

# Documentación del Proyecto: Verificador de Números Primos en Java

---

## Índice

1. [Introducción](#)
  2. [Descripción General](#)
  3. [Requisitos](#)
  4. [Instalación](#)
  5. [Uso](#)
  6. [Estructura del Código](#)
    - [Importaciones](#)
    - [Clase Principal](#)
    - [Inicialización de Recursos](#)
    - [Flujo del Programa](#)
    - [Manejo de Excepciones](#)
  7. [Ejemplos de Ejecución](#)
  8. [Manejo de Errores](#)
- 

## Introducción

El **Verificador de Números Primos en Java** es una aplicación de consola diseñada para determinar si un número ingresado por el usuario es primo. Un número primo es aquel que solo tiene dos divisores positivos distintos: 1 y sí mismo. Este proyecto es una herramienta educativa para practicar conceptos fundamentales de programación en Java, como manejo de entradas/salidas, estructuras de control y optimización de algoritmos.

---

## Descripción General

Este proyecto consiste en una única clase Java que interactúa con el usuario a través de la consola. El programa solicita al usuario que ingrese un número y verifica si dicho número es primo. La verificación se realiza comprobando si el número tiene divisores además de 1 y sí mismo, optimizando el proceso al verificar hasta la raíz cuadrada del número ingresado.

---

## Requisitos

- **Lenguaje de Programación:** Java (JDK 8 o superior)
  - **Entorno de Desarrollo:** Cualquier IDE compatible con Java (Eclipse, IntelliJ IDEA, NetBeans) o un editor de texto junto con la terminal.
  - **Sistema Operativo:** Compatible con cualquier sistema operativo que soporte Java (Windows, macOS, Linux).
-

# Instalación

## 1. Descargar e Instalar Java:

- Asegúrate de tener el Java Development Kit (JDK) instalado. Puedes descargarlo desde [Oracle](#) o usar una distribución OpenJDK.

## 2. Configurar el Entorno:

- Configura las variables de entorno (`JAVA_HOME` y `PATH`) según las instrucciones de tu sistema operativo.

## 3. Obtener el Código Fuente:

- Clona el repositorio o descarga el archivo `Main.java` que contiene el código fuente del proyecto.

---

# Uso

## 1. Compilar el Programa:

- Navega al directorio donde se encuentra el archivo `Main.java` y compílalo utilizando el siguiente comando:

```
javac Main.java
```

## 2. Ejecutar el Programa:

- Una vez compilado, ejecuta el programa con:

```
java Main
```

## 3. Interacción con el Usuario:

- El programa te solicitará que ingreses un número para evaluar.
- Ingresa un número entero y el programa determinará si es primo.
- El resultado se mostrará en la consola.

---

# Estructura del Código

## Importaciones

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
```

- **BufferedReader**: Facilita la lectura eficiente de caracteres de la entrada estándar.
- **IOException**: Maneja excepciones que puedan ocurrir durante las operaciones de entrada/salida.
- **InputStreamReader**: Convierte una secuencia de bytes (entrada estándar) en una secuencia de caracteres.

## Clase Principal

```
public class Main {  
    public static void main(String[] args) {  
        // Código principal  
    }  
}
```

- **public class Main**: Define la clase principal del programa.
- **public static void main(String[] args)**: Método de entrada del programa donde se ejecuta la lógica principal.

## Inicialización de Recursos

```
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
```

- **BufferedReader reader**: Instancia para leer la entrada del usuario desde la consola.

## Flujo del Programa

```
try {  
    // Solicitud y lectura del número  
    System.out.println("Enter a number to evaluate");  
    int number = Integer.parseInt(reader.readLine());  
  
    // Variable para determinar si el número es primo  
    boolean isPrime = true;  
  
    // Verificación básica: los números menores o iguales a 1 no son primos  
    if (number <= 1) {  
        isPrime = false;  
    } else {  
        // Verificación de divisores desde 2 hasta la raíz cuadrada del número  
        for (int i = 2; i <= Math.sqrt(number); i++) {  
            if (number % i == 0) {  
                isPrime = false; // Si se encuentra un divisor, el número no es  
                primo  
                break; // Salir del bucle ya que no es necesario continuar  
            }  
        }  
    }  
}
```

```

    }
}

// Resultado final
if (isPrime) {
    System.out.println(number + " is a Prime number");
} else {
    System.out.println(number + " is not a Prime number");
}

} catch (IOException e){
    System.out.println("An error occurred while reading the input: " +
e.getMessage());
}

```

## Detalles del Flujo del Programa:

### 1. Solicitud y Lectura del Número:

- El programa solicita al usuario que ingrese un número para evaluar.
- Utiliza `BufferedReader` para leer la entrada del usuario y `Integer.parseInt` para convertir la entrada a un entero.

### 2. Inicialización de la Variable `isPrime`:

- Se asume inicialmente que el número es primo (`isPrime = true`).

### 3. Verificación Básica:

- Si el número es menor o igual a 1, no es primo (`isPrime = false`).

### 4. Verificación de Divisores:

- Se itera desde 2 hasta la raíz cuadrada del número ingresado.
- Si el número es divisible por cualquier número en este rango (`number % i == 0`), no es primo (`isPrime = false`) y se rompe el bucle para optimizar el rendimiento.

### 5. Resultado Final:

- Después de la verificación, se imprime si el número es primo o no basado en el valor de `isPrime`.

## Manejo de Excepciones

```

catch (IOException e){
    System.out.println("An error occurred while reading the input: " +
e.getMessage());
}

```

- Captura cualquier excepción de tipo `IOException` que pueda ocurrir durante la lectura de la entrada del usuario.
  - Muestra un mensaje de error detallando el problema para informar al usuario.
- 

## Ejemplos de Ejecución

### Caso 1: Número Primo

```
Enter a number to evaluate
17
17 is a Prime number
```

### Caso 2: Número No Primo

```
Enter a number to evaluate
20
20 is not a Prime number
```

### Caso 3: Número Menor o Igual a 1

```
Enter a number to evaluate
1
1 is not a Prime number
```

### Caso 4: Entrada Inválida (No Entero)

```
Enter a number to evaluate
abc
An error occurred while reading the input: For input string: "abc"
```

---

## Manejo de Errores

El programa maneja posibles errores relacionados con la entrada/salida mediante un bloque `try-catch`. Si ocurre una excepción `IOException` durante la lectura de la entrada del usuario, el programa captura la excepción y muestra un mensaje de error con detalles específicos. Además, si el usuario ingresa un valor que no es un número entero válido, se lanzará una excepción `NumberFormatException`, la cual no está manejada explícitamente en el código actual. Para mejorar la robustez del programa, se recomienda agregar un manejo de excepciones adicional para capturar y gestionar errores de formato de entrada.