GUÍA TÉCNICA · MACOS · LINUX · WINDOWS

Guía de instalación del ecosistema Gentle

Qué es cada pieza, qué necesitas antes de empezar, cómo instalarla paso a paso en tu sistema operativo y cómo comprobar que todo funciona.

Gentle Shell
El espacio de trabajo del agente, construido sobre Pi (paquete npm `gentle-pi`).

Gentle AI
Configura los agentes que ya usas: Claude Code, Cursor, OpenCode, Codex, Pi y más.

Engram
Memoria persistente entre sesiones: un binario y una base SQLite.

Servidores MCP
Herramientas externas que el agente puede usar: documentación, navegador, archivos.

Fuentes: documentación oficial de Gentleman Programming y de Pi, consultada el 4 de noviembre de 2026 (lista completa en la última sección).
Los comandos de esta guía están copiados de esas fuentes. Si una versión nueva cambia un paso, manda la documentación oficial.

Contenido

1	Qué vas a instalar	las piezas y cómo encajan
2	Elige tu camino de instalación	cuatro caminos, uno por perfil
3	Requisitos por sistema operativo	qué debe haber antes
4	Camino 1 · Instalador descargable	el más simple, sin terminal
5	Camino 2 · `gentle-shell` con npm	no toca tu Pi
6	Camino 3 · Dentro de un Pi existente	para quien ya usa Pi
7	Camino 4 · Desde una copia del repositorio	instalador desde el código
8	Primer arranque y verificación	proveedor, modelo, diagnóstico
9	Servidores MCP	agregar, comprobar y depurar
10	Gentle AI para otros agentes	Claude Code, Cursor, OpenCode, Codex
11	Engram por separado	memoria para cualquier agente
12	Actualizar y desinstalar	canales, rollback, limpieza
13	Problemas frecuentes	síntoma → causa → solución
14	Lista de comprobación final	para imprimir y marcar
15	Fuentes	documentación oficial

> **Cómo leer esta guía**
> No hace falta leerla entera. Lee las secciones 1 a 3, elige *un* camino entre las secciones 4 y 7 y después sigue con la 8. Las secciones 9 a 13 son de consulta: vuelve a ellas cuando las necesites.

1 Qué vas a instalar

Antes de copiar comandos, conviene entender qué hace cada pieza. La mayoría de los problemas de instalación vienen de confundir estas capas.

Tú quien dirige	Decides qué se construye, apruebas los cambios y eliges qué se publica. El agente ejecuta; la decisión final es tuya.
Gentle Shell paquete `gentle-pi`	El espacio de trabajo: persona «el Gentleman», flujo de trabajo ODD, subagentes, lista de tareas, vista de cambios de Git, modos de permisos y comandos `/gentle:*`.
Gentle AI binario `gentle-ai`	Configura agentes con memoria, ODD, skills y revisión. Gentle Shell trae su propia copia fijada; también se instala aparte para otros agentes.
Engram memoria	Guarda decisiones, hallazgos y correcciones entre sesiones en `~/.engram/engram.db`. En Pi llega con el paquete `gentle-engram`.
Pi el agente base	El agente de programación en la terminal (`pi`). Gentle Shell funciona encima de Pi y requiere Pi 1.2.0 o posterior.
Modelo proveedor	El modelo de lenguaje que piensa: lo eliges con `/model` entre los proveedores en los que iniciaste sesión.

Tres nombres que se confunden

Nombre	Qué es
Gentle AI	Configura los agentes de programación que ya usas (Claude Code, OpenCode, Cursor, Codex, Pi y otros) con memoria, el flujo ODD, skills y controles de revisión. **Nunca instala un agente por ti.**
Gentle Shell	Un espacio de trabajo completo sobre Pi, con su propio Pi, agentes e interfaz. Su instalador incluye su Pi exacto, un Gentle AI fijado y Engram.
`gentle-pi`	No es un tercer producto: es el nombre del paquete npm y del repositorio de Gentle Shell mientras dura la migración de nombre.

En corto: Gentle AI mejora el agente que ya tienes; Gentle Shell es el espacio de trabajo completo sobre Pi, con Gentle AI adentro.

