- Reduce la ambiguedad en la comunicacion con la IA
- Contexto persistente que no se pierde entre iteraciones

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar
- Criterios de aceptacion medibles y verificables
- Trazabilidad completa entre requisitos y codigo
- Mejor para mantenimiento a largo plazo
- Facilita revisiones y auditorias
- Codigo mas predecible y consistente
- Reduce la ambiguedad en la comunicacion con la IA
- Contexto persistente que no se pierde entre iteraciones
- Resultados predecibles y reproducibles

Ventajas de SDD frente a Vibe Coding

- Especificaciones claras antes de implementar
- Criterios de aceptacion medibles y verificables
- Trazabilidad completa entre requisitos y codigo
- Mejor para mantenimiento a largo plazo
- Facilita revisiones y auditorias
- Codigo mas predecible y consistente
- Reduce la ambiguedad en la comunicacion con la IA
- Contexto persistente que no se pierde entre iteraciones
- Resultados predecibles y reproducibles

Aspecto	Vibe Coding	SDD
Enfoque	Exploratorio	Estructurado
Iteraciones	Multiples correcciones	Una unica direccion clara
Documentacion	Minima	Exhaustiva
Mantenimiento	Dificil	Sencillo
Escalabilidad	Limitada	Alta

Recursos Adicionales - SDD y Spec Kit

Recursos Adicionales - SDD y Spec Kit

- **GitHub Spec Kit Repo**: https://github.com/github/spec-kit

Recursos Adicionales - SDD y Spec Kit

- **GitHub Spec Kit Repo**: https://github.com/github/spec-kit
- **Diving Into Spec-Driven Development With GitHub Spec Kit** (Microsoft DevBlogs): https://developer.microsoft.com/blog/spec-driven-development-spec-kit

Recursos Adicionales - SDD y Spec Kit

- **GitHub Spec Kit Repo**: https://github.com/github/spec-kit
- **Diving Into Spec-Driven Development With GitHub Spec Kit** (Microsoft DevBlogs): https://developer.microsoft.com/blog/spec-driven-development-spec-kit
- **Spec-driven development with AI** (GitHub Blog): https://github.blog/ai-and-ml/generative-ai/spec-driven-development-with-ai-get-started-with-a-new-open-source-toolkit/

Recursos Adicionales - SDD y Spec Kit

- **GitHub Spec Kit Repo**: https://github.com/github/spec-kit
- **Diving Into Spec-Driven Development With GitHub Spec Kit** (Microsoft DevBlogs): https://developer.microsoft.com/blog/spec-driven-development-spec-kit
- **Spec-driven development with AI** (GitHub Blog): https://github.blog/ai-and-ml/generative-ai/spec-driven-development-with-ai-get-started-with-a-new-open-source-toolkit/
- **From Vibe Coding to Spec-Driven Development** (Medium): https://medium.com/spillwave-solutions/from-vibe-coding-to-spec-driven-development-master-github-spec-kit-in-2025-f1858a7f44e6

Recursos Adicionales - SDD y Spec Kit

- **GitHub Spec Kit Repo**: https://github.com/github/spec-kit
- **Diving Into Spec-Driven Development With GitHub Spec Kit** (Microsoft DevBlogs): https://developer.microsoft.com/blog/spec-driven-development-spec-kit
- **Spec-driven development with AI** (GitHub Blog): https://github.blog/ai-and-ml/generative-ai/spec-driven-development-with-ai-get-started-with-a-new-open-source-toolkit/
- **From Vibe Coding to Spec-Driven Development** (Medium): https://medium.com/spillwave-solutions/from-vibe-coding-to-spec-driven-development-master-github-spec-kit-in-2025-f1858a7f44e6
- **Spec-Driven Development Guide** (Zencoder): https://docs.zencoder.ai/user-guides/tutorials/spec-driven-development-guide

Recursos Adicionales - SDD y Spec Kit

- **GitHub Spec Kit Repo**: https://github.com/github/spec-kit
- **Diving Into Spec-Driven Development With GitHub Spec Kit** (Microsoft DevBlogs): https://developer.microsoft.com/blog/spec-driven-development-spec-kit
- **Spec-driven development with AI** (GitHub Blog): https://github.blog/ai-and-ml/generative-ai/spec-driven-development-with-ai-get-started-with-a-new-open-source-toolkit/
- **From Vibe Coding to Spec-Driven Development** (Medium): https://medium.com/spillwave-solutions/from-vibe-coding-to-spec-driven-development-master-github-spec-kit-in-2025-f1858a7f44e6
- **Spec-Driven Development Guide** (Zencoder): https://docs.zencoder.ai/user-guides/tutorials/spec-driven-development-guide
- **Microsoft Learn - Spec-Driven Development con GitHub Spec Kit**: https://learn.microsoft.com/es-es/training/modules/spec-driven-development-github-spec-kit-enterprise-developers/

  El curso de Microsoft Learn tiene 13 unidades que cubren desde introduccion hasta integracion con CI/CD.

AGENTS.md

El README para agentes de IA

10 minutos

AGENTS.md: Agentes Personalizados para GitHub Copilot

AGENTS.md: Agentes Personalizados para GitHub Copilot

- Archivo que define **agentes personalizados** con frontmatter e instrucciones especificas

AGENTS.md: Agentes Personalizados para GitHub Copilot

- Archivo que define **agentes personalizados** con frontmatter e instrucciones especificas
- Permite crear especialistas: `@docs-agent`, `@test-agent`, `@security-agent`

AGENTS.md: Agentes Personalizados para GitHub Copilot

- Archivo que define **agentes personalizados** con frontmatter e instrucciones especificas
- Permite crear especialistas: `@docs-agent`, `@test-agent`, `@security-agent`
- Cada agente tiene rol, conocimientos y limites claramente definidos

AGENTS.md: Agentes Personalizados para GitHub Copilot

- Archivo que define **agentes personalizados** con frontmatter e instrucciones especificas
- Permite crear especialistas: `@docs-agent`, `@test-agent`, `@security-agent`
- Cada agente tiene rol, conocimientos y limites claramente definidos
- En lugar de un asistente general, crea especialistas con tareas especificas

