

DIPLOMATURA EN

# NUEVAS TECNOLOGÍAS

## MÓDULO 1

# Introducción a la Programación

Docente:

Esp. Ing. Martín Polliotto





## Repaso de Semana anterior

- ✓ **Relaciones entre clases:**
  - Tiene un: **Asociaciones**
  - Es un: **Herencia**
  - Arreglos de objetos
  - Clases **abstractas**
  - **Polimorfismo**



DIPLOMATURA EN

**NUEVAS  
TECNOLOGÍAS**



Ministerio de  
**PROMOCIÓN DEL EMPLEO  
Y DE LA ECONOMÍA FAMILIAR**

Ministerio de  
**CIENCIA Y  
TECNOLOGÍA**



Con el apoyo de  
**Oficina de Montevideo**  
Oficina Regional de Genios  
para América Latina y el Caribe

## Contenidos Semana 07:

- 7.1 Api de Java
- 7.2 Paquetes
- 7.3 Clases de envoltorio
- 7.4 Strings
- 7.5 Date y Calendar
- 7.6 Interfaces Gráficas con swing
  - Ventanas
  - Componentes
  - Eventos



DIPLOMATURA EN

**NUEVAS  
TECNOLOGÍAS**

SECRETARÍA DE  
EXTENSIÓN  
UNIVERSITARIA  
UTN - FRC

**SEU**

**UTN**  
Facultad Regional Córdoba

Ministerio de  
**PROMOCIÓN DEL EMPLEO  
Y DE LA ECONOMÍA FAMILIAR**

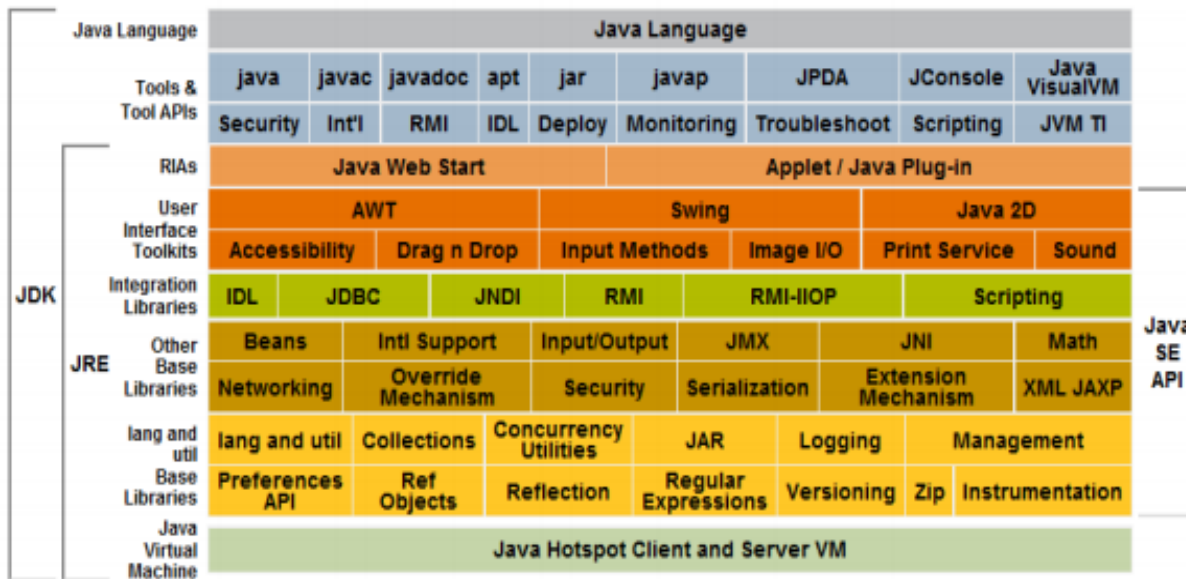
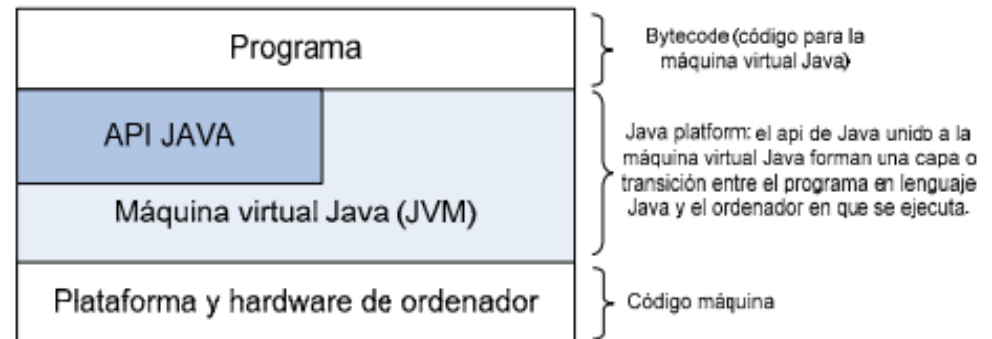
Ministerio de  
**CIENCIA Y  
TECNOLOGÍA**

