

Introducción a la Algorítmica y Programación (3300)

Prof. Ariel Ferreira Szpiniak - aferreira@exa.unrc.edu.ar

Departamento de Computación
Facultad de Cs. Exactas, Fco-Qcas y
Naturales

Universidad Nacional de Río Cuarto

Teoría 7 Composición Iterativa (Repetición, Ciclos)



2018 Lic. Ariel Ferreira Szpiniak

Noticias

1 de mayo: Día del Trabajador

Revuelta de Heymarket (1886, Chicago):

Dos recibieron cadena perpetua, uno 15 años de cárcel, cinco pena de muerte. A esos| cinco les espera la horca, pero Lingg madrugó a la muerte haciendo estallar entre sus dientes una cápsula de dinamita. Fischer se viste sin prisa, tarareando "La Marsellesa". Parsons, el agitador que empleaba la palabra como látigo o cuchillo, aprieta las manos de sus compañeros antes de que los guardias se las aten a la espalda. Engel, famoso por la puntería, pide vino de Oporto y hace reír a todos con un chiste. Spies, que tanto ha escrito pintando a la anarquía como la entrada en la vida, se prepara, en silencio, para entrar en la muerte.



“Mártires
de
Chicago”

Noticias

1 de mayo: Día del Trabajador

Chicago está llena de fábricas, hay fábricas hasta en pleno centro de la ciudad, en torno al edificio más alto del mundo. Chicago está llena de fábricas, Chicago está llena de obreros.

Al llegar al barrio de Heymarket, pido a mis amigos que me muestren el lugar donde fueron ahorcados, en 1886, aquellos obreros que el mundo entero saluda cada primero de mayo.

-Ha de ser por aquí- me dicen. Pero nadie sabe.

Ninguna estatua se ha erigido en memoria de los mártires de Chicago en la ciudad de Chicago.

Ni estatua, ni monolito, ni placa de bronce, ni nada.

El primero de mayo es el único día verdaderamente universal de la humanidad entera, el único día donde coinciden todas las historias y todas las geografías, todas las lenguas y las religiones y las culturas del mundo; pero en los Estados Unidos, el primero de mayo es un día cualquiera.

Ese día la gente trabaja normalmente, y nadie, o casi nadie, recuerda que los derechos de la clase obrera no han brotado de la oreja de una cabra, ni de la mano de Dios o del amo (Eduardo Galeano).

Noticias

1 de mayo: Día del Trabajador

Virginia Bolten (nació en San Luis en 1876), una luchadora por los derechos laborales femeninos a fines del siglo XIX, se atrevió a arengar a los obreros el 1 de mayo de 1890 en Rosario, cuando se celebró por primera vez el Día del Trabajador. Ella subió al escenario vestida de negro portando la bandera del anarquismo y denunciando la explotación laboral de las mujeres.

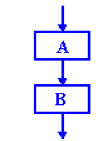
También fundó un periódico, cuyo slogan fue: "Ni Dios, ni patrón, ni marido", en tiempos en que las mujeres no tenían ni voz ni voto.



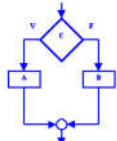
Trabajaba en una Refinería de Azúcar, donde se reveló contra la explotación de las mujeres.

Comenzó a agitar a todo el personal para cambiar las condiciones laborales hasta que finalmente la expulsaron del trabajo.

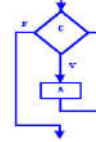
Tipos de Estructuras para componer Algoritmos



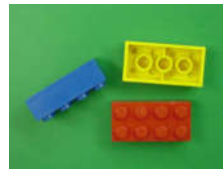
- Composición Secuencial



- Composición Condicional (Decisión, Selección)

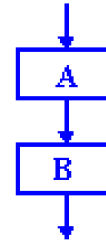


- Composición Iterativa (Repetición)



Composición Secuencial

Es el tipo de composición más simple, está representada por una **sucesión de acciones u operaciones** (por ej. asignaciones), que se realizan una después de la otra, es decir, que el orden de ejecución coincide con el orden físico de aparición de las mismas, es decir, de arriba hacia abajo.

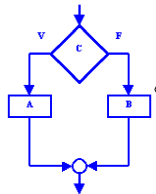


Las cajas A y B pueden ser definidas para ejecutar desde una simple acción hasta un módulo o algoritmo completo.



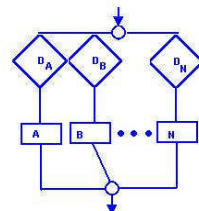
Composición Condicional

En un algoritmo representativo de un problema real es casi imposible que todo sea secuencial. Es necesario tomar **decisiones** en función de los datos del problema. La toma de decisión puede ser entre **dos o más** alternativas.



C es una condición que se evalúa. **A** es la acción que se ejecuta cuando la evaluación de este predicado resulta **verdadera** y **B** es la acción ejecutada cuando es **falsa**.

D_A, **D_B**, ..., **D_N** son decisiones a tomar. **A** es la acción que se ejecuta cuando la **D_A** es verdadera. **B** es la acción ejecutada cuando **D_B** es verdadera. **N** es la acción ejecutada cuando **D_N** es verdadera.



Composición Iterativa

Introducción

Hasta ahora hemos visto que un algoritmo está constituido estructuralmente por una serie de acciones organizadas de manera consecutiva (composición **secuencial**) o selectiva (composición **condicional**).

