

# Guía de Instalación Completa - Web Agent N8N

Esta guía te llevará paso a paso por la instalación y configuración completa del Web Agent N8N.

---

## Índice

1. [Requisitos del sistema](#)
  2. [Servicios necesarios](#)
  3. [Instalación de dependencias](#)
  4. [Configuración de credenciales](#)
  5. [Importación del workflow](#)
  6. [Configuración avanzada](#)
  7. [Verificación final](#)
- 

## Requisitos del sistema

### Software necesario

- **n8n:** Versión 1.0.0 o superior
- **Node.js:** Versión 18.x o superior (si instalas n8n manualmente)
- **Docker:** Versión 20.x o superior (recomendado)
- **Redis:** Versión 7.x o superior
- **Servidor web:** Nginx, Apache o Node.js (para servir la interfaz HTML)

### Hardware recomendado

- **CPU:** 2 cores mínimo
  - **RAM:** 4GB mínimo (8GB recomendado)
  - **Disco:** 20GB de espacio libre
  - **Red:** Conexión estable a internet
- 

## Servicios necesarios

### 1. n8n

Si aún no tienes n8n instalado:

### Opción A: Docker (Recomendado)

```
bash
```

```
docker run -it --rm \
  --name n8n \
  -p 5678:5678 \
  -v ~/.n8n:/home/node/.n8n \
  n8nio/n8n
```

## Opción B: NPM

```
bash
```

```
npm install n8n -g
n8n start
```

Accede a n8n en: <http://localhost:5678>

---

## 2. Redis

Redis se usa para mantener la memoria de las conversaciones.

### Con Docker:

```
bash
```

```
docker run -d \
  --name redis \
  -p 6379:6379 \
  -e REDIS_PASSWORD=tu_password_segura \
  redis:7 \
  redis-server --requirepass tu_password_segura
```

### Verifica que funciona:

```
bash
```

```
docker exec -it redis redis-cli -a tu_password_segura ping
```

Deberías ver: [PONG](#)

---

## 3. APIs de Inteligencia Artificial

Necesitas al menos una de estas:

### OpenAI:

1. Ve a <https://platform.openai.com>
2. Crea una cuenta o inicia sesión
3. Ve a **API Keys**
4. Crea una nueva API key
5. Guárdala de forma segura

#### **Modelos recomendados para este workflow:**

- `gpt-4.1` (para el agente principal)
- `gpt-4.1-mini` (para formateo de respuestas)

#### **Google AI (Gemini):**

1. Ve a <https://makersuite.google.com/app/apikey>
2. Crea un proyecto si no tienes uno
3. Genera una API key
4. Guárdala de forma segura

#### **Modelos disponibles:**

- `gemini-pro` (alternativa al agente principal)
- 

#### **4. Google Sheets (Opcional - para gestión de leads)**

Si vas a usar la función de registro de leads:

1. Ve a [Google Cloud Console](#)
  2. Crea un proyecto nuevo o selecciona uno existente
  3. Habilita la **Google Sheets API**
  4. Crea credenciales de **Service Account**
  5. Descarga el archivo JSON de credenciales
  6. Comparte tu Google Sheet con el email del service account
- 

#### **5. Servicios MCP (Model Context Protocol)**

Este workflow utiliza dos servicios MCP personalizados:

##### **MCP Calendario:**

- Gestiona la agenda de citas
- Requiere endpoint HTTP con autenticación por headers
- Funciones: Crear, leer, actualizar y eliminar eventos

### MCP Correo:

- Envía correos de confirmación automáticos
- Requiere endpoint HTTP con autenticación por headers
- Función: Envío de emails con plantillas

**Configuración:** Necesitarás los endpoints URL y las credenciales de autenticación para ambos servicios.

---

## Instalación de dependencias

### Paso 1: Verificar conectividad

Asegúrate de que todos los servicios están corriendo:

```
bash

# Verifica n8n
curl http://localhost:5678/healthz

# Verifica Redis
redis-cli -a tu_password ping

# Verifica que puedes acceder a las APIs de IA
curl https://api.openai.com/v1/models \
  -H "Authorization: Bearer tu_openai_api_key"
```

