

**COPILOT4ECLIPSE**

**ADOLFO SANZ DE DIEGO**

**DICIEMBRE 2024**

# INTRODUCCIÓN

Copilot4Eclipse - Adolfo Sanz De Diego - Diciembre 2024

## ¿QUE ES COPILOT4ECLIPSE?

- Es un plugin para Eclipse que integra las funcionalidades de codificación asistida por IA de GitHub Copilot directamente en tu IDE de Eclipse.

# SUGERENCIAS DE CÓDIGO

- Genera sugerencias de código en tiempo real mientras escribes en tu editor de Eclipse.
- Estas sugerencias se muestran en línea con tu código habitual, estilizadas como “texto fantasma” para diferenciarlas visualmente del texto normal.
- Puedes aceptar con **TAB** o **CTRL+DCHA** o rechazar con **ESC**.

# CHAT

- Herramienta conversacional similar a ChatGPT para discusiones técnicas con el agente de Chat de GitHub Copilot.
- Es una herramienta flexible que te ayuda a comprender rápidamente el código en tu proyecto, solucionar problemas de codificación y generar nuevo código y documentación.

# INTERFAZ DE USUARIO

- **Barra de Estado:** Proporciona una visión rápida del estado de Copilot4Eclipse.
- **Menú y menú contextual:** Ofrecen acceso a estados, comandos y preferencias, así como a recursos en línea.

# SUGERENCIAS DE CÓDIGO

# PROCESO DE SUGERENCIAS DE CÓDIGO

- **Copilot4Eclipse** usa el contexto del editor actual junto con los editores abiertos cercanos para solicitar sugerencias de código a través de una conexión cifrada TLS desde GitHub Copilot.
- Las sugerencias de código generadas por GitHub Copilot se devuelven a **Copilot4Eclipse**, donde se muestran como “texto fantasma” en la ubicación del cursor.
- Puedes aceptar con **TAB** o **CTRL+DCHA** o rechazar con **ESC**.



## EL “TEXTO FANTASMA”

- Las sugerencias de código de tipo “texto fantasma” se muestran de forma efímera en el editor justo a la derecha del cursor.
- El estilo predeterminado es un texto gris claro en cursiva.

## MOSTRAR SUGERENCIAS AUTOMÁTICAMENTE

- **Copilot4Eclipse** está configurado para activar automáticamente las sugerencias de código mientras escribes.
- El proceso de solicitud y la visualización de las sugerencias ocurre en unos pocos milisegundos después de dejar de escribir.

## SOLICITAR SUGERENCIAS DE CÓDIGO MANUALMENTE

- Puedes solicitar manualmente las sugerencias de código usando la acción del menú **Copilot > Show completion**.

# **IDENTIFICAR VISUALMENTE LA ACTIVIDAD DE SUGERENCIAS DE CÓDIGO**

- La actividad de generación de sugerencias se muestra en la barra de estado de Eclipse.
- Durante el proceso de solicitud, se muestra un mensaje “Copilot gen...” junto con un pequeño indicador de progreso visual.

# SUGERENCIAS DE CÓDIGO Y ECLIPSE CONTENT ASSIST SIMULTÁNEAMENTE

- **Copilot4Eclipse** permite mostrar simultáneamente las sugerencias de código inline y el asistente de contenido de eclipse. Usa las teclas **Tab** y **ESC** para aceptar o rechazar la sugerencias de Copilot.

## TRABAJANDO CON MÚLTIPLES SUGERENCIAS

- GitHub Copilot genera típicamente entre 0 y 3 sugerencias por solicitud. La barra de estado de **Copilot4Eclipse** muestra el número de sugerencias disponibles y el índice de la sugerencias actual.

## **DESHABILITAR SUGERENCIAS GLOBALMENTE O POR LENGUAJE**

- Puedes deshabilitar la generación de sugerencias de Copilot de forma global o para el lenguaje de programación del editor actual.

## **HABILITAR LA GENERACIÓN DE SUGERENCIAS DE COPILOT**

- Si la generación de sugerencias se ha deshabilitado, puedes habilitarla nuevamente desde el menú de Copilot > Habilitar sugerencias.



# CONSEJOS PARA GENERAR MEJORES SUGERENCIAS DE CÓDIGO

# GENERALES

- Utiliza **comentarios específicos** y **consultas claras** para obtener sugerencias más precisas. Por ejemplo, en lugar de escribir un comentario vago como “# ordenar esta lista”, sé más específico: “# ordenar esta lista de nombres alfabéticamente”.
- Usa **nombres descriptivos** para variables y funciones. Esto permite que Copilot entienda mejor el contexto y proporcione sugerencias más relevantes. Por ejemplo, utiliza `temperaturaEnCelsius` en lugar de `temp`.