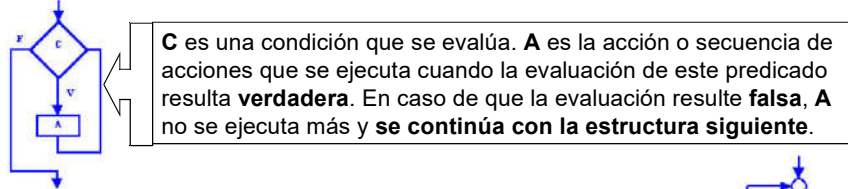
Sin embargo, en muchas ocasiones es necesario realizar las mismas acciones varias veces seguidas de manera consecutiva, ya sea una cantidad predeterminada de veces o no.

La composición **iterativa**, en todas sus alternativas, posibilita la ejecución repetida de un bloque de acciones.

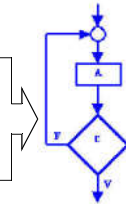


Composición Iterativa

En muchas ocasiones es necesario realizar una a varias acciones de manera repetida, ya sea una cantidad predeterminada de veces o no. La composición iterativa, en todas sus alternativas posibilita la ejecución repetida de un bloque de acciones. Veamos algunos ejemplos.



A es la acción o secuencia de acciones. **C** es una condición que se evalúa. Cuando la evaluación de este predicado resulta **falsa**, **A** vuelve a ejecutarse. Si la evaluación resulta **verdadera**, **A** no se ejecuta más y **se continúa con la estructura siguiente**.



Composición Iterativa

Introducción

- Analizaremos 3 formas diferentes de ejecutar acciones cíclicamente.

Mientras Repetir

Se aplican en casos donde se conoce o no la cantidad de veces que debemos ciclar

Para

Se aplica únicamente en el caso donde se conoce la cantidad de veces que debemos ciclar

Los lenguajes de programación generalmente implementan por lo menos una estructura de cada tipo.



Composición Iterativa

Introducción

- Una **ITERACIÓN**, **CICLO**, **BUCLE**, o **LOOP** es la ejecución de un bloque de acciones, tantas veces como sea necesario.
- La composición iterativa exige para su utilización **MUCHAS PRECAUCIONES**.
- Es preciso asegurarse a priori la coherencia en el encadenamiento sucesivo de las mismas acciones.
- Como veremos, éstas estructuras repetitivas presentan una **condición** que se debe satisfacer para continuar o para terminar. Un error en la formulación de esa condición puede producir a un fenómeno denominado **CICLO**, **BUCLE** o **LOOP INFINITO**.



Composición Iterativa

Introducción

Llamaremos iteración a toda repetición de la ejecución de una acción o de una secuencia de acciones.

Cuando se debe repetir una o varias acciones puede suceder alguna de las siguientes posibilidades:

- No se conoce la cantidad exacta de iteraciones.
- Si se conoce la cantidad exacta de iteraciones.

Para cada uno de los casos existen composiciones iterativas que resultan más convenientes.



Composición Iterativa

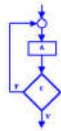
Muchas veces no se conoce de antemano la cantidad de veces que hay que ejecutar una serie de acciones sino que solo se sabe bajo que condiciones se debe seguir ejecutando o cuando se debe finalizar la ejecución. Para ese tipo de situaciones existen tres tipos de estructuras que permiten “ciclar” (ejecutar repetidamente) y parar solo cuando cierta condición es verdadera o es falsa.

Para este caso resultan adecuadas:

Mientras

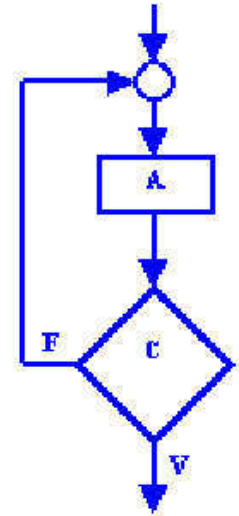


Repetir



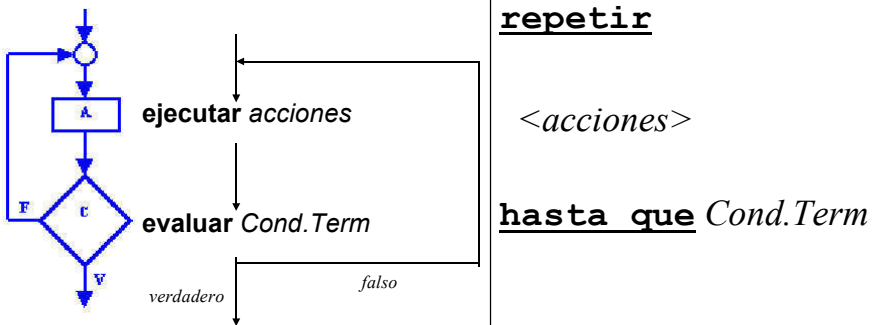
Composición Iterativa Repetir

- El número de ejecuciones de las acciones no se expresa directamente.
- La *condición* expresa lo que se debe cumplir para que *termine* el ciclo. Por ello se denomina *condición de terminación*.
- Las *acciones* se ejecutan *una o más veces*.
- El comportamiento es el siguiente:
 - Se ejecutan las acciones una vez.
 - Luego se evalúa la condición de terminación.
 - Si la condición es falsa se vuelven a ejecutar las acciones.
 - Si la condición es verdadera termina (sale del ciclo).



