Ciencia Digital
www.cienciadigital.org

**ISSN: 2602-8085**
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

# Métodos numéricos aplicados al conteo de operaciones para el cálculo del Costo computacional en Python

*Numerical methods applied to the counting of operations for the calculation of the Computational Cost in python*

Deysi Margoth Guanga Chunata, [1] Luis Enrique Sarmiento Torres, [2] Marianela de Jesús Inca Chunata. [3]

Recibido: 10-06-2019 / Revisado: 15-07-2019 /Aceptado: 14-08-2019/ Publicado: 06-09-2019

**Abstract.** DOI: https://doi.org/10.33262/cienciadigital.v3i3.3.825

In this article the quality of a numerical method is dealt with in a general way through the counting of operations to determine the storage, amount of time (number of operations), the effect by error of rounding of the programming codes in Python language on a microcontroller as a mechanism that allows inferring about the handling of large amounts of data in a comparative version of quality with the c ++ programming language. The relevance of applied mathematics in the evolution of the different technological tools at the service of humanity. The programming language contains a finite number of lines of code that allow controlled systems to execute operations. The computational cost is calculate by the quality of the programming algorithm by exemplifying the Gauss elimination method to run tasks without using interruptions.

**Keywords:** Quality, Code, Python programming language, Gaussian elimination, Numerical method.

[1]Escuela Superior Politécnica de Chimborazo, Ecuador. deysiguanga@espoch.edu.ec
[2] Petroamazonas EP, Sucumbíos, Ecuador. luis_sarmiento@petroamazonas.ec
[3] Secretaría Nacional de Educación Superior, Pichincha, Ecuador. minca@senescyt.gob.ec

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

**Resumen.**

En este artículo se aborda de manera general la calidad de un método numérico a través del conteo de operaciones para determinar el almacenamiento, cantidad de tiempo (número de operaciones), el efecto por error de redondeo de los códigos de programación en lenguaje Python sobre un microcontrolador como un mecanismo que permite inferir sobre el manejo de grandes cantidades de datos en una versión comparativa de calidad con el lenguaje de programación c++. La relevancia de la matemática aplicada en la evolución de las diferentes herramientas tecnológicas al servicio de la humanidad. El lenguaje de programación está construido sobre un número finito de líneas de código que permiten a los sistemas controlados ejecutar operaciones. El costo computacional ha sido calculado por la calidad del algoritmo de programación ejemplificando el método de eliminación de Gauss para correr tareas sin utilizar interrupciones.

**Palabras claves:** Calidad, Código, Lenguaje de programación Python, Eliminación de Gauss, Método numérico.

## Introducción.

A través de los tiempos la matemática ha jugado un papel fundamental en el desarrollo de la civilización. En la edad media el desarrollo de las conjeturas y demostraciones de la matemática marcaron un hito en el planteamiento de problemas cuya respuesta no pudo ser encontrada en los cálculos manuales, en la actualidad el ordenador sobrepasa nuestra capacidad de cálculo. No hay duda que el ordenado es un aliado precioso de los matemáticos en la exploración del mundo numérico, es también cierto que no podrá tomar nunca el lugar de un matemático. EL ordenador puede ganar fácilmente al matemático en cualquier cálculo finito, sin embargo, le falta aún la imaginación necesaria para comprender al mundo infinito y revelar la estructura y las regularidades que están en la base de la matemática. (Du Sautoy, M. 2007).

Los algoritmos de programación son el esquema plasmado de la aplicación de las matemáticas de modo que el presente artículo procura mostrar cómo contabilizar las operaciones de los algoritmos de programación para algunas operaciones importantes de los

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

microcontroladores PIC en tareas que no requieren interrupciones utilizando el lenguaje de programación Python y su correspondiente comparativo con C++. Si bien el lenguaje conjuga códigos sobre distintas funcionalidades, la investigación pretende establecer una métrica que determine la calidad del método numérico aplicado en las tareas desarrolladas por los sistemas inteligentes, tomando como ejemplo el método de eliminación de Gauss. Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos. Estos procesos o acciones son programados en lenguaje ensamblador por el usuario, y son introducidos en este a través de un programador. (Camacho, H. 2008).

