

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. [Puede saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web).

[< Anterior](#)[Posterior >](#)

Buscar...

3 - Variables

👤 Por: Nacho Cabanes

📅 Actualizado: 30-12-2019 15:09

🕒 Tiempo de lectura estimado: 5 min.

📖 Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (6:55 min):

EN LOS FOROS

17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)

17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)

16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)

16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)

16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)

[\(Anteriores...\)](#)

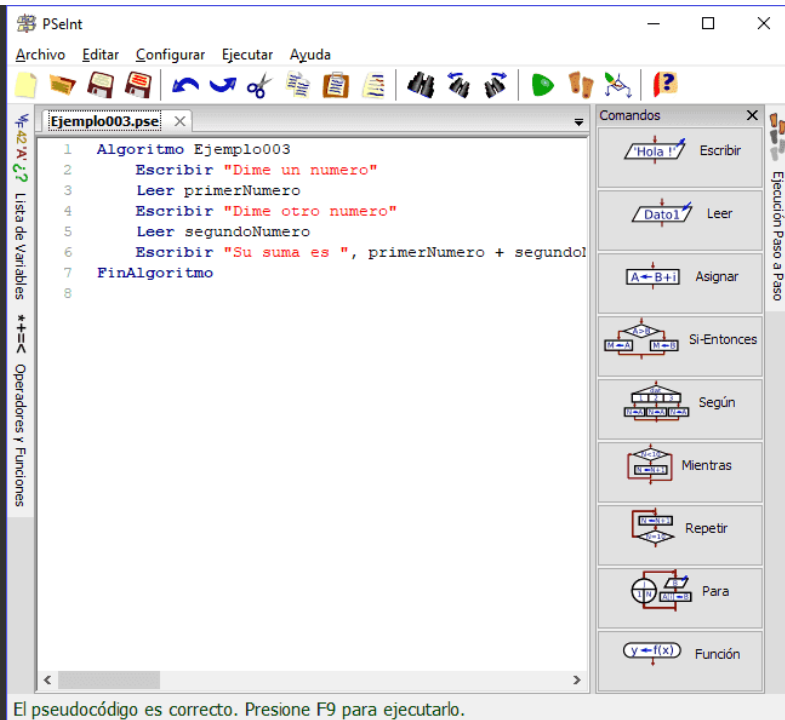
Pseudocódigo

3 - Variables

Vamos a hacer algo un poco más complejo: vamos a sumar dos números que no estén prefijados dentro del programa, sino que deberá teclear el usuario.

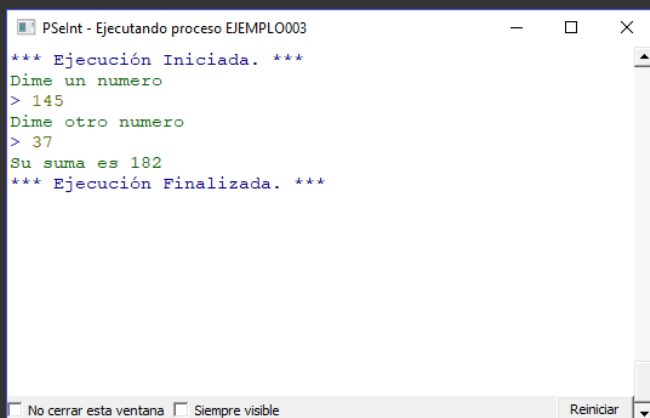
Para eso, usaremos la orden "Leer", que nos permite obtener un dato que el usuario teclee y dejarlo guardado para utilizarlo después. Deberemos dar un nombre temporal a estos datos que leemos del usuario. Parece razonable que el primer número que teclee el usuario se llame algo como "primerNumero", y el segundo sea algo como "segundoNumero". El resultado que queremos obtener será la suma de ese primer número y ese segundo número, así que nuestro programa podría quedar así:

```
Algoritmo Ejemplo003
  Escribir "Dime un numero"
  Leer primerNumero
  Escribir "Dime otro numero"
  Leer segundoNumero
  Escribir "Su suma es ", primerNumero + segundoNumero
FinAlgoritmo
```



El pseudocódigo es correcto. Presione F9 para ejecutarlo.

El resultado de este programa debería ser algo como (dependiendo de los datos que se introduzcan):



Esas dos palabras, "primerNumero" y "segundoNumero" representan a números que no tienen un valor prefijado. Eso es lo que llamaremos "variables". En las variantes de pseudocódigo que emplean en muchas universidades y centros de estudio (ya habíamos avisado de que no existe ningún estándar totalmente asentado) debemos "declarar las variables" antes de usarlas: decir qué tipo de dato es el que queremos representar con ese nombre. En nuestro ejemplo anterior se trataba de dos números enteros, así que el programa sería:

```

Algoritmo Ejemplo003b
  Definir primerNumero como Entero
  Definir segundoNumero como Entero

  Escribir "Dime un numero"
  Leer primerNumero
  Escribir "Dime otro numero"
  Leer segundoNumero
  Escribir "Su suma es ", primerNumero + segundoNumero
FinAlgoritmo

```

Esto se debe a que en la mayoría de lenguajes de programación "reales" es necesario detallar qué tipo de datos queremos guardar en cada variable, para que la herramienta de programación sepa exactamente qué cantidad de memoria será necesario reservar para guardar ese dato.

En el caso de PSeInt, se puede escoger entre distintas sintaxis. La sintaxis que viene "por defecto" (si no se cambia nada) es la llamada "flexible", que permite que no se declaren las variables antes de usarlas, pero existen otras variantes de sintaxis, empleadas por ciertas universidades, en las que sí puede ser obligatorio hacerlo.

Por otra parte, no siempre queremos que el valor de una variable lo introduzca el usuario. Habrá veces que seamos nosotros mismos los que demos el valor inicial a una variable desde nuestro programa, bien para usarlo en una serie de cálculos posteriores, o bien por legibilidad (es más fácil entender algo como "longitudCircunferencia = 2 * pi * radio" que algo como "longitudCircunferencia = 6.28 * radio").

La forma de dar un valor a una variable es con la secuencia de símbolos "<=":

```
radio <- 3
longitudCircunferencia <- 2 * pi * radio
```

(esos símbolos representan una flecha, para indicar que el valor 3 se va a guardar dentro del espacio de memoria reservado para la variable llamada "radio").

¿Y qué ocurre si usamos una variable sin haberle dado valor? Esto sucede a veces por despiste, si tectreamos mal el nombre de una variable, como en este fragmento de programa:

```
primerNumero <- 2
Escribir primerNmero
```

Si lees ese fragmento con cuidado, verás que el nombre de la variable que aparece en la segunda línea es incorrecto, falta la letra "u". ¿Qué sucede en ese caso? En algunos lenguajes (pocos, afortunadamente) se da por sentado que es una variable nueva, y se le da el valor 0; en el caso de PseInt, igual que en la mayoría de lenguajes actuales, obtendremos un mensaje de error que nos dirá que estamos usando una variable que no tenía valor.

Ejercicio de repaso propuesto 3.1: Crea un programa que escriba el resultado de multiplicar los dos números que introduzca el usuario

Ejercicio de repaso propuesto 3.2: Crea un programa que calcule la superficie de un rectángulo a partir de su base y su altura, y que después muestre el valor de dicha superficie.

Ejercicio de repaso propuesto 3.3: Crea un programa que calcule la superficie de un círculo a partir de su radio (la fórmula es " πr^2 ") y que después muestre el valor de dicha superficie (pista: para calcular el cuadrado de un número puedes usar la operación "potencia": x^2 , o bien multiplicar el número por él mismo: $x \times x$).

113300 visitas desde el 15-04-2019

[< Anterior](#)

[Posterior >](#)



APRENDEAPROGRAMAR.COM

Cursos gratuitos de programación de ordenadores, en español

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. [Puede saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web).

[< Anterior](#)[Posterior >](#)

4 - Comprobando condiciones (1: Si)

Por: Nacho Cabanes

Actualizado: 26-04-2019 07:59

Tiempo de lectura estimado: 6 min.

Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (7:11 min):

EN LOS FOROS

17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)

17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)

16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)

16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)

16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)

[\(Anteriores...\)](#)

Pseudocódigo

4 - Comprobando Condiciones (1: Si)

En casi cualquier secuencia de instrucciones para un ordenador, será vital poder comprobar si se cumple alguna condición. Una primera forma básica de comprobar condiciones es con la orden "SI". Su uso básico sería

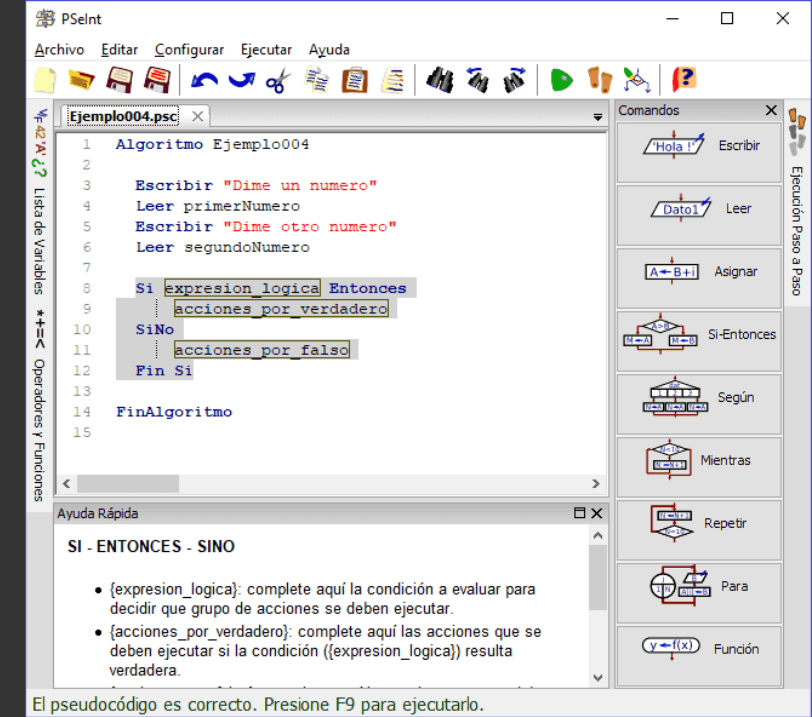
```
Si condicion Entonces
    pasos_a_dar_si_es_verdadero
SiNo
    pasos_a_dar_si_es_falso
FinSi
```

El bloque "SiNo" es opcional: podemos optar por no indicar lo que queremos que se haga cuando no se cumple la condición.

Para ayudarnos a planificar el comportamiento de una secuencia de instrucciones, se suele usar como ayuda los llamados "diagramas de flujo". En estos diagramas, una condición se representa como un rombo, del que salen dos flechas: una para la secuencia de acciones a realizar si se cumple la condición y otra para cuando no se cumple:



Así, si en PSeInt hacemos clic en el icono de la parte derecha que representa la condición SI-ENTONCES, aparece un esqueleto de programa casi completo, para que hagamos los cambios que nos interesen:



Sólo tenemos que escribir la condición que realmente nos interesa, y la serie de pasos que se deben dar si se cumple y si no se cumple, de modo que nuestro programa podría quedar así:

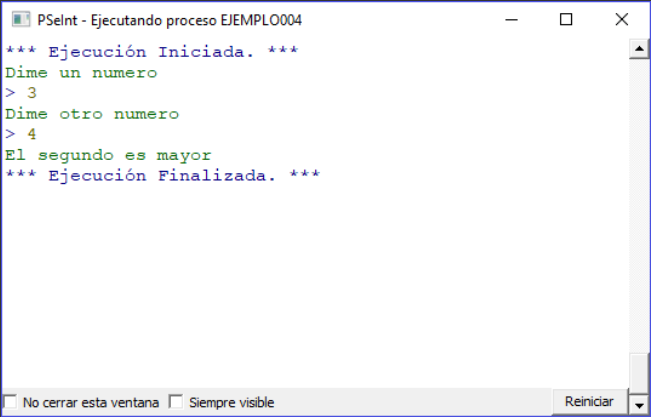
```
Algoritmo Ejemplo004

    Escribir "Dime un numero"
    Leer primerNumero
    Escribir "Dime otro numero"
    Leer segundoNumero

    Si primerNumero > segundoNumero Entonces
        Escribir "El primero es mayor"
    SiNo
        Escribir "El segundo es mayor"
    FinSi

FinAlgoritmo
```

Su resultado sería éste:



(Nota: si lo piensas con un poco de detenimiento -o lo pruebas-, te darás cuenta de que ese planteamiento no es correcto del todo: si introduces dos números iguales, te dirá que el segundo es el mayor; lo solucionaremos un poco más adelante, en el ejemplo 4c).

En ese ejemplo hemos comparado si un valor es mayor que el otro (>). Los operadores de comparación que tenemos disponibles son:

Operador relacional	Significado	Ejemplo
>	Mayor que	3>2
<	Menor que	2<3
=	Igual que	3=3
<=	Menor o igual que	5 <= 5

Operador relacional	Significado	Ejemplo
>=	Mayor o igual que	6 >= 5
<>	Distinto de	6 <> 5

Ejercicio de repaso propuesto 4.1: Crea un programa que pida dos números al usuario y responda si son iguales o no lo son.

También es posible comprobar varias condiciones a la vez, para hacer construcciones más complejas, como: "si a es mayor que b y b es mayor que c", o como "si a es igual a 1 o b es igual a 1 ". Los operadores lógicos que de los que disponemos son:

Operador lógico	Significado	Ejemplo
& ó Y	Conjunción (y).	(7>4) & (2=1) //falso
ó O	Disyunción (o).	(1=1 2=1) //verdadero
~ ó NO	Negación (no).	~(2<5) //falso

Un ejemplo de su uso sería:

```
Algoritmo Ejemplo004b

  Escribir "Dime un numero"
  Leer primerNumero

  Si primerNumero > 0 o primerNumero = 0 Entonces
    Escribir "Es mayor o igual que cero"
  Sino
    Escribir "Es negativo"
  FinSi

FinAlgoritmo
```

Además, podemos encadenar varias condiciones una detrás de otra. Como hemos comentado, el ejemplo 4 fallaba si los dos números son iguales. Por eso, un planteamiento más detallado (y más correcto) sería emplear dos comprobaciones "si", de la siguiente manera:

```
Algoritmo Ejemplo004c

  Escribir "Dime un numero"
  Leer primerNumero
  Escribir "Dime otro numero"
  Leer segundoNumero

  Si primerNumero > segundoNumero Entonces
    Escribir "El primero es mayor"
  Sino
    Si primerNumero < segundoNumero Entonces
      Escribir "El segundo es mayor"
    Sino
      Escribir "Son iguales"
    FinSi
  FinSi

FinAlgoritmo
```

Ejercicio de repaso propuesto 4.2: Haz una variante del ejemplo 004b, usando el operador ">=".

Ejercicio de repaso propuesto 4.3: Crea un programa que pida un número al usuario y diga si es positivo, negativo o cero.

Ejercicio de repaso propuesto 4.4: Haz un programa que pida al usuario dos números y diga cuántos de ellos son positivos.

Ejercicio de repaso propuesto 4.5: Crea un programa que pida al usuario dos números y muestre su división si el segundo no es cero, o un mensaje de aviso en caso contrario.

Ejercicio de repaso propuesto 4.6: Prepara un programa que pida al usuario tres números y diga cuál es el mayor de los tres.

[< Anterior](#)

[Posterior >](#)



APRENDEAPROGRAMAR.COM

Cursos gratuitos de programacion de ordenadores, en español

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta](#)

[Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. ✕
Puede [saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web).

[< Anterior](#)[Posterior >](#)

Buscar...

5 - Comprobando condiciones (2: Segun)

👤 Por: Nacho Cabanes

📅 Actualizado: 16-04-2019 08:25

🕒 Tiempo de lectura estimado: 4.5 min.

