Universidad de San Carlos de Guatemala

Facultad de ingeniería

Escuela de Ciencia y Sistemas

Inga. Damaris Campo

Aux. Mario Solís

Manual Técnico

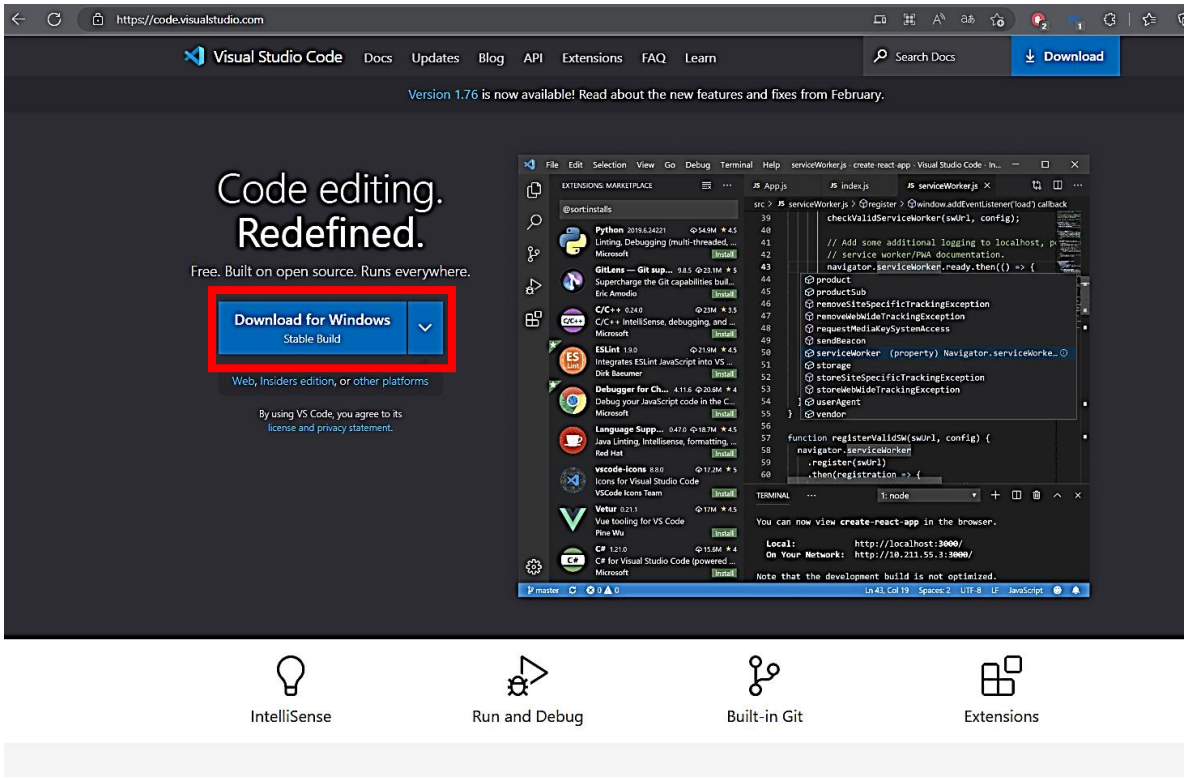
Nombre: Frander Oveldo Carreto Gómez.

Carné: 201901371

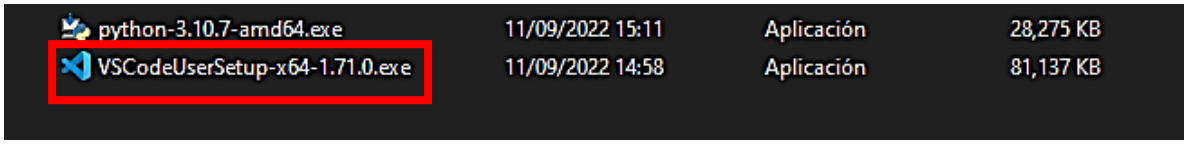
Tarea: Proyecto 1

Fecha: 22/03/23

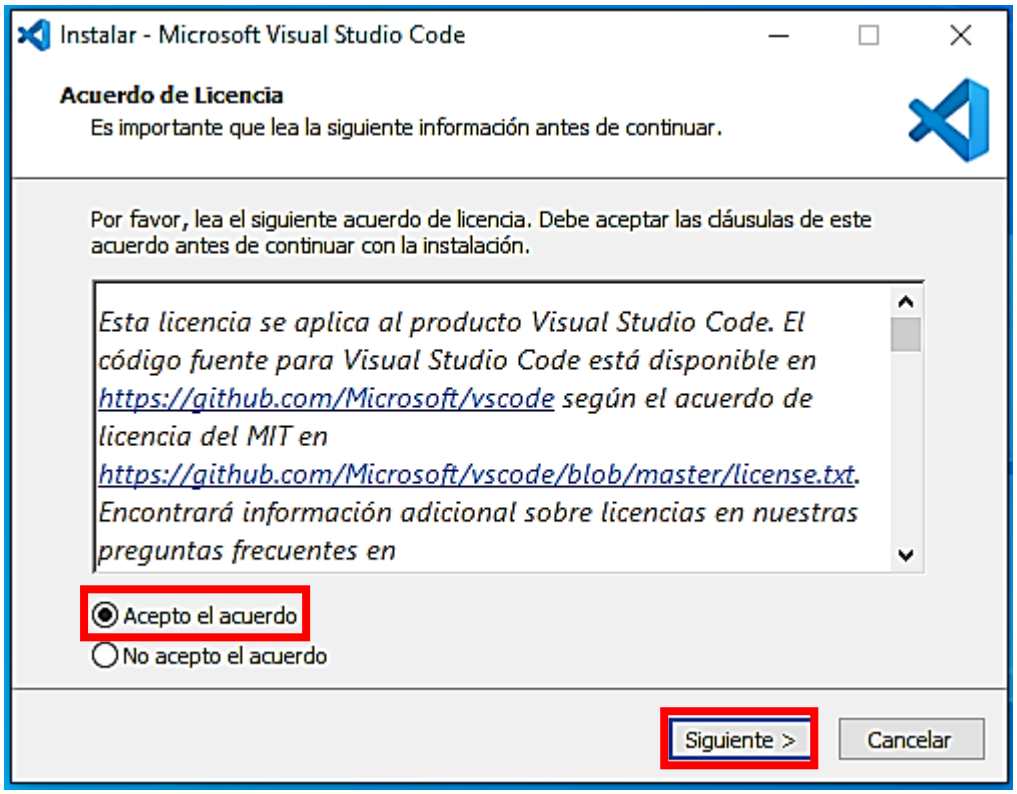
- Para realizar nuestro proyecto tenemos que tener instalado Visual Studio, Python en la instalación de visual code nos vamos al siguiente link: [Visual Studio Code - Code Editing. Redefined](#) y elegimos nuestro sistema operativo de preferencia.