---

### Paso 2: Crear red Docker (Opcional)

Si usas Docker, crea una red para que los servicios se comuniquen:

```
bash
```

```
docker network create n8n-network
```

```
# Conecta Redis a la red
```

```
docker network connect n8n-network redis
```

```
# Inicia n8n en la misma red
```

```
docker run -d \
```

```
--name n8n \
```

```
--network n8n-network \
```

```
-p 5678:5678 \
```

```
-v ~/.n8n:/home/node/.n8n \
```

```
n8nio/n8n
```

---

## Configuración de credenciales

### Paso 1: Accede a n8n

1. Abre tu navegador en `http://localhost:5678`
2. Si es la primera vez, crea tu cuenta de administrador

---

### Paso 2: Configura Redis

1. En n8n, ve a **Settings** → **Credentials**
2. Haz clic en **Add Credential**
3. Busca **Redis**
4. Completa los datos:

```
Host: localhost (o redis si usas Docker network)
```

```
Port: 6379
```

```
Password: tu_password_segura
```

```
Database: 0
```

5. Haz clic en **Save** y prueba la conexión

---

### Paso 3: Configura OpenAI

1. En **Credentials**, haz clic en **Add Credential**
  2. Busca **OpenAI**
  3. Pega tu API Key
  4. Haz clic en **Save**
- 

#### **Paso 4: Configura Google Gemini (Opcional)**

1. En **Credentials**, haz clic en **Add Credential**
  2. Busca **Google PaLM API** o **Google AI**
  3. Pega tu API Key
  4. Haz clic en **Save**
- 

#### **Paso 5: Configura Google Sheets (Opcional)**

Si vas a usar el registro de leads:

1. En **Credentials**, haz clic en **Add Credential**
  2. Busca **Google Service Account**
  3. Pega el contenido del archivo JSON de tu service account
  4. Haz clic en **Save**
- 

#### **Paso 6: Configura MCP Services**

Para cada servicio MCP (Calendario y Correo):

1. En **Credentials**, haz clic en **Add Credential**
2. Busca **HTTP Header Auth**
3. Configura:

Name: Authorization  
Value: Bearer tu\_token\_mcp

4. Haz clic en **Save**
- 

## **Importación del workflow**

#### **Paso 1: Descargar el workflow**

Descarga el archivo `AGENTE_WEB_CLEAN_2_.json` del repositorio.

---

## Paso 2: Importar en n8n

1. En n8n, haz clic en el menú superior derecho
  2. Selecciona **Import from file**
  3. Selecciona el archivo JSON descargado
  4. Haz clic en **Import**
- 

## Paso 3: Configurar nodos

Una vez importado, necesitas configurar varios nodos:

### a) OpenAI Chat Model4 (Agente principal)

- Selecciona tu credencial de OpenAI
- Modelo:
- Temperature:
- Top P:

### b) Google Gemini Chat Model (Alternativo)

- Selecciona tu credencial de Google
- Temperature:
- Top P:

### c) OpenAI Chat Model1 (Formateo)

- Selecciona tu credencial de OpenAI
- Modelo:

### d) Redis Chat Memory

- Selecciona tu credencial de Redis
- Context Window Length:

### e) MCP Calendario1

- Endpoint URL:
- Selecciona tu credencial HTTP Header Auth

### f) MCP Correo1

- Endpoint URL:
- Selecciona tu credencial HTTP Header Auth

## g) AddLEADS (Google Sheets)

- Selecciona tu credencial de Google Service Account
  - Document ID: ID de tu Google Sheet
  - Sheet Name: Nombre de la hoja (ej: "Leads")
- 

### Paso 4: Activar el workflow

1. En la parte superior del workflow, cambia el toggle de **Inactive** a **Active**
  2. El workflow ahora está escuchando peticiones
- 

## Configuración avanzada

### Personalización del prompt del sistema

Edita el nodo **AI Agent1** para personalizar:

- Horarios de atención
- Servicios ofrecidos
- Ubicación física
- Tono de respuesta
- Reglas de negocio

El prompt del sistema está en la opción **System Message**.

---

### Ajuste de memoria conversacional

En el nodo **Redis Chat Memory**, puedes ajustar:

- `contextWindowLength`: Número de mensajes previos a recordar (recomendado: 10-20)
- 

### Configuración de herramientas

El agente tiene acceso a estas herramientas:

- **Calculator**: Operaciones matemáticas
- **Think**: Razonamiento interno
- **MCP Calendario**: Gestión de agenda
- **MCP Correo**: Envío de emails
- **AddLEADS**: Registro en Google Sheets

Puedes habilitar/deshabilitar conectando o desconectando los nodos.



