

INGENIERÍA MECATRÓNICA



DI\_CERO

DIEGO CERVANTES RODRÍGUEZ

JAVA 8

APACHE NETBEANS IDE 12.5

Paquetes y API de Java

## Contenido

Clase Predefinida .....	2
Paquetes .....	3
Importación de Paquetes.....	6
¿Qué es una API? .....	6
Tipos de APIs: .....	7
Referencias: .....	10



# Clase Predefinida

Como cualquier programa Java está compuesto de puras clases (por lo menos de una), podemos hacer una clase propia o usar una de las que ya están creadas previamente en el lenguaje Java llamadas clases predefinidas, ya sea para trabajar con redes, manejar arrays, diseñar una interfaz gráfica, etc. de esta manera no tenemos que crear todo el código desde cero y más bien utilizar lo que ya está hecho (existen cientos de clases predefinidas para todo tipo de cosas).



Las clases propias son aquellas que creamos nosotros para organizar y/o comunicar las diferentes partes de nuestros proyectos.

Todas las clases no importando si son propias o predefinidas se ordenan por paquetes y éste viene indicado en la parte de arriba del nombre de la clase cuando vemos su documentación en la API.

Java™ Platform Standard Ed. 7

All Classes

Packages

- java.applet
- java.awt
- java.awt.color
- java.awt.datatransfer
- java.io
- java.lang
- java.math
- java.net
- java.nio
- java.rmi
- java.security
- java.sql
- java.util
- java.util.concurrent
- java.util.logging
- java.util.regex
- java.util.zip

StAXResult

StAXSource

Streamable

StreamableValue

StreamCorruptedException

StreamFilter

StreamHandler

StreamPrintService

StreamPrintServiceFactory

StreamReaderDelegate

StreamResult

StreamSource

StrictMath

**String**

StringBuffer

Overview Package **Class** Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

java.lang

## Class String

java.lang.Object  
java.lang.String

All Implemented Interfaces:  
Serializable, CharSequence, Comparable<String>

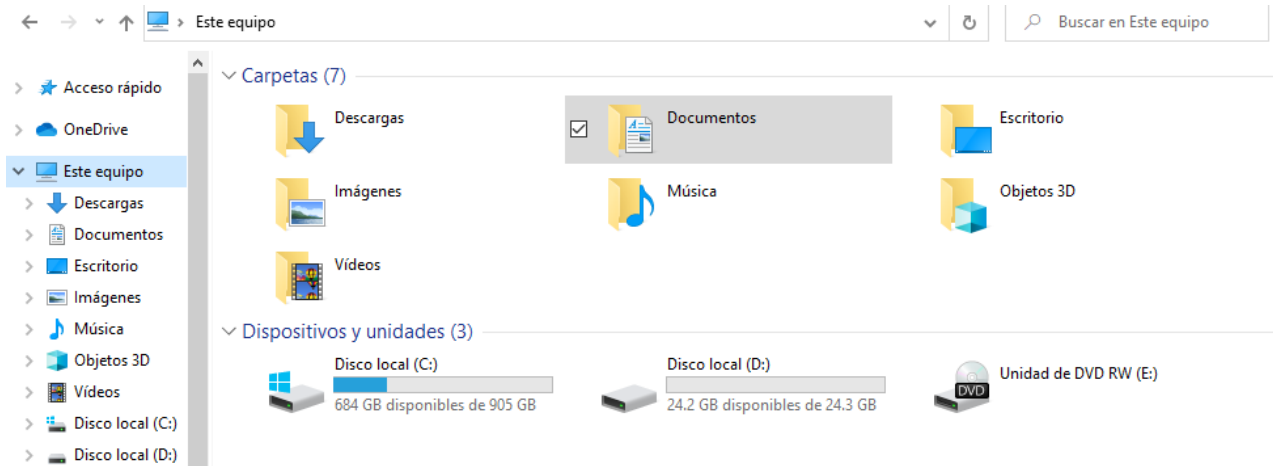
```
public final class String
extends Object
implements Serializable, Comparable<String>, CharSequence
```

The String class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class. Strings are constant; their values cannot be changed after they are created. String literals are used to create String objects.