**CBA**  
ENTRE TODOS

Organización  
de las Naciones Unidas  
para la Educación,  
la Ciencia y la Cultura

Con el apoyo de  
**Oficina de Montevideo**  
Oficina Regional de Genios  
para América Latina y el Caribe

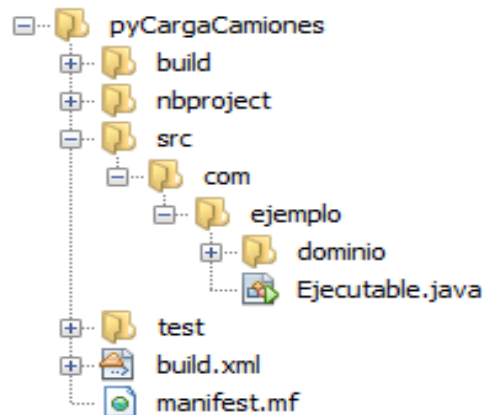
# 7.1 Api de Java



Conjunto de **clases agrupadas en paquetes** que ofrece Java y que están a disposición de todos los programadores listas para ser usadas y/o extendidas.

## 7.2 Paquetes

- ✓ Son la forma en la que Java nos permite agrupar de alguna manera lógica los componentes de nuestra aplicación que estén relacionados entre sí.
- ✓ Ayudan a darle una buena organización a la aplicación ya que permiten modularizar nuestra solución
- ✓ ***package*** *ruta.del.paquete*
- ✓ Son el equivalente a una carpeta de nuestro sistema de archivo

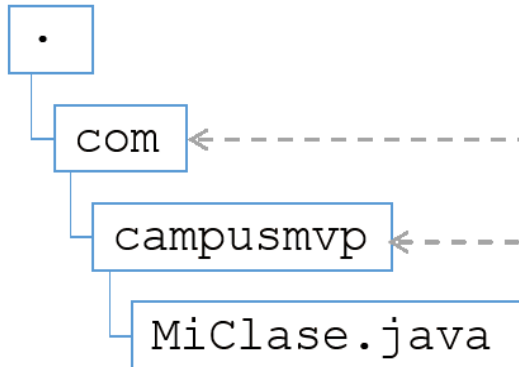


## 7.2 Paquetes

- ✓ El nombre del paquete se define de manera inversa al dominio de la organización o grupo. Por ejemplo, dominioempresa.com puede ser usado como nombre de paquete así: **com.dominio\_empresa.utilidades**.
- ✓ El uso de *packages* permite solucionar potenciales problemas en cuanto a nombres repetidos
- ✓ Para que una clase pueda acceder a otras clases que se encuentran en un package distinto, se usa la instrucción **import** al comienzo del archivo de la clase.

