

Suscríbete a DeepL Pro para poder editar este documento.  
Entra en [www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document) para más información.

Características

* Vista del gráfico Git:
  + Visualización:
    - Sucursales locales y remotas
    - Refs. locales Cabezas, etiquetas y mandos a distancia
    - Cambios no comprometidos
  + Realizar acciones Git (disponibles haciendo clic con el botón derecho del ratón en una confirmación / rama / etiqueta):
    - Crear, retirar, eliminar, recuperar, fusionar, tirar, empujar, volver a basar, renombrar y restablecer ramas
    - Añadir, eliminar y empujar etiquetas
    - Checkout, Cherry Pick, Drop, Merge & Revert Commits
    - Limpiar, reiniciar y almacenar los cambios no comprometidos
    - Aplicar, crear rama de, soltar y reventar alijos
    - Ver los detalles de las etiquetas anotadas (nombre, correo electrónico, fecha y mensaje)
    - Copiar los hashes de las confirmaciones y los nombres de las ramas, stash y etiquetas en el portapapeles
  + Vea los detalles de la confirmación y los cambios en los archivos haciendo clic en una confirmación. En la vista de detalles de la confirmación puede:
    - Vea el Visual Studio Code Diff de cualquier cambio de archivo haciendo clic en él.
    - Abra la versión actual de cualquier archivo que haya sido afectado en la confirmación.
    - Copie en el portapapeles la ruta de cualquier archivo que haya sido afectado en la confirmación.
    - Haga clic en cualquier url HTTP/HTTPS del cuerpo de la confirmación para abrirla en su navegador web predeterminado.
  + Compare dos confirmaciones cualesquiera haciendo clic en una confirmación y, a continuación, haciendo CTRL/CMD en otra confirmación. En la vista de comparación de confirmaciones puede:
    - Vea el Visual Studio Code Diff de cualquier cambio de archivo entre los commits seleccionados haciendo clic en él.
    - Abre la versión actual de cualquier archivo que haya sido afectado entre los commits seleccionados.
    - Copiar en el portapapeles la ruta de cualquier archivo que se haya visto afectado entre las confirmaciones seleccionadas.
  + Revisión del código - Mantenga un registro de los archivos que ha revisado en las vistas de detalles de confirmación y comparación.
    - La revisión del código puede realizarse en cualquier confirmación, o entre dos confirmaciones cualesquiera (no en los cambios no comprometidos).
    - Cuando se inicia una Revisión de Código, todos los archivos que necesitan ser revisados aparecen en negrita. Cuando vea el diff/abra un archivo, éste no estará en negrita.
    - Las Revisiones de Código persisten a través de las sesiones de Visual Studio Code. Se cierran automáticamente después de 90 días de inactividad.
  + Ver los cambios no comprometidos, y comparar los cambios no comprometidos con cualquier confirmación.
  + Pase el ratón por encima de cualquier vértice comprometido en el gráfico para ver una información sobre la herramienta que indica:
    - Si la confirmación se incluye en el HEAD.
    - Qué ramas, etiquetas y stashes incluye la confirmación.
  + Filtra las ramas mostradas en Git Graph utilizando el menú desplegable "Ramas". Las opciones para filtrar las ramas son:
    - Mostrar todas las sucursales
    - Seleccione una o varias ramas para verlas
    - Seleccionar de una matriz predefinida por el usuario de patrones glob personalizados (estableciendo git-graph.customBranchGlobPatterns)
  + Recuperar de la(s) remota(s) *(disponible en la barra de control superior)*
