

Explorando la Evolución y Clasificación de los Lenguajes de Programación: De la Máquina a la Programación Orientada a Objetos

Ingeniería | Ingeniería de sistemas | Aprendizaje Basado en Problemas

Descripción

Este plan de clase está diseñado para que los estudiantes universitarios comprendan qué es un programa de computación y conozcan su evolución histórica desde el lenguaje máquina hasta los paradigmas actuales como la Programación Orientada a Objetos (POO). A través de un enfoque centrado en el Aprendizaje Basado en Problemas, los estudiantes analizarán la transformación de los lenguajes de programación, la diferencia entre intérpretes y compiladores, y la clasificación de los lenguajes en estructurados, imperativos, funcionales y lógicos.

El conocimiento adquirido permitirá a los estudiantes entender la importancia y el impacto de los lenguajes de programación en el desarrollo de software moderno, conectando con aplicaciones reales y su futura profesión en ingeniería de sistemas. Además, fomentará habilidades de pensamiento crítico, análisis y trabajo colaborativo, a través de problemáticas reales y simuladas relacionadas con la evolución y clasificación de los programas de computación.

Objetivos de Aprendizaje

- Analizar la evolución histórica de los lenguajes de programación desde lenguaje máquina hasta POO.
- Comparar las diferencias entre intérpretes y compiladores y su impacto en el desarrollo de software.
- Clasificar los lenguajes de programación según sus paradigmas: estructurados, imperativos, funcionales y lógicos.
- Argumentar la importancia de la Programación Orientada a Objetos en el contexto actual de la ingeniería de sistemas.
- Aplicar el conocimiento adquirido para resolver un problema real relacionado con la selección y uso de lenguajes de programación.

Recursos Necesarios

- Computadoras con acceso a internet y software de presentación (PowerPoint o similar).
- Proyector y pantalla para mostrar presentaciones y videos.
- Material impreso: línea del tiempo histórica de los lenguajes de programación (1 por grupo de 4 estudiantes).
- Videos breves explicativos sobre compiladores vs intérpretes (2 videos de 3-4 minutos cada uno).
- Pliegos grandes para mapas conceptuales y marcadores de colores.
- Acceso a plataforma de colaboración en línea (Google Docs o similar) para trabajo grupal.

- Cuaderno o libreta para anotaciones personales.

Requisitos Previos

- Conocimientos básicos de computación e introducción a la programación.
- Familiaridad con conceptos elementales de software y hardware.
- Habilidad para trabajar en equipo y participar en discusiones académicas.
- Experiencia previa con algún lenguaje de programación básico (preferentemente).

Actividades

Fase de Inicio

Tiempo estimado: 20 minutos

Propósito de la sesión

Docente: Explica que en esta sesión se explorará qué es un programa de computación, su evolución histórica y cómo la clasificación de lenguajes influye en el desarrollo de software moderno, enfatizando su importancia para su formación como ingenieros de sistemas.

Activación de conocimientos previos

Docente: Inicia con la pregunta detonadora proyectada: "*¿Qué creen que es un programa de computación y cómo creen que ha evolucionado con el tiempo?*" Luego, pide a los estudiantes que discutan por 5 minutos en parejas y compartan sus ideas en plenaria.

Estudiantes: Participan en parejas para responder la pregunta y luego exponen brevemente sus ideas al grupo.

Motivación y enganche

Docente: Presenta un dato curioso: "*El primer programa de computadora fue escrito en lenguaje máquina, y hoy en día, con POO, los desarrolladores crean aplicaciones complejas con miles de líneas de código que interactúan entre sí.*" Luego muestra un video de 3 minutos sobre la evolución rápida de los lenguajes de programación desde los 1950s hasta la actualidad.

Estudiantes: Observan el video con atención y anotan ideas clave.

Contextualización

Docente: Conecta el contenido con la vida cotidiana del estudiante explicando cómo los lenguajes de programación están detrás de las aplicaciones móviles, videojuegos y sistemas que utilizan diariamente, resaltando la importancia de comprender su evolución para desarrollar mejores soluciones tecnológicas.

Estudiantes: Reflexionan sobre su experiencia personal con aplicaciones y software, relacionándolas con el tema.

Fase de Desarrollo

Tiempo estimado: 78 minutos

Presentación del contenido

Docente: Introduce el contenido con una breve explicación inicial apoyada en una línea del tiempo impresa y digital que incluye: lenguaje máquina, ensamblador, FORTRAN, COBOL, BASIC, C/C++, Java, y conceptos de intérpretes vs compiladores, y clasificación de lenguajes (estructurados, imperativos, funcionales, lógicos), concluyendo con POO.

Actividad 1: Análisis de la línea del tiempo histórica

- **Objetivo específico:** Analizar la evolución histórica de los lenguajes de programación.
- **Instrucciones:**
 - Formar grupos de 4 estudiantes.
 - Recibir una línea del tiempo impresa con los hitos principales.
 - Debatir en grupo sobre las características y aportes de cada lenguaje en la evolución tecnológica.
 - Responder la pregunta: "*¿Cuál creen que fue el cambio más significativo y por qué?*"
 - Preparar una breve explicación para compartir en plenaria.
- **Organización:** Grupos de 4
- **Producto:** Exposición grupal de 3 minutos y respuestas escritas en una hoja resumen.
- **Tiempo estimado:** 25 minutos
- **Rol del docente:** Facilita el debate, escucha, formula preguntas guía como: "*¿Cómo afectó el lenguaje ensamblador al desarrollo de software?*", y orienta a completar la tarea.

