

Currículo para enseñar Algoritmos y Programación

2009-05-27

Currículo para enseñar Algoritmos y Programación

Algoritmos y programación se definen como los procesos de formulación de una solución a una situación planteada, apoyándose en conceptos y estructuras propias de la programación. Se busca que el estudiante utilice metodologías y estructuras secuenciales, iterativas y condicionales para analizar problemas, diseñar algoritmos, traducir algoritmos a un lenguaje de programación y depurar procedimientos sencillos con el fin de solucionar problemas. Los estudiantes no elaborarán programas complejos, sólo se concentrarán en la elaboración de procedimientos.

Algoritmos y programación se definen como los procesos de formulación de una solución a una situación planteada, apoyándose en conceptos y estructuras propias de la programación. Se busca que el estudiante utilice metodologías y estructuras secuenciales, iterativas y condicionales para analizar problemas, diseñar algoritmos, traducir algoritmos a un lenguaje de programación y depurar procedimientos sencillos con el fin de solucionar problemas. Los estudiantes no elaborarán programas complejos, sólo se concentrarán en la elaboración de procedimientos.

Documento sin título

Algoritmos y Programación

DEFINICIÓN

Algoritmos y programación se definen como los procesos de formulación de una solución a una situación planteada, apoyándose en conceptos y estructuras propias de la programación.

ALCANCE

Se busca que el estudiante utilice metodologías y estructuras secuenciales, iterativas y condicionales para analizar problemas, diseñar algoritmos, traducir algoritmos a un lenguaje de programación y depurar procedimientos sencillos con el fin de solucionar problemas. Los estudiantes no elaborarán programas complejos, sólo se concentrarán en la elaboración de procedimientos.

OBJETIVO GENERAL

Al terminar la instrucción en Algoritmos y Programación, el estudiante debe estar en capacidad de utilizar metodologías y estructuras secuenciales, iterativas y condicionales para analizar problemas, diseñar algoritmos, traducir algoritmos a un lenguaje de programación y depurar los procedimientos resultantes.

OBJETIVOS ESPECÍFICOS

Al terminar la instrucción en esta herramienta, el estudiante debe estar en capacidad de :

- Comprender una metodología para resolver problemas matemáticos.
- Comprender las etapas del ciclo de programación de computadores.
- Comprender qué es un algoritmo.
- Utilizar el lenguaje pseudocódigo para representar algoritmos.
- Comprender qué son identificadores, variables y constantes.

- Conocer los símbolos que se utilizan para representar algoritmos mediante diagramas de flujo.
- Reconocer el entorno de trabajo que ofrece un lenguaje de programación.
- Utilizar apropiadamente las funciones básicas de un lenguaje de programación.
- Traducir algoritmos a un lenguaje de programación.
- Utilizar el recurso de interactividad con los usuarios de los procedimientos.
- Reconocer los diferentes tipos de fallas que puede presentar un procedimiento.
- Comprender en qué fases del ciclo de programación se pueden producir las fallas de sintaxis y de precaución.
- Comprender los pasos para analizar problemas.
- Definir y utilizar variables y constantes en los algoritmos.
- Comprender qué son operadores y expresiones.
- Elaborar diagramas de flujo para representar soluciones de problemas.
- Reconocer el entorno de trabajo que ofrece un software para elaborar diagramas de flujo (menús, barras, área de trabajo).
- Utilizar apropiadamente las funciones básicas de un software para elaborar diagramas de flujo.
- Realizar operaciones básicas con instrucciones y líneas de flujo.

- Elaborar procedimientos con estructura secuencial.
- Identificar los mensajes de error más comunes que presenta el lenguaje de programación utilizado (fallas de sintaxis) y si los hay corregirlos.
- Comprender en qué fases del ciclo de programación se pueden producir las fallas de concepción y de lógica.
- Realizar prueba de escritorio a los algoritmos elaborados.
- Analizar problemas utilizando una metodología con pasos ordenados.
- Realizar el primer paso de la etapa de análisis: "Formular problemas".
- Realizar el segundo paso de la etapa de análisis: "Precisar el resultado esperado".
- Realizar el tercer paso de la etapa de análisis: "Identificar los datos disponibles".
- Realizar el cuarto paso de la etapa de análisis: "Determinar las restricciones".
- Realizar el quinto paso de la etapa de análisis: "Establecer los procesos necesarios".
- Utilizar operadores y construir expresiones.
- Elaborar procedimientos con estructura iterativa.
- Identificar fallas de lógica en los algoritmos elaborados y si las hay corregirlas.

- Comprender la importancia de verificar los resultados que produce un programa de computador.
- Reflexionar sobre la conveniencia de emplear una metodología con pasos ordenados para analizar problemas.
- Comprender qué tipos de datos acepta el lenguaje de programación utilizado.
- Elaborar procedimientos con estructura condicional.
- Verificar los resultados que produce un procedimiento.
- Comprender la importancia de documentar los procedimientos.
- Comprender la importancia de ajustar (afinar o mejorar) los procedimientos.