📖 Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (3:28 min):

EN LOS FOROS

17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)

17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)

16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)

16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)

16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)

[\(Anteriores...\)](#)

Pseudocódigo

5 - Comprobando Condiciones (2: Segun)

Es frecuente tener que comprobar más de una condición a la vez, o bien varias condiciones consecutivas. En concreto, un caso especialmente habitual es el de que una variable pueda tomar un valor de entre varios. Por ejemplo, en el sistema de notas escolares español clásico, ciertas notas numéricas tienen "nombres" asociados: un 5 es un aprobado, un 9 y un 10 son sobresaliente, etc. Si queremos hacer un programa que convierta de la nota numérica a su equivalente escrito, podríamos emplear varias órdenes "SI", una tras la otra. Pero en muchos lenguajes de programación (y por tanto, también en muchas variantes de pseudocódigo) existe una alternativa más compacta y más legible: la orden "SEGUN". Esta orden permite hacer unas cosas u otras según el valor que tome una variable. Su uso sería así:

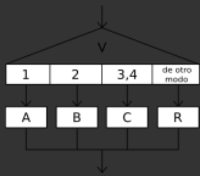
```
Segun variable Hacer
  valor1: pasos_a_dar_si_es_el_valor1
  valor2: pasos_a_dar_si_es_el_valor2
  valor3: pasos_a_dar_si_es_el_valor3
De Otro Modo:
  pasos_a_dar_si_es_otro_valor
FinSegun
```

El bloque "De Otro Modo" es opcional: si detallamos todos los valores posibles, no sería necesario utilizarlo.

El número de valores que podemos probar es indefinido: no tiene por qué ser 3, sino que pueden ser menos casos o muchos más.

Según la variante de pseudocódigo (o el lenguaje de programación) que empleemos, puede haber restricciones en el tipo de datos que es aceptable. Por ejemplo, en el caso de PseInt, la variable tiene que tener un valor numérico, no puede ser un texto.

Al igual que ocurría con la orden SI, existe un símbolo que podemos usar en los diagramas de flujo para ayudarnos a planificar nuestro programa (aunque este símbolo está menos extendido que el de SI):



Así, si en Pselnt hacemos clic en el icono del panel derecho que representa la condición SEGUN, aparece un esqueleto de programa casi completo, para que hagamos los cambios que nos interesen:

Sólo tenemos que escribir la condición que realmente nos interesa, y la serie de pasos a dar si se cumple y si no se cumple, de modo que nuestro programa podría quedar así:

```
Algoritmo EjemploCasos
  Escribir "Introduzca la nota"
  Leer nota
  Segun nota Hacer
    10:
      Escribir "Ha obtenido un sobresaliente alto"
    9:
      Escribir "Ha obtenido un sobresaliente bajo"
    8:
      Escribir "Ha obtenido un notable alto"
    7:
      Escribir "Ha obtenido un notable bajo"
    6:
      Escribir "Ha obtenido un aprobado alto"
    5:
      Escribir "Ha obtenido un aprobado"
    De Otro Modo:
      Escribir "Ha suspendido"
  FinSegun
FinAlgoritmo
```

Su resultado sería éste:

```
CA C:\temp\PSelnt\pseint.exe
*** Ejecucion Iniciada. ***
Introduzca la nota
> 5
Ha obtenido un aprobado
*** Ejecucion Finalizada. ***
```

Pero no siempre habrá que comprobar condiciones una única vez. Es muy frecuente que haya que hacerlo de forma repetitiva. Por ejemplo, "pedir una contraseña al usuario hasta que sea la correcta", o aún más cercano al mundo real, "pedir una contraseña al usuario hasta que sea la correcta o agote sus intentos". De eso hablaremos en la próxima entrega.

Ejercicio de repaso propuesto 5.1: Haz un programa que pida un número al usuario un número de día de la semana (del 1 al 7) y escriba el nombre de ese día (por ejemplo, "martes" para el día 2). Debes emplear la orden SEGUN.

Ejercicio de repaso propuesto 5.2: Crea una variante del ejercicio 5.1, en la que emplees varios SI-ENTONCES en vez de SEGUN.

Ejercicio de repaso propuesto 5.3: Crea un programa que pida al usuario un número de mes (por ejemplo, el 4) y escriba el nombre del mes correspondiente (por ejemplo, "abril"). Debes usar la orden SEGUN.

Ejercicio de repaso propuesto 5.4: Crea una variante del ejercicio 5.3, en la que emplees varios SI-ENTONCES en vez de SEGUN.

96317 visitas desde el 15-04-2019

[< Anterior](#)

[Posterior >](#)



APRENDEAPROGRAMAR.COM

Cursos gratuitos de programacion de ordenadores, en español

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. [Puede saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web). ✕

[< Anterior](#)[Posterior >](#)

6 - Condiciones repetitivas (1: Mientras)

Por: Nacho Cabanes

Actualizado: 29-05-2019 08:39

Tiempo de lectura estimado: 4 min.

Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (4:28 min):

EN LOS FOROS

17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)

17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)

16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)

16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)

16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)

[\(Anteriores...\)](#)

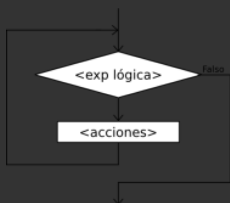
Pseudocódigo

6 - Condiciones Repetitivas (1: Mientras)

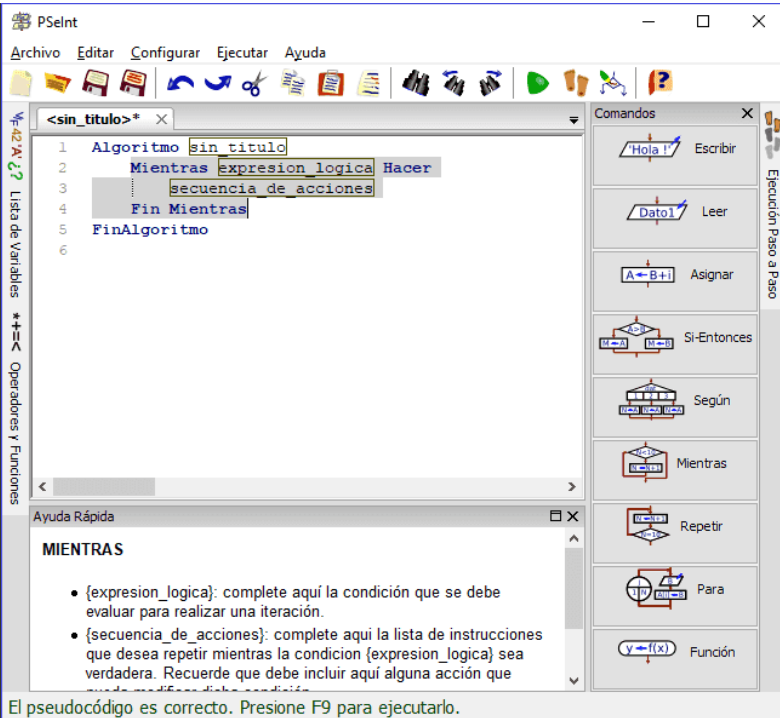
Habitualmente, una condición se deberá comprobar más de una vez. Por ejemplo, una condición de error puede repetirse: el usuario que introduce mal una contraseña por primera vez puede equivocarse también en una segunda ocasión.

Por eso, igual que cualquier lenguaje de programación tiene una orden "si", la gran mayoría de ellos tendrá una orden "mientras", que permite que un fragmento de un programa se repita mientras una cierta condición se siga cumpliendo (por ejemplo, mientras la contraseña que teclee el usuario sea incorrecta, el usuario deberá volver a introducirla).

También existe un símbolo habitual en los diagramas de flujo para representar este tipo de condiciones repetitivas, en las que si se cumple la condición, se realiza una serie de acciones y se vuelve a comprobar la condición, y así sucesivamente hasta que la condición no se cumpla:



Y en el caso de PseInt, ese ícono generaría un esqueleto de programa como éste:



El pseudocódigo es correcto. Presione F9 para ejecutarlo.

Por ejemplo, un programa capaz de sumar muchos números, todos los que el usuario quisiera, y en el que hubiera que escribir "0" para indicar que queremos terminar, podría ser así:

```
Algoritmo Mientras01
  Escribir "Dime un numero"
  Leer x
  suma <- 0
  Mientras x <> 0 Hacer
    suma <- suma + x
    Escribir "Hasta ahora, la suma es ", suma
    Escribir "Dime otro numero"
    Leer x
  FinMientras
  Escribir "Terminado"
FinAlgoritmo
```

Y su ejecución mostraría algo como:



Esta estructuras repetitivas reciben también el nombre de "bucles".

Hay más formas de comprobar condiciones repetitivas. Pronto las veremos...

Ejercicio de repaso propuesto 6.1: Crea un programa que pida al usuario una contraseña, de forma repetitiva mientras que no introduzca "1234". Cuando finalmente escriba la contraseña correcta, se le dirá "Bienvenido" y terminará el programa.