Qué instala cada camino

Componente	Instalador descargable	npm (`gentle-shell`)	Dentro de Pi
Node.js, npm, pnpm	Sí, si faltan (copia propia)	Debes tenerlos	Debes tenerlos
Pi	Sí, si falta	Debes tenerlo (≥ 1.2.0)	Ya lo tienes
Gentle Shell (`gentle-pi`)	Sí	Sí	Sí
Gentle AI fijado	Sí	Sí, dentro del paquete	Sí, dentro del paquete
Engram	Sí, en su propia carpeta	Paquete `gentle-engram` en el hogar aislado	Paquete `gentle-engram`
Toca tu Pi personal	No	No (salvo `--link`)	Sí

2 Elige tu camino de instalación

Hay cuatro formas de instalar Gentle Shell. Todas terminan en el mismo producto; cambia quién pone las herramientas previas y dónde queda instalado.

Si tú…	Usa	Sección
Quieres lo más simple y no usar la terminal	**Camino 1:** descargar y hacer doble clic	4
Te manejas con npm y quieres que tu Pi no cambie	**Camino 2:** `gentle-shell` independiente (recomendado)	5
Ya usas Pi y quieres el paquete dentro de él	**Camino 3:** dentro de un Pi existente	6
Trabajas desde una copia (clone) del repositorio	**Camino 4:** instalador desde el código	7
No quieres Gentle Shell: solo mejorar Claude Code, Cursor, OpenCode o Codex	**Gentle AI** por separado	10

> **Recomendación**
> Si es tu primera vez, usa el **Camino 1**. Instala lo que falta, actualiza lo que ya tienes y no cambia nada hasta que confirmas el plan que te muestra. Si ya tienes Node.js 22.19.0 o posterior y Pi 1.2.0 o posterior, el **Camino 2** es igual de corto.

> **¿Ya lo instalaste con npm antes de la versión 4.1.0?**
> Usa el instalador descargable (Camino 1): detecta tu instalación y te muestra el plan de actualización. Antes de la 4.1.0 el lanzador no tenía el comando `upgrade` .

3 Requisitos por sistema operativo

Requisito	Camino 1 (descargable)	Caminos 2 y 3 (npm / Pi)
Node.js	No hace falta: usa su propia copia fijada si el tuyo falta o es antiguo	**22.19.0 o posterior**
Pi	Lo instala si falta; reutiliza el tuyo si cumple el mínimo	**1.2.0 o posterior**
pnpm	Lo instala si falta (pnpm 11.1.1 o posterior se reutiliza)	No requerido por npm
Git	No hace falta para instalar, pero sí para revisión de código, worktrees y funciones del repositorio. Si falta, el plan te da el comando para instalarlo.	
Go	Solo cuando hace falta compilar (canal `main`, o Windows). Si el tuyo falta o es antiguo, el instalador descarga un Go fijado y verificado solo para compilar, sin tocar tu Go ni tu PATH.	

MACOS **Antes de empezar**

- Para el Camino 1 no necesitas nada más.
- Para Gentle AI con Homebrew: Homebrew en el PATH.
- La primera vez, macOS bloquea el instalador porque no está firmado (sección 4).

WINDOWS **Antes de empezar**

- Descomprime el zip antes de ejecutar: el instalador no corre desde dentro del zip.
- Gentle AI por separado se instala con Go 1.25.10 o posterior (sección 10).
- SmartScreen avisa la primera vez (sección 4).

LINUX **Antes de empezar**

- Para el Camino 1 basta con poder ejecutar un script `sh`.
- Para Gentle AI por separado: el gestor de paquetes de tu distribución (`apt-get` en Ubuntu/Debian, `pacman` en Arch, `dnf` en Fedora/RHEL), acceso `sudo` y `git`.

4 Camino 1 · Instalador descargable

Descarga el instalador de tu sistema, haz doble clic y sigue los pasos en el navegador. Instala Node.js, pnpm, Pi y Gentle Shell, o actualiza lo que ya tienes. **No instala nada hasta que confirmas el plan.**

Descarga solo desde la página de *Releases* del repositorio oficial:
github.com/Gentleman-Programming/gentle-shell/releases

MACOS gentle-shell-installer-macos.zip

1. Descarga `gentle-shell-installer-macos.zip`.
2. Abre el zip y entra en la carpeta **Gentle Shell Installer**.
3. Haz doble clic en **Install Gentle Shell.command**.
4. Si aparece *"Install Gentle Shell.command" Not Opened* (o *Apple cannot check it*): pulsa **Done**, abre **Ajustes del Sistema** → **Privacidad y seguridad**, baja hasta el aviso del instalador, pulsa **Open Anyway** y confirma con tu contraseña. En versiones antiguas de macOS: clic derecho en el archivo → **Abrir** → **Abrir**. Solo se hace una vez.