CONTENIDOS

- Comprender una metodología para resolver problemas matemáticos.
- Conocer los elementos que tienen en común la mayoría de los problemas matemáticos (estado inicial, meta, recursos y el estado actual de conocimientos de quien pretende resolverlos)
- Conocer las cuatro operaciones mentales que intervienen en la solución de problemas matemáticos (entender el problema, trazar un plan, ejecutarlo y revisar)
- Hacer conciencia sobre la utilización en la clase de matemáticas de estas cuatro operaciones para resolver problemas

EVALUACIÓN (Indicadores de Logro):

- Sin ayuda de referencias, describe brevemente, y en sus propias palabras, las cuatro operaciones mentales que intervienen en la solución de problemas matemáticos. [A, F]

NOTAS:

- Aunque Algoritmos y Programación se puede integrar con varias asignaturas, las matemáticas es un área muy adecuada ya que la forma de resolver problemas matemáticos se asemeja mucho al ciclo de programación.

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Comprender las etapas del ciclo de programación de computadores.
- Conocer las cuatro etapas del ciclo de programación para resolver problemas con ayuda del computador (analizar el problema, diseñar un algoritmo, traducir el algoritmo a un lenguaje de programación y depurar el programa)
- Comprender la similitud que hay entre las operaciones mentales que intervienen en la solución de problemas matemáticos y las etapas del ciclo de programación
- Entender que la solución de problemas matemáticos mediante programación tiene dos ciclos (uno en el que se resuelve el problema con lápiz y papel y otro en el que se automatiza la solución)
- Diferenciar entre Sistema Operativo y Software de Aplicación.
- Diferencias entre Software de Aplicación y Procedimientos.

EVALUACIÓN (Indicadores de Logro):

- Explica brevemente las cuatro etapas del ciclo de programación para resolver problemas con ayuda del computador. [A, F]

PERÍODOS DE CLASE: 1 (55 Minutos c/u).

- Comprender qué es un algoritmo.
- Comprender por qué no se debe empezar a diseñar un algoritmo hasta no haber analizado detalladamente los problemas que se desean resolver
- Identificar en el entorno: procesos, ciclos, rutinas o biorritmos que se puedan considerar como algoritmos (concepto intuitivo de algoritmo)
- Comprender que un algoritmo es un conjunto de pasos sucesivos y organizados en secuencia lógica
- Comprender la importancia de organizar en secuencia lógica los pasos de diversos procesos

EVALUACIÓN (Indicadores de Logro):

- Describe al menos dos procesos, ciclos, rutinas o biorritmos que se den en el entorno y que puedan considerarse como algoritmos. [A, F]
- Dada una lectura que describa una serie de instrucciones, sigue cada uno de los pasos indicados en esta, en el orden establecido. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Utilizar el lenguaje pseudocódigo para representar algoritmos.
- Utilizar frases o proposiciones en español para representar instrucciones

- Organizar en secuencia lógica las instrucciones que solucionan problemas planteados
- Refinar los algoritmos representados en pseudocódigo (escribir una primera versión y luego descomponerla en subproblemas, si fuera necesario)
- Comprender la importancia de detallar al máximo las instrucciones para que estas se puedan traducir a un lenguaje de programación

EVALUACIÓN (Indicadores de Logro):

- Dado un problema de la vida cotidiana (como hacer un jugo de fruta), construye un algoritmo en pseudocódigo para solucionarlo. [A, F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Comprender qué son identificadores, variables y constantes.
- Comprender que los identificadores son nombres que se dan a los elementos (variables, constantes, procedimientos) utilizados en los algoritmos
- Conocer un conjunto de reglas (convenciones) para asignar nombres a variables, constantes y procedimientos
- Conocer qué es una variable
- Entender cómo ayuda el uso de variables en la formulación de un algoritmo y en su utilización con diferentes conjuntos de datos iniciales (generalización)
- Conocer los tipos de variables y sus diferencias (globales y locales)
- Conocer cómo asignar un valor a una variable

- Conocer cómo utilizar el valor almacenado en una variable
- Conocer qué es una constante
- Conocer cómo asignar un valor a una constante
- Conocer cómo utilizar el valor almacenado en una constante

EVALUACIÓN (Indicadores de Logro):

- Dado un problema de la vida cotidiana, lista las variables y constantes presentes en este. [A, F]
- Dada una lista de variables y constantes, les asigna nombres que pueda entender el lenguaje de programación. [A, F]
- Dada una lista de variables y constantes, indica cómo asignarles valores a estos. [A, F]

PERÍODOS DE CLASE: 6 (55 Minutos c/u).

- Conocer los símbolos que se utilizan para representar algoritmos mediante diagramas de flujo.
- Comprender que los diagramas de flujo han sido una de las técnicas más utilizadas para representar gráficamente la secuencia de instrucciones de un algoritmo
- Identificar y recordar el significado de los principales símbolos estandarizados para elaborar diagramas de flujo (inicio, final, líneas de flujo, entrada por teclado, llamada a subrutina, salida impresa, salida en pantalla, conector, decisión,

iteración, etc)

- Conocer las principales reglas para elaborar diagramas de flujo (encabezado, dirección de flujo, iniciación de variables y constantes, etc)

EVALUACIÓN (Indicadores de Logro):

- Dada una serie de símbolos para representar algoritmos, escribe al frente su significado. [A, F]
- Dado un algoritmo sencillo, explica la función que realiza en cada uno de los pasos. [A, F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Reconocer el entorno de trabajo que ofrece un lenguaje de programación.
- Entender la barra de título
- Entender la barra de menús (Archivo, Edición, Ver, Insertar, Formato, Ventana)
- Entender las barras de herramientas
- Entender la barra de desplazamiento
- Entender la barra de estado
- Entender el área de trabajo

EVALUACIÓN (Indicadores de Logro):

- En sus propias palabras, describe brevemente, el entorno de trabajo que ofrece el entorno de programación. [A, F]

PERÍODOS DE CLASE: 1 (55 Minutos c/u).

