



MINISTERIO  
DE EDUCACIÓN

Programas de estudio de  
**Ciencias de la Computación**  
para Educación Media

**2024**

**José Mauricio Pineda Rodríguez**

Ministro de Educación, Ciencia y Tecnología

**Edgar Eliseo Alvarenga**

Viceministro de Educación y de Ciencia y Tecnología, *ad honorem*

**Wilfredo Alexander Granados Paz**

Director de Currículo y Materiales Educativos

**Gilberto Alexander Motto García**

Director de Educación en III Ciclo, Media y Tecnológica

**Gustavo Antonio Cerros Urrutia**

Gerente Curricular para el Diseño y Desarrollo de la Educación General

**Tonatiuh Eddie M. Orantes Ramos**

Jefe del Departamento de Ciencia y Tecnología

---

**Edición general**

Tonatiuh Eddie M. Orantes Ramos

Carolina Ruíz de Escobar

**Equipos técnicos**

Kevin Oswaldo Chávez Berríos

Tonatiuh Eddie M. Orantes Ramos

Jorge Alfredo Ávila Moreno

Luis Felipe Guevara García

Werner Francisco Romero Cañas

Edwin Adverdi Pérez Ventura

Óscar Mauricio Olmedo Martínez

**Corrección de textos**

Oswaldo Hernández Alas

**Diseño editorial y diagramación**

Patricia Damaris Rodríguez de Salmerón

ISBN EN TRÁMITE

## Estimado equipo docente

En el marco de la reforma educativa Mi Nueva Escuela, se hace entrega de esta primera versión de los programas de estudio de Ciencias de la Computación para Educación Media. Su contenido es coherente con los propósitos por cumplir desde el eje de Currículo Renovado, que es de carácter constructivista, humanista y vigente en el siglo XXI. Al mismo tiempo, profundiza en el desarrollo de competencias en innovación y desarrollo, pensamiento computacional y talento tecnológico mediante habilidades, procesos y actitudes que se conjuntan y estimulan en esta asignatura.

Como parte de las políticas educativas y de nación, se han incorporado contenidos relevantes y de vanguardia que ayuden a fortalecer la ciudadanía en respuesta a las demandas mundiales sobre habilidades para la resolución de problemas mediante el pensamiento lógico computacional, la capacidad de desarrollo tecnológico, la adaptación a nuevas tecnologías, el análisis y protección de datos, y una mentalidad innovadora con respeto, equidad y responsabilidad. Esto es posible si existen altas expectativas en la comunidad estudiantil y se les comunica que el esfuerzo y la constancia son vitales para lograr sus metas personales y su formación como ciudadanos del mundo.

Aprovechamos esta oportunidad para expresar la confianza en ustedes, que leerán y analizarán estos programas con una actitud dispuesta a aprender y mejorar, tomando en cuenta su experiencia y su formación docente. Su compromiso es sólido con la misión encomendada: alcanzar mejores aprendizajes en la niñez y la juventud salvadoreña para su desarrollo integral.

**José Mauricio Pineda Rodríguez**  
Ministro de Educación, Ciencia y Tecnología

<b>I. Introducción de los programas de estudio de Ciencias de la Computación para Educación Media .....</b>	<b>5</b>
1.1. Componentes curriculares .....	6
a. Competencias de unidad.....	6
b. Contenidos .....	6
c. Evaluación.....	8
1.2. Refuerzo académico .....	9
1.3. Estructura y descripción de una unidad de aprendizaje .....	10
<b>II. Plan de estudio de Ciencias de la Computación para Educación Media .....</b>	<b>12</b>
2.1. Malla curricular de Ciencias de la Computación para Educación Media .....	12
<b>III. Presentación de la asignatura .....</b>	<b>14</b>
3.1. Principio integrador .....	14
3.2. Objetivos .....	14
a. General .....	14
b. Específicos .....	15
3.3. Aproximación de la asignatura: IDI .....	15
3.4. Definición de los ámbitos IDI .....	15
3.5. Habilidades, procesos y actitudes.....	16
3.6. Características del currículo en Ciencias de la Computación .....	18
3.7. Perfiles de estudiante .....	18
3.8. Competencias de la asignatura .....	19
a. Competencia de pensamiento computacional .....	19
b. Competencia de talento tecnológico. ....	19
c. Competencia de innovación y desarrollo. ....	19
3.9. Competencias de grado.....	19
a. Primer año.....	19
b. Segundo año .....	20
3.10. Ejes integradores .....	20
<b>IV. Lineamientos metodológicos .....</b>	<b>23</b>
4.1 Experiencia directa: la práctica .....	23
4.2. Organización de las sesiones de clase .....	24

4.3 Secuencia didáctica de IDI .....	24
a. Etapa de inmersión.....	24
b. Etapa de desarrollo .....	24
c. Etapa de implementación.....	25
4.4. El estudiantado como protagonista .....	25
4.5. Modelaje docente.....	26
4.6. Atención a la diversidad .....	26

<b>V. Bases teóricas y lineamientos para la evaluación en Ciencias de la Computación.....</b>	<b>27</b>
5.1. La evaluación auténtica .....	27
5.2. Tipos de evaluación .....	28

<b>VI. Competencias y unidades didácticas .....</b>	<b>29</b>
Primer año .....	29
Competencias de grado .....	29
Unidades del programa de primer año de bachillerato .....	30
Segundo año .....	42
Competencias de grado .....	42

<b>VII. Referencias .....</b>	<b>43</b>
-------------------------------	-----------

## I. Introducción de los programas de estudio de Ciencias de la Computación para Educación Media

El desarrollo de los nuevos programas de estudio de Ciencias de la Computación para Educación Media pretende dar una respuesta curricular a las interrogantes que nacen desde la planificación docente:

Interrogantes	Componentes curriculares
¿Cuál es el propósito del aprendizaje?	Perfiles de estudiante Competencias
¿Qué debe aprender el estudiantado?	Contenidos Habilidades y procesos Actitudes Dominios clave
¿Cómo enseñar?	Orientaciones metodológicas Secuencia didáctica
¿Cómo, cuándo y qué evaluar?	Orientaciones sobre evaluación Indicadores de logro Indicadores avanzados Evidencias de aprendizaje

La propuesta de los programas de estudio está diseñada a partir de los componentes curriculares, que se desarrollan en el siguiente orden:

- Descripción de los componentes del programa, su interacción y papel en el logro de las competencias.
- Estructura general del plan de estudios de la asignatura para Educación Media.
- Presentación de la aproximación educativa que orienta el desarrollo de la asignatura en torno a habilidades, procesos y actitudes.

- Presentación de los ejes integradores que cohesionan los contenidos de la asignatura y ayudan a estructurar las unidades didácticas.
- Explicación de aspectos metodológicos relevantes para el desarrollo efectivo de las unidades, lo cual incluye justificaciones y recomendaciones específicas.
- Desarrollo de la evaluación, tomando en cuenta indicadores de logro, dominios clave, indicadores avanzados y criterios aplicables a las funciones de evaluación diagnóstica, formativa y sumativa.

Es importante destacar que los cambios programáticos obedecen a una transformación curricular completa, que abarca la concepción misma del **currículo como práctica** (Grundy, 1987; Smith, 2000).

De esta forma, el nuevo currículo en innovación y ciencias computacionales no prioriza la adquisición de conocimientos a través de contenido, lo que resultaría en una estructura más bien rígida; sino que considera a los contenidos como un medio dinámico; a través del cual cada estudiante, a su ritmo, podrá interactuar con sus semejantes y con el entorno mismo para adquirir gradualmente las habilidades, procesos y actitudes que son susceptibles de potenciar con el aprendizaje de las ciencias de la computación y que se deberán complementar con las demás áreas curriculares para alcanzar la integralidad (MINED, 2022).

La tercera sección introductoria a los programas de estudio expone con precisión las implicaciones más importantes del cambio curricular y su justificación.

## 1.1. Componentes curriculares

### a. Competencias de unidad

Las competencias se estructuran en función del logro de los aprendizajes, por lo que se orientan a la realización de una serie de acciones con el fin de obtener un desempeño concreto. Para la asignatura de Ciencias de la Computación, los desempeños se consolidan a partir de los perfiles de estudiante. En la enunciación de las competencias se distinguen, además, los saberes referidos a conceptos, procedimientos y actitudes como parte de los recursos de aprendizaje. También se evidencia el contexto y aplicación, el «para qué» o la finalidad de los aprendizajes, lo que le proporciona sentido y razón de ser a las acciones y los aprendizajes específicos.

### b. Contenidos

Los programas de estudio presentan los tres tipos de contenidos o recursos más importantes para el desarrollo de las competencias: los contenidos conceptuales (hechos, conceptos, principios o teorías), relacionados con el «saber»; los contenidos procedimentales (habilidades, procesos, técnicas, métodos y estrategias), conocidos como el «saber hacer»; y los contenidos actitudinales (actitudes, sentimientos, normas y valores), vinculados con «saber ser». En la planificación de aula es necesario que se tomen en cuenta con igualdad de importancia.

Los contenidos de Ciencias de la Computación se complementan, además, con los dominios clave, los cuales se relacionan con las ideas o principios que cada estudiante debería interiorizar con el desarrollo de los contenidos de una unidad particular.

#### b.1. Contenidos conceptuales

Los conocimientos configuran el saber. Un estudiante es com-

petente en este ámbito cuando los fundamentos y hechos que maneja son coherentes con la realidad. Si bien los campos de la ciencia y la tecnología son amplios y diversos, es posible encontrar conceptos unificadores que facilitan el estudio de dicha realidad. Es a partir de ellos que se han configurado **ejes integradores** como medios para cohesionar los contenidos conceptuales centrales de cada unidad de aprendizaje y año de estudio.

Los saberes de carácter conceptual ocupan un lugar muy importante en el aula y fuera de ella, pues son el motor que impulsa las acciones de la competencia, especialmente en torno a los demás recursos de aprendizaje, tales como habilidades, procedimientos y actitudes. Asimismo, el enfoque por competencias implica una trascendencia de lo conceptual al plano de los procedimientos, para que el saber no solo sea teórico sino significativo en la vida del estudiantado.

Es importante recalcar que los programas de estudio de Ciencias de la Computación no pormenorizan cada concepto, algoritmo o tecnología a abordar, ni intentan dotar de intencionalidad a la redacción de los contenidos conceptuales; en su lugar, proponen contenidos centrales y abiertos, en el entendido de que: 1) los contenidos conceptuales no son una lista de requisitos por cumplir, 2) el currículo no es una construcción rígida, sino que debe adecuarse al contexto del aula y de la comunidad, y 3) la competencia como principio organizador del currículo no significa introducir situaciones en los contenidos, sino que es una forma de trasladar la vida real al aula (*Jonnaert et al., 2008*).

Los subcontenidos conceptuales, así como su intencionalidad como motores de la competencia, se encuentran implícitos en los contenidos procedimentales.

## **b.2. Los contenidos procedimentales**

Este tipo de contenido tiene relación con las habilidades y procesos que se pretenden desarrollar en el estudiantado a fin de que sean capaces de enfrentarse con garantía de éxito ante una situación determinada. Las habilidades, por otra parte, también están inmersas en un contexto que las fortalece y en el cual se configuran e intervienen.

Para entender la relevancia que adquieren las habilidades y procesos dentro del nuevo currículo es indispensable contextualizar que la transformación curricular nacional de la reforma Mi Nueva Escuela (MNE) sucede dentro de una creciente ola de cambios en educación que tiene como principal detonante la transición global de economías dependientes de los recursos naturales a economías dependientes del conocimiento y la tecnología, exacerbado en el siglo XXI por la revolución industrial 4.0 (RI 4.0), que ha transformado rápidamente la demanda de habilidades para el trabajo y la vida (Hussin, 2018; Marope, 2017).

Los contenidos procedimentales de Ciencias de la Computación intentan plasmar la visión de cómo desarrollar escalonadamente estas nuevas habilidades y procesos dinámicos, impulsados por aquellos contenidos conceptuales que, dada su naturaleza, resultan afines a dichas habilidades y procesos. De cierta manera, los contenidos procedimentales también orientan la secuencia didáctica y dotan de intencionalidad a la unidad de aprendizaje, por lo que su papel es preponderante en la implementación del modelo educativo de la asignatura.

