

Descubriendo el Poder de los Algoritmos: ¡Resuelve y Crea!

Tecnología e Informática | Informática | Aprendizaje Basado en Problemas

Descripción

En esta sesión, los estudiantes explorarán el fascinante mundo de los algoritmos, aprendiendo a comprenderlos, diseñarlos y aplicarlos para resolver problemas cotidianos y tecnológicos. A través de la metodología de Aprendizaje Basado en Problemas (ABP), los jóvenes se enfrentarán a situaciones reales donde tendrán que analizar, planificar y ejecutar soluciones usando algoritmos, fomentando así su pensamiento crítico y habilidades lógicas.

Este aprendizaje es fundamental porque los algoritmos están presentes en muchas de las tecnologías que los estudiantes usan diariamente, desde aplicaciones móviles hasta juegos y sistemas de búsqueda. Entender cómo funcionan y cómo construirlos les da una ventaja en el mundo digital actual, preparándolos para estudios futuros y carreras en tecnología e ingeniería.

La sesión se centra en ejercicios prácticos y colaborativos donde los estudiantes aplican conceptos para reforzar y adquirir nuevos conocimientos, asegurando que el aprendizaje sea significativo y transferible a diferentes contextos.

Objetivos de Aprendizaje

- Analizar problemas cotidianos para identificar la necesidad y función de un algoritmo.
- Diseñar algoritmos sencillos que resuelvan problemas planteados.
- Ejecutar y verificar la lógica de algoritmos mediante pseudocódigo o diagramas de flujo.
- Colaborar en equipos para discutir y mejorar propuestas de solución algorítmica.
- Reflexionar sobre la importancia de los algoritmos en la vida diaria y la tecnología.

Recursos Necesarios

- Pizarrón o pizarra blanca y marcadores.
- Computadora o tablet con acceso a software de diagramas de flujo (ejemplo: Draw.io, Lucidchart) o papel cuadriculado para diagramas manuales (1 por estudiante o grupo).
- Proyector para mostrar ejemplos visuales y videos cortos.
- Hojas impresas con problemas para resolver (1 por grupo).
- Hojas de trabajo para elaboración de pseudocódigo y diagramas.
- Reloj o cronómetro para controlar tiempos.
- Material para escritura (lápices, bolígrafos, colores).

Requisitos Previos

- Conocimiento básico sobre conceptos de secuencia, decisión y repetición en programación.
- Experiencia previa con diagramas de flujo o pseudocódigo elemental.
- Habilidades básicas para trabajar en equipo y comunicarse eficazmente.
- Familiaridad con el uso de dispositivos digitales para consultar información o realizar actividades.

Actividades

Fase de Inicio

Tiempo estimado: 20 minutos

Propósito de la sesión:

Docente: Explica a los estudiantes que hoy entenderán qué es un algoritmo y cómo pueden diseñar uno para resolver problemas reales. Destaca la importancia de este conocimiento en la vida diaria y en la tecnología que usan.

Activación de conocimientos previos:

Docente: Presenta la siguiente pregunta para discusión rápida:

- "¿Alguna vez has seguido una receta para preparar una comida o armado un mueble? ¿Crees que esa receta o instrucciones son similares a un algoritmo? ¿Por qué?"

Estudiantes: Responden y comentan sus experiencias personales en plenaria, conectando con la idea de pasos ordenados para lograr un objetivo.

Motivación y enganche:

Docente: Muestra un video corto (2-3 minutos) que ejemplifica cómo un algoritmo ayuda a ordenar tareas en un robot o en una app popular (por ejemplo, algoritmo de recomendación en redes sociales). Luego lanza el reto:

- "¿Serán capaces de diseñar un algoritmo que ayude a resolver un problema real que les plantearé?"

Contextualización:

Docente: Relaciona el contenido con la vida cotidiana del estudiante destacando que los algoritmos están detrás de muchas decisiones automáticas en sus dispositivos y que aprender a crear algoritmos les permite controlar estas herramientas y pensar de forma lógica.

Estudiantes: Escuchan y hacen preguntas para aclarar dudas iniciales.

Fase de Desarrollo

Tiempo estimado: 80 minutos

Presentación del contenido:

Docente: Explica brevemente qué es un algoritmo como una serie de pasos claros para resolver un problema; introduce conceptos básicos como secuencia, decisión y repetición con ejemplos simples. Evita exposiciones largas y se apoya en preguntas para que los estudiantes reflexionen y participen.

Actividad 1: "Detectives del Problema"

- **Objetivo:** Analizar problemas cotidianos para identificar la función de un algoritmo.
- **Instrucciones:**
 - **Docente:** Divide la clase en grupos de 4. Entrega a cada grupo un problema cotidiano (ejemplo: organizar una fiesta, preparar un sándwich con pasos específicos, planificar una ruta para visitar amigos en la ciudad).
 - Pide a cada grupo que identifique qué pasos deberían seguirse y cuál es el problema principal a resolver.
 - Solicita que escriban en su hoja los pasos que consideran necesarios para resolver el problema.
- **Organización:** Grupos de 4 estudiantes.
- **Producto:** Lista de pasos ordenados y explicación breve del problema identificado.
- **Tiempo:** 25 minutos.
- **Rol del docente:** Circular por los grupos, hacer preguntas como: "¿Qué sucede si cambian el orden de los pasos?", "¿Falta algún detalle para que alguien pueda entender el proceso?", "¿Pueden identificar decisiones o condiciones en el proceso?"

