

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Organización de Lenguajes y Compiladores 1
Escuela de Vacaciones
Ing. Mario Bautista
Aux. Francisco Puac

MANUAL TÉCNICO

Proyecto

JPR



Realizado por:

3149675670901 - Jose Alejandro Barrios Rodas

Fecha: 4 de Julio de 2021

1. INTRODUCCIÓN

El sistema JPR es un editor de texto para el lenguaje de programación JPR, esta compuesto de forma que la información contenida pueda ser editada, ejecutada y exportada por el desarrollador. Para alcanzar estos propósitos se ha hecho uso de Python para crear el interprete y la herramienta Ply para realizar los analizadores léxico y sintáctico. El lenguaje utilizado para el interprete facilita la actualización y mantenimiento del código, además que maneja la información de manera eficaz. Este manual contiene los aspectos técnicos de dicho proyecto.

2. OBJETIVO

El objetivo de este manual es ayudar y guiar al desarrollador a informarse y utilizar las herramientas del sistema, así hacer un uso adecuado y optimo.

3. DIRIGIDO

Este manual esta dirigido a orientar a los desarrolladores que deseen hacer uso de este editor de texto. El sistema esta dirigido para la creación de programas basados en el lenguaje JPR, así mismo el manejo de los archivos y reportes.

4. ESPECIFICACIONES TÉCNICAS

Para el uso y la implementación del sistema JPR se requiere lo siguiente:

4.1 HARDWARE

El software requiere básicamente todo lo que una instalacion de Python requiere.

4.2 SOFTWARE

Se debe contar con Python 3 o superior y Ply 3.10 o superior.

El sistema operativo puede ser elegido libremente.

Se necesita un IDE que soporte Python, se recomienda Visual Studio Code.

Se requiere de Graphviz 2.4 o superior.

5. DESCRIPCIÓN DEL SISTEMA

El sistema consta de las herramientas necesarias para manejar archivos de extensión “.jpr”:

En estas funcionalidades para archivos se encuentran:

- Abrir: Se puede abrir cualquier archivo con la extensión especificada

- Guardar: Se puede guardar con un nombre y ubicación designada cada archivo, así como actualizar los cambios de alguno previamente guardado.
- Nuevo: Se puede generar un archivo en blanco.

El sistema cuenta con el interprete hecho en Python que al presionar el botón compilar interpreta el código fuente ingresado. El interprete de Python tomara la entrada y sera recorrida por el analizador léxico y sintáctico para detectar errores, estos errores son mostrados en la “Tabla de Errores”, luego se interpretara el código escrito para ejecutar las ordenes dadas, en este paso se detectaran todos los errores semánticos.

Al finalizar la ejecución se generaran tres reportes:

- Tabla de Símbolos: Este reporte mostrara todas las variables y funciones declaradas a lo largo del sistema.
- Tabla de Errores: Desplegara todos los errores encontrados
- Árbol Sintáctico: El árbol mostrara el flujo del analizador sintético, mostrando las derivaciones generadas.

Estos reportes pueden ser exportados, estos son generados por la herramienta de graphviz.

6. HERRAMIENTAS UTILIZADAS

6.1 Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License.

6.2 Ply

PLY es una implementación 100% Python de las herramientas lex y yacc comúnmente utilizadas para escribir analizadores y compiladores. El análisis se basa en el mismo algoritmo LALR (1) utilizado por muchas herramientas yacc. Aquí hay algunas características notables:

- Proporciona informes de errores e información de diagnóstico muy extensos para ayudar en la construcción del analizador. La implementación original se desarrolló con fines educativos. Como resultado, el sistema intenta identificar los tipos más comunes de errores cometidos por usuarios novatos.

- Proporciona soporte completo para producciones vacías, recuperación de errores, especificadores de precedencia y gramáticas moderadamente ambiguas.
- Se puede utilizar para construir analizadores para lenguajes de programación "reales". Aunque no es ultrarrápido debido a su implementación de Python, PLY se puede usar para analizar gramáticas que constan de varios cientos de reglas (como se puede encontrar para un lenguaje como C).

6.3 Graphviz

Graphviz es un programa de visualización gráfica de fuente abierta. La visualización de gráficos es una forma de representar información estructural como diagramas de gráficos y redes abstractos. Tiene importantes aplicaciones en redes, bioinformática, ingeniería de software, diseño de bases de datos y web, aprendizaje automático y en interfaces visuales para otros dominios técnicos.

