



1859



Universidad
Nacional
de Loja

Educamos para **Transformar**



Universidad
Nacional
de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables Computación

Fundamentos de Algoritmos y programas

1 ciclo A

Teoría de la Programación

Unidad 1

Septiembre 25 - Febrero 26

Educamos para **Transformar**



Universidad
Nacional
de Loja

Fundamentos de Algoritmos y programas



Contenido

Elementos básicos de algoritmos y programas:

- Dato e Información.
- Tipos de datos simples.
- Identificadores, variables y constantes.
- Instrucciones de control secuencial: asignación, lectura y escritura.
- Expresiones (Aritméticas, Lógicas y Relacionales) y literales.
- Pasos para escribir un algoritmo.
- Pruebas de escritorio.





Universidad
Nacional
de Loja

Elementos básicos de algoritmos y programas

{ Datos e información

Un dato es una cifra, letra, palabra... que se suministra a la computadora como entrada y la máquina almacena en un determinado formato (Figuerola Piscoya et al., 2021).

En un programa siempre se procesa datos, que pueden ser de diferente naturaleza. Dependiendo de su **tipo**, los datos se representan y almacenan en la memoria de la computadora (Guerra Salazar et al., 2023).



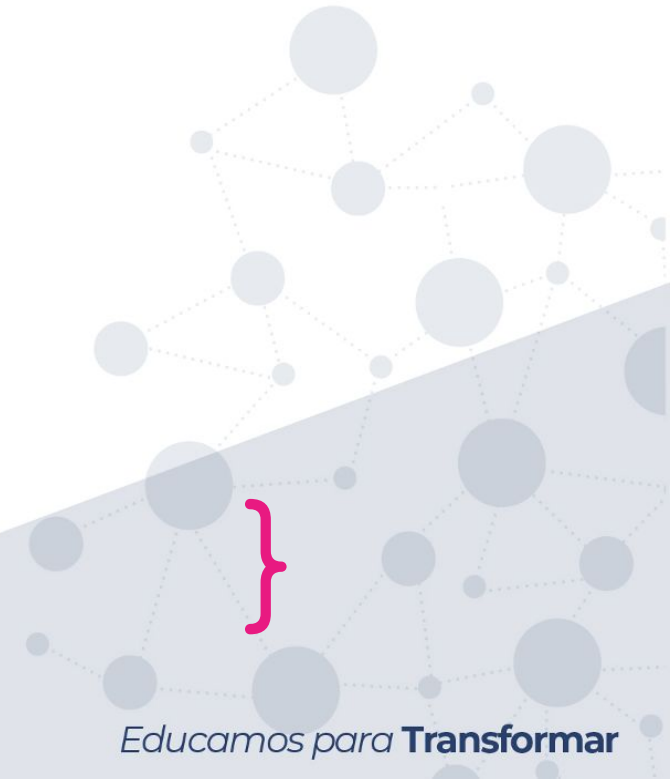


Universidad
Nacional
de Loja

{ Tipos de datos simples

Los **tipos de datos** básicos o primitivos:

- Enteros
- Reales
- Carácter
- Lógicos



{ Tipos de datos simples

- **Enteros (int):** valores que no tienen punto decimal o flotante, pueden ser positivos o negativos y se incluye el cero: 10, 120, 1200, etc.
- **Reales (float o double):** valores que tienen punto decimal o flotante, pueden ser positivos o negativos y se incluye el cero: 4.5, 3.1416, 0.83, etc.
- **Lógicos (boolean):** tipo de dato especial que solo puede almacenar uno de dos valores, falso o verdadero.
- **Carácter (char):** conformados por un solo carácter, van entre comillas dobles, y pueden ser letras, números, signos o símbolos; por ejemplo: “a”, “x”, “3”, “#”.
 - **Cadena** (también conocidas como alfanuméricos): La combinación de los caracteres forma una cadena. Ejemplo: “Calle 107 # 125-48A”, “Hola mundo”.





Universidad
Nacional
de Loja

Elementos básicos de algoritmos y programas

{ Tipos de datos simples

Dato	Tipo de Dato
"Bruce Wayne"	
"27 de mayo de 1970"	
"Gotham City"	
13 431	
1,88	
True	

```
1  Algoritmo sin_titulo
2
3      Definir var1 como Entero;
4      Definir var2 como Real;
5      Definir var3 como Caracter;
6      Definir var4 como Cadena;
7      Definir var5 como Logico;
8
9
10 FinAlgoritmo
```

{ Identificador

Es un **nombre asociado a un objeto de programa**, puede ser una variable, una constante, una función... El nombre de cada identificador debe identificar lo más claramente posible el objeto que identifica.

