

Ciclos de Repetición en Pseint

1

Variables especiales

- **Acumaladores:** son variables numéricas que acumulan un valor y lo guardan en la misma variable, podríamos decir que tiene un concepto implícito, por ejemplo: acumulador de notas de parciales

Ejemplo

acumParciales= acumParciales+nota

- **Contadores:** son variables numéricas que suman 1 y lo guardan en la misma variable

Ejemplo

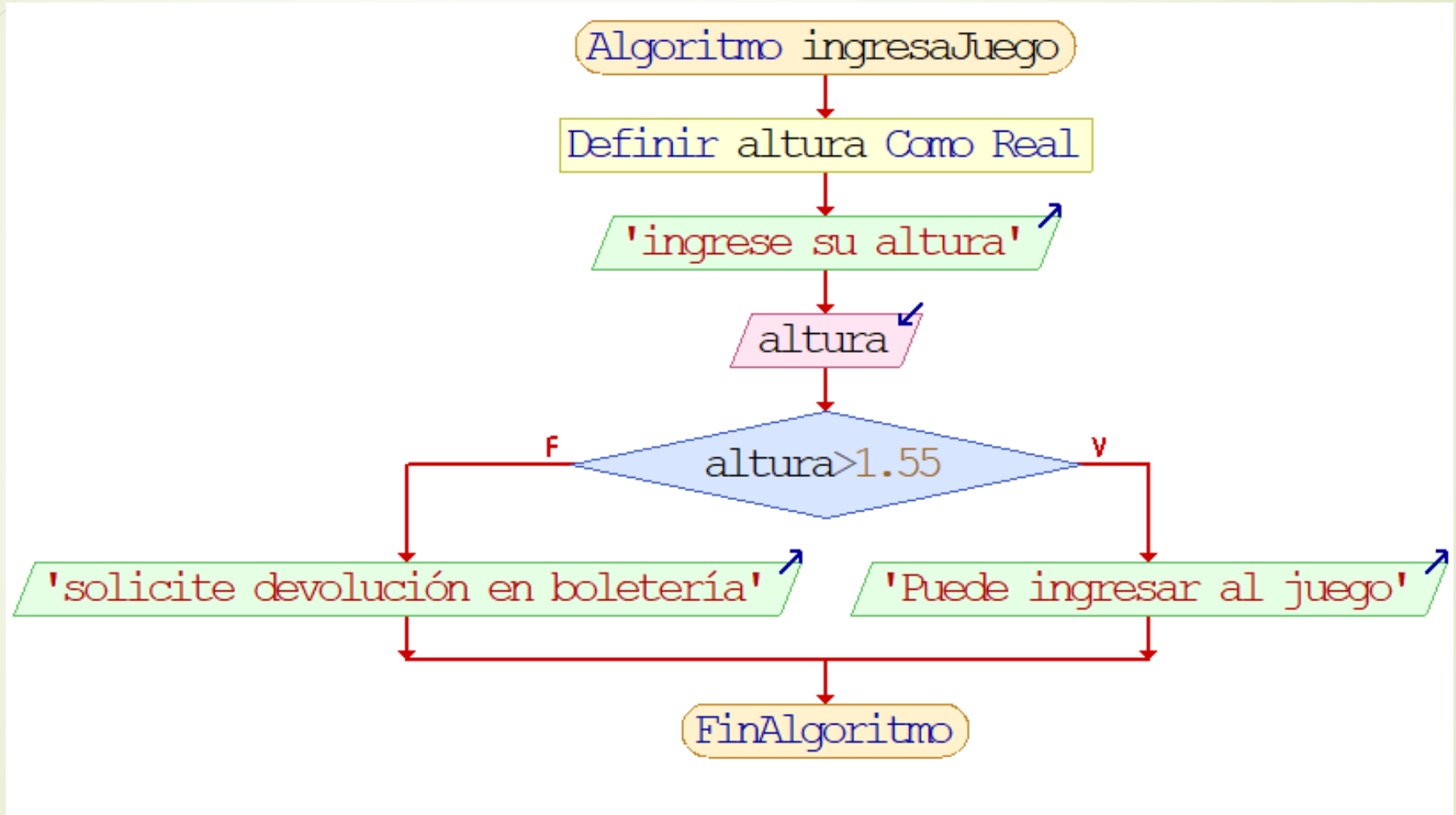
contadorAlumnos= contadorAlumnos+1

3

Tips para programar

- $\text{PROMEDIO} = \text{ACUMULADOR} / \text{CONTADOR}$
- $\text{PORCENTAJE} = (\text{CONTADOR} / \text{ACUMULADOR}) * 100$
- Similar para proporciones