- Utilizar apropiadamente las funciones básicas de un lenguaje de programación.
- Abrir y cerrar el ambiente de programación
- Abrir y cerrar un procedimiento existente
- Crear instrucciones nuevas dentro de un procedimiento existente
- Guardar un procedimiento en una unidad de almacenamiento local o remota
- Crear un proyecto nuevo
- Escribir, con la sintaxis correcta, instrucciones en el lenguaje de programación utilizado
- Compilar un procedimiento
- Ejecutar un procedimiento
- Utilizar las funciones de ayuda que ofrece el software

EVALUACIÓN (Indicadores de Logro):

- Crea un nuevo procedimiento, lo graba en un lugar establecido por el profesor, lo cierra; si es necesario, lo abre nuevamente para modificarlo. [A, F]

ACTIVIDADES:

- **Tema:** Caricaturas

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Traducir algoritmos a un lenguaje de programación.
- Identificar procedimientos que se utilicen frecuentemente en la vida diaria
- Conocer qué significa un procedimiento en un programa de computador
- Conocer la forma de elaborar un procedimiento con el lenguaje de programación seleccionado
- Comprender la estructura de un procedimiento (línea de título, instrucciones y final)
- Utilizar las reglas establecidas (convenciones) para nombrar procedimientos (identificadores)
- Conocer qué significa "palabra reservada"
- Conocer las principales primitivas (comandos) que ofrece el lenguaje de programación utilizado y tenerlas en cuenta para traducir los algoritmos a dicho lenguaje
- Conocer la sintaxis de las principales primitivas
- Conocer la forma de ejecutar un procedimiento en forma directa
- Conocer la forma de llamar un procedimiento desde otro procedimiento
- Hacer comentarios en procedimientos

- Traducir una a una las instrucciones de los diagramas de flujo al lenguaje de programación utilizado
- Elaborar procedimientos que acepten parámetros

EVALUACIÓN (Indicadores de Logro):

- A partir de un algoritmo construido sobre un problema matemático, lo traduce a un procedimiento en MMP. [A, F]
- Abre un procedimiento y lo ejecuta. [A]

ACTIVIDADES:

- **Tema:** Caricaturas

<http://eduteka.org/actividades/actividades.php?idH=501>

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Utilizar el recurso de interactividad con los usuarios de los procedimientos.
- Entender qué es ser usuario de un programa de computador
- Comprender la importancia de la interactividad con el usuario en la generalización de soluciones a problemas
- Reconocer diferentes métodos de interacción con el usuario (teclado y ratón)
- Utilizar los comandos apropiados para establecer interactividad con el usuario mediante el teclado

EVALUACIÓN (Indicadores de Logro):

- Elabora un procedimiento que solucione un problema planteado por el docente, en el que solicite al usuario digitar alguna información. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Reconocer los diferentes tipos de fallas que puede presentar un procedimiento.
- Comprender que hay fallas que detecta el computador (compilador) y otras no (fallas humanas)
- Conocer qué son las fallas de sintaxis y de precaución (detectables por el computador)
- Conocer qué son las fallas de concepción, de lógica y de procedimiento (fallas humanas)

EVALUACIÓN (Indicadores de Logro):

- En sus propias palabras, describe brevemente, los tipos de fallas que se pueden presentar en un procedimiento. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Comprender en qué fases del ciclo de programación se pueden producir las fallas de sintaxis y de precaución.
- Comprender qué es una falla de sintaxis
- Entender en qué fase del ciclo de programación se pueden producir fallas de sintaxis
- Conocer las causas más comunes por las que se producen las fallas de sintaxis

- Comprender el significado de los mensajes de error que presenta el compilador cuando detecta una falla de sintaxis y solucionarla.
- Conocer las medidas que se deben tomar para evitar las fallas de sintaxis
- Comprender qué es una falla de precaución (recomendaciones técnicas o "warning error")
- Entender en qué fase del ciclo de programación se pueden producir fallas de precaución
- Conocer las causas más comunes por las que se producen las fallas de precaución
- Conocer las medidas que se deben tomar para evitar las fallas de precaución

EVALUACIÓN (Indicadores de Logro):

- Dada una serie de fallas que se pueden presentar en un procedimiento, las relaciona con las fases del ciclo de programación en las cuales estas se pueden producir. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Comprender los pasos para analizar problemas.
- Conocer los pasos para analizar un problema que se quiere sistematizar mediante un procedimiento
- Comprender en qué consiste el paso "formular el problema" (determinar y comprender exactamente en qué consiste el problema)

- Comprender en qué consiste el paso "precisar los resultados esperados" (metas y submetas)
- Comprender en qué consiste el paso "identificar los datos disponibles"
- Comprender en qué consiste el paso "determinar las restricciones" (aquello que está permitido o prohibido hacer y/o utilizar para llegar a una solución)
- Comprender en qué consiste el paso "establecer los procesos necesarios" (operaciones)
- Hacer conciencia de cómo estos pasos ayudan a lograr el objetivo de la primera etapa del ciclo de programación