Los programas de estudio presentan los contenidos procedimentales enmarcados dentro de un contenido conceptual y numerados con un orden correlativo por cada unidad de aprendizaje.

Por ejemplo, el contenido procedimental 2.1 es el primero de la unidad 2.

## **b.3. Los contenidos actitudinales**

Las actitudes hacen referencia a las maneras habituales de reaccionar que tiene una persona. Son fruto de los conocimientos y las creencias. Con las actitudes las personas muestran lo que piensan de algo o alguien y sus emociones frente a una situación comunicativa ficcional o real. El ser se manifiesta cuando estos procesos son ejecutados.

Los contenidos actitudinales se construyen de manera transversal en todas las actividades de la unidad. La correspondencia entre los contenidos conceptuales, procedimentales e indicadores de logro tiene en su base otras habilidades básicas que están implícitas en las destrezas que los indicadores de logro evalúan. En este sentido, la labor docente en la planificación y secuenciación de los contenidos es muy importante, pues implica revisar con detalle cada procedimiento. Cabe destacar que el programa presenta una propuesta orientadora de la secuenciación, lo cual no significa que sea rígida. La flexibilidad curricular, siempre enfocada en la búsqueda por articular las competencias, es válida y necesaria.

## **b.4. Dominios clave**

Adicionalmente a los contenidos, los programas de estudio de Ciencias de la Computación retoman los dominios clave (MINED, 2022). Estos consisten en ideas o principios que deberían ser internalizados por cada estudiante, ya que tienen una vinculación directa con su vida cotidiana y sobre ellos se construirán posteriormente, saberes y procesos más complejos (OME, 2007), de tal manera que fortalecen la continuidad y gradualidad del currículo en su conjunto.

Cada dominio clave se construye a partir de los contenidos conceptuales y actitudinales comprendidos en una unidad de aprendizaje, resumiendo la estructura e intencionalidad de los mismos, por lo que pueden apoyar al docente para evaluar el dominio de los recursos de aprendizaje alcanzado por sus estudiantes durante una unidad específica.

### **c. Evaluación**

La propuesta curricular de la asignatura de Ciencias de la Computación se sustenta en la reforma educativa Mi Nueva Escuela (MNE) introduciendo la aproximación y secuencia de inmersión, desarrollo e implementación (IDI). La aproximación IDI se enmarca en el enfoque y la didáctica constructivista, en la que se articulan los diferentes momentos de la evaluación desde las competencias propuestas para cada unidad, ya que son las competencias y la claridad de su alcance las que permiten tener evidencias del desempeño del estudiantado. En lo específico, son los indicadores de logro los que permiten evaluar tal desempeño por medio de evidencias particulares.

Sobre los indicadores de logro se debe comprender que se pueden efectuar las adecuaciones que se consideren pertinentes. Sin embargo, modificar un indicador implica replantear los contenidos, las habilidades específicas e incluso la metodología empleada en la unidad de aprendizaje.

#### **c.1. Secuencia de los indicadores de logro**

Los programas de estudio presentan los indicadores de logro numerados con un orden correlativo por cada unidad de aprendizaje y están enlazados con los contenidos procedimentales, de forma tal que cada indicador de logro permite evidenciar el desempeño de un contenido procedimental específico. Por ejem-

plo, el indicador de logro 2.1 es el primer indicador de la unidad 2, y obedece al contenido procedimental 2.1, que es también el primero de la misma unidad. Ambos se enmarcan en un mismo contenido conceptual central.

Como ya se mencionó, en la asignatura de Ciencias de la Computación, los contenidos procedimentales desglosan la secuencia de abordaje de los contenidos conceptuales y detallan las habilidades y procesos que se espera potenciar en dicha secuencia; por lo tanto, los indicadores de logro secuenciados son susceptibles de emplearse para evidenciar la evolución de un aprendizaje de interés y para orientar las posibles adecuaciones metodológicas.

En una unidad de aprendizaje, las evidencias obtenidas por el conjunto de indicadores de logro, en contraste con el dominio clave, buscan reflejar el alcance de la competencia. Dado su carácter acumulativo, es de particular interés que cada docente plantee evaluaciones que le permitan determinar si sus estudiantes han interiorizado el dominio clave de la unidad.

#### **c.2. Indicadores avanzados**

Es muy probable que algunos estudiantes superen con facilidad los diferentes desafíos que le plantea el conjunto de contenidos abordados en una unidad de aprendizaje; o bien, que adquieran un interés particular que les motive a realizar un esfuerzo extra. Para ambos casos, los programas de estudio de Ciencias de la Computación retoman los indicadores avanzados (MINED, 2022). Como su nombre lo indica, su propósito es obtener evidencia de alto rendimiento en el estudiantado y ayudar a orientar la estrategia de atención, en el entendido de que señalan la vía de profundización del contenido o se enlazan directamente con el de años posteriores.

## 1.2. Refuerzo académico

El refuerzo académico resulta importante como estrategia de remediación planificada, ya que lograr las habilidades específicas y los aprendizajes prioritarios se vuelve imperativo. Para alcanzarlos con éxito, el refuerzo académico es la vía más idónea para estudiantes que enfrentan dificultades para consolidar sus conocimientos debido a condiciones cognitivas, sociales o circunstanciales, siempre que implique enfrentar a cada estudiante con otro grupo de situaciones de aprendizaje.

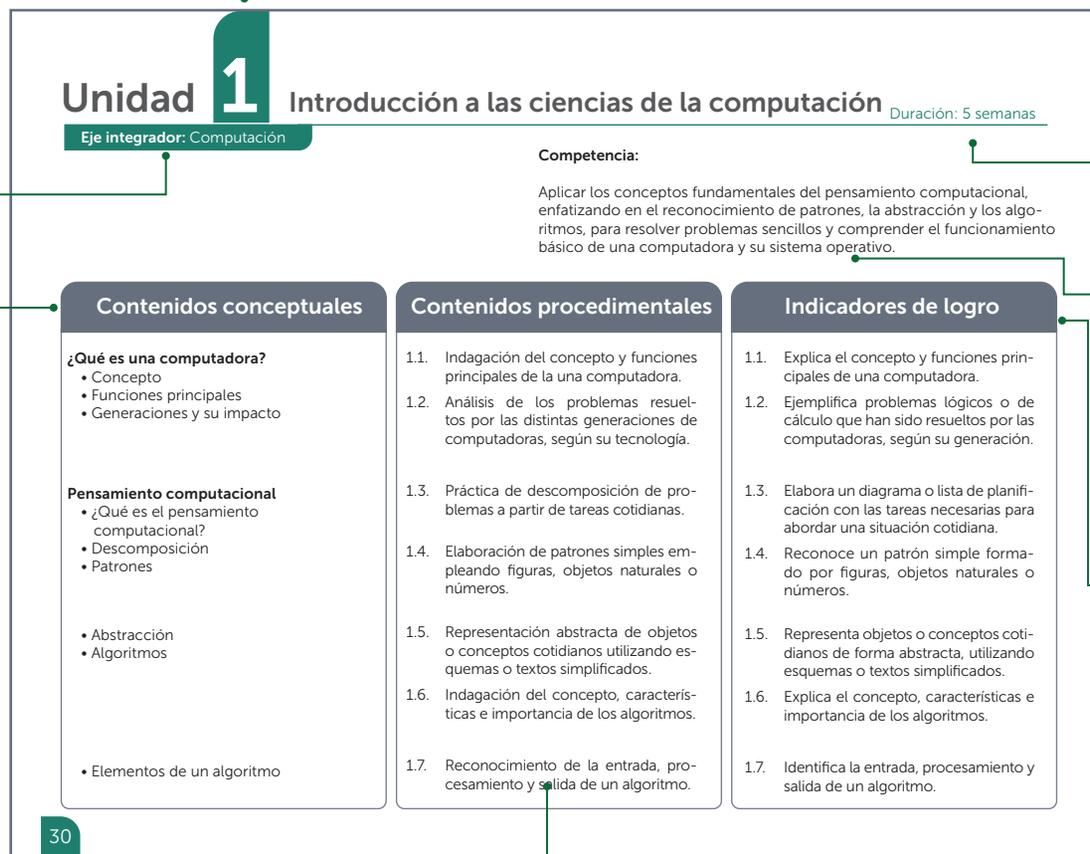
Para la asignatura de Ciencias de la Computación, se recomienda atender prioritariamente a aquellos estudiantes que no muestren claramente la apropiación de los dominios clave de las unidades de aprendizaje en cuestión.

### 1.3. Estructura y descripción de una unidad de aprendizaje

**Número y nombre de la unidad:** Describe los datos generales de la unidad

**Eje integrador:** Indica la idea clave en torno a la cual se pretenden conjuntar los contenidos de la unidad.

**Contenidos conceptuales:** Recogen la identidad y los atributos descriptivos de un objeto o sistema en estudio, así como los principios, modelos o teorías científicas que explican su comportamiento e interacciones. A menudo requieren de subcontenidos que se desarrollan de manera integrada con los contenidos procedimentales.



**Duración:** Cuantifica en semanas el tiempo previsto para desarrollar la unidad.

**Competencia de la unidad:** Enuncia lo que se espera que logre el alumnado.

**Indicadores de logro:** Reflejan las acciones y productos concretos que sirven como evidencias de aprendizaje. Presentan una numeración correlativa y coincidente con los contenidos procedimentales.

**Contenidos procedimentales:** Expresan las habilidades y los procesos físicos y mentales necesarios para alcanzar los indicadores de logro. Están numerados de forma correlativa para mostrar la secuencia de abordaje de los subcontenidos conceptuales, detallando las habilidades y los procesos a potenciar con ellos.

**Contenidos actitudinales:**  
Establecen con claridad las actitudes que se espera construir por medio de las situaciones de aprendizaje generadas en la unidad.

**Dominio clave:** Enuncia ideas o principios que deberían ser internalizados por cada estudiante, pues sobre ellos se construirán saberes más complejos.

**Indicadores avanzados:**  
Reflejan acciones o conductas concretas que sirven como evidencia de alto rendimiento en el estudiantado.

<ul style="list-style-type: none"> <li>• Representación de algoritmos</li> </ul>	
<b>Componentes de una computadora</b> <b>Sistemas operativos</b>	
1.8. Representación visual de algoritmos relacionados con situaciones cotidianas.  1.9. Clasificación de los componentes de una computadora como hardware y software.  1.10. Caracterización de los principales sistemas operativos.	1.8. Representa algoritmos empleando herramientas visuales como diagramas de flujo y pseudocódigo.  1.9. Desarrolla los conceptos de hardware y software brindando ejemplos.  1.10. Explica qué es un sistema operativo y su función en la computadora.
<b>Contenidos actitudinales</b>	<ul style="list-style-type: none"> <li>▪ Curiosidad por comprender el funcionamiento de las tecnologías digitales y sus aplicaciones.</li> <li>▪ Perseverancia para determinar patrones y resolver problemas lógicos siguiendo una estructura secuencial.</li> </ul>
<b>Dominio clave</b>	Las computadoras son herramientas multifuncionales que operan mediante la integración de hardware y software, mientras que el pensamiento computacional, al descomponer problemas y buscar patrones, nos permite abordarlos de manera lógica y creativa.
<b>Indicadores avanzados</b>	<ul style="list-style-type: none"> <li>▪ Relaciona las tecnologías de construcción de las computadoras de distinta generación con el tipo de problemas que han resuelto.</li> <li>▪ Utiliza un software para plantear un algoritmo simple.</li> </ul>

Primer año de bachillerato

31

**Grado:** Indica el año al que pertenece la unidad dentro del plan de estudios general.

**Notación:** Presenta indicaciones puntuales para homologar la introducción y presentación gradual de datos, definiciones y nomenclatura técnica científica involucrada en la unidad.

## II. Plan de estudio de Ciencias de la Computación para Educación Media

El plan de estudio para Educación Media se organiza en asignaturas con carga horaria definida. Ciencias de la Computación se desarrolla en cuatro horas por semana durante el año lectivo, que comprende 40 semanas.

Grado	Horas semanales	Horas anuales
Primer año	4	160
Segundo año	4	160

Para implementar el plan de estudio se deben hacer adecuaciones curriculares en función de las necesidades del estudiantado y de las condiciones del contexto.