Recordamos cómo se ve las secuencias en un diagrama de estructuras alternativas



Estructuras repetitivas o Ciclos de repetición

Repetir
MIENTRAS

Repetir
HASTA QUE

Repetir
PARA
valor Inicial – valor final
con paso

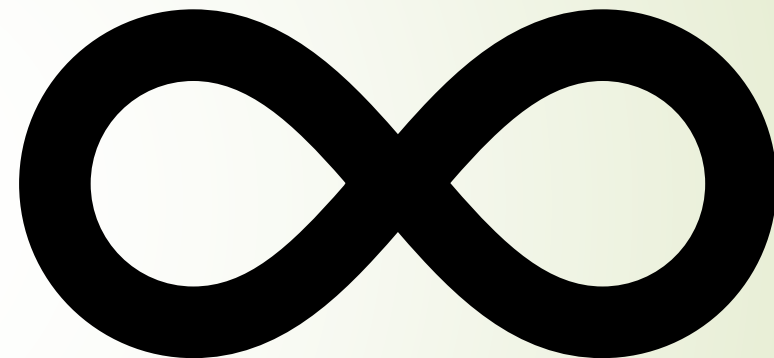
Cómo se ven las estructuras repetir



Condición de fin

Es muy importante analizar la **Condición de Fin**, si nunca se cumple entonces el ciclo de repetición continua infinitamente!!

¡¡ NO queremos eso !!



Estructuras Ciclos de repetición

Repetir MIENTRAS ejemplo

Algoritmo repetirMientrasSaldo

Definir saldo,extraccion Como Real

Definir continuar Como Caracter

continuar <- 'S'

saldo <- 100

Mientras saldo>0 Y continuar='S' Hacer

 Escribir 'ingrese monto de la extracción'

 Leer extraccion

 saldo <- saldo-extraccion

 Escribir 'saldo: ',saldo

 Escribir 'Continuar operando? S/N'

