Mirador

Manual del agente local

Instalación, configuración y operación del agente que conecta Mirador con las bases de datos de la red de su organización, sin abrir puertos entrantes ni entregar credenciales.

Agente
versión 0.1

Edición
septiembre de 2026

Para
administradores de Mirador e infraestructura

CONTENIDO

En este manual

1 Qué es el agente y cuándo hace falta

2 Antes de empezar

3 Generar el token de enrolamiento

4 Descargar y verificar el paquete

5 Configurar el agente

6 Iniciar el agente

7 Crear la conexión en Mirador y verificar

8 Diagnóstico de problemas

9 Operación: redundancia, actualización y baja

10 Seguridad: lo que el diseño garantiza

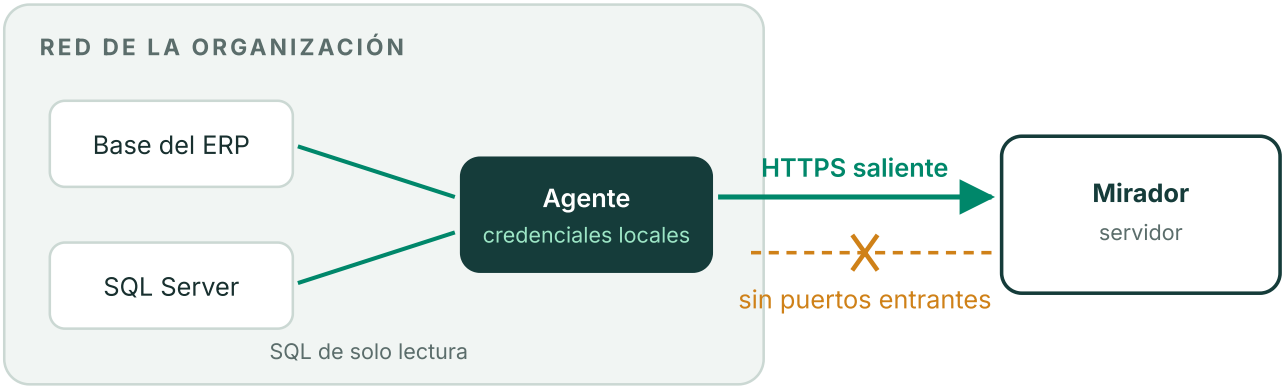
A Referencia: variables de entorno

1

Qué es el agente y cuándo hace falta

El agente es un programa pequeño que se instala dentro de la red de la organización y ejecuta, en nombre de Mirador, las consultas contra las bases que Mirador no puede alcanzar por sí mismo.

Mirador puede conectarse a una base de dos maneras. En la **conexión directa**, el servidor de Mirador abre la conexión a la base: sirve cuando la base es accesible desde internet o desde la red donde corre Mirador. Cuando la base vive en una red interna —el ERP en el servidor de la oficina, un SQL Server en el centro de datos—, abrirla hacia afuera suele ser inaceptable. Para ese caso está el **agente**.



El agente inicia todas las comunicaciones: pide trabajo a Mirador, ejecuta la consulta contra la base local y devuelve el resultado.

Tres propiedades lo definen:

- **No escucha en ningún puerto.** Solo hace conexiones HTTPS salientes hacia Mirador. No hay que abrir nada en el firewall de entrada.
- **Las credenciales de las bases no salen de su máquina.** Mirador solo conoce un nombre lógico por cada base (por ejemplo, `erp-produccion`); el servidor, el usuario y la contraseña quedan en el archivo de configuración del agente.
- **Revalida cada consulta** antes de ejecutarla: rechaza todo lo que no sea una única sentencia de lectura, aunque llegara desde el servidor.

Una vez creada la conexión, para el resto de Mirador es una conexión más: consultas, reportes, tableros y envíos programados funcionan igual que con una conexión directa.

2

Antes de empezar

Quién hace cada paso

El procedimiento casi siempre lo hacen dos personas distintas, y conviene coordinarlo de antemano:

Paso	Quién
3. Generar el token de enrolamiento	Administrador de Mirador
4–6. Descargar, configurar e iniciar el agente	Infraestructura de la organización
7. Crear la conexión en Mirador y verificar	Administrador de Mirador

Lo único que pasa de una persona a la otra es el token, y **vence en una hora**: conviene generarlo cuando infraestructura esté por instalar, no el día anterior.

