

Python

Funciones y manejo de errores

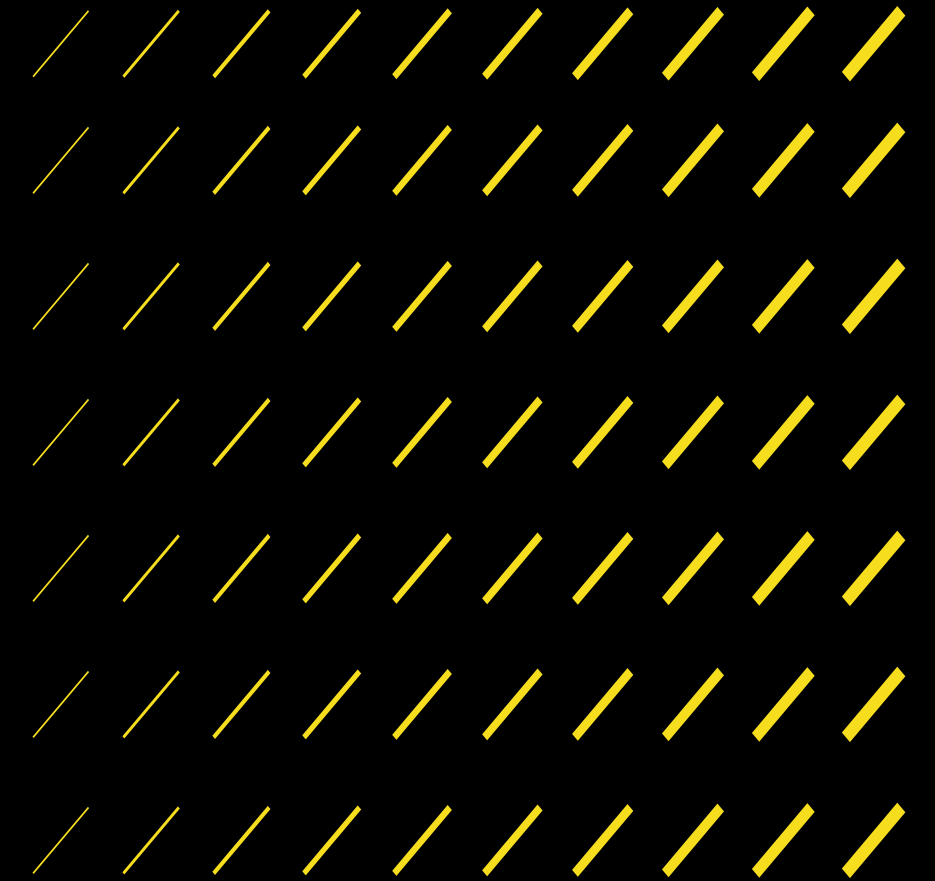
Semana #4

Instructora: Kattherine Hernandez

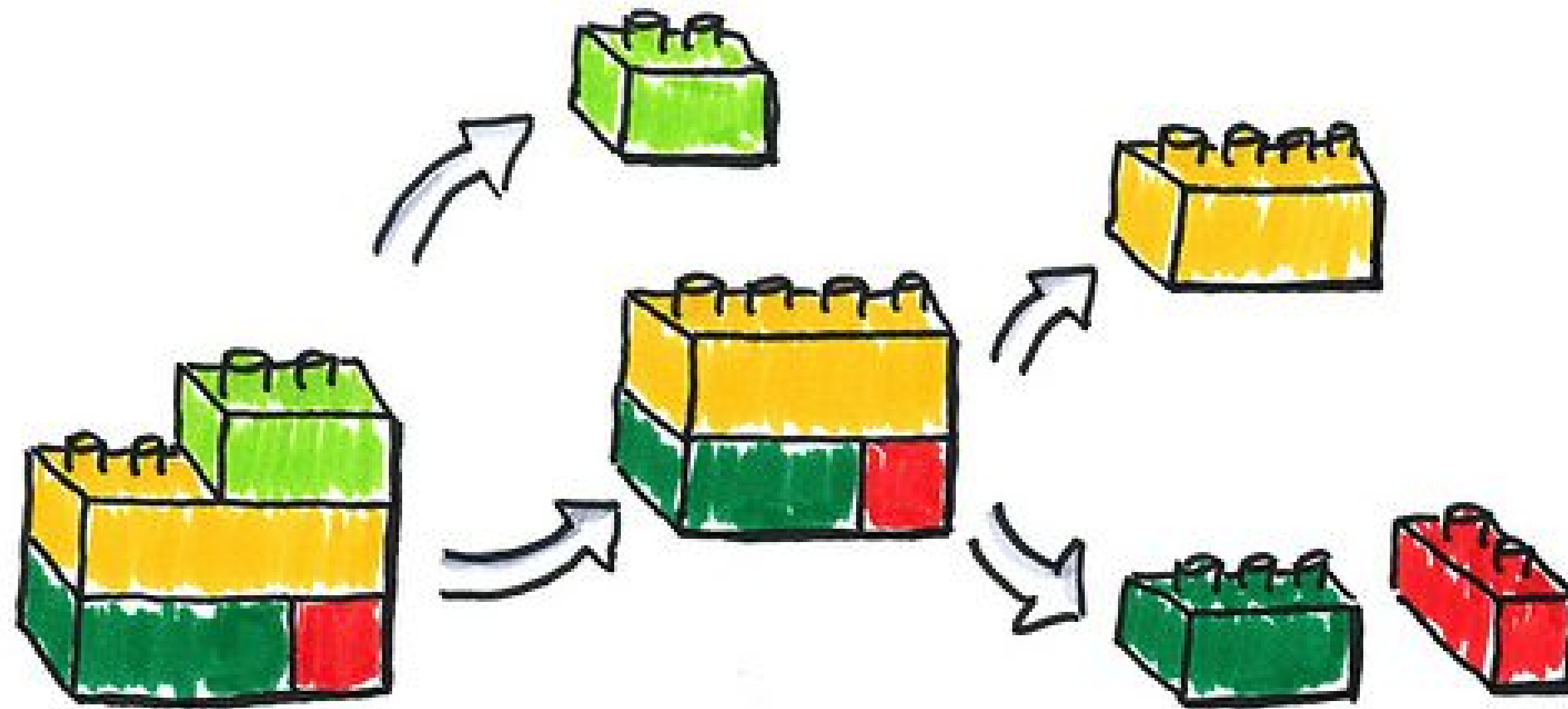


Día 13:

Funciones



Instr: Kattherine Hernandez



Un buen desarrollador divide el código (o mejor dicho: el problema) en piezas aisladas, y codifica cada una de ellas en la forma de una función.

Funciones

Una función es un bloque de código que realiza una tarea específica cuando la función es llamada (invocada). Las funciones son útiles para hacer que el código sea reutilizable, que este mejor organizado, y más legible. Las funciones contienen parámetros y pueden regresar valores.

Las funciones se obtienen:

1. Integradas en Python.
2. A partir de módulos preinstalados.
3. Directamente del código.

```
def nombre_funcion():  
    pass
```

Sintaxis de la **definición** de una función en Python

Uso de una función

- Definición.
- Llamado o invocación.



Imagen tomada de Skills for All, Cisco

