

Documentación Código de barras.

Etapa 0 (Previa): Configuración del entorno.....	2
Etapa 1:.....	2
Etapa 2:.....	2
Etapa 2.5 (Entre interfaz y BD): Gestión de errores y validación.....	2
Etapa 3:.....	2
Etapa 4: Persistencia avanzada + Backups.....	2
Etapa 5: Pruebas y documentación.....	2

Etapa 0 (Previa): Configuración del entorno

Objetivo: Garantizar que todas las dependencias funcionen correctamente desde el inicio.

Contenido:

- Instalación de pyzbar (wrapper de ZBar para lectura de códigos) + opencv-python
- Configuración de drivers de cámara (evitar fallos en tiempo real)
- Script de verificación rápida (test_cam.py) para confirmar acceso a la webcam

Etapa 1:

Objetivo: Leer código de barras 1D

Medios:

- Programación en: Python
- Visión artificial (Cv2 “OpenCV”)

Cuando voy a estar satisfecho: Cuando muestre exitosamente el código

Etapa 2:

Que me de una interfaz bonita donde pueda ver los códigos registrados, cuando detecte un código nuevo, deberá salir un rectángulo que nos avise de un producto nuevo y si queremos añadirlo a la lista. También se debe poder eliminar los códigos registrados.

Etapa 2.5 (Entre interfaz y BD): Gestión de errores y validación

Objetivo: Evitar comportamientos inesperados y garantizar integridad de datos.

Contenido:

- Validación de códigos duplicados antes de insertar en BD
- Manejo de falsos positivos (códigos leídos parcialmente por baja calidad de imagen)
- Feedback visual/auditivo al registrar (ej.: sonido + cambio de color del rectángulo)
- Protección contra inyección SQL (usar parámetros parametrizados en SQLite)

Etapa 3:

Debe de estar en una base de datos en SQLite

Etapa 4: Persistencia avanzada + Backups

Objetivo: Cumplir con buenas prácticas de seguridad de datos (¡importante para tu perfil!).

Contenido:

- Sistema de backups automáticos diarios/semanales de la BD SQLite
- Exportación/importación de códigos a CSV/JSON (útil para migrar datos)
- Registro de metadatos: fecha/hora de escaneo, contador de lecturas por código

Etapa 5: Pruebas y documentación

Objetivo: Entregar un proyecto listo para evaluación con evidencia de calidad.

Contenido:

- Script de pruebas automatizadas (test_barcode_reader.py) con códigos de ejemplo
- Documentación en README.md:
- Instrucciones de instalación paso a paso
- Capturas de pantalla de la interfaz funcionando
- Lista de códigos de barras de prueba usados (EAN-13, Code 128, etc.)

Historias:

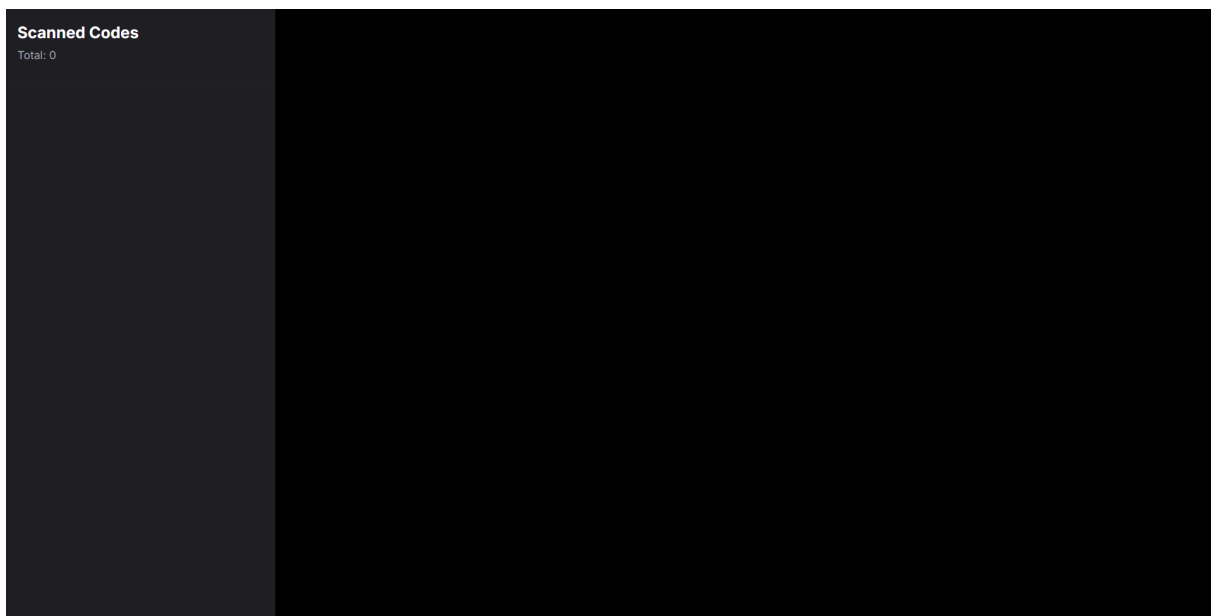
Me he quedado en la etapa 2.5, antes de empezar la 3.

