**ANEXO FORMATO COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Generación de códigos de *software* con inteligencia artificial |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 22050109: Desarrollar solución de *software* de acuerdo con especificaciones de diseño y marcos de referencia. | RESULTADOS DE APRENDIZAJE | Caracterizar herramienta de inteligencia artificial generativa según aplicabilidad en el desarrollo de *software*. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | 03 |
| NOMBRE DEL COMPONENTE FORMATIVO | Fundamentos de programación estructurada y herramientas de desarrollo |
| BREVE DESCRIPCIÓN | Este componente formativo está orientado al reconocimiento, la configuración y el uso eficiente de los editores de código fuente como herramienta fundamental para el desarrollo de *software*, aplicando la programación estructurada. Se exploran conceptos claves como la estructura de la interfaz de los editores, sus funcionalidades básicas o la configuración del entorno de desarrollo. |
| PALABRAS CLAVE | Código fuente, editor de código fuente, desarrollo de *software*, Visual Studio Code, lenguaje de programación, programación estructurada |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | CIENCIAS NATURALES, APLICADAS Y RELACIONADAS |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS**

1. Editor de código Fuente

1.1 Importancia

1.2 Editores populares

1.3 Interfaz y estructura

1.4 Instalación

1.5 Configuración del entorno de trabajo

* 1. Uso del terminal integrado

2. Programación estructurada

2.1 Principios

2.2 Características

2.3 Estructuras

2.4 Funciones y modularidad

2.5 Ejemplos

1. **INTRODUCCIÓN**

La industria del *software* camina a pasos agigantados, lo que demuestra que su avance acelerado junto al protagonismo que viene teniendo la inteligencia artificial, está llevando a que los profesionales en desarrollo de *software* sean más demandados; aunque esta actividad ya no es solo exclusividad de ellos, sino que por el contrario, profesionales de otras áreas también se han visto en la necesidad de adquirir conocimientos en este campo, para hacer frente a los desafíos que se van gestando y así mismo, contar con mejores herramientas para realizar y agilizar los procesos en sus áreas profesionales.

|  |  |
| --- | --- |
| Representación de la experiencia del usuario y diseño de la interfaz  [https://www.freepik.es/foto-gratis/representacion-experiencia-usuario-diseno-interfaz\_94963717.htm - fromView=search&page=1&position=0&uuid=b2f67ff2-4674-433b-9015-941e84011f4b&query=progrmacion](https://www.freepik.es/foto-gratis/representacion-experiencia-usuario-diseno-interfaz_94963717.htm#fromView=search&page=1&position=0&uuid=b2f67ff2-4674-433b-9015-941e84011f4b&query=progrmacion) | Lograr comprender la lógica que hay detrás de un algoritmo, aplicar un diseño estructurado y plasmarlo en un lenguaje de programación, había sido un reto exclusivo de pocos, pero con las herramientas que se tienen hoy en día y los nuevos objetivos que va persiguiendo la sociedad, es mayor el número de personas que ya están trabajando o buscando trabajar en la construcción de *software* de calidad, que pueda adaptarse fácilmente a los retos del entorno actual y seguirse optimizando, porque así como los desafíos no son estáticos, las soluciones que hoy se están planteando tampoco lo son. |

El componente Fundamentos de programación estructurada y herramientas de desarrollo, da respuesta a esa necesidad, brindando un espacio formativo para fortalecer las competencias en la construcción y generación lógica del código fuente, así como del aprovechamiento de los editores de código y demás herramientas de desarrollo que permitan analizar los problemas, plantear una solución lógica o en su defecto interactuar con modelos de inteligencia artificial generativa, para generar soluciones que se puedan integrar al flujo de trabajo del desarrollo de *software*. De este modo, este componente contribuye en la adquisición de habilidades que son valoradas en el sector tecnológico, facilitando esa interacción entre los desarrolladores, las herramientas de IA y el entorno profesional.

1. **DESARROLLO DE CONTENIDOS**
2. **Editor de código fuente**

El editor de código fuente es una herramienta utilizada para escribir, corregir, modificar y gestionar código en distintos lenguajes de programación. Su uso resulta fundamental en el desarrollo de software, ya que permite crear aplicaciones y programas para entornos de escritorio, móviles o web. Además, facilita la depuración de errores, promueve el trabajo colaborativo —especialmente al integrarse con sistemas de control de versiones— y contribuye a una mayor organización en el proceso de desarrollo (Id Digital School, 2024).

|  |  |
| --- | --- |
| Desarrollo de aplicaciones para computadoras de escritorio y teléfonos inteligentes  [https://www.freepik.es/vector-gratis/desarrollo-aplicaciones-computadoras-escritorio-telefonos-inteligentes\_10276838.htm - fromView=search&page=1&position=3&uuid=b2f67ff2-4674-433b-9015-941e84011f4b&query=progrmacion](https://www.freepik.es/vector-gratis/desarrollo-aplicaciones-computadoras-escritorio-telefonos-inteligentes_10276838.htm#fromView=search&page=1&position=3&uuid=b2f67ff2-4674-433b-9015-941e84011f4b&query=progrmacion) | Es un programa diseñado para redactar y editar código, ofreciendo funciones que optimizan el trabajo del programador, tales como autocompletado, resaltado de sintaxis e indentación automática. Suele consumir pocos recursos del sistema, y su disponibilidad varía según el sistema operativo y el modelo de licenciamiento (gratuito o de pago). La elección del editor depende en gran medida de las preferencias personales del desarrollador (UNIR, 2023). |

Se trata, en esencia, de una herramienta especializada en la edición y mantenimiento del código fuente en uno o varios lenguajes de programación. A diferencia de los editores de texto convencionales, los editores de código están diseñados para optimizar el trabajo del programador mediante características como navegación por la estructura del proyecto, autocompletado inteligente, detección temprana de errores, entre otras funcionalidades.

|  |  |
| --- | --- |
| Pueden funcionar de manera independiente o integrarse dentro de un entorno de desarrollo integrado (IDE). Su correcta elección influye directamente en la productividad y en la calidad del producto final. Entre los más populares se encuentran *Visual Studio Code*, *Sublime Text*, *Vim* e *IntelliJ IDEA*. En el flujo de trabajo actual, el editor de código fuente se ha convertido en una herramienta indispensable, permitiendo centralizar el diseño, la construcción y la evolución del *software* con mayor organización y control. | Representación de la experiencia del usuario y diseño de la interfaz  [https://www.freepik.es/foto-gratis/representacion-experiencia-usuario-diseno-interfaz\_94963696.htm - fromView=search&page=1&position=1&uuid=b2f67ff2-4674-433b-9015-941e84011f4b&query=progrmacion](https://www.freepik.es/foto-gratis/representacion-experiencia-usuario-diseno-interfaz_94963696.htm#fromView=search&page=1&position=1&uuid=b2f67ff2-4674-433b-9015-941e84011f4b&query=progrmacion) |

Dentro de las principales ventajas por las cuales se les recomienda a los programadores utilizar un buen editor de código fuente están (Id digital school, 2024):

|  |  |  |
| --- | --- | --- |
| **Pestañas** | | |
| **Resaltado de sintaxis** | Facilita la identificación de la estructura del código y la detección de errores sintácticos mediante el uso de colores que diferencian palabras clave, funciones, variables, entre otros. | trabajo en equipo de negocios con el sitio web de la computadora  [https://www.freepik.es/vector-gratis/trabajo-equipo-negocios-sitio-web-computadora\_5685540.htm - fromView=search&page=1&position=31&uuid=5192bc53-aaf6-4392-804b-1a91403152a8&query=sintaxis+codigo+programacion](https://www.freepik.es/vector-gratis/trabajo-equipo-negocios-sitio-web-computadora_5685540.htm#fromView=search&page=1&position=31&uuid=5192bc53-aaf6-4392-804b-1a91403152a8&query=sintaxis+codigo+programacion) |
| **Autocompletado** | Ayuda en la escritura del código al sugerir automáticamente palabras clave, variables, funciones y estructuras comunes, según el lenguaje de programación utilizado. | Arte del canal de YouTube de la noche del juego  [https://www.freepik.es/vector-gratis/arte-canal-youtube-noche-juego\_172780645.htm - fromView=search&page=1&position=0&uuid=720dc423-04be-4837-8ad6-b1b9baf54fde&query=codigo+programacion+ia](https://www.freepik.es/vector-gratis/arte-canal-youtube-noche-juego_172780645.htm#fromView=search&page=1&position=0&uuid=720dc423-04be-4837-8ad6-b1b9baf54fde&query=codigo+programacion+ia) |
| **Depuración integrada** | Permite identificar y corregir errores directamente desde el editor, con funciones como puntos de interrupción, ejecución paso a paso y seguimiento de variables, sin necesidad de herramientas externas. | Ilustración de npl de diseño plano dibujado a mano  [https://www.freepik.es/vector-gratis/ilustracion-npl-diseno-plano-dibujado-mano\_22379569.htm - fromView=search&page=1&position=6&uuid=720dc423-04be-4837-8ad6-b1b9baf54fde&query=codigo+programacion+ia](https://www.freepik.es/vector-gratis/ilustracion-npl-diseno-plano-dibujado-mano_22379569.htm#fromView=search&page=1&position=6&uuid=720dc423-04be-4837-8ad6-b1b9baf54fde&query=codigo+programacion+ia) |
| **Control de versiones** | Integra herramientas como Git para gestionar cambios en el código, hacer seguimiento del historial y facilitar la colaboración entre desarrolladores. | El día del programador Póster  [https://www.freepik.es/vector-gratis/dia-programador-poster\_31282203.htm - fromView=search&page=1&position=13&uuid=720dc423-04be-4837-8ad6-b1b9baf54fde&query=codigo+programacion+ia](https://www.freepik.es/vector-gratis/dia-programador-poster_31282203.htm#fromView=search&page=1&position=13&uuid=720dc423-04be-4837-8ad6-b1b9baf54fde&query=codigo+programacion+ia) |
| **Extensibilidad** | Admite la instalación de extensiones o *plugins* para personalizar y ampliar las funcionalidades del editor, adaptándolo a diferentes lenguajes, *frameworks* o flujos de trabajo. | Retrato de programador trabajando con pc.  [https://www.freepik.es/vector-premium/retrato-programador-trabajando-pc\_4956913.htm - fromView=search&page=1&position=34&uuid=720dc423-04be-4837-8ad6-b1b9baf54fde&query=codigo+programacion+ia](https://www.freepik.es/vector-premium/retrato-programador-trabajando-pc_4956913.htm#fromView=search&page=1&position=34&uuid=720dc423-04be-4837-8ad6-b1b9baf54fde&query=codigo+programacion+ia) |

