Proyecto: Web App Matemática Multifuncional

**Objetivo general:**  
Crear una web app que integre calculadora básica, científica y gráfica, con un workspace para guardar trabajos completos, un sistema de chat/compartición de problemas, soporte opcional de IA y funcionalidad de login.

**1. Calculadora**

* **Básica:** operaciones aritméticas simples (+, -, \*, /), porcentaje, cambio de signo.
* **Científica:** funciones trigonométricas, logaritmos, raíces, potencias, exponenciales, factorial.
* **Gráfica:** capacidad de graficar funciones matemáticas y mostrar resultados visuales.
* **Workspace:**
  + Espacio separado del historial y la memoria.
  + Guardar trabajos completos, incluyendo la expresión, resultado, pasos intermedios y gráficas.
  + Editar y modificar trabajos existentes.
  + Compartir trabajos a la zona de chat.
* **Persistencia:**
  + Guardar datos en **IndexedDB** para mantener información compleja de forma local.
  + No usar variables globales innecesarias.

**2. Función de captura de ecuaciones con cámara**

* Capacidad de capturar texto o ecuaciones desde una hoja usando la cámara.
* Detectar y extraer ecuaciones matemáticas.
* Resolver paso a paso mostrando el procedimiento.
* Permitir al usuario integrar la IA de su preferencia para el análisis o resolución.
* No depender de una IA integrada por defecto.

**3. Zona de compartición / chat**

* Espacio tipo red social (similar a Twitter) limitado a la app web.
* Permitir publicar problemas matemáticos y discusiones.
* Integración con el workspace para compartir trabajos completos.
* Capacidad de login opcional:
  + Usuarios registrados pueden publicar, guardar y gestionar trabajos.
  + Usuarios sin cuenta pueden navegar y usar funciones limitadas.

**4. Login y gestión de usuarios**

* Registro y autenticación básica.
* Área limitada para usuarios no registrados.
* Opcional: persistencia de sesiones y perfiles.

**5. UI / UX**

* **Responsive:** adaptarse a diferentes tamaños de pantalla, incluyendo ajuste al abrir la consola.
* **Colores y estilos:** uso de variables CSS para gestionar paleta de colores.
* Workspace, historial y memoria deben ser visualmente distintos y funcionalmente independientes.
* Botones claramente diferenciados (digit, operation, memory, etc.).
* Scrolls independientes para memoria y historial.

**6. Tecnologías recomendadas**

* **Frontend:** React.js para modularidad y componentes reutilizables.
* **Persistencia local:** IndexedDB para workspace, historial y memoria.
* **Opcional:** TypeScript para mayor seguridad y escalabilidad.
* **Chat en tiempo real:** Firebase o Socket.io.
* **Gráficas:** Chart.js, D3.js o similares.

**7. Características de calidad y mantenimiento**

* Código modular y limpio.
* Manejo adecuado de errores y validaciones (p. ej., división por cero, paréntesis mal colocados).
* Funciones independientes para cada módulo: calculadora, workspace, memoria, historial, chat.
* Persistencia y recuperación confiable de datos.
* Escalabilidad para futuras funciones y mejoras.

**Resumen:**  
El proyecto debe ser una web app completa de matemáticas con tres tipos de calculadora, workspace persistente, chat social interno, soporte opcional de IA, login opcional y una interfaz responsive y modular, usando React y considerando TypeScript a futuro.