

Práctica 5: Módulos Odoo Controlador, Herencia y Web Controllers.

Sistemas de Gestión Empresarial

Cristian Fernández

29 de enero de 2026

Índice

1. Introducción	2
2. Actividad 01 – Modificación <i>EJ07-LigaFutbol</i>	3
2.1. Reglas de puntuación especiales	3
2.2. Botones para alterar los goles de los partidos	11
2.3. Web Controller para eliminar empates	17
2.4. Informe PDF por cada partido	21
2.5. Wizard para crear nuevos partidos	27
2.6. Vista Graph	28
3. Actividad 02 – Pruebas <i>EJ08-API-REST_Socio</i>	29
4. Actividad 03 – Bot Telegram API REST	30
5. Actividad 04 – Generación imágenes aleatorias Web Controller	31

1. Introducción

El objetivo de esta práctica es aplicar los conocimientos adquiridos sobre vistas, wizards, informes y controladores web. Para ello, nos ocuparemos de la implementación de cuatro actividades que abarcan diferentes contextos:

- **EJ07-LigaFutbol** *4 puntos*

- Se modificará el módulo para que incluya las siguientes funcionalidades:
 - Reglas de puntuación especiales. *1 punto*
 - Botones para alterar los goles de los partidos. *1 punto*
 - Web controller para eliminar empates. *0,5 punto*
 - Informe PDF por cada partido. *0,5 punto*
 - Wizard para crear nuevos partidos. *0,5 punto*
 - Vista Graph. *0,5 punto*

- **EJ08-API-REST_Socios** *1,5 puntos*

- Se grabará un vídeo explicativo realizando operaciones CRUD.

- **Bot de Telegram** *3 puntos*

- Se creará un bot de Telegram que escuchará órdenes enviadas por los usuarios.

- **Generación de imágenes aleatorias con Web Controller** *1,5 puntos*

- Se creará un *Web Controller* que reciba parámetros, genere una imagen de píxeles aleatorios y la devuelva en Base64 o PNG.

Los ejemplos citados anteriormente se encuentran en el siguiente repositorio.

<https://github.com/sergarb1/OdooModulosEjemplos>

2. Actividad 01 – Modificación EJ07-LigaFutbol

2.1. Reglas de puntuación especiales

Se solicita modificar la lógica de puntuación de los partidos para que, en un partido con 4 o más goles de diferencia, el ganador se lleve 7 puntos y al perdedor se le reste 1.

Antes de comenzar será necesario modificar el código de las vistas para que el módulo se pueda instalar correctamente en Odoo. Deberemos sustituir las etiquetas <tree> por <list> tal como se ve en la siguiente imagen:

```
addons > EJ07-LigaFutbol > views > liga_equipo.xml
  2 <odoo>
  3
  4     <record id='liga_equipo_action' model='ir.act.list'>
  5         <field name='name'>Equipos de la liga</field>
  6         <field name='res_model'>liga.equipo</field>
  7         <!-- Indicamos que este modelo se puede ver en kanban, tree y form -->
  8         <field name='view_mode'>tree,kanban,form,graph</field>
  9     </record>
 10
 11     <menuitem name="Gestión de liga" id="liga_base_menu" />
 12     <menuitem name="Equipos" id="liga_equipo_menu" parent="liga_base_menu" action="liga_equipo_action" />
 13
 14     <!-- Vista formulario -->
 15     <record id="liga_equipo_view_form" model="ir.ui.view">
 16         <field name="name">Formulario de equipo de futbol</field>
 17         <field name="model">liga.equipo</field>
 18         <field name="arch" type="xml">
 19             <form>
 20                 <group>
 21                     <group>
 22                         <field name="nombre" />
 23                         <field name="escudo" widget="image" class="oe_avatar" />
 24                         <field name="fecha_fundacion" />
 25                     </group>
 26                     <group>
 27                         <field name="descripcion" />
 28                     </group>
 29                 </group>
 30             </form>
 31         </field>
 32     </record>
 33
 34     <!-- Tree(list) View -->
 35     <record id="liga_equipo_view_tree" model="ir.ui.view">
 36         <field name="name" value="Lista de Equipos de la liga" />
 37         <field name="model" value="liga.equipo" />
 38         <field name="arch" type="xml">
 39             <tree>
```

Figura 1: Aplicar estos cambios en las 3 vistas

Una vez instalado el módulo accedemos a él y vemos lo siguiente:

Figura 2: Las 4 vistas del módulo

Tras añadir los 10 primeros equipos de la primera división española nos quedará así:

<input type="checkbox"/>	Nombre equipo	Escudo equipo	Fecha fundación
<input type="checkbox"/>	Atlético		26/04/1903
<input type="checkbox"/>	Barcelona		29/11/1899
<input type="checkbox"/>	Betis		12/09/1907
<input type="checkbox"/>	Celta		23/08/1923
<input type="checkbox"/>	Espanyol		28/10/1900
<input type="checkbox"/>	Girona		23/07/1930
<input type="checkbox"/>	Osasuna		24/10/1920
<input type="checkbox"/>	R. Madrid		06/03/1902
<input type="checkbox"/>	R. Sociedad		07/09/1909
<input type="checkbox"/>	Villarreal		10/03/1923

Figura 3: Podemos simular varios partidos de prueba...