# Integración con interfaz web

## Opción A: Webhook directo

El workflow expone un webhook que puedes usar desde tu interfaz HTML/JavaScript:

javascript

*// Ejemplo de llamada desde JavaScript*

```
async function sendMessage(message, sessionId) {
  const response = await fetch('http://localhost:5678/webhook/AGENTE_WEB_GITHUB', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      chatInput: message,
      sessionId: sessionId
    })
  });

  const data = await response.json();
  return data.message;
}
```

---

## Opción B: Chat embed n8n

n8n ofrece un widget de chat integrable:

1. En el workflow, haz clic en el botón **Chat** en la parte superior
2. Copia el código de embed generado
3. Pégallo en tu HTML

html

```
<script type="module">
  import { createChat } from 'https://cdn.jsdelivr.net/npm/@n8n/chat/dist/chat.bundle.es.js';

  createChat({
    webhookUrl: 'http://localhost:5678/webhook/AGENTE_WEB_GITHUB',
    initialMessages: ['¡Hola! ¿En qué puedo ayudarte?'],
    mode: 'fullscreen'
  });
</script>
```

# Verificación final

## Paso 1: Test básico

1. Ve a tu interfaz web o usa el chat de n8n
  2. Envía un mensaje de prueba: "Hola"
  3. Deberías recibir la presentación del agente
- 

## Paso 2: Test de memoria

1. Envía un mensaje con tu nombre: "Me llamo Juan"
  2. Envía otro mensaje: "¿Cómo me llamo?"
  3. El agente debería recordar tu nombre
- 

## Paso 3: Test de herramientas

### Test de calendario:

"Quiero agendar una cita para mañana a las 10:00"

El agente debería:

- Verificar disponibilidad
  - Solicitar tus datos (nombre, email, teléfono)
  - Crear la cita en el calendario
  - Enviar confirmación por email
  - Registrar el lead en Google Sheets
- 

## Paso 4: Revisar logs

En n8n, ve a **Executions** para ver el historial de ejecuciones y detectar posibles errores.

---

## Solución de problemas comunes

### Error: "Cannot connect to Redis"

#### Solución:

```
bash
```

```
# Verifica que Redis está corriendo
```

```
docker ps | grep redis
```

```
# Si no está corriendo, inícialo
```

```
docker start redis
```

```
# Verifica la conexión
```

```
redis-cli -a tu_password ping
```

---

## Error: "Invalid API key" (OpenAI)

### Solución:

1. Ve a <https://platform.openai.com/api-keys>
2. Verifica que tu API key es válida
3. Actualiza la credencial en n8n

---

## El agente no responde

### Verifica:

1. El workflow está **Active**
2. Todos los nodos tienen credenciales configuradas
3. No hay errores en **Executions**
4. Redis está funcionando correctamente

---

## Error en MCP Services

### Solución:

1. Verifica que los endpoints MCP están accesibles
2. Comprueba que la autenticación es correcta
3. Revisa los logs del servidor MCP

---

## Mantenimiento

### Backup periódico

Exporta el workflow regularmente:

1. En n8n, abre el workflow
  2. Menú → **Download**
  3. Guarda el archivo JSON en un lugar seguro
- 

## Actualización de n8n

```
bash

# Con Docker
docker pull n8nio/n8n:latest
docker stop n8n
docker rm n8n
# Vuelve a ejecutar el comando de instalación
```

---

## Monitoreo de Redis

```
bash

# Ver uso de memoria
docker exec -it redis redis-cli -a tu_password INFO memory

# Ver número de claves
docker exec -it redis redis-cli -a tu_password DBSIZE
```

---

## Soporte

Si encuentras problemas durante la instalación:

1. Revisa los logs de n8n en **Executions**
  2. Consulta la [documentación oficial de n8n](#)
  3. Verifica que todas las versiones de software son compatibles
- 

## Próximos pasos

Una vez completada la instalación:

1. Personaliza el prompt del sistema según tus necesidades
  2. Diseña tu interfaz web de chat
  3. Configura tus servicios MCP personalizados
  4. Ajusta las herramientas del agente
  5. Realiza pruebas exhaustivas antes de producción
- 

**¡Instalación completada!** Tu agente web está listo para conversar con tus usuarios.