

# Programación con IA Generativa

## Sugerencias de Código

# Índice

1. Tipos de sugerencia
2. Sugerencia de finalización de código
3. Sugerencia de próxima edición (NES)

# **1 | Tipos de Sugerencia**

```

1 session_start();
2 ob_start();
3 include 'sys/togen.php';
4 include 'init.php';
5
6
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <meta content="text/html"/>
11 <meta charset="<?=$SESSION['CMSConfig']['charset']>" />
12 <meta http-equiv="Content-Language" content="<?=$SESSION['CMSConfig']['language']>" />
13 <meta http-equiv="Content-Type" content="text/html; charset="<?=$SESSION['CMSConfig']['charset']>" />
14 <!-- Basic Meta -->
15 <title><?=$SESSION['CMSConfig']['title']> - '.$CMSPage['name']>
16 <meta name="description" content="<?=$SESSION['CMSConfig']['description']>" />
17 <meta name="keywords" content="<?=$SESSION['CMSConfig']['keywords']>" />
18 <meta name="author" content="<?=$SESSION['CMSConfig']['author']>" />
19 <meta name="owner" content="<?=$SESSION['CMSConfig']['owner']>" />
20 <meta name="robots" content="<?=$SESSION['CMSConfig']['robots']>" />
21 <meta name="distribution" content="Global" />
22 <link rel="shortcut icon" href="media/images/favicon.png" />
23 <?php
24 isset($REQUEST['admin']) || isset($REQUEST['administration'])
25 ? $themeName='default'
26 : $themeName=$SESSION['CMSConfig']['theme'];
27 $path='media/themes/'.$themeName.'/mobile';
28 if($isMobile && is_dir($path))
29 print '<meta name="viewport" content="width=device-width, initial-scale=1" />';
30 ?>
31 <!-- OpenGraph Meta -->
32 <meta property="og:title" content="<?=$SESSION['CMSConfig']['title']>" />
33 <meta property="og:description" content="<?=$SESSION['CMSConfig']['description']>" />
34 <meta property="og:type" content="<?=$SESSION['CMSConfig']['type']>" />
35 <meta property="og:image" content="media/images/feature.png" />
36 <meta property="og:url" content="<?=$current_page_url()>" />
37 <!-- CMS Files -->
38 <script src="sys/jquery-1.11.1.min.js" type="text/javascript" />
39 <script src="sys/prog.js" type="text/javascript" />
40 <script src="lib/ckeditor/ckeditor.js" type="text/javascript" />
41 <link rel="stylesheet" type="text/css" href="sys/style.css" />

```



## Sugerencia de finalización de código

Sugerencias de codificación a medida que escribimos. También podemos describir algo en lenguaje natural dentro de un comentario y el asistente sugerirá el código para lograr tu objetivo.



## Sugerencias de próxima edición de código existente (NES)

En función del código que estemos editando (modificando), el asistente predecirá el próximo código que estaremos editando (que vamos a modificar).

# **1 | Sugerencia de finalización de código**

## Sugerencia de finalización de código

Sugerencias de codificación a medida que escribimos. Por ejemplo, si escribimos este encabezado de función en un archivo python:



```
def calculateDaysBetweenDates(begin, end):
```

El asistente nos sugerirá automáticamente el resto de la función. Para aceptar la sugerencia, debemos presionar la tecla **Tab**.

## Sugerencia de finalización de código

Podemos describir algo que deseamos hacer pero usando **lenguaje natural** dentro de un comentario, y el asistente nos sugerirá el código para lograr tu objetivo. Por ejemplo



```
# escribir una función que  
# reciba dos fechas  
# calcule la diferencia entre ambas fechas  
# devuelva el resultado en días
```

El asistente nos sugerirá automáticamente todo el código necesario. Para aceptar la sugerencia, debemos presionar la tecla **Tab**.

## Sugerencia de finalización de código

Para cualquier entrada dada, el asistente puede ofrecer **múltiples sugerencias**. Podemos seleccionar qué sugerencia usar o rechazar. Por ejemplo, podemos escribir este encabezado de función y presionar **Enter** en lugar de **Tab**:



```
def calculateDaysBetweenDates(begin, end):
```

Ahora, **pasar el cursor sobre la sugerencia** para mostrar el control del asistente para elegir sugerencias. Para mostrar la sugerencia siguiente o anterior, haga clic en el botón de la flecha hacia adelante o hacia atrás en el control.