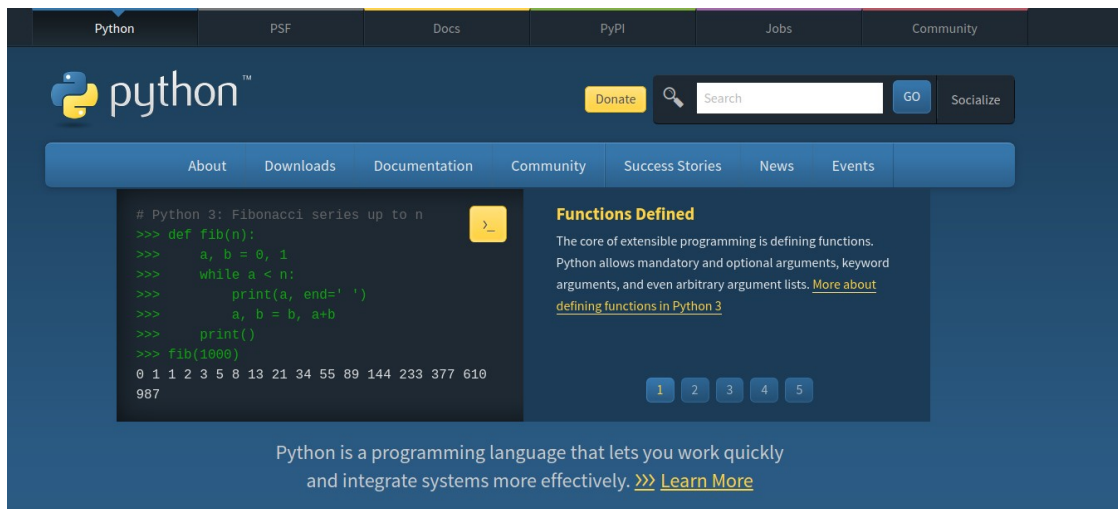
6.4 Visual Studio Code

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity).

7. INSTALACION DE HERRAMIENTAS

7.1 Python (Windows)

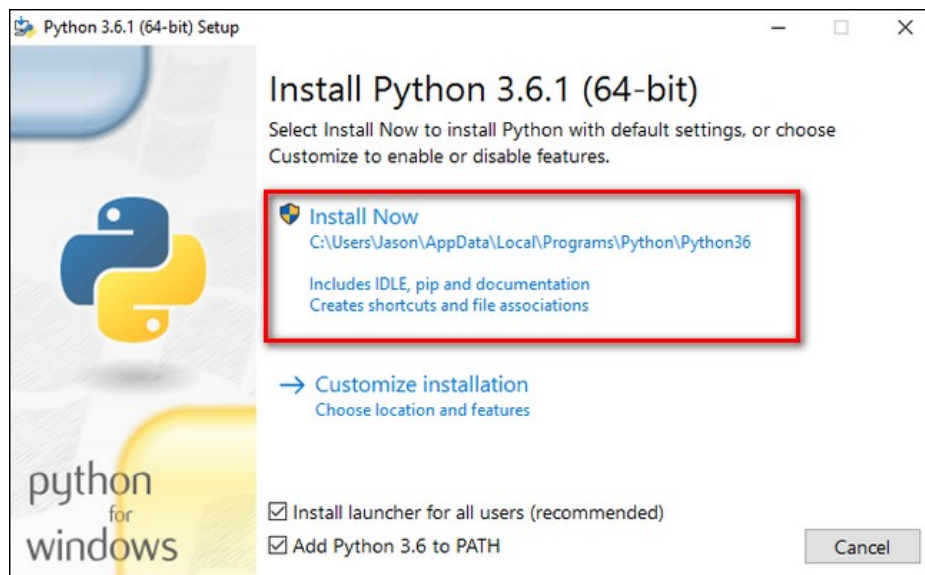
Debe ingresar a la pagina oficial de Python.