Es importante saber qué se diferencian las minúsculas de las mayúsculas (Goin, 2022).

{ Identificador

- Siempre debe comenzar por una letra o subguión. No debe empezar con número.
- No debe tener Espacios en Blanco.
- No debe tener caracteres especiales (#, \$, %, -, ...).
- No debe tener letras como la Ñ o ñ, Acento agudo (Á, é, í...), Diéresis (ü, ë, ...), etc.

- 2id
- Estudiante#
- Codigo estudiante
- Codigo-estudiante
- “estudiante”

El primer carácter siempre debe ser una letra.

El carácter “#” no es válido.

El espacio en blanco no es válido.

El carácter - no es válido.

El carácter no es válido.



{ Variables

Es un identificador que **guarda un valor**, que puede ser **modificado** durante la ejecución del programa. Las variables se declaran al inicio del programa, y antes de que se utilicen en las operaciones.

Una variable debe poseer: **un nombre, un tipo de dato y un valor inicial** que este último puede ser opcional.

}

{ Variables

```
1  Algoritmo sin_titulo
2
3  Definir nombre como Cadena;
4  Definir apellidos como Cadena;
5  Definir nombre,apellidos como Cadena;
6
7  nombres = "Lisette"
8  apellidos = "López"
9
10 FinAlgoritmo
```

Como regla general, para definir variables se usará la notación **Lower Camel Case**, que consiste en escribir toda la palabra inicial en minúsculas y las siguientes palabras con la primera letra en mayúscula.

}

{ Constantes

Es un identificador que **guarda un valor**, pero que **no van a cambiar** durante la ejecución del programa. Buena práctica escribir en **MAYÚSCULAS**.

```
1  Algoritmo sin_titulo
2
3      Definir EDAD como Entero;
4      Definir VALORPI como Real;
5      Definir UNIVERSIDAD como Cadena;
6
7      EDAD = 21;
8      VALORPI = 3.141597
9      UNIVERSIDAD = "ESPOCH";
10
11 FinAlgoritmo
```

}

{ Asignación

Es la acción de darle valor a una variable o constante. Se utiliza el signo \leftarrow o $=$ como símbolo de asignación.

`nota1 \leftarrow 5`

`nota1 = 5`

También se utiliza para asignarle a una variable el resultado de una operación matemática.

`suma \leftarrow valor1 + valor2`

`suma = valor1 + valor2`

Elementos básicos de algoritmos y programas

{ Asignación



El valor de la expresión en el lado derecho debe ser de un tipo de datos compatible con la variable en el lado izquierdo, o se producirán resultados inesperados.





Universidad
Nacional
de Loja

Elementos básicos de algoritmos y programas

{ Entrada de datos (lectura)

Instrucciones que permiten al usuario **ingresar datos** (por teclado, otros).

- Leer variable





Universidad
Nacional
de Loja

Elementos básicos de algoritmos y programas

{ Salida de datos (escritura)

Instrucciones
permiten **mostrar** **los**
resultados.

Visualizar un mensaje

- **Escribir** "Mensaje a visualizar"

Visualizar el contenido o valor de una variable

- **Escribir** variable

Visualizar un mensaje y el contenido de una variable

- **Escribir** "Mensaje a visualizar", variable



{ Comentarios en la codificación

Un comentario es un mensaje que sirve para documentar el código. No es tomado en cuenta al momento de ejecutarse el algoritmo.

```
//Definición o declaración de variables  
//Datos de entrada  
//Proceso  
//Datos de salida
```

}

{ Expresiones

Son una combinación entre variables, constantes, **operadores** y paréntesis.

$$a + (b * 5) / c$$

Dentro del desarrollo de los algoritmos y programas es necesario realizar **expresiones**, por lo cual es necesario conocer los tipos de operadores (Dueñas Meza et al., 2021):

- Operadores aritméticos o matemáticos
- Operadores lógicos
- Operadores relacionales





Universidad
Nacional
de Loja

Elementos básicos de algoritmos y programas

{ Expresiones

Operadores Aritméticos	Símbolo
Suma	+
Resta	-
Multiplicación	*
División	/
División Entera	DIV
Módulo (Resto de la división)	MOD %
Potencia	^ **

Operadores aritméticos y su simbología en programación

{ Expresiones

Operadores Relacionales	Símbolo
Mayor que	>
Menor que	<
Mayor o igual que	>=
Menor o Igual que	<=
Igual que	==
Distinto que	!= <> ≠

Operadores relacionales y su simbología en programación

{ Expresiones

Operadores Lógicos	Símbolo	
Conjunción	AND	&&
Disyunción	OR	
Negación	NOT	!

