Programa 111 mil
VOS PODÉS SER UNO.

Escuela de Educación Técnica N°1

PROGRAMA 111 MIL
Clase N°2

TÉCNICAS DE PROGRAMACIÓN

Raul German Avalos

Técnicas de Programación – Diseño de algoritmos

1. ¿QUÉ ES UN ALGORITMO?.

- UN **ALGORITMO** ES UN **MÉTODO PARA RESOLVER UN PROBLEMA**, QUE CONSISTE EN LA REALIZACIÓN DE UN CONJUNTO DE PASOS **LÓGICAMENTE ORDENADOS** TAL QUE, PARTIENDO DE CIERTOS **DATOS DE ENTRADA**, PERMITE OBTENER CIERTOS RESULTADOS QUE CONFORMAN LA **SOLUCIÓN DEL PROBLEMA**. ASÍ, COMO EN LA VIDA REAL, CUANDO TENEMOS QUE RESOLVER UN PROBLEMA, O LOGRAR UN OBJETIVO, POR EJEMPLO: "TENGO QUE ATARME LOS CORDONES", PARA ALCANZAR LA SOLUCIÓN DE ESE PROBLEMA, REALIZAMOS UN CONJUNTO DE PASOS, DE MANERA ORDENADA Y SECUENCIAL. ES DECIR, PODRÍAMOS DEFINIR UN ALGORITMO PARA ATARNOS LOS CORDONES DE LA SIGUIENTE FORMA:

Programa 111 mil
VOS PODÉS SER UNO.

Escuela de Educación Técnica N°1

Técnicas de Programación – Diseño de algoritmos

1. PONERME LAS ZAPATILLAS.
2. AGARRAR LOS CORDONES CON AMBAS MANOS.
3. HACER EL PRIMER NUDO.
4. HACER UN BUCLE CON CADA UNO DE LOS CORDONES.
5. CRUZAR LOS DOS BUCLES Y AJUSTAR.
6. CORROBORAR QUE AL CAMINAR LOS CORDONES NO SE SUELTAN Y LA ZAPATILLA SE ENCUENTA CORRECTAMENTE ATADA.

Raul German Avalos

Técnicas de Programación – Diseño de algoritmos

2. ¿Y QUÉ ES UN PROGRAMA?

- LUEGO DE HABER DEFINIDO EL **ALGORITMO NECESARIO**, SE DEBE TRADUCIR DICHO ALGORITMO EN UN CONJUNTO DE INSTRUCCIONES, **ENTENDIBLES POR LA COMPUTADORA**, QUE LE INDICAN A LA MISMA LO QUE DEBE HACER; ESTE **CONJUNTO DE INSTRUCCIONES** CONFORMA LO QUE SE DENOMINA, UN **PROGRAMA**.
- PARA ESCRIBIR UN PROGRAMA SE UTILIZAN **LENGUAJES DE PROGRAMACIÓN**, QUE SON LENGUAJES QUE PUEDEN SER **ENTENDIDOS Y PROCESADOS** POR LA COMPUTADORA. UN LENGUAJE DE PROGRAMACIÓN ES TAN SÓLO UN MEDIO PARA EXPRESAR UN ALGORITMO Y UNA COMPUTADORA ES SÓLO UN **PROCESADOR PARA EJECUTARLO**.

Técnicas de Programación – Diseño de algoritmos

3. CARACTERÍSTICAS DE LOS ALGORITMOS

- UN ALGORITMO DEBE SER PRECISO E INDICAR EL ORDEN DE REALIZACIÓN DE CADA PASO.
- UN ALGORITMO DEBE ESTAR ESPECÍFICAMENTE DEFINIDO. ES DECIR, SI SE EJECUTA UN MISMO ALGORITMO DOS VECES, CON LOS MISMOS DATOS DE ENTADA, SE DEBE OBTENER EL MISMO RESULTADO CADA VEZ.
- UN ALGORITMO DEBE SER FINITO. SI SE SIGUE UN ALGORITMO, SE DEBE TERMINAR EN ALGÚN MOMENTO; O SEA, DEBE TENER UN NÚMERO FINITO DE PASOS. DEBE TENER UN INICIO Y UN FINAL.
- UN ALGORITMO DEBE SER CORRECTO: EL RESULTADO DEL ALGORITMO DEBE SER EL RESULTADO ESPERADO.
- UN ALGORITMO ES INDEPENDIENTE TANTO DEL LENGUAJE DE PROGRAMACIÓN EN EL QUE SE EXPRESA COMO DE LA COMPUTADORA QUE LO EJECUTA.

