



# Trabajo de Fin de Grado

Leopoldo Ripoll Vitini

Junio de 2019

Consultor: Albert Grau Perisé

Responsable de la assignatura: Santi Caballé Llobet

# Metodología Seguida

- ▶ **Descripción del sistema:** Modelo de referencia para el procesamiento abierto y distribuido: **RM-ODP**
- ▶ **Basado en puntos de vista** (Empresa, información, computación, ingeniería y tecnología )
- ▶ **Organización, planificación y definición del proyecto especificado en P.V.**  
Empresa

# P.V. Empresa: Objetivos

GP  
PDP

- ▶ **Mostrar el proceso de construcción de un producto de software desde cero**
- ▶ **Reflejar conocimientos adquiridos durante el grado:**
  - ▶ Programación Web (PW)
  - ▶ Diseño Gráfico (DG)
  - ▶ Ingeniería de Componentes de software (EPCSD)
  - ▶ Proyecto de desarrollo de software / Gestión de proyectos (PDP / GP)
  - ▶ Ingeniería de requisitos (IR)
  - ▶ Interacción persona - ordenador (IPO)
  - ▶ Diseño de bases de datos (DBD)
- ▶ **Producir una aplicación web desde 0 que sea una red social para “futboleros”**

# P.V. Empresa: Análisis comparativo y justificación

- ▶ No existen redes sociales para “futboleros”
- ▶ Puntos de entrada interesantes:
  - ▶ Predicción de alineaciones
  - ▶ Encuestas
- ▶ Productos complementarios con gran masa social: Comunio / Futmondo
- ▶ Servicios Web encuestas

# P.V. Empresa: Descripción del proyecto

- ▶ Red Social para apasionados del futbol
- ▶ Título: Futboleros. Apasionados del fútbol
- ▶ Actores: Usuarios sin registrar / Usuarios / Administrador
- ▶ Secciones principales:
  - ▶ Alineaciones
  - ▶ Encuestas
  - ▶ Quiz
  - ▶ Foro