Composición Iterativa Repetir

En Notación Algorítmica la estructura **Repetir** se escribe de la siguiente manera:



Composición Iterativa Repetir - Ejemplo

Algoritmo EjemploRepetir

Lexico

$i \in \mathbb{Z}$

Inicio

$i \leftarrow 1$

repetir

$i \leftarrow i+1$

hasta que $i > 10$

Salida: i

Fin



Composición Iterativa

Repetir - Ejemplo

Algoritmo EjemploRepetir2

Lexico

$i \in \mathbb{Z}$

Inicio

$i \leftarrow 1$

repetir

Entrada: i

hasta que $i > 0$

Salida: i

Fin



Composición Iterativa

Repetir - Ejemplo

Algoritmo EjemploRepetir3

Lexico

$i \in \mathbb{Z}$

Inicio

$i \leftarrow 1$

repetir

$i \leftarrow i + 1$

hasta que $i \geq 10$

Salida: i

Fin



Composición Iterativa

Repetir - Ejemplo

Problema: obtener dos números enteros, que deben ser distintos entre sí, e informarlos en orden creciente.

Algoritmo OrdenCreciente

Lexico

$num1, num2 \in \mathbb{Z}$ // números a ordenar

Inicio

Entrada: $num1$ // primer número

repetir

Entrada: $num2$ // otro nro diferente al primero

hasta que $num1 < num2$

si $num1 < num2$

entonces

Salida: $num1$ $num2$

sino

Salida: $num2$ $num1$

fsi

Fin



Composición Iterativa

Repetir - Ejemplos

Analizar el siguiente algoritmo y determinar que hace.

Algoritmo BucleSioNo

Léxico

$i \in \mathbb{Z}$

$msg \in \text{Cadena}$

Inicio

$i \leftarrow 1$

repetir

$msg \leftarrow \text{"Hola, soy un bucle"}$

Salida: msg

hasta que $i \neq i$

$msg \leftarrow \text{"¡Terminé!"}$

Salida: msg

Fin

¿cuántas veces
hace el ciclo?



Composición Iterativa

Repetir - Ejemplos

Analizar el siguiente algoritmo y determinar que hace.

Algoritmo BucleSioNo2

Léxico

$i \in \mathbb{Z}$
 $msg \in \text{Cadena}$

Inicio

$i \leftarrow 2$

repetir

$msg \leftarrow \text{"Hola, ¿y yo que soy?"}$

Salida:msg

hasta que $i > 9$

$msg \leftarrow \text{"¡Terminé!"}$

Salida:msg

Fin

¿cuántas veces
hace el ciclo?



2018 Lic. Ariel Ferreira Szpiniak 21

Composición Iterativa

Repetir - Ejemplos

Analizar los siguientes algoritmos y determinar que hacen.

Algoritmo MeCuré?

Lexico

$j \in \mathbb{Z}$
 $msg \in \text{Cadena}$

Inicio

$j \leftarrow 1$

//¿Cuántas veces repito?

repetir

$j \leftarrow j + 3$

hasta que $j > 0$

$msg \leftarrow \text{"¡Terminé!"}$

Salida:msg

Fin

¿cuántas veces hace el ciclo?

Algoritmo KMasCinco

Lexico

$k \in \mathbb{Z}$
 $msg \in \text{Cadena}$

Inicio

$k \leftarrow 1$

repetir

// ¿Salgo o no salgo?

$k \leftarrow k + 1$

hasta que $(k+5) > 20$

$msg \leftarrow \text{"¡Terminé!"}$

Salida:msg

Fin

¿cuántas veces hace el ciclo?



2018 Lic. Ariel Ferreira Szpiniak 22

Composición Iterativa

Repetir - Ejemplos

Analizar los siguientes algoritmos y determinar que hacen.

Algoritmo Mafalda

Lexico

$jj \in \mathbb{Z}$
 $msg \in \text{Cadena}$

Inicio

$jj \leftarrow 0$

//Mi dicho preferido es:

repetir

$jj \leftarrow 20$

hasta que $jj > 1$

$msg \leftarrow \text{"Paren el mundo"}$

$msg \leftarrow msg + \text{" me quiero bajar!"}$

Salida:msg

Fin

¿cuántas veces hace el ciclo?

Algoritmo CuentaRegresiva

Lexico

$i \in \mathbb{Z}$
 $msg \in \text{Cadena}$

Inicio

$i \leftarrow 10$

//Iniciando conteo...

repetir

Salida:i

$i \leftarrow i - 1$

hasta que $i \leq 0$

$msg \leftarrow \text{"¡Despegue!"}$

Salida:msg

Fin

¿cuántas veces hace el ciclo?

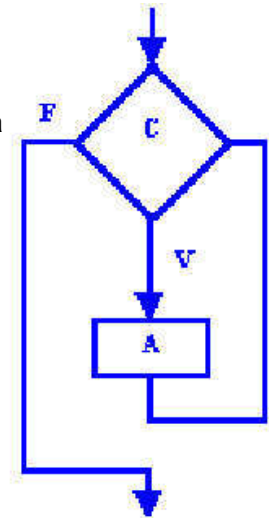


2018 Lic. Ariel Ferreira Szpiniak 23

Composición Iterativa

Mientras

- El número de ejecuciones de las acciones no se expresa directamente.
- La *condición* expresa lo que se debe cumplir para que *continúe* el ciclo. Por ello se denomina *condición de continuación*.
- Las *acciones* se ejecutan *cero, una o más veces*.
- El comportamiento es el siguiente:
 - Se evalúa la condición de continuación.
 - Si la condición es verdadera, entra al ciclo y se ejecutan las acciones una vez.
 - Si la condición es falsa termina, es decir, sale del ciclo.
 - Si la condición es verdadera se vuelven a ejecutar las acciones.



