



El futuro digital
es de todos

MinTIC



Bases para trabajar con Python

OPERADO POR:

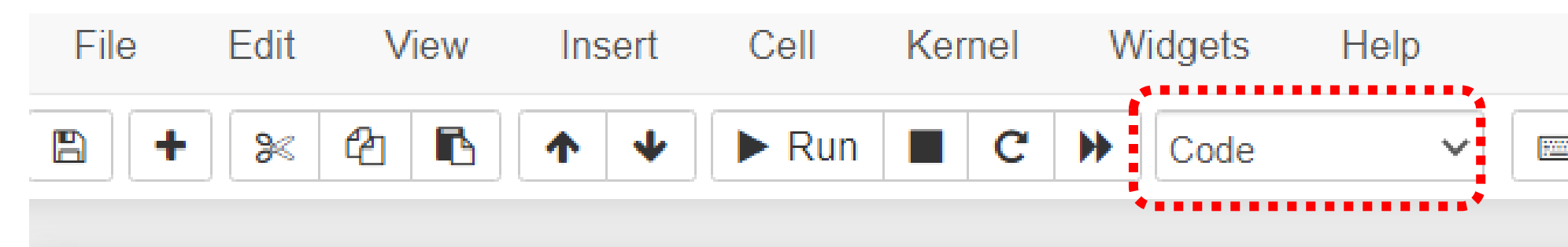


Mision
TIC 2022

ruta de aprendizaje 1



En esta nueva celda vamos a revisar si la opción **“Code”** esta activada. Por defecto al insertar una celda nueva, esta aparece como celda para programar (code). Si no lo esta, debemos seleccionar esa opción.



Primer documento en Jupiter

Notas de clase

Primera línea de Código:

In []: ▶

Una vez la celda sea tipo **“Code”** ¡vamos a escribir nuestra primera línea de código!

En la celda vamos a escribir la línea de código mas conocida por los programadores:
print(“ Hola mundo ”)



Nuestro documento se debe ver así:

Primer documento en Jupiter

Notas de clase

Primera línea de Código:

```
In [ ]: ▶ print("Hola mundo")
```

Con **Jupiter**, podemos ejecutar o correr una línea de código, como en cualquier consola. Esto facilita el proceso para la documentación mas adelante.

Cuando programamos, el cuaderno utiliza un código de colores especial para leer lo que estamos haciendo de forma mas fácil.

Para este ejemplo, el comando `print()` le dice a Python que debe imprimir un mensaje, en este caso el mensaje debe ir entre comillas `" "`.



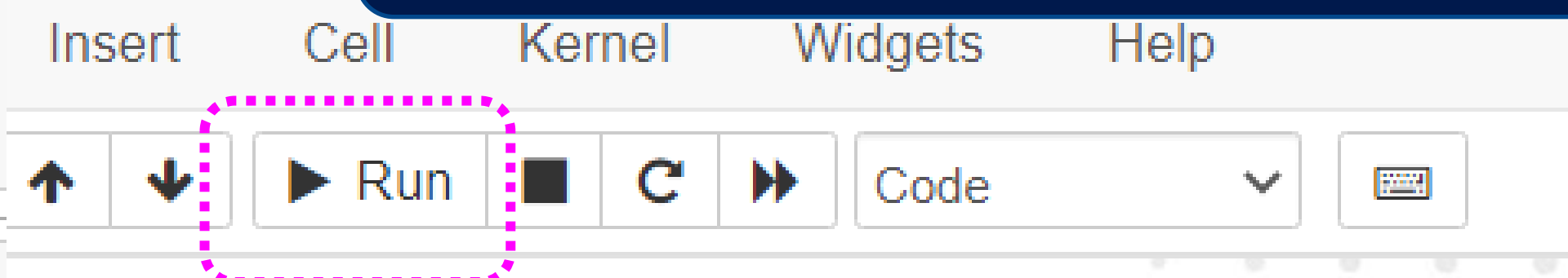
Celda
seleccionada

Notas de clase:

Primera línea de Código:

```
In [ ]: print("Hola mundo")
```

Ejecutamos con “Run” luego de
seleccionar la celda.



