

# **CURSO INICIACIÓN JAVA I**

# Tutoriales de pildorasinformaticas

## Descripción breve

Curso introductorio de Java por pildorasinformaticas.



Pere Manel Verdugo Zamora pereverdugo@gmail.com

# Presentación (Vídeo 1)

¿A quién va dirigido?

- Personas sin conocimientos de programación en general y Java en particular.
- Se utilizará una jerga sencilla y se explicarán los términos más farragosos y complejos.
- Insisto: Los primeros 20 vídeos no son aptos para personas con conocimientos de Java.

Entorno de desarrollo a utilizar en el curso

- Se utilizará Eclipse como entorno de desarrollo. ¿Por qué?
  - Porque es el que más le gusta al profesor.
  - Porque la curva de aprendizaje no es muy pronunciada.
  - Porque es uno de los entornos de desarrollo Java más utilizados hoy en día.
  - Porque es gratuito.
  - Porque hay numerosos foros, documentación online y tutoriales en línea sobre él.
  - El curso no trata sobre el aprendizaje de Eclipse. Se irá aprendiendo sobre la marcha a medida que avance el curso.

## Otros entornos de desarrollo

- JDK
- NetBeans
- BlueJ
- JBuilder
- JCreator
- JDevenloper

## Temario Parte 1

- Descarga e instalación de Eclipse
- Introducción a Jva
- Estructuras principales del lenguaje
- Objetos y clases
- Herencias
- Clases internas e interfaces.
- Programación de gráficos
- Eventos
- Componentes Swing
- Aplicaciones y Applets
- Tratamiento de errores (excepciones) y depuración
- Programación genérica
- Colecciones
- Programación multihilo (multihtrearding)

Temario Parte 2

- Programación con archivos
- XML
- Programación para redes
- Programación para BBDD
- Programación cliente-servidor (objetos distribuidos)
- Swing avanzado
- AWT avanzado
- JavaBeans
- Seguridad
- Programación internacional
- Método nativos
- Anotaciones

¿Y todo esto cómo se distribuye?

- Más de 250 vídeos con una duración media de 20 minutos cada uno.
- A lo largo del curso se irán haciendo ejercicios prácticos, muy sencillos al comienzo, más complejos según vayamos avanzando.





# Instalación JRE y Eclipse (Vídeo 2)

¿Qué es JRE?

- J.R.E = Java Runtime Enviroment (Entorno de ejecución Java), También denominado en sus inicios como J.V.M. = Java Virtual Machine (Máquina Virtual Java).
- ¿Por qué es necesaria su instalación?
  - Por culpa de la principal característica del lenguaje Java: ser MULTIPLATAFORMA.
  - La característica de ser multiplataforma implica que un programa escrito en Java debe ser <u>compilado</u> para posteriormente ser <u>interpretado</u> por la Máquina Virtual de Java o JRE.

¿Qué es compilar?



Según el sistema operativo que utilicemos utilizaremos distintos compiladores.



Como Java es un lenguaje multiplataforma el prceso es el siguiente:



El fichero que generamos tiene una extensión java, cuando lo compilamos nos genera un fichero intermedio (fichero de Bytecodes), tendrá extensión .class, le aplicaremos la máquina virtual de Java, el objetivo es traducir dicho fichero a código máquina independientemente de la plataforma en la que estamos trabajando, por eso se necesita JRE.



Una vez tengamos el fichero con extensión .class se podrá ejecutar desde cualquier ordenador que tenga instalada la maquina virtual de Java (JRE).



Escríbelo una vez y ejecútalo allí donde quiera.

Vamos a descargar la máquina virtual de Java, para ello accederemos al siguiente enlace:

https://www.java.com/es/



ORACLE

<u>Seleccionar idioma</u> | <u>Acerca de Java</u> | <u>Soporte</u> | Desarrolladores <u>Privacidad</u> | <u>Preferencias sobre cookies</u> | <u>Condiciones de uso</u> | <u>Marcas registradas</u> | <u>Descargo de responsabilidad</u>

#### Seleccionaremos Descarga gratuita de Java.

لمن العامة (Java	Buscar ्र Descargar Ayuda
Recursos de ayuda » ¿Qué es Java? » Eliminar versiones anteriores de Java » Desactivar Java » Mensajes de error » Solucionar problemas de Java » Otra ayuda	Descargar Java para Windows Recomendado Version 8 Update 261 (Tamaño de archivo: 1.99 MB) Fecha de lanzamiento: 14 de julio de 2020 Actualización importante de la licencia de Oracle Java La licencia de Oracle Java ha cambiado para las versiones publicadas a partir del 16 de abril de 2019.
Usuarios de Windows de 64 bits ¿Utiliza exploradores de 32 y 64 bits? » Preguntas frecuentes sobre Java de 64 bits para Windows Instalación fuera de línea ¿Problemas al descargar? Intente con el installer fuera	<ul> <li>In uevo <u>acuerdo de licencia de Oracle Technology Network para Oracle Java SE</u> es sustancialmente diferente a las licencias de Oracle Java anteriores. La nueva licencia permite ciertos usos, como el uso personal y de desarrollo, sin coste alguno (aunque podría haber otros usos autorizados en licencias de Oracle Java anteriores que ya no estén disponibles). Revise las condiciones con atención antes de descargar y utilizar este producto. Puede consultar las preguntas frecuentes <u>aquí</u>.</li> <li>La licencia comercial y el soporte están disponibles con una <u>suscripción de Java SE</u> de bajo coste.</li> <li>Oracle también ofrece la última versión de OpenJDK con la <u>licencia pública general</u> de código abierto en j<u>dk java net</u>.</li> </ul>
<u>de línea</u> .	<ul> <li>En Windows 10, el explorador Edge no es compatible con ningún plugin y, por lo tanto, no ejecutará Java. Cambie a un explorador diferente (internet Explorer, por ejemplo) para ejecutar el plugin de Java. Seleccione la opción Más acciones situada en la parte superior derecha del explorador Edge y haga clic en Abrir con Internet Explorer. Más información</li> </ul>

Al descargar Java, confirma que ha leído y acepta las condiciones del <u>acuerdo de licencia de Oracle Technology</u> <u>Network para Oracle Java SE</u>

Seleccionaremos el botón Aceptar e iniciar descarga gratuita.

## Iremos a descargas.

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	r Vista	Admini Herramientas d	strar le aplicación	Descargas					_		× ^ ?
Anclar al Acceso rápido Portapapel	☆ Cortar □ Copiar ruta d Pegar acceso les	le acceso directo	Mover a 🔹 🗙 Copiar a 👻 📑 Organ	Eliminar 👻 Cambiar nombre nizar	Nueva carpeta Nuevo	Propiedades Abrir	∎ • 2 2	Sele	eccional seleccio ertir sele seleccio	r todo mar nad ección nar	a
← → ∽ ↑ 🕇 > Este e	quipo > Desc	argas >				~	Ō	Q	Buscar	en Desc	arg
Acceso rápido	Nombre ~ hoy (4) —					Fecha de m	nodifica	ción	Тіро		^
Descargas	JavaSetup Java_logo. OfficeSetu	8u261.exe .jpg .p.exe 3,2016,2019 C2R	l Install + Lite 7.	• 04 + Activador cmc	1.7z	26/08/2020 26/08/2020 26/08/2020 26/08/2020	17:34 16:28 16:19 15:51		Aplicad Archive Aplicad Archive	ti <mark>ón</mark> 5 JPG tión 5 WinRA	٩R
<ul> <li>Documentos</li> <li>Office 2013-2016</li> <li>Sublime Text</li> </ul>	✓ ayer (1) — 20200825_ ✓ al principio SOLiteDat	155130[10817].jj o de este me abaseBrowserPc	og s (14) ortable 3.12.0 E	nolish.paf.exe		25/08/2020	15:54		Archive	o JPG	
Creative Cloud File	seleccionado 1,	98 MB				15/08/2020	17-/12		Archiv	WinRA	R 7 ▼ >

#### Ejecutaremos para su instalación.

Configuración de Java - Bienvenido	_		×
Bienvenido a Java - Condiciones de licencia actualizadas	;		
Condiciones que han cambiado en la licencia de esta versión del software <u>Acuerdo de licencia actualizado</u> Esta versión de Java Runtime únicamente otorga permisos para el uso de equipos po sobremesa (no comerciales). El uso comercial de este software requiere una licencia independiente de Oracle o de su software. Haga clic en Instalar para aceptar el acuerdo de licencia e instalar Java ahora o haga clic desinstalarlo del sistema. En este proceso de instalación no se registra ningún dato personal. <u>Detalles de la info</u> <u>recopilamos</u>	ortátiles I provee en Elimi ormación	y de dor de nar para n que	1
Cambiar la Carpeta de Destino		Elimina	

Seleccionaremos el botón Instalar.

Configuración de Java - Progreso	– 🗆 X
Estado: Instalación de Java	
Java™ 25 Years (	
#1 Development Platform	ORACLE
Configuración de Java - Desinstalar versiones obsoletas	- 🗆 X
Versiones de Java obsoletas de	tectadas
Mantener versiones obsoletas de Java instaladas en su sistema puede su	poner un riesgo para la seguridad.
☑ Java 8 Update 131	~
Haga ciic en "Desinstalar" para desinstalar las versiones de Java seleccion	adas.
Advertencia: Puede que al desinstalar las versiones obsoletas de Java algunas aplic funcionar. <u>Más información</u>	aciones Java mas antiguas dejen de
	Ahora No Desinstalar >

Ha detectado un versión anterior, seleccionaremos el botón Desintalar.



Seleccionaremos el botón Siguiente.



La instalación se ha realizado correctamente, seleccionaremos el botón Cerrar.

Ahora para la instalación de Eclipse accederemos al siguiente enlace:

https://www.eclipse.org/downloads/



Download Eclipse Technology that is right for you

	Tool Platforms	
Get Eclipse IDE 2020-06 Install your favorite desktop IDE packages. Download 64 bit Download Packages   Need Help?	Eclipse Che Eclipse Che is a developer workspace server and cloud IDE.	A modern, open source software development environment that runs in the cloud.

Seleccionaremos el botón Download 64 bit.

🕹   🛃 🥃 🗸	Administrar	Descargas			-	- 🗆	×	
Archivo Inicio Compartir Vista	Herramientas de aplicación						^ (	2
Anclar al Acceso rápido	de acceso o directo	Cambiar nombre	Nueva carpeta	Propiedades	No se	cionar todo eleccionar na tir selección	da	
Portapapeles	Orga	inizar	Nuevo	Abrir	Se	leccionar		
← → ~ ↑ 🕹 > Este equipo > Desc	cargas >			ٽ ~	,РВ	uscar en Des	carg	
Acceso rápido				Fecha de modific	ación T	īpo	,	^
Lescargas (O eclipse-in	nst-win64.exe			26/08/2020 17:51	Д	plicación		
Documentos     JavaSetup     Java_logo     Imágenes     OfficeSet	o8u261.exe Jipg up.exe			26/08/2020 17:34 26/08/2020 16:28 26/08/2020 16:19	А А А	Aplicación Archivo JPG Aplicación		
Disco local (D:)	12 2016 2010 COD ( III - I 2 7	· · · · · · · · · · · · · · · · · · ·	17-	<u> 16 /00 /2020 15.51</u>	^	L1 \//:-D	<u>&gt;</u>	4
118 elementos 1 elemento seleccionado 5	2.4 MB						8== 🛌	1

Ejecutamos para su instalación.



eclipseins	taller by Oomph	
Eclipse I The essenti Editor, Mav	<b>DE for Java Developers</b> details al tools for any Java developer, including a Java IDE, a Git client, XML an and Gradle integration.	
Java 1.8+ VM	C:\Program Files\Java\jdk-14.0.2 (Current)	
Installation Folder	C:\Users\pmver\eclipse\java-2020-06	
	<ul> <li>✓ create start menu entry</li> <li>✓ create desktop shortcut</li> </ul>	
	Cancel Installation	
васк		

Vamos a crear una carpeta porque la primera vez que se ejecute Eclipse nos pedirá en que carpeta queremos guardar nuestros proyectos.



#### Cerramos la ventana de bienvenida dándole a la x.

🖨 Curso de Java - Eclipse IDE					-		×
Eile Edit Source Refactor Navigate S	e <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> el; ▲ ▼   ∰ @ ▼   <u>@</u> @ <i>A</i> / ▼   ½]	╸ ╺╶╗╺ゃ┾╭┝╺╺┍┑╺│┏╴				۹ ا	8
Package Explorer 💥 🖵 🗖					🗐 Task List 🛛	<b>6</b> 9	- 0
There are no projects in your workspace. To add a project:					× 🎋 🖻   🗐 8	)	
Create a Java project           Create a project					Find •	All 🕨	Activ
▶∎ Import projects							
					📴 Outline 🔀		
					There is no active	e editor	that
					provides an outli	ine.	
	🛐 Problems 🛛 @ Javadoc 📴 🛙	Declaration			, ,	7 8	
	0 items		-				
	Description	Resource	Path	Location	Туре		
							0





## Introducción (Vídeo 3)

¿De dónde ha salido?

Características principales.

 Año 1991: un grupo de ingenieros de Sun Microsystems (liderados por James Gosling y Patrick Naughton) tenia que desarrollar un lenguaje de programación que se pudiera utilizar en pequeños electrodomésticos.



- Código pequeño
- Código compacto
- Neutro respecto arquitectura

En principio el nombre que se le puso fue OAK que significa roble ya que desde el despecho había una ventana en la que se divisaba un gran roble, pero este nombre ya estaba reservado y decidieron poner JAVA porque en aquel despacho solían tomar café y esto llama tomar Java.

Fracaso comercial

- De 1991 a 1994 intentaron vender la tecnología a diferentes empresas del ámbito tecnológico. No hubo éxito.
- El proyecto liderado por Gosling y Naughton queda en stand-by.
- Corría el año 1994 e Internet se hacía más grande... Pensaron que las características de Java se ajustaban como un guante a la naturaleza de Internet.



Alternativa

Hicieron un navegado llamado HotJava.

• Inventemos un navegador ligero, capaz de correr en cualquier plataforma y que pueda ejecutar código en su servidor (Applets).

Características de Java

- Sencillo: en el sentido de que se intentó quitar las características más engorrosas de otros lenguajes de programación (artimética de punteros, ficheros de encabezado, etc) y sobre todo que ocupa poco.
- Orientado a objetos.
- Distribuido: buen tratamiento de redes. Buena programación para Internet.
- Seguro: Como fue pensado para programar en red (Internet) se hizo seguro.
  - $\circ$   $\;$  Leer o escribir ficheros sin permiso.
  - Desbordar la pila de ejecución.
  - Corrupción de memoria.
- Neutro.
- Adaptable: tipo de datos primitivos iguales en todas las plataformas.
- Interpretado.
- Alto rendimiento.

Ideas erróneas

- Java no es una extensión de HTML.
- Java no tiene nada que ver con JavaScript.
- Todos los programas de Java se ejecutan en una página web.
- Java es inseguro.
- Java es 100% seguro.





# Estructuras principales I (Vídeo 4)

Tipos de programas Java

- Aplicaciones de consola.
- Aplicaciones de propósito general
- Applets

Las aplicaciones de consola se verán en una ventana de comandos.



También se sutiliza la consola de Eclipse.

Aplicaciones de propósito general son aplicaciones realizadas con Java, el mismo Eclipse es un programa que está realizado con Java.

Applets: son programas realizados en Java que se ejecutar en un navegador como un plugin.



En algunas páginas web se tienen que cargar el plugin de Java para que una parte de la página web pueda funcionar.

Vamos a empezar por las aplicaciones de consola.



Case sensitive distingue entre mayúsculas y minúsculas.

Public: Modificador de acceso.

Class: clase (Todo programa tiene que estar dentro de una clase, está delimitadas por { } llaves.

Ahora vamos a ejecutar Eclipse.

Vamos a crear un proyecto.

•	Curso de Java - Eclipse IDE					-	- C	) ×
File	Edit Source Refactor Navigate Se	arch Project	Ru	n Window Help				
	New A	Alt+Shift+N >	鬯	Java Project			Q	8
	Open File			Project Create a Java project	- 8	Task List	23	- 8
	Open Projects from File System		<del>ين</del> ه	Package		A + 1 😪	9- 8	
	D 1 51			2				

Del menú File seleccionamos New y de este Java Project.

New Java Project	>	<	
Create a Java Project Create a Java project in the workspace or in an external location.			7
Project name: PrimerosPasos			]
Use <u>d</u> efault location		B <u>r</u> owse	
JRE			
• Use an execution en <u>v</u> ironment JRE:	JavaSE-14	~	
○ U <u>s</u> e a project specific JRE:	jdk-14.0.2	$\sim$	
O Use def <u>a</u> ult JRE 'jdk-14.0.2' and workspace compiler preferences		Configure JREs	

Como nombre del proyecto PrimerosPasos.

