

Clasificación de los lenguajes de programación según su nivel de abstracción

Tecnología e Informática | Tecnología

Descripción del Curso

Este curso de Tecnología, orientado a estudiantes de 15 a 16 años, está organizado en cuatro unidades para desarrollar de forma progresiva conceptos fundamentales de computación, lógica y análisis tecnológico. Cada unidad propone experiencias de aprendizaje que conectan teoría con casos prácticos, fomentando la resolución de problemas, la colaboración y la capacidad de comunicar ideas técnicas a distintos públicos. La Unidad 4, denominada “Aplicación y criterios de clasificación de lenguajes según la abstracción”, cierra la secuencia al incorporar la clasificación de lenguajes por su nivel de abstracción y su aplicación en problemas reales, con énfasis en la justificación de elecciones técnicas ante requisitos concretos. En esta unidad final, se consolidan los conocimientos sobre clasificación por abstracción y se aplican a casos prácticos. Los estudiantes evaluarán proyectos y justificarán la selección de lenguaje en función de los requisitos de abstracción, complejidad y rendimiento. El enfoque combina análisis crítico y comunicación efectiva: analizar problemas simples, proponer el lenguaje adecuado según el nivel de abstracción requerido y defender la elección ante un público no técnico. Además, se espera que elaboren un informe corto que conecte los requisitos del problema con las características del lenguaje, demostrando la capacidad de traducir necesidades prácticas en argumentos técnicos claros. El objetivo general es que los alumnos apliquen la clasificación de lenguajes por nivel de abstracción en problemas reales y comuniquen de forma clara las razones para elegir un lenguaje concreto. Las actividades favorecen el desarrollo del pensamiento crítico, la capacidad de trabajar en equipo y la habilidad de explicar conceptos técnicos de forma accesible a audiencias diversas. Al finalizar la unidad, los estudiantes deben ser capaces de justificar decisiones de lenguaje con criterios de abstracción, complejidad y rendimiento, y de presentar un informe que sintetice la conexión entre problemáticas reales y las características de los lenguajes evaluados. Este enfoque integrador busca que el alumnado internalice criterios prácticos para la toma de decisiones tecnológicas y desarrolle una comunicación técnica competente.

Competencias

- Analizar problemas simples y proponer soluciones de lenguaje adecuadas según el nivel de abstracción requerido.
- Justificar la elección de un lenguaje ante audiencias no técnicas con argumentos claros y pertinentes.
- Elaborar informes breves que conecten requisitos del problema con características de los lenguajes, de forma concisa y rigurosa.
- Trabajar colaborativamente para organizar y presentar casos de estudio sobre clasificación de lenguajes por abstracción.
- Aplicar criterios de rendimiento y complejidad para comparar soluciones y tomar decisiones informadas.

Requerimientos

- Acceso a un dispositivo con conexión a internet y herramientas básicas de procesamiento de textos y presentaciones.
- Colaboración en equipo para el análisis de casos y la preparación de presentaciones orales y escritas.
- Lecturas y casos de estudio sobre clasificación de lenguajes por niveles de abstracción y rendimiento.
- Entrega de un informe corto que justifique la decisión de lenguaje, enlazando requisitos del problema con características del lenguaje.
- Participación en discusiones y exposiciones dirigidas a una audiencia no técnica.

Unidades del Curso

Unidad 1: Unidad 1: Introducción a la clasificación y conceptos de abstracción

Objetivos de Aprendizaje

- Definir conceptos clave: lenguaje de bajo nivel, lenguaje de alto nivel, lenguaje de medio nivel, ensamblador, máquina, compilado e interpretado.
- Identificar ejemplos de código o pseudo-código que correspondan a distintos niveles de abstracción.
- Analizar las ventajas y limitaciones de cada nivel de abstracción en situaciones reales.

Contenidos Temáticos

1. **Conceptos de abstracción en informática** – Descripción corta: qué implican la reducción de complejidad y la separación entre hardware y software.
2. **Niveles de abstracción en los lenguajes de programación** – Descripción corta: desde lenguaje de máquina hasta lenguajes de alto nivel y medio nivel.
3. **Comparación de ejemplos simples en distintos niveles** – Descripción corta: análisis de fragmentos de código para identificar nivel de abstracción.

Actividades

- **Actividad 1: Mapa de abstracciones** – En parejas, diseñarán un diagrama que represente una tarea cotidiana codificada en tres niveles de abstracción (día a día, pseudocódigo, código simple). Presentarán las ideas clave y discutirán por qué cada nivel facilita o dificulta el entendimiento y la ejecución.
- **Actividad 2: Debate corto** – Discusión guiada sobre cuál nivel de abstracción es más adecuado para diferentes proyectos (juegos, apps móviles, sistemas embebidos) y qué criterios guían la decisión. Se destacarán ventajas y limitaciones de cada enfoque.

Evaluación

La evaluación de la unidad se centra en la comprensión de conceptos y la capacidad de clasificación. Se emplearán:

- Participación y aportes en las actividades en clase (30%).
- Ejercicio de clasificación: seleccionar el nivel de abstracción adecuado para tres escenarios y justificar la elección (40%).
- Cuestionario corto de 5 preguntas sobre conceptos clave (30%).