<input type="checkbox"/> Escudo equipo	Nombre equipo	Puntos	Jugados	Goles A Favor	Goles En Contra	Victorias	Empates	Derrotas
	Barcelona	3	2	5	1	1	0	1
	Celta	3	1	1	0	1	0	0
	Girona	0	0	0	0	0	0	0
	Osasuna	0	0	0	0	0	0	0
	R. Sociedad	0	0	0	0	0	0	0
	Villarreal	0	0	0	0	0	0	0
	R. Madrid	0	1	0	5	0	0	1
	Betis	0	0	0	0	0	0	0
	Atlético	0	0	0	0	0	0	0
	Espanyol	0	0	0	0	0	0	0

Figura 4: Para ver la puntuación genérica en la clasificación general

Ahora echémosle un vistazo al código del módulo. Básicamente nos encontramos con 2 modelos, *liga_partido.py* y *liga_equipo.py*. En este último se calcula la puntuación con un campo computado llamado puntos tal como se ve en la siguiente imagen:

```
55     puntos= fields.Integer( compute="_compute_puntos",default=0, store=True)
56
57
58     @api.depends('victorias','empates')
59     def _compute_puntos(self):
60         for record in self:
61             record.puntos = record.victorias * 3 + record.empates
62
```

Dada la lógica que queremos implementar para la puntuación vamos a convertir ese campo computado en uno normal, tal que así:

```
56     puntos = fields.Integer(default=0)
57
58     # @api.depends('victorias','empates')
59     # def _compute_puntos(self):
60     #     for record in self:
61     #         record.puntos = record.victorias * 3 + record.empates
62
```

Cambiamos de modelo y nos vamos a *liga_partido.py* y añadimos el campo creado en el paso anterior dentro de la función *actualizoRegistrosEquipo()*:

```
62     def actualizoRegistrosEquipo(self):
63         #Recorremos partidos y equipos
64         for recordEquipo in self.env['liga.equipo'].search([]):
65             #Como recalculamos todo, ponemos de cada equipo todo a cero
66             recordEquipo.victorias=0
67             recordEquipo.empates=0
68             recordEquipo.derrotas=0
69             recordEquipo.goles_a_favor=0
70             recordEquipo.goles_en_contra=0
71             recordEquipo.puntos=0 ← AÑADIMOS CAMPO PUNTOS
```

En la misma función deberemos modificar todo este código...

```
73     for recordPartido in self.env['liga.partido'].search([]):
74
75         #Si es el equipo de casa
76         if recordPartido.equipo_casa.nombre==recordEquipo.nombre:
77
78             #Miramos si es victoria o derrota
79             if recordPartido.goles_casa>recordPartido.goles_fuera:
80                 recordEquipo.victorias=recordEquipo.victorias+1
81             elif recordPartido.goles_casa<recordPartido.goles_fuera:
82                 recordEquipo.derrotas=recordEquipo.derrotas+1
83             else:
84                 recordEquipo.empates=recordEquipo.empates+1
85
86         #Sumamos goles a favor y en contra
87         recordEquipo.goles_a_favor=recordEquipo.goles_a_favor+recordPartido.goles_casa
88         recordEquipo.goles_en_contra=recordEquipo.goles_en_contra+recordPartido.goles_fuera
89
90         #Si es el equipo de fuera
91         if recordPartido.equipo_fuera.nombre==recordEquipo.nombre:
92
93             #Miramos si es victoria o derrota
94             if recordPartido.goles_casa<recordPartido.goles_fuera:
95                 recordEquipo.victorias=recordEquipo.victorias+1
96             elif recordPartido.goles_casa>recordPartido.goles_fuera:
97                 recordEquipo.derrotas=recordEquipo.derrotas+1
98             else:
99                 recordEquipo.empates=recordEquipo.empates+1
100
101         #Sumamos goles a favor y en contra
102         recordEquipo.goles_a_favor=recordEquipo.goles_a_favor+recordPartido.goles_fuera
103         recordEquipo.goles_en_contra=recordEquipo.goles_en_contra+recordPartido.goles_casa
```

Figura 5: Código original...