EVALUACIÓN (Indicadores de Logro):

- Sin ayuda de referencias, lista los pasos propuestos para analizar problemas (formular el problema, precisar los resultados esperados, identificar los datos disponibles, determinar las restricciones y establecer los procesos necesarios) y describe brevemente en qué consiste cada uno. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Definir y utilizar variables y constantes en los algoritmos.
- Definir las variables y constantes necesarias para resolver un problema
- Nombrar las variables y constantes definidas utilizando las reglas establecidas para ello
- Inicializar las variables y constantes con los valores iniciales establecidos en el análisis del problema

EVALUACIÓN (Indicadores de Logro):

- Dado un problema de la vida cotidiana, lista las variables y constantes presentes en este. [A, F]
- Dada una lista de variables y constantes, les asigna nombres que pueda entender el lenguaje de programación. [A, F]
- Dada una lista de variables y constantes, indica cómo asignarles valores a estos. [A, F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Comprender qué son operadores y expresiones.
- Conocer que es un operador
- Entender la clasificación de operadores (aritméticos, alfanuméricos, relacionales y lógicos).
- Saber el orden de evaluación de los operadores
- Conocer qué es una expresión
- Entender los elementos que pueden conformar una expresión (valores, funciones, primitivas (comandos), constantes, variables, cadenas alfanuméricas, operadores)
- Conocer diferentes tipos de expresiones (aritméticas, alfanuméricas, lógicas y de asignación)

- Comprender cómo se pueden unir varios de estos elementos mediante operadores para formar una expresión compuesta

EVALUACIÓN (Indicadores de Logro):

- Dado un algoritmo por el docente, identifica y explica las expresiones y operadores presentes en este [F]

PERÍODOS DE CLASE: 4 (55 Minutos c/u).

- Elaborar diagramas de flujo para representar soluciones de problemas.
- Utilizar símbolos para representar instrucciones
- Recordar y utilizar los principales símbolos estandarizados para elaborar diagramas de flujo (inicio, final, líneas de flujo, entrada por teclado, llamada a subrutina, salida impresa, salida en pantalla, conector, decisión, iteración, etc)
- Aplicar las reglas para elaborar diagramas de flujo (encabezado, dirección de flujo, iniciación de variables y constantes, etc)
- Organizar en secuencia lógica las instrucciones que solucionan problemas planteados
- Elaborar diagramas de flujo para representar soluciones de problemas
- Refinar los algoritmos mediante la escritura de una primera versión y luego descomponerla en subproblemas (procedimientos), si fuera necesario
- Detallar al máximo las instrucciones para que estas se puedan traducir a un lenguaje de programación

EVALUACIÓN (Indicadores de Logro):

- Dado un problema matemático (como sumar los números pares comprendidos entre 2 y 1.000), construye un algoritmo en forma de diagrama de flujo para solucionarlo. [F]

PERÍODOS DE CLASE: 4 (55 Minutos c/u).

-
- Reconocer el entorno de trabajo que ofrece un software para elaborar diagramas de flujo (menús, barras, área de trabajo).
 - Entender la barra de título
 - Entender la barra de menús (Archivo, Edición, Ver, Insertar, Formato, Ventana)
 - Entender las barras de herramientas
 - Entender la barra de desplazamiento
 - Entender la barra de estado
 - Entender el área de trabajo
 - Entender las opciones de zoom (aumentar/disminuir la escala de visualización)

EVALUACIÓN (Indicadores de Logro):

- En sus propias palabras, describe brevemente, el entorno de trabajo que ofrece el software seleccionado para elaborar diagramas de flujo. [A, F]

NOTAS:

- Ver la reseña de algunas herramientas descargables de Internet que facilitan el Aprendizaje Visual. Incluye descripción de software para construir Diagramas de

Flujo - <https://eduteka.icesi.edu.co/HerramientasVisuales.php>

PERÍODOS DE CLASE: 1 (55 Minutos c/u).

- Utilizar apropiadamente las funciones básicas de un software para elaborar diagramas de flujo.
- Abrir y cerrar la aplicación
- Abrir y cerrar un diagrama de flujo existente
- Crear instrucciones nuevas dentro de un diagrama de flujo utilizando el símbolo apropiado
- Crear líneas de flujo entre las instrucciones de un diagrama de flujo
- Adicionar un título general que identifique un diagrama de flujo
- Seleccionar título, instrucciones o líneas de flujo
- Mover de posición el título, las instrucciones o las líneas de flujo
- Eliminar título, instrucciones o líneas de flujo
- Utilizar el comando deshacer
- Guardar un diagrama de flujo en una unidad de almacenamiento local o remota
- Guardar un diagrama de flujo para que pueda abrirse con otras versiones del mismo software.

