

# Descubriendo el Código: Fundamentos y Representación de Algoritmos

Ingeniería | Ingeniería de sistemas | Aprendizaje Basado en Problemas

## Descripción

Este plan de clase está diseñado para que los estudiantes universitarios de Ingeniería de Sistemas exploren y comprendan los fundamentos de los algoritmos y las técnicas para su representación. A través de situaciones problema reales y simuladas, los estudiantes desarrollarán habilidades para analizar, diseñar y representar algoritmos mediante diagramas de flujo, pseudocódigo y pseudocódigo estructurado. La relevancia de este aprendizaje radica en que los algoritmos son la base para la solución eficiente de problemas computacionales y la creación de software funcional y optimizado.

Esta experiencia de aprendizaje activa conecta con la vida cotidiana de los estudiantes, ya que la lógica algorítmica está presente en la organización de tareas, toma de decisiones y automatización de procesos en diversas áreas. Mediante el método de Aprendizaje Basado en Problemas, los estudiantes no solo adquieren conocimientos teóricos, sino que también desarrollan pensamiento crítico y habilidades prácticas que son esenciales para su futuro profesional en la ingeniería y las tecnologías de la información.

## Objetivos de Aprendizaje

- Analizar problemas reales para identificar procesos que pueden ser representados mediante algoritmos.
- Diseñar algoritmos utilizando técnicas de representación gráfica y textual adecuadas.
- Comparar diferentes técnicas de representación de algoritmos, identificando sus ventajas y limitaciones.
- Crear diagramas de flujo y pseudocódigos para resolver problemas planteados en clase.
- Evaluar la eficiencia y claridad de algoritmos desarrollados por pares mediante criterios definidos.

## Recursos Necesarios

- Computadoras con software de dibujo (como draw.io o Microsoft Visio) y editor de texto.
- Proyector y pantalla para presentaciones.
- Material impreso: hojas con ejercicios de problemas, guías de símbolos de diagramas de flujo, ejemplos de pseudocódigo.
- Pizarras y marcadores para trabajo grupal.
- Acceso a videos cortos explicativos sobre algoritmos y técnicas de representación.

## Requisitos Previos

- Conocimientos básicos de lógica matemática y programación elemental.
- Familiaridad con conceptos básicos de informática y sistemas computacionales.
- Habilidades para trabajar en equipo y comunicación efectiva.

## Actividades

### Sesión 1: Introducción y Análisis de Problemas para Algoritmos

#### Fase de Inicio

**Tiempo estimado: 15 minutos**

##### Propósito de la sesión:

Conectar con conocimientos previos y activar el interés para entender qué es un algoritmo y por qué es fundamental representarlo correctamente.

##### Activación de conocimientos previos:

- **Docente:** Inicia preguntando: "¿Pueden describir en pasos cómo preparar una taza de café? ¿Cómo organizarían esas instrucciones para que alguien más las siga sin problemas?"
- **Estudiantes:** Responden oralmente y en breve discusión grupal, describiendo pasos secuenciales y detectando la necesidad de orden y claridad.

##### Motivación y enganche:

- **Docente:** Presenta un dato curioso: "Cada aplicación que usan en su teléfono está basada en algoritmos que comenzaron como simples instrucciones ordenadas. Hoy veremos cómo crear esas instrucciones."
- **Estudiantes:** Escuchan y reflexionan sobre la importancia práctica de los algoritmos en su vida diaria y futura profesional.

##### Contextualización:

- **Docente:** Explica que entender y representar algoritmos es clave para diseñar soluciones en ingeniería y para resolver problemas complejos de manera sistemática.
- **Estudiantes:** Conectan el tema con sus expectativas profesionales y académicas.

#### Fase de Desarrollo

**Tiempo estimado: 95 minutos**

##### Presentación del contenido:

Se presenta un problema real: organizar el proceso de atención en una cafetería universitaria para optimizar tiempos. En lugar de una exposición magistral, se promueve el análisis grupal para identificar pasos, decisiones y procesos.