Operadores lógicos y su simbología en programación

}

{ Expresiones

Prioridad	Operador
1	Paréntesis (que, en realidad, no es un operador, pero sirve para dar prioridad a las operaciones determinadas dentro de una expresión que contenga varias operaciones)
2	NOT (Operador lógico)
3	* / DIV MOD (Operadores multiplicativos)
4	+ - (Operadores aditivos)
5	< <= > >= (Operadores de relación)
6	== != (Operadores de relación)
7	AND (Operador lógico)
8	OR (Operador lógico)

Prioridad de los operadores aritméticos, relacionales y lógicos.

{ Expresiones

Un operando situado entre dos operadores de diferente prioridad se liga al operador de mayor prioridad.

- **valor1** = $1+2*3 = 1+6 = 7$

Un operando situado entre dos operadores de igual prioridad se liga al operador de la izquierda.

- **valor2** = $1-2+3 = -1+3 = 2$

Las expresiones entre paréntesis se evalúan primero para ser tratadas como operandos simples.

- **valor3** = $1*(2+3) = 1*5 = 5$

Literal: Es una representación directa de un valor fijo en el código. Cuando se escribe 42, “Hola mundo” o true, se está utilizando literales.

}

{ Expresiones

Ejercicio: Evaluar la siguiente expresión, para $A=2$ y $B=5$:

- $3*2-4*5/2^2$

Solución

$$\begin{array}{r} 3 * 2 - 4 * 5 / 2 ^ 2 \\ 6 - 20 / 4 \\ 6 - 5 \\ 1 \end{array}$$

}

{ Expresiones

Ejercicio: Evaluar las siguientes expresiones:

- Para $A=2$ y $B=5$, $3*A-4*B/A^2$
- $4/2*3/6 + 6 / 2/1/5^2/4*2$

Respuestas

$$\begin{aligned} 3 * 2 - 4 * 5 / 2 ^ 2 \\ 6 - 20 / 4 \\ 6 - 5 \\ 1 \end{aligned}$$

$$\begin{aligned} 2*3/6 + 3/1 / 25 / 4 * 2 \\ 6/6 + 3/25 / 4 * 2 \\ 1 + 0.12/4 * 2 \\ 1 + 0.03*2 \\ 1 + 0.06 \\ 1.06 \end{aligned}$$

}



{ Expresiones

Ejercicio: Obtener el valor de cada una de las siguientes expresiones:

Expresión	Respuesta
7 DIV 2	3
7 / 2	3.5
7 MOD 2	1
12 DIV 3	4
12 MOD 3	0
0 MOD 5	0
15 MOD 5	0
$7 * 10 - 50 \text{ MOD } 3 * 4 + 9$	$70 - 2*4 + 9 = 71$
$(7 * (10 - 50) \text{ MOD } 3) * 4 + 9$	$(7 * -40 \text{ MOD } 3) * 4 + 9$
	$(-280 \text{ MOD } 3) * 4 + 9$
	$2 * 4 + 9$
	17



Expresiones por computadora

Algunas **expresiones matemáticas** utilizan símbolos que no se pueden escribir en un lenguaje de programación como la raíz cuadrada ($\sqrt{}$) o la potenciación en forma de superíndices (x^5).

- Es necesario convertir una expresión aritmética o algebraica a una expresión por computador utilizando únicamente los operadores antes mencionados.
- En las expresiones por computadora solamente se usan paréntesis para agrupar expresiones. No se utilizan corchetes ([]) y llaves ({ }).