Cuando la celda es de tipo “code” podemos ejecutar un código cualquiera, para eso debemos seleccionar la celda y luego hacemos click en:

Una vez hacemos click, **Jupyter** me muestra justo debajo de la celda, cual es el resultado que me entrega este código. Para este ejemplo me debe mostrar la frase: **Hola mundo**.



En nuestro documento creado en **jupyter**, podemos tener tantos códigos como queramos, se **recomienda** programar en **jupyter** cuando sean códigos simples, estamos aprendiendo algo nuevo en programación, o queramos compartir nuestro código.

Primer documento en Jupiter

Notas de clase

Primera línea de Código:

```
In [1]: ▶ print("Hola mundo")
```

Hola mundo

Como podemos ver, justo debajo de la celda aparece el resultado, el código imprime correctamente la frase **“Hola mundo”**



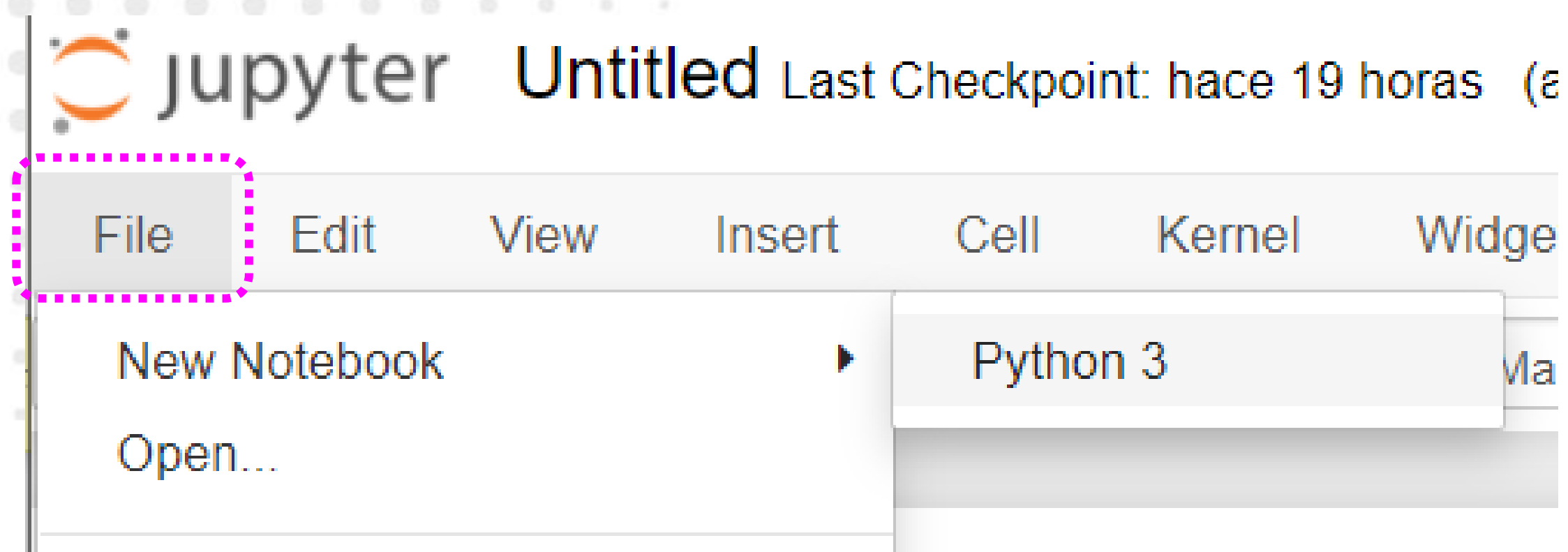
- ¡Todo en **jupyter** se puede ejecutar!
- **Jupyter** es solo una plataforma mas amigable de Phyton, ¡pero todo lo maneja como código de programación!
- Todos los proyectos pueden documentarse y programarse en el mismo lugar.

Como **buena practica de programación**, se recomienda escribir los códigos mas complejos o difíciles, en otro tipo de plataformas, o un editor de código de Anaconda.

