Inteligencia artificial y programación  
Asistente de programación tipo “Copilot” con CodeGPT y Mistral (Gratis y privado)

short line

Autor: Sergi García

Actualizado Enero 2024

Licencia

**Reconocimiento - No comercial - CompartirIgual** (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se ha de hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

📖 **Importante**

❕ **Atención**

💬 **Interesante**

**ÍNDICE**

[**1. Introducción 3**](#_dbh0n1vac4c8)

[**1.1 ¿Qué software utilizaremos? 3**](#_jtighp5yudqb)

[**2. Instalando Ollama 3**](#_mw4ckxdr9v3p)

[2.1 Instalando Ollama en local 3](#_taevr76zjx5y)

[2.2 Instalando Ollama dockerizado 3](#_u11fgs9suohs)

[2.2.1 Instalando Ollama dockerizado, solo CPU 4](#_s3l8mla412h2)

[2.2.2 Instalando Ollama dockerizado, con GPU Nvidia 4](#_xx53k8x0j8i4)

[**3. Obteniendo modelo “Mistral” para Ollama 4**](#_b4isp54o5bjb)

[3.1 Obteniendo el modelo Mistral si instalaste tu Ollama en local 4](#_qt9ucroodyr6)

[3.2 Obteniendo el modelo Mistral si instalaste Ollama dockerizado 4](#_mi8bkmsknj4s)

[**4. Instalando Visual Studio Code 4**](#_myzn9l6ach78)

[**5. Instalando extensión CodeGPT para Visual Studio Code 4**](#_akwjb64l029f)

[**6. Configurando la extensión para uso como Chat 5**](#_6jfa01gxyd9w)

[**7. Instalando otros modelos de Ollama 6**](#_yr45emacy0n2)

Asistente de programación tipo “Copilot” con CodeGPT y Mistral (Gratis y privado)

# Introducción

En el mundo de la programación, la demanda de herramientas eficientes y poderosas para agilizar el desarrollo de código es más evidente que nunca. La evolución constante de la tecnología y la creciente complejidad de los proyectos exigen soluciones innovadoras que no solo aceleren el proceso de codificación, sino que también mejoren la calidad y la precisión del código resultante.

La proliferación de herramientas de asistencia en programación, como Copilot o incluso ChatGPT, ha sido bien recibida en la comunidad de desarrollo de software gracias a su capacidad para acelerar el proceso de codificación. Sin embargo, surge un dilema significativo asociado con muchas de estas soluciones existentes: el costo de adquisición/suscripción y la preocupación por la privacidad (envías tu código a un tercero).

Para abordar estas inquietudes, en este documento ofrecemos una alternativa gratuita y “self-hosted” para garantizar la máxima privacidad.

## ¿Qué software utilizaremos?

Para esta tarea, en resumen, utilizaremos el siguiente software:

* **Ollama**: <https://ollama.ai>
  + Software para descargar, preparar y lanzar distintos modelos LLMs de LLama.
  + El modelo “Mistral”. Lo descargaremos usando Ollama.
    - Su web oficial <https://mistral.ai/>
* **Visual Studio Code:** <https://code.visualstudio.com/>
  + Editor de código ligero creado por Microsoft, con multitud de plugins disponibles.
* **Extensión de Visual Studio Code “CodeGPT”:** <https://www.codegpt.co/>
  + Extensión:<https://marketplace.visualstudio.com/items?itemName=DanielSanMedium.dscodegpt&ssr=false>.¡Cuidado! Hay otras extensiones con este nombre, recomendamos instalar desde el enlace.

❕ **Atención:** los requisitos mínimos pueden variar según el modelo utilizado. Por lo general siguen esta norma, según los billones (americanos) de parámetros: 7b -> 8GB de RAM, 13b -> 16 GB de RAM y 43b -> 32 GB de RAM

# Instalando Ollama

## Instalando Ollama en local

Descarga tu versión de Ollama de <https://ollama.ai/download> y sigue las instrucciones de instalación según tu sistema operativo.

Por ejemplo, en sistemas Linux ejecuta como “root” los siguientes comandos:

| apt update && apt install curl -y  curl https://ollama.ai/install.sh | sh |
| --- |

## Instalando Ollama dockerizado

Para ver como instalar Ollama de forma dockerizado, seguiremos la siguiente entrada: <https://ollama.ai/blog/ollama-is-now-available-as-an-official-docker-image>

Si no tienes instalado Docker o no sabes utilizarlo, puedes encontrar más información en mi curso

<https://github.com/sergarb1/CursoIntroduccionADocker>

### Instalando Ollama dockerizado, solo CPU

Si queremos usar la imagen “CPU only” con este comando tenemos suficiente:

| docker run -d -v ollama:/root/.ollama -p 11434:11434 --name ollama ollama/ollama |
| --- |

