



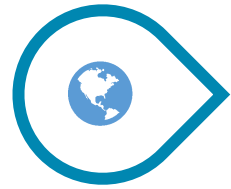
USO DE LIBRERÍAS PARA IMÁGENES PARA ANDROID

DESARROLLO DE APLICACIONES MÓVILES AVANZADAS

LUIS HUMBERTO RIVAS RODRÍGUEZ

INGENIERO EN SISTEMAS INFORMÁTICOS
Y MÁSTER EN DIRECCIÓN ESTRATÉGICA DE EMPRESAS.

Agenda



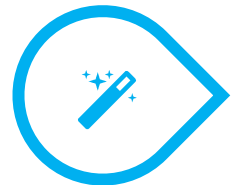
1 Generalidades

Conceptos

Librerías

Funcionalidades

2



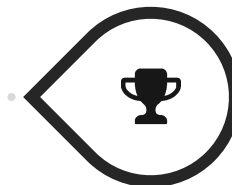
3 Desafío de Clases

Ejemplo Práctico

Cierre

Valoraciones Finales

4



RESULTADOS DE APRENDIZAJE

Competencias a desarrollar al finalizar la sesión.



INTERPRETA

Los componentes para usar imágenes dentro de una interfaz gráfica en Android.



COMPRENDE

La relación que tienen las librerías y su importancia en el desarrollo de aplicaciones basadas en Android.



