



**UNIVERSIDAD NACIONAL DE COLOMBIA  
SEDE MANIZALES**

**ANÁLISIS Y DISEÑO DE ALGORITMOS**

**MANUAL DE USUARIO PROYECTO FINAL**

**Presentado a**

Luz Enith Guerrero Mendieta

**Presentado por**

Manuela Zuluaga Cardona

Santiago Hoyos Gomez

**Fecha de entrega**

Diciembre 2023

# MANUAL DE USUARIO PROYECTO FINAL

## Introducción

Bienvenido al manual de usuario del proyecto final para la asignatura de Análisis y diseño de algoritmos. Este documento proporciona información detallada sobre cómo utilizar y comprender el funcionamiento del proyecto desarrollado en Python.

### Requisitos del equipo:

Para garantizar un rendimiento óptimo y la funcionalidad adecuada de nuestro software, es esencial cumplir con ciertos requisitos mínimos del sistema. Estos requisitos están diseñados para proporcionar un entorno de ejecución estable y eficiente. A continuación, se detallan los requisitos del equipo necesarios para aprovechar al máximo la aplicación:

- **Sistema Operativo:**  
Se recomienda utilizar sistemas operativos basados en Unix, como Linux o macOS, o en su defecto, Windows 10 o versiones superiores.
- **Procesador:**  
Se aconseja un procesador de doble núcleo o superior, con una frecuencia de reloj de al menos 2.0 GHz.
- **Memoria RAM (RAM):**  
Se recomienda un mínimo de 8 GB de RAM para garantizar un rendimiento fluido y eficiente del software.
- **Almacenamiento:**  
Asegúrese de contar con al menos 20 GB de espacio libre en disco para la instalación del software y el almacenamiento de datos adicionales.
- **Resolución de Pantalla:**  
Se recomienda una resolución de pantalla de 1280x720 píxeles o superior para una visualización clara de la interfaz de usuario.
- **Conexión a Internet:**  
Para acceder a actualizaciones y recursos en línea, se recomienda una conexión a Internet estable.
- **Software Adicional:**  
Asegúrese de tener instaladas las bibliotecas y dependencias especificadas en la documentación del software. Esto puede incluir bibliotecas de Python, herramientas de visualización, o cualquier otra dependencia mencionada.

- **Permisos de Usuario:**

Para garantizar el acceso adecuado a recursos del sistema y la ejecución sin restricciones, se recomienda contar con permisos de usuario que permitan la instalación y ejecución de software.

Estos requisitos del equipo proporcionan un marco sólido para garantizar que el software funcione de manera eficiente y cumpla con las expectativas de rendimiento. Al cumplir con estos requisitos, los usuarios pueden disfrutar de una experiencia fluida y aprovechar al máximo las capacidades del software.

① Especificaciones del dispositivo			
Nombre del dispositivo		LAPTOP-4K33KJ2R	
Procesador		Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz	
RAM instalada		8,00 GB	
Id. del dispositivo		431E4383-2BBC-4153-989C-5EEB207F7AB8	
Id. del producto		00327-30750-99546-AAOEM	
Tipo de sistema		Sistema operativo de 64 bits, procesador x64	
Lápiz y entrada táctil		La entrada táctil o manuscrita no está disponible para esta pantalla	

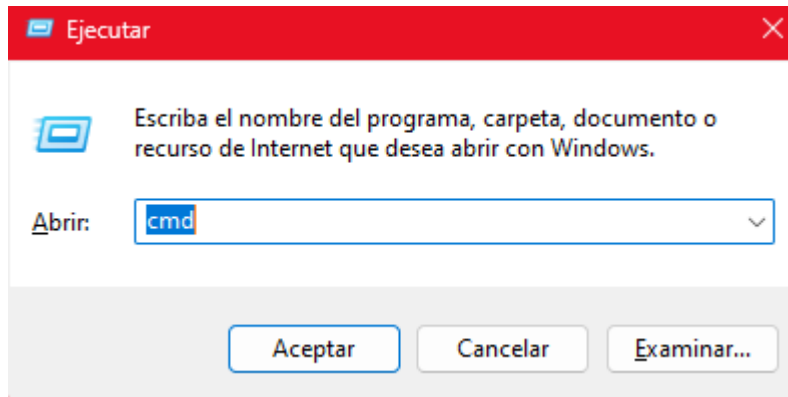
En uso (comprimido)	Disponible	Velocidad:	2400 MHz
<b>6,6 GB (376 MB)</b>	<b>1,2 GB</b>	Ranuras usadas:	1 de 2
Confirmada	En caché	Factor de forma:	SODIMM
<b>10,0/11,5 GB</b>	<b>1,2 GB</b>	Reservada para hardware:	88,9 MB
Bloque paginado	Bloque no paginado		
<b>490 MB</b>	<b>703 MB</b>		