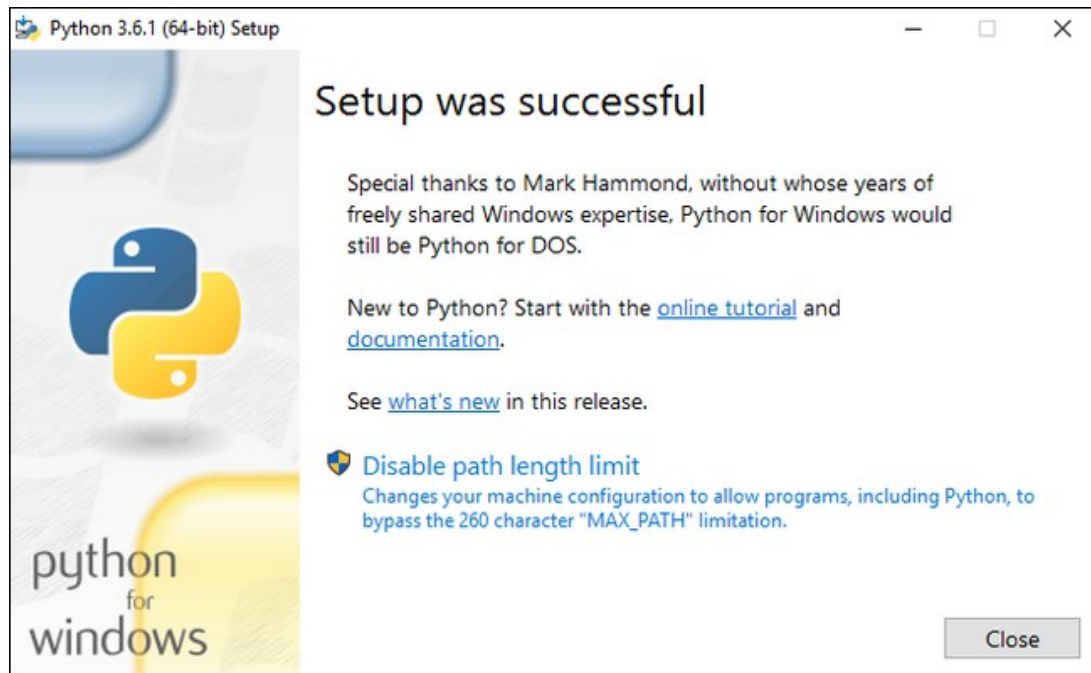
Luego presiona en Downloads y selecciona su sistema operativo, en este caso sera Windows. Luego debe seleccionar la siguiente opción.

- [Python 3.6.1 - 2017-03-21](#)
 - Download [Windows x86 web-based installer](#)
 - Download [Windows x86 executable installer](#)
 - Download [Windows x86 embeddable zip file](#)
 - Download [Windows x86-64 web-based installer](#)
 - Download [Windows x86-64 executable installer](#)
 - Download [Windows x86-64 embeddable zip file](#)
 - Download [Windows help file](#)

Después de descargar el archivo ejecutable se debe iniciar.

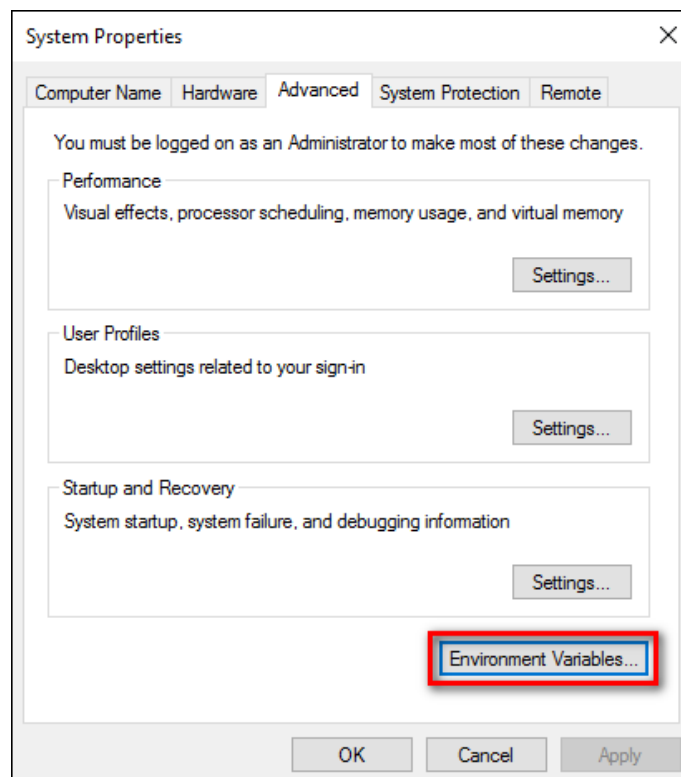


Luego de continuar con la instalación debe aparecer esta ventana.

