

MANUAL DE **MIROFISH**

Instalación paso a paso en Windows

 **DESDE CERO**

De la instalación a tu primera simulación



Git



Node.js



Python



uv



Qwen



Zep



Guía práctica
Pasos claros y sencillos



Configuración real
En tu entorno Windows



Primera simulación
Listo para explorar

MIROFISH

Manual de instalación, configuración y uso en Windows

Guía paso a paso para usuarios que parten desde cero

*Desde la verificación de herramientas hasta el informe final de una
simulación*

ALCANCE: Este manual no presupone que Git, Node.js, Python, uv ni ninguna otra herramienta estén instalados. Cada requisito se verifica primero y, si falta, se explica cómo instalarlo y cómo comprobar que quedó funcionando.

Versión 2.0 · 15 de septiembre de 2026

Entorno documentado: Windows 11, PowerShell, MiroFish desde código fuente

Cómo usar este manual

La guía está pensada para una persona que puede manejar Windows pero no necesariamente está familiarizada con herramientas de desarrollo. Los comandos se muestran completos y se indica exactamente dónde ejecutarlos.

REGLA BÁSICA: Todos los comandos de esta guía se ejecutan en PowerShell, salvo que se indique expresamente “navegador” o “Bloc de notas”. No copie el texto “PS C:\...” del ejemplo; copie solamente el comando.

Cómo abrir PowerShell

1. Presione la tecla Windows o haga clic en Inicio.
2. Escriba PowerShell. En Windows 11 también puede aparecer “Terminal”.
3. Abra “Windows PowerShell” o “Terminal” con un perfil de PowerShell. Para esta guía no es necesario abrirlo como administrador, salvo que Windows lo solicite durante una instalación.
4. Compruebe que la ventana muestra un prompt similar a “PS C:\Users\su_usuario>” o “PS C:\Windows\System32>”. Eso significa que está en PowerShell.

IMPORTANTE: Después de instalar Git, Node.js o uv, cierre PowerShell y abra una ventana nueva antes de verificar la instalación. Así Windows recarga la variable PATH.

Cómo ejecutar los comandos

- Pegue una línea de comando y presione Enter.
- Espere a que termine antes de pegar el siguiente comando.
- Si el prompt “PS ...>” vuelve a aparecer, el comando terminó.
- Si aparece un error, no continúe a ciegas: revise la sección de solución de problemas correspondiente.
- Cuando MiroFish esté ejecutándose con npm run dev, esa ventana debe permanecer abierta.

Contenido

1. Qué es MiroFish y qué piezas utiliza
 2. Preparar Windows y verificar herramientas
 3. Instalar Git
 4. Instalar Node.js y npm
 5. Instalar uv y Python 3.12
 6. Descargar MiroFish
 7. Instalar dependencias
 8. Crear el archivo .env
 9. Configurar Zep
 10. Configurar Qwen en Alibaba Cloud Model Studio
 11. Proteger la cuenta contra cargos inesperados
 12. Crear la API Key y copiar el endpoint correcto
 13. Primer arranque de MiroFish
 14. Crear la primera simulación de prueba
 15. Solucionar el error HTTP 401
 16. Configurar rondas y ejecutar
 17. Qué hacer si la simulación termina pero queda esperando
 18. Generar, abrir y localizar el informe final
 19. Detener y reanudar MiroFish
 20. Revisar consumo y costos
 21. Actualizar MiroFish en el futuro
 22. Seguridad de API keys
 23. Problemas frecuentes
- Apéndice A. Comandos de referencia
- Apéndice B. Enlaces oficiales

1. Qué es MiroFish y qué piezas utiliza

MiroFish no es un programa tradicional con un único instalador setup.exe. Es un proyecto que coordina una interfaz web, un backend en Python, un modelo de lenguaje externo y un servicio de memoria. Por eso se prepara por capas.

Componente	Función en MiroFish	Equivalencia práctica
Git	Descarga y actualiza el código fuente	Como descargar un proyecto y conservar el historial
Node.js + npm	Ejecuta y prepara la interfaz web	Runtime y administrador de bibliotecas JavaScript
Python 3.12	Ejecuta el backend	Motor de la lógica del servidor
uv	Gestiona Python, entorno virtual y paquetes	Administrador de dependencias/entorno aislado
Qwen	Modelo de IA que razona por los agentes	El “cerebro” generativo
Zep	Memoria y grafo de relaciones	La memoria contextual del mundo simulado

REQUISITO OFICIAL: El repositorio de MiroFish documenta Node.js 18 o superior, Python 3.11–3.12 y uv. En esta instalación usamos Python 3.12 administrado por uv.

Repositorio oficial: <https://github.com/666ghj/MiroFish>

2. Preparar Windows y verificar herramientas

Antes de instalar nada, revise qué herramientas existen en el equipo. Esto evita duplicar instalaciones y permite detectar incompatibilidades.

2.1 Abrir PowerShell y comprobar WinGet

1. Abra PowerShell siguiendo las instrucciones del inicio del manual.

2. Escriba el siguiente comando y presione Enter:

```
winget --version
```

Si devuelve una versión, WinGet está disponible. Si indica que winget no se reconoce, actualice o instale “App Installer” desde Microsoft Store y vuelva a abrir PowerShell.

Ayuda oficial de WinGet: [Microsoft Learn - WinGet](#)

2.2 Verificar Git, Node.js, npm, Python y uv

Ejecute estas líneas una por una:

```
git --version
node -v
npm -v
python --version
py --version
uv --version
```

CÓMO LEER EL RESULTADO: “The term ... is not recognized” significa que la herramienta no está instalada o que Windows todavía no conoce su ubicación mediante PATH. No es un error de MiroFish.

En el equipo usado para elaborar este manual, Git, Node.js y uv no estaban instalados y Python era 3.14.7. No se desinstaló Python 3.14: se añadió Python 3.12 en paralelo, exclusivamente para MiroFish.

2.3 Qué debe quedar al final de la preparación

Herramienta	Mínimo/objetivo	Comando de prueba
Git	Instalado	git --version
Node.js	18+ (se recomienda LTS)	node -v
npm	Incluido con Node.js	npm -v
uv	Instalado	uv --version
Python 3.12	Gestionado por uv	uv python list

3. Instalar Git

Git se usa para traer el repositorio oficial de MiroFish a su PC y, más adelante, actualizarlo.

3.1 Método recomendado: WinGet

1. Abra PowerShell.

2. Ejecute:

```
winget install --id Git.Git -e --source winget
```

3. Espere a que termine. Si Windows muestra un aviso de Control de cuentas de usuario, autorice la instalación.

4. Cierre PowerShell por completo y vuelva a abrirlo.

5. Verifique:

```
git --version
```

RESULTADO ESPERADO: Debe aparecer “git version ...”. La versión exacta puede cambiar.

3.2 Si WinGet falla: instalador manual

Descargue Git for Windows únicamente desde el sitio oficial: <https://git-scm.com/download/win>

- Descargue la versión de 64 bits para Windows si su PC es de 64 bits.
- Ejecute el instalador y puede conservar las opciones predeterminadas.
- Al terminar, cierre y reabra PowerShell.
- Vuelva a probar git --version.

PATH: Si Git se instaló pero sigue sin reconocerse, reinicie PowerShell. Si todavía falla, reinicie Windows antes de reinstalar.

4. Instalar Node.js y npm

Node.js ejecuta JavaScript fuera del navegador y npm administra las dependencias JavaScript de MiroFish. npm se instala junto con Node.js.

4.1 Método recomendado: WinGet

1. Abra PowerShell.

2. Ejecute:

```
winget install --id OpenJS.NodeJS.LTS -e --source winget
```

3. Cierre PowerShell y abra una ventana nueva.

4. Verifique ambas herramientas:

```
node -v  
npm -v
```

RESULTADO ESPERADO: Node debe mostrar una versión 18 o superior. El manual se probó con Node v24.19.0 y npm 11.17.0; versiones futuras pueden ser distintas.

4.2 Instalación manual

Descarga oficial de Node.js: <https://nodejs.org/en/download>

- Elija una versión LTS para Windows.
- Ejecute el instalador MSI y mantenga las opciones predeterminadas salvo que tenga una razón para cambiarlas.
- No necesita instalar npm aparte.
- Cierre y reabra PowerShell, luego repita node -v y npm -v.

5. Instalar uv y Python 3.12

MiroFish necesita Python compatible. uv permite instalar Python 3.12 para el proyecto sin reemplazar otras versiones de Python que ya existan en Windows.

