



## Actividad | 3 | Código en Lenguaje C

### Introducción al Desarrollo de Software

---

Ingeniería en Desarrollo de Software



TUTOR:

ALUMNO: Alejandro Norzagaray Martínez

FECHA: 10/04/2025

## ÍNDICE

Introducción .....	3
Descripción .....	4
Justificación .....	5
Desarrollo .....	6
Conclusión .....	10
Referencias .....	11

## Introducción

Entendemos por lenguaje en informática que es un sistema formal de comunicación que permite a los humanos dar instrucciones a una computadora. Estos lenguajes también llamados lenguajes de programación, permiten especificar que debe hacer la computadora, como hacerlo y cuando hacerlo.

El lenguaje en la informática nos permite a los humanos (programadores) transmitir instrucciones a la computadora de forma clara y precisa. Los lenguajes son claves ya que son la base para desarrollar programas, aplicaciones, sistemas operativos, etc.

Esto nos facilita la implementación de algoritmos y soluciones computacionales para diversos problemas.

Sus características claves son:

- Formal: Cuenta con una estructura y reglas específicas
- Sintaxis: Define la estructura de las instrucciones y la forma en que se combinan
- Semántica: Define el significado de las palabras claves y operadores del lenguaje
- Abstracción: Nos facilita la programación a un nivel más alto que el lenguaje de máquina.

Contamos con diferentes tipos de lenguajes:

- Lenguaje de nivel bajo.
- Lenguaje de nivel alto.
- Lenguajes compilados.
- Lenguajes interpretados.

## **Descripción**

A lo largo de esta actividad comprenderemos como se realiza un código en un lenguaje de programación que en esta ocasión haremos uso del lenguaje C. Observaremos como es que se desarrolla cada una de las funciones en específico para que tenga un funcionamiento correcto y lineal nuestro código.

Estaremos aprendiendo como es que cada característica es clave para que el código no tenga ningún error al momento de realizarlo y ejecutarlo en nuestra terminal de comandos.

Observaremos como este compuesto por símbolos, reglas sintácticas y reglas semánticas expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construyen el código fuente de una aplicación o pieza de software determinado.

Veremos los lenguajes de diferente nivel:

- Nivel Bajo
- Nivel Alto
- Nivel medio
- Imperativos
- Funcionales

Los diferentes lenguajes que existe para cada área en el que se utilizan:

- BASIC
- COBOL
- FORTRAN
- Java

Como distinguir los diferentes tipos de lenguajes informáticos ya que cada uno representa una categoría más amplia que otros.

## **Justificación**

Entender la importancia de cómo funciona los lenguajes de informática es vital ya que recientemente la tecnología ha cobrado un gran impacto en el mundo profesional, yan parte de eso a sido gracias a la programación, ya que tanto las aplicaciones, paginas web, softwares y además herramientas para el trabajo remoto están elaboradas a base de programación, volviéndose una pieza importante para el mundo laborar actual y una habilidad de alta demanda en campo laboral.

Conocer los distintos tipos de programación que pueden ser utilizados de acuerdo con las necesidades tecnológicas de cada persona o empresa nos ayuda a realizar de forma mas efectiva, sencilla y concretas nuestras actividades diarias con los niveles bajos, nivel medio y el nivel alto.

Son diversos los diferentes tipos de lenguaje de programación y al dominar uno de ellos pueden ayudar a manejar de forma más sencilla los demás lenguajes.

Aprender a programar se ha convertido en una necesidad para la toma de decisiones a partir del análisis de datos masivos, la inteligencia artificial y la robótica. Al conocer sobre la programación puedes abrir muchas puertas a mejores empleos y ver de diferentes formas el mundo a nuestro alrededor.

## Desarrollo

```
1  ****
2
3          Online C Compiler.
4      Code, Compile, Run and Debug C program online.
5  Write your code in this editor and press "Run" button to compile and execute it.
6
7 ****
8
9 #include <stdio.h>
10
11 int main() {
12     int n2;
13     int n3;
14     int num;
15     n3 = 0;
16     printf("Ingrese un numero\n");
17     scanf("%i", &num);
18     for (n2=1; n2<=num; ++n2) {
19         if ((num%n2)==0) {
20             n3 = n3+1;
21         }
22     }
23     if ((n3==2)) {
24         printf("El numero %i Si es primo\n", num);
25     } else {
26         printf("El numero %i No es primo\n", num);
27     }
28     return 0;
29 }
30
```