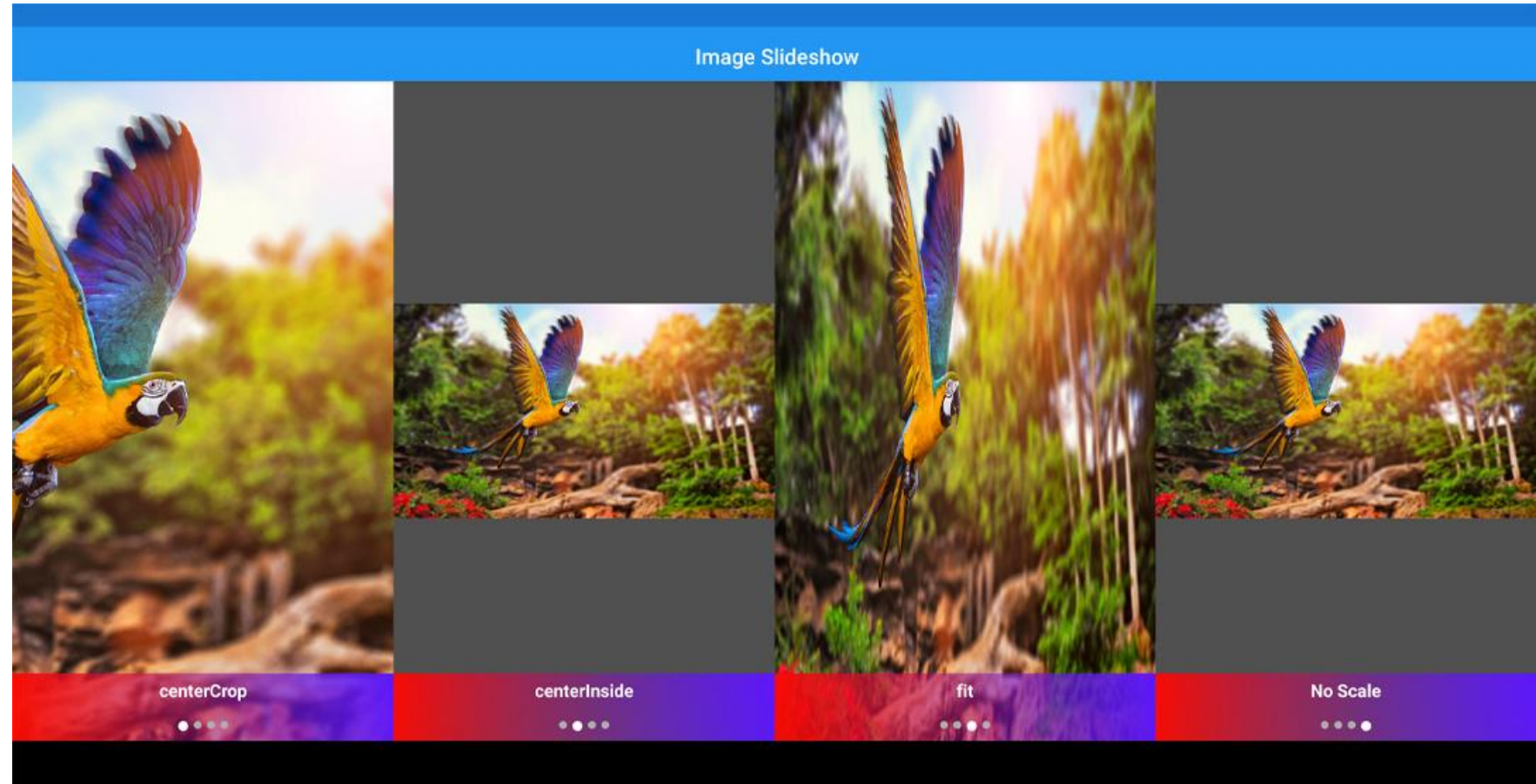
CONSTRUYE

Aplicaciones Android haciendo uso de los componentes avanzados como son las librerías en Android.

CONTEXTO DE USO

Requerimientos principales

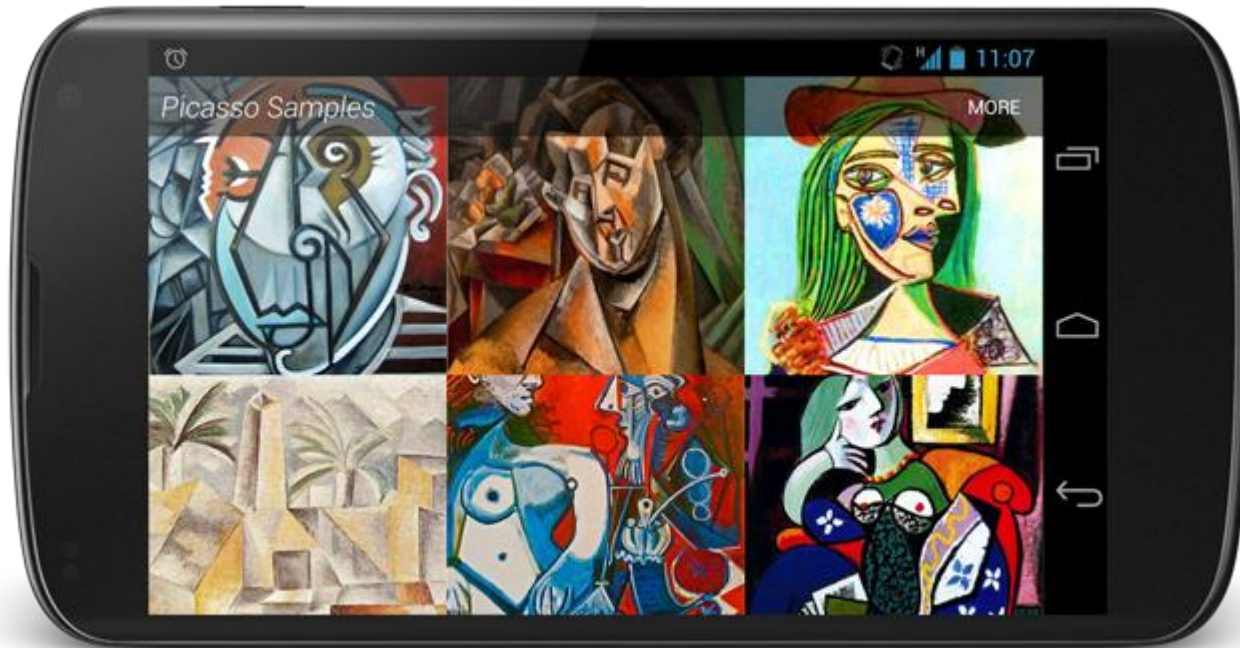
- En ocasiones se requiere hacer uso de recursos multimedia como las imágenes, sin embargo para un entorno Android podemos acceder a dichos recursos ya sea de manera local (**carpeta drawable**) o de manera remota (a través **de una URL** de imagen).



