

# Informática III



# Clase 1 de Informática III



# FUNDAMENTOS DE JAVA

Introducción a los fundamentos y objetivos del aprendizaje en JAVA.



## IMPORTANCIA DE JAVA

Se discutirá por qué aprender JAVA es fundamental en el ámbito de la programación moderna, resaltando su relevancia en el mercado laboral y en proyectos de desarrollo.



## OBJETIVOS DEL CURSO

El curso tiene como objetivo proporcionar una base sólida en JAVA, permitiendo a los estudiantes desarrollar una comprensión profunda del lenguaje y sus aplicaciones.



## EXPECTATIVAS DE APRENDIZAJE

Los alumnos podrán establecer expectativas sobre su propio aprendizaje en JAVA, incluidos los conocimientos que se espera adquirir al final del curso.



## Temas actuales "IA"



# IMPORTANCIA DE JAVA

Se discutirá por qué aprender JAVA es fundamental en el ámbito de la programación moderna, resaltando su relevancia en el mercado laboral y en proyectos de desarrollo.



# OBJETIVOS DEL CURSO

El curso tiene como objetivo proporcionar una base sólida en JAVA, permitiendo a los estudiantes desarrollar una comprensión profunda del lenguaje y sus aplicaciones.



# EXPECTATIVAS DE APRENDIZAJE

Los alumnos podrán establecer expectativas sobre su propio aprendizaje en JAVA, incluidos los conocimientos que se espera adquirir al final del curso.



# Temas actuales

## "IA"

# FUNDAMENTOS DE JAVA

Introducción a los fundamentos y objetivos del aprendizaje en JAVA.



## IMPORTANCIA DE JAVA

Se discutirá por qué aprender JAVA es fundamental en el ámbito de la programación moderna, resaltando su relevancia en el mercado laboral y en proyectos de desarrollo.



## OBJETIVOS DEL CURSO

El curso tiene como objetivo proporcionar una base sólida en JAVA, permitiendo a los estudiantes desarrollar una comprensión profunda del lenguaje y sus aplicaciones.



## EXPECTATIVAS DE APRENDIZAJE

Los alumnos podrán establecer expectativas sobre su propio aprendizaje en JAVA, incluidos los conocimientos que se espera adquirir al final del curso.



## Temas actuales "IA"

# Clase 1 de Informática III



## FUNDAMENTOS DE JAVA

Introducción a los fundamentos y objetivos del aprendizaje en JAVA.



IMPORTANCIA  
DE JAVA



OBJETIVOS  
DEL CURSO



EXPECTATIVAS  
DE APRENDIZAJE

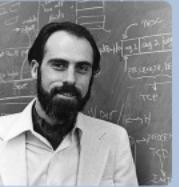


Temas actuales

# Introducción Personal

## Perfil Profesional

Ingeniero en informática con experiencia en enseñanza y desarrollo de software. Me dedico a fomentar el aprendizaje y la colaboración en el aula.



## Colaboración

La colaboración es clave en mi metodología. Fomento un ambiente donde los estudiantes pueden expresarse, hacer preguntas y compartir ideas.



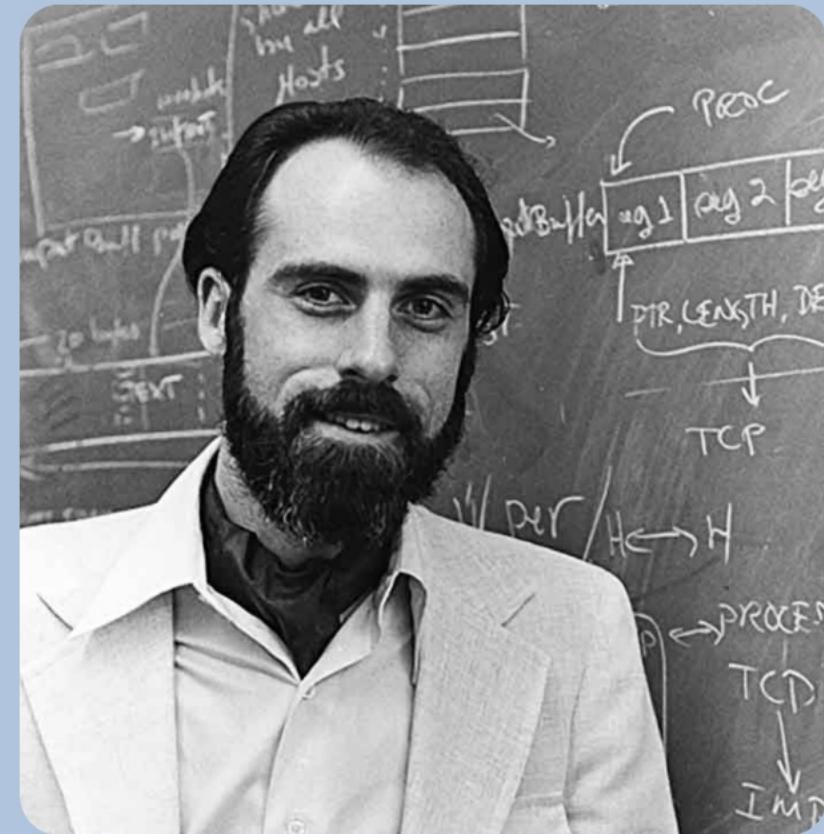
## Aprendizaje Continuo

La enseñanza es para mí una vía de aprendizaje continuo. Siempre busco nuevas formas de mejorar y adaptarme a las necesidades de mis estudiantes.