### 2.1. Malla curricular de Ciencias de la Computación para Educación Media

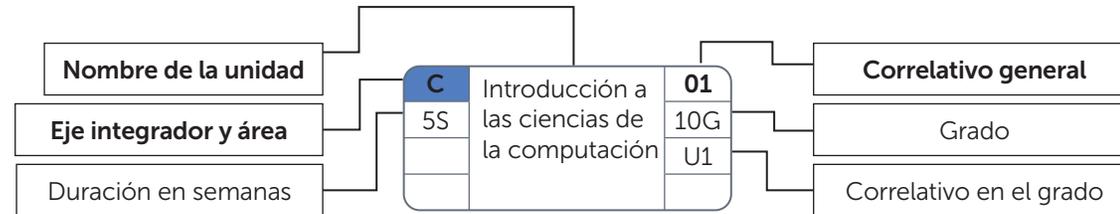
La asignatura de Ciencias de la Computación está diseñada para promover la adquisición de competencias en innovación, talento digital y pensamiento computacional durante la etapa comprendida entre los 16 y los 17 años; este proceso formativo de 2 años integra y expande las competencias promovidas durante la Educación Básica, principalmente en las asignaturas de Matemática y Datos y Ciencia y Tecnología. Esta etapa educativa también se caracteriza por requerir no solo la profundización en el conocimiento académico y el desarrollo de habilidades y actitudes especializadas, sino que, al mismo tiempo, se prepara a la persona para su probable vida laboral (MINED, 2022)

Para un desarrollo adecuado de las competencias planteadas es indispensable la construcción previa de habilidades fundamentales como la resolución de problemas, la búsqueda de información, la representación de datos numéricos y el ensamblaje de dispositivos siguiendo instrucciones. Estas habilidades deberán fomentarse en los años previos como parte del perfil de entrada a Educación Media. Para favorecer el desempeño durante la probable transición hacia la Educación Superior, los contenidos y las competencias de esta asignatura se construyen de forma acumulativa y articulada al perfil de salida del estudiante. Por otro lado, la adopción de los **ejes integradores** permite abordar diversos contenidos, ya sea de forma disciplinar o integrada, brindando coherencia entre las distintas asignaturas de la Educación Media.

A continuación se presenta la malla curricular de Ciencias de la Computación para Educación Media. En ambos años se abordan seis unidades de aprendizaje, y el tiempo efectivo de las clases se ajusta a 32 semanas, que incluye 8 semanas para las adecuaciones curriculares que se consideren pertinentes según las necesidades de cada institución educativa. Este período incluye el tiempo para planificación, evaluación y refuerzos académicos.

## Malla curricular de Ciencias de la Computación para Educación Media

Nomenclatura:



Leyenda:

Ejes integradores:

- C** – Computación
- A** – Aplicaciones
- D** – Datos
- S** – Sistemas

<b>C</b>	Introducción a las ciencias de la computación	<b>01</b>	<b>A</b>	Algoritmos avanzados	<b>07</b>
5S		10G	4S		11G
		U1			U1
<b>C</b>	Lógica de programación	<b>02</b>	<b>D</b>	Representación y análisis de datos	<b>08</b>
5S		10G	4S		11G
		U2			U2
<b>D</b>	Estructura de datos	<b>03</b>	<b>C</b>	Paradigmas y programación de software	<b>09</b>
5S		10G	7S		11G
		U3			U3
<b>A</b>	Desarrollo básico de software	<b>04</b>	<b>S</b>	Robótica y manufactura 3D	<b>10</b>
7S		10G	7S		11G
		U4			U4
<b>S</b>	Ciberseguridad y ciudadanía digital	<b>05</b>	<b>A</b>	Web 3.0	<b>11</b>
4S		10G	6S		11G
		U5			U5
<b>D</b>	Fundamentos de inteligencia artificial (IA)	<b>06</b>	<b>S</b>	Seguridad avanzada y protección de redes	<b>12</b>
6S		10G	4S		11G
		U6			U6

Primer año

Segundo año

### III. Presentación de la asignatura

La propuesta programática de Ciencias de la Computación está enmarcada en el currículo renovado de la reforma educativa MNE, centrado en el desarrollo del **potencial humano**, es decir, en la persona, donde cada estudiante es un actor propositivo de su aprendizaje, por lo que se promueven distintas prácticas pedagógicas efectivas, entre las que destaca el aprendizaje tecnológico y las habilidades relacionadas con la innovación.

Se trata de un giro en la concepción del currículo como un simple cuerpo de conocimientos que deben ser transmitidos y del intento de buscar fines generalizadores en el estudiantado hacia una aproximación mixta con sustento en el **currículo como práctica** (Smith, 2000), sin descuidar la adquisición equilibrada y gradual de conocimientos ni las habilidades y procesos propios de las tecnologías digitales, que son igualmente importantes para afrontar los desafíos multidimensionales y aprovechar las oportunidades específicas de cada país en la rápida ola de cambios que caracteriza al siglo XXI (Marope et al., 2017).

Por lo tanto, como ya se mencionó, el currículo en innovación y ciencias computacionales no prioriza la adquisición de conocimientos a través de contenidos, que resultaría en una estructura más bien rígida; en su lugar, se considera a los contenidos como un medio dinámico a través de los cuales cada estudiante, a su ritmo, podrá interactuar con sus semejantes y con el entorno mismo para adquirir gradualmente las habilidades, procesos y actitudes que son susceptibles de potenciar con el aprendizaje de las tecnologías digitales, y que se deberán complementar con las demás áreas curriculares para alcanzar la integralidad. En este sentido, se equipara en importancia la adquisición de habilidades, procesos y actitudes al dominio de conocimiento científico estructurado y a su aplicación en contexto (MINED, 2022).

A continuación se resumen los principales elementos de esta propuesta, según se plantea en los fundamentos curriculares de la asignatura.

#### 3.1. Principio integrador

Las tecnologías digitales son medios para desarrollar distintos tipos de pensamiento complejo, propiciar una mentalidad de innovación e incrementar la productividad. Su creciente diversidad e influencia está transformando todos los aspectos de la sociedad, siendo determinantes para abordar todo tipo de problemas, condicionar la forma en que nos relacionamos y adaptar los medios de producción en armonía con el ambiente.

Las ciencias de la computación son la base de la revolución digital y proporcionan herramientas y conocimientos necesarios para comprender, crear y transformar nuestro mundo.

Al cursar esta asignatura, los estudiantes desarrollarán habilidades de pensamiento crítico y lógico y de resolución de problemas y una mentalidad de innovación, lo que les permitirá asumir desafíos complejos y diseñar soluciones novedosas en un mundo cada vez más interconectado.

#### 3.2. Objetivos

##### a. General

- Preparar al estudiantado para ser ciudadanos digitales creativos y críticos, equipados con las habilidades técnicas y el pensamiento computacional necesarios para innovar, resolver problemas y contribuir al desarrollo de las nuevas tecnologías, adaptándose así a un mundo en constante evolución.

## b. Específicos

- Cultivar el pensamiento lógico computacional como medio para resolver problemas y enfrentar desafíos.
- Incentivar una mentalidad innovadora que permita idear y diseñar soluciones tecnológicas creativas que respondan a situaciones del mundo real.
- Promover la adquisición equilibrada y gradual de la teoría computacional y de las habilidades técnicas para dominar diversas herramientas digitales de vanguardia.

Para alcanzar los objetivos descritos, el nuevo diseño curricular retoma las habilidades, los procesos y las actitudes de la Educación 4.0, con énfasis en la innovación y el desarrollo, para el fomento de las competencias del siglo XXI.

### 3.3. Aproximación de la asignatura: IDI

La aproximación y secuencia IDI es el *marco conceptual de la asignatura que organiza el momento y la forma en que se promueven las habilidades, procesos y actitudes de la Educación 4.0, afines a las ciencias de la computación, durante la adquisición gradual, pertinente y práctica de conocimientos encaminados a la alfabetización tecnológica de la población, orientando así los perfiles de estudiante, articulando las competencias con momentos para la exploración y la reflexión y dotando de compatibilidad al currículo con diversas metodologías. Para ello se emplean tres ámbitos de habilidad* que funcionan como ejes de articulación y consolidación de los procesos, habilidades y actitudes fomentadas dentro de la secuencia didáctica de inmersión, desarrollo e implementación.

Al igual que una «metahabilidad» (SDS, 2018), *los ámbitos de habilidad se consideran atemporales, de construcción continua y de orden superior, al conjuntar habilidades, procesos y actitudes que*

*ayudan a formar estudiantes adaptables y facilitar el éxito en diferentes contextos que pueden suceder a futuro. No han sido seleccionados al azar, sino que se sustentan en aquellas habilidades consideradas comunes a la Educación 4.0 (Beers, 2011; Marope, 2017; MOE, 2014; SDS, 2018), que resultan de alta afinidad con las ciencias de la computación y que, por lo tanto, se perfilan como susceptibles de promover durante su aprendizaje práctico.*

### 3.4. Definición de los ámbitos IDI

**Inmersión.** *Es el ámbito relacionado con la capacidad de explorar el entorno y buscar información acerca de dónde y cómo se utilizan las tecnologías digitales, para así sumergirse en su mundo y adaptarse a los cambios rápidos con responsabilidad.*

En la aproximación IDI, donde se busca articular y enriquecer las competencias con momentos para la exploración y la reflexión, el ámbito de inmersión conjunta un dominio de habilidades y procesos relacionados principalmente con la exploración y la adaptabilidad, entre las que destacan la observación, el reconocimiento de problemas, la abstracción, la búsqueda de información y el autoaprendizaje. Al mismo tiempo, acuerpa el dominio de actitudes relacionadas con la curiosidad, la apertura mental y la asunción de desafíos.

**Desarrollo.** *Es el ámbito relacionado con la comprensión de la estructura, funcionamiento y producción de las tecnologías como medios para brindar soluciones eficientes y novedosas. Se trata del desarrollo del pensamiento lógico computacional y de la práctica manipulativa y creativa para generar experticia tecnológica y proponer innovaciones encaminadas a resolver problemas o aumentar la productividad.*

El ámbito de desarrollo es el eje central de la aproximación IDI,

pues conjunta un dominio de habilidades y procesos relacionados con el pensamiento computacional, la productividad y la innovación, entre los que destacan el razonamiento lógico y algorítmico, la gestión de datos, la programación, la resolución de problemas, la creatividad y el prototipado. Al mismo tiempo, acuerpa el dominio de actitudes relacionadas con la iniciativa, la atención al detalle, la perseverancia y la orientación a resultados.

**Implementación.** *Es el ámbito relacionado con la valoración y el pensamiento crítico sobre las experiencias con el uso o desarrollo de las tecnologías, así como el impacto que estas pueden tener en la sociedad, la economía y el ambiente.*

El ámbito de la implementación pretende conjuntar un dominio

de habilidades y procesos relacionados con la reflexión y la gestión, entre los que destacan el pensamiento crítico y sistémico, la representación y el análisis de datos, la organización, la simplificación y la síntesis. Al mismo tiempo, promueve un dominio de actitudes relacionadas con la objetividad, el respeto, la responsabilidad y la automejora.

### **3.5. Habilidades, procesos y actitudes**

Como ya se mencionó, la aproximación IDI orienta los momentos didácticos en los cuales resulta más favorable potenciar las distintas habilidades, procesos y actitudes afines a la educación 4.0 dentro de la asignatura. A continuación se presenta un cuadro resumen por cada ámbito de habilidad.

Resumen de las habilidades, procesos y actitudes afines a la Educación 4.0, que se promueven desde la asignatura de Ciencias de la Computación

INMERSIÓN		DESARROLLO			IMPLEMENTACIÓN	
Habilidades/Procesos	Actitudes	Habilidades/Procesos		Actitudes	Habilidades/Procesos	Actitudes
<b>Exploración y adaptabilidad</b>	Curiosidad	<b>Pensamiento computacional y productividad</b>	<b>Innovación</b>	Iniciativa	<b>Evaluación y gestión</b>	Ética
Observación	Motivación	Pensamiento computacional	Visualización	Colaboración	Evaluación	Responsabilidad
Indagación	Apertura mental	Razonamiento lógico y algorítmico	Creatividad	Atención al detalle	Pensamiento crítico	Automejora
Cuestionamiento	Asunción de desafíos	Programación	Resolución de problemas	Orientación a resultados	Pensamiento sistémico	Reflexión
Reconocimiento de problemas	Resiliencia	Desarrollo de software	Diseño de algoritmos	Asunción de errores	Representación y análisis de datos	Objetividad
Autoaprendizaje		Gestión de datos	Trabajo en equipo	Perseverancia	Gestión del tiempo	Empatía
Abstracción		Uso de herramientas de desarrollo	Pensamiento de diseño	Seguridad	Gestión de proyectos	
Aprendizaje continuo		Uso de sistemas operativos y redes	Deconstrucción	Respeto	Simplificación y síntesis	
Búsqueda y tamizaje de información		Experticia tecnológica y digital	Prototipado		Comunicación efectiva	
Concentración		Manipulación de dispositivos	Disruptividad		Perspectiva global	
Adaptación a nuevas tecnologías		Organización			Planificación	

Fuente: Elaboración propia.