Accept

Tab

Accept Word

Ctrl

+

RightArrow





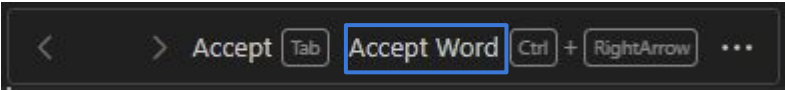
## Sugerencia de finalización de código

Para cualquier entrada dada, el asistente puede ofrecer **múltiples sugerencias**. Podemos seleccionar qué sugerencia usar o rechazar. Por ejemplo, podemos escribir este encabezado de función y presionar **Enter** en lugar de **Tab**:



```
def calculateDaysBetweenDates(begin, end):
```

Ahora, **pasar el cursor sobre la sugerencia** para mostrar el control del asistente para elegir sugerencias. Para mostrar la sugerencia siguiente o anterior, haga clic en el botón de la flecha hacia adelante o hacia atrás en el control.



< > Accept Tab Accept Word Ctrl + RightArrow ...

Aceptar sugerencia parcial, es decir, palabra a palabra

## Sugerencia de finalización de código

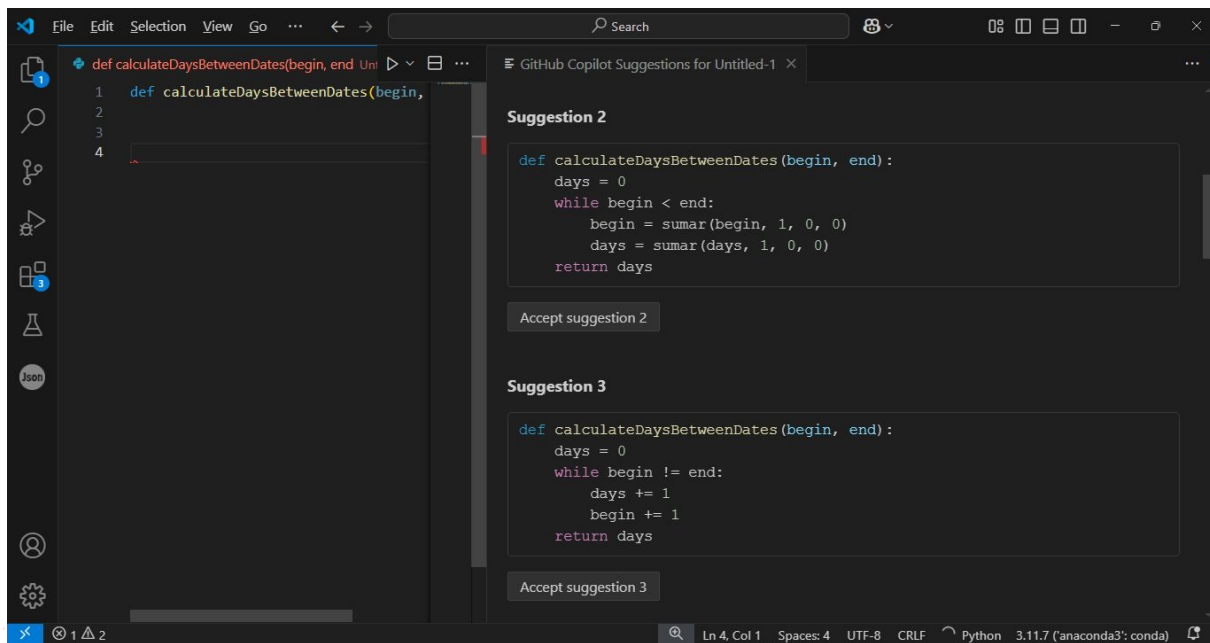
Podemos **mostrar múltiples sugerencias pero en una pestaña aparte**. Por ejemplo, podemos escribir este encabezado de función y presionar **Ctrl+ Enter** para abrir una nueva pestaña con varias opciones adicionales.



```
def calculateDaysBetweenDates(begin, end):
```

# Sugerencia de finalización de código

Para aceptar una sugerencia, debemos Aceptar **Suggestion NÚMERO** . Para rechazar todas las sugerencias, cerramos la pestaña.