24

Composición Iterativa

Mientras

En Notación Algorítmica la estructura **Mientras** se escribe de la siguiente manera:



Composición Iterativa

Mientras - Ejemplo

Algoritmo EjemploMientras

Lexico

$i \in \mathbb{Z}$

Inicio

$i \leftarrow 1$

mientras $i \leq 10$ hacer

$i \leftarrow i + 1$

fmientras

Salida: i

Fin



Composición Iterativa

Mientras - Ejemplo

Algoritmo EjemploMientras2

Lexico

$i \in \mathbb{Z}$

Inicio

$i \leftarrow 1$

mientras $i \leq 0$ hacer

Entrada: i

fmientras

Salida: i

Fin



Composición Iterativa

Mientras - Ejemplo

Algoritmo EjemploMientras3

Lexico

$i \in \mathbb{Z}$

Inicio

$i \leftarrow 1$

mientras $i < 10$ hacer

$i \leftarrow i + 1$

fmientras

Salida: i

Fin



Composición Iterativa

Mientras - Ejemplo

Problema: obtener dos números enteros, que deben ser distintos entre sí, e informarlos en orden creciente.

Algoritmo OrdenCreciente

Léxico

num1, num2 $\in \mathbb{Z}$ //números a ordenar

Inicio

Entrada:num1 // primer número

Entrada:num2 // otro número diferente al primero

mientras num1=num2 **hacer**

Entrada:num2 //se necesita un número diferente a num1

fmientras

si num1<num2

entonces

Salida:num1 num2

sino

Salida:num2 num1

fsi

Fin



2018 Lic. Ariel Ferreira Szpiniak 29

Composición Iterativa

Mientras - Ejemplos

Analizar el siguiente algoritmo y determinar que hace. Compare semejanzas y diferencias con los ejemplos dados para el **Repetir**.

Algoritmo BucleSioNo

Léxico

$i \in \mathbb{Z}$

msg \in Cadena

Inicio

$i \leftarrow 1$

mientras i=i **hacer**

msg \leftarrow "Hola, soy un bucle"

Salida:msg

fmientras

msg \leftarrow ";Terminé!"

Salida:msg

Fin



¿cuántas veces
hace el ciclo?

2018 Lic. Ariel Ferreira Szpiniak 30

Composición Iterativa

Mientras - Ejemplos

Analizar el siguiente algoritmo y determinar que hace. Compare semejanzas y diferencias con los ejemplos dados para el **Repetir**.

Algoritmo BucleSioNo2

Léxico

$i \in \mathbb{Z}$

msg \in Cadena

Inicio

$i \leftarrow 2$

mientras i<=9 **hacer**

msg \leftarrow "Hola, ¿y yo que soy?"

Salida:msg

fmientras

msg \leftarrow ";Terminé!"

Salida:msg

Fin



2018 Lic. Ariel Ferreira Szpiniak 31

¿cuántas veces
hace el ciclo?

Composición Iterativa

Mientras - Ejemplos

Analizar los siguientes algoritmos y determinar que hacen. Compare semejanzas y diferencias con los ejemplos dados para el **Repetir**.

Algoritmo MeCuré?

Lexico

$j \in \mathbb{Z}$

msg \in Cadena

Inicio

$j \leftarrow 1$

//¿Cuántas veces ciclo?

mientras j<=0 **hacer**

$j \leftarrow j + 3$

fmientras

msg \leftarrow ";Terminé!"

Salida:msg

Fin



¿cuántas veces hace el ciclo?

Algoritmo KMasCinco

Lexico

$k \in \mathbb{Z}$

msg \in Cadena

Inicio

$k \leftarrow 1$

mientras (k+5)>20 **hacer**

// ¿Salgo o no salgo?

$k \leftarrow k + 1$

fmientras

msg \leftarrow ";Terminé!"

Salida:msg

Fin

¿cuántas veces hace el ciclo?

2018 Lic. Ariel Ferreira Szpiniak 32

Composición Iterativa

Mientras - Ejemplos

Analizar los siguientes algoritmos y determinar que hacen. Compare semejanzas y diferencias con los ejemplos dados para el **Repetir**.

Algoritmo Mafalda

Lexico

$jj \in \mathbb{Z}$
 $msg \in \text{Cadena}$

Inicio

```
jj ← 0
//Mi dicho preferido es:
mientras jj<1 hacer
  jj ← 20
hasta que jj>=1
msg ← "Paren el mundo"
msg ← msg+" me quiero bajar!"
Salida:msg
```

Fin

¿cuántas veces hace el ciclo?

Algoritmo CuentaRegresiva

Lexico

$i \in \mathbb{Z}$
 $msg \in \text{Cadena}$

Inicio

```
i ← 10
//Iniciando conteo...
mientras i>=1 hacer
  Salida:i
  i ← i - 1
fmientras
msg ← "Despegue!"
Salida:msg
```

Fin

¿cuántas veces hace el ciclo?