Por este otro en el que añadiremos la nueva lógica de puntuación, tal como se muestran en las capturas de la página siguiente:

```

73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106

for recordPartido in self.env['liga.partido'].search([]):
    # AÑADIMOS NUEVA VARIABLE PARA SIMPLIFICAR LOS IF
    goles_de_diferencia = abs(recordPartido.goles_casa - recordPartido.goles_fuera)

    #Si es el equipo de casa
    if recordPartido.equipo_casa.nombre==recordEquipo.nombre:

        #Miramos si es victoria o derrota
        if recordPartido.goles_casa>recordPartido.goles_fuera:
            recordEquipo.victorias=recordEquipo.victorias+1
            # APPLICAMOS LA LÓGICA NUEVA (VICTORIAS)
            if goles_de_diferencia >= 4:
                recordEquipo.puntos += 7
            else:
                recordEquipo.puntos += 3

        elif recordPartido.goles_casa<recordPartido.goles_fuera:
            recordEquipo.derrotas=recordEquipo.derrotas+1
            # APPLICAMOS LA LÓGICA NUEVA (DERROTAS)
            if goles_de_diferencia >= 4:
                recordEquipo.puntos -= 1
            else:
                pass

        else:
            recordEquipo.empates=recordEquipo.empates+1
            # AÑADIMOS PUNTUACIÓN ESTÁNDAR EN CASO DE EMPATE
            recordEquipo.puntos += 1

    #Sumamos goles a favor y en contra
    recordEquipo.goles_a_favor=recordEquipo.goles_a_favor+recordPartido.goles_casa
    recordEquipo.goles_en_contra=recordEquipo.goles_en_contra+recordPartido.goles_fuera

```

Figura 6: Código para equipo local

```

107     #Si es el equipo de fuera
108     if recordPartido.equipo_fuera.nombre==recordEquipo.nombre:
109
110         #Miramos si es victoria o derrota
111         if recordPartido.goles_casa<recordPartido.goles_fuera:
112             recordEquipo.victorias=recordEquipo.victorias+1
113             # APLICAMOS LA LÓGICA NUEVA (VICTORIAS)
114             if goles_de_diferencia >= 4:
115                 recordEquipo.puntos += 7
116             else:
117                 recordEquipo.puntos += 3
118
119         elif recordPartido.goles_casa>recordPartido.goles_fuera:
120             recordEquipo.derrotas=recordEquipo.derrotas+1
121             # APLICAMOS LA LÓGICA NUEVA (DERROTAS)
122             if goles_de_diferencia >= 4:
123                 recordEquipo.puntos -= 1
124             else:
125                 pass
126
127         else:
128             recordEquipo.empates=recordEquipo.empates+1
129             # AÑADIMOS PUNTUACIÓN ESTÁNDAR EN CASO DE EMPATE
130             recordEquipo.puntos += 1
131
132     #Sumamos goles a favor y en contra
133     recordEquipo.goles_a_favor=recordEquipo.goles_a_favor+recordPartido.goles_fuera
134     recordEquipo.goles_en_contra=recordEquipo.goles_en_contra+recordPartido.goles_casa
135

```

Figura 7: Código para equipo visitante

Es necesario matizar que, según esta lógica, cuando un partido se resuelve con una diferencia de goles igual o superior a 4, el equipo perdedor recibirá una puntuación negativa de -1. Esto quiere decir que puede terminar la temporada con puntuación negativa.

Ahora echemos un vistazo al módulo de Odoo para ver si se han efectuado correctamente las modificaciones. Sobra decir que es necesario reiniciar el contenedor para que se apliquen los cambios.

Gestión de liga		Equipos	Clasificación	Partidos de la liga	Añadir equipo		
Nuevo Partidos de la liga						1-5 / 5	< >
<input type="text"/> Buscar...							
- Resultado - Barcelona : 2 R. Madrid : 2						- Resultado - Celta : 4 Atlético : 0	
- Resultado - Celta : 5 Espanyol : 1						- Resultado - Osasuna : 6 Celta : 2	
- Resultado - Espanyol : 5 Atlético : 0							

Figura 8: Crearemos nuevos partidos de prueba

Gestión de liga		Equipos	Clasificación	Partidos de la liga	Añadir equipo			
Nuevo Clasificación de la liga						1-10 / 10	< >	
<input type="text"/> Buscar...								
<input type="checkbox"/> Escudo equipo	Nombre equipo	Puntos	Jugados	Goles A F...	Goles En ...	Victorias	Empates	Derrotas
<input type="checkbox"/>	Celta	13	3	11	7	2	0	1
<input type="checkbox"/>	Osasuna	7	1	6	2	1	0	0
<input type="checkbox"/>	Espanyol	6	2	6	5	1	0	1
<input type="checkbox"/>	Barcelona	1	1	2	2	0	1	0
<input type="checkbox"/>	R. Madrid	1	1	2	2	0	1	0
<input type="checkbox"/>	Betis	0	0	0	0	0	0	0
<input type="checkbox"/>	Girona	0	0	0	0	0	0	0
<input type="checkbox"/>	R. Sociedad	0	0	0	0	0	0	0
<input type="checkbox"/>	Villarreal	0	0	0	0	0	0	0
<input type="checkbox"/>	Atlético	-2	2	0	9	0	0	2

Figura 9: Comprobamos que el sistema de puntuación se aplica correctamente

2.2. Botones para alterar los goles de los partidos

Se nos pide modificar la vista de los partidos para añadir 2 nuevos botones:

- **Botón “+2 goles equipos locales”** Sumará 2 goles a todos los equipos locales.
- **Botón “+2 goles equipos visitantes”** Sumará 2 goles a todos los equipos visitantes.

