

EXERCICIOS

INICIACIÓN A PROGRAMACIÓN CON PSEINT

INSTALAR PSEINT



[Descargar Instalador para Microsoft Windows](#) (exe - 9.0MB)

[Descargar Portable para Microsoft Windows](#) (zip - 12MB)

[Descargas para otras plataformas...](#)

PSeint é un software distribuído baixo unha licencia GNU GENERAL PUBLIC LICENSE v2. que funciona como unha ferramenta útil para a comprensión dos principios básicos da programación informática.

PSeint permite crear as instrucións dos programas en pseudocódigo para familiarizarse coas estruturas secuencias, condicionais e repetitivas da programación, así como o uso correcto das variables e os seus tipos.

PSeint permite seleccionar distintos perfís que condicionan o forma en que se mostra o pseudocódigo ou como se mostran as asignacións. Nos imos fixar as seguintes opcións da linguaxe.

Opciones del Lenguaje



- ☒ No permitir utilizar variables o posiciones de arreglos sin inicializar
- ☒ Obligar a definir los tipos de variables
- ☒ Controlar el uso de ; al final de sentencias secuenciales
- ☒ Permitir concatenar variables de texto con el operador +
- ☒ Habilitar funciones para el manejo de cadenas
- ☒ Permitir las palabras Y, O, NO y MOD para los operadores &, |, ~ y %
- ☒ Utilizar indices en arreglos y cadenas en base 0
- ☐ Permitir utilizar variables para dimensionar arreglos
- ☒ Permitir asignar con el signo igual (=)
- ☒ Permitir definir funciones/subprocesos
- ☐ Utilizar sintaxis flexible
- ☒ Permitir condiciones en lenguaje coloquial
- ☒ Limitar la estructura Según a variables de control numéricas

As opcións que non se mostran en pantalla están desactivadas

Nota: Se queremos ver o organigrama dun algoritmo podemos utilizar a opción do menú Archivo>Diagrama de flujo o F7

ESTRUTURAS SECUENCIAIS

1. Algoritmo que mostra por pantalla unha mensaxe.

```
Proceso Primer_Programa
    Escribir "Aquí estamos";
FinProceso
```

2. Algoritmo que permita introducir un nome por teclado e escribilo en pantalla.

Análise:

- *Que datos de entrada imos necesitar?:* O nome de usuario (cadea).
- *Que datos de saída temos que obter?:* O nome de usuario introducido.
- *Que procesos imos ter que realizar?:* Ler o nome de usuario e mostralo por pantalla.

Deseño:

1. Ler nome.
2. Escribir o nome introducido.

Implementación:

```
Proceso Saludar
    Definir nombre Como Cadena;
    Escribir "Dime tu nombre:";
    Leer nombre;
    Escribir "Hola ", nombre;
FinProceso
```

3. Calcular o perímetro e a área dun rectángulo dada a súa base e altura.

Análise:

- *Que datos de entrada imos necesitar?:* A base (real) e a altura (real) do triángulo.
- *Que datos de saída temos que obter?:* O perímetro (real) e a área (real) do rectángulo.
- *Que procesos imos ter que realizar?:*
 - *Definir as variables:*
 - de entrada: base e altura de tipo real.
 - de saída: perímetro e área de tipo real.
 - *Proceso:* Ler os datos de entrada e calcular o perímetro e a área do triángulo un función dos datos introducidos.

Deseño:

1. Ler a base e a altura por teclado.
2. Calcular o perímetro ($2 \cdot \text{base} + 2 \cdot \text{altura}$).
3. Calcular a área ($\text{base} \cdot \text{altura}$).
4. Mostrar perímetro e área.

Implementación:

```
Proceso Rectangulo
    Definir base, altura, perimetro, area Como Real;
    Escribir "Introduce la base:";
    Leer base;
    Escribir "Introduce la altura:";
    Leer altura;
    perimetro  $\leftarrow 2 * \text{base} + 2 * \text{altura}$ ;
    area  $\leftarrow \text{base} * \text{altura}$ ;
    Escribir "El perímetro es ", perimetro, " y el área es ", area;
FinProceso
```

4. Calcular o perímetro e a área dun rectángulo.
5. Ler o ano de nacemento e calcular cantos anos vai cumprir no ano actual.