Técnicas de Programación – Diseño de algoritmos

4. ENTRADA – PROCESO – SALIDA

- COMO VIMOS ANTERIORMENTE, EL **PROGRAMADOR** DEBE CONSTANTEMENTE RESOLVER PROBLEMAS DE **MANERA ALGORÍTMICA**, LO QUE SIGNIFICA PLANTEAR EL PROBLEMA DE FORMA TAL QUE QUEDEN INDICADOS LOS PASOS NECESARIOS PARA OBTENER LOS **RESULTADOS PEDIDOS**, A PARTIR DE LOS DATOS CONOCIDOS. LO ANTERIOR IMPLICA QUE UN ALGORITMO BÁSICAMENTE CONSTA DE **TRES** ELEMENTOS: **DATOS DE ENTRADA, PROCESOS Y LA INFORMACIÓN DE SALIDA.**

Técnicas de Programación – Diseño de algoritmos

5. CUANDO DICHO ALGORITMO SE TRANSFORMA EN UN PROGRAMA DE COMPUTADORA:

- LAS **ENTRADAS** SE DARÁN POR MEDIO DE UN DISPOSITIVO DE ENTRADA (COMO LOS VISTOS EN EL BLOQUE ANTERIOR), COMO PUEDEN SER EL TECLADO, DISCO DURO, TELÉFONO, ETC. ESTE PROCESO SE LO CONOCE COMO **ENTRADA DE DATOS**, OPERACIÓN DE LECTURA O ACCIÓN DE LEER.
- LAS **SALIDAS** DE DATOS SE PRESENTAN EN DISPOSITIVOS PERIFÉRICOS DE SALIDA, QUE PUEDEN SER PANTALLA, IMPRESORA, DISCOS, ETC. ESTE PROCESO SE LO CONOCE COMO **SALIDA DE DATOS**, OPERACIÓN DE ESCRITURA O ACCIÓN DE ESCRIBIR.

Técnicas de Programación – Diseño de algoritmos

6. UN POCO MAS PROFUNDO.

- DADO UN **PROBLEMA**, PARA PLANTEAR UN ALGORITMO QUE PERMITA RESOLVERLO, ES CONVENIENTE ENTENDER CORRECTAMENTE LA **SITUACIÓN PROBLEMÁTICA** Y SU **CONTEXTO**, TRATANDO DE DEDUCIR DEL MISMO LOS ELEMENTOS YA INDICADOS (ENTRADAS, PROCESOS Y SALIDA). EN ESTE SENTIDO ENTONCES, PARA CREAR UN **ALGORITMO**:

1. COMENZAR IDENTIFICANDO LOS RESULTADOS ESPERADOS, PORQUE ASÍ QUEDAN CLAROS LOS OBJETIVOS A CUMPLIR.
2. LUEGO, INDIVIDUALIZAR LOS DATOS CON QUE SE CUENTA Y DETERMINAR SI CON ESTOS DATOS ES SUFICIENTE PARA LLEGAR A LOS RESULTADOS ESPERADOS. ES DECIR, DEFINIR LOS DATOS DE ENTRADA CON LOS QUE SE VA A TRABAJAR PARA LOGRAR EL RESULTADO.
3. FINALMENTE SI LOS DATOS SON COMPLETOS Y LOS OBJETIVOS CLAROS, SE INTENTAN PLANTEAR LOS PROCESOS NECESARIOS PARA PASAR DE LOS DATOS DE ENTRADA A LOS DATOS DE SALIDA.

Técnicas de Programación – Diseño de algoritmos

7. NUESTRO PRIMER PROBLEMA

- OBTENCIÓN DEL ÁREA DE UN RECTÁNGULO:

1. RESULTADO ESPERADO: ÁREA DEL RECTÁNGULO.
- SALIDA: ÁREA FÓRMULA DEL ÁREA: BASE X ALTURA.
2. LOS DATOS CON LOS QUE SE DISPONE, ES DECIR LAS ENTRADAS DE DATOS SON:
- DATO DE ENTRADA 1: ALTURA: 5 CM
- DATO DE ENTRADA 2: BASE: 10 CM
3. EL PROCESO PARA OBTENER EL ÁREA DEL RECTÁNGULO ES: ÁREA = BASE * ALTURA - ÁREA = 50

Programa 111 mil
VOS PODÉS SER UNO.

Escuela de Educación Técnica N°1

Técnicas de Programación – Diseño de algoritmos

1. HERRAMIENTAS PARA LA REPRESENTACIÓN GRÁFICA DE LOS ALGORITMOS

COMO SE ESPECIFICÓ ANTERIORMENTE, UN **ALGORITMO** ES INDEPENDIENTE DEL LENGUAJE DE PROGRAMACIÓN QUE SE UTILICE. ES POR ESTO, QUE EXISTEN DISTINTAS **TÉCNICAS DE REPRESENTACIÓN DE UN ALGORITMO** QUE PERMITEN ESTA DIFERENCIACIÓN CON EL **LENGUAJE DE PROGRAMACIÓN ELEGIDO**. DE ESTA FORMA EL ALGORITMO PUEDE SER REPRESENTADO EN CUALQUIER LENGUAJE. EXISTEN DIVERSAS **HERRAMIENTAS** PARA REPRESENTAR GRÁFICAMENTE UN ALGORITMO. EN ESTE MATERIAL PRESENTAREMOS DOS:

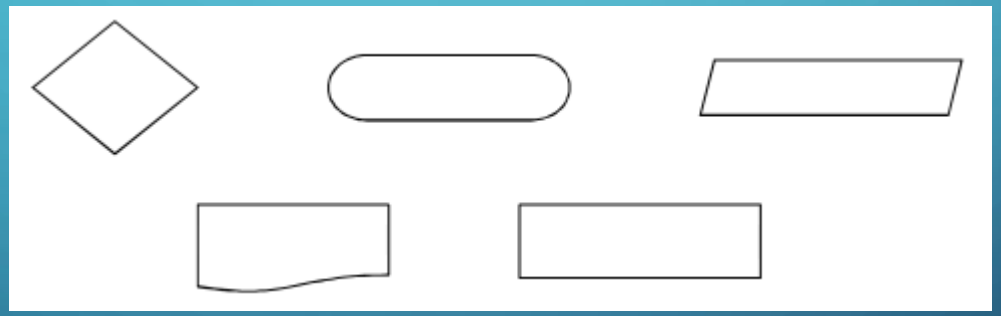
- DIAGRAMA DE FLUJO.
- LENGUAJE DE ESPECIFICACIÓN DE ALGORITMOS: PSEUDOCÓDIGO.

Raul German Avalos

Técnicas de Programación – Diseño de algoritmos

2. DIAGRAMAS DE FLUJO

- UN **DIAGRAMA DE FLUJO** HACE USO DE SÍMBOLOS ESTÁNDAR QUE UNIDOS POR FLECHAS, INDICAN LA **SECUENCIA** EN QUE SE DEBEN EJECUTAR. ESTOS SÍMBOLOS SON, POR EJEMPLO:



MÁS ADELANTE, A MEDIDA QUE PROFUNDIZAMOS EN LOS TEMAS, RETOMAREMOS EL USO DE ESTA HERRAMIENTA Y MOSTRAREMOS ALGUNOS EJEMPLOS DE SU USO.

Técnicas de Programación – Diseño de algoritmos

3. PSEUDOCÓDIGO

- CONOCIDO COMO **LENGUAJE DE ESPECIFICACIÓN DE ALGORITMOS**, EL **PSEUDOCÓDIGO** TIENE UNA ESTRUCTURA MUY SIMILAR AL LENGUAJE NATURAL Y SIRVE PARA PODER EXPRESAR ALGORITMOS Y PROGRAMAS DE FORMA INDEPENDIENTE DEL LENGUAJE DE PROGRAMACIÓN. ADEMÁS, ES MUY UTILIZADO PARA COMUNICAR Y REPRESENTAR IDEAS QUE PUEDAN SER ENTENDIDAS POR PROGRAMADORES QUE CONOZCAN DISTINTOS LENGUAJES.
- UN EJEMPLO BÁSICO DE PSEUDOCÓDIGO, CONSIDERANDO EL EJEMPLO UTILIZADO ANTERIORMENTE, ES EL SIGUIENTE:

```
INICIO FUNCION CALCULAR_AREA
    DEFINIR BASE: 5
    DEFINIR ALTURA: 10
    DEFINIR AREA: BASE * ALTURA
    DEVOLVER AREA
FIN
```

Programa 111 mil
VOS PODÉS SER UNO.

Escuela de Educación Técnica N°1

Técnicas de Programación – Lenguajes de Programación

1. CONCEPTO

- LOS LENGUAJES DE PROGRAMACIÓN SON LENGUAJES QUE PUEDEN SER ENTENDIDOS Y PROCESADOS POR LA COMPUTADORA.

2. TIPOS DE LENGUAJES DE PROGRAMACIÓN

- LENGUAJES MÁQUINA
- LENGUAJE DE BAJO NIVEL (ENSAMBLADOR)
- LENGUAJES DE ALTO NIVEL

Raul German Avalos

Técnicas de Programación – Lenguajes de Programación

3. EL LENGUAJE MÁQUINA

- LOS **LENGUAJES MÁQUINA** SON AQUELLOS QUE ESTÁN ESCRITOS EN LENGUAJES CUYAS INSTRUCCIONES SON **CADENAS BINARIAS** (CADENAS O SERIES DE CARACTERES -DÍGITOS- **O Y 1**) QUE ESPECIFICAN UNA OPERACIÓN, Y LAS POSICIONES (DIRECCIÓN) DE MEMORIA IMPLICADAS EN LA OPERACIÓN SE DENOMINAN **INSTRUCCIONES DE MÁQUINA O CÓDIGO MÁQUINA**. EL CÓDIGO MÁQUINA ES EL CONOCIDO **CÓDIGO BINARIO**

