

# Computación para la Vida Real: Pensamiento Computacional Aplicado

*Tecnología e Informática | Pensamiento Computacional | para estudiantes de media (15-17 años) | 8 semanas*

## Descripción del Curso

Este curso está diseñado para estudiantes de media que desean adquirir habilidades prácticas en pensamiento computacional aplicadas a problemas cotidianos. A través de un enfoque teórico-práctico, los participantes aprenderán a analizar situaciones reales, descomponer problemas complejos en partes manejables, identificar patrones repetitivos y diseñar soluciones estructuradas mediante algoritmos sencillos.

Dirigido a jóvenes de 15 a 17 años con interés en tecnología e informática, el curso promueve el desarrollo del pensamiento lógico y la capacidad de resolución de problemas mediante actividades interactivas y proyectos colaborativos. La metodología combina explicaciones conceptuales, ejercicios guiados y retos prácticos para consolidar el aprendizaje.

Al finalizar, los estudiantes serán capaces de aplicar herramientas básicas del pensamiento computacional para abordar y resolver desafíos cotidianos, fomentando así habilidades analíticas, creativas y de razonamiento crítico que serán útiles en su vida académica y personal.

## Objetivos Generales

- Comprender los principios fundamentales del pensamiento computacional y su aplicación en la vida diaria.
- Descomponer problemas complejos en subproblemas manejables mediante el análisis sistemático.
- Detectar patrones y regularidades que faciliten la creación de soluciones eficientes.
- Construir algoritmos básicos para representar procesos y resolver problemas específicos.
- Evaluar y comunicar soluciones computacionales con claridad y precisión.

## Competencias

- Analizar problemas cotidianos descomponiéndolos en partes fundamentales para facilitar su comprensión y solución.
- Identificar patrones y regularidades en conjuntos de datos o situaciones que permitan optimizar procesos.
- Diseñar algoritmos simples que representen soluciones paso a paso a problemas planteados.
- Aplicar técnicas básicas de pensamiento computacional para resolver retos prácticos de la vida diaria.
- Comunicar de forma clara y estructurada las soluciones propuestas utilizando lenguaje lógico y diagramas.

## Requerimientos

- Conocimientos básicos de informática y manejo elemental de dispositivos digitales (computadora o tablet).
- Acceso a un computador con conexión a internet para actividades y recursos digitales.
- Material de escritura para toma de notas y elaboración de esquemas.
- Actitud abierta para el trabajo colaborativo y la resolución de problemas.

## Unidades del Curso

### Unidad 1: Introducción al Pensamiento Computacional

#### Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de definir el concepto de pensamiento computacional y explicar su importancia en la resolución de problemas cotidianos.
- Al finalizar la unidad, el estudiante será capaz de identificar y describir ejemplos de pensamiento computacional aplicados en situaciones reales.
- Al finalizar la unidad, el estudiante será capaz de analizar un problema simple y descomponerlo en subproblemas manejables para facilitar su solución.
- Al finalizar la unidad, el estudiante será capaz de reconocer patrones y regularidades en problemas básicos para proponer soluciones iniciales.
- Al finalizar la unidad, el estudiante será capaz de comunicar de forma clara y ordenada las etapas del pensamiento computacional utilizadas en la resolución de un problema dado.

#### Contenidos Temáticos

##### 1. ¿Qué es el Pensamiento Computacional?

- Definición y origen del pensamiento computacional: Introducción al concepto y su evolución.
- Componentes principales: Descomposición, reconocimiento de patrones, abstracción y diseño de algoritmos.
- Importancia del pensamiento computacional en la vida diaria y en diferentes áreas del conocimiento.

##### 2. Aplicaciones del Pensamiento Computacional en Situaciones Cotidianas

- Ejemplos prácticos en la vida diaria: organización de tareas, toma de decisiones, planificación de actividades.
- Uso del pensamiento computacional en distintas profesiones y en problemas sociales.
- Análisis de casos reales para identificar el uso del pensamiento computacional.

##### 3. Descomposición de Problemas

- Definición de descomposición: dividir un problema complejo en partes más simples.
- Estrategias para descomponer problemas simples y cotidianos.
- Ejercicios prácticos para aplicar la descomposición en problemas reales.

#### 4. Reconocimiento de Patrones y Regularidades

- Concepto de patrones y su importancia en la solución de problemas.
- Identificación de patrones en datos, situaciones y problemas cotidianos.
- Ejemplos y ejercicios para detectar regularidades y usarlas para proponer soluciones.

#### 5. Comunicación del Proceso de Resolución

- Importancia de comunicar claramente el proceso de pensamiento computacional.
- Estructura para presentar la solución: etapas y lenguaje claro.
- Prácticas de explicación oral y escrita de soluciones basadas en el pensamiento computacional.

### Actividades

#### Actividad 1: Conceptualizando el Pensamiento Computacional

**Objetivo:** Definir el concepto de pensamiento computacional y explicar su importancia (Objetivo 1).

**Descripción paso a paso:**

- El docente presenta una breve introducción con ejemplos cotidianos.
- Los estudiantes, en grupos pequeños, discuten y elaboran una definición propia del pensamiento computacional.
- Cada grupo expone su definición y explicación sobre su importancia.
- Se realiza una puesta en común y se elabora una definición consensuada en clase.

**Organización:** Grupos pequeños (3-4 estudiantes)

**Producto esperado:** Definición escrita y explicación oral consensuada.

**Duración estimada:** 50 minutos

#### Actividad 2: Identificación de Pensamiento Computacional en la Vida Real

**Objetivo:** Identificar y describir ejemplos de pensamiento computacional aplicados en situaciones reales (Objetivo 2).

**Descripción paso a paso:**

- El docente presenta casos o videos cortos donde se muestran situaciones cotidianas y profesionales.
- Los estudiantes, en parejas, analizan cada caso para identificar qué aspectos del pensamiento computacional se aplican.
- Se redacta un breve resumen que explique el ejemplo y las técnicas utilizadas.
- Las parejas comparten sus análisis con la clase para ampliar la comprensión.

**Organización:** Parejas

**Producto esperado:** Resumen escrito de ejemplos y técnicas identificadas.

**Duración estimada:** 40 minutos

#### Actividad 3: Descomposición de un Problema Cotidiano

**Objetivo:** Analizar un problema simple y descomponerlo en subproblemas manejables (Objetivo 3).

**Descripción paso a paso:**

- El docente plantea un problema cotidiano (por ejemplo, organizar una fiesta o planificar un viaje).
- Individualmente, los estudiantes escriben las partes o subproblemas en que pueden dividir el problema general.
- En grupos, comparan sus descomposiciones y elaboran un esquema común.
- Se discuten las ventajas de la descomposición para facilitar la solución.

**Organización:** Individual y luego grupos pequeños

**Producto esperado:** Esquema de descomposición del problema.

**Duración estimada:** 50 minutos

#### **Actividad 4: Reconocimiento y Uso de Patrones**

**Objetivo:** Reconocer patrones y regularidades en problemas básicos para proponer soluciones (Objetivo 4).

**Descripción paso a paso:**

- Se presentan secuencias de números, objetos o situaciones con patrones evidentes y otros con patrones menos claros.
- Los estudiantes, en grupos, identifican y describen los patrones observados.
- Luego, proponen cómo esos patrones pueden ayudar a resolver o predecir soluciones en problemas similares.
- Presentan sus conclusiones a la clase.

**Organización:** Grupos pequeños

**Producto esperado:** Listado de patrones identificados y propuesta de uso para solución de problemas.

**Duración estimada:** 45 minutos

#### **Actividad 5: Comunicación del Proceso de Pensamiento Computacional**

**Objetivo:** Comunicar de forma clara y ordenada las etapas del pensamiento computacional utilizadas en la resolución de un problema (Objetivo 5).