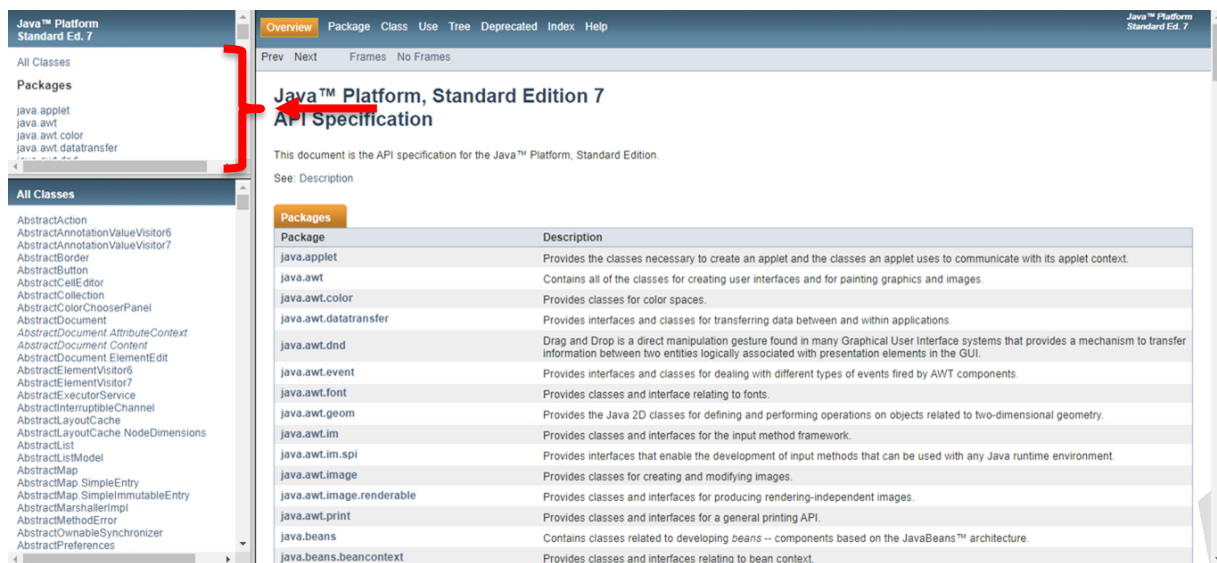
```
String str = "abc";
```

# Paquetes

Los paquetes realizan la misma función que las carpetas de nuestra computadora, ya que con ellas puedo juntar todas las clases que compartan ciertas características, así como en nuestra computadora hay una carpeta que almacena todas las imágenes, otra que almacena todos los videos, otra para los documentos, etc.



Por lo tanto, las clases de Java están organizadas en paquetes (que es lo mismo a pensar en carpetas) y todos los paquetes del lenguaje Java vienen indicados en la parte donde dice Packages dentro de la documentación de la API.