Fuente: <https://github.com/denzcoskun/ImageSlideshow>

LIBRERÍA PICASSO

COMPONENTES



- Las imágenes agregan un contexto muy necesario y un toque visual a las aplicaciones de Android, por lo que Picasso permite cargar imágenes sin problemas en nuestra aplicación, en una sola línea de código.

Fuente: <https://square.github.io/picasso/#features>

Repositorio: <https://github.com/square/picasso>

GRADLE

CONTEXTUALIZACIÓN

Para implementarlo bastaría agregar en nuestro Gradle dicha librería de la siguiente manera.

1. Primero agrega esto en `libs.versions.toml`

[versions]

`picasso = "2.71828"`

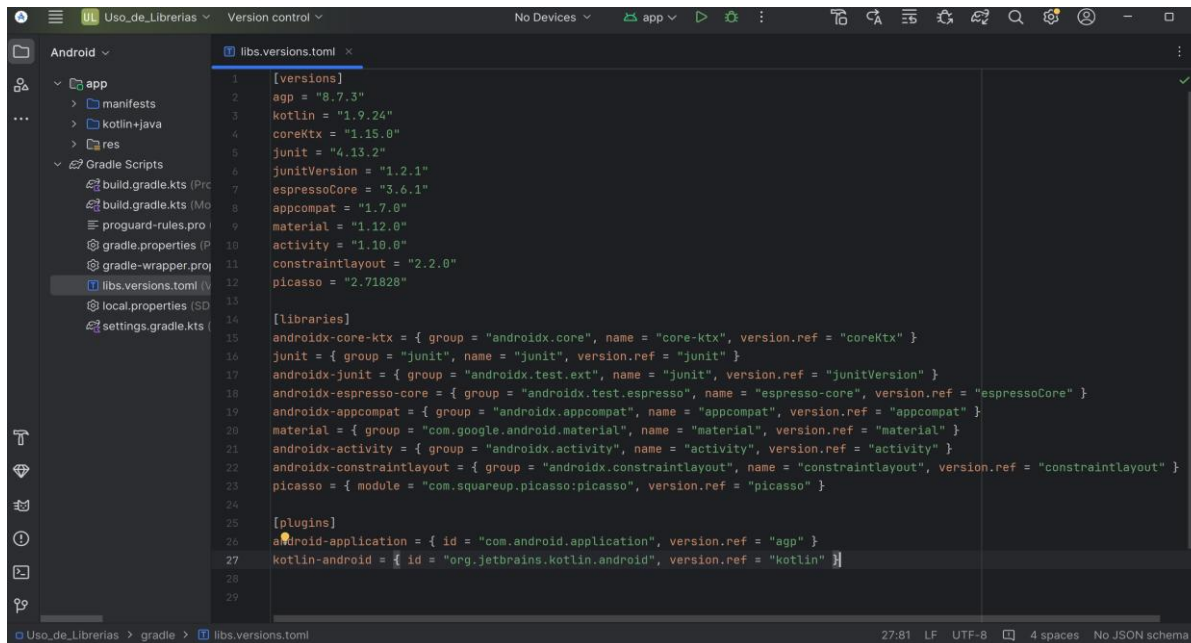
[libraries]

