Laboratorio: Algoritmos, diagramas de flujo y código fuente

Asmar, I. (1999). *Métodos numéricos. Un primer curso*. 2ª edición. Medellín: Universidad Nacional de Colombia. Recuperado de <http://bdigital.unal.edu.co/4831/>

De acuerdo con Asmar (1999), «un método numérico es un procedimiento mediante el cual se obtiene, casi siempre de manera aproximada, la solución de ciertos problemas realizando cálculos puramente aritméticos y lógicos» (p.2).

El procedimiento o algoritmo, que comprende una secuencia de operaciones y genera una solución numérica al problema o un mensaje, puede representarse mediante un diagrama de flujo. Este último, reduce considerablemente el tiempo de implementación del método numérico en cualquier lenguaje de programación.

**Objetivos**

El desarrollo del laboratorio te permitirá apropiar una herramienta para generar código fuente en diferentes lenguajes de programación a partir de un diagrama de flujo, al tiempo que fortalecerás conceptos y procedimientos relacionados con algoritmos, diagramas de flujo, pruebas de escritorio y programación, esenciales en la implementación de métodos numéricos.

**Descripción de la actividad**

Preparación del laboratorio:

Antes de la realización del laboratorio, es importante que cuentes con las siguientes herramientas instaladas en tu computador:

* PSeInt: herramienta libre que comprende un sencillo pseudolenguaje en español y un editor de diagramas de flujo. Permite exportar el pseudocódigo como código en C, C++, C#, HTML, Java, JavaScript, PHP y Python, entre otros lenguajes de programación. Disponible en <http://pseint.sourceforge.net/>
* El Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) de tu preferencia, que permita programar en alguno de los lenguajes listados en el numeral 1. Si aún no cuentas con un IDE, puedes instalar Dev C++ que está disponible en <http://orwelldevcpp.blogspot.com/>

Adicionalmente, debes descargar los siguientes archivos: sol\_cuadratica.psc y minimo.psc.

Desarrollo:

* **Encontrar las raíces reales de la ecuación .**

Lee el ejemplo 2 (página 10) del manual de la asignatura: *Curso de métodos numéricos*, de Virginia Muto Foresi.

|  |  |  |
| --- | --- | --- |
| Abre PSeInt y luego el archivo sol\_cuadratica.psc mediante la opción Archivo/Abrir. |  |  |
| Utiliza la opción Archivo/Editar Diagrama de Flujo, para abrir una nueva ventana con el diagrama de flujo del pseudocódigo. |  |  |
| Compara el diagrama de flujo con el del ejemplo 2 del manual de la asignatura. |  |  |
| Realiza una prueba de escritorio. Para hacerlo, regresa a la ventana principal de PSeInt, selecciona la opción Prueba de Escritorio ubicada a la derecha en la pantalla y luego agrega cuatro columnas en la ventana inferior para las variables A, B, C y  B2 – 4AC. |  |  |
| En la ventana del editor de diagramas de flujo, selecciona la opción para ejecutar el algoritmo paso a paso. Observa que cada paso se va marcando sobre el diagrama de flujo al tiempo que se actualiza la tabla de la Prueba de Escritorio. |  |  |
| Si realizas la prueba de escritorio con los valores A = 1, B = 2 y C = -3, el resultado debe ser X1 = -3 y X2 = 1. Repite la prueba de escritorio con A = 1, B = 3 y C = 5; en este caso el programa debe indicar que no hay raíces reales. | | |