### 3.6. Características del currículo en Ciencias de la Computación

Enmarcado en la reforma educativa MNE, el currículo de la asignatura Ciencias de la Computación muestra las siguientes características:

- **Centrado en el estudiante.** Se promueve la autonomía y la exploración por parte de los estudiantes.
- **Aprendizaje activo.** Los estudiantes aprenden haciendo, experimentando y resolviendo problemas.
- **Uso de herramientas tecnológicas.** Se utilizan distintas herramientas digitales y software para mediar el aprendizaje y lograr la experticia tecnológica y digital de los estudiantes.
- **Cultivo del pensamiento lógico computacional.** Se instaura un proceso de planteamiento y abordaje de problemas de manera que puedan ser expresados y resueltos por medio de un sistema computacional o bajo técnicas inspiradas en la informática.
- **Incentivo a la innovación y la mentalidad disruptiva.** Se proporcionan las herramientas y el conocimiento necesarios para desarrollar las habilidades que les permitirán enfrentar los desafíos del futuro con creatividad y confianza.
- **Profundización en las habilidades, los procesos y las actitudes de la educación 4.0.** Además del pensamiento computacional y la innovación, se fomentan habilidades y actitudes como la abstracción, el desarrollo de software, la deconstrucción, el prototipado, la colaboración, la empatía y el respeto.

### 3.7. Perfiles de estudiante

Un perfil de estudiante es el conjunto de características o cualidades que se desea apreciar en cada persona cuando culmine

un nivel educativo específico. Se debe hacer notar que muchas de ellas, al estar relacionadas con las habilidades 4.0, pretenden ayudar a largo plazo en la preparación de cada estudiante para su vida laboral o productiva, lo que es más enfático en la Educación Secundaria.

A continuación se presenta el perfil de estudiante para Educación Media. Se espera que, al finalizar su Educación Media, los estudiantes sean capaces de:

- Analizar dónde se utilizan y cómo influyen las tecnologías digitales en la vida cotidiana.
- Actuar como usuarios responsables, competentes y creativos de las tecnologías digitales.
- Aplicar el pensamiento lógico y computacional para enfrentarse a situaciones problema y proponer soluciones.
- Comprender y aplicar principios y conceptos fundamentales de la computación como ciencia, tales como la abstracción, la lógica, los algoritmos y el análisis de datos.
- Desarrollar programas informáticos mediante el uso de lenguajes de programación adecuados.
- Colaborar en equipo para llevar a cabo proyectos tecnológicos básicos.
- Adaptarse de forma crítica y proactiva a tecnologías nuevas o desconocidas.
- Argumentar sobre aspectos éticos, sociales, económicos y ambientales asociados al uso de las tecnologías digitales.
- Practicar el autocuidado y la responsabilidad ambiental con el uso y disposición final de dispositivos tecnológicos.
- Comprender y aplicar prácticas de ciberseguridad para proteger su información personal y navegar de forma segura por internet.

### 3.8. Competencias de la asignatura

Los principales procesos y habilidades que se pretende potenciar con Ciencias de la Computación se han empleado para diseñar las competencias de la asignatura. Estas deberán ser fomentadas en todo momento durante el desarrollo de la asignatura y en cada año de estudio; es decir, están en construcción constante y permanente

#### a. Competencia de pensamiento computacional.

Formular estrategias eficientes basadas en el reconocimiento de patrones, la abstracción o el análisis de datos, para abordar problemas complejos empleando secuencias de pasos lógicos.

Pretende fomentar habilidades y procesos como el razonamiento lógico y crítico, la abstracción, la evaluación y la automatización, así como el dominio teórico de conceptos y métodos propios de las ciencias computacionales. Además, la competencia incluye la promoción de actitudes transversales como la ética, la objetividad y la orientación a la calidad. Con estas herramientas, el estudiante podrá analizar dónde y cómo se utilizan las tecnologías digitales y aplicar pensamiento lógico computacional para enfrentar situaciones problemáticas.

#### b. Competencia de talento tecnológico.

Utilizar responsablemente y con criterio las herramientas tecnológicas apropiadas para diseñar, desarrollar o implementar soluciones efectivas a problemas.

Procura el uso informado y eficiente de las herramientas tecnológicas digitales, por lo que se centra en habilidades y procesos manipulativos como el uso de aplicaciones, la gestión de datos y la manipulación de dispositivos, así como los aspectos teóricos del funcionamiento de hardware y software. Además, la competencia

promueve actitudes como la responsabilidad, el autocuidado y la automejora. Con estas herramientas, se pretende que el estudiante no solo utilice diversas aplicaciones practicando medidas de ciberseguridad sino que pueda emplear sus fundamentos de programación para elaborar aplicaciones informáticas.

#### c. Competencia de innovación y desarrollo.

Generar ideas originales, prototipos o soluciones innovadoras para afrontar desafíos y oportunidades o adaptarse a las nuevas tecnologías.

Busca propiciar una mentalidad disruptiva, de creación y diseño de soluciones, para lo cual fomenta habilidades y procesos como la visualización, la creatividad, el prototipado y el trabajo colaborativo, además del dominio de estrategias y métodos de desarrollo tecnológico. También promueve actitudes como la iniciativa y la asunción de errores. Con estas herramientas se pretende que el estudiante proponga sus propias adecuaciones de procesos o dispositivos, que plantee proyectos colaborativos y que se adapte de forma crítica y proactiva a tecnologías nuevas o desconocidas.

### 3.9. Competencias de grado

En coherencia con el perfil del estudiante y las competencias generales de la asignatura, se han definido tres competencias por grado. A continuación se presentan sus enunciados, de acuerdo con la estructura propuesta por el Ministerio de Educación, Ciencia y Tecnología.

#### a. Primer año de bachillerato

**Pensamiento computacional.** Formular estrategias basadas en el reconocimiento de patrones y la abstracción, para resolver problemas simples empleando secuencias de pasos lógicos, estructuras de datos básicas y herramientas sencillas de programación.

**Talento tecnológico.** Utilizar responsablemente herramientas tecnológicas básicas para implementar soluciones sencillas a problemas, considerando la ciberseguridad y la lógica de programación.

**Innovación y desarrollo.** Experimentar con ideas originales o prototipos básicos para afrontar desafíos relevantes en su comunidad, utilizando herramientas de programación, hardware o elementos de inteligencia artificial.

### **b. Segundo año de bachillerato**

**Pensamiento computacional.** Formular estrategias eficientes basadas en la descomposición, el reconocimiento de patrones y el análisis de datos, para resolver problemas empleando algoritmos y códigos optimizados de acuerdo con su paradigma de programación.

**Talento tecnológico.** Utilizar con criterio herramientas tecnológicas apropiadas para diseñar y desarrollar soluciones efectivas a problemas, considerando aspectos de seguridad avanzada, protección de redes y desarrollo de software ajustado al hardware.

**Innovación y desarrollo.** Generar soluciones innovadoras o prototipos viables para abordar desafíos relevantes u oportunidades, aplicando conocimientos de programación, web 3.0, robótica y manufactura 3D.

## **3.10. Ejes integradores**

Como ya se mencionó, el nuevo currículo de innovación y ciencias computacionales se enfoca en el potencial de cada estudiante y busca fortalecer habilidades, procesos y actitudes. En la práctica, dichos componentes se activan a partir de los contenidos con-

ceptuales, generando así las unidades y sus competencias. Con el propósito de consolidar unidades de aprendizaje libres de la rigidez propia de un bloque de contenido disciplinar, capaces de brindar homogeneidad al abordaje de los saberes conceptuales a través de los distintos niveles educativos, la asignatura de Ciencias de la Tecnología hace uso de un sistema de ejes integradores.

*Los ejes integradores son conceptos que permiten cohesionar los contenidos de la asignatura en torno a una idea clave y a sus preguntas generadoras o de reflexión. Funcionan como hilo conductor y narrativa para desarrollar cualquiera de las temáticas relativas a ciencias de la computación. Además, están pensados para facilitar la articulación entre asignaturas, en búsqueda de un currículo holístico que promueva la formación de ciudadanos integrales. (MINED, 2022).*

Debido a su carácter catalizador de las distintas áreas de las ciencias de la computación, se han definido cuatro conceptos que cumplen las características antes descritas, que se convierten en los cuatro ejes integradores. A continuación se define cada uno de ellos junto a sus propiedades de cohesión y se presentan algunos ejemplos de integración.

Eje integrador	Definición y propiedades de cohesión	Ejemplos de integración
<b>Computación</b>	Entendida como el proceso de utilizar y manipular información a través de sistemas automáticos y la tecnología desarrollada para tal fin, es el sustento del pensamiento computacional y la puerta de entrada para aprender sobre una amplia gama de tecnologías. Desde la IA hasta la robótica, tienen su fundamento en la computación. Si bien el mundo tecnológico cambia rápidamente, estos principios siguen siendo subyacentes y relevantes.	<p><b>Hardware.</b> La teoría de la computación establece los límites de lo que una máquina puede hacer. Ayuda a diseñar circuitos más eficientes, a entender cómo almacenar y procesar información a nivel de hardware y, en general, a optimizar el rendimiento de los dispositivos.</p> <p><b>Aplicaciones.</b> La teoría de algoritmos es fundamental para el desarrollo de software. Permite diseñar algoritmos eficientes para resolver problemas, desde ordenar listas hasta encontrar la ruta más corta en un mapa. La complejidad algorítmica nos ayuda a evaluar la eficiencia de diferentes soluciones y a elegir la mejor opción para cada problema.</p> <p><b>Redes.</b> La teoría de la computación también juega un papel importante en el diseño de redes de computadoras. Nos ayuda a entender cómo transmitir información de manera eficiente y segura, a diseñar protocolos de comunicación y a resolver problemas como la congestión de la red.</p>
<b>Sistemas</b>	Los sistemas son conjuntos de componentes interrelacionados que trabajan juntos para lograr un objetivo común. Estos componentes pueden ser tanto físicos (hardware) como lógicos (software). En general, cualquier aplicación, método o dispositivo, como una computadora, puede abordarse como un sistema; de esta forma, un sistema puede ser tan simple como una calculadora o tan complejo como el internet.	<p><b>Hardware y software.</b> A menudo se ven como partes interdependientes, pero también pueden verse como un sistema. El hardware proporciona la plataforma física sobre la que se ejecuta el software, y el software le da instrucciones al hardware.</p> <p><b>Algoritmos y estructuras de datos.</b> Los algoritmos son las instrucciones que manipulan los datos, y las estructuras de datos son las formas en que se organizan esos datos. Ambos son componentes esenciales de cualquier sistema.</p> <p>Sistemas operativos y aplicaciones. El sistema operativo es el software que gestiona los recursos del hardware y proporciona una interfaz para que las aplicaciones puedan ejecutarse.</p> <p><b>Redes.</b> Las redes conectan múltiples sistemas, permitiendo la comunicación y el intercambio de datos. Las redes neuronales funcionan en sí como un sistema de procesamiento de información.</p> <p><b>Bases de datos.</b> Almacenan y organizan sistemáticamente grandes cantidades de datos que son utilizados por diferentes aplicaciones.</p>

