# **Flujo de Trabajo Modular de Codificación con IA para SOUP Emprendimientos**

Este flujo de trabajo está diseñado para construir SOUP Emprendimientos de forma incremental, maximizando la eficiencia de la IA y permitiendo una revisión y depuración efectiva en cada paso.

## **Fases de Generación de Código**

Dividiremos el proyecto en **fases lógicas**, y dentro de cada fase, generaremos **pequeños grupos de archivos (módulos)** que se necesitan mutuamente para una funcionalidad específica.

### **Fase 0: Preparación del Entorno y Estructura Base**

* **Descripción:** Crear las carpetas principales del proyecto, configurar los archivos de entorno y las dependencias base.
* **Archivos a Generar (por lotes muy pequeños):**
  + backend/requirements.txt
  + backend/.env.example
  + backend/app/main.py (con configuración básica de FastAPI)
  + backend/app/core/config.py
  + backend/app/database.py (conexión a PostgreSQL/SQLAlchemy)
  + frontend/package.json (dependencias de React)
  + frontend/tailwind.config.js, frontend/postcss.config.js
  + frontend/src/index.js, frontend/src/App.js (router básico)
  + frontend/src/index.css (Tailwind base)

**Instrucción a la IA para esta fase:** "Genera la estructura de archivos inicial para un proyecto full-stack con FastAPI y React. Configura la conexión básica a PostgreSQL con SQLAlchemy en el backend, y el entorno de React con Tailwind CSS en el frontend. Incluye los archivos main.py, database.py, config.py, requirements.txt para el backend, y index.js, App.js, package.json, index.css, tailwind.config.js, postcss.config.js para el frontend."

### **Fase 1: Autenticación y Gestión de Perfil de Usuario**

* **Descripción:** La funcionalidad más básica y esencial: que un usuario pueda registrarse, iniciar sesión y ver/editar su perfil.
* **Archivos a Generar (por lotes):**
  + **Backend:**
    - backend/app/models.py (Solo la entidad Usuario inicialmente)
    - backend/app/schemas.py (Schemas para Usuario - UserCreate, UserResponse, UserLogin)
    - backend/app/auth.py (Lógica de hashing, JWT)
    - backend/app/dependencies.py (Dependencia get\_db, get\_current\_user)
    - backend/app/crud/user.py (CRUD básico para Usuario)
    - backend/app/routers/auth\_router.py (/register, /login)
    - backend/app/routers/user\_router.py (/me, /me/cv)
    - *Actualizar main.py para incluir los routers.*
  + **Frontend:**
    - frontend/src/api/authApi.js (Funciones para llamar a la API de auth)
    - frontend/src/components/Auth/LoginScreen.js, RegisterScreen.js (Componentes de formularios)
    - frontend/src/screens/ProfileScreen.js
    - frontend/src/App.js (Rutas para login, registro, perfil)
    - frontend/src/context/AuthContext.js (Contexto de React para manejar el estado de autenticación global)
    - frontend/src/hooks/useAuth.js (Hook para consumir el contexto)

**Instrucción a la IA para esta fase:** "Basándote en la estructura de proyecto existente, implementa la autenticación completa para SOUP Emprendimientos. Esto incluye: el modelo Usuario en models.py, esquemas Pydantic, lógica de JWT y hashing de contraseña en auth.py, funciones CRUD para usuario, y los routers de FastAPI para registro, login y perfil de usuario (/users/me). En el frontend React, crea las pantallas de Login, Registro y Perfil de Usuario, con formularios, manejo de estado y conexión a estos endpoints de API."

