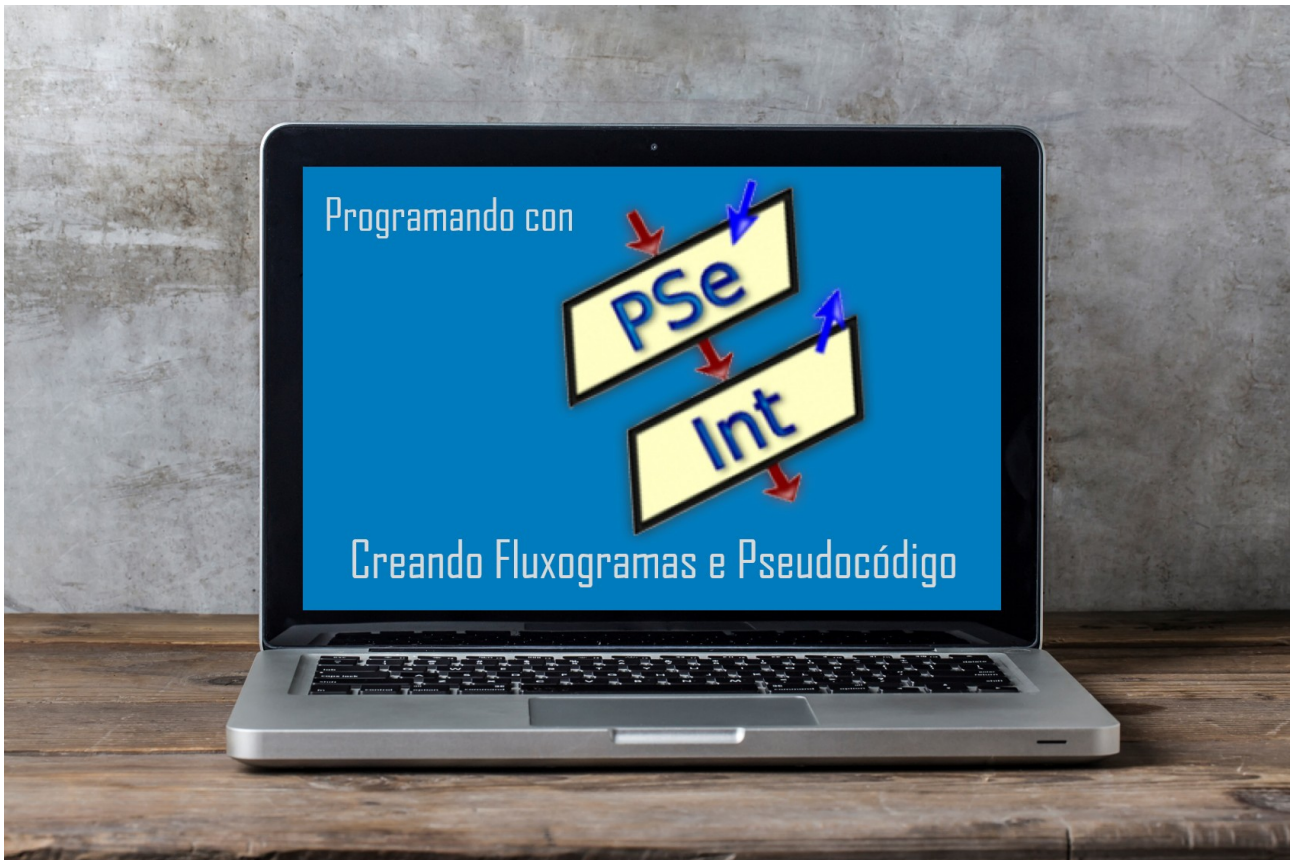


Programando coa aplicación Pseint



1.- Para comezar

CODE OF
PRACTICE



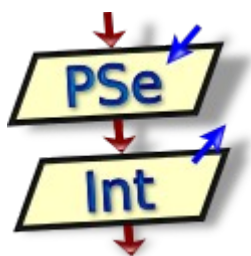
O mellor para deseñar e poñer en práctica un algoritmo computacional é traballando con diagramas de fluxo acompañados do correspondente pseudocódigo.

Para traballar con pseudocódigo e diagramas en paralelo, podemos empregar algunha ferramenta específica.

Non hai moita variedade de ferramentas sinxelas que permitan o traballo con diagramas

de fluxo e pseudocódigo en paralelo. Unha desas ferramentas, sinxela e de gran utilidade é Pseint.

Que é PSeint?



PSeint é unha ferramenta para aprender a lóxica da programación dirixida a persoas sen experiencia neste campo. É totalmente gratuíto e libre, cunha licenza GPL. Emprega unha pseudo-linguaxe intuitivo simple e limitado en castelán. Mediante Pseint podemos comprender conceptos básicos e fundamentais dos algoritmos computacionais.

Para que serve PseInt?

PSeint está deseñado para axudar á construción de algoritmos ou programas informáticos. O pseudocódigo adóitase utilizar como primeiro contacto para introducir conceptos básicos como o uso de estruturas de control, expresións, variables, etc., sen ter que xestionar as peculiaridades da sintaxe dunha linguaxe real.

Video Introducción a PseInt: [Programación con PseInt Introducción](#)

Video Primeros pasos: [Programación con PseInt - Primeros pasos](#)

Elementos básicos

Para deseñar un programa optamos entón polo emprego de Diagramas de Fluxo de Datos (DFD) e Pseudocódigo. E farémolo por medio da aplicación Pseint. O que debemos coñecer son as posibilidades que nos dá esta aplicación para deseñar, plasmar e probar os algoritmos que debemos crear.

a) Datos

Os algoritmos van funcionar en base á adquisición, operación e saída de datos. Un dato, no mundo da programación, será a expresión que caracteriza ás entidades coas que operamos nun algoritmo.

PSeint traballa cos seguintes tipos de datos:

a.1) Numéricos: Poden ser enteiros ou reais.

Enteiros: Conxunto dos números positivos, negativos e o cero. Representanse cos díxitos do 0 ao 9, precedidos dun guión para representar aos números negativos: 154, -154, 0, 700, 084, -084.

Reais: Os números reais que máis se empregarán serán os números con decimais. Os números decimais teñen unha parte enteira, unha parte decimal, un punto como separador decimal entre ambas partes e un guión ao comezo para indicar si se trata dun número negativo.

Números reais

Números racionais: -3/4, 5/8, 31/7
Números enteros: -7, -1, 0, 5, 20
Números irracionais: $\sqrt{2}$, $(1+\sqrt{5})/2$
Números trascendentes: e , π , $\ln(2)$

a.2) Lóxico: O tipo de datos lóxico, tamén chamado booleano é un tipo que só ten dous valores posibles: verdadeiro e falso. Este tipo de dato úsase para realizar operacións lóxicas sobre os datos, tomar decisións e controlar a execución dos programas.

b) Carácter: O tipo de datos de carácter permite procesar letras, números, símbolos e ideogramas ao asociar un código numérico a cada carácter que se desexa representar. O código máis empregado é código ASCII.

Os caracteres agrúpanse en cadeas de caracteres: 'Ola' "Ola mundo" '123' 'FALSO'...
(Péchanse entre comiñas dobres ou simples)

c) Expresións: Unha expresión en pseudocódigo é a que manifesta un cálculo. Polo tanto, nas expresións están involucrados os datos de partida do cálculo, e os operadores ou funcións.



d) Operadores: Os operadores son os elementos que empregamos para construír as expresións a partir dos datos de entrada do algoritmo.

