

Introducción a Java y entorno de desarrollo

Tecnología e Informática | Informática

Descripción del Curso

El curso de Informática está diseñado para estudiantes de 15 a 16 años, con el objetivo de desarrollar una base sólida en pensamiento computacional, resolución de problemas y habilidades prácticas de programación. A lo largo de las unidades, los alumnos avanzarán desde conceptos fundamentales como lógica, algoritmos y estructuras de control hasta la escritura de código claro, documentado y mantenible, acompañado de prácticas de depuración. Se fomenta el uso responsable de la tecnología, la creatividad y la capacidad de trabajar de forma colaborativa para plantear soluciones computacionales a situaciones reales. En la Unidad 8, Comentarios y depuración: escribir comentarios y leer/interpretar mensajes de error, se enfatiza la importancia de los comentarios claros en el código y la habilidad de leer e interpretar mensajes de error del compilador para corregir fallos y mejorar la legibilidad del programa. Esta unidad final consolida las prácticas de documentación, la interpretación de mensajes de error y las estrategias de depuración como componentes esenciales de un desarrollo de software responsable. Además, se refuerzan prácticas de legibilidad, mantenimiento y validación de soluciones mediante pruebas simples y revisión entre pares. La metodología combina explicaciones breves, ejercicios guiados y proyectos cortos que permiten aplicar lo aprendido en contextos cercanos a la vida diaria. Se busca potenciar la autonomía del estudiante, su capacidad de comunicar ideas técnicas y su competencia para trabajar en equipo, analizando problemas, proponiendo soluciones y justificando elecciones de diseño. El curso pretende formar ciudadanos digitales críticos, capaces de adaptarse a entornos tecnológicos en constante cambio y de continuar aprendiendo de forma constructiva a lo largo de su trayectoria educativa.

Competencias

- Comprender fundamentos de informática, lógica de programación y estructuras de control.
- Escribir código claro y mantenible apoyado en comentarios efectivos y documentados.
- Leer, interpretar y actuar sobre mensajes de error del compilador y del entorno de desarrollo para depurar.
- Aplicar buenas prácticas de legibilidad, mantenimiento y documentación de software.
- Analizar problemas, diseñar soluciones algorítmicas y verificarlas con pruebas adecuadas.
- Comunicar ideas técnicas de forma efectiva y colaborar en equipos de trabajo.

Requerimientos

- Dispositivo compatible (PC, portátil, o tablet) con acceso a un entorno de desarrollo y conexión a Internet.
- Conocimientos previos en lógica, algoritmos básicos y estructuras de control de flujo.
- Acceso a un editor de código y herramientas de depuración comunes, con permisos para instalar o usar versiones en línea.

- Compromiso de participación activa, realización de ejercicios prácticos y entrega de tareas en fechas establecidas.
- Materiales de lectura y recursos prácticos proporcionados por el curso para apoyar la práctica guiada.

Unidades del Curso

Unidad 1: Unidad 1: Conceptos clave de Java y entorno de desarrollo

Objetivos de Aprendizaje

- Definir qué es Java y cuál es su función en la programación.
- Distinguir entre JDK y JRE, y explicar qué es un IDE y la consola de desarrollo.
- Identificar conceptos de clase, objeto y método dentro de ejemplos simples.

Contenidos Temáticos

1. **Conceptos clave de Java y entorno de desarrollo:** explicación de JDK, JRE, IDE, consola, clase, objeto y método, con ejemplos simples de uso.
2. **Entorno de desarrollo y consola:** diferencias entre herramientas, instalación básica y roles en la programación.
3. **Estructura general de un programa Java:** declaración de clase, método main y llaves, conceptos de declaración y alcance.
4. **Compilación y ejecución:** cómo compilar y ejecutar programas desde consola y desde un IDE.

Actividades

1. **Actividad: Exploración guiada de herramientas** – En parejas, identificarán y describirán para qué sirve cada herramienta (JDK, JRE, IDE, consola) y cómo se relacionan entre sí. Se registrarán ejemplos prácticos de cuándo usar cada una y se discutirán errores comunes y soluciones básicas.
2. **Actividad: Conceptos en acción** – Se presentarán ejemplos cortos de clase, objeto y método; los estudiantes etiquetarán cada elemento y explicarán su función usando un diagrama simple.
3. **Actividad: Visualización de la consola** – Se ejecutarán comandos simples para observar el comportamiento de la consola (sin escribir código complejo). Los alumnos tomarán notas sobre qué información aparece y cómo interpretarla.

Evaluación

- Preguntas cortas sobre conceptos clave (JDK/JRE, IDE, consola, clase, objeto, método).
- Participación y aportes en las actividades prácticas de la unidad.
- Actividad de autoevaluación sobre la comprensión de herramientas y terminología.

