

Descubriendo la Programación: Primeros Pasos con Java

Ingeniería | Ingeniería de sistemas | Aprendizaje Basado en Problemas

Descripción

Este plan de clase está diseñado para introducir a los estudiantes universitarios de Ingeniería de Sistemas en los fundamentos básicos de la programación, con un enfoque práctico centrado en el lenguaje Java. A través de un problema real simulado, los estudiantes desarrollarán habilidades para analizar, diseñar y crear programas sencillos en Java, comprendiendo conceptos esenciales como variables, tipos de datos, estructuras de control y sintaxis básica. La relevancia de esta sesión radica en que la programación es una competencia clave para ingenieros en sistemas, habilitándolos para resolver problemas computacionales de forma lógica y efectiva. La metodología de Aprendizaje Basado en Problemas (ABP) permite que los estudiantes se enfrenten a desafíos auténticos que reflejan situaciones profesionales, promoviendo el pensamiento crítico, la colaboración y la autoexploración, lo cual potenciará su preparación para futuros cursos y su vida profesional en el campo de la ingeniería.

Objetivos de Aprendizaje

- Analizar la estructura básica de un programa en Java y sus componentes fundamentales.
- Crear programas simples en Java que utilicen variables, tipos de datos y estructuras de control condicionales.
- Aplicar la metodología de resolución de problemas para diseñar soluciones programáticas efectivas.
- Evaluar y corregir errores básicos de sintaxis y lógica en programas escritos en Java.

Recursos Necesarios

- Computadoras con entorno de desarrollo integrado (IDE) para Java instalado (por ejemplo, Eclipse, IntelliJ IDEA o NetBeans), una por cada dos estudiantes.
- Proyector y computadora del docente para demostraciones.
- Material impreso: guías rápidas de sintaxis básica de Java (1 por estudiante).
- Acceso a internet para consulta de documentación oficial de Java.
- Pizarra blanca y marcadores.
- Cuaderno o dispositivo para tomar notas.

Requisitos Previos

- Conocimientos básicos de lógica matemática y pensamiento algorítmico.
- Familiaridad previa con conceptos computacionales elementales (como qué es un programa y datos).
- Habilidades básicas en el uso de computadoras y software.

- Comprensión de instrucciones en español técnico y científico.

Actividades

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión:

Docente: Explica que en esta sesión se abordarán los fundamentos de la programación y se aprenderá a crear programas básicos en Java, destacando la importancia de estos conocimientos para su desarrollo profesional en Ingeniería de Sistemas.

Estudiantes: Escuchan y se preparan para participar activamente.

Activación de conocimientos previos:

Docente: Formula la pregunta detonadora: “¿Qué entienden por programa de computadora y para qué creen que sirve aprender a programar en Java?”

Estudiantes: Responden de forma voluntaria, compartiendo ideas y experiencias previas durante 3 minutos.

Motivación y enganche:

Docente: Presenta un dato curioso: “Java es uno de los lenguajes más usados en el mundo, desde aplicaciones móviles hasta sistemas empresariales, y aprender sus fundamentos abre muchas puertas en la industria tecnológica.” Muestra un breve video de 2 minutos donde se ven aplicaciones reales creadas con Java.

Estudiantes: Observan el video y reflexionan sobre la utilidad práctica del lenguaje.

Contextualización:

Docente: Relaciona la programación con situaciones cotidianas y profesionales, por ejemplo, cómo un ingeniero puede automatizar tareas, resolver problemas complejos y crear soluciones digitales mediante código.

Estudiantes: Comprenden la conexión entre la programación y su futura profesión.

Fase de Desarrollo

Tiempo estimado: 40 minutos

Presentación del contenido:

Docente: Introduce el problema: “Imaginen que debemos crear un programa que calcule el promedio de notas de un estudiante y determine si está aprobado o no.” Explica que para resolverlo se necesita conocer variables, tipos de datos, operadores y estructuras condicionales en Java. No se limita a exponer, sino que plantea preguntas para guiar la exploración: “¿Cómo representarían las notas? ¿Cómo compararían el resultado con un umbral de aprobación?”

Estudiantes: Escuchan y participan respondiendo preguntas.

Actividad 1: Análisis y diseño del programa

- **Objetivo:** Analizar la problemática y diseñar la solución algorítmica.
- **Instrucciones:**
 - **Docente:** Divide a los estudiantes en grupos de 3-4 personas e indica que deben discutir y elaborar un esquema o pseudocódigo para el programa descrito.
 - Solicita que identifiquen las variables necesarias, el tipo de datos y los pasos lógicos.
 - Entrega una hoja con el problema resumido para apoyar la discusión.
- **Organización:** Grupos de 3-4 estudiantes.
- **Producto:** Pseudocódigo o esquema escrito a mano o en computadora.
- **Tiempo:** 15 minutos.
- **Rol docente:** Circular entre grupos, fomenta la reflexión con preguntas como: “¿Por qué eligieron ese tipo de dato?”, “¿Cómo se decidirá si el estudiante aprueba?”, “¿Qué estructuras de control podrían usar?”