Operador	Significado
Relacionais	
>	Maior que
<	Menor que
=	Igual que
>=	Maior ou igual que
<=	Menor ou igual que
Lóxicos	
& ou Y	conxunción (and ou y)
ou O	Disxunción (or ou o)
~ ou NO	Negación (not ou non)
Alxebraicos	
+	Suma
-	Resta
*	Multiplicación
/	División
^	Potencia
% ou MON	Módulo (resto da división enteira)

e) Delimitadores: Son elementos que axudan a confeccionar as expresións separando ou coutando os datos. Na seguinte táboa temos os fundamentais.

DELIMITADOR	NOME	FUNCIÓN
"	Comiñas	Para introducir constantes alfanuméricas
'	Apóstrofo	O mesmo que as comiñas
()	Paréntese	Couta unha subexpresión
,	Coma	Separa un conxunto de datos
	Espazo	Separa palabras
	Salto de liña	Separa instrucións
[]	Corchetes	Delimita un subíndice dun arranxo

f) Expresións lóxicas: Os operadores lóxicos NOT (~), AND (&) e OR (|) operan as variables en función dos valores verdadeiro e falso. Así no operador NOT ou NO a variable toma o valor contrario, sendo verdadeiro se a variable é falsa e falso se a variable é verdadeira. O operador AND ou Y tomará resultados verdadeiros cando todas as variables coas que opera son verdadeiras, en caso contrario será falso. Por último o operador OR ou O tomará resultados verdadeiros cando algunha das variables é verdadeira, e só toma valor falso se as variables son falsas todas.

O que acabamos de ler represéntase habitualmente nas denominadas Táboas de Verdade:

AND (&): $C \leftarrow A \& B$

A	B	C
V	V	V
V	F	F
F	V	F
F	F	F

OR (|): $C \leftarrow A | B$

A	B	C
V	V	V
V	F	V
F	V	V
F	F	F

NOT (~): $B \leftarrow \sim A$

A	B
V	F
F	V

Exemplo: Operadores; vemos neste exemplo como sería o algoritmo da seguinte operación

$$\left[4 + 2 \times 2 - \sqrt[2]{20 - 4} + 8 / 2 - 6 \right]$$

Pseudocódigo:

Proceso precedencias

```
calcular <- 4+6*2-(20-4)^(1/2)+8/2-6
```

```
Escribir calcular
```

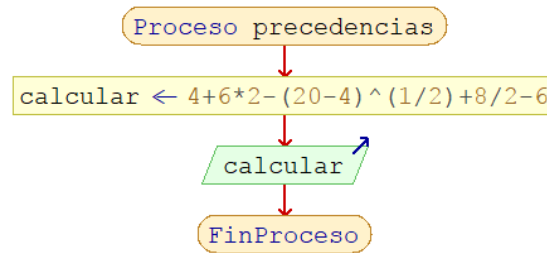
FinProceso

Resultado:

*** Ejecución Iniciada. ***

*** Ejecución Finalizada. ***

Diagrama de flujo:



Ejemplo: expresión lóxica

Exemplo do emprego de operadores lóxicos con datos numéricos.

Dados tres números, A=158, B=75, C=326 e D=15; determinar a veracidade das seguintes afirmacións:

- C é menor que A e D é menor que B. (variable E)
- C é menor que A ou D é menor que B. (variable F)
- O contrario do anterior. (variable G)

Pseudocódigo:

Proceso Operadores

```

A<-158
B<-75
C<-326
D<-15
E<-C<A Y D<B

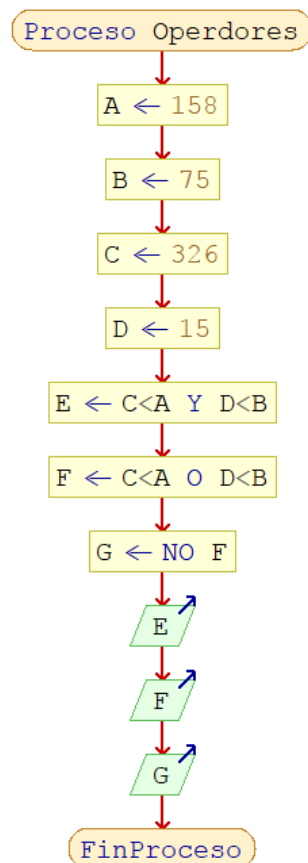
F<-C<A O D<B
G<-NO F
Escribir E
Escribir F
Escribir G
  
```

FinProceso

Resultado:

```
*** Ejecución Iniciada. ***  
FALSO  
VERDADERO  
FALSO  
*** Ejecución Finalizada. ***
```

Diagrama de flujo:



g) Función: Unha función é un subproceso ou subalgoritmo formado por unha secuencia de instrucións cunha tarefa específica. No caso de PSeint as funcións que ten o programa son case que todas matemáticas. Para empregalas colocamos o nome da función seguido dos argumentos para a mesma entre parénteses. Pódense empregar no interior de calquera algoritmo.

Función	Significado
RC(X) o RAIZ(X)	Raíz Cadrada de X

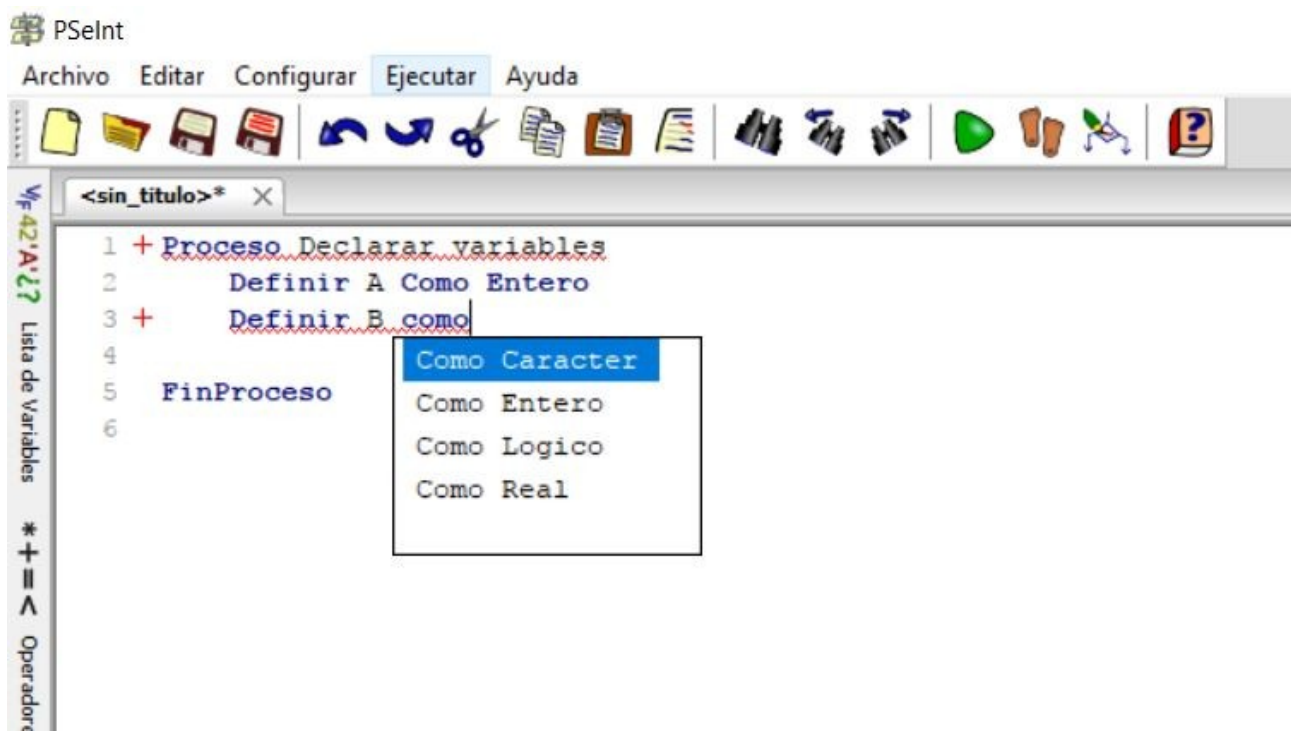
Función	Significado
ABS(X)	Valor Absoluto de X
LN(X)	Logaritmo Natural de X
EXP(X)	Función Exponencial de X
SEN(X)	Seo de X
COS(X)	Coseo de X
TAN(X)	TanXente de X
ASEN(X)	Arcoseo de X
ACOS(X)	Arcocoseo de X
ATAN(X)	Arcotanxente de X
TRUNC(X)	Parte enteira de X
REDON(X)	Enteiro máis próximo a X
AZAR(X)	Enteiro aleatorio no rango [0;x-1]
ALEATORIO(A,B)	Enteiro aleatorio no rango [A;B]
LONGITUD(S)	Cantidade de caracteres da cadea S
MAYUSCULAS(S)	Retorna unha copia da cadea S con todos os seus caracteres en maiúsculas
MINUSCULAS(S)	Retorna unha copia da cadea S con todos os seus caracteres en minúsculas
SUBCADENA(S,X,Y)	Retorna unha nova cadea que consiste na parte da cadea S que vai desde a posición X ata a posición E (incluíndo ambos os extremos). As posicións utilizan a mesma base que os arranxos, polo que a primeiro letra será o 0 ou a 1 de acordo ao perfil da linguaxe utilizada.
CONCATENAR(S1,S2)	Retorna unha nova cadea resulta de unir as cadeas S1 e S2.