Requisitos de la máquina

Requisito	Detalle
Sistema	Un servidor Linux o Windows que quede encendido. Puede ser una máquina virtual.
Docker *o* Java 21	La forma recomendada es Docker (Docker Engine con Compose). Alternativa: Java 21 (JRE), por ejemplo Eclipse Temurin.
Memoria	512 MB alcanzan para un agente con pocas conexiones.
Disco	Unos 600 MB con Docker (incluye la imagen base de Java); unos 50 MB más Java 21 sin Docker.
Salida HTTPS	Hacia la dirección de Mirador de la organización (puerto 443).
Red hacia las bases	Visibilidad hacia cada base que va a atender, en el puerto del motor.
Usuario de base	Un usuario **de solo lectura** por base.

Motores soportados

Motor	`database-type`	Puerto habitual
PostgreSQL	`POSTGRESQL`	5432
Microsoft SQL Server	`SQL_SERVER`	1433
Oracle	`ORACLE`	1521
MySQL	`MYSQL`	3306
MariaDB	`MARIADB`	3306

Los controladores de los cinco motores vienen dentro del agente: no hay que instalar nada aparte.

Sobre el usuario de solo lectura
El agente rechaza cualquier sentencia que no sea una lectura, pero esa es la segunda línea de defensa. La primera es que el usuario de la base no tenga permiso para escribir: si alguna vez fallara el validador, la base igual se negaría.

3

Generar el token de enrolamiento

El token es lo que permite que un agente nuevo se presente ante Mirador como parte de su organización. Lo genera un administrador de Mirador.

1. Entrar a Mirador con un usuario administrador.
2. Ir a **Administración** → **Agentes** y pulsar **Enrolar agente**.
3. Escribir el nombre del **grupo**. Si no existe, se crea.
4. Pulsar **Generar token** y copiarlo.

Cómo nombrar el grupo. Conviene nombrarlo por la **red donde viven las bases** — `montevideo`, `datacenter-norte` — y no por la máquina donde corre el agente. Los agentes de un mismo grupo se reparten el trabajo y se cubren entre sí, y eso solo tiene sentido si todos alcanzan las mismas bases.

Tres cosas del token

- **Se muestra una sola vez.**
  Mirador guarda solo un resumen; no se puede volver a ver. Si se pierde, se genera otro.
- **Es de un solo uso.**
  Enrola un agente. Para un segundo agente, se genera un token nuevo.
- **Vence en una hora.**
  Pasado ese plazo hay que generar otro.

Para hacerlo llegar a infraestructura, usar un canal que no lo deje archivado indefinidamente (por ejemplo, un mensaje que se borre después). Una vez usado o vencido no sirve para nada, pero mientras está vigente permite enrolar un agente.

¿No aparece la pantalla Agentes?
El modo agente está deshabilitado en la instalación. En el servicio de Mirador, pedir que lo habiliten a soporte. En una instalación propia, agregar `DATASCOPE_DATAPLANE_MODES=DIRECT,AGENT` al `.env` del servidor y recrear el servicio `backend`.

4

Descargar y verificar el paquete

El agente se distribuye en un único paquete, en dos formatos con el mismo contenido:

Archivo	Para	Dirección
`mirador-agente.zip`	Windows	www.mirador.uy/descargas/mirador-agente.zip
`mirador-agente.tar.gz`	Linux	www.mirador.uy/descargas/mirador-agente.tar.gz
`SHA256SUMS.txt`	Verificación	www.mirador.uy/descargas/SHA256SUMS.txt

En un servidor Linux sin navegador:

```
cd /opt
sudo curl -fLO https://www.mirador.uy/descargas/mirador-agente.tar.gz
sudo curl -fLO https://www.mirador.uy/descargas/SHA256SUMS.txt
sha256sum --check --ignore-missing SHA256SUMS.txt     # debe decir: OK
sudo tar -xzf mirador-agente.tar.gz                   # crea /opt/mirador-agente
```

En Windows, después de descargar, comparar el resultado de este comando con la línea del `.zip` en `SHA256SUMS.txt`, y descomprimir en una carpeta fija, por ejemplo `C:\mirador-agente`:

```
certutil -hashfile mirador-agente.zip SHA256
```

Qué trae el paquete

Archivo	Para qué sirve
`mirador-agent.jar`	El programa del agente.
`config/application.yml`	La configuración: dirección de Mirador y las bases a las que llega.
`.env.example`	Plantilla para el token y las contraseñas.
`compose.yml`, `Dockerfile`	Para correrlo con Docker.
`iniciar-sin-docker.sh` / `.cmd`	Para correrlo con Java 21, sin Docker.
`LEEME.txt`	Resumen de la puesta en marcha.

5

Configurar el agente

La configuración son dos archivos: `config/application.yml`, con lo que el agente tiene que saber, y `.env`, con los secretos.