- Exportar un diagrama de flujo a un formato gráfico para que lo puedan leer otros programas
- Utilizar las funciones de ayuda que ofrece el software

EVALUACIÓN (Indicadores de Logro):

- Utilizando un software para elaborar diagramas de flujo, crea un nuevo diagrama, lo graba en un lugar establecido por el profesor, lo cierra; si es necesario, lo abre nuevamente para modificarlo. [A, F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Realizar operaciones básicas con instrucciones y líneas de flujo.
- Editar el texto de instrucciones de un diagrama de flujo
- Cambiar la apariencia de las instrucciones (color, fuente, tamaño, forma (símbolo), etc)
- Cambiar la apariencia de las líneas de flujo (color, grosor de la línea, aspecto de la flecha, etc)
- Utilizar las opciones de copiar y pegar para duplicar instrucciones
- Utilizar la opción que ofrece el software para organizar automáticamente los diagramas de flujo

EVALUACIÓN (Indicadores de Logro):

- Dado un problema por el profesor, elabora un procedimiento para solucionarlo; la solución debe incluir el análisis del problema y el diagrama de flujo. [F]

PERÍODOS DE CLASE: 1 (55 Minutos c/u).

- Elaborar procedimientos con estructura secuencial.
- Conocer qué es una estructura secuencial
- Conocer qué tipo de instrucciones puede contener una estructura secuencial (declaración de variables y constantes, asignación de valores, entrada de datos, operaciones, reporte de resultados)
- Utilizar el orden correcto de ejecución de las instrucciones del algoritmo
- Elaborar procedimientos que contengan únicamente la estructura secuencial
- Reflexionar sobre la estructura utilizada en la solución de los problemas

EVALUACIÓN (Indicadores de Logro):

- A partir de un algoritmo construido sobre un problema matemático, lo traduce a un procedimiento en MMP. [A, F]
- Abre un procedimiento y lo ejecuta. [A]
- Elabora un procedimiento que solucione un problema planteado por el docente, en el que solicite al usuario digitar alguna información. [F]
- Sin ayuda de referencias, describe con sus propias palabras qué es en programación una estructura secuencial. [F]

PERÍODOS DE CLASE: 6 (55 Minutos c/u).

- Identificar los mensajes de error más comunes que presenta el lenguaje de programación utilizado (fallas de sintaxis) y si los hay corregirlos.
- Comprender el significado de los mensajes de error que presenta el compilador cuando detecta una falla de sintaxis
- Realizar acciones correctivas en el programa cuando el compilador reporte una falla de sintaxis

EVALUACIÓN (Indicadores de Logro):

- Dado por el docente un procedimiento que contenga fallas de sintaxis, identifica y corrige dichas fallas. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Comprender en qué fases del ciclo de programación se pueden producir las fallas de concepción y de lógica.
- Comprender qué es una falla de concepción (mala formulación del problema)
- Entender en qué fase del ciclo de programación se pueden producir fallas de concepción
- Conocer las causas más comunes por las que se producen las fallas de concepción
- Conocer las medidas que se deben tomar para evitar las fallas de concepción
- Comprender qué es una falla de lógica (algoritmos mal diseñados)

- Entender en qué fase del ciclo de programación se pueden producir fallas de lógica
- Conocer las causas más comunes por las que se producen las fallas de lógica
- Conocer las medidas que se deben tomar para evitar las fallas de lógica

EVALUACIÓN (Indicadores de Logro):

- Dado un algoritmo elaborado por otro estudiante, realiza la prueba de escritorio y predice que resultado arrojará el computador con un conjunto determinado de datos de entrada. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Realizar prueba de escritorio a los algoritmos elaborados.
- Comprender qué es una prueba de escritorio para un algoritmo
- Comprender cómo se realiza una prueba de escritorio
- Realizar la prueba de escritorio a los algoritmos diseñados (dando diferentes datos de entrada y siguiendo la secuencia indicada en el diagrama)

EVALUACIÓN (Indicadores de Logro):

- Dado un algoritmo elaborado por otro estudiante, realiza la prueba de escritorio y predice que resultado arrojará el computador con un conjunto determinado de datos de entrada. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Analizar problemas utilizando una metodología con pasos ordenados.
- Listar en forma ordenada los pasos indicados para analizar problemas ("formular el problema", "precisar los resultados esperados", "identificar los datos disponibles", "determinar las restricciones" y "establecer los procesos necesarios").
- Comprender que los pasos de la metodología para analizar problemas son dinámicos y cíclicos (no es necesario seguirlos en forma secuencial)
- Comprender la importancia y conveniencia de emplear una metodología para analizar problemas

EVALUACIÓN (Indicadores de Logro):

- Dado un problema por el docente, realiza el primer paso de análisis (formular problemas) siguiendo las indicaciones suministradas en clase. [F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Realizar el primer paso de la etapa de análisis: "Formular problemas".
- Formular por escrito problemas a partir de situaciones de la vida real, planteadas en forma verbal
- Hacer conciencia sobre la naturaleza ambigua, imprecisa, incompleta e incongruente que en muchas ocasiones tiene el lenguaje natural y cómo afecta esto la formulación de un problema
- Determinar si se puede definir mejor los problemas planteados

- Identificar y buscar en el diccionario las palabras desconocidas que aparecen en los problemas
- Reflexionar sobre si se ha resuelto problemas similares con anterioridad

EVALUACIÓN (Indicadores de Logro):

- Dado un problema por el docente, realiza el primer paso de análisis (formular problemas) siguiendo las indicaciones suministradas en clase. [F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Realizar el segundo paso de la etapa de análisis: "Precisar el resultado esperado".
- Precisar con claridad cuál es resultado final (producto) que debe devolver el programa elaborado
- Establecer el formato que debe tener el resultado final (impreso, en pantalla, diagramación, orden, etc)
- Identificar la información relevante de un problema

EVALUACIÓN (Indicadores de Logro):

- Dado un problema por el docente, realiza el segundo paso de análisis (precisar el resultado esperado) siguiendo las indicaciones suministradas en clase. [F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Realizar el tercer paso de la etapa de análisis: "Identificar los datos disponibles".