 Leer continuar

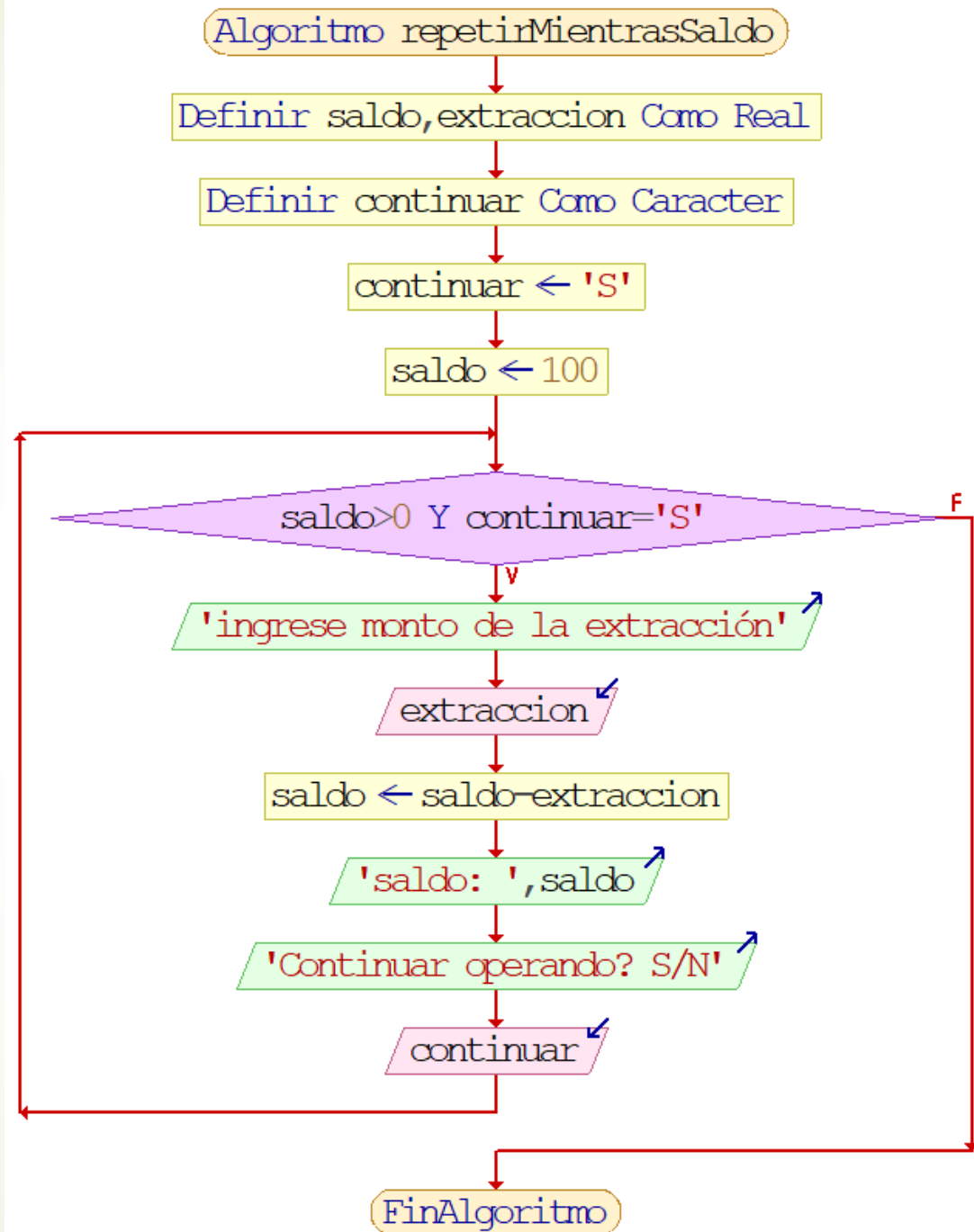
FinMientras

FinAlgoritmo

Diagrama de ejemplo de repetir mientras

Vemos en las secuencias que las flechas indican repetición o Bucle

9



Calibrador digital de neumáticos

Escribir un algoritmo permita calibrar un neumático a partir de la presión deseada ingresada por el usuario.

tener en cuenta que la rueda puede tener una presión mayor o menor a la que el usuario desea calibrar.

```

presionRueda.psc* x
1 Algoritmo CalibradorDeRuedas
2   //Autor: Fernando Princich
3   //Para: Elementos de Computación - FaCENA - UNNE
4   definir presionActual, presionIdeal Como Entero
5   mostrar Sin Saltar "Calibrar a: "
6   leer presionIdeal
7   leer presionActual // esto se debería leer desde un sensor
8   Mientras abs(presionActual-presionIdeal) > 0 Hacer
9     si presionActual>presionideal
10      presionActual=presionActual-1
11    SiNo
12      presionActual=presionActual+1
13    FinSi
14    // en la practica deberia leer la presion actual con un
15    //sensor en cada iteración entonces
16    // leer presionActual
17    imprimir "Presion : ", presionActual
18  Fin Mientras
19  Mostrar "*****"
20  Mostrar "Gracias por utilizar nuestro Calibrador digital de ruedas "
21  Mostrar "La presión de su neumático es: " presionActual ", Lbs."
22  Mostrar "*****"
23 FinAlgoritmo
24 // Por qué utilizamos la iteración mientras?
25 // Qué otros ejemplos se les ocurre donde se podría aplicar
26 //esta solución computacional?
27

```

Procesos para la solución de cualquier algoritmo o programa

1. Pensar globalmente la solución, las partes que la componen.
2. Dibujar la salida, como entiendo que debería ser la información que se presenta al usuario, mostrarla para validar.
3. Escribir cuales son las variables de entrada y de salida, los mensajes, esto tiene que ver con el punto anterior, y a partir de eso plasmar la tabla para la prueba de escritorio (seguramente después sufre modificaciones)
4. Escribir el algoritmo
5. Probarlo/modificarlo con el paso a paso y la prueba de escritorio
6. **Si funciona entonces**

FIN

SiNO Entonces

Volver_1

```
Iniciar_Proceso_Algoritmico
Mientras Algoritmo_no_funciona hacer:
    Punto1
    punto2
    Punto3
    Punto4
    punto5
    leer algoritmo_no_funciona
FinMientras
```

Ejercicios Complementarios

Empresa de electricidad

Una empresa factura la electricidad que consume cada usuario a partir de las lecturas del medidor correspondientes al mes actual y al anterior. Por cada cliente se lleva un registro que contiene:

- Numero de Cliente
- CantidadKWAnterior
- CantidadKWActual

El último registro contiene Número de Cliente = 0.

Calcular e imprimir:

- Consumo en Kwh.
- El importe a pagar por cada cliente, sabiendo que el \$/Kwh se cobra a razón de \$ 2,00 si el consumo es menor a 600 kw y \$2,5 si el consumo es igual o mayor a 600 kw.
- Realizar un listado de los clientes. (Imprimir código de cliente, consumo e importe a pagar)
- El importe total facturado por la empresa.

Ejercicios Complementarios

Sueldos

En una empresa se tiene que calcular el sueldo de los empleados teniendo en cuenta la cantidad de horas que ha trabajado cada uno.

Los datos que se tienen de cada empleado son:

- DNI Empleado
- Cantidad de horas trabajadas

Se sabe además que la cantidad de hs que debe trabajar cada empleado es de 160 hs al mes. y que por hora se paga \$200 y por horas extras se pagan \$300

Se pide:

- Calcular y listar las horas trabajadas por cada empleado y su sueldo
- El total a pagar por la empresa
- El promedio de sueldo de los empleados
- La cantidad de empleados que trabajaron horas extras.