Función	Significado
CONVERTIRANUMERO(X)	Recibe unha cadea de caracteres que contén un número e devolve unha variable numérica co mesmo.
CONVERTIRATEXTO(S)	Recibe un real e devolve unha variable numérica coa representación como cadea de caracteres do devandito real.

h) Variables: Unha variable é unha porción de memoria onde se pode almacenar información. Como o seu nome indica, o valor almacenado nunha variable pode ir cambiando a medida que o programa avanza. Nun algoritmo faise referencia a unha variable mediante un identificador (o nome da variable). Un identificador debe comezar con letras, e pode conter só letras, números e o guión baixo. Non pode conter nin espazos nin operadores, nin coincidir cunha palabra reservada ou función da linguaxe, para non xerar ambigüidade.

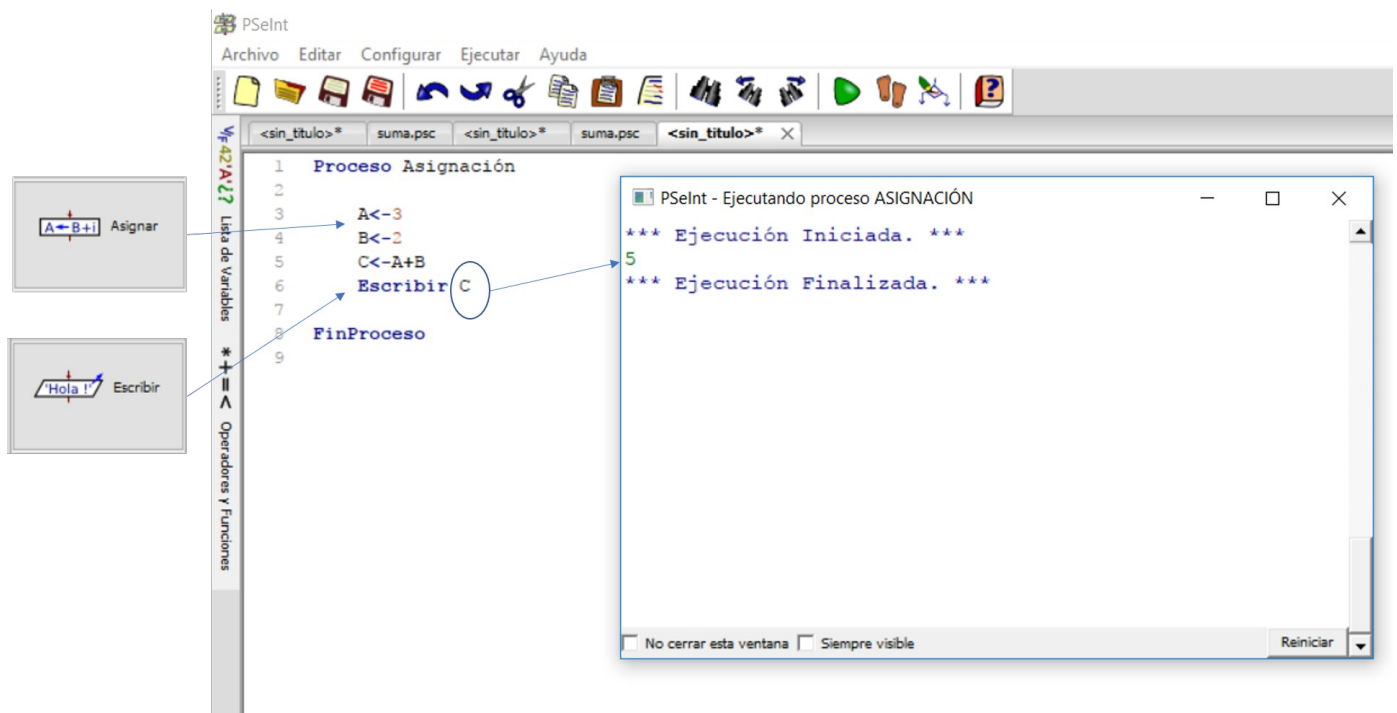
Declarar variables

En PSeInt as variables teñen un tipo de dato asociado, polo que durante a execución do algoritmo unha variable deberá gardar datos sempre do mesmo tipo. Pode declararse directamente o tipo dunha variable coa palabra clave Definir da seguinte forma:



Podemos declarar a variable coa instrución Definir como carácter, número enteiro, número real ou como lóxico.

Asignar variables: Se asignamos un valor nunha variable que non existe, esta crearase para o tipo de datos dos que forme parte o valor. Se a variable xa existía, esta toma o novo valor, perdendo o vello.



Unha vez inicializada, a variable pode utilizarse en calquera expresión (para realizar un cálculo nunha asignación, para mostrar en pantalla, como condición nunha estrutura de control, etc.)

i) Matrices ou arranxos: Os arranxos son un conxunto de datos do mesmo tipo ordenados nunha estrutura que ten un determinado nome identificador. Debemos primeiro dimensionar a estrutura para determinar cantos elementos poden almacenarse e como accederemos a eles.