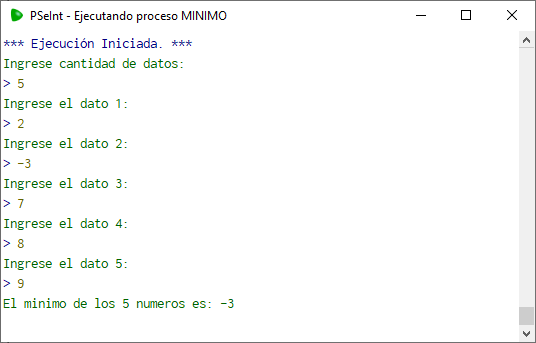
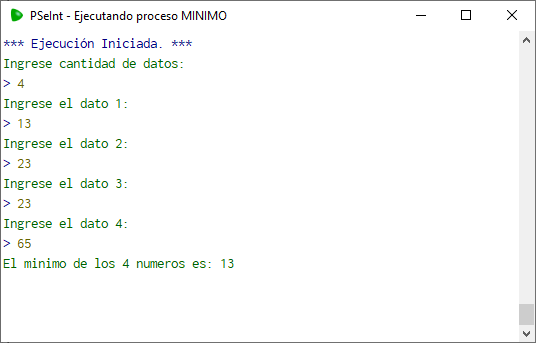
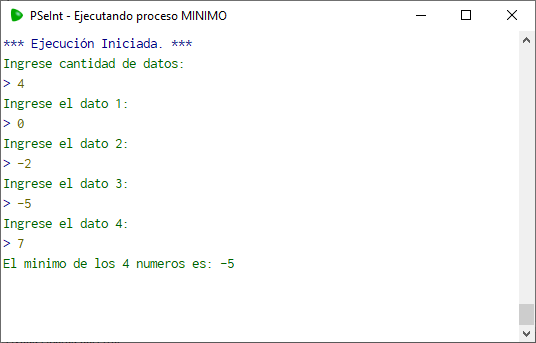
|  |  |  |
| --- | --- | --- |
| Exporta el pseudocódigo al lenguaje de tu preferencia utilizando la opción Archivo/Exportar. |  |  |
| Abre el código fuente en el IDE de tu preferencia. En este ejemplo utilizamos Dev C++. Compila el programa y en caso de ser necesario, corrige los errores que se reporten. |  |  |
| Ejecuta el programa. Realiza pruebas con los mismos valores empleados en la prueba de escritorio. Compara los resultados. |  |  |

* **Buscar el mínimo de N números reales.**

Lee el ejemplo 3 (página 11) del manual de la asignatura: *Curso de métodos numéricos*, de Virginia Muto Foresi.

Abre el archivo **minimo.psc** y compara el diagrama de flujo con el del ejemplo 3 del manual de la asignatura.

