

Universidad de Oviedo



Escuela de
Ingeniería
Informática



ARQUITECTURA
DEL SOFTWARE

Arquitectura del Software

Lab. 02

Diagramas UML

PlantUML

Introducción a documentación y arc42

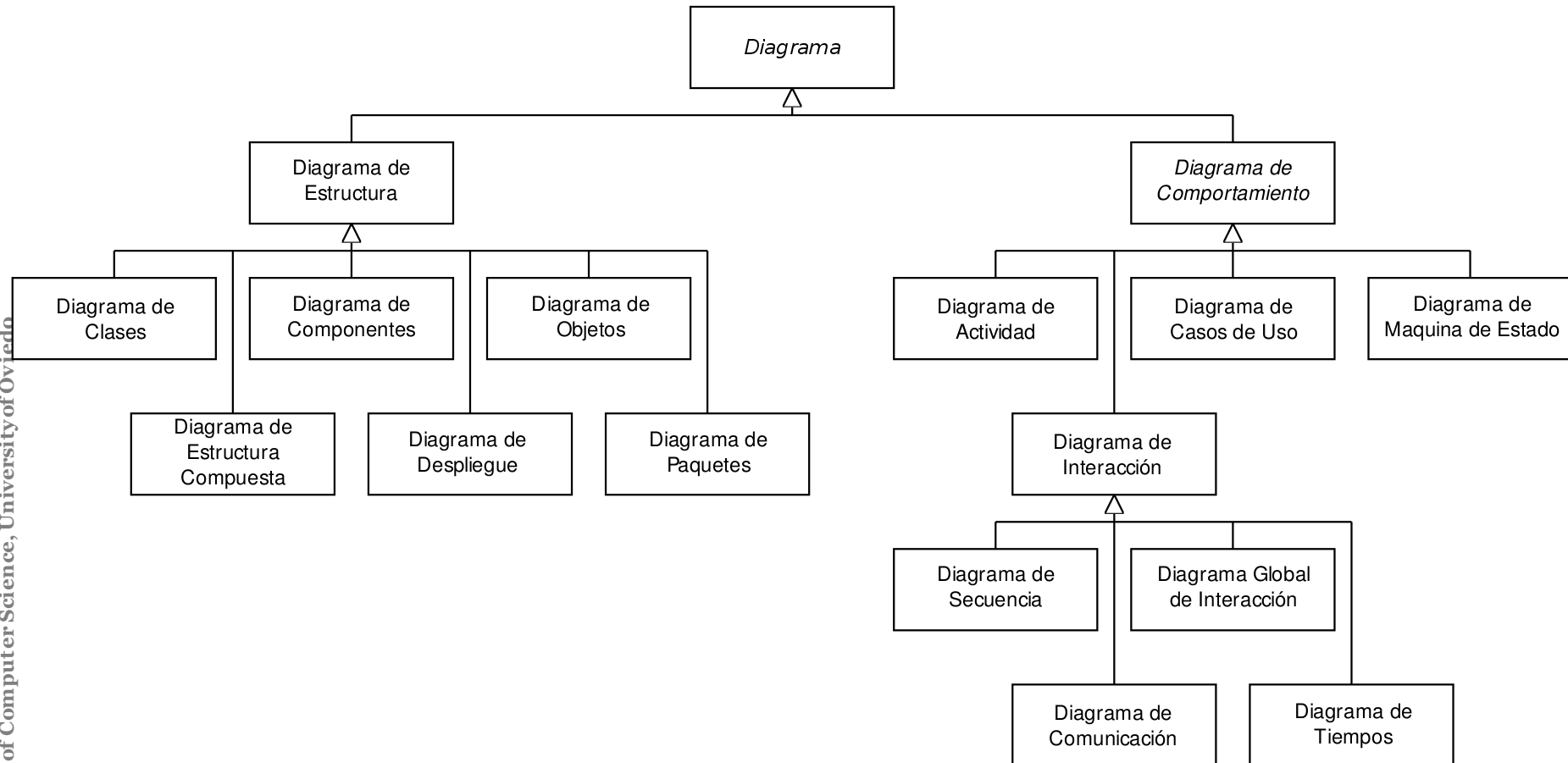
2022-23

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Cristian Augusto Alonso

UML

- **Unified Modeling Language**
 - Antes de UML había varias propuestas
 - Notación UML los unifica
 - Propuesta por OMG (Object Management Group)
 - Versión actual: UML 2.5.1 (2017)
- **Modelo = abstracción de un problema**
 - Puede tener varios diagramas diferentes
 - Diagrama = representación gráfica parcial de un modelo
- **OCL = Object Constraint Language**
 - Restricciones entre objetos usando lenguaje formal

