

HITO 1 DEL 2º TRIMESTRE DE ENTORNOS DE DESARROLLO

Alejandro Cortés Díaz

Índice

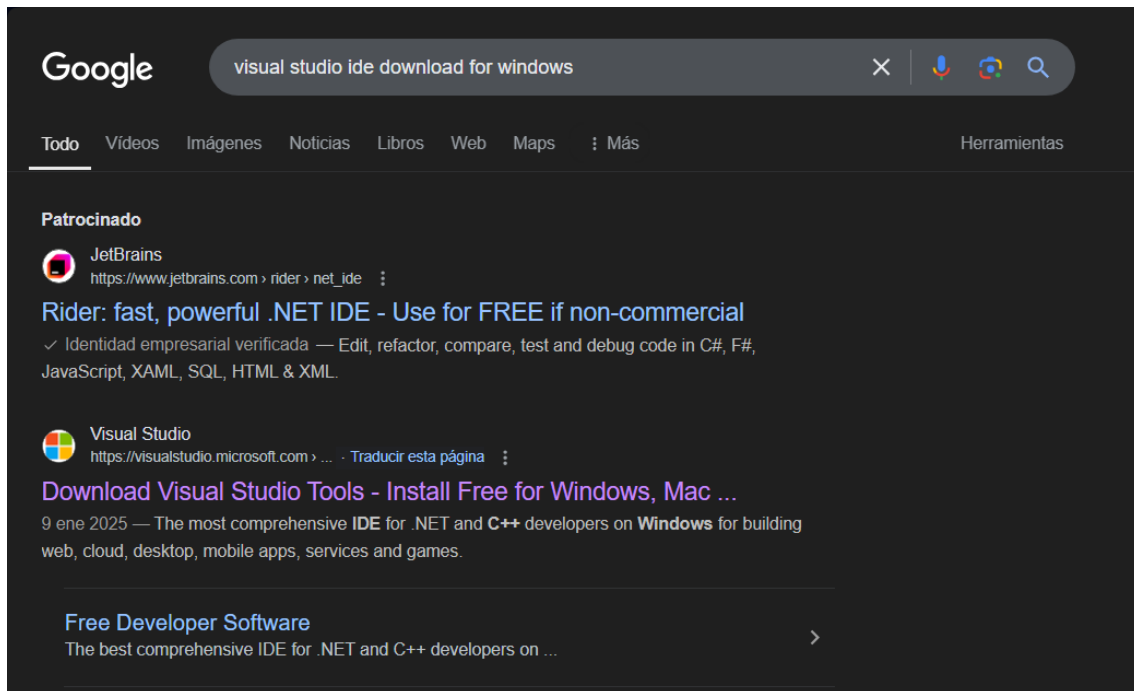
DESARROLLO	2
PUNTO 1;	2
Instalar adecuadamente un entorno de desarrollo no visto en clase, creando un	2
manual de uso paso a paso tanto de la instalación como de las partes de dicho	2
entorno de desarrollo y sus utilidades.	2
Modifica la apariencia del entorno de desarrollo.	29
Evalúa entornos integrados de desarrollo, analizando, sus características para	30
editar código fuente y generar ejecutables.	30
PUNTO 2;	31
Crea una aplicación sencilla (Un registro, un bucle sencillo...) utilizando el lenguaje de programación que prefieras, en el entorno instalado.	31
Define la aplicación realizada, lenguaje utilizado, la funcionalidad de la	31
aplicación, el proceso de elaboración del programa. Justificando de manera	31
completa indicando el porqué de ese lenguaje y el porqué de esa aplicación.....	31
Diseñar un diagrama de casos de uso sobre la aplicación realizada. Explica de	33
manera escrita lo realizado, detallando las relaciones entre los diferentes ítems.....	33
Describe mediante la técnica de texto informal el caso de uso realizado en el	33
punto anterior.	33
PUNTO 3;	34
Verifica el funcionamiento de programas, diseñando y realizando pruebas. Realiza un plan de pruebas que debe incluir la estrategia de pruebas que consideres necesaria para	34
comprobar el buen funcionamiento del desarrollo. El plan de pruebas deberá estar “ilustrado” con al menos cinco ejemplos de resultados positivos y de resultados negativos sobre las pruebas en cuestión.....	34
PUNTO 4;	43
Optimiza código empleando las herramientas disponibles en el entorno de desarrollo. Diseña una optimización para dicho desarrollo detallando los cambios que has de realizar, y la diferencia en los resultados, justificando el proceso.	43
Enlace a GitHub	44
Bibliografía	45

DESARROLLO

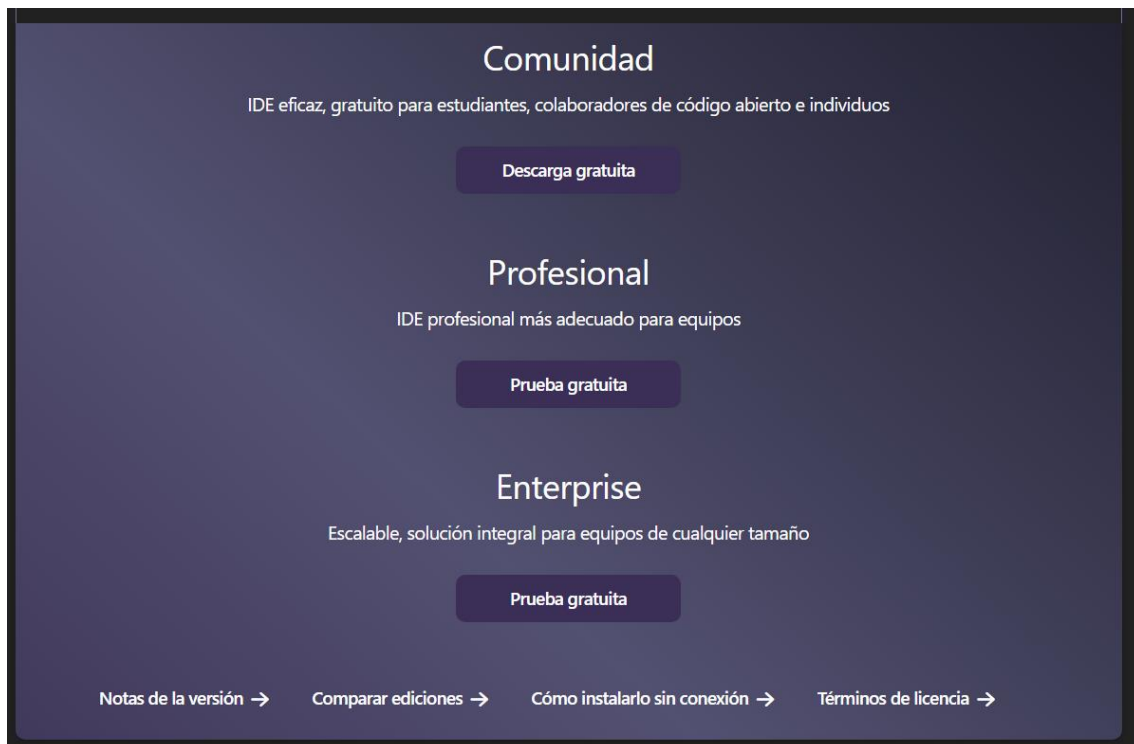
PUNTO 1;

Instalar adecuadamente un entorno de desarrollo no visto en clase, creando un manual de uso paso a paso tanto de la instalación como de las partes de dicho entorno de desarrollo y sus utilidades.


- Antes de proceder con la instalación de Visual Studio IDE, primero hemos de buscar por resultados.



- La mejor opción habitualmente es el lugar oficial, donde instalaré el programa. Y se nos ofrecen 3 opciones, de las cuales he elegido la de "Comunidad".



- Una vez hacer click en “Descarga gratuita”, automáticamente descarga el installer en tu dispositivo.

 VisualStudioSetup.exe	25/01/2025 13:33	Aplicación
---	------------------	------------

- Al abrirlo se procesa la siguiente ventana;



Visual Studio Installer

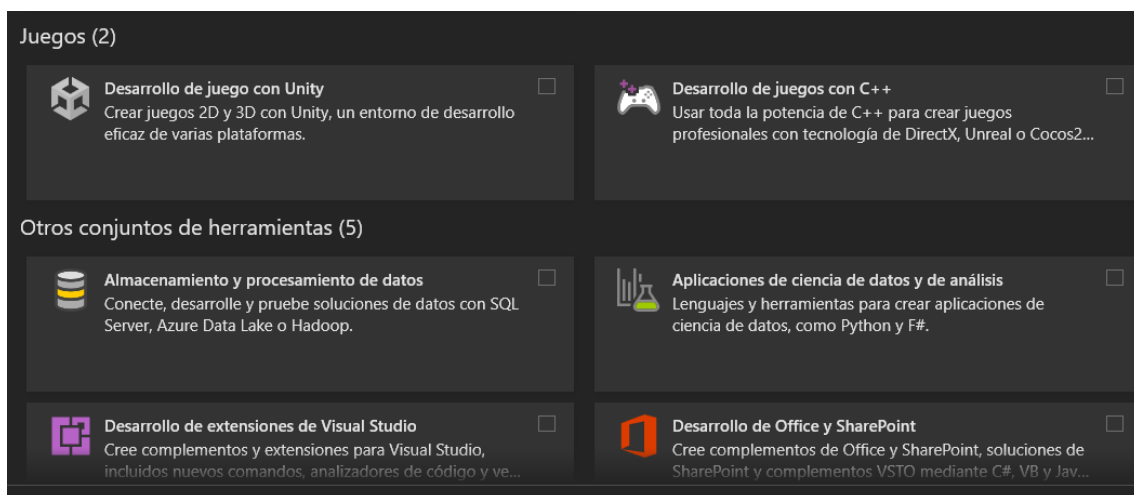
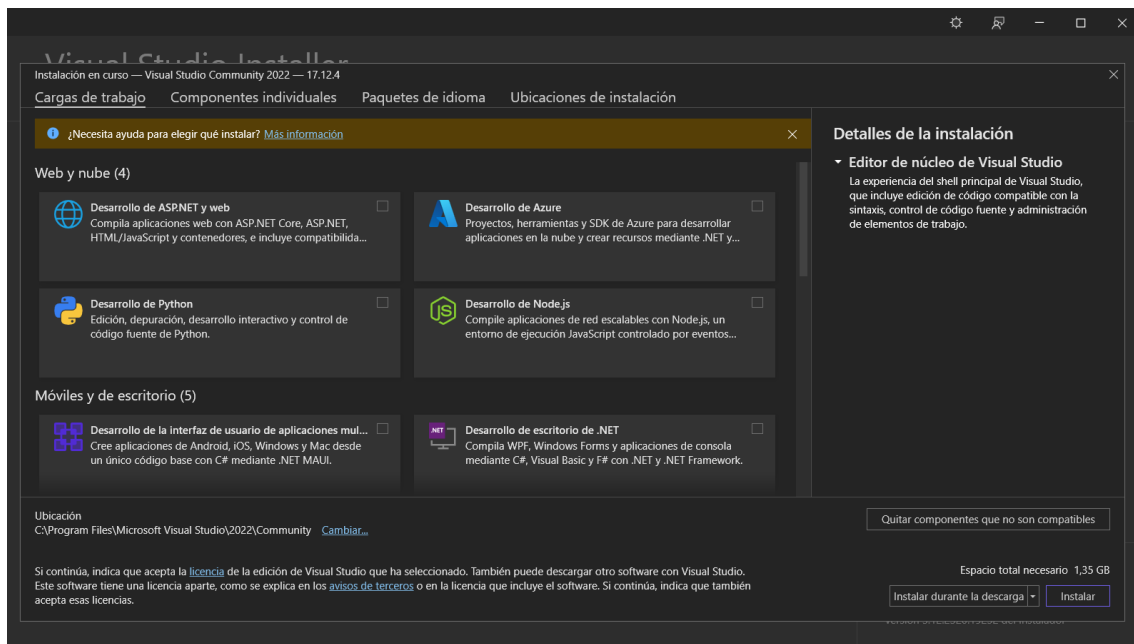
Antes de empezar, necesitamos realizar algunos ajustes para que pueda configurar la instalación.

Para más información sobre privacidad, vea la [Declaración de privacidad de Microsoft](#).

Al continuar, acepta los [Términos de licencia del software de Microsoft](#).

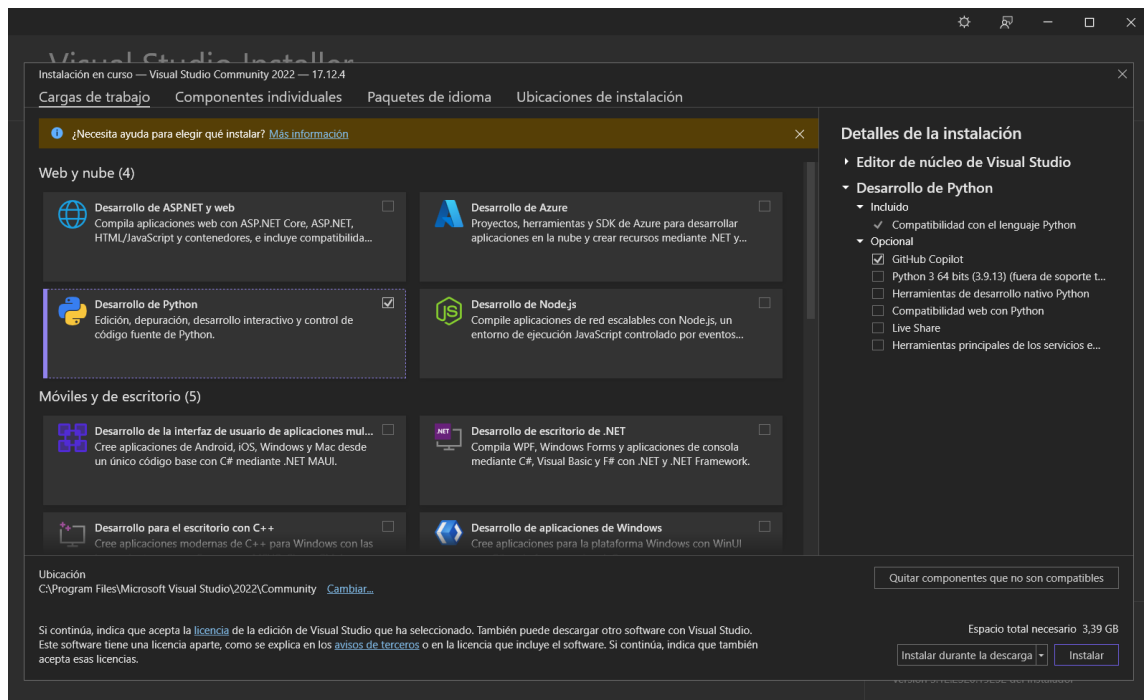
Continuar

- Presionamos “Continuar”. Y aparece la siguiente ventana con estos apartados;

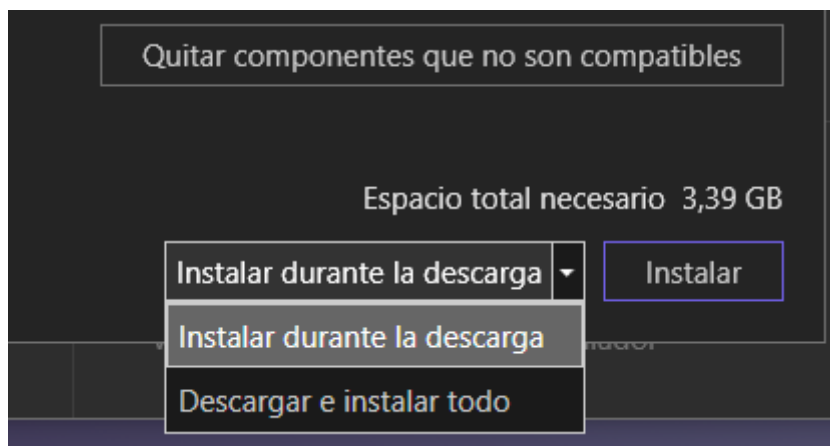


1. Por un lado, en “Cargas de trabajo”, podemos seleccionar qué tipo de desarrollo llevaremos a cabo, si queremos utilizar Python, Node.js, aplicaciones móviles y de escritorio, si deseamos codificar videojuegos y utilizar C++, etc.