Parámetros

Se puede pasar información a las funciones utilizando parámetros. Las funciones pueden tener tantos parámetros como sean necesarios.

```
def nombre_funcion(parametros):  
    pass
```

Sintaxis de la **definición** de una función en Python

Retorno

Para lograr que las funciones devuelvan un valor (pero no solo para ese propósito) se utiliza la instrucción **return** (regresar o retornar).

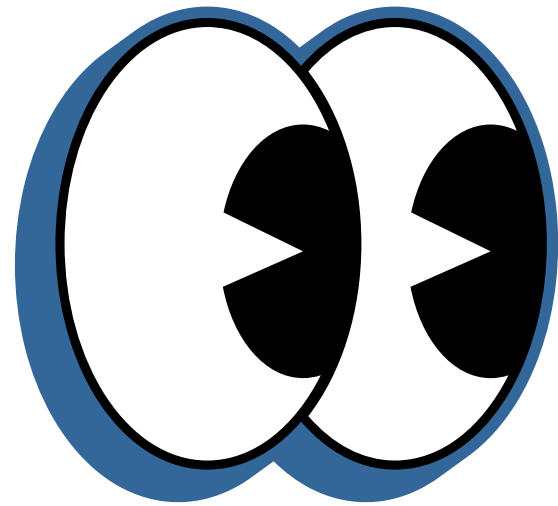
Se puede utilizar de dos formas:

- Sin expresión.
- Con expresión.

```
def nombre_funcion(parametros):  
    return 0
```

Sintaxis de la **definición** de una función en Python

Consideraciones



- No se deben invocar funciones que no se han definido.
- No se le puede dar a una función el mismo nombre que una variable.
- Se debe proveer el mismo número de argumentos como haya parámetros definidos.

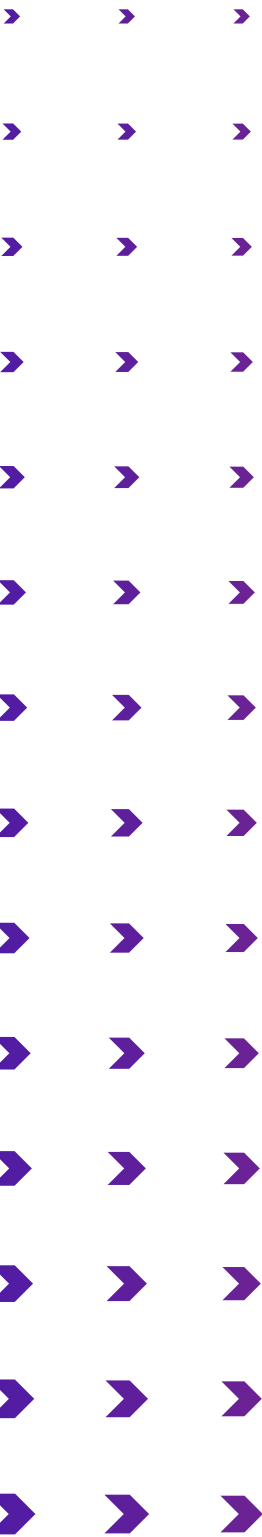


Proyecto final

Refinemos la calculadora de la semana #2

Aspectos a implementar:

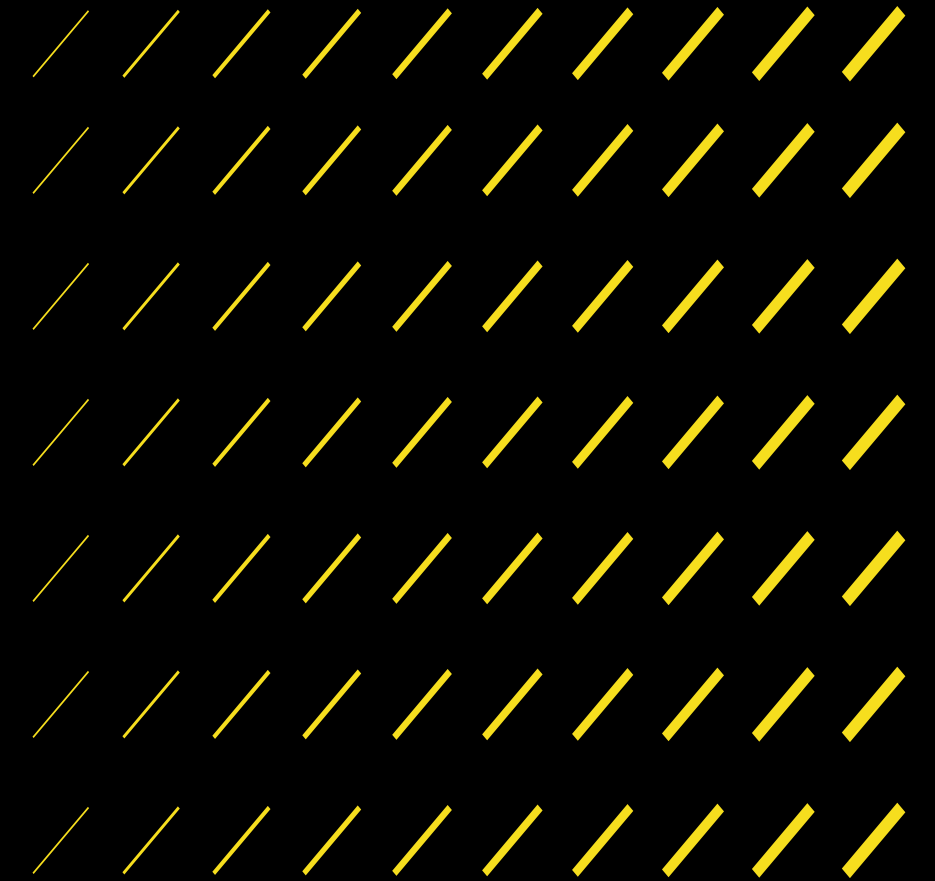
1. Estructuras de control: Validar entradas del usuario (por ejemplo, manejar errores en divisiones por cero o entradas no numéricas).
2. Estructuras de datos: Almacenar un historial de operaciones realizadas usando listas o diccionarios.
3. Funciones: Encapsular cada funcionalidad en funciones específicas.
4. Módulos: Separar el código en módulos, como uno para las operaciones y otro para la interfaz.
5. Bucles: Permitir que el usuario realice varias operaciones sin reiniciar el programa.



