

Pensamiento computacional aplicado a problemas del mundo real

Tecnología e Informática | Pensamiento Computacional

Descripción del Curso

Esta unidad, Pensamiento computacional aplicado al diseño de algoritmos para problemas del mundo real, forma parte del curso Pensamiento Computacional y está dirigida a estudiantes mayores de 17 años. En esta fase del aprendizaje se pasa de analizar problemas y patrones a la creación de algoritmos simples que resuelvan las subpartes identificadas. Los estudiantes expresarán sus soluciones en pseudocódigo o diagramas de flujo y explicarán las condiciones de uso y las pruebas básicas necesarias para validar el correcto funcionamiento de los algoritmos. Se enfatiza la descomposición de problemas en pasos ejecutables, la identificación de entradas, salidas, condiciones y casos límite, y la capacidad de justificar la adecuación de la solución propuesta. A lo largo de la unidad se trabajarán contextos reales y cercanos al alumnado para favorecer la transferencia de lo aprendido a situaciones cotidianas. Se propondrán ejercicios que involucren toma de decisiones simples, cálculos repetitivos y manejo de datos, con un enfoque en la claridad de la representación algorítmica, la lógica precisa y la verificación mediante pruebas básicas. El curso promueve la comunicación de ideas de forma estructurada y la capacidad de explicar, ante diferentes audiencias, cuándo un algoritmo es adecuado y dónde podría requerir ajustes. Al finalizar la unidad, el estudiantado deberá demostrar la habilidad para convertir un requerimiento en una secuencia de instrucciones comprensibles y verificables, justificar su adecuación en distintas condiciones y presentar soluciones de manera clara, tanto en formato escrito como diagramático.

Competencias

- Descomponer problemas en subpartes ejecutables y diseñar algoritmos para cada subparte.
- Expresar soluciones en pseudocódigo o diagramas de flujo de forma clara y correcta.
- Explicar las condiciones de uso, casos límite y pruebas básicas para validar el algoritmo.
- Aplicar pensamiento computacional para resolver problemas del mundo real de manera estructurada y razonada.
- Comunicar ideas algorítmicas de forma que sean comprensibles para distintos públicos.
- Trabajar de forma colaborativa para analizar problemas, dividir tareas y presentar resultados.

Requerimientos

- Asistencia regular a las sesiones y entrega oportuna de actividades prácticas.
- Materiales básicos: cuaderno o bloc de notas, lápiz y acceso a herramientas de diagramación o software simple para pseudocódigo/diagramas de flujo.
- Capacidad para realizar descomposición de problemas y justificar las decisiones de diseño.

- Participación en actividades de grupo y presentaciones de soluciones algorítmicas.
- Uso responsable de recursos digitales y entrega de tareas en la plataforma establecida por el curso.

Unidades del Curso

Unidad 1: Unidad 1: Pensamiento computacional aplicado a la identificación de problemas del mundo real

Objetivos de Aprendizaje

1. Proponer tres situaciones reales en las que el pensamiento computacional puede aportar una solución estructurada y eficiente.
2. Justificar, para cada problema propuesto, qué parte del problema se aborda con descomposición, patrones y algoritmos simples.
3. Desarrollar una explicación escrita que conecte cada problema con un enfoque computacional apropiado y con criterios de éxito claros.

Contenidos Temáticos

1. **Tema 1:** Introducción al pensamiento computacional y criterios de selección de problemas. Descripción corta: definición de pensamiento computacional y cómo evaluar si un problema es apto para este enfoque.
2. **Tema 2:** Análisis de problemas reales y construcción de justificación. Descripción corta: cómo analizar datos disponibles y justificar la utilidad del enfoque computacional para cada caso.
3. **Tema 3:** Casos prácticos de identificación de problemas. Descripción corta: revisión de ejemplos y discusión guiada para validar la idoneidad de los problemas propuestos.

Actividades

- **Actividad 1: "Explorando problemas reales"** Explora tres situaciones cercanas (transporte escolar, residuos y uso de laboratorios). Se presenta el problema, se discute su relevancia y se justifica el uso del pensamiento computacional. Puntos clave: identificar datos relevantes, justificar el enfoque y redactar una breve justificación. Aprendizajes: criterios de idoneidad y conexión entre problema real y pensamiento computacional.
- **Actividad 2: "Mapa de descomposición"** Para cada problema propuesto, descompón en subproblemas manejables y anota posibles reglas simples para cada subparte. Puntos clave: descomposición, límites de complejidad, señales de entrada/salida. Aprendizajes: capacidad de descomponer problemas complejos en partes tratables.
- **Actividad 3: "Justificación escrita"** Redacta una breve justificación para cada problema, explicando por qué el pensamiento computacional es adecuado y qué se espera lograr. Aprendizajes: claridad en la comunicación de criterios de idoneidad y criterios de éxito.

