

# SCANNER

En Java, Scanner es una clase que se encuentra en el paquete `java.util` y se utiliza para leer datos de diferentes tipos desde diversas fuentes, como la entrada estándar del teclado, archivos, o incluso cadenas. La clase Scanner proporciona métodos para leer datos primitivos y cadenas de manera sencilla.

Algunos de los métodos más comunes proporcionados por la clase Scanner son:

- `nextInt()`: Lee un entero.
- `nextDouble()`: Lee un número decimal.
- `nextLine()`: Lee una línea completa de texto.
- `next()` y `next(Patrón)`: Leen una cadena de caracteres hasta el siguiente espacio en blanco o hasta que coincida con un patrón específico.

A continuación, te doy un ejemplo básico de cómo se utiliza Scanner para leer la entrada del usuario desde la consola:

```
public class EjemploScanner {  
    public static void main(String[] args) {  
        // Crear un objeto Scanner para leer desde la entrada estándar (teclado)  
        Scanner scanner = new Scanner(System.in);  
  
        // Pedir al usuario que ingrese su nombre  
        System.out.print("Ingresa tu nombre: ");  
  
        // Leer la línea ingresada por el usuario  
        String nombre = scanner.nextLine();  
  
        // Mostrar un saludo con el nombre ingresado  
        System.out.println("Hola, " + nombre + "!");  
  
        // Cerrar el scanner al finalizar para liberar recursos  
        scanner.close();  
    }  
}
```

## Uso básico de Scanner:

El siguiente código muestra cómo crear un objeto Scanner y usarlo para leer entrada desde la consola:

```
import java.util.Scanner;

public class ScannerEjemplo {
    public static void main(String[] args) {
        // Crear un objeto Scanner que lee desde la entrada estándar (consola)
        Scanner scanner = new Scanner(System.in);

        // Pedir al usuario que ingrese un número entero
        System.out.print("Ingresa un número entero: ");
        int numero = scanner.nextInt();

        // Imprimir el número ingresado por el usuario
        System.out.println("Número ingresado: " + numero);

        // Cerrar el scanner al finalizar
        scanner.close();
    }
}
```

## Ejemplos de Scanner con diferentes tipos de datos:

Leer una cadena:

```
Scanner scanner = new Scanner(System.in);
System.out.print("Ingresa tu nombre: ");
String nombre = scanner.nextLine();
System.out.println("Hola, " + nombre);
```

Leer un número decimal:

```
Scanner scanner = new Scanner(System.in);
System.out.print("Ingresa un número decimal: ");
double decimal = scanner.nextDouble();
System.out.println("Número ingresado: " + decimal);
```

Leer un carácter:

```
Scanner scanner = new Scanner(System.in);
System.out.print("Ingresa un carácter: ");
char character = scanner.next().charAt(0);
System.out.println("Carácter ingresado: " + character);
```

### Manejo de excepciones con Scanner:

El uso de Scanner puede generar excepciones si la entrada no es del tipo esperado. Es importante manejar estas excepciones para evitar errores inesperados. Por ejemplo:

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class ManejoExcepcionesScanner {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Ingresa un número entero: ");
            int numero = scanner.nextInt();
            System.out.println("Número ingresado: " + numero);
        } catch (InputMismatchException e) {
            System.out.println("Error: Ingresa un número válido.");
        } finally {
            // Cerrar el scanner en el bloque finally para asegurar su cierre.
            scanner.close();
        }
    }
}
```

Recuerda que al usar Scanner, siempre debes cerrarlo para liberar los recursos asociados. Puedes hacerlo en el bloque finally o utilizando un bloque try-with-resources en Java 7 y versiones posteriores.