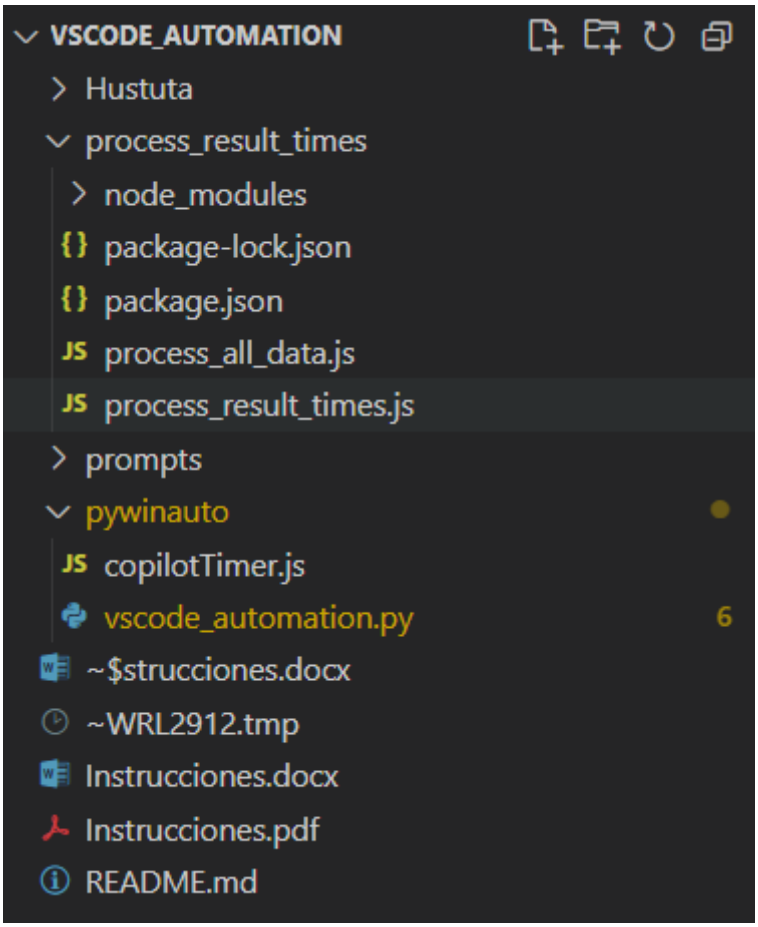
VSCODE Automation

Esta carpeta contiene los scripts necesarios para poder ejecutar prompts de manera automática en la extensión github copilot chat.

Estructura de carpetas



Hustuta: Contiene el proyecto “cypress-realworld-app”. Está modificado para que contenga la estructura de los test pero no el contenido de forma que pueda ser utilizado a modo de contexto a la hora de ejecutar prompts relacionados al proyecto.

Process_result_times: Por la forma en que el script principal ejecuta los prompts y guarda resultados, es necesario un procesado posterior. El script (process_result_times.js) de esta carpeta toma los resultados sin procesar y crea un archivo json en la carpeta. Este archivo contendrá por cada prompt ejecutado el tiempo que ha tardado en generarse junto con el código extraído de la transcripción del resultado.

```
[
  {
    "timestamp": "2025-05-20 20:37:46.114",
    "output_file": "output_claude_3_5_sonnet\\transaction-feeds11.spec_response_claude_
    "source_file": "C:\\Users\\xabia\\OneDrive\\Documentos\\4.Maila\\TFG-Bestelakoak\\B
    "requestTimestamp": "2025-05-20 20:37:46.321",
    "requestTimeMs": 1747766266321,
    "responseTimeMs": 1747766297231,
    "durationMs": 30910,
    "code": "// Get the user's contacts first\n\n  cy.database(\"filter\", \"contacts\"
  },
```


Prompts: En esta carpeta con formato txt se guardarán los prompts que tienen que ser ejecutados automáticamente

Pywinauto: En esta carpeta está el script que automatiza el uso de github copiot chat. Los resultados se guardarán en carpetas separadas dentro de esta. Con nombres como output_claude_sonnet_3_5 para la versión 3.5 de Claude Sonnet