- Identificar cuál es la información importante, para llegar a una solución, que se ofrece en la formulación de problemas
- Identificar cuál es la información no relevante, para llegar a una solución, que se ofrece en la formulación de problemas (que se puede omitir)
- Identificar los datos de entrada (conocidos) y la(s) incógnita(s) (datos desconocidos)
- Establecer las categorías en las cuales se pueden agrupar los datos
- Determinar el nivel de conocimiento que se posee en el ámbito de los problemas que se pretende resolver y establecer una estrategia para obtener los conocimientos que no se tienen actualmente, necesarios para llegar a una solución

EVALUACIÓN (Indicadores de Logro):

- Dado un problema por el docente, realiza el tercer paso de análisis (identificar los datos disponibles) siguiendo las indicaciones suministradas en clase. [F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Realizar el cuarto paso de la etapa de análisis: "Determinar las restricciones".
- Determinar lo que está permitido o prohibido hacer y/o utilizar para llegar a una solución (restricciones)
- Identificar las condiciones que se plantean en la formulación de los problemas
- Identificar los datos que pueden considerarse como fijos (constantes)

- Identificar los datos que deben considerarse como variables
- Identificar los datos que deben calcularse

EVALUACIÓN (Indicadores de Logro):

- Dado un problema por el docente, realiza el cuarto paso de análisis (determinar las restricciones) siguiendo las indicaciones suministradas en clase. [F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Realizar el quinto paso de la etapa de análisis: "Establecer los procesos necesarios".
- Determinar los procesos (operaciones) que permiten llegar a los resultados esperados a partir de los datos disponibles
- Determinar las fórmulas que deben emplearse
- Identificar como afectan las condiciones (restricciones) a los procesos
- Identificar el orden en el que deben realizarse las operaciones
- Dividir, si es el caso, un problema en otros más pequeños y fáciles de solucionar (procedimientos)

EVALUACIÓN (Indicadores de Logro):

- Dado un problema por el docente, realiza el quinto paso de análisis (establecer los procesos necesarios) siguiendo las indicaciones suministradas en clase. [F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Utilizar operadores y construir expresiones.
- Utilizar operadores aritméticos para construir expresiones aritméticas que tengan en cuenta el orden de evaluación de los operadores
- Utilizar operadores alfanuméricos para construir expresiones alfanuméricas
- Utilizar operadores lógicos y relacionales para construir expresiones lógicas
- Utilizar combinaciones apropiadas de operadores para construir expresiones de asignación

EVALUACIÓN (Indicadores de Logro):

- Dado un problema por el docente que requiera plantear expresiones, identifica variables y constantes y las une mediante operadores para establecer las expresiones correctas que resuelvan el problema. [F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Elaborar procedimientos con estructura iterativa.
- Conocer qué es una estructura iterativa
- Comprender en qué casos es ventajoso utilizar una estructura iterativa
- Conocer qué tipo de instrucciones puede contener una estructura iterativa (instrucciones de control de ciclo, todas las instrucciones de la estructura secuencial)
- Conocer los comandos con los cuales se implementa la estructura iterativa

- Utilizar el orden correcto de ejecución de las instrucciones del algoritmo
- Elaborar procedimientos con una estructura iterativa que contenga y controle a una estructura secuencial
- Reflexionar sobre los tipos de problemas que requieren utilizar la estructura iterativa en la solución

EVALUACIÓN (Indicadores de Logro):

- Sin ayuda de referencias, describe con sus propias palabras qué es en programación una estructura iterativa (de repetición). [F]
- Dado por el docente un problema que requiera para su solución una estructura iterativa (repetición), elabora un procedimiento con una estructura iterativa que contenga y controle una estructura secuencial; la solución debe incluir el análisis del problema, el diagrama de flujo y la prueba de escritorio. [F]

PERÍODOS DE CLASE: 6 (55 Minutos c/u).

- Identificar fallas de lógica en los algoritmos elaborados y si las hay corregirlas.
- Comprender las fallas de lógica no son detectables por el compilador
- Realizar acciones correctivas en el programa cuando mediante la prueba de escritorio se encuentre una falla de lógica

EVALUACIÓN (Indicadores de Logro):

- Dado por el docente un algoritmo y el respectivo procedimiento los cuales contengan fallas de lógica, identifica y corrige dichas fallas. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Comprender la importancia de verificar los resultados que produce un programa de computador.
- Comprender la importancia de verificar resultados
- Comprender cómo se verifican los resultados de un programa

EVALUACIÓN (Indicadores de Logro):

- Dado por el docente un algoritmo y el respectivo procedimiento los cuales contengan fallas de lógica, identifica y corrige dichas fallas. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Reflexionar sobre la conveniencia de emplear una metodología con pasos ordenados para analizar problemas.
- Hacer conciencia de la importancia y conveniencia de emplear una metodología para analizar problemas
- Reflexionar sobre la importancia de poner por escrito el resultado del análisis de problemas (para cada uno de los pasos)

EVALUACIÓN (Indicadores de Logro):

- Sin ayuda de referencias, explica con sus propias palabras cuál es la importancia de verificar los resultados que produce un programa de computador. [F]
- Elabora un ensayo en el que plasma sus reflexiones sobre la conveniencia o no de emplear una metodología con pasos ordenados para aprender a analizar

problemas. [F]

PERÍODOS DE CLASE: 1 (55 Minutos c/u).