Evaluación

La evaluación de la Unidad 1 se alinea con el objetivo general y sus niveles de logro se describen a continuación:

- Identificación y selección de al menos tres problemas reales aptos para pensamiento computacional (40%).
- Justificación escrita de por qué el enfoque computacional es adecuado para cada problema (40%).
- Participación y calidad de las actividades de descomposición y discusión (20%).

Unidad 2: Unidad 2: Pensamiento computacional aplicado a patrones y regularidades en problemas del mundo real

Objetivos de Aprendizaje

1. Analizar conjuntos de datos o procesos del problema para identificar patrones recurrentes.
2. Documentar al menos dos ejemplos que muestren patrones y sirvan como guías para el diseño de soluciones.
3. Explicar cómo esos patrones influyen en la formulación de algoritmos simples.

Contenidos Temáticos

1. **Tema 1:** Patrones y regularidades en datos del problema. Descripción corta: identificar tendencias, reglas simples y repeticiones en datos o procesos.
2. **Tema 2:** Técnicas básicas de reconocimiento de patrones. Descripción corta: enfoques simples para detectar similitudes y regularidades (p. ej., conteos, promedios, frecuencias).
3. **Tema 3:** Documentación de ejemplos que orienten soluciones. Descripción corta: registrar al menos dos ejemplos que sirvan como guía para el desarrollo de soluciones computacionales.

Actividades

- **Actividad 1: "Exploración de datos"** Analizar subconjuntos de datos del problema para identificar patrones. Puntos clave: extracción de tendencias, regularidades y supuestos. Aprendizajes: detectar patrones relevantes para la solución.
- **Actividad 2: "Ejemplos-guía"** Documentar al menos dos ejemplos que muestren patrones en diferentes escenarios y discutir cómo guían el diseño de soluciones. Aprendizajes: consolidar evidencia de patrones y su utilidad.
- **Actividad 3: "Representación de patrones"** Representar patrones mediante notas, tablas sencillas o diagramas simples para comunicar ideas rápidamente. Aprendizajes: comunicación clara de patrones y su impacto en soluciones.

Evaluación

- Capacidad para identificar patrones en los datos/procesos (30%).
- Calidad y relevancia de los dos ejemplos documentados (40%).
- Claridad de la explicación de cómo los patrones guían la solución (30%).

Unidad 3: Unidad 3: Pensamiento computacional aplicado al diseño de algoritmos para problemas del mundo real

Objetivos de Aprendizaje

1. Descomponer problemas en subpartes ejecutables y diseñar algoritmos para cada subparte.
2. Expresar las soluciones en pseudocódigo o diagramas de flujo de forma clara y correcta.
3. Explicar las condiciones de uso, casos límite y pruebas básicas para validar el algoritmo.

Contenidos Temáticos

1. **Tema 1:** Descomposición de problemas en subpartes. Descripción corta: dividir un problema en partes manejables para facilitar la solución computacional.
2. **Tema 2:** Pseudocódigo y diagramas de flujo. Descripción corta: conceptos básicos para expresar soluciones de forma estructurada y fácil de seguir.
3. **Tema 3:** Condiciones de uso y pruebas básicas. Descripción corta: definir condiciones de entrada, límites, casos extremos y criterios de verificación.

Actividades

- **Actividad 1: "Descomposición guiada"** Para un problema seleccionado de las unidades anteriores, descompón en subpartes y define la solución computacional para cada subparte. Aprendizajes: habilidad para dividir problemas y planificar soluciones escalables.
- **Actividad 2: "Escribe pseudocódigo"** Escribe pseudocódigo para una subparte identificada o crea un diagrama de flujo simple. Puntos clave: sintaxis básica, claridad y orden lógico. Aprendizajes: representación formal de soluciones.
- **Actividad 3: "Pruebas y condiciones de uso"** Describe las condiciones de uso, casos límite y pruebas mínimas para garantizar que el algoritmo funcione correctamente en distintos escenarios. Aprendizajes: verificación y robustez de la solución.

Evaluación

- Calidad de la descomposición en subpartes y diseño de algoritmos correspondientes (35%).
- Precisión y claridad del pseudocódigo o diagrama de flujo (35%).
- Capacidad para definir condiciones de uso y pruebas básicas (30%).