Tras cada acción la clasificación deberá actualizarse.

Lo primero que haremos será añadir 2 funciones de actualización en el modelo correspondiente (*liga_partido.py*) que posteriormente asignaremos a los botones de la vista.

```
addons > EJ07-LigaFutbol > models > liga_partido.py > ...
6   class LigaPartido(models.Model):
154     #Sobreescribo el metodo crear
155     @api.model
156     def create(self, values):
157       #hago lo normal del metodo create
158       result = super().create(values)
159       #Añado esto: llamo a la funcion que actualiza la clasificacion
160       self.actualizoRegistrosEquipo()
161       #hago lo normal del metodo create
162       return result
163
164     #NUEVAS FUNCIONES PARA LA BOTONERA DE GOLES
165     def sumar_goles_equipos_locales(self):
166       # BUSCAMOS TODOS LOS PARTIDOS
167       partidos = self.search([])
168
169       for partido in partidos:
170         partido.goles_casa += 2
171
172       # ACTUALIZAMOS CLASIFICACIÓN
173       self.actualizoRegistrosEquipo()
174
175     def sumar_goles_equipos_visitantes(self):
176       # BUSCAMOS TODOS LOS PARTIDOS
177       partidos = self.search([])
178
179       for partido in partidos:
180         partido.goles_fuera += 2
181
182       # ACTUALIZAMOS CLASIFICACIÓN
183       self.actualizoRegistrosEquipo()
184 |
```

Figura 10: Recorremos array, aplicamos +2 a la propiedad correspondiente y actualizamos

Ahora toca modificar la vista *liga_partido.xml* para añadir los nuevos botones. En esta vista hay 2 formas de visualizar los datos, una en modo Lista y la otra en modo Kanban (la que está por defecto). Para hacer correctamente la implementación solicitada deberemos añadir la nueva botonera en la cabecera de ambas etiquetas, forzando su visualización (si no la forzamos deberemos seleccionar obligatoriamente algún partido para que se visualicen los botones), tal como se ve en las siguientes capturas:

```
addons > EJ07-LigaFutbol > views > liga_partido.xml
 2 <odoo>
36     <!-- Vista list -->!! LISTA !!
37     <record id="liga_partido_view_list" model="ir.ui.view">
38         <field name="name">Lista de partidos de la liga</field>
39         <field name="model">liga.partido</field>
40         <field name="arch" type="xml">
41             <list>
42                 <header>
43                     <button name="sumar_goles_equipos_locales"
44                         string="+2 goles equipos locales"
45                         type="object"
46                         display="always" → IMPORTANTE!
47                         class="oe_highlight"
48                         confirm="Esta acción sumará 2 goles a todos los equipos locales, ¿estás seguro/a?"/>
49                     <button name="sumar_goles_equipos_visitantes"
50                         string="+2 goles equipos visitantes"
51                         type="object"
52                         display="always" → IMPORTANTE!
53                         class="oe_highlight"
54                         confirm="Esta acción sumará 2 goles a todos los equipos visitantes, ¿estás seguro/a?"/>
55                 </header>
56                 <!-- Indicamos que atributos usaremos al hacer la vista list -->
57                 <field name="equipo_casa" />
58                 <field name="goles_casa" />
59                 <field name="equipo_fuera" />
60                 <field name="goles_fuera" />
61             </list>
62         </field>

```

```
addons > EJ07-LigaFutbol > views > liga_partido.xml
 2 <odoo>
67     <!-- Vista Kanban -->!! KANBAN !!
68     <record id="liga_partido_view_kanban" model="ir.ui.view">
69         <field name="name">Lista de partidos de la liga</field>
70         <field name="model">liga.partido</field>
71         <field name="arch" type="xml">
72             <!-- Agrupamos por el atributo "parent_id"-->
73             <kanban>
74                 <header>
75                     <button name="sumar_goles_equipos_locales"
76                         string="+2 goles equipos locales"
77                         type="object"
78                         display="always" → IMPORTANTE!
79                         class="oe_highlight"
80                         confirm="Esta acción sumará 2 goles a todos los equipos locales, ¿estás seguro/a?"/>
81                     <button name="sumar_goles_equipos_visitantes"
82                         string="+2 goles equipos visitantes"
83                         type="object"
84                         display="always" → IMPORTANTE!
85                         class="oe_highlight"
86                         confirm="Esta acción sumará 2 goles a todos los equipos visitantes, ¿estás seguro/a?"/>
87                 </header>

```

Llegados a este punto sólo nos falta comprobar que los cambios se han realizado correctamente. Visualicemos el módulo de Odoo para comprobarlo.