EJEMPLO DE UNA INSTRUCCIÓN: 1110 0010 0010 0001 0000 0000 0010 0000

Técnicas de Programación – Lenguajes de Programación

4. EL LENGUAJE DE BAJO NIVEL

- LOS **LENGUAJES DE BAJO NIVEL** SON MÁS FÁCILES DE UTILIZAR QUE LOS LENGUAJES MÁQUINA, PERO, AL IGUAL, QUE ELLOS, DEPENDEN DE LA MÁQUINA EN PARTICULAR. EL LENGUAJE DE BAJO NIVEL POR EXCELENCIA ES EL ENSAMBLADOR O ASSEMBLER. LAS INSTRUCCIONES EN LENGUAJE ENSAMBLADOR SON **INSTRUCCIONES CONOCIDAS COMO MNEMOTÉCNICAS** (MNEMONICS). POR EJEMPLO, NEMOTÉCNICOS TÍPICOS DE OPERACIONES ARITMÉTICAS SON: EN INGLÉS, ADD, SUB, DIV, ETC.; EN ESPAÑOL, SUM, PARA SUMAR, RES, PARA RESTAR, DIV, PARA DIVIDIR ETC. .

Técnicas de Programación – Lenguajes de Programación

5. LENGUAJES DE ALTO NIVEL

- LOS **LENGUAJES DE ALTO NIVEL** SON LOS MÁS UTILIZADOS POR LOS PROGRAMADORES. ESTÁN DISEÑADOS PARA QUE LAS PERSONAS **ESCRIBAN Y ENTIENDAN** LOS PROGRAMAS DE UN MODO MUCHO **MÁS FÁCIL** QUE LOS LENGUAJES MÁQUINA Y ENSAMBLADORES. OTRA RAZÓN ES QUE UN PROGRAMA ESCRITO EN LENGUAJE DE ALTO NIVEL ES INDEPENDIENTE DE LA MÁQUINA; ESTO ES, LAS INSTRUCCIONES DEL PROGRAMA DE LA COMPUTADORA **NO DEPENDEN DEL DISEÑO DEL HARDWARE** O DE UNA COMPUTADORA EN PARTICULAR.

EJEMPLOS DE LENGUAJES DE ALTO NIVEL: C, C++, JAVA, PYTHON, VISUALBASIC, C#, JAVASCRIPT

Técnicas de Programación – Lenguajes de Programación

1. PASOS PARA LA CONSTRUCCIÓN DE UN PROGRAMA

- DEFINICIÓN DEL PROBLEMA:

- EN ESTE PASO SE DETERMINA LA **INFORMACIÓN INICIAL** PARA LA ELABORACIÓN DEL PROGRAMA. ES DONDE SE DETERMINA QUÉ ES LO QUE DEBE RESOLVERSE CON LA COMPUTADORA, EL CUAL REQUIERE UNA DEFINICIÓN **CLARA Y PRECISA**.
- ES IMPORTANTE QUE SE **CONOZCA LO QUE SE DESEA QUE REALICE LA COMPUTADORA**; MIENTRAS LA DEFINICIÓN DEL PROBLEMA NO SE CONOZCA DEL TODO, NO TIENE MUCHO CASO CONTINUAR CON LA SIGUIENTE ETAPA.

Técnicas de Programación – Lenguajes de Programación

1. PASOS PARA LA CONSTRUCCIÓN DE UN PROGRAMA

- ANÁLISIS DEL PROBLEMA :

UNA VEZ QUE SE HA COMPRENDIDO LO QUE SE DESEA DE LA COMPUTADORA, ES NECESARIO DEFINIR:

- LOS **DATOS DE ENTRADA**
- LOS **DATOS DE SALIDA**
- LOS **MÉTODOS Y FÓRMULAS** QUE SE NECESITAN PARA PROCESAR LOS DATOS

UNA RECOMENDACIÓN MUY PRÁCTICA ES LA DE COLOCARSE EN EL LUGAR DE LA COMPUTADORA Y ANALIZAR QUÉ ES LO QUE SE NECESITA QUE SE ORDENE Y EN QUÉ SECUENCIA PARA PRODUCIR LOS RESULTADOS ESPERADOS.

Técnicas de Programación – Lenguajes de Programación

1. PASOS PARA LA CONSTRUCCIÓN DE UN PROGRAMA

- DISEÑO DEL ALGORITMO:

- SE PUEDE UTILIZAR ALGUNAS DE LAS HERRAMIENTAS DE REPRESENTACIÓN DE ALGORITMOS MENCIONADAS ANTERIORMENTE. ESTE PROCESO CONSISTE EN DEFINIR LA SECUENCIA DE PASOS QUE SE DEBEN LLEVAR A CABO PARA CONSEGUIR LA SALIDA IDENTIFICADA EN EL PASO ANTERIOR.

- CODIFICACIÓN:

- LA CODIFICACIÓN ES LA OPERACIÓN DE ESCRIBIR LA SOLUCIÓN DEL PROBLEMA (DE ACUERDO A LA LÓGICA DEL DIAGRAMA DE FLUJO O PSEUDOCÓDIGO), EN UNA SERIE DE INSTRUCCIONES DETALLADAS, EN UN CÓDIGO RECONOCIBLE POR LA COMPUTADORA. LA SERIE DE INSTRUCCIONES DETALLADAS SE CONOCE COMO CÓDIGO FUENTE, EL CUAL SE ESCRIBE EN UN LENGUAJE DE PROGRAMACIÓN O LENGUAJE DE ALTO NIVEL.

