

# Guía práctica: GitHub Copilot Chat + Continue (OpenRouter)

Una referencia rápida y clara para combinar **GitHub Copilot Chat** y **Continue** con modelos de **OpenRouter**, maximizando el **contexto real del repositorio** mientras evitas los límites del chat de Copilot Free.

---

## Cómo usarlos juntos

1. **Copilot Chat → visión general y navegación**
  2. Verifica que el **Workspace Index** esté activo (barra de estado → Copilot → *Workspace index*).
  3. Aporta contexto puntual con **#-menciones** a archivos, carpetas o símbolos (`#api/`, `#config.php`, `#save_to_sheet.php`).
  4. **Continue → análisis profundo (RAG del repo)**
  5. Usa `` para recuperar, por embeddings, los archivos más relevantes de todo el workspace.
  6. Acota con `y` cuando quieras centrar el análisis.
  7. Añade `(estructura real)` y `(listado de ficheros/firmas)` para tener visión global.
  8. **Modelos (OpenRouter) en Continue**
  9. Decláralos en `config.yaml` / `config.json` con `apiBase: https://openrouter.ai/api/v1` y tu `apiKey`.
  10. Alterna modelos *free* o de pago según rendimiento.
  11. Ajusta el *recall* de recuperación con `nRetrieve` / `nFinal` para repos grandes.
- 

## Flujo de trabajo recomendado

- **Explorar / arquitectura:** Copilot Chat con `@workspace` y #-menciones para overview rápido.
  - **Refactors y análisis multiarchivo:** Continue con `@Codebase` + `@Tree` + `@repo-map` (y si hace falta `@Folder` / `@File`).
  - **Mantén limpio el índice:** usa `.continueignore` para saltarte binarios y builds; tras cambios grandes, ejecuta **Continue: Rebuild codebase index**.
-

# Advertencia: "Cuota de chat alcanzada" (Copilot Free)

## Qué significa

- En **Copilot Free**, el **chat** está limitado a **50 mensajes/mes** y las **autocompletaciones** a **~2000/mes**. El aviso lo aplica **Copilot Chat**, aunque uses **OpenRouter (BYOK)**. Se restablece automáticamente en tu fecha mensual.

## Cómo actuar

- Continúa la conversación en **Continue** (`@Codebase`, `@Tree`, `@repo-map`, `@Folder`, `@File`).
- Deja Copilot para **completions** en el editor hasta agotar su cupo.
- Si necesitas chat ilimitado en Copilot, valora **actualizar de plan**.

## Comprobaciones útiles

- Barra de estado → Copilot → **Workspace Index** activo.
- **Manage Models** en Copilot: confirma el proveedor/modelo (no quita el límite, pero asegura tu preferencia).

---

## Plantilla de prompts (copiar/pegar)

### 1) Árbol real del repo (ASCII)

Usa `@Tree` o `@Codebase` como pastilla y pega:

```
Devuélveme SOLO el árbol ASCII REAL del workspace, sin comentarios ni adornos.  
- Usa solo ASCII y los caracteres [A-Za-z0-9_./-]  
- No añadas descripciones a la derecha  
- Si el resultado es largo, envíalo en bloques de 300-400 líneas y espera mi "siguiente".
```

### 2) Diagrama Mermaid estable (`graph TD`)

Después del árbol, pide el gráfico así (con `@Tree` aún en contexto):

```
Genera un diagrama Mermaid `graph TD` de la estructura del repo con estas reglas:  
1) Cada nodo usa un **ID seguro** solo con [A-Za-z0-9_].  
2) El texto visible va SIEMPRE entre corchetes: root["root"] --> htaccess[".htaccess"].  
3) No uses comillas para IDs. No escribas "root" --> ".htaccess".  
4) Evita caracteres no ASCII en IDs; reemplázalos por "_".  
5) Si el diagrama excede el límite, devuélvelo en varios bloques
```

consecutivos.  
Devuelve SOLO el bloque ```mermaid ... ``` sin explicaciones.