# Perfil Profesional

Ingeniero en informática con experiencia en enseñanza y desarrollo de software. Me dedico a fomentar el aprendizaje y la colaboración en el aula.



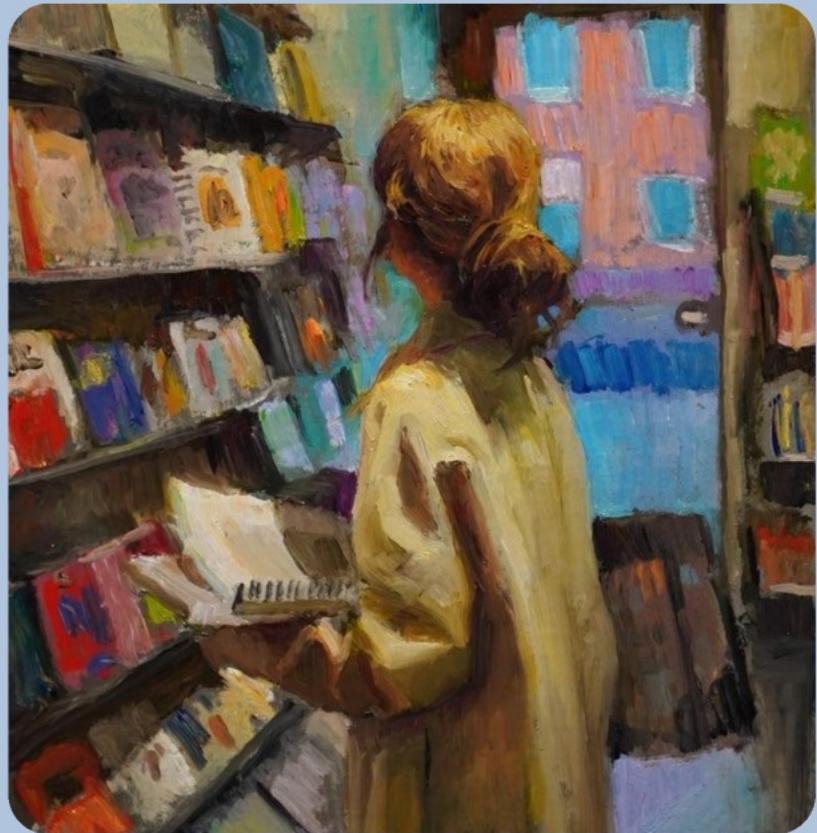
# Colaboración

La colaboración es clave en mi metodología. Fomento un ambiente donde los estudiantes pueden expresarse, hacer preguntas y compartir ideas.



# Aprendizaje Continuo

La enseñanza es para mí una vía de aprendizaje continuo. Siempre busco nuevas formas de mejorar y adaptarme a las necesidades de mis estudiantes.



# Objetivos de la Materia



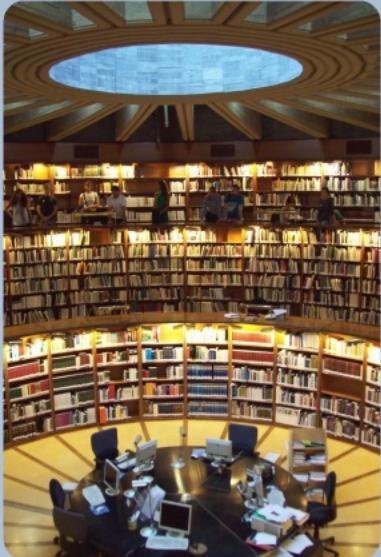
### Desarrollar en JAVA

Desarrollar habilidades en programación mediante la implementación de proyectos en JAVA, lo que permitirá a los estudiantes aplicar lo aprendido de manera práctica.



### Comprender los prácticos

Comprender los prácticos es clave para aplicar la teoría a situaciones reales y fortalecer el aprendizaje.



### Autogestión

Fomentar la capacidad de autogestión para que los estudiantes tomen control de su propio aprendizaje y establezcan metas.



### Trabajo en equipo

El trabajo en equipo es esencial para el desarrollo de proyectos, fomentando la colaboración y el intercambio de ideas.



### Planificación y cumplimiento de plazos

Planificar y cumplir con los tiempos de entrega ayudará a los estudiantes a desarrollar habilidades de gestión del tiempo, cruciales en el ámbito profesional.

# Primer Parcial Teórico

El primer parcial teórico se centrará en los conceptos fundamentales de Java, incluyendo teorías y prácticas esenciales que los estudiantes deben dominar para avanzar en el curso.



# Segundo Parcial - Proyecto Integrador

El segundo parcial consistirá en un proyecto integrador donde los alumnos aplicarán lo aprendido en un trabajo práctico, guiados por el Ing. Alejandro Silvestre, lo que permitirá fomentar el trabajo en equipo y la creatividad.



# Introducción a JAVA

Java, creado por James Gosling en 1995, se ha consolidado como uno de los lenguajes de programación más importantes y utilizados a nivel mundial. Su versatilidad le permite ser utilizado en desarrollo web, aplicaciones móviles, sistemas empresariales y más. Aprender Java es esencial para los desarrolladores debido a su robustez, seguridad y amplia comunidad de soporte.



## Portable

Java es un lenguaje de programación que se puede ejecutar en diferentes plataformas sin necesidad de recompilación, lo que lo hace altamente portable.