Técnicas de Programación – Lenguajes de Programación

1. PASOS PARA LA CONSTRUCCIÓN DE UN PROGRAMA

- PRUEBA Y DEPURACIÓN:

- SE DENOMINA **PRUEBA DE ESCRITORIO** A LA **COMPROBACIÓN** QUE SE HACE DE UN **ALGORITMO** PARA SABER SI ESTÁ BIEN REALIZADO. ESTA PRUEBA 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**.

Técnicas de Programación – Lenguajes de Programación

2. ELEMENTOS DE UN PROGRAMA

- VARIABLES Y CONSTANTES:

- A LA HORA DE **ELABORAR UN PROGRAMA** ES NECESARIO USAR DATOS; EN EL CASO DEL EJEMPLO DEL CÁLCULO DEL **ÁREA DEL RECTÁNGULO**, PARA PODER OBTENER EL ÁREA DEL MISMO, NECESITAMOS **ALMACENAR EN LA MEMORIA** DE LA COMPUTADORA EL VALOR DE LA BASE Y DE LA ALTURA, PARA LUEGO PODER MULTIPLICAR SUS VALORES.
- RECORDEMOS QUE NO ES LO MISMO GRABAR **LOS DATOS** EN **MEMORIA** QUE GRABARLOS EN EL **DISCO DURO**. CUANDO DECIMOS GRABAR EN MEMORIA NOS ESTAREMOS REFIRIENDO A **GRABAR ESOS DATOS** EN LA RAM. AHORA BIEN, PARA GRABAR ESOS DATOS EN LA RAM PODEMOS HACERLO UTILIZANDO DOS ELEMENTOS, LLAMADOS: **VARIABLES Y CONSTANTES**.

Técnicas de Programación – Lenguajes de Programación

2. ELEMENTOS DE UN PROGRAMA

- VARIABLES Y CONSTANTES:

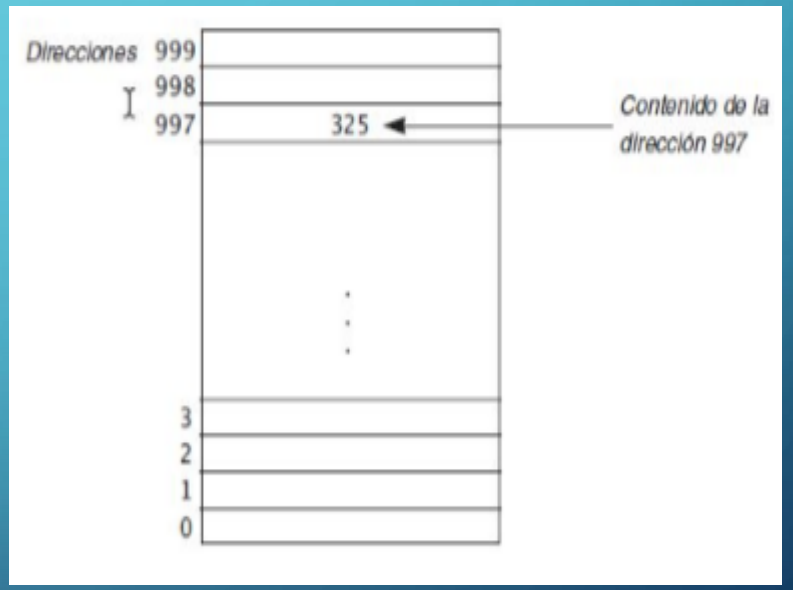
- PODRÍA DECIRSE QUE TANTO LAS **VARIABLES** COMO LAS **CONSTANTES**, SON **DIRECCIONES DE MEMORIA CON UN VALOR**, YA SEA UN NÚMERO, UNA LETRA, O **VALOR NULO** (CUANDO NO TIENE VALOR ALGUNO, SE DENOMINA VALOR NULO). ESTOS ELEMENTOS PERMITEN **ALMACENAR TEMPORALMENTE** DATOS EN LA COMPUTADORA PARA LUEGO PODER REALIZAR **CÁLCULOS Y OPERACIONES** CON LOS MISMOS. AL ALMACENARLOS EN MEMORIA, PODEMOS NOMBRARLOS EN CUALQUIER PARTE DE NUESTRO PROGRAMA Y OBTENER EL VALOR DEL DATO ALMACENADO, SE DICE QUE LA VARIABLE NOS **DEVUELVE EL VALOR ALMACENADO**.

Técnicas de Programación – Lenguajes de Programación

2. ELEMENTOS DE UN PROGRAMA

- VARIABLES Y CONSTANTES:

- A CONTINUACIÓN, SE MUESTRA UN ESQUEMA, QUE REPRESENTA LA MEMORIA RAM, COMO UN CONJUNTO DE FILAS, DONDE, EN ESTE CASO, CADA FILA REPRESENTA UN BYTE Y TIENE UNA DIRECCIÓN ASOCIADA.



