

ALGORITMOS QUE PERMITAN DAR SOLUCIÓN UTILIZANDO VECTORES Y MATRICES

Elemento competencia 1

Guía de evidencia

Bibliografía

Desarrollo temático

Tabla de contenido

De clic en los siguientes enlaces para acceder a un contenido específico.

[Definición, características, partes y tipos de algoritmos](#)

[Tipos de datos, operadores, variables, constantes y expresiones](#)

[Instrucciones de lectura, salida y asignación](#)

Instrucciones de lectura, salida y asignación

Al momento de desarrollar un algoritmo, se deben **identificar los datos** de entrada que generalmente son ingresados por el usuario desde el teclado u otro dispositivo de entrada mediante la sentencia de lectura. Igualmente debemos **identificar los procesos** (fórmulas o cálculos matemáticos) que se requieren para dar solución al problema planteado y finalmente **visualizar o mostrar los resultados** arrojados con la sentencias de salida.

Una vez que se ha terminado de escribir un algoritmo es necesario comprobar que realiza las tareas para las que se ha desarrollado y produce el resultado correcto y esperado, mediante la **validación o prueba de escritorio**, que consiste en tomar datos específicos como entrada y seguir la secuencia indicada en el algoritmo hasta obtener un resultado. El análisis de estos resultados indicará si el algoritmo está correcto o si por el contrario hay necesidad de corregirlo o hacerle ajustes.

En el primer video que observa abajo se explican algunas definiciones de de algoritmos, las partes de un algoritmo (entradas, procesos y salidas); se expone de forma general los tipos de sentencias básicas utilizadas en el desarrollo de algoritmos en pseudocódigo soportados por los lenguajes de programación y se presenta un ejemplo de una lista de canciones. En el segundo video se explica la forma de verificar si el diseño del algoritmo es correcto o no (**prueba de escritorio**):

Tomado de: <http://youtu.be/6oWJ4eBKEU0>

Tomado de: <http://youtu.be/TvHqEhmXE4A>

El estudiante está invitado a profundizar en esta temática, para lo cual se sugiere estudiar las fuentes de estudio siguientes:

	Recursos de aprendizaje
	<p>Unidad I. (Clic aquí): En este enlace se debe descargar el documento Unidad I.pdf. Leer con mucha atención, analizar e interpretar los conceptos teóricos que dan la fundamentación para el desarrollo de algoritmos cualitativos y cuantitativos de forma exitosa, así como los ejemplos desarrollados, cada uno de ellos incluye el análisis del problema y el desarrollo del algoritmo en pseudocódigo. Al finalizar el estudio de este documento junto con las explicaciones e insumos que se han compartido previamente, estará en capacidad de identificar qué es un algoritmo, sus partes y tipos, las fases del proceso de creación de un programa. Así mismo, podrá realizar el análisis a un problema específico, diseñar el algoritmo cualitativo en pseudocódigo, identificar los tipos de datos, variables, operadores matemáticos, operadores relacionales, operadores lógicos, crear expresiones, conocer las diferentes formas de representar un algoritmo y las sentencias de lectura, salida y asignación, analizar e interpretar el análisis de los múltiples problemas planteado, el diseño de los algoritmos y la validación o prueba de escritorio de estos.</p> <p>Prueba de escritorio (Clic aquí): En este enlace se debe descargar el archivo "Prueba de escritorio.ppt" en el que se muestra paso a paso la forma de hacerle seguimiento, validación o prueba de escritorio a un algoritmo que lee dos valores A y B y calcula el producto por sucesión de sumas. La presentación se abre con Microsoft PowerPoint. Seleccionar la opción presentación con diapositivas. Allí observará paso a paso el proceso, disponible en: Ver presentación</p> <p>Ejemplos_U1 (Clic aquí): En este enlace debe descargar los ejemplos desarrollados en el documento Unidad I.PDF, realizados en el interpretador de pseudocódigo PSeInt. Estos ejemplos están comprimidos en WinZip por lo que debe descomprimirlos, abrirlos con el PSeInt, analizarlos y ejecutarlos para que verifique su funcionalidad. También los puede ejecutar paso a paso haciendo clic en el menú <i>Ejecutar - Ejecutar Paso a Paso</i> o pulsar la tecla de función <i>F5</i>, para que siga la secuencia de ejecución. Visitar la página http://pseint.sourceforge.net/ para descargar la última versión del interpretador de pseudocódigo, ejemplos desarrollados, documentación y una serie de información que te serán de gran ayuda.</p>

Una vez estudiados los anteriores materiales, continuaremos el desarrollo temático con la observación de un video para aprender a instalar el interpretador de pseudocódigo PSeInt.

Tomado de: http://youtu.be/1b_ICVmpVps

A continuación observe otro video tutorial para aprender a instalar el interpretador de pseudocódigo PSeInt disponible en:

Tomado de: http://www.dailymotion.com/video/x76khr_videotutorial-pseint-1-instalacion_tech

Para aprender la sintaxis usada para escribir algoritmos en el interpretador de pseudocódigo PSeInt, debe observar:

Tomado de: http://www.dailymotion.com/video/x77ss5_videotutorial-pseint-2-sintaxis_tech

	Recursos de aprendizaje
	<p>Los siguiente tutoriales en línea, <i>Aprenda a programar</i> (Clic aquí), y Tutorial de diseño estructurado de algoritmos (Clic aquí): Tutoriales en línea donde se abordan conceptos básicos de la lógica computacional, algoritmos, diagramas de flujo, pseudocódigo, entre otros. También puedes descargar el curso Aprende a programar en PDF (Clic aquí).</p> <p>Del tutorial <i>Aprende a programar</i> debes estudiar del capítulo 1 al 9, del tutorial <i>diseño estructurado de algoritmos</i> debes estudiar desde la unida 1 a la Unidad 6, y de esta última sólo las estructuras secuenciales, ya que las estructuras condicionales las desarrollaremos en la siguiente unidad.</p>

En este momento, usted está en capacidad de realizar análisis a problemas, desarrollar algoritmos en pseudocódigo con sentencias secuenciales (lecturas, asignación y salidas) así como realizar la prueba de escritorio para verificar si el algoritmo arroja los resultados esperados y de no ser así, realizar los ajustes y correcciones hasta que cumpla con los requerimientos establecidos en el enunciado del problema. El desarrollo de la siguiente actividad le permitirá afianzar las competencias adquiridas hasta el momento. Una vez entregada la evidencia, se le invita a continuar con el estudio de la unidad 2, en la cual aprenderá a desarrollar algoritmos en pseudocódigo utilizando instrucciones condicionales y ciclos.