## Orientado a Objetos

Al estar diseñado en torno a la programación orientada a objetos, Java permite una mejor organización del código, facilitando la reutilización y mantenimiento.

## Seguro

Java incluye características de seguridad que protegen contra ataques y acceso no autorizado, lo que lo convierte en una opción confiable para el desarrollo de aplicaciones.

## Alto Rendimiento

Java es conocido por su rendimiento eficiente, siendo capaz de manejar grandes volúmenes de datos y operaciones a alta velocidad.

## Amplia Documentación y Librerías

La extensa documentación y las numerosas librerías disponibles facilitan el desarrollo, permitiendo a los desarrolladores implementar funciones complejas con facilidad.



# Características Clave de Java

# Portable

Java es un lenguaje de programación que se puede ejecutar en diferentes plataformas sin necesidad de recompilación, lo que lo hace altamente portable.

# Orientado a Objetos

Al estar diseñado en torno a la programación orientada a objetos, Java permite una mejor organización del código, facilitando la reutilización y mantenimiento.

# Seguro

Java incluye características de seguridad que protegen contra ataques y acceso no autorizado, lo que lo convierte en una opción confiable para el desarrollo de aplicaciones.

# Alto Rendimiento

Java es conocido por su rendimiento eficiente, siendo capaz de manejar grandes volúmenes de datos y operaciones a alta velocidad.

# Amplia Documentación y Librerías

La extensa documentación y las numerosas librerías disponibles facilitan el desarrollo, permitiendo a los desarrolladores implementar funciones complejas con facilidad.

# Configuración del Ambiente de Desarrollo





## Requerimiento de JDK v21

Es esencial contar con JDK v21 para desarrollar aplicaciones en Java, ya que proporciona las herramientas necesarias para la compilación y ejecución.



## IDE recomendados

Los IDEs como Eclipse, IntelliJ, NetBeans, Visual Studio y Cursor ofrecen entornos integrados que facilitan la escritura, depuración y ejecución del código Java.

# Configuración del Ambiente de Desarrollo



## Requerimiento de JDK v21

Es esencial contar con JDK v21 para desarrollar aplicaciones en Java, ya que proporciona las herramientas necesarias para la compilación y ejecución.



## IDE recomendados

Los IDEs como Eclipse, IntelliJ, NetBeans, Visual Studio y Cursor ofrecen entornos integrados que facilitan la escritura, depuración y ejecución del código Java.

# Sintaxis Básica en Java

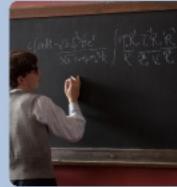
La sintaxis de un programa en Java se compone de varios elementos fundamentales. La estructura principal comienza con el método 'main', que es el punto de entrada del programa. Es crucial declarar clases, métodos y variables correctamente, asegurándose de utilizar los signos de puntuación como puntos y coma y llaves para definir bloques de código, además de agregar comentarios para mejorar la legibilidad.



# Clases y Objetos

## Clases como Plantillas

Las clases son plantillas que definen las propiedades y comportamientos de los objetos en programación. Cada clase puede tener atributos y métodos que operan sobre esos atributos.



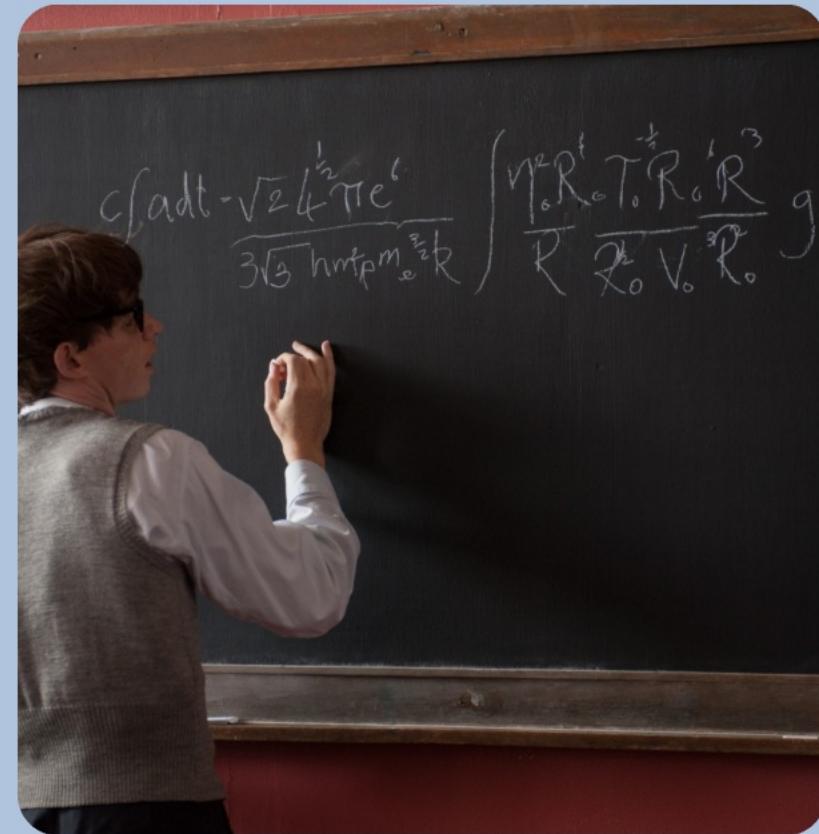
## Objetos como Instancias

Un objeto es una instancia concreta de una clase. Cada objeto puede tener un estado único representado por sus atributos, así como comportamientos definidos por sus métodos.