Técnicas de Programación – Lenguajes de Programación

3. VARIABLES

- SON ELEMENTOS DE **ALMACENAMIENTO DE DATOS**. REPRESENTAN UNA **DIRECCIÓN DE MEMORIA** EN DONDE SE **ALMACENA UN DATO**, QUE PUEDE VARIAR EN EL DESARROLLO DEL PROGRAMA. UNA **VARIABLE** ES UN GRUPO DE BYTES ASOCIADO A UN NOMBRE O IDENTIFICADOR, Y A TRAVÉS DE DICHO NOMBRE SE PUEDE **USAR O MODIFICAR** EL CONTENIDO DE LOS BYTES ASOCIADOS A ESA VARIABLE.
- EN UNA **VARIABLE** SE PUEDE **ALMACENAR DISTINTOS TIPOS DE DATOS**. DE ACUERDO AL **TIPO** DE DATO, DEFINIDO PARA CADA LENGUAJE DE PROGRAMACIÓN, SERÁ LA CANTIDAD DE BYTES QUE OCUPA DICHA VARIABLE EN LA MEMORIA.

Técnicas de Programación – Lenguajes de Programación

3. VARIABLES

Type	Size	Range
byte	1 byte	−128 to 127
short	2 bytes	−32,768 to 32,767
int	4 bytes	−2,147,483,648 to 2,147,483,647
long	8 bytes	−9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	$-3.40282347 \times 10^{38}$ to 3.4028347×10^{38}
double	8 bytes	$-1.79769313486231570 \times 10^{308}$ to $1.79769313486231570 \times 10^{308}$
char	2 bytes	one character
String	2 or more bytes	one or more characters

Técnicas de Programación – Lenguajes de Programación

3. VARIABLES

- LO MÁS IMPORTANTE DE LA **DEFINICIÓN DE LAS VARIABLES** Y LA **ELECCIÓN DEL TIPO** DE DATOS ASOCIADOS ES EL SIGNIFICADO DE LA VARIABLE, O SU **SEMÁNTICA**: YA QUE EN BASE AL TIPO DE DATOS SELECCIONADO SERÁN LAS OPERACIONES QUE PODAMOS REALIZAR CON ESA VARIABLE, POR EJEMPLO: SI TENEMOS LA VARIABLE EDAD DEBERÍAMOS SELECCIONAR UN TIPO DE DATOS COMO INTEGER (NÚMERO ENTERO) YA QUE LAS OPERACIONES RELACIONADAS SERÁN DE COMPARACIÓN, SUMAS O RESTAS Y NO ES NECESARIO TENER UNA PROFUNDIDAD DE DECIMALES.

- A CONTINUACIÓN, SE LISTAN ALGUNOS DE LOS TIPOS DE DATOS MÁS COMUNES Y SUS POSIBLES USOS:

Técnicas de Programación – Lenguajes de Programación

Tipo de Datos	Significado	Ejemplos de uso
Byte	Número entero de 8 bits. Con signo	Temperatura de una habitación en grados Celsius
Short	Número entero de 16 bits. Con signo	Edad de una Persona
Int	Número entero de 32 bits. Con signo	Distancia entre localidades medida en metros
Long	Número entero de 64 bits. Con signo	Producto entre dos distancias almacenadas en variables tipo int como la anterior
Float	Número Real de 32 bits.	Altura de algún objeto
Double	Número Real de 64 bits.	Proporción entre dos magnitudes
Boolean	Valor lógico: true (verdadero) o false (falso)	Almacenar si el usuario ha aprobado un examen o no

Técnicas de Programación – Lenguajes de Programación

4. CONSTANTES

- ELEMENTOS DE ALMACENAMIENTO DE DATOS. REPRESENTAN UNA DIRECCIÓN DE MEMORIA EN DONDE SE ALMACENA UN DATO PERO QUE NO VARÍA DURANTE LA EJECUCIÓN DEL PROGRAMA. SE PODRÍA PENSAR EN UN EJEMPLO DE NECESITAR UTILIZAR EN EL PROGRAMA EL NÚMERO PI, COMO EL MISMO NO VARÍA, SE PUEDE DEFINIR UNA CONSTANTE PI Y ASIGNARLE EL VALOR 3.14.

```
public class Ejemplo {
    public static void main(String[] args) {
        final double IVA = 0.18;
        int producto = 300;
        double resultado = producto * IVA;
        System.out.println("El IVA del producto es " +resultado);
    }
}
```