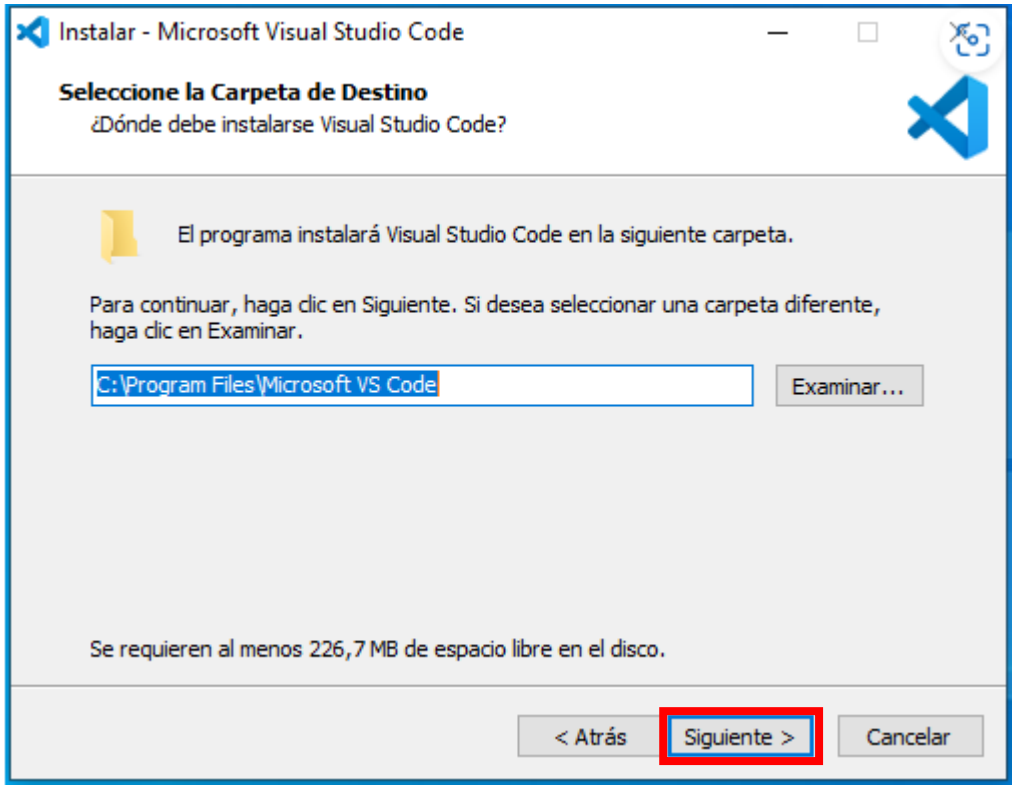
- Nos vamos a la carpeta donde guardamos nuestra descarga.



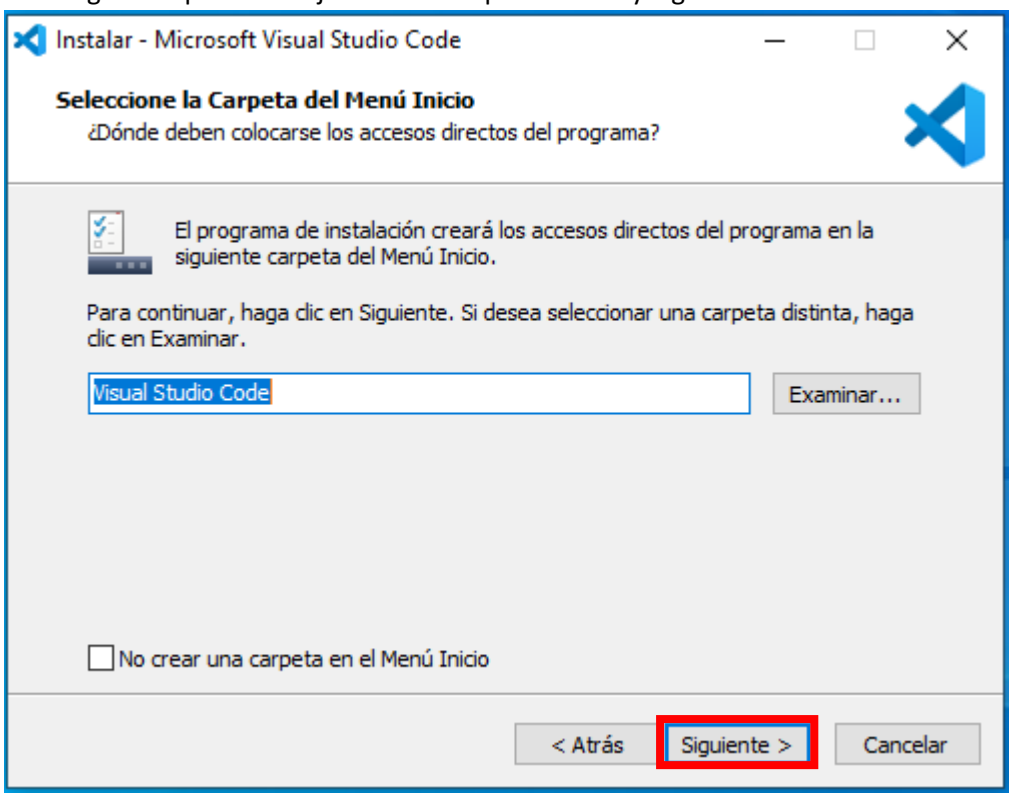
- Ejecutamos el archivo y le damos aceptar los términos y le damos en siguiente:



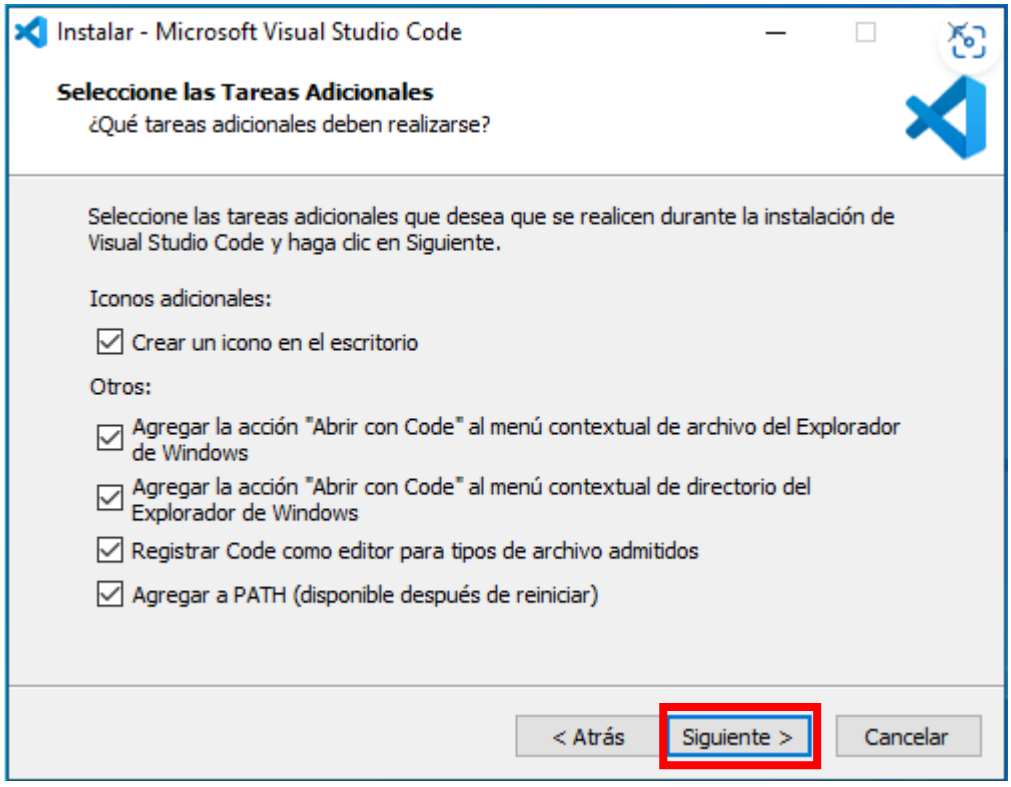
- Indicamos la ruta donde queremos instalarlo. Por defecto lo instala en Program Files. Después clickamos sobre Siguiente.



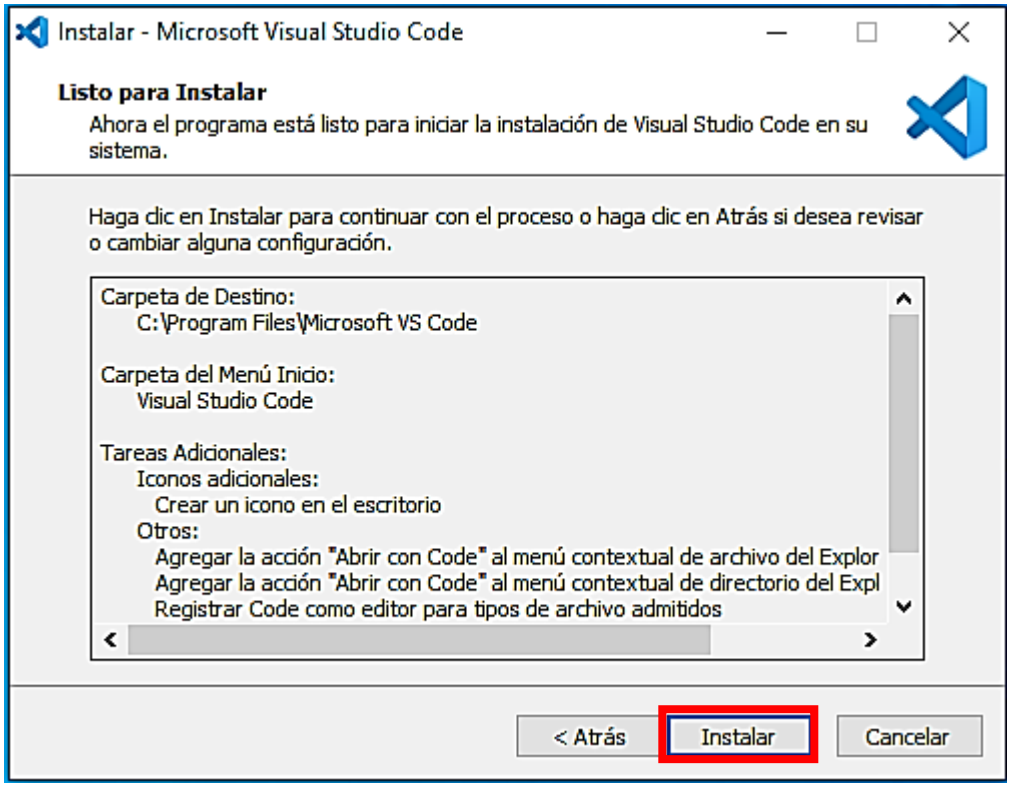
- En la siguiente pantalla dejaremos todo por defecto y Siguiente.