Ejercicio de repaso propuesto 6.2: Haz un programa que permita calcular la suma de pares de números. Pedirá dos números al usuario y mostrará su suma, volviendo a repetir hasta que ambos números introducidos sean 0.

Ejercicio de repaso propuesto 6.3: Crea un programa que genere dos números al azar entre el 0 y el 100, y pida al usuario que calcule e introduzca su suma. Si la respuesta no es correcta, deberá volver a pedirla tantas veces como sea necesario hasta que el usuario acierte. Pista: como verás en el apartado 10, para generar un número al azar del 0 al 100 puedes hacer

62243 visitas desde el 15-04-2019

[< Anterior](#)

[Posterior >](#)



APRENDEAPROGRAMAR.COM

[Cursos gratuitos de programacion de ordenadores, en español](#)

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta](#)

[Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. [Puede saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web).

[< Anterior](#)[Posterior >](#)

Buscar...

7 - Condiciones repetitivas (2: Repetir-hasta)

Por: Nacho Cabanes

Actualizado: 16-04-2019 08:37

Tiempo de lectura estimado: 3 min.

Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (3:21 min):

EN LOS FOROS

17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)

17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)

16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)

16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)

16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)

[\(Anteriores...\)](#)

Pseudocódigo

7 - Condiciones Repetitivas (2: Repetir-Hasta)

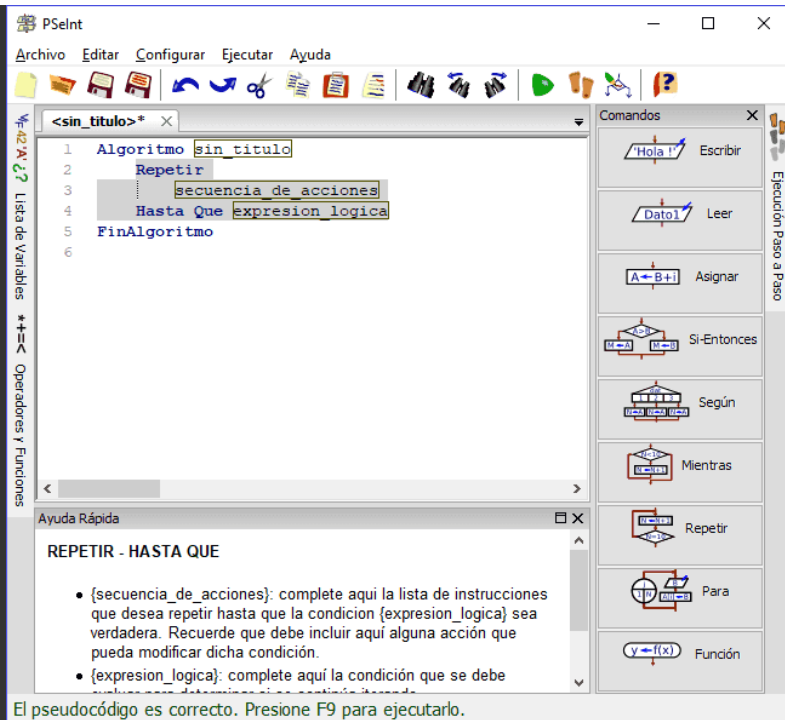
Es también muy frecuente que un bloque de programa repetitivo deba ejecutarse al menos una vez. Por ejemplo, si queremos pedir un dato al usuario, quizá exista algún error y haya que insistir, pero al menos deberemos pedírselo una primera vez.

En estos casos, la estructura "mientras" no es la más adecuada, porque no podemos comprobar antes de pedir el valor. En estos casos (que son muy frecuentes), sería más razonable usar otra estructura de programación en la que la condición se compruebe después de dar ciertos pasos. Esa estructura es "repetir... hasta":

Su representación en un diagrama de flujo sería:



Y en el caso de PSeInt, ese icono generaría un esqueleto de programa como éste:

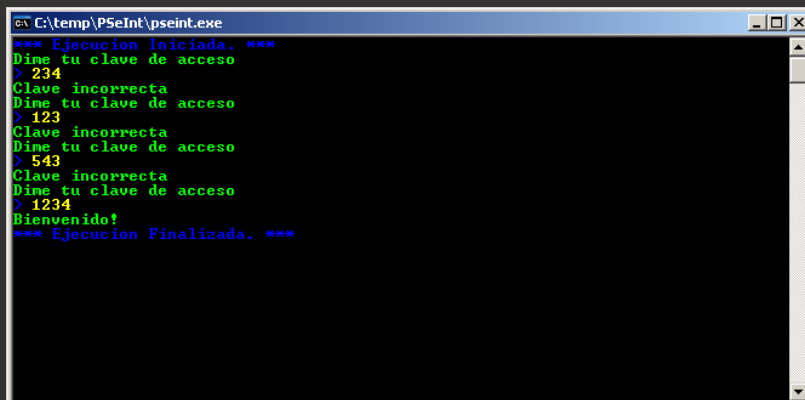


El pseudocódigo es correcto. Presione F9 para ejecutarlo.

Por ejemplo, un programa que pida al usuario una clave de acceso, y que no le permita seguir hasta que la introduzca correctamente, se podría hacer así:

```
Algoritmo Repetir01
  Repetir
    Escribir "Dime tu clave de acceso"
    Leer clave;
    Si clave <> 1234 Entonces
      Escribir "Clave incorrecta"
    FinSi
  Hasta Que clave=1234
  Escribir "Bienvenido!"
FinAlgoritmo
```

Cuyo resultado sería:



Queda otra forma de repetir fragmentos de programa. Pronto estaremos con ella...

Ejercicio de repaso propuesto 7.1: Crea un programa que pida al usuario un código de usuario y una contraseña. Deberá repetirse hasta que el código sea "1" y la contraseña sea "1234".

Ejercicio de repaso propuesto 7.2: Haz un programa que permita calcular la suma de pares de números. Pedirá dos números al usuario y mostrará su suma, volviendo a repetir hasta que ambos números introducidos sean 0. Esta vez deberás usar "Repetir", por lo que tu solución no será igual que la del ejercicio 6.2, que empleaba "Mientras".

Ejercicio de repaso propuesto 7.3: Prepara un programa que divida dos números que introduzca el usuario. Si el segundo número es cero, se le deberá avisar y volver a pedir tantas veces como sea necesario, hasta que introduzca un número distinto de cero, momento en que se calculará y mostrará el resultado de la división.

[< Anterior](#)

[Posterior >](#)



APRENDEAPROGRAMAR.COM

Cursos gratuitos de programacion de ordenadores, en español

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta](#)

[Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. [Puede saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web).

[< Anterior](#)[Posterior >](#)

Buscar...

8 - Repetir un cierto número de veces: Para

Por: Nacho Cabanes

Actualizado: 25-05-2019 22:50

Tiempo de lectura estimado: 7 min.

Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (3:18 min):

EN LOS FOROS

17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)

17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)

16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)

16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)

16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)

[\(Anteriores...\)](#)

Pseudocódigo

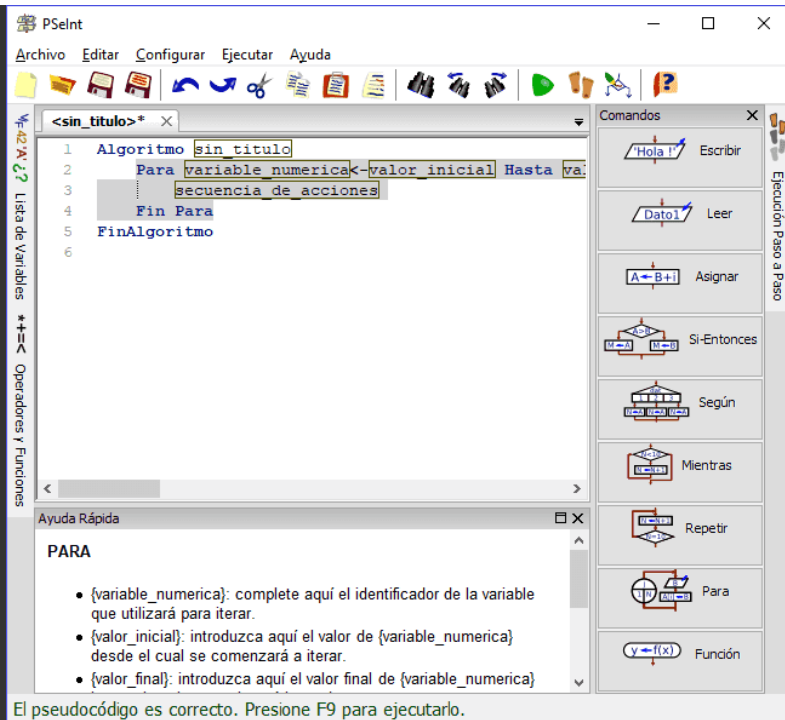
8 - Repetir Un Cierta Número De Veces: Para

