1. Define tu idea y objetivos:
   * ¿Cuál es el propósito principal de tu aplicación?
   * ¿Qué problema resuelve o qué necesidad satisface?
   * ¿Cuál es tu público objetivo?
2. Esboza las funcionalidades principales:
   * Haz una lista de las funciones esenciales que debe tener la aplicación.
   * Prioriza estas funciones en "must-have" (imprescindibles) y "nice-to-have" (deseables).
3. Realiza un esquema o diagrama:
   * Crea un esquema simple o wireframes de las pantallas principales.
   * Puedes usar papel, herramientas digitales como Figma, Figma, Sketch, Adobe XD, o incluso herramientas gratuitas como Pencil.
4. Planifica la estructura técnica:
   * Decide qué tecnologías usarás (lenguaje de programación, framework, base de datos, etc.).
   * Evalúa si necesitas un backend (servidor) o si será una aplicación solo del lado del cliente.
5. Diseña un plan de desarrollo:
   * Divide el trabajo en etapas o sprints.
   * Establece metas concretas para cada etapa.
6. Busca recursos:
   * Tutoriales, cursos, documentación oficial.
   * Comunidades en línea donde puedas hacer preguntas.
7. Comienza a construir:
   * Empieza con un prototipo básico o "mvp" (producto mínimo viable).
   * Prueba y ajusta según sea necesario.
8. Evalúa y mejora:
   * Pide feedback a potenciales usuarios.
   * Itera en base a los comentarios y errores detectados.

**1. Objetivo principal**

Crear una aplicación sencilla y fácil de usar para que los dueños de la tienda puedan registrar, consultar y gestionar las deudas con clientes y proveedores, reemplazando las notas adhesivas y evitando pérdidas de información.

**2. Funcionalidades clave**

* **Registro de deudas con clientes**: nombre, monto, fecha, estado (pagado/no pagado).
* **Registro de facturas con proveedores**: nombre del proveedor, monto, fecha, estado.
* **Listado y búsqueda**: poder buscar por cliente o proveedor, filtrar por estado o fecha.
* **Edición y eliminación**: modificar o eliminar registros.
* **Resumen o reporte**: total de deudas pendientes, pagos realizados, etc.
* **Seguridad básica** (opcional): autenticación simple si quieres que solo los dueños puedan acceder.

**3. Diseño y estructura visual**

* Pantalla principal con un menú para acceder a:
  + Deudas de clientes
  + Facturas de proveedores
  + Resumen general
* Formularios para agregar nuevas deudas y facturas.
* Listados con las entradas existentes, con botones para editar o eliminar.
* Opcional: filtros y búsquedas rápidas.

**4. Tecnologías recomendadas**

* **Frontend**: HTML, CSS, JavaScript (puedes usar frameworks como React o Vue si quieres algo más avanzado, pero para comenzar algo simple funciona muy bien).
* **Backend**: Node.js con Express, Python con Flask o Django, o incluso una solución sin backend si decides usar solo almacenamiento local (por ejemplo, LocalStorage o una base de datos sencilla).
* **Base de datos**: SQLite, MySQL, o incluso un archivo JSON si quieres algo muy simple para comenzar.
* **Opcional**: alguna plataforma de hosting gratuita para desplegar, como Vercel, Netlify, o Heroku.

**5. Plan de desarrollo**

* **Fase 1**: Crear las pantallas y formularios para ingresar deudas y facturas.
* **Fase 2**: Implementar la lista y búsqueda de registros.
* **Fase 3**: Agregar funciones de editar y eliminar.
* **Fase 4**: Crear reportes o resúmenes.
* **Fase 5**: Añadir protección con contraseña si quieres.