5.1 · config/application.yml

```
datascope:
  agent:
    server-url: https://tuempresa.mirador.uy   # la dirección con la que se entra a Mirador
    name: agente-erp                           # cómo se ve en la pantalla Agentes
    enrollment-token: ${DATASCOPE_AGENT_ENROLLMENT_TOKEN:}
    connections:
      - name: erp-produccion                   # el nombre que se elige después en Mirador
        database-type: POSTGRESQL
        host: db.interno
        port: 5432
        database-name: erp
        username: lector
        password: ${ERP_PASSWORD}
      - name: ventas
        database-type: SQL_SERVER
        host: sql.interno
        port: 1433
        database-name: Ventas
        username: mirador_ro
        password: ${VENTAS_PASSWORD}
        options:
          encrypt: "true"
          trustServerCertificate: "false"
```

Campo	Qué es
`server-url`	La dirección de Mirador de la organización, la misma que se usa en el navegador, con `https://`.
`name` (del agente)	Un nombre distinto por cada agente. Es el que aparece en la pantalla Agentes.
`connections[].name`	El nombre lógico de la base. **Es lo único que Mirador conoce de ella** y lo que se elige al crear la conexión.
`database-type`	Uno de los valores de la tabla de motores (sección 2).
`host`, `port`	Cómo llega *el agente* a la base: puede ser un nombre o una IP interna.
`database-name`	La base (en Oracle, el nombre de servicio o el SID).
`username`, `password`	El usuario de solo lectura. La contraseña conviene dejarla como `${VARIABLE}` y definirla en `.env`.

Respetar la sangría. YAML depende de los espacios: usar espacios (nunca tabuladores) y mantener la alineación del ejemplo. Un error de sangría impide que el agente arranque.

5.2 · Límites por conexión (opcionales)

Campo	Por defecto	Qué hace
`connect-timeout-seconds`	10	Espera máxima para abrir la conexión.
`query-timeout-seconds`	60	Corta una consulta que no termina.
`max-rows`	50000	Tope de filas que devuelve una consulta.
`max-concurrent-queries`	5	Consultas simultáneas contra esa base.

Ante dos límites —el del agente y el de Mirador— gana el más estricto. Es deliberado: el dueño de la base tiene la última palabra sobre cuánta carga recibe.

5.3 · Opciones del controlador

`options` pasa parámetros al controlador de cada motor. Las claves que no figuran en esta tabla se ignoran.

Motor	Claves	Por defecto
PostgreSQL	`sslMode`	`prefer`
SQL Server	`encrypt`, `trustServerCertificate`, `sslProtocol`, `driver` (`JTDS` para instancias antiguas)	`true`, `false`
Oracle	`serviceType` (`SID`; si se omite, nombre de servicio), `protocol` (`TCPS` para TLS), `verifyServerCertificate`	servicio, TCP, `true`
MySQL	`sslMode`, `allowPublicKeyRetrieval`	`PREFERRED`, `false`
MariaDB	`sslMode`	`disable`

Los valores booleanos van entre comillas (`"true"`), como en el ejemplo.

5.4 · El archivo .env

Copiar `.env.example` a `.env` y completarlo: el token de la sección 3 y una línea por cada contraseña que se haya referenciado en `application.yml`.

```
DATASCOPE_AGENT_ENROLLMENT_TOKEN=pegar-aquí-el-token
ERP_PASSWORD=contraseña-del-usuario-lector
VENTAS_PASSWORD=contraseña-del-usuario-mirador_ro
```

En Linux, dejarlo legible solo por su dueño:

```
chmod 600 /opt/mirador-agente/.env
```

Por qué separar los secretos

Así `application.yml` se puede respaldar, versionar o enviar a soporte sin exponer contraseñas. Lo sensible queda en un único archivo con permisos restringidos.

6

Iniciar el agente

6.1 · Con Docker (recomendado)

Desde la carpeta del paquete:

```
cd /opt/mirador-agente
docker compose up -d --build
```

La primera vez arma la imagen, para lo que descarga la imagen base de Java 21. El agente queda configurado para reiniciarse solo si se cae y al reiniciar la máquina.

Para ver el registro:

```
docker compose logs -f agent
```

La identidad del agente vive en un volumen

Al enrolarse, el agente recibe una credencial permanente que guarda en el volumen `mirador-agente_agent-data`. Esa credencial no se vuelve a emitir: si se borra el volumen (por ejemplo con `docker compose down -v`), hay que enrolar el agente de nuevo con un token nuevo. `docker compose down` sin `-v` lo conserva.

