

Descubriendo la inteligencia artificial: piensa como un programador

Tecnología e Informática | Informática | Aprendizaje Basado en Problemas

Descripción

En esta sesión los estudiantes explorarán cómo el pensamiento computacional permite resolver problemas cotidianos mediante la descomposición en partes más pequeñas y el diseño de algoritmos claros y precisos. Aprenderán a identificar patrones, abstraer información relevante y crear instrucciones ordenadas que sirvan para solucionar situaciones reales. Además, conocerán la relación entre estos procesos y el funcionamiento de tecnologías actuales como los chatbots y otros sistemas de inteligencia artificial, fortaleciendo su comprensión crítica sobre sus capacidades y limitaciones.

Esta experiencia es relevante porque les brinda herramientas para analizar problemas de forma estructurada, mejorar su lógica y comunicación, y entender cómo la tecnología que usan diariamente procesa información. Así, podrán desarrollar habilidades que no solo aplican en informática, sino en cualquier ámbito académico y personal.

Objetivos de Aprendizaje

- Reconocer y aplicar los pilares del pensamiento computacional para analizar y resolver problemas cotidianos.
- Identificar situaciones diarias que pueden solucionarse con instrucciones claras y ordenadas.
- Descomponer un problema en tareas más pequeñas y manejables.
- Diferenciar entre instrucciones ambiguas y precisas.
- Elaborar y representar un algoritmo sencillo mediante pasos o diagramas de flujo.
- Relacionar el uso de algoritmos con el procesamiento de información en sistemas de inteligencia artificial.
- Fortalecer el trabajo colaborativo, la comunicación y la reflexión crítica sobre la inteligencia artificial.

Recursos Necesarios

- Computadora o tablet con acceso a software para crear diagramas de flujo (ejemplo: draw.io o Google Drawings) - 1 por grupo
- Proyector o pantalla para mostrar videos y ejemplos
- Video corto sobre chatbots e inteligencia artificial (3-5 minutos)
- Hojas blancas y marcadores o lápices para diagramas y anotaciones
- Cartulinas para organizar ideas en grupos
- Tarjetas con ejemplos de instrucciones ambiguas y precisas (preparadas previamente)
- Cuaderno o libreta para anotaciones personales

- Reloj o cronómetro para controlar tiempos

Requisitos Previos

- Conocimiento básico sobre qué es un algoritmo (pasos ordenados para resolver un problema)
- Habilidad para trabajar en equipo y comunicar ideas
- Experiencias previas con problemas cotidianos o escolares que requieran seguir instrucciones
- Familiaridad básica con el uso de computadora o tablet para crear dibujos o diagramas

Actividades

Fase de Inicio

Tiempo estimado:

20 minutos

Propósito de la sesión:

Docente: Explica a los estudiantes que hoy aprenderán a pensar como programadores para resolver problemas usando pasos claros y ordenados, y descubrirán cómo esta habilidad está detrás de tecnologías que usan cotidianamente, como los chatbots.

Activación de conocimientos previos:

- **Docente:** Pregunta inicial para toda la clase: "¿Alguna vez han seguido una receta para preparar algo o las instrucciones para armar un mueble? ¿Qué pasa si las instrucciones no están claras?"
- **Estudiantes:** Responden y comparten ejemplos breves.

Motivación y enganche:

- **Docente:** Muestra un video corto (3 min) que explica qué son los chatbots y cómo usan algoritmos para entender preguntas y responder.
- **Docente:** Presenta un dato curioso: "¿Sabían que para que un chatbot responda bien, debe seguir miles de instrucciones bien ordenadas como un algoritmo?"
- **Estudiantes:** Observan el video y escuchan el dato.

Contextualización:

- **Docente:** Relaciona el pensamiento computacional con situaciones diarias, por ejemplo: "Cuando organizan su mochila para la escuela, ¿cómo deciden qué llevar primero? Eso es como planear una secuencia, un algoritmo."
- **Estudiantes:** Reflexionan y comparten ejemplos propios de su vida diaria.

Fase de Desarrollo

Tiempo estimado:

75 minutos

Presentación del contenido:

Docente: Introduce los cuatro pilares del pensamiento computacional (descomposición, reconocimiento de patrones, abstracción y diseño de algoritmos) con ejemplos sencillos y visuales, evitando solo exponer teoría, sino fomentando la participación activa.

Actividad 1: Descomponer un problema cotidiano

- **Objetivo:** Descomponer un problema en tareas más pequeñas y manejables.
- **Instrucciones:**
 - **Docente:** Divide a la clase en grupos de 3-4 estudiantes.
 - Presenta el problema: "Imaginen que quieren organizar una fiesta de cumpleaños en casa. ¿Cuáles son las tareas que deben hacer para lograrlo?"
 - **Estudiantes:** En grupo, identifican y enumeran las tareas principales y luego las dividen en subtareas más pequeñas, anotándolas en una cartulina.
- **Producto:** Lista descompuesta de tareas para organizar la fiesta.
- **Tiempo:** 20 minutos
- **Rol del docente:** Circula entre los grupos, pregunta: "¿Cómo decidieron dividir las tareas?", "¿Qué tarea creen que es la más importante?", "¿Falta alguna subtask?"

