



# **UNITAT 4**

## **INTRODUCCIÓ A JAVA**

### **GUIA API de Java**

**Programació  
CFGS DAW**

**Autors:**

Joan Vicent Cassany – [jv.cassanycoscolla@edu.gva.es](mailto:jv.cassanycoscolla@edu.gva.es)

**Revisat per:**

2021/2022

## Llicència



**CC BY-NC-SA 3.0 ES** **Reconeixement – No Comercial – Compartir Igual (by-nc-sa)**

No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original. Aquesta és una obra derivada de l'obra original de Carlos Cacho i Raquel Torres.

## Nomenclatura

Al llarg d'aquest tema s'utilitzaran diferents símbols per a distingir elements importants dins del contingut. Aquests símbols són:

	Important
	Atenció
	Interessant

# ÍNDEX

## 1. API DE JAVA

4

## 1. API de Java

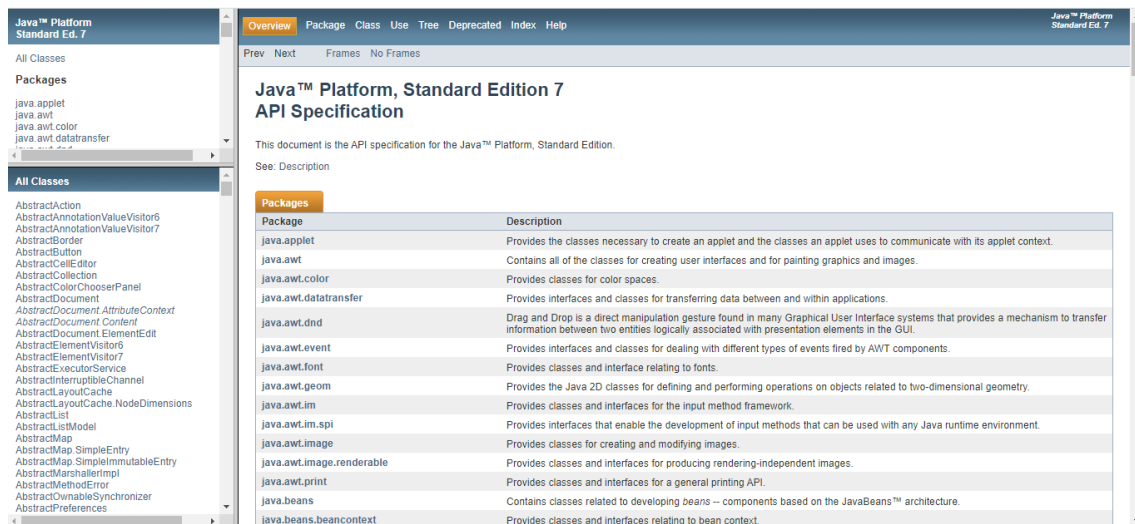
Tot programa Java ha d'estar compost per almenys una classe.

Les classes a Java poden ser de dos tipus: classes pròpies (són les que creem nosaltres) i classes definides (String, Math, Array, Thread, etc.).

Hi ha centenars de classes i cadascuna amb una gran diversitat de mètodes. És tan àmplia la xifra que no es poden saber totes de memòria i encara menys la seva sintaxi. És per això que a Java existeix el que s'anomena Biblioteca de Classes o API de Java, que no és ni més ni menys que una biblioteca on vénen relacionades totes les classes de Java i els seus mètodes.

Amb freqüència accedirem a aquesta biblioteca per fer consultes. Per això realitzarem una cerca des de qualsevol navegador amb el text "Java API". Que ens portarà al següent enllaç.

<https://docs.oracle.com/javase/7/docs/api/>



Al panell inferior esquerre podem veure totes les classes existents a Java i que analitzarem com consultar-ho més endavant.

Centrant-nos en la classe Math veurem alguns mètodes de gran utilitat.

Math.round(decimal): arrodoniment d'un número

Math.sqrt(n): arrel quadrada

Math.pow(base, exponent): calcula la potència d'un nombre base elevat a un exponent

Math.sin(angle): si d'un angle (igual que la resta de funcions trigonomètriques)

També hi ha les constants de classe.

Math.PI: número PI

A la part superior esquerra de la imatge veiem que apareix el concepte de Paquets.

Hem de tenir en compte que les classes estan sempre contingudes en un paquet, és a dir, que cada paquet conté un conjunt classes. Això no és ni més ni menys que una manera d'ordenació de les classes.

A més, els paquets també tenen una certa jerarquia (com passa amb les carpetes del nostre ordinador). Uns paquets poden contenir altres. Així, per exemple, podem veure aquesta jerarquia

`java.lang`

`java.lang.annotation`

`java.lang.instrument`

`java.lang.invoke`

`java.lang.management`

`java.lang.ref`

`java.lang.reflect`

on el paquet `java.lang.annotation` i tots els següents estan continguts al paquet principal `java.lang`.

Adicionalment, hem de tenir en compte que el paquet `java.lang` és el paquet per defecte, de manera que totes les funcions que veurem a continuació es poden utilitzar sense haver de fer cap menció explícita a aquest paquet. Més endavant veurem què passa amb la resta de paquets.