**Descripción paso a paso:**

- Se entrega a cada estudiante un problema sencillo para resolver aplicando pensamiento computacional.
- El estudiante realiza la descomposición, identifica patrones y propone solución.
- Escribe un informe breve explicando cada etapa del proceso utilizado.
- Opcional: presenta su solución y explicación oral a la clase o en grupos pequeños.

**Organización:** Individual

**Producto esperado:** Informe escrito y posible presentación oral.

**Duración estimada:** 60 minutos

#### **Evaluación**

## Evaluación Diagnóstica

**Qué se evalúa:** Conocimientos previos sobre pensamiento computacional y habilidades básicas para resolver problemas.

**Cómo se evalúa:** Cuestionario breve con preguntas abiertas y de opción múltiple sobre conceptos básicos y ejemplos cotidianos.

**Instrumento sugerido:** Cuestionario escrito o digital al inicio de la unidad.

## Evaluación Formativa

**Qué se evalúa:** Participación y desempeño en actividades de descomposición, reconocimiento de patrones, identificación de ejemplos y comunicación.

**Cómo se evalúa:** Observación directa, revisión de productos parciales (esquemas, resúmenes, informes) y retroalimentación continua.

**Instrumento sugerido:** Rúbrica de evaluación para actividades grupales e individuales, lista de cotejo y registro anecdótico.

## Evaluación Sumativa

**Qué se evalúa:** Capacidad para definir el pensamiento computacional, identificar ejemplos, descomponer un problema, reconocer patrones y comunicar el proceso completo.

**Cómo se evalúa:** Proyecto final individual que incluya la resolución de un problema aplicando el pensamiento computacional y un informe escrito o presentación oral detallando las etapas utilizadas.

**Instrumento sugerido:** Rúbrica que contemple claridad conceptual, aplicación práctica, análisis y comunicación.

## Unidad 2: Análisis de Problemas Cotidianos

### Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar problemas cotidianos específicos en su entorno mediante la observación y descripción detallada.
- Al finalizar la unidad, el estudiante será capaz de definir claramente los problemas seleccionados, estableciendo sus límites y objetivos de solución.
- Al finalizar la unidad, el estudiante será capaz de descomponer problemas complejos en subproblemas más simples, utilizando estrategias de análisis sistemático.
- Al finalizar la unidad, el estudiante será capaz de organizar y representar la descomposición de problemas mediante esquemas o diagramas que faciliten su comprensión y abordaje.

### Contenidos Temáticos

#### Análisis de Problemas Cotidianos

- **Introducción al análisis de problemas**

- Definición de problema en el contexto cotidiano
- Importancia de identificar y analizar problemas para la vida diaria y la computación

- **Identificación de problemas cotidianos**

- Técnicas de observación para detectar problemas en el entorno
- Descripción detallada del problema: quién, qué, cuándo, dónde y cómo
- Ejemplos prácticos de problemas comunes en la escuela, hogar o comunidad

- **Definición clara del problema**

- Establecimiento de los límites del problema
- Determinación de objetivos claros para la solución
- Diferenciación entre síntomas y causas del problema
- Formulación del enunciado del problema

- **Descomposición de problemas complejos**

- Concepto de descomposición y su importancia
- Estrategias para dividir un problema en subproblemas manejables
- Identificación de relaciones entre subproblemas
- Ejercicios prácticos de descomposición de problemas cotidianos

- **Organización y representación de la descomposición**

- Herramientas visuales: diagramas de árbol, esquemas y mapas conceptuales
- Creación de diagramas para representar la descomposición
- Uso de softwares básicos para diagramación (opcional)
- Interpretación y análisis de diagramas para facilitar soluciones

## **Actividades**

### **Actividad 1: Observa y Describe un Problema Cotidiano**

**Objetivo:** Identificar problemas cotidianos específicos en su entorno mediante la observación y descripción detallada.

**Descripción:**

- El docente pide a los estudiantes que observen su entorno inmediato (casa, escuela o comunidad) durante un día.
- Cada estudiante selecciona un problema cotidiano que haya detectado.
- Escriben una descripción detallada del problema, incluyendo quiénes están involucrados, dónde sucede, cuándo ocurre y cómo afecta.
- Comparten su descripción con un compañero para recibir retroalimentación y mejorarla.

**Organización:** Individual y posteriormente en parejas

**Producto esperado:** Documento escrito con la descripción detallada del problema seleccionado.

**Duración estimada:** 1 hora

## **Actividad 2: Definición Clara del Problema y Establecimiento de Objetivos**

**Objetivo:** Definir claramente los problemas seleccionados, estableciendo sus límites y objetivos de solución.

### **Descripción:**

- Los estudiantes utilizan la descripción del problema identificada en la actividad anterior.
- En grupos pequeños, analizan el problema para establecer sus límites: qué está incluido y qué no en el problema.
- Formulan un enunciado claro y conciso del problema.
- Definen objetivos específicos que una solución debería alcanzar para resolver el problema.
- Comparten y discuten sus definiciones y objetivos con el grupo completo para recibir retroalimentación.

**Organización:** Grupos pequeños (3-4 estudiantes)

**Producto esperado:** Enunciado del problema y lista de objetivos claros y delimitados.

**Duración estimada:** 1.5 horas

## **Actividad 3: Descomposición del Problema en Subproblemas**

**Objetivo:** Descomponer problemas complejos en subproblemas más simples, utilizando estrategias de análisis sistemático.

### **Descripción:**

- Cada grupo toma el problema definido en la actividad anterior.
- Identifican los componentes o aspectos principales del problema.
- Dividen el problema en subproblemas y describen cada uno brevemente.
- Relacionan los subproblemas entre sí para entender el flujo del problema.
- Discuten posibles prioridades o secuencias para abordar los subproblemas.

**Organización:** Grupos pequeños (3-4 estudiantes)

**Producto esperado:** Lista de subproblemas con descripción y esquema preliminar de relaciones.

**Duración estimada:** 1.5 horas

## **Actividad 4: Representación Visual de la Descomposición del Problema**

**Objetivo:** Organizar y representar la descomposición de problemas mediante esquemas o diagramas que faciliten su comprensión y abordaje.

### **Descripción:**

- Los grupos elaboran diagramas de árbol, mapas conceptuales o esquemas para representar la descomposición realizada.
- Utilizan papel, marcadores o alguna herramienta digital sencilla (opcional) para crear el esquema.

- Presentan su diagrama al grupo clase explicando cómo organizaron y relacionaron los subproblemas.
- Se realiza una sesión de retroalimentación grupal para mejorar la claridad y utilidad de las representaciones.

**Organización:** Grupos pequeños (3-4 estudiantes) y presentación grupal

**Producto esperado:** Diagramas o esquemas visuales claros que representen la descomposición del problema.

**Duración estimada:** 2 horas

## Evaluación

### Evaluación Diagnóstica

**Qué se evalúa:** Nivel inicial de los estudiantes para identificar y describir problemas cotidianos.

**Cómo se evalúa:** Mediante una breve actividad escrita donde describan un problema que hayan observado recientemente en su entorno.

**Instrumento sugerido:** Cuestionario abierto o lista de cotejo para verificar que incluya detalles como contexto, afectación y participación.

### Evaluación Formativa

**Qué se evalúa:** Progreso en la definición clara del problema, descomposición en subproblemas y organización visual.

**Cómo se evalúa:** Observación directa durante actividades grupales, revisión de productos parciales (enunciados, listas de subproblemas, esquemas) y retroalimentación oral.