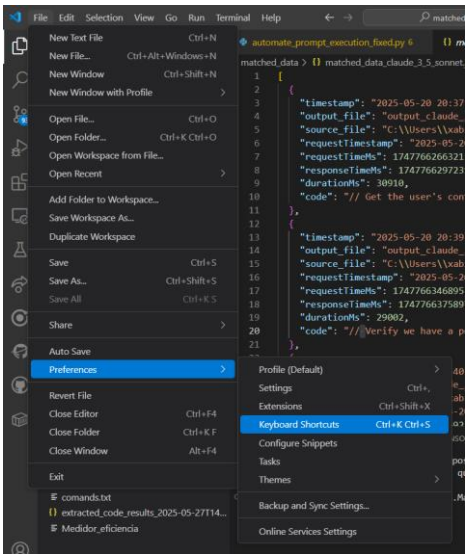
Configuración VSCODE

Hay un README generado por IA con explicaciones más precisas si hiciera falta.

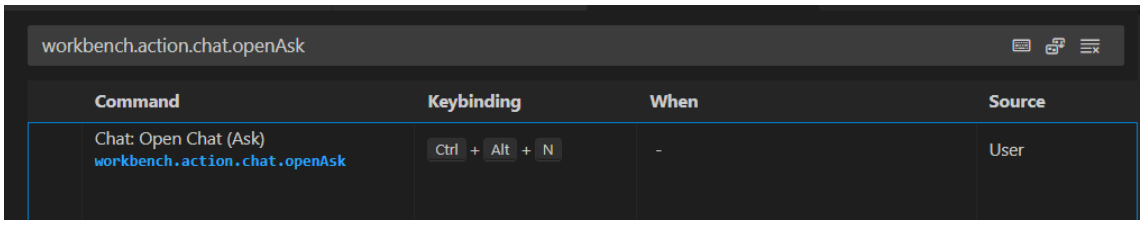
Configuración de shortcut

Deberá hacerse una sola vez en vscode. Añade atajo de teclado a la acción de cambiar el chat a modo Ask.

- File →Preferences → Keyboard Shortcuts o Ctrl + K Ctrl + S



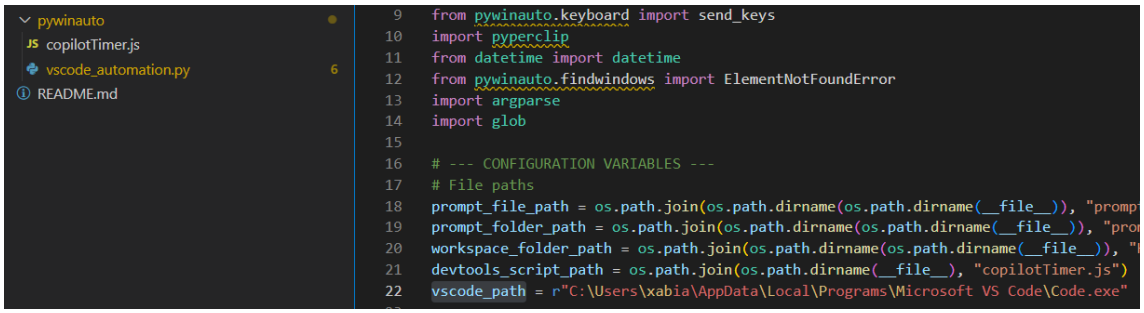
- En el menú que se abre, escribir “workbench.action.chat.openAsk”



- Editarlo para que el Keybinding sea **Ctrl + Alt + N**

Configuración script (vscode_automation)

Es necesario poner el path al ejecutable de vscode (code.exe) del sistema, para que el script pueda abrir una nueva instancia



La variable por editar es vscode_path (línea 21 del archivo pywinauto/vscode_automation.py)