# REFINAMIENTO ITERATIVO

- **Refina tu código** iterativamente. Si la sugerencia inicial no es perfecta, puedes agregar más contexto o modificar ligeramente el código. Replantear tu comentario o cambiar el nombre de una variable puede llevar a una mejor sugerencia.

## CARACTERÍSTICA DE AUTOCOMPLETADO

- **Aprovecha la función de autocompletado.** GitHub Copilot no solo genera bloques de código, sino que también completa líneas de código. Comienza a escribir y Copilot intentará completarlo por ti, lo cual es útil especialmente para código repetitivo o cuando usas bibliotecas con las que no estás completamente familiarizado.

# REVISIÓN Y PRUEBAS

- **Revisa y prueba** las sugerencias. Aunque Copilot es preciso, siempre es importante revisar y entender el código generado. Asegúrate de que sigue las mejores prácticas, se ajusta al contexto de tu proyecto y no introduce vulnerabilidades de seguridad. Prueba siempre el código para confirmar que funciona como se espera.

# RESUMEN

- GitHub Copilot es una herramienta para **asistir y aumentar** tu codificación, no para reemplazar tu conocimiento sobre el código que escribes.

# USO DEL CHAT

# INTRODUCCIÓN

- **Copilot4Eclipse** trae las características de Chat de GitHub Copilot a la plataforma Eclipse, utilizando el modelo GPT-4 de OpenAI especializado en programación y DevOps.
- Con un tamaño de contexto de aproximadamente 8,192 tokens, permite respuestas más relevantes y contextualizadas.



## PANEL DE CHAT

- Para ver el Panel de Chat, usa el menú principal de **Copilot4Eclipse** o el menú de la barra de estado para seleccionar “**Open Chat Panel**”.
- También puedes abrir el Panel de Chat desde el menú contextual (clic derecho) en la mayoría de los editores de texto de Eclipse.

# CONSEJOS PARA PROMPTS EFECTIVOS (I)

- **Sea Específico:** Cuanto más específico sea su prompt, más específica será la respuesta. Si es vago o ambiguo, la IA podría no entender exactamente lo que está pidiendo.
- **Proporcione Contexto:** Si su pregunta se relaciona con una pieza específica de código o un concepto de programación específico, proporcione ese contexto en su prompt. Esto puede ayudar a la IA a generar una respuesta más precisa.

## CONSEJOS PARA PROMPTS EFECTIVOS (II)

- **Use Terminología Correcta:** Utilice la terminología de programación correcta en sus prompts. Esto puede ayudar a la IA a entender mejor su pregunta y proporcionar una respuesta más precisa.
- **Pida lo que Desea:** Si desea un fragmento de código, pídale. Si desea una explicación de un concepto, pídale. Sea claro sobre lo que está solicitando.

## CONSEJOS PARA PROMPTS EFECTIVOS (III)

- **Experimente:** Si no está obteniendo la respuesta que desea, intente reformular su prompt o hacer su pregunta de una manera diferente. Diferentes prompts pueden generar diferentes respuestas de la IA.
- **Use Oraciones Completas:** Aunque no siempre es necesario, usar oraciones completas a veces puede ayudar a la IA a entender mejor su prompt.

## CONSEJOS PARA PROMPTS EFECTIVOS (IV)

- **Evite Jerga:** A menos que sea necesario para su pregunta, evite usar jerga en sus prompts. Esto puede hacer que su pregunta sea más clara y más fácil de entender para la IA.
- **Manténgalo Breve:** Los prompts largos y complejos pueden ser más difíciles de entender para la IA. Trate de mantener sus prompts breves y al grano.

## COMANDOS SLASH (I)

- El Chat proporciona un conjunto de comandos predefinidos conocidos como **comandos slash** que puede usar para solicitar a Chat.
- Puede ver los comandos slash disponibles en el Panel de Entrada ingresando un '/' o desde el menú contextual de **Copilot4Eclipse** dentro de la mayoría de los editores de texto de Eclipse.

## COMANDOS SLASH (II)

- **/test**: Genera casos de prueba para la porción de código seleccionada.
- **/simplify**: Simplifica la porción de código seleccionada.
- **/fix**: Arregla problemas y errores de compilación.
- **/explain**: Explica el código seleccionado en términos simples.
- **/doc**: Crea documentación para el código seleccionado.