# Clases como Plantillas

Las clases son plantillas que definen las propiedades y comportamientos de los objetos en programación. Cada clase puede tener atributos y métodos que operan sobre esos atributos.



# Objetos como Instancias

Un objeto es una instancia concreta de una clase. Cada objeto puede tener un estado único representado por sus atributos, así como comportamientos definidos por sus métodos.



# Métodos en Java



G.Vyšniauskaitė

## Organización del Código

Los métodos permiten organizar y estructurar el código, facilitando su mantenimiento y comprensión. Además, promueven la reutilización del código en diferentes partes del programa.



## Parámetros Flexibles

Los métodos pueden tener parámetros que permiten pasar información a la función, lo que la hace más flexible y reutilizable en distintos contextos.



## Valores Retornados

El valor returnedo por un método puede ser de cualquier tipo de dato, lo que permite utilizar el resultado en otras partes del programa, aumentando su funcionalidad.

## Comentarios de Una Línea

Los comentarios de una linea se utilizan para agregar notas breves y concisas, iniciando con '//' seguido del texto del comentario. Son útiles para aclarar el propósito de líneas específicas de código.

## Comentarios de Múltiples Líneas

Los comentarios de múltiples líneas se encierran entre '/\*' y '\*/', permitiendo escribir notas más extensas. Se utilizan para explicar secciones completas de código o para desactivar partes del código temporalmente.

## Comentarios de Documentación

Los comentarios de documentación, que se escriben con '/\*\*' y '\*/', son utilizados para generar documentación automática del código. Son esenciales para describir clases, métodos y atributos, facilitando la comprensión del código por otros desarrolladores.



# Importancia de los Comentarios en el Código

# Comentarios de Una Línea

Los comentarios de una línea se utilizan para agregar notas breves y concisas, iniciando con '//' seguido del texto del comentario. Son útiles para aclarar el propósito de líneas específicas de código.

# Comentarios de Múltiples Líneas

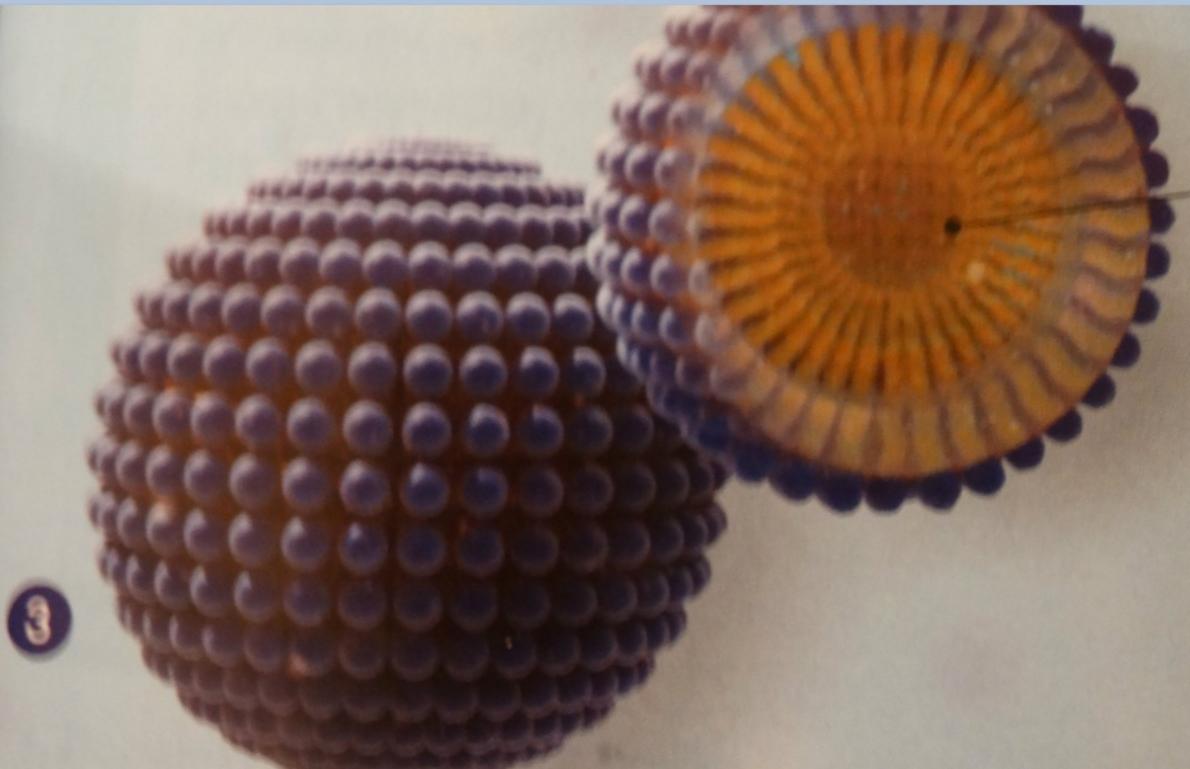
Los comentarios de múltiples líneas se encierran entre '/\*' y '\*/', permitiendo escribir notas más extensas. Se utilizan para explicar secciones completas de código o para desactivar partes del código temporalmente.