Project layout O Use project folder as root for sources and class files				
Or Create separate folders for sources and class files			Cor	figure default
Working sets				
Add project to working sets				Ne <u>w</u>
Working sets:			~	S <u>e</u> lect
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cancel

Seleccionaremos el botón Finish.

New module-info.java				Х
Create module-info.java	y start with a lov	vercase le	tter <b>G</b>	
Module name: PrimerosPasos				
<u>G</u> enerate comments (configure templates and default value <u>here</u> )				
	<u>C</u> reate		<u>D</u> on't Crea	ate

## Seleccionaremos el botón Create.

Curso de Java - PrimerosPasos/src/module-info.java - Eclipse IDE		– 🗆 X
<u>File Edit Source Reactor Navigate Search Project Kun Window H</u> elp		
🗂 ▼ 🗔 🕼 i 🍄 🍠 📴 🗉 👖 i 🔌 i 🏇 ▼ 🔾 ▼ 🎴 ▼ 🤮 ▼ 🎒 🧭 ▼ 🎒 🖉 ▼ 🧏 ▼ 🖗 ▼ 🖓 ▼ i 🧏 ▼ 🖓 ▼ i 🦓 ▼ i 🦓	1	Q । 😰 🖏
😫 Package Explorer 🔀 🖓 🗖 🚺 module-info.java 🔀	- 0	🗐 Task List 🛛 🗖 🗖
📄 😒 🖇 🔢 1 module PrimerosPasos {	^	🖆 🕶 🔚 😭 📄
✓ C→ PrimerosPasos		× 🛝 🗖 🚳
> 📉 JRE System Library [JavaSE-14]		<u>~</u> 10r ⊡   @
V 🕮 src		8
> 🚺 module-info.java		Find All Activ

📙   🛃 🖛 🗧 Curso de Java						_	o x
Archivo Inicio Compartir Vista							^ ዋ
Cortar	🖕 Mover a 👻 🗙	Eliminar 🝷	<b>Ⅰ</b> <sup>™</sup> -			Seleccionar No seleccio	r todo mar nada
Anclar al Copiar Pegar Acceso rápido	🛄 Copiar a 👻 💻	Cambiar nombre	Nueva carpeta	Propiedades		Invertir sele	ección
Portapapeles	Organ	nizar	Nuevo	Abrir		Seleccior	nar
← → ~ ↑ 📙 > Este equipo > Disco local (D:)	→ Curso de Java →	•		~	ر ن	D Buscar (	en Curso <mark>d</mark>
Escritorio ^ Nombre	`	Fecha de modi	ificación Ti	ро	Tamai	ĩo	
📰 Imágenes 🛛 🔤 .metadata		26/08/2020 18:	09 C	arpeta de archivo	s		
👌 Música 📃 PrimerosPasos 🗲		26/08/2020 19:	51 C	arpeta de archivo	s		
🧊 Objetos 3D							
😝 Vídeos 🗸 🗸							
2 elementos							

#### Si vamos a la carpeta que creamos para los proyectos de Java.

Ya ha creado la carpeta.

Primeros	Pasos						-		:
Ar E vo H Anclar al Acceso rápido	artir Vista V K Cortar S Copiar ruta de acceso Pegar acceso directo	Mover a 🔹 🕽	K Eliminar ▼ ∰ Cambiar nombre	Nueva carpeta	Propiedades	II - 2 ↓	Seleccionar	r todo nar nada ección	<b>C</b> E
Portapa	apeles	Orga	anizar	Nuevo	Abrir		Seleccio	nar	
← → · ↑ 📴 > Este equipo > Disco local (D:) > Curso de Java > PrimerosPasos > · v Ö 🖉 Busci						en Primero			
👌 Música 🔷 🐴	Nombre		Fecha de mod	ificación 1	īpo		Tamaño		
🗊 Objetos 3D	.settings		26/08/2020 19:	:51 (	Carpeta de archiv	vos			
📲 Vídeos	bin		26/08/2020 19:	:52 (	Carpeta de archiv	vos			
indows (C:)	src		26/08/2020 19:	:52 (	Carpeta de archiv	vos			
Disco local (D:)	.classpath		26/08/2020 19:	:51 A	Archivo CLASSPA	ATH	1 KB		
💣 Red	project		26/08/2020 19:	:51 A	Archivo PROJEC	Г	1 KB		
5 elementos									

Con subcarpetas y archivos necesarios para el proyecto.

## Ahora desde Eclipse

Curso de Java - PrimerosPasos/src/module-info.java - Eclipse IDE									
<u>File Edit Source Refactor Navigate Sea</u>	rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp								
🔁 🕶 🔚 🕼 📪 🍠 🖓 🗐 🔳 🐄	- O - G - G - 🖶 🐨 🐨 😂 🖨 🛷 - 🗄 - 🕅 - 🏷 🗘 - 🔿 - 🖂	<ul> <li>२ । 😰 । 🐯</li> </ul>							
😫 Package Explorer 🔀 📃 🗖	🚺 module-info.java 🔀	🗖 🗖 Task List 🔀 🗖 🗖							
E S 8 > ⊯ PrimerosPasos	1 module PrimerosPasos { 2 }	<ul> <li>↑ </li> <li></li></ul>							

Hacemos un clic en el proyecto y ahora del menú File seleccionaremos New y de este Class.

🖨 New Java Class	_		×	
Java Class . The use of the de	efault package is discouraged.		3	
Source fol <u>d</u> er:	PrimerosPasos/src	Bro	wse	
Pac <u>k</u> age:	(default)	Bro	<u>w</u> se	
Enclosing t <u>y</u> pe:		Bro	<u>w</u> se	
Na <u>m</u> e: Modifiers:	PrimeraClase     Image: Constraint of the state of the st			
Superclass:	java.lang.Object	Bro	ws <u>e</u>	
Interfaces:		<u>A</u> d	dd	
		<u>R</u> er	nove	
Which method stub	<ul> <li>would you like to create?</li> <li>public static void main(String[] args)</li> <li>Constructors from superclass</li> <li>Inherited abstract methods</li> </ul>			
Do you want to add	comments? (Configure templates and default value <u>here</u> )			
?	<u> </u>	Ca	ncel	

Como nombre de la clase PrimeraClase, seguido del botón Finish.

Curso de Java - PrimerosPasos/src/Prime	aClase.java - Eclipse IDE		- 🗆 ×
<u>File Edit Source Refactor Navigate S</u>	e <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp		
🔁 🕶 🔚 🕼 🚏 🗾 🕅 🔍	▓ ▾ ◑ ▾  ▾  ▾! #° ☞ ♥ 😂 😂 🖋 ▾! 월 ▾ 정 ▾ ♡ ♡ ▾ ▷ ▾   📷 .		Q 🗄 😭 🛛 🐉
🕆 Package Explorer 🔀 🗖 🗖	🚺 module-info.java 🚽 PrimeraClase.java 🔀		🗐 Task List 🛛 🗖 🗖
PrimerosPasos	1     2 public class PrimeraClase {	^ =	
> ➡ JRE System Library [JavaSE-14]	3 4 }		× 👫 🖻 🔞
✓ 进 src	5		8
V 🛃 (default package)			Find  All  Activ
> D module-info.java			

Ya se ha creado la clase.



Ahora del menú File seleccionamos Save.

Si vamos a la carpeta de nuestros proyectos y entramos a la carpeta PrimerosPasos.

Archivo Inicio Compat	sos						-		×
Anclar al Acceso rápido	Cortar Copiar ruta de acceso Pegar acceso directo	Mover a ▼	¥ Eliminar ▼ ■ Cambiar nombre	Nueva carpeta	Propiedades	<b>-</b> 2	Seleccion No selecc	ar todo ionar nac lección	da
← → × ↑ → Este	eles equipo → Disco local (D:)	○ → Curso de Jav	rganizar a > PrimerosPasos	Nuevo	Abrir	ō	Selecci	onar r en Prim	nero
👌 Música 🔷 🐴	Nombre	×	Fecha de mod	lificación T	ipo		Tamaño		
🧊 Objetos 3D	.settings		26/08/2020 19	:51 C	arpeta de archiv	vos			
Vídeos	bin		26/08/2020 20	:08 C	arpeta de archiv	vos			
🏪 Windows (C:)	src		26/08/2020 20	:08 C	arpeta de archiv	vos			
Disco local (D:)	classpath		26/08/2020 19	:51 A	rchivo CLASSPA	ATH	1 KB		
💣 Red	] .project		26/08/2020 19	:51 A	rchivo PROJECT	Г	1 KB		
5 elementos 1 elemento se	eleccionado							No.	

Vamos a ver el contenido de las carpetas src y bin.

#### En src

📊   🛃 📊 🖛   sre	c							- 🗆	×
Archivo Inicio	Compar	rtir Vista							~ ?
Anclar al Acceso rápido	Pegar	X         Cortar           Image: Copiar ruta de acceso         Image: Copiar ruta de acceso           Image: Pegar acceso directo         Image: Copiar ruta de acceso directo	Mover a 🔹 🗙	🕻 Eliminar 👻	Nueva carpeta	Propiedades	₽ - 2 €	Seleccionar todo	ja
	Portapap	peles	Orga	nizar	Nuevo	Abrir		Seleccionar	
$\leftrightarrow$ $\rightarrow$ $\checkmark$ $\uparrow$	> Este	equipo > Disco local (D:)	> Curso de Java	> PrimerosPasos >	src	~	Ū	℅ Buscar en src	
🁌 Música	^	Nombre		Fecha de mod	ificación T	ipo		Tamaño	
🧊 Objetos 3D		📄 module-info.java		26/08/2020 19:	:52 A	rchivo JAVA		1 KB	
😽 Vídeos		🌓 PrimeraClase.java		26/08/2020 20:	:17 A	rchivo JAVA		1 KB	
🏪 Windows (C:	)								
Disco local (I 2 elementos	):) ¥								

El proyecto con extensión java.

En bin

Image: Archivo     Inicio	n Compa	rtir Vista				_		-		× ^ ?
Anclar al Acceso rápido	Pegar Portanai	X Cortar Copiar ruta de acceso Pegar acceso directo	A Mover a 🔻	X Eliminar ▼ ➡ Cambiar nombre	Nueva carpeta	Propiedades		No seleccion	ar todo ionar nad Iección	a
$\leftarrow \rightarrow \circ \uparrow \square $ > Este equipo > Disco local (D:) > Curso de Java > PrimerosPasos > bin $\lor \circlearrowright$										
<ul> <li>Música</li> <li>Objetos 3D</li> <li>Vídeos</li> <li>Windows (C;)</li> </ul>	)	Nombre module-info.class PrimeraClase.class		Fecha de mod 26/08/2020 19 26/08/2020 20	lificación :52 :17	Tipo Archivo CLASS Archivo CLASS	-	Tamaño 1 KB 1 KB		
Disco local (D 2 elementos	);) ¥									

Ya nos ha creado el archivo compilado, recuerda que es multiplataforma.

Desde Elipse vamos a habilitar la consola, para ello del menú Windows seleccionamos Show View y de este console.



Es donde se verá el resultado de nuestros proyectos.

Ejecutamos el proyecto.



Este será el resultado:







# Estructuras principales II (Vídeo 5)

Tipos de datos en Java (Tipos primitivos)

- Enteros
  - Int: 4 bytes de espacio para almacenamiento. Desde -2.147.483.648 hasta 2.147.483.647.
  - Short: 2 bytes de espacio para almacenamiento. Desde -32.768 hasta 32.767.
  - o Long: 8 bytes de espacio para almacenamiento. Una barbaridad. Sufijo L.
  - Byte: 1 byte de espacio para almacenamiento. Desde -128 hasta 127.
- Coma flotante (decimales)
  - Float: 4 bytes de espacio para almacenamiento. Aproximadamente 6 a7 difras decimales significativas. Sufijo F.
  - Double: 8 bytes de espacio para almacenamiento. Aproximadamente 15 cifras decimales negativas.
- Char: Para representar caracteres.
- Boolean: True o False.

Variables en Java

- ¿Qué es una variable? Espacio en la memoria del ordenador donde se almacenará un valor que podrá cambiar durante la ejecución del programa.
- ¿Por qué hay que utilizar variables? Porque a la hora de crear un programa surge la necesidad de guardar datos temporalmente que necesitarás utilizar en el futuro en ese mismo programa.
- ¿Cómo se crea una variable en Java? Especificando el tipo de dato que almacenará en su interior + el nombre de la variable. P. ej: int salario;
- ¿Qué es iniciar una variable? Es darle valor. Nombre\_variable=valor. P. ej: salario=2000; Java no permite utilizar variables que no se hayan iniciado.





Tipos de variables que se utilizan con mayor frecuencia:

• Int, Double, Char y Boolean





# Estructuras principales III. Declaración variables Eclipse (Video 6)

Vamos a crear una nueva clase.

🖨 Curso de Java - Eclipse IDE		– 🗆 🗙
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> a	vigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
📑 🕶 🗟 🕼 🖉 🖛 🕶 🖸 🕶	<b>♀</b> ▼ <b>♀</b> ⇒ ⊜ <b>→</b> ⊜ <b>⇔</b> <i>⊘</i> <b>→</b> ⊴ <b>⇒</b> ⊕ <b>→</b> ⇔ <b>→</b>   <u>⇒</u>	ର୍ 🔡 💐
🛱 Package Explo 🔀 🖳 🗖		🗐 Task 🛛 🗖 🗖
E 😫 🕴	$\mathbf{i}$	🖆 🕶   🐮 🕼
> 🔁 PrimerosPasos		× 🎋 🖂 🚳
		🗄 Outline 🛛 🗖 🗖
		<b>6</b> 9 8
		There is no active editor
	🖹 Problems @ Javadoc 🚯 Declaration 📮 Console 🔀	🛃 🖻 🕶 📑 🗖
	No consoles to display at this time.	
		: <mark>0</mark>

Seleccionaremos el correspondiente botón.

🖨 New Java Class		_		×
Java Class	efault package is discouraged.		C	
Source fol <u>d</u> er:	PrimerosPasos/src		Br <u>o</u> wse	
Pac <u>k</u> age:		(default)	Bro <u>w</u> se	
Enclosing type:			Bro <u>w</u> se	
Na <u>m</u> e: Modifiers:	Variables       ● public     ○ package     ○ private     ○ private       □ abstract     □ final     □ static	o <u>t</u> ected		
<u>S</u> uperclass:	java.lang.Object		Browse	
<u>I</u> nterfaces:			<u>A</u> dd <u>R</u> emov	/e
Which method stut				
?	<u> </u>	sh	Cance	:

A esta clase la vamos a llamar Variables, activamos la casilla public static void main(String[] args). Seguido del botón Finish.

Curso de Java - PrimerosPasos/	src/Variables.java - Eclip	se IDE			_		×
<u>File Edit Source Refactor Na</u>	wigate Se <u>a</u> rch <u>P</u> roje	ct <u>R</u> un <u>W</u> indow	<u>H</u> elp				
📑 🗝 🔚 🕼 🔌 🕸 🕶 🔘 🕶	🍳 🕶 💁 🖬 🥵	) - 🤔 🗁 🛷 -	ं 🍄 🌛 📴 🗊 🖞 🝷	· 😽 🗕 *	þ 🔶 🗸 d	ə 🚽 🗖	1
						Q : 🖻	}   嬰
<ul> <li>Package Explo X</li> <li>PrimerosPasos</li> <li>JRE System Library [JavaSI</li> <li>Src</li> <li>(default package)</li> <li>PrimeraClase.java</li> <li>Variables.java</li> </ul>	Variables.java &          1         2         4         4         9         10    Image: Contract of the second seco	s Variables { static void mai TODO Auto-gener vadoc Q Declarati y at this time.	n(String[] args) { ated method stub	· · · · · · · · · · · · · · · · · · ·	Image: Task.         Image: Task.	Q : E ☆	Image: String[])       Image: String[])       Image: String[])
< >							
Writ	able Sm	art Insert	1:1:0				: 🖓

Vamos a declara e inicializar una variable.

Curso de Java - PrimerosPasos/s	rc/Variables.java - Eclipse IDE		_		×
<u>File Edit Source Refactor Nav</u>	igate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp				
➡ <b>-</b> 🖫 🕼 ! 🍬 ! 泰 <b>- O -</b>	Q_ ▼ Q_ ▼   ₩ @ ▼   ഈ (⇒ 🥔 ▼   ♀) № [[] π   ½ ▼ ∕	¶ <b>- </b> ⊀	> <> -	⇒ -   ⊡ Q   ⊞	1 }   歌
🛱 Package Explo 🔀 🖵 🗖	🚺 *Variables.java 🔀		🗐 Task .	🛛	
<ul> <li>PrimerosPasos</li> <li>A. JRE System Library [JavaS]</li> <li>Src</li> <li>Interpretation (default package)</li> <li>PrimeraClase.java</li> <li>Variables.java</li> </ul>	<pre>1 2 3 4 9 public class Variables { 4 4 0 public static void main(String[] args) { // TODO Auto-generated method stub byte edad; 7 8 edad=35; 9 10 System.out.println(edad); 11 } 12 13 14</pre>		C ← C ← C ← C ← C ← C ← C ← C ← C ← C ←	te S Variables S main()	5   .   ► . String[])
	<	>	<		>
	🖹 Problems @ Javadoc 😫 Declaration 🛢 Console 🛛			- 📬 🛨	
< >>	No consoles to display at this time.				
					: 🔉

Le vamos a dar al Play sin haber guardado la clase.