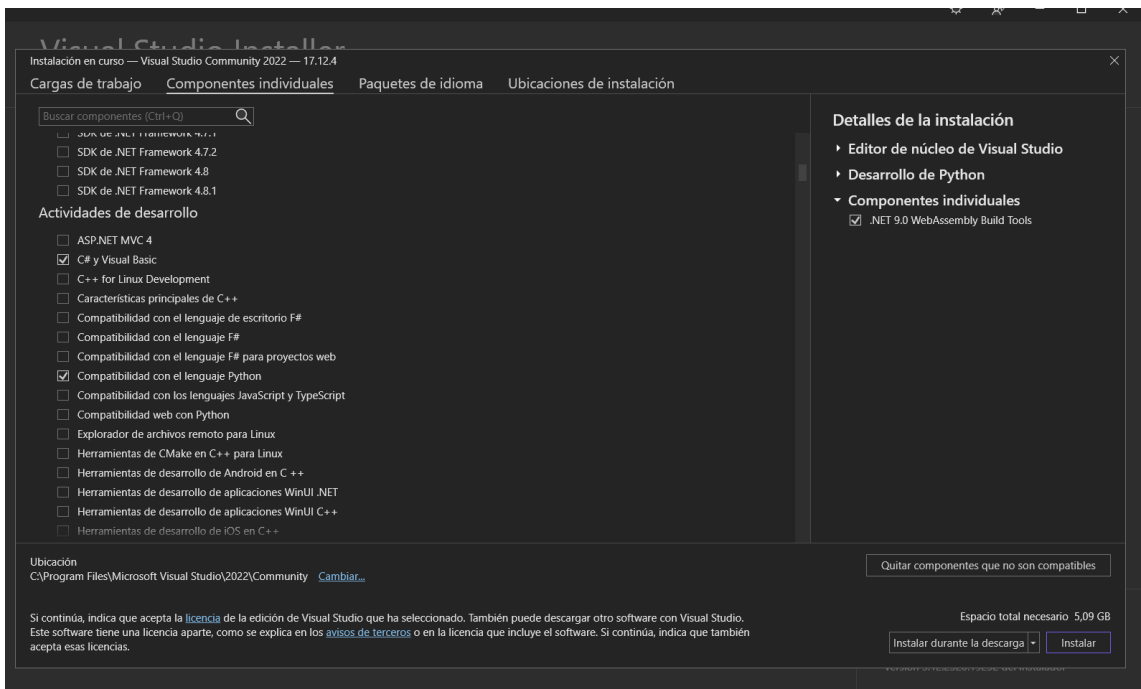
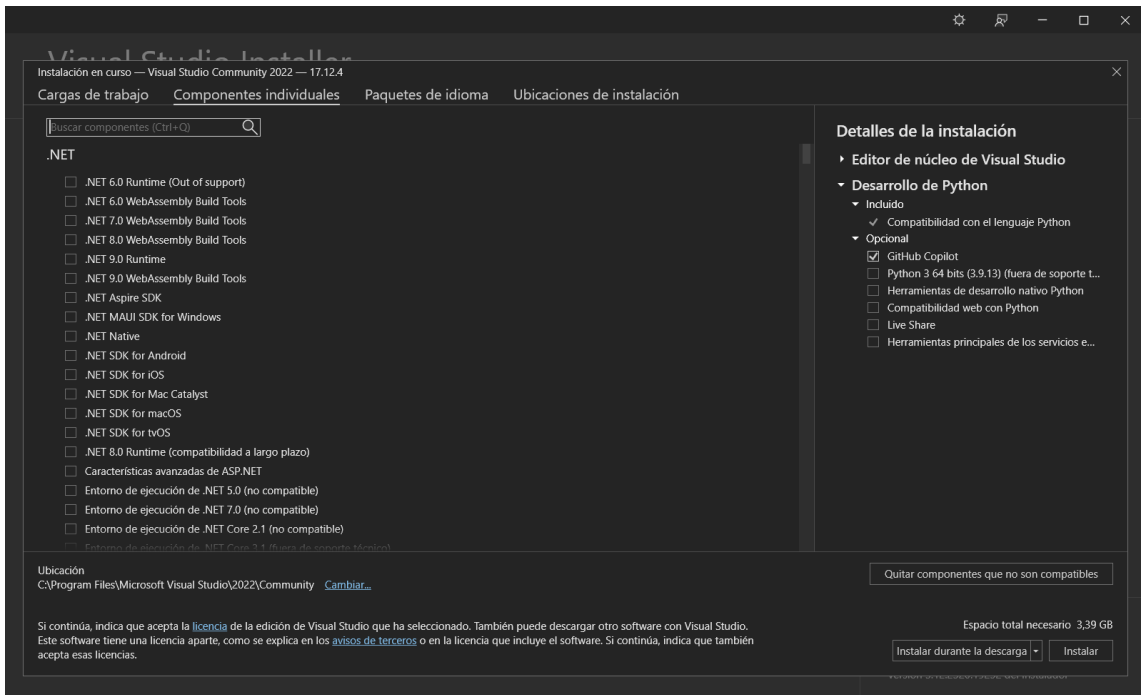
Además, en la ventana de la derecha se muestran aquellos componentes que deseamos instalar, y aquellos que se instalarían tan sólo con seleccionar alguna de las opciones que se nos presentan, puesto que yo utilizaré un código de Python, usaré esa opción;

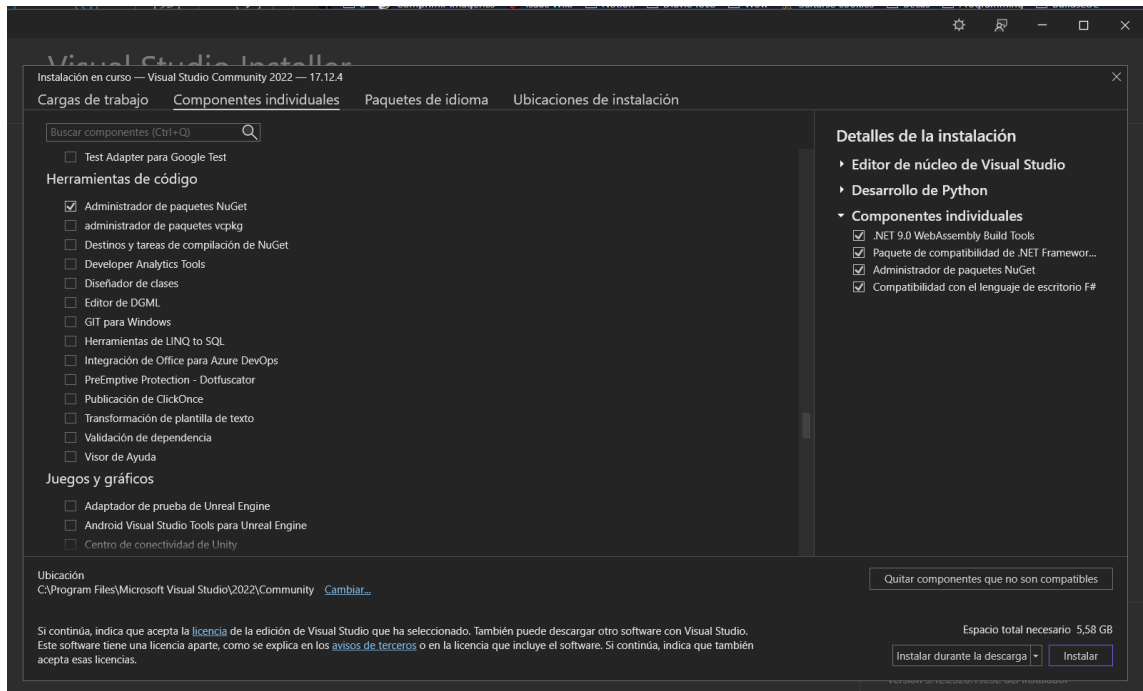


- Además, debajo de este elenco de opciones, nos aparece la presunta ubicación de los archivos que se descargarían, la cual podemos cambiar a conciencia. A la derecha, una opción para quitar aquellos componentes no compatibles con lo que deseemos instalar, y justo debajo, junto al espacio que ocuparía nuestra instalación, en relación a los programas seleccionados, 3 opciones de instalación;

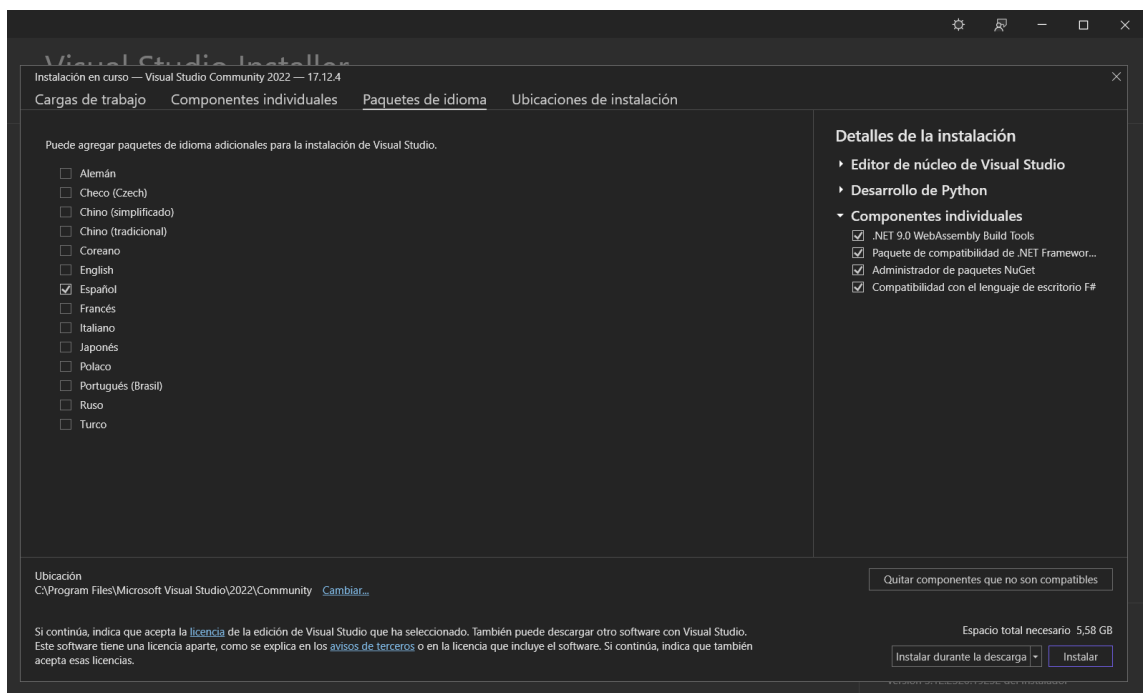


2. En componentes individuales, podemos seleccionar descargas más concretas para operaciones específicas con el IDE, así como en la pestaña anterior, está segmentada por categorías, según el tipo de utilidad de los programas presentados en ella, así como, en según qué casos, información sobre las versiones de aquellos programas que estaríamos instalando, al mismo tiempo de si posee o no soporte, aquella versión.