# Comentarios de Documentación

Los comentarios de documentación, que se escriben con '/\*' y '\*/', son utilizados para generar documentación automática del código. Son esenciales para describir clases, métodos y atributos, facilitando la comprensión del código por otros desarrolladores.

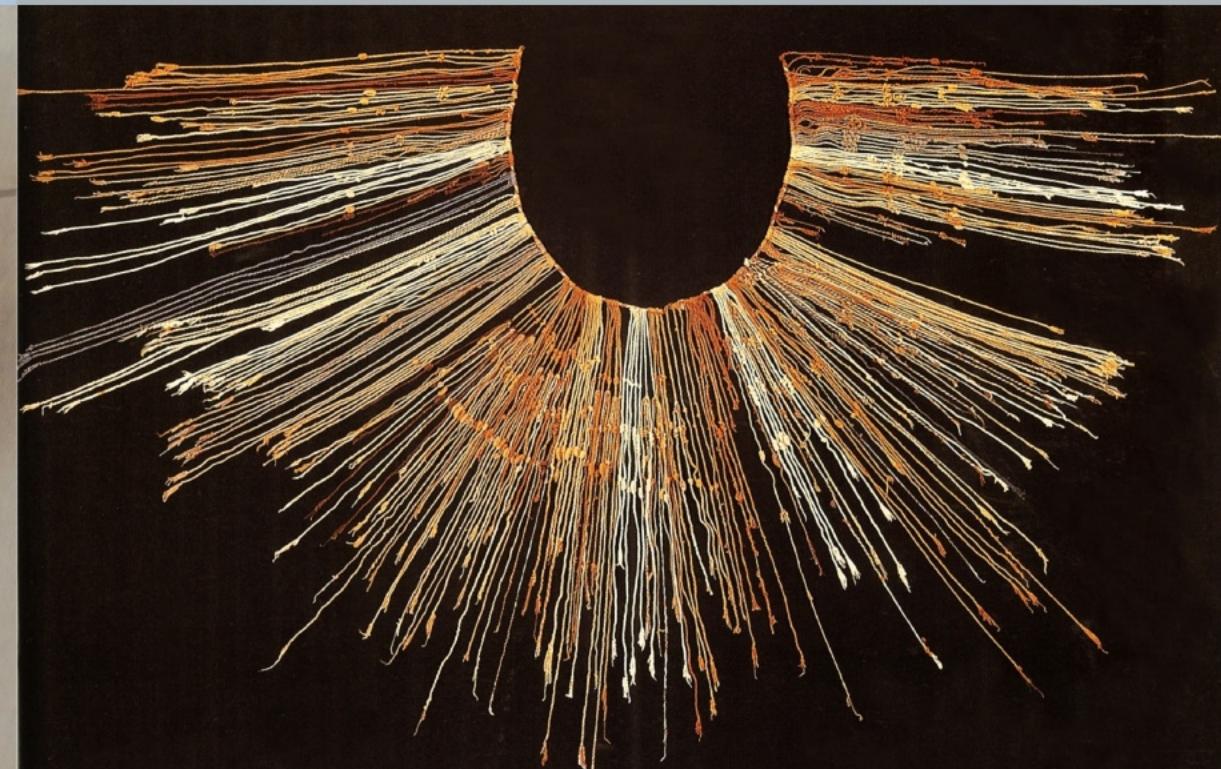
# Tipos de Datos Primitivos

Los tipos de datos primitivos en Java son fundamentales para el almacenamiento de información y operan en la memoria directamente. Incluyen: byte, short, int, long, float, double, char y boolean. Estos tipos son eficaces para realizar operaciones matemáticas y lógicas debido a su naturaleza simple y directa.

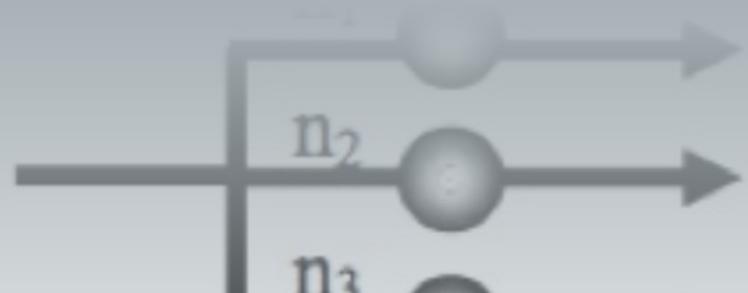


# Tipos de Datos de Referencia

Por otro lado, los tipos de datos de referencia son más complejos y permiten almacenar instancias de objetos. Incluyen String, arrays y clases personalizadas. Estos tipos son esenciales para estructurar datos más complejos y trabajar con colecciones de información en un programa.



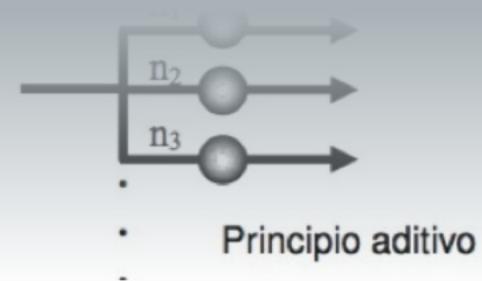
# Operadores Aritméticos en Java



# Operadores Aritméticos en Java



Principio multiplicativo



Principio aditivo

25

Los operadores aritméticos en Java son fundamentales para realizar cálculos matemáticos. Estos operadores incluyen la suma (+), resta (-), multiplicación (\*), división (/) y módulo (%), los cuales permiten manipular y combinar valores numéricos de diversas maneras.