14 tipos de diagramas UML

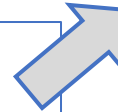


Diagramas de clase

- Modela la parte estática del proyecto, sin tener en cuenta la situación del sistema en un tiempo.
- Explica las relaciones que hay entre las distintas clases.
- Arc42:8-Concepts

Visibilidad:

- - Privado
- ~ Paquete
- # Protegido
- + Público

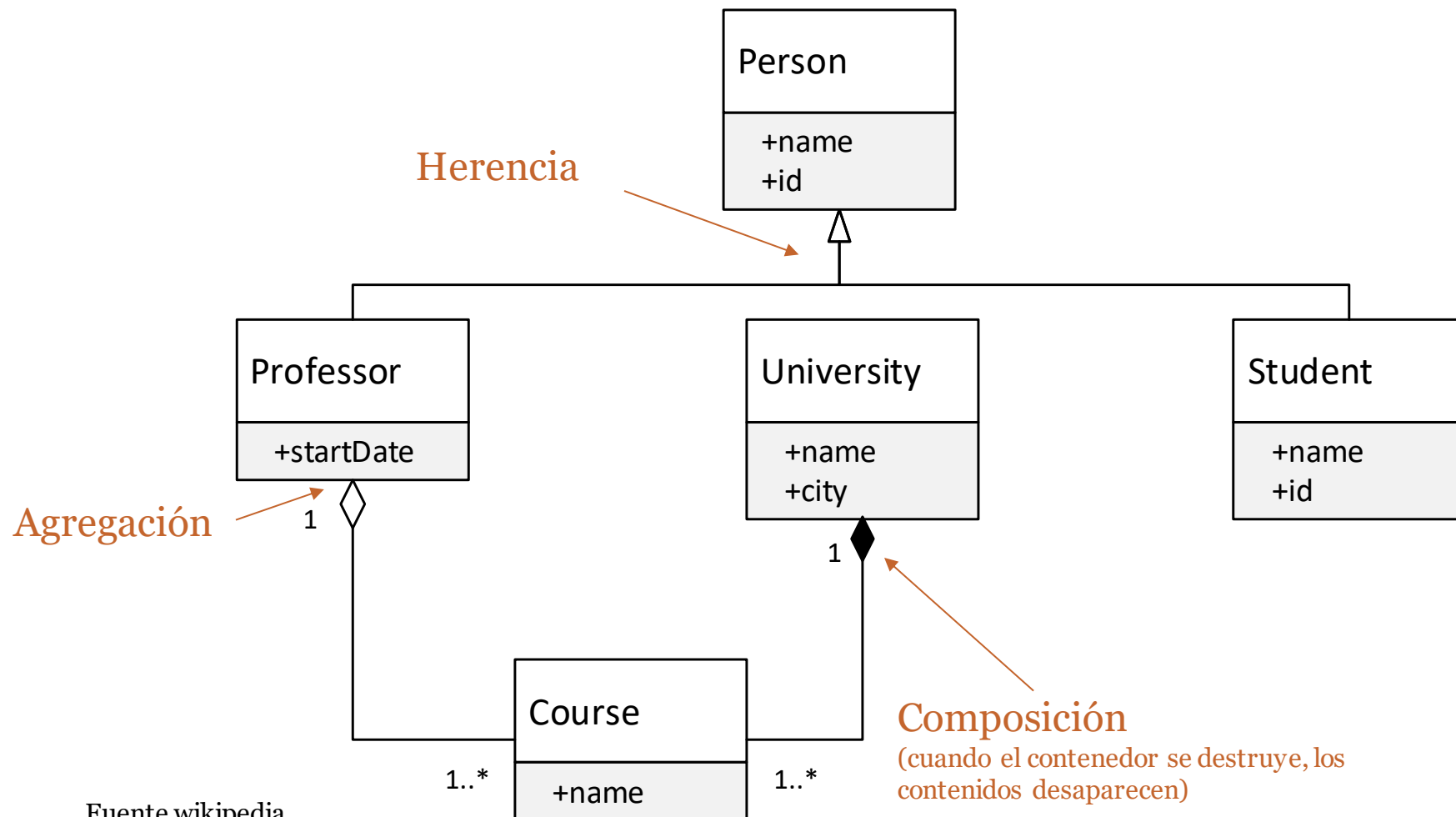


MusicaManager

+conf:Configuration
-playList:List<Song>
-activeSong:integer

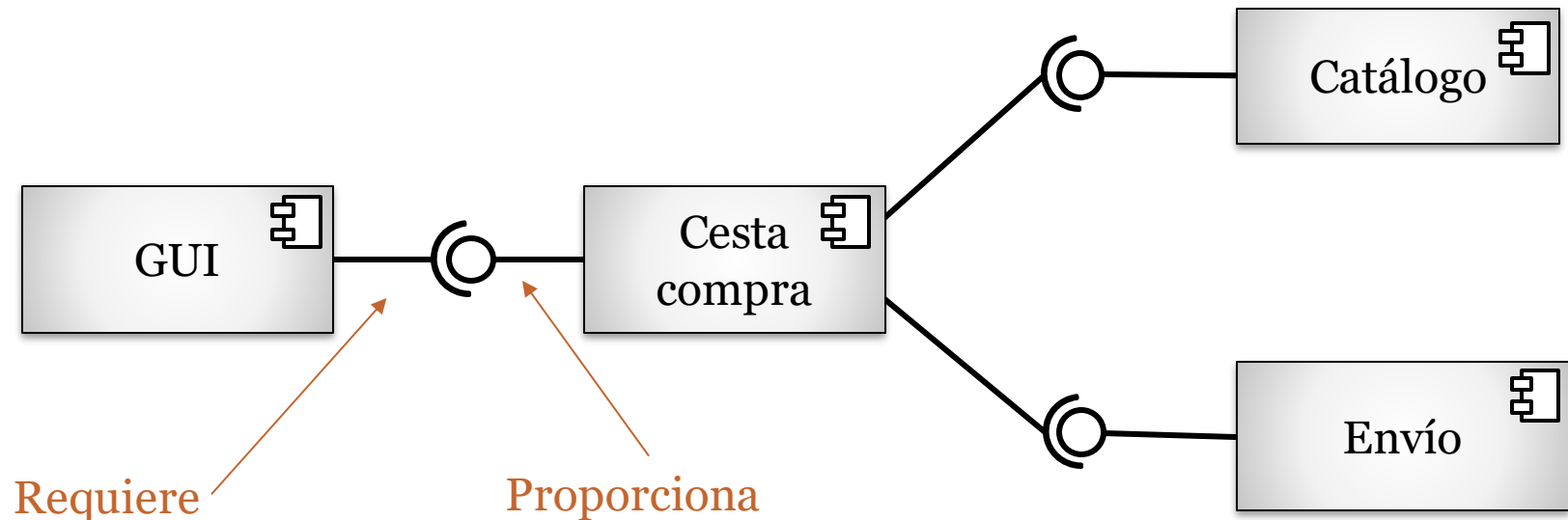
+randomPlay()
+añadirCancion(Song)
+consultarConf():Configuracion
..

Ejemplo



Fuente wikipedia

Diagrama de componentes



- Representa la relación estructural de los componentes de un sistema de software
- Sistemas complejos que tienen muchos componentes

Diagrama de secuencia

- Modela la comunicación entre los objetos de un sistema en un determinado momento
- Los objetos pueden enviarse dos tipos de mensajes: síncronos y asíncronos
- Arc42:6-RuntimeView