### Actividad 1: Análisis y descomposición del problema

- **Objetivo:** Analizar problemas reales para identificar procesos que pueden ser representados mediante algoritmos.
- **Instrucciones:**
  - **Docente:** Divide a la clase en grupos de 4. Entrega la descripción del problema (proceso de atención en cafetería) y pide que identifiquen las etapas y decisiones clave.
  - Solicita que escriban una lista secuencial de pasos para el proceso.
- **Organización:** Grupos de 4 estudiantes.
- **Producto:** Lista secuencial de pasos del proceso en hoja impresa.
- **Tiempo:** 30 minutos.
- **Rol del docente:** Camina entre grupos, formula preguntas como "¿Qué sucede si hay más de un cliente?", "¿Dónde hay una decisión que afecta el flujo?", "¿Hay pasos que podrían simplificarse?".

### Actividad 2: Introducción a técnicas de representación (diagramas de flujo y pseudocódigo)

- **Objetivo:** Diseñar algoritmos utilizando técnicas de representación gráfica y textual adecuadas.
- **Instrucciones:**
  - **Docente:** Presenta ejemplos de diagramas de flujo y pseudocódigo para procesos sencillos, utilizando proyector y pizarra.
  - Explica símbolos básicos, estructura y convenciones.
  - Invita a los estudiantes a relacionar su lista de pasos con estas técnicas.
- **Organización:** Plenaria con participación activa.
- **Producto:** Apuntes y esquema mental de técnicas para representación.
- **Tiempo:** 25 minutos.
- **Rol del docente:** Facilita la exploración con preguntas: "¿Cómo representarían una decisión en el diagrama?", "¿Qué ventaja tiene el pseudocódigo para describir detalles?".

### Actividad 3: Primer bosquejo de algoritmo en diagrama de flujo

- **Objetivo:** Crear diagramas de flujo para resolver problemas planteados en clase.
- **Instrucciones:**
  - **Docente:** Cada grupo utiliza hojas y marcadores para diseñar el diagrama de flujo del proceso de atención en la cafetería basado en su análisis previo.
  - Los estudiantes deben identificar entradas, procesos, decisiones y salidas.
- **Organización:** Grupos de 4 estudiantes.
- **Producto:** Diagrama de flujo manuscrito del proceso.

- **Tiempo:** 40 minutos.
- **Rol del docente:** Observa y guía con preguntas: "¿Está claro el inicio y fin?", "¿Se muestran todas las decisiones?", "¿Hay pasos redundantes?".

## Diferenciación

- Para estudiantes que terminan antes: se les invita a digitalizar su diagrama usando software y agregar comentarios explicativos.
- Para quienes necesitan apoyo: el docente proporciona ejemplos simplificados y apoyo individual para clarificar símbolos y secuencia.

## Transición a cierre

El docente solicita a los grupos que preparen una breve explicación de su diagrama para compartir en la siguiente sesión.

## Fase de Cierre

### Tiempo estimado: 10 minutos

#### Síntesis:

- **Docente:** Propone que cada estudiante escriba en una tarjeta las 3 ideas clave que aprendió hoy sobre algoritmos y su representación.
- **Estudiantes:** Escriben y entregan las tarjetas para formar un mapa mental colectivo en la pizarra.

#### Reflexión metacognitiva:

- ¿Qué pasos del problema fue más fácil o difícil representar y por qué?
- ¿Cómo ayuda un diagrama de flujo a entender mejor un proceso complejo?
- ¿Qué dudas o dificultades encontraste al diseñar tu algoritmo?

#### Retroalimentación:

El docente comenta las tarjetas y responde dudas rápidas, señalando fortalezas y aspectos a mejorar.

#### Transferencia:

Anuncia que en la próxima sesión se profundizará en pseudocódigo y evaluación de algoritmos, conectando con el diseño y análisis de soluciones en ingeniería.

## Sesión 2: Representación y Evaluación de Algoritmos

### Fase de Inicio

#### Tiempo estimado: 10 minutos

## Propósito de la sesión:

Revisar lo aprendido previamente y plantear el objetivo de representar algoritmos con pseudocódigo y evaluar su claridad y eficiencia.