WINDOWS gentle-shell-installer-windows.zip

1. Descarga `gentle-shell-installer-windows.zip`.
2. Clic derecho en el zip → **Extraer todo**. No funciona desde dentro del zip.
3. Abre la carpeta extraída, después la carpeta **Gentle Shell Installer**, y haz doble clic en **Install Gentle Shell.cmd**.
4. Si aparece *Windows protegió su PC*: pulsa **Más información** y después **Ejecutar de todas formas**. Si aparece *Abrir archivo – Advertencia de seguridad*, pulsa **Ejecutar**.

LINUX gentle-shell-installer-linux.tar.gz

1. Descarga `gentle-shell-installer-linux.tar.gz` y extráelo.
2. Haz doble clic en **install-gentle-shell.sh**.
3. Si tu gestor de archivos lo abre en un editor en vez de ejecutarlo, abre una terminal en esa carpeta y ejecuta:

```
sh install-gentle-shell.sh
```

Lo que pasa después del doble clic

1. Se abre una ventana pequeña y después el instalador aparece en tu navegador, en una dirección privada `127.0.0.1`.
2. Revisa el plan: dice qué va a instalar, qué va a actualizar y qué deja como está.
3. Confirma el plan en esa pestaña. **Deja la ventana pequeña abierta** hasta que el instalador diga que terminó.

> **Dónde queda instalado**
> Una instalación nueva guarda su propio Node.js, npm, pnpm y Pi en `~/.gentle-shell` (en Windows, dentro de tu carpeta de usuario). Tus herramientas no se usan ni se modifican.

Opcional: verificar la descarga

Cada release publica el archivo `gentle-shell-installers-SHA256SUMS.txt`. Calcula el SHA-256 de tu descarga y compáralo con la línea correspondiente:

macOS

```
shasum -a 256 gentle-shell-installer-macos.zip
```

Linux

```
sha256sum gentle-shell-installer-linux.tar.gz
```

Windows (PowerShell o cmd)

```
certutil -hashfile gentle-shell-installer-windows.zip SHA256
```

Es una versión preliminar

El instalador está probado en Linux y en CI; las pruebas en máquinas limpias de macOS y Windows todavía se están verificando. Si algo falla, el Camino 2 es la alternativa directa.

5 Camino 2 · `gentle-shell` con npm

`gentle-shell` abre Pi con Gentle Shell cargado, **sin** instalarlo dentro de tu Pi ni editar su `settings.json`. Requiere Pi 1.2.0 o posterior y Node.js 22.19.0 o posterior.

Paso 1 · Comprueba las versiones

macOS · Linux · Windows

```
node --version   # debe ser v22.19.0 o posterior
pi --version     # debe ser 1.2.0 o posterior
```

Paso 2 · Instala y arranca

macOS · Linux · Windows

```
npm i -g gentle-pi

# Hogar propio: nunca toca tu instalación de pi
gentle-shell
```

El primer arranque prepara solo su hogar en `~/.gentle-shell/agent`, con los paquetes complementarios de Gentle AI y sin complementos en conflicto. La configuración queda en `~/.gentle-shell/gentle-ai`. No hay un paso aparte.

¿Hogar propio o compartido con Pi?

Comando	Qué hace
`gentle-shell`	Hogar aislado. Tu Pi personal no cambia. Necesitas iniciar sesión en tus proveedores dentro de este hogar.
`gentle-shell --link`	Reutiliza `~/.pi/agent` tal cual: tus sesiones de proveedores, modelos y chats de Pi.
`gentle-shell home link`	Hace que `--link` sea el comportamiento por defecto.
`gentle-shell home import-pi`	Mantiene el hogar aislado, pero copia una vez `auth.json`, `mcp.json` y tu proveedor, modelo y nivel de razonamiento por defecto. Nunca sobrescribe nada ni cambia tu Pi.