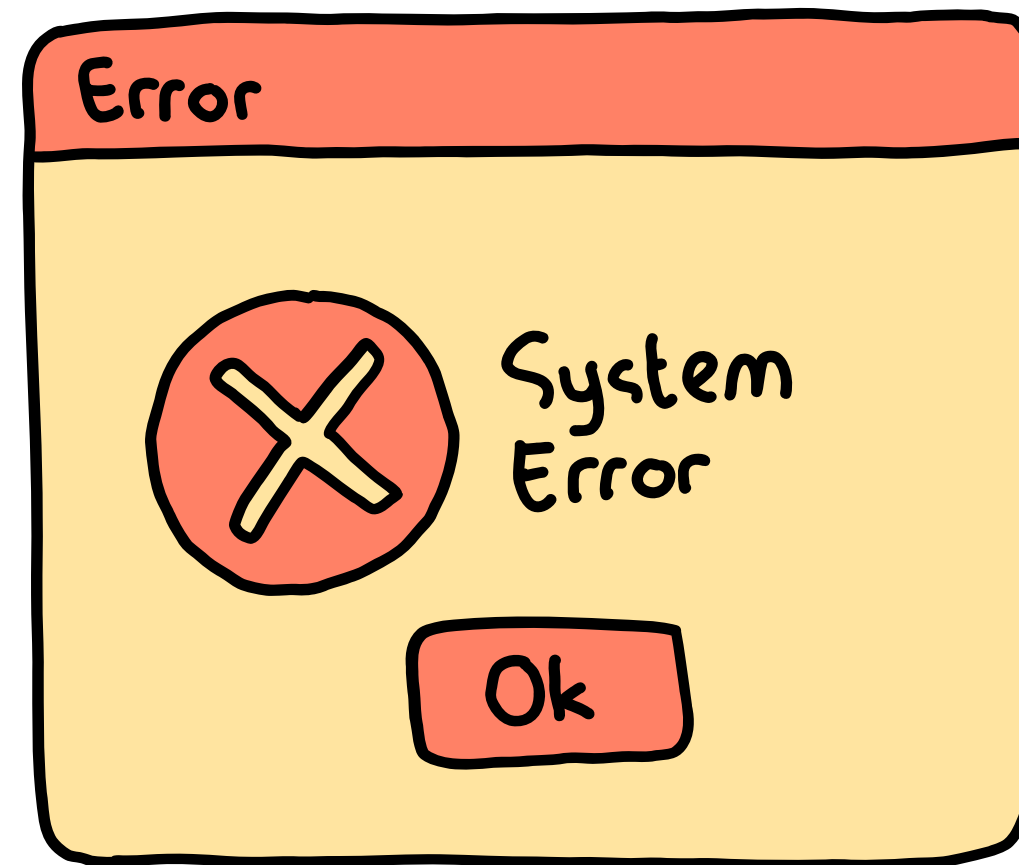


Día 14:

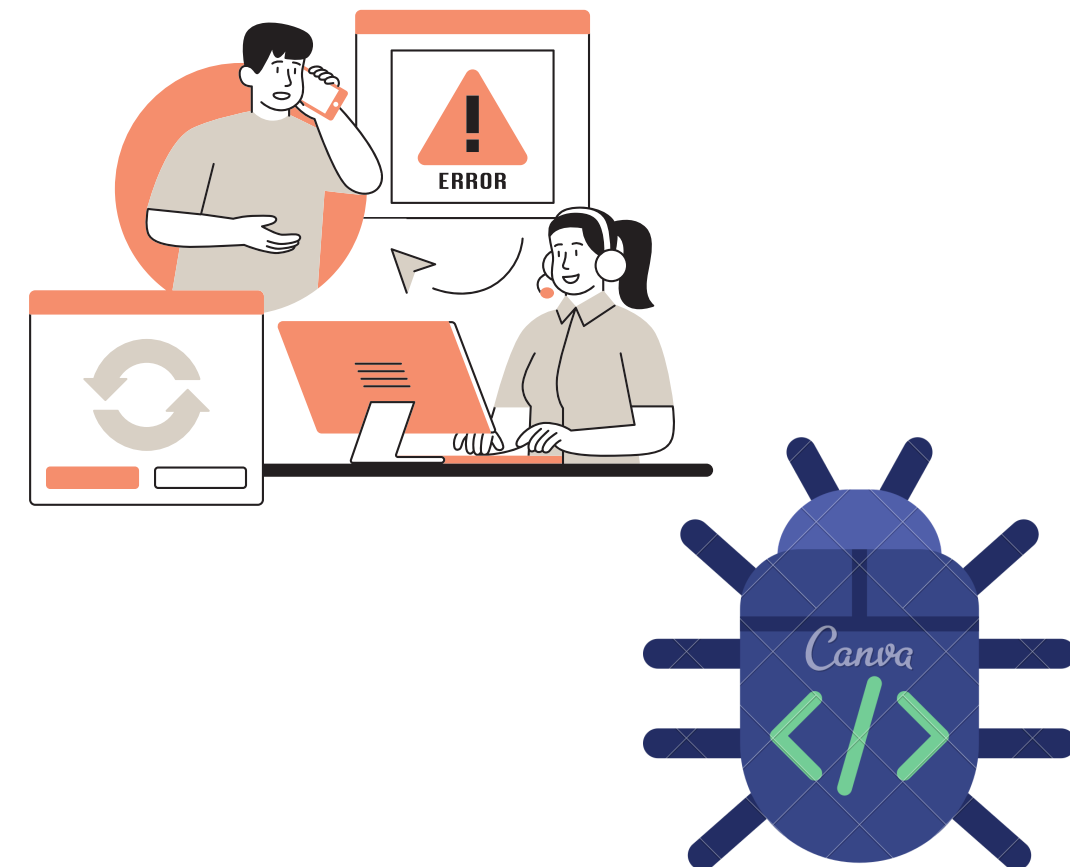
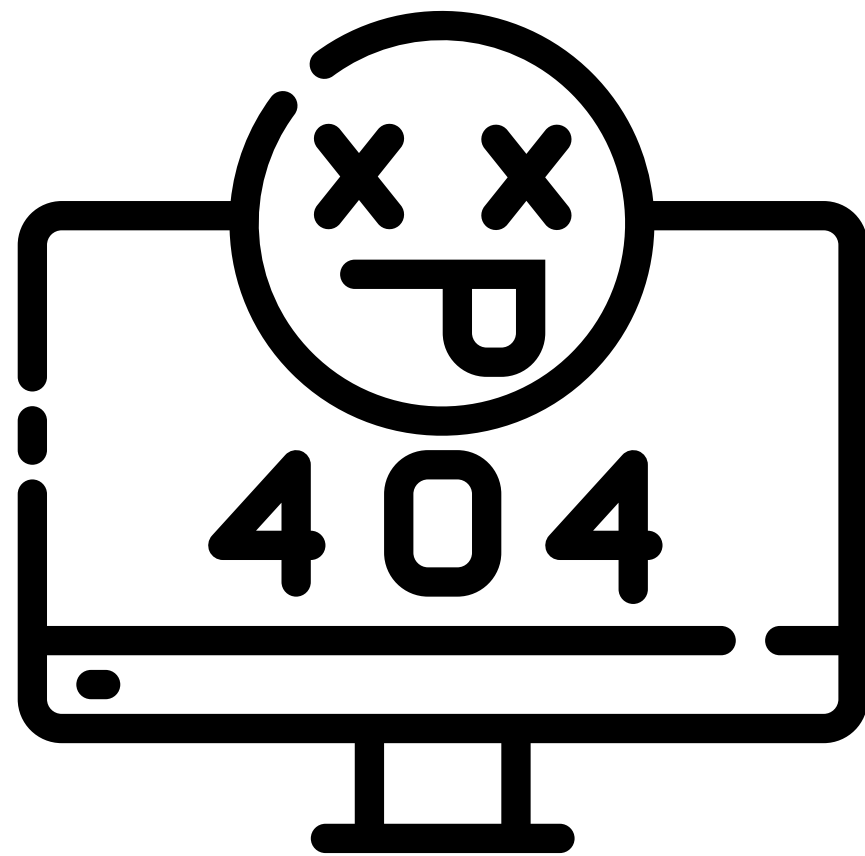
Manejo de errores