Transición:

Docente: Solicita a un representante de cada grupo compartir su problema y pasos detectados, destacando la importancia de la claridad y orden en un algoritmo. Conecta con la siguiente actividad donde diseñarán algoritmos más formales.

Actividad 2: "Diseña tu Algoritmo"

- **Objetivo:** Diseñar un algoritmo sencillo que resuelva el problema planteado.
- **Instrucciones:**
 - **Docente:** Entrega a cada grupo hojas para diagramas de flujo o acceso a software digital. Explica brevemente cómo representar pasos, decisiones (con rombos) y secuencias.
 - Los grupos deben convertir la lista de pasos creada previamente en un diagrama de flujo o pseudocódigo, cuidando que sea lógico y claro.
 - Invita a que revisen y mejoren su diseño entre ellos antes de finalizar.
- **Organización:** Grupos de 4 estudiantes.
- **Producto:** Diagrama de flujo o pseudocódigo completo del algoritmo.
- **Tiempo:** 35 minutos.
- **Rol del docente:** Apoya con ejemplos, responde dudas técnicas, plantea preguntas para profundizar en la lógica ("¿Qué pasa si esta condición no se cumple?", "¿Es necesario repetir este paso?").

Transición:

Docente: Propone revisar y ejecutar mentalmente los algoritmos diseñados para detectar errores o mejoras, preparando el terreno para la siguiente actividad.

Actividad 3: "Prueba y Mejora"

- **Objetivo:** Ejecutar y verificar la lógica de algoritmos diseñados.
- **Instrucciones:**
 - **Docente:** Solicita que cada grupo simule la ejecución de su algoritmo paso a paso, usando ejemplos concretos y comprobando que el resultado sea coherente.
 - Si encuentran errores o inconsistencias, deben discutir cómo corregirlos y aplicar las mejoras.
 - Finalmente, preparan una breve explicación para compartir con la clase.
- **Organización:** Grupos de 4 estudiantes.
- **Producto:** Algoritmo revisado y explicación oral breve.
- **Tiempo:** 20 minutos.
- **Rol del docente:** Observa la dinámica, plantea preguntas para que justifiquen decisiones, ayuda a solucionar dudas y motiva a la autoevaluación del trabajo.

Diferenciación:

- **Para estudiantes que terminan antes:** Propuesta de diseñar un algoritmo para un problema más complejo o realizar una comparación entre dos algoritmos diferentes para el mismo problema.
 - **Para estudiantes que necesitan más apoyo:** Asistencia personalizada para diseñar el algoritmo paso a paso, uso de ejemplos visuales adicionales y trabajo en parejas para facilitar la comprensión.
-

Fase de Cierre

Tiempo estimado: 20 minutos

Síntesis:

Docente: Propone un "ticket de salida" donde cada estudiante debe escribir en una hoja o digitalmente:

- Una definición breve de algoritmo con sus propias palabras.
- Un paso o concepto que encontró más interesante o complicado.
- Una aplicación práctica de algoritmos que usan en su vida diaria.

Estudiantes: Escriben sus respuestas individualmente.

Reflexión metacognitiva:

Docente: Hace preguntas para que los estudiantes reflexionen, por ejemplo:

- ¿Cómo te ayudó diseñar un algoritmo a entender mejor el problema?

- ¿Qué dificultades encontraste al representar el algoritmo y cómo las superaste?
- ¿Por qué crees que es importante pensar en algoritmos para la tecnología que usas?

Retroalimentación:

Docente: Recoge los tickets de salida y ofrece comentarios inmediatos a algunos ejemplos, destacando aciertos y señalando áreas de mejora para futuras actividades.

Transferencia:

Docente: Conecta la sesión con futuros temas de programación y lógica computacional, invitando a los estudiantes a observar en casa o en sus dispositivos cómo se aplican los algoritmos en apps, juegos o búsquedas en internet.

Tarea o reto:

Docente: Propone que cada estudiante diseñe un algoritmo simple (puede ser en pseudocódigo o diagrama de flujo) para una tarea cotidiana que realizan en casa, y lo traiga a la próxima clase para compartir.

Evaluación

Tipo de evaluación: Formativa durante la fase de desarrollo y cierre. Diagnóstica al inicio mediante la activación de conocimientos.

Criterios de evaluación:

- Capacidad para identificar y descomponer un problema en pasos claros (Relacionado con objetivo 1).
- Diseño lógico y coherente de algoritmos (Relacionado con objetivo 2).
- Aplicación correcta de conceptos básicos de algoritmos, como secuencia y decisión (Relacionado con objetivo 3).
- Participación activa y colaboración en equipo (Relacionado con objetivo 4).
- Reflexión crítica sobre la importancia y aplicación de algoritmos (Relacionado con objetivo 5).

Instrumentos sugeridos: Lista de cotejo para observación de trabajo en grupo, rúbrica para evaluar diagramas y pseudocódigos, portafolio digital o físico con productos generados, autoevaluación escrita sobre el aprendizaje.

Evidencias de aprendizaje:

- Listas de pasos y problemas identificados en la Actividad 1.
- Diagramas de flujo o pseudocódigos elaborados en la Actividad 2.
- Explicaciones orales y mejoras aplicadas en la Actividad 3.
- Respuestas en el ticket de salida y reflexiones escritas en la fase de cierre.