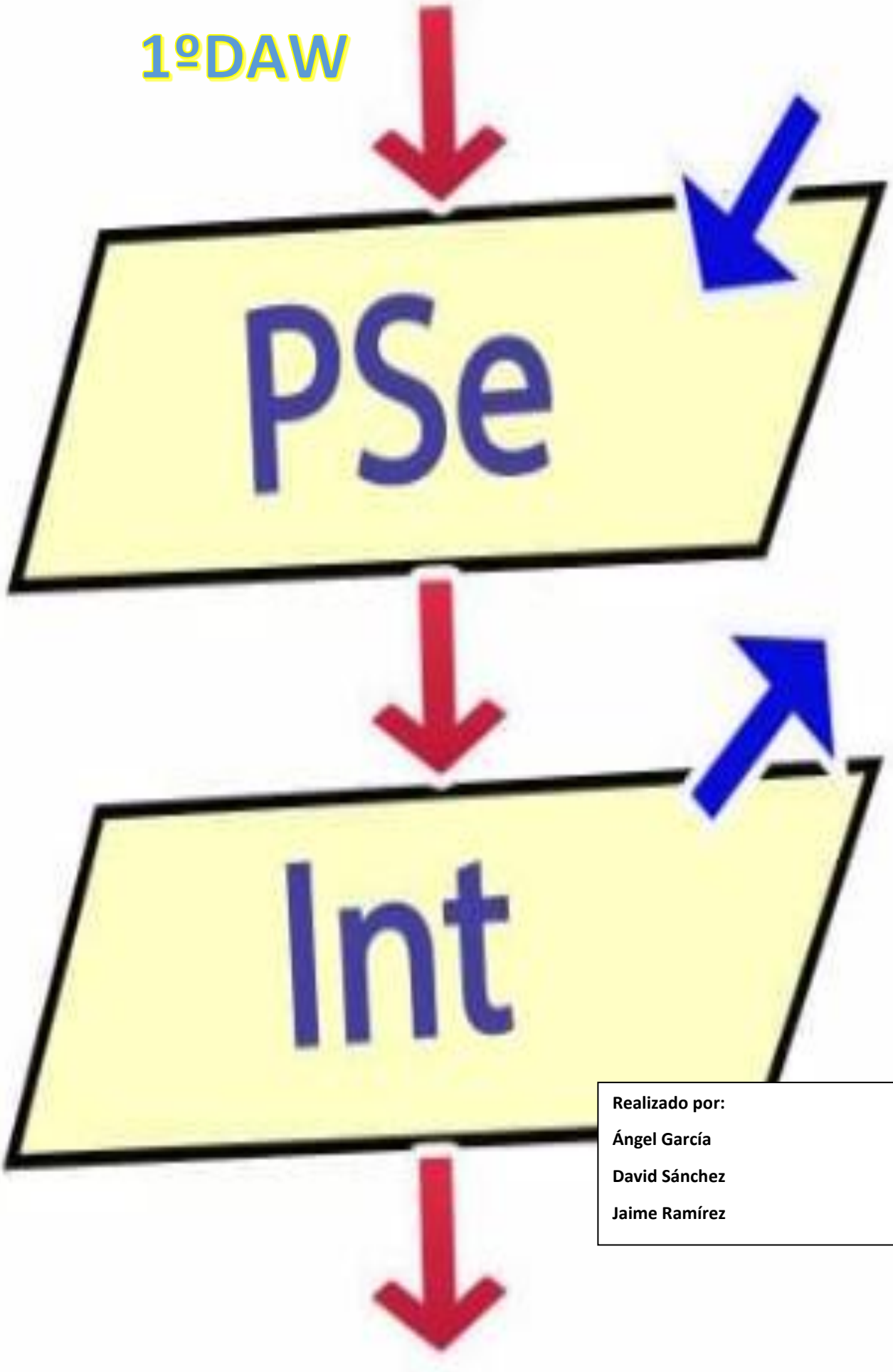
Proyecto Calculadora

1ºDAW



Realizado por:

Ángel García

David Sánchez

Jaime Ramírez

Índice

1. Instrucción

Ejemplo: Escribir “El resultado es ”, resultadoSuma

```
Escribir "El resultado es ", resultadoSuma
```

¿Por qué ese ejemplo?

Porque es una instrucción que realiza una acción, en este caso muestra el resultado de la suma en la salida.

2. Expresión

Ejemplo: resultadoMultiplicar * num

```
resultadoMultiplicar * num
```

¿Por qué ese ejemplo?

Porque es una expresión que calcula el producto de resultadoMultiplicar y num

3. Palabras reservadas

Ejemplo: FinAlgoritmo

```
FinAlgoritmo
```

¿Por qué ese ejemplo?

Porque es una palabra reservada que usa PSeint para indicar el final de un algoritmo.

4. Variable

Ejemplo: resultadoResta

```
resultadoResta
```

¿Por qué este ejemplo?