En muchas ocasiones, no queremos que algo se repita mientras se cumpla una condición, sino un cierto número de veces. Por ejemplo, para escribir "Hola" 3 veces en pantalla existe una orden más cómoda que la orden "mientras" o la orden "repetir... hasta". Es la orden "para", que hace que una variable tome una serie de valores que se van incrementando. Por ejemplo, una estructura como "para x con valores desde 2 hasta 4" haría que un bloque de programa se repitiera 3 veces. En la primera repetición, la variable "x" tendría el valor 2, en la segunda tendría el valor 3 y en la tercera tendría el valor 4. La sintaxis exacta en PSeInt es: "Para variable <- valorInicial Hasta valorFinal Hacer"

Su representación en un diagrama de flujo sería:



Y en el caso de PSeInt, ese icono generaría un esqueleto de programa como éste:

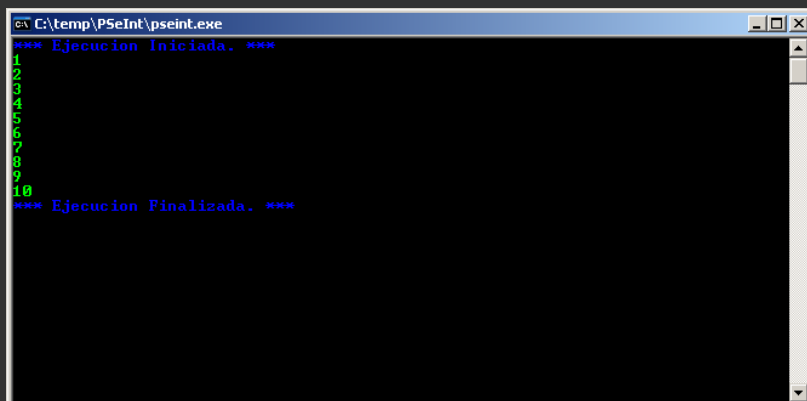


El pseudocódigo es correcto. Presione F9 para ejecutarlo.

Por ejemplo, un programa que mostrara los números del 1 al 10, podría ser:

```
Algoritmo Para01
  Para x <- 1 Hasta 10 Hacer
    Escribir x
  FinPara
FinAlgoritmo
```

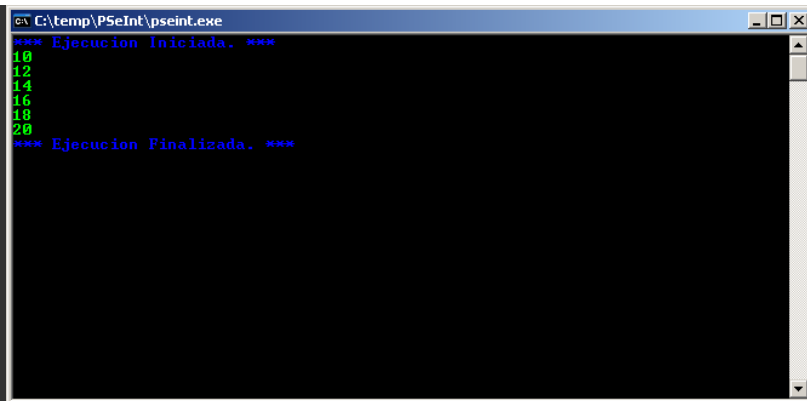
Cuyo resultado sería:



Si no queremos avanzar de uno en uno, sino con un incremento distinto, podemos indicar otro tamaño de "paso":

```
Algoritmo Para02
  Para x <- 10 Hasta 20 Con Paso 2 Hacer
    Escribir x
  FinPara
FinAlgoritmo
```

Y obtendríamos:



Hemos visto casi todas las construcciones que podremos encontrar en los lenguajes de programación convencionales (y, por tanto, al planificarlos usando pseudocódigo). En la próxima entrega veremos alguna otra posibilidad, como el manejo de matrices y las funciones matemáticas, para pasar después a algunos ejemplos completos.

Ejercicio de repaso propuesto 8.1: Crea un programa que escriba los números del 5 al 15, ambos incluidos.

Ejercicio de repaso propuesto 8.2: Crea un programa que escriba los múltiplos del 3, desde el 3 hasta el 30, usando un paso de tamaño 3.

Ejercicio de repaso propuesto 8.3: Crea un programa que escriba los múltiplos del 3, desde el 3 hasta el 30, contando del uno al diez pero mostrando ese contador multiplicado por tres.

Ejercicio de repaso propuesto 8.4: Crea un programa que escriba los números del 20 al 10, descendiendo.

Ejercicio de repaso propuesto 8.5: Crea un programa que escriba la tabla de multiplicar del 5: desde "5 x 0 = 0" hasta "5 x 10 = 50"

Ejercicio de repaso propuesto 8.6: También se puede contar usando una orden "mientras" o una orden "repetir", si usas una variable como contador e incrementas (o disminuyes) su valor en cada pasada de forma manual. Compruébalo creando un programa que escriba los números del 1 al 15 usando "mientras" en vez de "para".

Las estructuras repetitivas ("bucles") se pueden incluir una dentro de otra si fuera necesario. El resultado se conoce como un "bucle anidado". Por ejemplo, si se desea dibujar un rectángulo usando asteriscos, se puede plantear como escribir de forma repetitiva varias filas que, a su vez están formadas cada una por varias columnas, de forma también repetitiva, como muestra este ejemplo:

```
// Rectángulo formado por asteriscos

Algoritmo Rectangulo

    Escribir Sin Saltar "Introduce el ancho: "
    Leer ancho
    Escribir Sin Saltar "Introduce el alto: "
    Leer alto

    Para fila<-1 Hasta alto Hacer

        Para columna <- 1 Hasta ancho Hacer
            Escribir Sin Saltar "*"
        FinPara

        Escribir "" // Avance de línea tras cada fila

    FinPara

FinAlgoritmo
```

Ejercicio de repaso propuesto 8.7: A partir del ejemplo que dibuja un rectángulo de asteriscos, crea un que dibuje un cuadrado (deberá pedir sólo un dato, el lado, y ambas órdenes "para" deberán tener ese valor como límite).

Ejercicio de repaso propuesto 8.8: Dibuja un triángulo creciente de asteriscos, del tamaño que indique el usuario. Por ejemplo, si escoge 4, el resultado debería ser:

*
**

113485 visitas desde el 15-04-2019

[◀ Anterior](#)

[Posterior ▶](#)



APRENDEAPROGRAMAR.COM

[Cursos gratuitos de programacion de ordenadores, en español](#)

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta](#)

[Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. ✕
Puede [saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web).

[◀ Anterior](#)[Posterior ▶](#)

9 - Matrices

👤 Por: Nacho Cabanes

📅 Actualizado: 16-04-2019 08:47

🕒 Tiempo de lectura estimado: 6 min.

📖 Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (2:38 min):

EN LOS FOROS

17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)

17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)

16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)

16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)

16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)

[\(Anteriores...\)](#)

Pseudocódigo

9 - Matrices

Cuando necesitamos manejar muchos datos, generalmente hay soluciones más efectivas que tener muchas variables. Por ejemplo, si hay que guardar 10 números, resultará incómodo usar 10 variables llamadas n1, n2, n3, n4, n5, n6, n7, n8, n9, n10; si se trata de 100 números, puede pasar de resultar incómodo a ser totalmente inviable. En esos casos, será más eficiente almacenar esos 100 datos "juntos", formando lo que se conoce como una "matriz", en vez de usar 100 variables distintas.

La palabra "matriz" es una traducción del inglés "array". Algunos autores lo traducen alternativamente como tabla, vector o incluso "arreglo".

Normalmente, en una matriz podremos acceder individualmente a cada uno de sus elementos usando corchetes: el primer dato sería algo como "dato[1]".