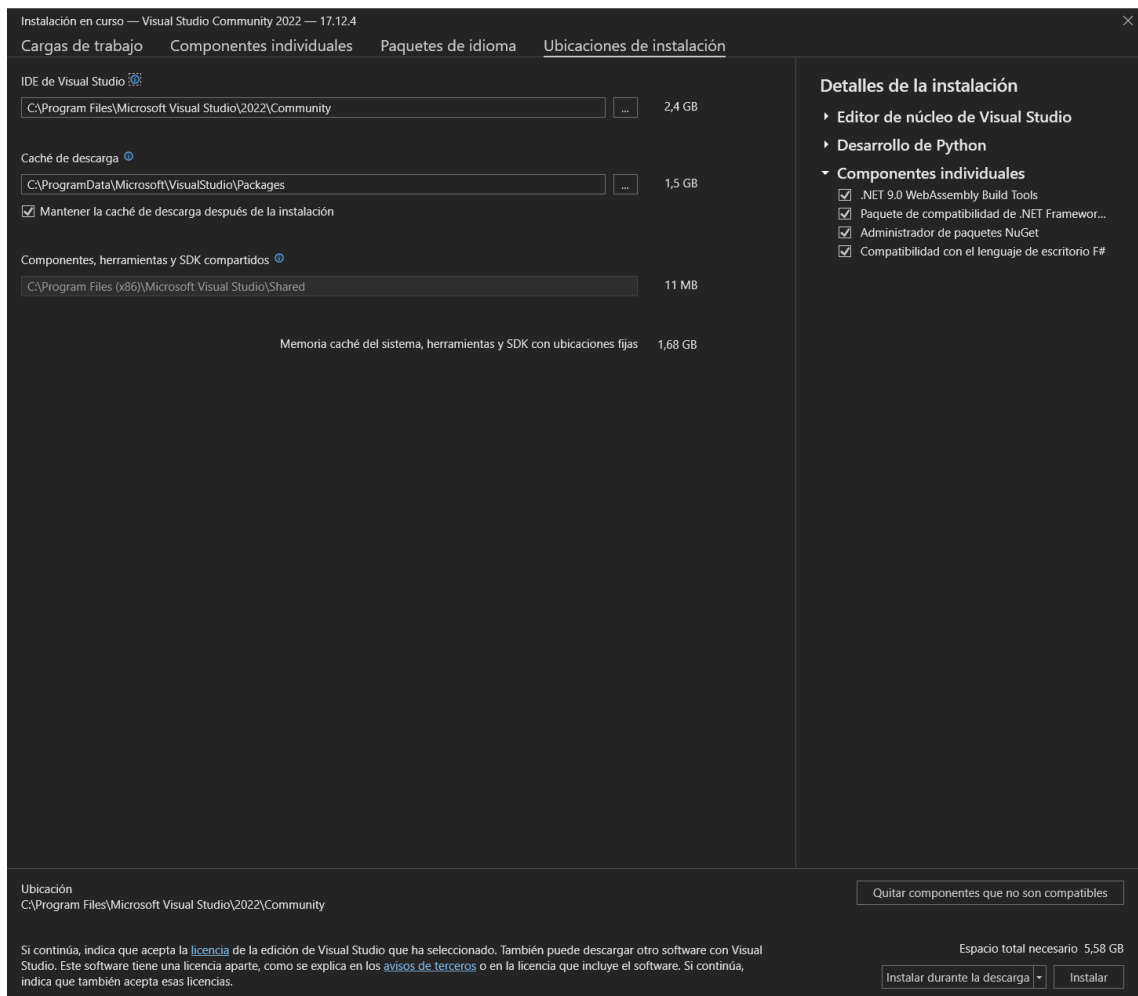




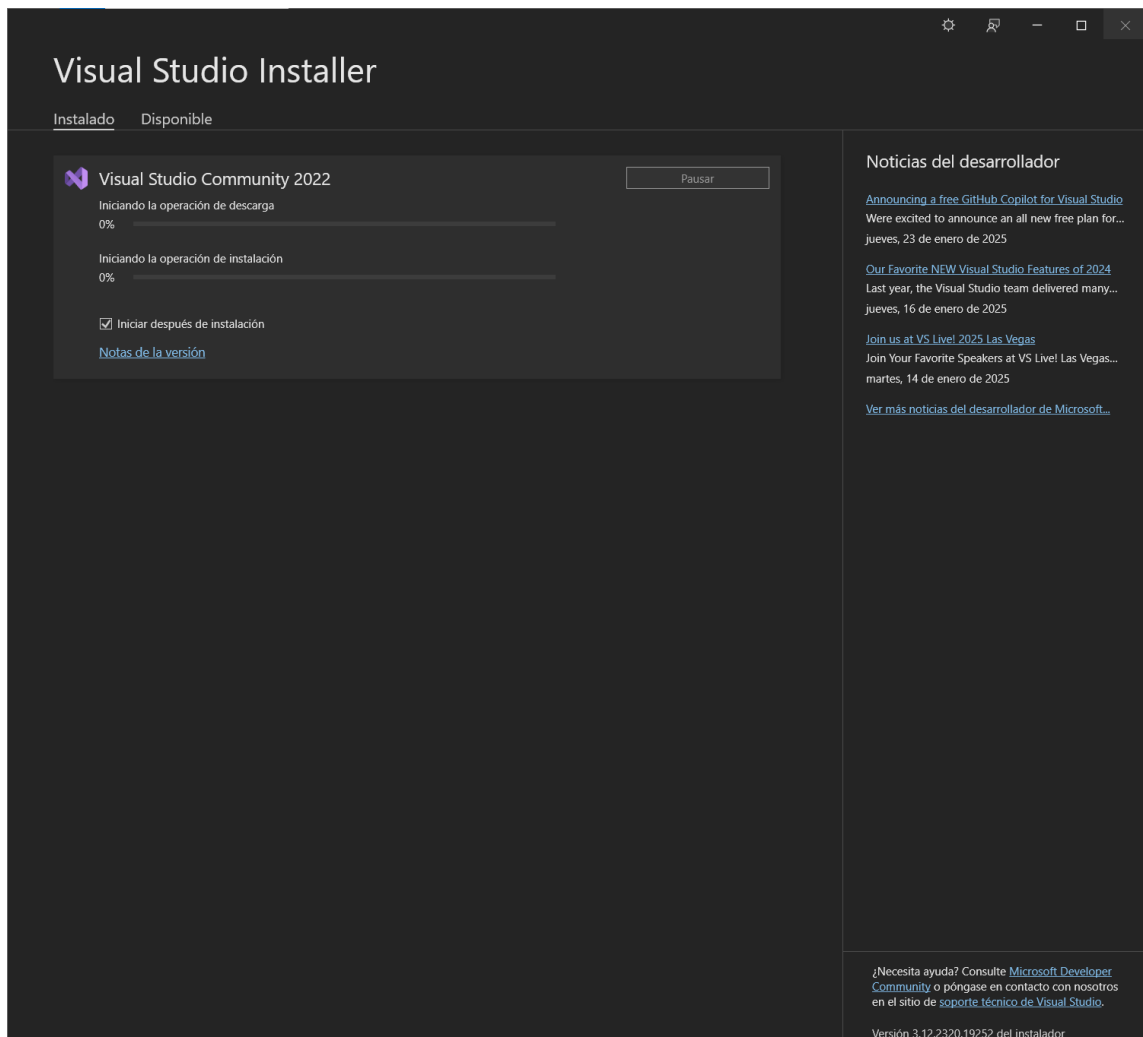
3. En Paquetes de idioma, como puede suponerse, podemos seleccionar en qué idioma/s deseamos que se muestre la información del IDE.



4. En Ubicaciones de instalación, podemos decidir dónde de instalarán, tanto el caché de la descarga, como la propia descarga del IDE, así como los componentes requeridos para la misma.



- Una vez definidas las posibilidades que nos aporta este punto de la instalación, continuamos la instalación.
- En este apartado, además del punto en el que se encuentra, a la derecha podemos ver una especie de “Panel de anuncios”, donde aparecen noticias relacionadas con el IDE, directa o indirectamente.
- También podemos modificar la apariencia del entorno gráfico, presionando en el símbolo del sol que se encuentra arriba a la derecha, junto al símbolo de “enviar comentarios”, minimizar pantalla, abrirlo en modo pantalla completa (en mi caso porque lo tenía en ventana), y cerrar ventana.
- Tras completarse la instalación, podemos cerrar el Visual Studio Installer.

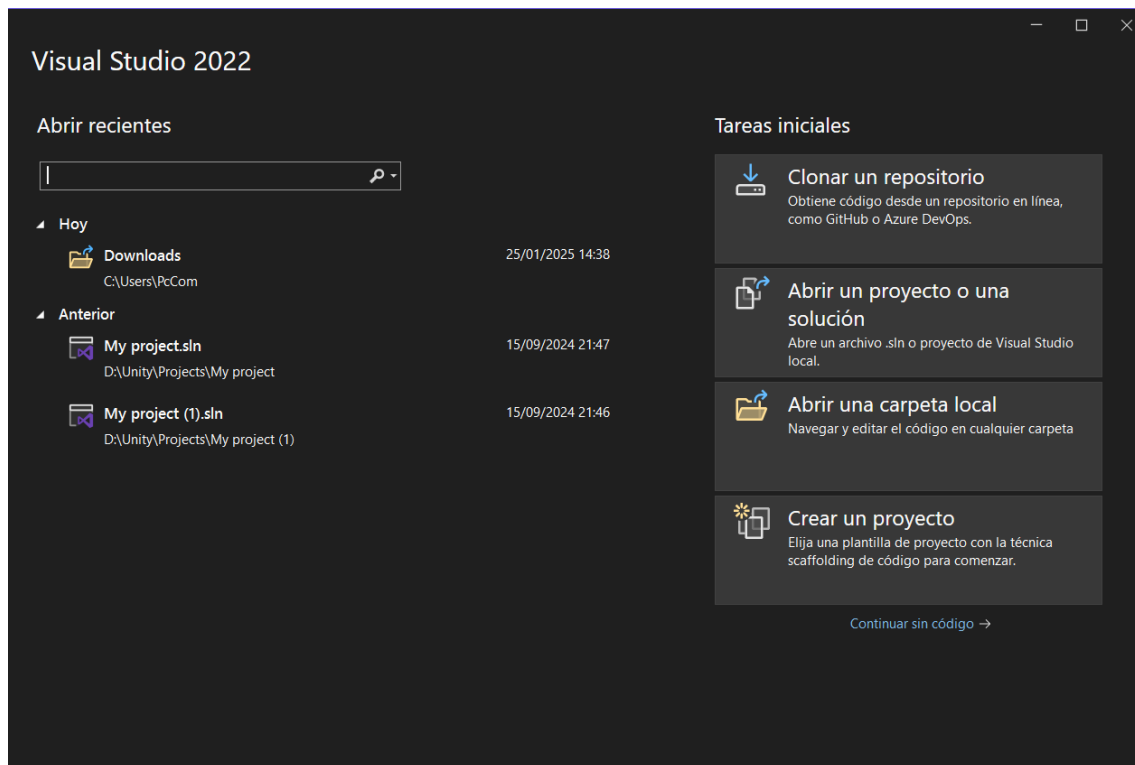


- Una vez finaliza este proceso, se nos permiten varias opciones en dos laterales.

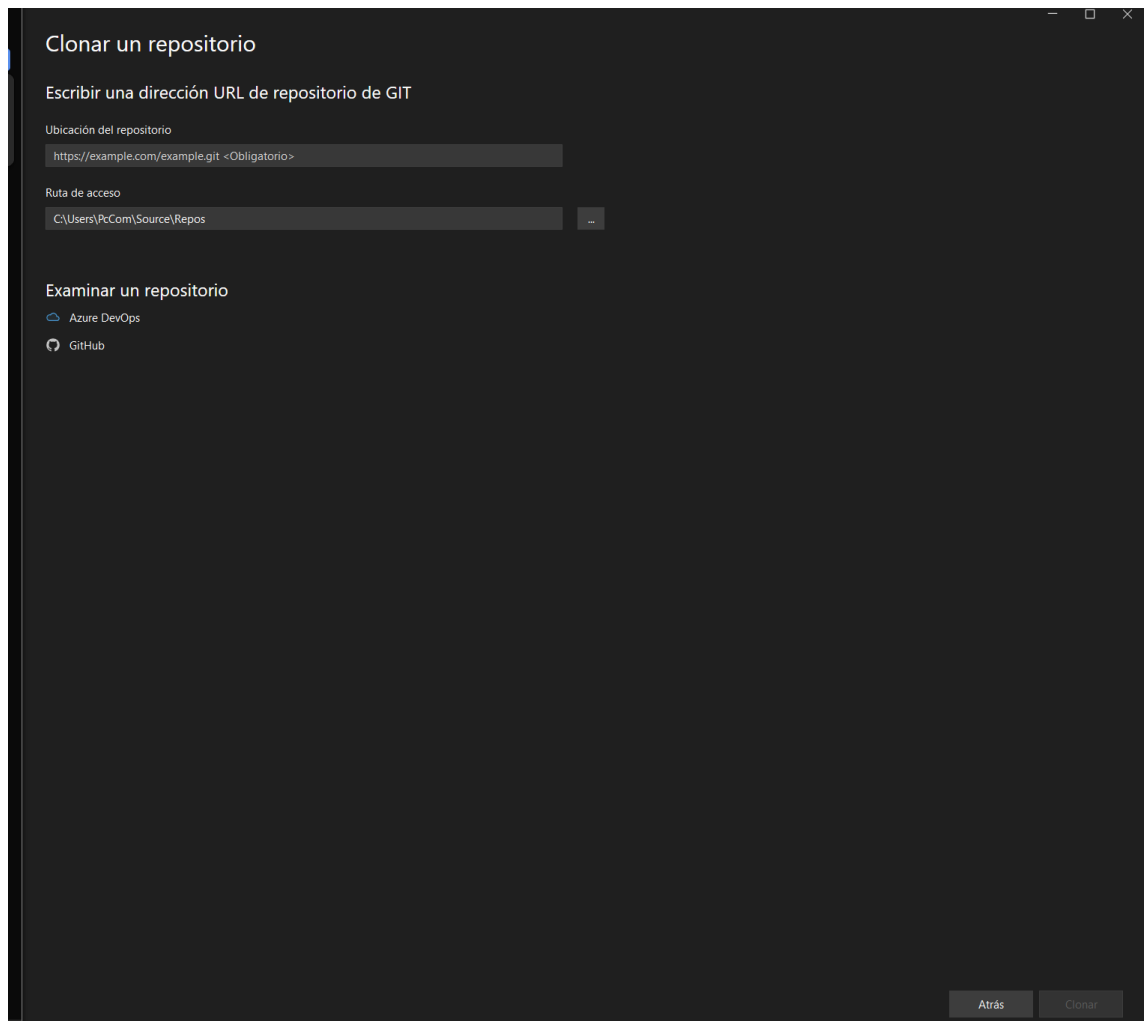
1º Puedes observar los archivos o carpetas que has abierto recientemente

2º Puedes realizar las siguientes operaciones;

- Clonar un repositorio, obteniendo el código desde GitHub o Azure DevOps.
- Abrir un proyecto que proceda de Visual Studio, o un .sln, al que se refieren como “solución”.
- Abrir una carpeta particular.
- Y comenzar un nuevo proyecto.
- También se nos permite continuar en este menú de tareas iniciales, sin ningún tipo de código en un primer momento.

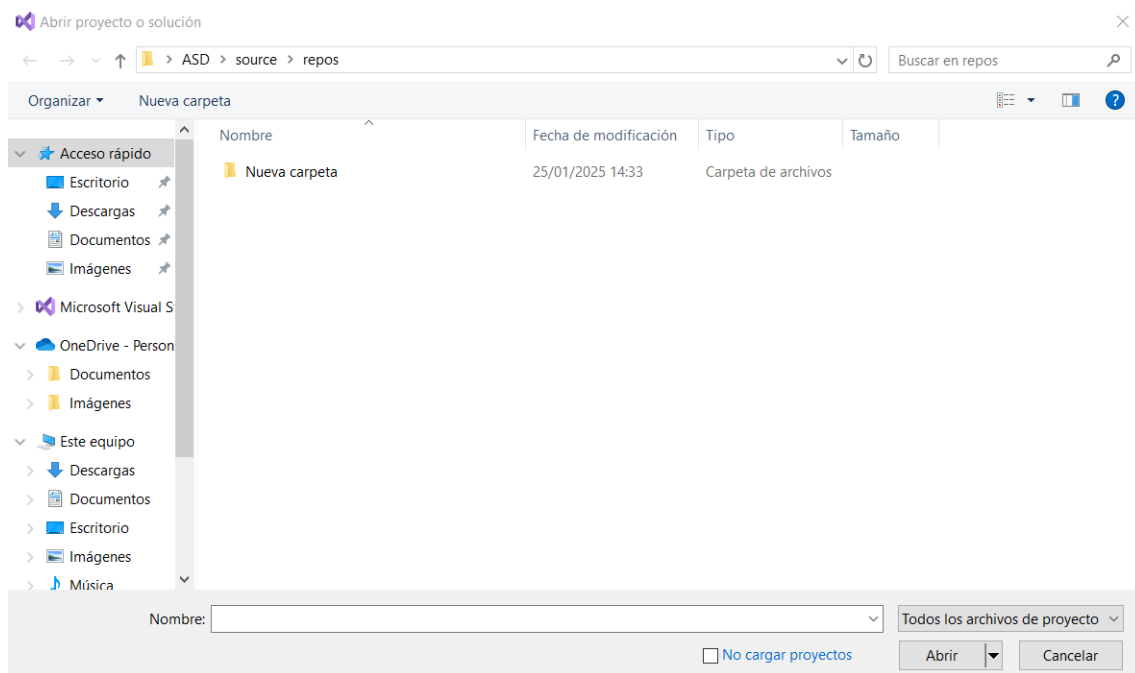


****Por un lado si presionamos “Clonar un repositorio”;**



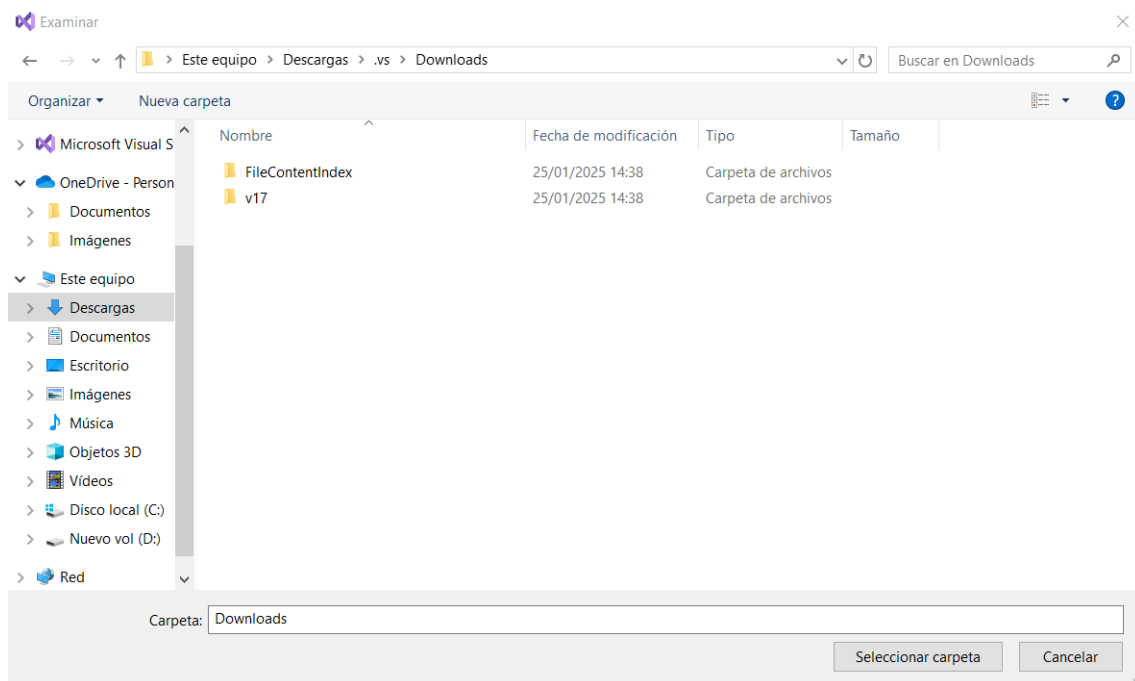
- Se nos permite escribir la dirección url del repositorio en GIT, con la ubicación del mismo rep. Así como la ruta donde se encuentre, en el caso de que tuviéramos a alguno que deseásemos clonar, únicamente con rellenar esos dos espacios, y seleccionar “Clonar”, abajo a la derecha, comenzaría este proceso.