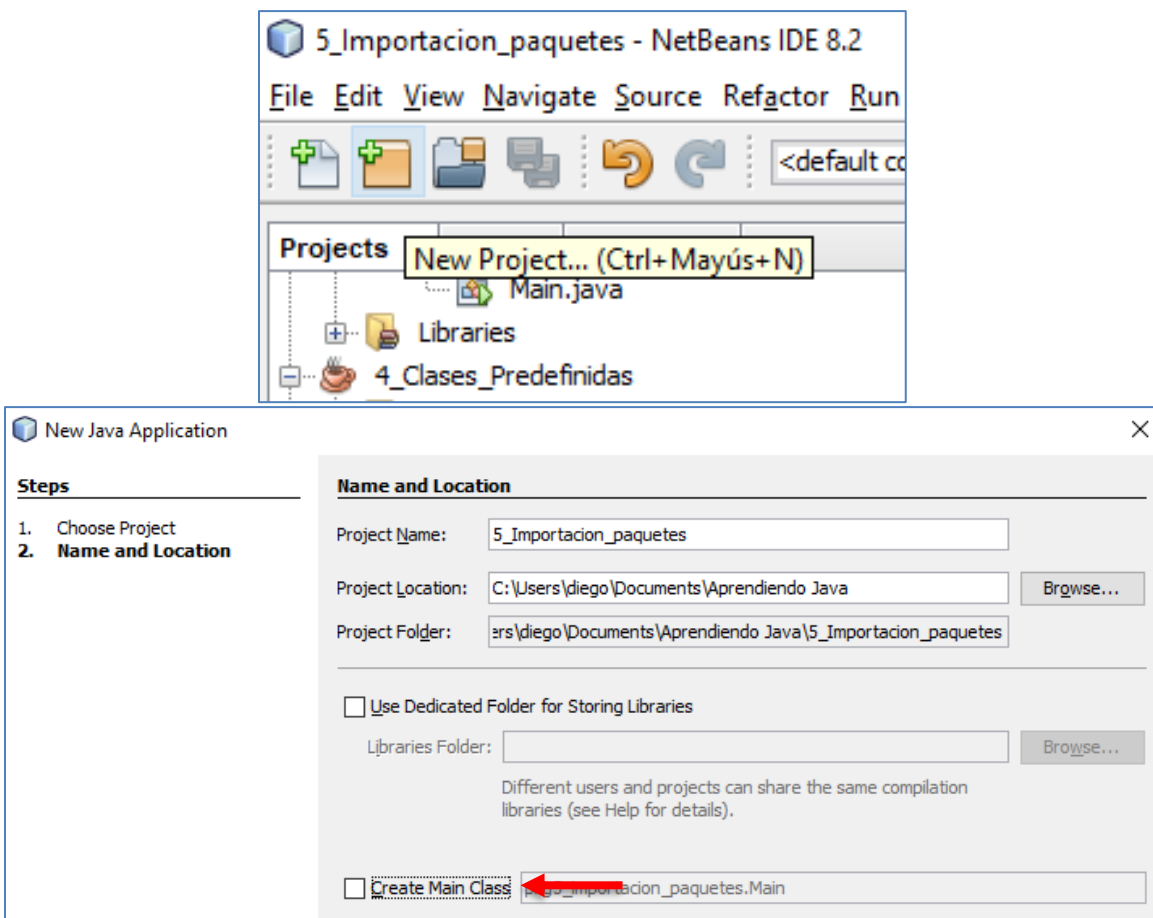
A esta parte se le llama panel de paquetes y lo que hace es representar el orden jerárquico de los paquetes en la API. Los **paquetes principales** de los que se desprenden todos los demás son el paquete **java**, el paquete **javax** y el paquete **org** (que es más raro de usar por eso no la vamos a mencionar ahorita), abajo se representan solo algunos subpaquetes de todos los existentes, esto es como meterse a una subcarpeta de cada carpeta de mi computadora.



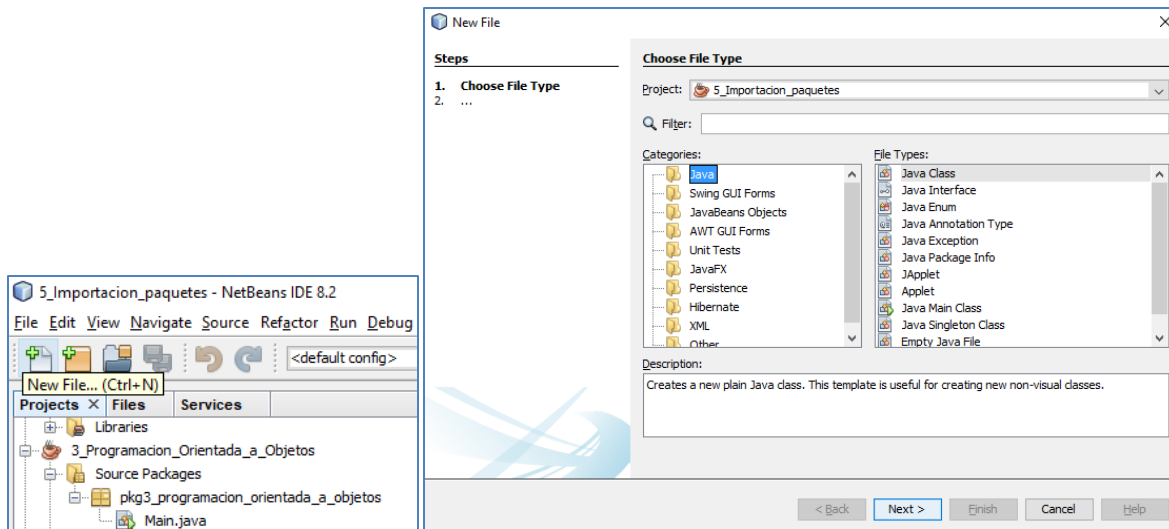
Dentro de cada uno de estos paquetes hay una serie de clases y se representa la jerarquía de los paquetes, subpaquetes y clases incluidas con la nomenclatura del punto:

**paquete\_principal.subpaquete.clase...**

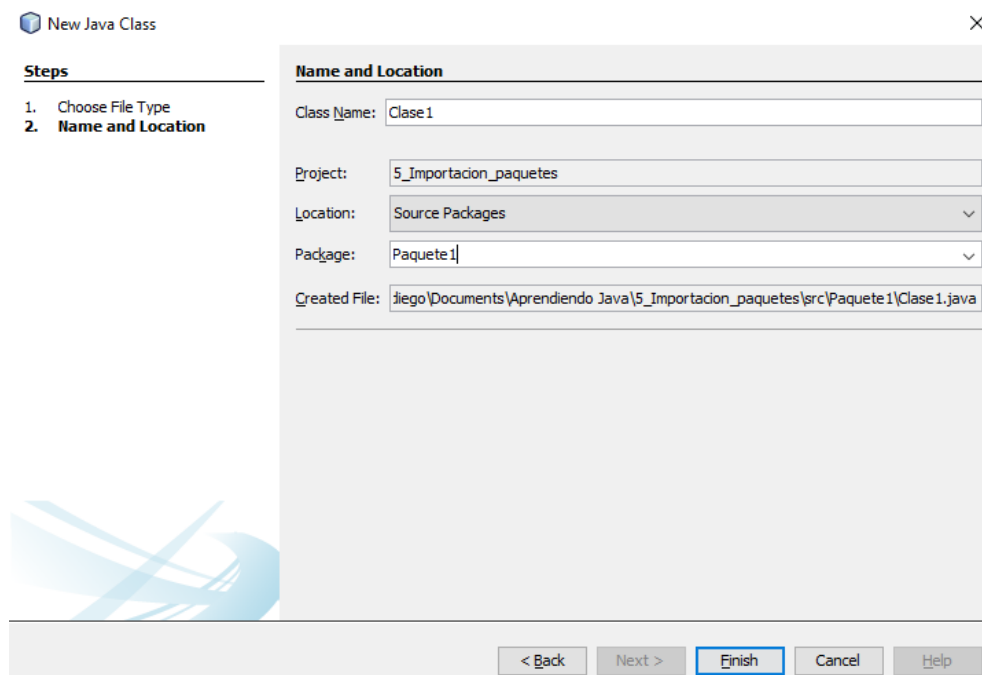
Si quiero crear un paquete personalizado para nuestras clases cuando creamos un nuevo proyecto, primero que nada, debo hacer un nuevo proyecto sin que el programa por sí solo cree una clase, esto se lo indico al programa deseleccionando el checkbox que dice Create Main Class:



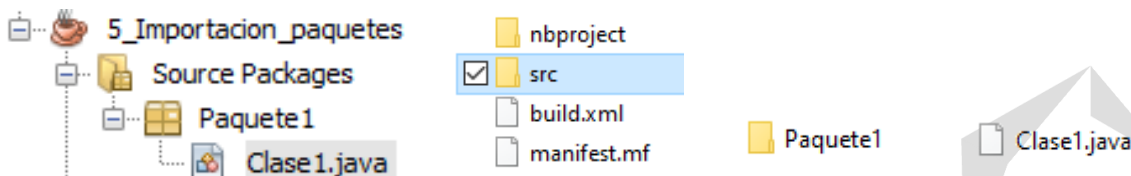
Luego deberé crear una clase en mi proyecto para indicarle en qué paquete se debe encontrar, anteriormente dejábamos que se creará en un paquete por defecto que hace Netbeans.



Pero ahora lo que haremos es crear un paquete en la parte donde dice Package.