Actividad 2: Implementación práctica en Java

- **Objetivo:** Crear un programa funcional en Java que resuelva el problema.
- **Instrucciones:**
 - **Docente:** Explica brevemente la sintaxis básica para declarar variables, leer datos y condicionales en Java usando el IDE instalado.
 - Los estudiantes trabajan en parejas para desarrollar el código del programa en sus computadoras, apoyándose en la guía rápida.
 - Se les indica probar el programa con diferentes entradas y corregir errores de sintaxis y lógica.
- **Organización:** Parejas.
- **Producto:** Código Java compilado y ejecutado con éxito para el problema planteado.
- **Tiempo:** 20 minutos.
- **Rol docente:** Asiste a las parejas, responde dudas técnicas, sugiere soluciones y fomenta el análisis crítico sobre errores y resultados.

Actividad 3: Presentación y discusión de soluciones

- **Objetivo:** Evaluar y reflexionar sobre las diferentes soluciones implementadas.
- **Instrucciones:**
 - **Docente:** Pide a un par de parejas que proyecten su código y expliquen su lógica y decisiones de programación.
 - Facilita una breve discusión sobre ventajas, posibles mejoras y errores comunes.
- **Organización:** Plenaria.

- **Producto:** Exposición oral y código compartido.
- **Tiempo:** 5 minutos.
- **Rol docente:** Modera, retroalimenta y destaca buenas prácticas y aprendizajes clave.

Diferenciación:

Para estudiantes que terminan antes: Se les propone extender el programa para que acepte múltiples estudiantes y calcule estadísticas adicionales (promedio general, número de aprobados).

Para estudiantes que requieren más apoyo: Se les ofrece ayuda individualizada para entender la sintaxis básica y se les proporcionan ejemplos comentados para facilitar la comprensión.

Transiciones:

El docente conecta cada actividad recordando el objetivo común y cómo cada paso acerca a los estudiantes a crear un programa funcional, asegurando que la comprensión se construya gradualmente.

Fase de Cierre

Tiempo estimado: 10 minutos

Síntesis:

Docente: Solicita que cada estudiante escriba en una hoja o en el chat virtual tres ideas clave aprendidas sobre programación en Java y la creación de programas.

Estudiantes: Escriben sus ideas y comparten algunas en plenaria.

Reflexión metacognitiva:

Docente plantea las preguntas exactas:

- ¿Cómo aplicaron las estructuras básicas de Java para resolver el problema planteado?
- ¿Qué dificultades encontraron al escribir y ejecutar el programa y cómo las superaron?
- ¿De qué manera este aprendizaje contribuirá a su formación como ingenieros de sistemas?

Estudiantes: Reflexionan y responden brevemente, ya sea oralmente o por escrito.

Retroalimentación:

Docente: Proporciona comentarios inmediatos sobre las ideas compartidas, reconoce los logros y sugiere áreas para seguir mejorando, enfatizando la importancia del proceso de aprendizaje.

Transferencia:

Docente: Vincula lo aprendido con futuras sesiones donde se profundizarán conceptos avanzados de programación, y con aplicaciones prácticas como desarrollo de software y automatización de procesos.

Tarea o reto:

Se asigna como reto opcional que cada estudiante cree un programa Java que solicite al usuario ingresar tres números y muestre cuál es el mayor, aplicando los conceptos vistos en clase.

Evaluación

Tipo de evaluación: La evaluación es formativa durante la fase de desarrollo al observar y guiar el trabajo en grupos y parejas, y sumativa al cierre mediante la reflexión y el producto final de programación.

- **Criterios de evaluación:**

- Analizar correctamente la estructura del problema y diseñar un algoritmo coherente (Objetivo 1).
- Crear un programa en Java que compile y funcione con variables, tipos de datos y condicionales (Objetivo 2).
- Demostrar capacidad para aplicar la metodología de resolución de problemas en programación (Objetivo 3).
- Identificar y corregir errores básicos en código Java (Objetivo 4).

- **Instrumentos sugeridos:**

- Lista de cotejo para evaluar el diseño del algoritmo y la implementación del código.
- Observación directa durante las actividades prácticas.
- Autoevaluación y reflexión escrita al final de la sesión.

- **Evidencias de aprendizaje:**

- Esquema o pseudocódigo elaborado durante la actividad 1.
- Programa en Java funcional desarrollado en la actividad 2.
- Respuestas a las preguntas de reflexión y síntesis escritas en la fase de cierre.