** Si presionamos “Abrir un proyecto o solución”, podemos abrir, en el explorador de archivos, en una ruta específica abierta por el propio IDE, los proyectos (ya sean archivos, o carpetas con archivos dentro), para comenzar a operar con ellos. En mi caso cree una carpeta nueva, esto aparece en blanco si instalamos por primera vez VS IDE.

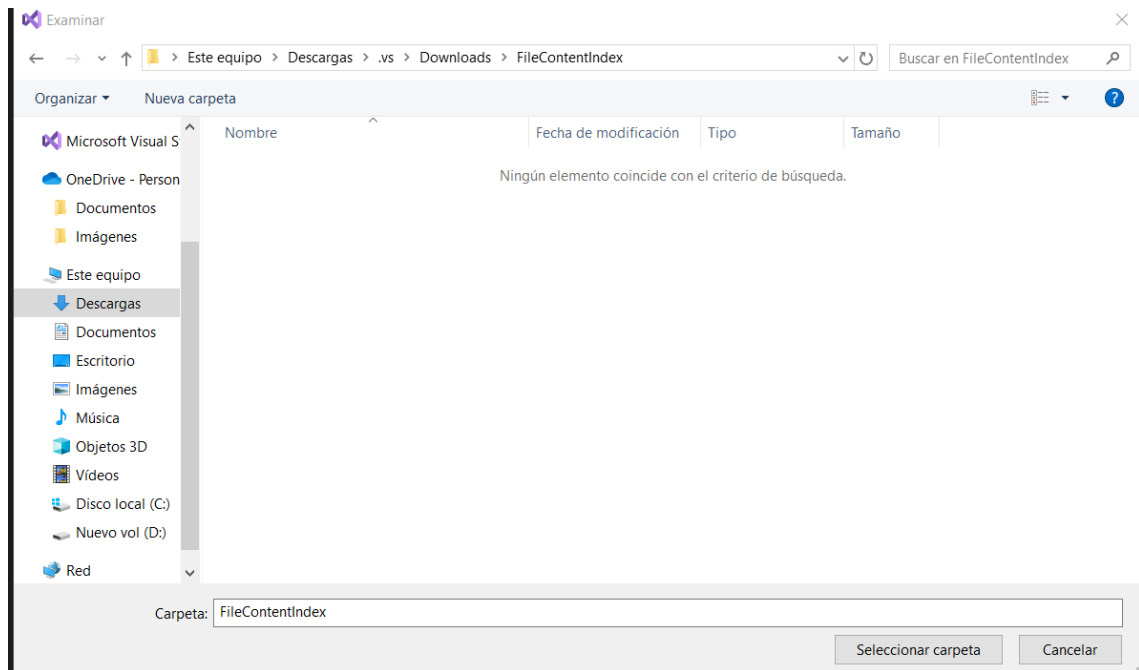


**** Si seleccionamos “Abrir una carpeta local”, operará de manera similar al anterior punto, pero en este caso, abre la carpeta “Descargas”, se denota una carpeta llamada “.vs” que este programa crea automáticamente, con el siguiente contenido;**

1º Carpeta Downloads, con el siguiente contenido;

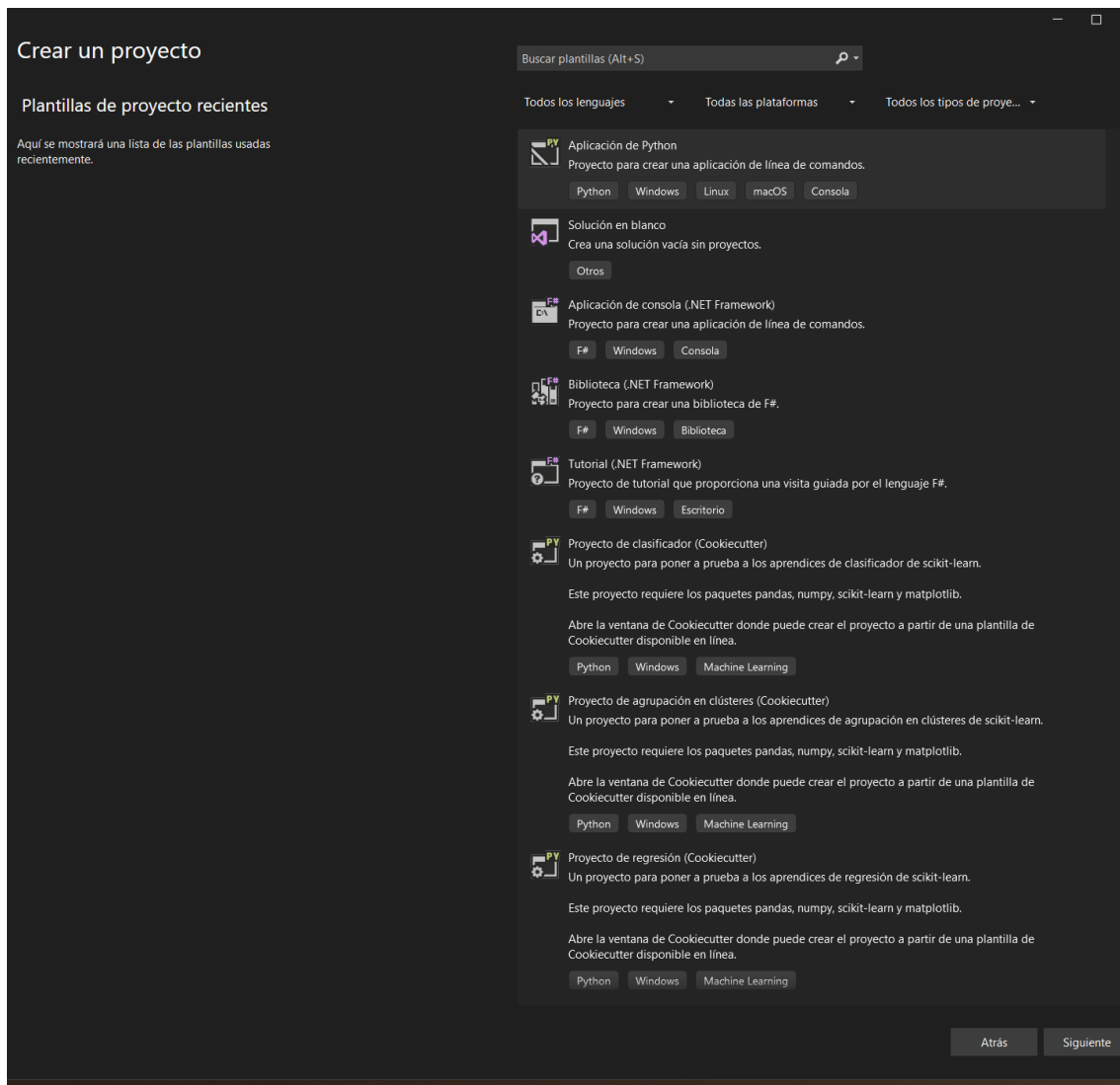


- Abrimos carpeta “FileContentIndex”, que aparece vacía;

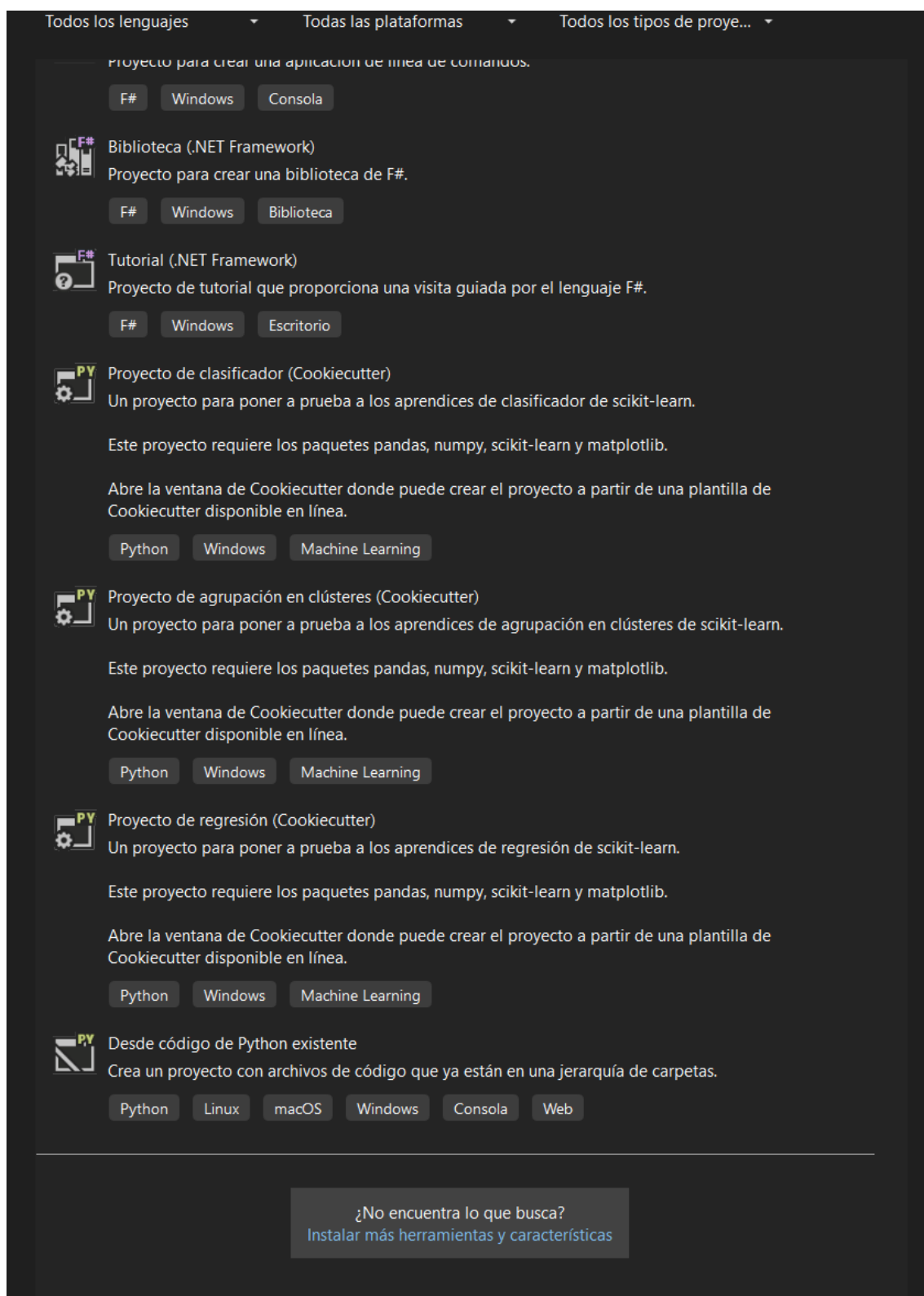


- Por otro lado, la carpeta v17, que tampoco posee contenido en sí misma.

**Con respecto a la opción “Crear un proyecto”, a la izquierda nos permite ver plantillas de proyectos recientes, al no tener ninguno no aparecerá nada. A la derecha, contenido relacionado con las plantillas, pudiendo insertar el nombre de la plantilla deseada de forma especificada en la barra de búsqueda, y bajo a esta, un cúmulo de plantillas para distintos propósitos.

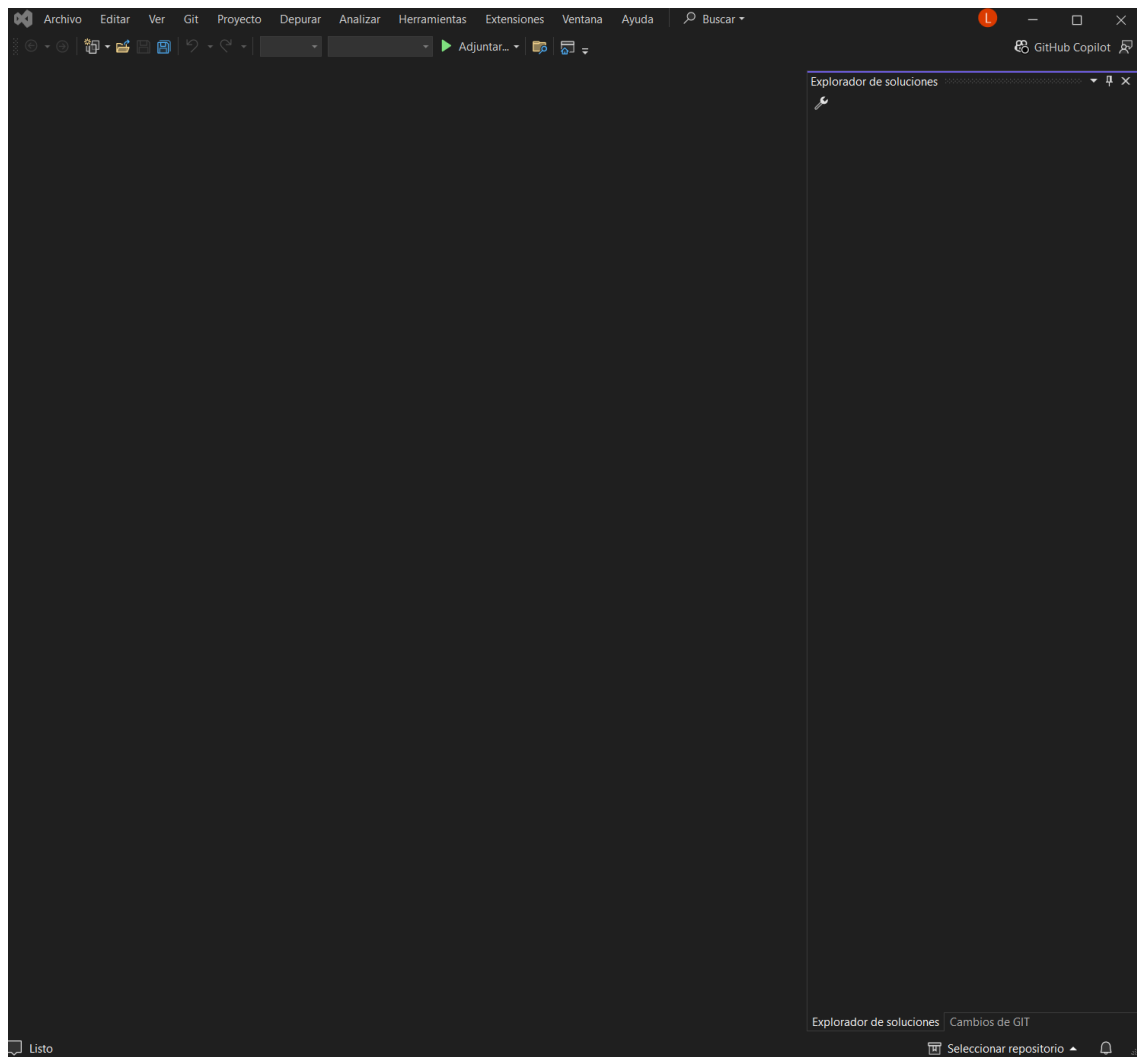


- Suponiendo que no encontremos las plantillas que deseamos, tendríamos una opción para Instalar más herramientas y características ;