Eje integrador	Definición y propiedades de cohesión	Ejemplos de integración
<b>Datos</b>	<p>Los datos son la representación digital de hechos, conceptos o instrucciones; se trata de la materia prima con la que trabajan los computadores. A través del análisis y procesamiento de datos, se puede extraer conocimiento valioso y tomar decisiones informadas. Los programas interactúan con datos de entrada y producen datos de salida; por ejemplo, la inteligencia artificial (IA) se basa en grandes cantidades de datos para aprender y realizar tareas complejas.</p>	<p><b>Algoritmos.</b> Los algoritmos son conjuntos de instrucciones diseñadas para procesar datos y obtener resultados específicos.</p> <p><b>Estructuras de datos.</b> Organizan los datos de manera eficiente para facilitar su acceso y manipulación.</p> <p><b>Bases de datos.</b> Almacenan y gestionan grandes volúmenes de datos.</p> <p><b>Redes.</b> Transmiten datos entre diferentes dispositivos.</p> <p><b>Seguridad informática.</b> Protege los datos de accesos no autorizados.</p> <p><b>Visualización de datos.</b> Presenta los datos de forma gráfica para facilitar su comprensión.</p>
<b>Aplicaciones</b>	<p>Pueden entenderse como programas informáticos diseñados para ejecutar un conjunto específico de funciones, proporcionando una interfaz que permite a los usuarios interactuar con el sistema operativo y con otros programas. Si bien las aplicaciones parecen existir en un mundo digital abstracto, están profundamente arraigadas en el hardware. Cada línea de código se traduce en instrucciones que el procesador ejecuta y cada dato que se procesa se almacena en algún componente físico.</p>	<p><b>Desarrollo de software <i>front-end</i>.</b> Las aplicaciones con interfaces gráficas interactivas impulsan el desarrollo de tecnologías web y móviles.</p> <p><b>Desarrollo de software <i>back-end</i>.</b> Las aplicaciones requieren de servidores y bases de datos para almacenar y procesar información, lo que a su vez impulsa el desarrollo de lenguajes de programación del lado del servidor.</p> <p><b>Aprendizaje automático.</b> Las aplicaciones utilizan algoritmos de aprendizaje automático para realizar tareas como reconocimiento de imágenes, procesamiento de lenguaje natural y recomendación de productos.</p> <p><b>Minería de datos.</b> Las aplicaciones pueden utilizar algoritmos de minería de datos para descubrir patrones y relaciones ocultas en los datos.</p> <p><b>Comunicación.</b> Las aplicaciones utilizan protocolos de red para comunicarse entre sí y con otros dispositivos.</p>

## IV. Lineamientos metodológicos

La asignatura de Ciencias de la Computación presenta un enfoque constructivista, centrado en la **resolución de problemas** por medio del **pensamiento computacional** y la **innovación**. Como resultado, la aproximación IDI permite la inclusión de distintas técnicas didácticas que impulsen al estudiante a usar su imaginación y las herramientas tecnológicas digitales para proponer y evaluar ideas que le permitan afrontar desafíos y resolver problemas relacionados prioritariamente con su ámbito cotidiano.

El desarrollo de los pensamientos lógico, algorítmico, computacional y crítico, así como la construcción de una mentalidad de innovación, no depende directamente del acceso a las tecnologías digitales, sino de poner en práctica los procesos subyacentes ante distintas situaciones de forma cotidiana. Asimismo, la naturaleza cambiante de las tecnologías y su capacidad para generar disrupción en distintos campos vuelve poco relevante centrar el aprendizaje en el uso de las aplicaciones particulares. Por estas razones, la propuesta metodológica de la asignatura se basa en generar una cultura escolar y del aula que promueva el planteamiento de situaciones problema que puedan resolverse de forma lógica y secuencial, incentivando el diálogo respetuoso, las preguntas abiertas y una disposición a la sorpresa. Se estimula la curiosidad, se aceptan los desafíos y distintas soluciones son consideradas una valiosa fuente de investigación, exploración y colaboración.

Por otro lado, para generar la experticia tecnológica y digital, así como las habilidades de gestión y evaluación, la asignatura propone experiencias educativas prácticas; no obstante, la diversidad de tecnologías disponibles, así como el contexto de cada centro educativo, impide establecer una metodología única para guiar su desarrollo, por lo que se proponen distintas estrategias com-

patibles. Entre las metodologías y estrategias didácticas compatibles con la aproximación IDI para conducir las experiencias de aprendizaje, se pueden destacar:

- Resolución de problemas.
- Proyectos educativos.
- Aprendizaje mediado.
- Gamificación.
- Investigación.
- Estudio de caso.
- Comprobación/Demostración.
- Aprendizaje basado en tecnologías.

### 4.1 Experiencia directa: la práctica

Para acercar las tecnologías y la mentalidad innovadora a cada estudiante, la secuencia IDI se vale principalmente de sus etapas de inmersión y desarrollo. De hecho, sumergirse en el mundo tecnológico implica que el estudiantado experimente de primera mano el funcionamiento e impacto cotidiano de las tecnologías, mientras que el desarrollo supone manipular, deconstruir, proponer ideas y prototipar. Existen dos razones para fomentar la práctica: 1) la experiencia es la llave que abre el entendimiento de los conceptos y, 2) un estudiante está continuamente utilizando sus experiencias para construir el entendimiento del mundo que lo rodea, sea científicamente correcto o no (Worth et al., 2009).

En consecuencia, la asignatura de Ciencias de la Computación está pensada para ser conducida de forma vivencial a través de las experiencias de aprendizaje, que pueden ser vistas como pequeños ejercicios de innovación y desarrollo tecnológico adaptados al aula.

## 4.2. Organización de las sesiones de clase

La asignatura de Ciencias de la Computación se organiza en semanas de 4 horas clase. En cada sesión de clases, en principio, deben desarrollarse las tres etapas de la secuencia IDI, lo que incluye al menos un momento de práctica. Debido a esto, se sugiere organizar las clases en bloques de dos horas, lo que permitirá usar el tiempo de forma más efectiva, acomodando momentos para la exploración, la práctica y la reflexión.

## 4.3 Secuencia didáctica de IDI

Para conjuntar las sesiones de clase con el desarrollo de las habilidades, procesos y actitudes, se plantea la secuencia didáctica IDI, que comprende tres etapas:

### a. Etapa de inmersión

Pretende desarrollar las habilidades, procesos y actitudes del ámbito de la inmersión, específicamente aquellas relacionadas con la exploración y la adaptabilidad. Esta etapa se centra en observar e investigar para descubrir dónde se encuentran y cómo funcionan las tecnologías y sumergirse en su mundo. Es decir, se pretende que cada estudiante sea capaz de abstraer y analizar dónde se utilizan y cómo influyen las tecnologías digitales en la vida cotidiana, lo que ayudará a deducir su funcionamiento técnico y, en última instancia, a adaptarse de forma crítica y proactiva a ellas, especialmente si se trata de tecnologías nuevas o desconocidas. Por otro lado, cuando el contenido no es nuevo, la etapa de inmersión también involucra la activación de presaberes.

Dado que la etapa de inmersión a menudo involucra la discusión a raíz de preguntas generadoras y puede llegar a ser bastante práctica, se recomienda centrarse en una actividad puntual de fácil conclusión. Además, la actividad de inmersión debería

arrancar sin un preámbulo teórico que conduzca al estudiantado a una respuesta predeterminada. De hecho, se considera que no existen respuestas o conclusiones erróneas en esta etapa, por lo que es importante no coartar opiniones ni ideas preconcebidas, incluso si carecen de asidero técnico (MINED, 2022). Se debe fomentar las repreguntas, el contraste de opiniones y el uso del pensamiento lógico entre el estudiantado.

A continuación se presentan ejemplos de actividades que permiten cumplir las características de la etapa de inmersión:

- Manipulación de aplicaciones o dispositivos.
- Demostraciones (por el docente).
- Lecturas.
- Recursos audiovisuales.
- Lluvias de ideas.
- Análisis de datos.
- Juegos (actividad gamificada).

### b. Etapa de desarrollo

Busca estimular las habilidades, procesos y actitudes del ámbito del desarrollo, específicamente aquellas relacionadas con el pensamiento computacional, la productividad y la innovación. Esta etapa se centra, por un lado, en determinar el funcionamiento específico de las tecnologías digitales y, por otro lado, en practicar y generar experticia tecnológica o digital para proponer adaptaciones o generar proyectos innovadores propios. Incluye construir, programar, crear soluciones, proponer y perfeccionar ideas o prototipos.

La etapa de desarrollo es el corazón de la experiencia educativa dentro de la aproximación IDI; en consecuencia, debería ser la parte más extensa de la clase. Típicamente, incluye una actividad central, pero también puede abordarse como una secuencia de

actividades menores, según el objetivo y la temática. Ocasionalmente, puede optarse por dividir al salón de clase en equipos de trabajo que efectúen variantes experimentales, prototipos o actividades particulares, según los mismos requerimientos del estudiantado.

Cualquiera que sea el foco de la actividad de desarrollo, es indispensable que toda la clase tenga claridad del objetivo o producto final y lleve un registro del proceso. Asimismo, dar rienda suelta a la creatividad y propiciar la innovación implica que el estudiante pueda atreverse a cuestionar su propio procedimiento, realizar adecuaciones y cometer errores, siempre que no altere el objetivo planteado inicialmente.

A continuación se presentan ejemplos de actividades que permiten cumplir las características de la etapa de desarrollo:

- Resolución de problemas.
- Desarrollo de un juego (gamificación).
- Construcción de un modelo o prototipo.
- Diseño de un algoritmo o programación de software
- Simulaciones o experimentos (individual o grupal).
- Recuperación y tratamiento de datos.
- Práctica con un software o un hardware.

### **c. Etapa de implementación**

Tiene por objetivo facilitar el desarrollo de habilidades, procesos y actitudes del ámbito de la implementación, específicamente aquellas relacionadas con la evaluación y la gestión. Esta etapa se centra en valorar y pensar críticamente sobre la experiencia con el uso o desarrollo de las tecnologías, así como el impacto en la sociedad, la economía y el ambiente.

Cuando se ha desarrollado un proyecto práctico, es importante

también brindar el espacio para presentarlo y demostrar su funcionamiento y utilidad, lo cual a menudo se acompaña con la representación y análisis de datos. También es de vital importancia reflexionar sobre los obstáculos superados durante el proceso de desarrollo y los puntos de mejora. En este sentido, la etapa de implementación fortalece la síntesis, la planificación y la comunicación efectiva y objetiva.

A continuación se presentan ejemplos de actividades que permiten cumplir las características de la etapa de desarrollo:

- Plenarias.
- Debates.
- Exposiciones.
- Demostraciones (por los estudiantes).
- Ferias de logro.

### **4.4. El estudiantado como protagonista**

El desarrollo de las competencias en innovación y ciencias computacionales, así como la adopción de las metodologías para el aprendizaje activo, conllevan a la adopción de un rol protagonista en el estudiantado, quien debe realizar una serie de actividades prácticas que aseguren la consecución de los indicadores de logro propuestos. Para que un estudiante pueda ser el protagonista de su aprendizaje e involucrarse en una experiencia educativa es vital que comprenda la cuestión o el problema en el que está trabajando y que este sea significativo para él. Una buena forma de lograrlo es propiciar su involucramiento para determinar la pregunta o problema de interés (Worth et al., 2009).

La habilidad de formular preguntas, la capacidad de observación y de seguir una secuencia de instrucciones son aspectos clave para las ciencias en general. En este sentido, el docente debe propiciar

con su ejemplo el uso de la pregunta correcta para cada situación y brindar indicaciones muy precisas sobre qué debería observar o hacer el estudiantado en cada momento, lo que ayuda a dejar en claro el objetivo de la actividad y brinda pistas sobre el proceso para conseguirlo.

Antes se mencionó que es recomendable usar las experiencias previas e ideas preconcebidas del estudiantado para conducir la etapa de inmersión. Además, se debe evitar responder inmediatamente a sus interrogantes sobre el tema para que pueda propiciarse una mentalidad de innovación. Una vez que un estudiante se haya centrado en la pregunta o problema, es el momento de estimular su involucramiento en la búsqueda de soluciones, pidiéndole su opinión o conjetura y la contrastación con las de sus compañeros. El uso de contraejemplos y de contextos situacionales para probar conjeturas también son estrategias recomendables.

#### 4.5. Modelaje docente

Se recomienda al equipo docente realizar el modelaje pertinente de las estrategias y actividades que se utilizan por primera vez con el estudiantado, con el fin de evitar ambigüedades y procesos innecesarios que los alejen de los indicadores de logro propuestos; esto es especialmente importante con la manipulación de dispositivos y el desarrollo de prototipos.