Inicialmente cuando no existían los microprocesadores las personas se ingeniaban en diseñar sus circuitos electrónicos y los resultados estaban expresados en diseños que implicaban muchos componentes electrónicos y cálculos matemáticos. Un circuito lógico básico requería de muchos elementos electrónicos basados en transistores, resistencias, etc. lo cual desembocaba en circuitos con muchos ajustes y fallos; pero en el año 1971 apareció el primer microprocesador el cual originó un cambio decisivo en las técnicas de diseño de la mayoría de los equipos.

Al principio se creía que el manejo de un microprocesador era para aquellas personas con un coeficiente intelectual muy alto; por lo contrario, con la aparición de este circuito integrado todo sería mucho más fácil de entender y los diseños electrónicos serían mucho más pequeños y simplificados. Entre los microprocesadores más conocidos tenemos el popular Z-80 y el 8085. Los diseñadores de equipos electrónicos ahora tenían equipos que podían realizar mayor cantidad de tareas en menos tiempo y su tamaño se redujo considerablemente; sin embargo, después de cierto tiempo aparece una nueva tecnología llamada microcontrolador que simplifica aún más el diseño electrónico. (Ruiz, L. 2008).

Un microcontrolador es un solo circuito integrado que contiene todos los elementos electrónicos que se utilizaban para hacer funcionar un sistema basado con un microprocesador; es decir; contiene en un solo integrado la Unidad de Proceso, la memoria RAM, memoria ROM, puertos de entrada, salidas y otros periféricos, con la consiguiente reducción de espacio.

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

El análisis numérico y su diversidad de métodos en realidad es la dialítica del análisis matemático cualitativo y cuantitativo. El análisis matemático nos afirma que bajo ciertas condiciones algo existe y que es único etc. Sin embargo, el otro complementa calculando aproximadamente el valor de aquello que existe. En resumen, podemos decir que el análisis numérico es una reflexión sobre el análisis matemático es decir sobre el álgebra lineal, ecuaciones diferenciales, etc. Desde el punto de vista numérico teniendo como sinergia una serie de métodos o algoritmos cuyo estudio y uso en diferentes áreas de ingeniería es de importancia (Song, G. 2000).

En la actualidad los métodos numéricos contando con una herramienta como la computadora ofrecen alternativas para el cálculo de problemas complejo que en oportunidades el análisis matemático tendría mucha dificultad.

**Metodología.**

Para la realización de esta investigación se empleó el método científico, puesto que es un proceso racional y lógico sistemático, el mismo que parte de la definición y delimitación del problema, precisando objetivos claros, concretos, recolectando información confiable y pertinente; así como también, organizando, analizando e interpretando la información en base a los resultados de la observación. Este método permite presentar el conocimiento científico logrado. La investigación fue de tipo experimental.

La población de estudio estuvo constituida por 40 estudiantes en la asignatura métodos numéricos en la unidad aproximación de la solución de ecuaciones lineales. La muestra fue de tipo aleatorio simple. Se realizaron observaciones planificadas mediante fichas para recoger y resumir los hechos observados, así como la medición de los tiempos de respuesta del algoritmo de aproximación. Se aplicaron cuestionarios a los estudiantes y se analizaron e interpretaron los resultados.

En el desarrollo de sistemas microprocesadores intervienen aspectos peculiares, que no aparecen en otros campos del diseño electrónico, como la dicotomía desarrollo del hardware/desarrollo del software; es decir, hay que desarrollar el circuito, pero por otra se debe programar el microprocesador.

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

Debido a que buena parte de la complejidad de la parte electrónica del sistema reside en el propio microprocesador (o microcontrolador), el cual nos lo da hecho el fabricante, el desarrollo del software suele ser la tarea que más tiempo (y coste económico) ocupa en el desarrollo de un sistema basado en microprocesador. El coste del software se ha venido incrementando progresivamente año a año; así, según un estudio reciente (Barrietos, A. 2007)., en 1980 el 25% del coste del desarrollo del sistema se relacionaba con la parte software y el 75% con hardware, mientras que en la actualidad aproximadamente el 85% es del software y el 15% del hardware. El motivo fundamental es que, debido al gran desarrollo de los microcontroladores, en un único (y barato) chip se dispone ya de CPU, memoria, temporizador, conversor A/D, UART, etc., con lo que la mayor parte del tiempo la emplearemos en programar todo ello.