Expresión algebraica	Expresión por computadora
$\frac{x^2 + y^2}{z^2}$	$(x^2 + y^2)/z^2$
$\frac{3x + 2y}{2z}$	$(3*x + 2*y)/(2*z)$
$\frac{a + b}{c + d}$	$(a+b)/(c+d)$
$4x^2 - 2x + 7$	$4*x^2 - 2*x + 7$
$\frac{x + y}{x} - \frac{3x}{5}$	$(x+y)/x - 3*x/5$
$\left(\frac{a}{bc}\right)^x$	$(a/(b*c))^x$
xyz	$x * y * z$
$\frac{(y_2 - y_1)}{x_2 - x_1}$	$(y_2-y_1)/(x_2-x_1)$
$2\pi r$	$2 * PI * r$
$\frac{4}{3} \pi r^3$	$4/3 * PI * r^3$
$(x_2 - x_1)^2 + (y_2 - y_1)^2$	$(x_2-x_1)^2 + (y_2-y_1)^2$

Pasos para escribir un algoritmo en pseudocódigo o diagrama de flujo

Análisis del problema

Implica entender y comprender al 100 % el problema que se pretende resolver. De ella depende el éxito en la siguiente fase.

- Los datos de entrada
- Cual es la información que se desea producir, los resultados
- Los métodos y fórmulas que se necesitan para arribar a los resultados, metodología de resolución

Desarrollo del algoritmo

Codificar o transcribir en sentencias o instrucciones el análisis que se le ha hecho al problema planteado.

- Variables, constantes
- Entrada y salida
- Asignación
- Operadores matemáticos, lógicos y relacionales
- Contadores, acumuladores
- Condicionales y ciclos
- ...

Validación o prueba de escritorio

Consiste en realizar un seguimiento manual paso a paso (sentencia por sentencia) del algoritmo para determinar si los resultados arrojados son los esperados. De no ser así, se procede a aplicar los respectivos ajustes.

- Utilizando papel, lápiz y nuestra mente

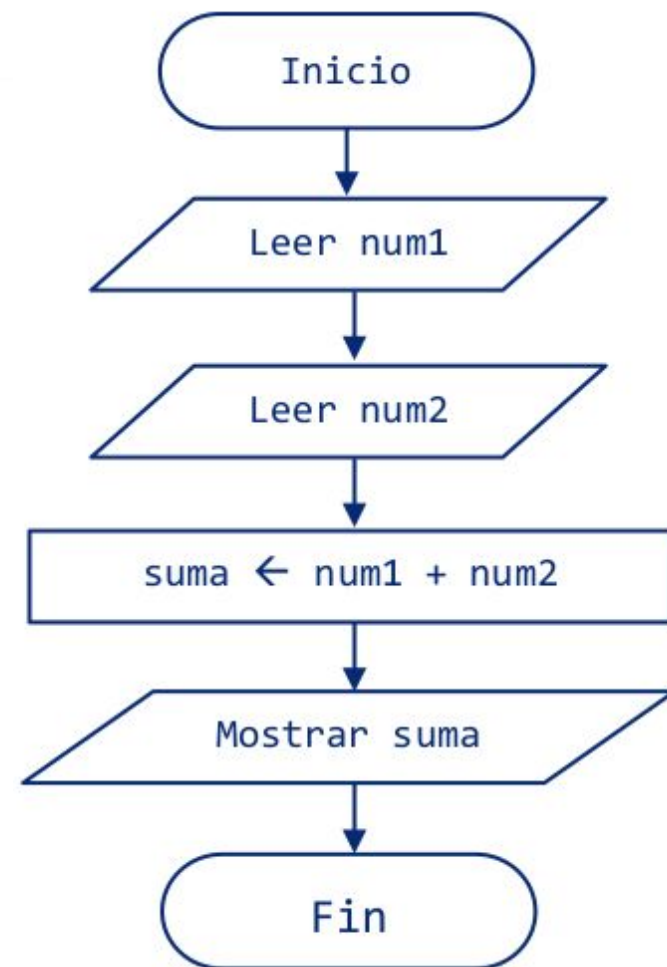
Elementos básicos de algoritmos y programas

ESTRUCTURA SECUENCIAL/LINEAL

{ Ejemplo:

Determinar la suma
de dos números.

```
1  Algoritmo suma
2  // Variables
3  Definir num1 como Real;
4  Definir num2 como Real;
5
6  // Entrada
7  Leer num1
8  Leer num2
9
10 // Proceso
11 sumar = num1 + num2
12
13 //Salida
14 Escribir sumar
15
16 FinAlgoritmo
```





Pruebas de escritorio

Simular datos de entrada, para comprobar que los resultados sean correctos.

De no ser así, será necesario **revisar** el **análisis del problema y el código del algoritmo** para aplicar las **respectivas correcciones** y repetir la prueba de escritorio hasta obtener los datos de salida esperados o correctos.

Prueba de escritorio “suma de dos números”.