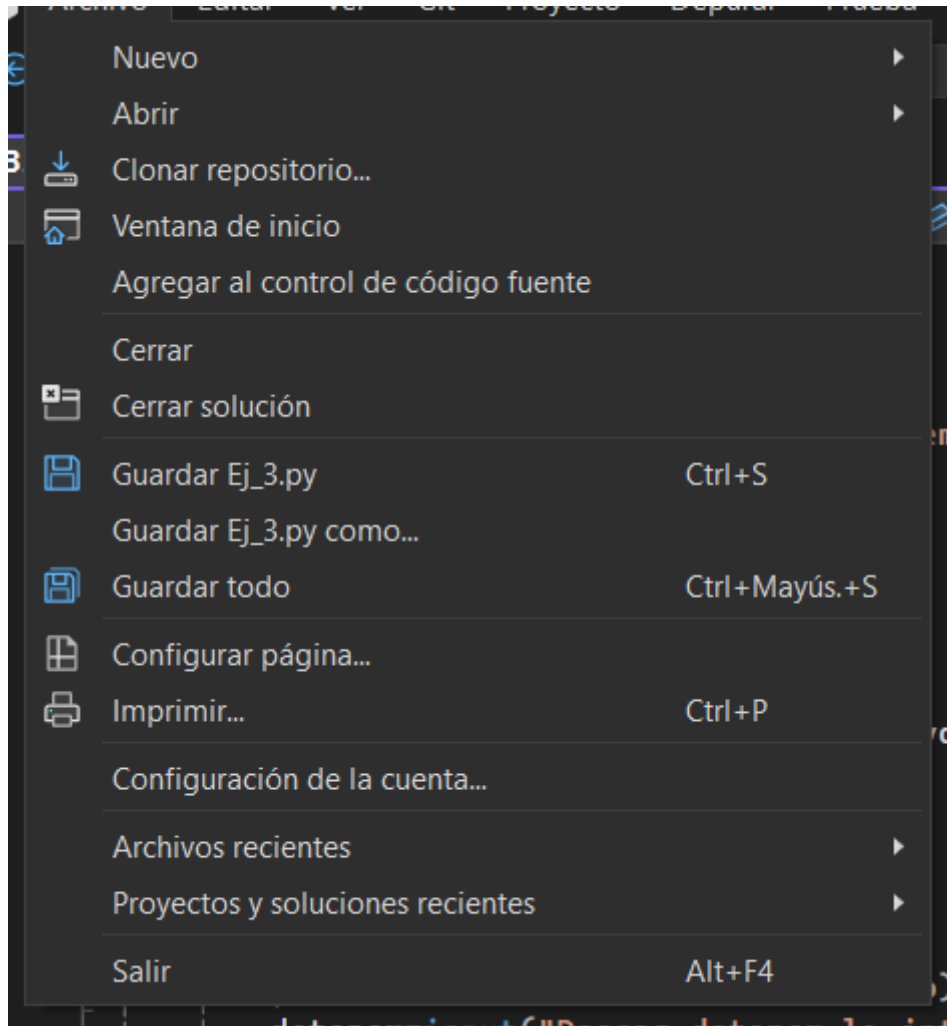
** A continuación, con la última opción “Continuar sin código”, se abre el apartado funcional del propio IDE, donde realizaremos los demás procesos deseados para completar el hito.

- Una vez seleccionada esta opción, se abre la siguiente ventana;



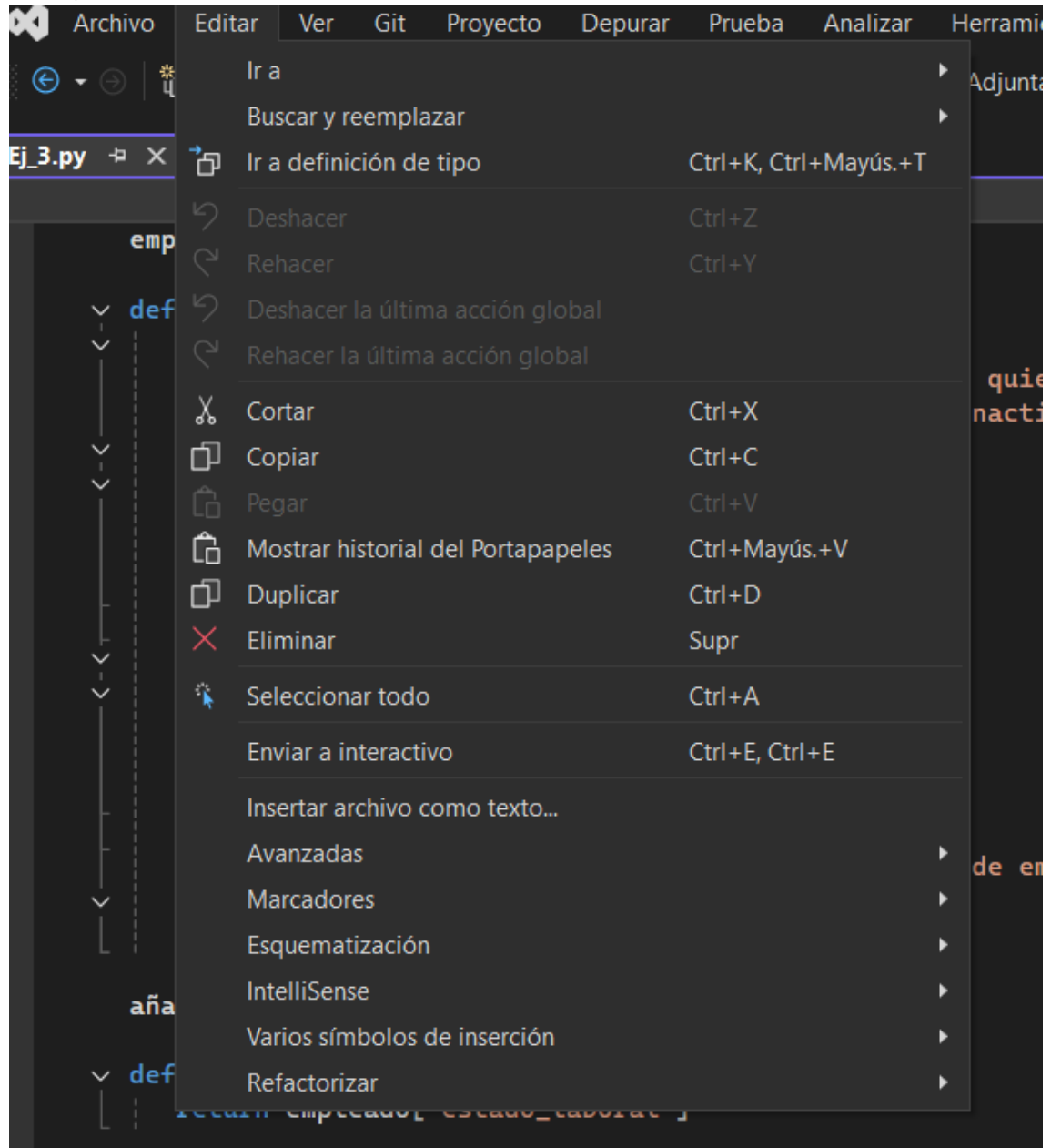
- En la primera barra superior de la ventana vemos las siguientes opciones;

1. Archivo;



- Podemos realizar todas las operaciones de esta imagen en relación al archivo en el que estaríamos trabajando en este momento, todas las opciones son reconocibles en este contexto, a excepción quizá, de “Clonar repositorio”, pues estaríamos tratando de opciones entorno al archivo. En ese caso, simplemente te deriva a la ventana de “Clonar repositorio” que hemos tratado en apartados anteriores de este documento.

2. Editar;



-En él, hay algunas opciones que podrían no ser reconocibles, o menos intuitivas, con respecto a estas;

1º Ir a definición de tipo te permite dar con una definición de tipo de alguna clase de objeto o clase, en nuestro caso puesto que el código no tiene clases, no aparecerá nada en particular.

2º Buscar y reemplazar, posibilita buscar un apartado del código que deseemos buscar, y reemplazarla con otro conjunto de código, esto es especialmente útil si el volumen de nuestro código es muy amplio.

3º Enviar interactivo. Nos permite utilizar la consola para ver resultados e interactuar con ella.

4º Marcadores, que nos permite colocar y gestionar marcadores a lo largo del código

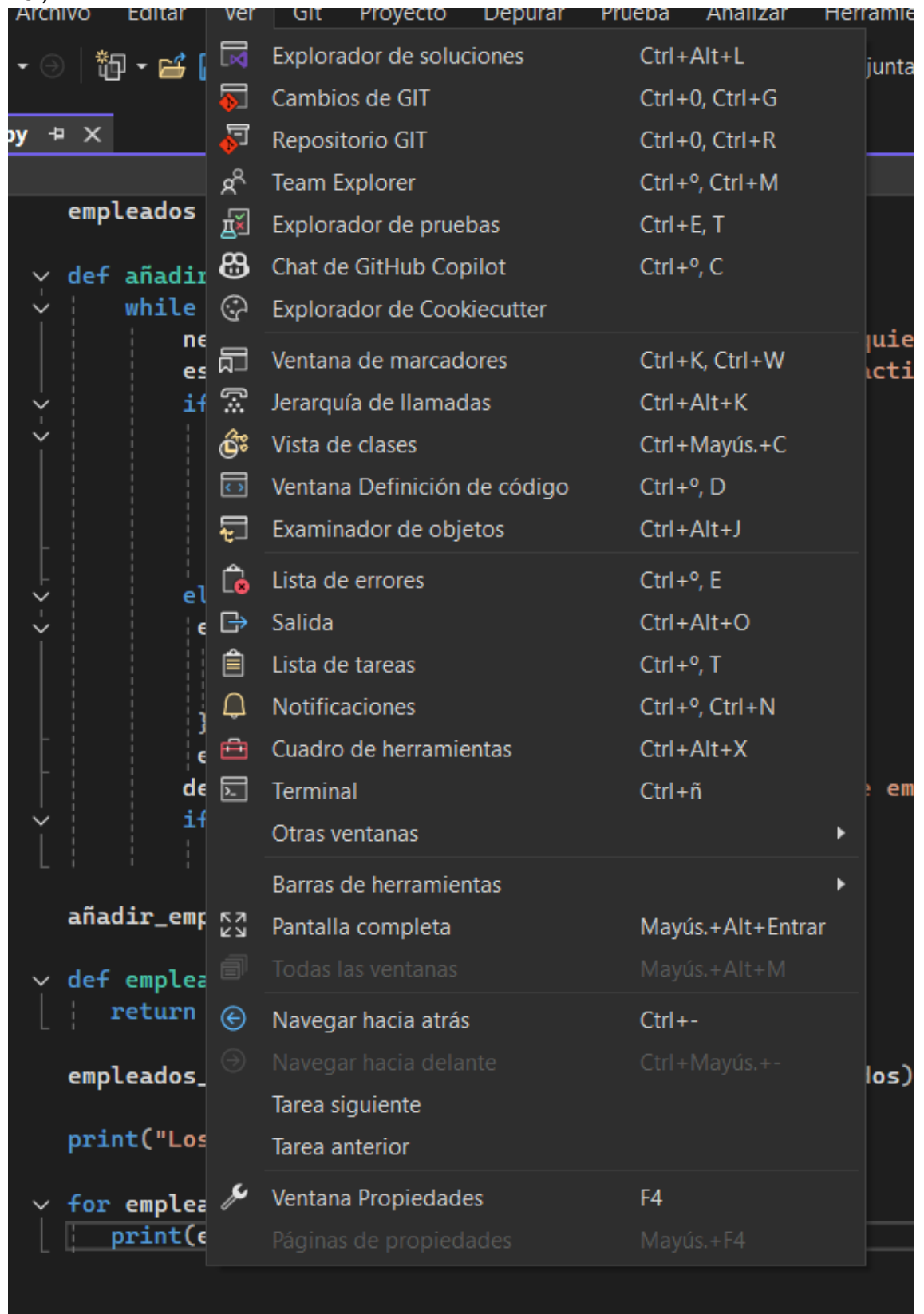
5º Esquematización, nos permite manipular conjuntos de código, pudiendo plegar ciertas partes o expandirlas según las necesidades que tengamos a la hora de entender o poner en perspectiva el código realizado.

6º IntelliSense, es una especie de asistencia de código inteligente, sugerencias de código, parámetros esperados si creamos una función, etc.

7º Varios símbolos de inserción, permite insertar, editar, eliminar, en sí operar con distintos puntos de inserción al mismo tiempo en la zona de código.

8º Refactorizar, nos modifica el código, ya sea para hacerlo más limpio, o simplemente ordenarlo si tenemos código repetido o manejado de forma poco ordenada.

3. Ver;



- Las más destacables son:

1º Explorador de Cookiecutter; Genera plantillas a través de plantillas predefinidas.

2º Jerarquía de llamadas; Muestra cómo se llaman las funciones o métodos dentro de un programa, es decir, desde qué métodos o funciones se realizan qué llamadas.

3º Vista de clases; Te permite explorar y ver las clases actuales, con sus objetos, métodos, propiedades, y relaciones de herencia.

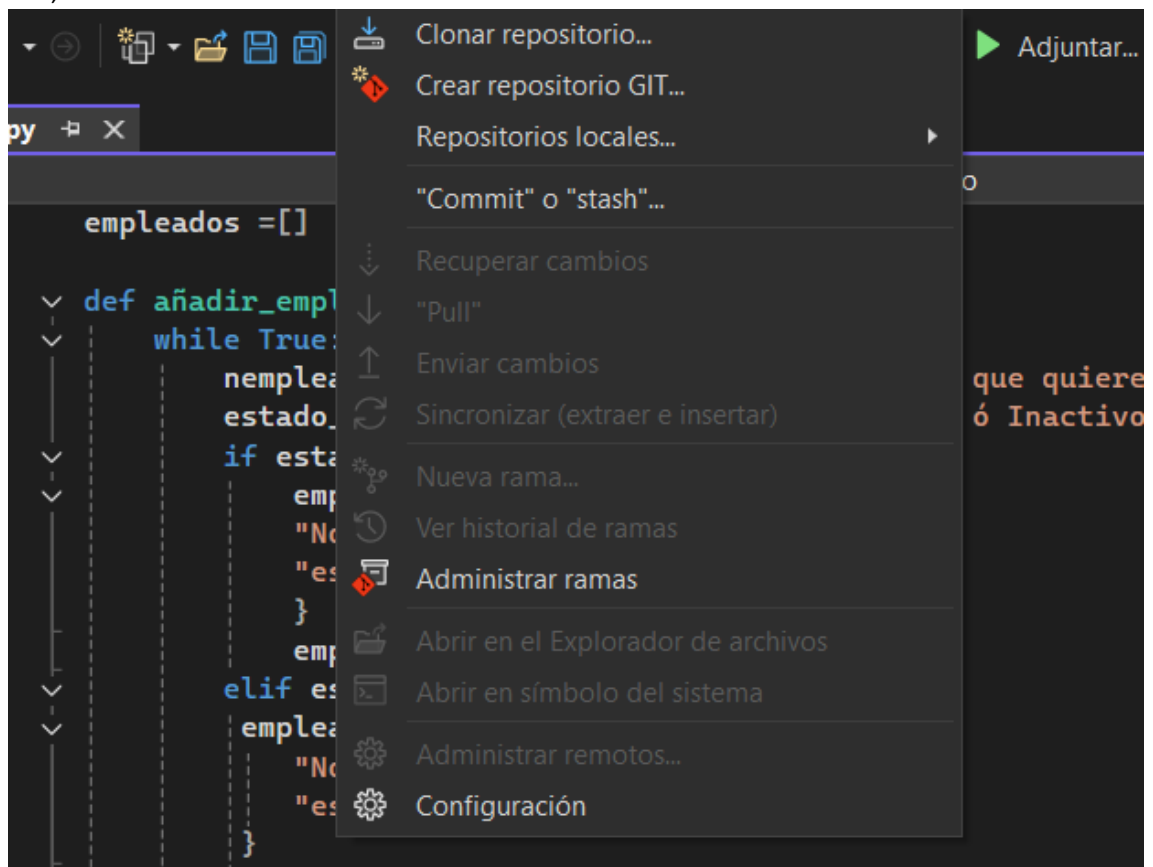
4º Ventana Definición de código; Muestra dónde se han definido ciertas entidades, si por ejemplo al usarla, seleccionas una función o una clase, te lleva a donde se ha definido la función o la clase misma.

5º Examinador de objetos; Permite inspeccionar y explorar objetos en tiempo de ejecución, es decir, en tiempo real, pudiendo ver el estado de, por ejemplo, las variables.