Ciclo repetir hasta que

- ▶ La acción se lleva a cabo al menos UNA vez, por la simple razón que la CONDICION de fin de la repetición está al final

Repetir

secuencia_de_acciones

Hasta Que expresion_logica

Ciclo repetir hasta que, veamos un ejemplo

Algoritmo division

Definir a,b,c Como real

Escribir 'Ingrese dividendo:'

Leer a

Repetir

 Escribir 'Ingrese divisor mayor a cero:'

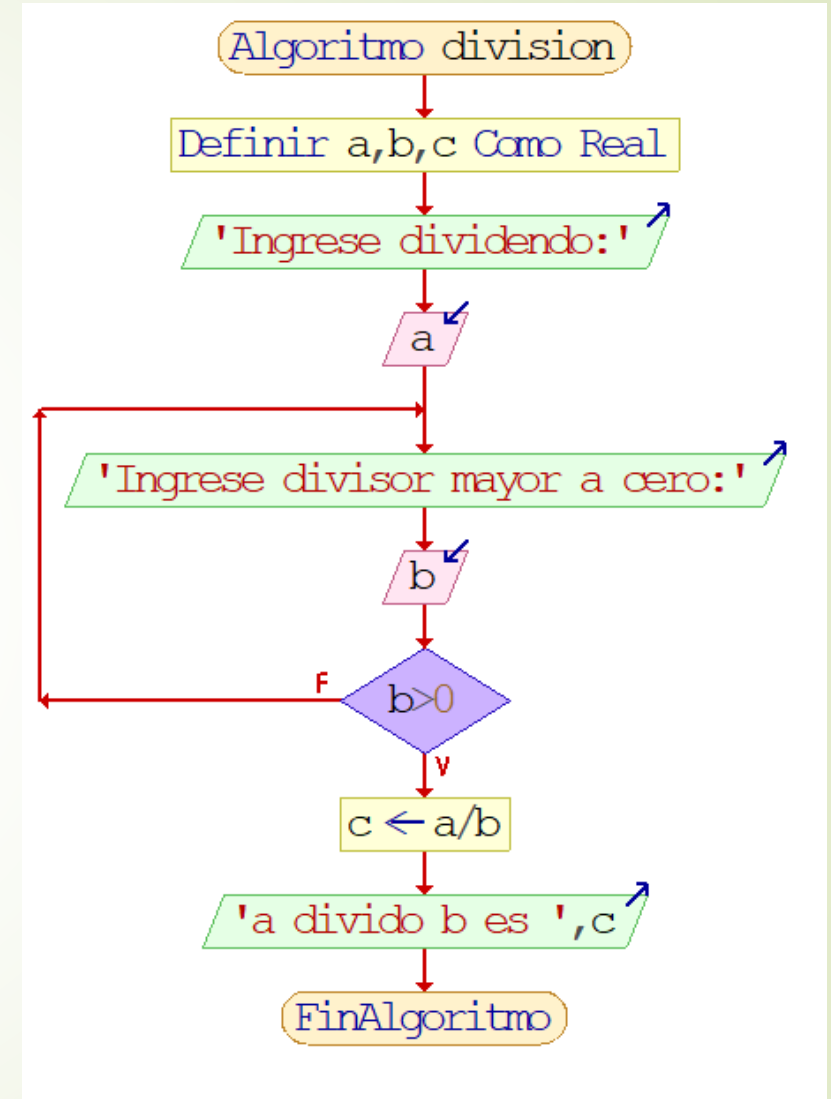
 Leer b

Hasta Que $b > 0$

$c \leftarrow a/b$

 Escribir 'a dividido b es ',c

FinAlgoritmo



Ciclos de repetición Hacer hasta que



¿Cuál es la diferencia principal de repetir **hasta que** y repetir **mientras**?



en **repetir hasta** que la **ACCIÓN** se realiza al menos **una vez**



Nuestro análisis del problema tiene que contemplar la conveniencia de usar un ciclo de repetición u otro

Tener en cuenta que la condición de fin está después de la acción

- En ciclos hacer hasta que **al menos UNA** vez se ejecutará la acción
- Al pensar la **condición de fin** controlar que **no se pase del límite**

Veamos el ejemplo de la Guía, **algoritmo para ingresar a un sistema**

Usuario

Contraseña

Permite **tres intentos**, si fallan, blanquea la contraseña

Algoritmo contraseña

definir usuario, usu1, contraUsu, contra1

Como Caracter

Definir contador Como Entero

usuario="Laurita"

contraUsu="1234"

Repetir

Escribir "ingrese usuario"

Leer usu1

Escribir "ingrese contraseña"

Leer contra1

si usuario=usu1 y contraUsu=contra1

Escribir "bienvenido al sistema"