# GESTIÓN DEL CONTEXTO DE LA CONVERSACIÓN

- El contexto de una conversación de Chat es toda la información que tiene sobre la conversación. Consiste en su historial de prompts, respuestas de Chat, perfil de usuario, información del IDE, el código fuente del editor activo y el texto seleccionado, metainformación del proyecto y cualquier archivo del proyecto adjunto como referencia. Este contexto ayuda a generar respuestas que sean relevantes y útiles para su solicitud.
- Puede agregar y eliminar referencias de archivos al contexto, así como eliminar prompts ejecutados previamente y sus respuestas asociadas.



## ELIMINACIÓN DE UN PROMPT

- Puede eliminar cualquier prompt de usuario y su respuesta de Chat de una conversación pasando el cursor sobre el mensaje del usuario que desea eliminar y haciendo clic en el botón 'X'.

## AGREGAR UNA ARCHIVO AL CONTEXTO

- **Desde el Panel de Chat:** Haz clic en el botón (+) en el Panel de Chat para abrir el diálogo de selección de archivos. Selecciona uno o más archivos y confirma. Las referencias aparecerán en la parte inferior del Panel de Chat.
- **Desde el Menú Contextual del Editor:** En el Project Explorer o Package Explorer, selecciona los archivos que deseas añadir al contexto de Chat, haz clic derecho y elige '\*\*\*\*Copilot4Eclipse\*\* > Reference File(s) in the Chat Panel\*\*'.

## **ELIMINACIÓN DE UNA REFERENCIA DE ARCHIVO**

- Para eliminar una referencia de archivo, haz clic en la 'X' junto al nombre de la referencia en el Panel de Chat.

## PROPORCIONAR RETROALIMENTACIÓN

- Puedes proporcionar retroalimentación calificando una respuesta con los botones de pulgar hacia arriba o hacia abajo. Esta información ayuda a mejorar la precisión futura de Chat.

## CREACIÓN DE UNA NUEVA CONVERSACIÓN

- Al abrir el Panel de Chat, se crea una nueva conversación. Si deseas iniciar una nueva conversación en cualquier momento, haz clic en el botón '+' en la barra de herramientas del Panel de Chat.

## **BORRAR MENSAJES**

- Si el Área de Mensajes está saturada, puedes eliminar rápidamente todos los mensajes sin afectar la conversación y el contexto haciendo clic en el botón 'Borrar Mensajes' en la barra de herramientas del Panel de Chat.

# GENERACIÓN DE MENSAJES DE COMMIT

# INTRODUCCIÓN

- Analiza y resume los cambios en tus archivos preparados para el commit.
- Utiliza la vista de Git Staging para facilitar la creación de mensajes de commit.



## GENERANDO UN MENSAJE DE COMMIT

- Abrir la Vista de Git Staging  
`Window > Show View > Other > Git > Git Staging`
- Selecciona y prepara los archivos que deseas incluir en el commit.
- Haz clic en el botón de **Copilot4Eclipse** en la barra de herramientas, justo encima del área de texto del mensaje de commit.

## REVISAR Y MODIFICAR EL MENSAJE DE COMMIT

- El texto generado aparecerá en el área de texto del mensaje de commit.
- Es recomendable revisar y, si es necesario, refinar el mensaje para que cumpla con los estándares de tu equipo.
- Junto al botón “**Generate**”, hay un nuevo botón “**Clear**” que permite eliminar rápidamente el mensaje actual.

## CONSEJOS PARA UN USO EFECTIVO

- El mensaje generado es un punto de partida útil.
- Es buena práctica revisarlo y ajustarlo según los estándares de tu equipo.
- Si añades más archivos a la preparación, haz clic nuevamente en el botón de generación de mensajes para asegurar que el mensaje refleje los cambios más recientes.

# FUNCIONAMIENTO INTERNO

- **Copilot4Eclipse** utiliza Copilot Chat para generar un breve mensaje de commit para cada archivo preparado, resumiendo sus diferencias individuales.
- Si tienes múltiples archivos preparados, **Copilot4Eclipse** utiliza el análisis de Chat para combinar estos mensajes individuales en un solo resumen cohesivo del commit.

## **LIMITACIONES A TENER EN CUENTA**

- Las descripciones de diferencias de archivos que superen los 20.000 caracteres se abrevian a esa longitud.
- Se pueden procesar un máximo de 100 archivos preparados.

# CONCLUSIÓN

- La función de generación de mensajes de commit agiliza el proceso y te ayuda a mantener historiales de commits informativos y claros.