6º Lista de tareas; muestra las tareas pendientes o comentarios del código, pudiendo mostrar qué partes del código necesitan ser modificadas o especificadas.

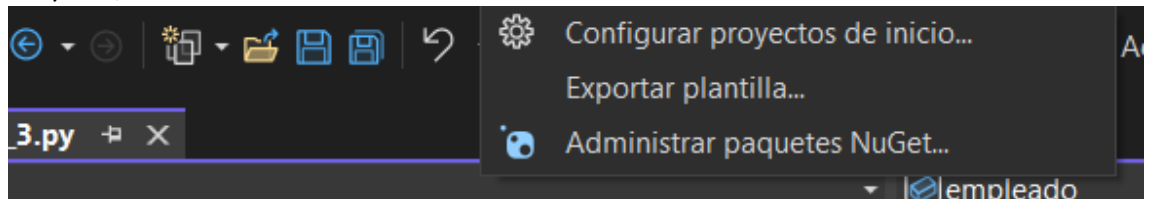
7º Cuadro de herramientas; Te permite ver los componentes y controles disponibles para ser manejados, sobre todo se usa para diseño de aplicaciones, puesto que ahí podrías arrastrar botones, cuadros de texto, etc.

4. Git ;



- Constituye en opciones para poder administrar un repositorio y sus ramas en git.

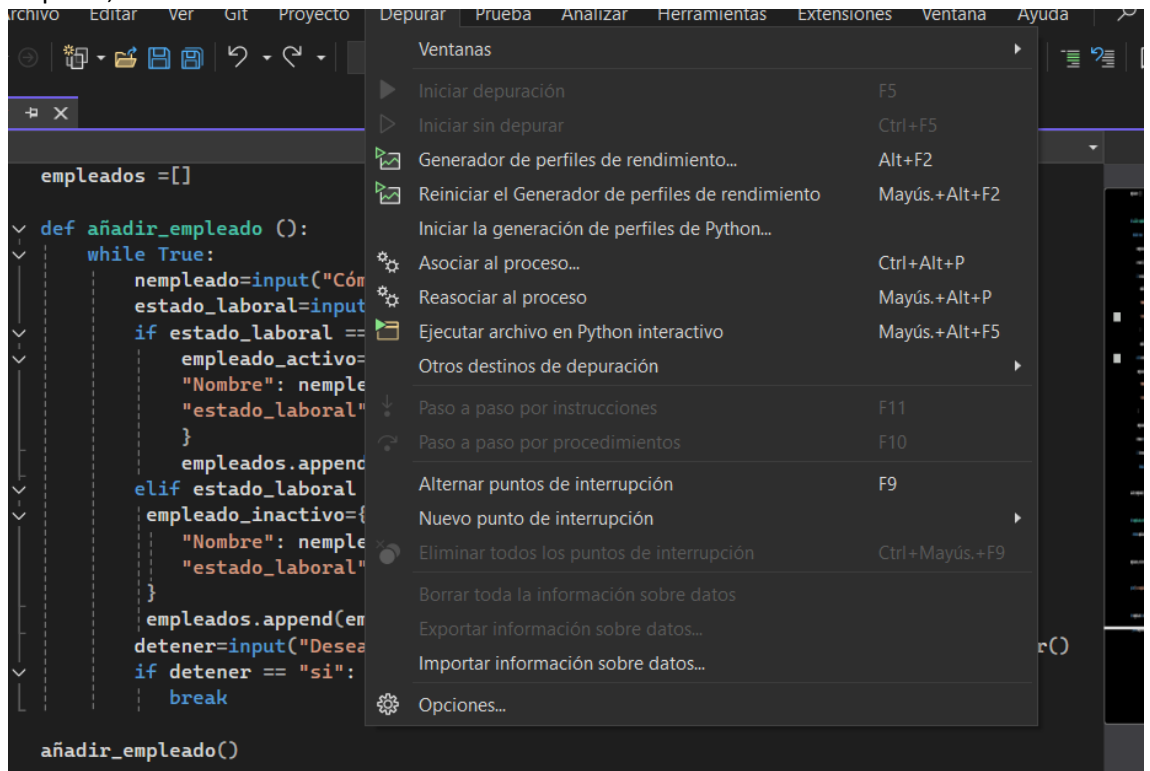
5. Proyecto;



- La más destacada es:

1º Administrar paquetes NuGet; Te permite saber qué paquetes son necesarios para que tu código funcione correctamente.

6. Depurar;



- Los más destacados fueron:

1º Generador de perfiles de rendimiento; Permite analizar el rendimiento de la aplicación mientras se ejecuta, identificando cuellos de botella, tiempos de ejecución muy largos, etc.

2º Reiniciar el Generador de perfiles de rendimiento; Simplemente la opción de reiniciar la previa.

3º Asociar al proceso; Permite conectar el depurador de Visual Studio a un proceso que se encuentre ya en ejecución.

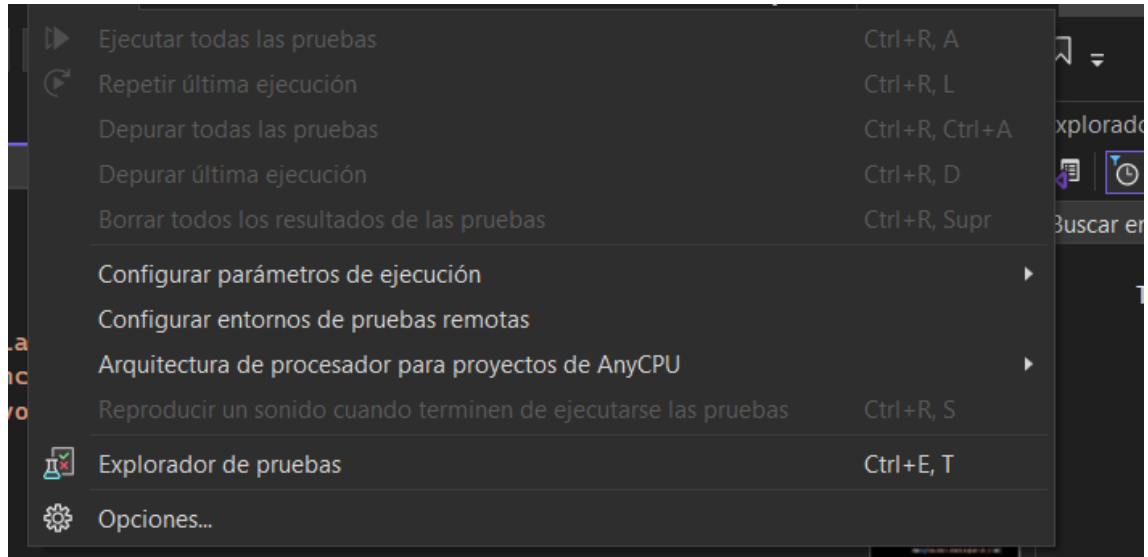
4º Reasociar proceso; Implica el mismo proceso que el anterior pero simplemente a un proceso al que ya te habías asociado con anterioridad.

5º Ejecutar archivo en Python interactivo; Implica ejecutar un script de Python sin tener que salir del IDE en un entorno interactivo dentro del Visual Studio.

6º Nuevo punto de interrupción; Esto detiene el recorrido de tu código en un punto en particular, viendo el funcionamiento del flujo de tu programa.

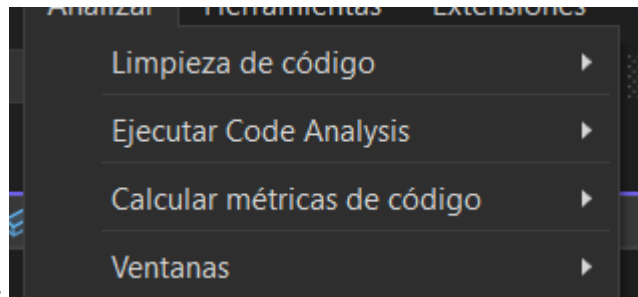
7º Importar información sobre datos; Importa datos para la depuración posterior de estos, desde archivos externos.

7. Prueba;



- Destaca:

1º Arquitectura de procesador para proyectos de AnyCPU; Permite que se ejecute el código en cualquier tipo de arquitectura del procesador, ya sea de 32 bits o de 64.

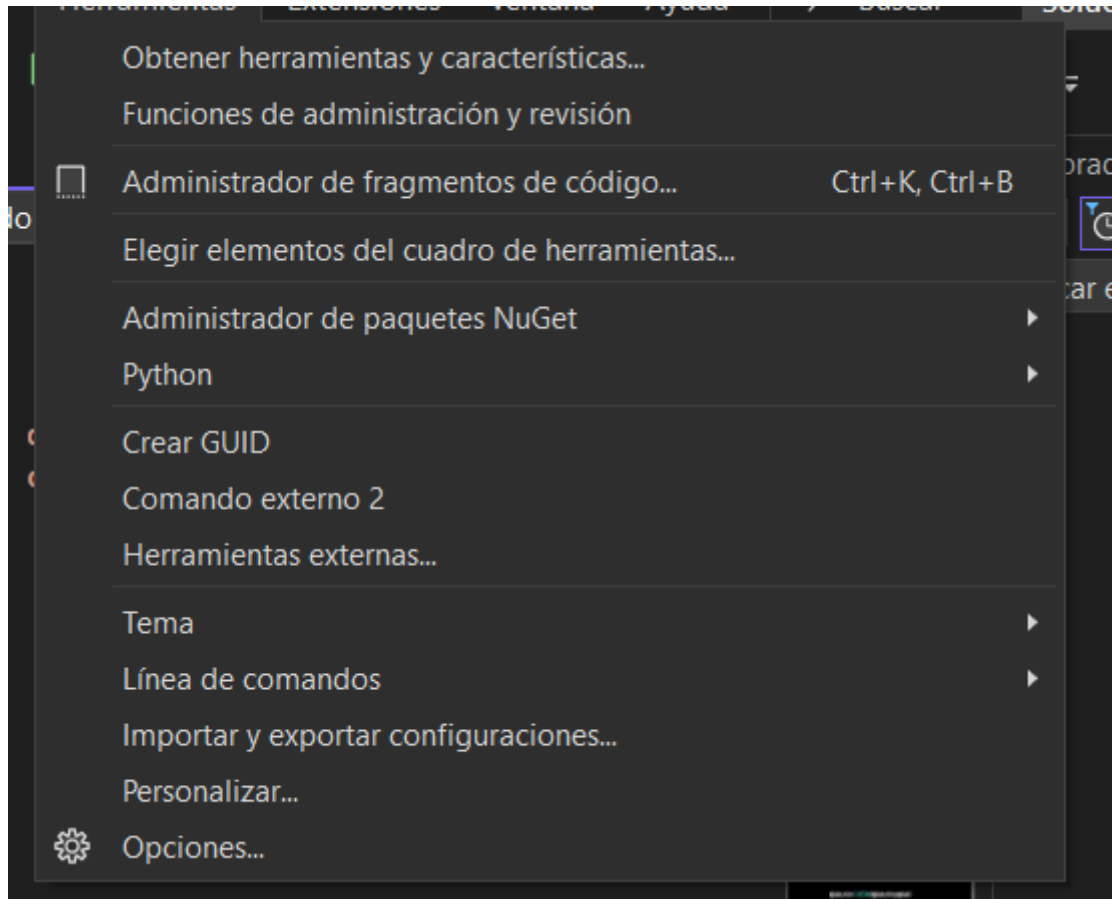


8. Analizar;

- Destaca ;

1º Calcular métricas de código; Son indicadores numéricos que valoran distintos aspectos del código, tales como si son fácilmente mantenibles, sencillos de entender, etc.

9. Herramientas;

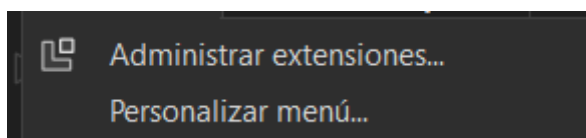


- Destacan;

1º Crear GUID; O Identificador Único Global en español, es un identificador que asegura que no haya duplicados en aplicaciones distribuidas o bases de datos, por ejemplo, registros de una base de datos, sesiones de usuario, etc.

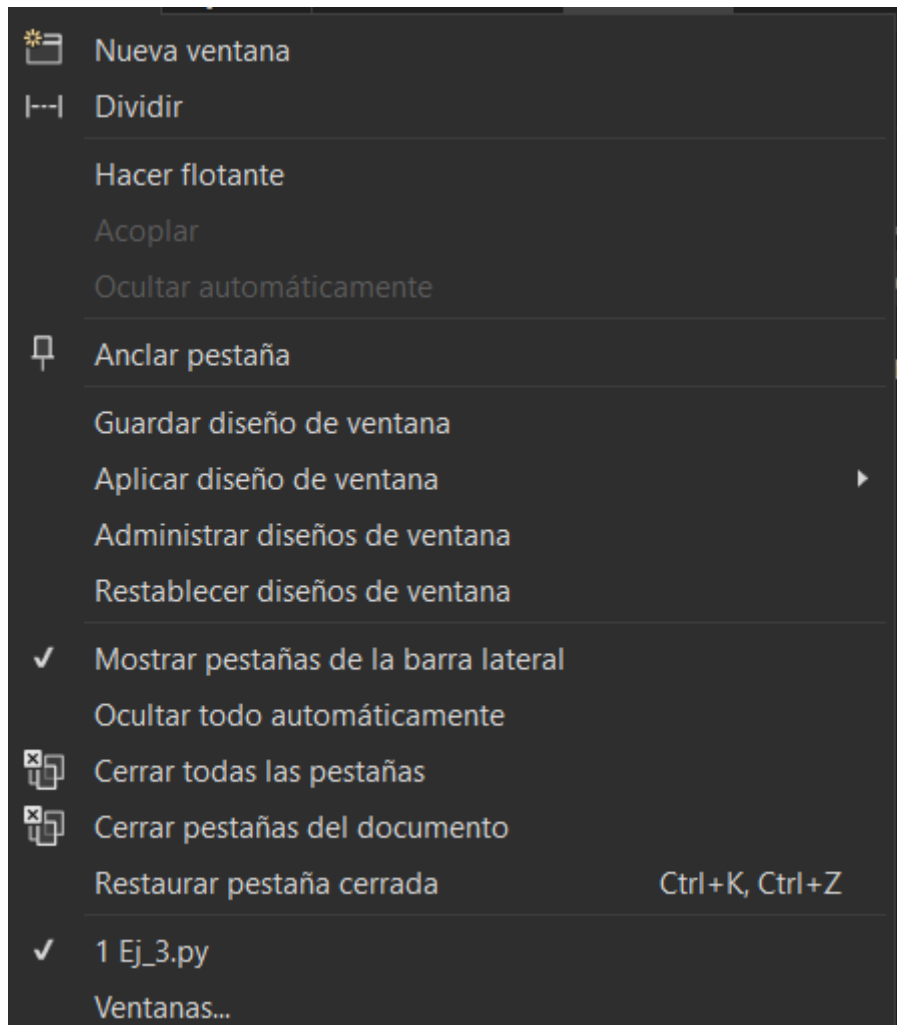
2º Comando externo 2; Permite ejecutar comandos externos sin tener que salir de Visual Studio.