Me ha dado un problema al iniciar el Start.bat, creo que ha habido un fallo con la cámara.

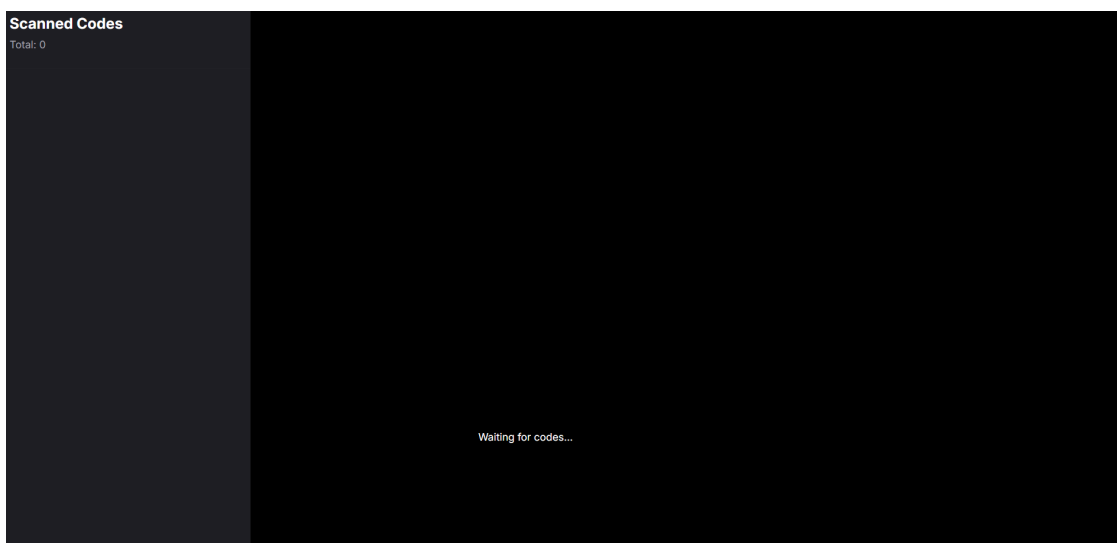
Iván - 3/2/2026 20:33

He conseguido que empiece a “hacer cosas” la cmd, ahora no podía coger los frames.

Iván - 3/2/2026 20:39



He conseguido cambiar de online a local. Ya se puede ver algo.



Ahora esta esperando códigos de barras.

Iván - 3/2/2026 20:51

Ahora ya no detecta 2 veces el código por error, solo lo escanea una sola vez

Iván - 3/2/2026 20:55

Fin de sesión, me he quedado por la etapa 2.5 para comenzar la etapa 3.





Iván - 3/2/2026 20:57

```
[DEBUG] Detected 1 codes
[INFO: 127.0.0.1:51478 - "GET /api/latest_codes HTTP/1.1" 200 OK
[DEBUG] Detected 1 codes
[DEBUG] Detected 1 codes
[DEBUG] Detected 1 codes
[DEBUG] Detected 1 codes
[DEBUG] Detected 1 codes
[DEBUG] Detected 1 codes
[DEBUG] Detected 1 codes
```

Códigos detectados correctamente

Iván - 3/2/2026 -21:23

Etapa 4 completada, ahora aunque cierre el servidor los datos se mantienen guardados en una base de datos SQLite, los datos guardados estarán en barcodes.db

mozzarella EAN13 • 8480000511263 • 22:46:56	
Burger Brioché EAN13 • 8402001013843 • 22:46:29	
Chips Paprika EAN13 • 5414359900273 • 22:46:07	
Jojitos EAN13 • 8410573020015 • 22:45:25	

Productos escaneados correctamente con el número de barras y también con la identificación del producto escaneado.

Iván - 3/2/2026 22:48

Es normal que algunos productos locales o específicos (como los de Juguettes o marcas blancas de supermercado) no estén en las bases de datos mundiales de internet.

Para solucionar esto, he añadido una función de **Identificación Manual**:

1. **Nombre Editable:** Ahora, cuando detecta un código, el nombre del producto aparece en un cuadro de texto que puedes editar.
2. **Entrada Rápida:** Si el sistema no reconoce el objeto, el cursor se pondrá automáticamente en el cuadro de nombre para que puedas escribirlo directamente.
3. **Confirmación con Enter:** Puedes escribir el nombre y pulsar **Enter** para añadirlo a la lista rápidamente.

Dada la incidencia de algunos objetos sin registrar en la base de datos, hemos añadido una versión para añadirlo nosotros automáticamente.

Iván - 3/2/2026 23:01

Finalizada la sesión.

Iván - 3/2/2026 23:01

El proyecto se ha ejecutado con éxito, objetivo completado.

Iván - 3/2/2026 23:16