Save and Launch	
Select resources to save:	
Variables.java	
<u>S</u> elect All	Deselect All
Always save <u>r</u> esources before	launching
	Cancel
UK UK	Cancel

Nos preguntará si queremos guardar la clase a lo que contestaremos OK.

Curso de Java - PrimerosPasos/s	rc/Variables.java - Eclipse IDE		– 🗆 ×
<u>File Edit Source Refactor Nav</u>	rigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp		
📩 🗕 🔚 🐚 🖉 🔅 🗸 🚺	🍳 + 💁 + 😰 🧭 + 🎥 🗁 🛷 + 👎 🗾 💀 🗐 👖 읽 + 🧞	- 🌾	⊃ (> ▼ <> ▼   🛃
			२ 🔡 🐯
🛱 Package Explo 🛛 🗖 🗖	🕽 Variables.java 🛛		🗐 Task 🛛 🗖 🗖
E 🔁 🖇	1 2 public class Variables (	^	🖆 🗕 📴 🐨
✓  PrimerosPasos			× 🕅 🗖 🚳
> ■ JRE System Library [JavaSł ∨ 伊 src	4⊖ public static void main(String[] args) { 2 5 // TODO Auto-generated method stub		8
🗸 🌐 (default package)	6 byte edad;		Find All A
> 🚺 PrimeraClase.java	7 8 edad-35:		
> 🚺 Variables.java	9		
	<pre>10 System.out.println(edad);</pre>		
	12		
	13 }		🗄 Outline 🛛 🗖 🗖
	14		🗊 🖹 🖓 🙀 🖓 🕒
			×L
			8
			Variables
			S main(String[])
		$\sim$	
	< >>		< >
	📳 Problems @ Javadoc 😟 Declaration 📃 Console 🔀		- 8
		1	🗩 🚇 🚽 🖻 👻 📑 👻
	<terminated> Variables [Java Application] C:\Program Files\Java\idk-14.0.2\b</terminated>	oin\java	aw.exe (27 ago. 2020 7:41:09
	35		^
			~
< >	<		>
Writa	ble Smart Insert 7 :127		l 😯

En la consola ya obtenemos el resultado.



En la misma línea declaramos e inicializamos la variable edad.



Cambio el valor de la variable edad durante la ejecución del proyecto.

🖨 Curso de Java - PrimerosPas	sos/src/Variables.java -	Eclipse IDE			_	
<u>File Edit Source Refactor</u>	<u>N</u> avigate Se <u>a</u> rch	<u>P</u> roject <u>R</u> un <u>W</u> indo	w <u>H</u> elp			
📑 🗝 🗐 🕼 🔌 🎋 🕶 🚺	) - 💁 - 💁 - 🔒	° 🕝 🕶 🥭 🗁 🔗	🝷 📪 🗾 🖬 🔲 п	ं 👰 👻 😽 👻 📢	- 🔶 🗸 🖒	-
					Q	C 🖻 💐
🛱 Package Explo 🛛 🗖	Variables.java	1 22			Task	X - 0
► 😓 V 🔁 PrimerosPasos	8 1 2 public	c <b>lass</b> Variables {		^		¶≣  <b>®</b>   
> 🛋 JRE System Library [Jav ✔ 进 src	vaSE 4⊕ pub.	lic static void m // TODO Auto-gen	ain(String[] args) erated method stub	{	8	1
✓	va 7	int edad=35;			Find	► All ► .
> 🚺 Variables.java	9 10 11 12	System.out.print edad=75;	ln(edad);			
	13 14 15 } 16	System. <i>out</i> .println(edad);		Bar Outline Sa I III Ja N	≍ □ □ × × ●	
	17 7 18				8 ✓ <b>G</b> ⊾ Va ● <sup>S</sup>	riables main(String[])
				~		
				-	<b>`</b>	>
	Roblems (	@ Javadoc 🗔 Declar	ation 📮 Console 🔀			
			■ 🗙	💥 🗟 🔊 🕅	5 🚱 🛃	🗉 🔻 📑 🔻
	<terminated> Va</terminated>	riables [Java Applicatio	n] C:\Program Files\Java	\jdk-14.0.2\bin\java	aw.exe (27 ag	go. 2020 7:50:40
	35 75					^
						× .
	2		0 122			>
V	Vritable	Smart Insert	8:133			2

Java es un lenguaje orientado a objetos POO.

Los objetos tienen propiedades y métodos.



Ejemplo de comentario de una y de varias líneas.





# Estructuras principales IV. Constantes y Operadores (Vídeo 7)

- Declaración de constantes
- Operadores

## Constantes en Java

- ¿Qué es una constante? Espacio en la memoria del ordenador donde se almacenará un valor que no podrá cambiar durante la ejecución del programa.
- ¿Por qué hay que utilizar constantes? Porque a la hora de crear un programa a veces surge la necesidad de guardar datos temporalmente que necesitarás utilizar en el futuro en ese mismo programa. Dichos datos deberán ser fijos.
- ¿Cómo se crea una constante en Java? Utilizando la palabra clave final y a continuación especificando el tipo de dato que almacenará en su interior + el nombre de la consante = valor. P. ej: final double a\_pulgadas=2,54;
- Recuerda: El valor de una constante no podrá cambiar.

## Operadores en Java

- Aritméticos
  - + : suma
  - o -: resta
  - \* : multiplicación
  - o /: división
- Lógicos, relacionales y booleanos
  - o > : mayor que
  - o <: menor que
  - <> : mayor o menor que
  - $\circ$  != : distinto que
  - == : igual que
  - o &&: y lógico
  - ||: o lógico
- Incremento y decremento
  - ++ : Incremento
  - o --: Decremento
  - +=nº : incremento
  - o -= nº : decremento
- Concatenación
  - + : une o concatena.

Ejecutamos Eclipse para crear una clase nueva.

Curso de Java - PrimerosPasos/src/Variables.java - Eclipse IDE		_		×
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp				
🛅 = 🗑 🔞 🔌 = 🔿 = 💁 = 🏰 🎯 = 🎒 😂 🖉 = 👘 🖉 =	招▼∜	- (-)	> -   🗖	1
New Java Class		(	২ ়≣	む
😫 Package Explo 🕱 🖓 🗖 🗋 🚺 Variables.java 💥		🗐 Task	×	- 0
PrimerosPasos ■ 2 a public class Variables []	^		5 9=   5   <b>43</b>	<b>6</b> 9

🖨 New Java Class			×
Java Class	efault package is discouraged.	C	
Source fol <u>d</u> er:	PrimerosPasos/src	Br <u>o</u> wse	
Pac <u>k</u> age:	(default)	Bro <u>w</u> se	
Enclosing type:		Bro <u>w</u> se	
Na <u>m</u> e: Modifiers:	Declaraciones_Opeadores		
Superclass:	java.lang.Object	Browse	
<u>I</u> nterfaces:		<u>A</u> dd <u>R</u> emov	/e
Which method stu Do you want to add	would you like to create?     public static void main(String[] args)     Constructors from superclass     Inherited abstract methods     comments? (Configure templates and default value here)     Generate comments		
?	<u> </u>	Cance	1

Seleccionaremos el botón Finish.



Curso de Java - PrimerosPasos/src/Declaracione	s_Opeadores.java - Eclipse IDE	– 🗆 ×
<u>File Edit Source Refactor Navigate Search</u>	<u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
	#`@`+`@>@~?`*`\$?]] @ @ 1 :}! + ?: + ?: <> +   ♂	ର୍ 🖻 🕏
😫 Package Explorer 🛛 📄 😫 🖇 🖵 🗖	🕐 Variables.java 🕼 Declaraciones_Opeadores.java 🛛 🗖 🗐 Ta	ask List 🛛 🗖 🗖
<ul> <li>PrimerosPasos</li> <li>Al JRE System Library [JavaSE-14]</li> <li>PrimeraClase]</li> <li>J Declaraciones_Opeadores.java</li> <li>J PrimeraClase.java</li> <li>Variables.java</li> </ul>	<pre> 1 public class Declaraciones_Opeadores [] 3  4  public static void main(String[] args) { 5 // TODO Auto-generated method stub 6 int a=5; 7 int b; 8 b=7; 9 int c=b+a; 10 System.out.println(c); 11 } 13 Problems @ Javadoc @ Declaration © Console 🖾 🖷 💥 👰 🗗 🖉 🖉 🖉 14 15 15 15 15 15 15 15 15 15 15 15 15 15</pre>	Image: Second
	<terminated> Declaraciones_Opeadores [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe</terminated>	e (27 ago. 2020 9:59:04 – 9:
	12	^
		× .
		>
	Writable Smart Insert 12:2:200	8

Si este punto está en rojo nos está diciendo que está compilando el programa.

Curso de Java - PrimerosPasos/src/Declaracione	s_Opeadores.java - Eclipse IDE	– 🗆 X
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch	<u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
📑 🕶 🔚 🐚 🔌 🎋 🕶 💽 🕶 💁 🕶	# @ - @ @ @ ~ + ₽ 🍠 💷 🗉 🖷 😓 - ╄ - ┺ ↔	Q i 🖻 😽
😫 Package Explorer 🛛 📄 😒 🖇 🗂 🗖	🕽 Variables.java 👔 Declaraciones_Opeadores.java 🛛 🗖	🗐 Task List 🛛 🗖 🗖
Image: PrimerosPasos         > ■ JRE System Library [JavaSE-14]         > ■ JRE System Library [JavaSE-14]         > ■ Gefaraciones_Opeadores.java         > □ PrimeraClase.java         > □ Variables.java	<pre> 1 2 public class Declaraciones_Opeadores { 3 4 4     public static void main(String[] args) {         // TODO Auto-generated method stub         6         int a=5;         int b;         b=7;         int c=b+a; 10         c++; 11         System.out.println(c); 12         } 13     } 4 </pre>	<sup>•</sup> ▼ 1 2 <sup>•</sup> 2 <sup>•</sup> 2 <sup>•</sup> 1 <sup>•</sup> 1 <sup>•</sup> ▼ 1 <sup>•</sup> 1 <sup>•</sup> ■ 1 <sup>•</sup> 4 <sup>•</sup> ■ 1 <sup>•</sup> All ▶ Act <sup>±</sup> ■ 1 <sup></sup>
	🖹 Problems @ Javadoc 😥 Declaration 📮 Console 🛛 🔲 🗮 💥 🗎 🛃 🛃	
	<terminated> Declaraciones_Opeadores [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\java</terminated>	vaw.exe (27 ago. 2020 10:01:39 – 1
	15	^
		×
		2
	Writable Smart Insert 10:13:175	<b>·</b>

Incremento en 1.

Curso de Java - PrimerosPasos/src/Declaracione	s_Opeadores.java - Eclipse IDE	- 🗆 X
<u>File Edit Source Refactor Navigate Search</u>	<u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
🔁 • 🗐 🕼 🔌 🕸 • 💽 • 🎴 •	# @ - @ @ @ ? - ₽ 🥖 🖻 🗉 👖 🖢 - २ - २ - 1 🛃	Q i 🖻 🛃
😫 Package Explorer 💥 📄 🗟 🖇 🗖 🗖	🚺 Variables.java 🚺 Declaraciones_Opeadores.java 💥	🗆 📳 Task List 💥 👘 🗖
Image: PrimerosPasos         > ▲ JRE System Library [JavaSE-14]         > ▲ JRE System Library [JavaSE-14]         > ● frimerosPasos         > ● CD Declaraciones_Opeadores.java         > ① PrimeraClase.java         > ② Variables.java	<pre> 1 2 public class Declaraciones_Opeadores { 3 4 4      public static void main(String[] args) {</pre>	• • • • • • • • • • • • • • •
	Problems @ Javadoc 😡 Declaration 🚍 Console 🛛 🔳 💥 🎇 🖳 🔐 🔐	
	<terminated> Declaraciones_Opeadores (Java Application) C:\Program Files/Java\gdk-14.0.2\b 18</terminated>	in yavaw.exe (27 ago. 2020 10:04:53 – 1
	Writable Smart Insert 14:1:211	0

Le incrementamos en 6.

Curso de Java - PrimerosPasos/src/Declaraciones_Opeadores.java - Eclipse IDE	-		×
<u>File Edit Source Refactor Navigate Search Project Run Window Help</u>			
🖻 • 🗟 🐚 🔌 🕸 • 🗛 • 🇣 • 🗣 🎯 • 🤔 🖉 🕫 🖉	2 ■ 1 ↓ ▼ ↓ ▼ ↓ ▼ ↓ ▼ ↓ ▼	-	

Cuando le damos al play y no hemos guardado el programa aparece una ventana para que confirmemos que nos guarde los cambios.

Save and	Launch			×
Select resour	ces to save:			
🗹 🕖 Decla	araciones_Op	eadores.j	ava	
	<u>S</u> elect A	All	Deselect /	All
Always sav	ve <u>r</u> esources	before lau	unching	
?	OK		Cance	ł

Si activamos "Always save resources before launching" ya no nos lo preguntará más.

Curso de Java - PrimerosPasos/src/Declaraciones_Opeadores.java - Eclipse IDE					
<u>File Edit Source Refactor Navigate</u>	Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp				
📑 <del>-</del> 🔚 🐚 🔌 🎋 - 🔿 - 💁 -	🌯 र 🏦 🎯 र 🅭 🗀 🛷 र 📪 🍠 💷 🗉 🦷 😓 र 🏷 🔶 र	⇒ ▼			
		Q 🗄 🖻 🛛 🐯			
🛱 Package Explorer 🔀 📃 🗖	🕽 Variables.java 🚺 *Declaraciones_Opeadores.java 🔀 🗖 🗖	🛛 📳 Task List 💥 🖳 🗖			
<b>三</b> 冬 8		🕆 🕇 🔁 🕼 🚺			
<ul> <li>✓ PrimerosPasos</li> <li>&gt; JRE System Library [JavaSE-14]</li> <li>✓ B src</li> <li>✓ (datall package)</li> <li>&gt; Declaraciones_Opeadores.ja</li> <li>&gt; PrimeraClase.java</li> <li>&gt; Variables.java</li> </ul>	<pre>2 public class Declaraciones_Opeadores { 3 4 public static void main(String[] args) { 5 // TODO Auto-generated method stub 6 int a=5; 7 int b; 8 b=7; 9 int c=b+a; 10 // c+=6; 11 System.out.println(c); 12 } 13 }</pre>	<ul> <li>▲ ●</li> <li>● ●</li> <li>● ●</li> <li>● ●</li> <li>● ●</li> <li>● □</li> <li>● ■</li> <li>● □</li> <li>● □</li></ul>			
	🖹 Problems @ Javadoc 😣 Declaration 📮 Console 🔀				
	🔲 🗶 💥 🖹 🗐	🥺 📮 🚝 📑 🖻 🔻 🔂 🔻			
	<terminated> Declaraciones_Opeadores [Java Application] C:\Program Files\Java\jd</terminated>	k-14.0.2\bin\javaw.exe (27 ago			
< >	<	> ''			
Writable	Smart Insert 12 : 6 : 209	: <mark>0</mark>			

Con // podemos comentar una línea de código para que esta no se ejecute, así no la tenemos que borrar y en un futuro quitando las // vuelve a estar en el código.

Curso de Java - PrimerosPasos/src/Decla	raciones Opeadores.ia	va - Eclipse IDE				_		×
File Edit Source Refactor Navigate Search Project Run Window Heln								
Ine Luic Source velacion Travilate Search Endert Frui Mundow Teb The Luic Source velacion Travilate Search Endert Frui Mundow Teb								
Package Explorer       Image: Constraint of the second secon	✔       ✔         ↓       ↓	Declaracio s Declaracio s Declaracio static void topo Auto-gr topo	nes_Opeadores.java ones_Opeadores 🖟 main(String[] args enerated method stu ttln(c); aration 🚍 Console res [Java Application] C:	) {		□       Task Lis         □       -         □       -         0       -     <	eclaracic main(S	anes (27 ago
<	<							>
Writable	Smart	nsert	12:2:209					8

Como esta división va a tener decimales esta tiene que ser de tipo double, pero además todas las variables que intervienen en el cálculo también tienen que ser de tipo double.





## Estructuras principales V. Constantes y Operadores II (Vídeo 8)

En el siguiente ejemplo vamos a definir una constante esta se declara poniendo al principio la clave final, si en el transcurso del programa intentamos cambiar su valor este nos dará error.