Instr: Kattherine Hernandez



Errar es de humanos.



Errores / Excepciones

```
Ingresa un número natural: .15
Traceback (most recent call last):
  File "c:\Users\USER\Documents\Cursos\Curso de Python\Semana4\
pciones.py", line 1, in <module>
    value = int(input('Ingresa un número natural: '))
ValueError: invalid literal for int() with base 10: '.15'
```

Nombre de la excepción



Breve explicación



"es mejor manejar un error cuando ocurre que tratar de evitarlo"

Manejo de excepciones

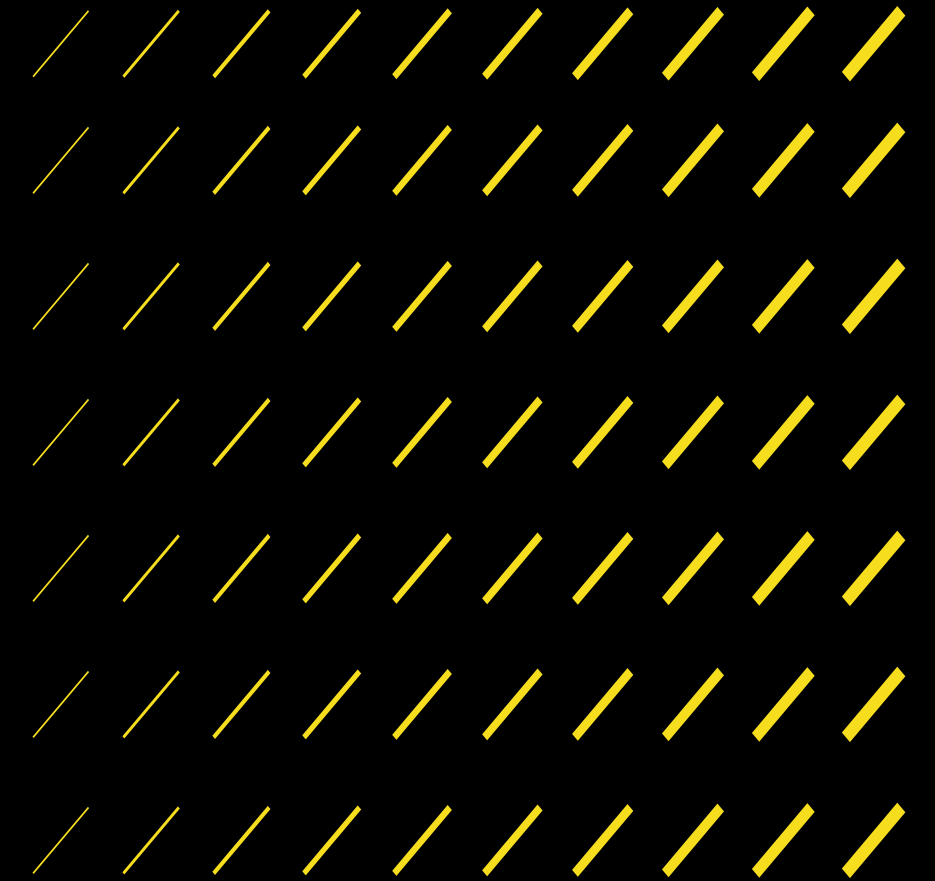
```
try:
    # Escribe las expresiones que puede generar una excepción
    pass
except:
    # Qué piensas hacer si ocurre el error
    pass
```