ESTRUTURAS CONDICIONAIS

1. Algoritmo que pida dous números por pantalla e indique se o primeiro é maior que o segundo.

Análise:

- *Que datos de entrada imos necesitar?:* Dous números (enteiros).
- *Que datos de saída temos que obter?:* A información de si o primeiro número é maior que o segundo ou non.
- *Que procesos imos ter que realizar?:*
 - Ler os dous números .
 - Comparar os dous números lidos para saber se o primeiro é maior que o segundo.
 - Mostrar a información por pantalla

Deseño:

1. Ler o número 1 e o número 2.
2. Si o número 1 é maior que o número 2 mostrar: "O primeiro é maior".
3. En caso contrario mostrar "O primeiro non é maior que o segundo".

Implementación:

```
Proceso CalcularMayor
    Definir num1, num2 como Entero;
    Escribir "Dime el número 1:";
    Leer num1;
    Escribir "Dime el número 2:";
    Leer num2;
    Si num1>num2 Entonces
        Escribir "Número 1 es mayor que número 2";
    SiNo
        Escribir "Número 1 NO es mayor que número 2";
    FinSi
FinProceso
```

2. Algoritmo que pida un número e diga si é positivo, negativo ou 0.

Análise:

- *Que datos de entrada imos necesitar?:* Un número (enteiros).
- *Que datos de saída temos que obter?:* A información de si o número é positivo, negativo ou 0.
- *Que procesos imos ter que realizar?:*
 - Ler o número .
 - Mirar se o número é menor, maior ou igual a 0 e mostrar a mensaxe correspondente.

Deseño:

1. Ler o número.
2. Si o número é igual a 0 mostrar a mensaxe correspondente.
3. Senón
 - 3.1. Preguntar si o número é maior que 0 e si é certo mostrar a mensaxe de que é positivo.
 - 3.2. Senón mostrar a mensaxe de que é negativo.

Implementación:

```
Proceso CalcularPositivoNegativoCero
  Definir num como Entero;
  Escribir "Dime el número :";
  Leer num;
  Si num=0 Entonces
    Escribir "Es igual a 0";
  SiNo
    Si num>0 Entonces
      Escribir "Es positivo";
    SiNo
      Escribir "Es negativo";
    FinSi
  FinSi
FinProceso
```

3. Algoritmo que pregunte pola idade do usuario e diga se é maior ou menor de idade.
4. Algoritmo que pida un nome por pantalla. Se o nome introducida é "Alejandro" que mostre a mensaxe "Está bache esperando" e senón que mostre "Eu non te coñezo".
5. Algoritmo que pida un usuario e un contrasinal. Se o nome é "rosa" e o contrasinal é "tururu" mostre a mensaxe: "Entraches no sistema" e senón, mostre "Usuario non autorizado".

ESTRUTURAS REPETITIVAS

1. Algoritmo que pida un número e calcule o seu factorial (O factorial dun número é o produto de todos os enteiros entre 1 e o propio número e represéntase polo número seguido dun signo de exclamación. Por exemplo $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$).

Análise:

- Que datos de entrada imos necesitar?: El número para calcular o seu factorial (enteiro).
- Que datos de saída temos que obter?: O factorial (enteiro) do número introducido.
- Que procesos imos ter que realizar?: Necesito acumular las sucesivas multiplicacións, empezando por 2 ata o número indicado.

Deseño:

1. Ler o número (enteiro) .
2. Inicializar contador a 0 (enteiro) e resultado (enteiro) a 1
3. Mentres o contador sexa \leq num facer:
 - 3.1. resultado=resultado*contador
 - 3.2. contador=contador+1
4. Mostrar resultado

Implementación →

```
Proceso Factorial
  Definir num, contador Como Entero;
  Definir resultado Como Real;
  resultado = 1;
  Escribir Sin Saltar "Dime un número:";
  Leer num;
  contador = 2;
  Mientras contador ≤ num Hacer
    resultado = resultado * contador;
    contador = contador + 1;
  FinMientras
  Escribir "El resultado es ", resultado;
FinProceso
```

2. Crea unha aplicación que faga o mesmo que a do punto 1 pero utilizando unha estrutura FOR:

```

Proceso Factorial
  Definir num, contador Como Entero;
  Definir resultado Como Real;
  resultado ← 1;
  Escribir Sin Saltar "Dime un número:";
  Leer num;
  Para contador←2 Hasta num Hacer
  | resultado ← resultado * contador;
  FinPara
  Escribir "El resultado es ", resultado;
FinProceso

```

3. Algoritmo que pida números hasta que se introduza un cero. Debe imprimir la suma y la media de todos los números introducidos.

Análise:

- *Que datos de entrada imos necesitar?* Números, un acumulador para sumar e un contador. Todos enteros.
- *Que datos de saída temos que obter?* La suma y la media. Todos enteros.
- *Que procesos imos ter que realizar?* Pida números hasta que se introduza un cero. Debe imprimir la suma y la media de todos los números introducidos.

Deseño:

1. Inicializar: suma = 0, cont = 0
2. Repetir
 - 2.1. Leer num
 - 2.2. Acumular: suma = suma + num
 - 2.3. Si num <> 0, → incremento o contador: cont=cont+1
3. Hasta num=0
4. Si cont=0 (No temos introducido ningún número) → media=0
5. SiNo → media=suma/cont;
6. Mostrar suma y media.

Proceso CalcularSumaMedia

```

Definir num Como Entero;
Definir suma, cont como Entero;
Definir media Como Real;
suma = 0;
cont = 0;
Repetir
  Escribir "Número (0 para salir):";
  Leer num;
  suma ← suma + num;
  //Si num=0 non o temos que ter en conta para a media
  Si num≠0 Entonces
  | cont = cont + 1;
  FinSi
Hasta que num=0;
//Si cont=0 non se pode dividir, daría un erro
Si cont≠0 Entonces
| media = suma / cont;
SiNo
| media = 0;
FinSi
Escribir "Suma:", suma;
Escribir "Media:", media;
FinProceso

```

Nota: As condicións de unha estrutura IF/SI ou WHILE/MIENTRAS pode ser composta. Si queremos que dúas ou mais condicións sexan obrigatorias temos que unilas con Y e se queremos que só unha delas sexa obrigatoria temos que unilas con O.

Por exemplo:

1. Condición: Que sexa maior de idade e de Vigo
 $IDADE > 18 \text{ Y } CIDADE = "VIGO"$
2. Condición: Que sexa de Vigo ou de Ourense
 $CIDADE = "VIGO" \text{ O } CIDADE = "OURENSE"$
3. Condición: Que sexa maior de 12 anos e menor de 15
 $IDADE > 12 \text{ Y } IDADE \leq 14$
 $IDADE \geq 13 \text{ Y } IDADE < 15$

4. Algoritmo que pida caracteres e imprima 'VOCAL' se son vocais e 'NON VOCAL' en caso contrario, o programa termina cando se introduce un espazo.

Análise:

- *Que datos de entrada imos necesitar?* Caracteres. Tipo char.
- *Que datos de saída temos que obter?* A información de si son vocais ou non. Mensaxes por pantalla.
- *Que procesos imos ter que realizar?* Ler caracteres un a un ata que se introduza un espazo en branco e por cada un dicir se son vocais ou non.

Deseño

1. Repetir -> Ler carácter hasta que sexa un carácter
2. Mentres o carácter lido non sexa un espazo
3. Si o carácter é A,E,I,O,U o a,e,i,o,u -> Mostrar "É vocal"
4. Si non mostrar "Non é vocal"
5. Repetir -> Ler carácter hasta que sexa un carácter