miprofeJorge

El Metro - múltiplos y submúltiplos

EJEMPLO 1

Convertir 24 hm a cm

10000

$$24 \text{ hm} = 24 \times 10000 \text{ cm} = 240000 \text{ cm}$$

EJEMPLO 2

Convertir 4,7 km a m

1000

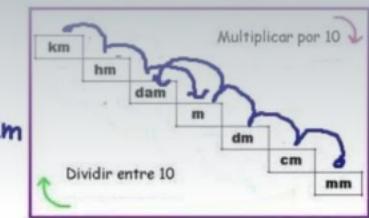
$$4,7 \text{ Km} = 4,7 \times 1000 \text{ m} = 4700 \text{ m}$$

EJEMPLO 3

Convertir 5400 mm a dam

÷ 10000

$$5400 \text{ mm} = 5400 \div 10000 \text{ dam} = 0,54 \text{ dam}$$



2010年10月31日在内蒙古呼和浩特进行教师培训。



### Operadores de Asignación en Java

Los operadores de asignación en Java son utilizados para asignar valores a las variables de manera sencilla y eficiente. Existen varios operadores que permiten realizar operaciones matemáticas y asignar el resultado a una variable en una sola línea, facilitando así la escritura del código.



### Tipos de Operadores de Asignación

Algunos de los operadores de asignación más comunes incluyen: '=' para asignar un valor; '+='; '-='; '\*='; '/='; y '%='; que permiten realizar la operación correspondiente y asignar el resultado a la variable en una sola instrucción.



# Operadores de Asignación

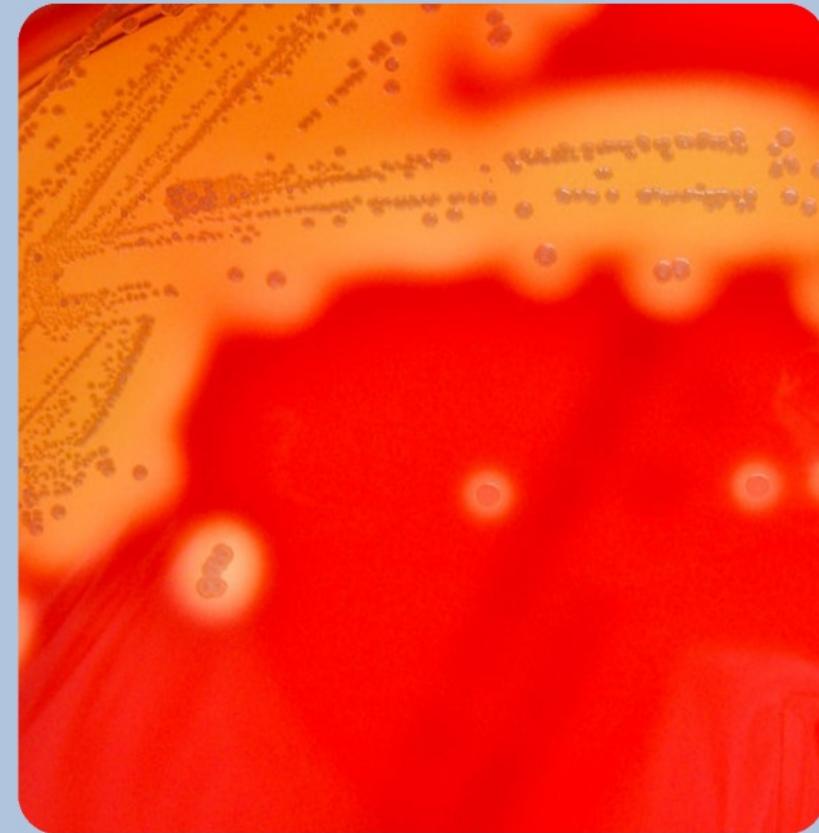
# Operadores de Asignación en Java

Los operadores de asignación en Java son utilizados para asignar valores a las variables de manera sencilla y eficiente. Existen varios operadores que permiten realizar operaciones matemáticas y asignar el resultado a una variable en una sola línea, facilitando así la escritura del código.



# Tipos de Operadores de Asignación

Algunos de los operadores de asignación más comunes incluyen: '=' para asignar un valor, '+=','-=','\*=','/=' y '%=' , que permiten realizar la operación correspondiente y asignar el resultado a la variable en una sola instrucción.



# Operadores de Comparación



## Importancia de los Operadores

Los operadores de comparación son fundamentales en la programación, ya que permiten tomar decisiones basadas en condiciones específicas.



## Tipos de Operadores de Comparación

Estos operadores incluyen '`==`', '`!=`', '`<`', '`>`', '`<=`', y '`>=`', que se utilizan para comparar valores en expresiones lógicas.

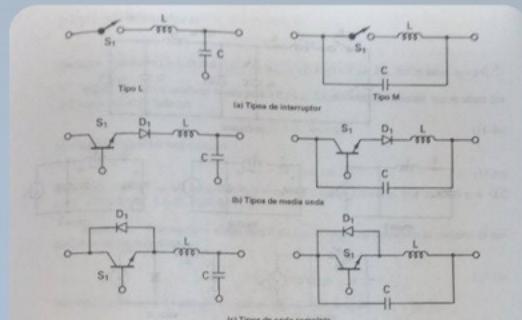


Figura 11-25 Configuraciones de interruptor para los convertidores resonantes ZCS.

Definiremos el origen del tiempo,  $t = 0$ , al principio de cada uno de los modos.

**Modo 1.** Este modo es válido para  $0 \leq t \leq t_1$ . El interruptor  $S_1$  se activa y conduce el diodo  $D_m$ . La corriente del inductor  $i_L$ , que se eleva en forma lineal, está dada por

$$i_L = \frac{V_s}{L} t \quad (11-47)$$

Este modo termina en el tiempo  $t = t_1$  cuando  $i_L(t = t_1) = I_p$ . Esto es,  $t_1 = I_p L / V_s$ .

**Modo 2.** Este modo es válido para  $0 \leq t \leq t_2$ . El interruptor  $S_1$  se mantiene activo, pero el diodo  $D_m$  está inactivo. La corriente del inductor  $i_L$  está dada por

$$i_L = I_m \operatorname{sen} \omega_0 t + I_o \quad (11-48)$$

donde  $I_m = V_s \sqrt{C/L}$  y  $\omega_0 = 1/\sqrt{LC}$ . El voltaje del capacitor  $v_c$  está dado por

$$v_c = V_s(1 - \cos \omega_0 t)$$

La corriente pico de comutación, que ocurre en  $t = (\pi/2)\sqrt{LC}$  es

$$I_p = I_m + I_o$$

## Importancia de los Operadores Lógicos

Los operadores lógicos son fundamentales en la programación, ya que permiten combinar o modificar condiciones booleanas, lo que facilita la toma de decisiones en el código.

### Operador AND (&&)

El operador AND (&&) devuelve true solo si ambas condiciones son verdaderas. Es útil para filtrar resultados que deben cumplir múltiples criterios.

### Operador OR (||)

El operador OR (||) devuelve true si al menos una de las condiciones es verdadera. Se utiliza cuando se desea que se cumpla al menos uno de varios criterios.

### Operador NOT (!)

El operador NOT (!) invierte el valor de una condición. Si la condición es verdadera, devuelve false, y viceversa, siendo útil para crear condiciones negativas.



# Operadores Lógicos en Java

# Importancia de los Operadores Lógicos

Los operadores lógicos son fundamentales en la programación, ya que permiten combinar o modificar condiciones booleanas, lo que facilita la toma de decisiones en el código.

# Operador AND (&&)

El operador AND (&&) devuelve true solo si ambas condiciones son verdaderas. Es útil para filtrar resultados que deben cumplir múltiples criterios.

# Operador OR (||)

El operador OR (||) devuelve true si al menos una de las condiciones es verdadera. Se utiliza cuando se desea que se cumpla al menos uno de varios criterios.

# Operador NOT (!)

El operador NOT (!) invierte el valor de una condición. Si la condición es verdadera, devuelve false, y viceversa, siendo útil para crear condiciones negativas.

# Estructuras de Control Condicionales en Java



# Estructuras de Control Condicionales en Java

Las estructuras de control condicionales son fundamentales en la programación, ya que permiten que un programa tome decisiones y ejecute diferentes bloques de código según el cumplimiento de ciertas condiciones. En Java, se utilizan principalmente las instrucciones if, else if y switch para manejar estas decisiones de manera efectiva.

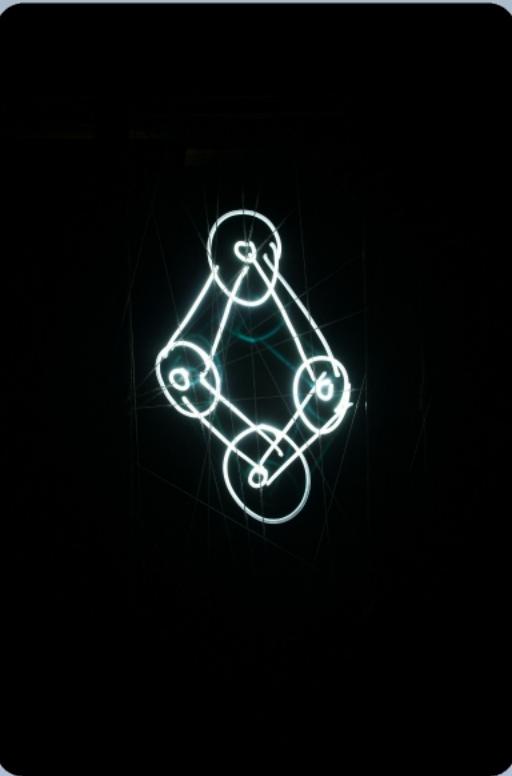


# Estructuras de Control Iterativas



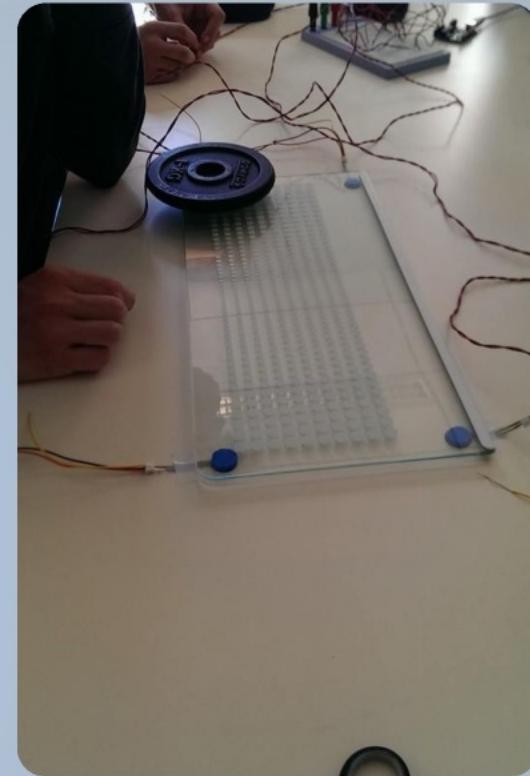
## Bucle 'while'

La estructura 'while' ejecuta un bloque de código mientras se cumpla una condición específica, evaluando la condición antes de cada iteración.