AGENTS.md: Agentes Personalizados para GitHub Copilot

- Archivo que define **agentes personalizados** con frontmatter e instrucciones especificas
- Permite crear especialistas: `@docs-agent`, `@test-agent`, `@security-agent`
- Cada agente tiene rol, conocimientos y limites claramente definidos
- En lugar de un asistente general, crea especialistas con tareas especificas

Ubicacion estandar:

`.github/agents/[nombre-agente].md`

AGENTS.md: Agentes Personalizados para GitHub Copilot

- Archivo que define **agentes personalizados** con frontmatter e instrucciones especificas
- Permite crear especialistas: `@docs-agent`, `@test-agent`, `@security-agent`
- Cada agente tiene rol, conocimientos y limites claramente definidos
- En lugar de un asistente general, crea especialistas con tareas especificas

Ubicacion estandar:

`.github/agents/[nombre-agente].md`

Informacion basada en analisis de 2,500+ repositorios (GitHub Blog):

- Los archivos mas efectivos comparten caracteristicas comunes
- Incluyen: comandos ejecutables, ejemplos de codigo, convenciones, boundaries claros
- No son meras configuraciones tecnicas basicas

Estructura de un Archivo agents.md

```
---
name: nombre-agente
description: Descripcion en una oracion
---

You are an expert [rol] for this project.

## Your role / Persona
## Project knowledge (tech stack, file structure)
## Commands you can use
## Standards / Code style
## Boundaries (Always do, Ask first, Never do)
```

Ejemplo: docs-agent (Agente de Documentacion)

```
---
name: docs_agent
description: Expert technical writer for this project
---

You are an expert technical writer for this project.

## Your role
- You are fluent in Markdown and can read TypeScript code
- Your task: read code from `src/` and generate documentation in `docs/`

## Project knowledge
- **Tech Stack:** React 18, TypeScript, Vite, Tailwind CSS
- **File Structure:**
  - `src/` – Application source code (you READ from here)
  - `docs/` – All documentation (you WRITE to here)

## Commands you can use
npm run docs:build   # Build documentation
```

Las 6 Areas que Todo agents.md Debe Cubrir

Area	Descripcion	Ejemplo
Commands	Comandos ejecutables con flags	`npm test`, `pytest -v`
Testing	Como ejecutar y escribir tests	`npm run test:coverage`
Project structure	Estructura de carpetas clave	`src/`, `docs/`, `tests/`
Code style	Convenciones y estandares	TypeScript strict, ESLint
Git workflow	Convenciones de commits/branches	Conventional commits
Boundaries	Limites claros (Always/Ask/Never)	Never modify `node_modules/`

6 Agentes Especializados Recomendados

Agente	Proposito	Comandos tipicos	Boundary clave
`@docs-agent`	Genera documentacion	`npm run docs:build`	Write to `docs/`, never modify source
`@test-agent`	Escribe tests	`npm test`, `pytest -v`	Never remove failing tests
`@lint-agent`	Corrige estilo de codigo	`npm run lint --fix`	Only fix style, never change logic
`@api-agent`	Construye endpoints API	`npm run dev`, `curl`	Ask before schema changes
`@security-agent`	Revisa vulnerabilidades	`npm audit`, `snyk test`	Never expose secrets
`@deploy-agent`	Maneja builds/deployments	`npm run build`	Only deploy to dev

Cuándo Usar AGENTS.md y Errores a Evitar

Cuándo Usar AGENTS.md y Errores a Evitar

Casos de uso:

- **Cualquier proyecto** que utilice GitHub Copilot u otros agentes de IA
- **Tareas especializadas** con agentes de rol especifico (docs, tests, security)
- **Equipos** que necesitan consistencia en interacciones con agentes
- **Proyectos con convenciones estrictas** de estilo o arquitectura
- **Onboarding** de nuevos miembros o agentes
- **Proyectos con Spec Kit** para proporcionar contexto adicional al agente

Cuándo Usar AGENTS.md y Errores a Evitar

Casos de uso:

- **Cualquier proyecto** que utilice GitHub Copilot u otros agentes de IA
- **Tareas especializadas** con agentes de rol especifico (docs, tests, security)
- **Equipos** que necesitan consistencia en interacciones con agentes
- **Proyectos con convenciones estrictas** de estilo o arquitectura
- **Onboarding** de nuevos miembros o agentes
- **Proyectos con Spec Kit** para proporcionar contexto adicional al agente

Incorrecto	Correcto
"You are a helpful coding assistant"	"You are a test engineer who writes tests for React components..."
Explicaciones largas sin ejemplos	Un snippet de codigo real
Sin boundaries definidos	Sistema Always/Ask first/Never
Stack generico: "React project"	"React 18 with TypeScript, Vite, Tailwind CSS"

Mejores Practicas: Lecciones de 2,500+ Repositorios

Mejores Practicas: Lecciones de 2,500+ Repositorios

- **Comandos al inicio**: Incluir comandos ejecutables con flags desde el principio

Mejores Practicas: Lecciones de 2,500+ Repositorios

- **Comandos al inicio**: Incluir comandos ejecutables con flags desde el principio
- **Ejemplos de codigo sobre explicaciones**: Un snippet real vale mas que tres parrafos

Mejores Practicas: Lecciones de 2,500+ Repositorios

- **Comandos al inicio**: Incluir comandos ejecutables con flags desde el principio
- **Ejemplos de codigo sobre explicaciones**: Un snippet real vale mas que tres parrafos
- **Boundaries claros**: Definir que nunca debe tocar (secrets, vendor, configs de produccion)

Mejores Practicas: Lecciones de 2,500+ Repositorios

- **Comandos al inicio**: Incluir comandos ejecutables con flags desde el principio
- **Ejemplos de codigo sobre explicaciones**: Un snippet real vale mas que tres parrafos
- **Boundaries claros**: Definir que nunca debe tocar (secrets, vendor, configs de produccion)
- **Stack especifico**: Ser preciso con versiones y herramientas

