



Universidad Tecnológica de Durango

Tecnologías de la Información

Fundamentos de Programación

Actividades

“Evidencias de Actividades y Tareas”

Alumnos:

- Barraza Torres Jesús Daniel

1ºA BIS

Docente:

- Ing. Dagoberto Fiscal Gurrola, M.T.I.

Enero 2025

Tabla de Ilustraciones

Ilustración 1: Metodología para la solución de problemas	2
Ilustración 2: Conceptos básicos de algoritmos.....	2
Ilustración 3: Reglas de Algoritmos, Diagramas de Flujo y Pseudocódigo	2
Ilustración 4: Tipos de datos básicos	2
Ilustración 5: Variables y constantes	2
Ilustración 6: Tipos de operadores.....	2
Ilustración 7: Estructura básica de un algoritmo secuencial.....	2

Actividad 1

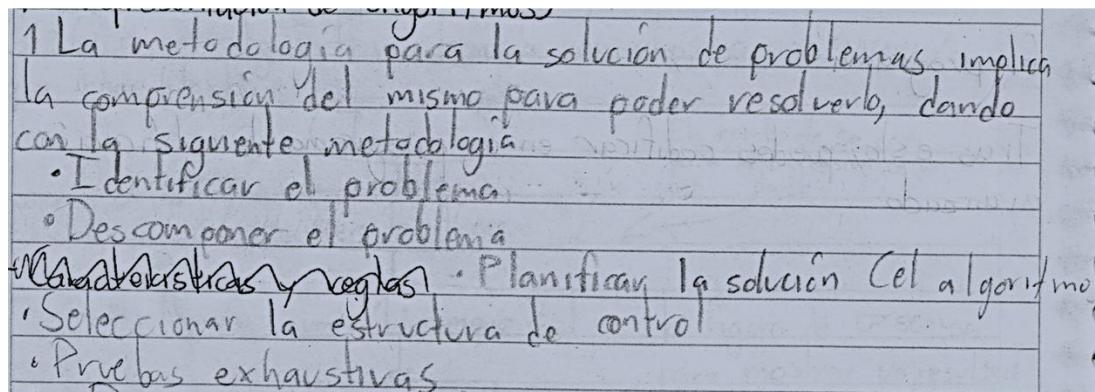


Ilustración 1: Metodología para la solución de problemas

En la imagen anterior se muestra la metodología básica que siguen los programadores a la hora de la resolución de problemas en base a su área y problemáticas específicas del mismo.

