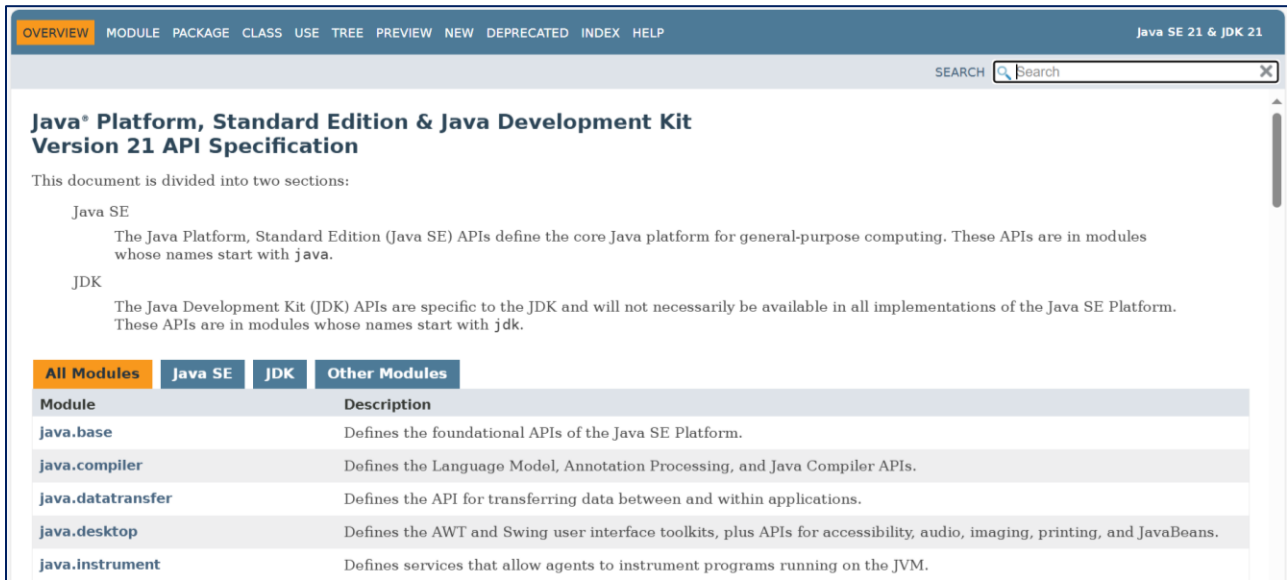


1 PAQUETES.

Los paquetes agrupan un conjunto de clases que trabajan conjuntamente sobre el mismo ámbito.

<https://docs.oracle.com/en/java/javase/21/docs/api/index.html>



OVERVIEW MODULE PACKAGE CLASS USE TREE PREVIEW NEW DEPRECATED INDEX HELP

Java SE 21 & JDK 21

SEARCH Search

Java® Platform, Standard Edition & Java Development Kit Version 21 API Specification

This document is divided into two sections:

Java SE

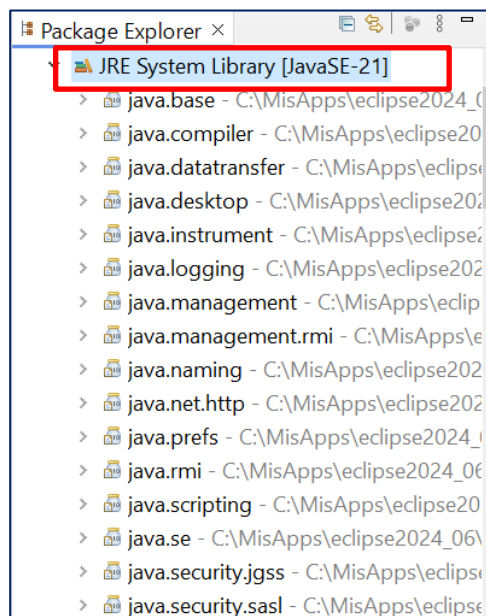
The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with `java`.

JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`.

Module	Description
<code>java.base</code>	Defines the foundational APIs of the Java SE Platform.
<code>java.compiler</code>	Defines the Language Model, Annotation Processing, and Java Compiler APIs.
<code>java.datatransfer</code>	Defines the API for transferring data between and within applications.
<code>java.desktop</code>	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.
<code>java.instrument</code>	Defines services that allow agents to instrument programs running on the JVM.

