

Chat Web con Ollama — Resumen Técnico y Estilización

1. Objetivo del proyecto

Crear una aplicación web de chat que:

- Se conecte directamente con un modelo LLM local (Ollama).
 - Use autenticación basada en áreas (no usuarios individuales).
 - Permita conversaciones por área.
 - Tenga un frontend estético y moderno con una experiencia fluida.
 - Esté preparada para expandirse (modo oscuro, RAG, persistencia).
-

2. Backend – Express + Ollama

Componentes clave:

- **API /api/login:** Valida área y contraseña con bcrypt.
- **API /api/chat:** Recibe prompt, lo envía a Ollama, retorna respuesta.
- **Configuración (area_config.json):**

```
{  
  "informatica": {  
    "password_hash": "<hash>",  
    "agent_config": {  
      "model": "llama3.2:3b",  
      "system_prompt": "...",  
      "temperature": 0.5,  
      "max_tokens": 2000  
    }  
  }  
}
```

- **Variables de entorno (.env):**

OLLAMA_BASE_URL=http://172.19.5.212:11434

3. Ollama con Docker

- Docker Compose en servidor Ubuntu (172.19.5.212).
 - Estructura de carpetas:
 - /opt/ollama/
 - └─ docker-compose.yml
 - └─ .env
 - └─ data/
 - Puerto expuesto: 11434
 - Modelo cargado exitosamente: llama3.2:3b
 - Prueba desde PC local con Invoke-RequestMethod funcionó correctamente.
-

4. Frontend (React + Tailwind CSS)

Estructura y funcionalidades:

- Estados gestionados con useState para controlar área, contraseña, chat, loading, modo oscuro.
- Login funcional, luego chat con mensajes del usuario y respuestas del bot.

Mejoras visuales aplicadas:

- Diseño centralizado, bordes redondeados, sombras, transiciones estilizadas.
 - Estilo de chat con burbujas diferenciadas (usuario – derecha / bot – izquierda).
 - Indicador “Pensando...” animado.
 - Scroll automático usando useRef y useEffect para mover ventana al fondo tras cada mensaje.
 - Dark mode parcialmente (toggle activado, pero estilo aún refinable).
-

5. Consideraciones importantes

- Almacenamiento de contraseñas: bcrypt para seguridad.
 - Verificar que Ollama esté cargado y disponible en red.
 - Evitar configuración errónea de Tailwind o falta de importación de CSS.
 - Implementar medidas de seguridad como rate limiting.
 - UI responsive y preparada para activar modo oscuro.
-

6. Próximos pasos sugeridos

1. Finalizar modo oscuro completo.
2. Mejoras UI adicionales: avatares, timestamps, botón para limpiar chat.
3. Integración RAG con Qdrant.
4. Dockerizar todo el stack completo.
5. Agregar logging y persistencia del historial.
6. Preparar entorno para producción (HTTPS, seguridad, optimización).