## Bucle 'do-while'

El 'do-while' garantiza que el bloque de código se ejecute al menos una vez, evaluando la condición solo después de la primera ejecución.



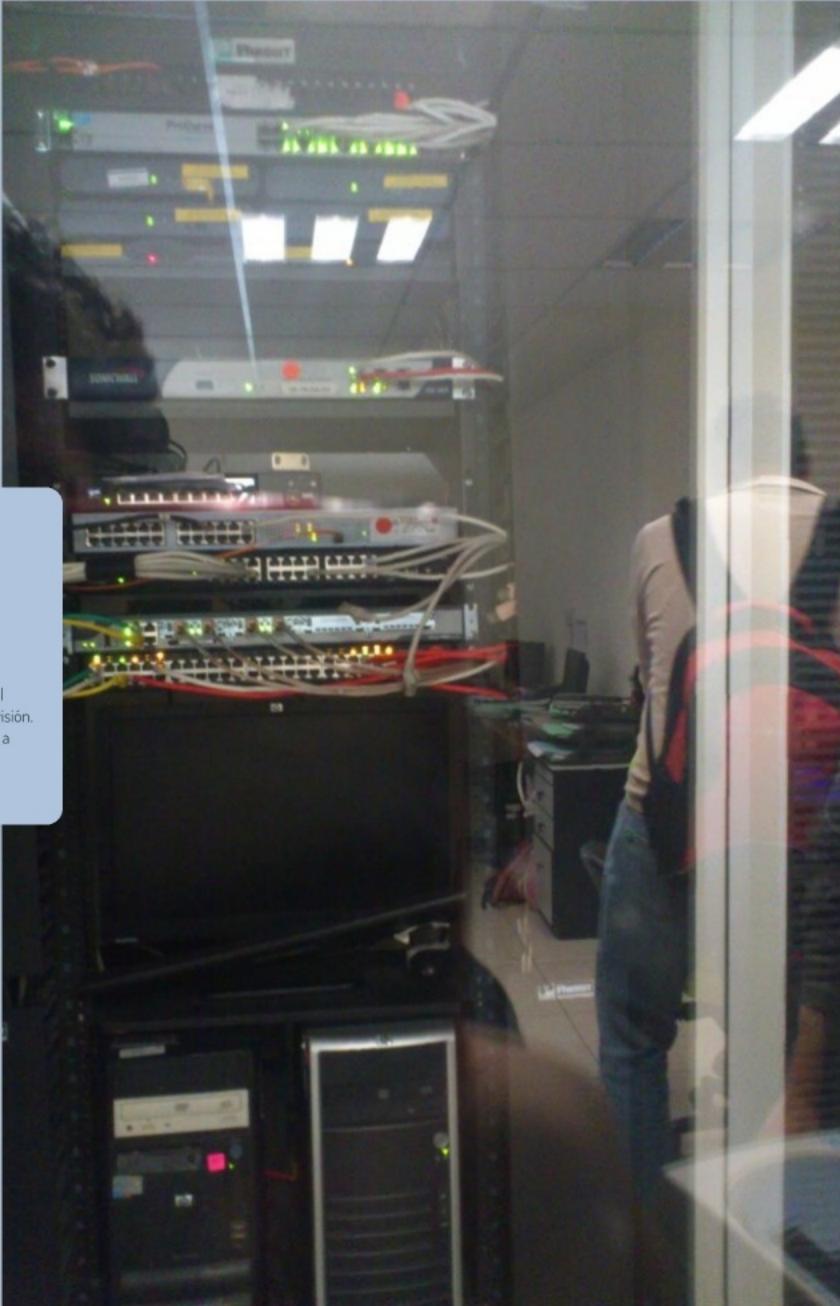
## Bucle 'for'

El bucle 'for' es ideal para ejecutar un bloque de código un número específico de veces, configurando la inicialización, la condición y el incremento en una sola línea.

# Actividades Prácticas

## Crear una Calculadora Simple

La actividad consiste en desarrollar una calculadora simple en Java que permita al usuario realizar una única operación matemática: suma, resta, multiplicación o división. El programa debe capturar la entrada del usuario para los números y la operación a realizar, asegurando que esté estructurado con métodos para cada función.



# Crear una Calculadora Simple

La actividad consiste en desarrollar una calculadora simple en Java que permita al usuario realizar una única operación matemática: suma, resta, multiplicación o división. El programa debe capturar la entrada del usuario para los números y la operación a realizar, asegurando que esté estructurado con métodos para cada función.

# Dudas y Cierre

Al finalizar la clase, se dedicará un tiempo para responder preguntas y aclarar cualquier inquietud sobre los temas tratados. Es una oportunidad para que los estudiantes expresen sus dudas y se aseguren de haber comprendido los conceptos antes de cerrar la sesión.



# Informática III