- Comprender qué tipos de datos acepta el lenguaje de programación utilizado.
- Conocer diferentes tipos de datos (números, palabras, listas, arreglos, etc)
- Conocer qué tipos de datos acepta el lenguaje de programación utilizado y qué tratamiento le da a cada tipo

EVALUACIÓN (Indicadores de Logro):

- Sin ayuda de referencias, contrasta las diferencias entre los tipos de datos que acepta el lenguaje de programación utilizado. [F]

PERÍODOS DE CLASE: 3 (55 Minutos c/u).

- Elaborar procedimientos con estructura condicional.
- Conocer qué es una estructura condicional (selección simple y doble)
- Comprender en qué casos es ventajoso utilizar una estructura condicional
- Conocer qué tipo de instrucciones puede contener una estructura condicional (instrucciones de decisión, de control de ciclo y todas las instrucciones de la estructura secuencial)
- Conocer y utilizar correctamente los comandos con los cuales se implementa la estructura condicional de selección simple

- Conocer y utilizar correctamente los comandos con los cuales se implementa la estructura condicional de selección doble
- Comprender que las proposiciones utilizadas en la estructura condicional deben poder evaluarse como verdaderas o falsas (solo dos valores posibles y excluyentes)
- Utilizar correctamente los operadores relacionales y lógicos para construir proposiciones (sencillas y compuestas)
- Expresar apropiadamente las proposiciones para que el lenguaje de programación las pueda entender y evaluar
- Utilizar el orden correcto de ejecución de las instrucciones del algoritmo
- Elaborar procedimientos con una estructura condicional que contenga y controle a una estructura secuencial
- Reflexionar sobre el papel que cumple el lenguaje en la formulación y uso de relaciones de orden y de proposiciones
- Reflexionar sobre los tipos de problemas que requieren utilizar la estructura condicional en la solución
- Reflexionar sobre la importancia que tiene reconocer las estructuras de solución de problemas en la forma de planear secuencias de acciones

EVALUACIÓN (Indicadores de Logro):

- Sin ayuda de referencias, describe con sus propias palabras qué es en programación una estructura condicional. [F]

- Dado por el docente un problema que requiera para su solución una estructura condicional, elabora un procedimiento con una estructura condicional que contenga y controle una estructura secuencial; la solución debe incluir el análisis del problema, el algoritmo en forma de diagrama de flujo y la prueba de escritorio. [F]

PERÍODOS DE CLASE: 6 (55 Minutos c/u).

- Verificar los resultados que produce un procedimiento.
- Estimar el resultado que debe producir un procedimiento y compararlo con el realmente producido para determinar si se encuentra cercano al valor estimado
- Realizar todas las operaciones manualmente con un conjunto de datos iniciales y comparar el resultado con el que arroja el procedimiento luego de introducirle el mismo conjunto de datos

EVALUACIÓN (Indicadores de Logro):

- A partir de la lectura de problemas, identifica correctamente el resultado esperado, los datos disponibles, las restricciones y los procesos necesarios para resolverlos. [F]
- Dada una situación del mundo real, enuncia (formula) un problema que tenga todos los elementos (resultado esperado, datos disponibles, restricciones y procesos necesarios) y explica por qué es un problema. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

- Comprender la importancia de documentar los procedimientos.

- Comprender qué es la documentación de procedimientos y las ventajas que ofrece
- Entender que los procedimientos documentados son más fáciles de leer y comprender por otras personas diferentes a quien los escribió

EVALUACIÓN (Indicadores de Logro):

- Sin ayuda de referencias, explica brevemente y con sus propias palabras cuál es la importancia de documentar los procedimientos. [F]

PERÍODOS DE CLASE: 1 (55 Minutos c/u).

-
- Comprender la importancia de ajustar (afinar o mejorar) los procedimientos.
 - Comprender en qué consiste ajustar un procedimiento
 - Comprender que los requerimientos iniciales pueden estar mal planteados, por tanto el procedimiento resultante se puede mejorar
 - Conocer el impacto que tiene en la imagen de un programador la realización de mejoras en los procedimientos (ir más allá de lo solicitado)

EVALUACIÓN (Indicadores de Logro):

- Sin ayuda de referencias, explica brevemente y con sus propias palabras cuál es la importancia de ajustar (afinar o mejorar) los procedimientos. [F]

PERÍODOS DE CLASE: 2 (55 Minutos c/u).