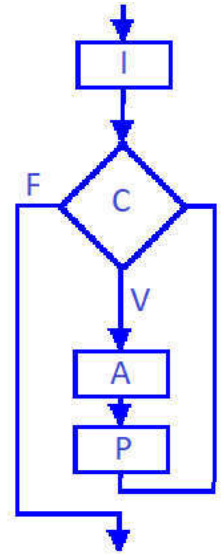


2018 Lic. Ariel Ferreira Szpiniak 33

Composición Iterativa

Para

- El número de ejecuciones de las acciones se expresa directamente. Las *acciones* se ejecutan *cero, una o más veces*.
- El *paso* puede ser positivo o negativo.
- Cualquier ciclo **para** debe contener una **variable de control** (i, j, k , los nombres más comunes). Dicha variable toma el valor indicado en la **inicialización**, se utiliza en la condición de continuación (**Cond.Cont**) para indicar hasta cuando debe ciclar, y en el **paso** (incremento o decremento) para saber que cambios sufre en cada ciclo.



2018 Lic. Ariel Ferreira Szpiniak 34

Composición Iterativa

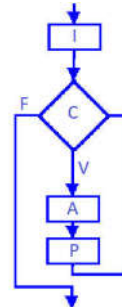
Para

para (*inicialización, Cond.Cont, paso*) **hacer**

<acciones>

fpara

- Se inicializa la variable de control (*inicialización*).
- Se evalúa la condición de continuación (**Cond.Cont**):
 - Si la condición de continuación es Verdadera, se ejecutan las acciones (*acciones*). Luego se incrementa o decrementa la variable de control (*paso*). Se vuelve a evaluar la condición de continuación.
 - Si la **Cond.Cont** es Falsa, sale del ciclo.



2018 Lic. Ariel Ferreira Szpiniak 35

Composición Iterativa

Para - Ejemplo

Problema: realizar y mostrar la sumatoria de los n primeros números enteros positivos.

Algoritmo SumatoriaEnteros

Lexico

$n \in \mathbb{Z}$ //cantidad de números
 $i \in \mathbb{Z}$ //variable de control del para
 $s \in \mathbb{Z}$ //sumatoria de los n primeros números enteros

Inicio

```
Entrada:n
s ← 0
para (i←0, i<=n, i←i+1) hacer
  s ← s + i
fpara
Salida:s
```

Fin



2018 Lic. Ariel Ferreira Szpiniak 36

Composición Iterativa

Para - Ejemplo

Analizar los siguientes algoritmos y determinar que hacen.

```
Algoritmo QueInformo?  
Lexico  
i ∈ Z  
Inicio  
  para (i←1, i≤10, i←i+1) hacer  
    Salida:i  
  fpara  
Fin
```

```
Algoritmo QueInformo2?  
Lexico  
j ∈ Z  
Inicio  
  para (j←2, j≤80, j←j+2) hacer  
    Salida:j  
  fpara  
Fin
```



Composición Iterativa

Para - Ejemplo

Analizar los siguientes algoritmos y determinar que hacen.

```
Algoritmo CuentaRegresiva  
Lexico  
i ∈ Z  
msg ∈ Cadena  
Inicio  
  //Iniciando conteo...  
  para (i←10, i≤1, i←i-1) hacer  
    Salida:i  
  fpara  
  msg ← ";Despegue!"  
  Salida:msg  
Fin
```

```
Algoritmo CuentaRegresiva2  
Lexico  
i ∈ Z  
msg ∈ Cadena  
Inicio  
  //Iniciando conteo...  
  para (i←10, i≤0, i←i-1) hacer  
    Salida:i  
  fpara  
  msg ← ";Despegue!"  
  Salida:msg  
Fin
```



Composición Iterativa

Para - Ejemplo

Analizar los siguientes algoritmos y determinar que hacen.

```
Algoritmo ParesDesde0  
Lexico  
j ∈ Z  
Inicio  
  para (j←0, j≤80, j←j+2) hacer  
    Salida:j  
  fpara  
Fin
```

```
Algoritmo BucleSioNo?  
Lexico  
k ∈ Z  
msg ∈ Cadena  
Inicio  
  msg ← ";Soy un bucle?"  
  para (k←1, k≤10, k←k-1) hacer  
    Salida:msg  
  fpara  
Fin
```



Ejercicios

Ahora que ya conocemos las estructuras iterativas vamos a resolver los siguientes problemas, validando que los datos ingresados sean correctos.

- Se tiene que pintar de color amarillo una señal de peligro contenida en un cartel para la vía pública.
- Desarrollar un juego que siga los siguientes pasos:
 - 1) Solicite un número.
 - 2) A continuación le sume 1 unidad al número.
 - 3) Informe el nuevo número y el mensaje TE GANE!!!
 - 4) A continuación consulte si quiere jugar otra vez (S/N).
 - 5) Si el usuario responde `S` vuelve al paso 1, sino termina.

Nota: use la estructura iterativa que considere más adecuada.



Composición Iterativa

C - Mientras

La estructura **Mientras**, en **C**, se escribe de la siguiente manera:

Notación Algorítmica

mientras *Cond.* *Cont* **hacer**

<acciones>

fmientras

C

while *Cond.* *Cont* {

<acciones>

}



Composición Iterativa

C - Mientras - Ejemplo

Problema: obtener dos números enteros, que deben ser distintos entre sí, e informarlos en orden creciente.