**Instrumento sugerido:** Rúbrica de desempeño que considere claridad, profundidad, organización y colaboración.

### Evaluación Sumativa

**Qué se evalúa:** Capacidad global del estudiante para identificar, definir, descomponer y representar un problema cotidiano.

**Cómo se evalúa:** Presentación final individual o grupal que incluya:

- Descripción detallada del problema
- Definición clara con límites y objetivos
- Lista de subproblemas bien estructurada
- Diagrama o esquema representativo

**Instrumento sugerido:** Rúbrica integradora que valore cada aspecto alineado a los objetivos de la unidad.

## Unidad 3: Descomposición de Problemas

### Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar y describir las partes componentes de un problema complejo utilizando ejemplos cotidianos.

- Al finalizar la unidad, el estudiante será capaz de dividir un problema complejo en subproblemas manejables aplicando técnicas de descomposición paso a paso.
- Al finalizar la unidad, el estudiante será capaz de organizar y representar visualmente la estructura de un problema descompuesto mediante diagramas o esquemas.
- Al finalizar la unidad, el estudiante será capaz de analizar subproblemas descompuestos para proponer soluciones parciales que contribuyan a resolver el problema general.
- Al finalizar la unidad, el estudiante será capaz de evaluar la efectividad de la descomposición realizada, ajustando la división de subproblemas según criterios de claridad y manejabilidad.

## **Contenidos Temáticos**

### **1. Introducción a la Descomposición de Problemas**

- Concepto de problema complejo: Definición y características.
- Importancia de la descomposición en el pensamiento computacional y en la vida diaria.
- Ejemplos cotidianos de problemas complejos.

### **2. Identificación y Descripción de Componentes de un Problema Complejo**

- Cómo reconocer las partes que componen un problema complejo.
- Uso de preguntas guía para identificar subproblemas.
- Ejemplos prácticos de descomposición inicial de problemas comunes.

### **3. Técnicas para Dividir Problemas en Subproblemas Manejables**

- Estrategias básicas para la descomposición paso a paso.
- Uso de listas, esquemas y mapas mentales para organizar ideas.
- Ejercicios prácticos para aplicar técnicas de descomposición.

### **4. Organización y Representación Visual de Problemas Descompuestos**

- Tipos de diagramas para representar problemas descompuestos (diagramas de árbol, esquemas, mapas conceptuales).
- Cómo construir un diagrama claro y efectivo.
- Ejemplos de diagramas aplicados a problemas cotidianos y académicos.

### **5. Análisis de Subproblemas y Propuesta de Soluciones Parciales**

- Cómo evaluar cada subproblema de forma independiente.
- Generación de soluciones parciales y su integración para resolver el problema general.
- Ejercicios para practicar la propuesta de soluciones a subproblemas.

### **6. Evaluación y Ajuste de la Descomposición**

- Criterios para evaluar la claridad y manejabilidad de la descomposición.
- Identificación de posibles mejoras en la división de subproblemas.
- Prácticas para ajustar y optimizar la descomposición inicial.

## **Actividades**

### **Actividad 1: Identificación de Componentes en Problemas Cotidianos**

**Objetivo:** Identificar y describir las partes componentes de un problema complejo utilizando ejemplos cotidianos.

**Descripción:**

- El docente presenta un problema cotidiano complejo (por ejemplo, planificar una fiesta de cumpleaños).
- Los estudiantes analizan el problema y listan en conjunto las partes que lo componen.
- Discusión grupal para validar y complementar las partes identificadas.

**Organización:** Grupos pequeños (3-4 estudiantes)

**Producto esperado:** Lista de componentes identificados del problema presentado.

**Duración:** 40 minutos

### **Actividad 2: Descomposición Paso a Paso de un Problema Complejo**

**Objetivo:** Dividir un problema complejo en subproblemas manejables aplicando técnicas de descomposición.

**Descripción:**

- Se entrega a los estudiantes un problema complejo (ejemplo: organizar un viaje escolar).
- En parejas, los estudiantes aplican técnicas para dividir el problema en subproblemas, utilizando listas o mapas mentales.
- Cada pareja presenta su descomposición al grupo, explicando su razonamiento.

**Organización:** Parejas

**Producto esperado:** Listado o mapa mental con la descomposición del problema.

**Duración:** 50 minutos

### **Actividad 3: Representación Visual de la Descomposición**

**Objetivo:** Organizar y representar visualmente la estructura de un problema descompuesto mediante diagramas o esquemas.

**Descripción:**

- Tomando la descomposición realizada en la actividad anterior, los estudiantes elaboran un diagrama de árbol o mapa conceptual.
- Se brinda una breve explicación sobre el uso de herramientas digitales simples (opcional) o papel y lápiz para crear el diagrama.
- Se realiza una exposición para compartir los diagramas y discutir la claridad de cada uno.

**Organización:** Parejas o grupos pequeños

**Producto esperado:** Diagrama visual que representa la descomposición del problema.

**Duración:** 60 minutos

#### **Actividad 4: Análisis y Propuesta de Soluciones Parciales**

**Objetivo:** Analizar subproblemas descompuestos para proponer soluciones parciales que contribuyan a resolver el problema general.

**Descripción:**

- Se asignan diferentes subproblemas del diagrama a grupos pequeños.
- Cada grupo analiza su subproblema y propone al menos una solución parcial concreta.
- Se comparten las soluciones propuestas y se discute cómo cada una ayuda a resolver el problema general.

**Organización:** Grupos pequeños (3-4 estudiantes)

**Producto esperado:** Documento o presentación breve con soluciones parciales para cada subproblema asignado.

**Duración:** 50 minutos

#### **Actividad 5: Evaluación y Ajuste de la Descomposición**

**Objetivo:** Evaluar la efectividad de la descomposición realizada, ajustando la división de subproblemas según criterios de claridad y manejabilidad.

**Descripción:**

- Se entrega un checklist o rúbrica con criterios para evaluar la descomposición (claridad, complejidad, manejabilidad).
- Los estudiantes revisan su diagrama y las soluciones propuestas, identificando áreas de mejora.
- Se realizan ajustes a la descomposición y se presenta la versión final mejorada.

**Organización:** Individual o parejas

**Producto esperado:** Diagrama ajustado y justificación de los cambios realizados.

**Duración:** 40 minutos

### **Evaluación**

#### **Evaluación Diagnóstica**

**Qué se evalúa:** Conocimiento inicial sobre problemas complejos y capacidad para identificar componentes básicos.

**Cómo se evalúa:** Preguntas abiertas y ejemplos breves donde el estudiante identifica partes de un problema cotidiano.

**Instrumento sugerido:** Cuestionario breve escrito o una lluvia de ideas en clase.

#### **Evaluación Formativa**

**Qué se evalúa:** Progreso en la división de problemas en subproblemas, representación visual y análisis de soluciones parciales.

**Cómo se evalúa:** Revisión continua de actividades prácticas, participación en discusiones y presentaciones grupales.

**Instrumento sugerido:** Rúbrica para evaluar listas, mapas mentales, diagramas y propuestas de soluciones durante las actividades.

### **Evaluación Sumativa**

**Qué se evalúa:** Capacidad integrada para identificar, dividir, representar, analizar y evaluar la descomposición de un problema complejo.

**Cómo se evalúa:** Proyecto final donde el estudiante debe seleccionar un problema complejo, descomponerlo, representarlo visualmente, proponer soluciones parciales y evaluar su propio proceso de descomposición.

**Instrumento sugerido:** Rubrica detallada que contemple todos los objetivos de la unidad, aplicada al proyecto final escrito o multimedia.