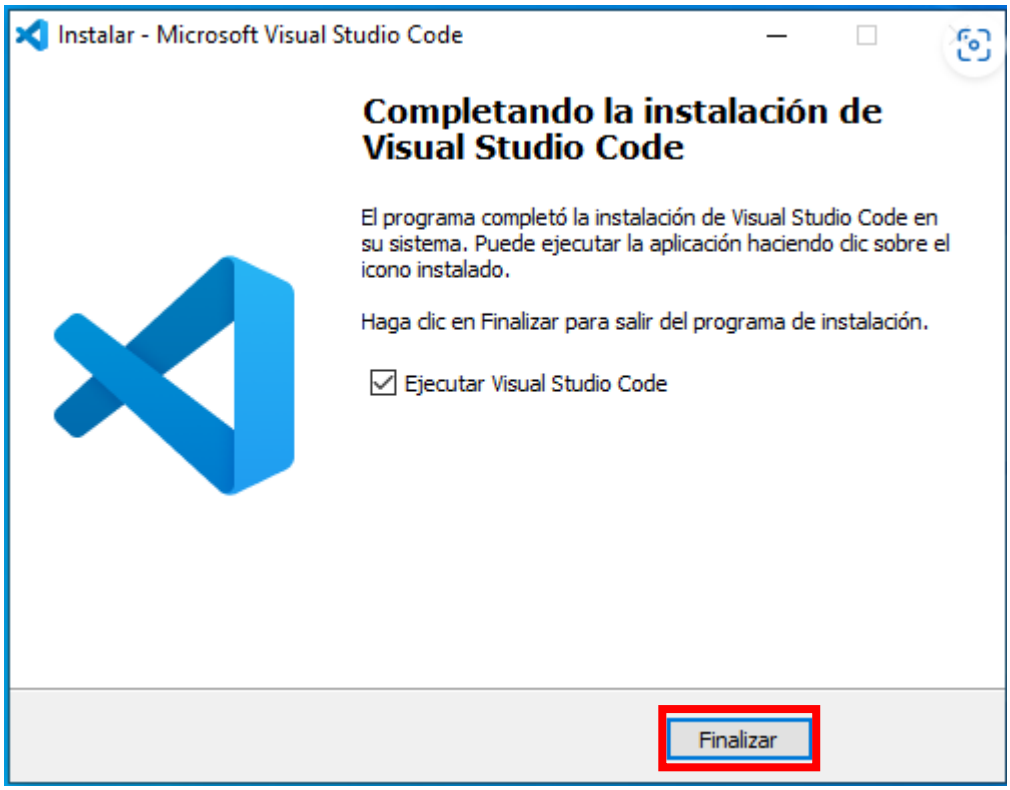
- En la siguiente pantalla marcaremos las opciones que creamos convenientes para nuestro entorno de trabajo y clickaremos sobre Siguiente.



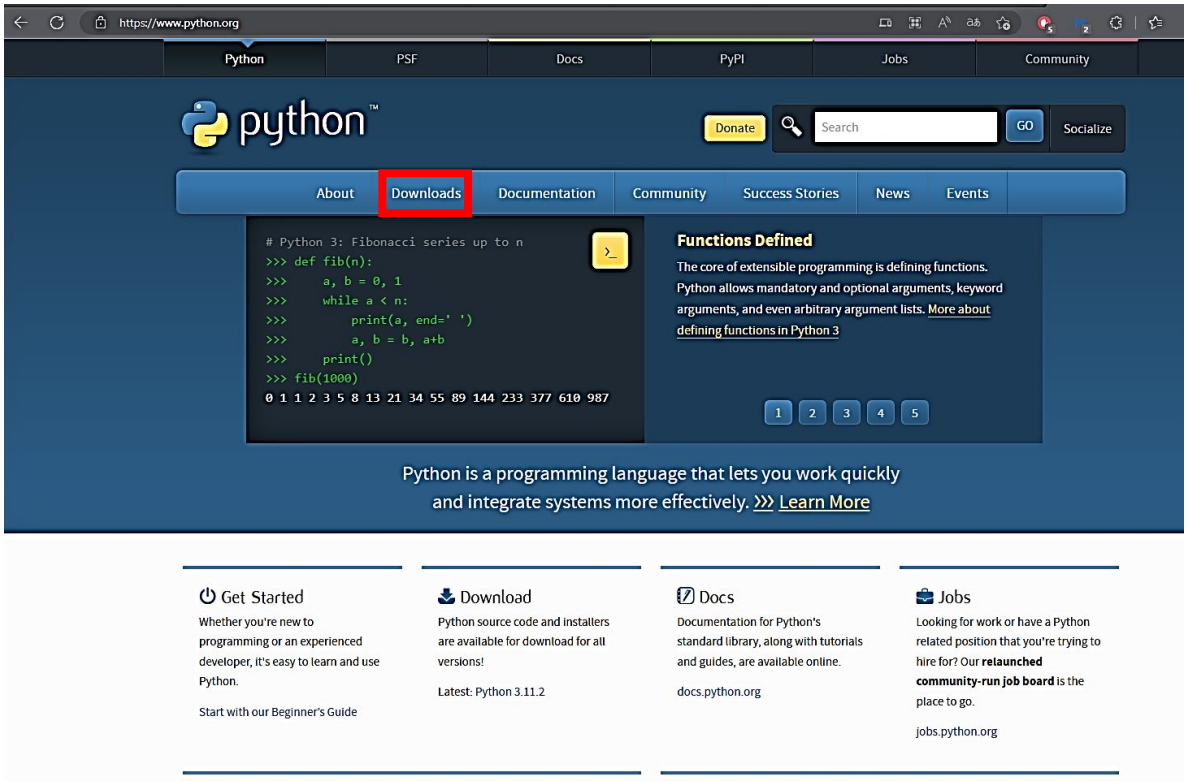
- Y ya pasaremos a la instalación del software clickando sobre Instalar.



- Una vez finalizada la instalación clickaremos sobre el botón Finalizar.

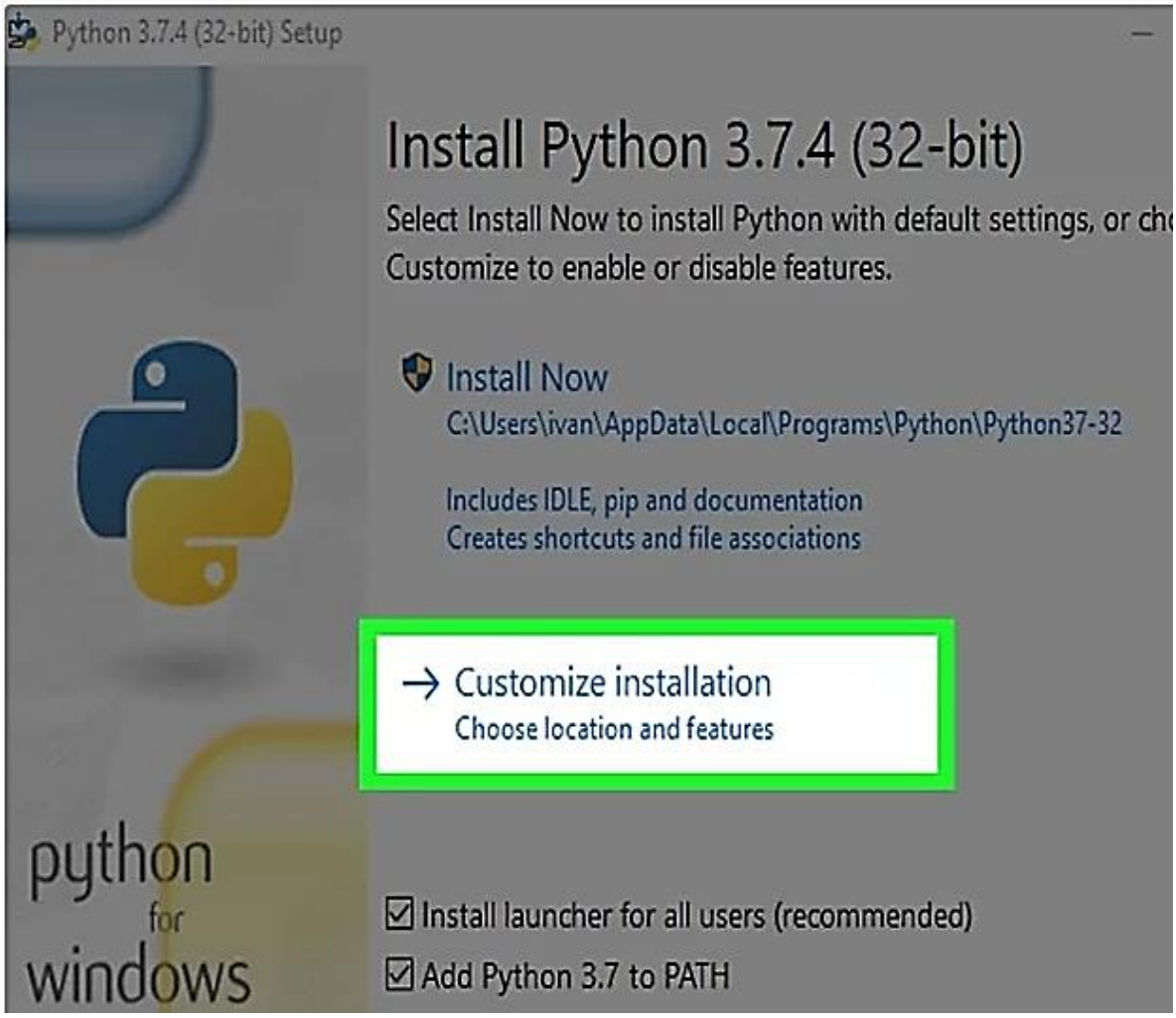


- Para instalar Python nos vamos al siguiente link: [Welcome to Python.org](https://www.python.org)