## Activación de conocimientos previos:

- **Docente:** Solicita a 2 grupos que expongan brevemente su diagrama de flujo y expliquen el proceso representado.
- **Estudiantes:** Presentan y reciben preguntas de sus compañeros.

## Motivación y enganche:

- **Docente:** Muestra un video corto (3 min) sobre cómo los algoritmos optimizan procesos logísticos en empresas reconocidas.
- **Estudiantes:** Observan y comentan la importancia de algoritmos claros y eficientes.

## Contextualización:

- **Docente:** Relaciona la representación del algoritmo con la necesidad de comunicar soluciones a equipos de desarrollo y clientes.
- **Estudiantes:** Reconocen la importancia de la documentación clara en ingeniería.

## Fase de Desarrollo

### Tiempo estimado: 95 minutos

### Presentación del contenido:

Se introduce el pseudocódigo como técnica textual para describir algoritmos con estructura y claridad, complementando diagramas de flujo.

### Actividad 1: Taller de pseudocódigo

- **Objetivo:** Diseñar algoritmos utilizando pseudocódigo para describir procesos complejos.
- **Instrucciones:**
  - **Docente:** Explica las convenciones básicas del pseudocódigo (estructuras de control, variables, sentencias básicas).
  - Entrega un ejercicio práctico: escribir el pseudocódigo para el proceso de atención en la cafetería considerando condiciones especiales (por ejemplo, clientes con pedido especial).
- **Organización:** Parejas de estudiantes.
- **Producto:** Documento con pseudocódigo escrito a mano o en computadora.
- **Tiempo:** 45 minutos.
- **Rol del docente:** Asiste a las parejas, pregunta "¿Cómo manejan las decisiones condicionales?", "¿Qué ocurre si cambia el orden de pasos?", "¿El pseudocódigo es claro para un tercero?".

## Actividad 2: Evaluación y comparación de algoritmos

- **Objetivo:** Comparar diferentes técnicas de representación de algoritmos, evaluando ventajas y limitaciones.
- **Instrucciones:**
  - **Docente:** Presenta dos representaciones distintas (diagrama de flujo y pseudocódigo) de un mismo algoritmo.
  - Solicita que los estudiantes en grupos de 3 analicen cuál es más claro, por qué, y cómo mejorarían cada una.
- **Organización:** Grupos de 3 estudiantes.
- **Producto:** Informe breve con conclusiones de la comparación.
- **Tiempo:** 30 minutos.
- **Rol del docente:** Facilita la discusión, señala aspectos clave para claridad y eficiencia, y motiva a plantear mejoras.

## Actividad 3: Revisión cruzada y retroalimentación

- **Objetivo:** Evaluar la eficiencia y claridad de algoritmos desarrollados por pares mediante criterios definidos.
- **Instrucciones:**
  - **Docente:** Proporciona una lista de cotejo con criterios claros (claridad, secuencia lógica, uso correcto de símbolos y estructuras).
  - Los estudiantes intercambian sus algoritmos (diagramas y pseudocódigos) y aplican la lista para dar retroalimentación constructiva.
- **Organización:** Parejas/grupos pequeños para revisión cruzada.
- **Producto:** Formato de retroalimentación completado.
- **Tiempo:** 20 minutos.
- **Rol del docente:** Supervisa, guía con preguntas para mejorar la crítica constructiva y apoya en dudas.

## Diferenciación

- Para estudiantes avanzados: se les invita a proponer mejoras al algoritmo para hacerlo más eficiente.
- Para quienes requieren apoyo adicional: se les facilita ejemplos detallados y soporte individual para aplicar la lista de cotejo.

## Transición a cierre

El docente solicita que cada grupo prepare una síntesis oral sobre la técnica de representación que consideran más efectiva y por qué.

## Fase de Cierre

**Tiempo estimado: 15 minutos**

**Síntesis:**

- **Docente:** Facilita una lluvia de ideas para construir un mapa mental colectivo en la pizarra que resuma las técnicas de representación, sus ventajas y aplicaciones.
- **Estudiantes:** Participan activamente integrando conceptos.