## **Unidad 4: Identificación de Patrones**

### **Objetivos de Aprendizaje**

- Al finalizar la unidad, el estudiante será capaz de identificar patrones y regularidades en conjuntos de datos presentados en tablas y gráficos.
- Al finalizar la unidad, el estudiante será capaz de analizar procesos cotidianos para reconocer secuencias repetitivas que permitan optimizar soluciones.
- Al finalizar la unidad, el estudiante será capaz de clasificar diferentes tipos de patrones (numéricos, visuales y secuenciales) en ejemplos prácticos.
- Al finalizar la unidad, el estudiante será capaz de aplicar el reconocimiento de patrones para anticipar resultados en situaciones problemáticas específicas.
- Al finalizar la unidad, el estudiante será capaz de explicar cómo la identificación de patrones contribuye a la creación de soluciones computacionales eficientes.

### **Contenidos Temáticos**

#### **1. Introducción a la Identificación de Patrones**

- Concepto de patrón: definición y ejemplos básicos en la vida diaria.
- Importancia de los patrones en el pensamiento computacional y en la resolución de problemas.
- Tipos de patrones: numéricos, visuales y secuenciales.

#### **2. Identificación de Patrones en Datos**

- Reconocimiento de patrones y regularidades en tablas: cómo leer y analizar datos tabulados.

- Reconocimiento de patrones en gráficos: interpretación de gráficos de barras, líneas y circulares para detectar tendencias y regularidades.
- Ejemplos prácticos: análisis de datos simples para identificar patrones numéricos y visuales.

### **3. Análisis de Procesos Cotidianos para Detectar Secuencias Repetitivas**

- Definición de secuencia repetitiva y su rol en la optimización de procesos.
- Observación y análisis de procesos diarios (por ejemplo, rutina escolar, preparación de alimentos, uso de transporte) para identificar secuencias y repeticiones.
- Representación de secuencias mediante diagramas de flujo o listas ordenadas.

### **4. Clasificación de Patrones**

- Patrones numéricos: secuencias aritméticas, geométricas y otras regularidades numéricas.
- Patrones visuales: repetición de formas, colores y figuras en imágenes y objetos.
- Patrones secuenciales: eventos o acciones que ocurren en un orden específico y repetitivo.
- Ejercicios prácticos para clasificar diferentes tipos de patrones en contextos reales.

### **5. Aplicación del Reconocimiento de Patrones para Anticipar Resultados**

- Uso de patrones para realizar predicciones simples en situaciones problemáticas.
- Ejemplos de anticipación de resultados en problemas matemáticos y situaciones cotidianas.
- Estrategias para extrapolar información a partir de patrones detectados.

### **6. Relación entre la Identificación de Patrones y las Soluciones Computacionales**

- Cómo la detección de patrones facilita la creación de algoritmos y programas eficientes.
- Ejemplos de soluciones computacionales basadas en patrones (por ejemplo, compresión de datos, reconocimiento de voz, detección de fraudes).
- Resumen y reflexión sobre la importancia de la identificación de patrones en la tecnología y la informática.

## **Actividades**

### **Actividad 1: Explorando Patrones en Datos Tabulados y Gráficos**

**Objetivo:** Identificar patrones y regularidades en conjuntos de datos presentados en tablas y gráficos.

**Descripción:**

- Se entregará a cada estudiante una tabla con datos numéricos relacionados con una situación real (por ejemplo, temperaturas diarias, ventas semanales).
- El estudiante analizará la tabla para encontrar patrones o tendencias (crecimientos, disminuciones, repeticiones).
- Luego se presentarán gráficos que representen esos datos y el estudiante deberá identificar patrones visuales y relacionarlos con la tabla.
- Finalmente, discutirán en clase las observaciones y conclusiones sobre los patrones encontrados.

**Organización:** Individual

**Producto esperado:** Informe breve con identificación y explicación de patrones detectados.

**Duración estimada:** 50 minutos

## **Actividad 2: Análisis de Secuencias en Procesos Cotidianos**

**Objetivo:** Analizar procesos cotidianos para reconocer secuencias repetitivas que permitan optimizar soluciones.

### **Descripción:**

- En grupos de 3 o 4 estudiantes, elegirán un proceso cotidiano (por ejemplo, la rutina de la mañana, preparación de un sándwich, secuencia de uso del transporte público).
- Observarán y describirán paso a paso el proceso, identificando acciones que se repiten o patrones en la secuencia.
- Representarán la secuencia mediante un diagrama de flujo o lista ordenada.
- Propondrán mejoras o formas de optimizar el proceso basado en la identificación de patrones.

**Organización:** Grupos pequeños

**Producto esperado:** Diagrama de flujo o lista ordenada con anotaciones de patrones y propuesta de optimización.

**Duración estimada:** 1 hora 15 minutos

## **Actividad 3: Clasificación de Patrones en Ejemplos Prácticos**

**Objetivo:** Clasificar diferentes tipos de patrones (numéricos, visuales y secuenciales) en ejemplos prácticos.

### **Descripción:**

- Se entregarán tarjetas con imágenes, secuencias numéricas y descripciones de procesos.
- Cada estudiante clasificará las tarjetas en las categorías: patrón numérico, visual o secuencial.
- Luego, en plenaria, compartirán sus clasificaciones y discutirán casos ambiguos o difíciles.

**Organización:** Individual con discusión grupal

**Producto esperado:** Listado clasificado de tarjetas con justificación breve.

**Duración estimada:** 45 minutos

## **Actividad 4: Aplicación Práctica para Anticipar Resultados**

**Objetivo:** Aplicar el reconocimiento de patrones para anticipar resultados en situaciones problemáticas específicas.

### **Descripción:**

- Presentar a los estudiantes problemas basados en patrones, por ejemplo: completar una secuencia numérica, prever el siguiente paso en un proceso o predecir la tendencia de un gráfico.
- Los estudiantes trabajarán en resolver los problemas anticipando resultados basados en la identificación de patrones.
- Luego, discutirán en grupo las estrategias utilizadas y compararán respuestas.

**Organización:** Individual y luego grupos pequeños

**Producto esperado:** Soluciones a problemas con explicación del razonamiento basado en patrones.

**Duración estimada:** 1 hora

## Evaluación

### Evaluación Diagnóstica

**Qué se evalúa:** Conocimientos previos sobre patrones, capacidad inicial para identificar tipos básicos de patrones en ejemplos simples.

**Cómo se evalúa:** Cuestionario corto con preguntas de opción múltiple y ejercicios breves para identificar patrones en imágenes, números y secuencias.

**Instrumento sugerido:** Cuestionario en papel o digital, con 10 preguntas variadas.

### Evaluación Formativa

**Qué se evalúa:** Progreso en la identificación, clasificación y análisis de patrones durante las actividades; capacidad para representar secuencias y anticipar resultados.

**Cómo se evalúa:** Revisión continua de productos de actividades (informes, diagramas, clasificaciones) y participación en discusiones.

**Instrumento sugerido:** Rúbrica de evaluación para actividades prácticas y observación directa del docente.

### Evaluación Sumativa

**Qué se evalúa:** Dominio integral de los objetivos: identificación, análisis, clasificación, aplicación y explicación de la importancia de los patrones en soluciones computacionales.

**Cómo se evalúa:** Proyecto final individual o en parejas que incluya:

- Análisis de un conjunto de datos con identificación de patrones.
- Representación de un proceso cotidiano con secuencias repetitivas.
- Clasificación de patrones encontrados.
- Predicción de resultados basados en esos patrones.
- Explicación escrita de la utilidad de la identificación de patrones para soluciones computacionales.