Porque es una variable que almacena el resultado de la resta realizada por el usuario

5. Constantes

Ejemplo: PI en área = PI * radio ^ 2

```
area = PI * radio↑2
```

¿Por qué este ejemplo?

Porque PI es una constante que representa un valor fijo y conocido en matemáticas

6. Literal

Ejemplo: “Calculadora Avanzada”

```
Escribir "Calculadora Avanzada"
```

¿Por qué este ejemplo?

Porque es un literal de tipo cadena o string que representa un texto específico y no va a cambiar nunca durante la ejecución del programa.

7. Operadores Aritméticos de cada tipo

Operador Aritmético	Operador Relacional
+	< >
`resultadoSuma + num`	`Mientras num2 ≠ 0 Hacer`
Realiza la suma de 2 números	Compara si 2 valores son diferentes
Operador Lógico	Operador Especial
Y	MOD
`Si num > 1 Entonces`	`num1 MOD num2`
Se utiliza para evaluar 2 condiciones	Devuelve el resto de una división

8. Un tipo de dato: Entero, Real, Cadena, Lógico

Definir num Como Entero	Definir nombre1, nombre2, nombre3 Como Cadena
`Definir num Como Ent`	`Definir nombre1, nombre2, nombre3 Como`
Permite almacenar valores enteros	Permite almacenar textos

En este ejercicio podríamos haber usado tipos de datos lógicos y reales, pero no hemos caído en usarlos .

9. Comentario

Ejemplo: // Realizado por Angel o // Hecho por David o // Realizado por Jaime

```
//Realizado por Ángel
```
```
//Hecho por David
```
```
// Realizado por Jaime
```

¿Por qué este ejemplo?

Simplemente es un comentario que no afecta a la ejecución del código y proporciona información a la persona que lo está leyendo

10. Contador

Ejemplo: Para i <- 3 Hasta num Hacer

```
Para i ← 3 Hasta num Hacer
```

¿Por qué este ejemplo?

Porque en programación tanto la letra i como la letra j se usan como contadores.

11. Acumulador

Ejemplo: resultadoSuma = resultadoSuma + num

```
resultadoSuma = resultadoSuma + num
```

¿Por qué este ejemplo?

Porque resultadoSuma se utiliza para acumular la suma de los números introducidos por el usuario.

12. While

Ejemplo: Mientras num2 <> 0 Hacer

```
Mientras num2 ≠ 0 Hacer
```

¿Por qué este ejemplo?

Porque este bucle siempre se repetirá mientras que num2 sea distinto de 0

13. Do While

Ejemplo: Repetir <secuencia de instrucciones> Hasta que num = 0

```
Repetir
    Escribir "Introduzca un número introduzca 0 para ver el resultado: "
    Leer num

    // Condición sí para que cuando el usuario introduzca 0 la función no tome ese valor y de 0 como resultado
    si num ≠ 0 Entonces
        resultadoMultiplicar = resultadoMultiplicar * num
    FinSi


Hasta Que num = 0
```

¿Por qué este ejemplo?

Porque ese bloque de instrucciones se ejecuta una vez y se va a repetir hasta que num sea igual a 0

14. FOR

Ejemplo: Para i <- 3 Hasta num Hacer

```
Para i ← 3 Hasta num Hacer
```

¿Por qué este ejemplo?

Porque se utiliza para repetir un bloque de código un número específico de veces, este bucle está incorporado en la sucesión de Fibonacci.

15. Dato de entrada

Ejemplo: Leer num

```
Leer num
```

¿Por qué este ejemplo?

Porque se utiliza para leer el dato introducido por el usuario.

16. Dato de Salida

Ejemplo: Escribir “El resultado es ”, resultadoSuma

```
Escribir "El resultado es ", resultadoSuma
```

¿Por qué este ejemplo?

Porque muestra el resultado de la operación en la salida.

17. Cambio de estado

Ejemplo: resultadoMultiplicar = resultadoMultiplicar * num

```
resultadoMultiplicar = resultadoMultiplicar * num
```

¿Por qué este ejemplo?

Porque cambia el valor de resultadoMultiplicar cada vez que se agrega un nuevo número para multiplicar por el anterior.

18. Aspecto de programación imperativa

Ejemplo: El uso de instrucciones que modifican el estado de las variables y ejecutan acciones secuenciales.

¿Por qué?

Porque el código sigue un enfoque paso a paso para realizar cálculos y mostrar resultados.

19. Aspecto de programación modular

Ejemplo: La división del código en funciones como por ejemplo realizarSuma, realizarResta, realizarMultiplicación, realizarDivision …

¿Por qué?

Porque cada función tiene una responsabilidad específica, facilitando así la lectura y posterior mantenimiento del código.

20. Código cohesionado o no cohesionado.

El código presenta una buena cohesión ya que cada función está diseñada para realizar una tarea específica, facilitando la claridad y el mantenimiento de dicho código.