Copiar tus sesiones de Pi al hogar aislado

```
# Primero mira qué copiaría, sin cambiar nada
gentle-shell home import-pi --dry-run
gentle-shell home import-pi
```

Compensación entre `--link` e `import-pi`

`--link` comparte un único almacén de credenciales con Pi; `import-pi` crea una segunda copia. Un proveedor que rota sus tokens puede cerrar la sesión en una de las dos: si pasa, ejecuta `/login` otra vez allí.

Volver a preparar el hogar a mano

```
# Repite la preparación y muestra la salida completa
gentle-shell setup
```

Cualquier otro argumento pasa a Pi sin cambios, por ejemplo `gentle-shell -p "..."`.

6 Camino 3 · Dentro de un Pi existente

Instala el paquete dentro de tu agente Pi, reinicia Pi y sincroniza los recursos instalados. A diferencia del Camino 2, este **sí** modifica tu Pi personal.

macOS · Linux · Windows

```
# 1. Instala la versión estable publicada
pi install npm:gentle-pi

# 2. Reinicia Pi y después sincroniza
gentle-ai sync

# 3. Arranca Pi en tu proyecto
cd mi-proyecto
pi
```

> **Fijar una versión concreta**
> Puedes instalar una versión exacta con `pi install npm:gentle-pi@<versión>`, por ejemplo la que indiquen las notas de la release que quieres usar.

La alternativa con Gentle AI

Si ya tienes Gentle AI instalado (sección 10), también puede preparar Pi por ti. Pi debe estar en el PATH antes de ejecutarlo:

```
gentle-ai install --agent pi
pi
```

Gentle AI ejecuta estos pasos de instalación:

```
pi install npm:gentle-pi
pi install npm:gentle-engram
pi install npm:pi-web-access
pi install npm:pi-btw
```

Paquete	Para qué sirve
`gentle-pi` (Gentle Shell)	Arnés de Pi, guía ODD, persona, modelos, skills, herramienta de preguntas y delegación.
`gentle-engram`	Memoria de sesión en Pi y herramientas de Engram.
`pi-web-access`, `pi-btw`	Acceso web y flujo de conversación paralela (`/btw`).

> **Paquetes que ya no se usan**
> No instales `pi-subagents-j0k3r`, `@juicesharp/rpiv-ask-user-question` ni `pi-mcp-adapter` junto a `gentle-pi`: el paquete trae sus propios reemplazos. Los nombres de herramientas de Pi son exclusivos y un duplicado puede impedir que Pi arranque. `gentle-ai install` y `gentle-ai sync` los retiran de la configuración administrada.

7 Camino 4 · Desde una copia del repositorio

Es el mismo instalador del Camino 1, pero arrancado desde un *clone*:

macOS · Linux

```
git clone https://github.com/Gentleman-Programming/gentle-shell.git
cd gentle-shell
sh scripts/bootstrap.sh
```

Windows (cmd)

```
git clone https://github.com/Gentleman-Programming/gentle-shell.git
cd gentle-shell
scripts\bootstrap.cmd
```

El script consigue Node.js y pnpm en una carpeta temporal si faltan y abre el asistente en tu navegador en una dirección privada `127.0.0.1`. **Usa la pestaña que abre:** el enlace funciona una sola vez y caduca a los pocos minutos.

8 Primer arranque y verificación

Paso 1 · Abre tu proyecto

```
cd ruta/a/tu-proyecto
gentle-shell      # o "pi" si usaste el Camino 3
```

Paso 2 · Inicia sesión en un proveedor

Dentro de la sesión, usa el comando nativo de Pi:

```
/login
```

Elige tu proveedor y sigue las instrucciones. Gentle Shell incluye el proveedor propio **NaN**; para usarlo hay dos opciones:

Opción A · dentro de la sesión

```
/login nan
```

Opción B · variable de entorno antes de arrancar

```
# macOS / Linux
export NAN_API_KEY="tu-clave"

# Windows (PowerShell)
$env:NAN_API_KEY = "tu-clave"
```

Una clave guardada con `/login` tiene prioridad sobre `NAN_API_KEY`. Nunca pegues claves en archivos del proyecto ni en commits.

Paso 3 · Elige el modelo

```
/model
```

Muestra los modelos que ofrecen los proveedores en los que iniciaste sesión.