`picasso = { module =
"com.squareup.picasso:picasso", version.ref =
"picasso" }`

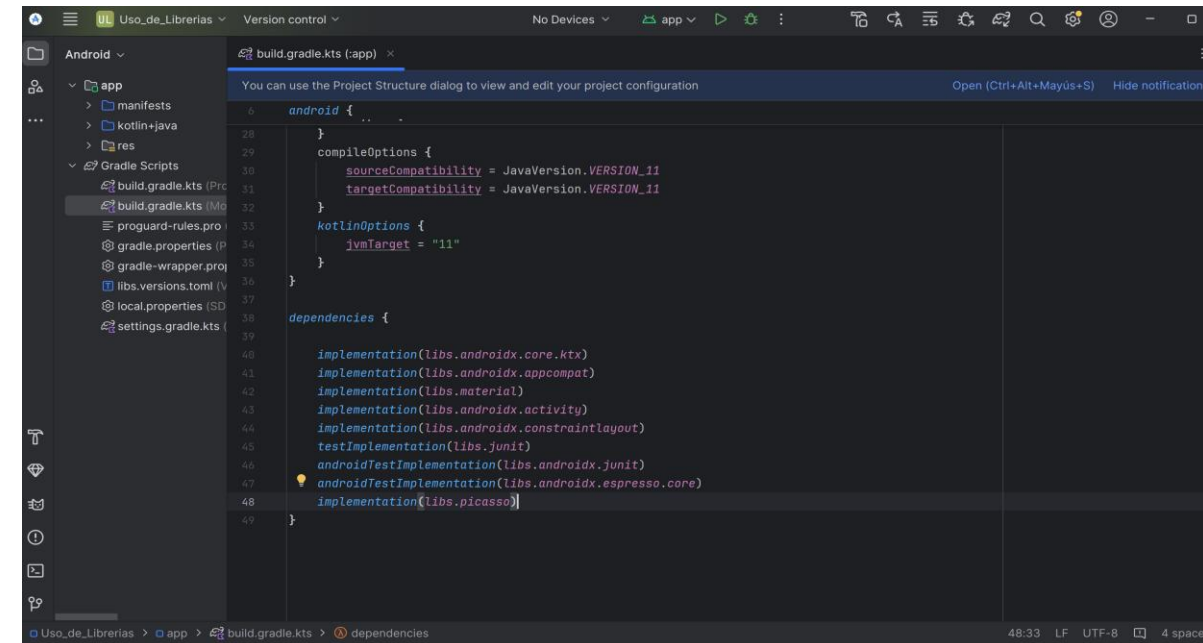
2. Luego agrega esto en `build.gradle.kts`

[dependencies]

`implementation(libs.picasso)`



```
1 [versions]
2 app = "8.7.3"
3 kotlin = "1.9.24"
4 coreKtx = "1.15.0"
5 junit = "4.13.2"
6 junitVersion = "1.2.1"
7 espressoCore = "3.6.1"
8 appcompat = "1.7.0"
9 material = "1.12.0"
10 activity = "1.10.0"
11 constraintlayout = "2.2.0"
12 picasso = "2.71828"
13
14 [libraries]
15 androidx-core-ktx = { group = "androidx.core", name = "core-ktx", version.ref = "coreKtx" }
16 junit = { group = "junit", name = "junit", version.ref = "junit" }
17 androidx-junit = { group = "androidx.test.ext", name = "junit", version.ref = "junitVersion" }
18 androidx-espresso-core = { group = "androidx.test.espresso", name = "espresso-core", version.ref = "espressoCore" }
19 androidx-appcompat = { group = "androidx.appcompat", name = "appcompat", version.ref = "appcompat" }
20 material = { group = "com.google.android.material", name = "material", version.ref = "material" }
21 androidx-activity = { group = "androidx.activity", name = "activity", version.ref = "activity" }
22 androidx-constraintlayout = { group = "androidx.constraintlayout", name = "constraintlayout", version.ref = "constraintlayout" }
23 picasso = { module = "com.squareup.picasso:picasso", version.ref = "picasso" }
24
25 [plugins]
26 android-application = { id = "com.android.application", version.ref = "app" }
27 kotlin-android = { id = "org.jetbrains.kotlin.android", version.ref = "kotlin" }
```



```
28 }
29 compileOptions {
30     sourceCompatibility = JavaVersion.VERSION_11
31     targetCompatibility = JavaVersion.VERSION_11
32 }
33 kotlinOptions {
34     jvmTarget = "11"
35 }
36
37 dependencies {
38
39     implementation(libs.androidx.core.ktx)
40     implementation(libs.androidx.appcompat)
41     implementation(libs.material)
42     implementation(libs.androidx.activity)
43     implementation(libs.androidx.constraintlayout)
44     testImplementation(libs.junit)
45     androidTestImplementation(libs.androidx.junit)
46     androidTestImplementation(libs.androidx.espresso.core)
47     implementation(libs.picasso)
48 }
49
```

