

Agents

👤 Creador	👤 Facundo Nicolás González
⚙️ Estado	Borrador
☰ Categoría	
🕒 Última edición	@6 de octubre de 2025 2:04
🕒 Hora de creación	@6 de octubre de 2025 2:04

agents.md

1. Contexto del proyecto

Este proyecto es una aplicación web que... (describe brevemente el propósito del proyecto, la tecnología principal y su función).

El objetivo principal es... (por ejemplo, mejorar la usabilidad, añadir nuevas funcionalidades, optimizar el rendimiento).

2. Axiomas y principios del agente

2.1 Principios de diseño

- **Modularidad:** Cada componente o módulo debe tener una responsabilidad única y bien definida.
- **Simplicidad:** El código debe ser fácil de entender, mantener y depurar.
- **Coherencia:** Mantener la uniformidad en los estilos de codificación, las convenciones de nomenclatura y los patrones de diseño.

2.2 Directrices de codificación

- **Comentarios:** Añadir comentarios solo en secciones de código complejas o no obvias.
- **Pruebas:** Toda nueva funcionalidad debe ir acompañada de pruebas unitarias o de integración.

- **Control de versiones:** Usar mensajes de confirmación claros y descriptivos.

3. Roles y tareas del agente

3.1 Agente de mantenimiento

- **Función:** Asistir en la revisión del código, la refactorización y la solución de errores.
- **Tarea:** Refactorizar el componente X para mejorar la legibilidad .
 - **Contexto:** El componente X está obsoleto y se necesita actualizar.
 - **Criterio de éxito:** El componente X cumple con los principios de diseño y los nuevos estándares.

3.2 Agente de desarrollo de funciones

- **Función:** Implementar nuevas funcionalidades según las especificaciones del proyecto.
- **Tarea:** Implementar un nuevo endpoint para la gestión de usuarios .
 - **Contexto:** Crear un nuevo endpoint de API REST para gestionar usuarios en la base de datos.
 - **Criterio de éxito:** El endpoint está documentado y pasa todas las pruebas de integración.

4. Archivos y directorios a tener en cuenta

- `src/` : Directorio principal del código fuente.
- `docs/` : Documentación del proyecto.
- `tests/` : Archivos de pruebas unitarias y de integración.
- `README.md` : Descripción general del proyecto.
- `.github/workflows/` : Flujos de trabajo de CI/CD.

5. Registro de código (ejemplo de anotación para el agente)

Cada entrada incluye: anclaje de ID de marca de tiempo, tipo de cambio, alcance, hash de confirmación, justificación.

- `Timestamp: 2025-10-05T23:00:00Z`
- `Type: fix`
- `Scope: authentication`
- `Justification: Solucionado un error de autenticación en el componente de login.`

6. Barreras de protección y límites

- **Evitar cambios en el archivo** `package-lock.json` sin previa consulta, ya que podría afectar a las dependencias.
- **No modificar archivos del núcleo del framework** a menos que sea absolutamente necesario.
- **Consultar con un desarrollador** si el cambio propuesto implica una reestructuración significativa de la arquitectura.

7. Herramientas y tecnologías

- **Lenguaje de programación:** `JavaScript` , `TypeScript` .
- **Frameworks:** `React` , `Node.js` , `Express` .
- **Base de datos:** `PostgreSQL` .
- **Dependencias principales:** `lodash` , `axios` , `jest` .