Curso de Java - PrimerosPasos/src/Decl	araciones_Opeadores.java - Eclipse IDE		_		×	
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate	Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp					
📑 🗕 🖫 🔌 🔅 🕶 🕒 🖛 🔁 🕶	🌯 + 🖶 🎯 + 🎘 🗁 🛷 + 🚏 🗾 💀 🗐 👖 🕌 + 👬 + 🏷 -	()	- 🔁 🗠			
			C	२ 🗄 🖪	<b>\$</b>	
🛱 Package Explorer 🔀 📃 🗖	🕽 Variables.java 🗊 *Declaraciones_Opeadores.java 🔀		🗐 Task List	t 23 - "		
) 🔁 PrimerosPasos	<pre>1 2 public class Declaraciones_Opeadores { 3</pre>	^ =				
	<pre>4⊖ public static void main(String[] args) { 2 5 6 // TODO Auto-generated method stub final int c=5; 2</pre>					
	<pre>10 System.out.println(c); 11 } 12 } 13</pre>					
	<	>	<	main(St	trina ∴ >	
	Revealed a state of the state o	≓ ⊑ •	<b>*</b> *			
Writable	Smart Insert 9:9:157				0	

#### Si ejecutamos.

Error	rs in Workspace >	<
$\bigcirc$	Errors exist in required project(s):	
	PrimerosPasos	
	Proceed with launch?	
□ <u>A</u> lw	ays launch without asking	
	<u>P</u> roceed Cancel	

#### Nos avisa de que hay un error.

12 } 13						⊫ Uutime & ⊫ ↓ªz ≷ ×	• 🗙 🕯 🖇	
						✓ ₽ Declara P S main	ciones_Opeadores n(String[]) : void	
					~			
<					>			
🖹 Problems @	Javadoc 😥 Declaration	📮 Console 🔀		- × 🗞   🛙	). 🚮 🔛	e 💌 🛃	• 📸 • 🗆 🗖	
<terminated> Dec</terminated>	laraciones_Opeadores [Jav	a Application] C:\Progra	m Files\Java\jdk-1	4.0.2\bin\javaw	.exe (27 ago	. 2020 11:04:52 – 1	11:05:31)	
Exception in thread "main" java.lang.Error: Unresolved compilation problem: The final local variable c cannot be assigned. It must be blank and not using a compound assignment								
at Dec	laraciones_Opeadore	s.main( <u>Declaracion</u> e	s Opeadores.ja	ava:8)			~	
<							>	
	Writable	Smart Insert	9:9:157				8	
Ejemplo de constante.



Definimos una constante con la conversión de pulgadas a cms. Que es 2,54 en una constante de tipo double llamada apulgadas.

Definimos una variable de tipo double con el nombre cm para almacenar el valor 6.

En la variable de tipo double llamada resultado calculamos la división del valor que tiene la variable cm con el valor de la constante apulgadas.

Por último enviamos a la consola la concatenación de texto y el valor de las variables.

Se pueden declarar varias varaibles en una misma línea.

Curso de Java - PrimerosPase	os/src/Declaraciones_Opeadore	s.java - Eclipse IDE			-		×
<u>File Edit Source Refactor</u>	<u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>F</u>	<u>Run W</u> indow <u>H</u> elp					
📑 🗕 📄 🔍 🎋 🗸 🔿	- Q - Q - # @ -	😂 🗁 🛷 👻 🝄 🗾	🖉 🗊 🖷 🐓	<ul> <li>→ </li> <li>→ </li> <li></li> <li></li></ul>	> -   🛃	۹ 🖻	}   🐉
🛱 P 🐹 🖳 🗖 🚺 Varial	oles.java 🚺 Declaraciones	_Opeadores.java 🔀			1 Task List 🔀		
	blic class Declassions	· Onenderer (		^	💣 🕶 📴 🕾	<b>6</b> 9	
> 🔁 PrimerosPasos 2 pt	DIIC CIASS DECIAPACIONE:	s_opeadores {			× 👫 🗖 🚳	)	
40	public static void ma	in(String[] args)	[		8		
6	int operador1, op	erador2, resultado			°		
7	operador1=8;				📴 Outline 🔀		
9	resultado=operado	r1+operador2;			🖻 📮 🏹 🗙	⊖ 🗙	69
10	System.out.printle	n(resultado);			8		
12	3			~	V O Declara	ciones_Ope	eado 👌
<				>	<		> ×
R Probl	ems @ Javadoc 🚯 Declarat	ion 📃 Console 🛛	(	🛛 🗙 🔆 🗎 🗟		- 📬 -	- 8
<termina< td=""><td>ted&gt; Declaraciones_Opeadores</td><td>[Java Application] C:\Pr</td><td>ogram Files\Java\jdk</td><td>k-14.0.2\bin\javaw.exe (</td><td>27 ago. 2020 11:23:10</td><td>- 11:23:12)</td><td></td></termina<>	ted> Declaraciones_Opeadores	[Java Application] C:\Pr	ogram Files\Java\jdk	k-14.0.2\bin\javaw.exe (	27 ago. 2020 11:23:10	- 11:23:12)	
15							\$
< >> <							>
	Writable	Smart Insert	10:39:263				1





# Estructuras principales IV Clase Math (Vídeo 9)

#### Otros cálculos numéricos.

Clase Math

Clases en Java	
Clase en Java Propias Predefinidas	
String	
Array	PLOORASINFORMÁTICAS

Hay clase propias y clases predefinidas.

Clase propia
<pre>public class Declaraciones_Operadores {</pre>
<pre>public static void main(String[] args) {     // TODO Auto-generated method stub</pre>
<pre>int operador1,operador2, resultado;</pre>
operador1=8;
operador2=7;
resultado=operador1+operador2;
<pre>System.out.println(resultado);</pre>
}

Este es un ejemplo de clase propia.

Ejemplos de predefinidas: String, Array, Math, Thread, etc.

API Java es una biblioteca donde vienen todas las clases de programación, muy útil para realizar consultas.

Accederemos al siguiente enlace:

🖸 (166) Curso Java. Estructuras prin 🗴 🥃 Overview (Java	Platform SE 7 ) × +	- 0 ×
← → C  a docs.oracle.com/javase/7/docs/api/		▶ ☆ ◎ 身 司 40 :
Aplicaciones		
Java™ Platform Standard Ed. 7	Overview Package Class Use Tree Dep	Janu <sup>ar</sup> Phatema Reacted Index. Help Readow Ed. 7
All Classes	Prev Next Frames No Frames	
Packages		
java.avvt java.avvt java.avvt.color	API Specification	a Ealaon 7
java.awt.datatransfer java.awt.dnd	This document is the API specification for the Ja	ava** Platform, Standard Edition.
java. avrt.event java. avrt.font java. avrt.geom	See: Description	
java.avvt.im.spi	Packages	
java.awi.image java.awi.image.renderable	* Package	Description
	java.appiet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
All Classes	Java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
AbstractAction AbstractActionValue/Isitor6	java.awt.color	Provides classes for color spaces.
AbstractAnnotationValueVisitor7	java.awc.datablarisier	Provoes interfaces and classes for transmiring data between and within appications.
AbstractBorder	java.awt.dnd	Urag and urop a oreic manipulation gesture tound in many unaprical user intertace systems that provides a mechanism to transfer information between two endities logically associated with presentiation elements in the GUI.
AbstractCellEditor	java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
AbstractColorChooserPanel	java.awt.font	Provides classes and interface relating to fonts.
AbstractDocument AbstractDocument AttributeContext	java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
AbstractDocument Content AbstractDocument ElementEdit	java.awt.im	Provides classes and interfaces for the input method framework.
AbstractElementVisitor6	java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
AbstractExecutorService	java.awt.image	Provides classes for creating and modifying images.
AbstractinterruptibleChannel	java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
AbstractLayoutCache.NodeDimensions	java.awt.print	Provides classes and interfaces for a general printing API.
AbstractList AbstractListModel	java.beans	Contains classes related to developing beans components based on the JavaBeans <sup>TM</sup> architecture.
AbstractMap AbstractMap SimpleEntry	java.beans.beancontext	Provides classes and interfaces relating to bean context.
AbstractMap.SimpleImmutableEntry	java.io	Provides for system input and output through data streams, serialization and the file system.
AbstractMarshallerimpi AbstractMethodError	java.lang	Provides classes that are fundamental to the design of the Java programming language.
AbstractOwnableSynchronizer	java.lang.annotation	Provides library support for the Java programming language annotation facility.
AbstractProcessor	java.lang.instrument	Provides services that allow Java programming language agents to instrument programs running on the JVM.
AbstractQueue AbstractQueuedLongSynchronizer	java.lang.invoke	The java.lang.invoke package contains dynamic language support provided directly by the Java core class libraries and virtual machine.
AbstractQueuedSynchronizer	java.lang.management	Provides the management interfaces for monitoring and management of the Java virtual machine and other components in the Java runtime.
AbstractRegionPainter.PaintContext	java.lang.ref	Provides reference-object classes, which support a limited degree of interaction with the garbage collector.
AbstractRegionPainter.PaintContext.CacheMode AbstractScriptEngine	java.lang.reflect	Provides classes and interfaces for obtaining reflective information about classes and objects.
AbstractSelectableChannel	java.math	Provides classes for performing arbitrary-precision integer arithmetic (BigInteger) and arbitrary-precision decimal arithmetic (BigDecimal).
AbstractSelector	java.net	Provides the classes for implementing networking applications.
AbstractSequentialList	java.nio	Defines buffers, which are containers for data, and provides an overview of the other NIO packages.
AbstractSpinnerModel	java.nio.channels	Defines channels, which represent connections to entities that are capable of performing I/O operations, such as files and sockets; defines selectors, for multiplexed, non-blocking I/O operations.
AbstractTableModel AbstractTypeVisitor6	java.nio.channels.spi	Service-provider classes for the fava.mio.channels package.
AbetractTuna\fieitor7	*	*

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

Con el tiempo tendremos las clases, según las necesidades que surjan, para ello habrá que actualizar la versión de JRE.

Se puede descargar para tenerla en tu disco duro local, pero si tienen internet siempre la tendrás actualizada.

Clase Math

- La clase Math contiene un muestrario de métodos que nos permitirán realizar cálculos matemáticos.
  - Math.sqrt(n): raíz cuadrada de un número.
  - Math.pow(base, exponente): potencia de un número. Base y exponentes son doubles.
  - o Math.sin(ángulo). Math.cos(ángulo). Math.tan(ángulo). Math.atan(/ángulo)
  - Math.round(decimal): redondeo de un número.
  - o Math.PI: contante de clase con el número PI

Vamos a Eclipse.

Vamos crear una nueva clase llamada Calculos\_conMath

New Java Class		
Java Class 😢 Type name is no	t valid. A Java type name must not start or end with a blank	C
Source fol <u>d</u> er:	PrimerosPasos/src	Br <u>o</u> wse
Pac <u>k</u> age:	(default)	Bro <u>w</u> se
Enclosing type:		Bro <u>w</u> se
Na <u>m</u> e:	Calculos_conMath	
Modifiers:	public Opackage Oprivate Oprotected     abstract final static	
<u>Superclass:</u>	java.lang.Object	Brows <u>e</u>
Interfaces:		<u>A</u> dd
		<u>R</u> emove
Which method stub	os would you like to create?	
►	public static void main(String[] args)	
	Constr <u>u</u> ctors from superclass	
Do you want to add	Inmented abstract methods  comments? (Configure templates and default value <u>here</u> ) <u>G</u> enerate comments	
?	<u>F</u> inish	Cancel

Seleccionamos el botón Finish.

Curso de Java - PrimerosPasos/src/Calculos_conMat	h.java - Eclipse IDE		– 🗆 ×
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> ro	ject <u>R</u> un <u>W</u> indow <u>H</u> elp		
📑 • 🗐 🕼 🔌 🕸 • 🔿 • 🎴 • 💁 •	▾ 🖄 🗀 🛷 ▾ 🕸 🗾 🛯 🗉 👔 😓 ▾ 🎘 ▾ 🏷 🗢 ▾ ⇔ ▾   🛃 🔹		Q 🗄 😭 😽
Package Explorer ☆      PrimerosPasos     Alter System Library [JavaSE-14]     Calculos_conMath.java     J Calculos_conMath.java     J Declaraciones_Opeadores.java     J PrimeraClase.java     J Variables.java	<pre> Calculos_conMath.java  public class Calculos_conMath {     public static void main(String[] args) {         // TODO Auto-generated method stub         //          // TODO Auto-generated method stub         //          //          //</pre>		Image: Task List 12       □         Image: Task List 12       □         Image: Task List 12       Image: Task List 12         Image: Task 12       Image: Task 12         Image: Task 12
	K	>	<pre>8</pre>
	Problems @ Javadoc 🔂 Declaration 📮 Console 🔀 No consoles to display at this time.		
	Writable Smart Insert 1:1:0		9

Ejemplo con la clase Math.

Curso de Java - PrimerosPasos/src/Calculos_conMat	n.java - Eclipse IDE				– 🗆 ×
<u>File Edit Source Refactor Navigate Search Pro</u>	ject <u>R</u> un <u>W</u> indow <u>H</u> e	ilp			
📑 🕶 📰 🕼 🗶 🔅 🕶 💽 🕶 💁 🕶 🖶	6 • 🔊 🗁 🛷 • 🐐	े 📝 🗟 🗉 👖 💱 🗸	· 👬 🕶 🏷 🔶 🕶 🖒	-	Q 🗄 😰 🛛 🕹
🛱 Package Explorer 🔀 🛛 📄 🚍	Calculos_conMath.jav	a 🛙			🗐 Task List 🛛 👘 🗖
<ul> <li>PrimerosPasos</li> <li>Al JRE System Library [JavaSE-14]</li> <li>Primeros commentation of the system of th</li></ul>	1 public class C 3 public class C 4 public sta 5 // TOC 6 7 double 9 System 10 1 12 } 13 14 } 15 <	alculos_conMath { tic void main(String 0 Auto-generated met raiz = Math.sqrt(9. out.println(raiz);	:[] args) { hod stub 0);		Image: Contract of the second sec
	🖹 Problems @ Javado	oc 😥 Declaration 📃 Co	onsole 🛛 🔳 🕽	( 🔆   🗟 🖬 🔛 🖃	🔎 🛃 🖳 🕶 📩 🖛 🗖
	<terminated> Calculos_c</terminated>	onMath [Java Application]	C:\Program Files\Java\j	dk-14.0.2\bin\javaw.exe (	27 ago. 2020 12:36:40 – 12:36:42)
	3.0				^
					~
	<				>
	Writable	Smart Insert	10:9:190		0

Tanto el parámetro que le pasamos como en que retorna es de tipo double.





#### Estructuras principales VII. Clase Math II. (Vídeo 10)

_					
Curso de Java - PrimerosPasos/src/Calcul	os_conMath.java - Eclipse IDE				– 🗆 ×
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate S	e <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> inde	ow <u>H</u> elp			
📑 • 🗐 🕲 🔯 • O • 💁 • O	🖌 🕈 🞯 🕶 🈂 😂 🥖	' 🗝 🕵 📝 🕫 🧃	<u>∲</u>   ▼ <del>  </del>   ▼ (⇔ <	> -   🛃	Q i 🖻 🛛 💐
Package Explorer 🛛 🗖 🚺 Ca	culos_conMath.java 🛛				□ Task List ☆ □ □
> PrimerosPasos	public class Calculos_c	conMath {	1		
2 5	// TODO Auto-ge	enerated method stub		-	8 Find All Activa
/ 8 9	<pre>int resultado =</pre>	Math.round(num1);			
10 11 12	System. <i>out</i> .prim	ntln(resultado);			E Outline X Dutline X Dutl
13 14 15	}				<ul> <li>Calculos_conMath</li> <li><sup>S</sup> main(String[]) : void</li> </ul>
16	3			~	
Pro	blems @ Javadoc 😥 Decl	aration 📃 Console 🔀	■ X ¾	B. 🔝 P	
<termi 6</termi 	nated> Calculos_conMath [Jav	a Application] C:\Progra	n Files\Java\jdk-14.0.2\bin\java	aw.exe (27 ag	jo. 2020 13:03:55 – 13:03:57)
<					>
	Writable	Smart Insert	10 : 13 : 189		1 (

En este ejemplo vamos a utilizar Math.round(float), para ello hemos de pasarle un parámetro de tipo Float y retorna un valor de tipo int.

Para asignar una variable de tipo float hemos de pasar el parámetro F.

float num1 = 5.85F; 🔸

Refundiciones \_\_\_\_\_

int raiz = (int)Math.round(num1);

Cuando la clase Math.round retorna un valor de tipo long y lo queremos asignar a una variable de tipo int.