### 3) Análisis dirigido por carpetas/archivos

@Codebase @Folder api @File assets/js/components/chatbot.js  
Resume los endpoints del backend y del frontend, riesgos (SQLi/CSRF/XSS) y propone fixes.  
Incluye diffs sugeridos y rutas exactas.

### 4) Recuperación con más contexto (Continue)

Añade en tu `config` si el repo es grande:

```
{
  "context": [
    { "provider": "codebase", "params": { "nRetrieve": 200, "nFinal": 60,
      "useReranking": true } },
    { "provider": "folder", "params": { "nRetrieve": 200, "nFinal": 60,
      "useReranking": true } }
  ]
}
```

Tras cambiar ignores o estructura, ejecuta **Continue: Rebuild codebase index**.

## Implementación desde cero (VS Code)

Premisas: cuentas gratuitas en **GitHub**, **Continue Hub** y **OpenRouter** (con API Key).

1. **Instala extensiones** \ VS Code → Marketplace: **GitHub Copilot** y **Continue – open-source AI code assistant**.
2. **Confía el workspace** \ Abre tu repo y marca **Trusted** si VS Code lo solicita.
3. **Configura OpenRouter en Continue** \ Crea `C:/Users/<tu_usuario>/.continue/config.yaml` (o `config.json`) y añade tus **modelos** y **context providers**.
4. **Ajusta recuperación** \ Sube `nRetrieve/nFinal` si el repo es grande; activa `useReranking`.
5. **Ignora ruido** \ Crea `.continueignore` en la raíz (imágenes, binarios, builds...).
6. **Reconstruye índice** \ Paleta → **Continue: Rebuild codebase index**.
7. **Workspace Index en Copilot** \ Barra de estado → Copilot → *Workspace index* (para overview y navegación).
8. **Prueba de humo** \ Continue: `@Tree` → árbol ASCII → Mermaid `graph TD` con IDs seguros. \ Copilot: `@workspace` + `#` -menciones para preguntas rápidas.
9. **Límite de Copilot Free** \ Si aparece “Cuota de chat alcanzada”, continúa el chat en **Continue**.

## Snippets de configuración

`config.yaml` (recomendado)

```
models:
  - title: GLM 4.5 Air (free)
    provider: openrouter
    apiBase: https://openrouter.ai/api/v1
    apiKey: sk-or-... # tu API key
    model: z-ai/glm-4.5-air:free
  - title: Devstral Small (free)
    provider: openrouter
    apiBase: https://openrouter.ai/api/v1
    apiKey: sk-or-...
    model: mistralai/devstral-small-2505:free
  - title: DeepSeek V3 (free)
    provider: openrouter
    apiBase: https://openrouter.ai/api/v1
    apiKey: sk-or-...
    model: deepseek/deepseek-chat-v3-0324:free

contextProviders:
  - name: tree
  - name: repo-map
  - name: codebase
  - name: folder
  - name: file

context:
  - provider: codebase
    params:
      nRetrieve: 200
      nFinal: 60
      useReranking: true
  - provider: folder
    params:
      nRetrieve: 200
      nFinal: 60
      useReranking: true

completionOptions:
  temperature: 0.1
  maxTokens: 8192
```