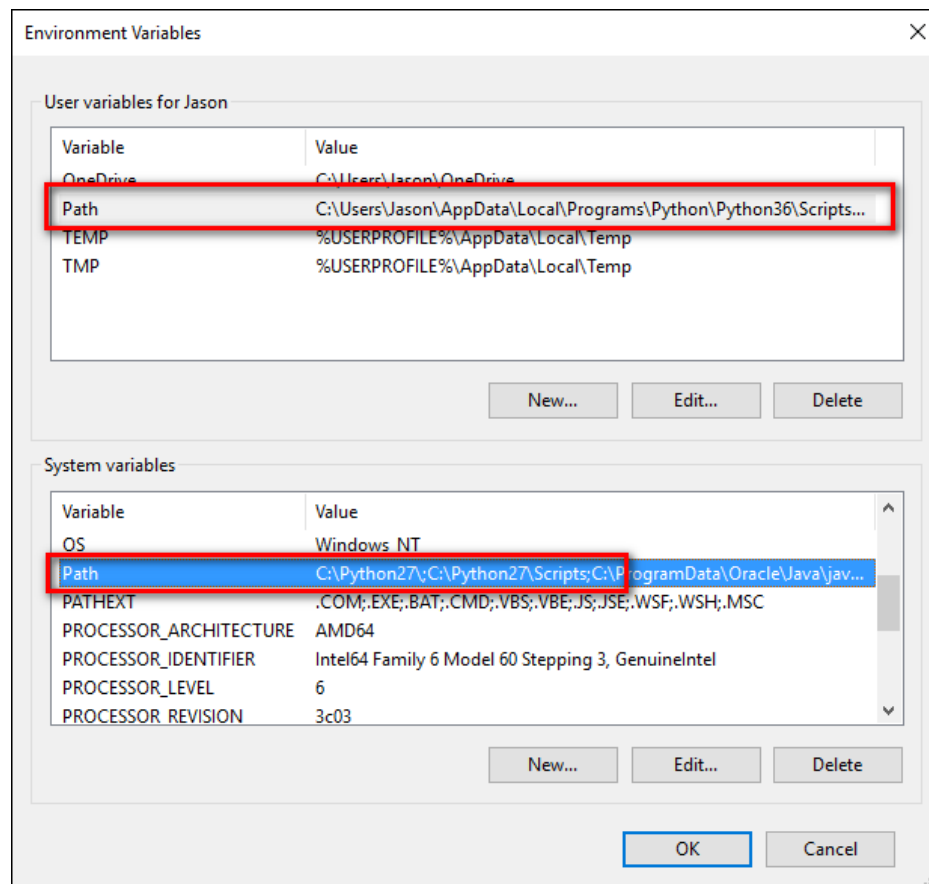


Ahora solo debemos configurar las variables de entorno en Windows, vamos a buscador e ingresamos “Propiedades avanzadas del sistema” y entramos a la primera opción.

Luego debemos presionar en Variables de Entorno.



Agregamos la variable de entorno en Variables del Sistema



7.2 Graphviz

Ingresamos a la pagina principal de graphviz y entramos a la seccion de Download
Seleccionamos la primera opción.