Unidad 2: Unidad 2: Escribir un programa Java básico que imprima

Objetivos de Aprendizaje

- Usar la sentencia `System.out.println` para imprimir texto en la consola.
- Configurar un archivo Java en un proyecto y ejecutarlo desde el IDE.
- Identificar la salida en consola y relacionarla con el código escrito.

Contenidos Temáticos

1. **Estructura mínima de un programa Java:** clase, método main y declaración de la clase.
2. **Salida por consola:** uso de `System.out.println` para imprimir mensajes.
3. **Compilación y ejecución desde el IDE:** pasos básicos para ver la salida en pantalla.

Actividades

1. **Actividad: Escribe y ejecuta Hello World** – Crear un proyecto, añadir una clase con un método main y imprimir el texto "Hola, mundo" en la consola. Observar la salida y verificar que no hay errores de compilación.
2. **Actividad: Imprime mensajes adicionales** – Ampliar el programa para imprimir dos o tres líneas de texto diferentes, comprobando el orden de las salidas.
3. **Actividad: Explora la salida** – Cambiar el mensaje y revisar cómo se ve en la consola, discutiendo cómo funciona la impresión en Java.

Evaluación

- Ejecutable final que imprime el texto esperado sin errores de compilación.
- Explicación breve de qué hace cada línea que imprime y por qué.

Unidad 3: Unidad 3: Estructura básica de un programa Java: clase, método main, llaves, declaraciones y comentarios

Objetivos de Aprendizaje

- Identificar la estructura de una clase y el método main dentro de un programa Java.
- Reconocer el uso de llaves `{ }` para delimitar bloques de código y la organización visual del programa.
- Utilizar comentarios para documentar el código y facilitar su lectura.

Contenidos Temáticos

1. **Clase y método main:** cómo definir una clase y dónde se ubica el método de entrada del programa.
2. **Bloques y llaves:** uso correcto de llaves para delimitar clases, métodos y estructuras de control.
3. **:** declarar variables simples y comentar el código para facilitar su comprensión.

Actividades

1. **Actividad: Esqueleto de clase** – Escribir un programa con una clase simple y un método main que contenga comentarios que expliquen cada parte del código.
2. **Actividad: Comentarios útiles** – Añadir comentarios claros a un código dado para describir la finalidad de cada sección y las decisiones tomadas.
3. **Actividad: Lectura de código** – Analizar un fragmento de código y explicar qué sucede en cada bloque, señalando posibles mejoras de legibilidad.

Evaluación

- Archivo con estructura adecuada (clase, main, llaves) y comentarios claros.
- Capacidad para explicar la finalidad de cada parte del código durante la revisión.

Unidad 4: Unidad 4: Instalar y configurar un entorno de desarrollo (IDE) y el JDK

Objetivos de Aprendizaje

- Descargar e instalar el JDK adecuado para el sistema operativo.
- Instalar y configurar un IDE y crear un proyecto nuevo básico.
- Configurar variables de entorno (JAVA_HOME y PATH) y verificar la instalación desde la consola.

Contenidos Temáticos

1. **Descarga e instalación del JDK:** versión adecuada, pasos básicos y verificación de instalación.
2. **Instalación del IDE:** elección de IDE y pasos para la primera ejecución.
3. **Configuración de entorno:** variables JAVA_HOME y PATH, verificación de versión desde la consola.

Actividades

1. **Actividad: Instalación del JDK y verificación** – Descargar e instalar el JDK, abrir la consola y ejecutar `java -version` y `javac -version` para verificar la instalación.
2. **Actividad: Instalación del IDE y primer proyecto** – Instalar un IDE, crear un proyecto Java simple y explorar la interfaz (barra de herramientas, explorador de archivos, consola integrada).
3. **Actividad: Configuración de entorno** – Configurar JAVA_HOME y actualizar PATH, luego crear y compilar un programa simple desde la consola o IDE.

Evaluación

- Comprobación de la instalación: ejecución de un programa simple desde consola/IDE sin errores.
- Capacidad para localizar y corregir problemas de configuración básica (PATH, JAVA_HOME).

Unidad 5: Unidad 5: Crear un proyecto Java en un IDE, añadir una clase con un método main y compilar para ejecutar

Objetivos de Aprendizaje

- Crear un proyecto Java en el IDE seleccionado y añadir una clase con main.
- Compilar y ejecutar el programa para ver la salida en la consola.
- Identificar y corregir errores de compilación básicos que aparezcan.