USO DE PICASSO

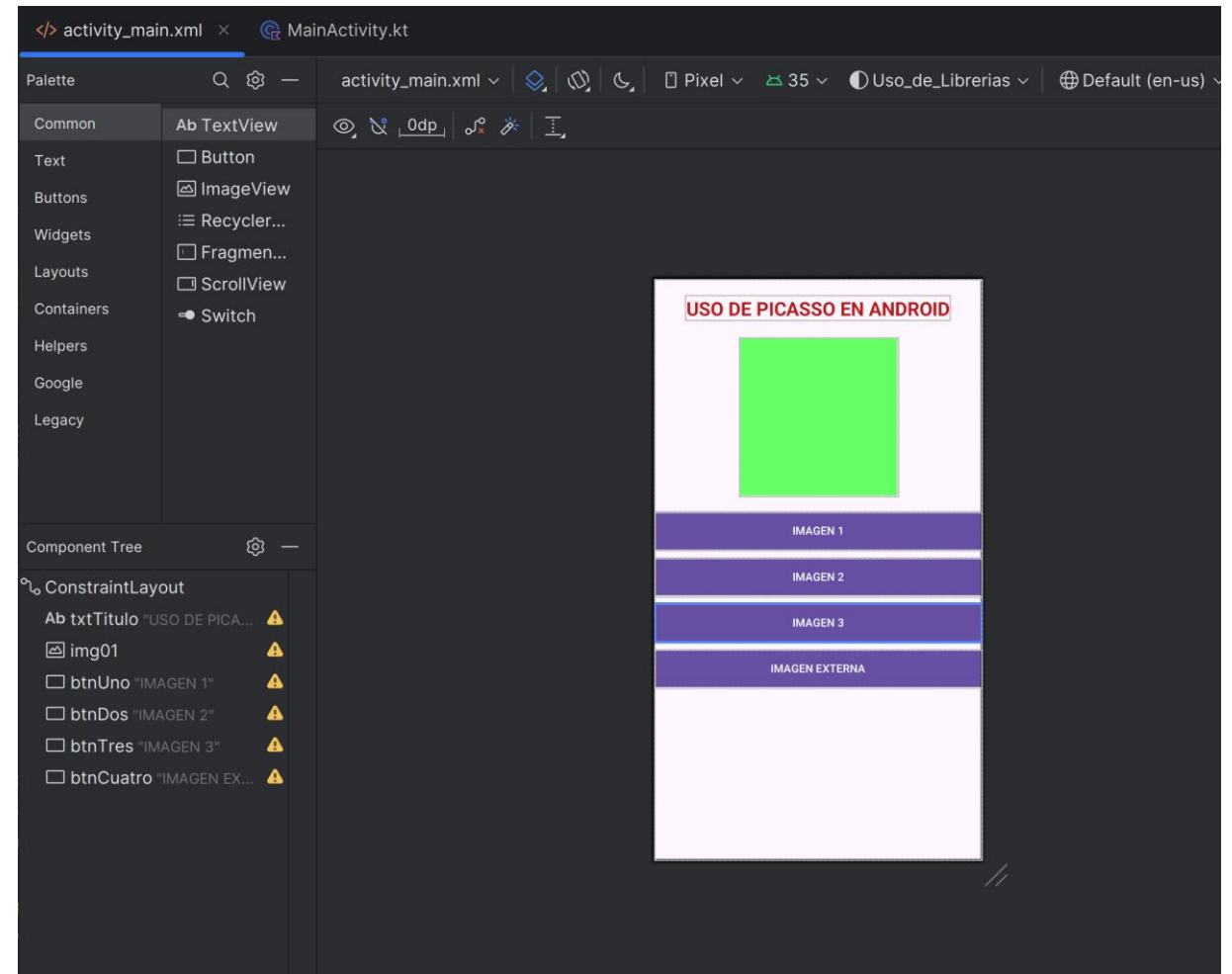
COMPONENTES

- Se requiere de un control ImageView sobre el cuál se pintará el resultado de nuestra imagen.

```
Picasso.get()  
    .load(R.drawable.imagen1)  
    .into(img01);
```

```
Picasso.get()  
    .load(R.drawable.imagen1)  
    .resize(512,512)  
    .centerCrop()  
    .into(img01);
```

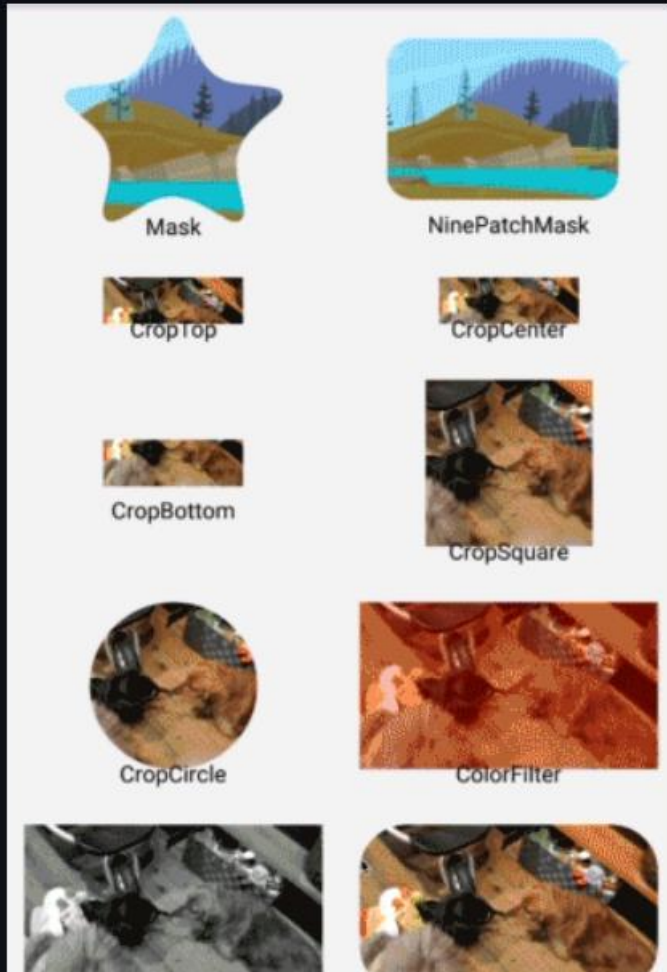
```
Picasso.get()  
    .load("http://url/imagen.jpg")  
    .into(img01);
```



EXTRAS

PERSONALIZANDO EL CONTROL PICASSO

Transformations



- Si queremos generar imágenes con efectos de una forma sencilla y rápida, se puede hacer uso de **Picasso-transformation**.

<https://github.com/wasabeef/picasso-transformations>



- Tomando como referencia los contenidos visto en la sesión, elabore una aplicación que permita generar un conjunto de imágenes a visualizar.
- Utilice la librería Picasso para dicha actividad.



VALORACIONES FINALES

Comentarios sobre el tema.

1

Picasso

Es la librería encargada de gestionar los recursos de imágenes en una aplicación android.

2

Transformations

Es un mecanismo para generar efectos de manera rápida, aprovechando la librería Picasso.

3

Creativo

Saber distribuir y seleccionar los diferentes métodos y propiedades de una clase, permitirá al programador trabajar de forma rápida y eficiente



¿PREGUNTAS?

DESARROLLO DE APLICACIONES MÓVILES AVANZADAS

LUIS HUMBERTO RIVAS RODRÍGUEZ

INGENIERO EN SISTEMAS INFORMÁTICOS
Y MÁSTER EN DIRECCIÓN ESTRATÉGICA DE EMPRESAS.