SiNo

contador=contador+1

Escribir "intento fallido " contador

FinSi

si contador=3

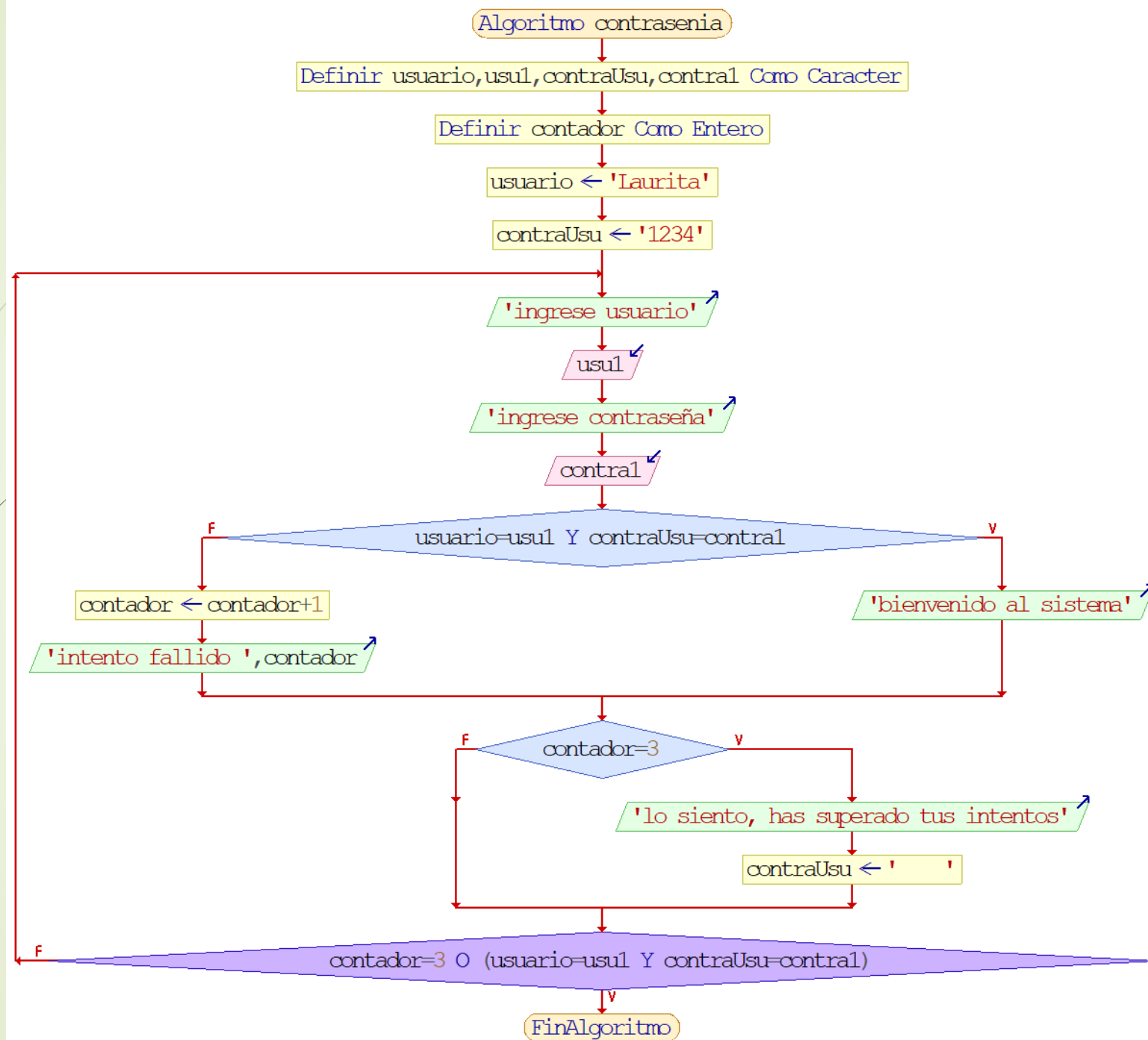
Escribir "lo siento, has superado tus
intentos"

contraUsu=""

