

**Aplicar conocimientos en pseudocódigo y diagramas de
flujo para la solución de problemas algorítmicos
GA3-220501093-AA2-EV03**



Isidro J Gallardo Navarro

Ficha:3070299

2025

**Tecnología en Análisis y Desarrollo de
Software.**

ADSO

Lista de chequeo a cubrir:

- Selecciona una herramienta de prototipado de acuerdo con las características del sistema a modelar.
- Construye los prototipos del sistema, siguiendo una estructura base de presentación de contenido.
- Se evidencia un flujo claro de interacción por medio de los prototipos para los requerimientos de mayor prioridad.

1. Introducción

Este documento presenta la implementación de cuatro algoritmos fundamentales utilizando la notación de PSeInt, acompañados de sus respectivos diagramas de flujo. Los algoritmos cubren cálculos matemáticos básicos, conversiones de unidades y procesamiento de datos múltiples, demostrando la aplicación práctica de estructuras algorítmicas en la resolución de problemas cotidianos.

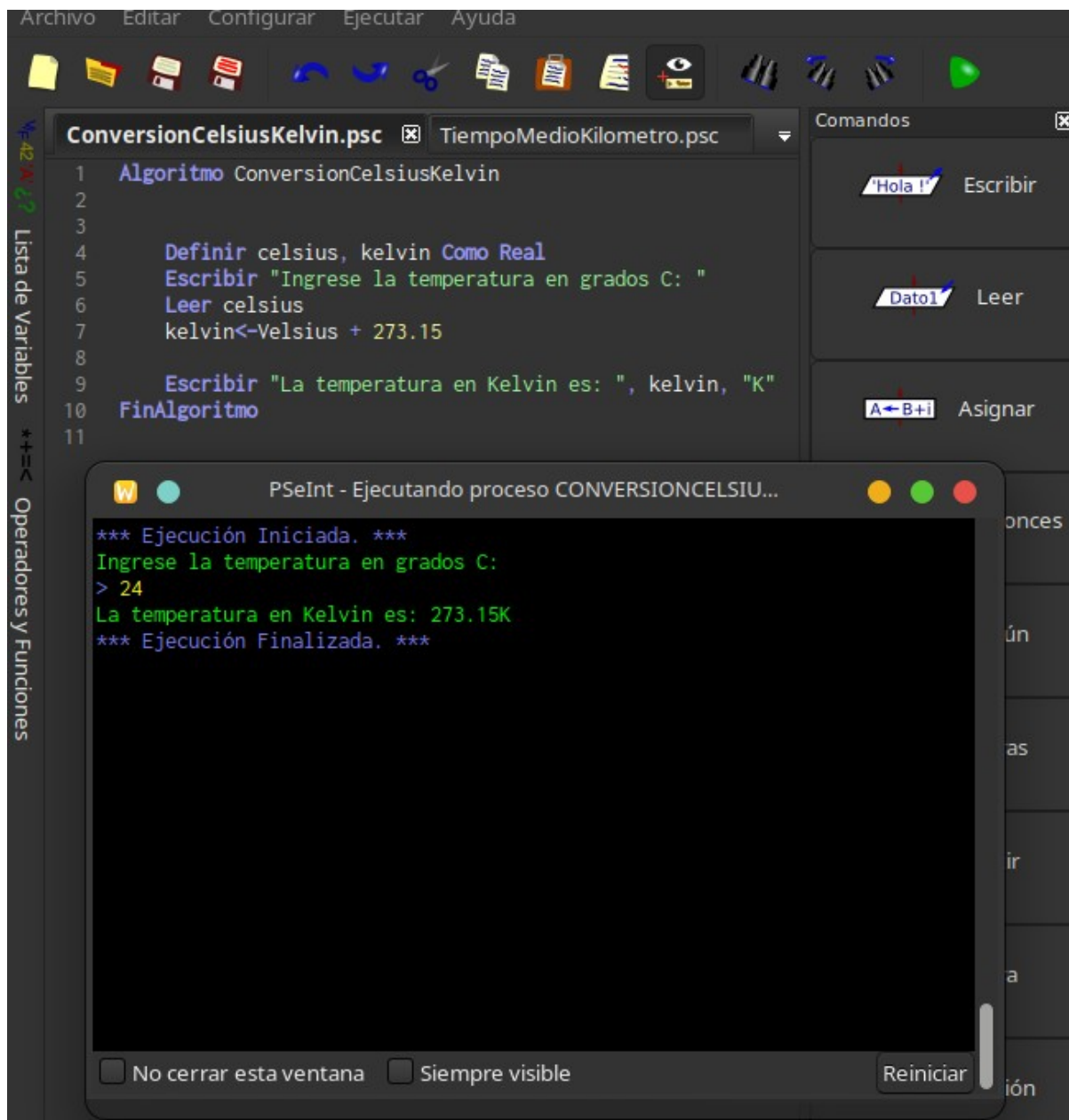
Cada algoritmo se presenta con su pseudocódigo correspondiente y diagrama de flujo, siguiendo las convenciones establecidas para facilitar la comprensión y posterior implementación en entornos de desarrollo.