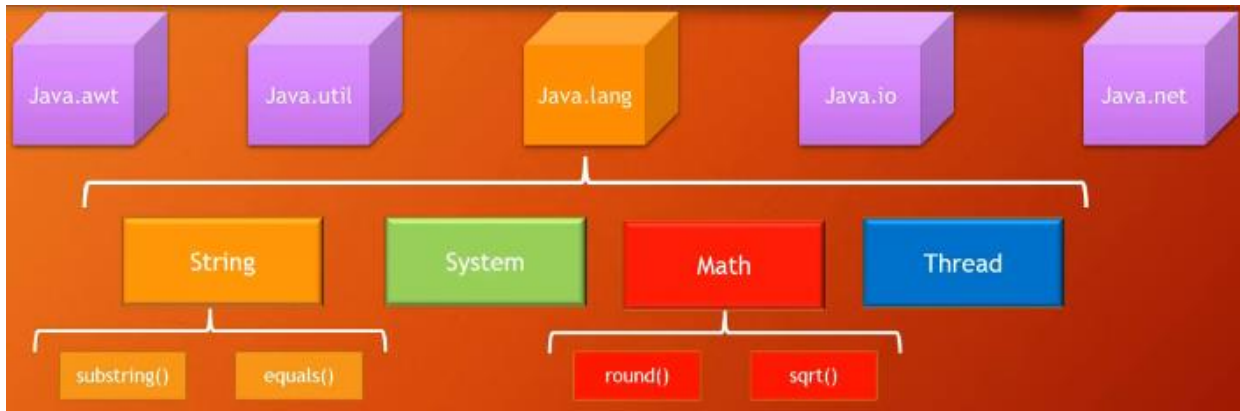


Y como podemos ver, el paquete (parecido a una carpeta) se creará en la carpeta del proyecto dentro de Source Packages (src):



# Importación de Paquetes

Es necesario saber el paquete de la clase predefinida que quiera usar porque a veces pueden existir dos clases con el mismo nombre dentro de la API siempre y cuando se encuentren en paquetes diferentes o porque cuando quiera usar una clase predefinida que no se encuentre en el paquete principal llamado **java.lang** tendré que importarlo para que me sea posible usarlo en mi código.



Para poder importar un paquete debemos escribir la siguiente línea de código hasta el principio del código, justo después de la instrucción que indica el paquete donde se encuentra mi programa:

```
package paquete_donde_se_encuentra_mi_programa;  
import nombre_del_paquete.subpaquete.nombre_clase_importada...;
```

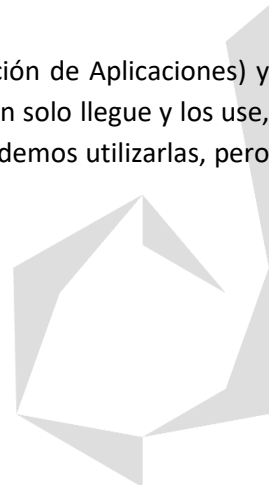
Esto se usa para cuando quiero importar un paquete con un subpaquete en específico y es útil porque de esta manera el programa que esté creando usará menos memoria RAM a que si importara todas las clases del paquete.

```
package paquete_donde_se_encuentra_mi_programa;  
import nombre_del_paquete.*;
```

Esto se usa para cuando quiero importar todos los subpaquetes del paquete que estoy importando.

## ¿Qué es una API?

Sus siglas significan Application Programming Interface (o Interfaz de Programación de Aplicaciones) y básicamente es una biblioteca de programas previamente hechos para que alguien solo llegue y los use, las APIs normalmente no muestran cómo realizan sus funciones, simplemente podemos utilizarlas, pero no ver su código fuente.



## Tipos de APIs:

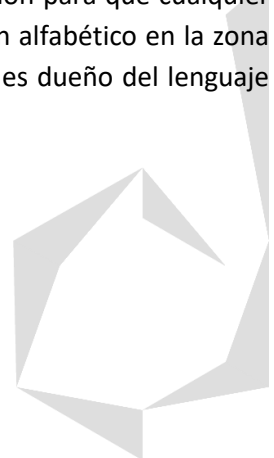
- **REST API:** Es el tipo de API que usualmente usan las empresas como Facebook, Twitter, PayPal, Google, etc. para compartir funcionalidades o datos de sus aplicaciones.
  - La forma en la que podemos acceder a la REST API de cualquier empresa es viendo sus documentaciones, cada empresa tiene una forma en específico de hacerlo y se puede encontrar buscando en internet con el url:  
**developer.empresa.com**  
Ejemplos:  
La API de Marvel es **developer.marvel.com**  
La API de Facebook es **developer.facebook.com**  
La API de Google es **developer.google.com**

Usualmente cuando consultamos este tipo de APIs lo que nos retornarán es una línea de código llamada JSON (JavaScript Object Notation) o XML que simplemente son datos organizados de cierta manera para que sean más entendibles y además puedan ser usados como un array dentro de nuestros programas.