10. Extensiones;



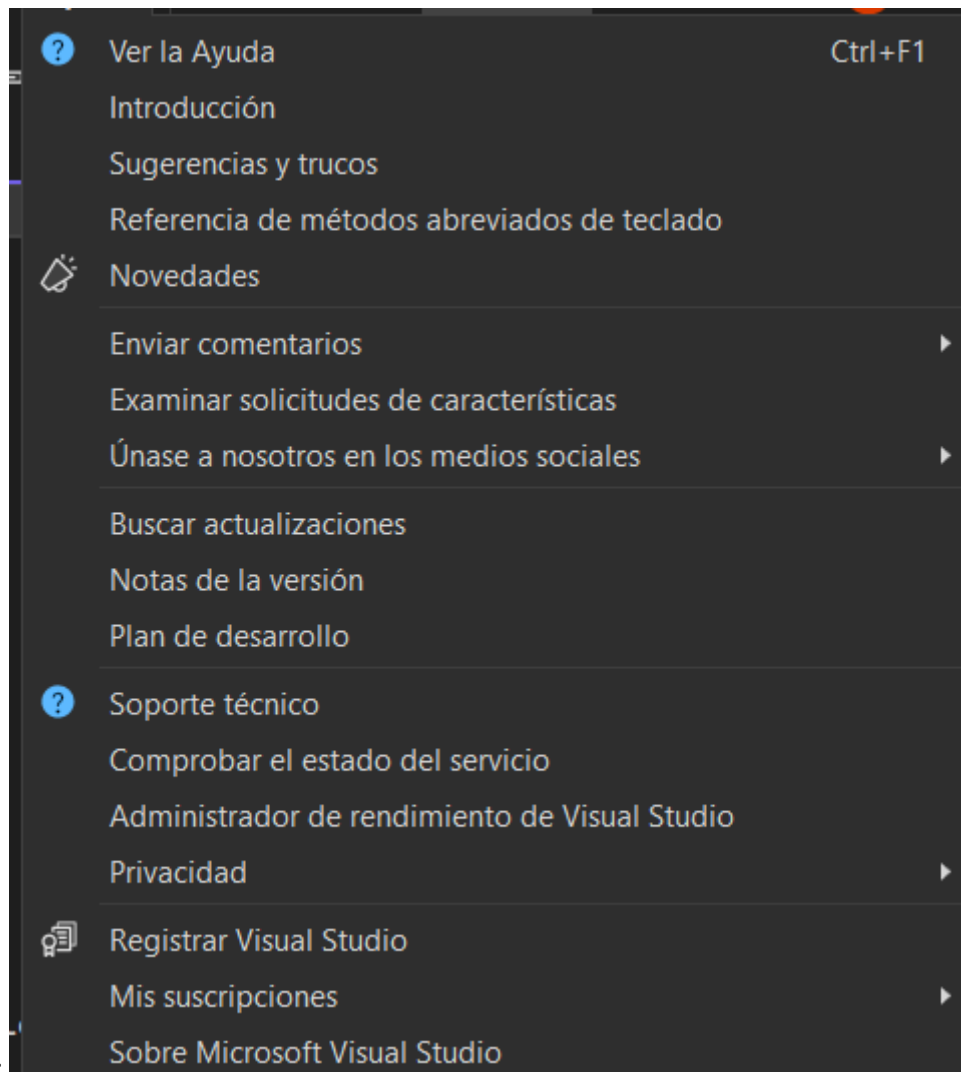
- Ninguna de las dos opciones me parece poco intuitiva como para dedicarles definiciones.

11.Ventana;



- Destaca:

1ºHacer flotante; separa el panel o ventana donde estés, de la principal, y pasa a tratarla como una totalmente independiente, con todas las opciones relacionadas con las ventanas individuales, lo que te da más flexibilidad con respecto al código o proyecto.



12. Ayuda;

- Destacan:

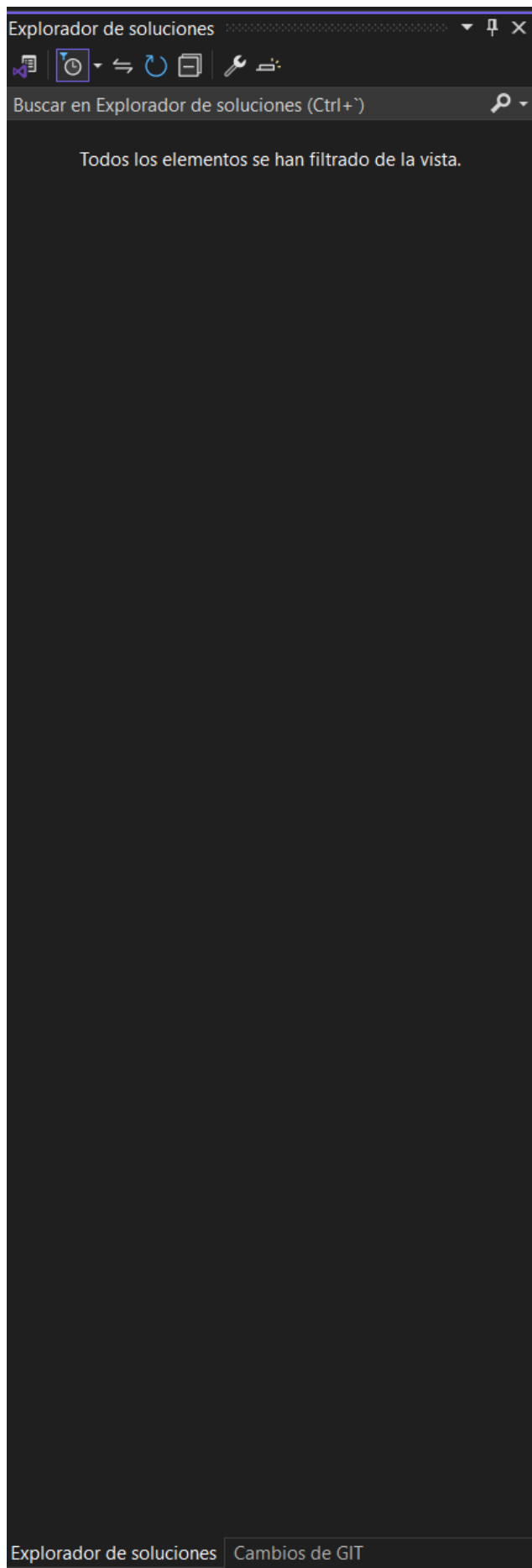
1º Referencia de métodos abreviados de teclado; Tan solo es una lista de métodos a los que puedes acceder a través de las combinaciones de teclas a las que se refiere como “métodos abreviados de teclado”.

2º Plan de desarrollo; Te deriva a una serie de consejos para poder lograr un eficiente desarrollo de ciclo de vida del software.

- En la barra justo debajo de la que acabamos de analizar se hayan los siguientes símbolos;

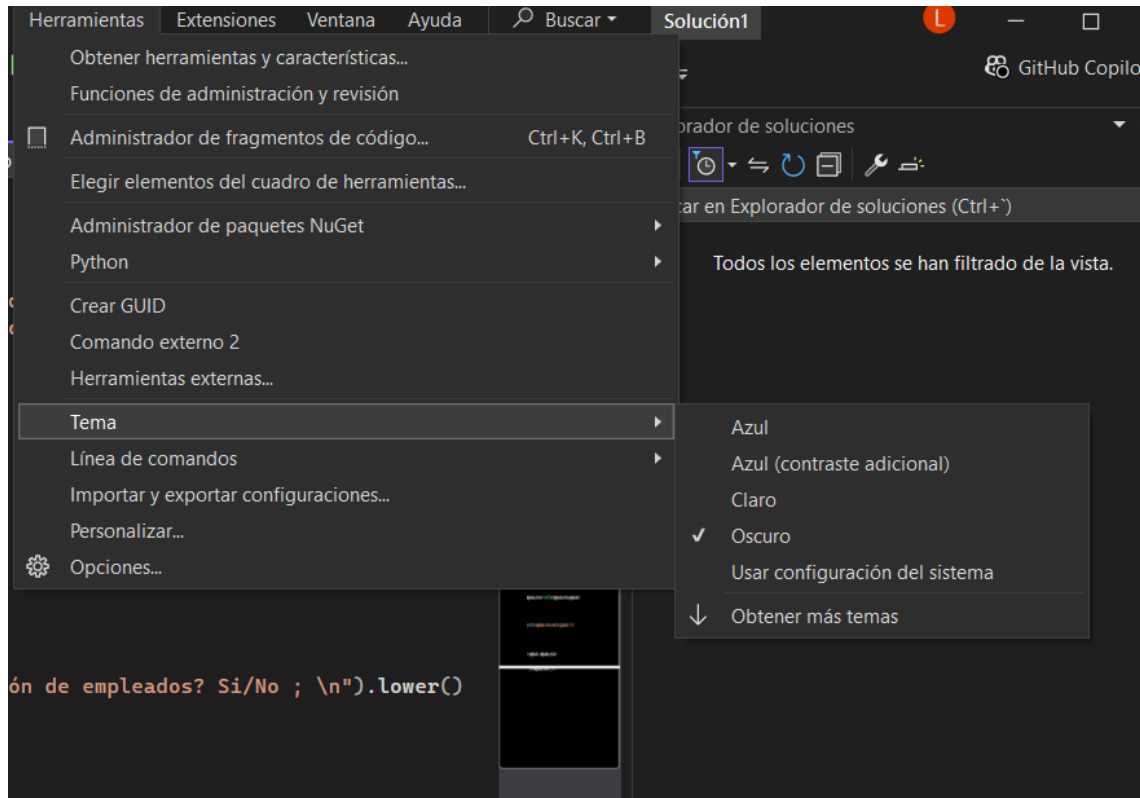


- 1º Podemos viajar entre momentos previos y posteriores del código.
 - 2º Crear nuevo proyecto.
 - 3º Abrir un archivo.
 - 4º Guardar y Guardar cómo.
 - 5º Deshacer y rehacer cambios.
 - 6º Ajustar el tamaño de los caracteres empleados.
 - 7º Conectarnos a un proceso ya en funcionamiento a través de “Adjuntar”.
 - 8º Buscar en archivos-
 - 9º Volver a la pantalla de inicio del IDE.
 - 10º Alternar corrector ortográfico.
 - 11º Mostrar información rápida.
 - 12º Marcar determinadas líneas de código como comentarios.
 - 13º Quitar esta misma marca de comentario.
 - 14º Añadir marcadores así como administrarlos.
- En lo que respecta al apartado derecho de la interfaz tenemos el “Explorador de soluciones”. Este nos muestra los archivos y proyectos dentro de una solución, pudiendo verlo de manera clara y estructurada;

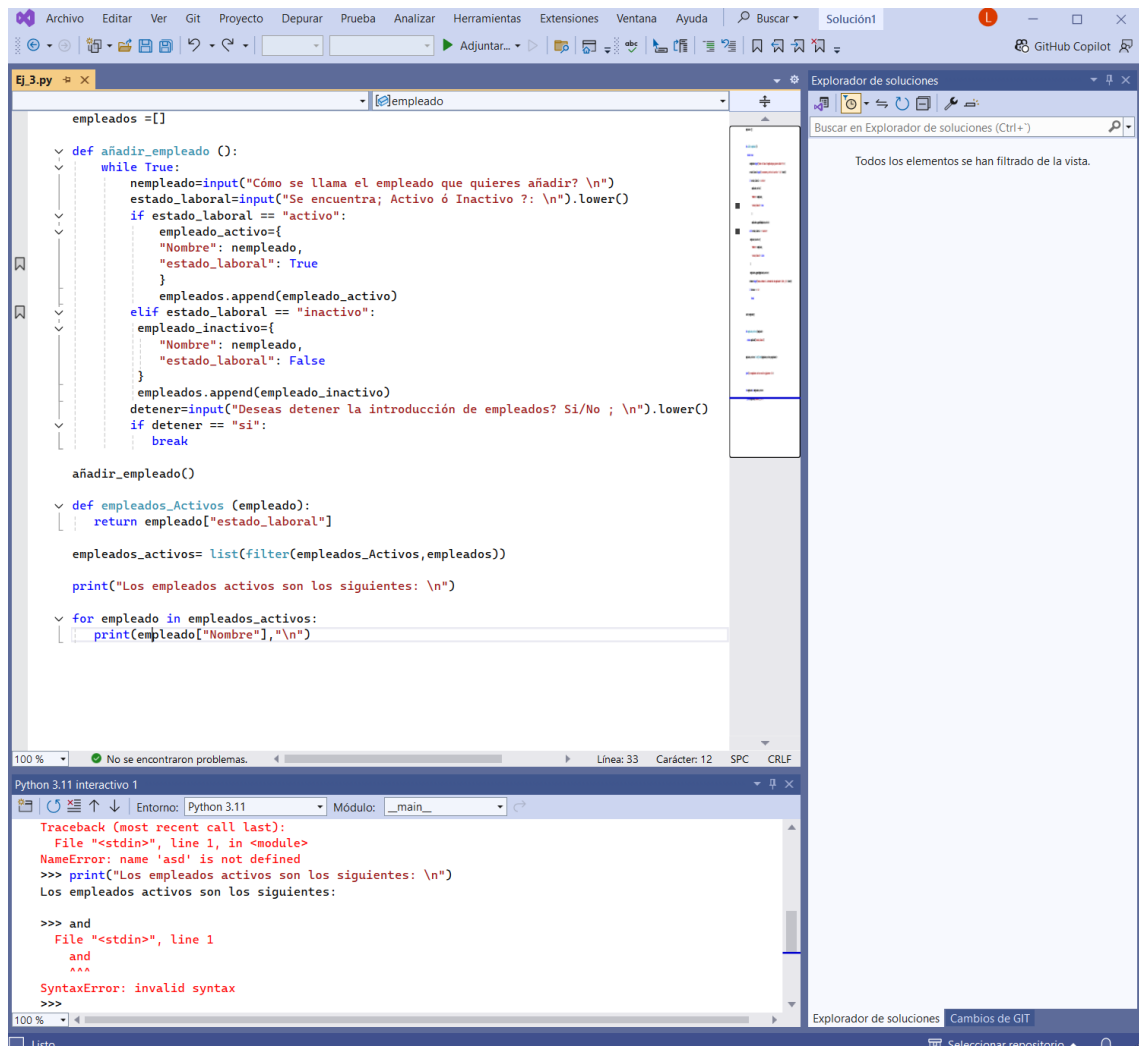


Modifica la apariencia del entorno de desarrollo.

-Con respecto al punto de cambio de tema del IDE, completando en este punto ese apartado del hito, he cambiado el estilo a “azul” en la siguiente ventana;



- Pasando a tener este estilo;



Evalúa entornos integrados de desarrollo, analizando, sus características para editar código fuente y generar ejecutables.

1. Visual Studio Code (VS Code)

- Se trata de un editor de código fuente, especializado en editar código, depurarlo, usar control de versiones, y posee una terminal integrada. Este es ligero y rápido, además de tener una cantidad de extensiones satisfactoria dada su amplitud, y está integrado con Git. No obstante, algunas funciones avanzadas requieren que añadas extensiones, y tiene menos herramientas integradas que otros IDEs, que, por este motivo entre otros, resultan más completos. No obstante, es más ligero que IDEs como Eclipse e IntelliJ, y presenta un menor consumo de recursos.

2. Eclipse

- Permite el desarrollo de software, refactorización de código, y depuración, tiene un soporte bastante destacable para Java, y añadiendo plugins puede ser extensible. No obstante, es posible que presente problemas de velocidad al ser utilizado, y en relación a la utilidad, esta es

más sencilla si eres conocedor de las herramientas y procesos que guarda con anterioridad, de lo contrario, es bastante común tener complicaciones en su uso. También, es más pesado de Vs code, y además de en lo que referencia al soporte, en general, está más orientado en un lenguaje, como es Java, lo que lo hace menos flexible de cara a otro carácter de desarrollos.

- **3. IntelliJ IDEA**

- **Presenta un autocompletado avanzado, así como la integración de herramientas, sobre todo en lo que respecta a la refactorización. Sin embargo, al igual que el anterior, consume muchos recursos. Además, sus versiones comerciales pueden ser difíciles de obtener dado su alto coste, aunque, por otro lado, posee más funciones para Java que VS Code, teniendo en cuenta el nivel de popularidad del lenguaje, puede ser algo a tener en cuenta a la hora de elegir Ides.**

- **4. NetBeans**

- **Desarrolla aplicaciones en entornos web, de escritorio, y móviles. Este es gratuito y de código abierto, lo que lo convierte en una opción muy rentable en lo que respecta a costes. Además, es muy intuitivo, por lo que no es costoso comenzar a usarlo tampoco a nivel mental. No obstante, es menos extensible que Eclipse y posee menor personalización que Visual Studio Code.**

PUNTO 2;

Crea una aplicación sencilla (Un registro, un bucle sencillo...) utilizando el lenguaje de programación que prefieras, en el entorno instalado.