Actividad 2: Comparación entre intérpretes y compiladores

- **Objetivo específico:** Comparar intérpretes y compiladores y su impacto en el desarrollo de software.
- **Instrucciones:**
 - Presentar dos videos cortos sobre intérpretes y compiladores.
 - Luego, en parejas, responder una tabla comparativa con ventajas y desventajas de cada uno.
 - Discutir en plenaria las respuestas y aclarar dudas.
- **Organización:** Parejas
- **Producto:** Tabla comparativa completada y discusión en plenaria.
- **Tiempo estimado:** 20 minutos
- **Rol del docente:** Proporciona videos, guía la elaboración de la tabla, fomenta la discusión y aclara conceptos erróneos.

Actividad 3: Clasificación de lenguajes y debate sobre POO

- **Objetivo específico:** Clasificar los lenguajes de programación y argumentar la importancia de la POO.

- **Instrucciones:**

- Entregar a cada grupo tarjetas con nombres y características breves de lenguajes representativos de cada paradigma (estructurado, imperativo, funcional, lógico, POO).
- Los grupos deben organizar las tarjetas en categorías y preparar un argumento de 2 minutos sobre por qué la POO es relevante hoy.
- Presentar sus categorías y argumentos en plenaria.

- **Organización:** Grupos de 4 (puede ser el mismo de la actividad 1)

- **Producto:** Mapa conceptual en pliego y exposición argumentativa.

- **Tiempo estimado:** 33 minutos

- **Rol del docente:** Observa la organización de categorías, fomenta el debate, formula preguntas como: "*¿Qué ventajas aporta la POO frente a los lenguajes imperativos?*" y retroalimenta.

Diferenciación

- **Estudiantes que terminan antes:** Se les invita a investigar brevemente sobre un lenguaje funcional o lógico y preparar una pequeña explicación para compartir.
- **Estudiantes que requieren apoyo:** Se les ofrece material adicional simplificado y apoyo directo del docente o un asistente, además de permitir usar esquemas visuales durante las actividades.

Transiciones

Después de cada actividad, el docente conecta el aprendizaje con la siguiente mediante preguntas guía y resúmenes breves, asegurando la continuidad del tema y la integración del conocimiento.

Fase de Cierre

Tiempo estimado: 22 minutos

Síntesis

Docente: Propone que cada estudiante escriba en una tarjeta tres ideas clave aprendidas sobre la evolución y clasificación de los lenguajes de programación y una pregunta que aún tengan. Luego, agrupa las tarjetas para generar un mapa mental colectivo en la pizarra digital.

Estudiantes: Realizan el resumen individual y participan en la elaboración del mapa mental colectivo.

Reflexión metacognitiva

Docente: Formula las siguientes preguntas para que los estudiantes respondan oralmente o por escrito:

- ¿Cómo ha cambiado tu comprensión sobre qué es un programa de computación a partir de esta sesión?
- ¿Por qué es importante conocer la diferencia entre intérpretes y compiladores cuando desarrollas software?
- ¿Cómo aplicarás el conocimiento sobre los paradigmas de programación en tus proyectos futuros?

Retroalimentación

Docente: Proporciona retroalimentación inmediata durante la actividad del mapa mental y la reflexión, destacando aportes acertados, corrigiendo conceptos erróneos y motivando a profundizar en las preguntas planteadas.

Transferencia

Docente: Conecta lo aprendido con próximas sesiones sobre programación práctica y diseño de software, y su aplicación en proyectos reales dentro de la carrera.

Tarea o reto

Docente: Propone investigar un lenguaje de programación de su elección, describiendo su tipo, uso principal y si utiliza intérprete o compilador, para presentar un breve informe en la siguiente clase.

Evaluación

Tipo de evaluación:

- **Diagnóstica:** Durante la fase de inicio, con la pregunta detonadora y discusión inicial para evaluar conocimientos previos.
- **Formativa:** Durante las actividades de desarrollo, observando participación, análisis y productos generados (línea del tiempo, tabla comparativa, mapa conceptual).
- **Sumativa:** En la fase de cierre mediante el mapa mental colectivo, respuestas a preguntas de reflexión y la tarea asignada como extensión.

Criterios de evaluación:

- Capacidad para analizar y explicar la evolución histórica de los lenguajes de programación (objetivo 1).
- Precisión en la comparación entre intérpretes y compiladores (objetivo 2).
- Correcta clasificación y argumentación sobre paradigmas de lenguajes y POO (objetivos 3 y 4).
- Aplicación adecuada del conocimiento en la resolución del problema propuesto y tareas (objetivo 5).

Instrumentos sugeridos:

- Lista de cotejo para participación y actividades grupales.
- Rúbrica para evaluar calidad de exposiciones y mapas conceptuales.
- Observación directa y registro anecdótico durante las actividades.
- Autoevaluación y coevaluación al final de la sesión para reflexión personal y grupal.

Evidencias de aprendizaje:

- Respuestas y exposiciones en la línea del tiempo histórica.
- Tabla comparativa de intérpretes vs compiladores.
- Mapa conceptual colectivo y argumentos sobre la POO.
- Respuestas escritas en la reflexión metacognitiva y tarea de investigación.