Algoritmo OrdenCreciente

Léxico

num1, num2 ∈ Z // números a ordenar

Inicio

Entrada:num1 // primer número

Entrada:num2 // otro número diferente al primero

mientras num1=num2 **hacer**

Entrada:num2 // se necesita un número diferente a num1

fmientras

si num1<num2

entonces

Salida:num1 num2

sino

Salida:num2 num1

fsi

Fin



Composición Iterativa

C - Mientras - Ejemplo

```
#include <stdio.h>
int num1, num2; //números a ordenar
int main(){
    printf("Ingrese el primer número ");
    scanf("%d",&num1);
    printf("Ingrese otro diferente al primero ");
    scanf("%d",&num2);
    while (num1==num2){
        printf("Se necesita un número diferente de %d \n", num1);
        printf("Vuelva a intentarlo \n");
        scanf("%d",&num2);
    }
    if (num1<num2){
        printf("Primer numero: %d ",num1);
        printf("Segundo número: %d ",num2);
    }
    else{
        printf("Segundo numero: %d ",num2);
        printf("Primer numero: %d ",num1);
    }
    return 0;
}
```



Composición Iterativa

C - Repetir

La estructura **Repetir**, en **C**, se escribe de la siguiente manera:

Notación Algorítmica

repetir

<acciones>

hasta que *Cond.Term*

C

do {

<acciones>

} while !(Cond.Term);



Composición Iterativa

C - Repetir - Ejemplo

Problema: obtener dos números enteros, que deben ser distintos entre sí, e informarlos en orden creciente.

Algoritmo OrdenCreciente

Lexico

num1, num2 $\in \mathbb{Z}$ //números a ordenar

Inicio

Entrada:num1 // primer número

repetir

Entrada:num2 // otro nro diferente al primero

hasta que num1<>num2

si num1<num2

entonces

Salida:num1 num2

sino

Salida:num2 num1

fsi

Fin



2018 Lic. Ariel Ferreira Szpiniak 45

Composición Iterativa

C - Repetir - Ejemplo

```
#include <stdio.h>
int num1, num2; //números a ordenar
int main(){
    printf("Ingrese el primer número ");
    scanf("%d",&num1);
    do {
        printf("Ingrese otro diferente al primero ");
        scanf("%d",&num2);
    } while (!(num1!=num2));
    if (num1<num2){
        printf("Primer numero: %d ",num1);
        printf("Segundo número: %d ",num2);
    }
    else{
        printf("Segundo numero: %d ",num2);
        printf("Primer numero: %d ",num1);
    }
    return 0;
}
```



2018 Lic. Ariel Ferreira Szpiniak 46

Composición Iterativa

C - Repetir - Ejemplo

```
#include <stdio.h>
int num1, num2; //números a ordenar
int main(){
    printf("Ingrese el primer número ");
    scanf("%d",&num1);
    do {
        printf("Ingrese otro diferente al primero ");
        scanf("%d",&num2);
    } while (num1==num2);
    if (num1<num2){
        printf("Primer numero: %d ",num1);
        printf("Segundo número: %d ",num2);
    }
    else{
        printf("Segundo numero: %d ",num2);
        printf("Primer numero: %d ",num1);
    }
    return 0;
}
```



2018 Lic. Ariel Ferreira Szpiniak 47

Composición Iterativa

C - Para

La estructura **Para**, en **C**, se escribe de la siguiente manera:

Notación Algorítmica

para (*inicialización, Cond.Cont, paso*) hacer

<acciones>

fpara

C

```
for (inicialización; Cond.Cont; paso) {
    <acciones>
}
```



2018 Lic. Ariel Ferreira Szpiniak 48

Composición Iterativa

C - Para - Ejemplo

Problema: realizar y mostrar la sumatoria de los n primeros números enteros positivos.

Algoritmo SumatoriaEnteros

Lexico

$n \in \mathbb{Z}$ //cantidad de números
 $i \in \mathbb{Z}$ //variable de control del para
 $s \in \mathbb{Z}$ //sumatoria de los n primeros números enteros

Inicio

Entrada: n

$s \leftarrow 0$

para ($i \leftarrow 0$, $i \leq n$, $i \leftarrow i+1$) **hacer**

$s \leftarrow s + i$

fpara

Salida: s

Fin



Composición Iterativa

C - Para - Ejemplo

```
#include <stdio.h>
int n; //cantidad de numeros
int i; // variable de control del para
int s; // sumatoria de los n primeros numeros
int main(){
    printf("Ingrese la cantidad de numeros a sumar: ");
    scanf("%d",&n);
    s = 0;
    for (i=0; i<=n; i=i+1){
        s = s + i;
    }
    printf("La suma de los %d primeros numeros es: %d \n", n, s);
    return 0;
}
```



Composición Iterativa

C - Para - Ejemplo

```
#include <stdio.h>
int n; //cantidad de numeros
int i; // variable de control del para
int s; // sumatoria de los n primeros numeros
int main(){
    printf("Ingrese la cantidad de numeros a sumar: ");
    scanf("%d",&n);
    s = 0;
    for (i=0; i<=n; i++){
        s = s + i;
    }
    printf("La suma de los %d primeros numeros es: %d \n", n, s);
    return 0;
}
```



Composición Iterativa

Equivalencias

repetir <acciones> hasta que Cond.Term	≡	<acciones> mientras no(Cond.Term) hacer <acciones> fmientras
para ($i \leftarrow a$, $i \leq n$, $i \leftarrow i+k$) hacer <acciones> fpara	≡	$i \leftarrow a$ mientras $i \leq n$ hacer <acciones> $i \leftarrow i+k$ fmientras
mientras Cond.Cont hacer <acciones> fmientras	≡	si Cond.Cont entonces repetir <acciones> hasta que no(Cond.Cont) fsi



Composición Iterativa

Riesgos - Ejemplos

Escribir un algoritmo que dado un **cesto** de papas permita **pelar** un número suficiente de **papas** para hacer un rico puré. Debe tenerse en cuenta que el cesto de papas puede estar vacío en un momento dado y que es posible volver a llenarlo.