6.2 · Sin Docker, con Java 21

Comprobar la versión con `java -version` (tiene que ser 21 o posterior) y, desde la carpeta del paquete:

```
# Linux
./iniciar-sin-docker.sh

# Windows (doble clic, o desde una consola)
iniciar-sin-docker.cmd
```

El agente corre en primer plano y se detiene con `Ctrl+C`. Lee `config/application.yml` y las variables de `.env`, y guarda su identidad en la subcarpeta `data/`, que **no hay que borrar**.

Sin Docker, `.env` se lee con las reglas de los archivos de propiedades de Java: si una contraseña contiene una barra invertida `\`, escribirla doble (`\\`).

6.3 · Dejarlo como servicio en Linux (sin Docker)

Para que arranque con la máquina y se reinicie si se cae, crear un usuario sin privilegios y una unidad de systemd en `/etc/systemd/system/mirador-agente.service`:

```
sudo useradd --system --home /opt/mirador-agente mirador
sudo chown -R mirador: /opt/mirador-agente
sudo chmod 600 /opt/mirador-agente/.env
```

```
[Unit]
Description=Agente de Mirador
After=network-online.target
Wants=network-online.target

[Service]
User=mirador
WorkingDirectory=/opt/mirador-agente
ExecStart=/opt/mirador-agente/iniciar-sin-docker.sh
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl enable --now mirador-agente
journalctl -u mirador-agente -f        # registro
```

En Windows sin Docker, el lanzador sirve para probar la configuración. Para una instalación permanente en Windows, consultar con soporte de Mirador la forma recomendada para su entorno.

6.4 · Confirmar que arrancó

En el primer arranque, el registro tiene que mostrar estas dos líneas:

```
Agente agente-erp iniciando con 2 conexion(es) configurada(s)
Agente enrolado en el grupo montevideo con id 3f2a…
```

En los arranques siguientes el agente ya tiene su identidad: aparece solo la primera línea y, si no hay advertencias después, está funcionando. Desde ese momento el token ya no hace falta y se puede vaciar en `.env`.

Si en lugar de la segunda línea aparece *«El agente no esta enrolado y no hay token configurado»*, el token no llegó al proceso: revisar que esté en `.env` con el nombre exacto `DATASCOPE_AGENT_ENROLLMENT_TOKEN`.

7

Crear la conexión en Mirador y verificar

1. En Mirador, ir a **Administración** → **Conexiones** → **Nueva conexión**.
2. Elegir **«A través de un agente instalado en el cliente»**. Los campos de servidor y credenciales desaparecen.
3. Elegir el **grupo** de agentes y el **nombre de la conexión**. Esta lista muestra lo que los agentes conectados publicaron, así que no hay que escribirlo a mano.
4. Guardar.

Si el nombre esperado no aparece en la lista, el agente todavía no reportó: esperar medio minuto y recargar la página.

Verificación

1. En **Agentes**, el agente figura como **Conectado**, con su versión y sus conexiones.
2. En la conexión recién creada, **Probar** responde correctamente.
3. Una consulta contra una tabla real devuelve filas.

El tercer punto es el que importa: los dos primeros prueban que el agente habla con Mirador; el tercero, que además alcanza la base con el usuario configurado.

8

Diagnóstico de problemas

Mensajes de «Probar» en Mirador

Mensaje	Qué pasó	Qué hacer
El agente del grupo «X» no está conectado (último latido: …)	Ningún agente del grupo reportó en los últimos 90 segundos.	Revisar que el proceso esté corriendo y que tenga salida HTTPS.
Hay N agentes conectados, pero ninguno tiene configurada la conexión «Y»	El agente funciona, pero ese nombre no está en su `application.yml`.	Corregir el `name` de la conexión y reiniciar el agente.
Este agente no tiene configurada la conexión …	Lo mismo, visto desde el agente.	Igual que el anterior.
Error del motor (usuario, base, servidor)	El agente llegó a la base y la base lo rechazó.	Revisar credenciales y permisos del usuario de solo lectura, y el `host` / `port`.

Mensajes en el registro del agente

Mensaje	Causa habitual
No se pudo contactar al plano de control: …	Salida HTTPS bloqueada, un proxy en el medio o `server-url` mal escrita. El agente reintenta solo en cada latido: no hace falta reiniciarlo.
El token de enrolamiento ya fue usado o expiró	Pasó más de una hora o el token ya enroló a otro agente. Generar uno nuevo.
La versión del agente no está autorizada en esta instalación	La instalación de Mirador solo acepta ciertas versiones del agente. Consultar a soporte qué versión corresponde.
Las credenciales del agente no son válidas / *El agente fue revocado*	Se revocó el agente en Mirador, o la identidad guardada es de una instalación anterior. Ver «Reenrolar» en la sección 9.
Muchos enrolamientos rechazados seguidos	Mirador acepta hasta 5 intentos por minuto desde una misma dirección. Un agente mal configurado que reintenta en bucle los agota.
El agente no arranca y menciona `application.yml`	Error de sangría o de sintaxis en el YAML, o una `${VARIABLE}` sin definir en `.env`.

