

Parte 4

Arquitectura CNN

```
(techx) C:\mini_proyectos_ia\Parte4>python summary.py
```

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 26, 26]	320
ReLU-2	[-1, 32, 26, 26]	0
MaxPool2d-3	[-1, 32, 13, 13]	0
Conv2d-4	[-1, 64, 11, 11]	18,496
ReLU-5	[-1, 64, 11, 11]	0
MaxPool2d-6	[-1, 64, 5, 5]	0
Flatten-7	[-1, 1600]	0
Linear-8	[-1, 128]	204,928
ReLU-9	[-1, 128]	0
Linear-10	[-1, 10]	1,290

```
=====  
Total params: 225,034  
Trainable params: 225,034  
Non-trainable params: 0  
=====  
Input size (MB): 0.00  
Forward/backward pass size (MB): 0.52  
Params size (MB): 0.86  
Estimated Total Size (MB): 1.38  
=====
```

Arquitectura CNN

Capas convolucionales

- Primera capa Conv2d con 32 filtros de tamaño 3x3, stride 1.
- Función de activación ReLU.
- MaxPooling 2x2 para reducir resolución espacial.
- Segunda capa Conv2d con 64 filtros de tamaño 3x3.
- ReLU.
- MaxPooling 2x2 nuevamente.

Capas totalmente conectadas (fully connected)

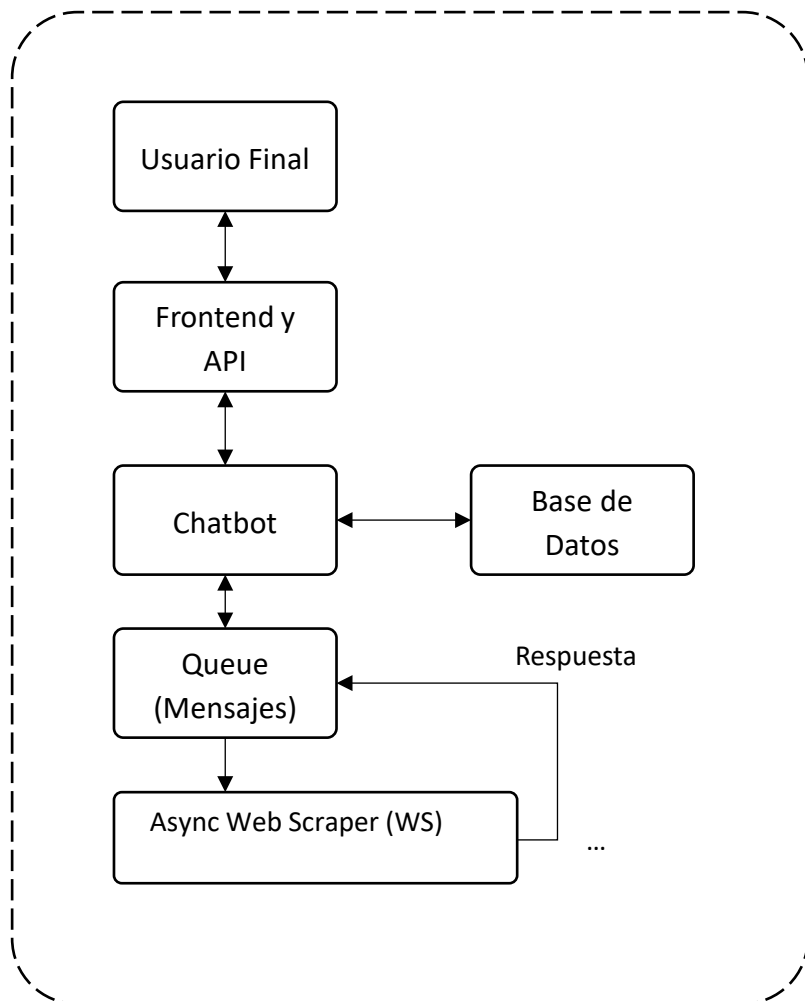
- Aplanamiento de la salida convolucional (64 canales \times 5 \times 5 píxeles).
- Capa densa de 128 neuronas con ReLU.
- Capa final de 10 neuronas (para clasificación en 10 clases).

Flujo

La imagen (1 canal) pasa por las convoluciones + pooling para extraer características espaciales, luego se aplanan y pasan a las capas densas para clasificación.

Parte 5

Caso 1



- Usuario hace una consulta (ej. ¿Qué precio tiene el producto X?)
- Chatbot consulta base de datos:
 - Si el producto existe → responde al instante
 - Si no existe
 - Encola un trabajo de scraping (no espera)
 - Responde algo como: “Estoy buscando el producto. Intenta en unos segundos.”
- Scraper (worker) consume la cola:
- Extrae producto
- Inserta o actualiza en base de datos
- Dashboard se refresca automáticamente o bajo demanda.

Componente	Rol
Redis / Base de datos (PostgreSQL, SQLite, Mongo)	Fuente común de datos para chatbot y dashboard. Redis se destaca por su alta velocidad y flexibilidad, almacenando los datos en memoria RAM en lugar de en disco, lo que permite operaciones rápidas de lectura y escritura.
Scraper Worker	Se ejecuta de forma asíncrona, independiente del cliente
Queue (RabbitMQ / Redis Queue / Celery)	Almacena trabajos de scraping sin bloquear al frontend
Chatbot y dashboard	Hacen solo consultas rápidas al backend
API Layer	Expone endpoints para dashboard, chatbot y nuevos productos

Caso 2

OCR (Pytesseract, Tesseract)

Posibles problemas:

- No está optimizado para layouts de recibos mexicanos u otros documentos ruidosos.
- Falla en detectar campos numéricos (totales) y patrones específicos (fechas).
- Imágenes borrosas, inclinadas o de baja calidad
- Sombras, oclusiones, bajos contrastes, brillo excesivo en las fotografías o documentos escaneados
- El sistema **no guía** al usuario para subir imágenes legibles.

Solución

Paso 1 (Mejoras suaves, auditoría del sistema actual):

- Extraer y analizar ejemplos con fallos
- Categorizar si el fallo es de la imagen o del OCR
- Generar un logging detallado del output del OCR y errores en los campos fecha y total, entre otros campos relevantes

Paso 2 (Mejoras técnicas):

Preprocesamiento de imágenes

- Binarización, denoise, reescalado, corrección de rotación (OpenCV)
- Conversión uniforme a escala de grises

Zonas específicas / detección de campos

- Extraer solo el área de fecha y de total (si es posible)
- Evitar pasar el recibo completo a OCR si tiene ruido

Post-procesamiento con reglas

- Usar regex para validar fecha (dd/mm/yyyy, etc) y total
- Si el OCR no detecta correctamente, marcar como “requiere revisión”

d. Considerar upgrade a OCR más robusto

Comunicación con stake holders

Estado actual:

El 35% de los OCR fallan por múltiples causas: calidad de imagen, errores del motor y falta de validación.

Acciones a tomar:

Auditar errores y clasificar las causas

Aplicar mejoras en preprocesamiento, realizar validaciones adicionales

Evaluar otros modelos OCR

Objetivos a corto plazo:

Reducir errores críticos

Establecer un umbral de confianza y revisión manual de ser posible

Explorar otros OCRs

Objetivos a largo plazo:

Aplicar filtros más avanzados a las imágenes

Desarrollar un OCR propio