## **2 | Sugerencia de próxima edición (NES)**

# Habilitar sugerencia de edición (NES)

## 1 Configuración

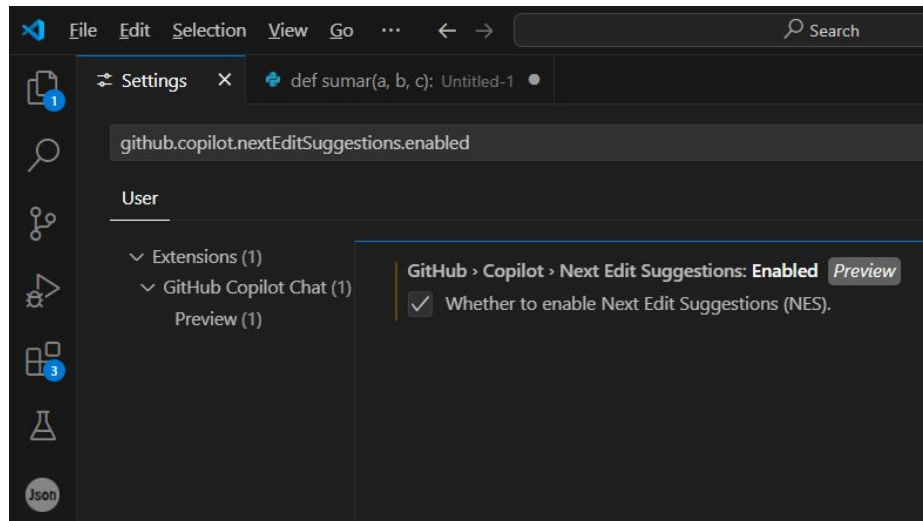
Abra el editor de configuraciones de VS Code  
(**Ctrl + ,**)

## 2 Buscar NES

Buscar la configuración (setting)  
**github.copilot.nextEditSuggestions.enabled**

## 3 Habilitar

Habilitar la configuración

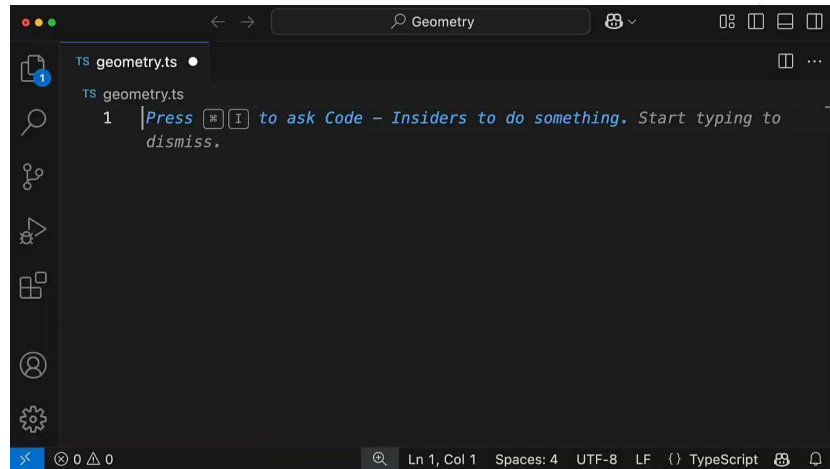


# Sugerencia de próxima edición (NES)

## Cuando estamos modificando código

Las sugerencias de próxima edición (NES) predicen dónde y qué ediciones pueden ser necesarias en función de los cambios en curso.

Las sugerencias pueden abarcar una sola instrucción, una línea completa o varias líneas, según el alcance del cambio potencial.