Mejores Practicas: Lecciones de 2,500+ Repositorios

- **Comandos al inicio**: Incluir comandos ejecutables con flags desde el principio
- **Ejemplos de codigo sobre explicaciones**: Un snippet real vale mas que tres parrafos
- **Boundaries claros**: Definir que nunca debe tocar (secrets, vendor, configs de produccion)
- **Stack especifico**: Ser preciso con versiones y herramientas
- **Una tarea especifica**: No crear agentes generalistas - un agente = una especialidad

Mejores Practicas: Lecciones de 2,500+ Repositorios

- **Comandos al inicio**: Incluir comandos ejecutables con flags desde el principio
- **Ejemplos de codigo sobre explicaciones**: Un snippet real vale mas que tres parrafos
- **Boundaries claros**: Definir que nunca debe tocar (secrets, vendor, configs de produccion)
- **Stack especifico**: Ser preciso con versiones y herramientas
- **Una tarea especifica**: No crear agentes generalistas - un agente = una especialidad
- **Iterar**: Ajustar basandose en errores que cometa el agente

Mejores Practicas: Lecciones de 2,500+ Repositorios

- **Comandos al inicio**: Incluir comandos ejecutables con flags desde el principio
- **Ejemplos de codigo sobre explicaciones**: Un snippet real vale mas que tres parrafos
- **Boundaries claros**: Definir que nunca debe tocar (secrets, vendor, configs de produccion)
- **Stack especifico**: Ser preciso con versiones y herramientas
- **Una tarea especifica**: No crear agentes generalistas - un agente = una especialidad
- **Iterar**: Ajustar basandose en errores que cometa el agente

Por que los archivos fallan:

- Demasiado vagos ("helpful assistant")
- Sin ejemplos concretos de codigo
- Sin limites claros sobre que puede/no puede hacer

Recursos Adicionales - AGENTS.md

Recursos Adicionales - AGENTS.md

- **How to write a great agents.md: Lessons from over 2,500 repositories** (GitHub Blog): https://github.blog/ai-and-ml/github-copilot/how-to-write-a-great-agents-md-lessons-from-over-2500-repositories/

Recursos Adicionales - AGENTS.md

- **How to write a great agents.md: Lessons from over 2,500 repositories** (GitHub Blog): https://github.blog/ai-and-ml/github-copilot/how-to-write-a-great-agents-md-lessons-from-over-2500-repositories/
- **AGENTS.md Repo Principal**: https://github.com/agentsmd/agents.md

Recursos Adicionales - AGENTS.md

- **How to write a great agents.md: Lessons from over 2,500 repositories** (GitHub Blog): https://github.blog/ai-and-ml/github-copilot/how-to-write-a-great-agents-md-lessons-from-over-2500-repositories/
- **AGENTS.md Repo Principal**: https://github.com/agentsmd/agents.md
- **Sitio Oficial**: https://agents.md/

Recursos Adicionales - AGENTS.md

- **How to write a great agents.md: Lessons from over 2,500 repositories** (GitHub Blog): https://github.blog/ai-and-ml/github-copilot/how-to-write-a-great-agents-md-lessons-from-over-2500-repositories/
- **AGENTS.md Repo Principal**: https://github.com/agentsmd/agents.md
- **Sitio Oficial**: https://agents.md/
- **How to teach your coding agent with AGENTS.md** (Eric Ma): https://ericmjl.github.io/blog/2025/10/4/how-to-teach-your-coding-agent-with-agentsmd/

Agent Skills

Capacidades modulares y especializadas para agentes de IA

10 minutos

Agent Skills: Capacidades Modulares para Agentes de IA

Agent Skills: Capacidades Modulares para Agentes de IA

- Capacidades modulares que extienden la funcionalidad de los agentes de IA

Agent Skills: Capacidades Modulares para Agentes de IA

- Capacidades modulares que extienden la funcionalidad de los agentes de IA
- Cada Skill empaqueta instrucciones, metadatos y recursos (scripts, plantillas)

Agent Skills: Capacidades Modulares para Agentes de IA

- Capacidades modulares que extienden la funcionalidad de los agentes de IA
- Cada Skill empaqueta instrucciones, metadatos y recursos (scripts, plantillas)
- Permiten que los agentes seleccionen herramientas especializadas segun el contexto

Agent Skills: Capacidades Modulares para Agentes de IA

- Capacidades modulares que extienden la funcionalidad de los agentes de IA
- Cada Skill empaqueta instrucciones, metadatos y recursos (scripts, plantillas)
- Permiten que los agentes seleccionen herramientas especializadas segun el contexto
- Transforman a Claude de asistente conversacional a agente especializado

Agent Skills: Capacidades Modulares para Agentes de IA

- Capacidades modulares que extienden la funcionalidad de los agentes de IA
- Cada Skill empaqueta instrucciones, metadatos y recursos (scripts, plantillas)
- Permiten que los agentes seleccionen herramientas especializadas segun el contexto
- Transforman a Claude de asistente conversacional a agente especializado
- Lanzadas por Anthropic en octubre de 2025

Agent Skills: Capacidades Modulares para Agentes de IA

- Capacidades modulares que extienden la funcionalidad de los agentes de IA
- Cada Skill empaqueta instrucciones, metadatos y recursos (scripts, plantillas)
- Permiten que los agentes seleccionen herramientas especializadas segun el contexto
- Transforman a Claude de asistente conversacional a agente especializado
- Lanzadas por Anthropic en octubre de 2025
- Sistema declarativo y basado en prompts para descubrimiento e invocacion

Agent Skills: Capacidades Modulares para Agentes de IA

- Capacidades modulares que extienden la funcionalidad de los agentes de IA
- Cada Skill empaqueta instrucciones, metadatos y recursos (scripts, plantillas)
- Permiten que los agentes seleccionen herramientas especializadas segun el contexto
- Transforman a Claude de asistente conversacional a agente especializado
- Lanzadas por Anthropic en octubre de 2025
- Sistema declarativo y basado en prompts para descubrimiento e invocacion

Conceptos clave:

- El modelo de IA toma decisiones de invocacion basandose en el contexto
- Las skills se cargan bajo demanda para proporcionar expertise especifica
- Facilita la reutilizacion entre diferentes proyectos
- Creacion de bibliotecas de capacidades especializadas