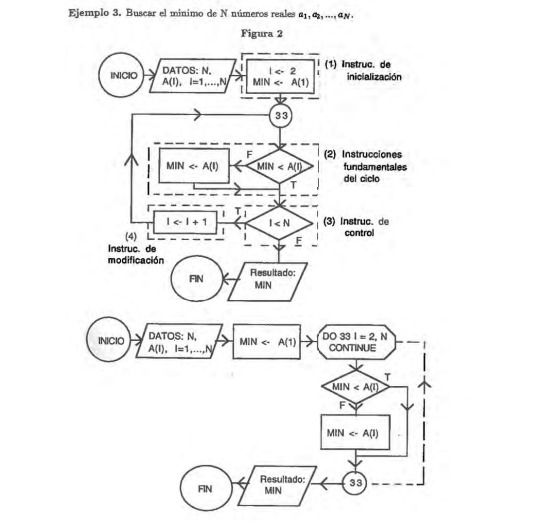
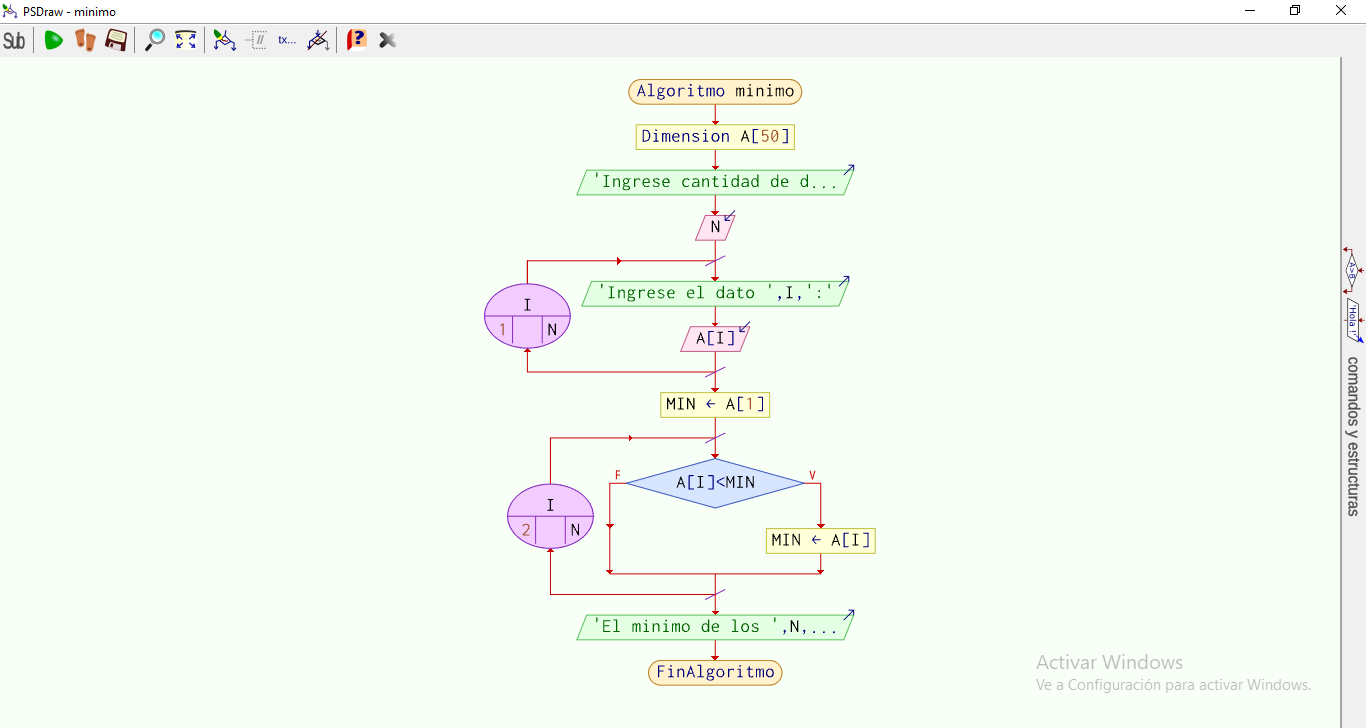
Realiza las pruebas de escritorio con los siguientes conjuntos de datos:

* + {2, -3, 7, 8, 9}
  + 
  + {13, 23, 23, 65}
  + 
  + {0, -2, -5, 7}
  + 

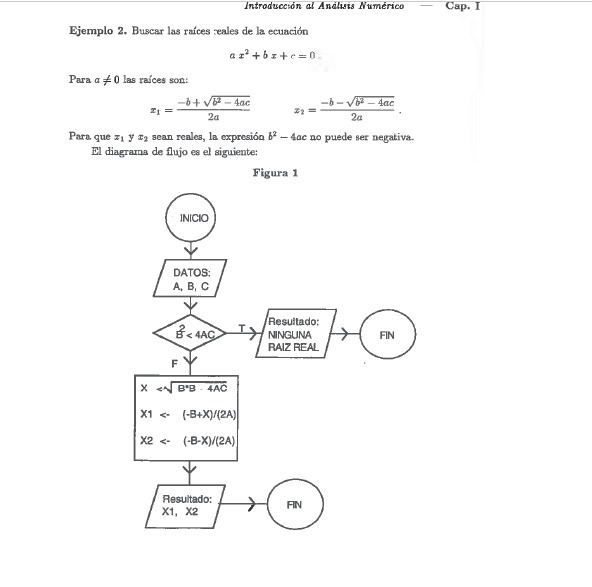
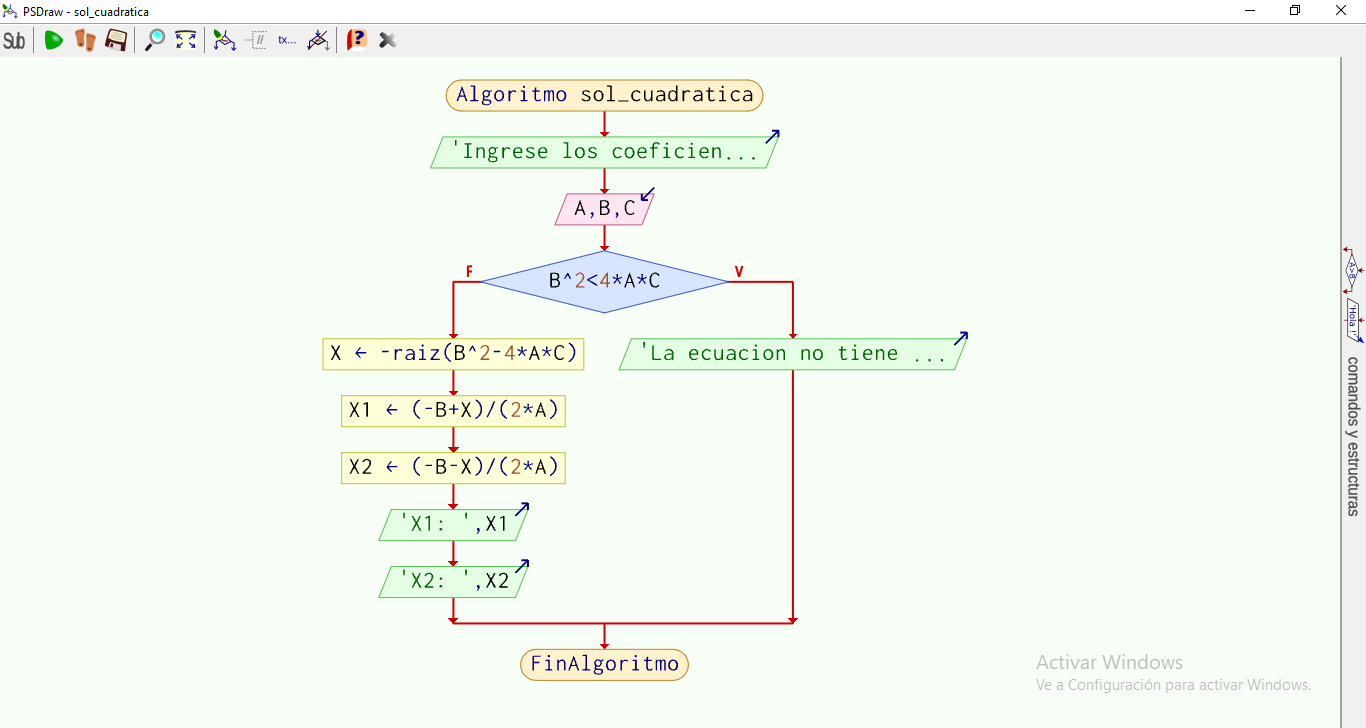
Exporta el pseudocódigo al lenguaje de tu preferencia. Utiliza el IDE para abrir el código fuente y verificarlo con los mismos conjuntos de datos del paso anterior. Compara los resultados.