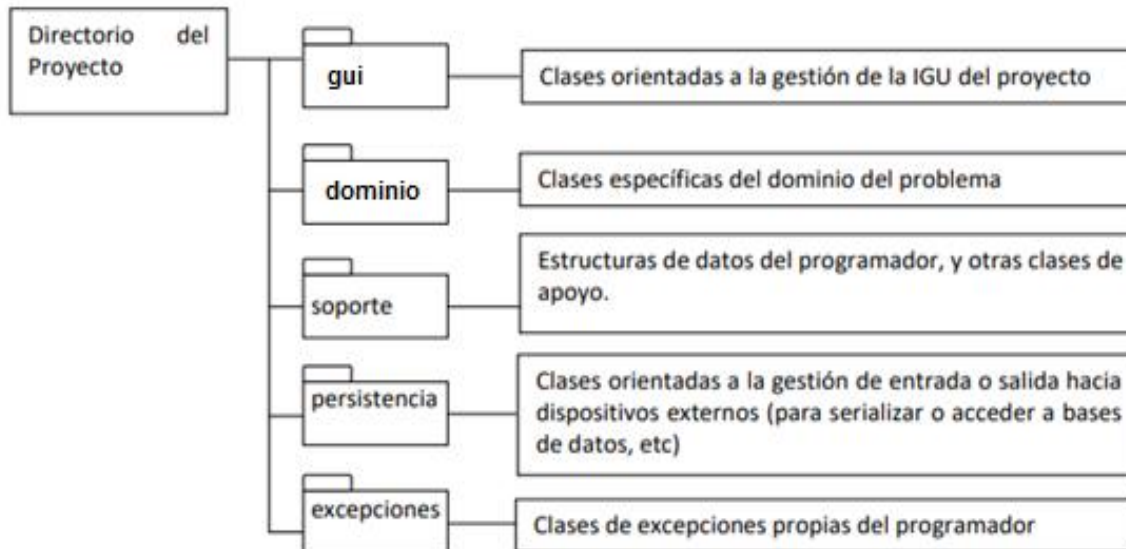
## 7.2 Paquetes

Sistema de archivos



Módulo de código fuente

```
// Declaración del paquete  
// al que pertenece la clase  
package com.campusmvp;  
  
class MiClase {  
    // Definición de la clase  
}
```



## 7.3 Clases de envoltorio

Tipo primitivo	Wrapper class
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

Todas marcadas como **final**  
y sus atributos también

- Asignar primitivos a **objetos**

*Integer entero = new Integer(10);*

**o bien**

*Integer entero = 10;*

*System.out.println("Entero: " + entero.intValue());*

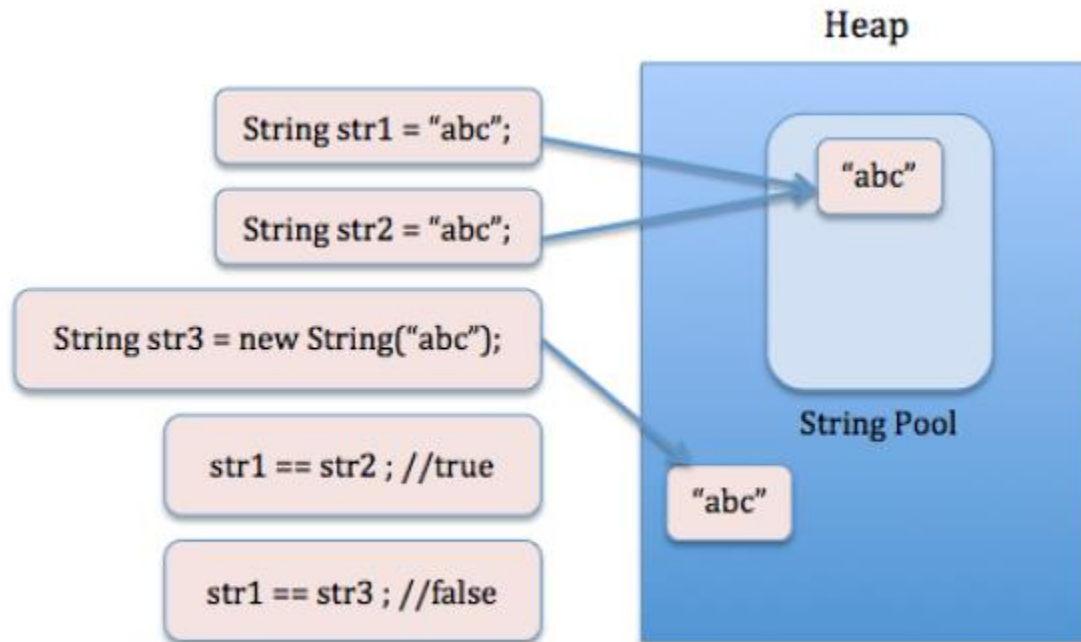
- **Convertir** cadenas a primitivos

*String valor = "10";*

*int val = Integer.parseInt(valor);*



## 7.4 Strings



### Para comparar:

El método `compareTo()` devuelve:

0 : cadenas **iguales**

> 0 cad1 **mayor que** cad2

< 0 cad1 **menor que** cad2

### Para concatenar:

Para concatenar + o +=

**StringBuilder**

Las cadenas son **Inmutables**. Para comparar cadenas también usamos: `equals()` para saber si son iguales o no



## 7.5 Date y Calendar

**Date** d = new **Date**(); // Crea el objeto Date

**Calendar** calendar = **Calendar.getInstance**(); // Obtiene una instancia de  
//Calendar

calendar.setTime(date); // Asigna la fecha al Calendar

int year = calendar.get(Calendar.**YEAR**);

int month = calendar.get(Calendar.**MONTH**);

int weekOfMonth = calendar.get(Calendar.**WEEK\_OF\_MONTH**);



Con Calendar podemos hacer aritmética de fechas: sumar o restar fechas, agregar días a una fecha, etc

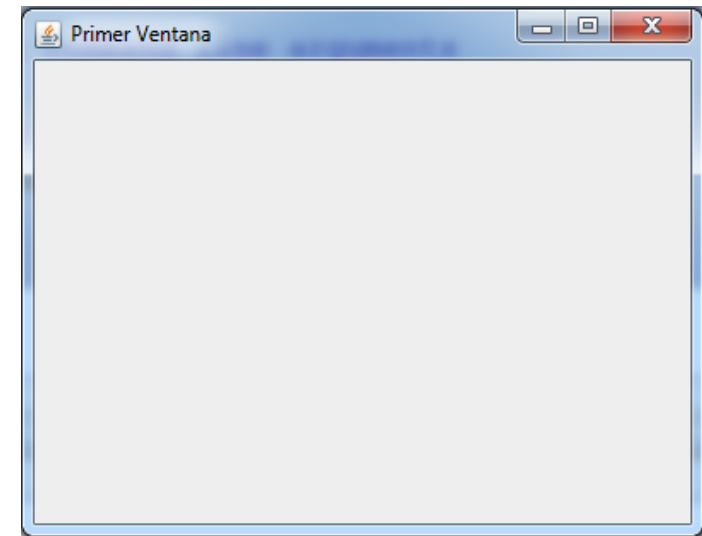
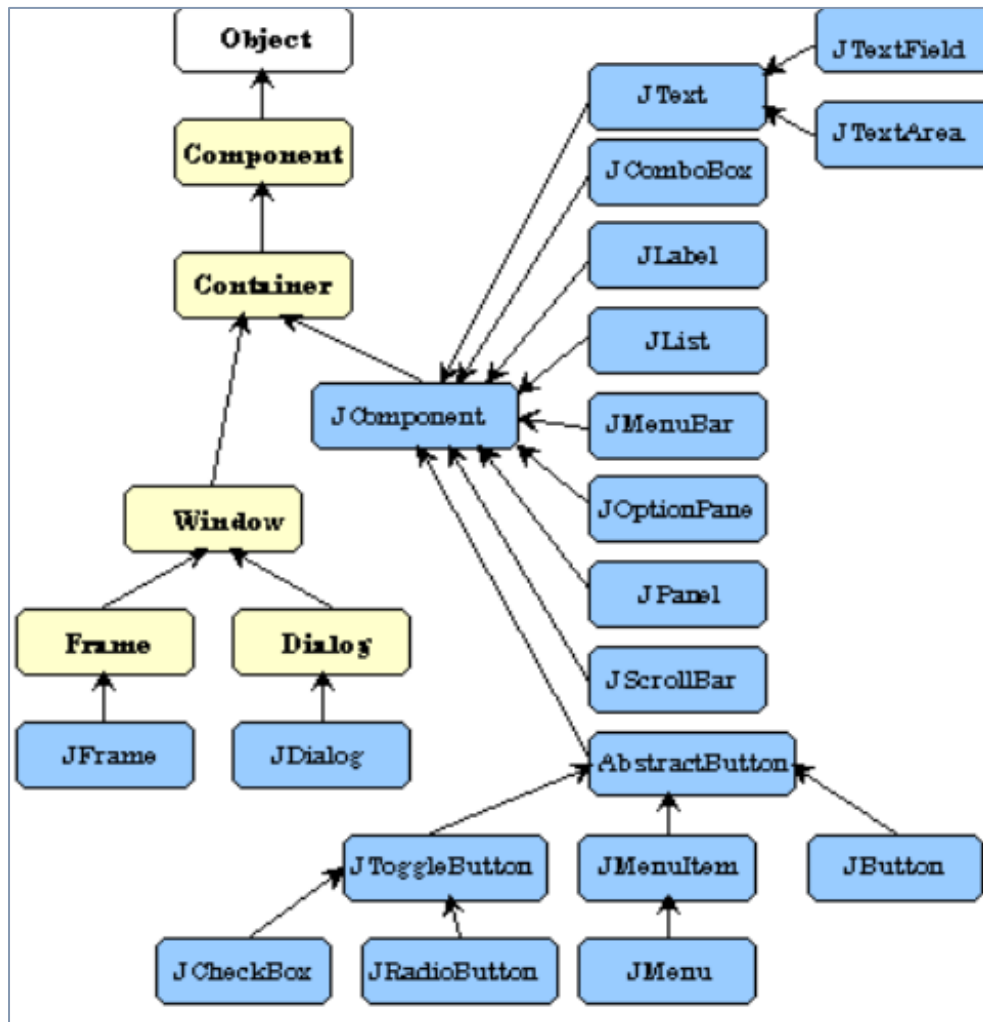
Date fechaHoy = new Date();