Figura 11: Vista Kanban (por defecto)

Equipo local	Goles Casa	Equipo visitante	Goles Fuera
<input type="checkbox"/> Barcelona	6	R. Madrid	13
<input type="checkbox"/> Celta	8	Atlético	6
<input type="checkbox"/> Celta	9	Espanyol	7
<input type="checkbox"/> Osasuna	10	Celta	8
<input type="checkbox"/> Espanyol	9	Atlético	6

Figura 12: Vista Lista

Probemos la funcionalidad de los botones:

The screenshot shows the 'Gestión de liga' interface. At the top, there are tabs: 'Nuevo', '+2 goles equipos locales' (highlighted with a red arrow), '+2 goles equipos visitantes', and 'Partidos de la liga'. Below the tabs is a search bar labeled 'Buscar...'. On the right side of the header, there are navigation icons and a page number '1-5 / 5'. A red box highlights the 'F' icon in the top right corner. The main area displays four cards, each titled '- Resultado -'. The first card shows 'Barcelona : 6' and 'R. Madrid : 13'. The second card shows 'Celta : 8' and 'Atlético : 6'. The third card shows 'Celta : 9' and 'Espanyol : 7'. The fourth card shows 'Osasuna : 10' and 'Celta : 8'. A modal dialog box titled 'Confirmación' is overlaid on the interface. It contains the message 'Esta acción sumará 2 goles a todos los equipos locales. ¿Estás seguro/a?' and two buttons: 'Aceptar' (highlighted with a red arrow) and 'Cancelar'.

Figura 13: Añadimos 2 goles a todos los equipos locales en Vista Kanban

The screenshot shows the 'Gestión de liga' interface after the action from Figure 13 has been completed. The interface is identical to Figure 13, but the results have changed. The first card now shows 'Barcelona : 8' and 'R. Madrid : 13'. The second card shows 'Celta : 10' and 'Atlético : 6'. The third card shows 'Celta : 11' and 'Espanyol : 7'. The fourth card shows 'Osasuna : 12' and 'Celta : 8'. All other elements of the interface remain the same.

Figura 14: El resultado es satisfactorio

The screenshot shows a table of football match results. A red arrow points from the '+2 goles equipos locales' button in the top navigation bar to the 'Aceptar' button in a modal dialog box. Another red arrow points from the 'Aceptar' button in the dialog to the text inside it.

Equipo local	Goles Casa	Equipo visitante	Goles Fu...
<input type="checkbox"/> Barcelona	8	R. Madrid	13
<input type="checkbox"/> Celta	10	Atlético	6
<input type="checkbox"/> Celta	11	Espanyol	7
<input type="checkbox"/> Osasuna	12	Celta	8
<input type="checkbox"/> Espanyol	11	Atlético	6

Confirmación

Esta acción sumará 2 goles a todos los equipos visitantes. ¿estás seguro/a?

Aceptar Cancelar

Figura 15: Añadimos 2 goles a todos los equipos locales en Vista Lista

The screenshot shows the same table of match results after the update. The 'Goles Fu...' column has been updated to reflect the addition of two goals to each away team.

Equipo local	Goles Casa	Equipo visitante	Goles Fu...
<input type="checkbox"/> Barcelona	8	R. Madrid	15
<input type="checkbox"/> Celta	10	Atlético	8
<input type="checkbox"/> Celta	11	Espanyol	9
<input type="checkbox"/> Osasuna	12	Celta	10
<input type="checkbox"/> Espanyol	11	Atlético	8

Figura 16: El resultado es satisfactorio

Por último, vayamos a la Vista de Clasificación para comprobar que los datos se han actualizado correctamente.

<input type="checkbox"/> Escudo equipo	Nombre equipo	Puntos	Jugados	Goles A Favor	Goles En Contra	Victorias	Empates	Derrotas
<input type="checkbox"/>	R. Madrid	7	1	15	8	1	0	0
<input type="checkbox"/>	Celta	6	3	31	29	2	0	1
<input type="checkbox"/>	Olasuna	3	1	12	10	1	0	0
<input type="checkbox"/>	Espanyol	3	2	20	19	1	0	1
<input type="checkbox"/>	Atlético	0	2	16	21	0	0	2
<input type="checkbox"/>	Girona	0	0	0	0	0	0	0
<input type="checkbox"/>	R. Sociedad	0	0	0	0	0	0	0
<input type="checkbox"/>	Villarreal	0	0	0	0	0	0	0
<input type="checkbox"/>	Betis	0	0	0	0	0	0	0
<input type="checkbox"/>	Barcelona	-1	1	8	15	0	0	1

Figura 17: Actualización realizada satisfactoriamente

2.3. Web Controller para eliminar empates

Se nos solicita crear un Web Controller que elimine todos los partidos que hayan terminado en empate y que devuelva el número de partidos eliminados. La url designada para dicha petición será `http://localhost:puerto/eliminarempates` (en este caso, debido a la configuración de Docker, el puerto será el 9001).