Curso de Java - PrimerosPasos/src/Calculos co	nMath java - Eclinse IDE				_		×
File Edit Course Defectes National Course	- Designation Designation						~
File Edit Source Refactor Navigate Search	<u>Project Kun windo</u>		. 1 Dec. 17-1 and a			0 -	
	: 🖭 🞯 🔻 : 😂 🗁 🔗	′ *: 🏆 🍠 📴 1	: ₩ <b>* </b>	₽▼⇔▼ ₫		Q : E	3   🕙
🛱 Package Explorer 🛛 🗖 🖸 Calculos	s_conMath.java 🛛				🗐 Task List 🔀		
E 😫 8 1	lie class Calcular a	enMath (		^	🗋 🖌 📴 🕼	🌮   🗙	👫 🖂
> 🔁 PrimerosPasos 3	lic class carculos_c	onnain i					
40	public static void	main(String[] args	{	_	8		
6	77 TODO AUCO-BO	ineraceu mechoù scu	,				
7	<pre>double num1 = 5</pre>	.85;			Find	All F AG	tiva
9	<pre>int resultado =</pre>	(int)Math.round(n	um1);				_
10	System.out.ncir	tln(resultado):			E Outline ⊠		
12					$\Box \downarrow^{a}_{Z} \aleph \aleph$	● × <sup>L</sup>	6 G
13	}				✓ O Calculos	_conMath	
15	,				e main(	sung[]) : ·	/010
16 }							
				× .			
		_					_
Problem	ns @ Javadoc 😣 Decl	aration 🖳 Console 🔀		< 🦓   🖳 🚮 🔛		* 🖸 <b>*</b>	
<terminated< th=""><th>d&gt; Calculos_conMath [Jav</th><th>a Application] C:\Progra</th><th>m Files\Java\jdk-14.0.2\</th><th>bin\javaw.exe (27 ag</th><th>jo. 2020 13:15:51 – 13</th><th>3:15:53)</th><th></th></terminated<>	d> Calculos_conMath [Jav	a Application] C:\Progra	m Files\Java\jdk-14.0.2\	bin\javaw.exe (27 ag	jo. 2020 13:15:51 – 13	3:15:53)	
°							^
<							>
	Writable	Smart Insert	12:5:231				: 🖓

Nos ha retornado un 6.



Con Math.pow(a,b) se calcula del número a elevado a la b, retorna un valor double, con lo cual haremos como en el ejemplo anterior.

```
int resultado=(int)Math.pow(base, exponente);
```

```
Otro ejemplo de refundición.
```

```
Vamos a concatenar el mensaje de consola.
```

```
public class Calculos_conMath {
  2
E
         public static void main(String[] args) {
  40
  5
              // TODO Auto-generated method stub
  6
  7
              double base = 5;
  8
9
              double exponente = 3;
 10
11
              int resultado=(int)Math.pow(base, exponente);
 12
 13
<u>14</u>
15
              System.out.println("El rersultado " + " de " + base + " elevado a " + exponente + " es " + resultado);
                      Т
         }
 16
 17
    }
 18
```

Este será el resultado:

El rersultado de 5.0 elevado a 3.0 es 125

En la biblioteca de API podremos consultar que tipo de valores hay que poner y que tipo de valor retorna.





## Manipulación de cadenas. Clase String I (Vídeo 11)

Clase String

- String no es un tipo primitivo.
- ¿Cómo almacenar una cadena de caracteres?
  - String mi\_nombre="Juan"; donde mi\_nombre es un objeto (instancia, ejemplar) de la clase String
- Métodos (más usado) de la clase String para manipulación de cadena de textos:
  - Lenght(): devuelve la longitud de una cadena de caracteres.
  - CharAt(n): devuelve la posición de un carácter dentro de una cadena. (Las posiciones empiezan a contar de 0)
  - Substring(x,y): devuelve una subcadena dentro de la cadena, siendo x el carácter a partir del cuál se extrae e y e nº de caracteres que se quieren extraer.
  - Equals(cadena): devuelve true si dos cadenas que se comparan son iguales y false si no lo son. Distingue mayúsculas y minúsculas.
  - equalsIgnoreCase(cadena): igual que el anterior pero sin tener en cuenta mayúsculas y mminúsculas.

Vamos a crear una nueva clase llamada manipula\_cadena.

New Java Class	_		×
Java Class	efault package is discouraged.	C	
Source fol <u>d</u> er:	PrimerosPasos/src	Br <u>o</u> wse	
Pac <u>k</u> age:	(default)	Bro <u>w</u> se	
Enclosing type:		Bro <u>w</u> se	
Na <u>m</u> e: Modifiers:	manipula_cadena <ul> <li> <u>public</u> <u>o package</u> <u>private</u> <u>o protected</u> </li> </ul> <u>abstract</u>		
<u>S</u> uperclass:	java.lang.Object	Brows <u>e</u>	
<u>I</u> nterfaces:		<u>A</u> dd <u>R</u> emov	e
Which method stul	bs would you like to create? public static <u>void main(String[] args)</u> Constr <u>u</u> ctors from superclass In <u>h</u> erited abstract methods comments? (Configure templates and default value <u>here</u> ) <u>G</u> enerate comments		
?	<u> </u>	Cance	I



En el objeto nombre almacenamos la cadena "Juan" que concatenamos para imprimirlo por la consola.

```
1
 2
    public class manipula_cadena {
 3
 4⊝
        public static void main(String[] args) {
 5
            // TODO Auto-generated method stub
 6
 7
            String nombre="Juan";
 8
 9
            System.out.println("Mi nombre es " + nombre);
10
            System.out.println("Mi nombre tine " + nombre.length()+ " letras.");
11
12
13
14
        }
15
16 }
17
```

Este será el resultado:

Mi nombre es Juan Mi nombre tine 4 letras.

```
1
 2 public class manipula_cadena {
 3
4⊝
5
        public static void main(String[] args) {
            // TODO Auto-generated method stub
 6
 7
            String nombre="Juan";
 8
 9
            System.out.println("Mi nombre es " + nombre);
 10
 11
            System.out.println("Mi nombre tine " + nombre.length()+ " letras.");
12
13
            System.out.println("Mi nombre empieza por " + nombre.charAt(0));
14
15
16
        }
17
18 }
19
```

Este será el resultado:

```
Mi nombre es Juan
Mi nombre tine 4 letras.
Mi nombre empieza por J
1
  2 public class manipula_cadena {
  З
4⊝
        public static void main(String[] args) {
E
 5
            // TODO Auto-generated method stub
  6
  7
            String nombre="Juan";
  8
  9
            System.out.println("Mi nombre es " + nombre);
 10
            System.out.println("Mi nombre tine " + nombre.length()+ " letras.");
 11
 12
 13
            System.out.println("Mi nombre empieza por " + nombre.charAt(0));
 14
 15
            System.out.println("Mi nombre termia por " + nombre.charAt(nombre.length()-1));
 16
 17
 18
        }
 19
 20 }
```

Este será el resultado:

Mi nombre es Juan Mi nombre tine 4 letras. Mi nombre empieza por J Mi nombre termia por n





# Manipulación de cadenas. Clase String II (Vídeo 12)

New Java Class			×
Java Class	fault package is discouraged.	C	
Source fol <u>d</u> er:	PrimerosPasos/src	Br <u>o</u> ws	e
Pac <u>k</u> age:	(default)	Bro <u>w</u> s	e
Enclosing type:		Bro <u>w</u> s	e
Na <u>m</u> e: Modifiers:	manipula_Cadena_II             gublic		
<u>S</u> uperclass:	java.lang.Object	Brows	<u>e</u>
Interfaces:		<u>A</u> dd. <u>R</u> emo	 ve
Which method stub	os would you like to create? ✓ public static void main(String[] args) Constructors from superclass ✓ Inherited abstract methods I comments? (Configure templates and default value here) Generate comments		
?	<u> </u>	Canc	el

Creamos un nuevo proyecto llamado manipula\_Cadena\_II.

Queremos extraer de una cadena parte de ella.

	1 2	public class manipula_Cadena_II {}
2	5 4⊜ 5	<pre>public static void main(String[] args) {     // TODO Auto-generated method stub</pre>
	6 7 8 9	<pre>String frase ="Hoy es un estupendo día para aprender a programar en Java"; String frase_resumen =frase.substring(29, 57); System out_println(frase_resumen);</pre>
	10 11 12	}
	13 14	K

Este será el resultado:

aprender a progamar en Java

Ahora en el siguiente ejemplo hacemos además una concatenación.

```
1
2 public class manipula Cadena II 🖁
3
40
       public static void main(String[] args) {
5
           // TODO Auto-generated method stub
 6
7
           String frase ="Hoy es un estupendo día para aprender a programar en Java";
           String frase_resumen =frase.substring(0, 28) + " irnos a la playa";
8
9
            System.out.println(frase resumen);
10
11
        }
12
   }
13
14
```

Este será el resultado:

Hoy es un estupendo día para irnos a la playa

Para hacer un salto de línea solo hay que hacer intro.

```
1
     public class manipula Cadena II 🖁
  2
  З
         public static void main(String[] args) {
  4⊝
Æ
 5
            // TODO Auto-generated method stub
  6
  7
             String frase ="Hoy es un estupendo día para aprender a programar en Java";
             String frase_resumen =frase.substring(0, 28) + " irnos a la playa...
  8
             " y " + frase.substring(29, 57);
  9
 10
             System.out.println(frase_resumen);
 11
 12
         }
 13
 14
     }
15
```

Este será el resultado:

Hoy es un estupendo día para irnos a la playa... y aprender a programar en Java

Vamos a crear otra clase llamada manipula\_cadena\_III.

```
1
    public class manipula_cadena_III {
  2
  3
  4Θ
         public static void main(String[] args) {
12
             // TODO Auto-generated method stub
  5
  6
  7
             String alumno1, alumno2;
  8
  9
             alumno1="David";
 10
 11
             alumno2="David";
 12
 13
             System.out.println(alumno1.equals(alumno2));
 14
 15
         }
 16
 17
     }
 18
```

En este ejemplo queremos comparar si las variables alumos1 y alumno2 contienen la misma información, si es afirmativo retornará true y sino false.

Este será el resultado:

True

Si queremos que no distinga entre mayúsculas y minúsculas, este será el ejemplo:



Este será el resultado:

true





### Acercamiento a la API Paquetes (Vídeo 13)

Las API de Java son las bibliotecas de clase que vienen de forma predefinida, para que las podamos utilizar en nuestros programas.

En Java tenemos las clases predefinidas y las propias.

Las clases de Java a las carpetas se les denomina paquetes.

🖸 (170) Curso Java. Acercamiento a 🗴 🥃 Overview	v (Java Platform SE 7.) × +	- 0
← → C	ı/api/	a 🕁 🙆 🏚 🗐 🧖
Aplicaciones		
Java™ Platform Standard Ed. 7	Overview Package Class Use Tree Depr	Jana <sup>3</sup> Pusitiva Sender El D
All Classes	Prev Next Frames No Frames	
Packages	Java IN Blatform, Standar	A Edition 7
java.applet java.avvt java.avvt.color	API Specification	
java.avvt.datatransfer java.avvt.dat	This document is the API specification for the Jar	va™ Flatform, Standard Edition.
java.awt.event java.awt.font java.awt.geom	See: Description	
java.avt.im	Packages	
java.awt.image	Package	Description
	java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
All Classes	java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
AbstractAction	java.awt.color	Provides classes for color spaces.
AbstractAnnotationValueVisitor6	java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
AbstractBorder AbstractButton	java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
AbstractCellEditor AbstractCollection	java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
AbstractColorChooserPanel	java.awt.font	Provides classes and interface relating to fonts.
AbstractDocument AttributeContext	java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
AbstractDocument Content AbstractDocument ElementEdit	java.awt.im	Provides classes and interfaces for the input method framework.
AbstractElementVisitor6	java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
AbstractExecutorService	java.awt.image	Provides classes for creating and modifying images.
AbstractinterruptibleChannel Abstracti avoutCache	java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
AbstractLayoutCache.NodeDimensions	java.awt.print	Provides classes and interfaces for a general printing API.
AbstractListModel	java.beans	Contains classes related to developing beans components based on the JavaBeans <sup>194</sup> architecture.
AbstractMap AbstractMap SimpleEntry	java.beans.beancontext	Provides classes and interfaces relating to bean context.
AbstractMap.SimpleImmutableEntry	java.io	Provides for system input and output through data streams, serialization and the file system.
AbstractMethodError	java.lang	Provides classes that are fundamental to the design of the Java programming language.
AbstractOwnableSynchronizer AbstractOrefarences	java.lang.annotation	Provides library support for the Java programming language annotation facility.
AbstractProcessor	java.lang.instrument	Provides services that allow Java programming language agents to instrument programs running on the JVM.
AbstractQueue AbstractQueuedLonoSynchronizer	java.lang.invoke	The java.lang.invoke package contains dynamic language support provided directly by the Java core class libraries and virtual machine.
AbstractQueuedSynchronizer	java.lang.management	Provides the management interfaces for monitoring and management of the Java virtual machine and other components in the Java runtime.
AbstractRegionPainter.PaintContext	java.lang.ref	Provides reference-object classes, which support a limited degree of interaction with the garbage collector.
AbstractRegionPainter.PaintContext.CacheMode AbstractScriptEngine	java.lang.reflect	Provides classes and interfaces for obtaining reflective information about classes and objects.
AbstractSelectableChannel	java.math	Provides classes for performing arbitrary-precision integer arithmetic (BigInteger) and arbitrary-precision decimal arithmetic (BigDecimal).
AbstractSelector	java.net	Provides the classes for implementing networking applications.
AbstractSequentialList	java.nio	Defines buffers, which are containers for data, and provides an overview of the other NIO packages.
AbstractSpinnerModel	java.nio.channels	Defines channels, which represent connections to entities that are capable of performing I/O operations, such as files and sockets; defines selectors, for multiplexed, non-blocking I/O operations.
AbstractTableModel AbstractTypeVisitor6	java.nio.channels.spi	Service-provider classes for the java.nio.channels package.
AbstractTypeVisitor7	*	

La parte superior izquierda se denomina panel de paquetes.

API Java (primer acercamiento)						
		java	paquetes	ja	vax	
	java.awt	java.util Java.util.regex	java.io	javax.activity	javax.annotation	WTEAL

- ¿Por qué son necesarios los paquetes? ¿Por qué se inventaron?
  - Para organizar las clases
  - Para evitar conflictos de nombres
  - o Para controlar la visibilidad de las clases

API Java (primer ace	rcamiento)
Java.awt Java.util Java.lang	Java.io Java.net Math Thread
substring() equals() round(	Sqrt()

Java.lang: Es el paquete principal que a su vez por este motivos hemos tenido acceso a las clases String, System, Math, Thread, así como a sus métodos sin ningún problema.

Si quieres utilizar otro paquete, por ejemplo Java.util que no se encuentre en Java.lang tendrás que indicárselo previamente.

Vamos a ejecutar Eclipse.

🖨 Curso de Java - Eclipse IDE			– 🗆 X
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> r	rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp		
🔁 • 🖓 • 🚱 🖉 🖉	▼ 🖶 🎯 ▼ 🤔 😂 🛷 ▼ 🗄 ▼ 🖓 ▼ 🏷 🔶 ▼ 🔿 ▼   🛃		Q 🗄 😰   🐉
😫 Package Explorer 💥 📃 🗖			🗐 Task List 🔀 🗖 🗖
			🗂 🕇 📲 📽   🐌   🗙 👫 🖻   🗐 🕴
PrimerosPasos     BE System Library [JavaSE-14]			Find I All Activate (?)
✓ → src			
🗸 🌐 (default package)			
> 🕖 Calculos_conMath.java			
> Declaraciones_Opeadores.java			
> J manipula_Cadena_II.java			
> I) manipula cadena.iava			
> D PrimeraClase.java			E Outline 🛛 🗊 🕴 🗖 🗖
> 🚺 Variables.java			There is no active editor that provides an outline.
	🖹 Problems @ Javadoc 😟 Declaration 📮 Console 🔀		
	No consoles to display at this time.		
			: 0
		1.0	: 🗸

Lo que señalamos con una flecha en un paquete.

Cuando tengamos que utilizar una clase que no pertenece al paquete principal, habrá que importarla.



Al agregar un \* al final, le estamos diciendo que importe todas las clases del paquete java.util.

Si del paquete java.util solo utilizarás la clase Scanner



Si utilizar el \* consumes más recursos porque le estás diciendo que te importe todas las clases en la memoria que si le dices que te importe solo ana clase.



Al utilizar el \* podemos utilizar todas las clases del paquete java.util.



Importamos las clases Scanner y Locate del paquete java.util, es mucho más practico importar con el \* y ahorramos código.

Supongamos que estamos trabajando con dos clases y no sabemos a que paquete pertenece.