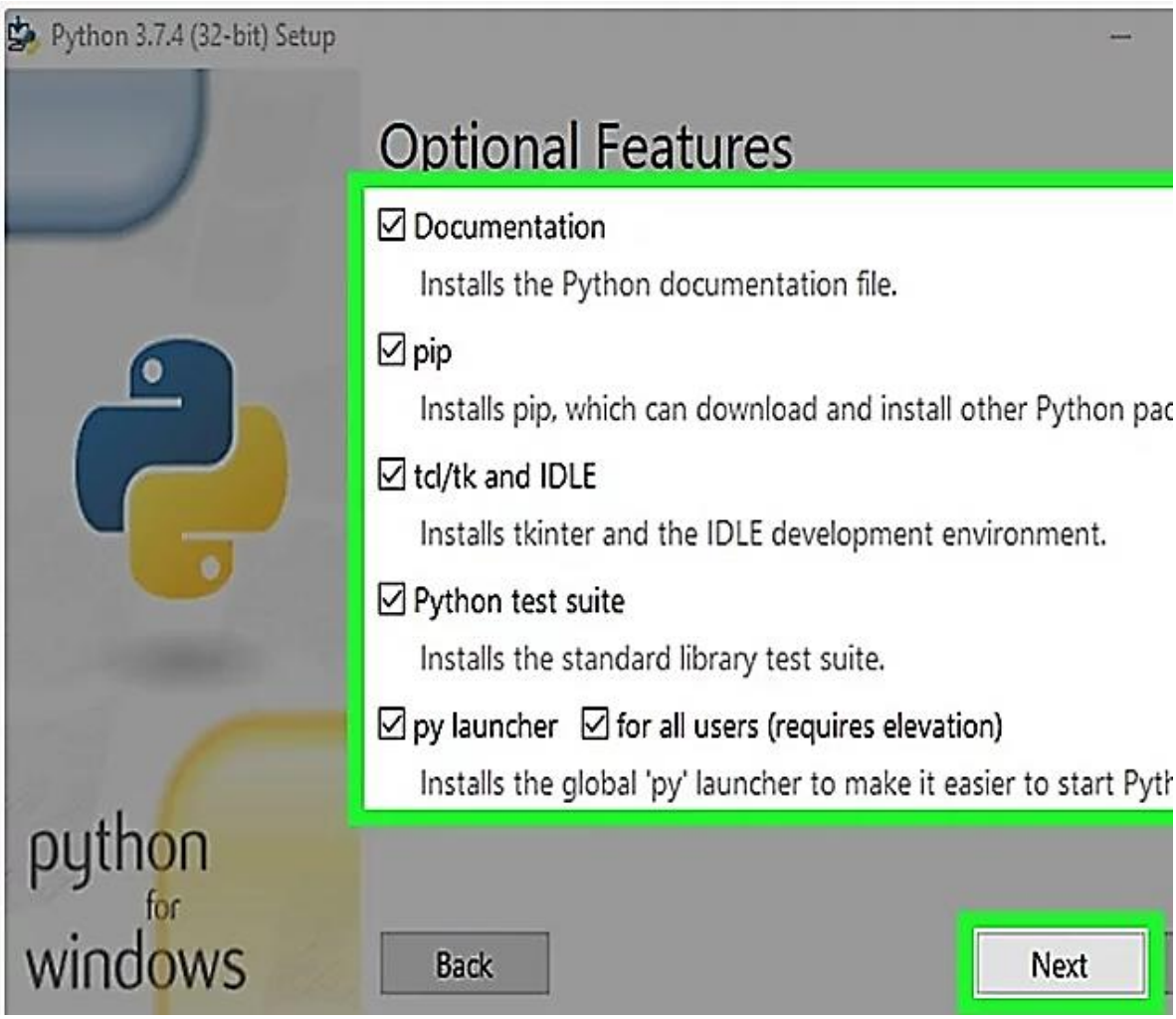


- Descargamos la última versión acorde a nuestro sistema operativo.

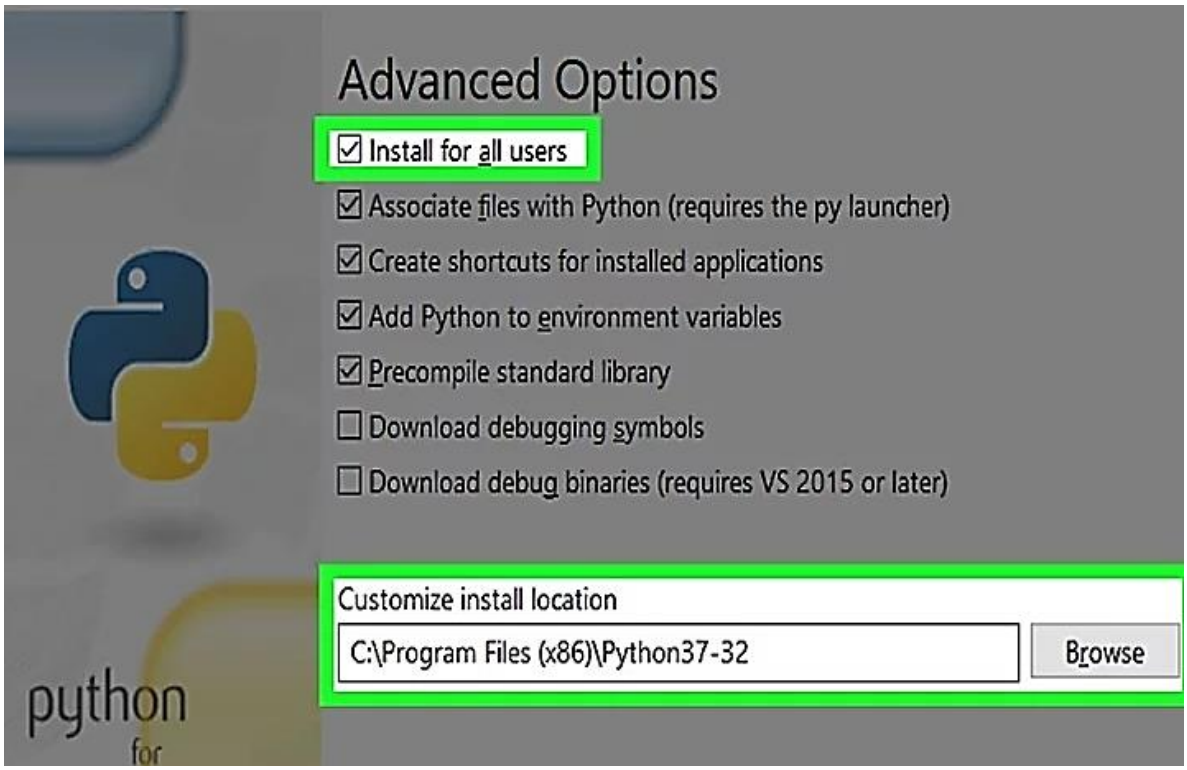
- Ejecutamos nuestra aplicación y le damos en personalizar instalación.