Lo primero que haremos será echar un vistazo a la clase `main.py` ubicada en la carpeta `controllers`. Aquí crearemos una nueva función en la que, al acceder a la url requerida, efectuará la lógica solicitada. Veamos el código:

```
addons > EJ07-LigaFutbol > controllers > main.py > ...
You, now | 1 author (You)
1  # -*- coding: utf-8 -*-
2  from odoo import http      Import "odoo" could not be resolved
3  from odoo.http import request  Import "odoo.http" could not be resolved
4  import json
5
6  #Clase del controlador web
7  You, now | 1 author (You)
8  class Main(http.Controller):
9      #Decorador que indica que la url "/ligafutbol/equipo/json" atenderá por HTTP, sin autenticación
10     #Devolverá texto que estará en formato JSON
11     #Se puede probar accediendo a http://localhost:8069/ligafutbol/equipo/json
12     @http.route('/ligafutbol/equipo/json', type='http', auth='none')
13     def obtenerDatosEquiposJSON(self):
14         #Obtenemos la referencia al modelo de Equipo
15         equipos = request.env['liga.equipo'].sudo().search([])
16
17         #Generamos una lista con información que queremos sacar en JSON
18         listaDatosEquipos=[]
19         for equipo in equipos:
20             listaDatosEquipos.append([equipo.nombre,str(equipo.fecha_fundacion),equipo.jugados,equipo.puntos,
21                                     equipo.victorias,equipo.empates,equipo.derrotas])
22         #Convertimos la lista generada a JSON
23         json_result=json.dumps(listaDatosEquipos)
24
25     return json_result
```

Figura 18: Web Controller por defecto

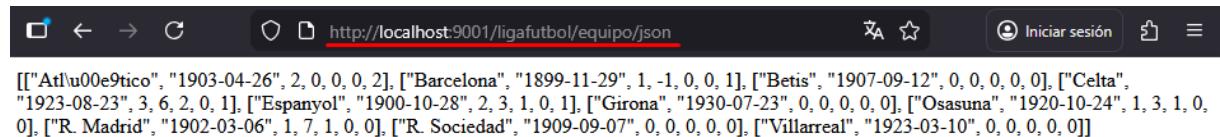


Figura 19: Comprobamos que efectivamente funciona

Creamos la nueva función solicitada, quedaría tal como se muestra en la siguiente captura:

```
addons > EJ07-LigaFutbol > controllers > main.py > ...
7   class Main(http.Controller):
26       # NUESTRO WEB CONTROLLER PARA ELIMINAR EMPATES
27       # http://localhost:9001/eliminarempates
28       @http.route('/eliminarempates', type='http', auth='none')
29       def eliminarPartidosConEmpate(self):
30           # OBTENEMOS TODOS LOS REGISTROS DE LA TABLA DE LA BD
31           partidos = request.env['liga.partido'].sudo().search([])
32
33           # LISTA TEMPORAL EN LA QUE GUARDAREMOS EMPATES EXISTENTES
34           listaEmpates = []
35
36           # RECORREMOS TODOS LOS REGISTROS
37           for partido in partidos:
38               # COMPROBAMOS SI HAY EMPATES Y LOS AÑADIMOS A LA LISTA
39               if partido.goles_casa == partido.goles_fuera:
40                   listaEmpates.append(partido)
41
42           # VARIABLE TEMPORAL DE TOTAL EMPATES, NECESARIA PORQUE DESPUÉS DEL BORRADO len(listaEmpates) SIEMPRE SERÁ 0
43           empatesTotales = len(listaEmpates)
44
45           # SI HAY EMPATES ELIMINAMOS LOS REGISTROS Y MOSTRAMOS LA RESPUESTA SOLICITADA
46           if empatesTotales > 0:
47               for empate in listaEmpates:
48                   empate.unlink()
49
50                   respuesta = "El total de empates eliminados ha sido de: " + str(empatesTotales)
51           # SI NO HAY EMPATES LO INDICAREMOS TAMBIÉN
52           else:
53               respuesta = "NO se han podido realizar el borrado porque NO se han detectado empates"
54
55           # DEVOLVEMOS LA RESPUESTA ESPECIFICANDO QUE ES TEXTO PLANO PARA EVITAR POSIBLES ERRORES
56           return request.make_response(respuesta, [('Content-Type', 'text/plain')])
```

Haremos una prueba que nos debería devolver un resultado negativo ya que nos tenemos almacenados partidos con empate:

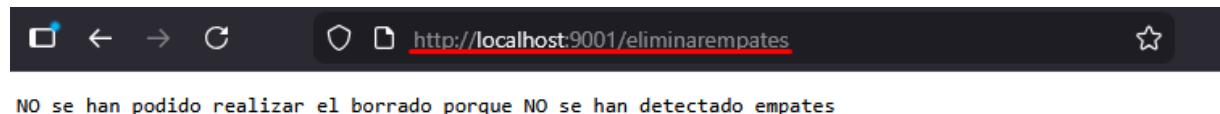


Figura 20: Este resultado es correcto

Como queremos comprobar un resultado positivo crearemos unos empates y volveremos a hacer la petición, tal como se muestra en las siguientes imágenes:

The screenshot shows a web interface for managing a league. The top navigation bar includes tabs for 'Gestión de liga', 'Equipos', 'Clasificación', and 'Partidos de la liga' (which is highlighted with a red box). Below the navigation is a search bar and a toolbar with icons for 'Nuevo', '+2 goles equipos locales', '+2 goles equipos visitantes', and 'Partidos de la liga'. The main content area displays seven match results in boxes:

- Resultado -
Barcelona : 8
R. Madrid : 15
- Resultado -
Celta : 10
Atlético : 8
- Resultado -
Celta : 11
Espanyol : 9
- Resultado -
Osasuna : 12
Celta : 10
- Resultado -
Espanyol : 11
Atlético : 8
- Resultado -
Girona : 0
R. Sociedad : 0
- Resultado -
Betis : 2
Espanyol : 2

Figura 21: Creamos 2 empates de prueba

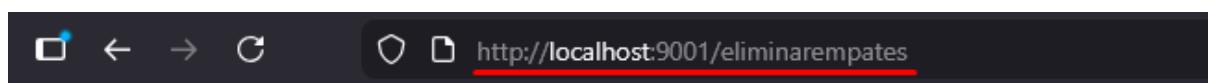


Figura 22: Ahora el resultado es el correcto tras la eliminación de empates

Finalmente vamos a comprobar la clasificación para verificar que las puntuaciones se han actualizado correctamente tras los borrados:

<input type="checkbox"/> Escudo equipo	Nombre equipo	Puntos	Jugados	Goles A Favor	Goles En Contra	Victorias	Empates	Derrotas
	R. Madrid	7	1	15	8	1	0	0
	Celta	6	3	31	29	2	0	1
	Espanyol	3	2	20	19	1	0	1
	Osasuna	3	1	12	10	1	0	0
	Villarreal	0	0	0	0	0	0	0
	Girona	0	0	0	0	0	0	0
	R. Sociedad	0	0	0	0	0	0	0
	Betis	0	0	0	0	0	0	0
	Atlético	0	2	16	21	0	0	2
	Barcelona	-1	1	8	15	0	0	1

Figura 23: Puntuaciones actualizadas satisfactoriamente

No hay duda... este Web Controller funciona a las mil maravillas.

2.4. Informe PDF por cada partido

Se nos pide generar un informe PDF que muestre los datos de un partido. Para ello crearemos una plantilla visual dentro de la carpeta *report* (que está relacionada con la generación de documentos impresos) y un botón dentro de la vista de partidos para que el usuario pueda generar el documento.

Echémosle un vistazo a la plantilla que ya está creada:

```
addons > EJ07-LigaFutbol > report > liga.equipo.clasificacion.report.xml
You, now | 1 author (You)
1  <?xml version="1.0" encoding="utf-8"?>
2  <odoo>
3
4      <!-- Ejemplo de informe asociado a equipo (se genera al darle aimprimir
5          desde la vista Form de un equipo) -->
6      <template id="report_clasificacion_view">
7          <!-- Aquí la estructura a seguir -->
8          <t t-call="web.html_container">
9              <t t-foreach="docs" t-as="doc">
10                 <t t-call="web.internal_layout">
11                     <div class="page">
12                         <!-- Cabecera con el nombre de equipo -->
13                         <h2 t-field="doc.nombre" />
14                         <!-- Datos de los partidos jugados por el equipo -->
15                         <p>
16                             Partidos jugados:
17                             <span t-field="doc.jugados" />
18                         </p><p>
19                             Partidos ganados:
20                             <span t-field="doc.victorias" />
21                         </p><p>
22                             Partidos empatados:
23                             <span t-field="doc.empates" />
24                         </p><p>
25                             Partidos perdidos:
26                             <span t-field="doc.derrotas" />
27                         </p><p>
28                             Puntos obtenidos:
29                             <span t-field="doc.puntos" />
30                         </p>
31                     </div>
32                 </t>
33             </t>
34         </t>
35     </template>
36
37     <!-- Realmente, este es el informe, lo de arriba es la plantilla que utilizará el informe -->
38     <report id="report_clasificacion" model="liga.equipo" string="Informe clasificación de cada equipo"
39         name="EJ07-LigaFutbol.report_clasificacion_view" file="EJ07-LigaFutbol.report_clasificacion_view"
40         report_type="qweb-pdf" />
41
42
43 </odoo>
```