Actividad 2

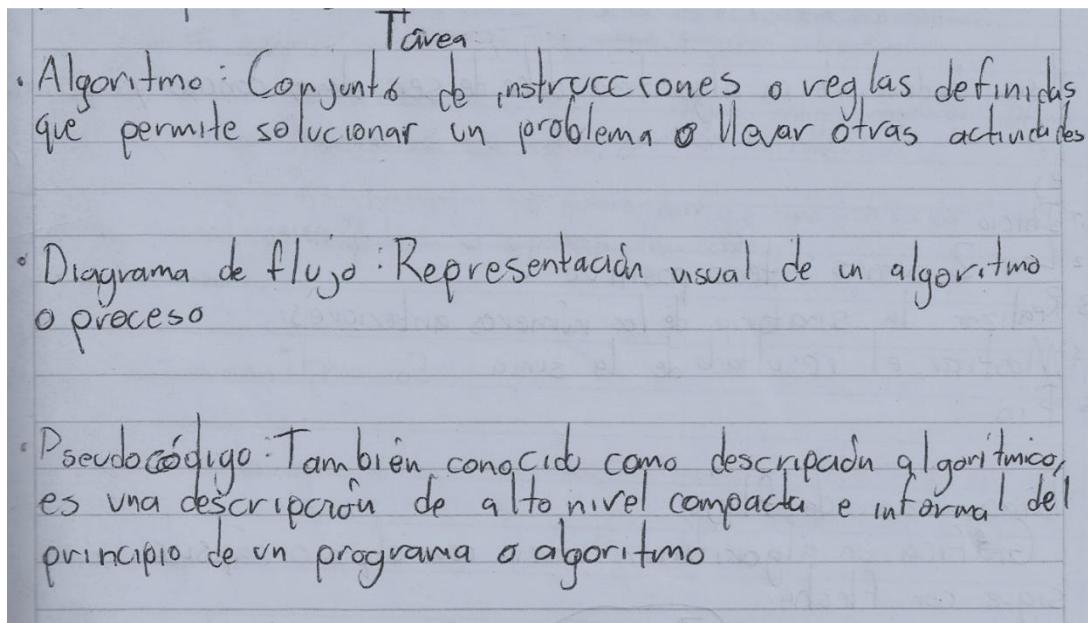


Ilustración 2: Conceptos básicos de algoritmos

En la imagen anterior se muestran los conceptos básicos de representación de un algoritmo de código mediante la simbología y la escritura simple, siendo estos: Algoritmo, Diagrama de Flujo y Pseudocódigo.

Actividad 3

2 Diagrama de Flujo - Representa visualmente el flujo de un proceso - Utiliza símbolos gráficos para mostrar los pasos del proceso Reglas/ formas:
- Inicio o Fin =  - Proceso =  para acciones u otros procesos - Decisión =  bifurcación basada en una condición - Entrada/Salida =  Datos de usuario - Conector =  Para enlazar distintas partes de un diagrama - Flechas = →. Dirección del flujo Pseudocódigo - Representa textualmente y similar a un código real los pasos Mezcla entre lenguaje natural y programación • Reglas/ formas - Instrucciones claras y fáciles de entender Palabras reservadas: "si", "entonces", "mientras", "hacer", "leer", "escribir", etc. Para estructuras acciones Es común el uso de sangrías para demostrar jerarquía Algoritmo Secuencia finita de pasos que resuelvan un problema o realizan una tarea Es la base y puede ser descrito por un diagrama, un pseudocódigo o incluso un lenguaje de programación formal • Reglas/ formas Precisa, inicio y final definidos Finito Ejecutable y bien definido

Ilustración 3: Reglas de Algoritmos, Diagramas de Flujo y Pseudocódigo

En la imagen anterior se muestran los distintos reglamentos de armado y de lectura de un algoritmo, un diagrama de flujo y un pseudocódigo junto con sus características principales y su simbología/significado.