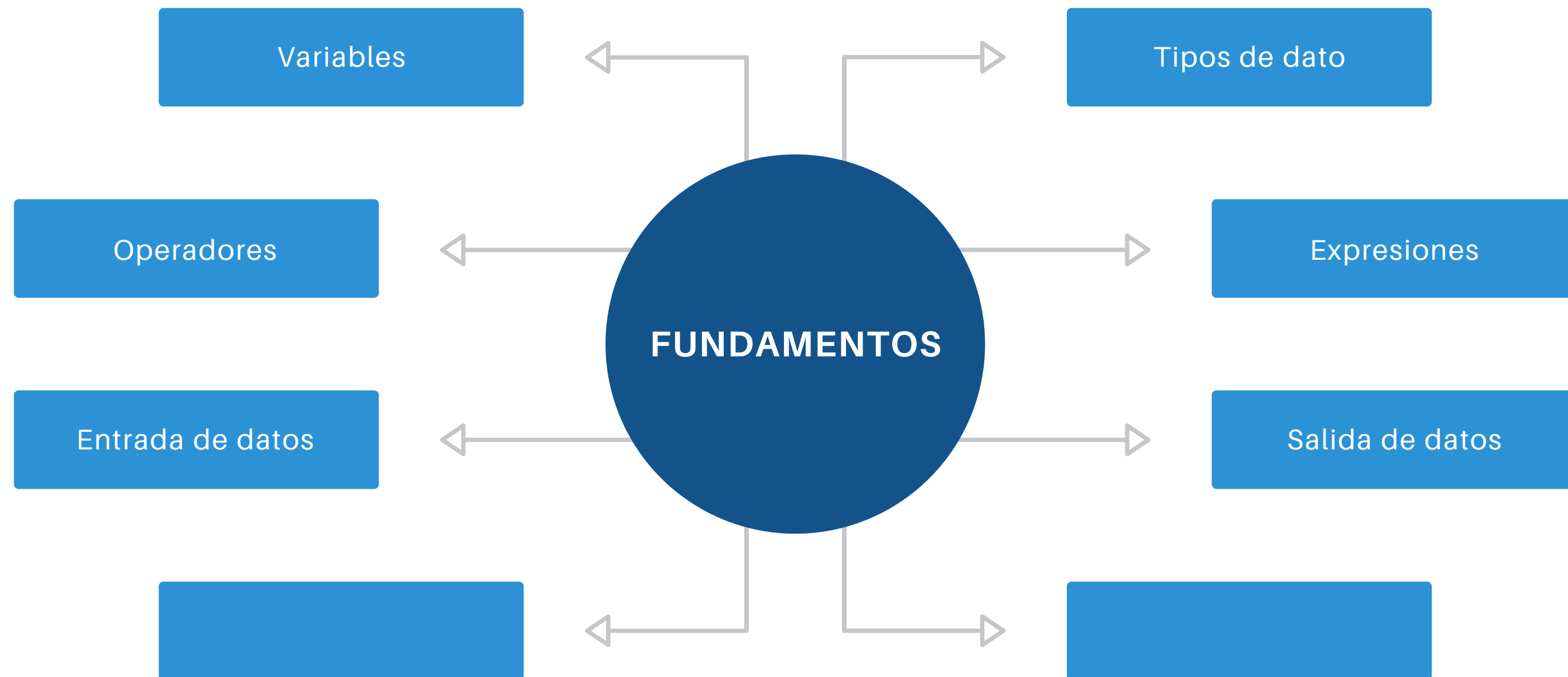
Día 15:

Repaso general



Instr: Kattherine Hernandez

Semana #1:



Semana #2:



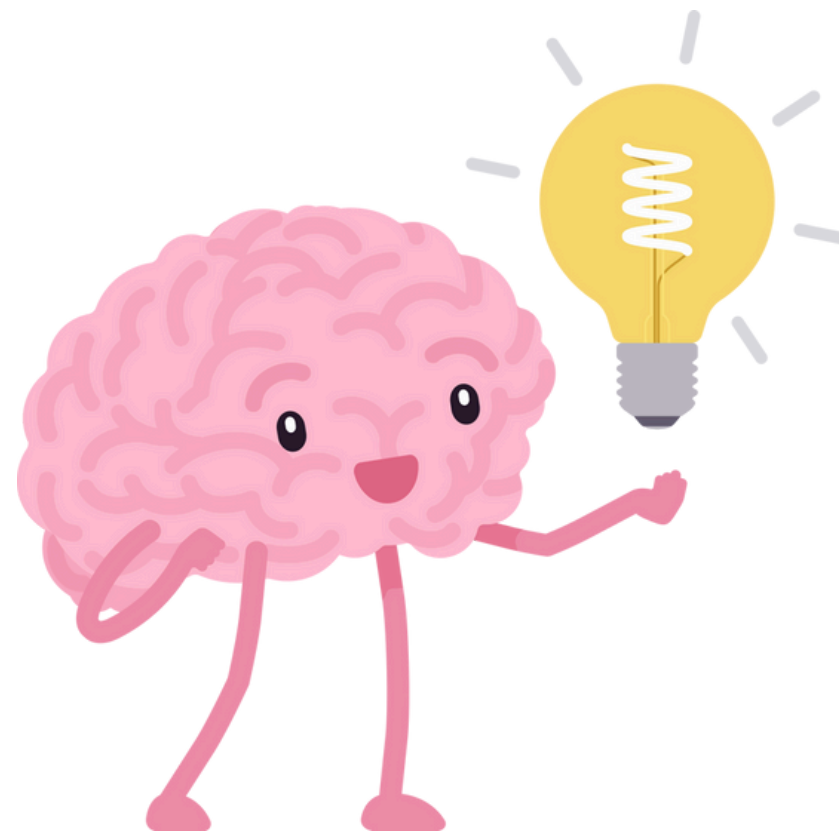
Semana #3:



Semana #4:



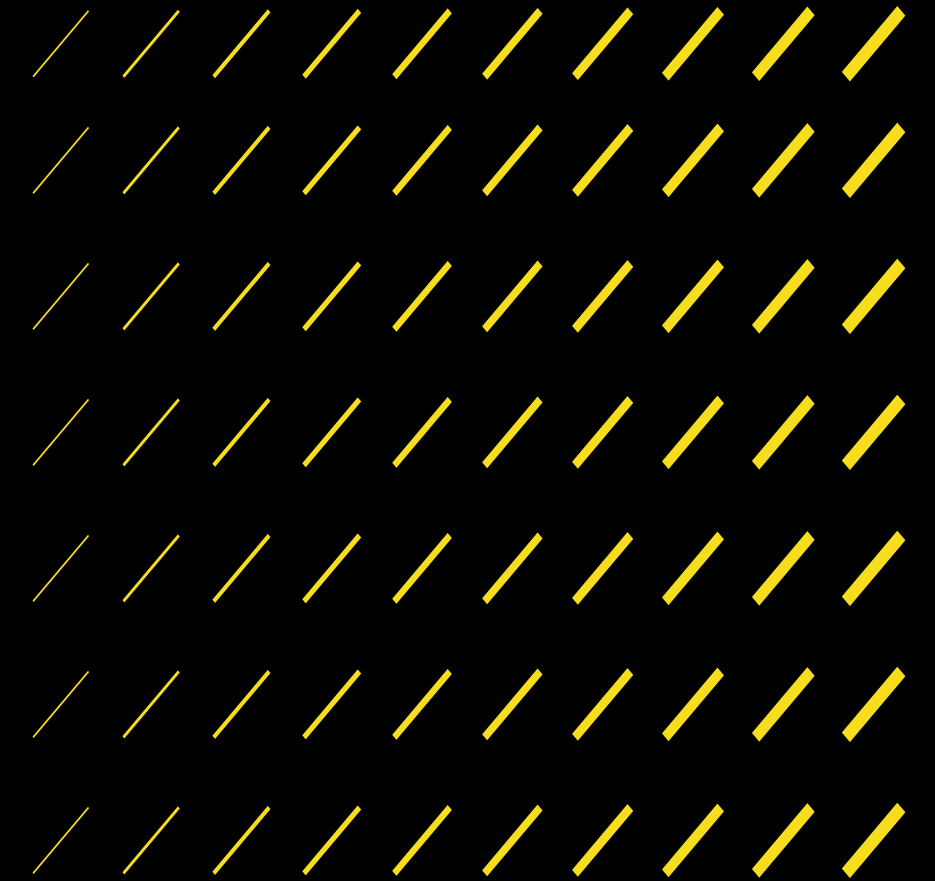
**Elaboremos una pequeña
calculadora de descuentos.**





Día 16:

Proyecto final



Instr: Kattherine Hernandez

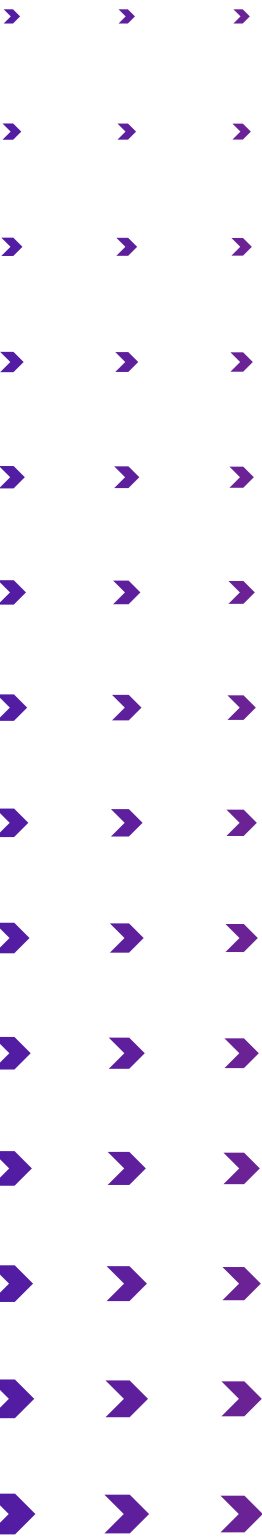


Proyecto final

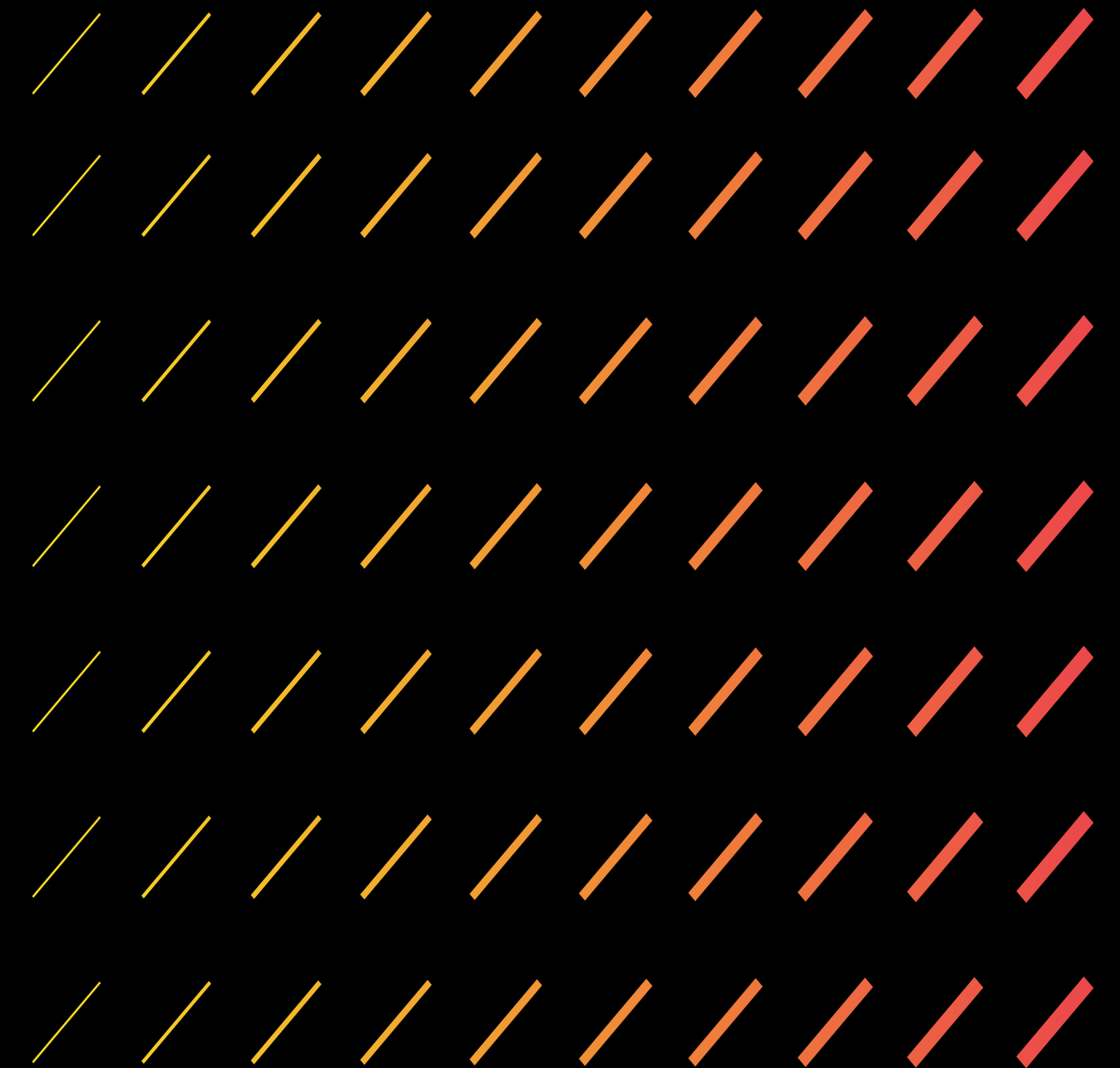
Refinemos la calculadora de la semana #2

Aspectos a implementar:

1. Estructuras de control: Validar entradas del usuario (por ejemplo, manejar errores en divisiones por cero o entradas no numéricas).
2. Estructuras de datos: Almacenar un historial de operaciones realizadas usando listas o diccionarios.
3. Funciones: Encapsular cada funcionalidad en funciones específicas.
4. Módulos: Separar el código en módulos, como uno para las operaciones y otro para la interfaz.
5. Bucles: Permitir que el usuario realice varias operaciones sin reiniciar el programa.
6. Función para ver historial de operaciones realizadas.
7. Mostrar el historial de operaciones al final de la ejecución con la cantidad estas.



**¡Felicidades por haber
llegado hasta el final!**



Instr. Katherine Hernandez