### **Reflexión metacognitiva:**

- ¿Cómo cambió tu comprensión de los algoritmos después de representarlos gráficamente y en pseudocódigo?
- ¿Qué técnica te parece más útil para comunicar soluciones y por qué?
- ¿Qué dificultades encontraste y cómo las superaste?

### **Retroalimentación:**

El docente ofrece comentarios finales, destacando los logros y brindando sugerencias para mejorar la representación y análisis de algoritmos en futuros proyectos.

### **Transferencia:**

Se invita a los estudiantes a aplicar estas técnicas en sus proyectos de programación y a explorar algoritmos más complejos en materias avanzadas.

### **Tarea o reto:**

Diseñar y representar un algoritmo para un problema cotidiano a elección, utilizando ambas técnicas (diagrama de flujo y pseudocódigo), y preparar una breve presentación para la próxima clase.

## **Evaluación**

### **Tipo de evaluación:**

- Diagnóstica: Activación de conocimientos previos en la primera sesión (fase de inicio).
- Formativa: Observación directa y guía durante actividades de análisis, diseño y representación de algoritmos en ambas sesiones.
- Sumativa: Revisión cruzada y uso de lista de cotejo para evaluar claridad, secuencia y corrección de algoritmos en la segunda sesión (actividad 3 fase de desarrollo).

### **Criterios de evaluación:**

- Identifica correctamente los pasos y decisiones en un problema real para diseñar un algoritmo (vinculado al análisis y diseño).
- Aplica correctamente técnicas de representación gráfica y textual para describir algoritmos (vinculado a la creación de diagramas y pseudocódigo).
- Evalúa la claridad y eficiencia en algoritmos propios y de pares (vinculado a la comparación y evaluación).
- Comunica de manera clara y organizada el proceso algorítmico (vinculado a la presentación y documentación).

### **Instrumentos sugeridos:**

- Lista de cotejo para evaluación de diagramas y pseudocódigo.
- Observación directa y registro anecdótico durante actividades grupales.
- Autoevaluación y coevaluación mediante formatos breves de retroalimentación.
- Portafolio con productos elaborados (listas de pasos, diagramas de flujo, pseudocódigos).

#### **Evidencias de aprendizaje:**

- Listas secuenciales de pasos para problemas analizados.
- Diagramas de flujo diseñados en grupo.
- Pseudocódigos elaborados individual o en pareja.
- Informes comparativos sobre técnicas de representación.
- Formatos de retroalimentación completados en revisión cruzada.

## **Enriquecimientos**

### **Inicio - Contextualizar**

#### **Contextualización para la fase de inicio**

En la actualidad, vivimos inmersos en un mundo digital donde prácticamente todos los dispositivos y servicios que utilizamos diariamente dependen de algoritmos para funcionar correctamente. Desde las aplicaciones que usamos para pedir comida o transporte, hasta la gestión de nuestras finanzas personales y la interacción en redes sociales, los algoritmos están detrás de cada proceso que facilita nuestra vida cotidiana.

Como estudiantes universitarios en Ingeniería de Sistemas, es fundamental comprender que un algoritmo no es solo un concepto abstracto, sino la base lógica que permite resolver problemas complejos de manera eficiente y estructurada. Por ejemplo, al planificar una ruta óptima para desplazarse por la ciudad o al diseñar un sistema que analice grandes volúmenes de datos para la toma de decisiones, el conocimiento profundo de cómo diseñar y representar algoritmos es esencial.

En estas sesiones abordaremos los fundamentos de los algoritmos y sus técnicas de representación, habilidades que serán la piedra angular para el desarrollo de software y sistemas inteligentes. Reconocerán cómo estas herramientas les permitirán transformar ideas y problemas reales en soluciones programables, incrementando su capacidad para innovar y aportar a la sociedad.

Emocionalmente, los invitamos a adoptar una actitud de curiosidad y exploración, reconociendo que dominar algoritmos abre puertas a múltiples oportunidades profesionales y les permitirá enfrentar desafíos técnicos con confianza y creatividad.