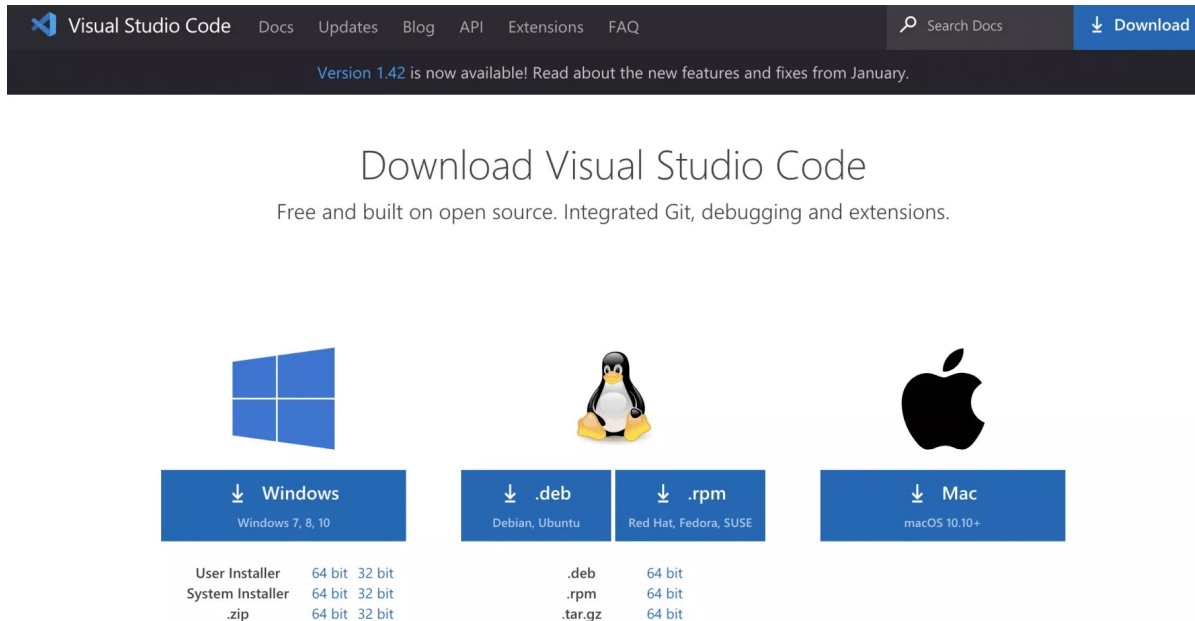
Windows

- Stable Windows install packages:
 - 2.47.3 EXE installer for Windows 10 (64-bit): [stable_windows_10_cmake_Release_x64_graphviz-install-2.47.3-win64.exe](#) (not all tools and libraries are included)
 - 2.47.3 EXE installer for Windows 10 (32-bit): [stable_windows_10_cmake_Release_Win32_graphviz-install-2.47.3-win32.exe](#) (not all tools and libraries are included)
 - 2.47.3 ZIP archive for Windows 10 (32-bit): [stable_windows_10_msbuild_Release_Win32_graphviz-2.47.3-win32.zip](#)
 - checksums: [stable_windows_10_cmake_Release_x64_graphviz-install-2.47.3-win64.exe.sha256](#) | [stable_windows_10_cmake_Release_Win32_graphviz-install-2.47.3-win32.exe.sha256](#) | [stable_windows_10_msbuild_Release_Win32_graphviz-2.47.3-win32.zip.sha256](#)
 - Further 2.47.3 variants available on [Gitlab](#)
 - Prior to 2.46 series

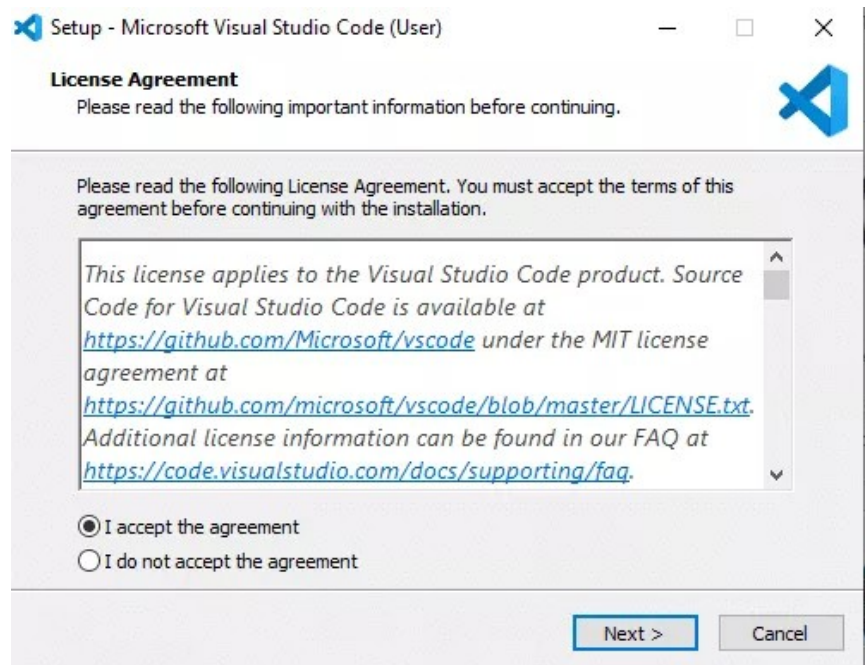
Luego ejecutamos el instalador y finalizamos la instalación.