Como primer contacto, se puede crear una matriz de 3 elementos, guardar datos en sus 3 posiciones y mostrar el primer dato y el tercero así:

```
Algoritmo Matriz3

    Dimension datos[3]

    datos[1] <- 10
    datos[2] <- 25
    datos[3] <- 50

    Escribir "El primer dato es ", datos[1]
    Escribir "Y el tercero es ", datos[3]

FinAlgoritmo
```

Como una matriz puede contener muchos datos, es frecuente recorrerlas de forma repetitiva, empleando una orden "para". Así un ejemplo que nos pida 6 datos y luego los muestre en orden contrario al que se han introducido podría ser:

```
Algoritmo MostrarAlReves

    Dimension datos[6]

    Para i <- 1 Hasta 6 Hacer
        Escribir "Dime el dato numero ", i
        Leer datos[i]
    FinPara

    Escribir "Los datos al reves son: "
    Para i <- 6 Hasta 1 Con Paso -1 Hacer
        Escribir datos[i]
    FinPara

FinAlgoritmo
```

Que mostraría algo como:

```
*** Ejecucion Iniciada. ***
Dime el dato numero 1
> 3
Dime el dato numero 2
> 4
Dime el dato numero 3
> 5
Dime el dato numero 4
> 6
Dime el dato numero 5
> 7
Dime el dato numero 6
> 8
Los datos al reves son:
8
7
6
5
4
3
*** Ejecucion Finalizada. ***
```

(Haremos ejemplos más avanzados como parte de los ejercicios)

Ejercicio de repaso propuesto 9.1: Crea un programa que pida un número al usuario un número de mes (por ejemplo, el 4) y diga cuántos días tiene (por ejemplo, 30). Debes usar una matriz.

Ejercicio de repaso propuesto 9.2: Haz un programa que pida al usuario 5 datos, los guarde en una matriz y luego muestre su media (la suma de los 5 datos, dividida entre 5).

Ejercicio de repaso propuesto 9.3: Crea una versión ampliada del ejercicio 9.2, en la que se pida al usuario 5 datos, se guarden en una matriz y finalmente se calcule su media, se muestre ésta y se muestren también los valores mayores que la media).

Ejercicio de repaso propuesto 9.4: Si has estudiado lo que es un vector, crea un programa que reserve espacio para dos vectores de 3 componentes, pida al usuario sus valores y calcule la suma de ambos vectores (su primera componente será x_1+y_1 , la segunda será x_2+y_2 y así sucesivamente).

Ejercicio de repaso propuesto 9.5: Si has estudiado lo que es un vector, prepara un programa que reserve espacio para dos vectores de 3 componentes, pida al usuario sus valores y calcule su producto escalar ($x_1 \cdot y_1 + x_2 \cdot y_2 + x_3 \cdot y_3$).

Ejercicio de repaso propuesto 9.6: Haz un programa que pida al usuario 7 números enteros y calcule (y muestre) cuál es el mayor de ellos. Nota: para calcular el mayor valor de una matriz, hay que comparar cada uno de los valores que tiene almacenados con el que hasta ese momento es el máximo provisional. El valor inicial de este máximo provisional no debería ser cero (porque el resultado sería incorrecto si todos los números son negativos), sino el primer elemento de la matriz.

Ejercicio de repaso propuesto 9.7: Crea un programa que prepare un array con 10 datos prefijados, luego pregunte al usuario qué dato desea buscar, avise si ese dato no aparece, y que en caso contrario diga cuántas veces se ha encontrado.

También se pueden crear arrays "de dos dimensiones" (por ejemplo, 3 filas y dos columnas), que se acercan más al concepto matemático de matriz:

```
Algoritmo Matriz32
```

```
    Dimension datos[3,2]
```

```
    datos[1,1] <- 10
```

```
    datos[2,1] <- 25
```

```
    datos[3,1] <- 50
```

```
    datos[1,2] <- 11
```

```
    datos[2,2] <- 26
```

```
    datos[3,2] <- 51
```

```
    Escribir "El primer dato en la primera columna es ", datos[1,1]
```

```
    Escribir "Y el ultimo de la segunda columna ", datos[3,2]
```

```
FinAlgoritmo
```

Ejercicio de repaso propuesto 9.8: Crea un programa que pida al usuario dos bloques de 10 números cada uno (usando un array de dos dimensiones). Después deberá mostrar el mayor dato que se ha introducido en cada uno de esos dos bloques.

Ejercicio de repaso propuesto 9.9: Si has estudiado álgebra matricial, haz un programa que calcule el determinante de una matriz de 2x2, a partir de datos que introduzca el usuario.

Ejercicio de repaso propuesto 9.10: Si has estudiado álgebra matricial, prepara un programa que calcule el determinante de una matriz de 3x3.

74467 visitas desde el 15-04-2019

[< Anterior](#)

[Posterior >](#)



APRENDEA PROGRAMAR.COM

Cursos gratuitos de programación de ordenadores, en español

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta](#)

[Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. Puede [saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web).

Anterior Posterior Buscar...

10 - Funciones matemáticas.

Por: Nacho Cabanes
Actualizado: 17-06-2019 21:15
Tiempo de lectura estimado: 5 min.
Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (2:14 min):

EN LOS FOROS

- 17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)
- 17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)
- 16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)
- 16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)
- 16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)
- (Anteriores...)

Pseudocódigo

10 - Funciones Matemáticas.

Casi cualquier lenguaje de programación tiene incorporadas ciertas funciones matemáticas, de modo que nos permita calcular raíces cuadradas, logaritmos, senos y cosenos, etc.

Lo mismo ocurre en muchas variantes de pseudocódigo. Por ejemplo, PseInt incluye las siguientes funciones matemáticas:

Función	Significado
RC(X) o RAIZ(x)	Raíz Cuadrada de X
ABS(X)	Valor Absoluto de X
LN(X)	Logaritmo Natural de X
EXP(X)	Función Exponencial de X
SEN(X)	Seno de un ángulo X (medido en radianes)
COS(X)	Coseno de un ángulo X (medido en radianes)
TAN(X)	Tangente de un ángulo X (medido en radianes)
ASEN(X)	Arco seno de X
ACOS(X)	Arco coseno de X
ATAN(X)	Arco tangente de X
TRUNC(X)	Parte entera de X
REDON(X)	Entero más cercano a X

Función	Significado
AZAR(X)	Entero aleatorio entre 0 y x-1

Si no sabes qué hace alguna de estas funciones, no te preocupes mucho, es señal de que hasta ahora no la has necesitado... ;-) Vamos a ver un ejemplo de algunas de ellas, y luego practicaremos las más importantes como parte de los ejercicios.

```

Algoritmo FuncionesMatematicas
  Escribir Raiz(4)    // Raiz de 4 = 2
  Escribir RC(9)      // Raiz de 9 = 3
  Escribir Trunc(7.9) // 7.9 truncado = 7
  Escribir Redon(7.9) // 7.9 redondeado = 8
  Escribir Azar(100)  // Al azar entre 0 y 99
  dado <- 1+Azar(6)   // 1 + azar entre 0 y 5
  Escribir dado
  Escribir Abs(-5)    // Valor absoluto de -5 = 5
  Escribir Cos(3.14)  // Coseno PI radianes
  Escribir Sen(90 * Pi/180) // Seno de 90 grados = 1
FinAlgoritmo

```

Ejercicio de repaso propuesto 10.1: Crea un programa que genere un número al azar entre el 1 y el 100. El usuario tendrá 7 oportunidades para adivinarlo. Tras cada intento, se le dirá si se ha pasado o se ha quedado corto.

Ejercicio de repaso propuesto 10.2: Haz un programa que calcule raíces cuadradas: el usuario introducirá un número y se le mostrará su raíz cuadrada. Se repetirá hasta que introduzca el número 0 (para el que no se deberá mostrar su raíz). Si introduce un valor negativo, se deberá mostrar un aviso en vez de intentar calcular su raíz.

Ejercicio de repaso propuesto 10.3: PseInt no incluye ninguna función para calcular raíces cúbicas ni con ningún otro índice distinto de 2, pero esto es algo fácil de solucionar: para hallar la raíz cúbica de un número basta con elevar a (1/3). Pruébalo con un programa que calcule la raíz cúbica de un número introducido por el usuario (puedes comprobar que funciona correctamente si pruebas con el número 8, cuya raíz cúbica es 2).