5.1 Instalar uv

1. Abra PowerShell.
2. Ejecute el instalador oficial de Astral:

```
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

3. Cierre PowerShell y vuelva a abrirlo.
4. Verifique:

```
uv --version
```

Documentación oficial de instalación: <https://docs.astral.sh/uv/getting-started/installation/>

5.2 Instalar Python 3.12 con uv

1. En PowerShell, ejecute:

```
uv python install 3.12
```

2. Cuando termine, liste las versiones detectadas:

```
uv python list
```

NO BORRE SU OTRO PYTHON: Si python --version sigue mostrando Python 3.14, no hay problema. uv puede utilizar Python 3.12 dentro de MiroFish aunque otra versión siga siendo la predeterminada del sistema.

5.3 Si uv no puede instalar Python

Como alternativa, Python puede descargarse desde el sitio oficial: <https://www.python.org/downloads/windows/>

Use una versión 3.12.x de 64 bits. Esta alternativa solo es necesaria si uv no consigue instalar Python 3.12.

6. Descargar MiroFish desde el repositorio oficial

ANTES DE CLONAR: No cree el proyecto dentro de C:\Windows\System32. Trabaje en su carpeta personal.

1. Abra PowerShell.

2. Vaya a su carpeta personal:

```
cd $HOME
```

3. Clone el repositorio oficial:

```
git clone https://github.com/666ghj/MiroFish.git
```

4. Entre a la carpeta recién creada:

```
cd MiroFish
```

5. Compruebe el estado del repositorio:

```
git status
```

6. Opcionalmente, liste los archivos para confirmar la estructura:

```
dir
```

RESULTADO ESPERADO: Git debe indicar que está en la rama main y que el working tree está limpio. Debe ver carpetas como frontend, backend, scripts y archivos como package.json y .env.example.

7. Instalar las dependencias de MiroFish

Este paso descarga las bibliotecas JavaScript y Python que el código necesita. No las instale una por una; el proyecto ya declara cuáles requiere.

1. Compruebe que PowerShell está dentro de la carpeta MiroFish. El prompt debería terminar en `\MiroFish>`.

2. Ejecute:

```
npm run setup:all
```

3. Espere. La primera ejecución puede tardar varios minutos y descargar una cantidad considerable de datos.

4. Busque en la salida mensajes equivalentes a “Using CPython 3.12...” y “Creating virtual environment at: .venv”.

ADVERTENCIAS DE NPM: Mensajes “warning”, “deprecated” o un resumen de vulnerabilidades no significan necesariamente que la instalación falló. No ejecute `npm audit fix --force` de forma automática: podría cambiar versiones que MiroFish espera.

QUÉ SE CREA: npm instala módulos JavaScript y uv crea `backend\.venv`, un entorno privado de Python para MiroFish. Esto evita mezclar sus paquetes con los de otros proyectos.

8. Crear el archivo .env

El archivo .env guarda la configuración que conecta MiroFish con Qwen y Zep. No modifique .env.example: úselo como plantilla.

1. En PowerShell, estando en C:\Users\su_usuario\MiroFish, cree una copia:

```
Copy-Item .env.example .env
```

2. Abra la copia con Bloc de notas:

```
notepad .env
```

3. No cierre el Bloc de notas: lo completaremos después de crear las claves.

SEGURIDAD: Nunca comparta su archivo .env ni publique capturas que muestren API keys completas. Una API key funciona como una contraseña para un servicio externo.

Al finalizar, la estructura esencial será:

```
LLM_API_KEY=TU_CLAVE_QWEN  
LLM_BASE_URL=TU_ENDPOINT_OPENAI_COMPATIBLE  
LLM_MODEL_NAME=qwen-plus  
  
ZEP_API_KEY=TU_CLAVE_ZEP
```

Si no va a usar un segundo modelo acelerador, elimine por completo las variables LLM_BOOST_API_KEY, LLM_BOOST_BASE_URL y LLM_BOOST_MODEL_NAME.

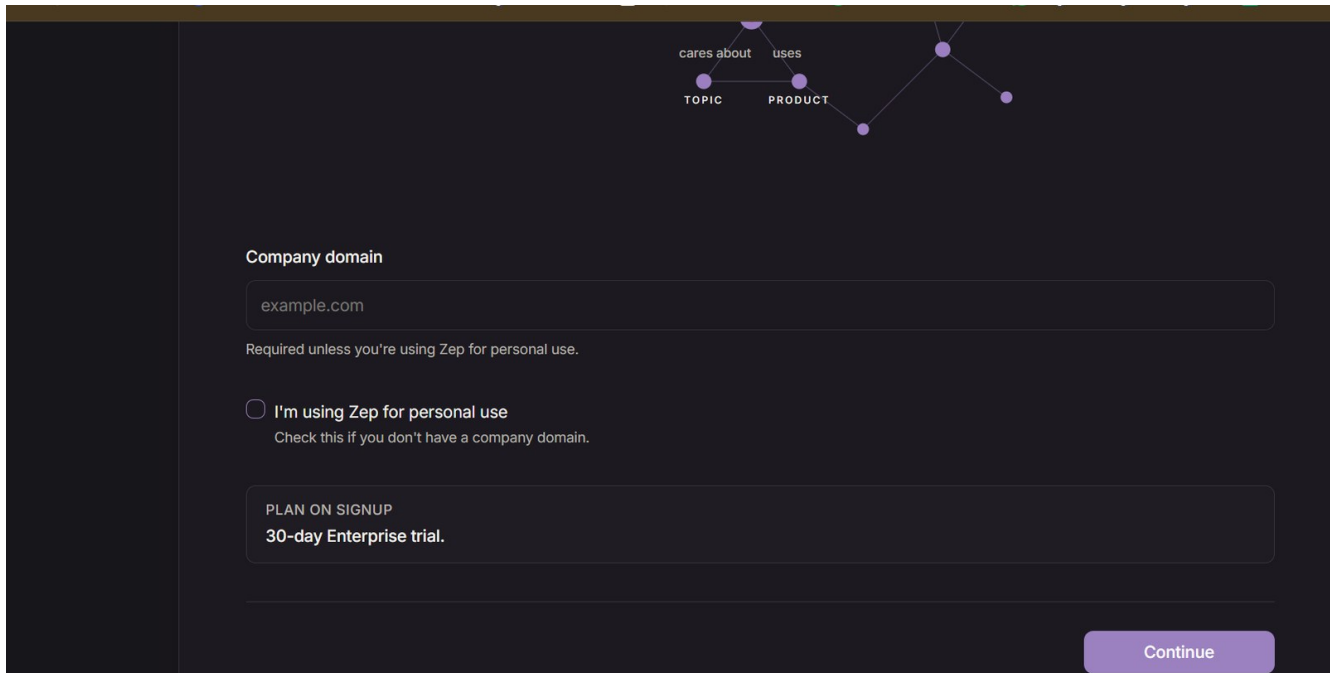
9. Configurar Zep

Zep mantiene memoria y relaciones de los agentes. Para esta instalación se creó una cuenta, un proyecto MiroFish y una API key.

Abra en el navegador: <https://app.getzep.com/>

9.1 Crear o iniciar sesión

1. Abra el enlace anterior en su navegador.
2. Cree una cuenta o inicie sesión.
3. Si pregunta por el dominio de una empresa y el uso será personal, marque “I’m using Zep for personal use” y continúe.



The screenshot shows the Zep onboarding interface. At the top, there is a diagram with nodes labeled 'TOPIC' and 'PRODUCT' connected by lines labeled 'cares about' and 'uses'. Below this, the 'Company domain' section has a text input field containing 'example.com'. A note below the field states: 'Required unless you're using Zep for personal use.' Below this, there is a radio button selection. The first option is 'I'm using Zep for personal use' with a subtext 'Check this if you don't have a company domain.' Below the radio buttons, there is a box labeled 'PLAN ON SIGNUP' containing the text '30-day Enterprise trial.' At the bottom right, there is a purple 'Continue' button.

Figura 1. Zep permite indicar uso personal sin introducir dominio de empresa.

9.2 Omitir el asistente de ejemplo

Si aparece “What are you building?”, no es necesario construir un ejemplo dentro de Zep. MiroFish ya se encargará del uso de memoria.

1. Haga clic en “Skip setup”.

What are you building?

Internal knowledge layer Sales & operations Software engineering agents Customer support

Consumer companion Other

Describe your product, your agent(s), and what you want it to remember

e.g. A support agent for our e-commerce store. It should remember each shopper's orders, preferences, and past issues.

Choose an example to start, then edit it to fit. The description drives your generated plan, graph schema, and sample data.

← Back Skip setup Continue

Figura 2. El asistente inicial de Zep puede omitirse.

9.3 Crear el proyecto

1. En “Create Project”, escriba MiroFish como nombre.
2. En Description puede escribir “Memory and graph storage for MiroFish simulations”.
3. Haga clic en “Create Project”, no en “Create with wizard”.

Create Project

Choose a recognizable name and describe what this project will power. Create with wizard can generate a tailored graph and sample data for you.

< Back Skip tour Next >

Project name

Enter project name

Description

Project description

Create Project Create with wizard

Figura 3. Creación del proyecto MiroFish dentro de Zep.

9.4 Crear y guardar la API key

1. Dentro del proyecto, cree una nueva API key.
2. Use un nombre reconocible, por ejemplo MiroFish.
3. Copie la clave cuando Zep la muestre.
4. Regrese al Bloc de notas con .env y reemplace ZEP_API_KEY=your_zep_api_key_here por su clave real.

5. Guarde con Ctrl+S.

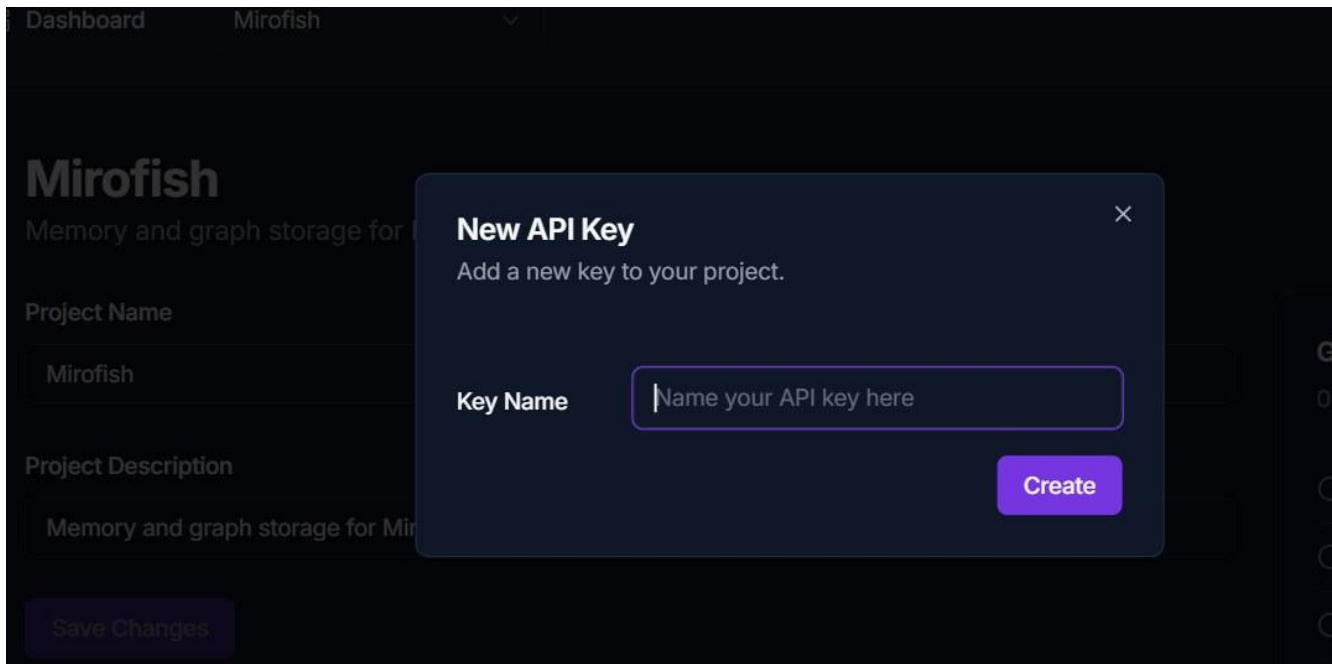


Figura 4. Creación de la API key de Zep.

NO COMPARTA LA CLAVE: Si sospecha que una clave se hizo pública, elimínela o rótele en Zep y actualice .env.

Documentación de Zep: <https://help.getzep.com/v2/quickstart>

10. Configurar Qwen en Alibaba Cloud Model Studio

Qwen será el modelo de lenguaje que MiroFish utilizará para razonar por los agentes. La configuración se hace en Alibaba Cloud Model Studio.

Abra: <https://modelstudio.console.alibabacloud.com/>

10.1 Crear la cuenta de Alibaba Cloud

1. Cree una cuenta o use el método de inicio de sesión disponible.
2. En Country/Region indique su país real. Esto pertenece a la cuenta y a la facturación.
3. Cree la contraseña y complete la verificación solicitada.

PAÍS VS. REGIÓN TÉCNICA: El país de su cuenta no debe falsificarse para obtener una región de API. La región técnica de Model Studio se selecciona después. En esta guía se utilizó Singapore.

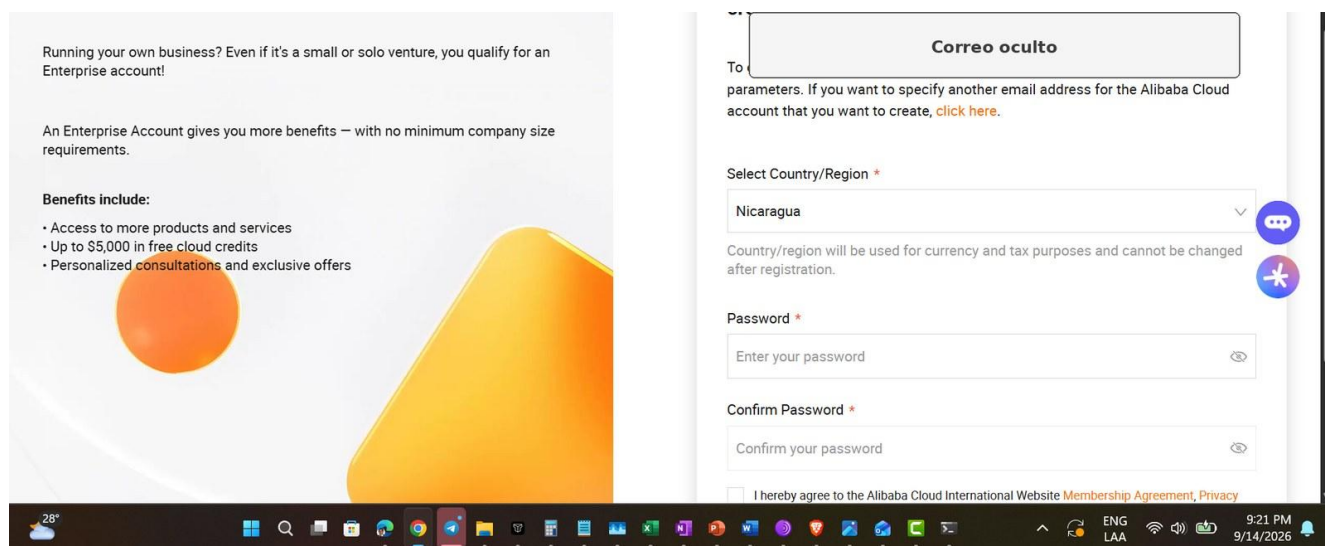


Figura 5. Registro de Alibaba Cloud; el correo se muestra oculto en esta captura.

10.2 Completar la información de cuenta

1. Seleccione Personal Account si corresponde a su caso.
2. Complete nombre, dirección, provincia/estado, ciudad y datos solicitados.
3. Alibaba puede solicitar asociar un método de pago para habilitar servicios pay-as-you-go.

Account created.

Complete sign-up and start your cloud journey!

[Return to Previous Page \[←\]](#)