En esta figura podemos observar cómo está pidiendo al usurario un numero para hacer el procedimiento para conocer si es un numero primo o no primo, se puede observar como se utiliza la función “int” que es usada para ingresar un tipo dato y almacenar números enteros, esta función no solo se utiliza en lenguaje C, en otros lenguajes como C++, C# y entre otros muchos lenguajes es utilizado esta función para almacenar un dato.

```
1 //  
2  
3             Online C Compiler.  
4             Code, Compile, Run and Debug C program online.  
5 Write your code in this editor and press "Run" button to compile and execute it.  
6  
7 ****  
8  
9 #include <stdio.h>  
10  
11 int main() {  
12     int n2;  
13     int num;  
14     printf("Ingrrese un numero\n");  
15     scanf("%i", &num);  
16     n2 = (int) num%2;  
17     if (n2==0) {  
18         printf("%i Es par\n", num);  
19     } else {  
20         printf("%i Es impar\n", num);  
21     }  
22     return 0;  
23 }  
24 }
```

En esta figura observamos como ahora queremos saber si el numero ingresado es par o impar, de igual forma se utiliza lo que seria el int pero también podemos observar que esta ocasión solo tenemos una comparativa para saber si el numero pertenece a un numero par o un número no par, observamos como usamos la función “scanf” para leer datos que ya sean escritos por el usuario o un de un archivo y los almacena en variables.

```

1  ****
2
3          Online C Compiler.
4          Code, Compile, Run and Debug C program online.
5  Write your code in this editor and press "Run" button to compile and execute it.
6
7 ****
8
9 #include<stdio.h>
10
11 int main() {
12     int n2;
13     int n3;
14     int n4;
15     int num;
16     printf("ingrese un numero de 4 digitos\n");
17     scanf("%i", &num);
18     n2 = num;
19     n4 = 0;
20     while (n2>0) {
21         n3 = n2%10;
22         n2 = (n2-n3)/10;
23         n4 = (n4*10)+n3;
24     }
25     printf("Este es el numero %i\n", num);
26     printf("Este es el numero al revés %i\n", n4);
27     return 0;
28 }
```

En esta figura observamos como ahora esta pidiendo algo mas complejo que es voltear los números o penderlos al revés, esta función comparte similitud con las anteriores en muchos aspectos sin embargo vemos una función llamada “while”, esta nos permite repetir un bloque de código hasta que se cumpla una condición en específico. Mediante comparativa, divisiones, restas, multiplicaciones e incluso sumas, podemos llegar a tener el resultado que habíamos especificado y así obtener nuestro número al revés

Todas estos códigos a pesar que se pide una cosa diferente comparten muchas similitudes con los demás la principal es que todas tienen la apertura de `<stdio.h>` Esto contiene la información necesaria para incluir rutinas de entrada y salida en nuestro programa.

Los diferentes niveles del lenguaje de programación pueden ser utilizados de acuerdo con la necesidad del usuario y observaremos cuales son y como se utilizan.

- Bajo nivel: Son lenguajes diseñados para un hardware específico, que no puede ser migrado o exportado.
- Nivel medio: Se refiere a aquellos lenguajes que permiten operaciones de alto nivel y así mismo la gestión local de la arquitectura del sistema.
- Alto nivel: Son lenguajes más universales que pueden emplearse a indistintamente en la arquitectura del hardware.
- Lenguajes imperativos: Son aquellos menos flexibles, dada la secuencialidad en el que construyen sus instrucciones.
- Lenguajes funcionales: También llamados procedimentales, estos lenguajes programan mediante funciones que son invocadas conforme a la entrada recibida, que a su vez son resultados de otra funciones.

## **Conclusión**

En esta unidad aprendimos como es importante dominar los lenguajes de programación en la era de la transformación digital. La programación es una habilidad de alta demanda que va más allá de cualquier lenguaje, ayudando a aprender a solucionar problemas de lógica e ingenio.

Como los lenguajes de programación se clasifican en diferentes niveles y existen varios tipos, como Java, Ruby, ASP.NET, PHP, Python, JavaScript y Objective-C.

La implementación de lenguajes de programación permite el trabajo conjunto y coordinado, a través de un conjunto de afín y finito de instrucciones posibles, de diversos programas o arquitectos de software, para el cual estos lenguajes imitan, la lógica de los lenguajes humanos o naturales.

Entendemos que cada lenguaje de programación tiene una amplia categoría, donde están contenidos los lenguajes de programación y muchos otros protocolos informáticos, como puede ser el HTML de las páginas web.

Como nos ayuda a aprender a solucionar problemas de lógica e ingenio, creando soluciones desde cero a paradigmas complejos con base en un pensamiento estructurado, donde no solo es un proceso rígido, si no también requiere de mucha creatividad ya que cada solución es única y cada persona tiene su propio método y razonamiento.