System.out.println(fechaHoy.toString());

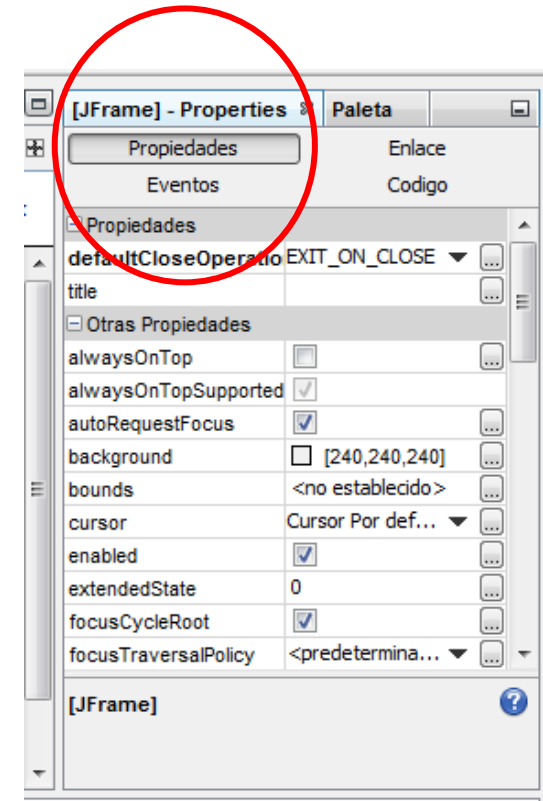
**SimpleDateFormat** sdf = new **SimpleDateFormat**("dd/MM/yyyy");

System.out.println(sdf.format(fechaHoy));

## 7.6 Interfaces gráficas con Swing



## 7.6 Interfaces gráficas con Swing



Todos los componentes heredan de **JComponent** >> **Container**, lo que los hace a su vez **contenedores** de otros objetos.

## Caso práctico



Vamos a NetBeans  
de nuevo!

# Caso práctico



Vamos a desarrollar una baraja de cartas españolas orientado a objetos.

Una carta tiene un número entre 1 y 12 (el 8 y el 9 no los incluimos), y un palo (espadas, bastos, oros y copas)

La baraja estará compuesta por un conjunto de cartas, 40 exactamente.

Las operaciones que podrá realizar la baraja son:

- **barajar**: cambia de posición todas las cartas aleatoriamente
- **siguienteCarta**: devuelve la siguiente carta que está en la baraja, cuando no haya más o se haya llegado al final, se indica al usuario que no hay más cartas.
- **cartasDisponibles**: indica el número de cartas que aún puede repartir
- **cartasMonton**: mostramos aquellas cartas que ya han salido, si no ha salido ninguna indicárselo al usuario
- **mostrarBaraja**: muestra todas las cartas hasta el final. Es decir, si se saca una carta y luego se llama al método, este no mostrara esa primera carta.

Diseñar además una ventana que permita mostrar el estado de la baraja, mezclarla y visualizar en una caja de texto la siguiente carta. En todo momento es posible conocer la cantidad de cartas disponibles en el montón y disponibles para sacar.