Dimesion `<identificador> (<max1>, ..., <maxN>);`

Exemplo: Neste exemplo que atopamos na axuda da aplicación PSeint podemos ver a estrutura dun algoritmo con arranxos. O exemplo amosa como cargar os datos nun arranxo para atopar o maior deles:

```
// Busca los dos mayores de una lista de N datos
```

Proceso Mayores

```
// primero se declara un arreglo de 200 elementos
```

```

Dimension datos[200]

// luego se pide al usuario ingresar la cantidad de datos,
// que no debera ser mayor a 200
Escribir "Ingrese la cantidad de datos (de 2 a 200):"
Leer n

// se leen uno por uno los 200 datos y se los guarda en el
arreglo
Para i<-1 Hasta n Hacer
    Escribir "Ingrese el dato ",i,":"
    Leer datos[i]
FinPara

// se comparan los dos primeros y se toman como may1 (el mayor de
los
// dos) y may2 (el segundo mayor).
Si datos[1]>datos[2] Entonces
    may1<-datos[1]
    may2<-datos[2]
SiNo
    may1<-datos[2]
    may2<-datos[1]
FinSi

// se recorren los demas elementos buscan si hay alguno mayor que
may1 o may2
Para i<-3 Hasta n Hacer
    Si datos[i]>may1 Entonces // si hay un valor mayor que
may1
        may2<-may1 // como may1 era el más grande, pasa a
estar en segundo lugar
        may1<-datos[i] // y el nuevo dato toma el primer
puesto (mayor de todos)

```

```

        SiNo // si no era mas grande que may1, todavia puede ser
mas grande que may2

        Si datos[i]>may2 Entonces // si supera al segundo
mayor que teniamos

            may2<-datos[i] // se lo guarda como segundo
mayor

        FinSi

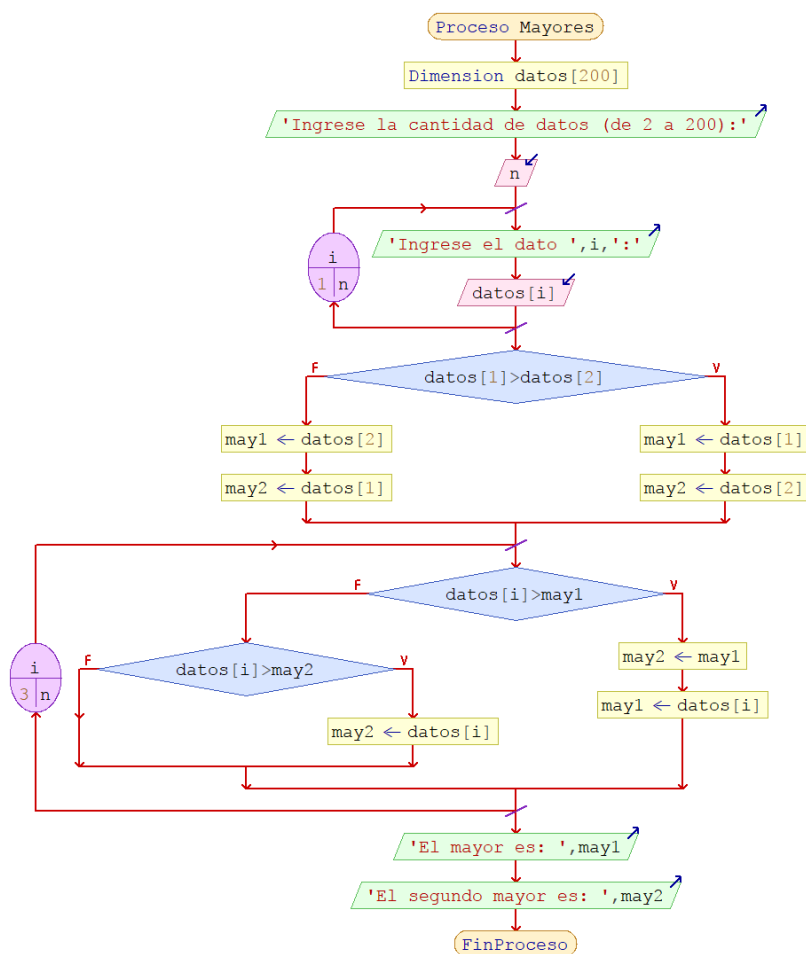
    FinSi

FinPara

// se muestran los resultados
Escribir "El mayor es: ",may1
Escribir "El segundo mayor es: ",may2

FinProceso

```



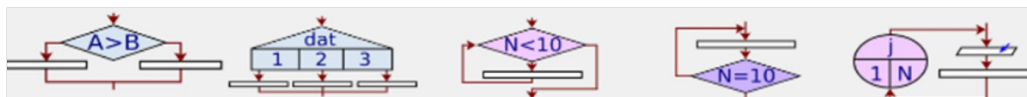
2.- Estruturas de control

Neste apartado coñeceremos como traballar coas estruturas de control (secuencial, condicional e repetitiva) en PSeint. Na imaxe temos as iconas que nos permiten crear as diferentes estruturas de control tanto en pseudocódigo como en fluxograma

Pseudocódigo



Fluxograma



a) Estrutura secuencial

As accións que permiten definir unha estrutura secuencial no algoritmo con PSeint son as que veñen a continuación



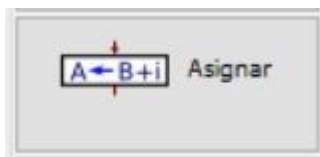
Asignación: A instrución de asignación permite almacenar un valor nunha variable. A sintaxe é a seguinte

```
<variable> <- <expresión> ;
```

Ao executarse a asignación, primeiro aválase a expresión da dereita e logo asígnase o resultado á variable da esquerda. O tipo da variable e o da expresión deben coincidir.

Se a variable da esquerda non existía previamente á asignación, créase. Se a variable existía pérdese o seu valor anterior e toma o valor novo. Os contidos das variables que interveñen na expresión da dereita non se modifican.

En PSeint podemos premer o botón correspondente:



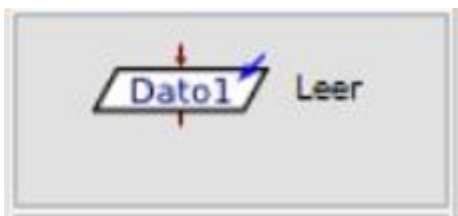
Lectura: A instrución Ler permite ingresar información dende fóra, polo usuario.

```
Ler <variable1> , <variable2> , ... , <variableN> ;
```

Esta instrución toma valores dende o exterior, mediante a súa introdución no teclado, e asigna eses valores ás variables mencionadas. Poden incluírse unha ou máis variables.

Se unha variable onde se debe gardar o valor lido non existe, créase durante a lectura. Se a variable existe pérdese o seu valor anterior xa que tomará o valor novo.

En PSeint podemos premer o botón correspondente:



Escritura: A instrución Escribir permite mostrar valores ao exterior.

```
Escribir <expr1> , <expr2> , ... , <exprN> ;
```

Esta instrución informa dos valores obtidos de avaliar as expresións 1, 2, 3.... Pode incluír unha ou máis expresións, e amosará polo tanto un ou máis valores.

Se hai máis dunha expresión, escríbense unha a continuación da outra sen separación, polo que o algoritmo debe explicitar os espazos necesarios para diferenciar dous resultados se así o require.

Se nalgún punto da liña están as palabras clave SIN SALTAR ou SIN BAJAR os valores amósanse na pantalla, pero non se avanza á liña seguinte, de modo que a próxima acción de lectura ou escritura continuará na mesma liña. En caso contrario, engádese un salto de liña logo das expresións mostradas.

```
Escribir Sen Saltar <expr1> , <expr2> , ... , <exprN> ;
```

```
Escribir <expr1> , <expr2> , ... , <exprN> Sen Saltar;
```

Outras accións secuenciais

A instrución Borrar Pantalla (ou Limpiar Pantalla) permite borrar a pantalla e colocar o cursor na esquina superior esquerda.

```
Borrar Pantalla;
```

A instrución Esperar Tecla detén o algoritmo ata que premamos unha tecla calquera do teclado.

