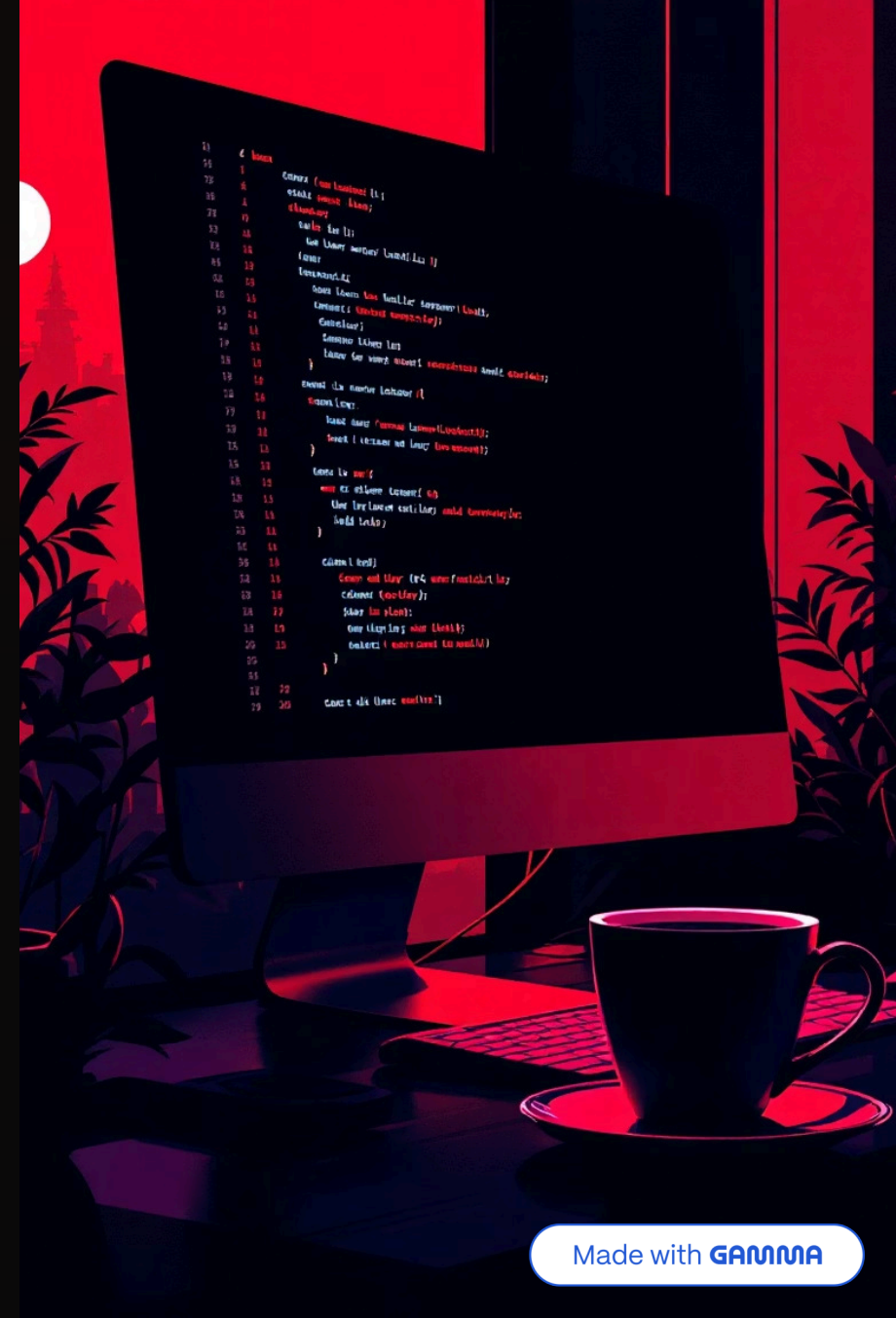


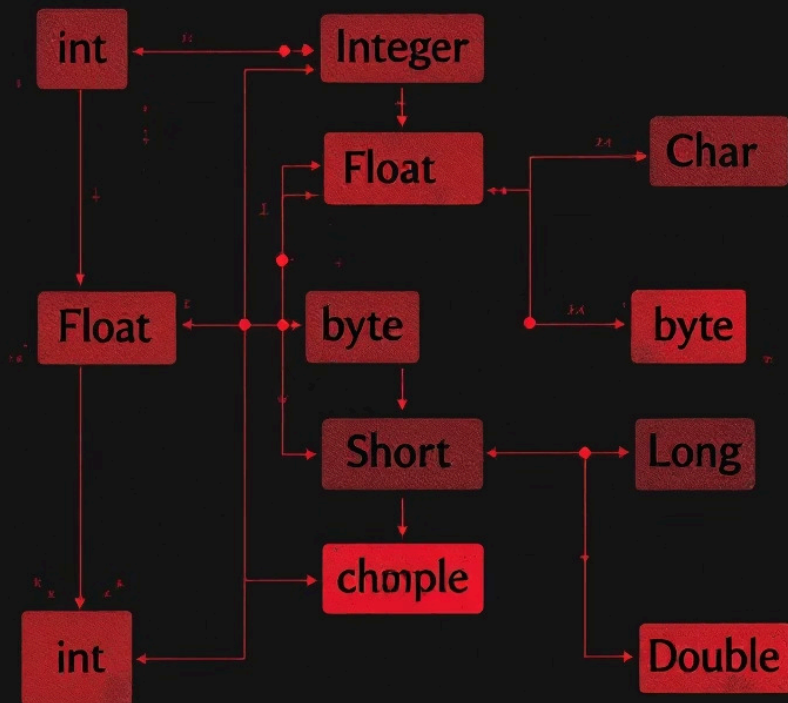
Fundamentos de Programación en Java

Este módulo cubre los conceptos fundamentales para comenzar a programar en Java, desde variables hasta estructuras de control.