Versión 1
mientras cesto no vacío hacer
 pelar una papa
 pelar una papa
fmientras

Versión 2
repetir
 pelar una papa
 pelar una papa
hasta que cesto no vacío

Versión 3
repetir
 pelar una papa
 pelar una papa
hasta que cesto vacío

Versión 4
mientras número de papas insuficiente hacer
 si cesto vacío
 entonces
 llenar cesto
 fsi
 pelar una papa
fmientras

Versión 5
mientras número de papas insuficiente hacer
 si cesto no vacío
 entonces
 pelar una papa
 sino
 llenar cesto
 fsi
fmientras



2018 Lic. Ariel Ferreira Szpiniak 53

Composición Iterativa

Riesgos - Ejemplos

Versión 6
mientras nro de papas insuficiente hacer
 si cesto no vacío
 entonces
 pelar una papa
 fsi
fmientras

Versión 8
mientras cesto lleno hacer
 si nro de papas insuficiente
 entonces
 pelar una papa
 sino
 llenar cesto
 pelar una papa
 fsi
fmientras

Versión 7
repetir
 si cesto no vacío
 entonces
 pelar una papa
 fsi
hasta que nro de papas suficiente

Versión 9
mientras nro de papas insuficiente o cesto lleno hacer
 pelar una papa
fmientras

Versión 10
repetir
 pelar una papa
hasta que nro de papas insuficiente o cesto vacío



2018 Lic. Ariel Ferreira Szpiniak 54

Bibliografía

- Biondi, J. y Clavel, G. “Introducción a la Programación. Tomo 1: Algorítmica y Lenguajes”. (pags. 70 - 122)
- Wirth, N. “Introducción a la Programación Sistemática”. Ateneo. (pags. 27 - 44)
- Quetglás, Toledo, Cerverón. “Fundamentos de Informática y Programación”. Capítulo 3. <http://robotica.uv.es/Libro/Indice.html>
 - Estructuras Repetitivas (pags 98 – 110)
- Watt, David: Programming Language Concepts and Paradigms, Prentice-Hall International Series in Computer Science (1990). Cap 10: The Imperative Programming Paradigm.
- Scholl, P. y Peyrin, J.-P. “Esquemas Algorítmicos Fundamentales: Secuencias e iteración”. (pags 107 – 128). Aparecen los ciclos en forma simultánea con el concepto de secuencias (este concepto lo veremos más adelante).



2018 Lic. Ariel Ferreira Szpiniak

Citar/Atribuir: Ferreira, Szpiniak, A. (2018). Teoría 7: Composición Iterativa. (Repetición, Ciclos). Introducción a la Algorítmica y Programación (3300). Departamento de Computación. Facultad de Cs. Exactas, Fco-Qcas y Naturales. Universidad Nacional de Río Cuarto.

Usted es libre para:

Compartir: copiar y redistribuir el material en cualquier medio o formato.
Adaptar: remezclar, transformar y crear a partir del material.

El licenciente no puede revocar estas libertades en tanto usted siga los términos de la licencia.

Bajo los siguientes términos:



Atribución: Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciente.



Compartir Igual: Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted podrá distribuir su contribución siempre que utilice la misma licencia que la obra original.

<https://creativecommons.org/licenses/by-sa/2.5/ar/>



Introducción a la Algorítmica y Programación (3300)

Prof. Ariel Ferreira Szpiniak - aferreira@exa.unrc.edu.ar
Departamento de Computación
Facultad de Cs. Exactas, Fco-Qcas y Naturales
Universidad Nacional de Río Cuarto

Teoría 7 - Anexo

Ejemplo de Modularización usando Acciones



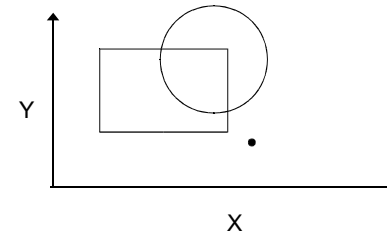
2018 Lic. Ariel Ferreira Szpiniak

Ejemplo

Dados un círculo, un rectángulo y un punto, determinar la posición del punto respecto a las dos figuras.

Análisis

Para comprender el problema debo contextualizarlo en el Plano, esto es:



¿cómo los represento?
el cuadrado con 2 puntos
el círculo con el punto central y radio

- ¿Qué resultados son posibles?

Las 4 posibilidades de posición del punto respecto de las 2 figuras.

- ¿Cómo puedo obtener los posibles resultados?

Puedo averiguar la pertenencia (estar dentro de) a cada una de las figuras y luego determinar la solución. Para el cuadrado puedo lograrlo comparando con los ejes de abscisas y ordenadas. Para el caso del punto con la distancia del mismo al centro respecto del radio.