Se distinguen dos tipos de programas, en lazo abierto y en lazo cerrado. Un programa o rutina en lazo abierto (o de usuario) es aquel que tiene principio y fin, es decir, que comienza en una instrucción y finaliza en otra. Un ejemplo sería una rutina que multiplique dos números y muestre por pantalla el resultado. La mayor parte de los programas que se ejecutan en un computador son de este tipo, pues comienzan cuando se envía al sistema operativo el comando de ejecución, y finalizan con una orden como quit o exit, devolviendo el control al sistema operativo.

## Resultados

La eliminación de Gauss-Jordan, llamada así debido a Carl Friedrich Gauss y Wilhelm Jordan, es un algoritmo del álgebra lineal para determinar las soluciones de un sistema de ecuaciones lineales, encontrar matrices e inversas. Un sistema de ecuaciones se resuelve por el método de Gauss cuando se obtienen sus soluciones mediante la reducción del sistema dado a otro equivalente en el que cada ecuación tiene una incógnita menos que la anterior. El método de Gauss transforma la matriz de coeficientes en una matriz triangular superior. El método de Gauss-Jordan continúa el proceso de transformación hasta obtener una matriz diagonal. El algoritmo para el método de Gauss Seidel emplea 72 líneas de código en lenguaje de programación C++ para lo cual se recomienda que se utilice guiones automáticos y se revise la ortografía. Debe verificarse que el enunciado esté completo están completos y que hay continuidad en los párrafos de su texto. Revise la numeración de sus figuras y asegúrese de

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

que todas las referencias estén incluidas, debe incluir las conclusiones más relevantes de su trabajo.

**Tabla 1**. Algoritmo en lenguaje C++, método eliminación de Gauss

| Line No. | gauss seidel method.- código | Line No. | gauss seidel method.- código |
|---|---|---|---|
| 1 | #include<stdio.h> | 29 | x[i]=temp; |
| 2 | #include<conio.h> | 30 | printf("\nx[%d] =%f",i,x[i]); |
| 3 | #include<math.h> | 31 | }printf("\n"); |
| 4 | #define e 0.01 | | } |
| 5 | void main() | 32 | while(big>=e); |
| | { | 33 | printf("\n\n Converge en la solucion"); |
| 6 | int i,j,k,n; | 34 | for(i=1;i<=n;i++) |
| 7 | float a[10][10],x[10]; | | { |
| 8 | float sum,temp,error,big; | 35 | printf("\nx[%d]=%f",i,x[i]); |
| 9 | printf("Introduce el numero de ecuaciones: "); | | } |
| 10 | scanf("%d",&n) ; | 36 | getch(); |
| 11 | printf("Introduce los coeficientes de las ecuaciones: \n"); | | } |
| 12 | for(i=1;i<=n;i++) | 37 | numero de ecuaciones: 3 |
| | { | 38 | coeficientes de las ecuaciones: |
| 13 | for(j=1;j<=n+1;j++) | 39 | a(Du Sautoy, M. 2007).(Du Sautoy, M. 2007).= 2 |
| | { | 40 | a(Du Sautoy, M. 2007).(Camacho, H. 2008).= 1 |