# Sugerencia de próxima edición (NES)

## Casos de Uso

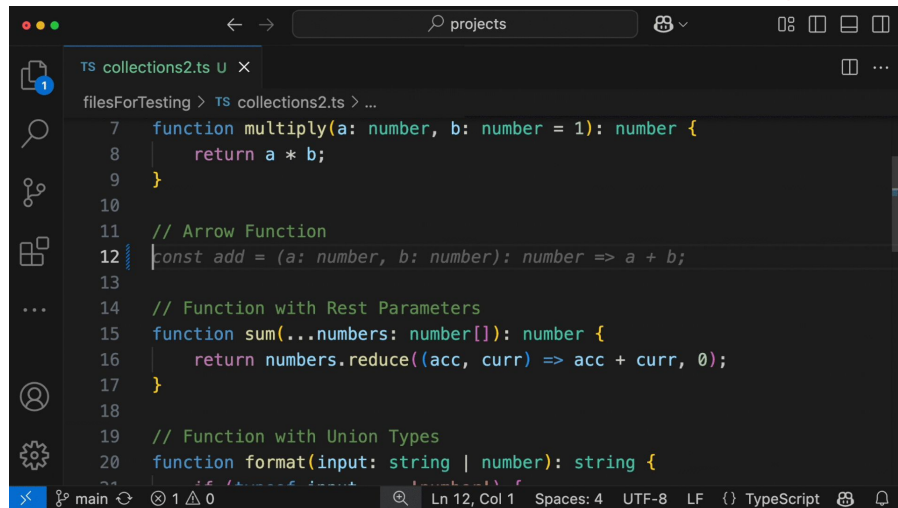
- ❑ Detectar y corregir errores
- ❑ Detectar errores más desafiantes en lógica
- ❑ Cambio de intención
- ❑ Refactorización



# Sugerencia de próxima edición (NES)

## Detectar y corregir errores

Ayuda con errores simples como errores tipográficos. Sugerirá soluciones cuando falten letras o estén intercambiadas, como `cont x = 5` o `conts x = 5`, que deberían haber sido `const x = 5`.



The screenshot shows a code editor window with a file named 'collections2.ts'. The code contains several TypeScript functions: 'multiply', 'add' (using an arrow function), 'sum' (using rest parameters), and 'format' (using union types). On line 12, the code is 'const add = (a: number, b: number): number => a + b;'. A blue vertical bar is positioned at the start of this line. At the bottom of the editor, a status bar shows 'Ln 12, Col 1' and 'Spaces: 4 UTF-8 LF {} TypeScript'. A small icon in the top left corner of the editor indicates a suggestion or error.

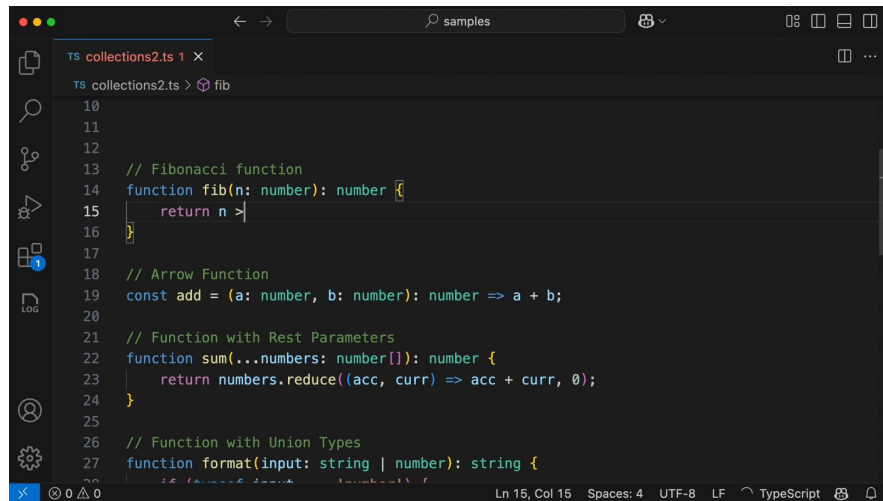
```
7 function multiply(a: number, b: number = 1): number {  
8   return a * b;  
9 }  
10  
11 // Arrow Function  
12 const add = (a: number, b: number): number => a + b;  
13  
14 // Function with Rest Parameters  
15 function sum(...numbers: number[]): number {  
16   return numbers.reduce((acc, curr) => acc + curr, 0);  
17 }  
18  
19 // Function with Union Types  
20 function format(input: string | number): string {
```



# Sugerencia de próxima edición (NES)

## Detectar errores más desafiantes en lógica

Puede ayudar con errores más desafiantes en lógica , como una expresión ternaria invertida.