**Asegurate de tener instalado lo siguiente:**

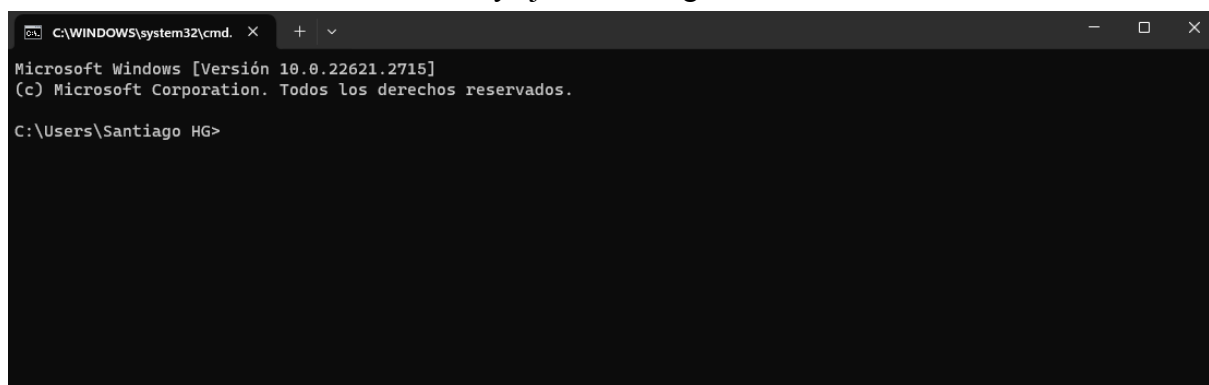
- Python: El código está escrito en python, por lo que necesitas tener python instalado en tu máquina. Puedes descargar la última versión de python desde [Descarga Python](#).
- Visual Studio Code (o el editor de código de tu preferencia): Es el entorno de desarrollo recomendado para este proyecto.

**Además, es necesario instalar las siguientes bibliotecas:**

Para realizar el proceso de instalación de las librerías, es necesario ejecutar primero el cmd. Para ello usamos la combinación de teclas **Tecla Windows + R** este comando te abre la siguiente ventana y en esta escribes el comando **cmd** que te permitirá abrir la consola cmd.



Abre tu terminal o línea de comandos y ejecuta los siguientes comandos:



```
pip install tabulate
```

Esto instalará la librería “tabulate”. Para las librerías estándar de Python como “math” y “random”, no necesitas instalarlas por separado, ya que están incluidas en la instalación estándar de Python.

Para instalar las librerías “numpy”, “biparts” y “scipy”, también debes ejecutar los siguientes comandos en tu terminal:

```
pip install biparts
```

```
pip install scipy
```

```
pip install numpy
```

Después de ejecutar estos comandos deberías poder importar estas librerías en tu script de python sin problemas. Si estás utilizando un entorno virtual, asegúrate de que el entorno esté activado antes de ejecutar el comando **pip**

*Se aclara que para el funcionamiento del proyecto debe estar instalada la librería scipy*

