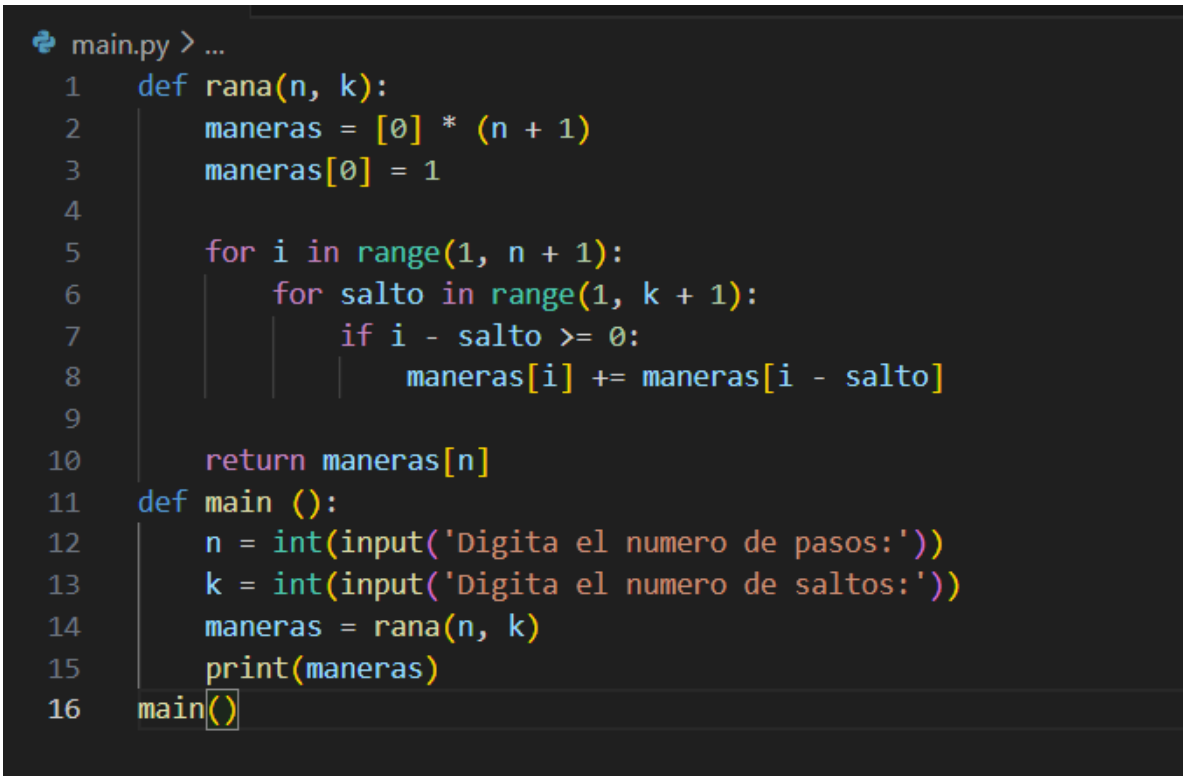
Análisis general del código



El código se encarda de almacenar dos variables, donde se le suma un n+1 cada ese n se recorre en un for donde va de 1 a n+1, donde ponemos unas condiciones para que el salto sea mayor o igual a cero, retorna el n o la posición, por así decirlo es un Fibonacci, pero guardando la posición en la que está.