The screenshot shows a code editor with a file named 'collections2.ts'. The code contains several TypeScript functions. The 'fib' function is highlighted, showing a logical error in its return statement. The code is as follows:

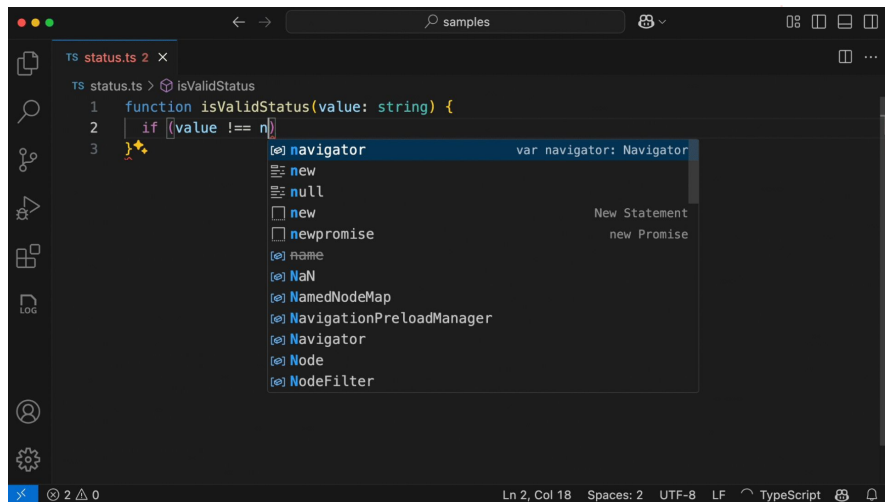
```
10
11
12
13 // Fibonacci function
14 function fib(n: number): number {
15     return n > 1 ? fib(n - 1) + fib(n - 2) : n;
16 }
17
18 // Arrow Function
19 const add = (a: number, b: number): number => a + b;
20
21 // Function with Rest Parameters
22 function sum(...numbers: number[]): number {
23     return numbers.reduce((acc, curr) => acc + curr, 0);
24 }
25
26 // Function with Union Types
27 function format(input: string | number): string {
28     if (typeof input === 'string') {
29         return input;
30     } else {
31         return input.toString();
32     }
33 }
```

The error is in line 15, where the return statement is `return n > 1 ? fib(n - 1) + fib(n - 2) : n;`. This is a logical error because it should be `return n > 1 ? fib(n - 1) + fib(n - 2) : n;`.

# Sugerencia de próxima edición (NES)

## Detectar errores más desafiantes en lógica

Puede ayudar con errores más desafiantes en lógica , una comparación que debería haberse utilizado &&(and) en lugar de || (or)

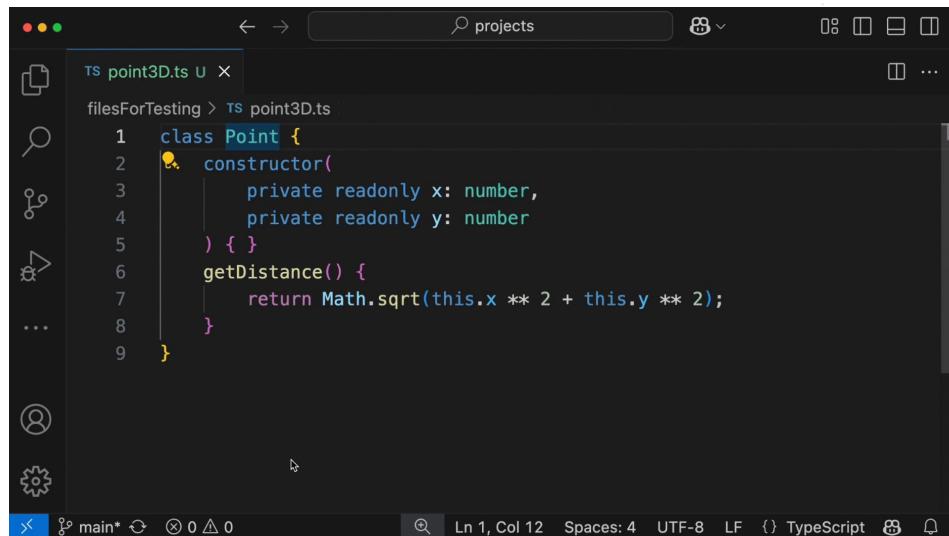


# Sugerencia de próxima edición (NES)

## Cambio de intención

Sugiere cambios en el resto del código que coinciden con un nuevo cambio de intención.