Proceso VocalConsonante

```

Definir car Como Caracter;
Repetir
    Escribir "Introduce un carácter:";
    Leer car;
Hasta que Longitud(car)=1;
Mientras car!=" " Hacer
    Si Mayusculas(car)="A" o Mayusculas(car)="E" o Mayusculas(car)="I" o Mayusculas(car)="O" o Mayusculas(car)="U" Entonces
        Escribir "VOCAL";
    SiNO
        Escribir "NO VOCAL";
    FinSi
    Repetir
        Escribir "Introduce un carácter:";
        Leer car;
    Hasta que Longitud(car)=1;
FinMientras

```

FinProceso

6. Realizar un algoritmo que pida números (pedirase por teclado a cantidade de números a introducir). O programa debe informar de cantos números introducidos son maiores que 0, menores que 0 e iguais a 0.

```

//#####
//Realizar un algoritmo que pida números (se pedirá por teclado la cantidad de

```

```
//números a introducir). El programa debe informar de cuantos números introducidos
//son mayores que 0, menores que 0 e iguales a 0.
//#####
//Análisis
//Se pide la cantidad de números que se van a leer. Vamos introduciendo números.
//Tenemos que contar los positivos, negativos y 0.
// Datos de entrada:Cantidad de números, los números.
// Información de salida: Cantidad de positivos, de negativos y de 0.
// Variables:cantidad_num,num,cont_positivos,cant_negativos,cantidad_ceros(entero)
//#####
//Diseño
//1.- Inicializo los contadores
//2.- Leer cantidad de números
//3.- Desde 1 hasta cantidad de números
//      4.- Leer num
//      5.- Si num> 0-> incremento cont_positivos
//      6.- Si num< 0-> incremento cont_negativos
//      7.- Si num = 0-> incremento cont_ceros.
// Muestro cont_positivos, cont_negativos, cont_ceros
//#####
```

Proceso ContarNumerosPositivosNegativosYCeros

```
Definir num Como Entero;
Definir cantidad_num,i Como Entero;
Definir cont_negativos,cont_positivos,cont_ceros Como Entero;
cont_negativos <- 0;
cont_positivos <- 0;
cont_ceros <- 0;
Escribir "¿Cuántos números vas a introducir?:";
Leer cantidad_num;
Para i<-1 Hasta cantidad_num Hacer
    Escribir "Número ",i," ";
    Leer num;
    Si num>0 Entonces
        cont_positivos <- cont_positivos + 1;
    Sino
        Si num<0 Entonces
            cont_negativos <- cont_negativos + 1;
        SiNo
            cont_ceros <- cont_ceros + 1;
        FinSi
    FinSi
FinPara
Escribir "Números positivos:",cont_positivos;
Escribir "Números negativos:",cont_negativos;
Escribir "Números igual a 0:",cont_ceros;
FinProceso
```

7. **Escribir un programa que imprima todos os números pares entre dous números que se lle pidan ao usuario. Existe un operador que é % que devolve o resto dunha división enteira, por**

ejemplo 5%2 da 1 e 6%2 da 0; este método é o que temos que utilizar para saber si o número é par ou impar; na condição podemos estabelecer Si num%2=0, então é par.

```
//#####  
//Escribir un programa que imprima todos los números pares entre dos números que  
//se le pidan al usuario.  
//#####  
//Análisis  
//Leemos un número. Si es impar, le sumo 1 y ya es par. Leemos el segundo número.  
//Escribo el número desde el primero hasta el segundo de dos en dos.  
// Datos de entrada: dos números  
// Información de salida: Los números pares que hay entre los dos anteriores  
// Variables: num, num1, num2 (entero)  
//#####  
//Diseño  
//1.- Leer num1, num2  
//2.- Si num1 es impar -> num1=num1+1  
//3.- Desde num1 hasta num2 de 2 en 2  
//      4.- Escribir el número  
//#####
```

Proceso NumerosPares

```
    Definir num,num1,num2 Como Entero;  
    Escribir Sin Saltar "Introduce el número 1:";  
    Leer num1;  
    Escribir Sin Saltar "Introduce el número 2:";  
    Leer num2;  
    Si num1 % 2 = 1 Entonces  
        num1<-num1+1;  
    FinSi  
    Para num<-num1 Hasta num2 Con Paso 2 Hacer  
        Escribir sin saltar num," ";  
    FinPara  
FinProceso
```