Vamos a crear una bitácora mas completa, utilizando imágenes , links y tablas:

Ejemplo: Implementar una bitácora para el proyecto de la tienda. Donde debíamos calcular si generaba beneficios o perdidas.

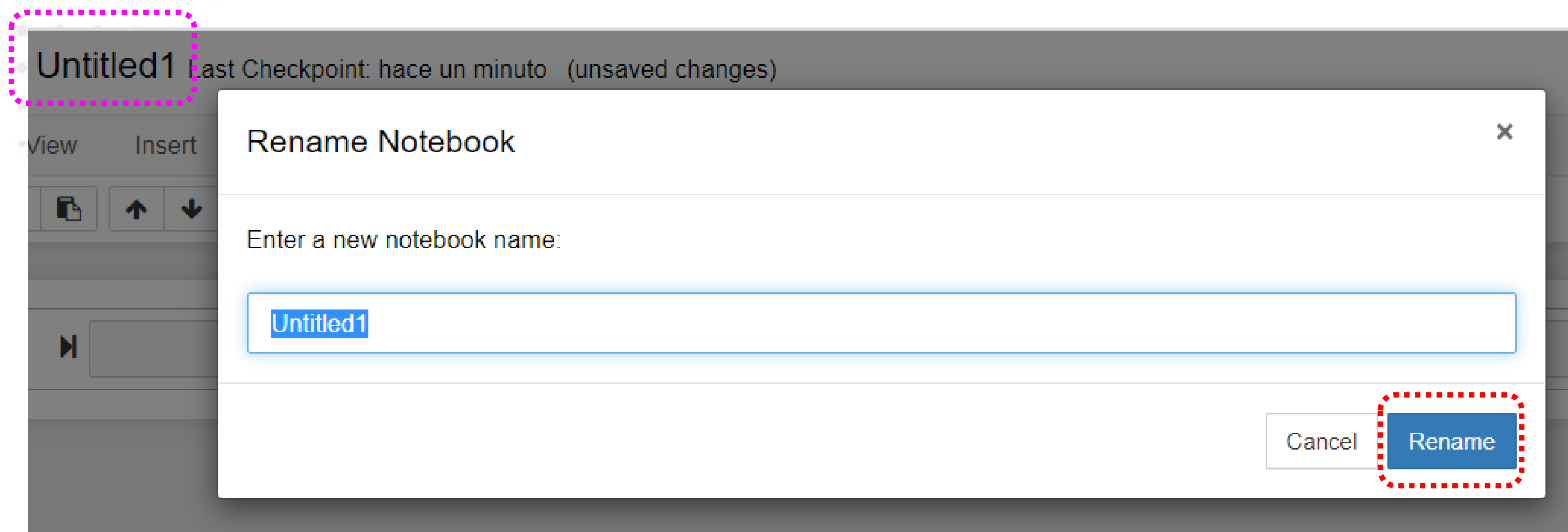




Vamos a crear un documento nuevo solo para el proyecto. Primero, lo creamos con la opción “New” o vamos a **“File”**, seleccionamos la opción de cuaderno nuevo “New Notebook” y seleccionamos el lenguaje, en este caso Python 3.



Ahora cambiemos el nombre del cuaderno, vamos a hacer click sobre el nombre actual. Nos mostrara una ventana donde podemos cambiar su nombre. Una vez escribamos el nuevo nombre solo debemos hacer click en renombrar **“Rename”**.





Como vimos en la clase anterior, debemos seleccionar la opción “**Markdown**” para poder escribir en Jupiter. Vamos a escribir un título, un subtítulo, y un texto corto. En este ejemplo vamos a usar ****** antes y después de una palabra y esto nos muestra una palabra en negrilla.

Bitácora del proyecto tienda

Definición del problema

El cliente necesita un algoritmo que pueda calcular si su negocio, o sus tiendas generan beneficios o perdidas. Tenemos acceso a los datos que maneja la tienda, tanto sus ****ingresos****, como sus ****gastos****.