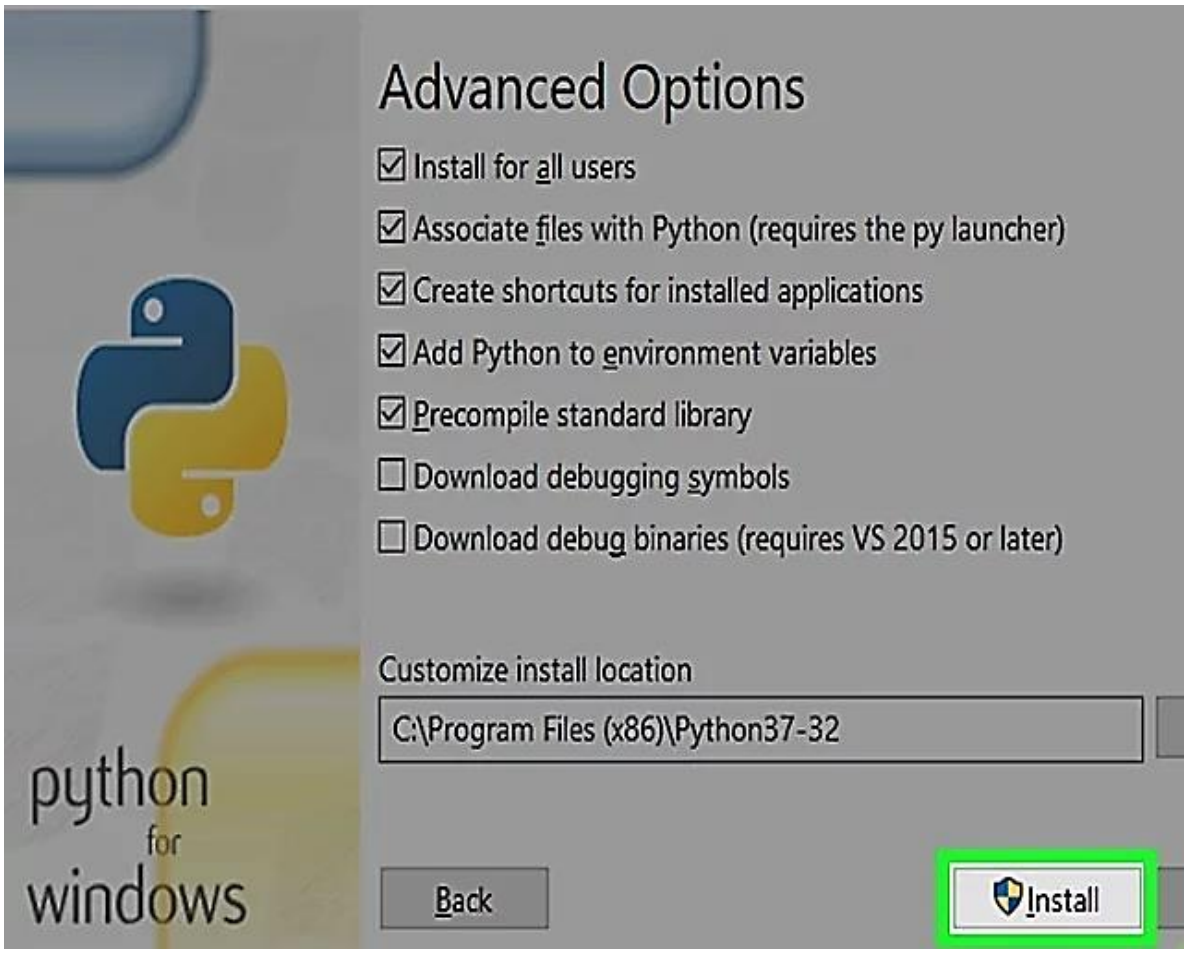
- Asignamos la instalación que deseemos personalizar y le damos en siguiente.



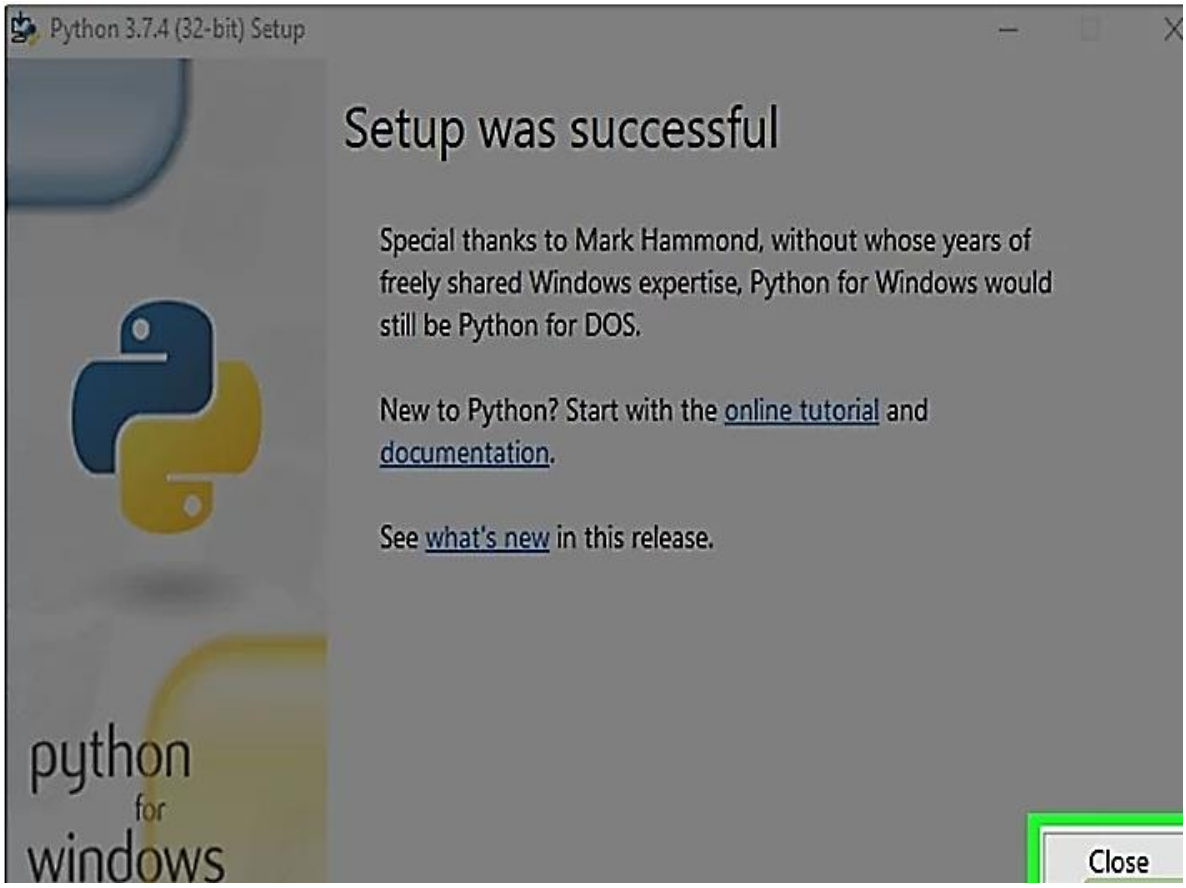
- Dejamos por defecto a la siguiente ventana y le damos en siguiente con la ruta que deseemos.



