

María Paula Llano

Departamento de Ciencias de
la Atmósfera y los Océanos

Laboratorio de Procesamiento de Información Meteorológica/Oceanográfica

Algoritmos

Algoritmos: "recetita en papel"

¿Cómo se resuelve un problema usando una computadora?

La idea de esta materia es brindar diferentes herramientas para poder solucionar un amplio rango de problemas “complejos” usando una computadora.

Resolver un problema usando computadoras es más que el simple hecho de escribir un programa.

Escribir un programa es claramente una parte del camino a la solución, pero no es toda la historia.

En esta clase vamos a ver algunos lineamientos para comenzar a responder esta pregunta.

Programa (script) es la solución a ese problema pero no es todo lo que hay que hacer. Esas otras cosas vamos a aprender hoy

Puntos de partida

- La computadora es un dispositivo que puede ser usado para resolver un problema determinado (con determinadas restricciones).
- Un programa de computadora es un conjunto de instrucciones que la computadora puede ejecutar. script: sentencias son instrucciones. Programa son todas esas instrucciones

Estas instrucciones describen los pasos que deben ser llevados a cabo de acuerdo con el plan diseñado para resolver el problema.

- Podemos definir a un algoritmo como el plan que vamos a utilizar para resolver el problema. como quiero resolver yo ese problema
- Los algoritmos son diseñados por las personas. depende de cada uno. Pueden ser distintas soluciones para el mismo problema
- Los algoritmos deben ser traducidos a un lenguaje de programación por una persona.

algoritmo no es el script. son esas ideas que voy pensando para hacer el script.

Algoritmo es como un pseudocodigo/ diagrama de flujo (creo)

Conjunto de pasos a seguir (puede variar de acuerdo con el enfoque)

- **Paso 1:** Obtener una descripción del desafío y problema. La descripción debe ser lo más precisa posible. lo mas desarrollado para que no tenga "baches" despues
- **Paso 2:** Analizar y entender el problema. No podemos avanzar con la solución si hay puntos del problema planteado que no son claros.
- **Paso 3:** Diseñar un algoritmo de alto-nivel. Alto nivel en esta instancia implica “sin mucho detalle”, destacando los aspectos más salientes. "tirar ideas" de lo que necesito para llevar a cabo la resolución
- **Paso 4:** Definir y revisar el algoritmo incluyendo los detalles. En este punto los pasos a seguir deben estar claros y deben estar dentro de lo que la computadora puede realizar. En esta instancia podemos usar herramientas como el *pseudo-código* y los *diagramas de flujo*.

Conjunto de pasos a seguir (continuación)

- **Paso 5:** Escribir el código. Es decir para cada acción del algoritmo, indicar cómo sería la instrucción correspondiente en el lenguaje de programación que hayamos elegido. aca me siento en la compu
- **Paso 6:** Probar el funcionamiento del código. Típicamente buscar casos de prueba donde sea fácil verificar que el resultado sea correcto y probar el código con la mayor cantidad de casos posibles.
- **Paso 7:** Hacer una corrección (debug) del código. Es frecuente que la primera versión de un código contenga muchos errores de diferente tipo. Esta etapa junto con el testeo es fundamental para garantizar que el código sea correcto. 6 y 7 van juntos. Primero 6 y desp 7 y asi sucesivamente
- **Paso 8:** Evaluar la eficiencia del código. ¿Es el código eficiente en términos del uso de los recursos de la computadora? ¿Podría ser más eficiente? Esto muchas veces requiere volver a repensar los algoritmos para sacar el mayor provecho posible de la capacidad de cálculo y almacenamiento de una computadora. eficiencia: poco costo computacional

Problema: Calcular la media aritmética de una lista de 10 valores almacenados en la memoria de la máquina.

● **Paso 1:** *Obtener una descripción del desafío y problema. La descripción debe ser lo más precisa posible.* Por ejemplo, calcular la media aritmética de una lista de números.

Si no sabemos lo que es la media aritmética podemos consultar en internet y obtenemos la expresión :

Media aritmética

sumar los 10 valores y dividirlo por 10

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$
$$= \frac{x_1 + x_2 + \dots + x_N}{N}$$

● **Paso 2:** *Analizar y entender el problema. No podemos avanzar con la solución si hay puntos del problema planteado que no son claros.*

En el ejemplo anterior, tenemos que estar familiarizados con el concepto de media aritmética. Tenemos además que comprender la naturaleza de los datos de entrada y exactamente qué tipo de dato pretendemos como resultado (si un valor mostrado en la pantalla, si un gráfico de algún tipo, si queremos almacenar el resultado en el disco en forma permanente, etc)

como hacer para hacer "venir" esos datos, el formato en el q estan los datos

Tenemos la expresión matemática que define la cantidad que tenemos que calcular. Sabemos también que en nuestro caso $n = 10$.