* 1. **Importancia**

Los editores de código fuente son importantes, ya que en ellos se centraliza el trabajo del desarrollo de *software*, convirtiéndose en la herramienta principal, en donde los programadores escriben, organizan, administran y mantienen el código que da vida a los programas, sistemas y demás plataformas digitales. Se puede decir que relevancia estaba basada en aspectos claves cómo:

|  |  |  |
| --- | --- | --- |
| **Slide** | | |
| **Eficiencia en la programación** | Funcionalidades como la navegación rápida por archivos, autocompletado, atajos de teclado y formateo automático aceleran la escritura del código y reducen errores. | Un chatbot de inteligencia artificial está ayudando a los programadores a escribir código fuente de renderización 3D  [https://www.freepik.es/fotos-premium/chatbot-inteligencia-artificial-esta-ayudando-programadores-escribir-codigo-fuente-renderizacion-3d\_149556329.htm - fromView=search&page=1&position=7&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/chatbot-inteligencia-artificial-esta-ayudando-programadores-escribir-codigo-fuente-renderizacion-3d_149556329.htm#fromView=search&page=1&position=7&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Prevención y detección de errores** | El resaltado de sintaxis permite identificar errores comunes como palabras mal escritas, paréntesis sin cerrar o estructuras mal formadas durante la escritura. | Ai tech utiliza Internet para conectar Chatgpt Chat con inteligencia artificial AI para generar contenido  [https://www.freepik.es/fotos-premium/ai-tech-utiliza-internet-conectar-chatgpt-chat-inteligencia-artificial-ai-generar-contenido\_137561398.htm - fromView=search&page=1&position=11&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/ai-tech-utiliza-internet-conectar-chatgpt-chat-inteligencia-artificial-ai-generar-contenido_137561398.htm#fromView=search&page=1&position=11&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Organización del código** | Permite estructurar archivos y carpetas, aplicar indentación automática y trabajar con múltiples lenguajes, lo que facilita el mantenimiento del software. | Experiencia en programación con persona que trabaja con códigos en computadora  [https://www.freepik.es/foto-gratis/experiencia-programacion-persona-que-trabaja-codigos-computadora\_38669465.htm - fromView=search&page=1&position=14&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/foto-gratis/experiencia-programacion-persona-que-trabaja-codigos-computadora_38669465.htm#fromView=search&page=1&position=14&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Personalización y extensibilidad** | Los editores modernos admiten extensiones, configuraciones y temas que adaptan el entorno a las necesidades del programador. | Collaje de antecedentes de programación  [https://www.freepik.es/foto-gratis/collaje-antecedentes-programacion\_34089161.htm - fromView=search&page=1&position=20&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/foto-gratis/collaje-antecedentes-programacion_34089161.htm#fromView=search&page=1&position=20&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Integración con otras herramientas** | Se pueden integrar con sistemas de control de versiones, depuradores, terminales, plataformas en la nube y gestores de paquetes, centralizando el flujo de trabajo. | Composición con sistema html para sitios web  [https://www.freepik.es/foto-gratis/profesional-ciberseguridad-oriente-medio\_136128971.htm - fromView=search&page=1&position=23&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/foto-gratis/profesional-ciberseguridad-oriente-medio_136128971.htm#fromView=search&page=1&position=23&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Compatibilidad y portabilidad** | La mayoría de los editores son multiplataforma, lo que permite trabajar en diferentes sistemas operativos sin pérdida de configuración o rendimiento. | Robot futurista inteligencia artificial concepto de tecnología de IA revolucionario  [https://www.freepik.es/fotos-premium/robot-futurista-inteligencia-artificial-concepto-tecnologia-ia-revolucionario\_30541254.htm - fromView=search&page=1&position=35&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/robot-futurista-inteligencia-artificial-concepto-tecnologia-ia-revolucionario_30541254.htm#fromView=search&page=1&position=35&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |

* 1. **Editores populares**

A la hora de elegir un buen editor de código para trabajar hay que tener en cuenta el tipo de proyecto en que se está trabajando o se desea desarrollar, los objetivos de programación que se están persiguiendo y el nivel de habilidad que se ha logrado como programador, con el fin de trabajar con el editor que más se ajuste a las necesidades. A la hora de elegir un editor, se recomienda tener en cuenta puntos como rendimiento optimizado, características del editor, funciones de navegación, código de referencias y que tan personalizable es (Weisheim, 2022). A continuación se comparte una lista de editores, donde podrán encontrar un alista de herramientas para facilitar el trabajo de la programación:

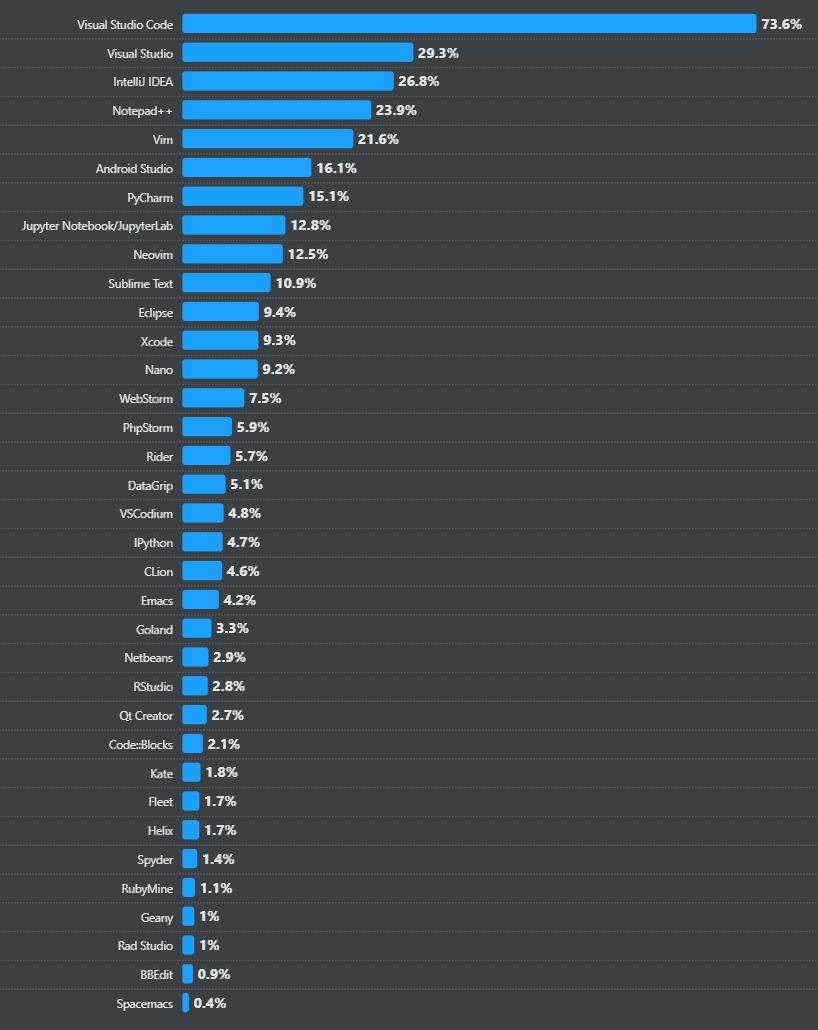
**Tabla 1.** Lista de editores de código fuente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Licencia** | **Link de descarga** | **Lenguajes compatibles** | **Plataformas compatibles** |
| Visual Studio Code | Gratuita | <https://code.visualstudio.com/> | JavaScript, TypeScript, Python, C++, entre otros | macOS, Windows, Linux |
| Sublime Text | Gratuita  (99 USD por licencia) | <https://www.sublimetext.com/> | C++, Python, PHP, Rails y más | Windows, Linux, macOS |
| Atom | Gratuita (descontinuado) | <https://github.com/atom/atom/releases> | C, C++, COBOL, HTML, CSS, Java, PHP, Ruby, Scala, SQL | Windows, macOS, Linux |
| Notepad++ | Gratuita | <https://notepad-plus-plus.org/downloads/> | Más de 70 lenguajes como HTML, CSS, JavaScript, PHP, XML, C++, Swift | Windows, Linux, UNIX |
| Vim | Gratuita | <https://www.vim.org/download.php> | Casi todos los lenguajes | Unix, Windows, macOS y más |
| NetBeans | Gratuita | <https://netbeans.apache.org/download/> | HTML5, C++, JavaScript, PHP, Java y más | Windows, Linux, macOS, BSD |
| TextMate | Gratuita | <https://macromates.com/> | AppleScript, C++, HTML, CSS, Java, PHP, Python, SQL y más | macOS |
| GNU Emacs | Gratuita | <https://www.gnu.org/software/emacs/download.html> | C, C++, Python, Lisp y más | Windows, Linux, macOS, BSD |
| Codeshare.io | Gratuita | <https://codeshare.io/24ejj7> | PHP, C#, SCSS, CSS, HTML | Navegadores web |