Unidad 2: Unidad 2: Lenguajes de bajo nivel y ensamblador

Objetivos de Aprendizaje

- Definir lenguaje de máquina y ensamblador y explicar su relación con la CPU y la arquitectura del equipo.
- Explicar el proceso de ensamblaje: código en ensamblador se traduce a código de máquina ejecutable.
- Identificar ejemplos simples de instrucciones en ensamblador y su equivalente conceptual en código de máquina.

Contenidos Temáticos

1. **Lenguaje de máquina** – Descripción corta: instrucciones binarias que la CPU entiende directamente.
2. **Ensamblador** – Descripción corta: lenguaje simbólico que facilita escribir instrucciones para la CPU, con mnemónos y direcciones.
3. **Arquitectura de la computadora y operaciones básicas** – Descripción corta: registros, memoria y flujo de instrucciones.

Actividades

- **Actividad 1: Construyendo un mini-ensamblador** – Los estudiantes escribirán un par de instrucciones simples en lenguaje ensamblador (p. ej., MOV, ADD) y, mediante una simulación guiada, asociarán cada instrucción con su código de máquina conceptual y el efecto en registros. Se enfatizarán los conceptos de lectura de código y traducción a máquina.
- **Actividad 2: Simulación de flujo de instrucciones** – En grupos, crearán una pequeña simulación que represente un conjunto de instrucciones moviéndose entre registro y memoria, discutiendo el rol del ensamblador y la necesidad de traducción para la ejecución.

Evaluación

Evaluación centrada en la comprensión de conceptos y la capacidad de aplicar la idea de ensamblaje. Se incluyen:

- Cuestionario corto de identificación de conceptos (25%).
- Actividad práctica de ensamblaje y correspondencia (35%).
- Resumen escrito explicando la relación entre código en ensamblador y código de máquina (40%).

Unidad 3: Unidad 3: Lenguajes de alto y medio nivel, compilación e interpretación

Objetivos de Aprendizaje

- Definir características de lenguajes de alto nivel y medio nivel, y ejemplos representativos.
- Explicar qué es compilación y qué es interpretación, con ejemplos de lenguajes comunes.
- Analizar ventajas y desventajas de lenguajes compilados vs interpretados para proyectos específicos.

Contenidos Temáticos

1. **Lenguajes de alto nivel** – Descripción corta: sintaxis cercana al lenguaje humano, abstracciones como estructuras de control y manejo de memoria automático.
2. **Lenguajes de medio nivel** – Descripción corta: equilibrio entre control de la máquina y abstracciones, ejemplos como C, que permiten manipulación de memoria y rendimiento.
3. **Compilación vs interpretación** – Descripción corta: procesos y herramientas (compiladores, intérpretes, entornos de ejecución).
4. **Herramientas y conceptos asociados** – Descripción corta: compiladores, enlazadores, máquinas virtuales, entornos de desarrollo.

Actividades

- **Actividad 1: Comparando código en C y Python** – Se presentarán dos tareas equivalentes (por ejemplo, imprimir números en un rango). Los estudiantes analizarán diferencias en sintaxis, rendimiento, manejo de memoria y legibilidad, destacando conceptos de alto y medio nivel y la idea de compilación vs interpretación.
- **Actividad 2: Experimento de compilación** – Con un pequeño programa, los estudiantes simularán el proceso de compilación, observando fases como análisis léxico, sintáctico y generación de código, para entender cómo se transforma el código fuente en ejecución.

Evaluación

La evaluación abordará los objetivos de la unidad mediante:

- Prueba corta sobre conceptos de alto/medio nivel y compilación/interpretación (30%).
- Actividad de comparación de lenguajes y justificación de elección en un escenario (40%).
- Informe breve de reflexión sobre cuándo usar un lenguaje compilado frente a uno interpretado (30%).

Unidad 4: Unidad 4: Aplicación y criterios de clasificación de lenguajes según la abstracción

Objetivos de Aprendizaje

- Analizar un conjunto de problemas simples y proponer el lenguaje adecuado según el nivel de abstracción requerido.
- Describir y defender la elección de lenguaje ante un público no técnico.

- Elaborar un informe corto que justifique la decisión de lenguaje, conectando requisitos del problema con características del lenguaje.

Contenidos Temáticos

1. **Criterios de selección de lenguajes según el problema** – Descripción corta: rendimiento, portabilidad, facilidad de desarrollo y mantenimiento.
2. **Procedimiento de evaluación de lenguajes** – Descripción corta: análisis de requisitos, pruebas de prototipos y estimación de recursos.
3. **Presentación de resultados** – Descripción corta: comunicación clara de decisiones técnicas a distintos públicos.

Actividades

- **Actividad 1: Caso práctico de clasificación** – En grupos, se presentarán tres escenarios de problemas y se deberá proponer el lenguaje más adecuado, explicando la elección y sus beneficios en términos de abstracción.
- **Actividad 2: Presentación de soluciones** – Cada grupo presentará su caso y defenderá su decisión ante la clase, respondiendo a preguntas y comparando enfoques distintos.

Evaluación

Evaluación final basada en el razonamiento y la claridad de la justificación. Se evaluarán:

- Informe escrito con criterios de selección y justificación (40%).
- Presentación oral y respuesta a preguntas (40%).
- Autoevaluación y coevaluación de pares (20%).