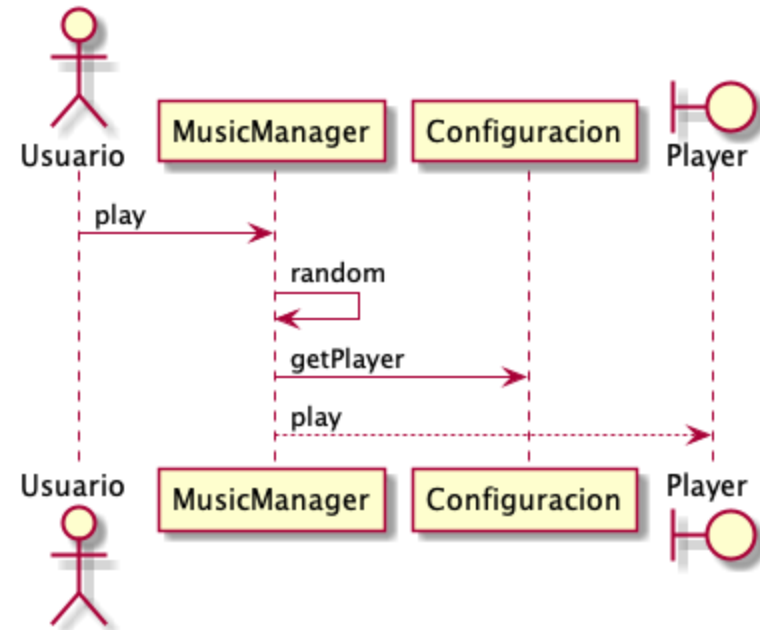
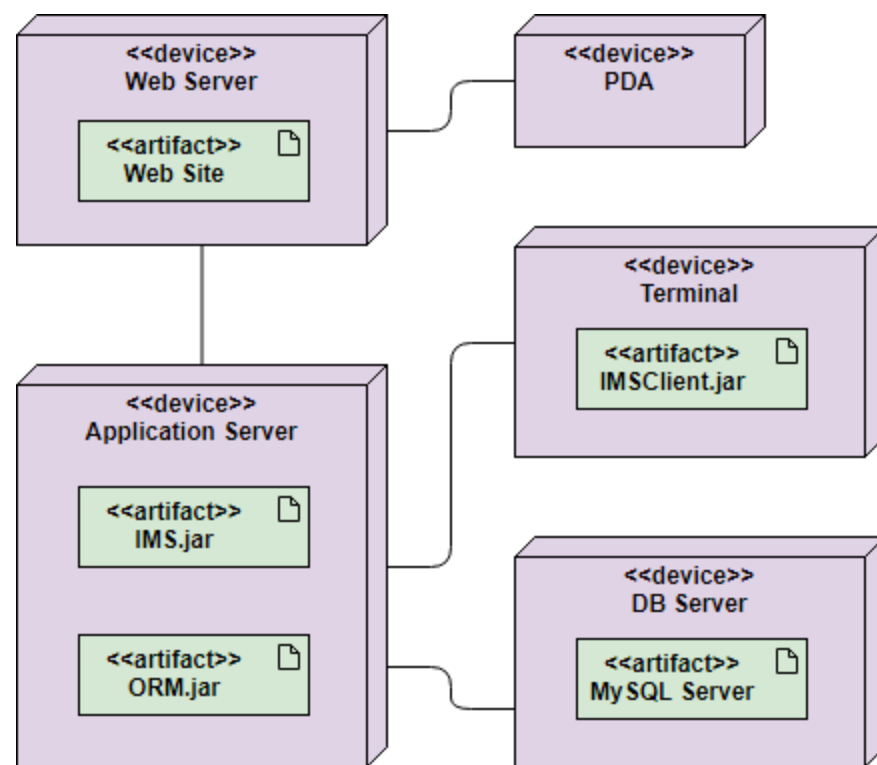


Diagrama de despliegue

- Representa la localización final de los componentes de la aplicación
- Elementos : Nodos , Componentes, relaciones
- Arc42: 07.DeploymentView



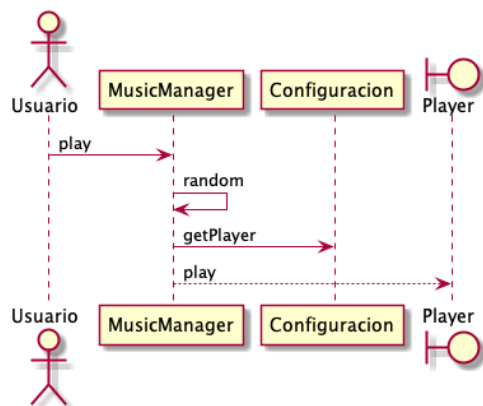
Herramientas Textuales

• PlantUML

```

@startuml component
actor Usuario
participant MusicManager
participant Configuracion
boundary Player
Usuario -> MusicManager: play
MusicManager -> MusicManager: random
MusicManager -> Configuracion : getPlayer
MusicManager --> Player : play
@enduml