2. Algoritmo 1: Cálculo del Tiempo Medio por Kilómetro

2.1 Descripción del Problema

Un ciclista ha recorrido una distancia total de 15 kilómetros en un tiempo de 40 minutos. Se requiere calcular el tiempo promedio que necesita para recorrer cada kilómetro.

2.2 Pseudocódigo en PSeInt



2.3 Diagrama de Flujo

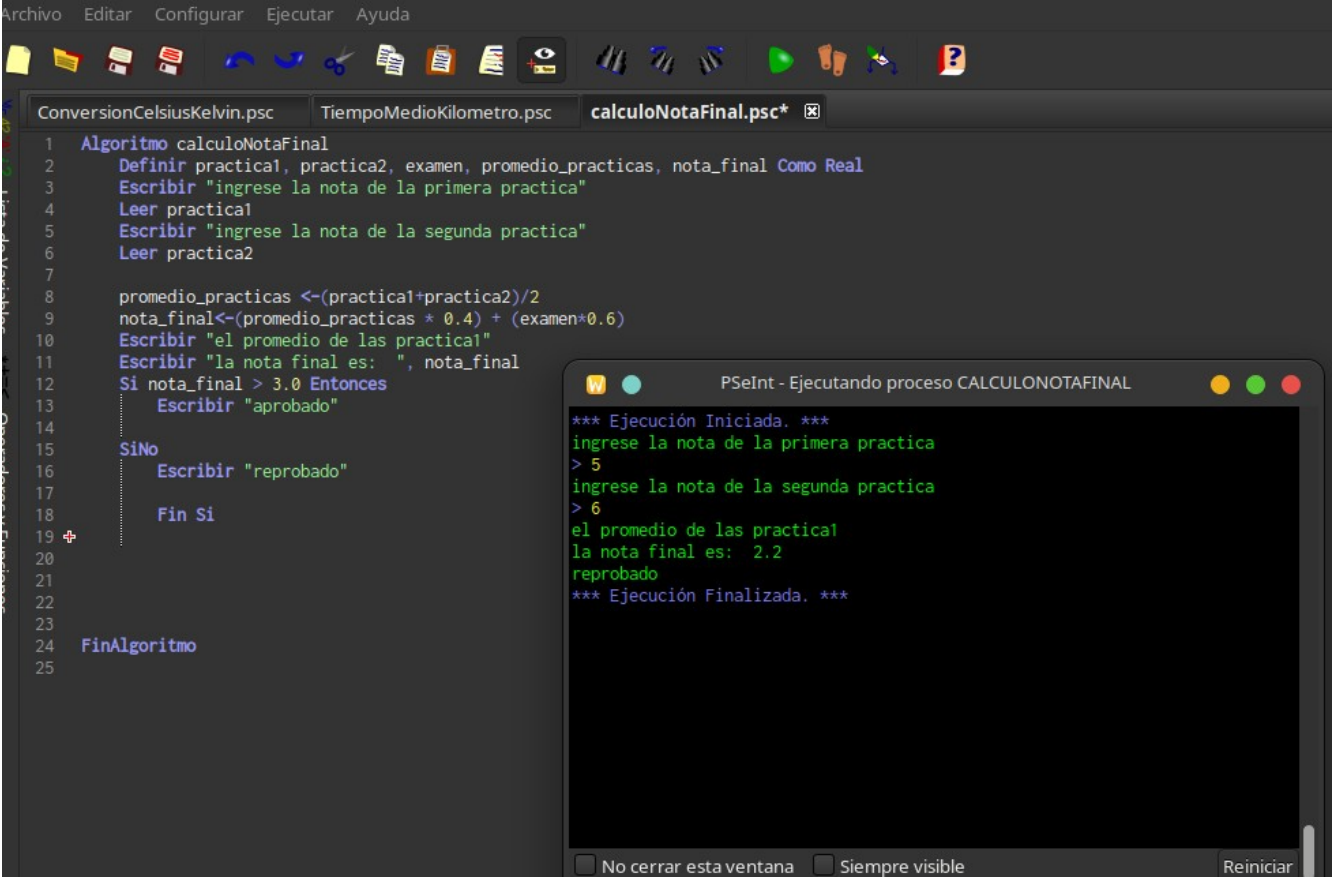
Algoritmo TiempoMedioKilometro
Definir distancia, tiempo, tiempo_medio Como Real
distancia \leftarrow 15
tiempo \leftarrow 40
tiempo_medio \leftarrow tiempo/distancia
Escribir 'El tiempo medio por Kilometro es: ', tiempo_medio, 'minutos'
FinAlgoritmo

4. Algoritmo 3: Cálculo de Nota Final

4.1 Descripción del Problema

Calcular la nota final de un estudiante considerando dos prácticas (40% del total) y un examen final (60% del total).

4.2 Pseudocódigo en PSeInt



The image shows the PSeInt software interface. The main window displays a pseudocode algorithm for calculating a final grade. The algorithm is as follows:

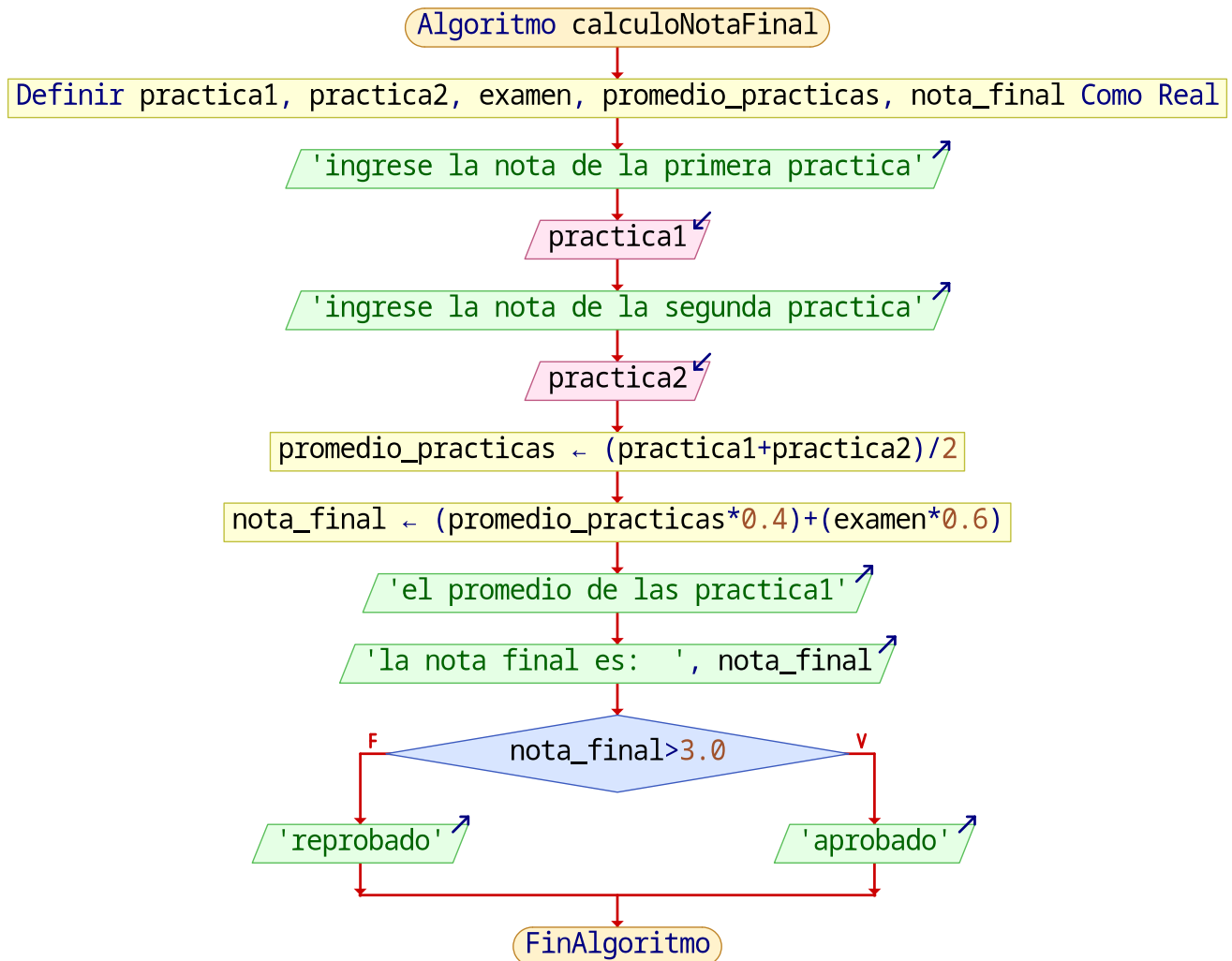
```
1  Algoritmo calculoNotaFinal
2  Definir practica1, practica2, examen, promedio_practicas, nota_final Como Real
3  Escribir "ingrese la nota de la primera practica"
4  Leer practica1
5  Escribir "ingrese la nota de la segunda practica"
6  Leer practica2
7
8  promedio_practicas <- (practica1 + practica2) / 2
9  nota_final <- (promedio_practicas * 0.4) + (examen * 0.6)
10 Escribir "el promedio de las practicas"
11 Escribir "la nota final es: ", nota_final
12 Si nota_final > 3.0 Entonces
13     Escribir "aprobado"
14
15 SiNo
16     Escribir "reprobado"
17
18 Fin Si
19
20
21
22
23
24 FinAlgoritmo
25
```

The execution window, titled "PSeInt - Ejecutando proceso CALCULONOTAFINAL", shows the output of the program:

```
*** Ejecución Iniciada. ***
ingrese la nota de la primera practica
> 5
ingrese la nota de la segunda practica
> 6
el promedio de las practicas
la nota final es: 2.2
reprobado
*** Ejecución Finalizada. ***
```

At the bottom of the execution window, there are checkboxes for "No cerrar esta ventana" and "Siempre visible", and a "Reiniciar" button.

4.3 Diagrama de Flujo



5. Algoritmo 4: Conversión de Dólares a Euros

5.1 Descripción del Problema

Sumar los precios de 7 artículos expresados en dólares y convertir el total a euros utilizando una tasa de cambio específica.

5.2 Pseudocódigo en PSeInt

``pseint

Algoritmo ConversionDolaresEuros

// Declaración de variables

Definir tasa_cambio, total_dolares, total_euros Como Real

Definir precio Como Real

Definir i Como Entero

// Inicialización

total_dolares \leftarrow 0

// Entrada de la tasa de cambio

Escribir "Ingrese la tasa de cambio (1 USD = ? EUR): "

Leer tasa_cambio

// Entrada de precios de los 7 artículos

Para i \leftarrow 1 Hasta 7 Con Paso 1 Hacer

 Escribir "Ingrese el precio del artículo ", i, " en dólares: "

 Leer precio

 total_dolares \leftarrow total_dolares + precio

FinPara

// Proceso de conversión

total_euros \leftarrow total_dolares * tasa_cambio

// Salida de resultados

Escribir "Total en dólares: \$", total_dolares

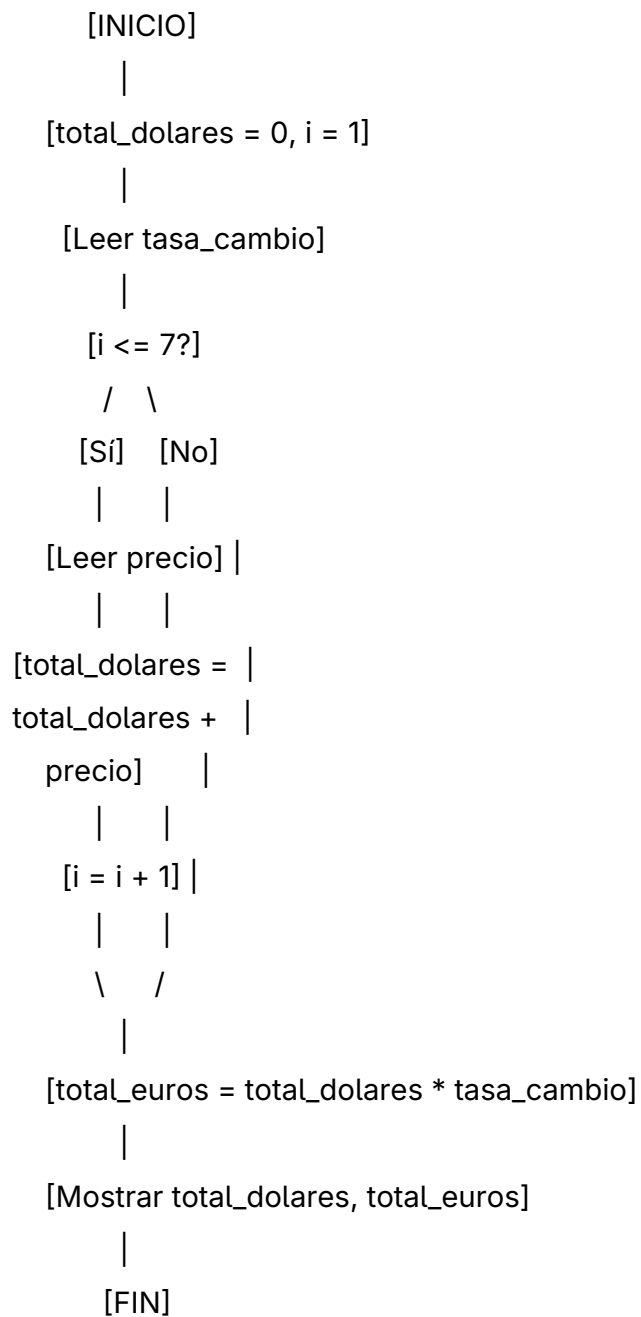
Escribir "Total en euros: €", total_euros

FinAlgoritmo

...

5.3 Diagrama de Flujo

...



...

6. Funciones y Procedimientos Complementarios

6.1 Función para Validación de Datos

```
``pseint
Funcion resultado ← ValidarNumero(numero)
    Si numero >= 0 Entonces
        resultado ← Verdadero
    Sino
        resultado ← Falso
    FinSi
FinFuncion
...
```

6.2 Procedimiento para Mostrar Menú

```
``pseint
SubProceso MostrarMenu
    Escribir "=== MENÚ DE ALGORITMOS ==="
    Escribir "1. Tiempo Medio por Kilómetro"
    Escribir "2. Conversión Celsius a Kelvin"
    Escribir "3. Cálculo de Nota Final"
    Escribir "4. Conversión Dólares a Euros"
    Escribir "5. Salir"
    Escribir "Seleccione una opción: "
FinSubProceso
...
```

6.3 Algoritmo Principal Integrado

``pseint

Algoritmo MenuPrincipal

Definir opcion Como Entero

Definir continuar Como Logico

continuar \leftarrow Verdadero

Mientras continuar Hacer

MostrarMenu

Leer opcion

Segun opcion Hacer

1: TiempoMedioKilometro

2: ConversionCelsiusKelvin

3: CalculoNotaFinal

4: ConversionDolaresEuros

5: continuar \leftarrow Falso

De Otro Modo:

Escribir "Opción no válida"

FinSegun

FinMientras

Escribir "Programa finalizado"

FinAlgoritmo

``

7. Análisis de Complejidad y Eficiencia

7.1 Complejidad Temporal

- Algoritmos 1, 2, 3: $O(1)$ - Tiempo constante
- Algoritmo 4: $O(n)$ - Tiempo lineal donde $n = 7$ (número de artículos)

7.2 Complejidad Espacial

Todos los algoritmos presentan una complejidad espacial de $O(1)$, utilizando un número fijo de variables independientemente del tamaño de entrada.

7.3 Optimizaciones Posibles

1. Validación de entrada: Implementar verificaciones para evitar errores de ejecución
2. Manejo de excepciones: Considerar casos especiales como división por cero
3. Formato de salida: Mejorar la presentación de resultados con redondeo apropiado

8. Conclusiones

Los algoritmos presentados demuestran la aplicación efectiva de estructuras básicas de programación en la resolución de problemas matemáticos y de conversión. La implementación en PSeInt facilita la comprensión de la lógica algorítmica antes de proceder a lenguajes de programación específicos.

La utilización de diagramas de flujo complementa la documentación del pseudocódigo, proporcionando una representación visual clara del flujo de control y la secuencia de operaciones. Esta metodología dual favorece tanto el aprendizaje como la verificación de la correctitud de los algoritmos desarrollados.

Herramientas utilizadas: PSeInt para pseudocódigo y diagramas de flujo, siguiendo las convenciones estándar para algoritmos estructurados.