### Instalando Ollama dockerizado, con GPU Nvidia

Si tenemos una tarjeta gráfica Nvidia y queremos aprovechar su potencia para esta imagen dockerizada, debermos dar los siguientes paso:

1. Instalar y configurar Nvidia container toolkit. Los distintos pasos están en <https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/install-guide.html#installation>
2. Una vez configurado, puedes poner en marcha el contenedor que utiliza la potencia de tu GPU con el siguiente comando:

| docker run -d --gpus=all -v ollama:/root/.ollama -p 11434:11434 --name ollama ollama/ollama |
| --- |

# Obteniendo modelo “Mistral” para Ollama

## Obteniendo el modelo Mistral si instalaste tu Ollama en local

En el caso de que hayas instalado Ollama en local, simplemente ejecuta estos dos comandos:

| ollama pull mistral ollama pull mistral:instruct |
| --- |

Con esto, estará todo listo para configurar la extensión de Visual Studio Code “CodeGPT”.

## Obteniendo el modelo Mistral si instalaste Ollama dockerizado

En el caso de que hayas instalado Ollama de forma dockerizada, simplemente ejecuta estos dos comandos:

| docker exec -it ollama ollama pull mistral  docker exec -it ollama ollama pull mistral:instruct |
| --- |

Con esto, estará todo listo para configurar la extensión de Visual Studio Code “CodeGPT”.

# Instalando Visual Studio Code

La instalación del editor Visual Studio Code es sencilla y está ampliamente documentada en tutoriales, videos, etc. y en multitud de formatos (ejecutable Windows, paquete “.deb”, paquete Snap, etc.). Para esto, simplemente, te recomiendo que descargues el software de <https://code.visualstudio.com/> y siguas las instrucciones.

# Instalando extensión CodeGPT para Visual Studio Code

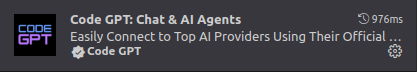
¡Cuidado! Hay otras extensiones con este nombre, recomendamos instalar desde el enlace.

<https://marketplace.visualstudio.com/items?itemName=DanielSanMedium.dscodegpt&ssr=false>

Además, también puedes instalarla si desde dentro de Visual Studio Code, pulsas la combinación de teclas “Control + P” y en la caja emergentes pegas este comando:

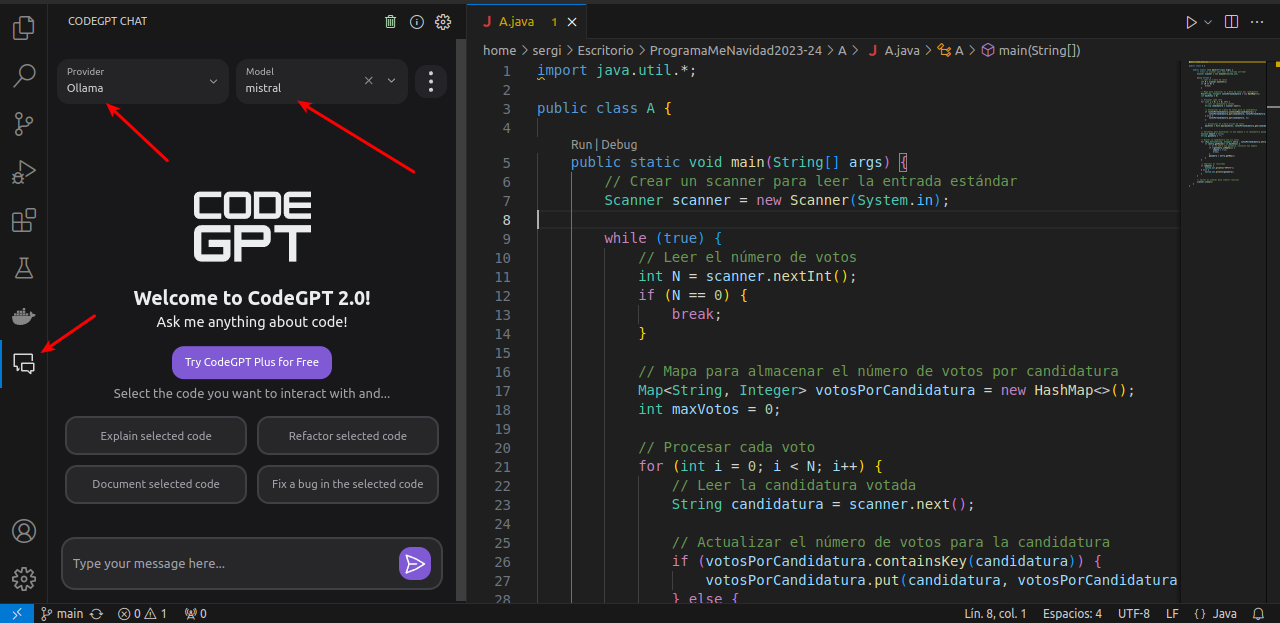
| ext install DanielSanMedium.dscodegpt |
| --- |