2018 Lic. Ariel Ferreira Szpiniak 58

Modularizar

Acciones

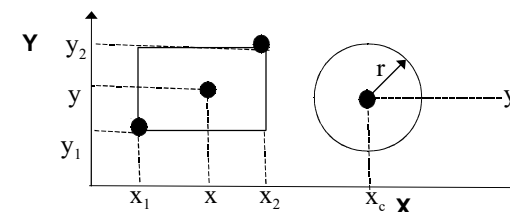
- Obtener los Datos
- Pertenencia al Cuadrado
- Pertenencia al Círculo



2018 Lic. Ariel Ferreira Szpiniak 59

Modularizar

Obtener los Datos: Modularizado como tres Acciones.



Como acotación del problema asumimos que los puntos que representan el cuadrado son el inferior izquierdo y el superior derecho (en ese orden), y que el radio es positivo.



2018 Lic. Ariel Ferreira Szpiniak 60

Obtener los Datos

Parámetros formales de salida

Acción ObtenerPunto(resultado xp, yp ∈ R)

Inicio

Entrada: xp yp

Facción

Parámetros formales de salida

Acción ObtenerCuadrado(resultado x1,y1,x2,y2 ∈ R)

Inicio

repetir

ObtenerPunto(x1,y1) ← Parámetros actuales

// Coordenada del punto sup der del cuadrado

ObtenerPunto(x2,y2) ← Parámetros actuales

hasta que (x1 < x2 y y1 < y2)

Parámetros formales de salida

Facción

Acción ObtenerCírculo(resultado xc,yc,r ∈ R)

Inicio

ObtenerPunto(xc,yc) ← Parámetros actuales

repetir

Entrada: r

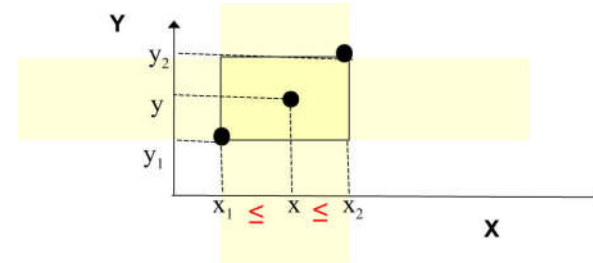
hasta que r > 0

Facción



Modularizar

• **Pertenece Rectángulo:** Modularizado como una Acción.

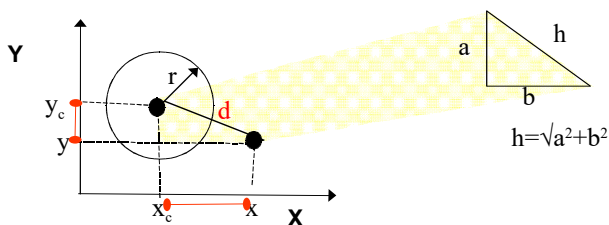


Para que el punto este dentro del rectángulo, debe estar en la intersección de las dos franjas, es decir, debe cumplir con las 2 condiciones (comparaciones) de su x e y.



Modularizar

• **Pertenece a Círculo:** Modularizado como una Acción.



Para determinar la pertenencia del punto al círculo, podemos comprobar si la distancia del mismo a centro es menor o igual que el radio. Para ello podemos determinar la distancia con Pitágoras, donde la distancia a averiguar es la hipotenusa.



Parámetro formal de salida
Parámetros formales de entrada

• **Pertenece Cuadrado:**

Acción PertAlCuadrado(dato x,y,x1,y1,x2,y2 ∈ R, resultado perteneceCud ∈ Lógico)

Inicio

perteneceCud <- (x1 ≤ x ≤ x2) y (y1 ≤ y ≤ y2)

Facción

Parámetro formal de salida

Parámetros formales de entrada

• **Pertenece Círculo:**

Acción PertAlCírculo(dato x,y,xc,yc,r ∈ R, resultado perteneceCirc ∈ Lógico)

Inicio

perteneceCirc <- (r ≥ √((x - xc)² + (y - yc)²))

Facción



Algoritmo FigurasYPunto

Léxico

$a1, a2, b1, b2, c1, c2, d1, d2, \text{radio} \in \mathbb{R}$ {Datos entrada}

$\text{estaCuad}, \text{estaCirc} \in \text{Logico}$ {Soluciones Parciales}

Acción ObtenerPunto(resultado $x_p, y_p \in \mathbb{R}$)

Acción ObtenerCuadrado(resultado $x1, y1, x2, y2 \in \mathbb{R}$)

Acción ObtenerCírculo(resultado $xc, yc, r \in \mathbb{R}$)

Acción PertAlCuadrado(dato $x, y, x1, y1, x2, y2 \in \mathbb{R}$, resultado
perteneceCuad $\in \text{Lógico}$)

Acción PertAlCírculo(dato $x, y, x1, y1, x2, y2 \in \mathbb{R}$, resultado
perteneceCirc $\in \text{Lógico}$)

Inicio

ObtenerCuadrado($b1, b2, c1, c2$)

ObtenerCírculo($d1, d2, \text{radio}$)

Entrada: $a1, a2$

PertAlCuadrado($a1, a2, b1, b2, c1, c2, \text{estaCuad}$)

PertAlCírculo($a1, a2, d1, d2, \text{radio}, \text{estaCirc}$)

según

estaCuad y estaCirc : Escribir('Está en ambos')

$\text{no}(\text{estaCuad})$ y estaCirc : Escribir('Está sólo en el Círculo')

estaCuad y $\text{no}(\text{estaCirc})$: Escribir('Está sólo en el Cuadrado')

$\text{no}(\text{estaCuad})$ y $\text{no}(\text{estaCirc})$: Escribir('Está fuera de ambos')

fsegún

Fin