Paso 4 · Comprueba la instalación

```
/gentle:status
/gentle:doctor
```

Comando	Qué revisa
`/gentle:status`	Estado de la configuración de Gentle Shell en esta sesión.
`/gentle:doctor`	Diagnóstico de la instalación y de sus piezas.
`/mcp`	Servidores MCP configurados y su estado de conexión (sección 9).

Paso 5 · Prueba el flujo de trabajo

Describe el resultado que quieres, en lenguaje natural. El agente sigue ODD (*Organic Driven Development*): autoriza, explora, resuelve dudas, clasifica el tamaño de la tarea, registra el trabajo grande, implementa tarea por tarea y cierra con la verificación. Las preguntas y las revisiones son de solo lectura hasta que pides un cambio.

> **Comprobación rápida de memoria**
> Pídele al agente que recuerde algo de una sesión anterior. Si Engram está activo, buscará en su memoria persistente; si no lo está, te lo dirá en vez de inventar.

> **¿Aviso sobre `codemode` al arrancar?**
> Gentle Shell reemplaza el `codemode` incluido en Pi y Pi avisa que no cargó el suyo. En tu propio hogar de Pi (Camino 3 o `--link`), Gentle Shell pregunta una vez si quieres agregar `"-builtin:codemode"` a `extensions` en `settings.json`. Los hogares aislados de `gentle-shell` ya lo traen.

9 Servidores MCP

Un servidor MCP (*Model Context Protocol*) le da herramientas nuevas al agente: documentación actualizada, un navegador, un editor de PDF, etc. Desde Pi 0.99, MCP viene incluido en Pi: no hace falta ningún adaptador.

Dónde se configuran

Archivo	Alcance
`<hogar del agente>/mcp.json`	Servidores de usuario. En Pi es `~/.pi/agent/mcp.json`; con `gentle-shell` en hogar aislado, el hogar es `~/.gentle-shell/agent`.
`.pi/mcp.json` (en el proyecto)	Servidores del proyecto. Pi solo los lee después de que confías en el proyecto.

Formato

El formato es el mismo que el de otros clientes MCP (Claude Desktop, Claude Code, Cursor):

mcp.json

```
{
  "mcpServers": {
    "filesystem": {
      "command": "npx",
      "args": ["-y", "@modelcontextprotocol/server-filesystem", "."]
    },
    "docs": {
      "url": "https://example.com/mcp",
      "headers": { "Authorization": "Bearer ${DOCS_TOKEN}" },
      "description": "Search and read the product documentation"
    }
  }
}
```

- **Servidores locales (stdio):** `command` es un solo ejecutable y `args` sus argumentos; no es una línea de shell. También aceptan `env` y `cwd`.
- **Servidores remotos (HTTP):** `url`, `headers` y `oauth`. El transporte SSE antiguo no está soportado.
- **Secretos:** usa variables de entorno como `${GITHUB_TOKEN}`; nunca escribas la clave en el archivo. Guarda los servidores con credenciales en el archivo de usuario, no en el del proyecto.
- **Nombres:** solo letras, dígitos, `_` y `-`. Las herramientas aparecen como `mcp__<servidor>__<herramienta>`.

Agregar y comprobar desde la terminal

Estos comandos escriben en el hogar de Pi (~/.pi/agent)

```
pi mcp add filesystem -- npx -y @modelcontextprotocol/server-filesystem .
pi mcp list      # conecta cada servidor y muestra estado, herramientas y errores
```

Para el hogar aislado de `gentle-shell`, edita su `mcp.json` o copia el de Pi con `gentle-shell home import-pi`.

Dentro de la sesión

Comando	Para qué
`/mcp`	Lista servidores, estado, cantidad de herramientas y origen; permite reconectar, iniciar sesión, activar o desactivar.
`/reload`	Recarga la configuración después de editar `mcp.json` fuera de la sesión.
`/mcp login <servidor>`	Inicia sesión OAuth en un servidor remoto que lo pide.

Si agregas un servidor y el agente no lo ve

Las herramientas MCP se registran al iniciar la sesión. Ejecuta `/reload` o reinicia la sesión, y revisa `/mcp`: muestra el error completo y el final del *stderr* del servidor que falló. El registro está en `mcp.log`, dentro del hogar del agente.

10 Gentle AI para otros agentes

Usa esta sección si quieres llevar memoria, ODD, skills y controles de revisión a **Claude Code, Cursor, OpenCode, Codex** u otro agente que ya usas. Gentle AI configura el agente, pero no lo instala: instálalo tú antes.