  + El widget de búsqueda le permite encontrar rápidamente uno o más commits que contengan una frase específica (en el mensaje de commit / fecha / autor / hash, rama o nombres de etiqueta).
  + Widget de configuración del repositorio:
    - Permite ver, añadir, editar, eliminar, recuperar y podar remotas del repositorio.
    - Configurar la "Vinculación de la cuestión" - Convierte los números de la cuestión en los mensajes de confirmación en hipervínculos, que abren la cuestión en su sistema de seguimiento de la cuestión.
    - Configurar "Creación de Pull Request" - Automatiza la apertura y el prellenado de un formulario de Pull Request, directamente desde el menú contextual de una rama.
      * La compatibilidad con los proveedores públicos de Bitbucket, GitHub y GitLab Pull Request está integrada.
      * Los proveedores de Pull Request personalizados pueden ser configurados utilizando la configuración de la extensión git-graph.customPullRequestProviders (por ejemplo, para su uso con proveedores de Pull Request privados). La información sobre cómo configurar los proveedores personalizados está disponible [aquí](https://github.com/mhutchie/vscode-git-graph/wiki/Configuring-a-custom-Pull-Request-Provider).
    - Exporta la configuración de tu repositorio Git Graph a un archivo que se puede confirmar en el repositorio. Permite que otras personas que trabajen en el mismo repositorio utilicen automáticamente la misma configuración de Git Graph.
  + Atajos de teclado (disponibles en la vista del gráfico Git):
    - CTRL/CMD + F: Abre el widget de búsqueda.
    - CTRL/CMD + H: Desplaza la vista del gráfico de Git para centrarse en la confirmación referenciada por HEAD.
    - CTRL/CMD + R: Actualizar la vista del gráfico Git.
    - CTRL/CMD + S: Desplaza la vista del gráfico de Git al primer (o siguiente) alijo de los commits cargados.
    - CTRL/CMD + SHIFT + S: Desplaza la vista del gráfico de Git hasta el último (o anterior) alijo de los commits cargados.
    - Cuando la vista de detalles de la confirmación está abierta en una confirmación:
      * Arriba / Abajo: La vista de detalles de la confirmación se abrirá en la confirmación directamente encima o debajo de ella en la vista del gráfico de Git.
      * CTRL/CMD + Arriba / CTRL/CMD + Abajo: La vista de detalles del commit se abrirá en su commit hijo o padre en la misma rama.
        + Si también se pulsa la tecla Shift (es decir, CTRL/CMD + SHIFT + Arriba / CTRL/CMD + SHIFT + Abajo), cuando se encuentran ramas o fusiones se sigue la rama alternativa.
    - Intro: Si hay un diálogo abierto, al pulsar Intro se envía el diálogo, realizando la acción principal (izquierda).
    - Escapar: Cierra el diálogo activo, el menú contextual o la vista de detalles de confirmación.
  + Redimensionar el ancho de cada columna, y mostrar/ocultar las columnas de Fecha, Autor y Compromiso.
  + Los Emoji Shortcodes comunes se sustituyen automáticamente por el emoji correspondiente en los mensajes de confirmación (incluyendo todos los [gitmoji](https://gitmoji.carloscuesta.me/)). Las asignaciones de Emoji Shortcode personalizadas se pueden definir en git-graph.customEmojiShortcodeMappings.
* Una amplia gama de ajustes configurables (por ejemplo, el estilo del gráfico, los colores de las ramas, y más...). Para más información, consulte la sección "Ajustes de la extensión".
* Botón de lanzamiento de "Git Graph" en la barra de estado
* "Git Graph: Ver gráfico Git" en la paleta de comandos