`config.json` (equivalente)

```
{
  "models": [
    {"title": "GLM 4.5 Air (free)", "provider": "openrouter",
```

```

"apiBase": "https://openrouter.ai/api/v1", "apiKey": "sk-or-...", "model": "z-
ai/glm-4.5-air:free"},
  {"title": "Devstral Small (free)", "provider": "openrouter",
"apiBase": "https://openrouter.ai/api/v1", "apiKey": "sk-or-...",
"model": "mistralai/devstral-small-2505:free"},
  {"title": "DeepSeek V3 (free)", "provider": "openrouter",
"apiBase": "https://openrouter.ai/api/v1", "apiKey": "sk-or-...",
"model": "deepseek/deepseek-chat-v3-0324:free"}
],
"contextProviders": [
  {"name": "tree"}, {"name": "repo-map"}, {"name": "codebase"},
{"name": "folder"}, {"name": "file"}
],
"context": [
  {"provider": "codebase", "params": {"nRetrieve": 200, "nFinal": 60,
"useReranking": true}},
  {"provider": "folder", "params": {"nRetrieve": 200, "nFinal": 60,
"useReranking": true}}
],
"completionOptions": {"temperature": 0.1, "maxTokens": 8192}
}

```

Para añadir o cambiar **modelos** en el futuro: agrega otro bloque en `models` con su *slug* de OpenRouter y selecciónalo en el panel de Continue.

## Checklist diario

- Copilot → **Workspace Index** activo.
- Continue con tu modelo de OpenRouter seleccionado.
- Para mapas: `@Tree` → árbol ASCII → Mermaid `graph TD`.
- Tras cambios de estructura/ignores: **Rebuild codebase index**.

## Atajos de teclado útiles (VS Code)

- **Paleta de comandos:** `Ctrl+Shift+P` (o `F1`).
- **Abrir archivo rápido:** `Ctrl+P`.
- **Buscar en archivos:** `Ctrl+Shift+F`.
- **Ir a símbolo en workspace:** `Ctrl+T`.
- **Renombrar símbolo:** `F2`.
- **Quick Fix / Acciones:** `Ctrl+.`.
- **Peek Definition:** `Alt+F12`.
- **Formatear documento:** `Shift+Alt+F`.
- **Terminal integrada (toggle):** `Ctrl+``.
- **Vista previa Markdown al lado:** `Ctrl+K`, luego `V`.

Tip: usa la paleta para **Continue: Rebuild codebase index** cuando cambies ignores o sumes muchos archivos.

---

## Diagnóstico rápido

1. **Continue no usa tu config** → Asegura `title` en modelos; usa **config.yaml** o **config.json** (uno u otro) y **recarga la ventana**.
  2. **No ve archivos / "no tengo acceso"** → Habilita **Workspace Trust** y abre la carpeta raíz del repo. Revisa `.continueignore` / `.gitignore` (si ignoras una carpeta completa, no podrás re-incluir hijos).
  3. **Contexto pobre** → Sube `nRetrieve/nFinal` y combina pastillas (`@Codebase @Folder api @File assets/js/main.js`). Reconstruye el índice; si persiste, borra `~/.continue/index`.
  4. **Salida se corta** → Aumenta `completionOptions.maxTokens` o pide respuesta en bloques.
  5. **Cuota de chat en Copilot** → Límite del plan Free; sigue en **Continue** para chat. Las **completions** de Copilot seguirán hasta su tope mensual.
- 

## Enlaces base (oficiales)

- Copilot Free – límites por plan: <https://docs.github.com/copilot/concepts/copilot-billing/understanding-and-managing-requests-in-copilot>
- Planes de Copilot: <https://docs.github.com/en/copilot/get-started/plans>
- Workspace Index en VS Code: <https://code.visualstudio.com/docs/copilot/reference/workspace-context>
- Copilot Chat – contexto del workspace: <https://code.visualstudio.com/docs/copilot/chat/copilot-chat-context>
- Tips y paleta de comandos (VS Code): <https://code.visualstudio.com/docs/getstarted/tips-and-tricks>
- Continue – ``: <https://docs.continue.dev/customize/context/codebase>
- Continue – Context Providers: <https://docs.continue.dev/customize/context-providers>
- Continue – Selección de contexto: <https://docs.continue.dev/chat/context-selection>
- OpenRouter – Quickstart (endpoint base): <https://openrouter.ai/docs/quickstart>
- OpenRouter – API Reference: <https://openrouter.ai/docs/api-reference/overview>