Sabemos que una computadora puede recibir las siguientes instrucciones:

- obtener (leer) datos de la memoria y almacenar (escribir) nuevos datos en la memoria saber leer y tamb guardarlos en distintos formatos
- operar aritméticamente sobre los datos sea el tipo de dato q sea
- ejecutar diferentes instrucciones dependiendo de una prueba lógica (condicionales) los if
preguntar y si es T seguir un camino y si es F otro
- repetir un determinado conjunto de acciones las veces que le indiquemos segun el ciclo: for, while o repeat
usando los ciclos repito algo sistemáticamente

Esta información representa las herramientas con las que contamos para resolver el problema. Esto es fundamental ya que la solución debe hacer uso de estas herramientas.

- **Paso 3:** *Diseñar un algoritmo de alto-nivel. Alto nivel acá implica sin mucho detalle, destacar los aspectos más salientes.*

Esencialmente debemos acceder a los 10 valores almacenados en la memoria, sumarlos uno a uno y luego tomar la suma y dividir por 10.

- **Paso 4:** *Definir y revisar el algoritmo incluyendo los detalles.*

Podemos empezar dándole un “nombre” a la variable donde almacenamos el resultado.

Podemos llamarlo **ValorMedio** por ejemplo.

Es una buena práctica (aunque no todos los lenguajes lo requieran) dar un valor inicial a cada una de las variables que vayamos definiendo y usando en nuestro algoritmo.

ValorMedio = 0.0 al definir una variable, inicialmente darle el valor cero esta bueno (segun lo que entendí je)

Es decir, al inicio del cálculo la media vale 0, y su valor va ir cambiando a medida que vayamos sumando los números de la lista.

Paso 4 (continuación): Definir y revisar el algoritmo incluyendo los detalles.

ValorMedio = 0.0

Una vez que definimos el nombre a través del cual nos vamos a referir al resultado, podemos proceder a calcularlo.

Supongamos que los números de la lista están guardados en otra variable que se llama **MiLista** . MiLista es vector que contiene los 10 valores que voy a usar

Un posible algoritmo consiste en lo siguiente:

Repetir la siguiente acción 10 veces (con i variando desde 1 a 10):

ValorMedio \leftarrow ^{Valía cero} **ValorMedio** + el i -ésimo elemento de **MiLista**

La repetición es una de las instrucciones que la computadora puede ejecutar, con lo cual este algoritmo está dentro de las restricciones planteadas. escribir un ciclo sin usar la palabra for

Finalmente dividimos por 10 para obtener la media.

ValorMedio \leftarrow **ValorMedio** / 10

Paso 4 (continuación): *Definir y revisar el algoritmo incluyendo los detalles.*

Usando pseudo-código el algoritmo se vería como lo siguiente:

MiLista (ya estaba en la memoria de la máquina, es una variable con 10 elementos) el vector

ValorMedio = 0.0 ese primer valor que le asigno

repetir con i desde 1 hasta 10

ValorMedio = ValorMedio + el *i*-esimo elemento de MiLista

final de repetición

ValorMedio = ValorMedio / 10

vuelvo a lo q escribí antes pero le pongo un poco mas de detalle

Paso 5: Traducir el algoritmo en términos del lenguaje de programación elegido. En este caso R. # son comentarios."con

#Defino la lista de valores sobre la cual voy a calcular la media.

MiLista <- c(10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 , 100)

#Le doy un valor inicial a la variable donde voy a almacenar la media de la
lista

ValorMedio <- 0.0

#Armo una repetición (loop, ciclo) que va tomando uno a uno los valores de
MiLista y los va sumando en ValorMedio

valor es lo mismo q poner i o j

for (valor in MiLista) {

ValorMedio <- ValorMedio + valor

}

Forma de escribir con i (CHEQUEAR POR LAS DUDAS)

for(i in 1:10)

ValorMedio<-ValorMedio+MiLista[i]

#Divido ValorMedio por 10 de acuerdo a la definición de media aritmética

ValorMedio <- ValorMedio / 10

#Muestro el resultado de la media aritmética por pantalla.

print(ValorMedio)

```
1
2
3
4 #Defino la lista de valores sobre la cual voy a calcular la media.
5 MiLista <- c( 10,20,30,40,50,60,70,80,90,100 )
6
7 #Le doy un valor inicial a la variable donde voy a almacenar la media de la lista
8 valorMedio <- 0.0
9
10 #Armo una repeticion (loop, ciclo) que va tomando uno a uno los valores de MiLista y los va sumando en valorMedio
11 for ( valor in MiLista ) {
12   valorMedio <- valorMedio + valor
13 }
14
15 #Divido valorMedio por 10 de acuerdo a la definicion de media aritmetica
16 valorMedio <- valorMedio / 10
17
18 #Muestro el resultado de la media aritmetica por pantalla.
19 print( valorMedio)
20
```

Paso 6, 7: *Probar el funcionamiento del código y corregirlo si es necesario.*

Podemos probar con diferentes valores de números en **MiLista**, y verificar que el resultado sea correcto. Por ejemplo, si todos los números son iguales, el valor medio obtenido debe ser igual a los elementos de **MiLista**.

Si existe alguna otra herramienta que realice alguna función similar podemos usarla para comparar los resultados. Por ejemplo, como R es un lenguaje de muy alto nivel, tiene una ***función intrínseca*** que calcula la media de una secuencia de números.

Las funciones intrínsecas son aquellas que forman parte del lenguaje y por ende han sido extensamente chequeadas.

#Defino la lista de valores sobre la cual voy a calcular la media.

MiLista <- c(10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 , 100)

#Le doy un valor inicial a la variable donde voy a almacenar la media de la lista

ValorMedio <- mean(MiLista)

Por supuesto que de entrada hubiera sido más fácil utilizar esta función que programar la media aritmética desde 0.

Pero en este caso (y en muchos casos en esta materia) lo hacemos para entender la lógica de la programación y cómo utilizar los elementos del lenguaje en la resolución de problemas.

funciones intrínsecas tienen un script oculto q hace todo eso

Paso 6, 7 (continuación): Probar el funcionamiento del código y corregirlo si es necesario.

Resulta que los valores que estoy promediando representan la altura de un edificio por encima del nivel de la calle. Por ende deben ser valores positivos. No obstante, mientras aplico mi programa para promediar la altura de diferentes conjuntos de edificios. Encuentro la siguiente serie:

```
MiLista <- c( 10 , 20 , 30 , 40 , -999 , 60 , 70 , 80 , 90 , 100 )
```

Con lo que el resultado obtenido es -49.9 !! Un valor negativo y que por ende no puede ser una altura. Esto claramente se debe a que el 5to elemento de la serie es -999.

Hablo con el proveedor de los datos y me dice que cuando falta el dato correspondiente a algún edificio, la falta de información se codifica como -999. Entonces ese valor no corresponde a un dato y no debe ser considerado en el cálculo de la altura promedio.

Esto es un ejemplo de un aspecto imprevisto en el diseño original que requiere de un rediseño del algoritmo. problema con los datos reales

-999 es un numero, pero no es un dato que me sirva

Paso 6, 7 (continuación):

Repensado el algoritmo me doy cuenta que antes de sumar un dato, voy a tener que verificar si ese dato es válido o no. Esto se puede resolver en programación mediante el uso de condicionales.

MiLista (ya estaba en la memoria de la máquina, es una variable con 10 elementos)

ValorMedio = 0.0

CantidadDatosValidos = 0

repetir con i desde 1 hasta 10

El valor i-esimo es un dato válido ?

Verdadero:

ValorMedio = ValorMedio + el *i-esimo* elemento de **MiLista**

CantidadDatosValidos = CantidadDatosValidos + 1

final de repetición

ValorMedio = ValorMedio / CantidadDatosValidos


```
#Defino la lista de valores sobre la cual voy a calcular la media.  
MiLista <- c( 10 , 20 , 30 , 40 , -999 , 60 , 70 , 80 , 90 , 100 )
```

```
#Le doy un valor inicial a la variable donde voy a almacenar la media de la lista  
valorMedio <- 0.0  
CantidadDatosValidos <- 0
```

tamb podía poner #Inicializo mis variables

```
#Armo una repeticion (loop, ciclo) que va tomando uno a uno los valores de MiLista  
# y los va sumando en valorMedio
```

```
for ( valor in MiLista ) {  
  if ( valor != -999 ) {  
    valorMedio <- valorMedio + valor  
    CantidadDatosValidos <- CantidadDatosValidos + 1  
  }  
}
```

Forma de escribir con i (CHEQUEAR POR LAS DUDAS)

for(i in 1:10)

aca usaria el if

!= es como preguntar
¿sos distinto?

```
#Divido valorMedio por la cantidad de datos validos  
valorMedio <- valorMedio / CantidadDatosValidos
```

```
#Muestro el resultado de la media aritmetica por pantalla.  
print( valorMedio)
```

```

#Defino la lista de valores sobre la cual voy a calcular la media.
MiLista <- c( 10 , 20 , 30 , 40 , -999 , 60 , 70 , 80 , 90 , 100 )

#Le doy un valor inicial a la variable donde voy a almacenar la media de la lista
ValorMedio <- 0.0
CantidadDatosValidos <- 0

#Armo una repeticion (loop, ciclo) que va tomando uno a uno los valores de MiLista
# y los va sumando en ValorMedio
for ( valor in MiLista ) {
  if ( valor != -999 ) {
    ValorMedio <- ValorMedio + valor
    CantidadDatosValidos <- CantidadDatosValidos + 1
  }
}

#Divido ValorMedio por la cantidad de datos validos
ValorMedio <- ValorMedio / CantidadDatosValidos

#Muestro el resultado de la media aritmetica por pantalla.
print( ValorMedio)

```

¿Este programa me va a servir para calcular el promedio de 20 valores?

```

#Defino la lista de valores sobre la cual voy a calc
MiLista <- c( 10,20,30,40,50,60,70,80,90,100 )

#Le doy un valor inicial a la variable donde voy a a
ValorMedio <- 0.0

#Armo una repeticion (loop, ciclo) que va tomando un
for ( valor in MiLista ) {
  ValorMedio <- ValorMedio + valor
}

#Divido ValorMedio por 10 de acuerdo a la definicion
ValorMedio <- ValorMedio / 10

#Muestro el resultado de la media aritmetica por pan
print( ValorMedio)

```

El último programa es en algún sentido más general que nuestro programa original.

```

#Defino la lista de valores sobre la cual voy a calcular la media.
MiLista <- c( 10 , 20 , 30 , 40 , -999 , 60 , 70 , 80 , 90 , 100 )

#Le doy un valor inicial a la variable donde voy a almacenar la med
ValorMedio <- 0.0
CantidadDatosValidos <- 0

#Armo una repeticion (loop, ciclo) que va tomando uno a uno los val
# y los va sumando en ValorMedio
for ( valor in MiLista ) {
  if ( valor != -999 ) {
    ValorMedio <- ValorMedio + valor
    CantidadDatosValidos <- CantidadDatosValidos + 1
  }
}

#Divido ValorMedio por la cantidad de datos validos
ValorMedio <- ValorMedio / CantidadDatosValidos

#Muestro el resultado de la media aritmetica por pantalla.
print( ValorMedio)

```

Paso 6, 7 (continuación):

Le falta algo que me diga por pantalla la cantidad de datos validos.
Puede ser que sean 0 y quedaria la indeterminación 0/0.
Podría solucionar con un if

El programa todavía tiene un problema (lógico). ¿Cuál es?

```
#Defino la lista de valores sobre la cual voy a calcular la media.
MiLista <- c( 10 , 20 , 30 , 40 , -999 , 60 , 70 , 80 , 90 , 100 )

#Le doy un valor inicial a la variable donde voy a almacenar la med
ValorMedio <- 0.0
CantidadDatosValidos <- 0

#Armo una repeticion (loop, ciclo) que va tomando uno a uno los val
# y los va sumando en ValorMedio
- for ( valor in MiLista ) {
-   if ( valor != -999 ) {
-     ValorMedio <- ValorMedio + valor
-     CantidadDatosValidos <- CantidadDatosValidos + 1
-   }
- }

#Divido ValorMedio por la cantidad de datos validos
ValorMedio <- ValorMedio / CantidadDatosValidos
|
#Muestro el resultado de la media aritmetica por pantalla.
print( ValorMedio)
```

Paso 8: *Evaluar la eficiencia del código. Consideremos por ejemplo la siguiente variación.*

¿Es correcta desde el punto de vista del resultado? ¿Es más o menos eficiente?

#Defino la lista de valores sobre la cual voy a calcular la media.

MiLista <- c(10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 , 100)

#Le doy un valor inicial a la variable donde voy a almacenar la media de la lista

ValorMedio <- 0.0

#Armo una repeticion (loop, ciclo) que va tomando uno a uno los valores de MiLista y los va sumando en ValorMedio

for (valor in MiLista) {

ValorMedio <- ValorMedio + (valor / 10) nos ahorramos el paso de la división

}

#Muestro el resultado de la media aritmetica por pantalla.

print(ValorMedio)

Existen infinidad de maneras de escribir un programa para obtener un resultado, pero no todas son igualmente eficientes. En general evaluar la eficiencia de un algoritmo no es trivial y para eso existen las herramientas de ***profiling***.

Escribir un programa es mucho más que simplemente expresar en un lenguaje de programación un conjunto de sentencias. Requiere entender, diseñar, testear, corregir, re-diseñar y optimizar algoritmos hasta obtener el resultado deseado de manera robusta y de la forma lo más eficiente posible.

incluir algo erroneo en esa prueba me ayuda a ver si hay cosas para corregir