Instalación de dependencias Python

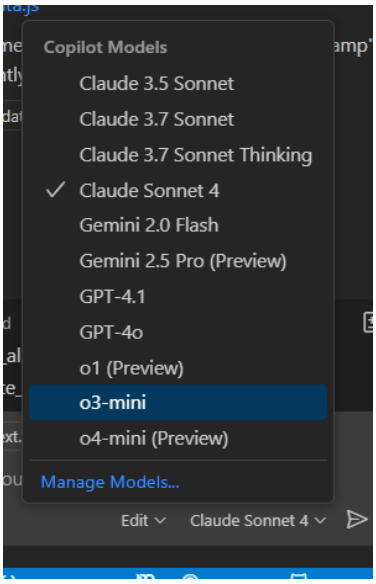
Ejecutar **pip install pywinauto pyperclip psutil**

Utilización

El script pywinauto/vscode_automation.py ejecutará todos los prompts de la carpeta prompts de forma secuencial y guardará la transcripción de la conversación en formato txt en una carpeta llamada output_ + nombre_llm. Para inicial la automatización

Python vscode_automation [0-10]

El script recibirá como argumento un número del 0 al 10 para que el pueda guardar los resultados en la carpeta adecuada. Este número corresponde a la posición de cada llm en la lista del chat ask en vscode siendo 0 Claude Sonnet 3.5 y 10 o4-mini preview

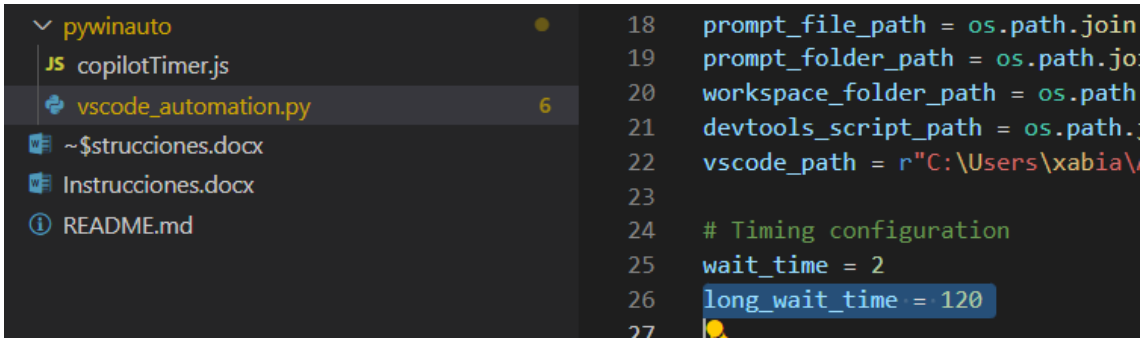


El funcionamiento del script es el siguiente:

1 – Abre la consola de las herramientas de desarrollador (DevTools) y carga un script (copilotTimer.js) que medirá el tiempo de generación de cada respuesta del llm

2 – Comenzará un nuevo chat, añadirá la carpeta de contexto con # (#cypress-realworld-app) y pegará el contenido del prompt. Pulsará enter para comenzar la generación de respuesta y la medición de tiempo. El tiempo de espera para la generación del resultado es una aproximación. El script está configurado para esperar 2 minutos. Pasado este tiempo guardará la conversación en un archivo txt empezará un nuevo chat y seguirá con el siguiente prompt.

Dado que dependiendo del tipo de prompt y llm el tiempo de generación de respuesta puede ser mayor o menor, el tiempo de espera es un valor modificable en el script **long_wait_time**, línea 26 del script **pywinauto/vscode_automation.py**

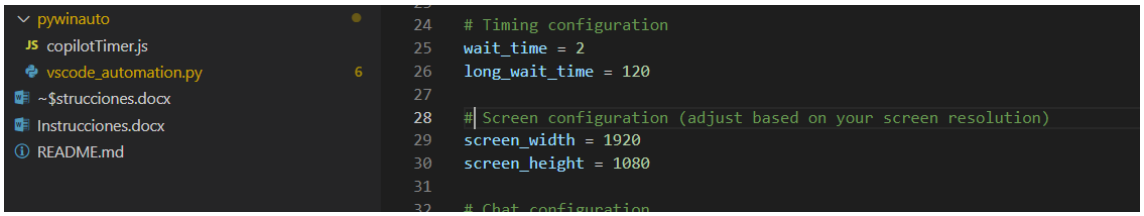


3 - Una vez ejecutados todos los prompts el script volverá a abrir la consola de herramientas de desarrollador y copiara un objeto que contiene los tiempos que ha tardado cada respuesta en generarse y lo guardará en archivo de formato Json