Actividad 2: Diferenciar instrucciones ambiguas y precisas

- **Objetivo:** Diferenciar entre instrucciones ambiguas y precisas.
- **Instrucciones:**
 - **Docente:** Entrega a cada grupo tarjetas con ejemplos de instrucciones ambiguas y otras precisas (ejemplos: "Pon la mesa" vs "Coloca 4 platos, 4 vasos y 4 tenedores sobre la mesa").
 - **Estudiantes:** En grupo clasifican las instrucciones en ambiguas o precisas y explican por qué.
 - **Docente:** Solicita que cada grupo comparta un ejemplo y su razonamiento con la clase.
- **Producto:** Clasificación y justificación de instrucciones.
- **Tiempo:** 20 minutos
- **Rol del docente:** Facilita la discusión, formula preguntas aclaratorias como: "¿Por qué una instrucción precisa ayuda a evitar errores?"

Actividad 3: Diseñar un algoritmo y representarlo

- **Objetivo:** Elaborar y representar un algoritmo sencillo mediante pasos ordenados o un diagrama de flujo.

- **Instrucciones:**

- **Docente:** Presenta un problema breve: "Cómo prepararse para salir de casa por la mañana".
- **Estudiantes:** En los mismos grupos, escriben los pasos para resolver el problema en orden lógico.
- **Docente:** Luego guía a los estudiantes para que transformen esos pasos en un diagrama de flujo usando el software indicado o en papel.
- **Estudiantes:** Crean su algoritmo visual y lo preparan para presentar.

- **Producto:** Algoritmo escrito y diagrama de flujo.

- **Tiempo:** 35 minutos

- **Rol del docente:** Asiste en la organización lógica de pasos, pregunta: "¿Qué pasa si se cambia el orden?", "¿Cómo podemos representar decisiones en el diagrama?"

Diferenciación

- **Para estudiantes que terminan antes:** Proponer que creen una pequeña historia donde un chatbot usa un algoritmo para responder preguntas y dibujen el proceso.
- **Para estudiantes que necesitan apoyo:** Trabajo guiado en parejas con ejemplos más simples y apoyo directo del docente para organizar ideas y crear diagramas.

Transiciones

Docente: Después de cada actividad, invita a reflexionar brevemente: "¿Cómo esta actividad nos ayuda a pensar como un programador? Ahora usaremos lo que aprendimos para diseñar algoritmos que funcionan como los chatbots."

Fase de Cierre

Tiempo estimado:

25 minutos

Síntesis:

- **Docente:** Solicita que cada grupo comparta un algoritmo creado y explique cómo usaron la descomposición, instrucciones precisas y patrones.
- **Estudiantes:** Presentan y participan en un mapa mental colectivo que el docente va escribiendo en la pizarra, destacando los pilares del pensamiento computacional.

Reflexión metacognitiva:

- ¿Cuál de los pilares del pensamiento computacional te pareció más útil para resolver el problema?
- ¿Por qué es importante que las instrucciones sean claras y precisas?
- ¿Cómo crees que los chatbots usan estas ideas para responder preguntas?

Retroalimentación:

Docente: Da retroalimentación positiva y constructiva a cada grupo, resaltando ideas claras y proponiendo mejoras específicas en la organización de los algoritmos.

Transferencia:

Docente: Relaciona lo aprendido con otras materias y con situaciones fuera del aula, por ejemplo, siguiendo instrucciones en juegos o apps, y anticipa que en futuras clases profundizarán en inteligencia artificial y programación.

Tarea o reto:

Docente: Propone que en casa identifiquen un problema cotidiano y lo descompongan en tareas, luego escriban un algoritmo para resolverlo, usando instrucciones claras que puedan compartir en la próxima clase.

Evaluación

Tipo de evaluación:

- Diagnóstica en el inicio mediante la pregunta detonadora sobre instrucciones y experiencias previas.
- Formativa durante el desarrollo al observar y guiar la descomposición, clasificación de instrucciones y diseño de algoritmos.
- Sumativa en el cierre con la presentación de algoritmos, participación en el mapa mental colectivo y respuestas a la reflexión metacognitiva.

Criterios de evaluación:

- Identifica y aplica correctamente los pilares del pensamiento computacional al analizar un problema.
- Descompone problemas en tareas pequeñas de forma lógica y ordenada.
- Distingue y explica diferencias entre instrucciones ambiguas y precisas.
- Elabora algoritmos claros y representaciones gráficas coherentes.
- Relaciona el uso de algoritmos con el funcionamiento de sistemas de inteligencia artificial.

Instrumentos sugeridos:

- Lista de cotejo para evaluar participación y productos de actividades grupales.
- Rúbrica para valorar claridad y organización de algoritmos y diagramas de flujo.
- Observación directa con registro anecdótico durante las intervenciones y reflexiones.
- Autoevaluación y coevaluación guiada al final de la sesión mediante preguntas específicas.

Evidencias de aprendizaje:

- Listas descompuestas de tareas presentadas en la primera actividad.
- Clasificación y justificación de instrucciones ambiguas y precisas.
- Algoritmos escritos y diagramas de flujo elaborados.
- Participación activa en el mapa mental colectivo y respuestas a preguntas de reflexión.