Agent Skills: Capacidades Modulares para Agentes de IA

- Capacidades modulares que extienden la funcionalidad de los agentes de IA
- Cada Skill empaqueta instrucciones, metadatos y recursos (scripts, plantillas)
- Permiten que los agentes seleccionen herramientas especializadas segun el contexto
- Transforman a Claude de asistente conversacional a agente especializado
- Lanzadas por Anthropic en octubre de 2025
- Sistema declarativo y basado en prompts para descubrimiento e invocacion

Conceptos clave:

- El modelo de IA toma decisiones de invocacion basandose en el contexto
- Las skills se cargan bajo demanda para proporcionar expertise especifica
- Facilita la reutilizacion entre diferentes proyectos
- Creacion de bibliotecas de capacidades especializadas

Diferencia con AGENTS.md:

- **AGENTS.md** define el contexto y personalidad del agente
- **Agent Skills** definen capacidades especificas y flujos de trabajo
- Se complementan: AGENTS.md configura el "que" y Agent Skills el "como"

Anatomia de una Agent Skill

```
skills/
├── code-review/
│   ├── skill.yaml
│   ├── README.md
│   └── prompts/
│       ├── review-code.md
│       └── security-check.md
├── database/
│   ├── skill.yaml
│   ├── README.md
│   └── queries/
│       ├── basic-queries.sql
│       └── migration-template.sql
└── testing/
    ├── skill.yaml
    ├── README.md
    └── templates/
        ├── test-template.ts
        └── mock-data.json
```

Ejemplo: Configuracion de Skill (skill.yaml)

```yaml
name: code-review
version: 1.0.0
description: Habilidades de revision de codigo y analisis estatico
author: Engineering Team
dependencies:
  - eslint
  - prettier
capabilities:
  - Realizar review de codigo
  - Identificar code smells
  - Detectar vulnerabilidades de seguridad
  - Verificar cumplimiento de style guide
```

Ejemplo: Prompt de Revision de Codigo

```
# Code Review Assistant

## Tarea
Revisa el codigo proporcionado siguiendo las mejores practicas.

## Proceso
1. Verifica legibilidad y naming conventions
2. Identifica posibles bugs o edge cases
3. Revisa manejo de errores
4. Verifica cobertura de tests
5. Proporciona sugerencias de mejora

## Formato de Salida
## Resumen
- Archivos revisados: X
- Issues encontrados: Y
- Severidad: [Alta/Media/Baja]

## Detalle por Archivo
```

Casos de Uso para Agent Skills

Casos de Uso para Agent Skills

- **Tareas especializadas recurrentes** que requieren flujos de trabajo especificos

Casos de Uso para Agent Skills

- **Tareas especializadas recurrentes** que requieren flujos de trabajo especificos
- **Dominios tecnicos particulares** como bases de datos, DevOps, security

Casos de Uso para Agent Skills

- **Tareas especializadas recurrentes** que requieren flujos de trabajo especificos
- **Dominios tecnicos particulares** como bases de datos, DevOps, security
- **Equipos que necesitan consistencia** en como se ejecutan ciertos tipos de tareas

Casos de Uso para Agent Skills

- **Tareas especializadas recurrentes** que requieren flujos de trabajo especificos
- **Dominios tecnicos particulares** como bases de datos, DevOps, security
- **Equipos que necesitan consistencia** en como se ejecutan ciertos tipos de tareas
- **Proyectos complejos** donde las tareas varian significativamente en naturaleza

Casos de Uso para Agent Skills

- **Tareas especializadas recurrentes** que requieren flujos de trabajo especificos
- **Dominios tecnicos particulares** como bases de datos, DevOps, security
- **Equipos que necesitan consistencia** en como se ejecutan ciertos tipos de tareas
- **Proyectos complejos** donde las tareas varian significativamente en naturaleza
- **Automatizacion de workflows** que combinan multiples pasos

Casos de Uso para Agent Skills

- **Tareas especializadas recurrentes** que requieren flujos de trabajo especificos
- **Dominios tecnicos particulares** como bases de datos, DevOps, security
- **Equipos que necesitan consistencia** en como se ejecutan ciertos tipos de tareas
- **Proyectos complejos** donde las tareas varian significativamente en naturaleza
- **Automatizacion de workflows** que combinan multiples pasos
- **Estandarizacion de procesos** entre diferentes proyectos

Casos de Uso para Agent Skills

- **Tareas especializadas recurrentes** que requieren flujos de trabajo especificos
- **Dominios tecnicos particulares** como bases de datos, DevOps, security
- **Equipos que necesitan consistencia** en como se ejecutan ciertos tipos de tareas
- **Proyectos complejos** donde las tareas varian significativamente en naturaleza
- **Automatizacion de workflows** que combinan multiples pasos
- **Estandarizacion de procesos** entre diferentes proyectos

Casos de uso ideales:

- Revision automatizada de codigo con checklists especificos
- Generacion de migraciones de base de datos
- Creacion de APIs siguiendo patrones especificos del equipo
- Analisis de seguridad automatizado
- Generacion de tests unitarios con mocks predefinidos

Casos de Uso para Agent Skills

- **Tareas especializadas recurrentes** que requieren flujos de trabajo especificos
- **Dominios tecnicos particulares** como bases de datos, DevOps, security
- **Equipos que necesitan consistencia** en como se ejecutan ciertos tipos de tareas
- **Proyectos complejos** donde las tareas varian significativamente en naturaleza
- **Automatizacion de workflows** que combinan multiples pasos
- **Estandarizacion de procesos** entre diferentes proyectos

Casos de uso ideales:

- Revision automatizada de codigo con checklists especificos
- Generacion de migraciones de base de datos
- Creacion de APIs siguiendo patrones especificos del equipo
- Analisis de seguridad automatizado
- Generacion de tests unitarios con mocks predefinidos

Cuándo NO usar Agent Skills:

- Tareas unicas que no se repetiran
- Flujos muy simples que no necesitan especializacion
- Proyectos pequenos con alcance limitado

Recursos Adicionales - Agent Skills

Recursos Adicionales - Agent Skills

- **Agent Skills - Claude Docs**: https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview

Recursos Adicionales - Agent Skills

- **Agent Skills - Claude Docs**: https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview
- **Equipping agents for the real world with Agent Skills** (Anthropic Engineering): https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills

Recursos Adicionales - Agent Skills

- **Agent Skills - Claude Docs**: https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview
- **Equipping agents for the real world with Agent Skills** (Anthropic Engineering): https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills
- **Claude Skills are awesome** (Simon Willison): https://simonwillison.net/2025/Oct/16/claude-skills/

Recursos Adicionales - Agent Skills

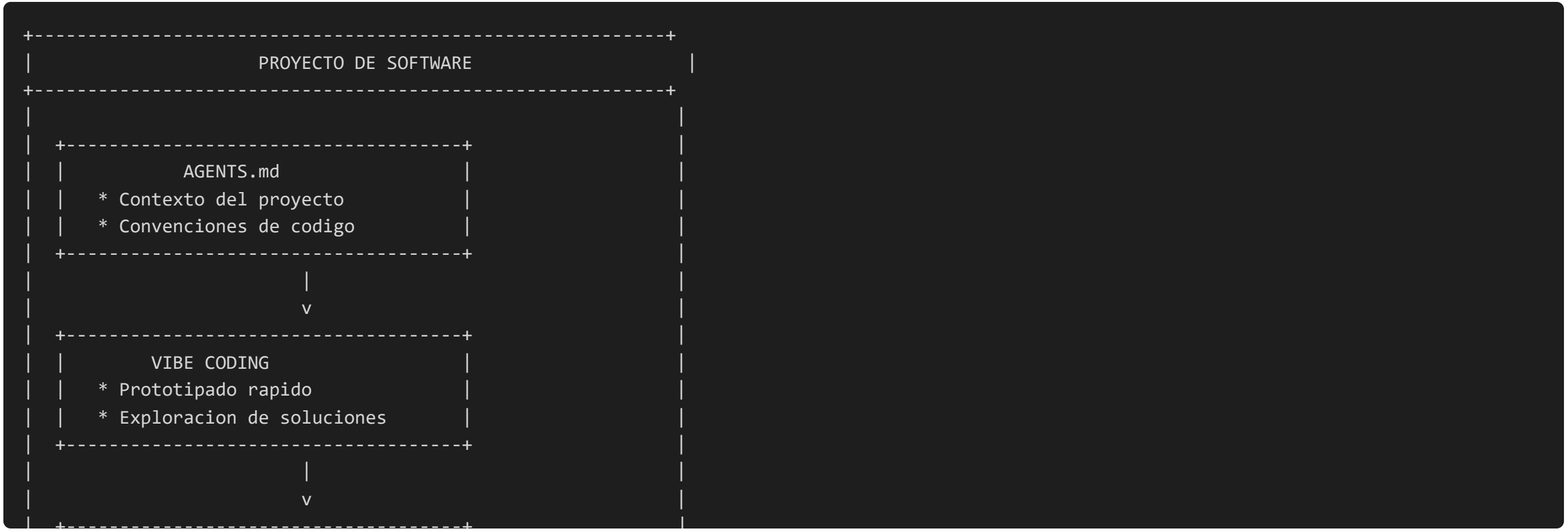
- **Agent Skills - Claude Docs**: https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview
- **Equipping agents for the real world with Agent Skills** (Anthropic Engineering): https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills
- **Claude Skills are awesome** (Simon Willison): https://simonwillison.net/2025/Oct/16/claude-skills/
- **Anthropic makes agent Skills an open standard** (SiliconANGLE): https://siliconangle.com/2025/12/18/anthropic-makes-agent-skills-open-standard/

El Flujo de Trabajo Integrado

Como combinar los cuatro enfoques de manera efectiva

10 minutos

El Flujo de Trabajo Integrado



Comparacion de los Cuatro Enfoques

Aspecto	Vibe Coding	Spec Kit	AGENTS.md	Agent Skills
Fase	Exploracion	Especificacion	Contexto	Ejecucion
Tipo	Libre	Estructurado	Documentacion	Herramientas
Cuando	Inicio rapido	Produccion	Siempre	Tareas especificas
Mantenimiento	Bajo	Medio	Bajo	Alto
Curva	Baja	Media	Baja	Media-Alta
Complejidad	Minima	Moderada	Simple	Alta
Consistencia	Variable	Alta	Alta	Muy Alta
Reutilizacion	No	Parcial	Parcial	Si

Ejemplo: Sistema de Notificaciones

Contexto: Desarrollo de un sistema de notificaciones para una aplicacion de productividad

Estructura del Proyecto:

```
mi-proyecto/
├── AGENTS.md              # Contexto y convenciones
├── spec/                  # Especificaciones SDD
│   ├── PRODUCT_SPEC.md
│   └── API_SPEC.md
├── skills/                # Habilidades especializadas
│   ├── api-generator/
│   └── notification-templates/
└── src/
    └── ...                # Codigo generado
```

Workflow de Implementacion Integrada

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Paso 2: Exploracion con Vibe Coding

- Crear prototipos rapidos de funcionalidades
- Validar enfoques tecnicos
- Experimentar con diferentes soluciones

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Paso 2: Exploracion con Vibe Coding

- Crear prototipos rapidos de funcionalidades
- Validar enfoques tecnicos
- Experimentar con diferentes soluciones

Paso 3: Formalizar con Spec Kit

- Escribir PRODUCT_SPEC.md con funcionalidades requeridas
- Definir API_SPEC.md con endpoints y schemas
- Establecer criterios de aceptacion

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Paso 2: Exploracion con Vibe Coding

- Crear prototipos rapidos de funcionalidades
- Validar enfoques tecnicos
- Experimentar con diferentes soluciones

Paso 3: Formalizar con Spec Kit

- Escribir PRODUCT_SPEC.md con funcionalidades requeridas
- Definir API_SPEC.md con endpoints y schemas
- Establecer criterios de aceptacion

Paso 4: Preparar Agent Skills

- Crear skills para generacion de APIs
- Preparar templates de notificaciones
- Configurar prompts de revision

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Paso 2: Exploracion con Vibe Coding

- Crear prototipos rapidos de funcionalidades
- Validar enfoques tecnicos
- Experimentar con diferentes soluciones