En adición, es indispensable para el equipo docente modelar en todo momento el respeto a las normas establecidas para:

- Aportar ideas o comentarios.
- Organizar el espacio de trabajo.
- Manipular dispositivos y aplicaciones.
- Reducir y tratar el error.
- Descartar residuos.

#### 4.6. Atención a la diversidad

Hay que recordar que cada estudiante aprende de diferente manera y a distintos ritmos, lo que obliga al equipo docente a implementar diversas estrategias didácticas con el fin de dar una respuesta que permita atender y desarrollar de forma adecuada el potencial de cada estudiante.

La diversidad también incluye el uso diferente de la lengua. En un aula puede haber estudiantes de origen rural y urbano, así como de diferente estrato social y económico; sin embargo, el lenguaje y la simbología son aspectos cruciales para el logro de las competencias de la asignatura.

Durante la etapa de inmersión es importante atender la formulación apropiada de preguntas y la discusión. En la etapa de desarrollo se debe tener cuidado en estimular el pensamiento computacional, la manipulación y la creatividad, mientras que en la etapa de implementación se debe prestar especial atención a la reflexión y a la evaluación del trabajo propio.

Finalmente, es indispensable para el adecuado desarrollo de la asignatura que se normalice el cometimiento de errores, los cuales no deben ser castigados, sino asumidos como parte inherente de la innovación y el desarrollo, por lo que solo pueden reducirse y tratarse. Existen tres razones de peso detrás de este principio: 1) las tecnologías, y cualquier las innovaciones, son falibles y perfectibles; 2) la asunción de errores es una actitud deseable en distintos contextos de la vida laboral y personal, y 3) castigar errores coarta la creatividad y genera barreras entre los estudiantes.

## V. Bases teóricas y lineamientos para la evaluación en Ciencias de la Computación

Es necesario retomar lo establecido en el manual *Evaluación al servicio del aprendizaje y del desarrollo*, en donde se explicitan los métodos, las técnicas y las normas de evaluación en su carácter diagnóstico, formativo y sumativo (MINED, 2015).

El nuevo currículo en innovación y ciencias computacionales se sustenta en el enfoque por competencias y en la aproximación IDI. Este paradigma práctico exige una evaluación a través de situaciones prácticas concretas y simuladas, pero en contextos cercanos y reales (escolar, familiar y comunitario). Estas experiencias de aprendizaje deben buscar que el estudiantado aplique los aprendizajes adquiridos y evidencie los desempeños (ser, hacer, conocer y convivir) de las competencias.

### 5.1. La evaluación auténtica

Este tipo de evaluación otorga relevancia a las tareas o actividades basadas en el «saber hacer» (proceso) en situaciones prácticas de innovación o desarrollo tecnológico que ocurren en el aula, o en cualquier otro espacio escolar, para construir significado. Por ejemplo, resolver un problema, programar software, representar datos, construir un modelo, efectuar una simulación por computadora, elaborar un prototipo, entre otras actividades. La evaluación auténtica se centra en un estudiante real, considera sus diferencias, lo ubica en su propio contexto y lo enfrenta a situaciones significativas y complejas de aprendizajes, tanto a nivel individual como grupal (Condemarín & Medina, 2000).

Principios de la evaluación auténtica:

- Es una instancia destinada a mejorar la calidad de los aprendizajes.

- Constituye parte integral de la enseñanza.
- Evalúa competencias dentro de contextos significativos.
- Se realiza a partir de situaciones problemáticas.
- Se centra en las fortalezas del estudiantado.
- Constituye un proceso colaborativo.
- Diferencia evaluación de calificación.
- Constituye un proceso multidimensional.
- Utiliza el error como ocasión de aprendizaje.

Una de las críticas en el sistema educativo tradicional es que la evaluación que se realiza no es congruente con lo que se fomenta aprender. A partir de esto y de una idea del constructivismo, que plantea que no debe haber una ruptura ni un desfase entre los episodios de aprendizaje y los de evaluación, se establece que los criterios de evaluación deben determinarse a partir de los indicadores de logro definidos en la transformación curricular realizada como evidencias de los aprendizajes prioritarios.

Los criterios de evaluación son el término más usado en la evaluación por competencias para dar cuenta de las pautas que deben considerarse al evaluar un producto o evidencia de semana, unidad, o período, pues buscan considerar los diferentes saberes de la competencia (ser, hacer, conocer y convivir) y el desarrollo integral de un proceso transformador del estudiantado (Tobón et al., 2010).

Es importante que el personal docente utilice estrategias e instrumentos para diferentes tipos de evaluación. Estos deben reunir los criterios, con base en los indicadores de logro, de manera que se realice una evaluación integral. Se sugiere el uso de rúbricas, listas de verificación o de cotejo, matrices de valoración, el registro anecdótico o un diario de clases, el portafolio de tareas, los proyectos, los prototipos, entre otros.

## 5.2. Tipos de evaluación

**Evaluación diagnóstica.** Un docente de Educación Media, por medio de diversas técnicas, como la formulación de preguntas exploratorias, de discusiones, de explicaciones previas y del seguimiento en la resolución de tareas o problemas iniciales, conoce el saber y las habilidades que posee el estudiantado al inicio del año escolar, de una semana o de una unidad didáctica.

Para ello, se presentan tareas en función de las necesidades y desempeños esperados de sus estudiantes.

**Evaluación formativa.** La evaluación formativa es continua y tiene como punto de partida los aprendizajes esperados de acuerdo con los indicadores de logro y las evidencias al final de cada semana, unidad o trimestre, según lo manifiesta la secuencia didáctica de cada año. La finalidad de la evaluación formativa es conocer los logros y las dificultades de los aprendizajes para tomar decisiones o acciones de enseñanza necesarias a fin de que cada estudiante y el grupo en su conjunto logren las competencias.

Este tipo de evaluación se apoya en la observación y el registro sistemático durante la clase para detectar las necesidades, potencialidades y dificultades de cada uno de sus alumnos en cuanto al desarrollo de las habilidades, procesos y actitudes, así como de las competencias.

Al indagar las razones de su conducta y ritmo de aprendizaje, se puede identificar el tipo de ayuda u orientación que necesita el estudiantado y definir estrategias didácticas diferenciadas, adaptadas a los distintos estilos de aprendizaje y contextos sociales de cada estudiante, para lograr las competencias en innovación y tecnológicas. Además, permite orientar el desarrollo de actividades de refuerzo y de ampliación como apoyo específico al final de la unidad o del ciclo escolar.

**Evaluación sumativa.** La evaluación sumativa certifica y asigna una nota a la calidad del desempeño del estudiantado (MINED, 2008a, 2008b). Según la secuencia didáctica propuesta, al inicio de cada unidad se debe definir el producto final (evidencia de aprendizaje), lo que orienta las actividades en las que el estudiantado aplica la comprensión de conceptos y los procedimientos y demuestra actitudes a través del trabajo en pares, en equipos y con el docente.

En este sentido, como evidencia del aprendizaje en cada semana y unidad, se deben revisar las acciones especificadas en los contenidos procedimentales correspondientes, que incluyen procesos como identificar aplicaciones tecnológicas, organizar secuencias de pasos lógicos, resolver problemas, representar o analizar datos, desarrollar una aplicación, crear prototipos, elaborar modelos, entre otros. Todos estos procesos deberían quedar registrados adicionalmente en el cuaderno de los estudiantes, como proyectos tangibles, o en registros grupales.

Para evaluar cada producto, se recomienda elaborar un instrumento en el que se detallen los criterios según los indicadores de la unidad y se expliciten las escalas valorativas, para que la evaluación esté ligada a lo enseñado y lo aprendido en clases desde el enfoque por competencias.



MINISTERIO  
DE EDUCACIÓN

# Programa de estudio de Primer año de bachillerato

## Competencias de grado

Al finalizar el primer año de bachillerato el estudiantado será competente para:

- Formular estrategias basadas en el reconocimiento de patrones y la abstracción para resolver problemas simples empleando secuencias de pasos lógicos, estructuras de datos básicas y herramientas sencillas de programación.
- Utilizar responsablemente herramientas tecnológicas básicas para implementar soluciones sencillas a problemas, considerando la ciberseguridad y la lógica de programación.
- Experimentar con ideas originales o prototipos básicos para afrontar desafíos relevantes en su comunidad, utilizando herramientas de programación, hardware o elementos de inteligencia artificial.

## Competencia:

Aplicar los conceptos fundamentales del pensamiento computacional, enfatizando en el reconocimiento de patrones, la abstracción y los algoritmos, para resolver problemas sencillos y comprender el funcionamiento básico de una computadora y su sistema operativo.

### Contenidos conceptuales

#### ¿Qué es una computadora?

- Concepto
- Funciones principales
- Generaciones y su impacto

#### Pensamiento computacional

- ¿Qué es el pensamiento computacional?
- Descomposición
- Patrones

- Abstracción
- Algoritmos

- Elementos de un algoritmo

### Contenidos procedimentales

- 1.1. Indagación del concepto y funciones principales de la una computadora.
- 1.2. Análisis de los problemas resueltos por las distintas generaciones de computadoras, según su tecnología.
- 1.3. Práctica de descomposición de problemas a partir de tareas cotidianas.
- 1.4. Elaboración de patrones simples empleando figuras, objetos naturales o números.
- 1.5. Representación abstracta de objetos o conceptos cotidianos utilizando esquemas o textos simplificados.
- 1.6. Indagación del concepto, características e importancia de los algoritmos.
- 1.7. Reconocimiento de la entrada, procesamiento y salida de un algoritmo.

### Indicadores de logro

- 1.1. Explica el concepto y funciones principales de una computadora.
- 1.2. Ejemplifica problemas lógicos o de cálculo que han sido resueltos por las computadoras, según su generación.
- 1.3. Elabora un diagrama o lista de planificación con las tareas necesarias para abordar una situación cotidiana.
- 1.4. Reconoce un patrón simple formado por figuras, objetos naturales o números.
- 1.5. Representa objetos o conceptos cotidianos de forma abstracta, utilizando esquemas o textos simplificados.
- 1.6. Explica el concepto, características e importancia de los algoritmos.
- 1.7. Identifica la entrada, procesamiento y salida de un algoritmo.

<ul style="list-style-type: none"> <li>• Representación de algoritmos</li> </ul> <p><b>Componentes de una computadora</b> <b>Sistemas operativos</b></p>	<p>1.8. Representación visual de algoritmos relacionados con situaciones cotidianas.</p> <p>1.9. Clasificación de los componentes de una computadora como hardware y software.</p> <p>1.10. Caracterización de los principales sistemas operativos.</p>	<p>1.8. Representa algoritmos empleando herramientas visuales como diagramas de flujo y pseudocódigo.</p> <p>1.9. Desarrolla los conceptos de hardware y software brindando ejemplos.</p> <p>1.10 Explica qué es un sistema operativo y su función en la computadora.</p>
--	---	---

<p><b>Contenidos actitudinales</b></p>	<ul style="list-style-type: none"> <li>▪ Curiosidad por comprender el funcionamiento de las tecnologías digitales y sus aplicaciones.</li> <li>▪ Perseverancia para determinar patrones y resolver problemas lógicos siguiendo una estructura secuencial.</li> </ul>
<p><b>Dominio clave</b></p>	<p>Las computadoras son herramientas multifuncionales que operan mediante la integración de hardware y software, mientras que el pensamiento computacional, al descomponer problemas y buscar patrones, nos permite abordarlos de manera lógica y creativa.</p>
<p><b>Indicadores avanzados</b></p>	<ul style="list-style-type: none"> <li>▪ Relaciona las tecnologías de construcción de las computadoras de distinta generación con el tipo de problemas que han resuelto.</li> <li>▪ Utiliza un software para plantear un algoritmo simple.</li> </ul>

# Unidad **2** Lógica de programación

Duración: 5 semanas

Eje integrador: Computación

## Competencia:

Diseñar algoritmos con estructuras de control, representándolos en pseudocódigo y diagramas de flujo, para resolver problemas lógicos y prácticos implementándolos en un entorno de programación visual.

