

Laboratorio 8
Exploración y Uso Avanzado de Plataformas IA, Repositorios
Profesionales y Herramientas Globales para el desarrollo de IA
y de SW

Materia:

Profundización de inteligencia artificial

Participantes:

Ana Maria Navarro Fernandez

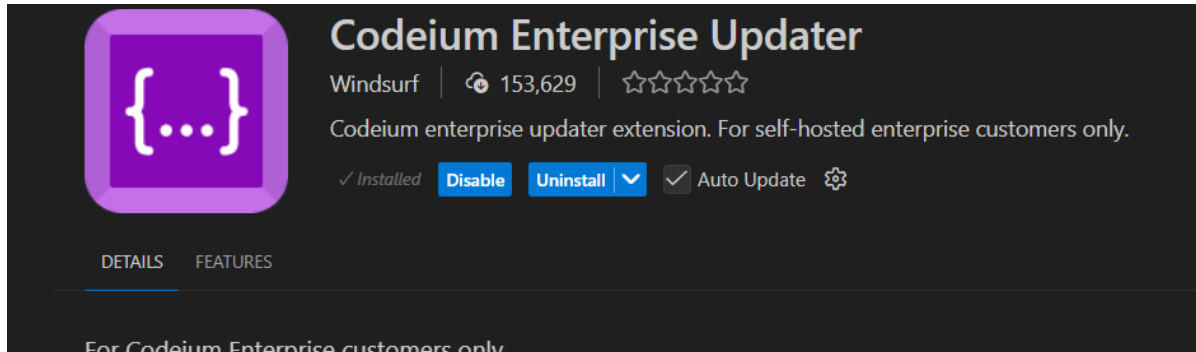
Profesor:

Carlos Betancourt correa

Universidad de Manizales
ingeniería en sistemas y telecomunicaciones
Manizales, Caldas, Colombia

1. Instalar la extensión Codeium en Visual Studio Code

- Se abre VS Code → Extensions.
- Se busca “Codeium”.
- Se instala la extensión Codeium Enterprise Updater.
- En la captura se verifica que aparece como Installed.



2. Crear un archivo de prueba

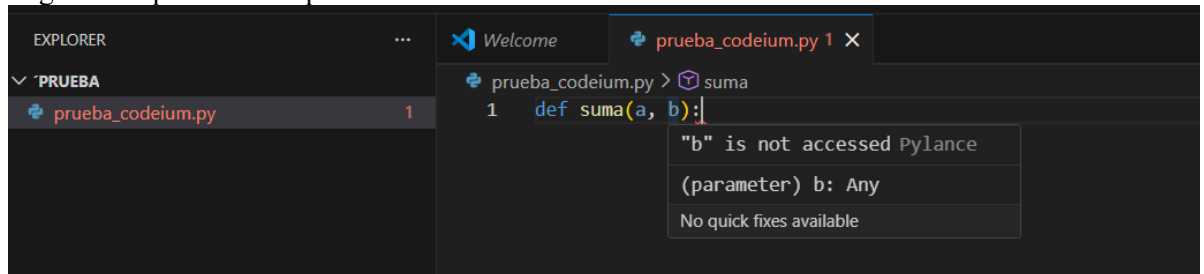
- Se crea una carpeta llamada PRUEBA.
- Dentro de ella se crea el archivo: prueba_codeium.py

3. Probar autocompletado de Codeium

En el archivo se comienza a escribir:

```
def suma(a, b):
```

- ✓ Lo que ocurrió (según tus capturas):
 - Codeium detectó que el parámetro b no estaba siendo usado.
 - Mostró advertencias del analizador.
 - Sugirió completar el cuerpo de la función.



El código final quedó:

```
def suma(a, b):  
    return a + b
```

- ✦ Resultado: Codeium funcionó como asistente de código, ofreciendo sugerencias automáticas.

4. Probar el Chat de Codeium

En el panel derecho de VS Code se abre:

➡ Codeium Chat

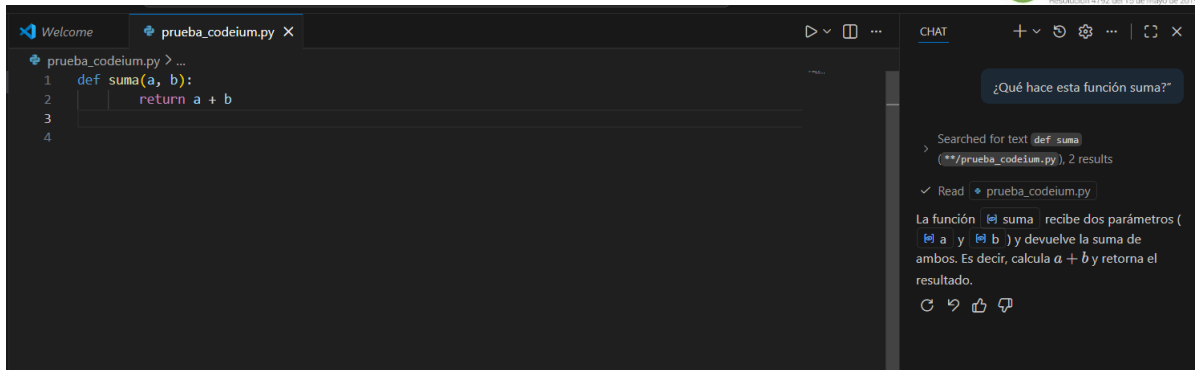
Se realiza la consulta:

“¿Qué hace esta función suma?”

- ✓ Respuesta de Codeium (según tu captura):

“La función suma recibe dos parámetros (a y b) y devuelve la suma de ambos. Es decir, calcula a + b y retorna el resultado.”

Resultado:



Codeium entendió el código del archivo, lo analizó y respondió en lenguaje natural.

5. Conclusión

- La instalación se completó correctamente.
- Codeium asistió en la escritura del código mediante autocompletado inteligente.
- El panel de chat permitió realizar preguntas contextuales sobre el código.
- Se confirmó que Codeium funciona como herramienta de IA integrada en el entorno de desarrollo, cumpliendo con lo solicitado en el laboratorio.