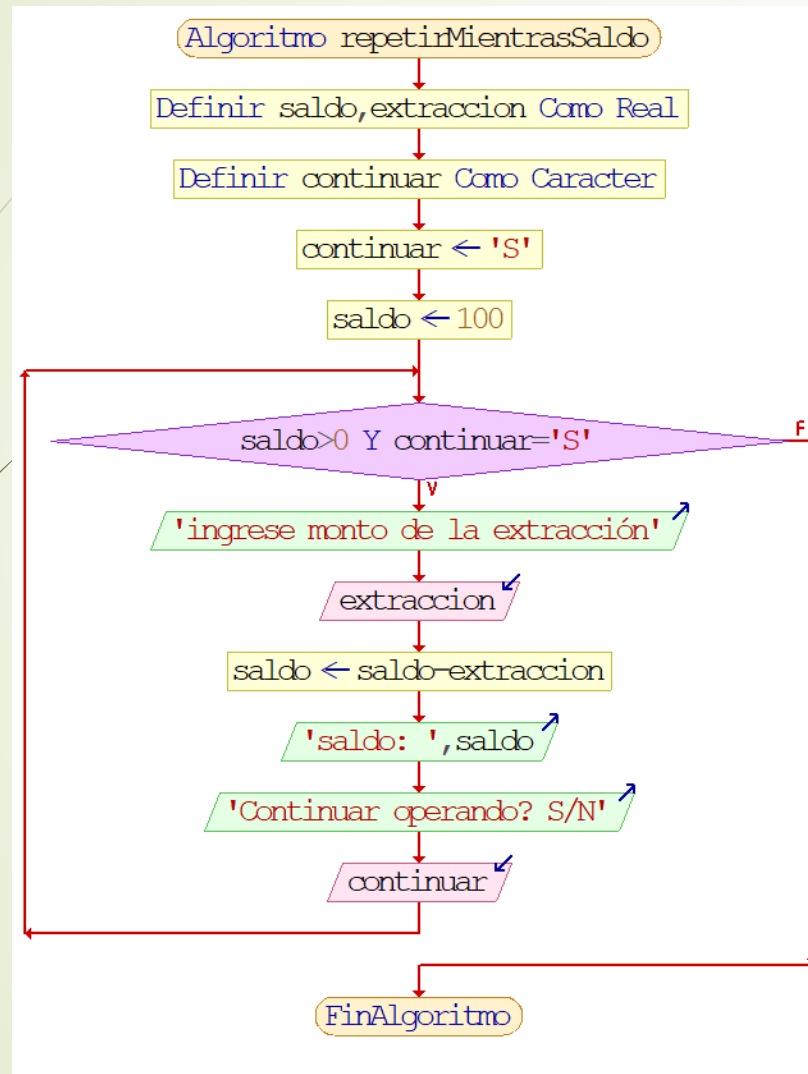
FinSi

Hasta Que contador=3 o (usuario=usu1 y
contraUsu=contra1)