**Comparaciones**

Minimo

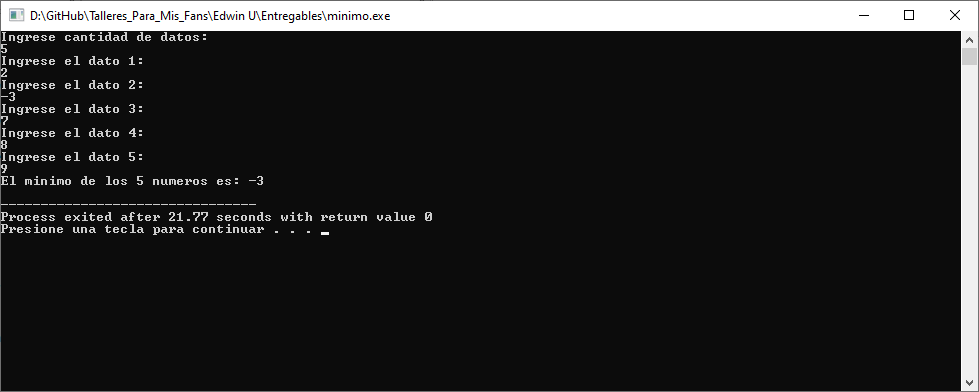
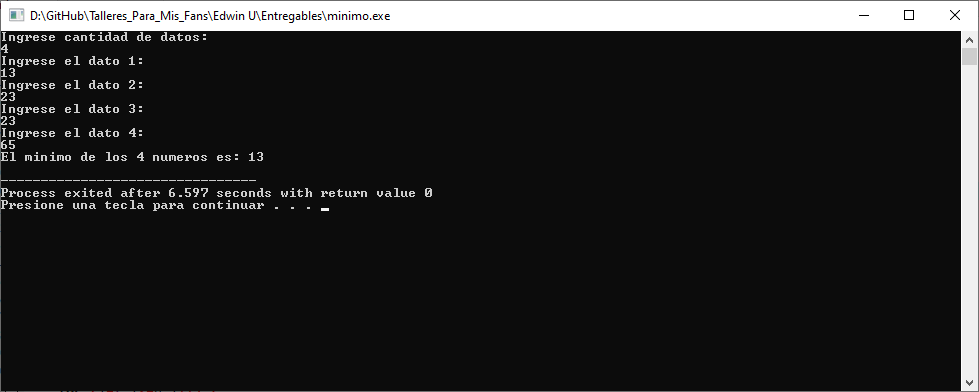
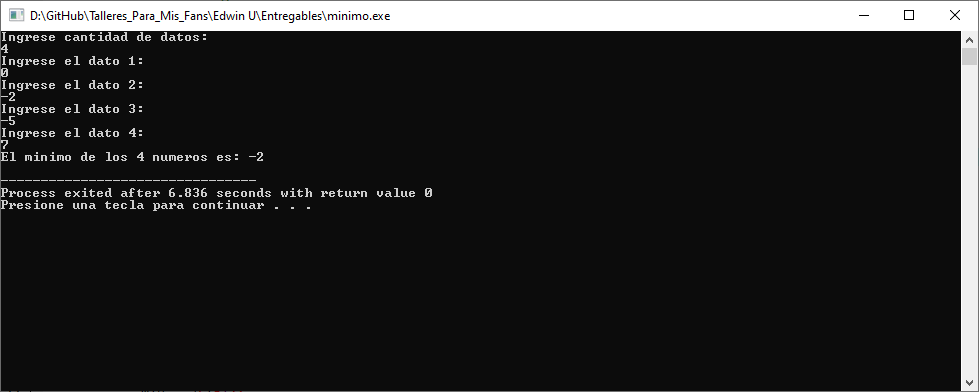
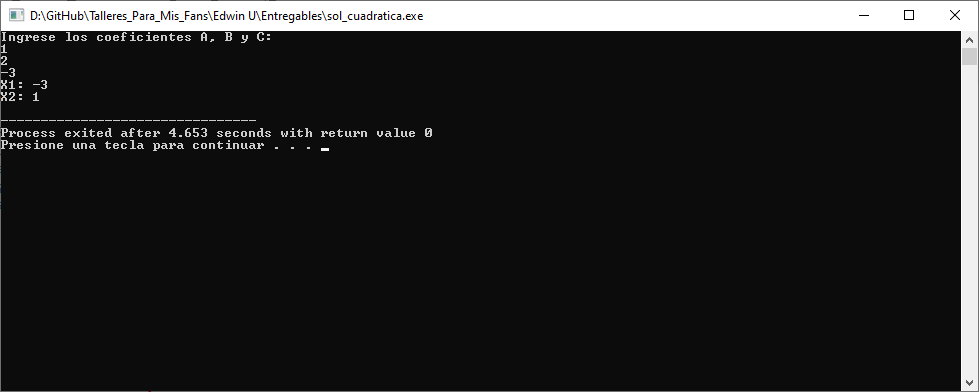
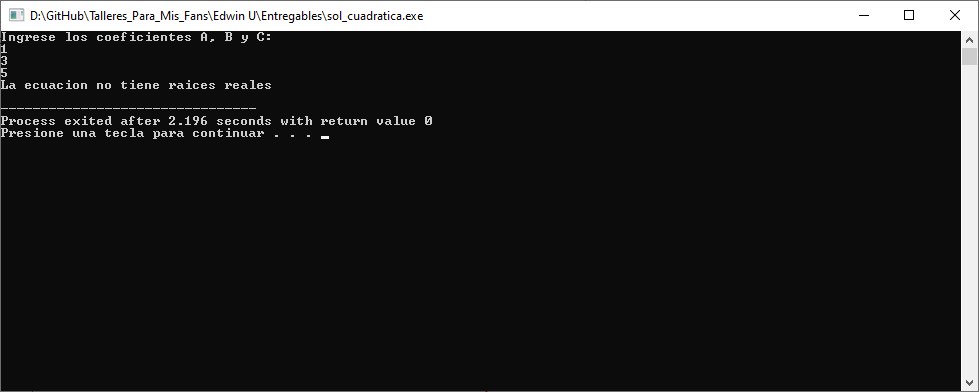
 