### Contenidos conceptuales

#### Fundamentos de lógica

- Introducción a la lógica de programación
- Operadores lógicos y relacionales

#### Estructuras secuenciales

- Pseudocódigo
- Escritura de secuencias
- Bloques de código

#### Estructuras condicionales

- Condicionales simples y compuestos
- Diagramas de flujo para decisiones
- Implementación en un entorno de programación visual

### Contenidos procedimentales

- 2.1. Resolución de ejercicios para evaluar expresiones de verdad o falsedad utilizando operadores lógicos y relacionales.
- 2.2. Uso de operadores lógicos (AND, OR, NOT) y operadores relacionales (<, >, ==, !=, <=, >=).
- 2.3. Abordaje de problemas simples empleando estructuras secuenciales y sintaxis en lenguaje natural.
- 2.4. Escritura de secuencias en papel usando pseudocódigo y bloques para afrontar situaciones cotidianas.
- 2.5. Diseño de estructuras condicionales simples (*if*) y compuestas (*if-else*) en pseudocódigo y diagramas de flujo.
- 2.6. Implementación de estructuras condicionales en un entorno de programación visual.

### Indicadores de logro

- 2.1. Evalúa expresiones de verdad o falsedad utilizando operadores lógicos y relacionales.
- 2.2. Utiliza apropiadamente los operadores lógicos (AND, OR, NOT) y los operadores relacionales (<, >, ==, !=, <=, >=).
- 2.3. Expresa problemas simples empleando estructuras secuenciales y sintaxis en lenguaje natural.
- 2.4. Escribe situaciones cotidianas empleando secuencias en pseudocódigo o bloques de código y viceversa.
- 2.5. Utiliza apropiadamente estructuras condicionales simples (*if*) y compuestas (*if-else*) en pseudocódigo y diagramas de flujo.
- 2.6. Implementa estructuras condicionales en un entorno de programación visual.

**Estructuras repetitivas**

- Concepto
- Tipos de bucles: *for* y *while*
- Representación en pseudocódigo
- Implementación en un entorno de programación visual

**Depuración y optimización de algoritmos**

- Identificación y corrección de errores
- Mejora de algoritmos

2.7. Explicación del concepto de bucles y su importancia en la automatización de tareas repetitivas.

2.8. Aplicación de los bucles *for* y *while* en pseudocódigo y diagramas de flujo.

2.9. Aplicación de técnicas básicas para la identificación y corrección de errores en algoritmos y programación visual.

2.10. Discusión acerca de la importancia de la depuración en la programación e identificación de errores en un algoritmo.

2.7. Explica el concepto de bucles y su importancia en la automatización de tareas repetitivas.

2.8. Aplica correctamente los tipos bucles *for* y *while* en pseudocódigo o diagramas de flujo.

2.9. Identifica errores en algoritmos dentro de un entorno de programación visual.

2.10. Explica la importancia de la depuración en la programación e identificación de errores en un algoritmo.

**Contenidos actitudinales**

- Asunción de desafíos al organizar secuencias de instrucciones para enfrentar problemas lógicos.
- Automejora de algoritmos en la búsqueda de soluciones eficientes.

**Dominio clave**

La lógica de programación es un proceso estructurado que nos permite afrontar problemas de manera sistemática y eficiente, utilizando algoritmos y estructuras de control para crear soluciones paso a paso.

**Indicador avanzado**

Propone soluciones óptimas a errores frecuentes de algoritmos implementados en un entorno de programación visual.

# Unidad **3** Estructuras de datos

Duración: 5 semanas

Eje integrador: Datos

## Competencia:

Implementar estructuras de datos básicas en un entorno de programación visual para organizar información y resolver problemas cotidianos aplicando métodos de búsqueda y algoritmos de ordenamiento.

### Contenidos conceptuales

#### Introducción a estructuras de datos

- Concepto
- Estructuras simples y complejas
- Tipos de datos: variables y listas

#### Listas y arreglos

- Concepto
- Manipulación de listas y arreglos
- Implementación en un entorno de programación visual

#### Pilas y colas

- Concepto y aplicaciones
- Implementación en un entorno de programación visual

### Contenidos procedimentales

- 3.1. Organización de datos en estructuras simples (variables) y complejas (listas, matrices, árboles y grafos) dentro de un entorno de programación visual.
- 3.2. Diferenciación de los tipos de datos y su forma de almacenamiento en variables y listas.
- 3.3. Organización de datos empleando listas y arreglos.
- 3.4. Implementación de listas y arreglos para el almacenamiento y la manipulación dinámica de datos en un entorno de programación visual.
- 3.5. Identificación de aplicaciones comunes de pilas y colas en la vida cotidiana y en la computación.
- 3.6. Implementación de pilas y colas en un entorno de programación visual mediante el uso de listas y estructuras condicionales.

### Indicadores de logro

- 3.1. Ejemplifica tipos de datos cotidianos que pueden organizarse en estructuras simples (variables) y complejas (listas, matrices, árboles y grafos).
- 3.2. Diferencia entre tipos de datos y su forma de almacenamiento en variables y listas.
- 3.3. Realiza operaciones básicas (agregar, eliminar y modificar) en una lista de al menos 5 elementos.
- 3.4. Implementa listas y arreglos para el almacenamiento y la manipulación dinámica de datos en un entorno de programación visual.
- 3.5. Ejemplifica aplicaciones comunes de pilas y colas en la vida cotidiana y en la computación.
- 3.6. Implementa pilas y colas en un entorno de programación visual mediante el uso de listas y estructuras condicionales.

### Tablas y diccionarios

- Concepto, estructura y usos
- Estructuras de datos en un entorno de programación visual
- Simulación de tablas y diccionarios

### Aplicaciones prácticas de las estructuras de datos

- Métodos básicos de búsqueda: lineal y binaria
- Algoritmos de ordenamiento: *Bubble Sort* y *Selection Sort*
- Algoritmos de búsqueda y ordenamiento en un entorno de programación visual

3.7. Explicación del concepto, estructuras y usos de las tablas y diccionarios.

3.8. Implementación de estructuras de datos en un entorno de programación visual aplicando listas para simular tablas y diccionarios.

3.9. Diferenciación entre los métodos de búsqueda lineal y binaria.

3.10. Implementación de algoritmos de búsqueda y ordenamiento en un entorno de programación visual aplicando estructuras de datos.

3.7. Explica cómo acceder a un valor específico en una tabla o diccionario sencillo de elaboración propia.

3.8. Implementa estructuras de datos en un entorno de programación visual aplicando listas para simular tablas o diccionarios.

3.9. Diferencia entre los métodos de búsqueda lineal y binaria.

3.10. Implementa un algoritmo de búsqueda y ordenamiento en un entorno de programación visual aplicando estructuras de datos.

<b>Contenido actitudinal</b>	Orden y seguridad en la implementación de estructuras de datos.
<b>Dominio clave</b>	Las estructuras de datos son herramientas para organizar y almacenar información que resultan fundamentales para manipular datos de manera eficiente y ordenada, tanto en la programación como en la vida cotidiana.
<b>Indicador avanzado</b>	Importa una lista estructurada desde un archivo CSV para su lectura en una aplicación de desarrollo propio.

# Unidad 4 Desarrollo básico de software

Duración: 7 semanas

Eje integrador: Aplicaciones

## Competencia:

Desarrollar programas básicos utilizando estructuras de control, funciones y manejo de datos para implementar soluciones a problemas sencillos y explorar las posibilidades de la programación como herramienta de innovación.

### Contenidos conceptuales

#### Introducción a los lenguajes de programación

- Entorno de desarrollo y aprendizaje integrado (IDLE)
- Editores de código fuente
  
- Sintaxis básica: variables, operadores y comentarios

#### Condicionales

- Sentencias *if*, *else* y *elif (else if)*
- Operadores relacionales y lógicos
  
- Aplicaciones prácticas de condicionales

#### Bucles

- Bucles *for* y *while*
  
- Combinación de bucles y condicionales
- Casos de uso y buenas prácticas

### Contenidos procedimentales

- 4.1. Instalación y configuración del entorno de desarrollo y aprendizaje integrado (IDLE).
- 4.2. Configuración de un entorno de edición de código fuente para programar en Python.
- 4.3. Uso de la sintaxis básica de Python para variables, operadores y comentarios.
- 4.4. Explicación de la sintaxis y usos de las sentencias *if*, *else* y *elif (else if)*.
- 4.5. Desarrollo de aplicaciones prácticas que utilicen estructuras condicionales para resolver problemas cotidianos.
- 4.6. Implementación de bucles *for* y *while* para automatizar tareas repetitivas.
- 4.7. Integración de bucles y condicionales en aplicaciones para la resolución de problemas prácticos.

### Indicadores de logro

- 4.1. Configura el entorno de desarrollo y aprendizaje integrado (IDLE).
- 4.2. Configura un entorno de edición de código fuente para programar en Python.
- 4.3. Utiliza apropiadamente la sintaxis básica de Python para variables, operadores y comentarios.
- 4.4. Utiliza apropiadamente las sentencias *if*, *else* y *elif (else if)*.
- 4.5. Aplica estructuras condicionales en el desarrollo de una aplicación simple.
- 4.6. Implementa los bucles *for* y *while* para automatizar tareas repetitivas.
- 4.7. Integra bucles y condicionales en el desarrollo de una aplicación simple.

### Funciones y modularidad

- Definición de funciones
- Parámetros y valores de retorno
- Modularización de código
- Buenas prácticas al escribir código

### Tipos de datos y manejo básico de archivos

- Cadenas, listas, tuplas y diccionarios
- Operaciones con archivos de texto
- Lectura y escritura de datos

### Librerías

- Operaciones matemáticas
- Números aleatorios
- Fecha y hora

4.8. Definición y uso de funciones estableciendo parámetros y valores de retorno.

4.9. Uso de la modularización de código en el desarrollo de aplicaciones.

4.10. Reconocimiento de buenas prácticas para facilitar la legibilidad y el mantenimiento del código.

4.11. Implementación de cadenas, listas, tuplas y diccionarios para almacenar y manipular información.

4.12. Lectura y escritura de archivos de texto aplicando operaciones de entrada/salida (E/S) para persistir datos.

4.13. Implementación de librerías integradas para realizar operaciones matemáticas.

4.14. Aplicación de librerías para generar números aleatorios y trabajar con fechas y horas.

4.8. Ejecuta funciones estableciendo parámetros y valores de retorno.

4.9. Organiza una aplicación simple en dos o más módulos independientes y reutilizables.

4.10. Reconoce prácticas que facilitan la legibilidad y el mantenimiento del código.

4.11. Implementa cadenas, listas, tuplas y diccionarios para almacenar y manipular información.

4.12. Escribe archivos de texto aplicando operaciones de entrada/salida (E/S) para persistir datos.

4.13. Resuelve problemas simples mediante la implementación de una librería integrada para realizar operaciones matemáticas.

4.14. Resuelve problemas simples mediante la implementación de una librería integrada para generar números aleatorios o trabajar con fechas y horas.

### Contenidos actitudinales

- Orientación a resultados en el desarrollo de aplicaciones sencillas.
- Atención al detalle para comprender errores de sintaxis o compilación.

### Dominio clave

La programación consiste en brindar instrucciones a una computadora para que realice tareas. Es una herramienta poderosa para la creación de soluciones innovadoras a problemas, permitiendo transformar ideas en aplicaciones funcionales mediante el uso de lenguaje Python.

### Indicador avanzado

Utiliza un sistema de control de versiones para administrar y compartir sus proyectos de programación propio.

# Unidad 5 Ciberseguridad y ciudadanía digital

Duración: 4 semanas

Eje integrador: Sistemas

## Competencia:

Aplicar principios y herramientas de ciberseguridad para proteger la información personal, detectar y evitar ciberamenazas y navegar de forma segura y ética en internet.

### Contenidos conceptuales

#### Fundamentos de ciberseguridad y ciudadanía digital

- Rastro digital y privacidad en línea
- Reputación en línea
- Marco de contenido

#### Ciberamenazas

- Ciberamenazas comunes: *phishing*, *malware* y estafas
- Información falsa y suplantación de identidad

#### Huella digital y protección de la información personal

- Contraseñas y cuentas
- Datos que permanecen en línea
- Configuraciones de privacidad
- Supervisión de dispositivos

#### Ética en el uso de internet

- Empatía, respeto y bondad en la interacción digital

### Contenidos procedimentales

- 5.1. Reconocimiento de datos sensibles que no deberían compartirse en línea, cómo encontrarlos y verificarlos.
- 5.2. Reflexión acerca de las prácticas que permiten mantener una buena reputación y presencia positiva en línea.
- 5.3. Detección de señales que caracterizan los ataques de *phishing*, *malware* y estafas en línea.
- 5.4. Detección de información falsa en línea y la confiabilidad de fuentes.
- 5.5. Aplicación de estrategias de seguridad para gestionar contraseñas y cuentas.
- 5.6. Análisis de los datos que pueden permanecer en línea en relación con las configuraciones de privacidad.
- 5.7. Análisis de situaciones que requieren acciones amables y empáticas durante la interacción en línea.