Configuración de la extensión

[Aquí](https://github.com/mhutchie/vscode-git-graph/wiki/Extension-Settings) encontrará información detallada de todos los ajustes de Git Graph, incluyendo: descripciones, capturas de pantalla, valores por defecto y tipos.

Un resumen de la configuración de la extensión Git Graph es:

* **Vista de los detalles del compromiso**:
  + **Centrar automáticamente**: Centra automáticamente la vista de detalles de la comisión cuando se abre.
  + **Vista de archivos**:
    - **Árbol de archivos**:
      * **Compactar carpetas**: Renderiza el árbol de archivos en la vista de detalles de la confirmación en una forma compactada, de tal manera que las carpetas con una sola carpeta hija se comprimen en un solo elemento de carpeta combinado.
    - **Tipo**: Establece el tipo de vista de archivo por defecto utilizado en la vista de detalles de la confirmación.
  + **Ubicación**: Especifica dónde se muestra la vista de detalles de confirmación en la vista de gráfico de Git.
* **Visibilidad de las acciones del menú contextual**: Personalice qué acciones del menú contextual son visibles. Para más información, consulte la documentación [aquí](https://github.com/mhutchie/vscode-git-graph/wiki/Extension-Settings#context-menu-actions-visibility).
* **Patrones Globales de Ramas Personalizadas**: Una matriz de Patrones Globales Personalizados que se mostrarán en el desplegable de "Ramas". Ejemplo: [{"name": "Feature Requests", "glob": "heads/feature/\*"}]
* **Asignaciones personalizadas de Emoji Shortcode**: Una matriz de mapeos de Emoji Shortcode personalizados. Ejemplo: [{"shortcode": ":sparkles:", "emoji":"✨"}]
* **Proveedores de Pull Request personalizados**: Un conjunto de proveedores de Pull Request personalizados que se pueden utilizar en la Integración "Creación de Pull Request". Para obtener información sobre cómo configurar este ajuste, consulte la documentación [aquí](https://github.com/mhutchie/vscode-git-graph/wiki/Configuring-a-custom-Pull-Request-Provider).
* **La fecha**:
  + **Formato**: Especifica el formato de la fecha que se utilizará en la columna "Fecha" en la vista del gráfico de Git.
  + **Tipo**: Especifica el tipo de fecha que se mostrará en la columna "Fecha" de la vista de gráfico de Git, ya sea la fecha de autor o de confirmación.
* **Visibilidad de la columna por defecto**: Un objeto que especifica la visibilidad por defecto de las columnas Fecha, Autor y Compromiso. Ejemplo: {"Fecha": true, "Autor": true, "Commit": true}
* **Diálogo > \***: Establecer las opciones por defecto en los siguientes diálogos: Añadir etiqueta, Aplicar Stash, Cherry Pick, Crear rama, Eliminar rama, Obtener en la rama local, Obtener en la rama remota, Fusionar, Sacar Stash, Sacar rama, Rebase, Reiniciar y Almacenar cambios no comprometidos
* **Accesibilidad mejorada**: Indicadores visuales de cambio de archivo A|M|D|R|U en la vista de detalles del commit para usuarios con daltonismo. En el futuro, este ajuste activará cualquier característica adicional de Git Graph relacionada con la accesibilidad que no esté activada por defecto.
* **Codificación de archivos**: La codificación del conjunto de caracteres que se utiliza cuando se recupera una versión específica de los archivos del repositorio (por ejemplo, en la vista Diff). Puede encontrar una lista de todas las codificaciones soportadas [aquí](https://github.com/ashtuchkin/iconv-lite/wiki/Supported-Encodings).
* **Gráfico**:
  + **Colores**: Especifica los colores utilizados en el gráfico.
  + **Estilo**: Especifica el estilo del gráfico.
  + **Cambios no comprometidos**: Especifica cómo se muestran los cambios no comprometidos en el gráfico.
* **Terminal Integrado Shell**: Especifica la ruta y el nombre de archivo del ejecutable de Shell que utilizará el Terminal Integrado de Visual Studio Code, cuando sea abierto por Git Graph.
* **Atajo de teclado > \***: Configura las combinaciones de teclas utilizadas para todos los atajos de teclado en la vista Git Graph.
* **Markdown**: Analiza y representa un subconjunto de reglas de formato Markdown en línea que se utilizan con frecuencia en los mensajes de confirmación y en los detalles de las etiquetas (negrita, cursiva, negrita y cursiva, y bloques de código en línea).
* **Profundidad máxima de búsqueda de repositorios**: Especifica la profundidad máxima de las subcarpetas a buscar cuando se descubren repositorios en el espacio de trabajo.
* **Abrir nueva pestaña del grupo de editores**: Especifica el grupo de editores en el que Git Graph debe abrir nuevas pestañas, al realizar las siguientes acciones desde la vista de Git Graph: Ver la vista de diferencias de código de Visual Studio, Abrir un archivo, Ver un archivo en una revisión específica.
* **Abrir al repositorio del documento activo del Editor de Textos**: Abre la vista Git Graph al repositorio que contiene el documento activo del Editor de Texto.
* **Etiquetas de referencia**:
  + **Alineación**: Especifica cómo se alinean las etiquetas de referencia de rama y etiqueta para cada confirmación.