Actividad 4

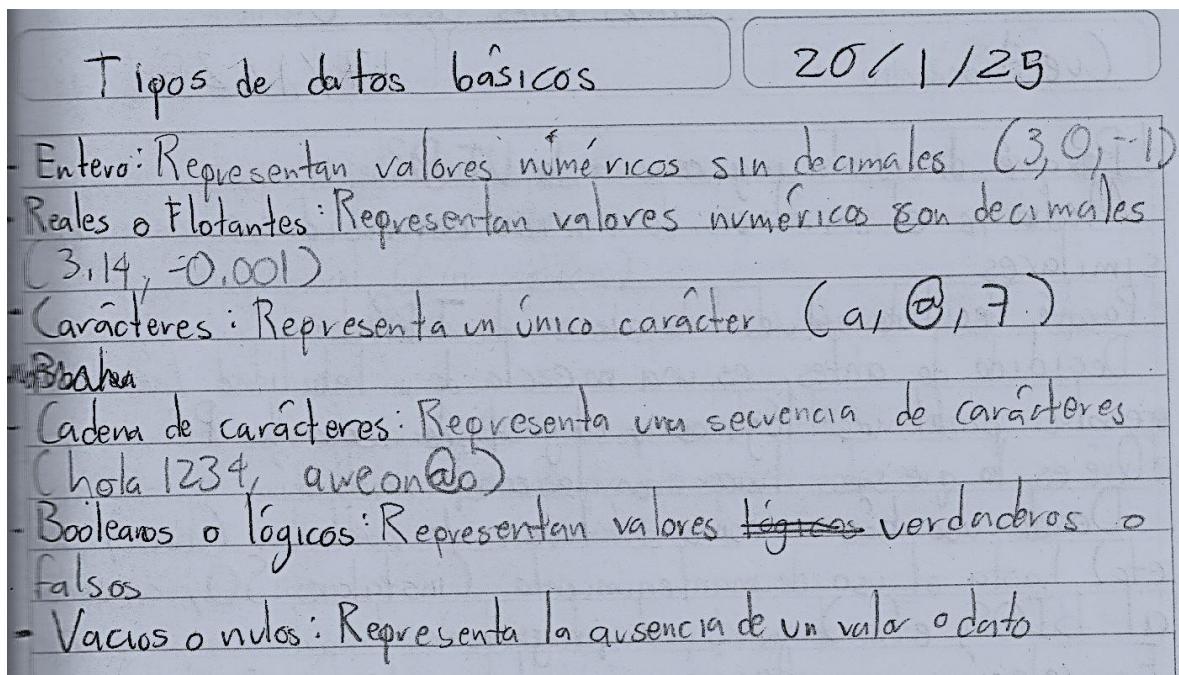


Ilustración 4: Tipos de datos básicos

En la imagen anterior se muestran los distintos tipos de datos básicos que pueden ser representados en un algoritmo ya sean datos existentes como datos que se deben de insertar para que el algoritmo funcione correctamente.

Actividad 5

Variables y constantes (Programación) 20 / 1 / 25

- Variables -

Una variable es una parte del código que puedes cambiar. Las variables se caracterizan por los siguientes procesos:

- Valor mutable: el valor puede ser cambiado en cualquier momento
- Tipo de dato: Pueden almacenar distintos tipos de datos como enteros, cadenas de texto, booleanos, etc.
- Nombre identificador: El valor insertado debe tener un nombre único identificable

- Constantes -

Una constante es el valor que, una vez asignado, no puede cambiar. Las constantes se utilizan para valores que deben permanecer fijos a lo largo del tiempo. Aunque no todos los lenguajes de programación usan una sintaxis para constantes, muchos permiten simularlo de alguna manera.

- Mutabilidad: Las constantes no pueden cambiar al iniciar el programa
- Uso: Se usan para valores fijos ^{no} modificables

Ilustración 5: Variables y constantes

En la imagen anterior se muestran los conceptos de variables y constantes dentro del ámbito de programación. Esto permite identificar datos que posteriormente pueden ser cambiados o datos que serán constantes permanentemente.