Observamos dos errores porque no las reconoce, pero además no sabemos a que paquete pertenece, para ello vamos a realizar los siguientes pasos:

Curso e	de Java - PrimerosPasos/src/Prueba.java -	Eclipse IDE	- 🗆 X
File Edit	Source Refactor Navigate Search	Project Run Window	Help
<ul> <li>➡ □</li> <li>➡ □</li> <li>➡ □</li> <li>➡ □</li> <li>□</li> <li>➡ □</li> <li>□</li>     &lt;</ul>	Toggle Comment Add Block Comment Remove Block Comment Generate Element Comment Add Text Block Shift Right Shift Left Correct Indentation Format	Ctrl+7 Ctrl+Shift+/ Ctrl+Shift+\ Alt+Shift+J Ctrl+Shift+' Ctrl+Shift+F Ctrl+Shift+F	Image: Second state of the second
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Format Element Add Import Organize Imports Sort Members Clean Up Override/Implement Methods Generate Getters and Setters Generate Delegate Methods Generate Delegate Methods	Ctrl+Shift+M Ctrl+Shift+O	Brueba ● S main(String[]): void
	Generate nashCobey and equalsy Generate toString() Generate Constructor using Fields Generate Constructors from Superc Surround With Externalize Strings Find Broken Externalized Strings	lass Alt+Shift+Z >	Declaration 📮 Console 🖾 🖻 👻 🖻 🖻 🖛 🖻 🖛
		Writable	Smart Insert 1:1:0

Del menú Source seleccionaremos Organize Imports.

Organize Imports	_	
<u>C</u> hoose type to import:		Page 1 of 1
<ul> <li>G<sup>F</sup> java.util.Scanner</li> <li>com.sun.java_cup.internal.runtime.Scanner</li> <li>com.sun.tools.javac.parser.Scanner</li> </ul>		
<ul> <li>jdk.nashorn.internal.parser.Scanner</li> <li>sun.tools.java.Scanner</li> </ul>		
Skip     < Back     Next >     Fi	nish	Cancel

Seguido del botón Finish.



Ya hemos importado las clases Locale y Scanner del paquete java.util.





### Entrada Salida datos I (Vídeo 14)



Un método es estático cuando delante del método hay que poner la clase separada por punto.

Los métodos que no son estáticos no vamos a utilizar el nombre de la clase, para ello hemos de crear un objeto de la clase Scanner.

Vamos a Eclipse.

Vamos a crear una nueva clase llamada Entrada\_ejemplo1.

	🖨 New Java Clas	s					×		
	Java Class					e			
	👍 The use of the	default package is c	liscouraged.			C			
	Source fol <u>d</u> er:	PrimerosPasos/s	src			Br <u>o</u> ws	e		
	Pac <u>k</u> age:			(d	efault)	Bro <u>w</u> s	e		
	Enclosing type	:		,		Brows	e		
	Name	Entrada ejemplo	1						
	Modifiers:		nackage Opri	vate Oprot	ected				
	mounters	abs <u>t</u> ract fi	na <u>l</u> stati <u>c</u>						
	<u>Superclass:</u>	java.lang.Object	:			Brows	<u>e</u>		
	Interfaces:					<u>A</u> dd.			
						<u>R</u> emo	ve		
	Which method st	ubs would you like t	o create?						
		✓ public static v	oid main(String[]	args)					
		Constr <u>u</u> ctors f	rom superclass						
	Do you want to av	In <u>h</u> erited absti	figure templates :	and default value	here)				
	Do you want to at	Generate com	ments	and default value	nere)				
	?			<u>F</u> inish		Cance	el		
🛢 Curso de Java - Pri	merosPasos/src/Entrada_ej	emplo1.java - Eclipse IDE						- 0	×
<u>File Edit Source</u>	Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rc	h <u>P</u> roject <u>R</u> un <u>W</u> indo	w <u>H</u> elp	t Day tail way of	1N	1		0	
: □2 ▼ III VIII : ♥ :	ו••••	D Entrada eiemplo1.java	* * * <u>&gt;</u> * • • •			Task List Σ	3	Q	
	E 😫 🖇	1   2 public class En	trada ejemplo1 /		^	ें 🕶   🐄	8= 5	× 👫 🗆	<b>100</b> 100
✓  → PrimerosPasos →  ▲ JRE System L	ibrary [JavaSE-14]	3 4⊖ public stat	ic void main(String	z[] args) {		Find	► All	Activate	e ?
✓ 进 src ✓ 🌐 (default p	ackage)	2 5 // TODO	Auto-generated met	thod stub	-				
> 🚺 Calculo	os_conMath.java aciones Opeadores.java	7 }							
> 🚺 Entrada	a_ejemplo1.java	9 } 10							
> D manip	ula_cadena_III.java					😓 Outline 🖾			
> 🗾 manipi > 🚺 Primer	ula_cadena.java aClase.java					<u> </u>	î⇒ ⊫ ļa	R x <sup>5</sup>	• <b>∖</b> 8
> J Variabl	es.Java					✓ ☺ <sub>▶</sub> Entra <sup>S</sup> m	da_ejemplo1 ain(String[])	: void	
		<			>				
		Problems @ Javadoc	😣 Declaration 📮 Co	onsole 🛛			ľ	🗉 🔻 📑	•
		No consoles to display at th	iis time.						
		Writable	Smart Insert	1:1:0	1				: 🖓

Constructor: Tiene el mismo nombre que a la clase que pertenece.

Vamos a escribir el siguiente código:

```
1
    import java.util.*;
    public class Entrada ejemplo1 🛛
 2
 З
 48
        public static void main(String[] args) {
 5
            // TODO Auto-generated method stub
            Scanner entrada=new Scanner(System.in);
 6
h
            System.out.println("Introduce tu nombre, por favor: ");
 7
            String nombre_usuario = entrada.nextLine();
 8
 9
            System.out.println("Introduce edad, por favor: ");
            int edad = entrada.nextInt();
10
            System.out.println("Hola " + nombre_usuario + " el año que viene tendrás " +
11
            (edad + 1) + " años.");
12
13
        }
14
15
    }
```

En la línea 1 importamos el paquete java.util, con el \* todas sus clases.

En la línea 6 definimos un objeto llamado entrada de tipo Scanner, que nos permite realizar entradas por la consola.

En la línea 8 definimos una variable de tipo String asignándole el objeto entrada que es de la clase Scanner para que la consola se detenga esperando que introduzcamos un datos de tipo string.

En la línea 10 definimos una variable de tipo int asignándole el objeto entrada para que la consola se detenga en espera que introduzcas un valor de tipo int.

Si ejecutamos este será el resultado:

```
Introduce tu nombre, por favor:
Pedro
Introduce edad, por favor:
60
Hola Pedro el año que viene tendrás 61 años.
```

Observarás que durante la ejecución del programa observamos el siguiente cuadrado en rojo.



Nos indica que el programa está en ejecución, cuando termine se volverá de color gris.





#### Entrada Salida datos II. (Vídeo 15)

Vamos a ejecutar Eclipse y vamos a crear otra clase llama Entrada\_ejemplo2.

```
1 import javax.swing.*;
2
   public class Entrada_ejemplo2 {
 З
40
       public static void main(String[] args) {
5
           // TODO Auto-generated method stub
           String Nombre_usuario=JOptionPane.showInputDialog("Introduce tu nombre, por favor");
6
           System.out.println("Tu nombre es " + Nombre_usuario + ".");
7
8
       }
9
10 }
```

En la línea 1 importamos todas las clases del paquete javax.swing.

En la línea 6 declaramos una variable de tipo String llamada Nombre\_usuario que le asignaremos el dato introducido en la ventana de diálogo que se abrirá, que nos preguntará "Introduce tu nombre, por favor".

En la línea 7 imprimiremos por consola tu nombre es + el nombre introducido en la ventana de diálogo.

Input		×
?	Introduce tu nombre, por favor         Pere Manel         OK       Cancel	

Le damos al botón Ok, y este será el resultado:

Tu nombre es Pere Manel.

Ahora además que nos pregunte por la edad.

```
1 import javax.swing.*;
2
   public class Entrada_ejemplo2 {
3
       public static void main(String[] args) {
40
5
           // TODO Auto-generated method stub
           String Nombre usuario=JOptionPane.showInputDialog("Introduce tu nombre, por favor");
6
         String edad = DOptionPane.showInputDialog("Introduce tu edad, por favor");
7
8
           System.out.println("Tu nombre es " + Nombre_usuario + " y tienes " +
9
10
           edad + " años");
11
       }
12
13
   }
14
```

Creamos variable de tipo String que retorna un valor de este tipo.

```
1 import javax.swing.JOptionPane;
 2 public class Entrada_ejemplo2 {
 З
J.
 4⊝
        public static void main(String[] args) {
 5
            // TODO Auto-generated method stub
            String Nombre_usuario=JOptionPane.showInputDialog("Introduce tu nombre, por favor");
 6
 7
            String edad = JOptionPane.showInputDialog("Introduce tu edad, por favor");
 8
            int edad1 = Integer.parseInt(edad);
 9
 10
            System.out.println("Tu nombre es " + Nombre_usuario + " el año que viene tendrás | " +
            (edad1 + 1) + " años");
11
        }
12
13
14 }
15
```

En la línea 8 hacemos que la variable edad1 asuma el valor como un dato int, ya que en su momento en que introdujéramos una edad esta era almacenada de tipo String.

El método parseInt y observamos que es estático tendré que utilizar el nombre de la clase como argumento: Integer.parseInt(edad);

El resultado será:

```
Tu nombre es Pere Manel el año que viene tendrás 61 años
```

Vamos a utilizar el operador incremento.

```
1 import javax.swing.JOptionPane;
     public class Entrada_ejemplo2 {
  2
  З
  40
          public static void main(String[] args) {
e
  5
              // TODO Auto-generated method stub
              String Nombre_usuario=JOptionPane.showInputDialog("Introduce tu nombre, por favor");
String edad = JOptionPane.showInputDialog("Introduce tu edad, por favor");
  6
  7
  8
              int edad1 = Integer.parseInt(edad);
  9
 10
               edad1++; <
 11
 12
               System.out.println("Tu nombre es " + Nombre_usuario + " el año que viene tendrás " +
 13
               edad1 + " años");
 14
          }
15
16 }
```

Lo que hace es incrementar su valor en 1.

¿Cómo formatear los resultados?

```
1
 2
    public class Entrada Numeros {
 3
        public static void main(String[] args) {
 4⊝
 5
            // TODO Auto-generated method stub
 6
 7
            double x=10000.0;
 8
 9
            System.out.printf("%1.2f", x/3);
10
        }
11
12
    }
```

En la línea 9 hacemos un printf de format y en comillas le decimos con el "%1.2f", que al número que viene a continuación le asignemos un formato de 2 decimales, este será el resultado:

3333,33

```
1 import javax.swing.*;
 2
    public class Entrada_Numeros {
 3
        public static void main(String[] args) {
 4⊖
 5
            // TODO Auto-generated method stub
 6
            String num1=JOptionPane.showInputDialog("Introduce un número: ");
 7
 8
            Double num2=Double.parseDouble(num1);
 9
 10
            System.out.print("La raiz del número " + num1 + " es ");
 11
12
13
            System.out.printf("%1.2f", Math.sqrt(num2));
14
15
        }
16
17 }
```

En el siguiente ejemplo en la línea 9 hacemos que un valor de tipo string pase a ser de tipo Double.

En la línea 11 hacemos un print este no hace salto de línea, con lo cual el siguiente print lo hará en la misma línea.

En la línea 13 hacemos un printf para darle formato de 2 decimales.

Si introducimos el valor 25 este será el resultado:

Input		$\times$
?	Introduce un número: 25 OK Cancel	

La raiz del número 25 es 5,00





## Condicionales I. (Vídeo 16)

Control de flujo.

Condicionales y bucles.

Flujo de ejecución	
<ul> <li>Public static void main (String args[])</li> </ul>	
• Línea código 1	
• Línea código 2	
• Línea código 3	
• Línea código 4	
• Línea código 5	
•}	PLOGRASINFORMÁTICAS

Para el salto usamos el condicional y para repetir unas líneas de código x veces utilizamos el bucle.

Condicionales

•

- Condicional if
  - If(condición){
    - Código a ejecutar si la condición es verdadera (true);

0 }

- Condicional switch
  - Switch (valor a evaluar){
    - Case valor1:
      - Código a ejecutar;
      - Break;
    - Case valor 2:
      - Código a ejecutar;
      - Break;

El condicional if puede llevar otra estructura que es else.

El condicional switch puede llevar otra estructura que es default.

Vamos a ejecutar Eclipse con una nueva clase llamada Evalua\_edad.

```
import java.util.*;
 1
 2
    public class Evalua_edad {
 3
 4⊝
        public static void main(String[] args) {
 5
            // TODO Auto-generated method stub
            Scanner entrada=new Scanner(System.in);
 6
 7
            System.out.println("Introduce tu edad, por favor");
 8
            int edad=entrada.nextInt();
 9
10
            if(edad>=18) {
11
                System.out.println("Eres mayor de edad");
12
            }
13
            else {
14
                System.out.println("No eres mayor de edad");
15
            }
16
17
        }
18
19
    3
```

En la línea 10 estamos comparando si la variable edad es mayor o igual a 18, si es así ejecutará el código que se encuentra entre llaves después del if, de lo contrario si no eres mayor o igual a 18 se ejecutará lo que viene después del else que se encuentra entre llaves.

Vamos a ver ahora la instrucción else if.

```
1 import java.util.*;
    public class Evalua_edad {
 2
 3
 4⊝
        public static void main(String[] args) {
 5
            // TODO Auto-generated method stub
 6
            Scanner entrada=new Scanner(System.in);
 7
            System.out.println("Introduce tu edad, por favor");
 8
            int edad=entrada.nextInt();
 9
10
             if(edad<18) {</pre>
                 System.out.println("Eres un adolescente");
11
12
             }
13
             else if(edad<40){</pre>
                 System.out.println("Eres joven");
14
15
             }
16
             else if(edad<65){</pre>
17
                 System.out.println("Eres maduro");
18
             }
19
             else{
20
                 System.out.println("Cuidate");
21
             }
22
        }
    }
23
```

En la línea 10 compara si edad es menor de 18 si es así ejecuta lo que se encuentra entre llaves, si no es así compara si edad es menor de 40 si es así ejecuta lo siguiente que encuentra entre llaves, si no es así compara si edad es menor de 65 ejecuta el código siguiente pero else (si no se cumple nada de lo anterior, ejecuta la siguiente línea.

Ejecuta el programa con diferentes edades.





#### Condicionales II (Vídeo 17)

Vamos a aprender como funciona la estructura Switch.

Abrimos Eclipse.

Vamos a crear un nuevo proyecto para que nos calcule el área de un cuadrado, rectángulo, triangulo y círculo.

Vamos a crear una clase nueva llamada Áreas.

```
1⊖ import java.util.*;
 2 import javax.swing.*;
 4 public class Areas {
 60
         public static void main(String[] args) {
              // TODO Auto-generated method stub
 7
              Scanner entrada=new Scanner(System.in);
System.out.println("Elige una opción: \n1: Cuadrado "
+ "\n2: Rectángulo \n3: Triangulo \n4: Círculo");
 8
 9
10
11
              int figura=entrada.nextInt();
12
13
              switch(figura) {
14
              case 1:
                   int lado=Integer.parseInt(JOptionPane.showInputDialog("Introduce el lado:"));
15
16
                   System.out.println("El área del cuadrao es "
                                                                           + Math.pow(lado, 2));
17
                   break;
18
              case 2:
19
                   int base=Integer.parseInt(JOptionPane.showInputDialog("Introduce la base del rectángulo:"));
                   int altura=Integer.parseInt(30ptionPane.showInputDialog("Introduce la altura del rectángulo:"));
System.out.println("El área del rectángulo es " + base * altura);
20
21
22
                   break;
23
              case 3:
24
                   base=Integer.parseInt(JOptionPane.showInputDialog("Introduce la base del rectángulo:"));
                   altura=Integer.parseInt(JOptionPane.showInputDialog("Introduce la altura del rectángulo:"));
System.out.println("El área del triangulo es " + (base * altura)/2);
25
26
27
                   break;
28
              case 4:
29
                   double radio=Double.parseDouble(JOptionPane.showInputDialog("Introduce el radio de un circulo:"));
                   System.out.print("El area del circulo es ");
System.out.printf("%1.2f",(Math.PI * Math.pow(radio, 2)));
30
31
32
                   break:
33
              default:
                   System.out.println("El número introducido es erróneo");
34
```

Ahora lo ejecutas y pruebas todas las opciones de cálculo.





#### Bucles I (Vídeo 18)

- Bucles indeterminados
  - o While
  - Do while
- Bucles determinados
  - o For
  - o For each

Bucle while

- Sintaxis:
  - While (condición){
    - Línea1
    - Línea2
    - Línea3
    - Línea4

0 }

Vamos a realizar un proyecto en el que va a pedir una contraseña al usuario y mientras no la acierte la irá preguntado.

Vamos a crear una clase nueva llamada Acceso\_aplicacion.

```
1 import javax.swing.*;
   public class Acceso aplicacion {
 2
 З
 4Θ
        public static void main(String[] args) {
            // TODO Auto-generated method stub
 5
            String clave="Juan";
String pass="";
 6
 7
 8
            while(clave.equals(pass)==false) {
 9
                pass=JOptionPane.showInputDialog("Introduce la contaseña, por favor:");
                if(clave.equals(pass)==false) {
10
11
                     System.out.println("Contraseña incorecta.");
12
                 }
13
14
15
            System.out.println("Contraseña correcta, acceso permitido.");
16
        }
17
18
   }
```

En la línea 8 mientras clave sea distinto a pass el bucle que llega hasta la línea 13 se irá repitiendo.