Técnicas de Programación – Lenguajes de Programación

5. OPERADORES

- LOS **PROGRAMAS DE COMPUTADORAS** SE APOYAN ESENCIALMENTE EN LA REALIZACIÓN DE NUMEROSAS OPERACIONES **ARITMÉTICAS Y MATEMÁTICAS** DE DIFERENTE COMPLEJIDAD. LOS OPERADORES SON SÍMBOLOS ESPECIALES QUE SIRVEN PARA **EJECUTAR UNA DETERMINADA OPERACIÓN**, DEVOLVIENDO EL RESULTADO DE LA MISMA. PARA COMPRENDER LO QUE ES UN OPERADOR, DEBEMOS PRIMERO INTRODUCIR EL CONCEPTO DE EXPRESIÓN. UNA EXPRESIÓN ES, NORMALMENTE, UNA ECUACIÓN MATEMÁTICA, TAL COMO 3 + 5. EN ESTA EXPRESIÓN, EL SÍMBOLO MÁS (+) ES EL OPERADOR DE SUMA, Y LOS NÚMEROS 3 Y 5 SE LLAMAN OPERANDOS. EN SÍNTESIS, UNA EXPRESIÓN ES UNA **SECUENCIA DE OPERACIONES Y OPERANDOS QUE ESPECIFICA UN CÁLCULO**.

Técnicas de Programación – Lenguajes de Programación

6. TIPOS DE OPERADORES

- OPERADOR DE ASIGNACIÓN

- UN **OPERADOR DE ASIGNACIÓN** ES EL OPERADOR MÁS SIMPLE QUE EXISTE, SE UTILIZA PARA **ASIGNAR UN VALOR** A UNA VARIABLE O A UNA CONSTANTE. EL SIGNO QUE REPRESENTA LA ASIGNACIÓN ES EL **=** Y ESTE OPERADOR INDICA QUE EL **VALOR A LA DERECHA DEL =** SERÁ **ASIGNADO** A LO QUE ESTÁ A LA IZQUIERDA DEL MISMO.

EJEMPLO EN PSEUDOCÓDIGO:

ENTERO EDAD = **20**

DECIMAL PRECIO = **25.45**

Técnicas de Programación – Lenguajes de Programación

6. TIPOS DE OPERADORES

- OPERADORES ARITMÉTICOS

- SON **OPERADORES BINARIOS** (REQUIEREN SIEMPRE **DOS OPERANDOS**) QUE REALIZAN LAS OPERACIONES ARITMÉTICAS HABITUALES: SUMA (+), RESTA (-), MULTIPLICACIÓN (*), DIVISIÓN (/) Y RESTO DE LA DIVISIÓN ENTERA (%), POR EJEMPLO 50%8 = 2 PORQUE NECESITAMOS OBTENER UN NÚMERO ENTERO QUE QUEDA LUEGO DE DETERMINAR LA CANTIDAD DE VECES QUE 8 ENTRA EN 50.

Expresión	Operador	Operandos	Resultado arrojado
5 * 7	*	5 , 7	35
6 + 3	+	6 , 3	9
20 - 4	-	20 , 4	16
50 % 8	%	50 , 8	2
45/5	/	45,5	9

Técnicas de Programación – Lenguajes de Programación

6. TIPOS DE OPERADORES

- LOS OPERADORES UNITARIOS

- LOS **OPERADORES UNITARIOS** REQUIEREN SÓLO UN OPERANDO; QUE LLEVAN A CABO DIVERSAS OPERACIONES, TALES COMO **INCREMENTAR/DECREMENTAR** UN VALOR DE A UNO, NEGAR UNA EXPRESIÓN, O INVERTIR EL VALOR DE UN BOOLEANO.

Operador	Descripción	Ejemplo	Resultado
++	operador de incremento; incrementa un valor de a 1	int suma=20; suma++;	suma=21
--	operador de decremento; Reduce un valor de a 1	int resta=20; resta--;	resta=19
!	operador de complemento lógico; invierte el valor de un valor booleano	boolean a=true; boolean b= !a;	b=false

Técnicas de Programación – Lenguajes de Programación

6. TIPOS DE OPERADORES

- OPERADORES CONDICIONALES

- SON AQUELLOS OPERADORES QUE SIRVEN PARA **COMPARAR VALORES**. SIEMPRE DEVUELVEN VALORES BOOLEANOS: **TRUE O FALSE**. PUEDEN SER RELACIONALES O LÓGICOS.

- OPERADORES RELACIONALES

- LOS **OPERADORES RELACIONALES** SIRVEN PARA REALIZAR COMPARACIONES DE **IGUALDAD, DESIGUALDAD Y RELACIÓN DE MENOR O MAYOR**. LOS OPERADORES RELACIONALES DETERMINAN SI UN OPERANDO ES MAYOR QUE, MENOR QUE, IGUAL A, O NO IGUAL A OTRO OPERANDO. LA MAYORÍA DE ESTOS OPERADORES PROBABLEMENTE LE RESULTARÁ FAMILIAR. TENGA EN CUENTA QUE DEBE UTILIZAR " == ", NO " = ", AL PROBAR SI DOS VALORES PRIMITIVOS SON IGUALES.