FinAlgoritmo

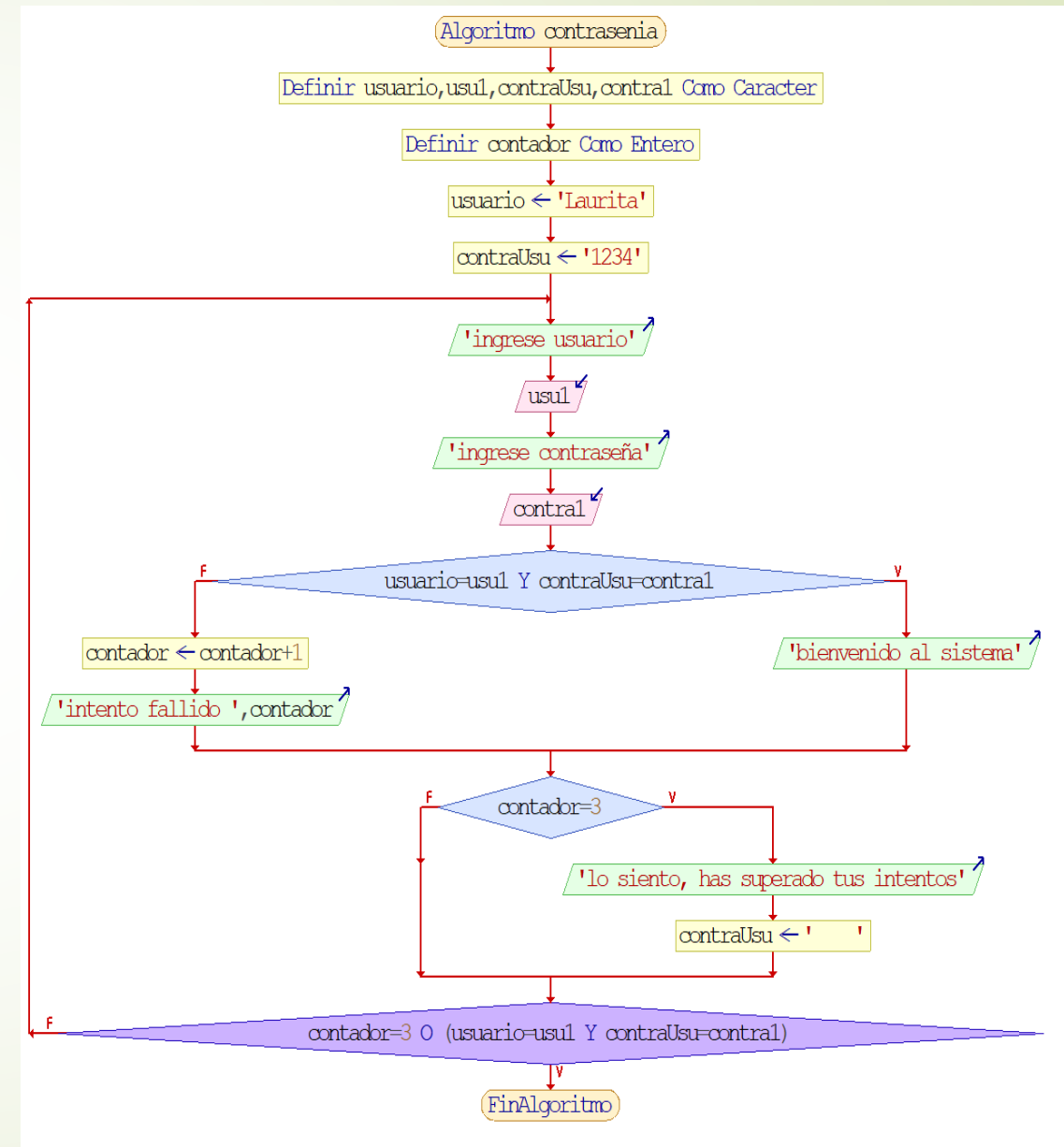


Ciclo repetir Mientras



Elementos de Computación

Ciclo repetir hasta que



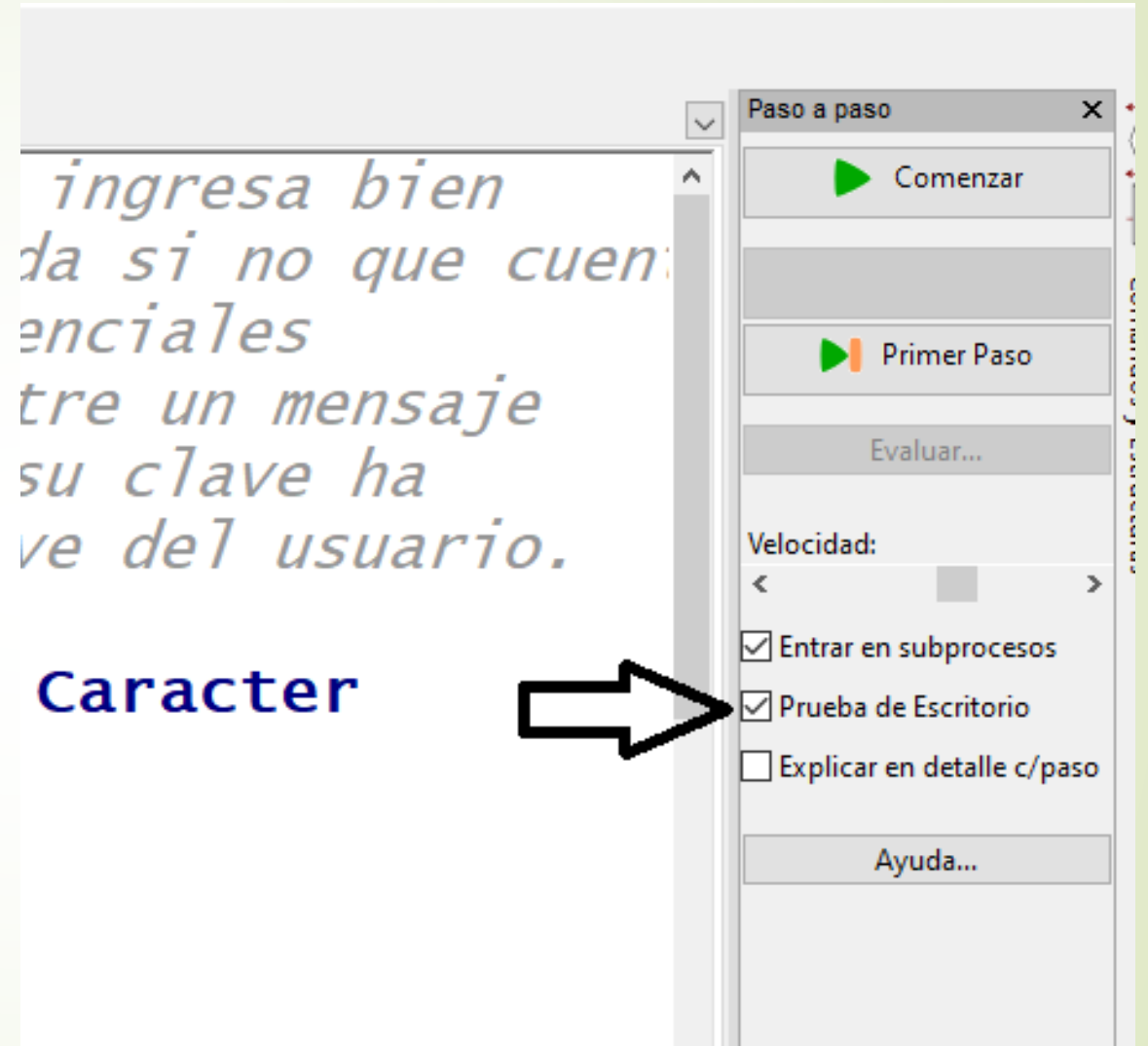
Cómo usar la prueba de escritorio de Pseint