**Instrumento sugerido:** Rúbrica detallada para evaluar cada apartado del proyecto.

## Unidad 5: Diseño de Algoritmos Básicos

### Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar y describir los pasos necesarios para resolver un problema simple mediante la construcción de un algoritmo básico.
- Al finalizar la unidad, el estudiante será capaz de diseñar algoritmos secuenciales que representen procedimientos claros para resolver problemas específicos utilizando diagramas de flujo o pseudocódigo.

- Al finalizar la unidad, el estudiante será capaz de analizar y corregir algoritmos básicos para mejorar su precisión y eficiencia en la solución de problemas.
- Al finalizar la unidad, el estudiante será capaz de aplicar la descomposición de problemas en subproblemas para facilitar la construcción de algoritmos efectivos.
- Al finalizar la unidad, el estudiante será capaz de comunicar de manera clara y estructurada el funcionamiento de un algoritmo básico, justificando cada paso en función del problema planteado.

## **Contenidos Temáticos**

### **1. Introducción al concepto de algoritmo**

- Definición y características de un algoritmo: claridad, finitud, orden y efectividad.
- Importancia de los algoritmos en la vida diaria y en la computación.
- Ejemplos simples de algoritmos cotidianos (recetas, instrucciones para armar un objeto).

### **2. Identificación y descripción de pasos para solucionar problemas simples**

- Análisis de problemas simples: comprensión y determinación de objetivos.
- Descomposición del problema en pasos secuenciales.
- Redacción clara de pasos que conforman un algoritmo básico.

### **3. Diseño de algoritmos secuenciales**

- Concepto de secuencialidad en algoritmos: orden y ejecución paso a paso.
- Representación de algoritmos mediante pseudocódigo: sintaxis básica y ejemplos.
- Introducción a diagramas de flujo: símbolos estándar y su significado.
- Construcción de diagramas de flujo para algoritmos simples.

### **4. Análisis y corrección de algoritmos básicos**

- Identificación de errores comunes en algoritmos (pasos omitidos, secuencia incorrecta, ambigüedades).
- Estrategias para mejorar la precisión y eficiencia de un algoritmo.
- Revisión crítica y corrección de algoritmos propuestos.

### **5. Descomposición de problemas en subproblemas**

- Concepto y ventajas de la descomposición de problemas.
- Cómo dividir un problema en subproblemas manejables.
- Construcción de algoritmos a partir de subproblemas integrados.

### **6. Comunicación clara y estructurada de algoritmos**

- Técnicas para explicar un algoritmo paso a paso.
- Justificación de cada paso en función del problema planteado.

- Presentación oral y escrita de algoritmos básicos.

## Actividades

### Actividad 1: Identificación de pasos en un problema cotidiano

**Objetivo:** Identificar y describir los pasos necesarios para resolver un problema simple mediante la construcción de un algoritmo básico.

**Descripción:**

- Se presenta a los estudiantes un problema cotidiano sencillo (por ejemplo, preparar un sándwich).
- En grupos pequeños, los estudiantes analizan el problema y enumeran los pasos necesarios para resolverlo.
- Cada grupo redacta un algoritmo básico en lenguaje natural describiendo estos pasos.
- Los grupos comparten sus algoritmos y discuten posibles diferencias o mejoras.

**Organización:** Grupos de 3-4 estudiantes.

**Producto esperado:** Algoritmo básico escrito en lenguaje natural para el problema planteado.

**Duración estimada:** 45 minutos.

### Actividad 2: Diseño de algoritmos secuenciales con pseudocódigo y diagramas de flujo

**Objetivo:** Diseñar algoritmos secuenciales que representen procedimientos claros para resolver problemas específicos utilizando diagramas de flujo o pseudocódigo.

**Descripción:**

- Se asigna un problema simple (por ejemplo, calcular el área de un rectángulo).
- Individualmente, los estudiantes escriben el pseudocódigo que resuelve el problema.
- Posteriormente, en parejas, transforman el pseudocódigo en un diagrama de flujo utilizando símbolos estándar.
- Se realiza una puesta en común para comparar y mejorar los diseños.

**Organización:** Individual para pseudocódigo, parejas para diagramas de flujo.

**Producto esperado:** Pseudocódigo y diagrama de flujo que resuelven el problema asignado.

**Duración estimada:** 1 hora.

### Actividad 3: Análisis y corrección de algoritmos

**Objetivo:** Analizar y corregir algoritmos básicos para mejorar su precisión y eficiencia en la solución de problemas.

**Descripción:**

- Se entrega a los estudiantes un algoritmo con errores o ineficiencias (por ejemplo, pasos faltantes o redundantes).
- En grupos, los estudiantes identifican los errores y proponen correcciones.
- Se discuten las correcciones y se elaboran versiones mejoradas del algoritmo.
- Finalmente, cada grupo explica las razones de sus correcciones.

**Organización:** Grupos de 3-4 estudiantes.

**Producto esperado:** Algoritmo corregido y justificación escrita de las mejoras realizadas.

**Duración estimada:** 50 minutos.

#### **Actividad 4: Descomposición de un problema en subproblemas y construcción de algoritmo**

**Objetivo:** Aplicar la descomposición de problemas en subproblemas para facilitar la construcción de algoritmos efectivos.

##### **Descripción:**

- Se presenta un problema un poco más complejo (por ejemplo, organizar una fiesta: lista de invitados, compra de comida, decoración).
- En grupos, los estudiantes identifican y dividen el problema en subproblemas manejables.
- Cada grupo diseña un algoritmo básico para cada subproblema y luego integra estos en un algoritmo general.
- Se realiza una presentación oral donde explican cada subproblema y la integración final.

**Organización:** Grupos de 4-5 estudiantes.

**Producto esperado:** Conjunto de algoritmos para subproblemas y algoritmo general integrado, presentado oralmente con soporte escrito.

**Duración estimada:** 1 hora 15 minutos.

#### **Evaluación**

##### **Evaluación diagnóstica**

**Qué se evalúa:** Comprensión inicial del concepto de algoritmo y capacidad para identificar pasos en la resolución de problemas simples.

**Cómo se evalúa:** Preguntas abiertas y ejercicios cortos donde los estudiantes describen pasos para resolver tareas cotidianas.

**Instrumento sugerido:** Cuestionario escrito breve al inicio de la unidad.

##### **Evaluación formativa**

**Qué se evalúa:** Desarrollo progresivo en la construcción, diseño, análisis y comunicación de algoritmos básicos durante las actividades.

**Cómo se evalúa:** Observación directa, revisión de productos parciales (pseudocódigos, diagramas de flujo, correcciones) y retroalimentación continua.

**Instrumento sugerido:** Rúbricas de evaluación para pseudocódigo, diagramas de flujo y presentaciones orales; listas de cotejo para revisión de pasos y descomposición.

##### **Evaluación sumativa**

**Qué se evalúa:** Capacidad para diseñar un algoritmo básico completo, descomponer un problema, corregir errores y comunicar el proceso de manera clara y estructurada.

**Cómo se evalúa:** Proyecto final individual o en parejas que incluye:

- Diseño de un algoritmo para un problema dado (pseudocódigo y diagrama de flujo).
- Descomposición del problema en subproblemas con algoritmos correspondientes.
- Análisis y corrección de un algoritmo propuesto.
- Explicación escrita y oral justificando cada paso.