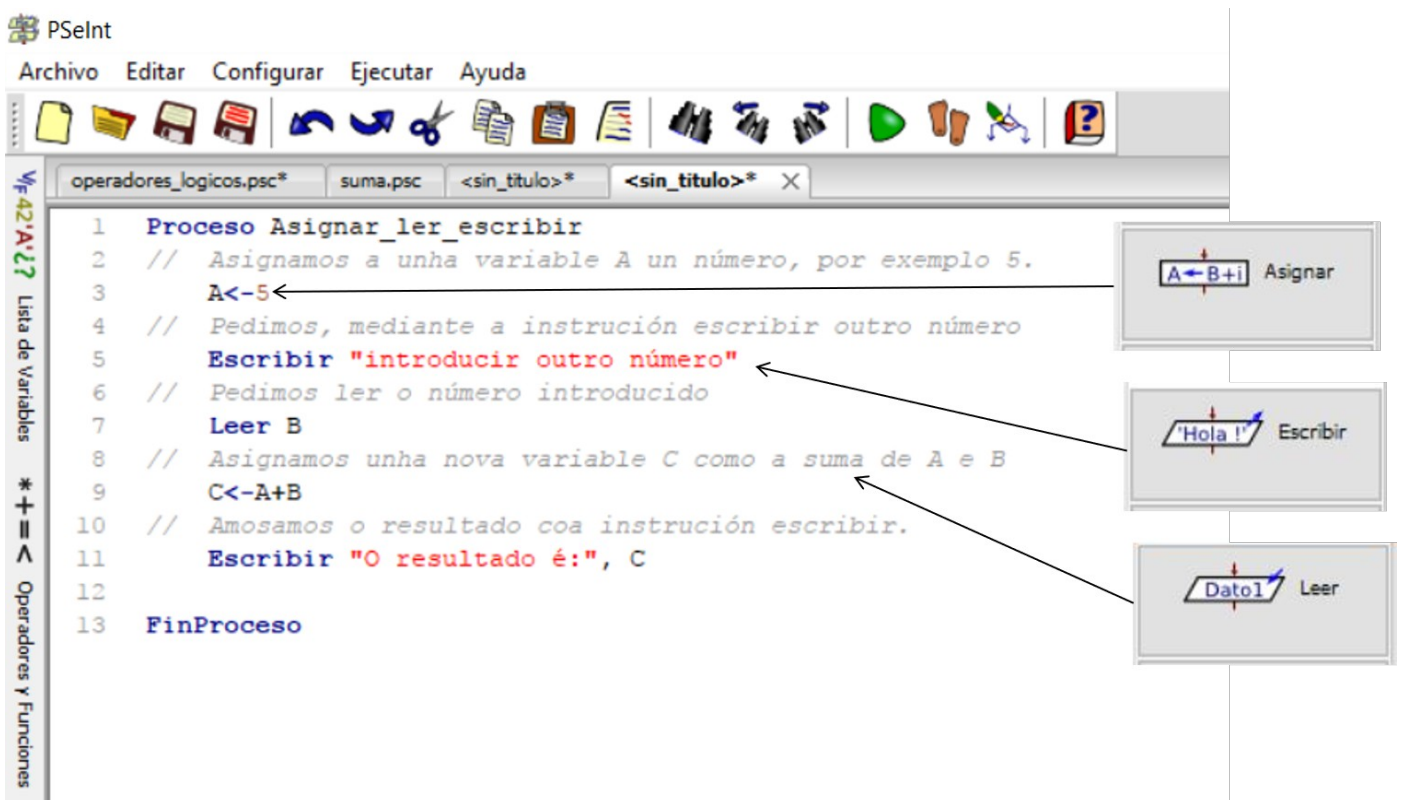
Esperar Tecla;

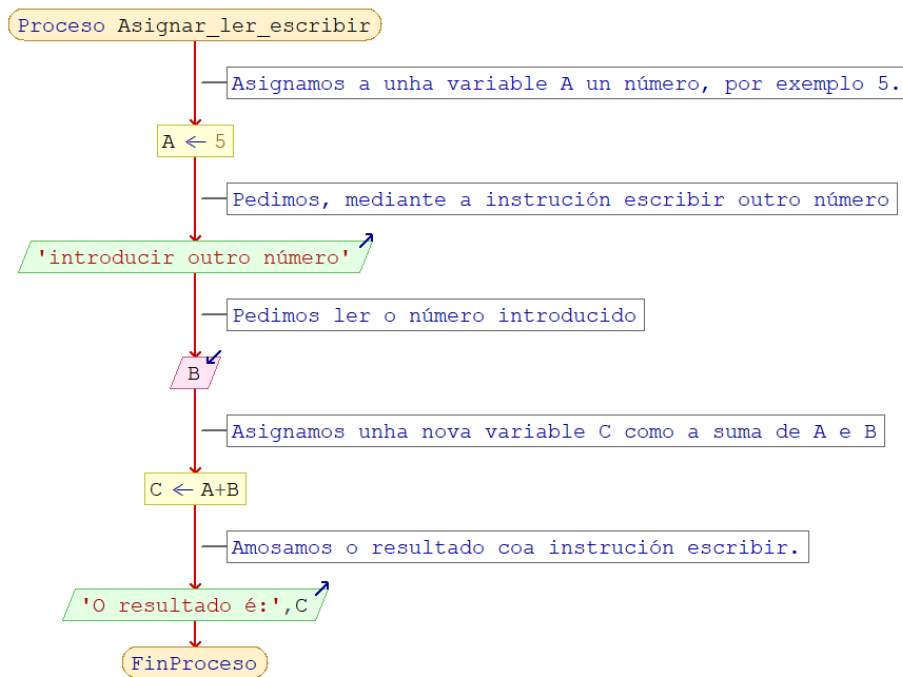
A instrución Esperar tamén pode utilizarse para parar o algoritmo durante un intervalo de tempo predefinido, indicando a continuación da palabra clave a lonxitude e unidade do devandito intervalo. As unidades válidas son Segundos e Milisegundos.

Esperar 3 Segundos;

Exemplo: Realizamos de novo un algoritmo da suma de dous números para observar as instrucións asignar, ler e escribir.

Pseudocódigo





Execución:

```

PSeInt - Ejecutando proceso ASIGNAR_LER_ESCRIBIR
*** Ejecución Iniciada. ***
introducir outro número
> 3
O resultado é:8
*** Ejecución Finalizada. ***
  