Cuadratica

 ****

**Entrega**

Al finalizar el laboratorio, entrega un documento de Microsoft Office Word con la siguiente información:

* Código fuente generado para el programa que encuentra las raíces reales de la ecuación y una impresión de pantalla con la ejecución en el IDE elegido
  + En la carpeta entregables está el código fuente con el nombre minimo.cpp.
  + {2, -3, 7, 8, 9}
  + 
  + {13, 23, 23, 65}
  + 
  + {0, -2, -5, 7}
  + 
* Código fuente generado para el programa que busca el mínimo de N números reales y una impresión de pantalla con la ejecución en el IDE elegido.
  + En la carpeta Entregables se encuentra el código fuente con el nombre sol\_cuadratica.cpp.
  + 
  + 
* Conclusiones del laboratorio, considerando: ventajas y desventajas de utilizar diagramas de flujo, importancia de las pruebas de escritorio, y ajustes realizados al código fuente generado desde PSeInt, entre otros aspectos. Máximo una (1) página.
  + Podemos concluir que la herramienta PSeInt, nos permite entender cómo se estructuran los problemas para convertirlos en algoritmos y a través de los diagramas podemos organizar las ideas, organizar el orden y el flujo que debe seguir el programa, además es muy bueno que pueda generar código en diferentes lenguajes de programación, además su cercanía al español ayuda a pensar los programas a diferencia de Dev C++, que lo hace más confuso.
  + Otra conclusión del laboratorio es que al comparar los diagramas del libro y los generados desde el código encontramos bastante similitud, lo que nos ayuda a entender cómo va a funcionar el algoritmo.
  + Diagramas de flujo
    - Ventajas:
      1. Ayudan a visualizar los problemas en un paso a paso, lo que los hace más fáciles de entender.
      2. Se puede crear los algoritmos a partir de los diagramas.
      3. Son fáciles de leer, si se necesita entender un algoritmo, se puede leer y comprender su funcionamiento a partir de los diagramas.
      4. Es una forma de crear algoritmos con lenguaje universal, un algoritmo desarrollado en diagrama se puede comprender sin importar el lenguaje.
    - Desventajas:
      1. Se deben conocer lo que significa cada icono para saber cómo funciona el algoritmo.
      2. Si un algoritmo no es corto se puede extender mucho el diagrama, haciéndolo tedioso de leer.
      3. Difícil de cambiar, al ser gráficos corregir pasos anteriores suele ser un dolor de cabeza.
  + Son importantes ya que nos permiten verificar que el pseudocódigo hace lo que esperamos.
  + El código estaba bien, un ajuste fue al final colocar “Esperar 10 Segundos”, ya que la pantalla de ejecución desaparecía y no podía tomar los pantallazos.
  + En conclusión, a través de este laboratorio se cumplió el objetivo de introducirnos a la herramienta PSeInt, y dar el primer paso para el mundo de la programación y el análisis numérico a través de la programación. Tambien nos apropiamos del uso de los diagramas de flujo y sus ventajas.