

Ciclos y sumatorias

Objetivos

- Conocer las operaciones de sumatoria.
- Crear diagramas de flujo de problemas de sumatoria.
- Escribir en Ruby el código de una sumatoria.
- Conocer la diferencia de un contador y de un acumulador.

Motivación

En este capítulo aprenderemos a crear programas que, además de utilizar ciclos, operen sobre otra variable.

Esto nos permitirá resolver diversos tipos de problemas, la mayoría de ellos corresponde a problemas del tipo matemáticos pero son aplicables a múltiples contextos.

Introducción a sumatorias

Para algunos -que no gustan de las matemáticas- el término sumatoria puede sonar algo intimidante. Sin embargo, para resolver una sumatoria, solo necesitamos saber una cosa: sumar.

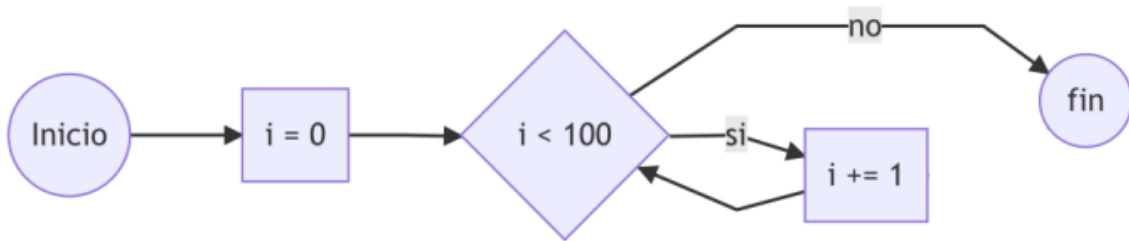
Sumando números

$$1 + 2 + 3 + \dots + 100 = ?$$

La sumatoria consiste en sumar todos los números de una secuencia. Por ejemplo: sumar todos los números entre 1 y 100. Esto no solo sirve para resolver ecuaciones matemáticas, sino también para que generemos las habilidades de abstracción necesarias para resolver diversos problemas.

Resolver esto es muy similar a contar las cien veces, pero además de contar, vamos ir guardando la suma **de cada iteración**.

Comencemos desde la base del código anterior



```

i = 0
while i < 100
  i += 1
end
  
```

Luego, si queremos guardar la suma en cada iteración, necesitamos una variable para ir guardando los datos.

```

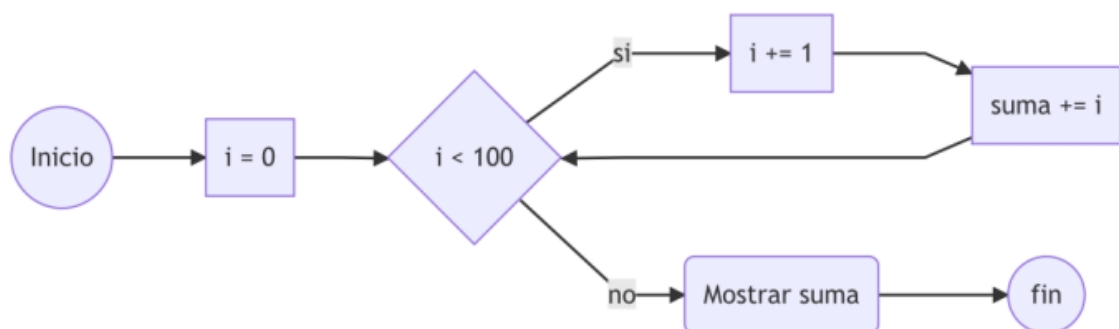
i = 0
while i < 10
  i += 1
  suma += i
end
  
```

Todavía nos falta un detalle para que funcione. No podemos sumarle algo a una variable que no existe; las variables con las que contamos o sumamos deben ser, primero, **inicializadas**.

```

i = 0
suma = 0
while i < 10
  i += 1
  suma += i
end
  
```

Veamos el diagrama de flujo para reforzar lo aprendido.



La instrucción `suma = suma + i` es la encargada de aumentar el valor de la variable `suma` en cada iteración.

Este comportamiento, sumar y almacenar secuencialmente valores variables, se conoce como la implementación de un **acumulador**. En este ejemplo, la variable `suma` está acumulando la suma de los valores en cada iteración.

Resumen

En este capítulo aprendimos sobre contadores y acumuladores.

Tanto los contadores como acumuladores son **ampliamente utilizados** en programación.

- Contador: Aumenta de 1 en 1
 - `cont = cont + 1`
 - `cont += 1`
- Acumulador: Aumenta en función a `valor`
 - `acu = acu + valor`
 - `acu += valor`