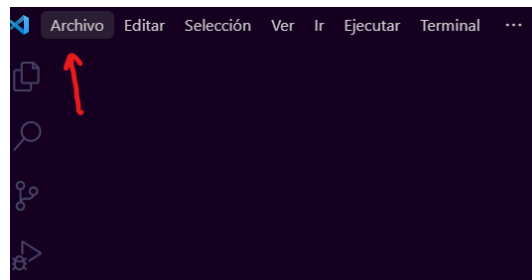
## Instalación del proyecto

### 1. Descarga el proyecto:

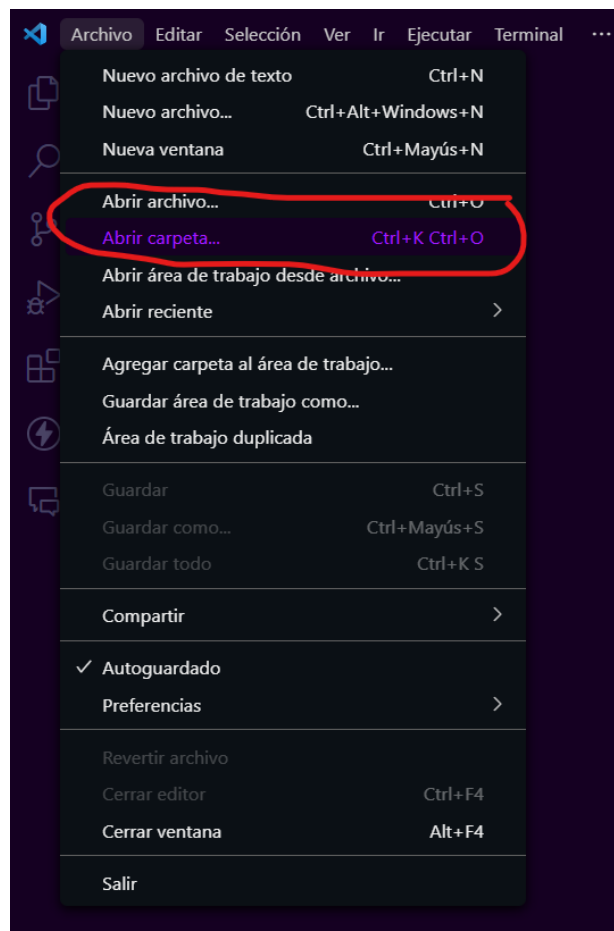
Descarga el proyecto y descomprime el archivo en una ubicación de tu elección.

### 2. Abrir en Visual Studio Code:

Abre VSC y selecciona “archivo” o en caso de tenerlo en ingles “file”, como se muestra en la siguiente imagen:



Siguiente paso: “Abrir carpeta” u “Open folder”, navega hasta la carpeta del proyecto y ábrela.



## Ejecución del programa, funcionalidades y pantallazos

Una vez instaladas las librerías podrás ejecutar el programa, para esto tenemos dos opciones:

### Opción 1 (se ejecuta el archivo *final.py*)

Primero se ingresan los canales para el actual (letra, letra) y luego los valores (0-1)

```
**Menu de particiones**
{'Actual': {}, 'Futuro': []}
Los canales disponibles para realizar las particiones son: ['a', 'b', 'c']

Ingresa los canales de la particion Actual Ejemplo: a,b,c o b,c o a,c
a,c
Ingresa el valor actual para el canal a
1
Ingresa el valor actual para el canal c
0

Ingresa los canales de la particion Futuro Ejemplo: a,b,c o b,c o a,c
a,b,c

Tiempos iniciales: {'Actual': {'a': '1', 'c': '0'}, 'Futuro': ['a', 'b', 'c']}
Desea realizar comprobaciones de distancia entre particiones(s/n)?
n
```

Si seleccionamos algo diferente de “s”. Solo se mostrará la distribución de probabilidades para, el caso indicado

Estado//T+1	000	100	010	110	001	101	011	111
{'a': '1', 'c': '0'}	0.25	0.25	0	0	0.25	0.25	0	0

En caso de seleccionar “s”. Primero se mostrará la distribución original

```
**Se mirara el original**
```

Estado//T+1	000	100	010	110	001	101	011	111
{'a': '1', 'c': '0'}	0.25	0.25	0	0	0.25	0.25	0	0

Seguidamente se mostrarán de forma iterativa diferentes cálculos de las diferentes posibles particiones. Se debe buscar aquellos lugares donde aparece, de este modo:

Estado//T+1	000	100	010	110	001	101	011	111
{'a': '1', 'c': '0'}	0.25	0.25	0	0	0.25	0.25	0	0

Para el actual ['a', 'c']  
Para el futuro []

Estado//T+1	
{'a': '1', 'c': '0'}	1

Comparacion de distancia con la distrinucion compuesta: ['a', 'c']['a', 'b', 'c'] // ['a', 'c'][]  
valores previos: [0.25, 0.25, 0.0, 0.0, 0.25, 0.25, 0.0, 0.0]  
valores previos: [0.25, 0.25, 0.0, 0.0, 0.25, 0.25, 0.0, 0.0]  
Distancia EMD: 0.0

Primero encontraremos la tabla de distribución de probabilidades de las 2 particiones. Aunque lo importante es los valores que se encuentran abajo, donde se puede encontrar el cálculo de la distancia.

## Opcion 2 (se ejecuta el archivo *base.py* )

A diferencia de la opción 1, en esta se presenta un menú que te permite acceder a otras funcionalidades, sin embargo para acceder a la misma funcionalidad que se presenta en la opción 1 directamente debes:

```

Menu Principal
1) Generar estados aleatorios
2)Generar estados aleatorios para EMD
3) Usar datos de prueba
4) Usar datos de prueba para EMD
5) Salir
Ingresa una opcion: 4

```

Idealmente se debe seleccionar “4” para usar valores de una muestra ya existente para evitar el uso de valores aleatorios.