Flujo de Trabajo Paso a Paso para Vibe Coding

Este flujo de trabajo detalla el proceso práctico para crear productos digitales utilizando la metodología Vibe Coding, aprovechando herramientas de IA gratuitas con capacidad de iteración y deploy.

Fase 1: Preparación y Planificación

Paso 1: Definir el Producto Digital

- **Acción**: Describe claramente qué producto digital deseas crear (sitio web, landing page, aplicación, etc.)
- **Herramienta recomendada**: Documento de texto o nota para registrar la idea
- **Resultado esperado**: Descripción concisa del producto, su propósito y audiencia objetivo

Paso 2: Investigar Referencias y Ejemplos

- **Acción**: Busca productos similares para inspiración y análisis
- **Herramienta recomendada**: Navegador web
- **Resultado esperado**: Colección de ejemplos, características deseables y elementos a evitar

Paso 3: Seleccionar la Herramienta de IA Adecuada

- **Acción**: Elige la herramienta según tus necesidades específicas
- Para planificación completa: Manus, GenSpark
- Para máximas iteraciones: DeepSeek, Qwen
- Para desarrollo web especializado: Yourware, Lovable
- **Herramienta recomendada**: Consulta la tabla comparativa en este documento
- **Resultado esperado**: Herramienta seleccionada y cuenta creada (si es necesario)

Paso 4: Preparar los Requisitos Detallados

- **Acción**: Elabora una lista de funcionalidades, diseño y comportamientos esperados
- **Herramienta recomendada**: Documento de texto estructurado
- **Resultado esperado**: Lista detallada de requisitos técnicos y funcionales

Fase 2: Generación del Prototipo Inicial

Paso 5: Formular el Prompt Inicial

- **Acción**: Redacta una descripción clara y detallada para la IA
- **Estructura recomendada**: ``` Quiero crear [tipo de producto] que [propósito principal].

Funcionalidades clave: 1. [Funcionalidad 1] 2. [Funcionalidad 2] 3. [Funcionalidad 3]

Estilo visual: - [Descripción del estilo] - [Referencias o ejemplos]

Audiencia objetivo: - [Descripción de usuarios]

Tecnologías preferidas (opcional): - [Lenguajes o frameworks] ``` - **Resultado esperado**: Prompt estructurado y completo

Paso 6: Generar el Prototipo Base

- **Acción**: Envía el prompt a la herramienta de IA seleccionada
- **Consideraciones**:
- Si usas un agente (Manus, GenSpark): Deja que planifique y genere el prototipo
- Si usas un LLM (DeepSeek, Qwen): Solicita explícitamente la estructura y código
- **Resultado esperado**: Código inicial o estructura del producto

Paso 7: Revisar y Analizar el Resultado

- **Acción**: Examina el prototipo generado para identificar aciertos y áreas de mejora
- **Aspectos a evaluar**:
- Funcionalidad básica
- Estructura del código
- Diseño visual
- Alineación con requisitos
- **Resultado esperado**: Lista de aspectos a mejorar o modificar

Fase 3: Refinamiento Iterativo

Paso 8: Iterar con Instrucciones Específicas

- **Acción**: Proporciona feedback claro y solicita mejoras concretas

- **Estructura recomendada**: ``` El prototipo se ve bien, pero necesito los siguientes cambios:
- [Cambio específico 1]: [Descripción detallada]
- [Cambio específico 2]: [Descripción detallada]
- [Cambio específico 3]: [Descripción detallada]

Mantén [elementos que funcionan bien]. ``` - **Resultado esperado**: Versión mejorada del prototipo

Paso 9: Probar Funcionalidades

- **Acción**: Verifica que las funcionalidades clave operen correctamente
- **Método**:
- Para agentes con entorno integrado: Prueba directamente en la plataforma
- Para código generado: Implementa localmente o en un entorno de pruebas
- **Resultado esperado**: Lista de funcionalidades verificadas y errores detectados

Paso 10: Refinar Diseño y Experiencia de Usuario

- **Acción**: Solicita mejoras específicas en la interfaz y experiencia
- **Estructura recomendada**: ``` Ahora necesito mejorar el diseño:
- [Elemento visual 1]: [Cambio deseado]
- [Elemento visual 2]: [Cambio deseado]
- [Experiencia de usuario]: [Mejora específica] ```
- **Resultado esperado**: Prototipo con diseño y experiencia mejorados