Para obtener más detalle, agregar `DATASCOPE_AGENT_LOG_LEVEL=DEBUG` al `.env` y reiniciar. Volver a `INFO` al terminar.

9

Operación

Redundancia: un segundo agente en el mismo grupo

Los agentes de un grupo se reparten el trabajo y, si uno se cae, el otro sigue atendiendo. Para agregar uno, repetir el procedimiento desde la sección 3 con un **token nuevo**, el **mismo grupo** y un `name` de agente distinto (`agente-erp-2`).

Las conexiones tienen que llamarse **exactamente igual** en los dos agentes: si uno publica `erp-produccion` y el otro `erp-prod`, cada consulta va a funcionar o no según a cuál le toque.

Si los dos agentes corren con Docker en la misma máquina, cambiar en el segundo la línea `name: mirador-agente` de `compose.yml` para que cada uno tenga su volumen.

Actualizar el agente

1. Descargar el paquete vigente y verificarlo (sección 4).
2. Reemplazar solo `mirador-agent.jar` en la carpeta actual. **No reemplazar** `config/application.yml`, `.env` ni la carpeta `data/`.
3. Reiniciar: con Docker, `docker compose up -d --build`; con systemd, `sudo systemctl restart mirador-agente`.

La identidad se conserva, así que no hay que volver a enrolar. Cuando se actualiza Mirador, conviene actualizar los agentes en la misma ventana de mantenimiento.

Dar de baja un agente

1. En Mirador, **Agentes** → **Revocar**. Desde ese momento su credencial deja de valer.
2. En la máquina del agente, detenerlo y borrar la identidad: con Docker, `docker compose down -v`; sin Docker, borrar la carpeta `data/`.

El orden importa: revocar primero. Un agente que se apaga sin revocar deja una credencial válida en el disco.

Reenrolar después de perder la identidad

Si se perdió el volumen o la carpeta `data/` (máquina recreada, volumen borrado), el agente no puede volver a autenticarse. Revocar el registro viejo en Mirador, generar un token nuevo, ponerlo en `.env` y reiniciar el agente.

10

Seguridad: lo que el diseño garantiza

Para la conversación con el área de seguridad de la organización.

- **Las credenciales de las bases no salen de su infraestructura.** No es solo una promesa de la aplicación: la tabla del servidor que guarda las conexiones por agente tiene una restricción que impide almacenar servidor, usuario o contraseña.
- **Sin puertos entrantes.** El agente solo inicia conexiones HTTPS hacia Mirador.
- **El secreto del agente existe en claro una sola vez**, en la respuesta del enrolamiento. En Mirador queda solo su resumen SHA-256: una copia de la base de Mirador no permite hacerse pasar por el agente.
- **Cada sentencia se revalida en su infraestructura** antes de ejecutarse: el agente acepta una única sentencia de lectura y rechaza escrituras, cambios de estructura y bloqueos.
- **Límites bajo su control.** Los topes de filas, tiempo y concurrencia del agente prevalecen sobre los de Mirador si son más estrictos.
- **Revocación inmediata.** Revocar un agente en Mirador invalida su credencial en el acto.

Lo que hoy no incluye

- La autenticación del agente es un secreto compartido sobre TLS, no TLS mutuo con certificado de cliente.
- El paquete se verifica con SHA-256 publicado en el sitio; no tiene todavía firma digital.

ANEXO A

Referencia: variables de entorno

Además de las contraseñas, en `.env` se pueden definir estas variables:

Variable	Por defecto	Qué hace
`DATASCOPE_AGENT_ENROLLMENT_TOKEN`	vacío	Token del primer arranque.
`DATASCOPE_AGENT_HEARTBEAT_SECONDS`	`30`	Intervalo entre latidos.
`DATASCOPE_AGENT_LOG_LEVEL`	`INFO`	Nivel de registro (`DEBUG` para diagnóstico).
`DATASCOPE_AGENT_IDENTITY_FILE`	`data/agent-identity.json`	Dónde guarda su identidad.

Soporte

Ante cualquier duda con la instalación, escribir a soporte de Mirador indicando la versión del agente, el sistema operativo y las últimas líneas del registro. Nunca enviar el archivo `.env` ni la carpeta `data/` .