Requisitos

- **Git 2.38 o posterior** en todas las plataformas.
- **Node.js 18 o posterior y npm**: `gentle-ai install` los verifica y, si faltan, te sugiere cómo instalarlos para tu distribución. No los instala por ti.
- **Go 1.25.10 o posterior** para compilar desde el código (obligatorio en Windows).
- Si eliges el agente Pi u OpenCode, `pi` (o `opencode --version`) debe funcionar antes de instalar.

Instalación

MACOS · LINUX Script oficial

```
curl -fsSL https://raw.githubusercontent.com/Gentleman-Programming/gentle-ai/main/scripts/install.sh | bash
```

MACOS · LINUX Homebrew

```
brew tap Gentleman-Programming/homebrew-tap
brew trust --formula gentleman-programming/tap/gentle-ai
brew install gentle-ai
```

El comando `brew trust` solo hace falta si Homebrew pide confiar en la fórmula.

WINDOWS · CUALQUIER SISTEMA CON GO go install

```
go install github.com/gentleman-programming/gentle-ai/v4/cmd/gentle-ai@latest
gentle-ai version
```

El sufijo `/v4` es obligatorio: Go exige el número de versión mayor en la ruta del módulo. Las rutas `/v2` o `/v3` se quedan en versiones antiguas. En Windows, los binarios oficiales y Scoop están en pausa mientras se gestiona la firma de código.

Configurar tus agentes

```
gentle-ai install                     # asistente interactivo: agentes, componentes, persona
gentle-ai install --scope=workspace   # solo para el proyecto actual
gentle-ai doctor                      # chequeo de salud de solo lectura
gentle-ai sync                        # refresca los recursos tras actualizar el binario
```

Preset	Incluye
`full-gentleman`	Todos los componentes (Engram, SDD, Skills, Context7, GGA, Permisos, Tema) y todas las skills.
`ecosystem-only`	Engram, SDD, Skills, Context7 y GGA, con todas las skills.
`minimal`	Engram y las skills de SDD.
`custom`	Eliges componentes y skills a mano.

11 Engram por separado

Engram es un solo binario: memoria persistente para cualquier agente (Claude Code, OpenCode, Gemini CLI, Codex, VS Code, Cursor…). **Si usas Gentle Shell o Gentle AI, ya viene incluido**; instálalo aparte solo si lo quieres sin ellos.

macOS · Linux (Homebrew)

```
brew install gentleman-programming/tap/engram
```

Windows · cualquier sistema con Go

```
go install github.com/Gentleman-Programming/engram/cmd/engram@latest
```

Dato	Valor
Datos (macOS, Linux)	`~/.engram` · base `~/.engram/engram.db`
Datos (Windows)	`%USERPROFILE%\.engram`
Variables útiles	`ENGRAM_DATA_DIR` (carpeta), `ENGRAM_PORT` (puerto HTTP), `ENGRAM_PROJECT` (forzar proyecto)

Uso diario (opcional)

```
engram tui                    # explorar memorias
engram search "auth refactor"   # buscar desde la terminal
engram sync                   # exportar memorias del proyecto a .engram/
```

12 Actualizar y desinstalar

Actualizar Gentle Shell

`gentle-shell upgrade` usa el gestor de paquetes que es dueño de tu instalación.

```
gentle-shell upgrade                    # canal actual
gentle-shell upgrade --channel release  # versiones estables
gentle-shell upgrade --channel main     # cambios sin publicar (compila con Go)
gentle-shell upgrade --rollback         # volver a la versión anterior
```

En el Camino 3 (dentro de Pi): `pi install npm:gentle-pi@<versión>`, reinicia Pi y ejecuta `gentle-ai sync`.

Actualizar Gentle AI y Engram

```
gentle-ai upgrade      # después: gentle-ai sync
brew update && brew upgrade engram
```

Desinstalar Gentle Shell

```
# Mira qué se eliminaría y después elimina tras confirmar
gentle-shell self-uninstall --dry-run
gentle-shell self-uninstall
```

- Elimina los datos propios de Gentle Shell, su hogar aislado (con la configuración de `~/.gentle-shell/gentle-ai`) y el paquete `gentle-pi`.
- Conserva la configuración compartida de `~/.pi/gentle-ai`, salvo que aceptes o pases `--include-shared`.
- **Nunca toca tu hogar de Pi** ni los hogares personalizados creados con `--home`.
- En una instalación hecha con el instalador descargable, elimina la carpeta incluida y la única entrada del PATH que registró el instalador.