Paso 11: Optimizar para Diferentes Dispositivos

- **Acción**: Solicita adaptaciones para móviles, tablets y escritorio
- **Estructura recomendada**: ``` Necesito que el diseño sea responsive:
- En móviles: [Comportamiento esperado]
- En tablets: [Comportamiento esperado]
- En escritorio: [Comportamiento esperado] ```
- **Resultado esperado**: Prototipo responsive adaptado a múltiples dispositivos

Fase 4: Preparación para Deploy

Paso 12: Revisar y Corregir Errores

- **Acción**: Identifica y solicita corrección de bugs o problemas técnicos
- **Estructura recomendada**: ``` He encontrado los siguientes problemas:
- [Problema 1]: [Descripción y comportamiento esperado]
- [Problema 2]: [Descripción y comportamiento esperado] ```
- **Resultado esperado**: Prototipo con errores corregidos

Paso 13: Optimizar Rendimiento

- **Acción**: Solicita mejoras de rendimiento y optimización
- **Aspectos a considerar**:
- Tiempo de carga
- Tamaño de archivos
- Eficiencia del código
- **Resultado esperado**: Prototipo optimizado para mejor rendimiento

Paso 14: Preparar Archivos para Deploy

- **Acción**: Organiza y prepara todos los archivos necesarios
- **Consideraciones**:
- Para agentes con deploy integrado: Sigue las instrucciones de la plataforma
- Para código generado: Organiza en estructura de carpetas adecuada
- **Resultado esperado**: Proyecto listo para deploy

Fase 5: Deploy y Validación

Paso 15: Realizar el Deploy

- **Acción**: Despliega el producto en un entorno accesible
- **Opciones**:
- Usando capacidades de deploy de la herramienta (Manus, Replit)
- Mediante plataformas gratuitas (GitHub Pages, Netlify, Vercel)
- **Resultado esperado**: Producto accesible mediante URL pública

Paso 16: Probar en Entorno Real

- **Acción**: Verifica todas las funcionalidades en el entorno de producción
- **Aspectos a probar**:
- Navegación completa
- Funcionalidades interactivas
- Formularios y entradas de datos
- Visualización en diferentes dispositivos
- **Resultado esperado**: Lista de verificación completada

Paso 17: Iterar Post-Deploy

- **Acción**: Realiza ajustes finales basados en pruebas reales
- **Estructura recomendada**: ``` Después de probar el producto desplegado, necesito estos ajustes finales:
-
- ```
- **Resultado esperado**: Versión final optimizada y funcional

Fase 6: Documentación y Entrega

Paso 18: Documentar el Producto

- **Acción**: Solicita a la IA que genere documentación del producto
- **Elementos a incluir**:
- Descripción general
- Funcionalidades implementadas
- Tecnologías utilizadas
- Instrucciones de uso
- **Resultado esperado**: Documentación completa del producto

Paso 19: Exportar y Respaldar

- **Acción**: Guarda copias de todo el código y recursos
- **Métodos**:
- Descarga directa desde la plataforma
- Repositorio Git (GitHub, GitLab)
- Archivo comprimido local
- **Resultado esperado**: Respaldo completo del proyecto

Paso 20: Planificar Evolución Futura

- **Acción**: Define próximos pasos y mejoras potenciales
- **Elementos a considerar**:
- Funcionalidades adicionales
- Optimizaciones técnicas
- Escalabilidad
- **Resultado esperado**: Plan de evolución documentado

Consejos para Maximizar Resultados

Para Agentes de IA (Manus, GenSpark, Replit)

- Proporciona contexto completo desde el inicio
- Permite que el agente planifique antes de solicitar cambios específicos
- Aprovecha las capacidades de planificación autónoma
- Revisa y valida cada etapa del proceso

Para LLMs con Muchas Iteraciones (DeepSeek, Qwen)

- Divide el proyecto en componentes manejables
- Solicita código completo para cada componente
- Itera componente por componente
- Integra manualmente las partes al final

Para Optimizar Iteraciones Gratuitas

- Prepara prompts detallados y específicos
- Agrupa cambios relacionados en una sola iteración
- Documenta cada versión para evitar repeticiones
- Prioriza cambios críticos sobre ajustes menores

Para Facilitar el Deploy

- Elige tecnologías compatibles con plataformas gratuitas
- Minimiza dependencias externas
- Utiliza CDNs para bibliotecas comunes
- Considera soluciones serverless para funcionalidades backend