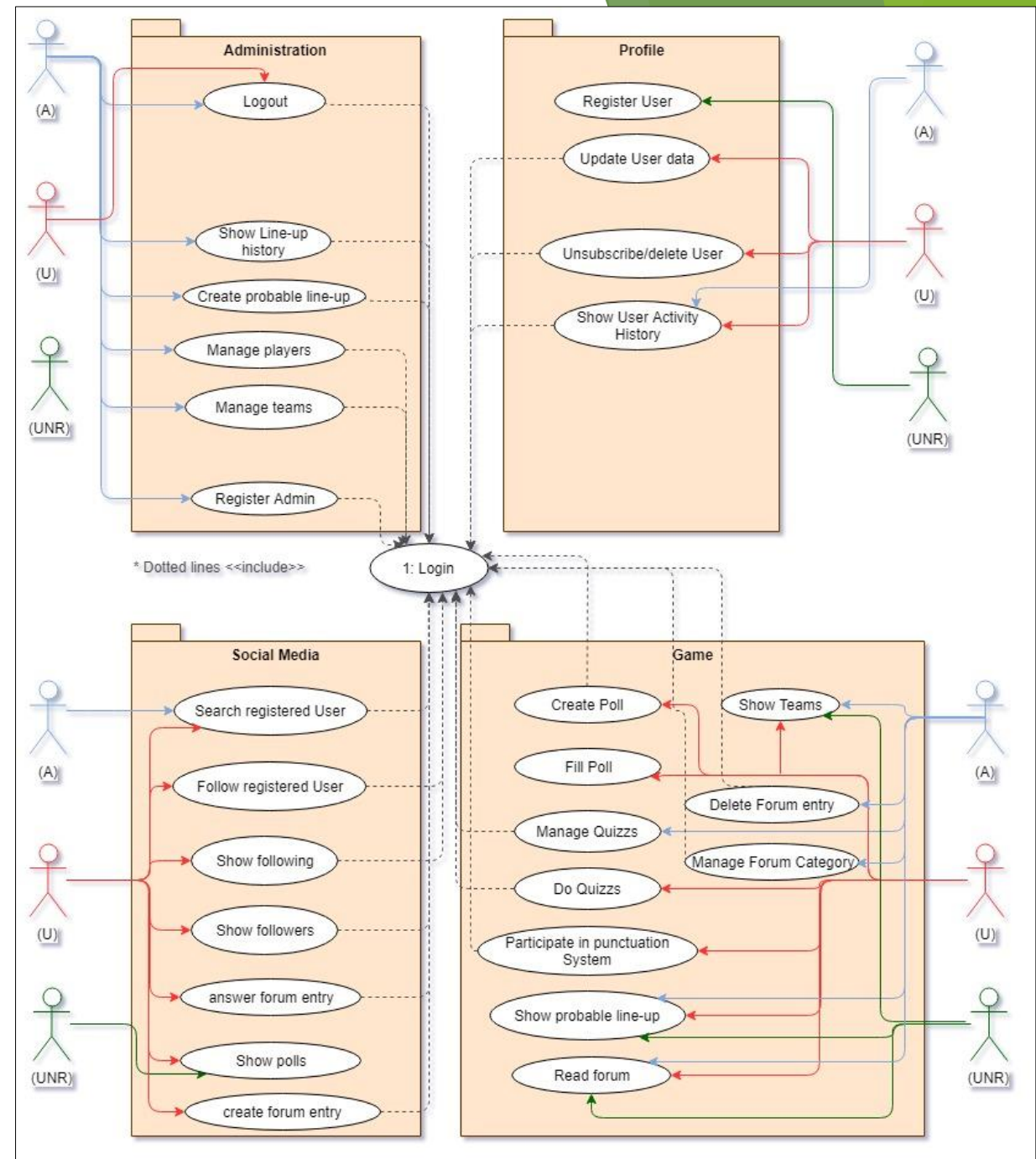


# P.V. Empresa: Requisitos

- ▶ No funcionales:
  - ▶ Responsivo, ágil, usable, flexible, mantenible, escalable
- ▶ Funcionales (ver memoria):
  - ▶ Usuarios no registrados
  - ▶ Usuarios Registrados
  - ▶ Administradores
  - ▶ Alineaciones
  - ▶ Encuestas
  - ▶ Quizz
  - ▶ ...

# P.V. Empresa: Casos de uso

- Especificación de casos de Uso
- Refinamiento de requisitos



# P.V. Empresa: Calidad y planificación



- ▶ Calidad:
  - ▶ Pruebas unitarias
  - ▶ Estudio de calidad del código con SonarQube
  - ▶ Pruebas de interacción con usuarios IPO
  