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

| | | | |
|---|---|---|---|
| 14 | printf("a[%d][%d]= ",i,j); | 41 | a(Du Sautoy, M. 2007).(Ruiz, L. 2008).= 1 |
| 15 | scanf("%f",&a[i][j]); | 42 | a(Du Sautoy, M. 2007).(Song, G. 2000).= 5 |
| | } | 43 | a(Camacho, H. 2008).(Du Sautoy, M. 2007).= 3 |
| | } | 44 | a(Camacho, H. 2008).(Camacho, H. 2008).= 5 |
| 16 | for(i=1;i<=n;i++) | 45 | a(Camacho, H. 2008).(Ruiz, L. 2008).= 2 |
| | { | 46 | a(Camacho, H. 2008).(Song, G. 2000).= 15 |
| 17 | x[i]=0; | 47 | a(Ruiz, L. 2008).(Du Sautoy, M. 2007).= 2 |
| | } | 48 | a(Ruiz, L. 2008).(Camacho, H. 2008).= 1 |
| 18 | do | 49 | a(Ruiz, L. 2008).(Ruiz, L. 2008).= 4 |
| | { | 50 | a(Ruiz, L. 2008).(Song, G. 2000).= 8 |
| 19 | big=0; | 51 | x(Du Sautoy, M. 2007). =2.500000 |
| 20 | for(i=1;i<=n;i++) | 52 | x(Camacho, H. 2008). =1.500000 |
| | { | 53 | x(Ruiz, L. 2008). =0.375000 |
| 21 | sum=0; | 54 | x(Du Sautoy, M. 2007). =1.562500 |
| 22 | for(j=1;j<=n;j++) | 55 | x(Camacho, H. 2008). =1.912500 |
| | { | 56 | x(Ruiz, L. 2008). =0.740625 |
| 23 | if(j!=i) | 57 | x(Du Sautoy, M. 2007). =1.173437 |
| | { | 58 | x(Camacho, H. 2008). =1.999688 |
| 24 | sum=sum+a[i][j]*x[j]; | 59 | x(Ruiz, L. 2008). =0.913359 |
| | } | 60 | x(Du Sautoy, M. 2007). =1.043477 |

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

| | | | |
|---|---|---|---|
| | } | 61 | x(Camacho, H. 2008). =2.008570 |
| 25 | temp=(a[i][n+1]-sum)/a[i][i]; | 62 | x(Ruiz, L. 2008). =0.976119 |
| 26 | error=fabs(x[i]-temp); | 63 | x(Du Sautoy, M. 2007). =1.007655 |
| 27 | if(error>big) | 64 | x(Camacho, H. 2008). =2.004959 |
| | { | 65 | x(Ruiz, L. 2008). =0.994933 |
| 28 | big=error; | 66 | x(Du Sautoy, M. 2007). =1.000054 |
| | } | 67 | x(Camacho, H. 2008). =2.001995 |
| 29 | x[i]=temp; | 68 | x(Ruiz, L. 2008). =0.999474 |
| 30 | printf("\nx[%d] =%f",i,x[i]); | 69 | converge en la solucion: |
| 31 | }printf("\n"); | 70 | x(Du Sautoy, M. 2007).=1.000054 |
| | } | 71 | x(Camacho, H. 2008).=2.001995 |
| 32 | while(big>=e); | 72 | x(Ruiz, L. 2008).=0.999474 |

**Fuente:** Elaboración propia

**Método De Eliminación Gaussiana Con Sustitución Hacia Atrás**

Algoritmo

Entrada: Número de incógnitas y de ecuaciones n; matriz aumentada A = [aij] donde 1 ≤ i ≤n y 1 ≤ j ≤ n + 1
Salida: Solución x1; x2;…;xn o mensaje de que el sistema lineal no tiene solución única
Paso 1: Para i = 1;…; n-1 seguir los pasos 2-4 (proceso de eliminación)
Paso 2: Sea p el menor entero con 1 ≤ p ≤ n y api ≠0
Si p no puede encontrarse entonces salida ("no existe solución única")
Parar
Paso3: Si p = i entonces ejecutar (Ep) ⇓◊ (Ei)
Paso 4: Para j = i + 1; :::; n seguir los pasos 5 y 6
Paso 5: Tomar mji = (aji)/aii
Paso 6: Efectuar (Ej- mji Ei) ◊ (Ej)
Paso 7: Si ann = 0 entonces Salida("no existe solución única")
Parar

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

Paso 8: Tomar xn = (an,n+1)/ann

Paso 9: Para i = n -1, , 1 tomar xi = $\frac{a_{in+1} - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}$

Paso 10: Salida(x1; x2;…; xn) (Procedimiento completado satisfactoriamente Parar.