Técnicas de Programación – Lenguajes de Programación

6. TIPOS DE OPERADORES

Operador	Significado
==	Igual a
!=	No igual a
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que

Expresión	Operador	Resultado
a > b	>	true: si a es mayor que b false: si a es menor que b
a >= b	>=	true:si a es mayor o igual que b false: si a es menor que b
a < b	<	true: si a es menor que b false: si a es mayor que b
a <= b	<=	true: si a es menor o igual que b false: si a es mayor que b.
a == b	==	true: si a y b son iguales. false: si a y b son diferentes.
a != b	!=	true: si a y b son diferentes false: si a y b son iguales.

Técnicas de Programación – Lenguajes de Programación

6. TIPOS DE OPERADORES

- OPERADORES LÓGICOS

- LOS **OPERADORES LÓGICOS** (AND, OR Y NOT), SIRVEN PARA EVALUAR CONDICIONES COMPLEJAS. SE UTILIZAN PARA CONSTRUIR **EXPRESIONES LÓGICAS**, COMBINANDO VALORES LÓGICOS (TRUE Y/O FALSE) O LOS RESULTADOS DE LOS OPERADORES RELACIONALES.

Expresión	Nombre Operador	Operador	Resultado
a && b	AND	&&	true: si a y b son verdaderos. false: si a es falso, o si b es falso, o si a y b son falsos
a || b	OR	||	true: si a es verdadero, o si b es verdadero, o si a y b son verdaderos. false: si a y b son falsos.

Técnicas de Programación – Lenguajes de Programación

7. RUTINAS

- LAS **RUTINAS** SON UNO DE LOS **RECURSOS MÁS VALIOSOS** CUANDO SE TRABAJA EN PROGRAMACIÓN YA QUE PERMITEN QUE LOS PROGRAMAS SEAN **MÁS SIMPLES**, DEBIDO A QUE EL PROGRAMA PRINCIPAL SE COMPONE DE DIFERENTES RUTINAS DONDE CADA UNA DE ELLAS REALIZA UNA TAREA DETERMINADA.
- UNA RUTINA SE DEFINE COMO UN **BLOQUE**, FORMADO POR UN **CONJUNTO DE INSTRUCCIONES** QUE REALIZAN UNA **TAREA ESPECÍFICA** Y A LA CUAL SE LA PUEDE LLAMAR DESDE **CUALQUIER PARTE** DEL PROGRAMA PRINCIPAL. ADEMÁS, UNA RUTINA PUEDE OPCIONALMENTE TENER UN VALOR DE RETORNO Y PARÁMETROS. EL VALOR DE RETORNO PUEDE ENTENDERSE COMO EL **RESULTADO DE LAS INSTRUCCIONES** LLEVADAS A CABO POR LA RUTINA, POR EJEMPLO SI PARA UNA RUTINA LLAMADA SUMAR(A, B) PODRÍAMOS ESPERAR QUE SU VALOR DE RETORNO SEA LA SUMA DE LOS NÚMEROS A Y B. EN EL CASO ANTERIOR, A Y B SON LOS DATOS DE ENTRADA DE LA RUTINA NECESARIOS PARA REALIZAR LOS CÁLCULOS CORRESPONDIENTES. A ESTOS DATOS DE ENTRADA LOS DENOMINAMOS PARÁMETROS Y A LAS RUTINAS QUE RECIBEN PARÁMETROS LAS DENOMINAMOS FUNCIONES, PROCEDIMIENTOS O MÉTODOS, DEPENDIENDO DEL LENGUAJE DE PROGRAMACIÓN.

Técnicas de Programación – Lenguajes de Programación

7. RUTINAS

- EJEMPLOS DE RUTINAS:

```
SumarPrecioProductos(precioProducto1, precioProducto2) {
```

- RUTINA QUE REALIZA LA SUMA DE LOS PRECIOS DE LOS PRODUCTOS COMPRADOS POR UN CLIENTE Y DEVUELVE EL MONTO TOTAL CONSEGUIDO.

```
AplicarDescuento(montoTotal) {
```

- RUTINA QUE A PARTIR DE UN MONTO TOTAL APLICA UN DESCUENTO DE 10% Y DEVUELVE EL MONTO TOTAL CON EL DESCUENTO APLICADO.

Técnicas de Programación – Lenguajes de Programación

7. RUTINAS

- ENTONCES NUESTRO PROGRAMA PUEDE HACER USO DE DICHAS RUTINAS CUANDO LO NECESITE. POR EJEMPLO CUANDO UN CLIENTE REALICE UNA COMPRA DETERMINADA, PODEMOS LLAMAR A LA RUTINA SUMARPRECIOPRODUCTOS Y COBRARLE EL MONTO DEVUELTO POR LA MISMA. EN EL CASO QUE EL CLIENTE ABONARA CON UN CUPÓN DE DESCUENTO, PODEMOS ENTONCES LLAMAR A LA RUTINA APLICARDESCUENTO Y ASÍ OBTENER EL NUEVO MONTO CON EL 10% APLICADO.