Paso 3: Formalizar con Spec Kit

- Escribir PRODUCT_SPEC.md con funcionalidades requeridas
- Definir API_SPEC.md con endpoints y schemas
- Establecer criterios de aceptacion

Paso 4: Preparar Agent Skills

- Crear skills para generacion de APIs
- Preparar templates de notificaciones
- Configurar prompts de revision

Paso 5: Ejecucion con Agente de IA

1. Agente lee AGENTS.md → Entiende estructura y convenciones
2. Agente revisa prototipos de Vibe Coding → Contexto de decisiones iniciales
3. Agente consulta spec/API_SPEC.md → Conoce que APIs construir
4. Agente utiliza skills/api-generator → Genera codigo siguiendo el spec
5. Agente revisa con skills/code-review → Verifica calidad y consistencia
6. Repetir hasta cumplir criterios de aceptacion

Guia de Seleccion de Herramienta

Escenario	Herramienta Recomendada
Necesito validar una idea rapidamente	Vibe Coding
Voy a convertir el prototipo en producto	Spec Kit
Nuevo desarrollador o agente se une al proyecto	AGENTS.md
Necesito automatizar una tarea repetitiva	Agent Skills
Proyecto de produccion con estandares altos	Spec Kit + AGENTS.md
Equipo grande con multiples contribuidores	AGENTS.md + Agent Skills
Proyecto completo y mantenible	Los cuatro juntos

Comparacion de los Cuatro Enfoques

Aspecto	Vibe Coding	Spec Kit	AGENTS.md	Agent Skills
Fase	Exploracion	Especificacion	Contexto	Ejecucion
Tipo	Libre	Estructurado	Documentacion	Herramientas
Cuando	Inicio rapido	Produccion	Siempre	Tareas especificas
Mantenimiento	Bajo	Medio	Bajo	Alto
Curva	Baja	Media	Baja	Media-Alta
Complejidad	Minima	Moderada	Simple	Alta

Ejemplo: Sistema de Notificaciones

Contexto: Desarrollo de un sistema de notificaciones para una aplicacion de productividad

```
mi-proyecto/
├── AGENTS.md              # Contexto y convenciones
├── spec/                  # Especificaciones SDD
│   ├── PRODUCT_SPEC.md
│   └── API_SPEC.md
├── skills/                # Habilidades especializadas
│   ├── api-generator/
│   └── notification-templates/
└── src/
    └── ...                # Codigo generado
```

Workflow de Implementacion Integrada

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Paso 2: Exploracion con Vibe Coding

- Crear prototipos rapidos de funcionalidades
- Validar enfoques tecnicos
- Experimentar con diferentes soluciones

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Paso 2: Exploracion con Vibe Coding

- Crear prototipos rapidos de funcionalidades
- Validar enfoques tecnicos
- Experimentar con diferentes soluciones

Paso 3: Formalizar con Spec Kit

- Escribir PRODUCT_SPEC.md con funcionalidades requeridas
- Definir API_SPEC.md con endpoints y schemas
- Establecer criterios de aceptacion

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Paso 2: Exploracion con Vibe Coding

- Crear prototipos rapidos de funcionalidades
- Validar enfoques tecnicos
- Experimentar con diferentes soluciones

Paso 3: Formalizar con Spec Kit

- Escribir PRODUCT_SPEC.md con funcionalidades requeridas
- Definir API_SPEC.md con endpoints y schemas
- Establecer criterios de aceptacion

Paso 4: Preparar Agent Skills

- Crear skills para generacion de APIs
- Preparar templates de notificaciones
- Configurar prompts de revision

Workflow de Implementacion Integrada

Paso 1: Configurar AGENTS.md

- Definir stack tecnologico y convenciones
- Especificar estructura del proyecto
- Documentar comandos disponibles
- Establecer restricciones arquitectonicas

Paso 2: Exploracion con Vibe Coding

- Crear prototipos rapidos de funcionalidades
- Validar enfoques tecnicos
- Experimentar con diferentes soluciones

Paso 3: Formalizar con Spec Kit

- Escribir PRODUCT_SPEC.md con funcionalidades requeridas
- Definir API_SPEC.md con endpoints y schemas
- Establecer criterios de aceptacion

Paso 4: Preparar Agent Skills

- Crear skills para generacion de APIs
- Preparar templates de notificaciones
- Configurar prompts de revision

Paso 5: Ejecucion con Agente de IA

1. Agente lee AGENTS.md -> Entiende estructura y convenciones
2. Agente revisa prototipos de Vibe Coding -> Contexto de decisiones iniciales
3. Agente consulta spec/API_SPEC.md -> Conoce que APIs construir

Guia de Seleccion de Herramienta

Escenario	Herramienta Recomendada
Necesito validar una idea rapidamente	Vibe Coding
Voy a convertir el prototipo en producto	Spec Kit
Nuevo desarrollador o agente se une al proyecto	AGENTS.md
Necesito automatizar una tarea repetitiva	Agent Skills
Proyecto de produccion con estandares altos	Spec Kit + AGENTS.md
Equipo grande con multiples contribuidores	AGENTS.md + Agent Skills
Proyecto completo y mantenible	Los cuatro juntos

Checklist de Implementacion

Pasos practicos para comenzar

5 minutos

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

- [ ] Crear AGENTS.md con contexto y convenciones del proyecto

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

- [ ] Crear AGENTS.md con contexto y convenciones del proyecto
- [ ] Usar Vibe Coding para prototipado inicial de ideas

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

- [ ] Crear AGENTS.md con contexto y convenciones del proyecto
- [ ] Usar Vibe Coding para prototipado inicial de ideas
- [ ] Documentar aprendizajes del prototipado

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

- [ ] Crear AGENTS.md con contexto y convenciones del proyecto
- [ ] Usar Vibe Coding para prototipado inicial de ideas
- [ ] Documentar aprendizajes del prototipado
- [ ] Crear estructura de spec/ con especificaciones basicas

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

- [ ] Crear AGENTS.md con contexto y convenciones del proyecto
- [ ] Usar Vibe Coding para prototipado inicial de ideas
- [ ] Documentar aprendizajes del prototipado
- [ ] Crear estructura de spec/ con especificaciones basicas
- [ ] Identificar skills necesarias para el dominio del proyecto