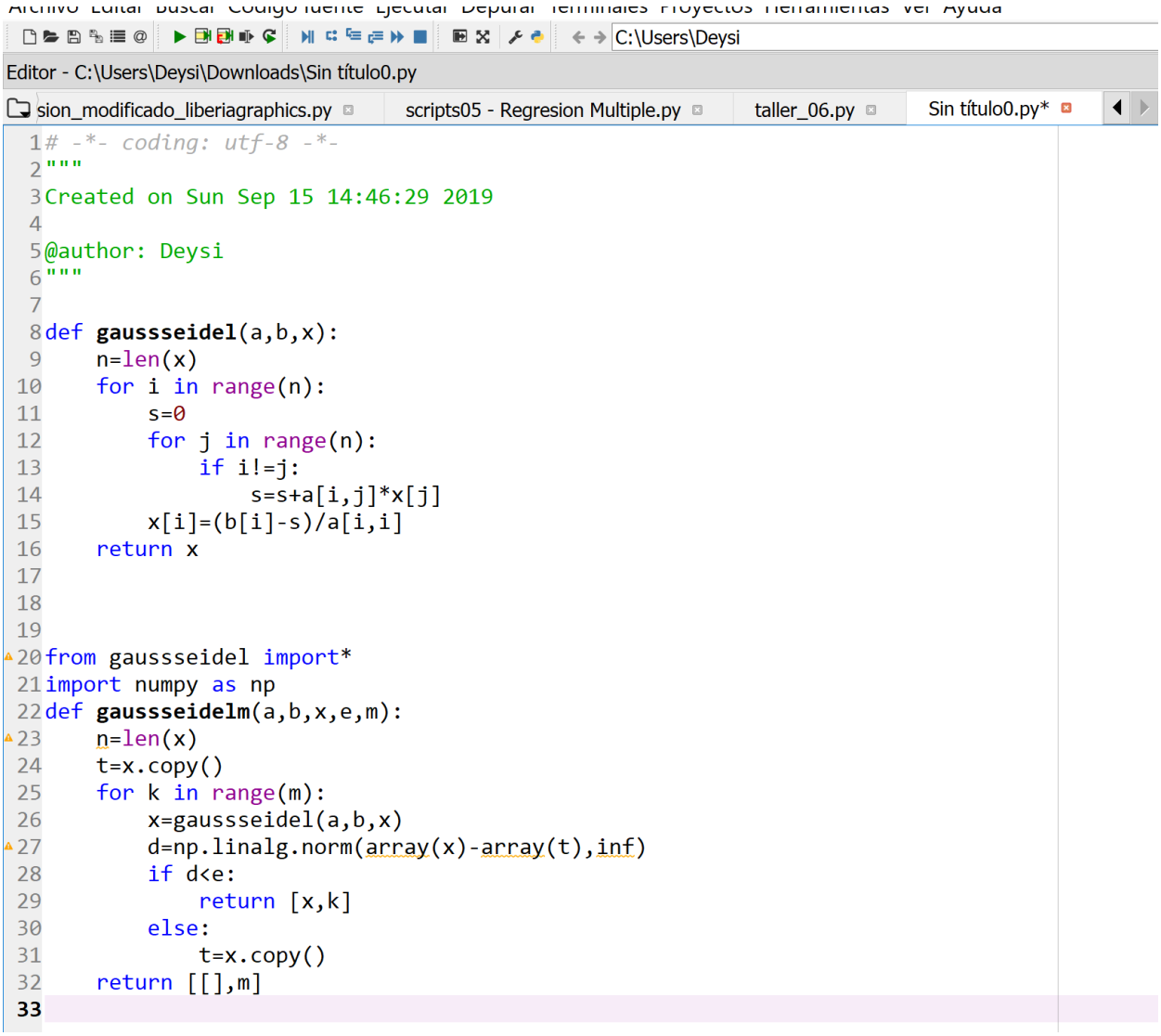


**Figura 1** Algoritmo de eliminación de gauss Seidel en Python
**Fuente:** Elaboración propia

Para la eliminación de gauss el conteo de operaciones se lo realiza de la siguiente manera:

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

En el paso $k$ se elimina $x_k$ de $n-k$ ecuaciones. Para efectuar lo anterior se requieren $n-k$ divisiones para calcular $m_{jk}$ (reglón 3) y $(n-k)(n-k+1)$ multiplicaciones y el mismo número de sustracciones (ambas en el renglón 4). Como se ejecutan $n-1$ pasos entonces $k$ varía desde 1 hasta $n-1$ y así el número total de operaciones en esta eliminación hacia adelante es:

$$f(x) = \sum_{k=1}^{n-1}(n-k) + 2\sum_{k=1}^{n-1}(n-k)\,(n-k+1)$$

$$= \sum_{s=1}^{n-1} s + 2\sum_{s=1}^{n-1} s(s+1) = \frac{1}{2}(n-1)n + \frac{2}{3}(n^2-1)n$$

$$\approx \frac{2}{3}n^3$$

En donde $\frac{2}{3}n^3$ se obtiene al eliminar las potencias mas bajas de $n$. Se observa que $f(n)$ crece de manera casi proporcional a $n^3$. Se dice que $f(n)$ es de orden $n^3$ y se escribe:

$$f(n) = \emptyset(n^3)$$

Donde $\emptyset$ sugiere el término orden. La definición general de $\emptyset$ se escribe:

$$f(n) = \emptyset\big(h(n)\big)$$

Si el cociente $\frac{f(n)}{h(n)}$ permanece acotado, es decir que no tiende al infinito.

Cuando $n \to \infty$ en este caso $h(n) \to n^3$ y en efecto, $\frac{f(n)}{n^3} \to \frac{2}{3}$
Porque los términos omitidos entre $n^3$ tienden a cero cuando $n \to \infty$.

En la sustitución hacia atrás $x_i$ se efectúan $n-i$ multiplicaciones y el mismo número de sustracciones, así como una división. Por tanto, el número de operaciones en la sustitución hacia atrás es:

$$b(n) = \sum_{i=1}^{n}(n-i) + n = \sum_{s=1}^{n} s + n = \frac{1}{2}n(n+1) + n$$

$$\approx \frac{1}{2}n^2 = \emptyset(n^2)$$

Se observa que la expresión anterior crece más lentamente que el número de operaciones en la eliminación hacia adelante del algoritmo de Gauss. De modo que es depreciable para sistemas grandes, debido a que es menor por un factor n, aproximadamente. Por ejemplo si una operación se efectúa a $10^{-6}$s, entonces los tiempos requeridos son:

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

**Tabla 2.** Comparativo de costo computacional entre algoritmos

| *N Líneas de código* | *Eliminación* | *Sustitución hacia atrás* | *Lenguaje de programación* |
|---|---|---|---|
| 72 | 0.7 segundos | 0.5 segundos | C++ |
| 34 | 0.4 segundos | 0. 2 segundos | Python |

**Fuente:** Elaboración propia

**Análisis:** Los diferentes algoritmos desarrollados en el lenguaje de programación Python requieren de menor cantidad de tiempo en la ejecución de las sentencias programadas para aproximar los métodos de Gauss:

**Tabla 1.** ¿Ha desarrollado algoritmos para aproximación en lenguaje Python?

| Preguntas | Estudiantes | Porcentaje |
|---|---|---|
| Si | 1 | 2.5% |
| No | 39 | 97.5% |

**Fuente:** Encuesta aplicada a los estudiantes

**Análisis:** El 97.5 % de los estudiantes reconoce que no han desarrollado ningún tipo de algoritmo que utilice el lenguaje de programación Python y el 2.5% reconoce se ha involucrado con el lenguaje Python.