Nota. Elaboración propia con datos de (Weisheim, 2022), (UNIR, 2023)

Basados en la tabla anterior, donde se muestra una variedad de editores, elegir un buen entorno de programación es primordial para la productividad y la experiencia del desarrollador. De ahí que cuando se está comenzando, es bueno conocer sobre los diferentes *rankings* que se realizan a nivel mundial para ver cómo están posicionadas las herramientas que se utilizan en el campo de la informática, con el fin de estar al día con las tecnologías que se trabajan en el mercado. En ese orden de ideas, la encuesta global de Stack overflow, recopila una gran cantidad de datos, basado en el consenso de miles de profesionales para identificar las más utilizadas en diferentes áreas de la industria. Los resultados del 2024 posicionan al editor Visual Studio Code como líder indiscutible, con una preferencia del 73,6 %, lo que demuestra actualmente su hegemonía.

**Figura 1.** Ranking de editores de código del año 2024

****

Nora. Tomado de Stack Overflow (2024).

* 1. **Interfaz y estructura**

Hasta este punto, está bien definido que los editores de código fuente han sido diseñados para facilitar la escritura, lectura y gestión del código. Esto se debe, en parte, a la interfaz y a la estructura del editor, que permiten al programador interactuar de forma más fluida con la herramienta. Es importante tener presente que, aunque existe una amplia variedad de editores, cada uno posee particularidades propias. No obstante, la mayoría comparte ciertos componentes comunes que les otorgan una organización tanto visual como funcional. Entre estos componentes comunes, se destacan los siguientes:

**Figura 2.** Interfaz de Visual Studio Code con sus componentes principales.

A screenshot of a computer program

AI-generated content may be incorrect.

|  |  |
| --- | --- |
| Acordeón | |
| Barra de menú | Ubicada en la parte superior. Contiene funciones como: • Archivo: abrir, guardar, cerrar archivos. • Editar: copiar, pegar, buscar, reemplazar. • Ver: mostrar u ocultar paneles. • Ejecutar / Depurar: iniciar scripts, abrir consola o *debugger.* • Extensiones / Preferencias: configuración del editor. |
| Explorador de archivos (panel lateral) | Permite visualizar y navegar por archivos y carpetas del proyecto. • Crear, modificar o eliminar archivos y carpetas. • Abrir archivos en diferentes pestañas. • Ver la estructura general del proyecto. |
| Área de edición (editor de texto) | Espacio principal para escribir el código fuente. Incluye: • Resaltado de sintaxis. • Numeración de líneas. • Plegado de bloques. • Autocompletado y sugerencias. |
| Panel de extensiones o *plugins* | Permite buscar, instalar y gestionar extensiones para ampliar las funciones del editor: • Soporte para nuevos lenguajes. • Integración con bases de datos. • Sistemas de control de versiones. • Herramientas de depuración e inteligencia artificial. |
| Consola | Terminal integrada para ejecutar comandos y ver resultados o realizar pruebas sin salir del editor. |
| Depurador | Herramienta para: • Establecer puntos de interrupción. • Analizar variables y su comportamiento. • Ejecutar el código paso a paso. |
| Barra de estado | Ubicada en la parte inferior. Muestra información contextual como: • Lenguaje activo. • Codificación. • Tabulaciones. • Mensajes del sistema. • Estado del proyecto. |

* 1. **Instalación**

Antes de iniciar la programación o la generación de código, es fundamental contar con una herramienta que facilite este proceso. Visual Studio Code, conocido también como VS Code, se ha consolidado como el editor de código más utilizado en el ámbito del desarrollo de software, gracias a su ligereza, versatilidad y múltiples funciones integradas. Por esta razón, en esta sección se abordará su instalación. Para ello, se invita a seguir los pasos descritos en el video:

|  |
| --- |
| VIDEO 1 |

* 1. **Configuración del entorno de trabajo**

|  |  |
| --- | --- |
| Después de instalado el editor *Visual Studio Code*, el siguiente paso es realizar los ajustes al entorno de trabajo según las necesidades del desarrollador. Para ello se realizará una configuración básica inicial del editor y del lenguaje de programación de preferencia, el cual para esta ocasión se hará con *Python*, por ser uno de los lenguajes de programación más utilizados en el mundo y el de mayor crecimiento por el aporte que viene brindando a la inteligencia artificial y a otras tecnologías disruptivas. | Persona que trabaja con HTML en la computadora  [https://www.freepik.es/foto-gratis/persona-que-trabaja-html-computadora\_36190655.htm - fromView=search&page=1&position=33&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/foto-gratis/persona-que-trabaja-html-computadora_36190655.htm#fromView=search&page=1&position=33&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |

Dentro de la configuración del editor es posible elegir entre temas claros u oscuros para ajustar el color del entorno, cambiar el idioma a uno de preferencia, activar el guardado automático e instalar las extensiones necesarias, por ejemplo, para trabajar con un lenguaje específico. Por ello, se invita a revisar el video donde se explican los pasos para configurar el editor:

|  |
| --- |
| VIDEO 2 |

Para el desarrollo de los ejercicios se utilizará Python como lenguaje de programación de apoyo. Por ello, se invita a revisar el video donde se explican los pasos para su instalación y configuración en el editor.

|  |
| --- |
| VIDEO 3 |

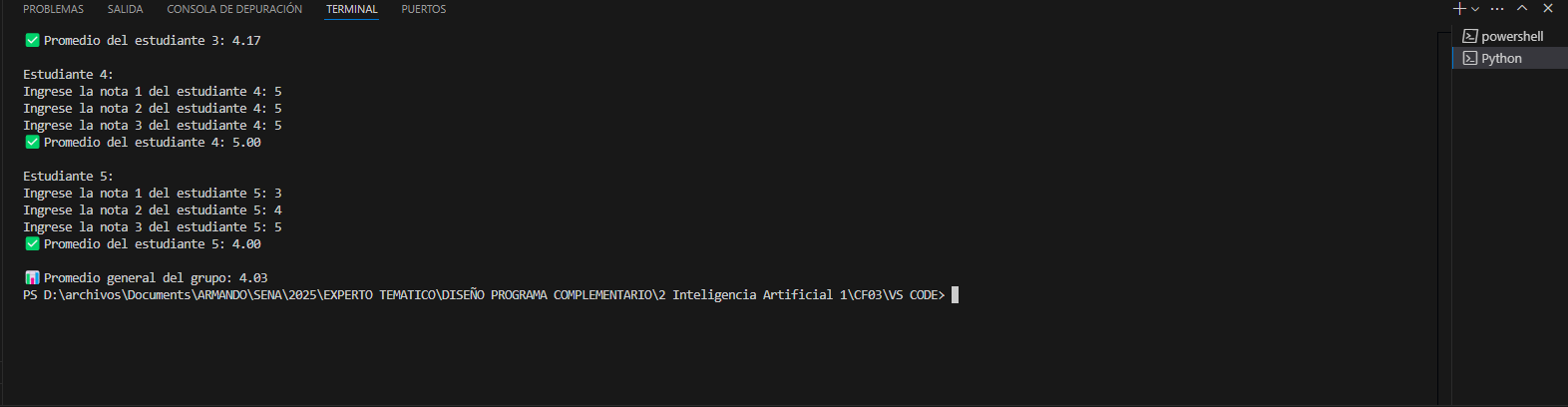
**1.6 Uso del terminal integrado**

Muchos de los editores de código incluyen un terminal integrado para que se puedan ejecutar directamente en ellos comandos desde el editor, sin la necesidad de abrir o utilizar una terminal externa. Esto permite acelerar el flujo de trabajo y mejorar la productividad cuando se trabaja a nivel profesional.

|  |  |
| --- | --- |
| En el caso del editor *Visual Studio Code*, contiene su propio terminal, lo cual lo hace una herramienta más robusta y autosuficiente para el desarrollo de programas y demás aplicaciones. Esta funcionalidad elimina la incomodes que representa en el programador el estar cambiando entre una ventana y otra, al tener que estar utilizando herramientas externas al momento de necesitar ejecutar comandos, tener que compilar programas, iniciar servidores locales, ejecutar scripts, instalar dependencias o trabajar con sistemas de control de versiones como *Git*. | Robot futurista inteligencia artificial concepto de tecnología de IA revolucionario  [https://www.freepik.es/fotos-premium/robot-futurista-inteligencia-artificial-concepto-tecnologia-ia-revolucionario\_81410162.htm - fromView=search&page=1&position=47&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/robot-futurista-inteligencia-artificial-concepto-tecnologia-ia-revolucionario_81410162.htm#fromView=search&page=1&position=47&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |

En el caso de *Visual Studio Code*, el terminal integrado actúa como una línea de comandos que se encuentra alojada en el editor y que se adapta al sistema operativo con el cual se esté trabajando, lo que agiliza el trabajo al permitir esa interacción fluida entre el código y la consola sin salir del entorno de desarrollo. Este editor permite trabajar con múltiples terminales en paralelo, así mismo elegir diferentes intérpretes de comando como *CMD, Bash, o PowerShell*, entre otros.