Bitácora del proyecto tienda

Definición del problema

El cliente necesita un algoritmo que pueda calcular si su negocio, o sus tiendas generan beneficios o pérdidas. Tenemos acceso a los datos que maneja la tienda, tanto sus ****ingresos****, como sus ****gastos****.

Algoritmo

Inicio

1. Leer todos los ingresos
2. Leer todos los gastos
3. Si los Ingresos son mayores o iguales a los gastos
 - Calcular beneficios = Ingresos - Gastos
4. Si los Gastos con mayores o iguales a los ingresos
 - Calcular las pérdidas = Gastos - Ingresos

Fin

Ahora vamos a escribir el algoritmo.

Primero escribimos el sub-subtítulo con **###Algoritmo** y escribimos los pasos del algoritmo.



Para las palabras **inicio** y **fin**, vamos a usar * al inicio y al final de la palabra para que sean **cursivas**. Y luego simplemente escribimos un **numero**, **seguido de un punto**, o un **guion**, para definir que es un **item** o un paso en el algoritmo.



Algoritmo

Inicio

1. Leer todos los ingresos
2. Leer todos los gastos
3. Si los Ingresos son mayores o iguales a los gastos
 - Calcular beneficios = Ingresos - Gastos
4. Si los Gastos con mayores o iguales a los ingresos
 - Calcular las perdidas = Gastos - Ingresos

Fin

Podemos escribir mas información relacionada con el tema si deseamos.



Diagrama de flujo

```
![Imagen](Desktop\Diagrama_de_flujo.png)
```



0

/ Desktop

..

Diagrama_de_flujo.png

Por ultimo vamos a agregar una imagen. Tenemos que usar el comando `![Imagen]()` y entre los paréntesis colocamos la dirección donde se encuentra la imagen en nuestro computador. En nuestro navegador siempre estará abierta la pagina inicial de **jupyter**, así que siempre podemos revisar donde se encuentra el archivo.

hace unos segundos

Solo copiamos el nombre de la carpeta donde esta la imagen, y el nombre del archivo con su extensión = .png



Bitácora del proyecto tienda

Definición del problema

El cliente necesita un algoritmo que pueda calcular si su negocio, o sus tiendas generan beneficios o pérdidas. Tenemos acceso a los datos que maneja la tienda, tanto sus ****ingresos****, como sus ****gastos****.

Algoritmo

Inicio

1. Leer todos los ingresos
2. Leer todos los gastos
3. Si los Ingresos son mayores o iguales a los gastos
 - Calcular beneficios = Ingresos - Gastos
4. Si los Gastos con mayores o iguales a los ingresos
 - Calcular las pérdidas = Gastos - Ingresos

Fin

Diagrama de flujo

! [Imagen](#) (Desktop\Diagrama_de_flujo.png)

Si ejecutamos la celda haciendo click en “Run” el resultado de ser como el de la derecha.





Bitácora del proyecto tienda

Definición del problema

El cliente necesita un algoritmo que pueda calcular si su negocio, o sus tiendas generan beneficios o pérdidas. Tenemos acceso a los datos que maneja la tienda, tanto sus ingresos, como sus gastos.

Algoritmo

Inicio

1. Leer todos los ingresos
2. Leer todos los gastos
3. Si los Ingresos son mayores o iguales a los gastos
 - $\text{Calcular beneficios} = \text{Ingresos} - \text{Gastos}$
4. Si los Gastos son mayores o iguales a los ingresos
 - $\text{Calcular las pérdidas} = \text{Gastos} - \text{Ingresos}$

Fin

Diagrama de flujo





Ejercicios en clase:

Vamos a implementar estas dos preguntas en Jupiter, donde vamos a tener dos preguntas y dos secciones donde debemos escribir el código.



1) ¿Que imprime el siguiente programa ?

```
In [ ]: ▶ x = 43  
        x = x +1
```

```
In [ ]: ▶ print(x)
```

2) Calcule el promedio de las siguientes variables

```
In [ ]: ▶ var1 = 10  
        var2 = 4  
        var3 = 5.5  
        var4 = 67
```



El futuro digital
es de todos

MinTIC

GRACIAS

OPERADO POR:

