Evidencia de producto: GA3-220501093-AA1-EV02 taller de resolución de problemas de algoritmos en pseudocódigo y diagramas de flujo

Trabajo elaborado por:

Daniel Bustamante Ortega

Programa:

Tecnología en análisis y desarrollo de software

Ficha: 2834872

Servicio nacional de aprendizaje Sena, modalidad Virtual

Medellín, Colombia

Junio de 2024

# TABLA DE CONTENIDO.

[Portada 1](#_Toc150612692)

[Tabla de contenido 2](#_Toc150612692)

[Introducción](#_Toc150612692) 3

[Propósito](#_Toc150612692) 4

[Requerimientos de la actividad](#_Toc150612693) 5-6

[Solución a la actividad Sección 1 Ejercicios 1, 2, 3, 4](#_Toc150612693) 7-26

[Solución a la actividad Sección 2 ¿Qué es un diagrama de flujo?](#_Toc150612698) 27-30

Que es un pseudocódigo31-33

Conclusiones34

Referencias35

# INTRODUCCIÓN.

# En esta actividad, se trabajará en el taller de resolución de problemas de algoritmos en pseudocódigo y diagramas de flujo, correspondiente a la evidencia de producto GA3-220501093-AA1-EV02. Siguiendo las pautas establecidas en la guía proporcionada, desarrollaremos soluciones utilizando Python como lenguaje de programación y la herramienta PSeInt de manera sencilla para formular el pseudocódigo y los diagramas de flujo.

# PROPÓSITO.

El objetivo principal es que, basándonos en las recomendaciones para el análisis de problemas presentadas durante el componente formativo, comprendamos detenidamente una serie de problemas planteados. A partir de esto, identificaremos los elementos de entrada y los resultados esperados. Adicionalmente, introduciremos el uso de diagramas de flujo, pseudocódigo en la herramienta PSeInt y un pequeño ejemplo en lenguaje de programación python por consola resolviendo aquellos problemas.

# 

# REQUERIMIENTOS DE LA ACTIVIDAD.

Teniendo en cuenta las recomendaciones para el análisis de problemas desarrollado durante el componente formativo en esta actividad se plantean algunos problemas, los cuales deberá leer detenidamente para luego identificar los elementos de entrada y los resultados esperados. Adicionalmente, se introduce el tema de diagramas de flujo para lo cual deberá realizar una investigación corta respondiendo algunos lineamientos establecidos en esta actividad. Elementos para tener en el desarrollo del taller:

**Sección 1:**

Teniendo en cuenta los siguientes problemas identificar los posibles elementos de entrada y resultados esperados de cada uno:

● Se desea elaborar un algoritmo que permita identificar la cantidad de dólares equivalentes a una cantidad de pesos colombianos.

● Se desea elaborar un algoritmo que permita determinar la temperatura equivalente en grados centígrados a la cantidad de grados Fahrenheit actuales en la ciudad de New York.

● Suponiendo que nos encontramos descansando en una nuestra casa en una ciudad de Colombia requiero hacer un plan detallado para llegar a tiempo a mi sitio de trabajo el día siguiente.

● Suponiendo que tengo habilidades en la elaboración de comida necesito elaborar un arroz con pollo para 5 personas. Respecto a los problemas planteados: Para precisar los elementos de entrada de cada una de las situaciones enunciadas no olvide las preguntas guía:

● ¿Qué información es importante y necesaria para resolver el problema?

● ¿Qué información no es importante y se puede prescindir?

● ¿Cuáles son los datos de entrada conocidos?

● ¿Cuál es la incógnita o qué debemos calcular?

● ¿Los datos se pueden agrupar en categorías?

● ¿Qué información adicional hace falta para resolver el problema? Para precisar los resultados esperados se recomienda utilizar las preguntas guía:

● ¿Qué información están solicitando?

● ¿En qué formato se debe entregar esta información?

**Sección 2**

Taller Elabore una investigación corta usando los materiales disponibles en la biblioteca o Internet respecto a los fundamentos para la resolución de problemas con algoritmos, seleccione por lo menos tres fuentes que le permitan resolver las siguientes preguntas:

● Definición de diagrama de flujo.

● Símbolos más importantes.

● Seleccionar uno de los problemas de la sección 1 y representarlo en su equivalente diagrama de flujo. Respecto a los diagramas de flujo:

● Referenciar las fuentes de información utilizadas para resolver cada una de las preguntas utilizando el formato de las Normas APA. Lineamientos generales para la entrega de la evidencia:

● Producto para entregar: documento con la solución del taller propuesto.

● Formato: PDF.

● Extensión: libre.

# SOLUCIÓN DE LA ACTIVIDAD.

# SECCIÓN 1

# Ejercicio no 1:

# Se desea elaborar un algoritmo que permita identificar la cantidad de dólares equivalentes a una cantidad de pesos colombianos.

**Solución al ejercicio**

Para convertir pesos colombianos a dólares, se divide la cantidad de pesos por la tasa de cambio actual de pesos a dólares (3840 COP por USD).

Para convertir dólares a pesos colombianos, se multiplica la cantidad de dólares por la tasa de cambio (3840 COP por USD).

**Pasos de Lectura y Escritura:**

Se muestra un menú al usuario con dos opciones: convertir pesos a dólares o convertir dólares a pesos.

Se lee la opción seleccionada por el usuario.

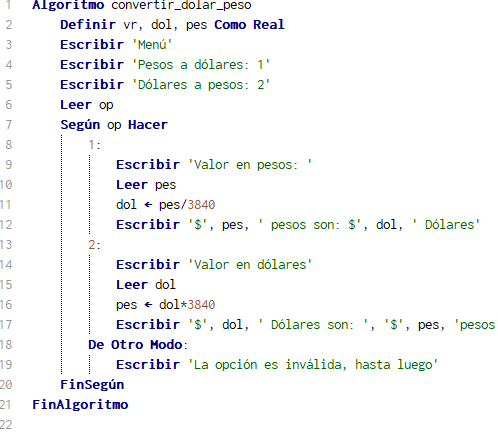
Dependiendo de la opción elegida, se solicita al usuario que ingrese la cantidad en la moneda de origen (pesos o dólares).

Se realiza la conversión utilizando la operación matemática correspondiente.

Se muestra el resultado al usuario.

A continuación se muestra en formato de imagen los procesos de pseudocódigo, diagramas de flujo y códigos de programación para los problemas planteado: (Los archivos serán adjuntados con la actividad y en este caso se subirán capturas de pantalla para no generar un efecto visual tan intranquilo en el documento.)

*Pseudocódigo* (Psint, 2024)



**Definir vr, dol, pes Como Real**: Este paso se utiliza para declarar las variables vr, dol y pes como números reales (decimales). Esto para asegurar que las operaciones matemáticas se realicen con precisión, especialmente en conversiones de moneda que pueden involucrar valores fraccionarios.

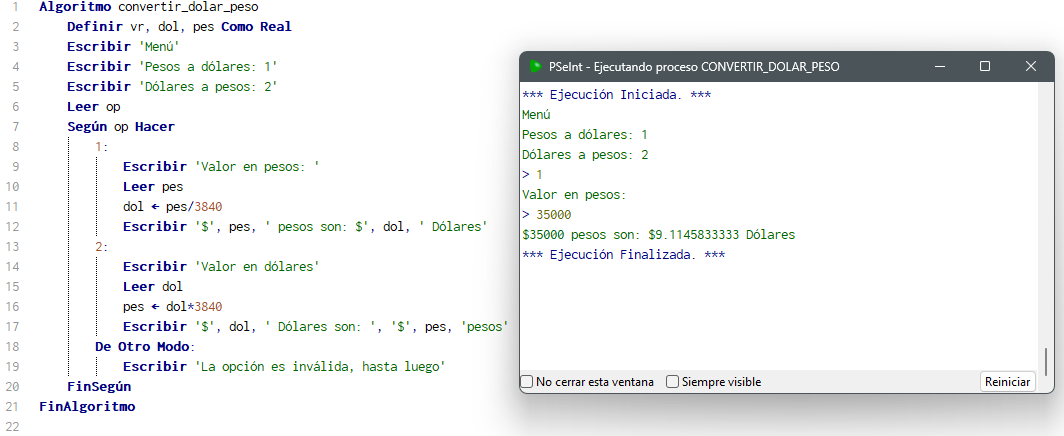
**Escribir 'Menú'**: La instrucción Escribir se usa para mostrar mensajes al usuario. Aquí, se utiliza para presentar un menú que explica las opciones disponibles para la conversión de moneda.

**Leer op**: La instrucción Leer permite capturar la entrada del usuario y almacenarla en la variable op. Esto es esencial para que el programa pueda actuar en función de la selección del usuario.

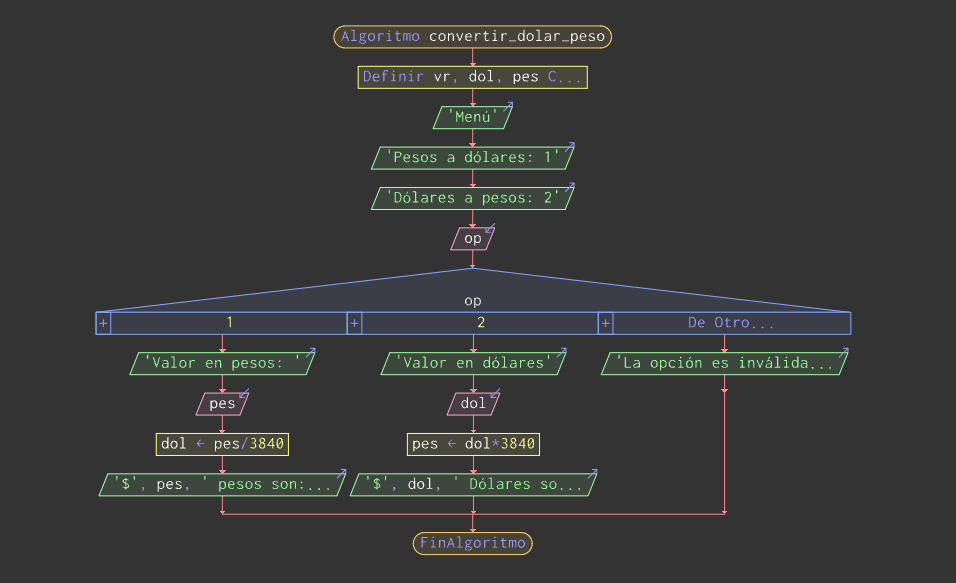
**Según op Hacer**: Esta es una estructura de control condicional que ejecuta diferentes bloques de código según el valor de op. Es similar a una sentencia switch en otros lenguajes de programación. Aquí, se utiliza para determinar qué conversión realizar en función de la opción elegida por el usuario.

**De Otro Modo**: Dentro de la estructura Según, De Otro Modo actúa como una cláusula de "else" que se ejecuta si ninguna de las condiciones anteriores es verdadera. En este caso, muestra un mensaje de error si la opción seleccionada es inválida.

*Resultados (Psint, 2024)*

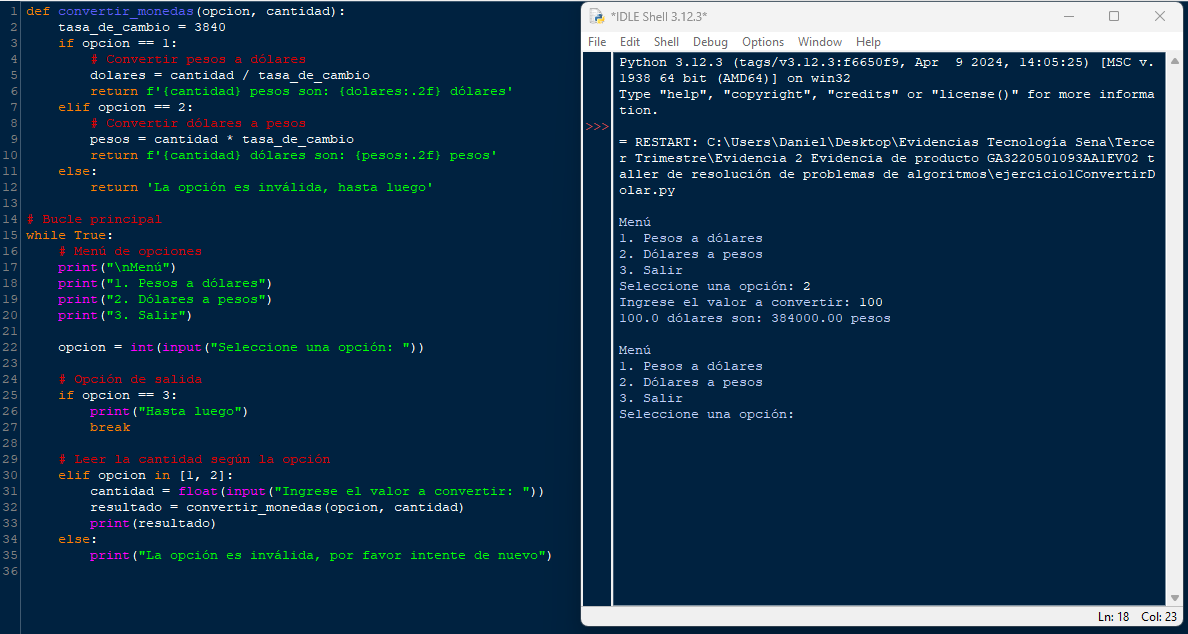


*Diagrama de flujo (Psint, 2024)*



*Los diagramas de flujo son generados a partir de los pseudocódigos en la herramienta PseInt o viceversa, y proporcionan una representación visual del flujo del programa, mostrando claramente las decisiones y los cálculos realizados en cada paso.*

*Algoritmo de programación en python - Resultados (Python.org, 2024)*



*Se puede apreciar como con el ciclo ‘while’ se crea una opción de salir para que mientras el usuario no decida salir del programa, el menú sigue apareciendo.*

**Definición de la Función convertir\_monedas:**

Esta función toma dos parámetros: opción y cantidad. Según la opción seleccionada por el usuario, retorna la conversión entre pesos colombianos y dólares estadounidenses utilizando una tasa de cambio fija de 3840 COP por USD. Esto la hace creando una sentencia condicional en donde si la opción es 1 almacena en la variable dólares la operación cantidad \* tasa de cambio y si la opción es 2 todo lo contrario.

**Bucle Principal:**

El bucle principal while True permite al usuario interactuar con un menú de opciones para convertir moneda o salir del programa.

**Mostrar el Menú:**

Se presenta un menú al usuario con tres opciones: convertir pesos a dólares, convertir dólares a pesos o salir del programa.

**Leer la Opción del Usuario:**

Se solicita al usuario que seleccione una opción. Si la opción es '1' o '2', el programa solicita la cantidad a convertir y realiza la conversión utilizando la función convertir\_monedas.

**Opción de Salida:**

Si el usuario selecciona la opción '3', el programa imprime un mensaje de despedida y finaliza su ejecución.

# Ejercicio no 2:

Se desea elaborar un algoritmo que permita determinar la temperatura equivalente en grados centígrados a la cantidad de grados Fahrenheit actuales en la ciudad de New York.

**Solución al ejercicio**

Para convertir grados Fahrenheit a grados Centígrados, se usa la fórmula:

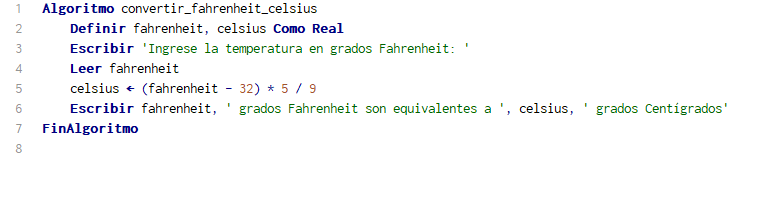
**Celsius = (Fahrenheit − 32) \***

**Pasos de Lectura y Escritura:**

* Se solicita al usuario que ingrese la temperatura en grados Fahrenheit.
* Se lee el valor ingresado por el usuario.
* Se realiza la conversión utilizando la fórmula matemática correspondiente.
* Se muestra el resultado de la conversión al usuario.

A continuación se muestra en formato de imagen el proceso de pseudocódigo y diagrama de flujo en PSeInt y código de programación para el problema planteado:

*Pseudocódigo* (Psint, 2024)



1. **Definir fahrenheit, celsius Como Real:**

Este paso se utiliza para declarar las variables fahrenheit y celsius como números reales (decimales). Esto asegura que las operaciones matemáticas, que pueden incluir decimales, se realicen con precisión.

1. **Escribir** 'Ingrese la temperatura en grados Fahrenheit: ':

La instrucción Escribir se usa para mostrar un mensaje al usuario pidiéndole que ingrese la temperatura en grados Fahrenheit.

1. **Leer fahrenheit:**

La instrucción Leer captura la entrada del usuario y la almacena en la variable fahrenheit. Esta entrada es la temperatura en grados Fahrenheit que se desea convertir.

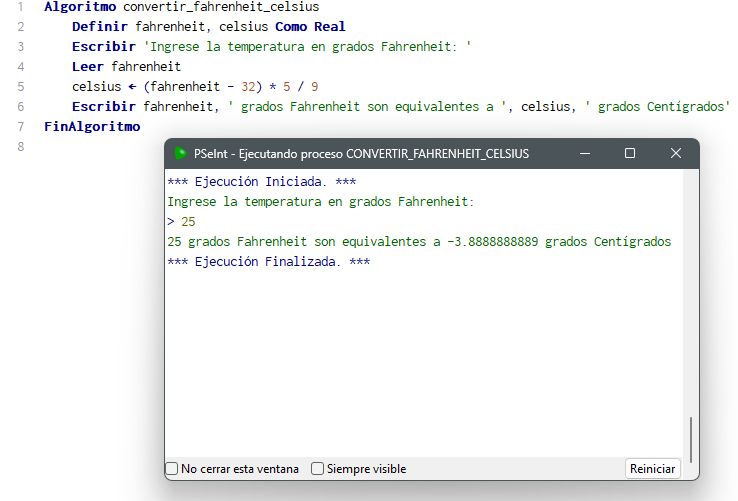
1. **celsius <- (fahrenheit - 32) \* 5 / 9:**

Esta línea ejecuta la fórmula de conversión de Fahrenheit a Celsius. Primero, se resta 32 de la temperatura en Fahrenheit, luego el resultado se multiplica por 5 y se divide por 9. El resultado se almacena en la variable celsius.

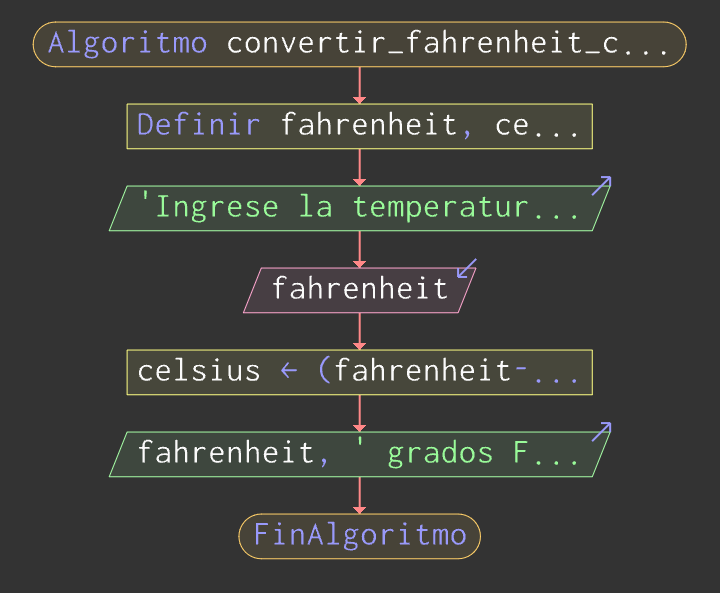
1. **Escribir fahrenheit, ' grados Fahrenheit son equivalentes a ', celsius, ' grados Centígrados':**

Esta instrucción utiliza Escribir para mostrar el resultado de la conversión al usuario. Indica cuántos grados Celsius equivalen a la temperatura en Fahrenheit que fue ingresada.

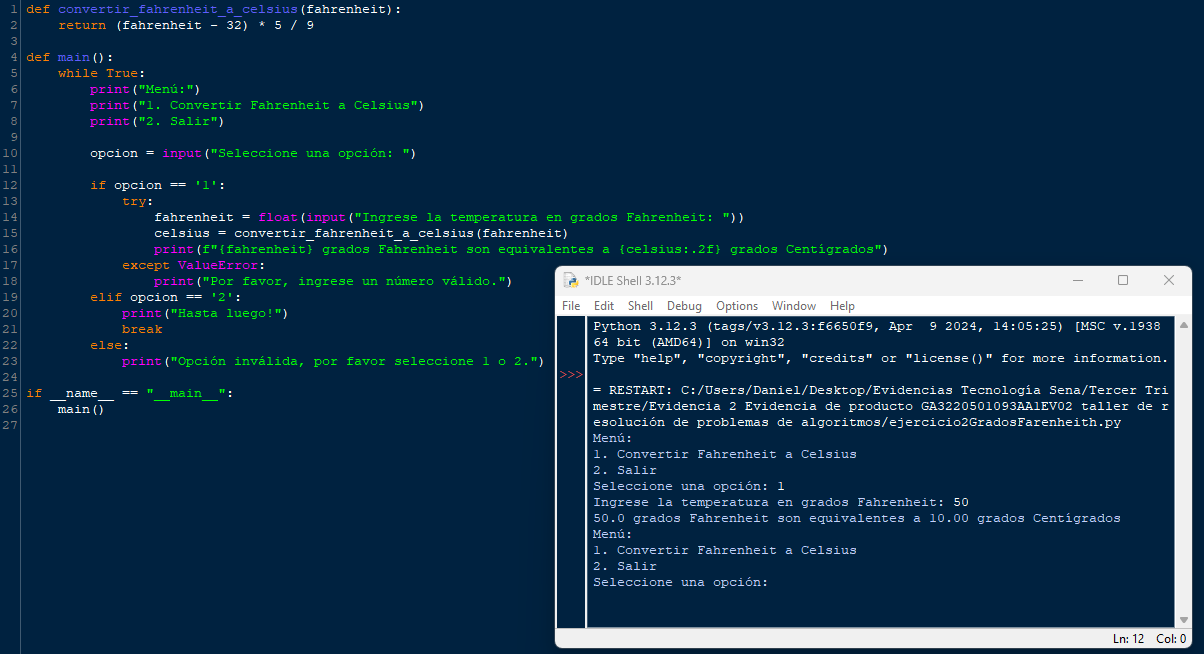
*Resultados (Psint, 2024)*

****

*Diagrama de flujo (Psint, 2024)*

**

*Algoritmo de programación en python - Resultados (Python.org, 2024)*

**

**Definición de la Función convertir\_fahrenheit\_a\_celsius**:

Esta función toma una temperatura en grados Fahrenheit y la convierte a grados Centígrados usando la fórmula de conversión: Celsius = (Fahrenheit−32) × 5/9

**Función main con Ciclo while**:

La función main contiene un ciclo while que se ejecuta indefinidamente hasta que el usuario decide salir.

**Mostrar el Menú**: Se presenta un menú con dos opciones al usuario: convertir temperaturas o salir.

**Leer la Opción del Usuario**: Se solicita al usuario que seleccione una opción. Si la opción es '1', el programa procede a realizar la conversión; si es '2', el programa finaliza; cualquier otra entrada se considera inválida.

**Conversión de Temperatura**:

Si el usuario selecciona la opción '1', se le pide que ingrese la temperatura en grados Fahrenheit. El programa lee este valor, intenta convertirlo a un número de punto flotante y luego usa la función convertir\_fahrenheit\_a\_celsius para realizar la conversión.

**Ejercicio no 3:**

**Descripción del ejercicio:** Suponiendo que nos encontramos descansando en nuestra casa en una ciudad de Colombia, necesitamos elaborar un plan detallado para llegar a tiempo a nuestro sitio de trabajo al día siguiente.

**Solución al ejercicio**

**Se definen las variables** ***ciudad\_origen, ciudad\_destino, distancia, velocidad y tiempo.*** Las primeras dos son de tipo texto para almacenar el nombre de las ciudades de origen y destino, respectivamente. Las tres variables restantes son de tipo real para almacenar la distancia del viaje, la velocidad promedio y el tiempo estimado de viaje, respectivamente.

**Solicitud de información al usuario**:

Se le pide al usuario que ingrese la ciudad de origen, la ciudad de destino, la distancia del viaje en kilómetros y la velocidad promedio a la que planea conducir en kilómetros por hora. Estos valores son capturados utilizando la instrucción Leer.

**Cálculo del tiempo estimado de viaje**:

Se calcula el tiempo estimado de viaje dividiendo la distancia entre la velocidad. Este cálculo se realiza utilizando la fórmula básica de la física: tiempo = distancia / velocidad.

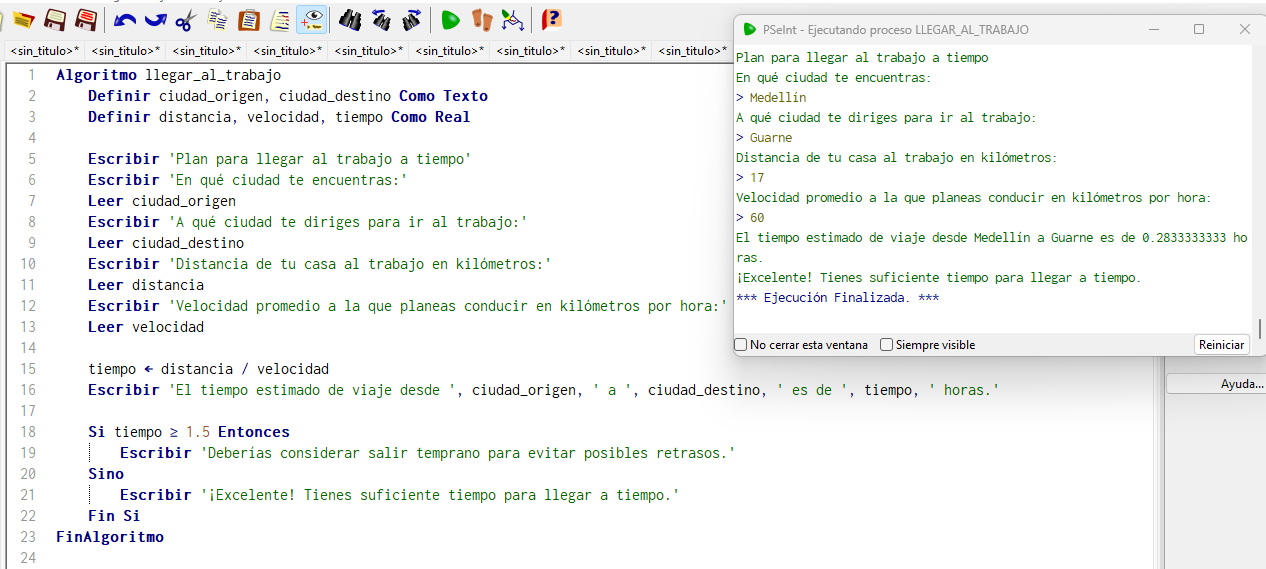
**Mostrar el tiempo estimado de viaje al usuario**:

Se utiliza la instrucción Escribir para mostrar al usuario el tiempo estimado de viaje calculado en el paso anterior. Se muestra el tiempo estimado de viaje desde la ciudad de origen a la ciudad de destino.

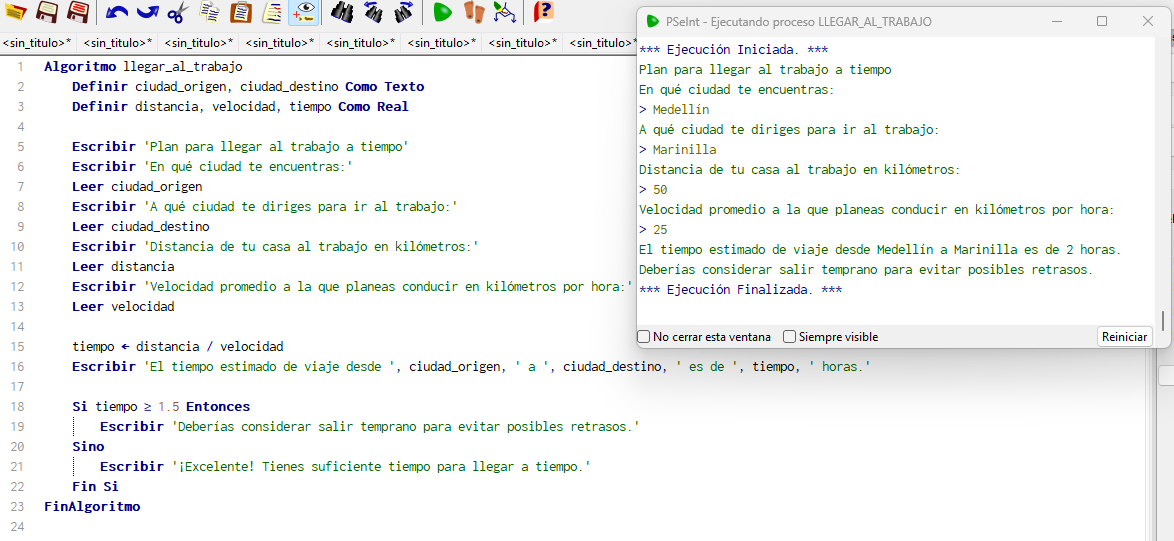
**Condición para sugerir acciones**:

Se evalúa si el tiempo estimado de viaje es mayor o igual a 1.5 horas. Si es así, se sugiere al usuario considerar salir temprano para evitar posibles retrasos. En caso contrario, se muestra un mensaje indicando que tiene suficiente tiempo para llegar a tiempo.

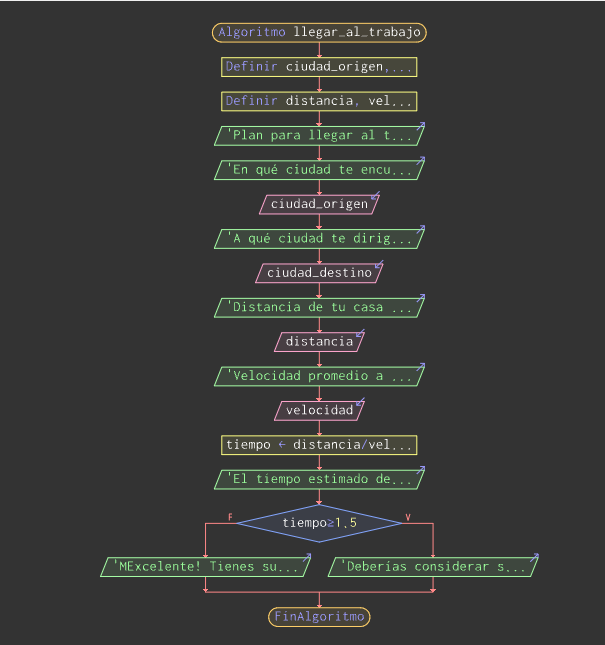
*Pseudocódigo condición uno (Psint, 2024)*



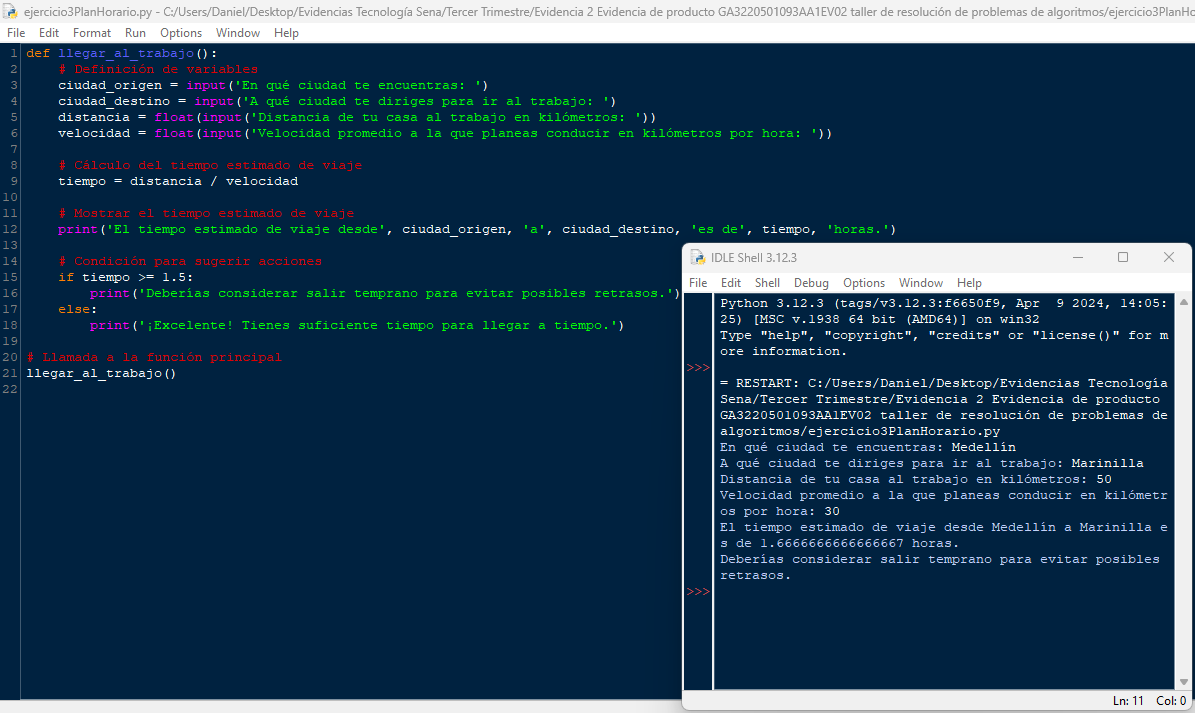
*Pseudocódigo condición Dos (Psint, 2024)*



*Diagrama de flujo (Psint, 2024)*



*Algoritmo de programación en python - Resultados (Python.org, 2024)*



**Función llegar\_al\_trabajo()**:

Esta es la función principal que ejecuta todo el proceso de planificación del viaje al trabajo.

**Definición de variables**:

Se definen las variables ciudad\_origen, ciudad\_destino, distancia y velocidad para almacenar la información proporcionada por el usuario. Las dos primeras son de tipo texto para almacenar el nombre de las ciudades de origen y destino, respectivamente. Las dos últimas son de tipo flotante para almacenar la distancia del viaje en kilómetros y la velocidad promedio en kilómetros por hora.

**Cálculo del tiempo estimado de viaje**:

Se calcula el tiempo estimado de viaje dividiendo la distancia entre la velocidad. Esto se realiza utilizando la fórmula básica de la física: tiempo = distancia / velocidad.

**Mostrar el tiempo estimado de viaje al usuario**:

Se utiliza la función print() para mostrar al usuario el tiempo estimado de viaje calculado en el paso anterior. Se muestra el tiempo estimado de viaje desde la ciudad de origen a la ciudad de destino.

**Condición para sugerir acciones**:

Se evalúa si el tiempo estimado de viaje es mayor o igual a 1.5 horas. Si es así, se sugiere al usuario considerar salir temprano para evitar posibles retrasos. En caso contrario, se muestra un mensaje indicando que tiene suficiente tiempo para llegar a tiempo.

**Ejercicio no 4:**

Suponiendo que tengo habilidades en la elaboración de comida necesito elaborar un arroz con pollo para 5 personas.

**Solución al ejercicio**

**Definición de variables:**

Se definen las variables personas\_adultos, personas\_niños, total\_personas para representar la cantidad de adultos, la cantidad de niños y el total de personas respectivamente, todas como enteros.

Se definen las variables cantidad\_arroz, cantidad\_pollo, cantidad\_agua, cantidad\_sal, cantidad\_condimentos para representar las cantidades necesarias de arroz, pollo, agua, sal y condimentos respectivamente, todas como números reales.

**Entrada de datos:**

Se solicita al usuario ingresar la cantidad de adultos y niños.

**Cálculo de cantidades de ingredientes:**

Se calculan las cantidades de ingredientes basadas en el número de adultos y niños.

Si hay adultos presentes, se calculan las cantidades de arroz, pollo, agua, sal y condimentos multiplicando las cantidades por persona por los coeficientes adecuados (1.4, 0.3, 3, 0.5, 0.5 respectivamente).

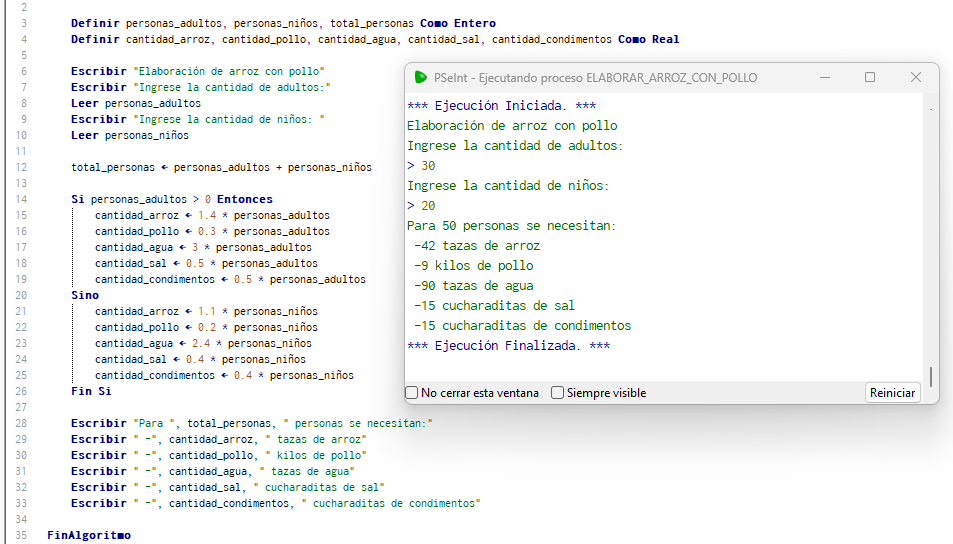
Si no hay adultos presentes (es decir, solo hay niños), se calculan las cantidades de ingredientes utilizando los coeficientes correspondientes para niños (1.1, 0.2, 2.4, 0.4, 0.4 respectivamente).

**Salida de resultados:**

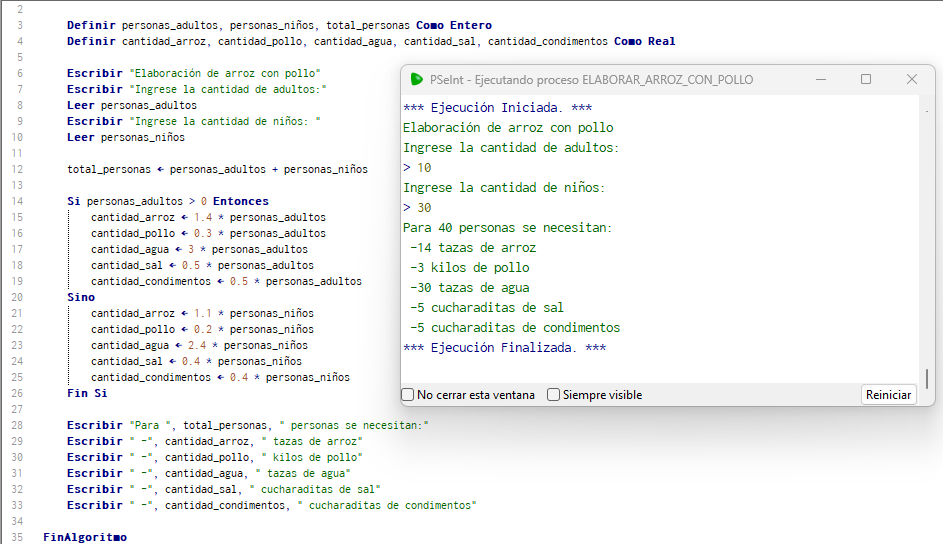
Se imprime un mensaje indicando las cantidades de ingredientes necesarias para el total de personas ingresado, incluyendo la cantidad de arroz, pollo, agua, sal y condimentos.

**Fin del algoritmo.**

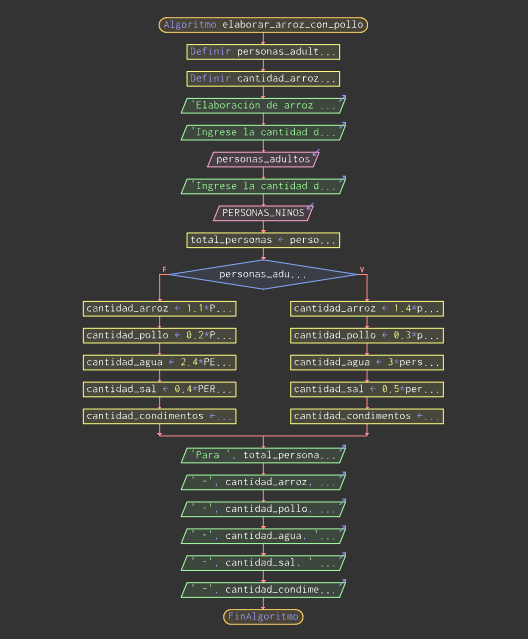
*Pseudocódigo ejemplo uno (Psint, 2024)*

****

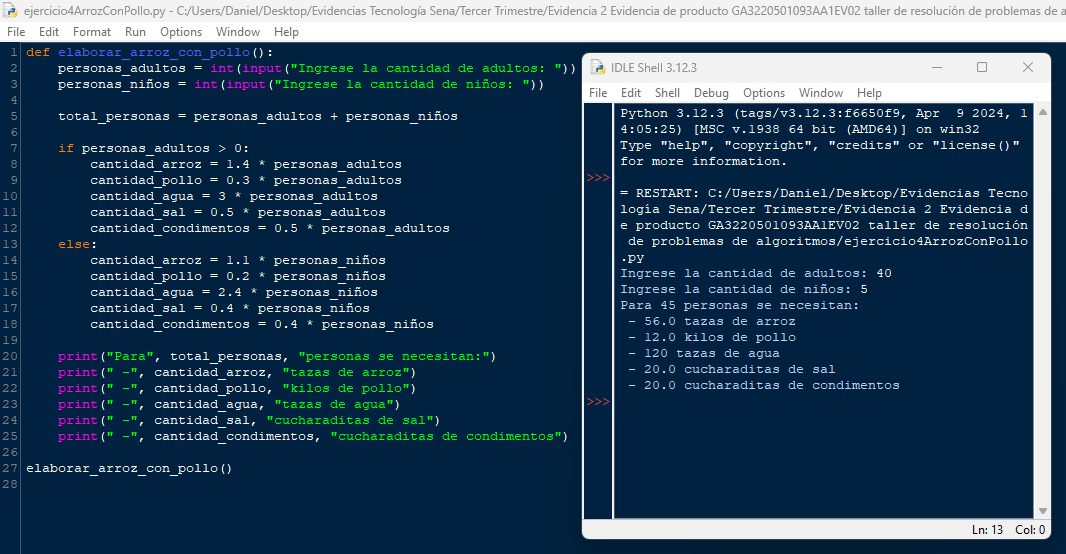
*Pseudocódigo ejemplo dos (Psint, 2024)*

****

*Diagrama de flujo (Psint, 2024)*

****

*Algoritmo de programación en python - Resultados (Python.org, 2024)*

****

1. **Definición de la función elaborar\_arroz\_con\_pollo():**

Esta función no acepta argumentos y simplemente solicita al usuario que ingrese la cantidad de adultos y niños que van a comer arroz con pollo.

1. **Entrada de datos:**

personas\_adultos = int(input("Ingrese la cantidad de adultos: ")): Solicita al usuario que ingrese la cantidad de adultos que van a comer arroz con pollo y almacena este valor como un número entero en la variable personas\_adultos.

personas\_niños = int(input("Ingrese la cantidad de niños: ")): Solicita al usuario que ingrese la cantidad de niños que van a comer arroz con pollo y almacena este valor como un número entero en la variable personas\_niños.

1. **Cálculo de ingredientes:**

total\_personas = personas\_adultos + personas\_niños: Calcula el número total de personas sumando la cantidad de adultos y niños.

La lógica para calcular la cantidad de ingredientes se basa en si hay adultos o no:

Si personas\_adultos > 0, se calculan las cantidades de arroz, pollo, agua, sal y condimentos basados en el número de adultos.

De lo contrario, se asume que solo hay niños y se calculan las cantidades de ingredientes basadas en la cantidad de niños.

1. **• Impresión de resultados:**

La función imprime la cantidad necesaria de cada ingrediente para el total de personas:

"Para", total\_personas, "personas se necesitan:": Imprime el número total de personas.

" -", cantidad\_arroz, "tazas de arroz", " -", cantidad\_pollo, "kilos de pollo", " -", cantidad\_agua, "tazas de agua", " -", cantidad\_sal, "cucharaditas de sal", " -", cantidad\_condimentos, "cucharaditas de condimentos": Imprime la cantidad necesaria de arroz, pollo, agua, sal y condimentos, respectivamente.

**Respecto a las incógnitas que se pueden presentar durante el desarrollo…**

Los problemas planteados en el taller de resolución de problemas de algoritmos requieren una cuidadosa consideración de diversos aspectos para su solución efectiva. Al abordar estos problemas, es esencial identificar claramente los elementos de entrada y los resultados esperados. Esto implica determinar qué información es esencial para resolver el problema, como la ciudad de origen y destino en el caso de la planificación del viaje al trabajo, o la cantidad de dólares y pesos colombianos en las conversiones de moneda. Además, se debe discernir entre la información relevante y la que puede prescindirse, agrupar los datos de entrada conocidos, identificar la incógnita o lo que se debe calcular, y considerar si los datos pueden categorizarse para simplificar el proceso de resolución. Respecto a los resultados esperados, es fundamental comprender qué información se está solicitando y en qué formato debe entregarse, ya sea un valor numérico, una descripción detallada o una recomendación específica

**Sección dos**

**¿Qué es un Diagrama de Flujo?**

Un diagrama de flujo es un tipo de diagrama que explica visualmente un proceso o flujo de trabajo. Mediante el uso de símbolos y definiciones estandarizados, puede crear una representación visual útil de los diversos pasos y puntos de decisión de cualquier proceso.

Los diagramas de flujo han sido populares durante muchos años en diversos campos, desde ingeniería y educación hasta programación de computadoras y gestión de proyectos. Son útiles para comunicar procesos simples y complejos y son increíblemente dinámicos y versátiles. Al mirar un diagrama de flujo, un espectador debe ser capaz de obtener una visión general de alto nivel de un proceso.

Por lo general, se crea un diagrama de flujo utilizando varios símbolos, cada uno representando un paso diferente dentro de una secuencia o proceso. Estos diagramas generalmente incluyen una variedad de elementos, como acciones, materiales, servicios, entradas y salidas. Si necesita tomar una decisión pero no está seguro del proceso y los diferentes elementos necesarios para informar esa decisión, el uso de un diagrama de flujo podría simplificar enormemente el proceso.

**Beneficios de crear un diagrama de flujo**

Los diagramas de flujo son una gran herramienta para visualizar procesos en su vida personal y profesional. Ofrecen varias ventajas que pueden ayudarlo a usted y a su equipo. Estos son algunos de los beneficios de usar un diagrama de flujo:

Claridad visual

El principal beneficio de usar diagramas de flujo es que le permiten comprender operaciones complejas en un formato simple. Muchas empresas en fabricación, logística y administración los usan para trazar secuencias importantes. Además, los diagramas de flujo pueden ayudar a las partes interesadas a comprender los flujos de trabajo que emplea su empresa.

Comunicación mejorada

Los diagramas de flujo son útiles para reuniones donde su equipo necesita comprender y comunicar procesos importantes. Al simplificar operaciones complejas con símbolos y definiciones de diagramas de flujo, puede comunicar claramente información que de otro modo sería difícil de discutir.

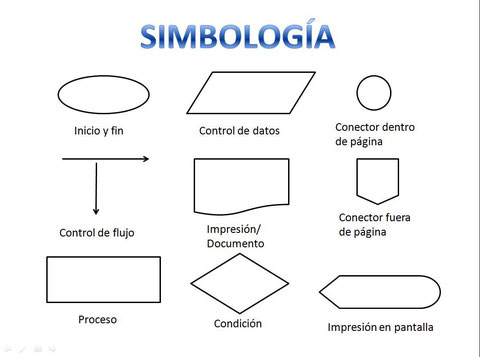
Documentación adecuada

El uso de diagramas de flujo para trazar procesos dentro de su negocio es una excelente manera de documentar lo que se debe hacer. También le ayuda a comprender el progreso de su empresa en completar una tarea. Al utilizar diagramas de flujo digitales, puede incorporar documentación sin papel en su flujo de trabajo.

**Cuando usar un diagrama de flujo**

Los diagramas de flujo se utilizan en una variedad de contextos e industrias diferentes. Realmente no importa en qué sector opere usted o su empresa – se puede usar un diagrama de flujo para ayudar a comprender los procesos y las operaciones fácilmente. (Miro, 2024)

*Simbología básica para los diagramas de flujo (TutosPOO, 2024)*



*A continuación se da una breve descripción del uso de cada conector*

**Óvalo (Inicio y Fin):**

Indica el inicio o el final de un proceso.

Uso: Representa el punto de entrada o salida del diagrama.

Ejemplo: "Inicio" al comenzar un proceso y "Fin" al concluirlo.

**Paralelogramo (Control de Datos):**

Descripción: Representa la entrada o salida de datos en el proceso.

Uso: Utilizado para indicar operaciones de entrada/salida, como leer datos del usuario o mostrar resultados.

Ejemplo: "Leer longitud" o "Imprimir resultado".

**Círculo (Conector Dentro de Página):**

Descripción: Indica la conexión a otra parte del diagrama dentro de la misma página.

Uso: Facilita la continuidad del diagrama en la misma página cuando hay líneas de flujo largas.

Ejemplo: Usado para mantener el diagrama limpio y evitar líneas cruzadas.

**Flecha (Control de Flujo):**

Descripción: Indica la dirección del flujo del proceso.

Uso: Muestra la secuencia en que se ejecutan las acciones.

Ejemplo: Conecta los distintos símbolos mostrando el orden del proceso.

**Rectángulo Ondulado (Impresión/Documento):**

Descripción: Representa un documento generado en el proceso.

Uso: Indica la creación de un documento o impresión.

Ejemplo: "Generar reporte".

**Pentágono (Conector Fuera de Página):**

Descripción: Conecta partes del diagrama cuando el flujo continúa en otra página.

Uso: Facilita la conexión de diferentes secciones de un diagrama extenso.

Ejemplo: Conectar a un diagrama de flujo complementario en otra hoja.

**Rectángulo (Proceso):**

Descripción: Representa una operación o un paso en el proceso.

Uso: Utilizado para cualquier tipo de acción o tarea dentro del proceso.

Ejemplo: "Calcular área".

**Diamante (Condición):**

Descripción: Indica un punto donde se toma una decisión que ramifica el flujo en diferentes caminos.

Uso: Utilizado para condiciones de decisión con salidas alternativas, como sí/no.

Ejemplo: "¿El valor es mayor a 10?".

**Rectángulo con Bordes Redondeados (Impresión en Pantalla):**

Descripción: Representa una salida en pantalla, como la impresión de datos en un monitor.

Uso: Indica la salida de información en una pantalla digital.

Ejemplo: "Mostrar resultado en pantalla".

**¿QUE ES UN PSEUDOCÓDIGO?**

Una de las mejores formas de aprender a programar es empezar por los diagramas de flujo y el pseudocódigo. Ambos facilitan al estudiante su inmersión en la resolución de problemas mediante algoritmos.

El pseudocódigo es una forma de expresar los distintos pasos que va a realizar un programa, de la forma más parecida a un lenguaje de programación. Su principal función es la de representar por pasos la solución a un problema o algoritmo, de la forma más detallada posible, utilizando un lenguaje cercano al de programación. El pseudocódigo no puede ejecutarse en un ordenador ya que entonces dejaría de ser pseudocódigo, como su propio nombre indica, se trata de un código falso (pseudo = falso), es un código escrito para que lo entienda el ser humano y no la máquina.

Aprender a escribir pseudocódigo para la resolución de un problema permite hacer mucho más sencilla su programación en un lenguaje convencional, por lo que si estás interesado en comenzar tu formación como programador y no tienes conocimientos previos, resulta muy recomendable y conveniente formarse en pseudocódigo antes de empezar a estudiar cualquier lenguaje de programación.

Podemos considerar al pseudocódigo como un lenguaje intermedio, que se encuentra en medio de nuestro propio lenguaje y el lenguaje de programación que entiende el ordenador.

## **Principales características del pseudocódigo**

Para comprender qué es el pseudocódigo podemos utilizar un símil: los planos de una casa sería el pseudocódigo y la casa en sí el programa.

Su principal característica es la de representar un método que facilita la programación y solución del algoritmo del programa. También se caracteriza por ser una forma de representación, fácil de utilizar y de manipular, que simplifica el paso del programa, al lenguaje de programación.

Otra característica que tiene el pseudocódigo es su independencia al código en el que se va a escribir el programa, proporcionando un método que facilita la posterior programación y la resolución del algoritmo del programa.

Conviértete en un Backend Developer

Domina los lenguajes de programación más demandados. Accede a cursos, talleres y laboratorios para crear proyectos con Java, Python, PHP, Microsoft .NET y más

[Comenzar gratis ahora](https://openwebinars.net/registro/prueba-gratis/?ref=blog)

## Ventajas y desventajas del pseudocódigo

Las tareas más complejas o repetitivas pueden representarse de forma más sencilla ya que está escrito en un lenguaje sencillo y no estructurado que permite una transición sencilla al lenguaje de programación, más complejo y estructurado. Tener un programa escrito en pseudocódigo facilita la tarea de programar en un lenguaje formal y mejora la calidad en la resolución de problemas, además de reducir el espacio necesario a la hora de desarrollar un problema.

El pseudocódigo llega donde el diagrama de flujo no lo hace. La solución de un diagrama de flujo suele ser la ideal, pero no suele ser fácil de implementar al crear el programa. El pseudocódigo permite que el diseño del programa y su implementación sean muy parecidos.

La curva de aprendizaje del pseudocódigo es baja por lo que facilitan enormemente el aprendizaje de la programación y la iniciación a lenguajes de programación más avanzados y complejos. Por lo tanto, se trata de una herramienta educativa interesante.

El pseudocódigo, al ser independiente del lenguaje de programación, permite que su uso se pueda aplicar utilizando diferentes lenguajes y permitiendo que el programador no tenga que ser la misma persona que escribió el pseudocódigo.

Una de las desventajas del uso de pseudocódigo es la falta de normas, que puede hacer que la lógica de un programa, resulte complicada de ver por el programador que va a implementar este pseudocódigo. Además, en el caso de problemas muy extensos, puede llegar a ser difícil de entender.

## Componentes y sintaxis del pseudocódigo

Para escribir programas utilizando pseudocódigo es necesario seguir unas pautas o normas de sintaxis para que puedan ser leídos y comprendidos por los programadores a la hora de pasarlos a un lenguaje de programación. Es muy útil utilizar herramientas que faciliten esta escritura de pseudocódigo, como es el caso de [PSeInt](http://pseint.sourceforge.net/), que asiste con un simple e intuitivo pseudolenguaje en español y que además incluye un editor de diagramas de flujo.

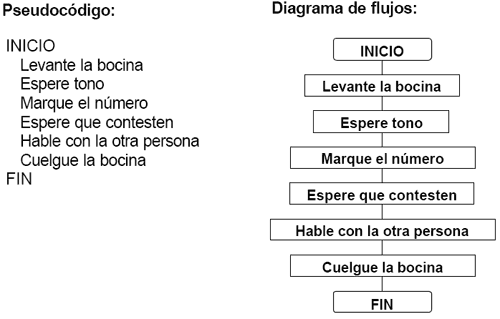
Un programa escrito en pseudocódigo debe permitir instrucciones primitivas, de proceso, de control, de descripción y compuestas.

## Opciones del lenguaje (perfiles)

El lenguaje que utilizamos para construir el pseudocódigo no es estándar. Podemos añadir o eliminar algunas reglas de sintaxis sin ningún problema. En la opción Configurar-Opciones del Lenguaje (perfiles), podemos escoger las características del pseudocódigo que vamos a utilizar. Tenemos tres alternativas:

* Escoger un perfil que define un pseudocódigo utilizado en distintos centros educativos y universidades.
* Perfil flexible: Está escogido por defecto, y no es muy exigente con las reglas que hay que utilizar para escribir el pseudocódigo.
* Perfil estricto: Establece unas reglas que hacen que el pseudocódigo se parezca más a un lenguaje de programación: se debe definir las variables y sus tipos, las instrucciones deben terminar en punto y coma. (Robledano, 2019)

*Pseudocódigo y diagrama de flujo (desarrolloweb.com, 2022)*



**CONCLUSIONES**

La actividad ha permitido desarrollar y aplicar conocimientos en la resolución de problemas mediante algoritmos, tanto en pseudocódigo como en diagramas de flujo. Estos enfoques ayudan a estructurar el pensamiento lógico y sistemático necesario para abordar problemas complejos de manera eficiente.

Uso de Herramientas y Lenguajes de Programación:

La utilización de Python y PSeInt ha facilitado la implementación práctica de los algoritmos diseñados. Python, con su sintaxis clara y directa, junto con PSeInt, que permite una visualización simple de los algoritmos, se complementan para ofrecer una experiencia de aprendizaje completa y efectiva.

Desglose y Análisis de Problemas:

Al descomponer los problemas en elementos de entrada y resultados esperados, se ha resaltado la importancia de identificar información relevante y prescindir de la irrelevante. Este proceso es crucial para la resolución efectiva de problemas, asegurando que se enfoquen los recursos y esfuerzos en los aspectos que realmente importan.

Diagramas de Flujo y Simbología Estándar:

La creación de diagramas de flujo ha proporcionado una representación visual clara de los procesos y decisiones dentro de un algoritmo. El uso de símbolos estandarizados facilita la comprensión y comunicación de los procesos, haciendo más accesible la información a todos los involucrados en el desarrollo y revisión de los algoritmos.

Metodología para la Planificación de Soluciones:

La planificación detallada y estructurada, como la realizada para los ejercicios de conversión de moneda, cálculo de temperatura, planificación de viajes y preparación de comida, demuestra la efectividad de un enfoque metódico para la resolución de problemas. Este enfoque asegura que todos los aspectos del problema se consideren y que las soluciones sean completas y precisas.

Mejora de Habilidades en Programación y Resolución de Problemas:

A través de la práctica con pseudocódigo, diagramas de flujo y la implementación en Python, se ha mejorado significativamente la habilidad para resolver problemas mediante la programación. Este conjunto de habilidades es esencial no solo para el ámbito académico, sino también para el desarrollo profesional en campos relacionados con la tecnología y la ingeniería.

# REFERENCIAS

desarrolloweb.com. (22 de noviembre de 2022). *desarrolloweb*. Obtenido de desarrolloweb.com: https://desarrolloweb.com/articulos/2199.php

Miro. (1 de Abril de 2024). *Miro*. Obtenido de miro.com: https://miro.com/flowchart/what-is-a-flowchart/

plataforma.josedomingo.org. (17 de March de 2024). *plataforma.josedomingo*. Obtenido de plataforma.josedomingo.org: https://plataforma.josedomingo.org/pledin/cursos/programacion/curso/u05/

Psint. (31 de may de 2024). captura de pantalla pseudocódigo. Medellín, Antioquia, Colombia.

Psint. (31 de May de 2024). Diagrama de flujo. Medellín, Antioquia, Colombia.

Python.org. (31 de May de 2024). Función en lenguaje de programación python. Medellín, Antioquia, Colombia.

Robledano, A. (18 de junio de 2019). *openwebinars*. Obtenido de openwebinars.net: https://openwebinars.net/blog/que-es-pseudocodigo/

TutosPOO. (1 de Abril de 2024). *TutosPOO*. Obtenido de tutospoo.jimdofree.com: https://tutospoo.jimdofree.com/fundamentos/diagramas-de-flujo/