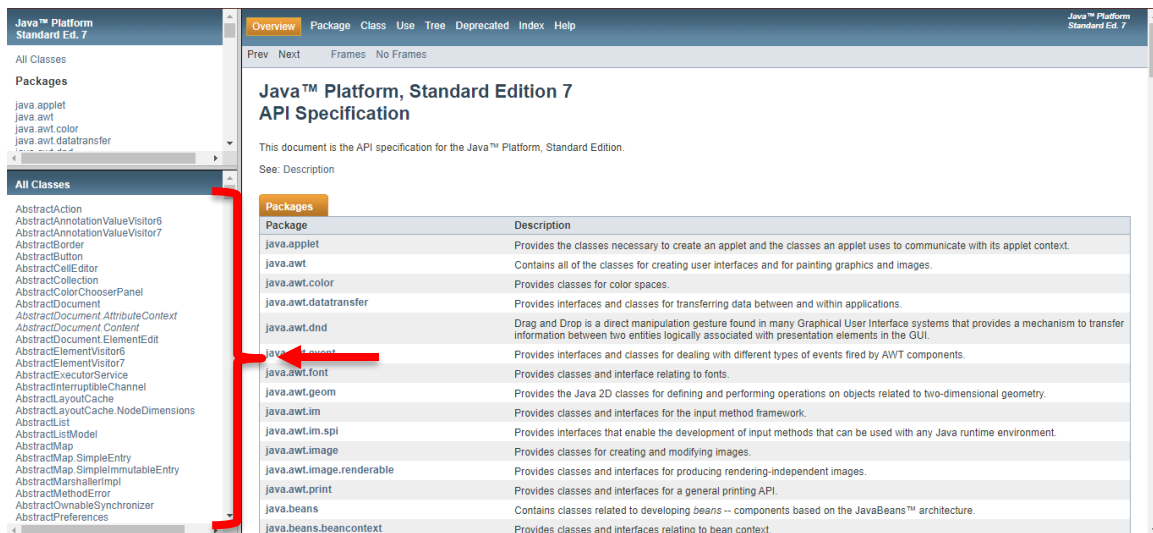
- **APIs para lenguajes orientados a objetos:** Sirven para proporcionar funcionalidades o constantes a algunos lenguajes de programación orientados a objetos como Java, PHP, Arduino, etc. para que los programadores que quieran crear su código con estos lenguajes no deban hacer todo desde cero.
  - Siempre para acceder a este tipo de APIs es necesario haber descargado algún tipo de software que me permita usar el lenguaje de programación para después incluir una línea de código que me permita utilizar cada parte (osea funcionalidad o constante) de la API.
- **APIs para lenguajes de procedimientos:** Sirven para comunicar dos aplicaciones que estén escritas en un lenguaje que no esté orientado a objetos.
  - Podemos acceder a este tipo de APIs de la misma manera con la que se accede a las APIs para lenguajes orientados a objetos.

La API de Java contiene todas las clases disponibles en el lenguaje de programación para que cualquier programador pueda consultarla y usarla en sus códigos. Está organizada en orden alfabético en la zona de la pantalla que dice All Classes y la página de documentación de Oracle (que es dueño del lenguaje Java) la podemos encontrar en el siguiente link:

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







En la API están incluidas las clases predefinidas que vienen incluidas con el lenguaje Java como la clase Math que nos permite realizar cálculos matemáticos, la clase String que se encarga de la manipulación o análisis de palabras (o también llamadas cadenas de caracteres), etc. y consultando la API puedo ver cómo usar cada una de ellas.

En general las APIs son un conjunto de programas escritos en cualquier lenguaje de programación que nos permiten obtener información o funcionalidades de una empresa (como Oracle, Google, Facebook, NASA, etc.), lenguaje de programación (como Java, PHP, Arduino, etc.) o un programador cualquiera (como la API de Pokémon llamada PokéAPI, la de Harry Potter llamada potterapi, etc.) a través de sus librerías para que podamos implementarlas a nuestros propios proyectos sin tener la necesidad de programarlos desde cero, simplificando la chamba de los programadores, en el caso específico del lenguaje Java la empresa que proporciona la API es Oracle, el programa proporcionado por la API se puede ejecutar o no dentro de nuestra propia computadora o servidor, esto depende de si la descargamos en nuestro disco duro o simplemente la jalamos de internet por medio de una línea de código, si elijo hacerlo de manera remota la API agarrará los datos de mi página o programa y se los enviará al servidor donde está alojada la API para que éste pueda ejecutar la acción que le estoy solicitando y después me regrese el resultado.