  + **Combinar etiquetas de rama local y remota**: Combina las etiquetas de rama local y remota si se refieren a la misma rama, y están en el mismo commit.
* **Repositorio**:
  + **Se compromete**:
    - **Obtención de avatares**: Obtén los avatares de los autores de las confirmaciones y de los confirmantes.
    - **Carga inicial**: Especifica el número de commits a cargar inicialmente.
    - **Cargar más**: Especifica el número de commits adicionales a cargar cuando se pulsa el botón "Cargar más commits", o se cargan automáticamente más commits.
    - **Cargar más automáticamente**: Cuando la vista se ha desplazado hasta el fondo, carga automáticamente más commits si existen (en lugar de tener que pulsar el botón "Cargar más commits").
    - **Silencio**:
      * **Compromisos que no son ancestros de HEAD**: Muestra los compromisos que no son ancestros de la rama/compromiso comprobado con un color de texto apagado.
      * **Confirmaciones de fusión**: Muestra las confirmaciones de fusión con un color de texto apagado.
    - **Orden**: Especifica el orden de los commits en la vista gráfica de Git. Ver [git log](https://git-scm.com/docs/git-log#_commit_ordering) para más información sobre cada opción de orden.
    - **Mostrar el estado de la firma**: Muestra el estado de la firma del commit a la derecha del committer en la vista de detalles del commit (sólo para commits firmados). Al pasar por encima del icono de la firma se muestra una información sobre herramientas con los detalles de la firma.
  + **Recuperar y eliminar**: Antes de obtener información de la(s) remota(s) utilizando el botón Fetch en la barra de control de la vista de Git Graph, elimine cualquier referencia de seguimiento remoto que ya no exista en la(s) remota(s).
  + **Recuperar y eliminar etiquetas**: Antes de obtener las etiquetas de las remotas utilizando el botón Fetch de la barra de control de la vista Git Graph, elimine cualquier etiqueta local que ya no exista en las remotas.
  + **Incluir confirmaciones mencionadas por reflogs**: Incluir los commits sólo mencionados por reflogs en la vista de gráfico de Git (sólo se aplica cuando se muestran todas las ramas).
  + **En carga**:
    - **Desplazamiento a la cabecera**: Desplaza automáticamente la vista del gráfico de Git para centrarse en la confirmación referenciada por HEAD.
    - **Mostrar la rama comprobada**: Muestra la rama check out cuando se carga un repositorio en la vista Git Graph.
    - **Mostrar ramas específicas**: Mostrar ramas específicas cuando se carga un repositorio en la vista Git Graph.
  + **Seguir sólo al primer padre**: Sólo sigue al primer padre de los commits cuando descubre los commits a cargar en la vista Git Graph. Consulte [--first-parent](https://git-scm.com/docs/git-log#Documentation/git-log.txt---first-parent) para saber más sobre esta configuración.
  + **Mostrar los commits sólo referenciados por etiquetas**: Mostrar los commits a los que sólo se hace referencia mediante etiquetas en Git Graph.
  + **Mostrar ramas remotas**: Mostrar ramas remotas en Git Graph por defecto.
  + **Mostrar cabezas remotas**: Mostrar referencias simbólicas de HEAD remotas en el gráfico Git.
  + **Mostrar Stashes**: Mostrar Stashes en Git Graph por defecto.
  + **Mostrar etiquetas**: Mostrar etiquetas en Git Graph por defecto.
  + **Mostrar cambios no comprometidos**: Mostrar los cambios no comprometidos. Si trabajas en repositorios grandes, deshabilitar este ajuste puede reducir el tiempo de carga de la vista gráfica de Git.
  + **Mostrar archivos sin seguimiento**: Mostrar los archivos no rastreados al ver los cambios no comprometidos. Si trabajas en repositorios grandes, deshabilitar este ajuste puede reducir el tiempo de carga de la vista gráfica de Git.
  + **Firma**:
    - **Confirmaciones**: Activa la firma de commits con GPG o X.509.
    - **Etiquetas**: Activa la firma de etiquetas con GPG o X.509.
  + **Utilizar Mailmap**: Respetar los archivos [.mailmap](https://git-scm.com/docs/git-check-mailmap#_mapping_authors) cuando se muestren los nombres y direcciones de correo electrónico de los autores y compromisarios.
* **Orden de Despliegue de Repositorios**: Especifica el orden en el que se ordenan los repositorios en el desplegable de repositorios en la Vista Gráfica de Git (sólo visible cuando existe más de un repositorio en el Espacio de Trabajo de Visual Studio Code actual).
* **Conservar el contexto cuando se oculta**: Especifica si el contexto de la vista Git Graph de Visual Studio Code se mantiene cuando el panel ya no está visible (por ejemplo, cuando se mueve a la pestaña de fondo). Si se activa esta opción, Git Graph se cargará mucho más rápido al volver a la pestaña de Git Graph, aunque tiene una mayor sobrecarga de memoria.
* **Mostrar elemento de la barra de estado**: Muestra un elemento de la barra de estado que abre la vista del gráfico de Git cuando se hace clic en él.
* **Ubicación de la integración del proveedor de código fuente**: Especifica dónde aparece la acción "Ver gráfico Git" en el título de los Proveedores SCM.
* **Tema de color del icono de la pestaña**: Especifica el tema de color del icono mostrado en la pestaña Git Graph.