Security Verification

Payment Information

For business purposes, you can apply for business payments in addition to bank cards (credit or debit cards) and e-wallets. After you enter your account information, our dedicated staff will contact you to assist you in the follow-up process.[>]

Account Type

Note

Please carefully review the differences between account types and select the appropriate one. If you run your own business—even as a solo entrepreneur—we recommend signing up for a business account to enjoy greater benefits and support.

☐ Enterprise Account
 ☒ Personal Account

Billing Information

First Name *

Last Name *

Address Line - 1 *

Address Line - 2

Country/Region

State/Province *

City *

Postal Code

Figura 6. Información de cuenta y facturación.

☐ Apple Pay
Associate your Alibaba Cloud account with your Apple Pay account.

☐ Bank Card
MASTERCARD,VISA,JCB,AMEX credit cards or debit cards are supported.

☐ Google Pay
Associate your Alibaba Cloud account with your Google Pay account.

☒ PayPal
Associate your Alibaba Cloud account with your PayPal account.

Método de pago oculto

PayPal accounts registered in the Chinese mainland are not supported.
Each PayPal account can be linked to only one Alibaba Cloud account.
When you activate pay-as-you-go products and incur charges, Alibaba Cloud pre-authorizes your PayPal account.

Complete →

Figura 7. Ejemplo de autorización de PayPal como método de pago. Los datos sensibles se ocultaron.

COSTO: Vincular un método de pago significa que la cuenta puede facturar uso de pago. Antes de ejecutar MiroFish, active Free Quota Only/Stop-on-Exhaust para el modelo que vaya a utilizar.

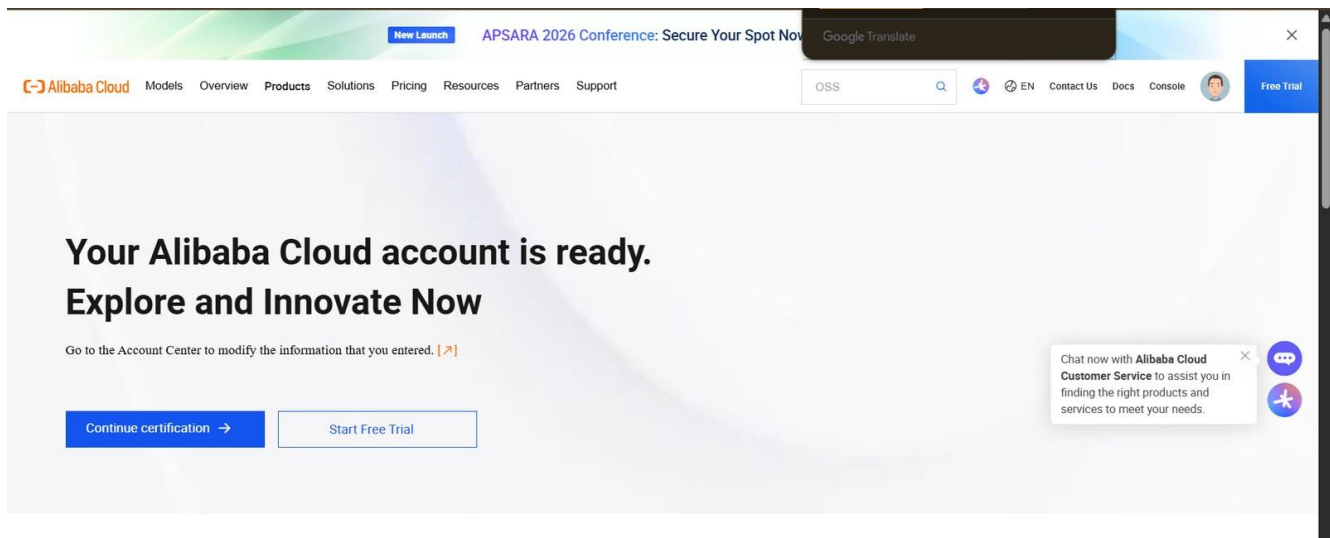


Figura 8. Cuenta de Alibaba Cloud activada.

10.3 Entrar a Model Studio y seleccionar Singapore

1. Abra Alibaba Cloud Model Studio.
2. En el selector de región, elija Singapore para esta guía.
3. Si se solicita activar Model Studio o aceptar términos, complételo.

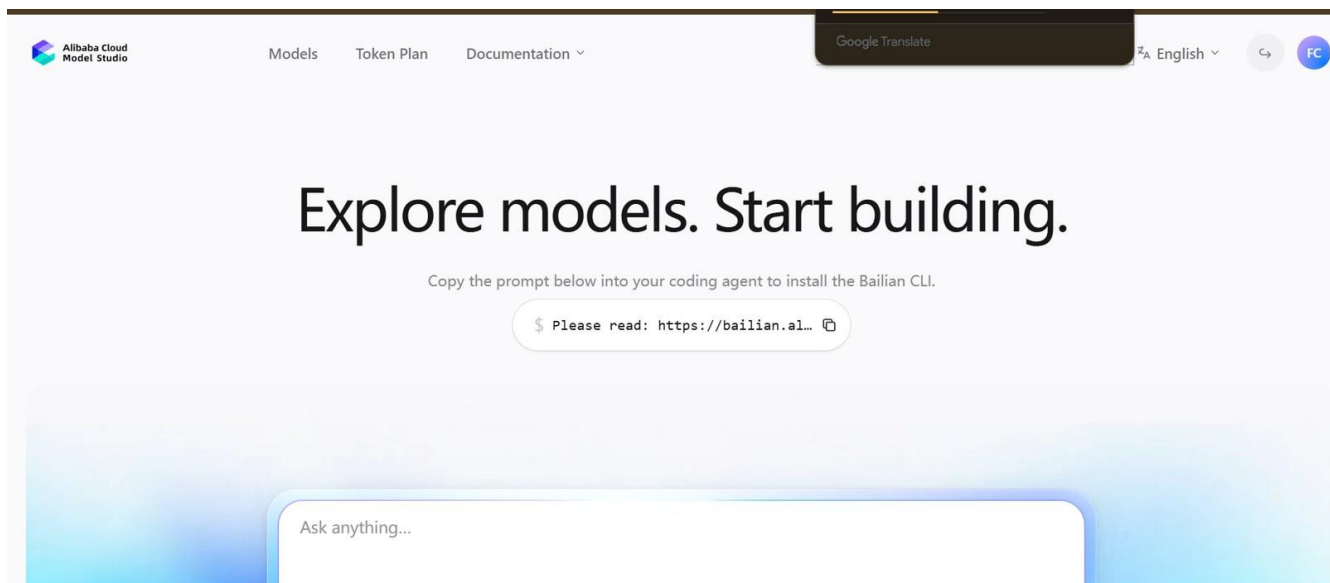


Figura 9. Página principal de Alibaba Cloud Model Studio.

11. Proteger la cuenta contra cargos inesperados

Este paso debe hacerse antes de la primera simulación si quiere limitarse a la cuota gratuita disponible.

1. En Model Studio, abra Usage & Billing.
2. Entre en Free Quota.
3. Busque qwen-plus.
4. Active “Stop-on-Exhaust” o “Free Quota Only” para qwen-plus.
5. Confirme que el interruptor queda activado antes de hacer llamadas desde MiroFish.

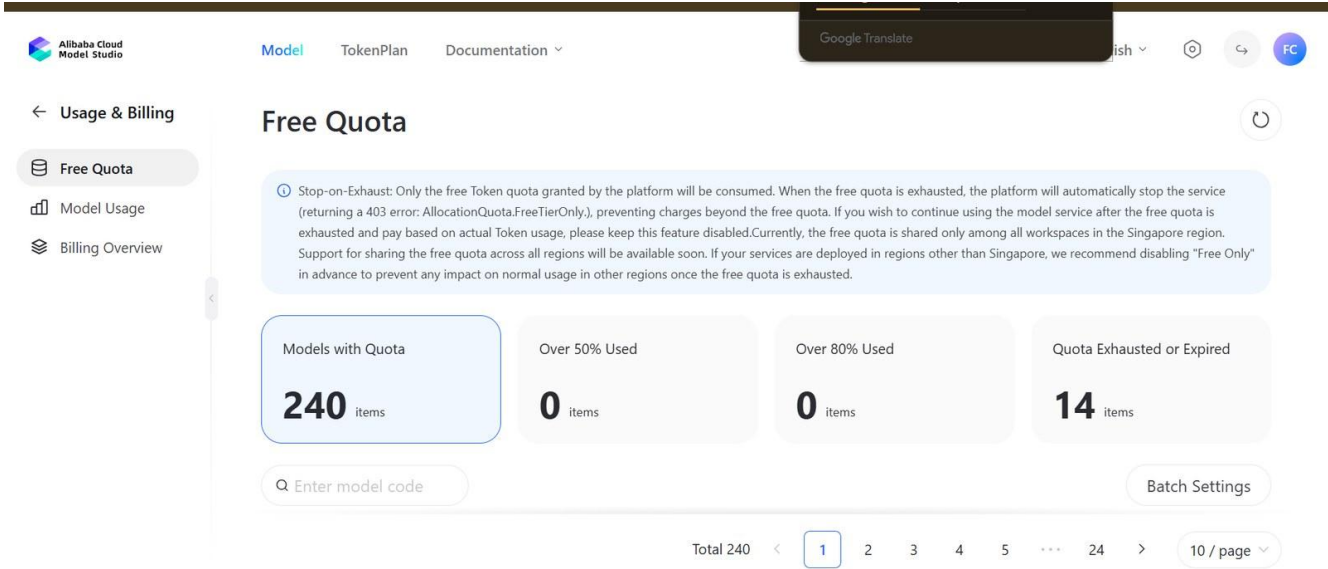


Figura 10. Página Free Quota, donde se explica el modo de detener el servicio al agotar la cuota.

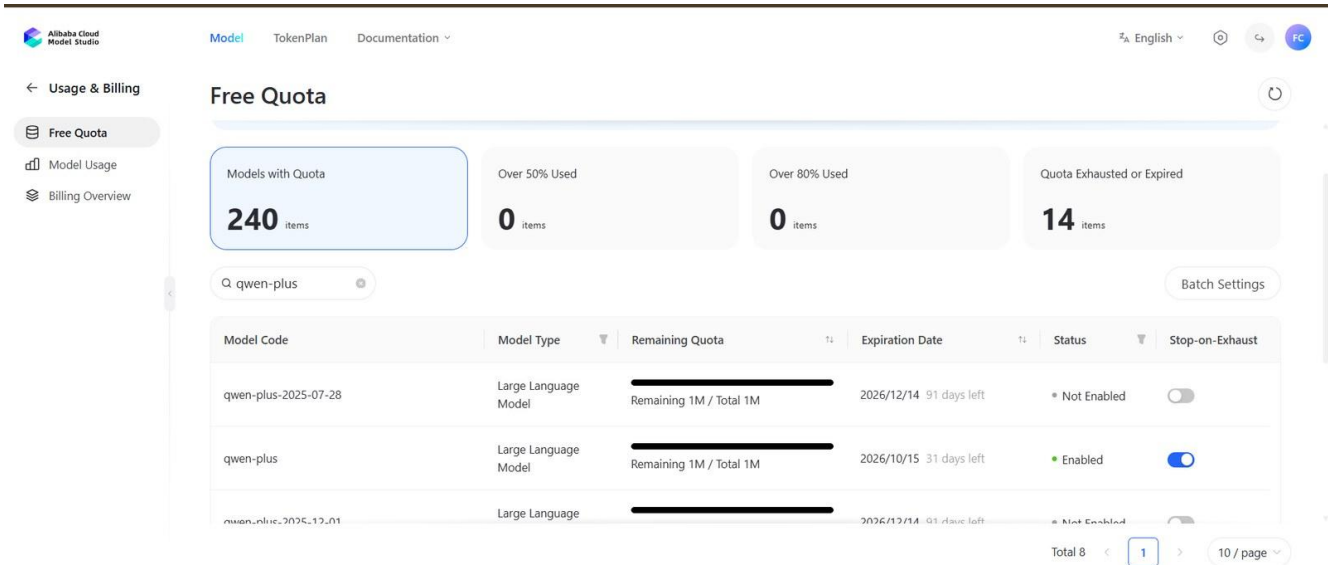


Figura 11. qwen-plus con Stop-on-Exhaust activado.

QUÉ HACE ESTA PROTECCIÓN: Cuando la cuota gratuita se agota, el servicio rechaza nuevas llamadas en vez de seguir consumiendo pay-as-you-go. La cuota y su fecha de vencimiento dependen de la cuenta y pueden cambiar.

Referencia oficial de cuota gratuita: [Alibaba Cloud - Free quota for new users](#)

12. Crear la API Key de Qwen y copiar el endpoint correcto

La API key y el endpoint deben pertenecer a la misma región. Además, Alibaba puede mostrar un dominio específico de workspace. En la instalación documentada, usar ese endpoint fue necesario para resolver el error 401.

1. Abra API Management → API Key dentro de Model Studio.
2. Verifique que arriba figure Singapore.
3. Haga clic en Create API Key.
4. Use el workspace predeterminado y una descripción como MiroFish.
5. Copie la API key cuando se muestre. No la publique ni la envíe por chat.
6. En la misma página, localice el cuadro “OpenAI compatible” y use el icono de copiar para copiar la URL exacta que muestra su workspace.
7. Regrese a .env y complete LLM_API_KEY con la clave y LLM_BASE_URL con la URL copiada.
8. Mantenga LLM_MODEL_NAME=qwen-plus.
9. Guarde .env con Ctrl+S y cierre el Bloc de notas.

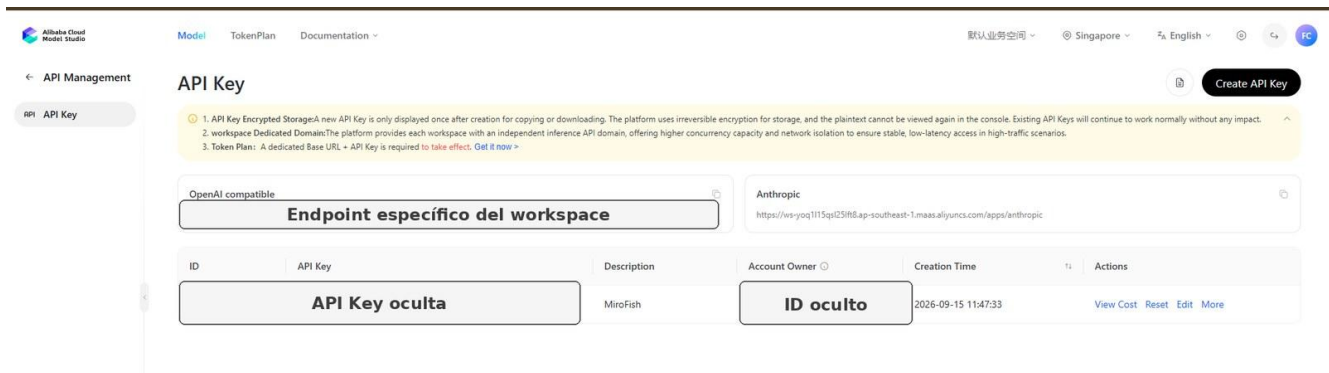


Figura 12. API Management: endpoint OpenAI compatible específico del workspace. API key e identificadores están ocultos.

NO ESCRIBA EL ENDPOINT A MANO: Copiarlo desde el botón de Alibaba reduce el riesgo de errores. Las regiones tienen endpoints y claves diferentes; no mezcle una API key de Singapore con un endpoint de otra región.

Referencia oficial de API key: [Alibaba Cloud - Obtain an API key](#)

Referencia de regiones y endpoints: [Alibaba Cloud - Regions and endpoints](#)

13. Primer arranque de MiroFish

A partir de aquí ya están configuradas todas las piezas externas.

1. Abra PowerShell.

2. Entre en la carpeta del proyecto:

```
cd $HOME\MiroFish
```

3. Inicie frontend y backend:

```
npm run dev
```

4. No cierre esa ventana. Es el proceso que mantiene MiroFish en funcionamiento.

5. Abra Chrome, Edge u otro navegador y escriba:

```
http://localhost:3000
```

DOS DIRECCIONES LOCALES: La interfaz usa localhost:3000. El backend/API trabaja en localhost:5001. “localhost” significa su propia PC; no es una página pública en Internet.



Figura 13. Pantalla de MiroFish ejecutándose localmente en localhost:3000.

14. Crear la primera simulación de prueba

Antes de utilizar documentos extensos o muchas rondas, haga una prueba pequeña. Esto valida que Qwen, Zep, el grafo y el informe funcionen.

14.1 Crear la semilla en Bloc de notas

1. Presione Inicio y abra Bloc de notas.
2. Pegue un texto de prueba breve. En esta guía se usó el siguiente:

PRUEBA DE MIROFISH

La ciudad ficticia de Puerto Claro anunció que, a partir del próximo mes, el centro de la ciudad será peatonal todos los domingos entre las 8:00 a. m. y las 2:00 p. m.

La alcaldía afirma que la medida reducirá la contaminación, favorecerá a las familias y aumentará la actividad recreativa.

La Asociación de Comerciantes está dividida. Algunos restaurantes apoyan la medida porque esperan más visitantes a pie, mientras otros negocios creen que perderán clientes que llegan en automóvil.

Una organización ambiental apoya públicamente la decisión.

Un grupo de residentes se opone porque considera que aumentará el tráfico en los barrios cercanos.

Los conductores de transporte público esperan un aumento de pasajeros.

La población todavía no tiene una posición clara y el anuncio comienza a discutirse en redes sociales.

3. Guarde el archivo como prueba_mirofish.txt en una carpeta fácil de encontrar. Si el Bloc de notas ofrece codificación, utilice UTF-8.

14.2 Cargar la semilla y escribir la predicción

1. En MiroFish, arrastre o seleccione prueba_mirofish.txt en “Semilla de la realidad”.
2. En el cuadro de indicaciones, pegue el prompt de prueba:

Simula la reacción inicial de la población ante esta medida durante las primeras 2 horas después del anuncio.

Identifica qué actores la apoyan o rechazan, los principales argumentos, posibles alianzas inesperadas y cómo cambia el sentimiento público.

No supongas de antemano que la medida tendrá éxito o fracasará.

Esta es una prueba técnica corta. Mantén la simulación pequeña.

3. Pulse “Arranca el motor”.

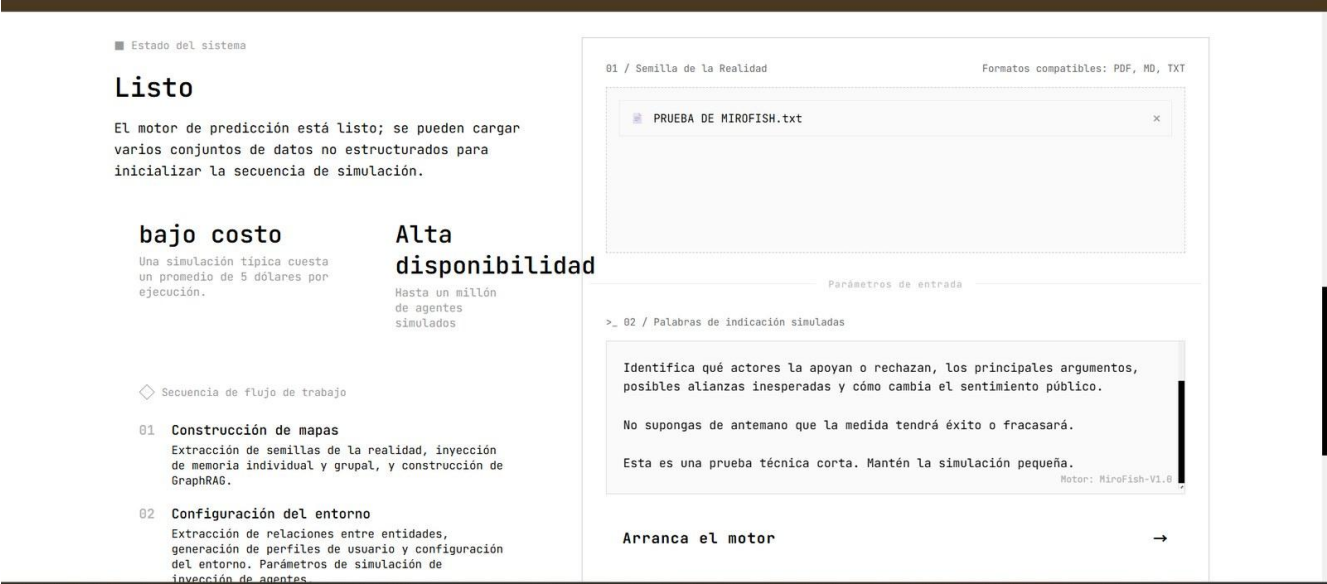


Figura 14. Semilla cargada y prompt de simulación listo.

DURANTE EL PROCESAMIENTO: No presione F5 ni cierre la pestaña durante una simulación activa. Tampoco cierre PowerShell.

15. Solucionar el error HTTP 401

En el primer intento de esta instalación, MiroFish mostró que la solicitud al proveedor LLM falló con HTTP 401. Esto significa que el proveedor rechazó la autenticación.

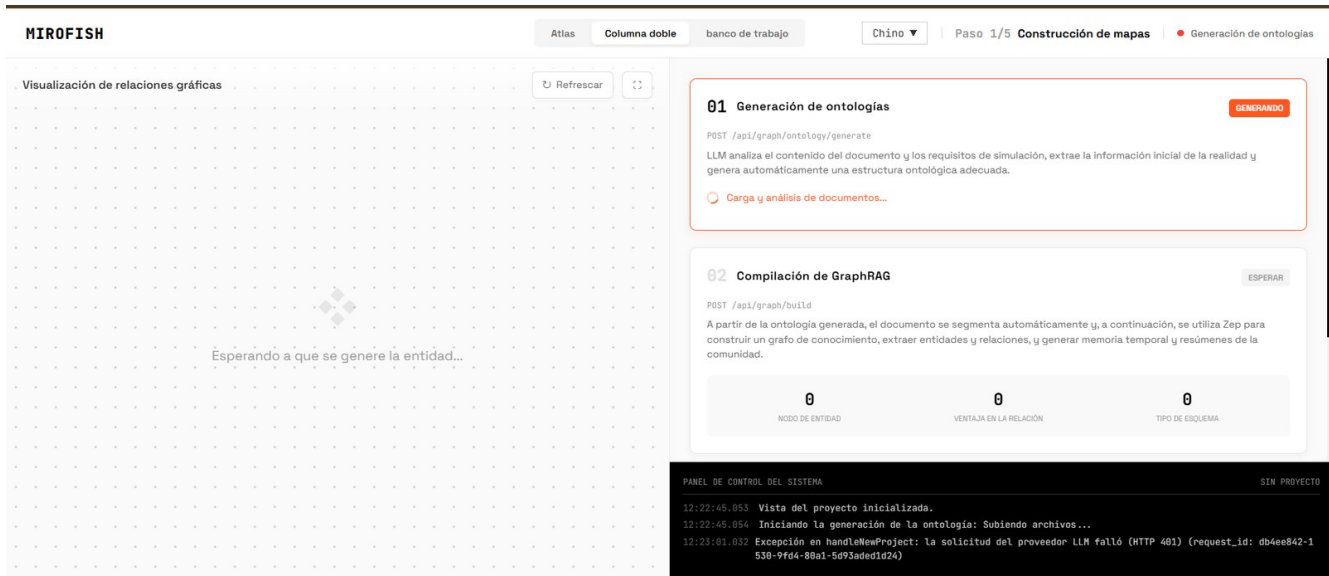


Figura 15. Error HTTP 401 al intentar generar la ontología.

15.1 Qué revisar antes de crear otra clave

1. Detenga MiroFish en la ventana donde se ejecuta `npm run dev` presionando `Ctrl+C`.
2. Si pregunta “Terminate batch job (Y/N)?”, responda Y y presione Enter.
3. Abra `.env`:

```
notepad %HOME%\MiroFish\.env
```
4. Confirme que `LLM_MODEL_NAME` sea `qwen-plus`.
5. Confirme que la API key fue creada en Singapore.
6. En Alibaba API Management, copie de nuevo la URL que aparece bajo “OpenAI compatible”.
7. Reemplace únicamente `LLM_BASE_URL` por esa URL exacta. Mantenga la misma API key si sigue vigente.
8. Guarde `.env`.
9. Vuelva a iniciar:

```
cd %HOME%\MiroFish
npm run dev
```

LECCIÓN DE ESTA PRUEBA: El endpoint genérico no fue suficiente para la clave de workspace usada en la cuenta documentada. La URL de “OpenAI compatible” del propio workspace resolvió la autenticación.

16. Configurar rondas y ejecutar la simulación

Cuando la construcción del grafo termina, MiroFish prepara agentes y muestra la configuración de rondas.

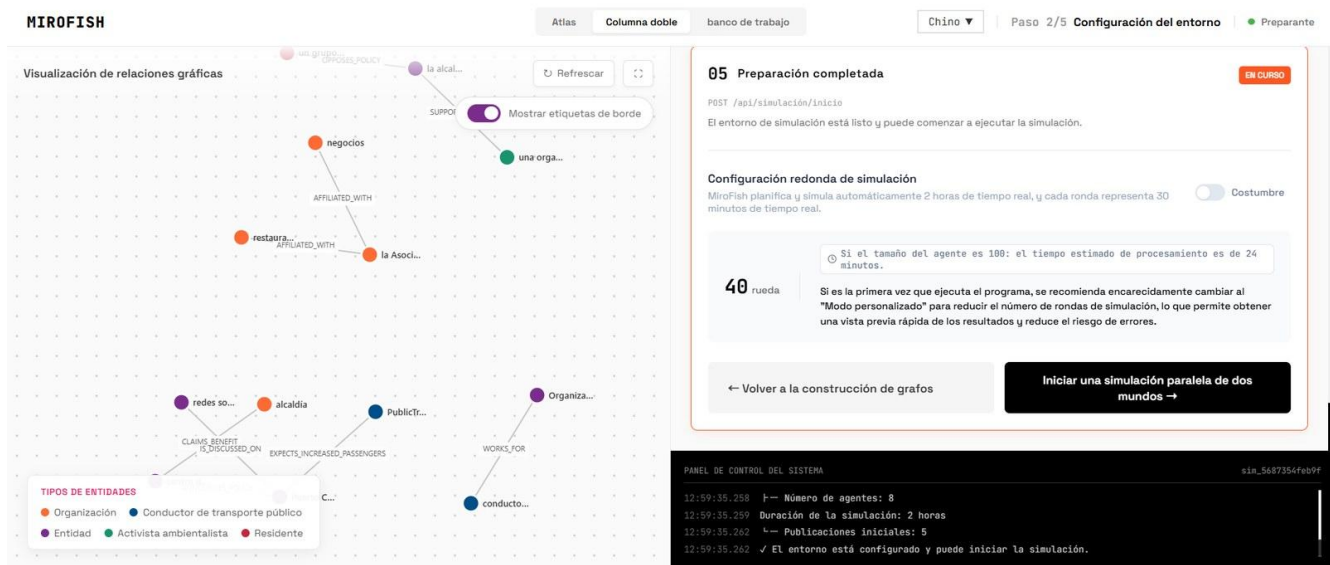


Figura 16. Entorno preparado: agentes detectados y configuración de la simulación.

16.1 Activar el modo personalizado

1. Localice el interruptor Custom. Si el navegador traduce automáticamente la interfaz, puede aparecer como “Costumbre”.
2. Actívelo.
3. En la versión documentada, el mínimo del control era 10 rondas.

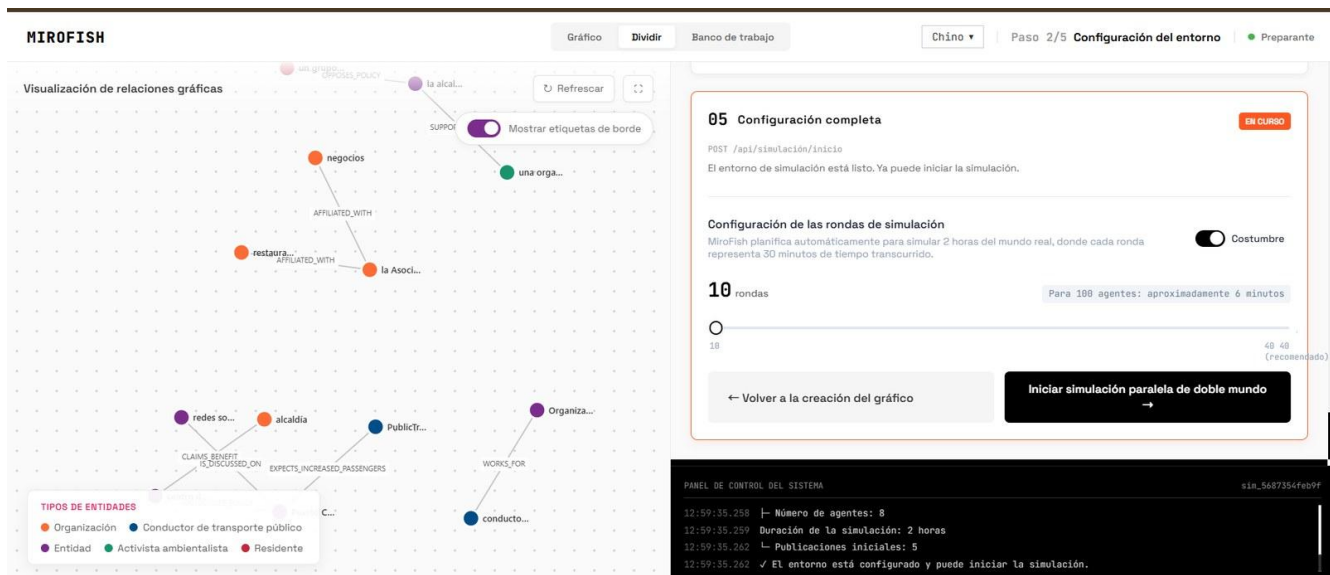


Figura 17. Modo Custom/“Costumbre” activado y control de rondas.

LÍMITE VS. RONDAS EFECTIVAS: El valor del control puede funcionar como máximo. En la prueba de 2 horas, donde cada ronda representó 30 minutos, la ejecución efectiva terminó en 4 rondas por mundo.

4. Haga clic en “Iniciar simulación paralela de doble mundo”.
5. Espere sin recargar la página.

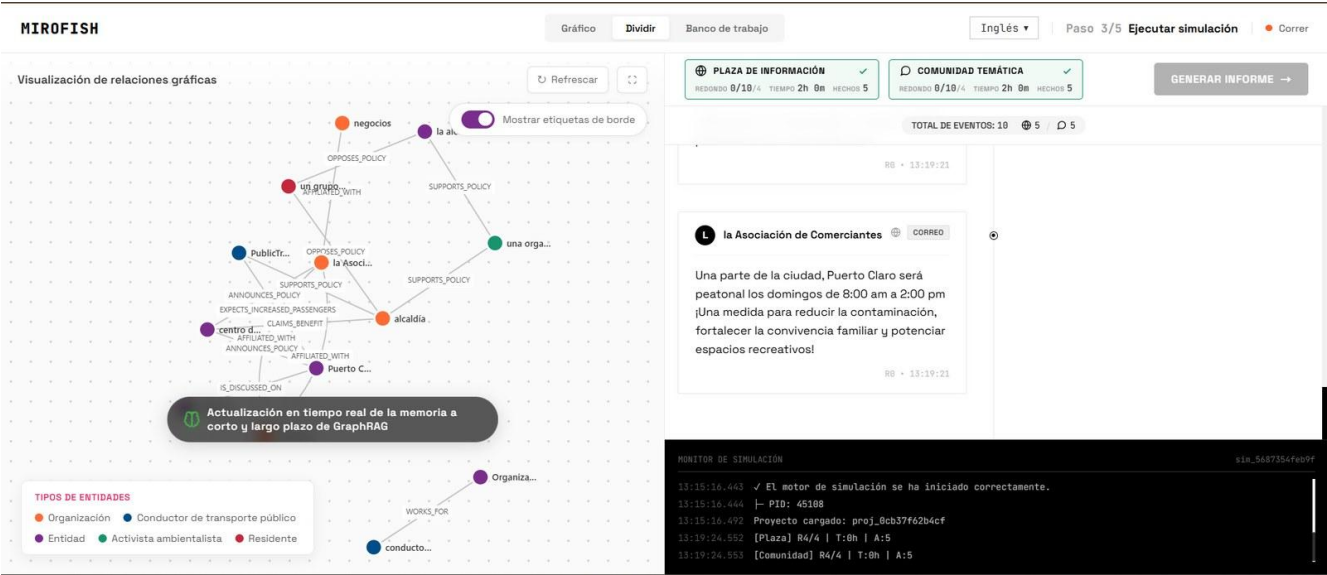


Figura 18. Simulación en ejecución con dos mundos y eventos en tiempo real.

17. Qué hacer si la simulación termina pero queda esperando

En nuestra prueba, ambos mundos llegaron a R4/4, pero “Generar informe” siguió deshabilitado. Después de 40 minutos, revisamos el log en lugar de seguir esperando.

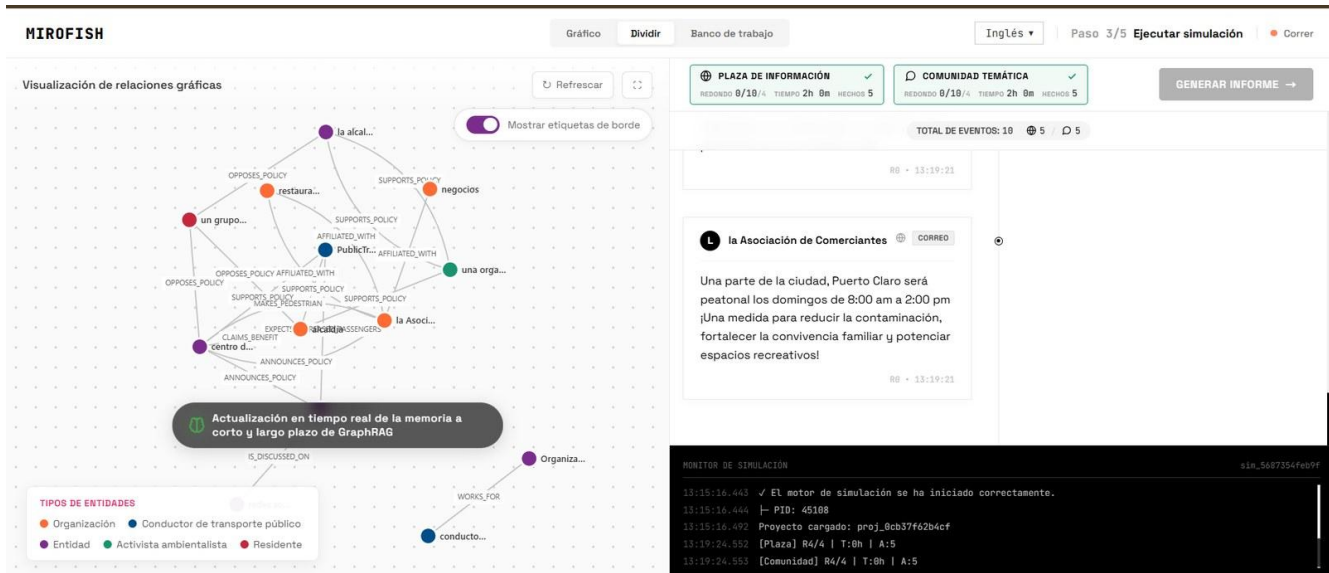


Figura 19. Ambos mundos en R4/4 mientras el entorno continúa activo.

17.1 Abrir una segunda ventana de PowerShell

1. No cierre la ventana donde npm run dev sigue activo.
2. Abra otra ventana de PowerShell desde Inicio.
3. Entre a la carpeta MiroFish:

```
cd $HOME\MiroFish
```

17.2 Identificar el ID de la simulación

La interfaz muestra un identificador parecido a sim_5687354feb9f. El valor cambia en cada ejecución. Copie el suyo y sustitúyalo en los comandos siguientes.

17.3 Leer el log en UTF-8

```
Get-Content ".\backend\uploads\sims\simulations\SIM_ID\simulation.log" -Encoding UTF8 -Tail 40
```

Reemplace SIM_ID por su identificador real. Si el log indica que los ciclos terminaron y el entorno entró en modo de espera de comandos, puede cerrarse de forma ordenada.

17.4 Cerrar el entorno local de forma ordenada

PRECAUCIÓN: Este es un procedimiento de recuperación observado en la versión usada para el manual. Use el ID de su propia simulación. No ejecute este comando mientras las rondas sigan avanzando.

```
Invoke-RestMethod `
-Method Post `
-Uri "http://localhost:5001/api/simulation/close-env" `
-ContentType "application/json" `
-Body '{"simulation_id":"SIM_ID","timeout":30}'
```

Si la operación devuelve `success=True` o equivalente, vuelva al navegador. El botón para generar el informe debería quedar disponible tras actualizar el estado de la interfaz.

18. Generar, abrir y localizar el informe final

1. Cuando “Generar informe” esté habilitado, púlselo.
2. Espere a que MiroFish complete todas las secciones. No cierre PowerShell durante esta fase.
3. Busque en la consola mensajes de “Generación del informe completada” y “Informe guardado”.

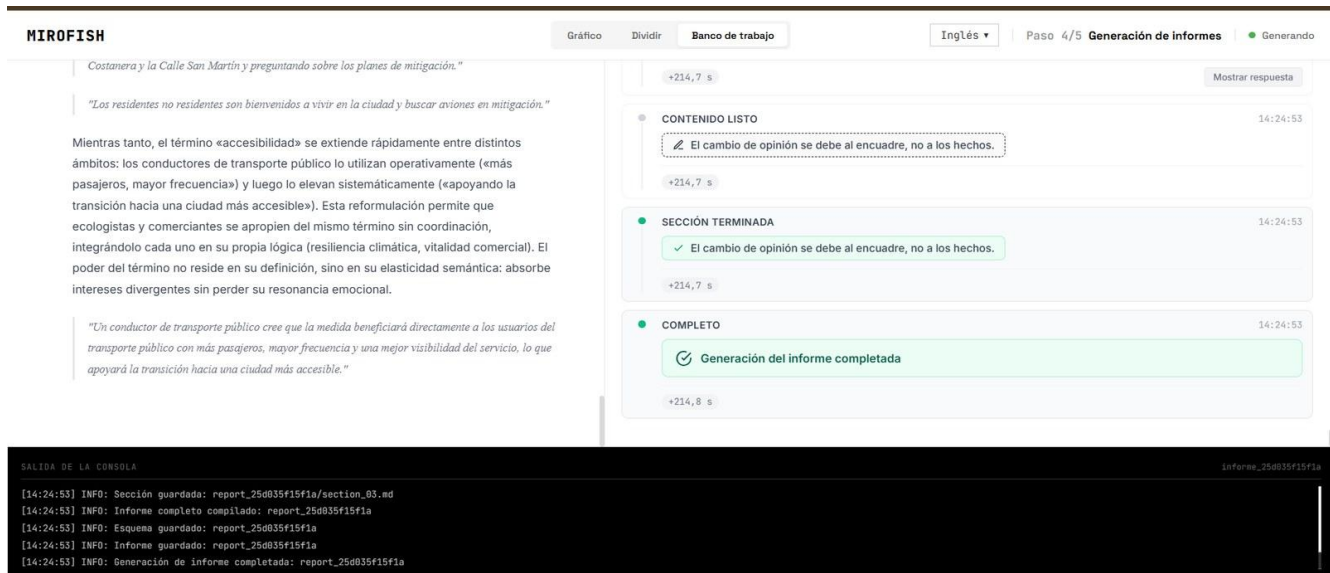


Figura 20. Informe generado y guardado correctamente.

PASO 4/5: Que la interfaz diga Paso 4/5 no significa que falte contenido del informe. El paso 5 es interacción profunda con el mundo simulado y con el ReportAgent.

18.1 Leer el informe en la interfaz

El informe aparece en la columna de lectura. Puede desplazarse para revisar todas las secciones.

18.2 Abrir el archivo `full_report.md` en Windows

Cada informe tiene un ID propio, por ejemplo `report_25d035f15f1a`. Puede verse en la URL del navegador o en la consola.