**Tabla 2.** ¿Utilizaría en sus futuras simulaciones el lenguaje Python?

| Preguntas | Estudiantes | Porcentaje |
|---|---|---|
| Si | 36 | 90% |
| No | 4 | 10% |

**Fuente:** Encuesta aplicada a los estudiantes

**Análisis:** El 90% de los estudiantes reconoce que el lenguaje de programación Python tiene grandes ventajas y lo utilizarán en futuras simulaciones y el 10% reconoce que no lo utilizará

ISSN: 2602-8085
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

Con la finalidad de determinar la aceptación de los lenguajes de programación aplicados a los diferentes métodos numéricos que nos permiten aproximar operaciones algebraicas a través de software el resultado en el grupo de 40 estudiantes es el siguiente

**Tabla 3.** ¿Cómo califica el rendimiento del lenguaje de programación Python?

| Preguntas | Estudiantes | Porcentaje |
|---|---|---|
| bajo | 2 | 5% |
| medio | 15 | 37% |
| alto | 23 | 58% |
| Total | 40 | 100% |

**Fuente:** Encuesta aplicada a los estudiantes

**Análisis:** El 5% de los estudiantes reconoce que tuvo dificultades en la ejecución de algoritmos y califican como bajo el rendimiento de este lenguaje, el 37 % reconoce que el rendimiento es bueno en comparación a otros lenguajes y el 58% reconoce que el rendimiento y calidad en la ejecución de algoritmos fue alto

## Conclusiones

- Los métodos numéricos constituyen una herramienta de análisis científico y tecnológico valiosa en nuestros días. El desarrollo de las computadoras ha permitido su desarrollo para resolver problemas de gran complejidad, desde la simulación de un fenómeno o dispositivo, hasta el estudio de sistemas complejos.
- La implementación real de un método numérico se hace mediante el uso de lenguajes de programación, como C++, que es uno de los lenguajes técnicos más difundidos, no obstante, se puede considerar otro como Python, cuyo desarrollo actual lo posiciona como uno de los lenguajes más sencillos.
- La calidad de un algoritmo se mide por el número de líneas de comando empleadas para la ejecución de los diferentes métodos. La ejecución de los algoritmos requiere tiempo mismo que determina el costo computacional

**ISSN: 2602-8085**
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

- El lenguaje de programación Python posee alternativas de solución estructurada con un tiempo mínimo de respuesta lo cual hace que cualquier algoritmo adquiera calidad en sus características.

**Referencias Bibliográficas**

Du Sautoy, M. (2007). A música dos números primos: a história de um problema não resolvido na matemática. Zahar.

Camacho, H., Casilla, D., & de Franco, M. F. (2008). La indagación: una estrategia innovadora para el aprendizaje de procesos de investigación. Laurus, 14(26), 284-306.

L. I. Ruiz, A. García, J. García, G. Taboada. *"Criterios para la optimización de sistemas eléctricos en refinerías de la industria petrolera: influencia y análisis en el equipo eléctrico,*" IEEE CONCAPAN XXVIII, Guatemala 2008

G. Song, S. P. Schmidt and B. N. Agrawal, "Active vibration suppression of a flexible structure using smart material and modular control patch," *Proc. Inst. Mech. Eng.* vol .214, 2000, pp. 217–29.

Barrientos A.; Peñin L.; y otros, *"Fundamentos de Robótica,"* Segunda Edición. Editorial McGraw-Hill. España. 2007..

G. O. Young, "Synthetic structure of industrial plastics," in *Plastics*, 2nd ed., vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15-64.

J. Jones. (1991, May 10). *Networks*. (2nd ed.) [Online]. Disponible: http://www.atm.com

Minchala, Ismael, Presentación curso *"Visión Artificial"*. National Instruments. Ecuador, 2009.

**ISSN: 2602-8085**
Vol. 3, N°3.3, p. 331 - 344, septiembre, 2019

**PARA CITAR EL ARTÍCULO INDEXADO.**

Guanga Chunata, D., Sarmiento Torres, L., & Inca Chunata, M. de J. (2019). Métodos numéricos aplicados al conteo de operaciones para el cálculo del Costo computacional en Python. *Ciencia Digital*, *3*(3.3), 331-344. https://doi.org/10.33262/cienciadigital.v3i3.3.825