Actividad 6

<p>Tipos de operadores</p> <p>Los operadores en programación son símbolos que se utilizan para realizar operaciones sobre valores y variables. Los operadores se pueden clasificar en varios tipos:</p> <ol style="list-style-type: none"> 1. Operadores aritméticos Se utilizan para realizar operaciones matemáticas. + = Suma - = Resta * = Multiplicación / = División % = Módulo (Resto de una división) 2. Operadores de comparación Se utilizan para comparar 2 valores. ==: Igual a !=: Diferente de >: Mayor que <: Menor que >=: Mayor o igual que =: Menor o igual que 3. Operadores lógicos Se utilizan para combinar expresiones booleanas (V/F). && o "and": Y lógico o "or": O lógico ! o "not": Negación lógica 	<p>21/1/25</p> <p>4. Operaciones de asignación Estos operadores asignan un valor a una variable.</p> <ul style="list-style-type: none"> =: Asignación simple +=: Suma y asigna -=: Resta y asigna *=: Multiplicación y asigna /=: División y asigna %=: Módulo y asigna <p>5. Operadores de incremento y decremento (C, C++) Se utilizan para aumentar o disminuir en uno el valor de una variable.</p> <ul style="list-style-type: none"> ++: incremento --: decremento <p>6. Operadores binario Realizan operaciones sobre los bits de los operandos (binario).</p> <ul style="list-style-type: none"> &: Y lógico : O lógico ^: XOR (solo un entrante, no 2 o O) ~: Negación <<: Desplazamiento izq >>: Desplazamiento der <p>7. Pertinenencia Verifica si un dato está en una secuencia (lista o cadena).</p> <ul style="list-style-type: none"> in: Pertenece not in: No pertenece
<p>8. Operador de identidad</p> <p>Comprueba si 2 objetos son iguales.</p> <ul style="list-style-type: none"> is: Es el mismo not is: No es el mismo <p>9. Operadores ternarios (condicionales)</p> <p>Permite realizar una evaluación condicional en una sola línea.</p> <ul style="list-style-type: none"> Condición ? valor_si_true : valor_si_false (C, C++, Java) Condición if else (Python) <p>- Orden de precedencia -</p> <ol style="list-style-type: none"> 1. Parentesis () 2. Operadores unarios (+, -, ++, --, !) 3. Operaciones de multiplicación, división, módulo (*, /, %) 4. Suma y resta (+, -) 5. Desplazamiento (<<, >>) 6. Relacionales (comparación) (==, !=, >, <, >=, <=) 7. Lógicos (and, or) 8. Asignación (=, +=, -=, *=, /=, etc.) 9. Identidad y pertenencia (is, is not, in, not in) 	<p>21/1/25</p>

Ilustración 6: Tipos de operadores

En la imagen anterior se muestran los distintos tipos de operadores así como sus ejemplos y su orden de precedencia. Estos tipos de operadores permiten realizar modificaciones a las variables y a las constantes colocadas en el dispositivo.

Actividad 7

Estructura básica de un algoritmo sec. 21 / 1 / 25

Un algoritmo secuencial es un conjunto de pasos o instrucciones ejecutados de manera lineal sin ninguna repetición. Cada paso se ejecuta en orden, y el flujo sigue una secuencia específica.

La estructura de un algoritmo secuencial incluye:

- 1 Entrada (Input): La fase donde el algoritmo recibe datos para su ejecución. Desde la introducción manual hasta ejecutar un archivo ó obtener datos.
- 2 Proceso (Cálculo): Se realizan las operaciones o transformaciones necesarias a los datos, y a soñ matemáticas, lógicas o de datos
- 3 Salida (Output): Los resultados entregados al usuario o al sistema, ya sea mostrar datos en pantalla, guardar datos o enviar info. a otros procesos
- 4 Finalización: Último paso donde el algoritmo termina su ejecución. Ya sea implícito o explícito.

- Importancia en la resolución de problemas -

Los algoritmos secuenciales son vitales por:

- Simplicidad
- Claridad
- Fácilidad de implementar
- Adecuado para problemas deterministas
- Base para algoritmos más complejos
- Eficiencia

Ilustración 7: Estructura básica de un algoritmo secuencial

En la imagen anterior se nos describe la estructura básica que siguen los algoritmos de secuencia, así como una breve descripción de lo que hacen y su utilidad para la resolución de problemas.

Retroalimentación

Mayoría de las actividades demostradas en esta unidad nos da las herramientas simples para lograr el armado, estructurado y presentado de un algoritmo lógico para los diversos programas por el lado del software.

Esta unidad cumple con la función de demostrar los conceptos básicos de la programación mediante sus temas y actividades. Si bien este tema no se ha puesto en práctica todavía, debido a la brevedad y claridad de los temas, es posible obtener un buen rendimiento al primer intento.

Al iniciar las clases de la unidad, al principio uno puede llegar a perderse, sin embargo, conforme los temas se iban mostrando, las dudas se iban aclarando de manera automática y permite una compresión plena de las bases de los algoritmos para la programación.