```

b) Estrutura condicional

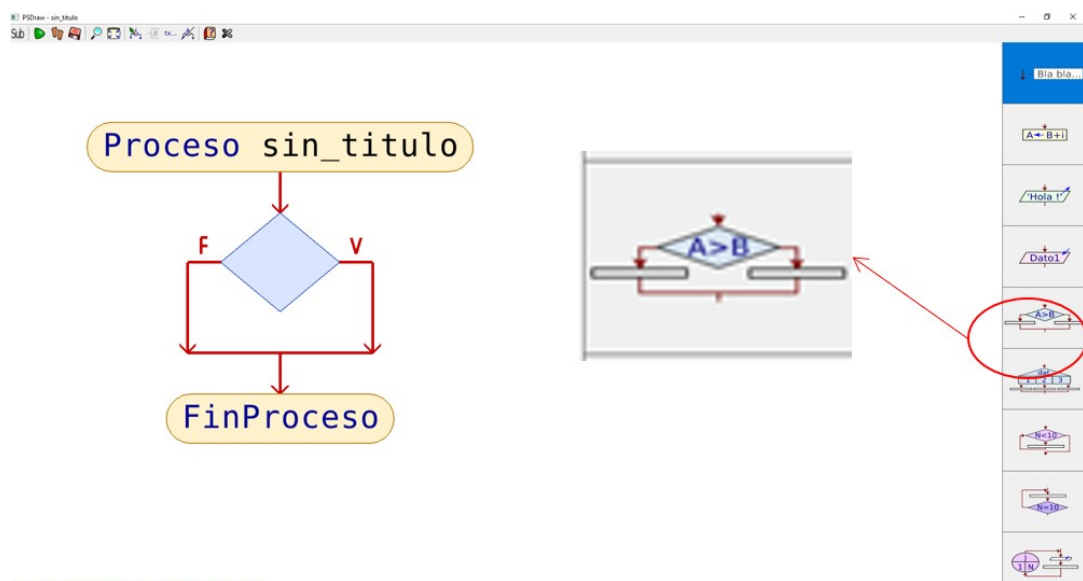
b.1) Se....entón: Esta instrución de PSeint vai nos valer para o condicional simple e o condicional dobre. Se empregamos todos os campos sería un condicional dobre: se ocorre a condición, entón temos esta acción. Se non ocorre temos estoutra. Convertamos este condicional dobre en simple eliminando a parte de Senón: Se ocorre a condición ocorre esta acción. Se non ocorre non pasa nada (continúa o fluxo de datos).

1 **Proceso** sin_titulo
 2 **Si** expresion_logica **Entonces**
 3 acciones_por_verdadero
 4 **SiNo**
 5 acciones_por_falso
 6 **Fin Si**
 7 **FinProceso**
 8

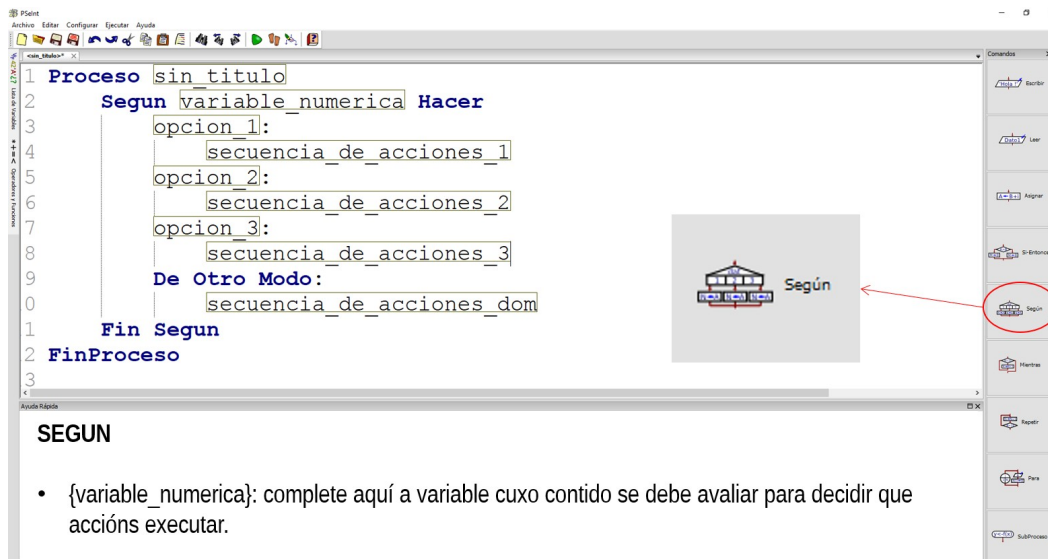
SE - ENTÓN - SENÓN

- {expresion_loxica}: complete aquí a condición para avaliar para decidir que grupo de accións se deben executar.

- {accións_por_verdadeiro}: complete aquí as accións que se deben executar se a condición ({expresion_loxica}) resulta verdadeira.
- {accións_por_falso}: complete aquí as accións que se deben executar se a condición ({expresion_loxica}) resulta falsa. Se non desexa realizar nada neste caso, elimine esta liña e a anterior (SENÓN).



b.2) Segundo: Esta instrucción corresponde co condicional múltiple.

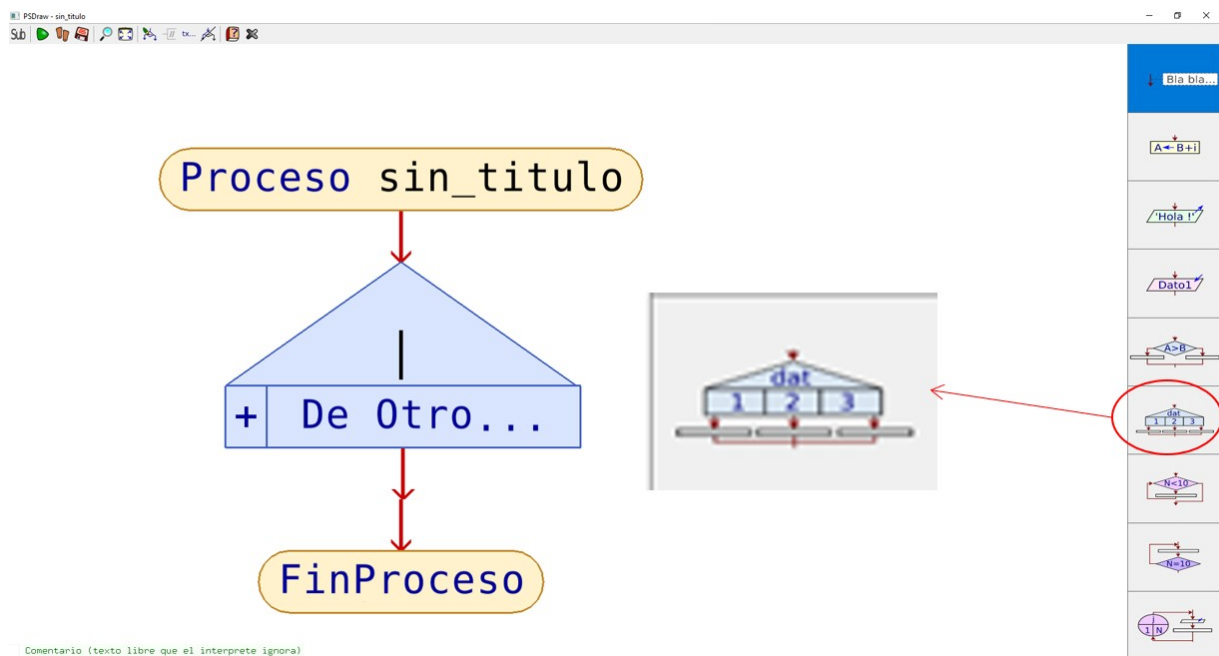


```

1 Proceso sin_titulo
2   Segun variable_numerica Hacer
3     opcion_1:
4       secuencia_de_acciones_1
5     opcion_2:
6       secuencia_de_acciones_2
7     opcion_3:
8       secuencia_de_acciones_3
9     De Otro Modo:
10      secuencia_de_acciones_dom
11   Fin Segun
12 FinProceso
  