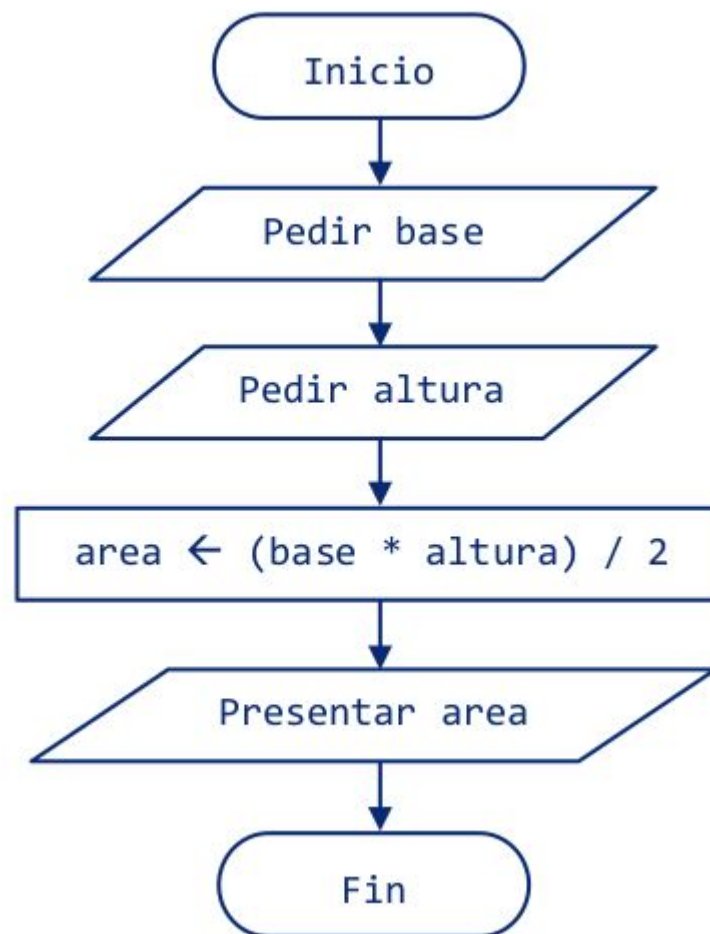
Instrucción	a	b	suma	Salida
inicio				
leer a	5			
leer b		2		
suma = a + b			7	
imprimir				7
fin				



Elementos básicos de algoritmos y programas

{ Ejercicio:

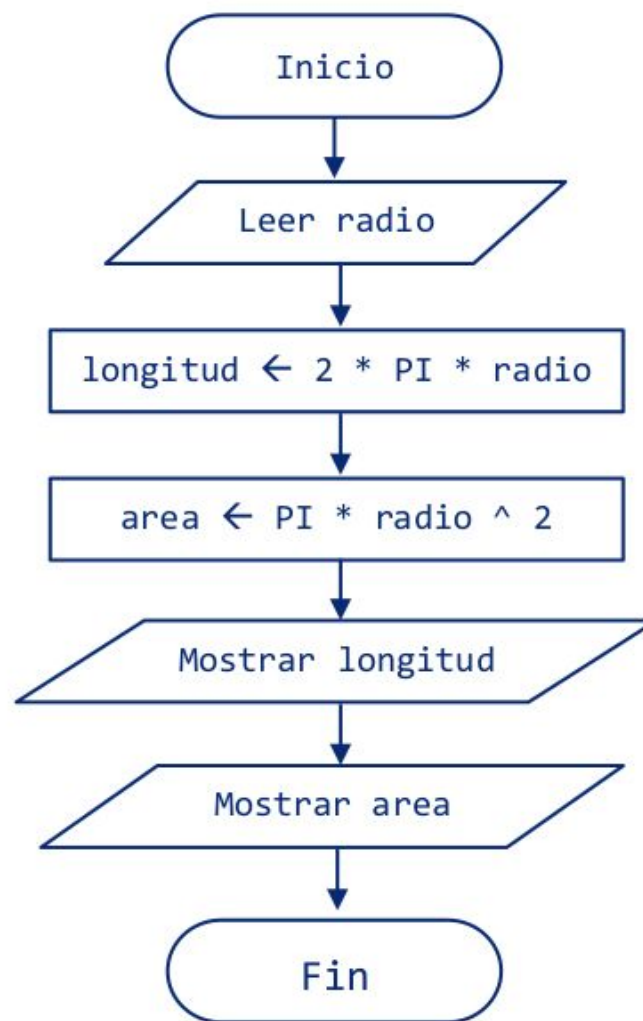
Se requiere conocer la **superficie** de un **triángulo** en función de la base y la altura.