En la línea 9 hacemos que nos pregunte con una contraseña en una ventana emergente, si introducimos una contraseña distinta a Juan en la línea 11 con una condición nos dirá que la contraseña es incorrecta y no saldremos del bucle, cundo introduzcamos la contraseña correcta imprimirá "Contraseña correcta, acceso permitido" y saldremos del bucle.

Vamos a ejecutar el programa:
Input		$\times$
?	Introduce la contaseña, por favor: Pedro OK Cancel	

Pulsamos OK.

Nos dirá Clave incorrecta.

Input		$\times$
?	Introduce la contaseña, por favor: Juan OK Cancel	

Pulsamos OK.

Nos dirá Contraseña correcta, acceso permitido.

Y terminará el programa.





### Bucles II. (Vídeo 19)

En este ejemplo vamos elaborar una especie de juego en que el programa generará un numero aleatorio entre 0 y 100 y en la que tendremos que adivinar el número, mientras no lo acertemos nos lo irá preguntando.

Vamos a crear una nueva clase llamada Adivina\_numero.

En este programa vamos a introducir la consola tanto para introducir los números como para ver el resultado.

```
1 import java.util.*;
 2
   public class Adivina_numero {
 3
        public static void main(String[] args) {
 4⊝
            // TODO Auto-generated method stub
 5
            int aleatorio = (int)(Math.random()*100);
 6
 7
            Scanner entrada=new Scanner(System.in);
 8
            int numero=0;
 9
            int intentos=0;
10
            while(numero!=aleatorio) {
11
                System.out.println("Introduce un número, por favor:");
12
                intentos++;
13
                numero=entrada.nextInt();
14
                if(aleatorio<numero) {</pre>
15
                    System.out.println("Más bajo");
16
                3
17
                else if (aleatorio>numero) {
18
                    System.out.println("Más alto");
19
                3
20
21
            System.out.println("Correcto has acertado al intento " + intentos + ".");
22
        3
23
   }
24
```

En la línea 6 en la variable aleatorio almacenamos un valor entero aleatorio entre 0 y 100.

En la línea 7 creamos un objeto de tipo Scanner para decirle que los datos los introduciremos por la consola.

Declaramos las variables numero e intentos de tipo int y las inicializamos a 0.

Línea 10 Mientras las variables numero y aleatorio sean distintas ¡=

Línea 11 Pregunta por un número.

Línea 12 intentos++ es un contador que incrementará en 1 cada vez que introduzcamos un número.

Línea 13 el valor introducido por teclado se almacena en la variable número.

Línea 14 comparamos si el valor que hemos introducido es mayor al valor de la variable aleatorio, si es así dirá "Más bajo".

Línea 17 comparamos si el valor que hemos introducido es menor al valor de la variable aleatorio, si es así dirá "Más alto".

Este bucle se repetirá hasta que las variables numero y aleatorio sean iguales, como la condición mientras deja de cumplirse vamos directamente a la línea 21 donde nos dice "Correcto has

acertado al intento x, donde x es el valor de intentos variable que hemos utilizado como contador.





### Bucles III. (Vídeo 20)

Bucle Do – while

- Sintaxis:
  - **Do** {
    - Línea1
    - Línea2
    - Línea3
    - Linea4
    - o }while(condición);

El bucle do-while se asegura que aun siendo la condición falsa, por lo menos se ejecuta una vez.



En este ejemplo garantizamos que al menos una vez entre en el bucle do-while ya que la comparación la realiza al final. Si lo ejecutamos este será el resultado:

<pre>PrimercyPass  PrimercyPass  PrimercyPass Primerc</pre>	Edit Source Refactor Nav	igate Search Project Run Window Help	
<pre>iProtect 20</pre>		입 = Q = : # @ = @ @ # # # # # @ @ # # @ = 한 = 한 = 다 = [ []	Q 🛛 🔡
<pre>PrimercoPaoos  i i igent java.utll.; public tatai: vidi main(String[] arg) (</pre>	ackage Explorer 🖂	🖻 😫 🕴 🗖 🔲 💭 Adivina_numero.java 📖	Task List 🖸
Problems @ Javadoc Q Declaration C Console 32      externinated Advins, numero [Java Application] C Console 32      externinated Advins, numero [Java Application] C Chrogram Filed Java/dk-14.0.2 (bin/javawcze (29 ago. 2020 921:15 - 921:34)      Introduce un numero, por favor:     action of the second of	∯ PitmerosPasos	<pre>i import jows util.*; public static void main(String[] args) {     // TOO Auto-generated method stud     // TOO Auto-generated method stud     // TOO Auto-generated method stud     int alactorio &lt; (inf)(Math. mode()/100);     int alactorio &lt; (inf)(Math. mode()/100);     int intertoo-=0;     do {         system.out.println("Introduce un nümero, por favor:");         int intertoo-=0;         do {             system.out.println("Introduce un nümero, por favor:");             int intertoo++;             int intertoo++;             if(alactoriconuero) {                  System.out.println("Mas late");             }             system.out.println("Mas late");             }             builta(mumerol=alactorio);             System.out.println("Correcto has acertado al intento " + intentos + ".");             }             // **********</pre>	Image: Contine to the second secon
<pre>steminated: Advins_numero[Java Application] C/Program Files/Java/jdk-14.0.2/bin/javaw.see (20 ago. 2020 9:21:15 - 9:21:34) Introduce un numero, por favor: Mis bajo Introduce un numero, por favor: 0 His bajo 0 His ba</pre>		🖹 Problems 🕜 Javadoc 🔯 Declaration 📮 Console 🙁	= 🗶 💥 🗟 🖬 🖉 🖉 🚽 🗆 v 😁 v
		<pre>cterminated&gt; Advinu, numere (Java Application) CLYProgram Files/Java)gld:-14.0.2\binjavau.exe (29 ago. 2020 9:21:15 - 9:21: 16 16 alto 16 for alto 16 for advo: 16 for advo: 17 for advo: 16 for advo: 16 for advo: 17 for advo: 16 for advo: 16 for advo: 17 for advo: 16 for advo: 17 for advo: 16 for advo: 17 for advo: 17 for advo: 18 for advo:</pre>	39

```
Introduce un número, por favor:

50

Más alto

Introduce un número, por favor:

80

Más bajo

Introduce un número, por favor:

70

Más bajo

Introduce un número, por favor:

60

Más alto

Introduce un número, por favor:

65

Correcto has acertado al intento 5.
```

Vamos a realizar un programa para que calcule el peso ideal de un hombre y de una mujer, para el peso del hombre se mide la altura en cm y se resta 110 y para una mujer se mide la altura en centímetros y se resta 120.

Vamos a crear una clase nueva con el nombre peso\_ideal.

```
1 import javax.swing.*;
 2
   public class peso_ideal {
 3
        public static void main(String[] args) {
 4⊝
 5
            // TODO Auto-generated method stub
 6
            String genero="";
 7
            do {
 8
                genero=JOptionPane.showInputDialog("Introduce tu género (H/M):");
 9
            }while(genero.equalsIgnoreCase("H")==false && genero.equalsIgnoreCase("M")==false);
10
            int altura=Integer.parseInt(JOptionPane.showInputDialog("Introduce altura en cm. "));
11
12
13
            int pesoideal=0;
14
15
            if(genero.equalsIgnoreCase("H")){
16
                pesoideal=altura-110;
17
            3
18
            else if(genero.equalsIgnoreCase("M")){
19
                pesoideal=altura-120;
20
            }
21
22
            System.out.println("Te peso ideal es " + pesoideal);
<u>23</u>
24
        }
   }
```

Línea 8 a la variable genero le asignamos un valor por teclado.

Entre las 7 y 9 controlamos que solo puedes contestar H o M indistinto mayúsculas o minúsculas, mientras no se así no saldremos de bucle.

La línea 11 asignamos a la variable altura de tipo int un valor por teclado.

La línea 13 a la variables pesoideal la definimos como int y un valor inicial de 0.

En la línea 15 comparamos que el valor de genero sea H o h para pasar el valor pesoideal=altua110.

En la línea 18 comparamos que el valor de genero sea M o m para pasar el valor pesoideal=altura-120.

En la línea 22 te dice cual es tu peso ideal.

#### Vamos a ejecutarlo.

Input		×
?	Introduce tu género (H/M): h	
	OK Cancel	

Pulsamos Ok.

Input		×
?	Introduce altura en cm. 190 OK Cancel	

Pulsamos Ok.

El resultado será:

Te peso ideal es 80





### Bucles IV (Vídeo 21)

El bucle For

- Sintaxis
  - For(inicio bucle, condición, contador bucle){
    - Línea1
    - Línea2
    - Línea3
    - Línea4

0 }

Vamos a ejecutar Eclipse.

Vamos a crear una clase nueva llamada Uso\_Bucle\_For.

1 2 public class Uso Bucle For 🛛 3 4⊝ public static void main(String[] args) { 5 // TODO Auto-generated method stub 6 7 8 for(int i=0;i<10;i++) {</pre> System.out.println("Juan"); 9 10 11 } 12 13 }

En el bucle for declaramos s una variable entera iniciando su valor a 0; la condición que ha de cumplir que i sea menor de 10; y el incremento sea de 1. I++

Vamos a ejecutar.

Para ver los valores que adquiere en cada vuelta vamos a realizar los siguente.

```
1
 2 public class Uso_Bucle_For {
 3
 4⊝
        public static void main(String[] args) {
P
 5
            // TODO Auto-generated method stub
 6
 7
            for(int i=0;i<10;i++) {</pre>
 8
                System.out.println("Juan " + i);
 9
            }
10
11
        }
12
13
    }
```

Cuando ejecutemos este será el resultado:

Juan 0
Juan 1
Juan 2
Juan 3
Juan 4
Juan 5
Juan 6
Juan 7
Juan 8
Juan 9
for(int i=10;i>0;i--) {
 System.out.println("Juan " + i);
}

Ahora cambiamos el valor inicial i igual 10 condición i mayor a 0 y como incremento i -- decrecerá en -1, este será el resultado:

```
Juan 10
Juan 9
Juan 8
Juan 7
Juan 6
Juan 5
Juan 4
Juan 3
Juan 2
Juan 1

for(int i=0;i<20;i+=2) {
    System.out.println("Juan " + i);
}</pre>
```

Inicializamos la variable i a 0, como condición i menor a vente y como incremente le suma cada vez 2, este será el resultado:

Juan 0 Juan 2 Juan 4 Juan 6 Juan 8 Juan 10 Juan 12 Juan 14 Juan 16 Juan 18

Vamos a realizar una nueva clase llamada comprueba\_mail que evalúe si una dirección de correo electrónica es correcta, para ello solo vamos a controlar si lleva una "@" arroba.

```
import javax.swing.JOptionPane;
 1
 2
 3
   public class comprueba mail 🛛
 4
 5Θ
        public static void main(String[] args) {
 6
            // TODO Auto-generated method stub
 7
            boolean arroba=false;
 8
            String mail=JOptionPane.showInputDialog("Introduce mail: ");
 9
            for(int i=0; i<mail.length();i++) {</pre>
10
                if(mail.charAt(i)=='@') {
11
                     arroba=true;
12
                }
                                                  Va con comillas simples
13
14
            if(arroba==true) {
                System.out.println("El correo intrducido es correcto.");
15
16
            }
            else{
17
18
                System.out.println("El correo introducido es incorrecto.");
19
            }
20
        }
21
22
   }
```

En la línea 7 definimos una variable de tipo boolean su valor puede ser true o false.

En la línea 8 asignamos el valor de la variable mail desde una ventana de diálogo.

En la línea 9 hacemos un ciclo for donde i empieza por cero has que sea menor a la longitud de la palabra que hemos introducido para ello es mail.length(), con un incremento de 1. De este modo en el interior de bucle comprueba carácter por carácter si encuentra la @, en caso afirmativo la variable arroba pasará a valer true, si en todo el bucle no ha encontrado la arroba la variable arroba seguirá valiendo false.

En la línea 14 cuando ya hemos salido del bucle comparamos si vale true para decirnos que "El correo introducido es correcto", de lo contrario nos dirá "El correo introducido es incorrecto".





## Bucles V (Vídeo 22)

Siguiendo con el ejemplo anterior queremos controlar que al introducir la dirección de correo electrónico este solo contenga una @ es decir que si tiene 0 o más de 1 que nos diga que el correo electrónico no es correcto.

```
import javax.swing.JOptionPane;
 1
 2
   public class comprueba mail 🛛
 З
 4
 5Θ
        public static void main(String[] args) {
 6
            // TODO Auto-generated method stub
            int arroba=0;
 7
            String mail=JOptionPane.showInputDialog("Introduce mail: ");
 8
            for(int i=0; i<mail.length();i++) {</pre>
 9
10
                if(mail.charAt(i)=='@') {
11
                    arroba++; <
12
                }
13
14
            if(arroba==1) {
15
                System.out.println("El correo intrducido es correcto.");
16
            }
17
            else{
18
                System.out.println("El correo introducido es incorrecto.");
19
            }
20
        }
21
22
   }
```

En la línea 7 hemos cambiado el tipo de variable a int con un valor inicial de 0.

En la línea 11 hemos puesto un contador para saber cuántas @ tiene el correo electrónico que hemos introducido.

En la línea 14 controlamos que si solo ha encontrado 1 @ de nuestro correo electrónico como correcto de lo contrario en la línea 17 nos dirá que lo contrario nos diga que el correo electrónico no es correcto.

Ejecuta el proyecto poniendo más de una arroba.

Input		×
?	Introduce mail: Pere@@pere.com OK Cancel	

Este será el resultado:

El correo introducido es incorrecto.

Ahora queremos controlar que nuestra dirección de correo por lo menos tenga un punto.

```
1 import javax.swing.JOptionPane;
2
3 public class comprueba_mail {
4
       public static void main(String[] args) {
5Θ
6
           // TODO Auto-generated method stub
7
            int arroba=0;
            boolean punto = false;
8
9
            String mail=JOptionPane.showInputDialog("Introduce mail: ");
            for(int i=0; i<mail.length();i++) {</pre>
10
                if(mail.charAt(i)=='@') {
11
                    arroba++;
12
13
                }
                if(mail.charAt(i)=='.') {
14
                    punto=true;
15
16
                }
17
            3
            if(arroba==1 && punto==true) {
18
19
                System.out.println("El correo intrducido es correcto.");
20
            }
            else{
21
                System.out.println("El correo introducido es incorrecto.");
22
23
            }
24
```

En la línea 8 hemos creado una variable llamada punto de tipo boolean con un valor inicial de false.

En la línea 14 controlamos que si encontramos un carácter que sea "." La variable punto pase a valer true.

En la línea 18 con un condicionante con el operador Y && queremos comprobar y ha encontrado una @ y por lo menos un . ya que una dirección de correo electrónico puede tener más de un punto.

Prueba una dirección de correo electrónica que no tenga punto.

Input		×
?	Introduce mail: pere@perees OK Cancel	

Este será el resultado:

El correo introducido es incorrecto.

Vamos a probar con una correo que tenga mas de un punto.

Input		$\times$
?	Introduce mail: pere.manel@gmail.com OK Cancel	

El resultado será:

```
El correo intrducido es correcto.
```

Vamos a realizar un nuevo proyecto para calcular el factorial de un número este se representa de la siguiente forma 6! Y esto es igual a 6 \* 5 \* 4 \* 3 \* 2 \* 1 es 720.

Vamos a crear una nueva clase llamado Factorial.

```
1 import javax.swing.*;
 2 public class Factorial {
 3
 4⊝
       public static void main(String[] args) {
 5
           // TODO Auto-generated method stub
 6
 7
           int resultado=1;
           int numero=Integer.parseInt(JOptionPane.showInputDialog("Introduce un número"));
 8
 9
           for (int i=numero; i>0;i--) {
               resultado=resultado*i;
10
11
           }
12
13
           System.out.println("El factorial de "+ numero + " es " + resultado);
14
       }
15
16 }
```

En la línea 7 definimos una variable de tipo int llamada resultado con el valor inicial de 1, esto se hace porque en la línea 10 acumularemos el producto de resultado por el valor de i que tenemos en el bucle for, si fuera 0 multiplicaría por 0 siempre su resultado sería 0.

En la línea 10 resultado en multiplicado por el valor de i, luego por i-1 y así hasta llegar al final del bucle.

En la línea 13 fuera del ciclo for imprimimos el valor de la variable resultado que tiene el valor del factorial del número introducido desde la ventana de diálogo.

Vamos a ejecutarlo:

Input		$\times$
?	Introduce un número 6 OK Cancel	

Este será el resultado:

El factorial de 6 es 720

Si al introducir valores estos se desbordan es porque la variable no puede almacenar dicho valor es decir en lugar de declararla int la podemos declarar long.

Long resultado=1L;

Recuerda poner la letra L después del valor de la variable.