```

SEGUN

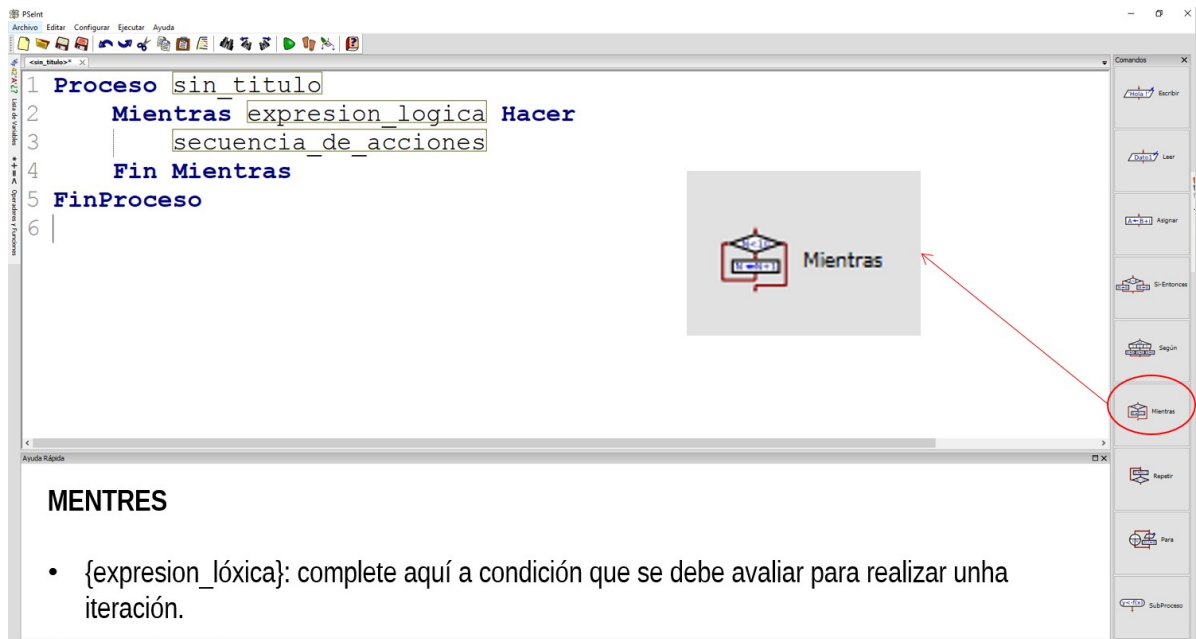
- {variable_numerica}: complete aquí a variable cuxo contido se debe avaliar para decidir que accións executar.
- {opcion_n} e {secuencia_de_accions_n}: reemplaze {opcion_n} por uns posible valores para {variable_numerica} e complete {secuencia_de_accions_n} coas instrucións para executar no caso de que o valor coincida. Pode agregar {opcion_n} e {secuencia_de_accions_n} tantas veces como sexa necesario para cubrir os casos que correspondan. Pode asociar máis dunha opción a un mesmo grupo de accións colocando en {opcion_n} a lista de valores separados por comas.
- {secuencia_de_accions_dom}: complete aqui o conxunto de instrucións a executar se o valor de {variable_numerica} non coincide con ningunha das opcións anteriormente expostas. Se non desexa incluír un conxunto de accións para estes casos elimine esta línea e a anterior (Doutro xeito).



b.3) Estructura anidada: Para crear unha estrutura anidada en PSeint empregamos a instrucción SE ... ENTÓN ... de forma sucesiva. Se cumpre unha condición, entón prodúcese unha acción, senón temos outra condición.

c) Estructura repetitiva

c.1) **Mientras:** A estrutura repetitiva MENTRES correspóndese coa instrución MENTRES de Pseint



The screenshot shows the Pseint IDE interface. The code editor contains the following code:

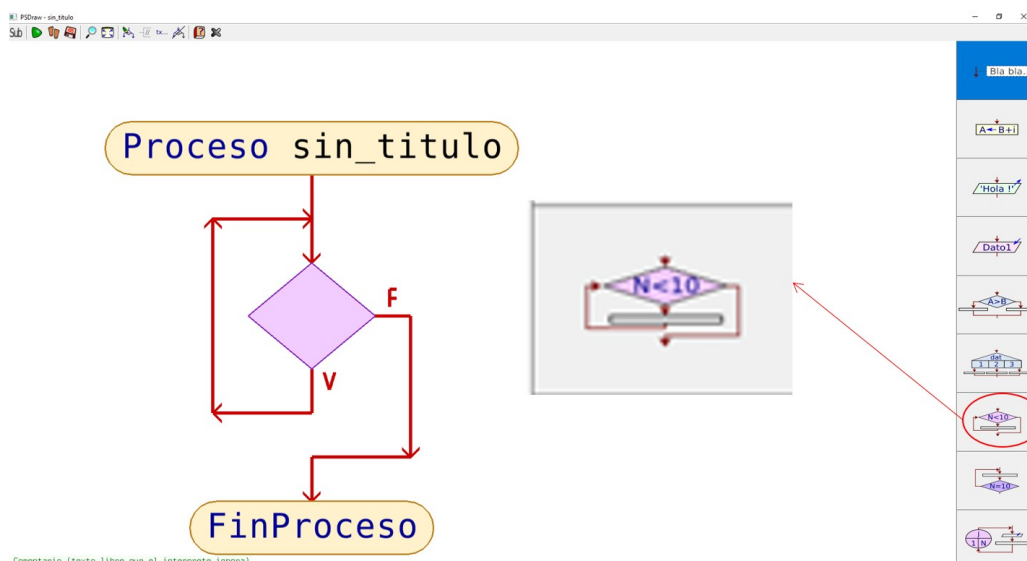
```
1 Proceso sin_titulo
2   Mientras expresion_logica Hacer
3     secuencia_de_acciones
4   Fin Mientras
5 FinProceso
6
```

The 'Mientras' command is highlighted in the toolbar on the right. Below the code editor, the 'Ayuda Rápida' (Quick Help) window is open, displaying the 'MENTRES' command details.

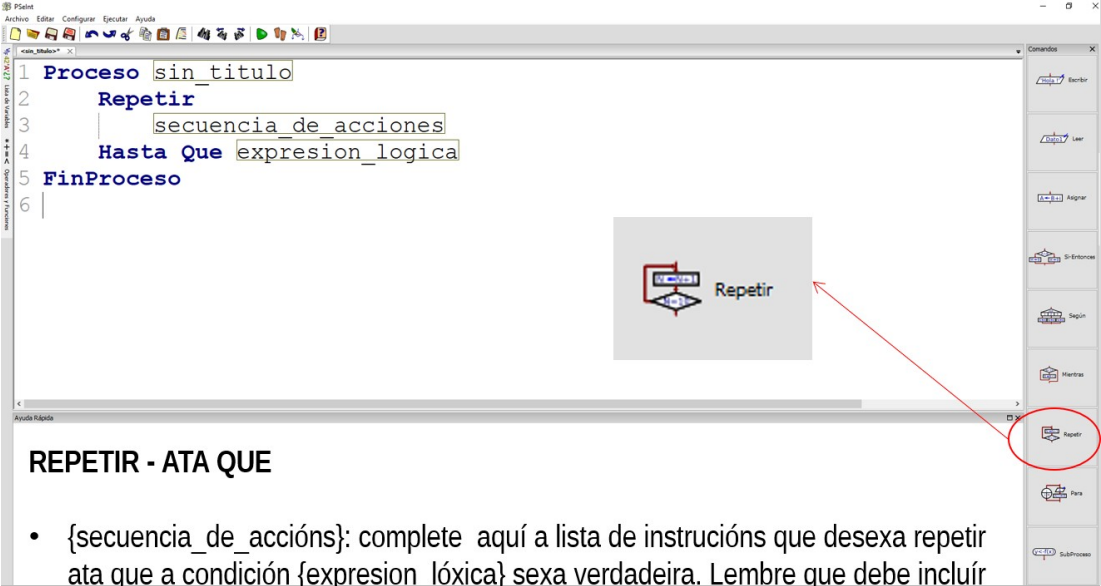
MENTRES

- {expresion_lógica}: complete aquí a condición que se debe avaliar para realizar unha iteración.

- {secuencia_de_accións}: complete aquí a lista de instrucións que desexa repetir mentres a condicion {expresion_lógica} sexa verdadeira. Lembre que debe incluír aquí algunha acción que poida modificar dita condición.



c.2) Repetir ata que: Esta instrución correspóndese coa estrutura repetitiva FACER MENTRES.



The screenshot shows the PSeint IDE interface. The code editor on the left contains the following code:

```

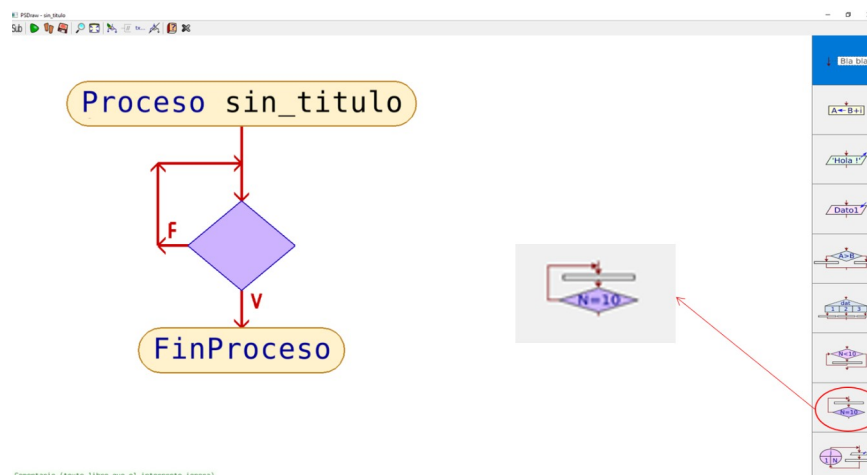
1 Proceso sin_titulo
2   Repetir
3     secuencia_de_acciones
4   Hasta Que expresion_logica
5 FinProceso
6

```

On the right, the 'Comandos' (Commands) palette is visible. The 'Repetir' (Repeat) command is highlighted with a red circle. A red arrow points from this command to a diagram of the 'Repetir' structure in the center of the screen.

REPETIR - ATA QUE

- {secuencia_de_accions}: complete aquí a lista de instrucións que desexa repetir ata que a condición {expresion_lóxica} sexa verdadeira. Lembre que debe incluír aquí algunha acción que poida modificar dita condición.
- {expresion_lóxica}: complete aquí a condición que se debe avaliar para determinar se se continúa iterando.



c.3) Para: O nome da instrución en PSeint chámase tamén PARA.

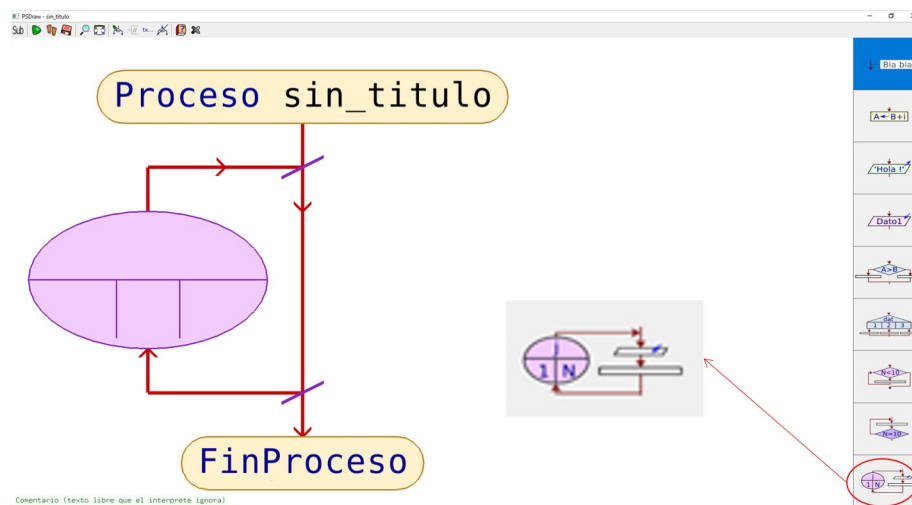
The screenshot shows the PSeint software interface. In the main command window, the following code is entered:

```
1 Proceso sin_titulo
2   Para variable_numerica<-valor_inicial Hasta valor_final Con Paso paso Hacer
3     secuencia_de_acciones
4   Fin Para
5 FinProceso
6
```

To the right, a vertical toolbar contains various icons. The 'Para' icon, which depicts a loop, is circled in red. A red arrow points from this icon to a larger, semi-transparent version of the 'Para' icon displayed in the center of the workspace.

PARA

- {variable_numerica}: complete aquí o identificador da variable que utilizará para iterar.
- {valor_inicial}: introduza aquí o valor de {variable_numerica} dende o cal se comezará a iterar.
- {valor_final}: introduza aquí o valor final de {variable_numerica} ata o cal se continuará iterando.
- {paso}: introduza aquí o incremento que se debe realizar a {variable_numerica} en cada iteración.
- {secuencia_de_accions}: complete aqui a lista de instrucións que desexa repetir en cada iteración.



Ej.1 - Exemplo 1: Un algoritmo sinxelo que nos pregunta o nome e si queremos continuar.

```
Proceso Saudo    //Bucle repetir
    Repetir

        Escribir "Ola, Como te Chamas?";
        Leer nombre;
        Escribir "Ola, ",nombre," Que tal estás?";

        Escribir "Queres comezar de novo? S/N";
        Leer seguir;

        Mientras seguir<>"n" y seguir<>"N" y seguir<>"s" y
seguir<>"S"  Hacer
            Escribir "Tes que escribir S ó N.";
            Leer seguir;
        FinMientras

        Hasta Que seguir = "n" o seguir = "N"

FinProceso
```

Ej.2 - Exemplo 2: Outro sinxelo exemplo. Neste caso, ademais de preguntarnos o nome preguntáanos tamén a hora para personalizar o saúdo. Emprégase a estrutura condicional SI aninada e o operador lóxico Y (&).

```
//Programa que visualiza un saludo distinto en función da hora
//
//Bos días: 6-12h
//Boa tarde: 12 - 20h
//Boas noites: 20 - 6h

Proceso SaludoSHora
    Escribir "Introduce a hora (0-24h)";
    Leer hora;
    Escribir "Cal é o teu nome?"
    Leer N
```

```

        Si (hora>6) Y (hora<=12) Entonces
            Escribir "Bos dias", N;
//          Acompañamos o texto entre "", co nome da variable N,
separados con comas.
        Sino
            Si (hora>12) Y (hora<=20) Entonces
                Escribir "Boa tarde", N;
            Sino
                Si ((hora>20) Y (hora<24)) O ((hora>=0) Y
(hora<=6)) Entonces
                    Escribir "Boas noites" N;
                Sino
                    Escribir "A hora introducida non é
correcta", N;
                FinSi
            FinSi
        FinSi
FinProceso

```

Ej.3 - Exemplo 3: Un algoritmo que calcula o maior de dous números mediante unha estrutura condicional simple (se-entón).

Proceso Maior

```

        Escribir "Introduce dous números";
        Leer mayor,A;

        Si A>mayor Entonces
            mayor<-A;    // En la variable mayor almaceno el
valor del numero más alto.
        FinSi
        Escribir "El mayor es ",mayor;
FinProceso

```