```kotlin
package com.example.caracteristicaskotlin

// AQUI VAMOS A VER LAS CARACTERISTICAS DE KOTLIN

// Creamos la funcion main
fun main(){ // No necesita nada mas para declararla

    //Tenemos dos opciones de declarar la variables: AÑADIENDO DE QUE TIPO SON O NO

    var nombre : String = "Maria" // Seria una variable mutable a la que puedo cambiar el valor
    nombre = "Alejandro" //Aqui le hemos reasignado el valor
    println(nombre) // Cuando lo imprimimos podemos ver que el valor que nos muestra es el ultimo
    var edad = 27

    val nombrePrincipal = "Maria" // Esta variable es inmutable (nunca cambiara su valor)
    // nombrePrincipal = "Lola" nos da error

    println(nombrePrincipal + " " +  nombre)

    // Si yo estoy definiendo variables, pero una de inicio, quiero que sea de valor nulo:

    var correo : String? = null // En este caso estamos diciendo que es un posible nulo, siempre poner el tipo de variable con la ?

    var tarea1 = Tarea("AD", "Resumenes Tema 1")
    tarea1.mostrarDatos()

    var tarea2 = Tarea.TareaImagen("PSP","Videos de clase","Imagen Kotlin") //Siempre poniendo la clase padre antes de la nueva clase
    tarea2.mostrarDatos()

    //Crearemos tambien un objeto tareatexto

    var tarea3 = Tarea.TareaTexto("SGE", "Sistemas de Gestion empresarial", "Inicio del tema 1")
    tarea3.mostrarDatos()

    // Voy a hacer un array para que recorra todas las tareas
```

```
    val conjuntoTareas : Array<Tarea> =
arrayOf(tarea1,tarea2,tarea3) //Añadimos los objetos tarea
creados
    for (i in conjuntoTareas){ //Recorremos con un for each
        i.mostrarDatos() //Pedimos que nos muestre los datos
    }
    tarea3.funcionLambdaTexto("Parametro texto") // aqui
invocamos la funcion lambda y lo que le añadimos es el
parametro

    val conjuntoTareasLista : ArrayList<Tarea> =
ArrayList() //Si queremos un array list lo haremos de la
siguiente manera
    conjuntoTareasLista.add(tarea2) // Aqui añadimos los
elementos que queremos recorrer
    conjuntoTareasLista.add(tarea3)
    conjuntoTareasLista.add(tarea1)
    //Para recorrerlos haremos lo siguiente
    //Se podria hacer perfectamente con un for, pero vamos a
utilizar las funciones Lambda
    conjuntoTareasLista.forEachIndexed { index, tarea ->
println("La tarea en la posicion $index tiene como titulo $
{tarea.getTitulo()}") }
    //Este for each indexed hace que me diga por ejemplo el
titulo de las tareas en cada indice


    // Ahora de esta lista quiero filtrar y que solo me
enseñe la que contenga el titulo SGE
    conjuntoTareasLista.filter { it.getTitulo() ==
"SGE" }.forEach { it.mostrarDatos() }

    // En el caso del find importante poner la interrogacion
porque en caso que el valor sea nulo no lo mostrara
    conjuntoTareasLista.find {it .getDescripcion() == "Videos
de clase"}?.mostrarDatos()

}
//Podemos trabajar con clases y esta clase la meteremos en el
main
open class Tarea{
    private var titulo :String
    private var descripcion :String
```

```
    //Crearemos el constructor de la clase
    constructor(titulo :String, descripcion:String){
        this.titulo = titulo;
        this.descripcion = descripcion;


    }
    fun getTitulo():String { // como las variables las hemos
hecho privadas, tenemos que añadir los getters
        return this.titulo
    }



    fun getDescripcion():String{
        return this.descripcion
    }

    fun setTitulo(titulo: String){ // como las variables las
hemos hecho privadas, tenemos que añadir los setters
        this.titulo = titulo
    }

    fun setDescripcion(descripcion: String){
        this.descripcion = descripcion
    }

    //Vamos a crear una funcion mostrar datos por ejemplo
    open fun mostrarDatos(){
        println("Titulo: $titulo")
        println("Descripcion: $descripcion")
    }
class TareaImagen (titulo: String, descripcion: String,
private var imagen: String) : Tarea(titulo,descripcion) {//
Otra forma de declarar las var dentro de una clase
    // Aqui estamos diciendo que la clase TareaImagen
implementa de la clase tarea, ya que a parte
    //de las dos variables de Tarea, añade la variable imagen
    //Hay que tener en cuenta que para que pueda implementar,
la clase Tarea debe ser --> open class Tarea

    // Haremos el get y set de la nueva variable privada

    fun getImagen(): String {
        return this.imagen
```

```
    }

    fun setImagen(imagen: String){
        this.imagen = imagen
    }

    //Si tambien quiero implementar el metodo mostrar datos,
tengo que mostrarlo como open en la clase donde esta
declarado

    override fun mostrarDatos() { //Siempre con override para
sobrescribir
        super.mostrarDatos() // esto marca lo que hacia en la
clase Tarea
        println("Imagen: $imagen")// A parte le pedimos que
imprima la otra variable
    }
}
class TareaTexto( titulo: String, descripcion: String,
private var texto : String): Tarea(titulo, descripcion){

    var funcionLambdaTexto: (String) -> Unit = {// significa
que la variable es una funcion de tipo String que va a
devolver un Unit
     parametro: String -> println("El texto de la nota es
$texto y el parametro es $parametro")
        //Aqui lo que le paso es un parametro que es un
String y devuelve el impreso por pantalla
    }

    override fun mostrarDatos() {
        super.mostrarDatos()
        println("Texto: $texto")
    }
    fun getTexto(): String{
        return this.texto
    }
    fun setTexto(texto :String){
        this.texto = texto
    }
}
}
```