**Figura 3.** Visualización de terminal integrada en Visual Studio Code.



1. **Programación estructurada**

La programación estructurada es una de las formas de utilizar más utilizadas, sobre todo cuando se inicia en el mundo académico de la programación, y pertenece al paradigma de programación imperativo, cuya finalidad está basada en ir indicando paso a paso cómo resolver un problema. Su enfoque radica en buscar que el código sea sencillo, claro, ordenado y de fácil mantenibilidad, utilizando subrutinas y las estructuras de control (secuencia, selección e iteración):

Esta forma de programar surge del teorema del programa estructurado de Böhm y Jacopini en el año 1966, y fue apoyada por Dijkstra en 1968, quien promovió la eliminación del uso de instrucciones como *goto*, porque se tornaban confusas haciendo tediosa la programación; logrando que hoy en día se pueda contar con lenguajes de programación como *Java*, *C*, *Python* y *C+*+, los cuales soportan la programación estructurada (UNIR, 2022).

* 1. **Principios**

Antes de comenzar con la escritura de código, es bueno conocer algunos fundamentos que ayudaran a codificar de manera clara, confiable y organizada; para ello es bueno conocer unos principios básicos en la programación estructurada, que servirán de guía para trabajar en el desarrollo de *software* con la mejor calidad y adoptando buenas prácticas:

|  |  |  |
| --- | --- | --- |
| **Slide** | | |
| **Uso de estructuras de control básicas** | Se deben emplear tres estructuras clave: secuencia, selección (*if*, *switch*) e iteración (*for*, *while*, *do-while*) para estructurar adecuadamente un programa. | Neural networks expert in server farm fixing insufficient memory errors  [https://www.freepik.es/fotos-premium/neural-networks-expert-in-server-farm-fixing-insufficient-memory-errors\_416754337.htm - fromView=search&page=1&position=37&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/neural-networks-expert-in-server-farm-fixing-insufficient-memory-errors_416754337.htm%20-%20fromView=search&page=1&position=37&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Modularización del código** | El programa debe dividirse en subrutinas, funciones o procedimientos que realicen tareas específicas, facilitando la organización, reutilización y mantenimiento del código. | Hombre de tiro medio trabajando en una computadora portátil  [https://www.freepik.es/fotos-premium/hombre-tiro-medio-trabajando-computadora-portatil\_40728377.htm - fromView=search&page=1&position=45&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/hombre-tiro-medio-trabajando-computadora-portatil_40728377.htm#fromView=search&page=1&position=45&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Claridad y legibilidad** | El código debe ser comprensible para cualquier desarrollador. Se recomienda usar nombres descriptivos, comentarios explicativos y una organización ordenada del código. | Encendido el ordenador portátil gris  [https://www.freepik.es/foto-gratis/encendido-ordenador-portatil-gris\_12661377.htm - fromView=search&page=1&position=49&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/foto-gratis/encendido-ordenador-portatil-gris_12661377.htm#fromView=search&page=1&position=49&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Evitar instrucciones de salto incontrolado** | No se deben usar sentencias como *goto*, ya que dificultan el seguimiento del flujo y la comprensión del código. | Experiencia en programación con persona que trabaja con códigos en computadora  [https://www.freepik.es/fotos-premium/experiencia-programacion-persona-que-trabaja-codigos-computadora\_38669447.htm - fromView=search&page=2&position=18&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/experiencia-programacion-persona-que-trabaja-codigos-computadora_38669447.htm#fromView=search&page=2&position=18&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Diseño descendente** | El desarrollo debe iniciar con un análisis general del problema, descomponiéndolo en subproblemas que se resuelven mediante módulos, funciones o subrutinas. | Persona frente a la computadora trabajando en html  [https://www.freepik.es/foto-gratis/persona-frente-computadora-trabajando-html\_36189972.htm - fromView=search&page=2&position=31&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/foto-gratis/persona-frente-computadora-trabajando-html_36189972.htm#fromView=search&page=2&position=31&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| **Prueba y verificación sistemática** | El código debe permitir una verificación formal y pruebas que confirmen que cumple sus objetivos y que puede corregirse fácilmente si se requieren ajustes. | Un experto en el centro de servidores que utiliza inteligencia artificial para reducir el tiempo de inactividad  [https://www.freepik.es/foto-gratis/experto-centro-servidores-que-utiliza-inteligencia-artificial-reducir-tiempo-inactividad\_418121208.htm - fromView=search&page=2&position=39&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/foto-gratis/experto-centro-servidores-que-utiliza-inteligencia-artificial-reducir-tiempo-inactividad_418121208.htm#fromView=search&page=2&position=39&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |

* 1. **Características**

A continuación, se presenta un pódcast que resume las cualidades esenciales de un programa bien estructurado, con énfasis en claridad, modularidad y facilidad de mantenimiento.

|  |
| --- |
| PÓDCAST |

* 1. **Estructuras**

Todo programa escrito bajo el estándar de la programación estructurada sigue una lógica de programación que permite resolver problemas de una forma sistemática y organizada por medio de tres estructuras básicas: secuencia, selección e iteración. El dominar estas tres estructuras les permite a los programadores tener el control del flujo de un programa, tomar decisiones y automatizar los procesos repetitivos; con lo cual llegan a lograr soluciones eficientes. A continuación se ahonda en cada una de estas tres estructuras, con el fin de que se puedan conocer, dominar y aplicar en futuros desarrollos de *software*:

**Estructura de secuencia**

Representa la estructura de control más básica en la programación estructurada, y su principal función consiste en la ejecución de las instrucciones de un programa paso a paso, línea a línea, en el orden en que se han escrito en el código, en orden descendente, desde el inicio hasta el final del código.

|  |  |
| --- | --- |
| Mientras no se utilice una estructura de selección o iteración, cada instrucción se ejecuta de una vez, garantizando con ello un flujo lineal de ejecución, que puede llevar un seguimiento simple y predecible, al que incluso se le pueden realizar fácilmente pruebas para verificar la eficacia del código. | Ingeniero que integra la inteligencia artificial en la producción de fábricas fotovoltaicas  [https://www.freepik.es/foto-gratis/ingeniero-que-integra-inteligencia-artificial-produccion-fabricas-fotovoltaicas\_416930013.htm - fromView=search&page=2&position=49&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/foto-gratis/ingeniero-que-integra-inteligencia-artificial-produccion-fabricas-fotovoltaicas_416930013.htm#fromView=search&page=2&position=49&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |

Esta estructura representa la base del código, ya que las estructuras de selección e iteración están escritas bajo la base de la estructura de secuencia

Para conocer como se ve la estructura de secuencia, se comparten unos ejemplos:

|  |  |
| --- | --- |
| **Pestañas** | |
| **Código en Python** | edad\_actual = 25 edad\_futura = edad\_actual + 5 print("Edad dentro de 5 años:", edad\_futura) |
| **Código en C++** | #include <iostream> using namespace std;  int main() {  int edad\_actual = 25;  int edad\_futura = edad\_actual + 5;  cout << "Edad dentro de 5 años: " << edad\_futura << endl;  return 0; } |
| **Código en Java** | public class Main {  public static void main(String[] args) {  int edad\_actual = 25;  int edad\_futura = edad\_actual + 5;  System.out.println("Edad dentro de 5 años: " + edad\_futura);  } } |

**Estructura de selección**

Es una estructura de control que permite que se tomen decisiones mientras el programa esté corriendo. En otras palabras, para cualquier problema planteado, se presentan varios caminos, y de acuerdo a unas condiciones el programa elige un camino, dependiendo del resultado de las condiciones.

|  |  |
| --- | --- |
| A diferencia de la estructura de secuencia, aquí el camino no siempre será lineal, sino que podrá tomar un orden diferente en la ruta. Según la condición lógica, si es verdadera o falsa, se ejecuta una u otra línea o bloque de código. | Collaje de antecedentes de programación  [https://www.freepik.es/foto-gratis/collaje-antecedentes-programacion\_34089180.htm - fromView=search&page=3&position=2&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/foto-gratis/collaje-antecedentes-programacion_34089180.htm%20-%20fromView=search&page=3&position=2&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |

De acuerdo al lenguaje de programación en que se esté trabajando existen unos tipos de selección (pueden variar de un lenguaje a otro), de los cuales se comparten los más usados:

|  |  |  |
| --- | --- | --- |
| **Tarjetas** | | |
| *If simple* | Ejecuta un bloque si y solo si la condición es verdadera. Se ejecuta la condición con una sola acción. | mano icono  [https://www.freepik.es/icono/mano\_15119062 - fromView=search&page=1&position=1&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539](https://www.freepik.es/icono/mano_15119062#fromView=search&page=1&position=1&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539) |
| *If-else* | Cuando la condición es verdadera ejecuta un bloque; cuando es falsa, ejecuta un bloque diferente. Se usa cuando hay dos caminos alternativos. | codificación icono  [https://www.freepik.es/icono/codificacion\_10679123 - fromView=search&page=1&position=13&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539](https://www.freepik.es/icono/codificacion_10679123#fromView=search&page=1&position=13&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539) |
| *If-elif-else (Python) / else if (C++, Java)* | Permite elegir entre varias condiciones. De acuerdo a la que se cumpla, ejecuta un bloque exclusivo. Se emplea cuando hay más de dos condiciones posibles. | código icono  <https://cdn-icons-png.freepik.com/256/17432/17432078.png?ga=GA1.1.1998440075.1749696705&semt=ais_hybrid> |
| *Switch-case (C++, Java)* | Maneja múltiples opciones según el valor de una variable, ejecutando el bloque que corresponda. Se recomienda cuando existen opciones fijas, como en un menú. | texto icono  <https://cdn-icons-png.freepik.com/256/6867/6867388.png?ga=GA1.1.1998440075.1749696705&semt=ais_hybrid> |

Para conocer como se ve la estructura de selección, se comparten unos ejemplos:

|  |  |
| --- | --- |
| **Slide** | |
| **Código en C++ de if simple** | #include <iostream> using namespace std;  int main() {  int numero;  cout << "Ingrese un número: ";  cin >> numero;   if (numero > 0)  cout << "El número es positivo." << endl;   return 0; } |
| **Código en Java de if else** | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.print("Ingrese un número: ");  int numero = sc.nextInt();   if (numero % 2 == 0) {  System.out.println("El número es par.");  } else {  System.out.println("El número es impar.");  }  } } |
| **Código en C++ de else if** | #include <iostream> using namespace std;  int main() {  float nota;  cout << "Ingrese la nota: ";  cin >> nota;   if (nota >= 4.5)  cout << "Excelente" << endl;  else if (nota >= 3.5)  cout << "Bueno" << endl;  else if (nota >= 3.0)  cout << "Aceptable" << endl;  else  cout << "Reprobado" << endl;   return 0; } |
| **Código en Java de Switch-case** | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.println("1. Sumar\n2. Restar\n3. Salir");  System.out.print("Ingrese una opción: ");  int opcion = sc.nextInt();   switch (opcion) {  case 1:  System.out.println("Has elegido Sumar.");  break;  case 2:  System.out.println("Has elegido Restar.");  break;  case 3:  System.out.println("Has elegido Salir.");  break;  default:  System.out.println("Opción no válida.");  }  } } |

**Estructura de iteración**

La iteración, también conocida como ciclo o bucle, es una estructura de control utilizada para automatizar procesos repetitivos, y su finalidad es permitir la repetición de un bloque de instrucciones mientras se esté cumpliendo una condición lógica. Cuando la condición deja de cumplirse, el flujo del programa sale del bucle y sigue con el resto de instrucciones si las hay, de lo contrario finaliza. La iteración es un proceso clave de mucha ayuda para la solución de problemas donde se hace necesaria la repetición de acciones, por ejemplo:

|  |  |
| --- | --- |
| Administrador certificado que actualiza los sistemas de IA  [https://www.freepik.es/fotos-premium/mano-robot-renderizado-3d-trabajando-teclado-computadora\_17154918.htm - fromView=search&page=3&position=9&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/mano-robot-renderizado-3d-trabajando-teclado-computadora_17154918.htm#fromView=search&page=3&position=9&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) | * Procesar listas o series de datos. * Repetir operaciones matemáticas. * Crear menús interactivos. * Realizar cálculos acumulativos. |

De acuerdo al lenguaje de programación en que se esté trabajando existen unos tipos de iteración (pueden variar de un lenguaje a otro), de los cuales se comparten los más usados:

|  |  |  |
| --- | --- | --- |
| **Tarjetas** | | |
| *While* | La condición se evalúa antes de cada repetición y, si se cumple, el bloque de código se repite mientras siga siendo verdadera. Se puede usar cuando no se sabe cuántas veces se ejecutarán las instrucciones. | computadora icono  [https://www.freepik.es/icono/codificacion\_6991035 - fromView=search&page=1&position=61&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539](https://www.freepik.es/icono/codificacion_6991035#fromView=search&page=1&position=61&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539) |
| *Do-while (C++, Java)* | Funciona como el *while*, pero la condición se evalúa después de la primera ejecución, lo que garantiza que el bloque se ejecuta al menos una vez. Se utiliza cuando se requiere que las instrucciones se realicen al menos una vez. | resolviendo icono  [https://www.freepik.es/icono/resolviendo\_4765938 - fromView=search&page=1&position=93&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539](https://www.freepik.es/icono/resolviendo_4765938#fromView=search&page=1&position=93&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539) |
| *For* | La repetición ocurre un número determinado de veces, controlado por un contador. Se usa cuando se conoce de antemano cuántas veces se desea repetir la acción. | programación icono  [https://www.freepik.es/icono/programacion\_6062646 - fromView=search&page=2&position=9&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539](https://www.freepik.es/icono/programacion_6062646#fromView=search&page=2&position=9&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539) |

Para conocer como se ve la estructura de selección, analice los siguientes ejemplos:

|  |  |
| --- | --- |
| **Slide** | |
| **Código en C++ de while** | #include <iostream> using namespace std;  int main() {  int i = 1;  while (i <= 5) {  cout << "Número: " << i << endl;  i++;  }  return 0; } |
| **Código en Java de do while** | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  int opcion;  do {  System.out.println("1. Opción A\n2. Opción B\n3. Salir");  System.out.print("Elija una opción: ");  opcion = sc.nextInt();  } while (opcion != 3);   System.out.println("Programa finalizado.");  } } |
| **Código en C++ de else if** | #include <iostream> using namespace std;  int main() {  float nota;  cout << "Ingrese la nota: ";  cin >> nota;   if (nota >= 4.5)  cout << "Excelente" << endl;  else if (nota >= 3.5)  cout << "Bueno" << endl;  else if (nota >= 3.0)  cout << "Aceptable" << endl;  else  cout << "Reprobado" << endl;   return 0; } |
| **Código en Python del for** | suma = 0 for i in range(1, 11):  suma += i print("La suma es:", suma) |

* 1. **Funciones y modularidad**

Uno de los principios de la programación estructurada es dividir el código de un programa en partes más pequeñas, que a la vez sean independientes y reutilizables; esto se logra con funciones (conocidas también como subrutinas o procedimientos), las cuales permiten aplicar el principio de modularidad. Cuando se logra entender el concepto de funciones, será más fácil para el programador desarrollar y mantener sus programas. Ahondando en lo conceptos se puede decir que:

|  |  |  |
| --- | --- | --- |
| **Pestañas** | | |
| Funciones | Bloque de código que realiza una operación particular dentro de un programa. Puede incluir parámetros de entrada y devolver un valor como salida. Facilitan el encapsulamiento de operaciones comunes en un bloque reutilizable. Se recomienda usar nombres claros y nemotécnicos para identificar fácilmente la función (Microsoft, 2023). | Optimización SEO. Tecnología moderna. Se resalta la sintaxis de PHP. Escribir funciones de programación en una computadora portátil. Tendencia de Big Data e Internet de las cosas.  [https://www.freepik.es/fotos-premium/optimizacion-seo-tecnologia-moderna-resalta-sintaxis-php-escribir-funciones-programacion-computadora-portatil-tendencia-big-data-e-internet-cosas\_17975856.htm - fromView=search&page=3&position=19&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/optimizacion-seo-tecnologia-moderna-resalta-sintaxis-php-escribir-funciones-programacion-computadora-portatil-tendencia-big-data-e-internet-cosas_17975856.htm#fromView=search&page=3&position=19&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |
| Modularidad | Práctica de dividir un programa en módulos más pequeños, cada uno con funciones y responsabilidades claras. Esta descomposición permite una mejor gestión, mantenimiento y comprensión del programa, en contraste con un código extenso y monolítico (Casero, 2023). | Especialista en IA revisa gráficos de datos que trabajan en la industria de la tecnología  [https://www.freepik.es/fotos-premium/especialista-ia-revisa-graficos-datos-que-trabajan-industria-tecnologia\_258642441.htm - fromView=search&page=3&position=44&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia](https://www.freepik.es/fotos-premium/especialista-ia-revisa-graficos-datos-que-trabajan-industria-tecnologia_258642441.htm#fromView=search&page=3&position=44&uuid=8ce6fcaf-c5c0-47c2-b703-4e7c55348638&query=codigo+programacion+ia) |

Una función puede recibir unos datos de entrada (parámetros), puede devolver un resultado (valor de retorno) y puede ser reutilizada en diferentes partes de un programa, para ello después de definida, solo tiene que ser invocada y podrá cumplir con el objetivo o la tarea para la cual fue creada.