**Instrumento sugerido:** Rúbrica detallada que considere precisión, claridad, estructura, corrección y comunicación.

## **Unidad 6: Representación de Algoritmos**

### **Objetivos de Aprendizaje**

- Al finalizar la unidad, el estudiante será capaz de identificar los elementos básicos de un diagrama de flujo y explicar su función en la representación de algoritmos.
- Al finalizar la unidad, el estudiante será capaz de construir diagramas de flujo que representen algoritmos simples para resolver problemas cotidianos, asegurando la secuencia lógica y claridad en la comunicación.
- Al finalizar la unidad, el estudiante será capaz de redactar pseudocódigo para describir algoritmos básicos, utilizando una sintaxis clara y estructurada que facilite su comprensión.
- Al finalizar la unidad, el estudiante será capaz de comparar y contrastar la representación de un algoritmo mediante diagramas de flujo y pseudocódigo, evaluando la eficacia de cada método según el contexto del problema.
- Al finalizar la unidad, el estudiante será capaz de interpretar algoritmos representados en diagramas de flujo y pseudocódigo, identificando posibles errores o mejoras en la lógica del proceso.

### **Contenidos Temáticos**

#### **1. Introducción a la Representación de Algoritmos**

- Concepto de algoritmo: definición y ejemplos cotidianos.
- Importancia de representar algoritmos para facilitar su comprensión y comunicación.
- Formas comunes de representar algoritmos: diagramas de flujo y pseudocódigo.

#### **2. Elementos Básicos de un Diagrama de Flujo**

- Definición de diagrama de flujo y su utilidad.
- Símbolos fundamentales:
  - Óvalo: inicio y fin.
  - Rectángulo: proceso o acción.
  - Paralelogramo: entrada/salida de datos.
  - Rombo: decisión o condición.
  - Flechas: dirección y secuencia.

- Funcionamiento y significado de cada símbolo en la representación de algoritmos.

### **3. Construcción de Diagramas de Flujo para Problemas Simples**

- Identificación del problema y sus pasos para resolverlo.
- Organización lógica de las acciones y decisiones.
- Diseño del diagrama de flujo paso a paso:
  - Definir inicio y fin.
  - Agregar procesos y decisiones en secuencia lógica.
  - Incluir entradas y salidas relevantes.
  - Asegurar claridad y evitar ambigüedades.
- Ejemplos prácticos: cálculo de promedio, decisión para aprobar un examen, determinar si un número es par o impar.

### **4. Redacción de Pseudocódigo para Algoritmos Básicos**

- Definición y propósito del pseudocódigo.
- Características de un pseudocódigo efectivo:
  - Sintaxis clara y estructurada.
  - Uso de palabras clave comunes (inicio, fin, si, entonces, mientras, repetir, etc.).
  - Indentación para mejorar legibilidad.
- Elementos básicos del pseudocódigo:
  - Declaración de variables.
  - Entrada y salida de datos.
  - Condicionales y bucles.
  - Asignaciones y operaciones.
- Ejemplos prácticos de pseudocódigo para problemas cotidianos.

### **5. Comparación entre Diagramas de Flujo y Pseudocódigo**

- Ventajas y desventajas de cada método para representar algoritmos.
- Contextos o situaciones donde uno es más eficaz que el otro.
- Ejemplos comparativos con un mismo algoritmo representado con ambos métodos.

### **6. Interpretación y Evaluación Crítica de Algoritmos Representados**

- Lectura e interpretación de diagramas de flujo y pseudocódigo dados.
- Identificación de errores lógicos o de secuencia en las representaciones.
- Sugerencias para mejorar la claridad o la eficiencia del algoritmo.
- Prácticas de corrección y optimización de algoritmos simples.

## Actividades

### Actividad 1: Identificación y explicación de elementos básicos en diagramas de flujo

**Objetivo:** Identificar los elementos básicos de un diagrama de flujo y explicar su función en la representación de algoritmos.

**Descripción:**

- El docente presenta diferentes símbolos usados en diagramas de flujo y sus significados.
- Los estudiantes analizan un diagrama de flujo sencillo y etiquetan cada símbolo con su función.
- En parejas, discuten cómo cada símbolo contribuye a la representación del algoritmo.
- Finalmente, cada pareja expone a la clase la función de un símbolo específico.

**Organización:** Parejas

**Producto esperado:** Diagrama de flujo con símbolos etiquetados y explicación oral de su función.

**Duración estimada:** 45 minutos

### Actividad 2: Construcción de un diagrama de flujo para un problema cotidiano

**Objetivo:** Construir diagramas de flujo que representen algoritmos simples para resolver problemas cotidianos, asegurando la secuencia lógica y claridad en la comunicación.

**Descripción:**

- Se plantea un problema cotidiano (por ejemplo, decidir si una persona puede votar según su edad).
- Los estudiantes identifican los pasos necesarios para resolver el problema.
- Individualmente dibujan el diagrama de flujo correspondiente usando papel y lápiz o herramientas digitales básicas.
- Luego, en grupos pequeños, comparan sus diagramas, discuten las diferencias y mejoran su diseño final.
- Se realiza una puesta en común con varios diagramas para analizar la claridad y secuencia lógica.

**Organización:** Individual y grupos pequeños

**Producto esperado:** Diagrama de flujo claro y lógico que resuelva el problema planteado.

**Duración estimada:** 1 hora 15 minutos

### Actividad 3: Redacción de pseudocódigo para un algoritmo simple

**Objetivo:** Redactar pseudocódigo para describir algoritmos básicos con una sintaxis clara y estructurada.

**Descripción:**

- El docente explica las reglas básicas para escribir pseudocódigo y muestra ejemplos.
- Los estudiantes reciben un problema simple (por ejemplo, calcular el área de un rectángulo).
- En parejas, redactan el pseudocódigo que resuelve el problema.
- Presentan su pseudocódigo al grupo para recibir retroalimentación sobre claridad y estructura.

**Organización:** Parejas

**Producto esperado:** Pseudocódigo bien estructurado y comprensible que resuelva el problema.

**Duración estimada:** 50 minutos

#### **Actividad 4: Análisis comparativo entre diagrama de flujo y pseudocódigo**

**Objetivo:** Comparar y contrastar la representación de un algoritmo mediante diagramas de flujo y pseudocódigo, evaluando la eficacia de cada método según el contexto.

**Descripción:**

- Se proporciona a los estudiantes un mismo algoritmo representado en diagrama de flujo y pseudocódigo.
- En grupos, analizan las ventajas y desventajas de cada representación.
- Discuten en qué situaciones sería más útil cada método.
- Preparan un breve informe o presentación que resuma sus conclusiones.

**Organización:** Grupos pequeños

**Producto esperado:** Informe o presentación con el análisis comparativo y conclusiones.

**Duración estimada:** 1 hora

#### **Actividad 5: Interpretación y corrección de algoritmos representados**

**Objetivo:** Interpretar algoritmos en diagramas de flujo y pseudocódigo, identificando errores o mejoras en la lógica del proceso.

**Descripción:**

- Se entregan a los estudiantes ejemplos de diagramas de flujo y pseudocódigos con errores lógicos o secuenciales.
- Individualmente, analizan cada representación para identificar errores o puntos de mejora.
- En plenaria, discuten las posibles correcciones y mejoras.
- Finalmente, reescriben o rediseñan el algoritmo corregido.

**Organización:** Individual y plenaria

**Producto esperado:** Algoritmo corregido y justificación de los cambios realizados.

**Duración estimada:** 1 hora

### **Evaluación**

#### **Evaluación diagnóstica**

**Qué se evalúa:** Conocimientos previos sobre algoritmos y comprensión básica de diagramas de flujo y pseudocódigo.

**Cómo se evalúa:** Cuestionario breve con preguntas de opción múltiple y ejercicios simples de identificación de símbolos y estructuras.

**Instrumento sugerido:** Test escrito o digital de 10 preguntas, duración 20 minutos.

#### **Evaluación formativa**

**Qué se evalúa:** Progreso en la identificación, construcción y redacción de algoritmos en diagramas de flujo y pseudocódigo, así como la capacidad de análisis crítico.

**Cómo se evalúa:** Revisión continua de las actividades prácticas (diagramas, pseudocódigos, análisis comparativos), retroalimentación oral y escrita durante las sesiones.

**Instrumento sugerido:** Rúbrica de evaluación para cada actividad considerando claridad, correcta secuencia, uso adecuado de símbolos o sintaxis, y lógica del algoritmo.

## **Evaluación sumativa**

**Qué se evalúa:** Competencia para construir, redactar, interpretar y comparar algoritmos representados mediante diagramas de flujo y pseudocódigo.

**Cómo se evalúa:** Proyecto final donde el estudiante debe:

- Desarrollar un algoritmo para un problema planteado.
- Representarlo mediante un diagrama de flujo correcto y claro.
- Redactar el pseudocódigo correspondiente.
- Realizar un análisis comparativo breve justificando qué representación considera más adecuada para el problema.
- Identificar posibles mejoras o errores en su propio algoritmo.

**Instrumento sugerido:** Rúbrica integral que evalúe precisión, claridad, lógica, comparación crítica y autocorrección.

## **Unidad 7: Aplicación Práctica del Pensamiento Computacional**

### **Objetivos de Aprendizaje**

- Al finalizar la unidad, el estudiante será capaz de analizar problemas reales seleccionados y descomponerlos en subproblemas manejables utilizando técnicas de pensamiento computacional.
- Al finalizar la unidad, el estudiante será capaz de identificar patrones y regularidades en datos o procesos relacionados con problemas cotidianos para facilitar el diseño de soluciones computacionales.
- Al finalizar la unidad, el estudiante será capaz de diseñar y construir algoritmos básicos que representen soluciones efectivas a problemas reales aplicando principios de pensamiento computacional.
- Al finalizar la unidad, el estudiante será capaz de evaluar la eficiencia y funcionalidad de sus soluciones computacionales mediante pruebas y criterios definidos.
- Al finalizar la unidad, el estudiante será capaz de comunicar claramente sus soluciones y procesos computacionales mediante presentaciones escritas o orales, utilizando terminología adecuada.

### **Contenidos Temáticos**

#### **1. Introducción a la Aplicación Práctica del Pensamiento Computacional**

- Concepto y relevancia de aplicar pensamiento computacional en problemas cotidianos.
- Repaso breve de las habilidades básicas de pensamiento computacional.

## 2. Análisis y Descomposición de Problemas Reales

- Identificación de problemas reales adecuados para solución computacional.
- Técnicas para descomponer problemas complejos en subproblemas manejables.
- Ejemplos prácticos: descomposición paso a paso.

## 3. Identificación de Patrones y Regularidades en Datos y Procesos

- Definición de patrones y su importancia en el diseño de soluciones.
- Reconocimiento de patrones en conjuntos de datos y procesos cotidianos.
- Uso de la abstracción para simplificar el análisis de datos y procesos.

## 4. Diseño y Construcción de Algoritmos Básicos

- Principios para diseñar algoritmos efectivos.
- Representación de algoritmos mediante pseudocódigo y diagramas de flujo.
- Construcción de algoritmos para resolver subproblemas identificados.
- Ejercicios prácticos de diseño algorítmico.

## 5. Evaluación de la Eficiencia y Funcionalidad de Soluciones Computacionales

- Definición de criterios de eficiencia y funcionalidad en algoritmos.
- Técnicas de prueba y validación de soluciones computacionales.
- Identificación y corrección de errores comunes.
- Iteración y mejora continua de soluciones.

## 6. Comunicación de Soluciones y Procesos Computacionales

- Elementos clave para comunicar soluciones técnicas claramente.
- Uso adecuado de terminología computacional en presentaciones orales y escritas.
- Preparación de informes y presentaciones efectivas.
- Prácticas de exposición y retroalimentación.

## Actividades

### Actividad 1: Descomposición de un Problema Real

**Objetivo:** Analizar problemas reales seleccionados y descomponerlos en subproblemas manejables.

**Descripción:**

- El docente presenta un problema cotidiano (por ejemplo, organizar un evento escolar o planificar una ruta eficiente).
- Los estudiantes, en grupos pequeños, identifican y listan todos los componentes y subproblemas involucrados.
- Cada grupo crea un esquema visual de la descomposición del problema.

- Discusión grupal para comparar enfoques y justificar las divisiones realizadas.

**Organización:** Grupos de 3-4 estudiantes.

**Producto esperado:** Esquema de descomposición del problema con explicación oral.

**Duración:** 1 hora.

## **Actividad 2: Identificación de Patrones en Datos Cotidianos**

**Objetivo:** Identificar patrones y regularidades para facilitar el diseño de soluciones.

**Descripción:**

- Se proporciona a los estudiantes conjuntos de datos sencillos (por ejemplo, horarios de transporte, temperaturas diarias, o registros de consumo de agua).
- En parejas, analizan los datos para identificar patrones, repeticiones o tendencias.
- Explican cómo esos patrones podrían ayudar a diseñar una solución computacional (por ejemplo, automatizar alertas o planificar recursos).

**Organización:** Parejas.

**Producto esperado:** Informe breve que describe patrones encontrados y posibles aplicaciones.

**Duración:** 1 hora.

## **Actividad 3: Diseño y Representación de Algoritmos**

**Objetivo:** Diseñar y construir algoritmos básicos que representen soluciones efectivas.

**Descripción:**

- Partiendo de un subproblema de la actividad 1, cada estudiante diseña un algoritmo para resolverlo.
- Utilizan pseudocódigo y diagramas de flujo para representar su algoritmo.
- Intercambian sus diseños con un compañero para revisión y sugerencias.

**Organización:** Individual con pares para revisión.

**Producto esperado:** Algoritmo representado en pseudocódigo y diagrama de flujo.

**Duración:** 1.5 horas.

## **Actividad 4: Prueba, Evaluación y Presentación de Soluciones**

**Objetivo:** Evaluar la eficiencia y funcionalidad de soluciones y comunicar resultados claramente.

**Descripción:**

- Los estudiantes prueban sus algoritmos con diferentes casos o escenarios para verificar su funcionamiento.
- Registran resultados, identifican posibles mejoras y ajustan su diseño si es necesario.
- Preparan una presentación oral o escrita explicando el problema, la solución diseñada, pruebas realizadas y conclusiones.
- Presentan a la clase y reciben retroalimentación.

**Organización:** Individual o en parejas según preferencia.

**Producto esperado:** Informe o presentación oral con evaluación y conclusiones.

**Duración:** 2 horas.

## **Evaluación**

### **Evaluación Diagnóstica**

**Qué se evalúa:** Conocimientos previos sobre pensamiento computacional y habilidades para analizar problemas.

**Cómo se evalúa:** Cuestionario breve y actividad práctica de descomposición de un problema sencillo.

**Instrumento sugerido:** Lista de cotejo para descomposición y cuestionario de opción múltiple.

### **Evaluación Formativa**

**Qué se evalúa:** Progreso en la identificación de patrones, diseño de algoritmos y capacidad de prueba de soluciones.

**Cómo se evalúa:** Revisión continua de productos parciales (esquemas, informes, pseudocódigo), observación durante actividades y autoevaluación guiada.

**Instrumento sugerido:** Rúbricas específicas para cada actividad y registros de observación docente.

### **Evaluación Sumativa**

**Qué se evalúa:** Competencia global para analizar problemas, diseñar y evaluar soluciones computacionales, y comunicar los resultados.