Esta extensión consume las siguientes configuraciones:

* git.path: Especifica la ruta y el nombre de archivo de una instalación portátil de Git.

Comandos de extensión

Esta extensión aporta los siguientes comandos:

* git-graph.view: Git Graph: Ver Git Graph
* git-graph.addGitRepository: Git Graph: Add Git Repository... *(se utiliza para añadir sub-repositorios a Git Graph)*
* git-graph.clearAvatarCache: Git Graph: Borrar caché de avatar
* git-graph.endAllWorkspaceCodeReviews: Git Graph: Finalizar todas las revisiones de código en el espacio de trabajo
* git-graph.endSpecificWorkspaceCodeReview: Git Graph: EndSpecificWorkspaceCodeReview: Git Graph: Terminar una revisión de código específica en el espacio de trabajo... *(se utiliza para terminar una revisión de código específica sin tener que abrirla primero en la vista de Git Graph)*
* git-graph.fetch: Git Graph: Fetch from Remote(s) *(se utiliza para abrir la vista de Git Graph y ejecutar inmediatamente "Fetch from Remote(s)")*
* git-graph.removeGitRepository: Git Graph: Remove Git Repository... *(se utiliza para eliminar repositorios de Git Graph)*
* git-graph.resumeWorkspaceCodeReview: Git Graph: Reanudar una revisión de código específica en el espacio de trabajo... *(se utiliza para abrir la vista de Git Graph a una revisión de código que ya está en curso)*
* git-graph.version: Git Graph: Obtener información de la versión

Notas de publicación

Las notas de la versión detalladas están disponibles [aquí](https://github.com/mhutchie/vscode-git-graph/blob/HEAD/CHANGELOG.md).

Mercado de Visual Studio

Esta extensión está disponible en el [Visual Studio Marketplace](https://marketplace.visualstudio.com/items?itemName=mhutchie.git-graph) para Visual Studio Code.

Agradecimientos

Gracias a todos los colaboradores que ayudan al desarrollo de Git Graph.

Algunos de los iconos utilizados en Git Graph proceden de las siguientes fuentes, ¡apoyadlas por su excelente trabajo!

* [GitHub Octicons](https://octicons.github.com/) ([Licencia](https://github.com/primer/octicons/blob/master/LICENSE))
* [Iconos8](https://icons8.com/icon/pack/free-icons/ios11) ([licencia](https://icons8.com/license))