Así es la estructura de una función:

|  |
| --- |
| Función nombre\_función(parámetros) → valor\_retorno  instrucciones  return valor\_retorno |

Y aunque ya se han compartido ejemplos de funciones en anteriores códigos, se comparten ejemplos específicos de funciones en diversos lenguajes de programación:

|  |  |
| --- | --- |
| **Pestañas** | |
| **Función en Python** | def calcular\_area\_circulo(radio):  return 3.1416 \* radio \* radio  # Uso de la función r = float(input("Ingrese el radio: ")) area = calcular\_area\_circulo(r) print("El área del círculo es:", area) |
| **Función en C++** | #include <iostream> using namespace std;  // Definición de la función double calcular\_area\_circulo(double radio) {  return 3.1416 \* radio \* radio; }  int main() {  double r;  cout << "Ingrese el radio: ";  cin >> r;  double area = calcular\_area\_circulo(r);  cout << "El área del círculo es: " << area << endl;  return 0; } |
| **Función en Java** | import java.util.Scanner;  public class Main {  // Definición de la función  public static double calcularAreaCirculo(double radio) {  return 3.1416 \* radio \* radio;  }   public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.print("Ingrese el radio: ");  double r = sc.nextDouble();  double area = calcularAreaCirculo(r);  System.out.println("El área del círculo es: " + area);  } } |

Después de conocer los ejemplos anteriores, hay que tener en cuenta que las funciones no se comportan igual, es decir, algunas pueden devolver un resultado, otras reciben datos y otras simplemente realizan la tarea para la cual fueron creadas. Conocer los diferentes tipos de funciones, le permite a los programadores construir programas mejor organizados. A continuación la siguiente tabla resume los tipos de funciones que se pueden utilizar:

|  |  |  |
| --- | --- | --- |
| **Tarjetas** | | |
| Función sin parámetro y sin retorno | Solo ejecuta las instrucciones o la tarea encomendada. Ejemplo: mostrar un mensaje. | programación icono  [https://www.freepik.es/icono/programacion\_3749622 - fromView=search&page=2&position=50&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539](https://www.freepik.es/icono/programacion_3749622%20-%20fromView=search&page=2&position=50&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539) |
| Función con parámetros, pero sin retorno | Procesa los datos, pero no devuelve el valor o el resultado. Ejemplo: impresión de un reporte. | programación icono  [https://www.freepik.es/icono/programacion\_6328462 - fromView=search&page=2&position=61&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539](https://www.freepik.es/icono/programacion_6328462#fromView=search&page=2&position=61&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539) |
| Función sin parámetros, pero con retorno | Devuelve un valor fijo. Ejemplo: obtener la fecha actual. | codificación icono  [https://www.freepik.es/icono/codificacion\_2097285 - fromView=search&page=2&position=93&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539](https://www.freepik.es/icono/codificacion_2097285#fromView=search&page=2&position=93&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539) |
| Función con parámetros y con retorno | Procesa los datos y devuelve un valor o un resultado. Ejemplo: calcular el área cuadrada de un cultivo. | pruebas icono  [https://www.freepik.es/icono/pruebas\_4694494 - fromView=search&page=3&position=4&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539](https://www.freepik.es/icono/pruebas_4694494#fromView=search&page=3&position=4&uuid=577075fb-0fdb-4eaa-abea-4872c95b4539) |

Tal como se ha podido evidenciar, las funciones permiten organizar mejor el código de un programa y hacerlo más robusto con su reutilización. En el caso de la modularidad, en el siguiente cuadro se aprecia cómo se puede ver un programa organizado de esta forma:

|  |
| --- |
| **Ejemplo de código que muestra la modularidad en la programación estructurada** |
| def leer\_numero():  return float(input("Ingrese un número: "))  def sumar(a, b):  return a + b  def mostrar\_resultado(resultado):  print("La suma es:", resultado)  # Programa principal num1 = leer\_numero() num2 = leer\_numero() resultado = sumar(num1, num2) mostrar\_resultado(resultado) |

* 1. **Ejemplos**

En este apartado se comparten unos ejemplos de código generado en Python con ChatGPT y corriendo en Visual Studio Code. Lo primero que se hizo fue darle un *prompt* general para que supiera como iba a trabajar:

|  |
| --- |
| ***Prompt* inicial** |
| Necesito que actúes como un experto en programación, pero que tengas en cuenta los siguientes parámetros para trabajar:   * Lenguaje de programación a trabajar: Python * Paradigma de programación: programación estructurada * Recursos a utilizar para el desarrollo: funciones y estructuras de control. * Lógica de programación: lo más sencillo posible, ya que el público objetivo apenas está aprendiendo a programar. * Cuando estés listo me dices para irte dando los problemas a resolver por medio de los programas que vas a generar. |

Luego se le fueron dando cada uno de los *prompt* con el que generaría los códigos para los programas solicitados.

**Ejemplo 1: operación aritmética**

***Prompt*:** Necesito un programa que pregunte qué operación aritmética se desea realizar, solicite dos números, y con ellos haga la operación y muestre como resultado: la operación y el valor resultante de la operación.

|  |
| --- |
| **Código generado en Python con ChatGPT** |
| # Función para realizar la operación seleccionada  def calculadora\_simple():  print("Bienvenido a la calculadora básica.")  # Mostrar opciones  print("\nElige la operación que deseas realizar:")  print("1. Suma")  print("2. Resta")  print("3. Multiplicación")  print("4. División")  opcion = input("Ingresa el número de la opción elegida (1-4): ")  # Pedir los dos números  num1 = float(input("Ingresa el primer número: "))  num2 = float(input("Ingresa el segundo número: "))  # Realizar la operación y mostrar la operación completa con el resultado  if opcion == "1":  resultado = num1 + num2  print(f"{num1} + {num2} = {resultado}")  elif opcion == "2":  resultado = num1 - num2  print(f"{num1} - {num2} = {resultado}")  elif opcion == "3":  resultado = num1 \* num2  print(f"{num1} \* {num2} = {resultado}")  elif opcion == "4":  if num2 != 0:  resultado = num1 / num2  print(f"{num1} / {num2} = {resultado}")  else:  print("Error: no se puede dividir entre cero.")  else:  print("Opción no válida. Intenta de nuevo.")  # Llamar a la función principal  calculadora\_simple() |

**Figura 4.** Pantallazo del código en Visual Studio Code y ejecución en la terminal.

A computer screen shot of a black screen

AI-generated content may be incorrect.

**Ejemplo 2: calculadora**

***Prompt*:** Un estudiante al que se le dañó su calculadora, desea hacer un programa que le permita realizar las 4 operaciones aritméticas.

|  |
| --- |
| **Código generado en Python con ChatGPT** |
| # Función para sumar dos números  def sumar(a, b):  return a + b  # Función para restar dos números  def restar(a, b):  return a - b  # Función para multiplicar dos números  def multiplicar(a, b):  return a \* b  # Función para dividir dos números  def dividir(a, b):  if b != 0:  return a / b  else:  return "Error: no se puede dividir entre cero."  # Función principal para interactuar con el usuario  def calculadora():  print("Calculadora Básica")  print("Operaciones disponibles: suma, resta, multiplicación, división\n")  # Pedir los dos números  num1 = float(input("Ingrese el primer número: "))  num2 = float(input("Ingrese el segundo número: "))  # Pedir la operación a realizar  print("\nElige la operación a realizar:")  print("1. Suma")  print("2. Resta")  print("3. Multiplicación")  print("4. División")  opcion = input("Ingresa el número de la opción elegida (1-4): ")  # Realizar la operación seleccionada  if opcion == "1":  resultado = sumar(num1, num2)  print("El resultado de la suma es:", resultado)  elif opcion == "2":  resultado = restar(num1, num2)  print("El resultado de la resta es:", resultado)  elif opcion == "3":  resultado = multiplicar(num1, num2)  print("El resultado de la multiplicación es:", resultado)  elif opcion == "4":  resultado = dividir(num1, num2)  print("El resultado de la división es:", resultado)  else:  print("Opción no válida. Intenta de nuevo.")  # Llamar a la función principal  calculadora() |

**Figura 5. Pantallazo del código en Visual Studio Code y ejecución en la terminal.**

A screenshot of a computer program

AI-generated content may be incorrect.

**Ejemplo 3: lista ordenada**

***Prompt: J***uanito desea ayudar a su profesora, quien ha tomado los nombres de los 10 estudiantes que han ingresado por primera vez a su clase; para ello Juanito necesita crear un programa donde escriba los 10 nombres y se los entregue en orden alfabético ascendente.

|  |
| --- |
| **Código generado en Python con ChatGPT** |
| # Función para registrar y ordenar los nombres de los estudiantes  def ordenar\_nombres\_estudiantes():  # Lista para guardar los nombres  lista\_nombres = []  # Pedir los 10 nombres  for i in range(1, 11):  nombre = input(f"Ingrese el nombre del estudiante {i}: ")  lista\_nombres.append(nombre)  # Ordenar la lista alfabéticamente en orden ascendente  lista\_nombres.sort()  # Mostrar la lista ordenada  print("\n--- Lista de Estudiantes en Orden Alfabético ---")  for nombre in lista\_nombres:  print(nombre)  # Llamar a la función principal  ordenar\_nombres\_estudiantes() |

**Figura 6.** Pantallazo del código en Visual Studio Code y ejecución en la terminal.

A screenshot of a computer program

AI-generated content may be incorrect.

**Ejemplo 4: promedio de notas**

*Prompt*: Un estudiante está cursando 3 materias, cada materia es calificada con 3 notas, donde la nota 1 equivale al 30%, la nota 2 al 30% y la nota 3 al 40% del total del curso. Se necesita obtener el promedio por materia y el promedio general del estudiante.

|  |
| --- |
| **Código generado en Python con ChatGPT** |
| # Función para calcular el promedio ponderado de una materia  def promedio\_materia():  print("\nIngresa las notas de la materia:")  nota1 = float(input("Nota 1 (30%): "))  nota2 = float(input("Nota 2 (30%): "))  nota3 = float(input("Nota 3 (40%): "))  # Calcular el promedio ponderado  promedio = (nota1 \* 0.3) + (nota2 \* 0.3) + (nota3 \* 0.4)  return promedio  # Función principal para manejar el proceso  def promedio\_general():  total\_promedios = 0  # Calcular el promedio para las 3 materias  for i in range(1, 4):  print("\n--- Materia", i, "---")  promedio = promedio\_materia()  print("Promedio de la materia", i, ":", promedio)  total\_promedios += promedio  # Calcular el promedio general  promedio\_general = total\_promedios / 3  print("\n--- Promedio General del Estudiante ---")  print("Promedio general:", promedio\_general)  # Llamar a la función principal  promedio\_general() |

**Figura 7. Pantallazo del código en Visual Studio Code y ejecución en la terminal.**

A screenshot of a computer

AI-generated content may be incorrect.

**Ejemplo 5: venta de productos electrónicos**

***Prompt***: Ricardo está vendiendo dispositivos electrónicos. Carlos le ha hecho una compra y necesita hacerle la factura. Para hacer la factura necesita saber la cantidad de dispositivos vendidos, el valor del dispositivo vendido, el descuento por la venta, el impuesto por la venta y el total a pagar en la factura

|  |
| --- |
| **Código generado en Python con ChatGPT** |
| # Función para calcular la factura de la venta  def calcular\_factura():  # Pedir la cantidad de dispositivos vendidos  cantidad = int(input("Cantidad de dispositivos vendidos: "))  # Pedir el valor unitario de cada dispositivo  valor\_unitario = float(input("Valor de cada dispositivo: "))  # Calcular el subtotal  subtotal = cantidad \* valor\_unitario  # Pedir el descuento (en porcentaje)  descuento = float(input("Descuento en porcentaje (%): "))  # Calcular el monto de descuento  monto\_descuento = subtotal \* (descuento / 100)  # Calcular el subtotal después del descuento  subtotal\_con\_descuento = subtotal - monto\_descuento  # Pedir el impuesto (en porcentaje)  impuesto = float(input("Impuesto en porcentaje (%): "))  # Calcular el monto de impuesto  monto\_impuesto = subtotal\_con\_descuento \* (impuesto / 100)  # Calcular el total a pagar  total\_pagar = subtotal\_con\_descuento + monto\_impuesto  # Mostrar la factura  print("\n--- Factura de Compra ---")  print("Cantidad de dispositivos:", cantidad)  print("Valor unitario:", valor\_unitario)  print("Subtotal:", subtotal)  print("Descuento aplicado:", monto\_descuento)  print("Subtotal con descuento:", subtotal\_con\_descuento)  print("Impuesto aplicado:", monto\_impuesto)  print("Total a pagar:", total\_pagar)  # Llamar a la función principal  calcular\_factura() |

**Figura 8. Pantallazo del código en Visual Studio Code y ejecución en la terminal.**

A screenshot of a computer program

AI-generated content may be incorrect.

**Ejemplo 6: venta de leche**

***Prompt:*** en la finca La Costa, todos los días se está vendiendo la leche a la empresa la Pradera. El valor del litro de leche es $2.000. La leche se paga quincenal. Se necesita saber la cantidad de litros producidos desde el día 1 al día 15. Se necesita saber el total de leche vendida y el total de dinero recibido por venta de leche.

|  |
| --- |
| **Código generado en Python con ChatGPT** |
| # Función para calcular la venta de leche en 15 días  def venta\_leche\_quincenal():  # Precio del litro de leche  precio\_litro = 2000  # Inicializar el total de litros  total\_litros = 0  # Pedir la cantidad de litros de leche producidos cada día (día 1 al 15)  for dia in range(1, 16):  litros = float(input(f"Ingrese los litros de leche del día {dia}: "))  total\_litros += litros  # Calcular el total de dinero recibido  total\_dinero = total\_litros \* precio\_litro  # Mostrar los resultados  print("\n--- Resumen de Venta Quincenal ---")  print("Total de litros vendidos:", total\_litros)  print("Total de dinero recibido:", total\_dinero)  # Llamar a la función principal  venta\_leche\_quincenal() |

**Figura 9. Pantallazo del código en Visual Studio Code y ejecución en la terminal.**

A screenshot of a computer program

AI-generated content may be incorrect.

**Ejemplo 7: entrenamiento atlético**

***Prompt:*** un atleta aficionado se está preparando para la media maratón de la independencia realizada en Valledupar. El correrá en la categoría de 5 kilómetros. El récord histórico está en 25 minutos. Después de un entrenamiento riguroso, él está haciendo el recorrido por 5 días, en donde lleva una libreta y anota el tiempo que se toma en el recorrido de los 5 kilómetros, y quiere saber cuánto le falta por día para batir el récord histórico. También desea saber el promedio de tiempo que se ha tomado en el recorrido de los 5 días.

|  |
| --- |
| **Código generado en Python con ChatGPT** |
| # Función para calcular las diferencias con el récord y el promedio  def entrenamiento\_atleta():  # Récord histórico en minutos  record = 25  # Inicializar variables  total\_tiempo = 0  # Pedir los tiempos para cada uno de los 5 días  for dia in range(1, 6):  tiempo = float(input(f"Ingrese el tiempo del día {dia} (en minutos): "))  diferencia = tiempo - record  if diferencia > 0:  print(f"El atleta necesita mejorar {diferencia} minutos para igualar el récord.")  elif diferencia == 0:  print("¡El atleta igualó el récord!")  else:  print(f"¡El atleta superó el récord por {-diferencia} minutos!")  # Acumular el tiempo para el promedio  total\_tiempo += tiempo  # Calcular el promedio de tiempo  promedio = total\_tiempo / 5  # Mostrar el promedio  print("\n--- Resumen del Entrenamiento ---")  print("Promedio de tiempo en los 5 días:", promedio, "minutos")  # Llamar a la función principal  entrenamiento\_atleta() |

**Figura 10. Pantallazo del código en Visual Studio Code y ejecución en la terminal.**

A screenshot of a computer program

AI-generated content may be incorrect.

Como se ha podido verificar en los diferentes ejemplos, los códigos fueron generados con la versión gratuita de *ChatGPT* y probados en el editor de código *Visual Studio Code*. Hay que tener presente que para que el código pueda correr sin problemas, deben tener instalado Python y configurado el editor. Se hace la salvedad, porque algunos intentan correr el código de cualquier lenguaje, pero no tienen instaladas las extensiones o cualquier otro recurso tanto en el editor como en el equipo para que el código pueda ser ejecutado sin ningún problema.

|  |  |
| --- | --- |
| Concepto de nube ai con brazos robóticos  [https://www.freepik.es/foto-gratis/concepto-nube-ai-brazos-roboticos\_32471122.htm - fromView=search&page=1&position=20&uuid=901032bc-49d4-41d9-8eee-3714bec7a42f&query=importante++ia](https://www.freepik.es/foto-gratis/concepto-nube-ai-brazos-roboticos_32471122.htm%20-%20fromView=search&page=1&position=20&uuid=901032bc-49d4-41d9-8eee-3714bec7a42f&query=importante++ia) | **Recomendación:** para las personas que desean aprender o asimilar mejor sus conocimientos, pueden intentar desarrollar el programa o en su defecto generarlo en la IA gen y luego escribirlo a mano en una hoja, después digitarlo en el editor; esto se hace con el fin de ir estimulando su memoria muscular, lo cual le ayuda a irse familiarizando con el lenguaje de programación, ir estimulando la lógica, ir practicando y cuando menos lo piense podrá ir desenvolviéndose y codificando sus propios programas. |

En el siguiente video se explorarán ejemplos de *prompts* que permiten generar códigos en Python con ayuda de ChatGPT.

|  |
| --- |
| VIDEO EJEMPLO 1 |

En el siguiente video se pondrán a prueba los códigos generados con ChatGPT utilizando Visual Studio Code, con el propósito de verificar que funcionen según lo esperado.

|  |
| --- |
| VIDEO EJEMPLO 2 |

1. **SÍNTESIS**

A continuación, se presenta una síntesis de la temática estudiada en el componente formativo.

1. **ACTIVIDADES DIDÁCTICAS *(Se debe incorporar mínimo 1, máximo 2)***

|  |  |
| --- | --- |
| **DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA** | |
| **Nombre de la Actividad** | Reconocimiento de aspectos de inteligencia artificial generativa |
| **Objetivo de la actividad** | Fortalecer los conocimientos relacionados con inteligencia artificial generativa |
| **Tipo de actividad sugerida** | **Según catálogo de recursos educativos.**  **Para consultarlo, puede hacer clic** [**AQUÍ**](https://ecored-sena.github.io/ECOREDSENA_CATALOGO_WEB/#/curso/tema5) |
| **Archivo de la actividad**  **(Anexo donde se describe la actividad propuesta)** | **El ejercicio siempre debe tener realimentación positiva sobre las respuestas que seleccione el aprendiz…si queda mal o bien** |

1. **MATERIAL COMPLEMENTARIO:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| Editor de código Fuente | Moure, M. by B. [@mouredev]. (2023). Las 15 MEJORES EXTENSIONES para VSCODE. [Archivo de video] Youtube. | Vídeo | [**https://www.youtube.com/watch?v=lv88bCi7eyg**](https://www.youtube.com/watch?v=lv88bCi7eyg) |
| Editor de código Fuente | Programador X. (2023). Visual Studio Code: Tutorial (curso completo de VSCode). [Archivo de video] Youtube. | Vídeo | [**https://www.youtube.com/watch?v=FzRGSlhQIyY**](https://www.youtube.com/watch?v=FzRGSlhQIyY) |
| Editor de código Fuente | Visual Studio Code - code editing. Redefined. (2025). | Sitio web | [**https://code.visualstudio.com/**](https://code.visualstudio.com/) |
| Editor de código Fuente | Geeky Script [@GeekyScript]. (2025). Cómo instalar Visual Studio Code en Windows 10/11 [Actualización 2025]. [Archivo de video] Youtube. | Vídeo | [**https://www.youtube.com/watch?v=2Gz-uuQWxu4**](https://www.youtube.com/watch?v=2Gz-uuQWxu4) |
| Editor de código Fuente | Roelcode [@roelcode]. (2023). Configura Visual Studio Code para Python y crear proyectos | Paso a paso. [Archivo de video] Youtube. | Vídeo | [**https://www.youtube.com/watch?v=MVsNOtvwwlw**](https://www.youtube.com/watch?v=MVsNOtvwwlw) |
| Programación estructurada | Contreras, M. [@martincontreras9589]. (2022). Python- programación estructurada. [Archivo de video] Youtube. | Vídeo | [**http://www.youtube.com/playlist?list=PLcRgzfqsp7xpg1NEQ\_CZMl7bLdPQlBVFm**](http://www.youtube.com/playlist?list=PLcRgzfqsp7xpg1NEQ_CZMl7bLdPQlBVFm) |
| Programación estructurada | Huet, P. (2024). Fundamentos de Python: Sintaxis, variables y estructuras. Openwebinars.net. [Archivo de video] Youtube. | Sitio web | [**https://openwebinars.net/blog/fundamentos-de-python-sintaxis-variables-y-estructuras-de-control/**](https://openwebinars.net/blog/fundamentos-de-python-sintaxis-variables-y-estructuras-de-control/) |
| Programación estructurada | Dalto, L. [@soydalto]. (2025). Curso de PYTHON desde CERO (Completo). [Archivo de video] Youtube. | Vídeo | [**https://www.youtube.com/watch?v=nKPbfIU442g**](https://www.youtube.com/watch?v=nKPbfIU442g) |

1. **GLOSARIO:**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| Código fuente: | es el conjunto de instrucciones escritas por un programador en un lenguaje de programación legible por humanos. Representa la base de cualquier programa informático y debe ser interpretado o compilado para ejecutarse en una máquina. Su calidad influye directamente en el rendimiento, mantenimiento y seguridad del *software*. |
| Desarrollo de *software:* | es el proceso de diseñar, construir, probar y mantener programas informáticos. Involucra diferentes etapas como análisis, codificación, pruebas y documentación, y puede llevarse a cabo de manera individual o colaborativa. Es una actividad clave en la transformación digital y la automatización de procesos. |
| Editor de código fuente: | es una herramienta digital utilizada para escribir, editar y gestionar el código fuente de programas. Ofrece funciones como resaltado de sintaxis, autocompletado, terminal integrada y soporte para múltiples lenguajes. Facilita la productividad y organización del trabajo de los desarrolladores. |
| Extensiones: | son complementos que amplían las funcionalidades de un editor de código fuente. Permiten integrar nuevas herramientas como depuradores, asistentes de inteligencia artificial, soporte para nuevos lenguajes o gestores de versiones. Su uso personaliza y optimiza el entorno de desarrollo. |
| Estructuras de control: | son construcciones que permiten controlar el flujo de un programa mediante secuencia, selección e iteración. Facilitan la toma de decisiones, la repetición de acciones y la ejecución ordenada del código. |
| Funciones: | son bloques de código reutilizables que realizan tareas específicas. Pueden recibir datos (parámetros) y devolver resultados, facilitando la modularización y la organización del programa. |
| Interfaz: | es el conjunto de elementos visuales y de interacción que presenta una herramienta o *software* al usuario. En los editores de código, incluye menús, barras laterales, terminales y paneles que permiten navegar, configurar y escribir código de forma intuitiva. |
| Lenguaje de programación: | es un conjunto de reglas y sintaxis que los programadores usan para escribir instrucciones que la computadora pueda entender y ejecutar. Ejemplos incluyen *Python, Java, C++ y JavaScript*. |
| Modularidad: | es un principio que divide el programa en componentes independientes. Mejora la legibilidad, facilita el mantenimiento y permite la reutilización del código. |
| Multiplataforma: | hace referencia a la capacidad de un *software* para funcionar en diferentes sistemas operativos como *Windows*, *macOS* y *Linux* sin necesidad de modificaciones. Esta característica garantiza mayor compatibilidad, accesibilidad y flexibilidad para los usuarios. |
| Programación estructurada: | es un paradigma que organiza el código utilizando estructuras de control básicas y funciones. Mejora la claridad, la modularidad y la mantenibilidad del *software*. |
| Sintaxis: | es el conjunto de reglas que define la estructura correcta de las instrucciones en un lenguaje de programación. Una sintaxis incorrecta genera errores de compilación o ejecución. Los editores de código suelen ayudar resaltando errores de sintaxis en tiempo real. |
| Terminal: | es una interfaz de línea de comandos que permite ejecutar instrucciones directamente al sistema operativo o entorno de desarrollo. En editores como *Visual Studio Code*, la terminal integrada facilita la ejecución de scripts, comandos *Git* o compilaciones sin salir del entorno de trabajo. |
| Visual Studio Code: | es un editor de código fuente desarrollado por *Microsoft*, ampliamente utilizado por su velocidad, versatilidad y soporte para múltiples lenguajes. Incluye terminal integrada, extensiones, depuración, control de versiones y herramientas colaborativas. Es gratuito, multiplataforma y de código abierto. |

1. **REFERENCIAS BIBLIOGRÁFICAS**

Casero, A. (2023). ¿Qué es la modularidad en programación? *KeepCoding.io*. <https://keepcoding.io/blog/que-es-la-modularidad-en-programacion/>

Id digital school. (2024). *¿Qué es un editor de código?* Id digital school. <https://iddigitalschool.com/bootcamps/que-es-un-editor-de-codigo/>

Maldonado, R. (2024). Descubre la terminal de Visual Studio Code. *KeepCoding*. <https://keepcoding.io/blog/terminal-de-visual-studio-code/>

Microsoft. (2023). *Funciones (C++)*. Microsoft.com. <https://learn.microsoft.com/es-es/cpp/cpp/functions-cpp?view=msvc-170>

Stack Overflow. (2024). *Integrated development environment*. Stackoverflow.Co. <https://survey.stackoverflow.co/2024/technology#1-integrated-development-environment>

UNIR. (2022). *¿Qué es la programación estructurada?* Unir.net. <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/programacion-estructurada/>

UNIR. (2023). *5 editores de código indispensables para todo desarrollador*. Unir.net. <https://www.unir.net/revista/ingenieria/editores-codigo/>

Weisheim, R. (2022). *Los 17 mejores editores de código*. Hostinger; Tutoriales Hostinger. <https://www.hostinger.com/es/tutoriales/editores-de-codigo>

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| Autor (es) | Armando Javier López Sierra | Experto temático | Regional Tolima. Centro de Comercio y Servicios. | Mayo de 2025 |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| Autor (es) |  |  |  |  |  |