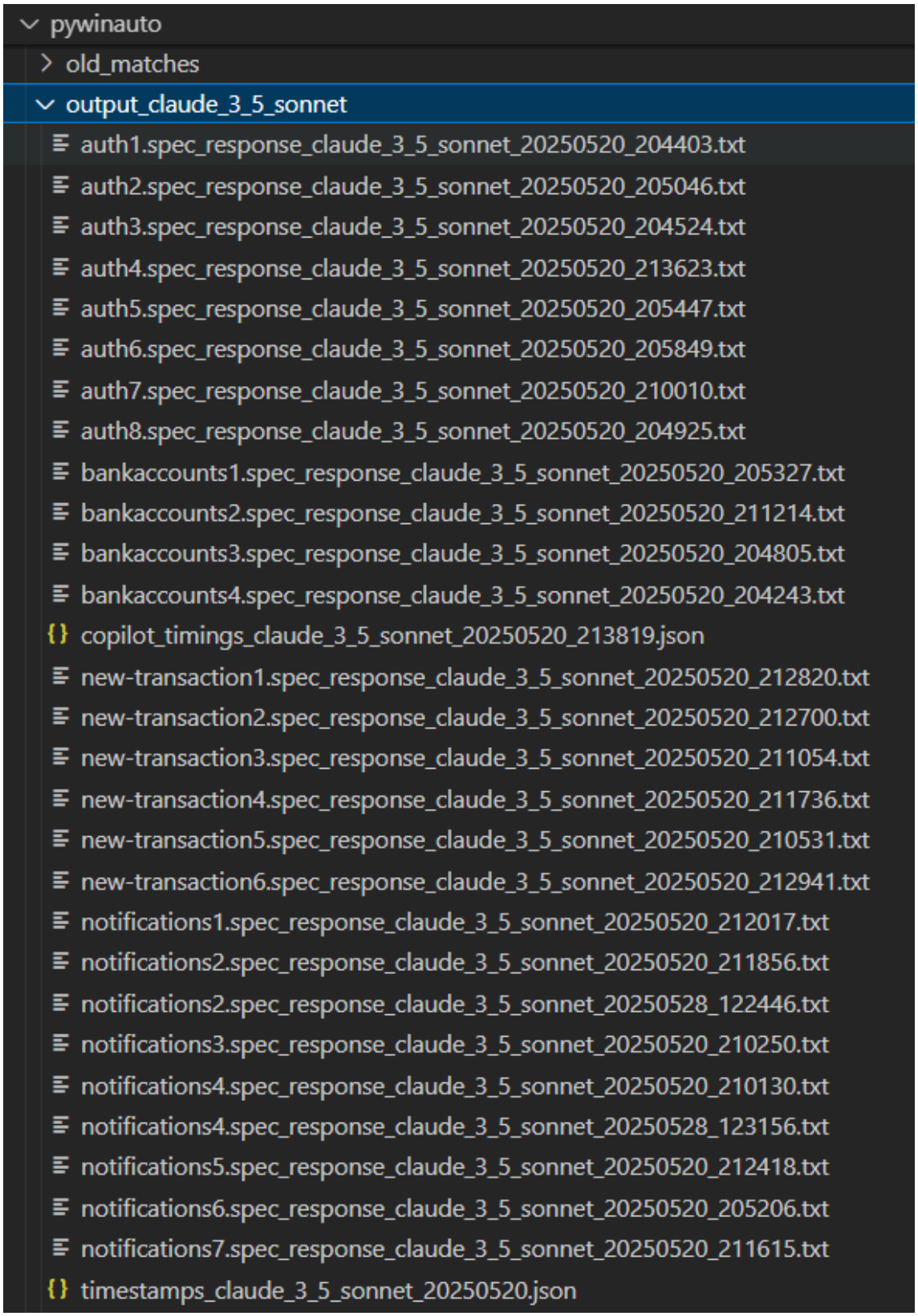
Este sería el contenido de una carpeta con los resultados de Claude sonnet 3.5

4 – Solo hay una parte de la automatización que requiere la simulación del ratón. Esto es a la hora de cerrar una ventana de dialogo. Para enfocar en esa ventana de dialogo se hace clic en el centro de la pantalla. Dependiendo del tamaño de la pantalla la posición de clic puede variar.