Contenidos Temáticos

1. **Creación de proyectos en IDE:** estructura de un proyecto, archivos y carpetas comunes.
2. **Clase con método main:** implementación de una clase con punto de entrada.
3. **Compilar y ejecutar:** pasos para ver la salida y resolver errores simples.

Actividades

1. **Actividad: Proyecto HelloWorld** – Crear un proyecto, añadir una clase con main y escribir un mensaje en la consola; compilar y ejecutar para verificar la salida.
2. **Actividad: Ampliar la clase** – Añadir una segunda clase con su propio main para comparar resultados y entender la organización de paquetes.
3. **Actividad: Depuración básica** – Introducir un error intencional (por ejemplo, un error de sintaxis) y practicar su localización y corrección a partir del mensaje del compilador.

Evaluación

- Proyecto funcional que compila y ejecuta correctamente dentro del IDE.
- Informe breve de la estructura del proyecto y de la salida esperada.

Unidad 6: Unidad 6: Declarar y usar variables de tipos primitivos y String

Objetivos de Aprendizaje

- Declarar variables de los tipos int, double y boolean, y asignarles valores adecuados.
- Trabajar con la clase String y concatenación simple para generar salidas más completas.
- Imprimir en consola los valores de estas variables para verificar el almacenamiento y la visualización.

Contenidos Temáticos

1. **Tipados primitivos:** int, double y boolean, sus rangos y usos básicos.
2. **Clase String y concatenación:** manejo de textos y unión de cadenas con números.
3. **Salida por consola:** imprimir variables y textos combinados en una línea o varias.

Actividades

1. **Actividad: Perfil simple** – Declarar variables para nombre (String), edad (int), altura (double) y estado (boolean); imprimir un perfil básico en consola concatenando texto y variables.
2. **Actividad: Operaciones simples** – Realizar una suma con enteros y mostrar el resultado junto con una explicación textual.
3. **Actividad: Formateo de salida** – Practicar diferentes formas de imprimir texto y variables para mejorar la legibilidad de la salida.

Evaluación

- Programa que declare correctamente los tipos solicitados y muestre la salida esperada.
- Explicación breve de qué muestra cada línea de imprimir y por qué.

Unidad 7: Unidad 7: Estructuras de control básicas: if/else y bucles (for/while)

Objetivos de Aprendizaje

- Escribir y usar condicionales if/else para tomar decisiones en el código.
- Utilizar bucles for y while para repetir acciones de forma controlada.
- Resolver un problema práctico sencillo que requiera decisiones y/o repeticiones.

Contenidos Temáticos

1. **Estructuras condicionales:** if, else, operadores de comparación y lógica básica.
2. **Bucle for:** sintaxis y uso para iteraciones fijas.
3. **Bucle while:** condición de continuación y uso en casos dinámicos.
4. **Aplicación de un problema sencillo:** clasificación o conteo utilizando if/else y bucles.

Actividades

1. **Actividad: Clasificación con if/else** – Escribir un programa que clasifique un número como positivo, negativo o cero usando if/else.
2. **Actividad: Conteo con for** – Usar un bucle for para imprimir los números del 1 al 10.
3. **Actividad: Par/impar con while** – Implementar un bucle while que verifique si un número es par o impar y muestre el resultado.

Evaluación

- Programa que resuelva correctamente el problema propuesto usando if/else y bucles.
- Justificación breve de la decisión tomada en el código.

Unidad 8: Unidad 8: Comentarios y depuración: escribir comentarios y leer/interpretar mensajes de error

Objetivos de Aprendizaje

- Utilizar comentarios de línea y de bloque para documentar ideas, supuestos y decisiones de diseño.
- Leer, interpretar y actuar sobre mensajes de error del compilador y del entorno de desarrollo.
- Aplicar buenas prácticas de legibilidad y mantenimiento del código.

Contenidos Temáticos

1. **Comentarios en Java:** tipos de comentarios (`//`, `/* */`) y buenas prácticas.
2. **Lectura de errores:** interpretación de mensajes de compilación y stack traces básicos.
3. **Depuración y legibilidad:** estrategias simples de depuración y organización de código.

Actividades

1. **Actividad: Añadir comentarios explicativos** – Tomar un fragmento de código sin comentarios y añadir explicaciones claras en líneas apropiadas.
2. **Actividad: Interpretación de errores** – Proporcionar fragmentos con errores intencionados y revisar los mensajes del compilador para proponer correcciones.
3. **Actividad: Mejora de la legibilidad** – Reorganizar código para mejorar sangría, nombres de variables y comentarios, y justificar las mejoras.

Evaluación

- Proyecto corto con comentarios útiles y código legible.
- Informe breve explicando tres errores comunes y cómo se resolvieron a partir de los mensajes del compilador.