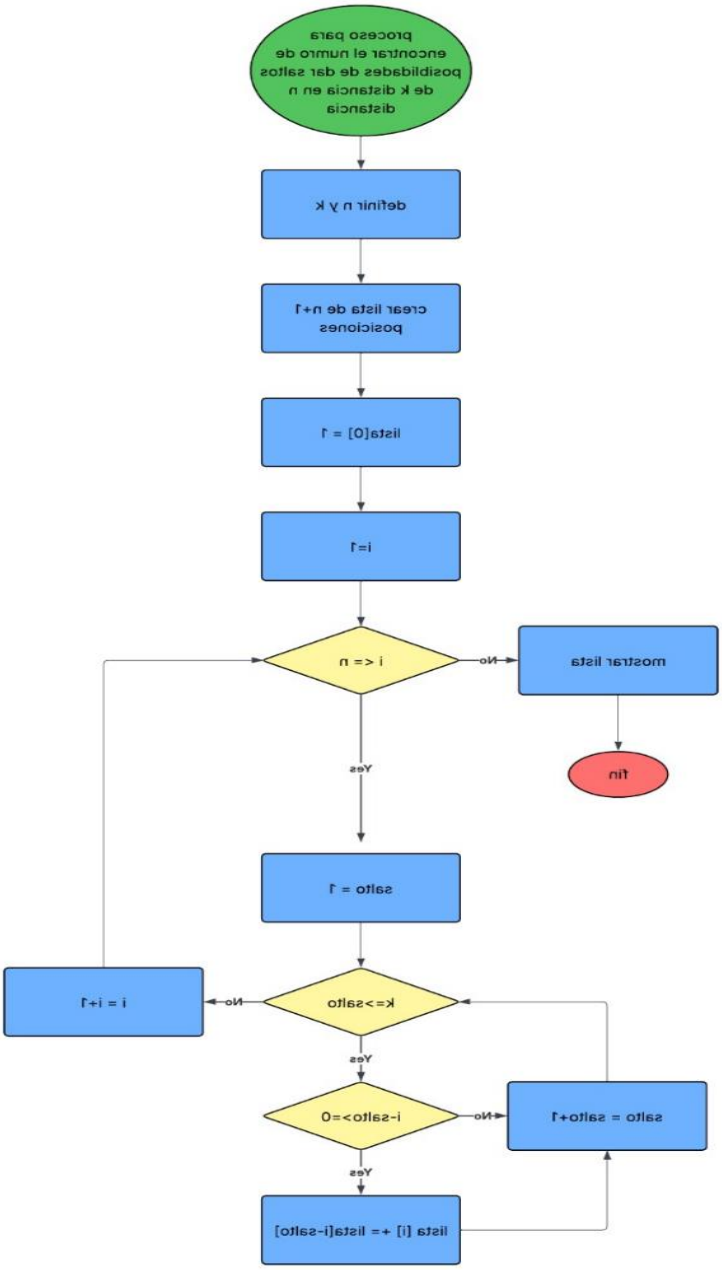
REQUERIMIENTOS	HISTORIA DE USUARIO
RF-001 – Mostrar resultados del recorrido El sistema debe mostrar la información resultante del recorrido del sapo.	Mostrar la posición final del sapo, la cantidad de saltos realizados y las maneras posibles en que puede recorrer el trayecto.
RF-002 – Calcular saltos El sistema debe calcular los saltos realizados por el sapo durante el recorrido.	calcular los saltos del sapo desde la posición inicial hasta la posición final para conocer su recorrido total.
RF-003 – Determinar recorridos posibles El sistema debe determinar las diferentes maneras en que el sapo puede recorrer el trayecto.	Determinar las posibles formas en que el sapo puede completar el recorrido según los saltos definidos.
REQUERIMIENTOS NO RELACIONALES	HISTORIA DE USUARIOS
RNF-001 – Validar posición inicial El sistema debe validar que la posición inicial de la rana sea un número entero.	Validar que la posición ingresada para la rana sea un número entero antes de iniciar el recorrido.
RNF-002 – Validar rango de entrada El sistema debe validar que el valor ingresado esté en el rango de 1 hasta n+1.	Validar que el valor ingresado se encuentre dentro del rango permitido de 1 hasta n+1.
RNF-003 – Validar valor de saltos El sistema debe validar que el número de saltos sea mayor o igual a cero.	Validar que el número de saltos ingresado no sea negativo para garantizar un recorrido válido.
RNF-004 – Validar tipo de dato El sistema debe validar que todas las entradas numéricas sean valores enteros.	Verificar que los datos ingresados sean enteros y rechazar valores decimales o no numéricos.

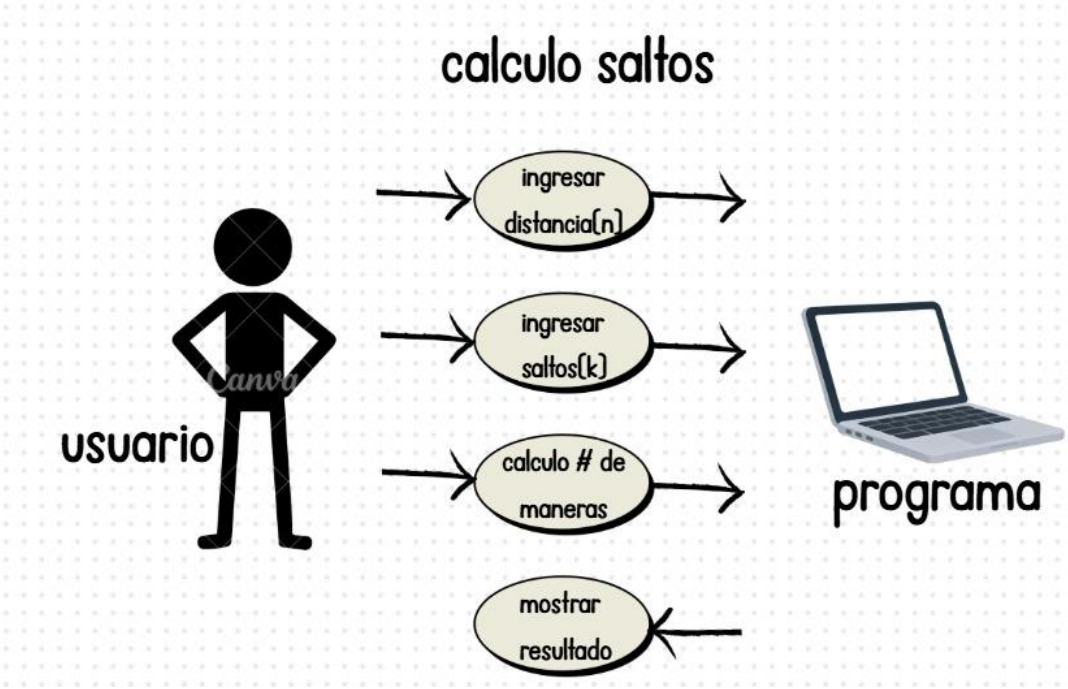
Tabla de complejidad:

COST	TIMES
C1	1
C2	1
C3	$n+1$
C4	n
C5+T(i-k)	1
C6	1
C7	1
C8	1
C9	1

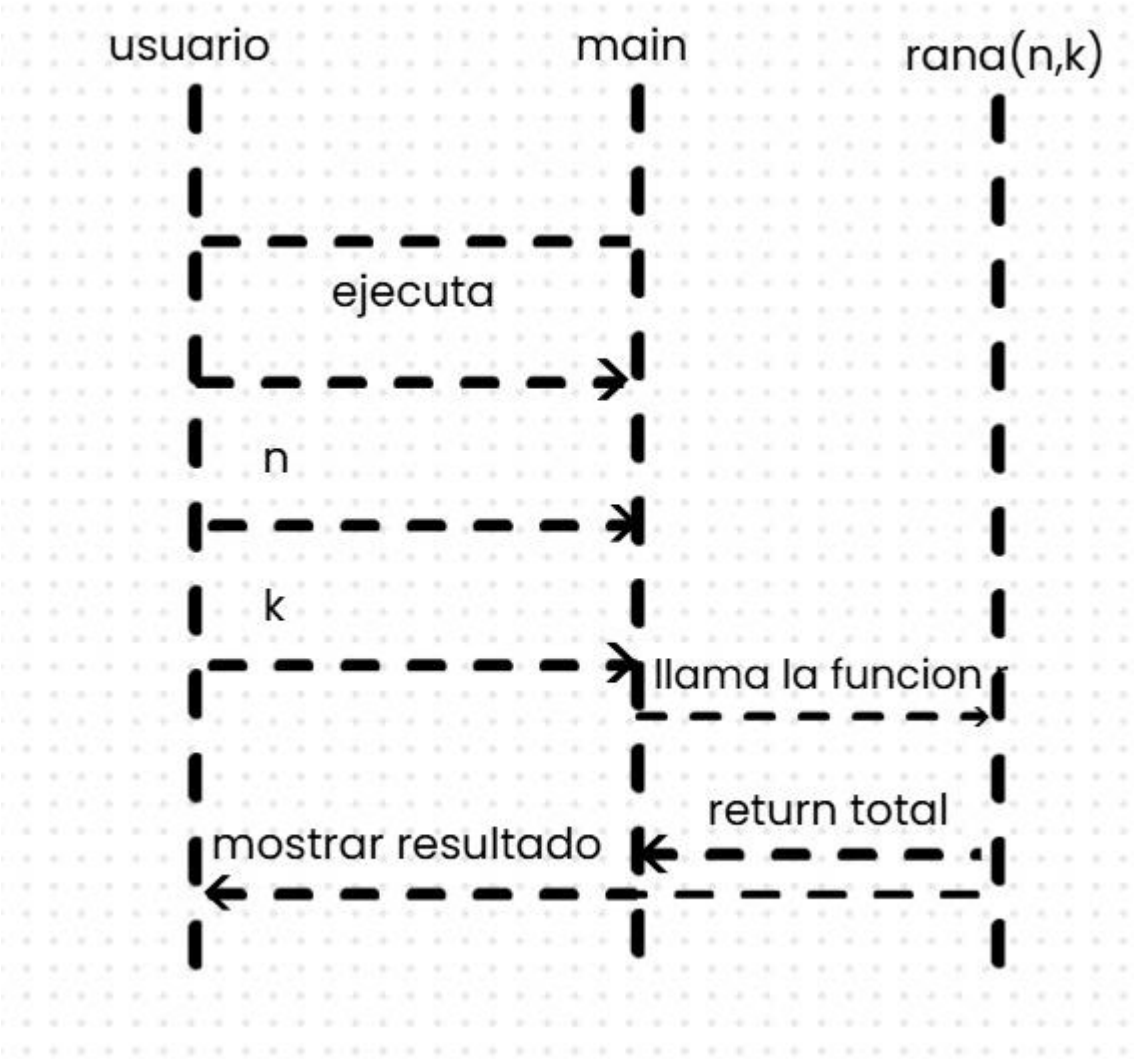
$$c1 + c2 + c3 + c4 + \big(c5 + (i-k)\big) + c6 + c7 + c8 + c9 = n^2$$

Diagrama de flujos:




calculo saltos
ingresar distancia(n)
ingresar saltos(k)
calculo # de maneras
mostrar resultado
usuario
programa

X|


usuario
main
rana(n,k)
ejecuta
n
k
llama la funcion
return total
mostrar resultado