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

- [ ] Crear AGENTS.md con contexto y convenciones del proyecto
- [ ] Usar Vibe Coding para prototipado inicial de ideas
- [ ] Documentar aprendizajes del prototipado
- [ ] Crear estructura de spec/ con especificaciones basicas
- [ ] Identificar skills necesarias para el dominio del proyecto
- [ ] Crear o importar skills relevantes

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

- [ ] Crear AGENTS.md con contexto y convenciones del proyecto
- [ ] Usar Vibe Coding para prototipado inicial de ideas
- [ ] Documentar aprendizajes del prototipado
- [ ] Crear estructura de spec/ con especificaciones basicas
- [ ] Identificar skills necesarias para el dominio del proyecto
- [ ] Crear o importar skills relevantes
- [ ] Verificar que el agente lee todos los archivos de contexto

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

- [ ] Crear AGENTS.md con contexto y convenciones del proyecto
- [ ] Usar Vibe Coding para prototipado inicial de ideas
- [ ] Documentar aprendizajes del prototipado
- [ ] Crear estructura de spec/ con especificaciones basicas
- [ ] Identificar skills necesarias para el dominio del proyecto
- [ ] Crear o importar skills relevantes
- [ ] Verificar que el agente lee todos los archivos de contexto

Configuracion inicial recomendada:

1. **AGENTS.md**: Definir stack, estructura, comandos y boundaries
2. **spec/CONSTITUTION.md**: Establecer principios de seguridad, rendimiento y cumplimiento
3. **spec/PRODUCT_SPEC.md**: Documentar funcionalidades requeridas
4. **skills/**: Crear o importar skills especificas del dominio

Checklist: Inicio de Proyecto con los Cuatro Enfoques

Fase de Fundacion:

- [ ] Crear AGENTS.md con contexto y convenciones del proyecto
- [ ] Usar Vibe Coding para prototipado inicial de ideas
- [ ] Documentar aprendizajes del prototipado
- [ ] Crear estructura de spec/ con especificaciones basicas
- [ ] Identificar skills necesarias para el dominio del proyecto
- [ ] Crear o importar skills relevantes
- [ ] Verificar que el agente lee todos los archivos de contexto

Configuracion inicial recomendada:

1. **AGENTS.md**: Definir stack, estructura, comandos y boundaries
2. **spec/CONSTITUTION.md**: Establecer principios de seguridad, rendimiento y cumplimiento
3. **spec/PRODUCT_SPEC.md**: Documentar funcionalidades requeridas
4. **skills/**: Crear o importar skills especificas del dominio

Verificacion de setup:

- [ ] El agente lee AGENTS.md al inicio de la sesion
- [ ] Las especificaciones estan actualizadas y sincronizadas
- [ ] Las skills estan correctamente configuradas
- [ ] Los comandos documentados funcionan correctamente

Mejores Practicas por Enfoque

Mejores Practicas por Enfoque

Con Vibe Coding:

- Mantener prototipos separados del codigo de produccion
- Documentar decisiones tomadas durante prototipado
- Usar para exploracion, no para implementacion final
- Revisar y refactorizar codigo generado

Mejores Practicas por Enfoque

Con Vibe Coding:

- Mantener prototipos separados del codigo de produccion
- Documentar decisiones tomadas durante prototipado
- Usar para exploracion, no para implementacion final
- Revisar y refactorizar codigo generado

Con Spec Kit:

- Especificar antes de implementar
- Incluir criterios de aceptacion medibles
- Mantener specs sincronizadas con codigo
- Documentar decisiones de diseno

Mejores Practicas por Enfoque

Con Vibe Coding:

- Mantener prototipos separados del codigo de produccion
- Documentar decisiones tomadas durante prototipado
- Usar para exploracion, no para implementacion final
- Revisar y refactorizar codigo generado

Con Spec Kit:

- Especificar antes de implementar
- Incluir criterios de aceptacion medibles
- Mantener specs sincronizadas con codigo
- Documentar decisiones de diseno

Con AGENTS.md:

- Mantenerlo conciso y focalizado
- Usar revelacion progresiva de informacion
- Actualizar cuando cambien convenciones
- Incluir ejemplos de codigo correcto

Mejores Practicas por Enfoque

Con Vibe Coding:

- Mantener prototipos separados del codigo de produccion
- Documentar decisiones tomadas durante prototipado
- Usar para exploracion, no para implementacion final
- Revisar y refactorizar codigo generado

Con Spec Kit:

- Especificar antes de implementar
- Incluir criterios de aceptacion medibles
- Mantener specs sincronizadas con codigo
- Documentar decisiones de diseno

Con AGENTS.md:

- Mantenerlo conciso y focalizado
- Usar revelacion progresiva de informacion
- Actualizar cuando cambien convenciones
- Incluir ejemplos de codigo correcto

Con Agent Skills:

- Crear skills reutilizables entre proyectos
- Documentar cada skill claramente
- Versionar skills cuando evolucionen
- Compartir skills entre equipos similares

Recursos Adicionales

Continua tu aprendizaje

3 minutos

Recursos de Documentacion General

Recursos de Documentacion General

- **Prompt Engineering Guide**: https://www.promptingguide.ai/

Recursos de Documentacion General

- **Prompt Engineering Guide**: https://www.promptingguide.ai/
- **OpenAI Best Practices**: https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api

Recursos de Documentacion General

- **Prompt Engineering Guide**: https://www.promptingguide.ai/
- **OpenAI Best Practices**: https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api
- **LangChain/LangGraph**: https://www.langchain.com/

Recursos de Documentacion General

- **Prompt Engineering Guide**: https://www.promptingguide.ai/
- **OpenAI Best Practices**: https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api
- **LangChain/LangGraph**: https://www.langchain.com/
- **Model Context Protocol (MCP)**: https://modelcontextprotocol.io/

Recursos de Documentacion General

- **Prompt Engineering Guide**: https://www.promptingguide.ai/
- **OpenAI Best Practices**: https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api
- **LangChain/LangGraph**: https://www.langchain.com/
- **Model Context Protocol (MCP)**: https://modelcontextprotocol.io/

Recursos de aprendizaje estructurado:

- Microsoft Learn - Spec-Driven Development con GitHub Spec Kit (13 unidades)
- Cursos de GitHub Copilot para desarrolladores
- Documentacion oficial de Anthropic sobre Agent Skills

Recursos de Documentacion General

- **Prompt Engineering Guide**: https://www.promptingguide.ai/
- **OpenAI Best Practices**: https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api
- **LangChain/LangGraph**: https://www.langchain.com/
- **Model Context Protocol (MCP)**: https://modelcontextprotocol.io/

Recursos de aprendizaje estructurado:

- Microsoft Learn - Spec-Driven Development con GitHub Spec Kit (13 unidades)
- Cursos de GitHub Copilot para desarrolladores
- Documentacion oficial de Anthropic sobre Agent Skills

Libros recomendados:

- "AI-Assisted Programming" (O'Reilly, 2025)
- "Building AI-Powered Applications" (Manning, 2025)

Recursos de Comunidad

Recursos de Comunidad

- **r/PromptEngineering (Reddit)**: https://www.reddit.com/r/PromptEngineering/

Recursos de Comunidad

- **r/PromptEngineering (Reddit)**: https://www.reddit.com/r/PromptEngineering/
- **GitHub Copilot Community**: https://github.com/community/copilot

Recursos de Comunidad

- **r/PromptEngineering (Reddit)**: https://www.reddit.com/r/PromptEngineering/
- **GitHub Copilot Community**: https://github.com/community/copilot
- **Stack Overflow Developer Survey 2025** - Seccion AI

Recursos de Comunidad

- **r/PromptEngineering (Reddit)**: https://www.reddit.com/r/PromptEngineering/
- **GitHub Copilot Community**: https://github.com/community/copilot
- **Stack Overflow Developer Survey 2025** - Seccion AI
- **Conferencias**: MLOps World, GenAI Summit

Recursos de Comunidad

- **r/PromptEngineering (Reddit)**: https://www.reddit.com/r/PromptEngineering/
- **GitHub Copilot Community**: https://github.com/community/copilot
- **Stack Overflow Developer Survey 2025** - Seccion AI
- **Conferencias**: MLOps World, GenAI Summit

Comunidades en espanol:

- Reddit r/es/comments sobre programacion
- Grupos de Meetup sobre IA y desarrollo
- Discord servers de comunidades de desarrolladores

Recursos de Comunidad

- **r/PromptEngineering (Reddit)**: https://www.reddit.com/r/PromptEngineering/
- **GitHub Copilot Community**: https://github.com/community/copilot
- **Stack Overflow Developer Survey 2025** - Seccion AI
- **Conferencias**: MLOps World, GenAI Summit

Comunidades en espanol:

- Reddit r/es/comments sobre programacion
- Grupos de Meetup sobre IA y desarrollo
- Discord servers de comunidades de desarrolladores

Blogs y Newsletters recomendados:

- Plausible Futures Substack: https://plausiblefutures.substack.com/
- Simon Willison's Blog: https://simonwillison.net/
- GitHub Blog: https://github.blog/

Glosario de Terminos

Referencia rapida de conceptos clave

2 minutos

Glosario Rapido de Terminos

Termino	Definicion
Vibe Coding	Desarrollo usando lenguaje natural con IA, enfoque fluido e intuitivo
SDD (Spec-Driven Development)	Metodologia donde las especificaciones formales guian la generacion de codigo
Spec Kit	Toolkit de GitHub que implementa SDD con comandos estructurados
AGENTS.md	Archivo de documentacion que proporciona contexto a agentes de IA
Agent Skills	Capacidades modulares que extienden la funcionalidad de los agentes
Prompt	Instrucciones en lenguaje natural dadas a un modelo de IA
LLM	Large Language Model - Modelo de lenguaje de gran escala
MCP	Model Context Protocol - Protocolo para gestion de contexto
Frontload	Cargar toda la informacion necesaria al inicio del proceso
Brownfield	Proyecto que modifica o extiende codigo existente (vs. greenfield)
Agente	Sistema de IA que puede planificar, ejecutar y verificar tareas
Skill	Capacidad modular y reutilizable para agentes de IA

Terminos Adicionales

Termino	Definicion
Context Window	Cantidad maxima de texto que un LLM puede procesar
Token	Unidad basica de texto procesada por un LLM
System Prompt	Instrucciones base que definen el comportamiento del agente
User Prompt	Instrucciones especificas de una tarea particular
Code Generation	Generacion automatica de codigo por modelos de IA

Resumen y Proximos Pasos

Integrando los Cuatro Enfoques en Tu Flujo de Trabajo

Puntos Clave:

Integrando los Cuatro Enfoques en Tu Flujo de Trabajo

Puntos Clave:

- Vibe Coding para exploracion rapida de ideas

Integrando los Cuatro Enfoques en Tu Flujo de Trabajo

Puntos Clave:

- Vibe Coding para exploracion rapida de ideas
- Spec Kit para proyectos que requieren estructura

Integrando los Cuatro Enfoques en Tu Flujo de Trabajo

Puntos Clave:

- Vibe Coding para exploracion rapida de ideas
- Spec Kit para proyectos que requieren estructura
- AGENTS.md como base de contexto para cualquier proyecto

Integrando los Cuatro Enfoques en Tu Flujo de Trabajo

Puntos Clave:

- Vibe Coding para exploracion rapida de ideas
- Spec Kit para proyectos que requieren estructura
- AGENTS.md como base de contexto para cualquier proyecto
- Agent Skills para automatizacion de tareas especializadas

Integrando los Cuatro Enfoques en Tu Flujo de Trabajo

Puntos Clave:

- Vibe Coding para exploracion rapida de ideas
- Spec Kit para proyectos que requieren estructura
- AGENTS.md como base de contexto para cualquier proyecto
- Agent Skills para automatizacion de tareas especializadas

Llamada a la Accion:

- Comenzar con AGENTS.md en tu proximo proyecto
- Experimentar con Vibe Coding para prototipado
- Adoptar Spec Kit cuando la estructura sea necesaria
- Desarrollar Agent Skills para tu dominio especifico

Gracias

Introduccion Completada

Siguiente: Modulo 1 - Ingeniería de Prompts y Personalización