Elementos básicos de algoritmos y programas

{ Ejercicio:

Calcular y visualizar la **longitud** de la **circunferencia** y el **área** de un círculo de radio dado.





Universidad
Nacional
de Loja

Elementos básicos de algoritmos y programas

{



ACD

Pseudocódigo

De los ejercicios de Identificación de Entrada-Proceso-Salida transformar a Pseudocódigo.

}



Universidad
Nacional
de Loja

Elementos básicos de algoritmos y programas



Lista de ejercicios

- Un almacén requiere determinar cuánto cobrar por trabajos de pintura. Considere que se cobra por m². Realice el algoritmo que le permita ir generando presupuestos para cada cliente.
- Un alumno necesita calcular el promedio de sus 3 notas, cada una tiene una ponderación de: primera nota del 30%, segunda nota del 30% y tercera notas del 40%.
- Una modista, para realizar sus prendas de vestir, encarga las telas al extranjero. Para cada pedido, tiene que proporcionar las medidas de la tela en pulgadas, pero ella generalmente las tiene en metros. Realice un algoritmo que determine cuántas pulgadas debe pedir con base en los metros que requiere (1 pulgada = 0.0254 m O 1 metro = 39.37 pulgadas).
- Un productor de leche lleva el registro de lo que produce en litros, pero cuando entrega le pagan en galones. Realice un algoritmo que ayude al productor a saber cuánto recibirá por la entrega de su producción de un día (1 galón = 3.785 litros).
- Un estacionamiento requiere determinar el cobro que debe aplicar a las personas que lo utilizan. Considere que el cobro es con base en las horas que utiliza y que las fracciones de hora se toman como completas. Realice el algoritmo que permita determinar el cobro.
- Realice el algoritmo para determinar cuánto dinero ahorra una persona en un año si considera que cada semana ahorra 15% de su sueldo (considere cuatro semanas por mes y que no cambia el sueldo).





Universidad
Nacional
de Loja

REFERENCIAS BIBLIOGRÁFICAS

- Arteaga Martínez M. M. (2023). *Lógica de programación con Pseint. Enfoque práctico (Primera edición)*. Fondo Editorial Remington. Disponible en: <https://research.ebsco.com/linkprocessor/plink?id=0c1115b8-e552-38e4-bc75-bf84bbdd293f>
- Figueroa Piscocoy, E.N., Maldonado Ramirez, I., y Santa Cruz Acosta, R.C. (2021). *Fundamentos de programación*. Biblioteca Nacional del Perú. https://virtual.autonoma.edu.pe/uploads/10-03-2023_141635939_ANICAMASILVAJOSECARLOS.pdf
- Goin, M. (2022). *Caminando junto al Lenguaje C*. Editorail UNRN Disponible en: https://editorial.unrn.edu.ar/index.php/catalogo/346/view_bl/62/lecturas-de-catedra/26/caminando-junto-al-lenguaje-c?tab=getmybooksTab&is_show_data=1
- Guerra Salazar, J. E, Ramos Valencia, M. V, Vallejo Vallejo, G. E. (2023). *Programando en C desde la práctica problemas resueltos*. Puerto Madero Editorial. Disponible en: <https://dialnet.unirioja.es/servlet/libro?codigo=933288>
- Toro Bonilla, J. M. (2022). *FUNDAMENTOS DE PROGRAMACIÓN: JAVA*. Universidad de Sevilla, Editorial Universidad de Sevilla. Disponible en: <https://dialnet.unirioja.es/servlet/libro?codigo=871118>
- Dueñas Meza, J. M., Valdés Benjumea, J. J., Rodríguez Niño, M. C. (2021). *Didáctica para el aprendizaje de algoritmos*. Universidad Nacional Abierta y a Distancia – UNAD. <https://www.digitaliapublishing.com/a/133651/didactica-para-el-aprendizaje-de-algoritmos>

An aerial photograph of the Universidad Nacional de Loja campus, showing various buildings, a large sports field, and surrounding greenery. A network graphic consisting of blue circles connected by dotted lines is overlaid on the right side of the image.

Educamos para **Transformar**

Redes UNL Oficiales



UNLoficial



UNLoficial



@UNLoficial



Universidad Nacional
de Loja-UNL