Si todo va bien, tu Visual Studio Code tendrá instalada esta extensión:



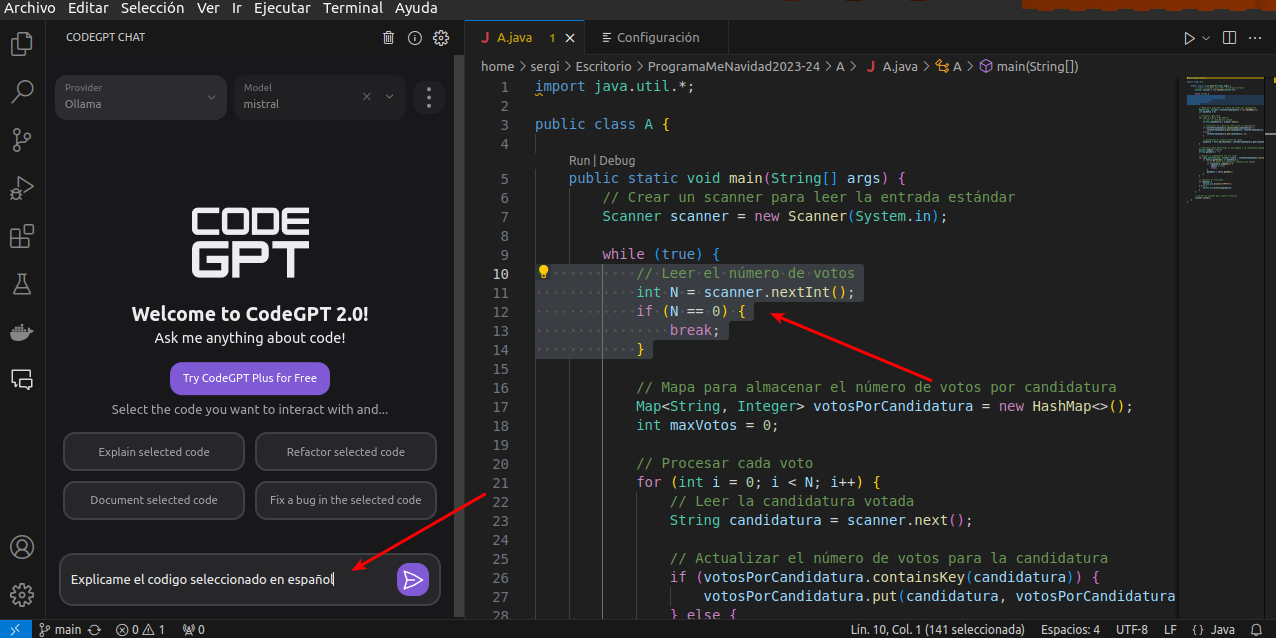
# Configurando la extensión para uso como Chat

Una vez lista la extensión, ábrela e indica que estás usando “Ollama” y el modelo “Mistral”, como se puede ver en la siguiente imagen:

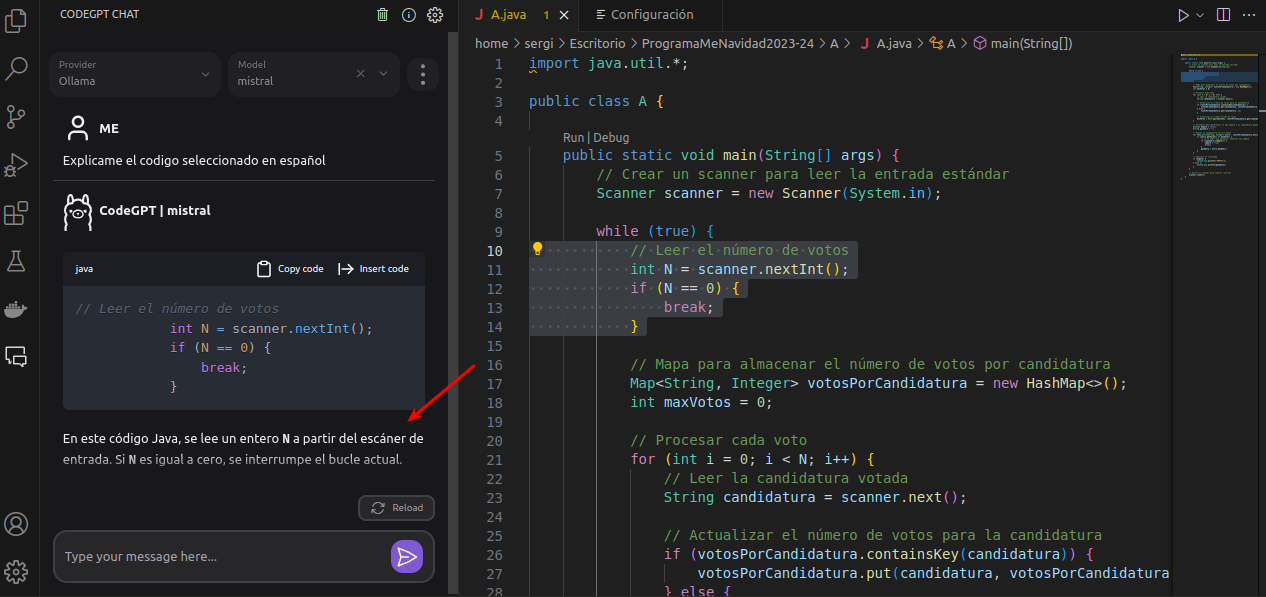


Con esta configuración, tendrás un chat similar a “Chat GPT” para tu código.

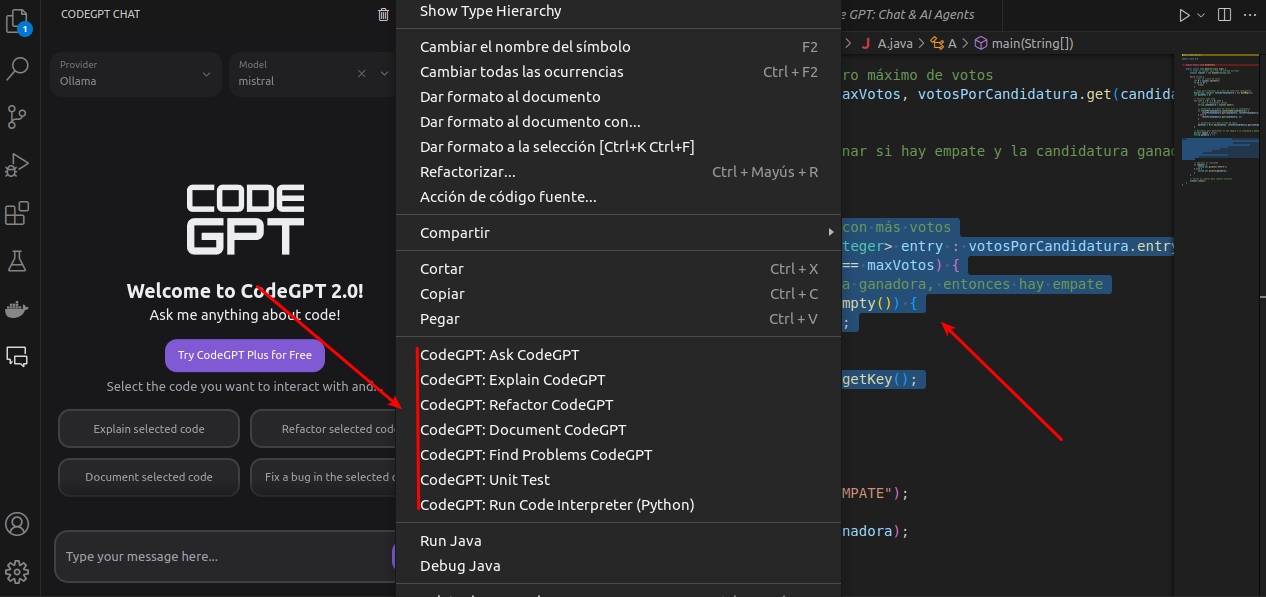
Aquí un ejemplo de uso del Chat:



Aquí una porción de la respuesta obtenida:



También se puede, simplemente, seleccionando código y pulsando botón derecho:



# Instalando otros modelos de Ollama

En este ejemplo hemos utilizado el modelo “Mistral”, pero realmente pueden utirlizarse otros modelos como “codellama” u otros que surjan en el futuro (incluso en Ollama se puede importar cualquier modelo en formato GGUF tal como indican aquí <https://github.com/jmorganca/ollama>)

Un ejemplo para añadir “codellama”:

) Si tienes una instalación local de Ollama, puedes añadirlo con:

| ollama pull codellama  ollama pull codellama:instruct |
| --- |

B) Si tienes una instalación dockerizada de Ollama, puedes añadirlo con:

| docker exec -it ollama ollama pull codellama  docker exec -it ollama ollama pull codellama:instruct |
| --- |