### **Fase 2: Gestión de Negocios/Servicios y Productos/Ofertas (Core del Emprendedor)**

* **Descripción:** Permitir que el emprendedor cree sus perfiles de negocio/freelancer y defina sus productos/servicios con la calculadora de precios.
* **Archivos a Generar (por lotes):**
  + **Backend:**
    - backend/app/models.py (Añadir Negocio, Producto, Insumo y sus relaciones)
    - backend/app/schemas.py (Añadir esquemas para Negocio, Producto, Insumo)
    - backend/app/crud/negocio.py, backend/app/crud/producto.py, backend/app/crud/insumo.py
    - backend/app/services/pricing\_service.py (Lógica de cálculo COGS, precio sugerido)
    - backend/app/routers/negocio\_router.py
    - backend/app/routers/producto\_router.py
    - backend/app/routers/calculadora\_precios.py
  + **Frontend:**
    - frontend/src/api/negocioApi.js, frontend/src/api/productosApi.js
    - frontend/src/screens/BusinessManagementScreen.js
    - frontend/src/screens/ProductServiceManagementScreen.js
    - frontend/src/components/Forms/ProductServiceForm.js (con la sección dinámica de insumos y la calculadora de precios).
    - frontend/src/screens/DashboardScreen.js (Para mostrar un resumen de negocios/productos).

**Instrucción a la IA para esta fase:** "Partiendo del código anterior, implementa la gestión de Negocios y Productos/Servicios. Esto incluye: añadir los modelos Negocio, Producto e Insumo y sus esquemas. Crea los endpoints CRUD para estos en FastAPI. Desarrolla el servicio de cálculo de precios (pricing\_service.py) y un endpoint /calculate\_price\_suggestion. En el frontend React, crea las pantallas para gestionar Negocios y Productos, incluyendo un formulario ProductServiceForm con un campo dinámico para Insumos y la calculadora de precios integrada."

### **Fase 3: Marketplace Público**

* **Descripción:** Construir la interfaz pública donde los clientes pueden buscar y ver los negocios y productos.
* **Archivos a Generar (por lotes):**
  + **Backend:**
    - backend/app/routers/marketplace\_router.py (Search, Perfil Público, Detalle de Producto Público)
    - backend/app/utils/geospatial\_utils.py (Funciones básicas de PostGIS, si se implementa ubicación).
  + **Frontend:**
    - frontend/src/api/marketplaceApi.js
    - frontend/src/screens/MarketplacePublicScreen.js (Barra de búsqueda, filtros, listado de tarjetas)
    - frontend/src/components/Common/ProductCard.js, BusinessCard.js (Componentes para las tarjetas)
    - frontend/src/screens/PublicProfileScreen.js
    - frontend/src/screens/PublicProductDetailScreen.js

**Instrucción a la IA para esta fase:** "Crea la interfaz del Marketplace Público de SOUP Emprendimientos. En el backend, implementa los endpoints de FastAPI para búsqueda (/marketplace/search) con filtros de texto, rubro y ubicación, y los endpoints para perfiles públicos de negocios y detalles de productos. En el frontend React, genera la pantalla principal del Marketplace con búsqueda y filtros, así como los componentes de tarjetas para negocios y productos, y las pantallas de detalle público."

### **Fases Siguientes:**

* **Fase 4: Clientes, Proveedores y Mensajería.**
* **Fase 5: Gestión de Encargos.**
* **Fase 6: Informes y Publicidad.**
* **Fase 7: Integración Real de la IA (API de Gemini).**

### **Beneficios de Este Flujo de Trabajo**

* **Manejable:** Cada instrucción a la IA es un "chunk" de código razonable.
* **Depurable:** Si algo falla en la Fase 1, es más fácil encontrar el error antes de pasar a la Fase 2.
* **Feedback Constante:** Puedes ver resultados funcionales en cada fase.
* **Optimización de IA:** La IA se mantiene dentro de su ventana de contexto y puede "aprender" de lo que ya generó en los pasos anteriores para mantener la coherencia.
* **SRP Integrado:** La división de archivos ya está predefinida, así que la IA seguirá esa estructura al generar el código.

Estoy listo para empezar a generar el código para la **Fase 0** cuando me indiques.