7.3 Visual Studio Code

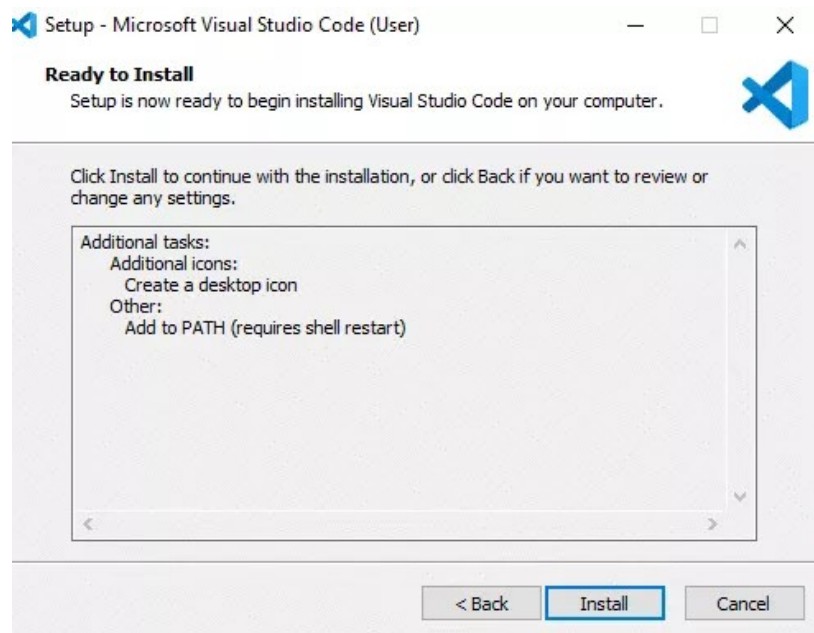
Descargamos el archivo ejecutable



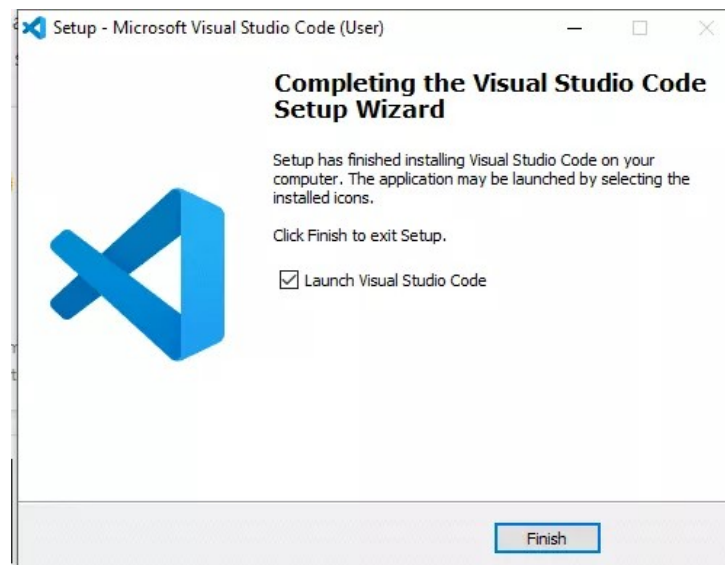
Aceptamos los términos



Continuamos y damos en instalar.