- ▶ Planificación
  - ▶ Diagrama Gantt
  - ▶ Adecuación a PECs



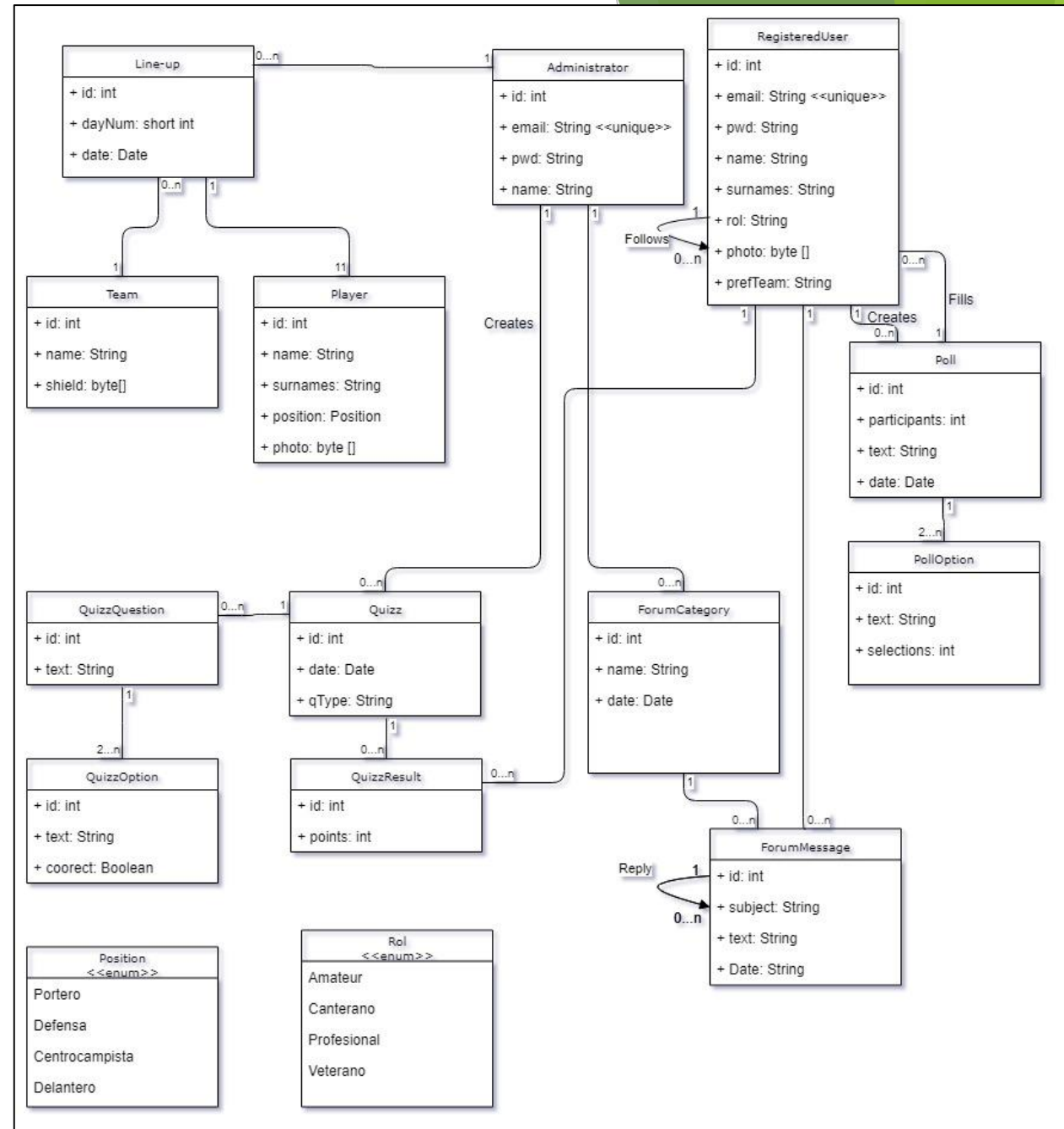


# P.V. de la información

PostgreSQL



DBD  
IR



# P.V. de la computación

- ▶ Arquitectura cliente - servidor
- ▶ Arquitectura de componentes distribuidos
- ▶ Arquitectura por capas

3r refinamiento:

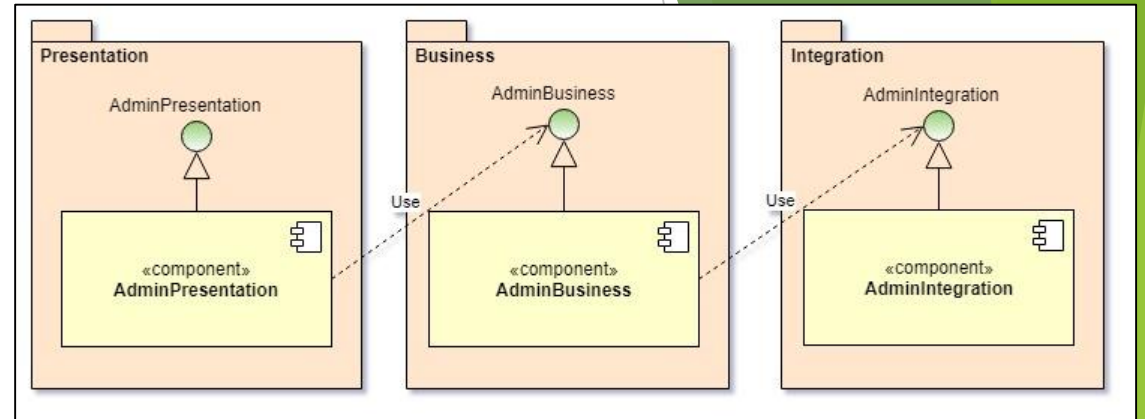
Patrones:

FrontController

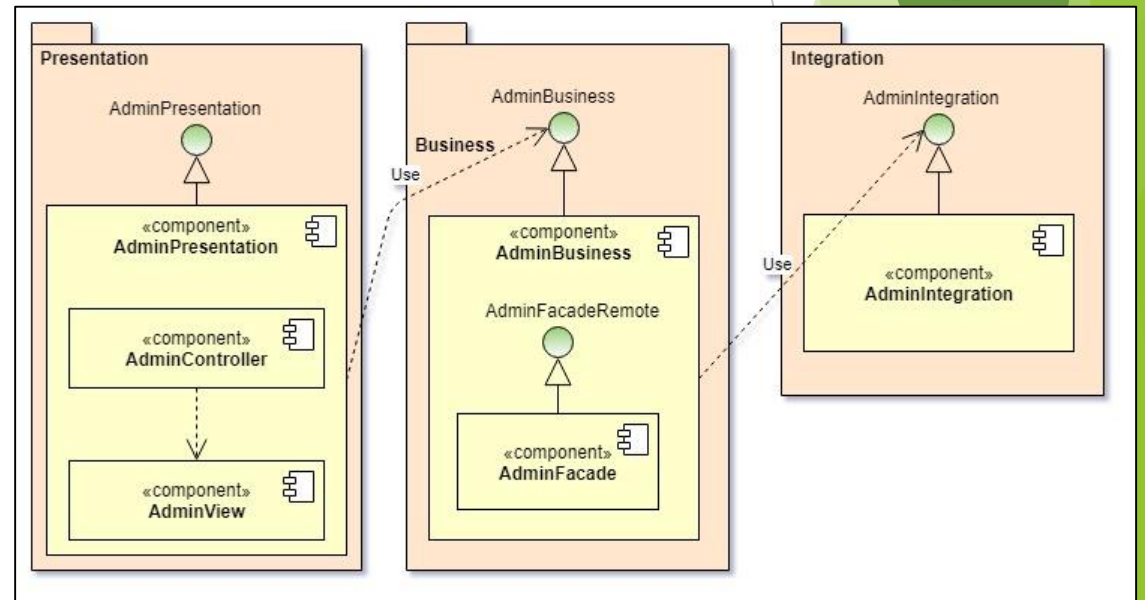
Command (Acciones)