**Cómo se evalúa:** Presentación final escrita u oral integrando todos los elementos trabajados y defensa ante el grupo.

**Instrumento sugerido:** Rúbrica integral que incluya análisis, diseño, evaluación y comunicación.

## **Unidad 8: Evaluación y Presentación de Soluciones**

### **Objetivos de Aprendizaje**

- Al finalizar la unidad, el estudiante será capaz de evaluar la efectividad y eficiencia de soluciones computacionales desarrolladas mediante criterios definidos para identificar mejoras.
- Al finalizar la unidad, el estudiante será capaz de comparar diferentes soluciones a un problema dado utilizando evidencias recogidas durante el proceso de desarrollo.
- Al finalizar la unidad, el estudiante será capaz de organizar y presentar soluciones computacionales de forma clara y estructurada usando recursos visuales y lenguaje técnico apropiado.
- Al finalizar la unidad, el estudiante será capaz de justificar las decisiones tomadas en el diseño y desarrollo de soluciones computacionales durante una presentación oral o escrita.
- Al finalizar la unidad, el estudiante será capaz de reflexionar críticamente sobre el impacto y aplicabilidad de sus soluciones en contextos reales, proponiendo posibles mejoras.

### **Contenidos Temáticos**

## 1. Evaluación de Soluciones Computacionales

- **1.1 Criterios para evaluar soluciones:** Definición y explicación de criterios como efectividad, eficiencia, usabilidad, escalabilidad y mantenibilidad.
- **1.2 Métodos para medir efectividad y eficiencia:** Uso de pruebas funcionales, análisis de rendimiento, y revisión de requisitos cumplidos.
- **1.3 Identificación de mejoras:** Cómo detectar áreas de mejora a partir de la evaluación y establecer prioridades para optimización.

## 2. Comparación de Soluciones

- **2.1 Recopilación de evidencias durante el desarrollo:** Registro de resultados, errores, tiempos de ejecución y retroalimentación.
- **2.2 Técnicas para comparar soluciones:** Análisis comparativo basado en criterios definidos y evidencias recogidas.
- **2.3 Toma de decisiones basada en comparaciones:** Selección de la solución más adecuada según contexto y objetivos.

## 3. Presentación de Soluciones Computacionales

- **3.1 Estructura de una presentación clara y efectiva:** Introducción, desarrollo, conclusión y espacio para preguntas.
- **3.2 Uso de recursos visuales:** Diagramas, gráficos, prototipos y demostraciones prácticas para apoyar la explicación.
- **3.3 Lenguaje técnico apropiado:** Uso correcto de términos informáticos y explicación clara para diferentes audiencias.

## 4. Justificación de Decisiones en Diseño y Desarrollo

- **4.1 Argumentación lógica:** Explicación de las razones detrás de cada decisión tomada durante el proyecto.
- **4.2 Comunicación oral y escrita:** Técnicas para expresar ideas con claridad y persuasión en presentaciones y reportes.
- **4.3 Manejo de preguntas y críticas:** Estrategias para responder a dudas y defender las decisiones con evidencia.

## 5. Reflexión Crítica sobre el Impacto y Aplicabilidad

- **5.1 Evaluación del impacto social, económico y ambiental:** Análisis de cómo la solución afecta a usuarios y contextos reales.
- **5.2 Identificación de limitaciones y áreas de mejora:** Reflexión sobre debilidades y oportunidades para futuras iteraciones.
- **5.3 Propuestas para mejoras y ampliaciones:** Ideas para optimizar y adaptar la solución a diferentes escenarios.

## Actividades

### Actividad 1: Evaluación de una Solución Computacional Existente

**Objetivo:** Evaluar la efectividad y eficiencia de soluciones computacionales desarrolladas mediante criterios definidos para identificar mejoras.

**Descripción:**

- El docente proporciona a cada estudiante o pareja una solución computacional simple (puede ser un algoritmo, un programa pequeño o un prototipo).
- Los estudiantes revisan la solución aplicando criterios de evaluación previamente explicados (efectividad, eficiencia, usabilidad).
- Registran resultados y proponen al menos dos mejoras concretas para optimizar la solución.

**Organización:** Individual o parejas

**Producto esperado:** Informe breve con evaluación y propuestas de mejora.

**Duración:** 90 minutos

### Actividad 2: Comparación y Selección de Soluciones para un Problema Común

**Objetivo:** Comparar diferentes soluciones a un problema dado utilizando evidencias recogidas durante el proceso de desarrollo.

**Descripción:**

- Presentar un problema computacional (por ejemplo, ordenar una lista, buscar información, resolver un cálculo).
- Los estudiantes en grupos desarrollan dos o más soluciones alternativas.
- Recogen evidencias (tiempos de ejecución, facilidad de uso, resultados correctos).
- Analizan y comparan las soluciones usando tablas o matrices de comparación.
- Elaboran una justificación escrita de la solución seleccionada.

**Organización:** Grupos de 3-4 estudiantes

**Producto esperado:** Informe comparativo y justificación de la solución seleccionada.

**Duración:** 2 horas

### Actividad 3: Preparación y Presentación de una Solución Computacional

**Objetivo:** Organizar y presentar soluciones computacionales de forma clara y estructurada usando recursos visuales y lenguaje técnico apropiado; justificar las decisiones tomadas.

**Descripción:**

- Cada grupo prepara una presentación oral y un soporte visual (diapositivas, diagramas, demostración del prototipo).
- Incluyen explicación de la solución, criterios usados para evaluar y comparar, y justificación de decisiones.
- Presentan ante la clase, respondiendo preguntas y comentarios.

**Organización:** Grupos de 3-4 estudiantes

**Producto esperado:** Presentación oral con apoyo visual y sesión de preguntas.

**Duración:** 1.5 horas (incluye preparación y presentación)

#### **Actividad 4: Reflexión Crítica y Propuesta de Mejoras**

**Objetivo:** Reflexionar críticamente sobre el impacto y aplicabilidad de sus soluciones en contextos reales, proponiendo posibles mejoras.

**Descripción:**

- Después de la presentación, cada estudiante escribe una reflexión personal sobre el impacto social, económico o ambiental de la solución desarrollada.
- Identifican limitaciones y proponen al menos dos mejoras o adaptaciones para futuros desarrollos.
- Comparten sus reflexiones en un foro o discusión grupal moderada por el docente.

**Organización:** Individual y discusión en grupo

**Producto esperado:** Ensayo breve de reflexión y propuestas de mejora.

**Duración:** 60 minutos

#### **Evaluación**

##### **Evaluación Diagnóstica**

**Qué se evalúa:** Conocimientos previos sobre evaluación y comparación de soluciones computacionales; habilidades básicas de presentación.

**Cómo se evalúa:** Preguntas orales o escritas breves al iniciar la unidad; discusión en clase sobre criterios de evaluación y experiencias previas.

**Instrumento sugerido:** Cuestionario diagnóstico corto y observación participativa.

##### **Evaluación Formativa**

**Qué se evalúa:** Progreso en la aplicación de criterios de evaluación, análisis comparativo, habilidades de presentación y justificación, calidad de la reflexión crítica.

**Cómo se evalúa:** Revisión y retroalimentación de informes escritos, observación durante presentaciones, participación en discusiones y foros.

**Instrumento sugerido:** Rúbrica para informes y presentaciones, listas de cotejo para participación y reflexión.

##### **Evaluación Sumativa**

**Qué se evalúa:** Capacidad para evaluar soluciones, comparar alternativas, presentar con claridad y justificar decisiones, y reflexionar críticamente sobre el impacto y mejoras.

**Cómo se evalúa:** Calificación de un proyecto integrador que incluya un informe escrito con evaluación y comparación de soluciones, presentación oral y visual, y reflexión crítica escrita.

**Instrumento sugerido:** Rúbrica integral de evaluación que contemple criterios técnicos, comunicativos y reflexivos.