Define la aplicación realizada, lenguaje utilizado, la funcionalidad de la aplicación, el proceso de elaboración del programa. Justificando de manera completa indicando el porqué de ese lenguaje y el porqué de esa aplicación.

- Para este punto he creado una aplicación que te permite introducir el nombre de los empleados, y si están en este momento trabajando o no, introduciendo los datos en un diccionario con formato "Clave - Valor", para posteriormente introducirlo en una lista, y, a través de una función "filter", filtrar los empleados, de los recogidos, que estén trabajando en este momento, para poder mostrarlos en consola como último paso;
- El lenguaje empleado ha sido Python puesto que me resulta intuitivo y familiar, y la aplicación fue elegida porque presenta elementos interesantes y variables como el manejo de funciones, la introducción de diccionarios en listas, el filtrado a través de funciones propias de Python, así como el bucle While y un pequeño bucle for. Es variado pero sencillo a su vez.


```

empleados =[]

def añadir_empleado ():
    while True:
        npleado=input("Cómo se llama el empleado que quieres añadir? \n")
        estado_laboral=input("Se encuentra; Activo ó Inactivo ?: \n").lower()
        if estado_laboral == "activo":
            empleado_activo={
                "Nombre": npleado,
                "estado_laboral": True
            }
            empleados.append(empleado_activo)
        elif estado_laboral == "inactivo":
            empleado_inactivo={
                "Nombre": npleado,
                "estado_laboral": False
            }
            empleados.append(empleado_inactivo)
        detener=input("Deseas detener la introducción de empleados? Si/No ; \n").lower()
        if detener == "si":
            break

    añadir_empleado()

def empleados_Activos (empleado):
    return empleado["estado_laboral"]

empleados_activos= list(filter(empleados_Activos,empleados))

print("Los empleados activos son los siguientes: \n")

for empleado in empleados_activos:
    print(empleado["Nombre"],"\n")

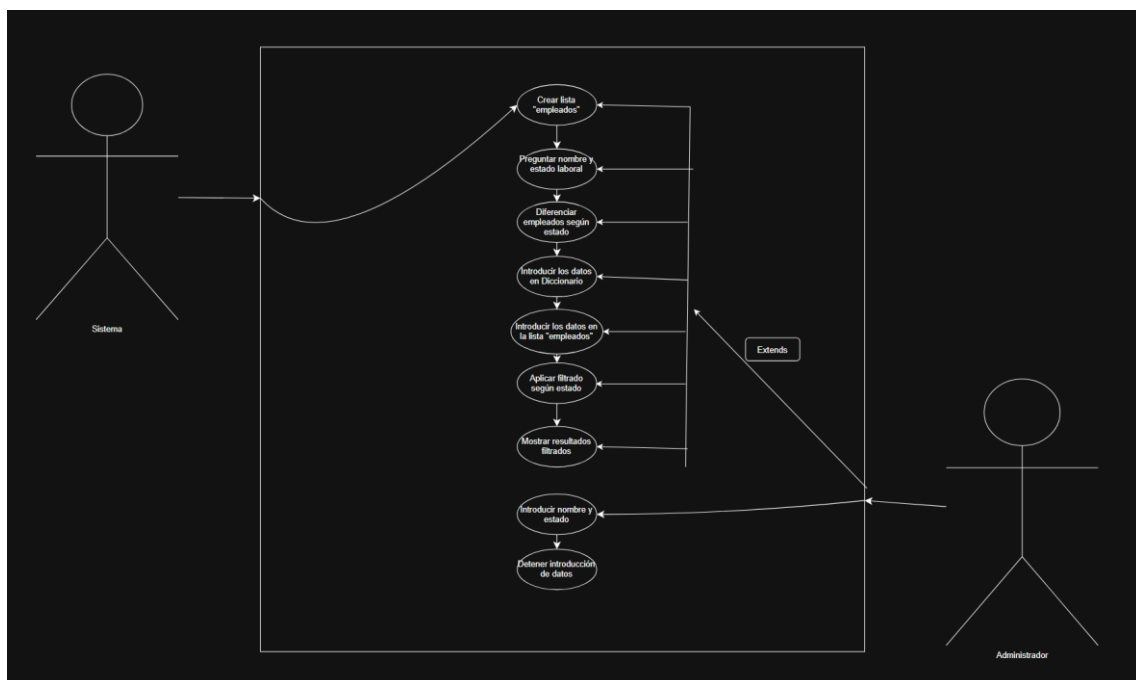
```

Diseñar un diagrama de casos de uso sobre la aplicación realizada. Explica de manera escrita lo realizado, detallando las relaciones entre los diferentes ítems.

En este diagrama de casos de uso, hay 2 actores;

1º El sistema, que realizaría las operaciones mostradas en el código a excepción de introducir datos de empleados, y dar la orden de detener el bucle.

2º Lo que sería el administrador, que introduciría los nombres y detendría la introducción de la misma.



Describe mediante la técnica de texto informal el caso de uso realizado en el punto anterior.

Nombre: Filtrar empleados activos de inactivos

Descripción: Se realizará un filtrado de empleados activos e inactivos tras haber introducido sus nombres y estados.

Actores : Administrador y Sistema

Pre-condiciones: Abrir un IDE, exportar el archivo en el IDE correspondiente, y tener espacio para operar con Python, así como acceso a una consola o terminal, pues es donde se mostraran los resultados.

Flujo de eventos:

1º Se crea la lista empleados

2º Se define función “añadir_empleado”, en la cual;

2.1 Se solicitan datos de nombre y estado del empleado que se desea introducir.

2.2 Pasan los datos por 2 condicionales que tienen en cuenta el estado, para introducirlos, o bien en un diccionario “empleado_activo”, bien en otro “empleado_inactivo”. En cualquier caso, se añaden a la lista empleados creada en el punto 1. Además de otro condicional donde se pregunta si deseas detener la introducción de empleados, en cuyo caso la función “añadir_empleado” termina.

3º Una vez termina la función, se define la función “empleados_Activos”, que devuelve el estado laboral.

4º Se crea “empleados_activos”, que alojará, a través de un filtrado via “filter”, únicamente a los empleados que cumplan como “activo” en el estado.

5º Se pinte, a modo de enunciado “Los empleados activos son los siguientes”.

6º Se utiliza un bucle for, para imprimir, uno a uno, los nombres de los empleados que aparecen en el array del paso 4.

Alternativas:

- El programa funciona adecuadamente, pero al no haber ningún trabajador activo, no muestra ninguno en la lista definitiva.

Excepciones:

- El programa no recibe empleados, pero se desea recibir un output
- El programa recibió números como empleado, estado, o ambos.

PUNTO 3;

Verifica el funcionamiento de programas, diseñando y realizando pruebas. Realiza un plan de pruebas que debe incluir la estrategia de pruebas que consideres necesaria para comprobar el buen funcionamiento del desarrollo. El plan de pruebas deberá estar “ilustrado” con al menos cinco ejemplos de resultados positivos y de resultados negativos sobre las pruebas en cuestión.

- Para comprobar el buen funcionamiento del programa, mostraremos 5 ejemplos donde funciona adecuadamente y 5 donde no.
- Dado que es un programa donde el único malfuncionamiento puede darse en la introducción de 3 valores, puesto que el resto está bien optimizado y no posee código de “relleno”, repetido, o similares, el plan de pruebas consistirá en mostrar variedades de datos introducidos válidos, ante los cuales el programa funciona, y datos donde no.

-Resultados positivos;

```
Cómo se llama el empleado que quieres añadir?  
Martín  
  
Se encuentra; Activo ó Inactivo ?:  
Activo  
  
Deseas detener la introducción de empleados? Si/No ;  
Si  
  
Los empleados activos son los siguientes:  
  
Martín  
  
>>> |
```

1º Pese a mezclar mayúsculas, minúsculas, y tildes, el programa los recibe y muestra sin problema.

```
Cómo se llama el empleado que quieres añadir?  
martin  
  
Se encuentra; Activo ó Inactivo ?:  
inactivo  
  
Deseas detener la introducción de empleados? Si/No ;  
no  
  
Cómo se llama el empleado que quieres añadir?  
lola  
  
Se encuentra; Activo ó Inactivo ?:  
activo  
  
Deseas detener la introducción de empleados? Si/No ;  
si  
  
Los empleados activos son los siguientes:  
lola
```

2º Pese a utilizar todo en minúsculas, el programa capta el mensaje

```
Cómo se llama el empleado que quieres añadir?  
fahasdfh  
  
Se encuentra; Activo ó Inactivo ?:  
inactivo  
  
Deseas detener la introducción de empleados? Si/No ;  
no  
  
Cómo se llama el empleado que quieres añadir?  
adjgn  
  
Se encuentra; Activo ó Inactivo ?:  
inactivo  
  
Deseas detener la introducción de empleados? Si/No ;  
si  
  
Los empleados activos son los siguientes:  
  
>>> |
```

3º Aún cuando no hay ninguno en activo, no te muestra un error o parecido, cumple con el funcionamiento posible del programa pues puede no haber nadie disponible.

```
Como se llama el empleado que quieres añadir?  
PABLO  
  
Se encuentra; Activo ó Inactivo ?:  
ACTIVO  
  
Deseas detener la introducción de empleados? Si/No ;  
NO  
  
Cómo se llama el empleado que quieres añadir?  
MARA  
  
Se encuentra; Activo ó Inactivo ?:  
INACTIVO  
  
Deseas detener la introducción de empleados? Si/No ;  
SI  
  
Los empleados activos son los siguientes:  
  
PABLO
```

4ºAsí como en minúsculas capta bien las respuestas, en mayúsculas también, aunque no se haya especificado en el código, funciona correctamente.

```

Cómo se llama el empleado que quieres añadir?
+++++++

Se encuentra; Activo ó Inactivo ?:
activo

Deseas detener la introducción de empleados? Si/No ;
no

Cómo se llama el empleado que quieres añadir?
-----

Se encuentra; Activo ó Inactivo ?:
inactivo

Deseas detener la introducción de empleados? Si/No ;
si

Los empleados activos son los siguientes:

+++++++

```

5º Incluso si el nombre del empleado no está disponible, quizá por un error en la base de datos, sigue clasificando correctamente a los empleados, mostrando un funcionamiento que cumple con el objetivo del algoritmo.

-Resultados Negativos;

```

Cómo se llama el empleado que quieres añadir?
hfdadf

Se encuentra; Activo ó Inactivo ?:
acctivo

Deseas detener la introducción de empleados? Si/No ;
si

Los empleados activos son los siguientes:

>>>

```


1ºAnte una ligera falta, puesto que no tiene una c extra, no asume el resultado al que te refieres, en ese sentido, no es inteligente o adaptable.

```
El proceso de ingreso de empleados se ha completado
Cómo se llama el empleado que quieres añadir?
alioeurhj

Se encuentra; Activo ó Inactivo ?:
activo

Deseas detener la introducción de empleados? Si/No ;
si

Los empleados activos son los siguientes:

alioeurhj

>>> |
```

2ºAsí como podía funcionar como un aspecto positivo si fallaba algo en la base de datos, pero permitiendo que no haya fugas de empleados, que puedas introducir nombres no válidos, puede suponer una complicación, en el caso de que intentes averiguar a qué empleado corresponde, aquel que estaría, por ejemplo de baja.

```
Cómo se llama el empleado que quieres añadir?
aiueorh

Se encuentra; Activo ó Inactivo ?:
999

Deseas detener la introducción de empleados? Si/No ;
no

Cómo se llama el empleado que quieres añadir?
123

Se encuentra; Activo ó Inactivo ?:
5222

Deseas detener la introducción de empleados? Si/No ;
si

Los empleados activos son los siguientes:
```

3º No se muestra un respaldo de errores para impedir que puedas introducir caracteres int, o float, en un programa donde introducirlos no tiene sentido.

```
El proceso de Python interactivo se ha cerrado.  
Cómo se llama el empleado que quieres añadir?  
kike  
  
Se encuentra; Activo ó Inactivo ?:  
activo  
  
Deseas detener la introducción de empleados? Si/No ;  
N0  
  
Cómo se llama el empleado que quieres añadir?  
asf  
  
Se encuentra; Activo ó Inactivo ?:  
inactivo  
  
Deseas detener la introducción de empleados? Si/No ;  
3  
  
Cómo se llama el empleado que quieres añadir?  
2  
  
Se encuentra; Activo ó Inactivo ?:  
5  
  
Deseas detener la introducción de empleados? Si/No ;  
2  
  
Cómo se llama el empleado que quieres añadir?  
1  
  
Se encuentra; Activo ó Inactivo ?:
```

4º Puedes responder con un número en la pregunta que puede romper el bucle, cosa que no tiene sentido y no da una respuesta exacta, por lo que continua interpretando que no ha de detener el bucle While.

```
El proceso de Python interactivo se ha cerrado.  
Cómo se llama el empleado que quieres añadir?  
yuls  
  
Se encuentra; Activo ó Inactivo ?:  
  
Deseas detener la introducción de empleados? Si/No ;  
  
Cómo se llama el empleado que quieres añadir?  
  
Se encuentra; Activo ó Inactivo ?:  
  
Deseas detener la introducción de empleados? Si/No ;  
  
Cómo se llama el empleado que quieres añadir?  
  
Se encuentra; Activo ó Inactivo ?:  
  
Deseas detener la introducción de empleados? Si/No ;
```

5ºAplica algo similar, aunque en este caso simplemente se te permite responder con espacios enters, sin ningún tipo de caracter

PUNTO 4;

Optimiza código empleando las herramientas disponibles en el entorno de desarrollo. Diseña una optimización para dicho desarrollo detallando los cambios que has de realizar, y la diferencia en los resultados, justificando el proceso.

- Teniendo en cuenta que el siguiente código cumple satisfactoriamente con la función que se le asignó realizar;

```
empleados = []

def añadir_empleado ():
    while True:
        empleado=input("Cómo se llama el empleado que quieres añadir? \n")
        estado_laboral=input("Se encuentra; Activo ó Inactivo ?: \n").lower()
        if estado_laboral == "activo":
            empleado_activo={
                "Nombre": empleado,
                "estado_laboral": True
            }
            empleados.append(empleado_activo)
        elif estado_laboral == "inactivo":
            empleado_inactivo={
                "Nombre": empleado,
                "estado_laboral": False
            }
            empleados.append(empleado_inactivo)
        detener=input("Deseas detener la introducción de empleados? Si/No ; \n").lower()
        if detener == "si":
            break

añadir_empleado()

def empleados_Activos (empleado):
    return empleado["estado_laboral"]

empleados_activos= list(filter(empleados_Activos,empleados))

print("Los empleados activos son los siguientes: \n")

for empleado in empleados_activos:
    print(empleado["Nombre"],"\n")
```

- No hay herramienta que pueda solventar un malfuncionamiento en el desempeño de su propósito que no presenta porque almacena correctamente las entradas dadas.

Enlace a GitHub

<https://github.com/Cortes-cmd/Entornos-de-Desarrollo.git>

Bibliografía

Bibliografía

Boada, D. (2022, diciembre 6). *¿Qué es un entorno de desarrollo y en qué se diferencia de un entorno de desarrollo integrado (IDE)?* Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-un-entorno-de-desarrollo>

ChatGPT. (s/f). Chatgpt.com. Recuperado el 25 de enero de 2025, de <https://chatgpt.com>

(S/f). Amazon.com. Recuperado el 25 de enero de 2025, de [https://aws.amazon.com/es/what-is/ide/#:~:text=Un%20entorno%20de%20desarrollo%20integrado%20\(IDE\)%20es%20una%20aplicación%20de,una%20aplicación%20fácil%20de%20usar.](https://aws.amazon.com/es/what-is/ide/#:~:text=Un%20entorno%20de%20desarrollo%20integrado%20(IDE)%20es%20una%20aplicación%20de,una%20aplicación%20fácil%20de%20usar.)