```

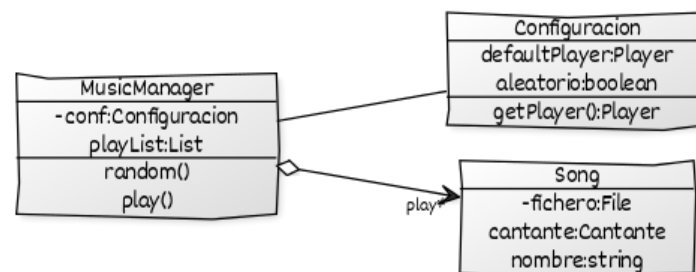


YUML

```

// Cool Class Diagram
[MusicManager|-conf:Configuracion;
playList:List |random();play()]
[MusicManager]<>-play*>[Song|-
fichero:File;cantante:Cantante;nombre
:string]
[MusicManager]-
[Configuracion|defaultPlayer:Player;a
leatorio:boolean|getPlayer():Player]
]

```



Herramientas de dibujo

PowerPoint

Visio (Microsoft)

UMLet (<https://www.umlet.com/>)

Herramientas CASE

EnterpriseArchitect

- Solo para Windows
- Entiende todo tipo de diseño
- Ingeniería Inversa con Java/C++
- Conecta con Oracle modelos datos relacionales
- Plantillas editables para Word, HTML

MagicDraw

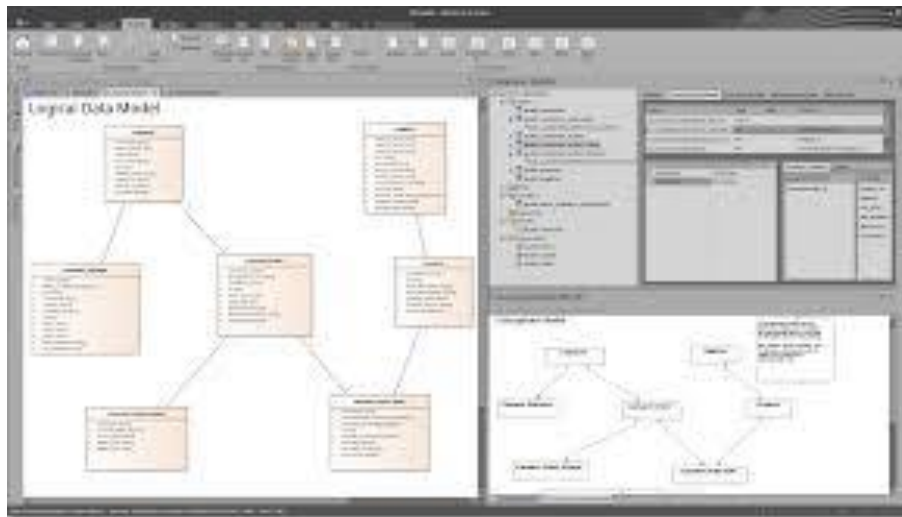
- Para todo sistema con Java
- Diagramas UML
- Ingeniería Inversa Java , C++

Visual Paradigm

- Comercial (Licencia estudiantes)

Modelio

- Código abierto
- Java based
- Ingeniería Inversa



Dibujando la arquitectura

Vídeo con pautas para diagramas

<https://www.youtube.com/watch?v=wgpSdpny-0c>

Checklist utilizado en C4

<https://c4model.com/assets/software-architecture-diagram-review-checklist.pdf>

Plantillas arc42

<https://arc42.org/>

DeDe ya sigue la plantilla:

https://arquisoft.github.io/dede_o/

Generación de documentación en local:

> `npm run docs`

Generación de documentación en repositorio:

`git push ...`



Despliegue de la documentación

Utilizaremos GitHub Pages para desplegar la documentación

GitHub Pages permite a los usuarios publicar un sitio web sencillo directamente en GitHub.

El sitio web de la documentación se enviará a la rama **gh-pages**.

Los archivos asciidoc se enviarán a la rama develop del repositorio (Manualmente)

El package.json de **gh-pages** se encarga de enviar la documentación del sitio web a gh-pages

Todo esto está automatizado ejecutando el siguiente comando:

```
> npm run deploy
```



Fin