### Indicadores de logro

- 5.1. Reconoce datos sensibles que no deberían compartirse en línea, cómo encontrarlos y verificarlos.
- 5.2. Ejemplifica prácticas que permiten mantener una buena reputación y presencia positiva en línea.
- 5.3. Identifica señales características de los ataques de *phishing*, *malware* y estafas en línea.
- 5.4. Evalúa la veracidad de información disponible en línea considerando la fiabilidad de sus fuentes.
- 5.5. Lista estrategias de seguridad útiles en la gestión de contraseñas y cuentas.
- 5.6. Ejemplifica los datos que pueden permanecer en línea dependiendo de las configuraciones de privacidad.
- 5.7. Reconoce situaciones que requieren acciones amables y empáticas durante la interacción en línea.

- Ciberacoso
- Denuncia y bloqueo de contenido inapropiado
- Cómo pedir ayuda en situaciones de riesgo o acoso

5.8. Análisis de situaciones en línea que requieren acciones de reporte, bloqueo de contenido y ayuda profesional.

5.8. Reconoce situaciones en línea que requieren acciones de reporte, bloqueo de contenido o ayuda profesional.

<b>Contenidos actitudinales</b>	<ul style="list-style-type: none"> <li>▪ Respeto a los límites de la privacidad de los demás, aunque sean distintos a los propios.</li> <li>▪ Promoción de la amabilidad y la empatía en la interacción en línea.</li> </ul>
<b>Dominio clave</b>	La ciberseguridad es una responsabilidad compartida que exige proteger la información personal y navegar con ética en el mundo digital utilizando herramientas y principios para prevenir y responder ante ciberamenazas.
<b>Indicador avanzado</b>	Utiliza una aplicación de autenticación de identidad o un gestor de contraseñas para robustecer la seguridad de sus cuentas personales.

# Unidad 6 Fundamentos de inteligencia artificial (IA) Duración: 6 semanas

Eje integrador: Datos

## Competencia:

Explorar los fundamentos de la inteligencia artificial y sus aplicaciones, incluyendo el aprendizaje automático y la visión artificial, para comprender su impacto en la sociedad y proponer ideas innovadoras que utilicen estas tecnologías.

### Contenidos conceptuales

#### Introducción a la inteligencia artificial (IA)

- ¿Qué es una inteligencia artificial (IA)?
- Aplicaciones cotidianas de IA
- Historia y evolución de la IA
- Bases de aprendizaje automático (AA)
- IA débil e IA fuerte

#### Visión artificial (VA)

- ¿Cómo observan las máquinas?
- Clasificación de imágenes
- Generación de imágenes

#### Procesamiento del lenguaje natural (PLN)

- *Chatbots*
- Clasificación de textos
- Generación de textos y sus sesgos potenciales

### Contenidos procedimentales

- 6.1. Reconocimiento de las características, evolución y usos cotidianos de la inteligencia artificial (IA).
- 6.2. Indagación de los conceptos fundamentales del aprendizaje automático (AA), diferenciando entre IA fuerte e IA débil.
- 6.3. Simulación de algoritmos básicos para visión artificial (VA) y clasificación de imágenes por IA.
- 6.4. Simulación de un algoritmo básico para generación de imágenes por IA.
- 6.5. Diferenciación entre el funcionamiento de *chatbots* basados en reglas y los *chatbots* de IA.
- 6.6. Análisis de las potencialidades y limitaciones de los algoritmos de IA para clasificación y generación de textos.

### Indicadores de logro

- 6.1. Describe las características, evolución y usos cotidianos de la inteligencia artificial (IA).
- 6.2. Explica los conceptos fundamentales del aprendizaje automático (AA), incluyendo la diferencia entre IA fuerte e IA débil.
- 6.3. Escribe un algoritmo básico para visión artificial (VA) o clasificación de imágenes por IA empleando pseudocódigo.
- 6.4. Identifica los retos de formular instrucciones precisas para un algoritmo de generación de imágenes con IA.
- 6.5. Caracteriza el funcionamiento general de los *chatbots* basados en reglas y los *chatbots* de IA.
- 6.6. Identifica los retos en la programación de IA para comprender y generar textos en lenguaje natural.

**Datos y aprendizaje automático (AA)**

- Tipos de datos para IA
- Preprocesamiento de datos para AA
- Fundamentos de modelado predictivo

**Aprendizaje supervisado**

- Algoritmos de clasificación: árboles de decisión, clasificación basada en reglas y clasificación k-NN
- Fundamentos de regresión

**Aprendizaje no supervisado**

- Agrupamiento
- Aprendizaje supervisado vs. no supervisado
- Redes neuronales
- IA generativas

- 6.7. Exploración del papel de los datos en el AA y la eficiencia de los sistemas de IA.
- 6.8. Indagación del funcionamiento de los modelos predictivos basados en AA.
- 6.9. Simulación de algoritmos de clasificación usados en IA para la toma de decisiones.
- 6.10. Exploración de los modelos predictivos de IA basados en métodos de regresión.
- 6.11. Caracterización del aprendizaje no supervisado a partir de algoritmos de agrupamiento para el análisis de datos.
- 6.12. Simulación de la estructura y función de las redes neuronales y su potencial generativo.

- 6.7. Argumenta acerca del papel crucial de los datos en el AA y la eficiencia de los sistemas de IA.
- 6.8. Explica los conceptos fundamentales del modelado predictivo basado en AA.
- 6.9. Reconoce la aplicabilidad de un algoritmo básico de clasificación usado en IA para la toma de decisiones en situaciones cotidianas.
- 6.10. Describe la regresión lineal simple y su uso en el modelado predictivo por IA.
- 6.11. Explica los fundamentos del aprendizaje no supervisado a partir de un algoritmo de agrupamiento para el análisis de datos.
- 6.12. Describe la estructura y función de las redes neuronales, incluyendo su potencial generativo.

<b>Contenidos actitudinales</b>	<ul style="list-style-type: none"> <li>▪ Reflexión en los aspectos éticos de utilizar IA predictivas y generativas.</li> <li>▪ Apertura mental al analizar los modelos y aplicaciones de IA.</li> </ul>
<b>Dominio clave</b>	La IA, entendida como la capacidad de las máquinas para simular la inteligencia humana, es un campo en constante evolución. Sus herramientas, como el aprendizaje automático (AA) y la visión artificial (VA), enriquecen la comprensión del mundo y la innovación dentro de la sociedad.
<b>Indicadores avanzados</b>	<ul style="list-style-type: none"> <li>▪ Desarrolla un <i>chatbot</i> básico entrenado para responder a preguntas de un tema o contexto particular.</li> <li>▪ Genera código para una aplicación funcional a partir de una IA generativa.</li> </ul>



MINISTERIO  
DE EDUCACIÓN

# Programa de estudio de Segundo año de bachillerato

## Competencias de grado

Al finalizar el segundo año de bachillerato el estudiantado será competente para:

- Pensamiento computacional. Formular estrategias eficientes basadas en la descomposición, el reconocimiento de patrones y el análisis de datos, para resolver problemas empleando algoritmos y códigos optimizados de acuerdo con su paradigma de programación.
- Talento tecnológico. Utilizar con criterio herramientas tecnológicas apropiadas para diseñar y desarrollar soluciones efectivas a problemas, considerando aspectos de seguridad avanzada, protección de redes y desarrollo de software ajustado al hardware.
- Innovación y desarrollo. Generar soluciones innovadoras o prototipos viables para abordar desafíos relevantes u oportunidades, aplicando conocimientos de programación, web 3.0, robótica y manufactura 3D.

## VI. Referencias

- Beers, S. (2011). *21st Century Skills: Preparing Students for THEIR Future STEM*, 1-6. Recuperado de [bit.ly/3JeYtkr](https://bit.ly/3JeYtkr)
- Condemarín, M., y Medina, A. (2000). *La evaluación auténtica de los aprendizajes*. Santiago, Chile: Andrés Bello–Didáctica y Escritura. Recuperado de <https://bit.ly/2ZGTEhU>
- Fisk, P. (2017). *Education 4.0... the future of learning will be dramatically different, in school and throughout life*. Recuperado de [bit.ly/3Jc5zGl](https://bit.ly/3Jc5zGl)
- Grundy, S. (1987). *Curriculum: Product Or Praxis?* Recuperado de <https://bit.ly/3UDSGuJ>
- Hussin, A. A. (2018). Education 4.0 Made Simple: Ideas For Teaching. *International Journal of Education and Literacy Studies*, 6(3), pp. 92-98. Recuperado de <https://bit.ly/3UEbLNv>
- Jonnaert, P., Barrette, J., Masciotra, D., y Yaya, M. (2008). La competencia como organizadora de los programas de formación: hacia un desempeño competente. *Revista del Currículum y Formación del Profesorado*, 12(3), 1-32.
- Marope, M. (2017). *Reconceptualizing and Repositioning Curriculum in the 21st Century*. A Global Paradigm Shift. In International Bureau of Education-UNESCO (Ed.), Recuperado el 7 de octubre de 2020, <https://bit.ly/3sAzlyV>
- Marope, M., Griffin, P., & Gallagher, C. (2017). *Future competences and the future of curriculum*. In International Bureau of Education - UNESCO (Ed.). Recuperado de <https://bit.ly/3FrM5v0>
- Ministerio de Educación de El Salvador. (2008a). *Programa de Estudio: Cuarto Grado. Educación Básica*. San Salvador. Recuperado de <https://bit.ly/3upJLOO>
- Ministerio de Educación de El Salvador. (2008b). *Programa de Estudio. Segundo Grado. Educación Básica*. San Salvador. Recuperado de <https://bit.ly/3bSsWWE>
- Ministerio de Educación de El Salvador. (2015). *Evaluación al servicio del aprendizaje y del desarrollo*. San Salvador. Recuperado de <https://bit.ly/3EobaG2>
- Ministerio de Educación, Ciencia y Tecnología. (2022). *Programa de Estudio de Ciencia y Tecnología. Segundo a Sexto Grado de Educación Básica*. San Salvador. Recuperado de <https://bit.ly/CienciaEducativa>
- Ministry of Education Singapore, MOE (2014). *Science Syllabus Primary. Singapore, MOE. National Science Teaching Association, NSTA (2011). NSTA position statement: Quality science education and 21st century skill (1-4)*. Arlington, Virginia, USA: NSTA. Recuperado de <https://bit.ly/3svNexZ>
- Ontario Ministry of Education, OME (2007). *The Ontario Curriculum, Grades 1-8, Science and Technology*. Recuperado de <https://bit.ly/3qpRnBf>
- Puncreobutr, V. (2016). Education 4.0: New Challenge of Learning. *Journal of Humanities and Social Sciences*, 2(2), 92-97. Recuperado de <https://bit.ly/3EqxzCq>
- Skills Development Scotland, SDS. (2018). *Skills 4.0. A skills model to drive Scotland's future*. Recuperado de <https://bit.ly/3MKNYcT>
- Smith, M. K. (2000). *Curriculum theory and practice. The Encyclopedia of Pedagogy and Informal Education*. Recuperado de <https://bit.ly/3suHeFL>
- Tobón, S., Pimienta, J., & García, J. (2010). *Secuencias didácticas: aprendizaje y evaluación de competencias*. México: Pearson Educación. Recuperado de [www.pearsoneducacion.net](http://www.pearsoneducacion.net)
- Worth, K., Duque, M., & Saltiel, E. (2009). *Desinging and Implementing Inquiry-based Science Units for Primary Education. Pollen Project*, 58. Recuperado de [www.polleneuropa.net](http://www.polleneuropa.net)

**Material en  
validación**



MI  
**NUEVA  
ESCUELA**  
Reforma Educativa



GOBIERNO DE  
EL SALVADOR

MINISTERIO  
DE EDUCACIÓN