Por ejemplo, al cambiar una clase de Point a Point3D, Copilot sugerirá agregar una z variable a la definición de la clase. Después de aceptar el cambio, Copilot NES recomienda agregar z al cálculo de la distancia:

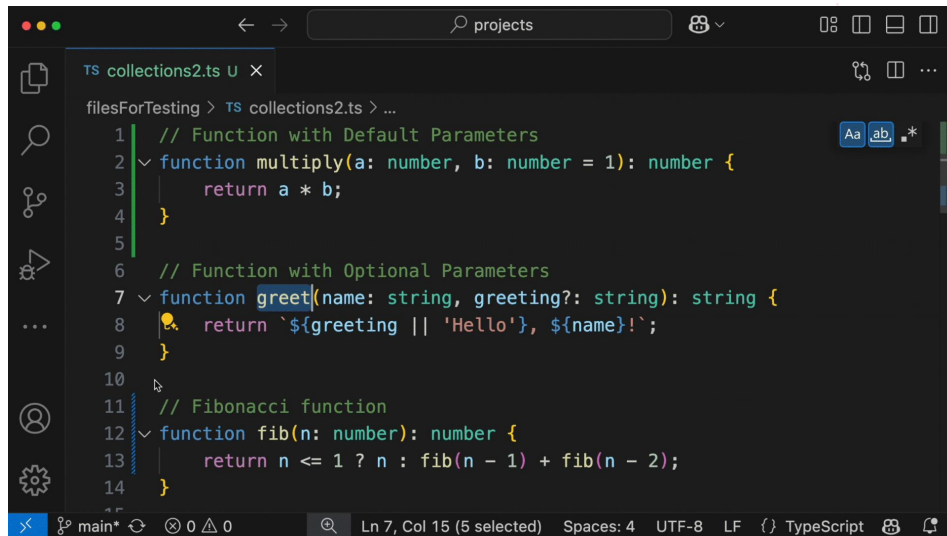


```
TS point3D.ts U x
filesForTesting > TS point3D.ts
1 class Point {
2   constructor(
3     private readonly x: number,
4     private readonly y: number
5   ) { }
6   getDistance() {
7     return Math.sqrt(this.x ** 2 + this.y ** 2);
8   }
9 }
```

# Sugerencia de próxima edición (NES)

## Refactorización

Si utiliza un nombre o patrón de nombres nuevo, el asistente le sugerirá que actualice el código subsiguiente de manera similar



The screenshot shows a VS Code editor window with a file named 'TS collections2.ts'. The code contains three functions: 'multiply', 'greet', and 'fib'. The 'greet' function is currently selected, and a refactor suggestion is visible in the left margin, indicating a change from 'greet' to 'ab'. The suggestion is shown as a blue box with the text 'Aa ab.\*'.

```
1 // Function with Default Parameters
2 function multiply(a: number, b: number = 1): number {
3     return a * b;
4 }
5
6 // Function with Optional Parameters
7 function greet(name: string, greeting?: string): string {
8     return `${greeting || 'Hello'}, ${name}!`;
9 }
10
11 // Fibonacci function
12 function fib(n: number): number {
13     return n <= 1 ? n : fib(n - 1) + fib(n - 2);
14 }
```

Ln 7, Col 15 (5 selected) Spaces: 4 UTF-8 LF {} TypeScript

# Actividad en grupo

Armar un ejemplo con el asistente para que nos genere sugerencias de código utilizando al menos una de las siguientes opciones:

- Sugerencia de finalización de código
- Sugerencia de próxima edición (NES)



**?** | **Dudas**