Java data types

primitive types



The link
Java types with
data type

Variables y Tipos de Datos

Tipos primitivos

- int → números enteros (5, 100)
- double → números decimales (3.14, 2.5)
- char → un solo carácter ('A')
- boolean → valores lógicos true o false
- byte, short, long, float → otros tipos numéricos

Clases Wrapper

- Integer, Double, Character, Boolean
- Byte, Short, Long, Float



Ventajas de las Clases Wrapper

1 Autoboxing y Unboxing

Conversión automática entre primitivo y objeto.

2 Métodos útiles

`Integer.parseInt("123")` convierte texto en número.

3 Constantes

Integer.MAX_VALUE devuelve el máximo valor de un int.

Operadores Aritméticos

- + → suma
- - → resta
- * → multiplicación
- / → división
- % → módulo (residuo)

Ejemplo:

Si $a = 10$ y $b = 3$:

- $a + b = 13$
- $a - b = 7$
- $a * b = 30$
- $a / b = 3$
- $a \% b = 1$



Condicionales

if - else if - else

Ejecuta bloques de código dependiendo de condiciones.

```
if (edad >= 18) {  
    System.out.println("Mayor de edad");  
} else {  
    System.out.println("Menor de edad");  
}
```

switch

Evalúa una variable contra múltiples opciones.

```
switch (dia) {  
    case 1: System.out.println("Lunes");  
        break;  
    case 2: System.out.println("Martes");  
        break;  
    default: System.out.println("Otro día");  
}
```


Bucles (Ciclos)



for

Se usa cuando sabes cuántas veces repetir algo.

```
for (int i = 0; i < 5; i++) {  
    System.out.println(i);  
}
```



while

Repite mientras una condición sea verdadera.

```
while (contador < 10) {  
    contador++;  
}
```



do-while

Siempre se ejecuta al menos una vez.

```
do {  
    accion();  
} while (condicion);
```

Sentencias de Control de Flujo

break

Rompe el bucle actual y sale de él inmediatamente.

```
for (int i = 0; i < 10; i++) {  
    if (i == 5) break;  
    System.out.println(i);  
}
```

continue

Salta a la siguiente iteración del bucle.

```
for (int i = 0; i < 10; i++) {  
    if (i % 2 == 0) continue;  
    System.out.println(i);  
}
```

return

Termina un método y devuelve un valor.

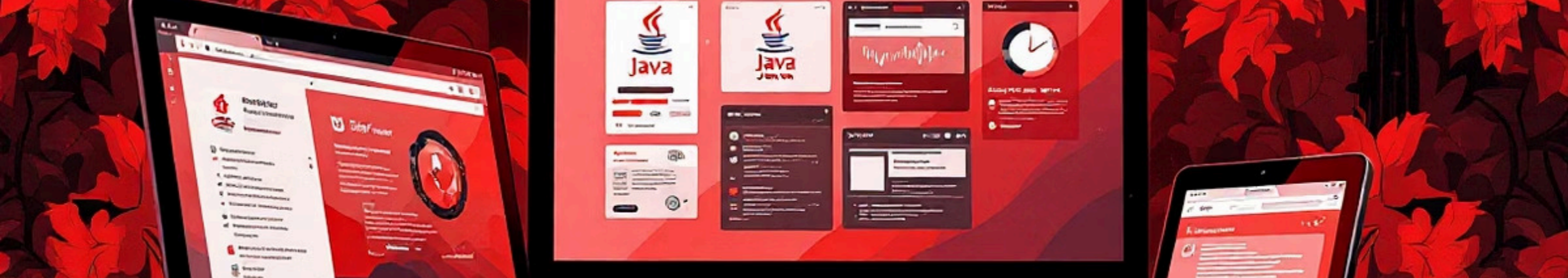
```
int suma(int a, int b) {  
    return a + b;  
}
```

Ejemplo Práctico

Veamos cómo se combinan estos conceptos en un programa simple:

```
public class Ejemplo {  
    public static void main(String[] args) {  
        int suma = 0;  
  
        for (int i = 1; i <= 10; i++) {  
            if (i % 2 == 0) {  
                suma += i;  
            }  
        }  
  
        System.out.println("La suma es: " + suma);  
    }  
}
```

Este programa suma todos los números pares del 1 al 10.



Aplicaciones Prácticas



Desarrollo de Apps

Java es ampliamente utilizado en el desarrollo de aplicaciones Android.



Sistemas Empresariales

Muchos sistemas bancarios y de gestión empresarial están desarrollados en Java.



Aplicaciones Web

Frameworks como Spring permiten crear aplicaciones web robustas con Java.

Resumen del Módulo

Variables y Tipos de Datos

Primitivos y Wrappers para almacenar información.

1

2

Operadores Aritméticos

Suma, resta, multiplicación, división, módulo.

3

Condicionales

if-else y switch para tomar decisiones.

4

Bucles

for, while, do-while para repetir código.

5

Sentencias de Control

break, continue, return para controlar el flujo.

¡Ahora estás listo para comenzar a programar en Java!