## Referencias

Gortázar, F. (2016). “*Lenguajes de programación y procesadores.*” Editorial Universitaria Ramón Areces: España. Vol. 2, núm. 21, págs. 193-240. ISBN: 9788499612492. Recuperado de: [https://books.google.es/books?hl=es&lr=&id=eHL-DAAAQBAJ&oi=fnd&pg=PA193&dq=tipos+de+lenguajes+de+programaci%C3%B3n&ots=QSs76XE35H&sig=OqPfiqczMvHAEKai8kond\\_i-QKE#v=onepage&q=tipos%20de%20lenguajes%20de%20programaci%C3%B3n&f=false](https://books.google.es/books?hl=es&lr=&id=eHL-DAAAQBAJ&oi=fnd&pg=PA193&dq=tipos+de+lenguajes+de+programaci%C3%B3n&ots=QSs76XE35H&sig=OqPfiqczMvHAEKai8kond_i-QKE#v=onepage&q=tipos%20de%20lenguajes%20de%20programaci%C3%B3n&f=false)

González, F. “*Paper Lenguajes de Programación.*” Escuela de Ingeniería Informática: Chile. Obtenido de: [https://d1wqxts1xzle7.cloudfront.net/31513342/90710967-Paper-Lenguajes-de-Programacion-2012-libre.pdf?1392346540=&response-content-disposition=inline%3B+filename%3D90710967\\_Paper\\_Lenguajes\\_de\\_Programacion.pdf&Expires=1744336740&Signature=QLIXAnINIPTfMFNIElbq5ArzKkM7zMwTR-yh8ETO~A7LC94sXuJP4zkL38LKuUYtrlbWH5GMYvng~-Rdd7D5rPVInlyNIjQh0DiUaeCOsQeBMBrnjvyA7IZYucARzJ3~FDCyGB9SK-AeDBEuxzfb0t8qZErv11KDjyJrwaYoouHpVruLuIUNoXR0Kz~Y2NIHxI-Fa2S4v~9dTo9ErQ8woGEAoa4D63ZRzlzYqHO50duP2J-zXfUAh4HGp~w9eaLqO8Qgp9BG3UFsx5Qi~6vNgUUM~uQUM9So~yCJ1-JE6DfefzT9UmRi4dn1pe-ugVRXYFJsKEkDwXy9umoGMRZLxg &Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqxts1xzle7.cloudfront.net/31513342/90710967-Paper-Lenguajes-de-Programacion-2012-libre.pdf?1392346540=&response-content-disposition=inline%3B+filename%3D90710967_Paper_Lenguajes_de_Programacion.pdf&Expires=1744336740&Signature=QLIXAnINIPTfMFNIElbq5ArzKkM7zMwTR-yh8ETO~A7LC94sXuJP4zkL38LKuUYtrlbWH5GMYvng~-Rdd7D5rPVInlyNIjQh0DiUaeCOsQeBMBrnjvyA7IZYucARzJ3~FDCyGB9SK-AeDBEuxzfb0t8qZErv11KDjyJrwaYoouHpVruLuIUNoXR0Kz~Y2NIHxI-Fa2S4v~9dTo9ErQ8woGEAoa4D63ZRzlzYqHO50duP2J-zXfUAh4HGp~w9eaLqO8Qgp9BG3UFsx5Qi~6vNgUUM~uQUM9So~yCJ1-JE6DfefzT9UmRi4dn1pe-ugVRXYFJsKEkDwXy9umoGMRZLxg &Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA)

Lizcano, D. (2018). “*Fundamentos de la programación.*” Repositorio Institucional de la Universidad a Distancia de Madrid: España. ISBN: 9788445436820. Recuperado de: <https://udimundus.udima.es/handle/20.500.12226/705>

Minaya, C., Loor, J., Briones, J. (2018). “*LOS PILARES DE LA PROGRAMACIÓN ORIENTADA A LA WEB: UN ENFOQUE TEÓRICO.*” Universidad Ciencia y Tecnología: Colombia. Vol. 1, núm. 9. Recuperado de: <https://uctunexpo.autanabooks.com/index.php/uct/article/view/42>

Challenger, I., Díaz, R., Becerra, R. (2014). “*El lenguaje de programación Python.*” Ciencias Holguín: Cuba. Vol. 2, núm. 2, págs. 1-13. EISSN: 10272127. Obtenido de: <https://www.redalyc.org/pdf/1815/181531232001.pdf>

Galarza, J. (2022) “*Evolución de los sistemas de lenguaje de programación a lo largo de la historia.*” Journal of Engineering Science: Ecuador. Vol. 4, núm. 10, págs: 14-26. DOI: <https://doi.org/10.53734/esci.vol4.id237> Recuperado de: <https://revista.estudioidea.org/ojs/index.php/esci/article/view/237>

Osio, R. (2013) “*Metodología innovadora para el estudio y programación de microprocesadores en arquitectura de computadoras.*” Repositorio Institucional de la Universidad Nacional de La Plata: Argentina. Págs: 1624-1630. Recuperado de: <http://sedici.unlp.edu.ar/handle/10915/31694>