

Programación II

Tarea 6

Investigación Api Java

Autor:
Luis Mendoza Madriz

¿Qué es?

API viene de *Application Programming Interface* que es un grupo de clases, paquetes e interfaces predefinidos en java que ayudan a los programadores a desarrollar sin tener que escribir tantos algoritmos desde cero. Estas ayudan principalmente en no tener que volver a tener que utilizar algoritmos, metodos o funciones ya creadas y compiladas en una aplicación de la interfaz ahorrando tiempo y esfuerzo; y a su vez este es compatible con las implementaciones de java.

Estas aplicaciones se dividen en:

- **Paquetes (java.lang):** Este paquete se encuentran importadas automaticamente en java tales como:
 1. **Object:** Clase base de todas clases de java
 2. **String:** Maneja cadenas de Texto
 3. **Math:** Maneja operaciones matemáticas
 4. **System:** Maneja Entradas y salidas básicas
 5. **Datos Primitivos:** Tales como Integer, Double, Boolean, etc.
- **Colecciones y Utilidades (java.util):** Contiene utilidades diversas que requieren de ser importadas tales como:
 1. **Estructuras de Datos:** Manejan estructuras de datos como ArrayList, LinkedList, HashMap, HashSet, etc.
 2. **Collections:** Contiene utilidades como ordenar, copiar, buscar, etc.
 3. **Random:** Utilidad para generacion de numeros aleatorios (Un sustituto a este es el metodo Math.random();)
 4. **Manejo de tiempo:** Maneja datos de fecha y hora tales como Date, Calendar, TimeZone
- **Entradas y salidas (java.io):** Se utiliza para trabajar con flujo de datos, archivos y lectura/escritura:
 1. **File:** Hace referencia a archivos y directorios
 2. **Lectura de Archivos:** Se basa en la lectura de archivos tales como FileReader o BufferedReader
 3. **Escritura de Archivos:** Se basa en la escritura de archivos tales como FileWriter, BufferedWriter o PrintWriter
 4. **Flujo de bytes:** Tales como Input Stream, OutputStream
- **Redes (java.net):** Se utiliza para la programacion de red (Sockets, URL's, Comunicación entre programas, etc.):
 1. **Conexión Cliente/Servidor:** Socket y ServerSocket (Para el cliente y el servidor respectivamente)
 2. **URL, URLConection:** Conexión a sitios web y recursos remotos

- *Concurrencia (java.util.concurrent): Manejo de múltiples hilos y tareas simultáneas:*
 1. *Thread, Runnable: Para crear y ejecutar hilos*
 2. *ExecutorService: Para gestionar múltiples tareas*
 3. *Semaphore, Lock, CountdownLatch: Sincronización entre hilos*
- *Extenciones Estándar (javax.): Componentes para interfaces gráficas, seguridad, XML, y más:*
 1. *javax.swing – para crear interfaces gráficas con ventanas, botones, menús, etc.*
 2. *javax.xml – para leer y escribir archivos XML*
 3. *javax.crypto – para cifrado y seguridad*
 4. ***Nota: Muchas de las clases de javax. Pasaron a ser parte de otros módulos o reemplazados por otras bibliotecas***
- *Fecha y hora moderna (java.time): Funciona en java 8 en adelante y es una version que reemplaza a Date y Calendar:*
 1. *LocalDate, LocalTime, LocalDateTime, ZonedDateTime: Los cuatro funcionan de manera similar*
 2. *Duration, Period: Se utiliza para diferencia entre tiempo*
 3. *DateTimeFormatter: Para dar formato a las fechas*
 4. ***Nota: En el código de la tarea 6 utilice este en lugar de Date y utilice DateTimeFormatter para dar formato a la fecha.***
- *Manejo de Errores (java.exceptions): No es exactamente un paquete pero sirve para manejar errores que se producen al manejar de manera incorrecta los diversos objetos; al igual se puede utilizar un try-catch-finally para capturar estos errores y manejarlos de la manera deseada.*
Nota: Utilice esta funcion para crear errores personalizados para resolver problemas con desbordamientos y mal manejo de objetos tanto para encontrar el punto de error como para hacer el código mas robusto
- *Otras bibliotecas útiles*
 1. *java.nio – Entrada/Salida no bloqueante, muy eficiente (archivos, buffers, canales).*
 2. *java.sql – Conexión con bases de datos.*
 3. *java.security – Criptografía, autenticación y permisos*

Nota: Junto a este archivo dejare un código de ejemplo con el cual tambien me dio ideas de como funcionan ciertas aplicaciones.

Nota2: El código no fue hecho por mí sino por inteligencia artificial, no quiere decir que no haya entendido las funcionalidades si no que me permite ver como funcionan.

**Se puede encontrar
mas informacion en el
siguiente enlace:**

[oracle.com/java](https://docs.oracle.com/en/java/javase/21/docs/api/)

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