El uso de APIs se ve por ejemplo en los botones de redes sociales dentro de las páginas web de noticias.



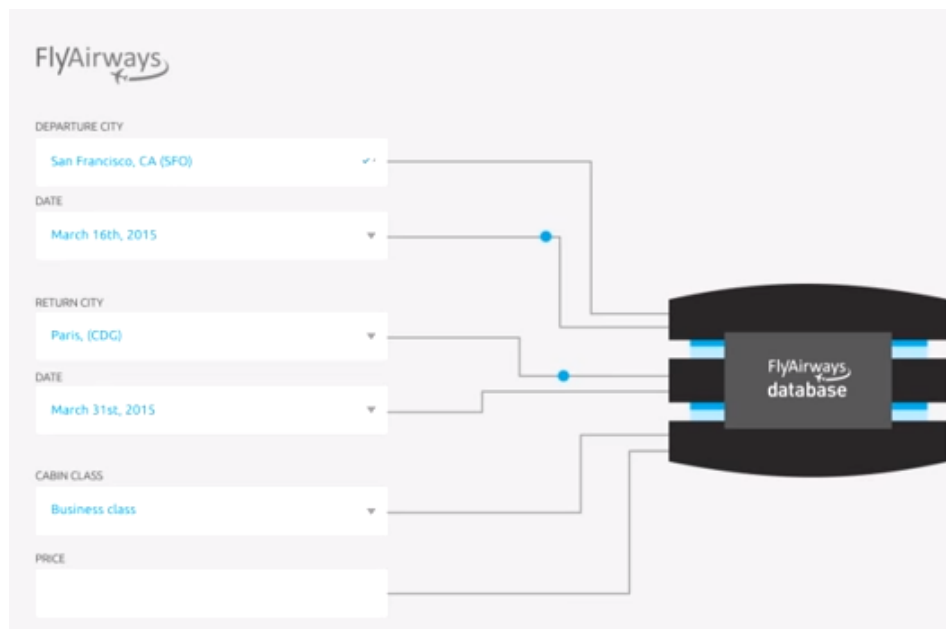
Dentro de este tipo de páginas cuando le doy click a los botones de redes sociales se me permite compartir dicha noticia en mi propio muro; esto es posible gracias a las APIs que proporcionan una

extensión del código de las redes sociales dentro de la del periódico sin tener la necesidad de copiar la noticia, salir de la página del periódico y pegarla en mi muro ni que tampoco las empresas de dichas redes sociales me compartan literal su código, ya que las APIs ejecutan acciones pero no nos muestran el código fuente para saber cómo lo hizo, si quisiéramos hacer lo mismo sin el uso de APIs, Facebook o Twitter tendrían que compartir su código públicamente lo que sería difícil, tedioso y hasta peligroso para ellos ya que esa información los podría hacer hasta más vulnerables a hackeos.

Esto también se usa por ejemplo cuando hacemos un pedido por internet, cuando hacemos una reservación en un restaurante, cuando reservamos un vuelo, cuando queremos acceder al GPS del celular de mi usuario, etc. Todo esto normalmente se hace a través de librerías usadas por las APIs.



Otro ejemplo del uso de APIs son las páginas de internet que sirven para darte los mejores precios para tu vuelo por medio de una comparación de precios entre una gran gama de aerolíneas, como Trivago, Booking, etc.



Lo que hace la página web es que interactúa con las APIs de las diferentes aerolíneas, esto no significa que tiene acceso directo a sus bases de datos, sino que los datos que el usuario ingresó al sitio web como la ciudad de partida, la fecha de partida, la ciudad de regreso y la fecha de regreso se los pasa a las APIs para que estos datos le lleguen a las páginas de las diferentes aerolíneas, después las APIs regresan el costo total de los vuelos de cada aerolínea y el usuario puede visualizar la recopilación de los diferentes precios que se obtuvo, así se crea la conectividad entre diferentes aplicaciones.



## Referencias:

Pildoras Informáticas, “Curso Java”, 2023 [Online], Available:  
<https://www.youtube.com/playlist?list=PLU8oAlHdN5BktAXdEVCLUYzvDyqRQJ2Ik>

MuleSoft Videos, “What is an API?”, 2015 [Online], Available:  
<https://www.youtube.com/watch?v=s7wmiS2mSXY>