Ejercicio de repaso propuesto 10.4: Si has estudiado trigonometría, quizá te suene que el seno de un ángulo de 45 grados es (raíz de 2) / 2. Haz un programa que muestre los resultados de ambas operaciones, para ver si parecen iguales. Recuerda que el ángulo se ha de indicar en radianes y que puedes convertir de grados a radianes multiplicando por PI y dividiendo por 180: SEN(45*PI/180).

Ejercicio de repaso propuesto 10.5: Pide al usuario dos números y muestra la "distancia" entre ellos (el valor absoluto de su diferencia, de modo que el resultado sea siempre positivo).

Ejercicio de repaso propuesto 10.6: Pide al usuario dos pares de números x1,y2 y x2,y2, que representen dos puntos en el plano. Calcula y muestra la distancia entre ellos (raíz cuadrada de ((x1-x2)² + (y1-y2)²)).

54457 visitas desde el 15-04-2019

[< Anterior](#)

[Posterior >](#)



APRENDE A PROGRAMAR.COM

Cursos gratuitos de programación de ordenadores, en español

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta](#)
[Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. [Puede saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web).

[< Anterior](#)[Posterior >](#)

11.- Funciones de cadenas de texto

Por: Nacho Cabanes

Actualizado: 16-04-2019 08:51

Tiempo de lectura estimado: 5 min.

Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (3:27 min):

EN LOS FOROS

17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)

17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)

16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)

16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)

16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)

[\(Anteriores...\)](#)

Pseudocódigo

11.- Funciones De Cadenas De Texto

Desde octubre de 2012, PseInt incluye ciertas funciones de manejo de cadenas de texto que es habitual encontrar en casi cualquier lenguaje de programación.

Las dos más sencillas son las que permiten convertir un texto a mayúsculas y a minúsculas:

```
Algoritmo Cadenas01
  Escribir "Dime tu nombre"
  Leer nombre

  Escribir "En mayúsculas quedaría:"
  Escribir Mayusculas(nombre)

  Escribir "En minúsculas quedaría:"
  Escribir Minusculas(nombre)
FinAlgoritmo
```

Para resolver problemas reales de computación, hay otras dos funciones que dan mucho más juego: una nos dicen la longitud de una cadena y otra permite extraer una subcadena, desde cierta posición inicial hasta cierta posición final:

```

Algoritmo Cadenas02a
  Escribir "Dime tu nombre"
  Leer texto

  Escribir "Su longitud es:"
  Escribir Longitud(texto)

  Escribir "Su primera letra es:"
  Escribir Subcadena(texto, 1, 1)
FinAlgoritmo

```

Como ejemplo de su uso real, es posible mostrar un texto incluyendo un espacio en blanco adicional entre cada letra y la siguiente, si se crea un programa como éste:

```

Algoritmo Cadenas02b
  Escribir "Dime tu nombre"
  Leer texto

  Escribir "Con espacios adicionales, quedaría:"
  Para x <- 1 Hasta Longitud(texto) Hacer
    Escribir Sin Saltar Subcadena(texto, x, x)
    Escribir Sin Saltar " "
  FinPara
  Escribir ""
FinAlgoritmo

```

(Si usamos la configuración normal de PseInt, las posiciones dentro de la cadena se empezarán a contar desde uno; ya sabemos, como vimos en el apartado sobre "Matrices", que en muchos otros lenguajes de programación se empieza a contar desde cero; PseInt también se puede configurar para trabajar así).

También es posible "concatenar" dos cadenas, es decir, formar una nueva cadena a partir de la unión de esas dos. Por ejemplo, el resultado de Concatenar("1a", "2b") sería la cadena de texto "1a2b". Así, el ejemplo del rectángulo de asteriscos se podría hacer, de forma alternativa, creando primero una cadena que fuera la fila de asteriscos y luego mostrándola varias veces, así:

```

// Rectángulo formado por asteriscos

// Versión concatenando cadenas.
// No funciona con versiones de PseInt de antes de Octubre de 2012

Algoritmo Rectangulo2

  Escribir Sin Saltar "Introduce el ancho: "
  Leer ancho
  Escribir Sin Saltar "Introduce el alto: "
  Leer alto

  // Creamos una fila formada por asteriscos
  filaDeAsteriscos <- ""
  Para columna <- 1 Hasta ancho Hacer
    filaDeAsteriscos <- Concatenar( filaDeAsteriscos, "*" )
  FinPara

  // Y la mostramos varias veces
  Para fila<-1 Hasta alto Hacer
    Escribir filaDeAsteriscos
  FinPara

FinAlgoritmo

```

Ejercicio de repaso propuesto 11.1: Crea un programa que pida su nombre al usuario y lo escriba al revés (de la última letra a la primera: a partir de "Nacho" escribiría "ohcaN").

Ejercicio de repaso propuesto 11.2: Crea un programa que pida su nombre al usuario y lo escriba alternando letras mayúsculas y minúsculas (por ejemplo, "nAcho" se mostraría como "NaChO").

Ejercicio de repaso propuesto 11.3: Crea un programa que pida su nombre al usuario y diga cuántas vocales contiene (por ejemplo, "Aurora" tiene 4 vocales).

Ejercicio de repaso propuesto 11.4: Crea un programa que pida su nombre al usuario y diga qué vocales contiene (en orden y sin repetir: por ejemplo, para "Aurora" deberá responder "aou").

Ejercicio de repaso propuesto 11.5: Crea un programa que pida una frase al usuario y diga cuántas palabras contiene (pista: puedes contar los espacios, prestando atención en que no estén repetidos).

Ejercicio de repaso propuesto 11.6: Crea un programa que pida al usuario su nombre y apellidos y los muestre con las mayúsculas y minúsculas correctas (aparecerán en mayúsculas la primera letra y la que haya tras cada espacio; las demás aparecerán en minúsculas. Por

ejemplo, si introduce "nAcho cabaNes", es escribirá "Nacho Cabanes").

119365 visitas desde el 15-04-2019

[◀ Anterior](#)

[Posterior ▶](#)



APRENDEAPROGRAMAR.COM

[Cursos gratuitos de programacion de ordenadores, en español](#)

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta](#)

[Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)

Este sitio web usa cookies de terceros para analizar el tráfico y personalizar los anuncios. Si no está de acuerdo, abandone el sitio y no siga navegando por él. [Puede saber más sobre nuestra política de cookies](#) (Este aviso se muestra sólo una vez en cada visita al sitio web).

[< Anterior](#)[Posterior >](#)

12.- Creación de funciones y procedimientos (subprocesos)

👤 Por: Nacho Cabanes

📅 Actualizado: 25-05-2019 22:45

🕒 Tiempo de lectura estimado: 10 min.

📖 Curso: [Introducción a la programación - Pseudocódigo](#)

Versión en video (10:13 min):

EN LOS FOROS

17-11-2021 22:34 [Necesito ayuda urgente con un programa en c++ \[C++\]](#)

17-11-2021 02:36 [un código que cumpla con lo siguiente \[Pseudoc\]](#)

16-11-2021 23:23 [Resolucion Ejercicio \[C\]](#)

16-11-2021 19:59 [Como puedo poner botones dentro de un boton? \[Python\]](#)

16-11-2021 18:25 [Tarjetas NFC \[Java\]](#)

[\(Anteriores...\)](#)

Pseudocódigo

12.- Creación De Funciones Y Procedimientos (Subprocesos)

En muchos casos, nos encontraremos con tareas que tenemos que repetir varias veces en distintos puntos de nuestro programa. Si tecleamos varias veces el mismo fragmento de programa no sólo tardaremos más en escribir: además el programa final resultará menos legible, será más también será más fácil que cometamos algún error alguna de las veces que volvemos a teclear el fragmento repetitivo, o que decidamos hacer una modificación y olvidemos hacerla en alguno de los fragmentos. Por eso, conviene evitar que nuestro programa contenga código repetitivo. Una de las formas de evitarlo es usar "subrutinas", una posibilidad que la mayoría de lenguajes de programación permiten, y que en ocasiones recibe el nombre de "procedimientos" o de "funciones" (existe algún matiz que hace que esas palabras no sean realmente sinónimas y que comentaremos más adelante).