13 Problemas frecuentes

Síntoma	Causa	Solución
macOS: *“Install Gentle Shell.command” Not Opened*	El instalador no está firmado por Apple.	Ajustes del Sistema → Privacidad y seguridad → **Open Anyway** (sección 4).
Windows: *Windows protegió su PC*	Instalador sin firma de Microsoft.	**Más información** → **Ejecutar de todas formas**.
Windows: el instalador no arranca	Se ejecutó desde dentro del zip.	Clic derecho → **Extraer todo** y ejecútalo desde la carpeta extraída.
Linux: el doble clic abre un editor	El gestor de archivos no ejecuta scripts.	`sh install-gentle-shell.sh` en esa carpeta.
`gentle-ai install` se detiene al elegir Pi	`pi` no está en el PATH.	Instala Pi y comprueba `pi --version` antes de repetir.
Pi no arranca después de instalar `gentle-pi`	Paquete duplicado con nombres de herramientas en conflicto.	Retira `@juicesharp/rpiv-ask-user-question` y similares; ejecuta `gentle-ai sync`.
Un proveedor pide iniciar sesión otra vez	Rotó su token y tienes dos copias (`--link` / `import-pi`).	`/login` en el hogar afectado.
El agente no ve un servidor MCP nuevo	Las herramientas se cargan al iniciar la sesión.	`/reload` o reinicia; revisa `/mcp` y `pi mcp list`.
Aviso de `codemode` al arrancar	Gentle Shell reemplaza el `codemode` de Pi.	Acepta agregar `"-builtin:codemode"` a `extensions`.
Las skills no se cargan	Pi arrancó con `pi -ns` (sin hooks).	`gentle-ai skill-registry refresh`.
La revisión nativa falla en WSL	Montaje DrvFS sin metadatos POSIX.	Trabaja en el sistema de archivos de Linux, o desactiva RDD con `/gentle:review-mode disable`.
Algo no cuadra y no sabes qué	—	`/gentle:doctor` en la sesión; `gentle-ai doctor` en la terminal.

14 Lista de comprobación final

Instalación

- [ ] Elegí un solo camino de instalación (sección 2).
- [ ] El instalador terminó sin errores, o `gentle-shell` / `pi` arranca.
- [ ] Git está instalado (`git --version`).

Primer arranque

- [ ] Inicié sesión en al menos un proveedor con `/login`.

- [ ] Elegí un modelo con `/model`.
- [ ] `/gentle:status` y `/gentle:doctor` no muestran errores.
- [ ] El agente se presenta como «el Gentleman» y responde en mi idioma.

Extras

- [ ] Mis servidores MCP aparecen conectados en `/mcp`.
- [ ] La memoria (Engram) responde: el agente puede buscar trabajo anterior.
- [ ] Sé actualizar (`gentle-shell upgrade`) y desinstalar (`self-uninstall --dry-run`).
- [ ] Ninguna clave quedó escrita en archivos del proyecto ni en commits.

15 Fuentes

Consultadas el 4 de noviembre de 2026. Ante cualquier diferencia, mandan estas páginas.

- Gentle Shell (sitio): https://gentlemanprogramming.com/gentle-shell/
- Gentle Shell (repositorio y README): https://github.com/Gentleman-Programming/gentle-shell
- Asistente de instalación: https://github.com/Gentleman-Programming/gentle-shell/blob/main/docs/install-wizard.md
- Releases de Gentle Shell: https://github.com/Gentleman-Programming/gentle-shell/releases
- Gentle AI · quickstart: https://github.com/Gentleman-Programming/gentle-ai/blob/main/docs/quickstart.md
- Gentle AI · integración con Pi: https://github.com/Gentleman-Programming/gentle-ai/blob/main/docs/pi.md
- Gentle AI · wiki: https://gentle-ai-wiki.gentlemanprogramming.com/
- Engram · instalación: https://engram.gentlemanprogramming.com/getting-started/installation/
- Pi · servidores MCP: https://pi.dev/docs/latest/mcp
- Paquete `gentle-pi`: https://pi.dev/packages/gentle-pi