- Le damos en instalar.



- Y le damos en cerrar.



- Para poder iniciar con nuestro programa importamos la librería que vayamos a utilizar para nuestra interfaz gráfica o las que sean necesarias para que nuestro programa se ejecute correctamente.

```
1  import tkinter as tk
2  import webbrowser
3  from tkinter import filedialog
4  import json
5
```

- Declaramos nuestras clases para nuestro programa.

```
6
7  class Application(tk.Frame):
8      def __init__(self, master=None):
9          super().__init__(master)
10         self.master = master
11         self.master.title("Lenguajes formales A")
12         self.pack()
13         self.create_widgets()
```

- Creamos nuestro parámetro para la barra del menú.

```
15
16      def create_widgets(self):
17          # Crear la barra de menú
18          menubar = tk.Menu(self.master)
19          self.master.config(menu=menubar)
```

- Creamos el parámetro de los labels para el menú archivo.

```
20
21          # Crear el menú "Archivo"
22          file_menu = tk.Menu(menubar, tearoff=0)
23          file_menu.add_command(label="Abrir", command=self.abrir_archivo)
24          file_menu.add_command(label="Guardar", command=self.guardar_archivo)
25          file_menu.add_command(label = "Guardar Como", command=self.guardar_como)
26          file_menu.add_command(label="Analizar", command=self.analizar_texto)
27          file_menu.add_command(label="Errores", command=self.mostrar_errores)
28          file_menu.add_separator()
29          file_menu.add_command(label="Salir", command=self.master.quit)
30          menubar.add_cascade(label="Archivo", menu=file_menu)
```

- Creamos el parámetro de los labels para el menú herramientas.

```
32          # Crear el menú "Herramientas"
33          tools_menu = tk.Menu(menubar, tearoff=0)
34          tools_menu.add_command(label="Manual de Usuario", command=self.manual_usuario)
35          tools_menu.add_command(label="Manuan Técnico", command=self.manual_tecnico)
36          tools_menu.add_command(label="Temas de Ayuda", command=self.temas_ayuda)
37          menubar.add_cascade(label="Herramientas", menu=tools_menu)
```

- Agregamos un cuadro para que se muestre la información dentro de un cuadro.

```
        # Agregar un cuadro de texto para editar el archivo
        self.textbox = tk.Text(self)
        self.textbox.pack(fill="both", expand=True)
```

- Le asignamos la función para el botón de abrir los archivos.

```
def abrir_archivo(self):
    file_path = filedialog.askopenfilename()
    if file_path:
        with open(file_path, "r") as f:
            self.textbox.delete("1.0", tk.END)
            self.textbox.insert(tk.END, f.read())
```

- Asignamos la función para el botón Guardar.

```
    def guardar_archivo(self):
        file_path = filedialog.asksaveasfilename()
        if file_path:
            with open(file_path, "w") as f:
                f.write(self.textbox.get("1.0", tk.END))
```

- Asignamos la función para el botón Guardar Como.

```
def guardar_como(self):
    filetypes = (("Text files", "*.txt"), ("All files", "*.*"))
    file_path = filedialog.asksaveasfilename(filetypes=filetypes)
    if file_path:
        with open(file_path, "w") as f:
            f.write(self.textbox.get("1.0", tk.END))
```

- Asignamos la función para nuestro botón Analizar.

```
def analizar_texto(self):
    # Obtener el texto del cuadro de texto
    texto = self.textbox.get("1.0", tk.END)

    # Intentar cargar el texto como un objeto JSON
    try:
        datos = json.loads(texto)
    except json.JSONDecodeError as e:
        # Mostrar un mensaje de error si el texto no es un JSON válido
        tk.messagebox.showerror("Error", "El texto no es un JSON válido: {}".format(str(e)))
        return

    # Mostrar los elementos reconocidos en una nueva ventana
    lista_elementos = list(datos.keys())
    elementos_window = tk.Toplevel(self.master)
    elementos_window.title("Elementos reconocidos")
    elementos_label = tk.Label(elementos_window, text="\n".join(lista_elementos))
    elementos_label.pack()
    pass
```

- Asignamos el valor para el botón Errores.

```
def mostrar_errores(self):
    # Obtener el texto del cuadro de texto
    texto = self.textbox.get("1.0", tk.END)

    # Intentar cargar el texto como un objeto JSON
    try:
        datos = json.loads(texto)
    except json.JSONDecodeError as e:
        # Mostrar un mensaje de error si el texto no es un JSON válido
        tk.messagebox.showerror("Error", "El texto no es un JSON válido: {}".format(str(e)))
        return

    # Analizar el texto para buscar errores
    errores = []
    for elemento, valor in datos.items():
        if not isinstance(valor, str):
            errores.append("El valor para el elemento {} no es una cadena: {}".format(elemento, valor))

    # Mostrar los errores en una nueva ventana
    if errores:
        errores_window = tk.Toplevel(self.master)
        errores_window.title("Errores encontrados")
        errores_label = tk.Label(errores_window, text="\n".join(errores))
        errores_label.pack()
    else:
        tk.messagebox.showinfo("Errores", "No se encontraron errores.")
    pass

def manual_usuario(self):
    #cambiar la direccion del archivo para que se abra la documentacion de este proyecto
    pathTecnico = "Proyecto1\Documentacion\Manual Tecnico.pdf"
    webbrowser.open_new(pathTecnico)
    pass
```

- Asignaos los valores para el botón manuales.

```
def manual_usuario(self):
    #cambiar la direccion del archivo para que se abra la documentacion de este proyecto
    pathTecnico = "Proyecto1\Documentacion\Manual Tecnico.pdf"
    webbrowser.open_new(pathTecnico)
    pass
```

- No olvidemos cerrar el programa con el siguiente código para que se ejecute la interfaz gráfica.

```
root = tk.Tk()
app = Application(master=root)
app.mainloop()
```