PseInt permite definir "subrutinas" (o "funciones") dentro del pseudocódigo, desde la versión del 10 de octubre de 2012. En su caso, se llaman "subprocesos". Veamos un ejemplo de su uso:

Vamos a empezar por crear un subproceso (o "subrutina", o "procedimiento") que escriba 20 guiones, que podríamos utilizar para subrayar textos. Un programa completo que escribiera tres textos y los subrayara podría ser:

```

Algoritmo SubProcesos01
  Escribir " Primer ejemplo"
  Para x <- 1 Hasta 20 Hacer
    Escribir Sin Saltar "-"
  FinPara
  Escribir ""

  Escribir " Segundo ejemplo"
  Para x <- 1 Hasta 20 Hacer
    Escribir Sin Saltar "-"
  FinPara
  Escribir ""

  Escribir " Tercer ejemplo"
  Para x <- 1 Hasta 20 Hacer
    Escribir Sin Saltar "-"
  FinPara
  Escribir ""
FinAlgoritmo

```

Muy repetitivo. Sería un poco más elegante si lo reescribimos así:

```

Algoritmo SubProcesos02
  Escribir " Primer ejemplo"
  Subrayar

  Escribir " Segundo ejemplo"
  Subrayar

  Escribir " Tercer ejemplo"
  Subrayar
FinAlgoritmo

Subproceso Subrayar
  Para x <- 1 Hasta 20 Hacer
    Escribir Sin Saltar "-"
  FinPara
  Escribir ""
FinSubproceso

```

Mucho más legible, pero todavía no está tan bien como debería: siempre estamos escribiendo 20 guiones, aunque el texto sea más largo o más corto. En la mayoría de lenguajes de programación se puede indicar detalles adicionales ("parámetros") para que se puedan utilizar desde dentro de esa subrutina. Por ejemplo, en nuestro caso podríamos indicarle qué texto queremos escribir y qué longitud queremos que tenga la secuencia de guiones:

```

Algoritmo SubProcesos03
  EscribirSubrayado(" Primer ejemplo", 16)
  EscribirSubrayado(" Segundo ejemplo", 17)
  EscribirSubrayado(" Tercer ejemplo", 16)
FinAlgoritmo

Subproceso EscribirSubrayado(texto, cantidad)
  Escribir texto
  Para x <- 1 Hasta cantidad Hacer
    Escribir Sin Saltar "-"
  FinPara
  Escribir ""
FinSubproceso

```

Eso todavía es un poco redundante: en general, queremos escribir tantos guiones como letras tenga el texto, así que no será necesario indicar ese dato. Desde octubre de 2012, PseInt incluye ciertas funciones predefinidas para manejo de cadenas de texto; una de ellas es "Longitud", que nos indica la cantidad de letras que tiene un texto, de modo que nuestro programa se podría simplificar así:

```

Algoritmo SubProcesos04
  EscribirSubrayado("Primer ejemplo")
  EscribirSubrayado("Segundo ejemplo")
  EscribirSubrayado("Tercer ejemplo")
FinAlgoritmo

Subproceso EscribirSubrayado(texto)
  Escribir texto
  Para x <- 1 Hasta Longitud(texto) Hacer
    Escribir Sin Saltar "-"
  FinPara
  Escribir ""
FinSubproceso

```

También podemos crear subprocesos que realicen ciertas operaciones aritméticas y devuelvan un resultado, como las funciones matemáticas que vimos en el tema 10. Se puede hacer con la misma palabra "subproceso" que hemos empleado hasta ahora, pero muchos lenguajes de programación

distinguen entre un "procedimiento" o "subrutina", cuando se da una serie de pasos pero no se devuelve ningún valor (como habíamos hecho hasta ahora), y una "función", cuando sí se devuelve un resultado (como haremos en esta ocasión), así que usaremos esta última nomenclatura. Su desarrollo se parece mucho a lo que hemos hecho hasta ahora, con la diferencia de que escogeremos un nombre para la variable que guardará el resultado, así:

```
Funcion resultado <- Maximo( n1, n2)
  Si n1 > n2 Entonces
    resultado <- n1
  Sino
    resultado <- n2
  FinSi
FinFuncion

Algoritmo Funciones01
  Escribir "El máximo de 5 y 12 es:"
  Escribir Maximo(5,12)
  Escribir "El máximo de 25 y 12 es:"
  Escribir Maximo(25,12)
FinAlgoritmo
```

Este programa crea una función que calcula cuál es el mayor de los dos números que se le indican como parámetro, y la usa dos veces, para mostrar el máximo de dos valores prefijados.

No sólo podemos devolver valores numéricos; también podemos devolver cadenas (como las funciones que vimos en el apartado 13) o "valores de verdad" (verdadero, falso). Por ejemplo, podemos crear una función que calcule si un número es primo o no (lo vamos a hacer de la forma más simple pero también de la menos eficiente: aplicar la definición, probando a dividir entre todos los números que hay entre 1 y n; si hemos encontrado dos divisores -o menos, para el número uno-, entonces el número es primo):

```
SubAlgoritmo resultado <- Primo ( num )
  cantidadDivisores <- 0
  Para cont <- 1 Hasta num Hacer
    Si num % cont = 0 Entonces
      cantidadDivisores <- cantidadDivisores + 1
    FinSi
  FinPara
  Si cantidadDivisores <= 2 Entonces
    resultado <- verdadero
  Sino
    resultado <- falso
  FinSi
FinSubAlgoritmo

Algoritmo PrimosDel1Al30
  Para n <- 1 hasta 30
    si Primo(n) Entonces
      Imprimir n
    FinSi
  FinPara
FinAlgoritmo
```

Hay más detalles que comentar sobre funciones, pero son un poco más avanzados, así que vamos a descansar un poco aquí de nuevos conceptos y a practicar lo que hemos visto...

Ejercicio de repaso propuesto 12.1: Crea un procedimiento EscribirCentrado, que reciba como parámetro un texto y lo escriba centrado en pantalla (suponiendo una anchura de 80 columnas; pista: deberás escribir 40 - longitud/2 espacios antes del texto).

Ejercicio de repaso propuesto 12.2: Crea una función CantidadDeDivisores, que reciba un número entero y devuelva la cantidad de divisores (por ejemplo, para el número 16, sus divisores son 1, 2, 4, 8, 16, por lo que la respuesta debería ser 5).

Ejercicio de repaso propuesto 12.3: Crea un programa que pida dos números enteros al usuario y diga si alguno de ellos es múltiplo del otro. Crea una función EsMultiplo que te ayude a que el proceso principal sea legible.

Ejercicio de repaso propuesto 12.4: Crea un procedimiento EscribirEspaciado, que reciba como parámetro un texto y lo escriba con un espacio adicional tras cada letra. Por ejemplo, "Hola, tú" se escribiría "H o l a , t ú ".

Ejercicio de repaso propuesto 12.5: Crea una función MayorDeTres, que reciba tres números enteros y devuelva el valor del mayor de ellos. Por ejemplo, para los números 5, 7 y 5, devolvería el valor 7.

Ejercicio de repaso propuesto 12.6: Crea una función EsPar que devuelva el valor lógico "verdadero" o "falso" según si el número que se indique como parámetro es par o no lo es.

Ejercicio de repaso propuesto 12.7: Crea una función Cubo, que reciba un número y lo

devuelva elevado al cubo.

Ejercicio de repaso propuesto 12.8: Crea una función Iniciales, que devuelva una cadena formada por las iniciales de la frase que se indique como parámetro (primera letra y la letra que haya tras cada espacio; por ejemplo, para "Nacho Cabanes" devolvería "NC").

Ejercicio de repaso propuesto 12.9: Crea una función Contiene, que reciba una cadena y una (posible) subcadena, y devuelva "verdadero" o "falso", según si la primera contiene a la segunda (como "Nacho" contiene "ac") o no la contiene, (como "Aurora" no contiene "sol").

81080 visitas desde el 15-04-2019

[< Anterior](#)

[Posterior >](#)

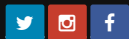


APRENDEAPROGRAMAR.COM

[Cursos gratuitos de programacion de ordenadores, en español](#)

[¿Por qué?](#)

[Preguntas frecuentes](#)



MAPA DEL SITIO

[Tutoriales](#)

[Foros](#)

[Referencia](#)

NOVEDADES

[24-12-2018: Algunos apartados incluyen... ¡vídeo!](#)

[22-03-2015: Puedes votar cuando te guste una respuesta](#)

[Anteriores...](#)

- [Contactar](#)
- [¿Quiénes somos?](#)