NOTA GENERAL: El número de clases que aparece asociada a cada uno de los objetivos específicos indica la cantidad de períodos de clase, de 55 minutos,

*estimado por los profesores participantes en la construcción del **Modelo**, necesarios para impartir la instrucción y realizar los ejercicios y prácticas necesarios para que el estudiante pueda dominar lo que se pretende enseñar.*

ESTÁNDARES

A. Operaciones y Conceptos Básicos B. Problemas Sociales, Éticos y Humanos C. Herramientas Tecnológicas para la Productividad D. Herramientas Tecnológicas para la Comunicación E. Herramientas Tecnológicas para la Investigación F. Herramientas Tecnológicas para la Solución de Problemas y la Toma de Decisiones
Estándares "NETS" para estudiantes- Estándares en TIC para estudiantes desarrollados por el proyecto NETS, liderado por el comité de acreditación y criterios profesionales de ISTE (Sociedad Internacional para la Tecnología en Educación). Los criterios de formación básica en TIC para estudiantes se dividen en seis grandes categorías, son muy concretos y pertinentes para la educación en América Latina. <https://eduteka.icesi.edu.co/estandaresestux.php>

ACTIVIDADES PARA ENSEÑAR INFORMÁTICA (APEI):

En la enseñanza de la herramienta: Algoritmos y Programación, se deben dedicar algunas clases de informática a realizar Actividades especialmente diseñadas para desarrollar habilidades básicas en esta herramienta.

Estas Actividades se utilizan cada vez que se inicia el aprendizaje de alguna herramienta, dedicando algunas sesiones de clase a llevarlas a cabo. Tienen como característica principal apartarse de la enseñanza mecánica de comandos y funciones, y dedicarse a promover su aprendizaje con situaciones de la vida real. Deben ser interesantes, retadoras, reales, variadas y divertidas, de forma que ayuden a generar rápidamente las habilidades necesarias para que puedan utilizarse con éxito en los Proyectos de Integración.

Área: Informática

Tema: Caricaturas

Url: <http://eduteka.org/actividades/actividades.php?idH=501>

Herramienta: Algoritmos y Programación

Espacio: La actividad se desarrolla durante la clase de informática (Periodo: 2)

Resumen: Con esta actividad se busca que el estudiante refuerce el concepto de ángulo, reconociendo ángulos internos y externos en una figura. Para ello debe programar desde el área de procedimientos la tortuga de MicroMundos Pro para la elaboración de las caricaturas propuestas con formas geométricas.

LINEAMIENTOS DE INTEGRACIÓN

Con la Integración se busca por un lado, afianzar las habilidades adquiridas en el manejo de las herramientas informáticas y por el otro, facilitar, mejorar o profundizar, con el uso significativo de las TIC, el aprendizaje en otras asignaturas. Es necesario dar a la integración un propósito y unos objetivos claros, para lograr una mejoría real en el aprendizaje y la comprensión de temas que sean fundamentales en las asignaturas seleccionadas para realizarla.

Materias Sugeridas:

- Aunque Algoritmos y Programación se puede integrar con varias asignaturas, las matemáticas es un área muy adecuada ya que la forma de resolver problemas matemáticos se asemeja mucho al ciclo de programación.

Productos Sugeridos:

1. Elaborar procedimientos que resuelvan problemas matemáticos que involucren operaciones aritméticas.

Precauciones:

- Cuando se integra con matemáticas, los estudiantes deben demostrar excelente comprensión de los temas que se van a utilizar para elaborar procedimientos.

LECTURAS RECOMENDADAS:

En pro de los computadores (Parte I) - Artículo del Dr. Gary Stager, pionero en aprendizaje, quien describe las oportunidades que para la educación ofrece el uso creativo de los computadores. Cuenta experiencias exitosas en las que se evidencia la utilización del computador como laboratorio intelectual y mecanismo de expresión.

<https://eduteka.icesi.edu.co/ProComputadores.php>

En pro de los computadores (Parte II) - Gary Stager cuestiona en esta segunda parte el haber excluido la programación de computadores de la formación de los estudiantes. Defiende con argumentos de peso su reinclusión, aduciendo razones como el desarrollo de capacidades intelectuales de orden superior.

<https://eduteka.icesi.edu.co/ProComputadores2.php>

Experiencia de un proyecto de formación en Micromundos- Jaime Alberto Osorio, investigador de la Fundación Tránsito, comparte la experiencia de desarrollar un proyecto de la Secretaría de Educación del Distrito de Bogotá sobre Ambientes Colaborativos en Micromundos.

<https://eduteka.icesi.edu.co/reportaje.php?ReportID=0005>

En las siguientes direcciones de Internet se puede encontrar información sobre software para la elaboración de Diagramas de Flujo:

- SmartDraw <http://www.smartdraw.com>
- Inspiration <http://www.inspiration.com>
- WinEsquema <http://www.softonic.com/ie/27771/WinEsquema>
- Dia Win32 Installer <http://www.softonic.com/ie/33781/dia>
- DFD 1.0 <http://www.softonic.com/ie/16035/DFD>
- Paraben's Flow Charter <http://www.paraben.com/html/flow.html>

- Flow Charter <http://www.tucows.com/preview/199025.html>
- Novagraph Chartist <http://www.tucows.com/preview/289535.html>
- Flow Charting 5 <http://www.patton-patton.com>
- OrgPlus <http://www.tucows.com/preview/281861.html>
- Antechinus Draw Magic <http://www.tucows.com/preview/254904.html>
- ArisFlow <http://www.tucows.com/preview/362408.html>