Si hubiera problemas existe la posibilidad de configurar el tamaño de la pantalla, para que el cálculo sea más preciso



Líneas 29 y 30 de vscode-automation.py


pywinauto
old_matches
output_claude_3_5_sonnet
auth1.spec_response_claude_3_5_sonnet_20250520_204403.txt
auth2.spec_response_claude_3_5_sonnet_20250520_205046.txt
auth3.spec_response_claude_3_5_sonnet_20250520_204524.txt
auth4.spec_response_claude_3_5_sonnet_20250520_213623.txt
auth5.spec_response_claude_3_5_sonnet_20250520_205447.txt
auth6.spec_response_claude_3_5_sonnet_20250520_205849.txt
auth7.spec_response_claude_3_5_sonnet_20250520_210010.txt
auth8.spec_response_claude_3_5_sonnet_20250520_204925.txt
bankaccounts1.spec_response_claude_3_5_sonnet_20250520_205327.txt
bankaccounts2.spec_response_claude_3_5_sonnet_20250520_211214.txt
bankaccounts3.spec_response_claude_3_5_sonnet_20250520_204805.txt
bankaccounts4.spec_response_claude_3_5_sonnet_20250520_204243.txt
copilot_timings_claude_3_5_sonnet_20250520_213819.json
new-transaction1.spec_response_claude_3_5_sonnet_20250520_212820.txt
new-transaction2.spec_response_claude_3_5_sonnet_20250520_212700.txt
new-transaction3.spec_response_claude_3_5_sonnet_20250520_211054.txt
new-transaction4.spec_response_claude_3_5_sonnet_20250520_211736.txt
new-transaction5.spec_response_claude_3_5_sonnet_20250520_210531.txt
new-transaction6.spec_response_claude_3_5_sonnet_20250520_212941.txt
notifications1.spec_response_claude_3_5_sonnet_20250520_212017.txt
notifications2.spec_response_claude_3_5_sonnet_20250520_211856.txt
notifications2.spec_response_claude_3_5_sonnet_20250528_122446.txt
notifications3.spec_response_claude_3_5_sonnet_20250520_210250.txt
notifications4.spec_response_claude_3_5_sonnet_20250520_210130.txt
notifications4.spec_response_claude_3_5_sonnet_20250528_123156.txt
notifications5.spec_response_claude_3_5_sonnet_20250520_212418.txt
notifications6.spec_response_claude_3_5_sonnet_20250520_205206.txt
notifications7.spec_response_claude_3_5_sonnet_20250520_211615.txt
timestamps_claude_3_5_sonnet_20250520.json

Para obtener un objeto resumen con los tiempos de ejecución + la extracción del código generado se debe utilizar el script

process_result_times/process_result_times.js los resultados un json por carpeta (llm) estarán en la carpeta matched_data

Ejecutar con el comando **node process_result_times.js**

A TENER EN CUENTA ANTES DE EJECUTAR!!!

1 – El número que se le pasa a script vscode_automation se usa para guardar los resultados en la carpeta correspondiente y nombrar los archivos, pero NO configura el llm en la instancia de VSCODE. Esto deberá hacerse a mano. Ejecutar el script, interrumpirlo una vez abierta la ventana vscode (ctrl + c) y poner manualmente el llm que se quiera. Una vez hecho esto volver a ejecutar

2 – Algunos llm como sonnet 3.7 thinking pueden tardar más de los 2 minutos que están configurados. Por lo que se debería cambiar el tiempo de espera en las pruebas que he hecho 3 minutos han sido suficientes

3 – Por el momento el script no tiene mecanismos de pausa por lo que, si hay algún error en la generación de un prompt, limite de tokens,... y se interrumpe la ejecución, los tiempos no podrán obtenerse ya que se guardan al final. Los resultados de los prompts ejecutados en cambio sí estarán ya que se van guardando durante la ejecución