- Una vez escrito el algoritmo, activamos la opción **Paso a Paso y tildamos en prueba de escritorio**
- Regulamos la **velocidad** del Paso a Paso a gusto
- **Agregamos las variables** a controlar en la prueba de escritorio, utilizando el botón agregar

Es un método muy útil para seguir el comportamiento de las variables durante la ejecución del algoritmo y la corrección de errores

Prueba de escritorio

- Para iniciar una prueba de escritorio vamos a la columna derecha y seleccionamos paso a paso, y luego tildamos prueba de escritorio



Agregamos las variables a evaluar

3 // la cantidad de veces que ingresa mal sus credenciales
 4 // cuando supere las tres entradas erróneas muestre un mensaje
 5 // usted ha superado los tres intentos válidos, su clave ha
 6 // sido invalidada, el algoritmo blanquea la clave del usuario.
 7 **Algoritmo** contraseña
 8 **Definir** usuario,usu1,
 9 **Definir** contador Como **no** Caracter
 10 usuario <- 'Laurita'
 11 contraUsu <- '1234'
 12 **Repetir**
 13 **Escribir** 'ingrese usuario'
 14 **Leer** usu1
 15 **Escribir** 'ingrese contraseña'

Prueba de Escritorio

Variable o expresión a evaluar:
 contador

OK Cancel

paso 2

paso 3

paso 1

| Proceso/SubProceso | Linea(inst) | contador | usu1 | contra1 |
|--------------------|-------------|---|----------|---------|
| 1: CONTRASEÑA | 17(1) | <<Variable no inicializada (CONTADOR)>> | Milagros | 1234 |
| 1: CONTRASEÑA | 19(1) | <<Variable no inicializada (CONTADOR)>> | Milagros | 1234 |
| 1: CONTRASEÑA | 20(1) | <<Variable no inicializada (CONTADOR)>> | Milagros | 1234 |

Prueba de Escritorio

Agregar...
 Limpiar
 Ayuda...

Prueba de escritorio, luego de ejecutar el algoritmo para tres valores de usuario y contraseña

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

Prueba de Escritorio

| Proceso/SubProceso | Linea(inst) | contador | usu1 | contra1 |
|--------------------|-------------|---|----------|---------|
| 1: CONTRASENIA | 17(1) | <<Variable no inicializada (CONTADOR)>> | Milagros | 1234 |
| 1: CONTRASENIA | 19(1) | <<Variable no inicializada (CONTADOR)>> | Milagros | 1234 |
| 1: CONTRASENIA | 20(1) | <<Variable no inicializada (CONTADOR)>> | Milagros | 1234 |
| 1: CONTRASENIA | 21(1) | 1 | Milagros | 1234 |
| 1: CONTRASENIA | 22(1) | 1 | Milagros | 1234 |
| 1: CONTRASENIA | 23(1) | 1 | Milagros | 1234 |
| 1: CONTRASENIA | 26(1) | 1 | Milagros | 1234 |
| 1: CONTRASENIA | 27(1) | 1 | Milagros | 1234 |
| 1: CONTRASENIA | 13(1) | 1 | Milagros | 1234 |
| 1: CONTRASENIA | 14(1) | 1 | Milagros | 1234 |
| 1: CONTRASENIA | 15(1) | 1 | maria | 1234 |
| 1: CONTRASENIA | 16(1) | 1 | maria | 1234 |
| 1: CONTRASENIA | 17(1) | 1 | maria | 2222 |
| 1: CONTRASENIA | 19(1) | 1 | maria | 2222 |
| 1: CONTRASENIA | 20(1) | 1 | maria | 2222 |
| 1: CONTRASENIA | 21(1) | 2 | maria | 2222 |
| 1: CONTRASENIA | 22(1) | 2 | maria | 2222 |
| 1: CONTRASENIA | 23(1) | 2 | maria | 2222 |
| 1: CONTRASENIA | 26(1) | 2 | maria | 2222 |
| 1: CONTRASENIA | 27(1) | 2 | maria | 2222 |
| 1: CONTRASENIA | 13(1) | 2 | maria | 2222 |
| 1: CONTRASENIA | 14(1) | 2 | maria | 2222 |
| 1: CONTRASENIA | 15(1) | 2 | Laurita | 2222 |
| 1: CONTRASENIA | 16(1) | 2 | Laurita | 2222 |
| 1: CONTRASENIA | 17(1) | 2 | Laurita | 1234 |
| 1: CONTRASENIA | 18(1) | 2 | Laurita | 1234 |
| 1: CONTRASENIA | 22(1) | 2 | Laurita | 1234 |
| 1: CONTRASENIA | 23(1) | 2 | Laurita | 1234 |
| 1: CONTRASENIA | 26(1) | 2 | Laurita | 1234 |
| 1: CONTRASENIA | 27(1) | 2 | Laurita | 1234 |
| 1: CONTRASENIA | 28(1) | 2 | Laurita | 1234 |

Agregar...
Limpiar
Ayuda...

21:02
12/5/2020

Preguntas