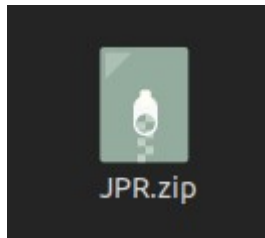
La instalacion se completara



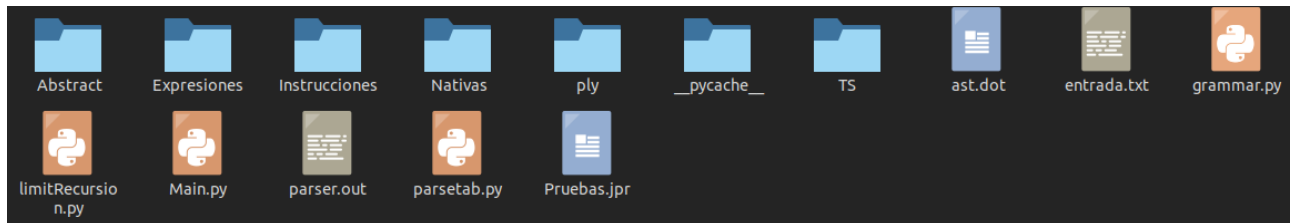
8. ABRIR PROGRAMA

Para ejecutar el archivo debemos:

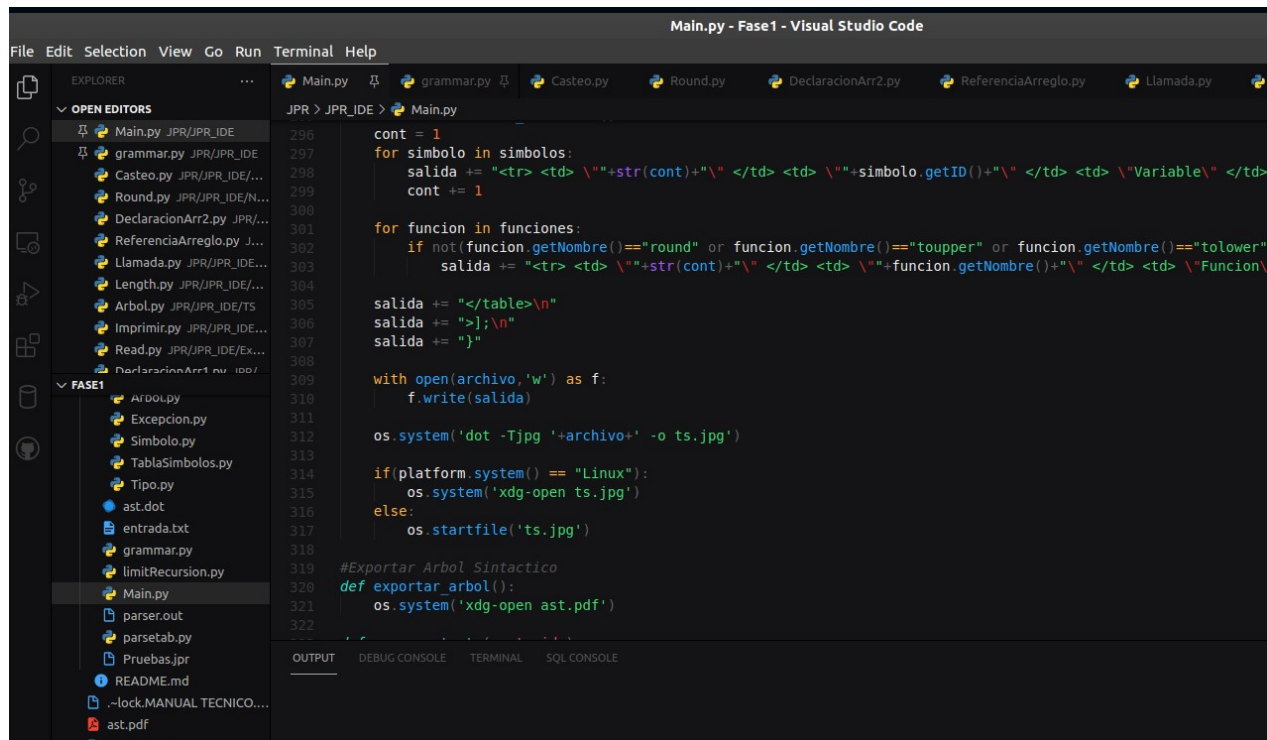
1. Descargar código fuente



2. Luego se descomprime el archivo



3. Luego procedemos a abrir la carpeta con Visual Studio Code



4. Ejecutamos la clase Main.py en la raíz
5. Debe abrir el programa