## Arrays I (Vídeo 23)

#### Matrices (Arrays, Arreglos)

- ¿Qué es? Estructura de datos que contiene una colección de valores del mismo tipo.
- ¿Para qué? Para almacenar valores que normalmente tienen alguna relación entre sí.
- Sintaxis:
  - Declaración: int[] mi\_matriz=new int[10];

(j) Java	Hipotética representación gráfica				
		35	Variable de tipo entero		
	15 25	8 -7	92	Matriz de tipo entero	
				PLOORABIN	-ояматісле





Para declarar la matriz e inicializarla en la misma línea:

Int [] mi\_matriz = { 15, 25, 8, -7, 92 };

Vamos a crear una nueva clase llamada Uso\_Arrays.

```
1
 2
    public class Uso_Arrays 🛛
 3
 4⊝
        public static void main(String[] args) {
 5
            // TODO Auto-generated method stub
 6
            int [] mi_matriz=new int[5];
 7
            //También es correcto (int mi:matriz[]=new int[5];)
 8
            mi_matriz[0]=5;
 9
            mi_matriz[1]=38;
10
            mi_matriz[2]=-15;
11
            mi_matriz[3]=92;
            mi_matriz[4]=71;
12
13
14
            System.out.println(mi_matriz[3]);
15
16
        }
17
18 }
19
```

En la línea 6 declaramos la matriz para almacenar 5 valores.

De la línea 8 hasta 12 vamos asignando los valores de la matriz.

En la línea 14 imprimimos por consola el valor de la matriz almacenado en la posición 4 que es el tres, ya que las matrices empiezan por 0.

Ejecutamos y observamos el resultado:

92

```
1
   public class Uso Arrays 🖁
 2
 3
 4⊝
        public static void main(String[] args) {
 5
            // TODO Auto-generated method stub
 6
            int [] mi_matriz=new int[5];
            //También es correcto (int mi:matriz[]=new int[5];)
 7
 8
            mi matriz[0]=5;
 9
            mi_matriz[1]=38;
            mi_matriz[2]=-15;
10
            mi_matriz[3]=92;
11
            mi_matriz[4]=71;
12
13
            System.out.println(mi_matriz[0]);
14
            System.out.println(mi_matriz[1]);
15
            System.out.println(mi_matriz[2]);
16
            System.out.println(mi_matriz[3]);
17
18
            System.out.println(mi_matriz[4]);
19
        }
20
   }
21
```

Si queremos ver todos los valores que tiene almacenada la matriz, este será el resultado:

Para imprimir todos los valores de una matriz lo mejor es un bucle for.

```
1
 2
   public class Uso_Arrays {
 3
 40
        public static void main(String[] args) {
 5
            // TODO Auto-generated method stub
            int [] mi_matriz=new int[5];
 6
 7
            //También es correcto (int mi:matriz[]=new int[5];)
 8
            mi matriz[0]=5;
            mi_matriz[1]=38;
 9
10
            mi matriz[2]=-15;
11
            mi_matriz[3]=92;
12
            mi_matriz[4]=71;
13
            for(int i=0; i<5;i++) {</pre>
14
15
                 System.out.println("Valor del índice
                                                          +
                                                                         mi matriz[i]
16
            }
17
        }
18
19
    }
```

La variable i dentro del bucle for asumirá los valores de 0 hasta 4, de este modo podemos consultar toda la matriz, este será el resultado:

```
Valor del índice 0 = 5
Valor del índice 1 = 38
Valor del índice 2 = -15
Valor del índice 3 = 92
Valor del índice 4 = 71
```

Ahora vamos a ver otra forma de declarar una matriz.

```
1
   public class Uso_Arrays {
 2
 3
        public static void main(String[] args) {
 4⊝
 5
            // TODO Auto-generated method stub
 6
            int[] mi_matriz= {5, 38, -15, 92, 71};
 7
            for(int i=0; i<5;i++) {</pre>
 8
                System.out.println("Valor del índice " + i + " = " + mi_matriz[i]);
9
10
            }
11
        }
12
   }
13
```

En la línea 6 declaramos una matriz y además le asignamos los valores, esto se llama declaración simplificada o implícita.

Si ejecutamos, este será el resultado:

```
Valor del índice 0 = 5
Valor del índice 1 = 38
Valor del índice 2 = -15
Valor del índice 3 = 92
Valor del índice 4 = 71
1
 2
   public class Uso Arrays {
 3
40
        public static void main(String[] args) {
 5
           // TODO Auto-generated method stub
 6
           int[] mi_matriz= {5, 38, -15, 92, 71, 95, 85, 65, 25, 14, 78};
 7
 8
           for(int i=0; i<mi matriz.length;i++) {</pre>
 9
                System.out.println("Valor del indice " + i + " = " + mi_matriz[i]);
10
            }
        }
11
12
   }
13
```

Si queremos saber la longitud de una matriz, es decir cuantos valores almacena podemos utilizar la función lenght, tal como se muestra en el código, este será el resultado:

Valor del índice 0 = 5 Valor del índice 1 = 38 Valor del índice 2 = -15 Valor del índice 3 = 92 Valor del índice 4 = 71 Valor del índice 5 = 95 Valor del índice 6 = 85 Valor del índice 7 = 65 Valor del índice 8 = 25 Valor del índice 9 = 14 Valor del índice 10 = 78





### Arrays II (Vídeo 24)

Bucle for each (por cada).

Vamos a crear una clase nueva llamada Uso\_Arrays\_II.

```
1
    public class Uso Arrays II 🖁
 2
 3
        public static void main(String[] args) {
 4⊝
 5
            // TODO Auto-generated method stub
 6
            String []paises=new String[8];
 7
            paises[0]="España";
 8
            paises[1]="Mejico";
 9
            paises[2]="Colombia";
10
            paises[3]="Perú";
11
            paises[4]="Chile";
12
            paises[5]="Argentina";
13
            paises[6]="Ecuador";
14
            paises[7]="Venezuela";
15
            for(int i=0;i<8;i++) {</pre>
16
17
                 System.out.println("Pais "+ (i+1) +" " +paises[i]);
18
            }
19
        }
20
21 }
```

Este será el resultado:

Pais 1 España Pais 2 Mejico Pais 3 Colombia Pais 4 Perú Pais 5 Chile Pais 6 Argentina Pais 7 Ecuador Pais 8 Venezuela

Ahora vamos a modificar el bucle for con each.

```
1
   public class Uso_Arrays_II {
 2
 3
 40
        public static void main(String[] args) {
            // TODO Auto-generated method stub
 5
            String []paises=new String[8];
 6
            paises[0]="España";
 7
            paises[1]="Mejico";
 8
            paises[2]="Colombia";
9
            paises[3]="Perú";
10
            paises[4]="Chile";
11
            paises[5]="Argentina";
12
            paises[6]="Ecuador";
13
            paises[7]="Venezuela";
14
15
16
            for(String elemento:paises) {
17
                System.out.println(elemento);
18
            }
19
        }
20
   }
```

El nuevo for sabe el número de elementos que tiene la matriz.

Para inicializar la matriz y declarar los valores en una sola línea sería de la siguiente forma: String [] países={"España", "Méjico", "Colombia", "Perú", "Chile", "Argentina", "Ecuador", "Venezuela"};

Ahora vamos a rellenar una matriz por mediación de una ventana de diálogo:

```
import javax.swing.*;
 1
   public class Uso_Arrays_II {
 2
 3
        public static void main(String[] args) {
40
 5
            // TODO Auto-generated method stub
            String [] paises=new String[8];
 6
 7
            for (int i=0; i<8; i++){</pre>
                paises[i]=JOptionPane.showInputDialog("Introduce pais (" + (i+1)+") :");
8
9
            }
10
11
            for(String elemento:paises) {
12
                System.out.println(elemento);
13
            }
14
        3
```

Ejecuta el programa y una vez introducidos los 8 países estos se verán por la consola. Vamos a crear una matriz que se rellenará con valores aleatorios.



Este será el resultado:

```
61 8 88 72 58 95 73 73 94 15 9 34 97 60 53
97 4 19 5 2 16 49 100 48 50 12 54 90 97 51
7 51 43 75 51 29 36 2 61 9 86 91 64 46 71
3 33 27 63 4 93 11 55 40 31 65 52 0 46 17
24 100 47 98 37 11 97 4 84 94 50 16 33 69 56
62 10 27 0 96 59 52 75 69 28 74 86 21 19 13
87 44 38 51 29 42 69 76 23 18 89 97 82 99 92
_____
35 73 61 3 99 28 41 2 0 92 25 30 34 62 27
 .....
54 73 45 66 77 99 97 87 23 44 56 78 21 60 40
.....
75 80 43 12 49 28 39 20 44 91 93 12 4 27 25
 -----
```





### Arrays III. Arrays bidimensionales I. (Vídeo 25)

Matrices bidimensionales

چ)) java	Hipotética representación gráfica					
	0,0	1,0	2,0	3,0		
	0,1	1,1	2,1	3,1		
	0,2	1,2	2,2	3,2		
	0,3	1,3	2,3	3,3		
	0,4	1,4	2,4	-1.3,4	иАтісле	

Vamos a crear una clase nueva llamada Arrays\_bidimensionales.

```
1 public class Arrays_bidimensionales {
 2
       public static void main(String[] args) {
3Θ
 4
           // TODO Auto-generated method stub
 5
            int[][] matrix=new int[4][5];
 6
7
           matrix[0][0]=15;
           matrix[0][1]=21;
8
           matrix[0][2]=18;
9
10
           matrix[0][3]=9;
11
           matrix[0][4]=15;
12
13
           matrix[1][0]=10;
14
           matrix[1][1]=52;
15
           matrix[1][2]=17;
           matrix[1][3]=19;
16
17
           matrix[1][4]=7;
18
19
           matrix[2][0]=19;
20
           matrix[2][1]=2;
21
           matrix[2][2]=19;
22
           matrix[2][3]=17;
23
           matrix[2][4]=7;
24
25
           matrix[3][0]=92;
           matrix[3][1]=13;
26
           matrix[3][2]=13;
27
           matrix[3][3]=32;
28
29
           matrix[3][4]=41;
30
31
            System.out.println(matrix[2][3]);
32
33
        }
   }
```

En la línea 5 definimos la matriz bidimensional y desde la línea 7 hasta la línea 29 le asignamos los valores.

En la línea 31 queremos ver en consola el valor que se guarda en la matriz en la posición 2-3 matrix[2][3].

```
public class Arrays_bidimensionales {
      public static void main(String[] args) {
             // TODO Auto-generated method stub
             int[][] matrix=new int[4][5];
             matrix[0][0]=15;
             matrix[0][1]=21;
             matrix[0][2]=18;
             matrix[0][3]=9;
             matrix[0][4]=15;
             matrix[1][0]=10;
             matrix[1][1]=52;
             matrix[1][2]=17;
             matrix[1][3]=19;
             matrix[1][4]=7;
             matrix[2][0]=19;
             matrix[2][1]=2;
             matrix[2][2]=19;
             matrix[2][3]=17;
             matrix[2][4]=7;
             matrix[3][0]=92;
             matrix[3][1]=13;
             matrix[3][2]=13;
             matrix[3][3]=32;
             matrix[3][4]=41;
             for(int i=0;i<4;i++) {</pre>
                    for(int j=0;j<5;j++) {</pre>
                           System.out.print(matrix[i][j]+ " ");
                    }
                    System.out.println();
             }
      }
}
Este sería el resultado:
```

Este seria el resultado

15 21 18 9 15 10 52 17 19 7 19 2 19 17 7 92 13 13 32 41





### Arrays IV. Arrays bidimensionales II. (Vídeo 26)

Otra forma de declarar una matriz bidimensional.

```
1 public class Arrays_bidimensionales {
 2
        public static void main(String[] args) {
 30
 4
            // TODO Auto-generated method stub
 5
            int[][] matrix= {
 6
                     {10,15,18,19,21},
 7
                     {5,25,37,41,15},
8
                     {7,19,32,14,90},
9
                     {85,2,7,40,27}
10
            };
11
12
13
14
            for(int i=0;i<4;i++) {</pre>
15
                for(int j=0;j<5;j++) {</pre>
                     System.out.print(matrix[i][j]+ " ");
16
17
                }
18
                System.out.println();
19
            }
20
        }
21 }
```

Si ejecutamos este será el resultado:

10 15 18 19 21 5 25 37 41 15 7 19 32 14 90 85 2 7 40 27

Si nosotros a la hora de leer una matriz por error intentamos leer valores que no tiene.

```
for(int i=0;i<4;i++) {
    for(int j=0;j<6;j++) {
        System.out.print(matrix[i][j]+ " ");
    }
    System.out.println();
}</pre>
```

Observaremos el siguiente error.

Vamos a leer la matriz bidimensional con el método for con each.

```
1 public class Arrays_bidimensionales {
 2
       public static void main(String[] args) {
 30
 4
            // TODO Auto-generated method stub
 5
            int[][] matrix= {
 6
                    {10,15,18,19,21},
 7
                    {5,25,37,41,15},
 8
                    {7,19,32,14,90},
9
                    {85,2,7,40,27}
10
            };
11
            for(int[]fila:matrix) {
12
                System.out.println("");
13
14
                for(int z: fila) {
                   System.out.print(z + " ");
15
16
                }
17
            }
18
19
        }
20 }
```

Si ejecutamos este será el resultado:

10 15 18 19 21 5 25 37 41 15 7 19 32 14 90 85 2 7 40 27

	10%	11%	12%	13%	14%	15%
1	10.000,00€	10.000,00€	10.000,00€	10.000,00€	10.000,00€	10.000,00€
	11.000,00€	11.100,00€	11.200,00€	11.300,00€	11.400,00€	11.500,00€
	12.100,00€	12.321,00€	12.544,00€	12.769,00€	12.996,00€	13.225,00€
	13.310,00€	13.676,31€	14.049,28€	14.428,97€	14.815,44 €	15.208,75€
	14.641,00€	15.180,70€	15.735,19€	16.304,74€	16.889,60€	17.490,06€

Vamos a realizar un ejercicio para poder almacenar esta array en Java.

Vamos a crear una nueva case llamada Ejemplo\_Array\_2D.

```
1
   public class Ejemplo_Array_2D {
2
3
4⊝
        public static void main(String[] args) {
 5
            // TODO Auto-generated method stub
 6
            double acumulado;
 7
            double interes=0.10;
8
            double[][] saldo=new double[6][5];
9
10
            for(int i=0; i<6;i++) {</pre>
11
                saldo[i][0]=10000;
12
                acumulado=10000;
13
                for(int j=1;j<5;j++) {</pre>
14
                     acumulado=acumulado+(acumulado*interes);
15
                     saldo[i][j]=acumulado;
                }
16
17
                interes=interes+0.01;
18
19
            }
20
            for(int i=0;i<6;i++) {</pre>
21
                System.out.println();
22
                for(int j=0;j<5;j++) {</pre>
                     System.out.printf("%1.2f",saldo[i][j]);
23
                                              ");
24
                     System.out.print(" €
25
                }
26
            }
27
        }
28
29
   }
```

Este será el resultado:

10000,00	€	11000,00	€	12100,00	€	13310,00	€	14641,00	€
10000,00	€	11100,00	€	12321,00	€	13676,31	€	15180,70	€
10000,00	€	11200,00	€	12544,00	€	14049,28	€	15735,19	€
10000,00	€	11300,00	€	12769,00	€	14428,97	€	16304,74	€
10000,00	€	11400,00	€	12996,00	€	14815,44	€	16889,60	€
10000,00	€	11500,00	€	13225,00	€	15208,75	€	17490,06	€





# Contenido

Presentación (Vídeo 1) 1
Instalación JRE y Eclipse (Vídeo 2)3
Introducción (Vídeo 3)13
Estructuras principales I (Vídeo 4)15
Estructuras principales II (Vídeo 5)23
Estructuras principales III. Declaración variables Eclipse (Video 6)
Estructuras principales IV. Constantes y Operadores (Vídeo 7)
Estructuras principales V. Constantes y Operadores II (Vídeo 8)
Estructuras principales IV Clase Math (Vídeo 9)
Estructuras principales VII. Clase Math II. (Vídeo 10)42
Manipulación de cadenas. Clase String I (Vídeo 11)45
Manipulación de cadenas. Clase String II (Vídeo 12)49
Acercamiento a la API Paquetes (Vídeo 13)52
Entrada Salida datos I (Vídeo 14) 58
Entrada Salida datos II. (Vídeo 15)62
Condicionales I. (Vídeo 16)66
Condicionales II (Vídeo 17)69
Bucles I (Vídeo 18)71
Bucles II. (Vídeo 19)73
Bucles III. (Vídeo 20)75
Bucles IV (Vídeo 21)78
Bucles V (Vídeo 22)
Arrays I (Vídeo 23)
Arrays II (Vídeo 24)91
Arrays III. Arrays bidimensionales I. (Vídeo 25)94
Arrays IV. Arrays bidimensionales II. (Vídeo 26)97