```
notepad "$HOME\MiroFish\backend\uploads\reports\REPORT_ID\full_report.md"
```

18.3 Abrir la carpeta del informe

```
explorer "$HOME\MiroFish\backend\uploads\reports\REPORT_ID"
```

ARCHIVO PRINCIPAL: `full_report.md` es el informe consolidado. Los otros archivos de la carpeta pueden incluir secciones y esquema generados durante el proceso.

18.4 Qué aprendimos de la prueba Puerto Claro

La simulación mostró apoyos funcionales entre transporte público, ambientalistas y parte del comercio; preocupaciones de residentes por desvío de tráfico; y división interna de comerciantes según dependencia del automóvil. El informe también interpretó que el encuadre del discurso influyó más que la aparición de hechos nuevos durante las primeras dos horas.

19. Detener y reanudar MiroFish

19.1 Detener al terminar

1. Vaya a la ventana donde está ejecutándose `npm run dev`.
2. Presione `Ctrl+C`.
3. Si pregunta “Terminate batch job (Y/N)?”, escriba `Y` y presione `Enter`.

19.2 Reanudar otro día

1. Abra PowerShell.
2. Entre a MiroFish y arranque:

```
cd $HOME\MiroFish
npm run dev
```

3. Abra `http://localhost:3000` en el navegador.

NO REINSTALE: Git, Node.js, uv, Python 3.12, las dependencias y `.env` permanecen guardados. No repita la instalación cada vez que reinicie la PC.

20. Revisar consumo y costos después de una prueba

Antes de una simulación grande, conviene observar cuántos tokens consumió la prueba pequeña.

1. Abra Model Studio en el navegador.
2. Vaya a Usage & Billing → Model Usage.
3. Filtre o localice qwen-plus.
4. Revise consumo de entrada, salida y cuota restante si la consola los muestra.
5. Vuelva a Free Quota y confirme que Free Quota Only/Stop-on-Exhaust sigue activo.

PLANIFICACIÓN: El consumo crece con el número de agentes, las rondas, la longitud de los documentos y el tamaño de las respuestas. Haga pruebas pequeñas antes de aumentar el escenario.

Documentación de uso del modelo: [Alibaba Cloud - Model usage](#)

21. Actualizar MiroFish en el futuro

MiroFish es un proyecto activo. Una actualización puede cambiar dependencias, interfaz o procedimientos. Antes de actualizar, detenga MiroFish.

1. Abra PowerShell y entre al proyecto:

```
cd $HOME\MiroFish
```

2. Haga una copia de seguridad de su configuración:

```
Copy-Item .env .env.backup
```

3. Revise que no haya cambios inesperados:

```
git status
```

4. Descargue los cambios oficiales:

```
git pull
```

5. Vuelva a sincronizar dependencias por si el proyecto cambió sus requisitos:

```
npm run setup:all
```

6. Inicie de nuevo con npm run dev y haga una prueba corta antes de una simulación importante.

SI GIT PULL SE NIEGA: No fuerce la actualización si Git indica cambios locales. Primero revise qué archivos modificó. El archivo .env normalmente está excluido del repositorio, pero otros cambios manuales sí pueden bloquear la actualización.

22. Seguridad de API keys y archivos sensibles

- No publique .env en foros, repositorios ni capturas.
- No copie una API key completa en un chat o correo salvo que el procedimiento de seguridad de su organización lo autorice.
- Si una clave se filtra, rótelas/elimine en el proveedor y actualice .env.
- Antes de compartir capturas de Alibaba, oculte API key, Account Owner, correo, teléfono y datos de pago.
- No suba la carpeta completa de MiroFish a un repositorio público sin revisar archivos sensibles.
- Mantenga Stop-on-Exhaust/Free Quota Only activado si desea evitar consumo de pago una vez agotada la cuota gratuita.

VERIFICACIÓN SENCILLA: Puede ejecutar `git status --short` y confirmar que .env no aparece como archivo listo para subir. Si aparece, deténgase y revise .gitignore antes de hacer cualquier push.

23. Problemas frecuentes

Síntoma	Causa probable	Qué hacer
git/node/npm/uv no se reconoce	No está instalado o PATH no se recargó	Instale la herramienta; cierre y reabra PowerShell; pruebe de nuevo.
python muestra 3.14	Es la versión global de Windows	No la borre. Verifique uv python list y que MiroFish use CPython 3.12 en npm run setup:all.
npm run setup:all muestra warnings	Advertencias de paquetes	Revise si termina con error real. No aplique --force automáticamente.
HTTP 401	API key/endpoint/región no coinciden	Verifique Singapore y copie el endpoint OpenAI compatible exacto del workspace.
HTTP 403 FreeTierOnly	Cuota gratuita agotada y protección activa	Es el comportamiento esperado de Stop-on-Exhaust. No desactive si quiere evitar cargos.
localhost:3000 no abre	npm run dev no está activo o hubo error	Revise la ventana de PowerShell. Manténgala abierta y lea el error del frontend/backend.
Rondas llegaron al final pero informe no se habilita	Entorno quedó esperando comandos	Revise simulation.log y use close-env solo si las rondas ya terminaron.
Caracteres raros en log	Codificación de PowerShell	Use Get-Content con -Encoding UTF8.
No encuentro el informe	Cada ejecución usa un REPORT_ID distinto	Mire la URL/consola y abra backend\uploads\reports\REPORT_ID\full_report.md.
No quiero cargos	Cuenta pay-as-you-go habilitada	Active Free Quota Only/Stop-on-Exhaust y revise Usage & Billing antes de simular.

Apéndice A. Comandos de referencia rápida

Verificar herramientas

```
winget --version  
git --version  
node -v  
npm -v  
python --version  
py --version  
uv --version
```

Instalar herramientas

```
winget install --id Git.Git -e --source winget  
winget install --id OpenJS.NodeJS.LTS -e --source winget  
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"  
uv python install 3.12
```

Descargar e instalar MiroFish

```
cd $HOME  
git clone https://github.com/666ghj/MiroFish.git  
cd MiroFish  
npm run setup:all
```

Crear configuración

```
Copy-Item .env.example .env  
notepad .env
```

Iniciar

```
cd $HOME\MiroFish  
npm run dev
```

Leer log

```
Get-Content ".\backend\uploads\simulations\SIM_ID\simulation.log" -Encoding UTF8 -Tail 40
```

Abrir informe

```
notepad "$HOME\MiroFish\backend\uploads\reports\REPORT_ID\full_report.md"
```

Apéndice B. Enlaces oficiales

MiroFish - repositorio oficial: <https://github.com/666ghj/MiroFish>

Git for Windows: <https://git-scm.com/download/win>

Node.js - descargas: <https://nodejs.org/en/download>

uv - instalación oficial: <https://docs.astral.sh/uv/getting-started/installation/>

Python para Windows (alternativa manual): <https://www.python.org/downloads/windows/>

Microsoft WinGet: <https://learn.microsoft.com/windows/package-manager/winget/>

Zep Cloud: <https://app.getzep.com/>

Zep Quickstart: <https://help.getzep.com/v2/quickstart>

Alibaba Cloud Model Studio: <https://modelstudio.console.alibabacloud.com/>

Alibaba - obtener API key: <https://www.alibabacloud.com/help/en/model-studio/get-api-key>

Alibaba - regiones y endpoints: <https://www.alibabacloud.com/help/en/model-studio/regions>

Alibaba - cuota gratuita: <https://www.alibabacloud.com/help/en/model-studio/new-free-quota>

Alibaba - estadísticas de uso: <https://www.alibabacloud.com/help/en/model-studio/model-usage-statistics>

FECHA DE VERIFICACIÓN: Los enlaces y pantallas se verificaron para esta edición el 15 de septiembre de 2026.
Servicios externos y nombres de menú pueden cambiar con el tiempo.

Checklist final antes de una simulación importante

- ☐ Git, Node.js/npm y uv responden en PowerShell.
- ☐ uv tiene Python 3.12 instalado.
- ☐ npm run setup:all terminó correctamente.
- ☐ .env contiene las claves correctas y no tiene valores de ejemplo.
- ☐ Zep API key está vigente.
- ☐ Qwen API key pertenece a Singapore.
- ☐ LLM_BASE_URL fue copiado del cuadro OpenAI compatible del workspace.
- ☐ qwen-plus tiene Free Quota Only/Stop-on-Exhaust activado si desea evitar cargos.
- ☐ npm run dev está activo y localhost:3000 abre.
- ☐ Se ejecutó una simulación corta de prueba.
- ☐ Se revisó Model Usage antes de aumentar agentes/rondas/documentos.

Fin del manual.

Elaborado por:



Ministerio LD

WhatsApp: +505 8693 9441

Web: LeyDominical.com

Email: info@leydominical.com