EPCSD  
PDP

## 1r refinamiento (división por capas)



## 2º refinamiento (MVC + Patron fachada en negocio)



# P.V. de la computación: Diseño previo de vistas / prototipo



# P.V. de la ingeniería:

- ▶ Acceso por navegadores.
- ▶ Base de datos relacional
- ▶ Componentes de la capa de negocio se ejecutarán de manera concurrente en el servidor.
- ▶ Presentación y negocio en máquinas virtuales diferentes
- ▶ Negocio e integración en misma máquina virtual
- ▶ No se considera replicación de datos.

# P.V. de la tecnología



PostgreSQL



Enterprise Java Bean



# Comentarios de implementación

- ▶ Imágenes (escudos equipos, fotos jugadores, fotos usuarios) en base de datos
- ▶ *Beans* extienden clase genérica *BaseBusiness*
- ▶ Sistema de autenticación con filtros
- ▶ Uso de *Java Mail*
- ▶ Paquete *Game* más complejo de lo previsto
- ▶ Errores centralizados con `<error-page>` en `web.xml`
- ▶ Evitar *fetch-eager*
- ▶ *Jquery*
- ▶ *Templates* y `Html` semántico

# Calidad del código



GP  
PDP

**▼ Maintainability** ? Overview

Overview

Overall

Code Smells	310
Debt	4d 1h
Debt Ratio	1.6%
Rating	<b>A</b>
Effort to Reach A	0

**▼ Duplications** Overview

Overview

Overall

Density	4.5%
Duplicated Lines	320
Duplicated Blocks	12
Duplicated Files	11

**▼ Reliability** ? Overview

Overview

Overall

Bugs	4
Rating	<b>C</b>
Remediation Effort	18min

**▼ Security** ? Overview

Overview

Overall

Vulnerabilities	14
Rating	<b>E</b>
Remediation Effort	2h 40min

**▼ Complexity** ? Cyclomatic Complexi...

Cyclomatic Complexity 627

Cognitive Complexity 241

Example of SonarQube Scanner for Ant Usage > src

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Coverage	Duplications
src	4,174	4	14	310	0.0%	4.5%
ejb	773	2	1	53	0.0%	0.0%
jpa	915	0	0	23	0.0%	7.6%
managedbean	1,938	2	8	193	0.0%	4.8%
META-INF	27	0	0	0		0.0%
utils	521	0	5	41	0.0%	6.9%

**▼ Size**

Lines of Code	4,174
Lines	7,040
Statements	1,314
Functions	542
Classes	89
Files	91
Comment Lines	516
Comments (%)	11.0%







# Pruebas de usabilidad

- ▶ Definición de perfiles, dispositivos, plantillas
- ▶ Presenciales / online
- ▶ Resultados:
  - ▶ Incidencias críticas
  - ▶ Observaciones de usabilidad
  - ▶ Observaciones de diseño gráfico
  - ▶ Observaciones relacionadas con el navegador

IPO



# Conclusiones del proyecto



- ▶ **Planificación previa: El proceso de implementación se dirige mejor y resulta más efectivo**
- ▶ Alcance y priorización de secciones: trabajo de forma secuencial y consciencia de la limitación de tiempo
- ▶ Alteración de planificación: paquete *Game*
- ▶ **Previo a producción:** observaciones IPO, indicaciones SonarQube, y perfeccionar *responsive* tablas usuarios. (5 días)
- ▶ Línea de trabajo a **corto plazo:** sistema de puntuaciones, foro, “recuperar contraseña”, participación pública en encuestas y pruebas de usuario. (8 días)
- ▶ Línea de trabajo a **largo plazo:** “Valoración de jugadores”, “porra de la semana”, JAAS, optimización de tablas, integración Facebook/Twitter, pruebas de usuario. (18 días)