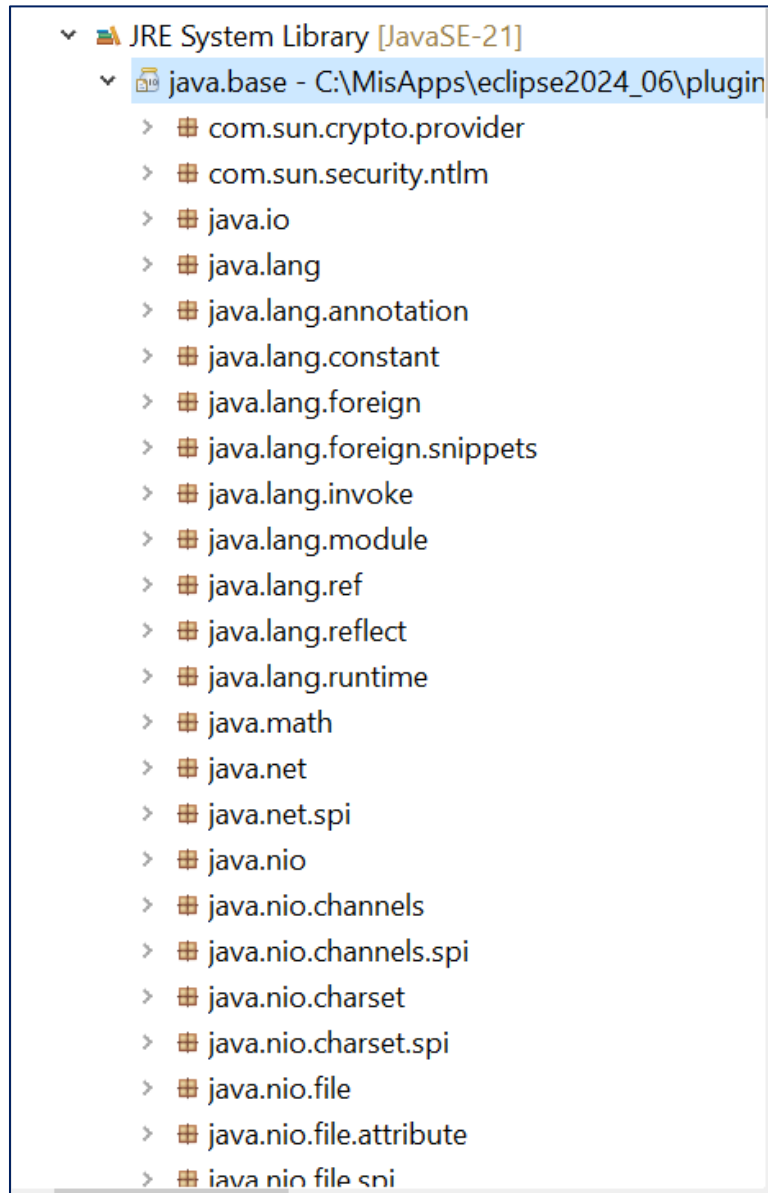
Dentro del proyecto Eclipse, pinchando en el JRE se pueden ver los módulos que forman parte de JDK 21:



java.base: Define las API fundamentales de la plataforma Java SE.

Las siglas **API** tienen su origen en **Application Programming Interface** y consiste en un conjunto de librerías de código Java compilado o clases ofrecidas por la compañía multinacional ya listas para que sean usadas por todos los desarrolladores o programadores.

Dentro de los módulos anteriores podemos ver los paquetes:



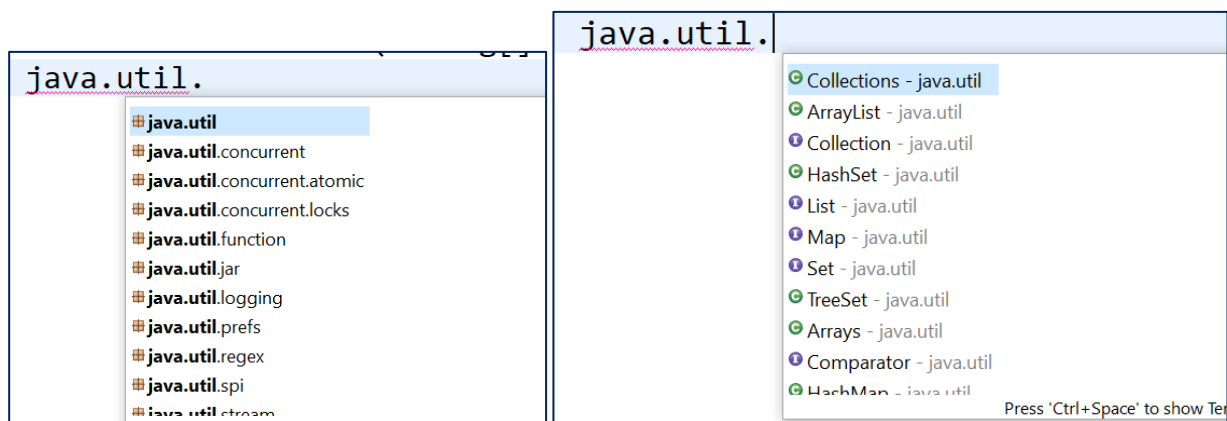
Algunos de ellos son:

- **java.lang** incluye clases básicas o esenciales.

El paquete **java.lang** se importa automáticamente. Esto quiere decir que no es necesario poner en la clase donde se vaya a hacer uso de este paquete la cláusula o instrucción correspondiente **import java.lang**.

PRINCIPALES CLASES EN EL PAQUETE JAVA.LANG	
Boolean	Math
Double	String
Float	StringBuffer y StringBuilder
Integer	System

- **java.io** incluye clases para la serialización de objetos y manejar la entrada y salida.
- **java.util** facilita y permite el manejo de horas, fechas, excepciones, estructuras de datos o colecciones o más a través de las clases que lo integran. Lectura de teclado con **Scanner**.



- **Java.time** incluye clases para manejo de fecha y hora.
- **java.awt** (módulo **java.desktop**) contiene clases que permiten pintar imágenes y gráficos, manejar la GUI.
- **java.sql** incluye clases que permiten el manejo de bases de datos relacionadas.
- **java.net** provee soporte para sockets.

La palabra reservada “**import**” se usa para importar las diferentes clases de estos paquetes. Ejemplo:

```
import java.time.LocalDateTime;
```

Importa la clase **LocalTime** del paquete **time**

2 SALIDA POR CONSOLA.

Para mostrar los resultados por pantalla podemos usar los métodos de la clase **System**, que tiene para diferentes formas de mostrar datos por pantalla.

```
System.out.println("Hola Mundo");
System.out.print("Hola Mundo");
String nombre = "Maria";
short edad = 20;
System.out.println("Valor de edad: " + edad);

System.out.printf("Nombre: %s Edad: %d %n", nombre, edad);
```

<https://oregoom.com/java/salida-de-datos/>

Para mostrar decimales con formato (punto para los miles y coma de los decimales). Ejemplo:

```
double d = 10000;
System.out.printf("Valor1: %,9.2f %n", d);
System.out.printf("Valor2: %f %n", d);
System.out.printf("Valor3: %,12.2f %n", d);
```

```
Valor1: 10.000,00
Valor2: 10000,000000
Valor3: 10.000,00
```

```
int a= 4123456;
System.out.printf("Valor entero: %,15d %n", a);
//Valor entero: 4.123.456
```

3 ENTRADA POR CONSOLA.

Scanner es una clase en el paquete **java.util** utilizada para obtener la entrada de los tipos primitivos como int, double etc. y también String.

<https://oregoom.com/java/entrada-de-datos/>

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        //Instanciar la clase Scanner o
        // crear el objeto leer

        Scanner leer = new Scanner(System.in);

        System.out.println("Ingrese su nombre: ");
        String nombre = leer.nextLine();

        System.out.println("Ingrese su Edad: ");
        int edad = leer.nextInt();

        System.out.println("Ingrese un caracter: ");
        char c = leer.next().charAt(0);

        System.out.println("Ingrese una palabra: ");
        String palabra = leer.next();

        System.out.println("Nombre: "+ nombre);
        System.out.println("Edad: "+ edad);
        System.out.println("Valor de Char: "+ c);
        System.out.println("Valor de palabra: "+ palabra);

    }
}
```

HACER EJEMPLO DE CONTROLAR EXCEPCIONES en la entrada de datos por teclado.

LEER DECIMALES (PÁGINA 23)

```
Scanner sc = new Scanner(System.in);
sc.useLocale(Locale.US);
```

```
System.out.print("Introduce un decimal (.): ");  
double d = sc.nextDouble();
```

Introduce un decimal: 1.4

```
Scanner sc = new Scanner(System.in);  
System.out.print("Introduce un decimal (,): ");  
double d = sc.nextDouble();
```

Introduce un decimal: 1,4