Basándonos en esa plantilla crearemos una nueva desde cero que cumpla con los requisitos solicitados. La llamaremos *liga_partido_report.xml* y la ubicaremos dentro de la carpeta *report*. Será necesario enlazarla dentro del archivo *__manifest__.py* del proyecto, tal como se muestra en la siguiente captura:

```
addons > EJ07-LigaFutbol > __manifest__.py
2  {
10     #Indicamos que es una aplicación
11     'application': True,
12     'author': "Sergi García",
13     'website': "http://apuntesfpinformatica.es",
14     'category': 'Tools',
15     'version': '0.1',
16     'depends': ['base'],
17
18     'data': [
19
20         #Estos dos primeros ficheros:
21         #1) El primero indica grupo de seguridad basado en rol
22         #2) El segundo indica la política de acceso del modelo
23         #Mas información en https://www.odoo.com/documentation/17.0/es/developer/howtos/rdtraining/05_securityintro.html
24         #Y en www.odoo.yenthevg.com/creating-security-groups-odoo/
25         #'security/groups.xml',
26         #'security/ir.model.access.csv',
27
28         #Aqui distintas vistas de equipo (vistas diferentes, mismo modelo)
29         'views/liga_equipo.xml',
30         'views/liga_equipo_clasificacion.xml',
31         #Vista a un informe
32         'report/liga_equipo_clasificacion_report.xml',
33         'report/liga_partido_report.xml', ←
34         #Aqui vista sobre los partidos
35         'views/liga_partido.xml',
36         #Añadimos un Wizard para introducir equipos
37         'wizard/liga_equipo_wizard.xml'
38
39     ],
40     # Fichero con data de demo si se inicializa la base de datos con "demo data" (No incluido en ejemplo)
41     # 'demo': [
42     #     'demo.xml'
43     # ],
44
45 }
```

¡AVISO! ES IMPORTANTE QUE LA PLANTILLA VAYA ANTES QUE LA VISTA PORQUE DE LO CONTRARIO PUEDE HABER CONFLICTOS

La creación de la plantilla es problemática ya que **Odoo 18** y la librería **wkhtmltopdf**, ya que existen ciertas incompatibilidades. Odoo 18 usa Bootstrap 5, que basa casi todo su diseño en Flexbox y CSS Grid. Sin embargo, wkhtmltopdf (el motor que genera el PDF) utiliza un motor de renderizado basado en una versión muy antigua de WebKit (similar a un navegador de hace 10 años). Por las dificultades de crear una plantilla decente me he visto obligado a generar gran parte de la parte visual con asistencia de la IA.

Veamos por partes cómo ha quedado el código de la plantilla en las siguientes capturas:

```

addons > EJ07-LigaFutbol > report > liga_partido_report.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <odoo>
3
4      <record id="action_report_liga_partido" model="ir.actions.report"> ← ESTE ID SERVIRÁ PARA ENLAZAR CON
5          <field name="name">Acta del Partido</field> EL BOTÓN DE LA VISTA
6          <field name="model">liga.partido</field>
7          <field name="report_type">qweb-pdf</field>
8          <field name="report_name">EJ07-LigaFutbol.report_partido_liga_template</field> ← ESPECIFICAR NOMBRE MÓDULO
9          <field name="report_file">EJ07-LigaFutbol.report_partido_liga_template</field> Y ID TEMPLATE
10         <field name="print_report_name">'Partido - %s vs %s' % (object.equipo_casa.nombre,
11                                         object.equipo_fuera.nombre)</field>
12         <field name="binding_model_id" ref="model_liga_partido"/>
13         <field name="binding_type">report</field>
14     </record>
15
16     <template id="report_partido_liga_template"> ←
17         <t t-call="web.html_container">
18             <t t-foreach="docs" t-as="doc">
19                 <t t-call="web.external_layout">
20                     <div class="page">
21                         <div class="text-center" style="border-bottom: 2px solid black; margin-bottom: 20px;">
22                             <h2>Acta Oficial de Partido</h2>
23                         </div>
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

Figura 24: La acción (orden de generar informe) e inicio de la plantilla

```

24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

<h3 style="margin-bottom: 10px;" t-field="doc.equipo_casa.nombre"></h3> <div style="font-size: 48px; font-weight: bold;"> </div> <p style="color: #666; font-size: 12px;">LOCAL</p>	VS	<h3 style="margin-bottom: 10px;" t-field="doc.equipo_fuera.nombre"></h3> <div style="font-size: 48px; font-weight: bold;"> </div> <p style="color: #666; font-size: 12px;">VISITANTE</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 25: Esta parte mostrará los equipos y el resultado

```

48         <div style="width: 100%; margin-top: 30px;">
49             <table class="table table-sm" style="width: 100%; border-top: 1px solid #dee2e6;">
50                 <thead>
51                     <tr style="background-color: #f8f9fa;">
52                         <th style="padding: 8px; text-align: left; border-bottom: 2px solid #333;">Detalle</th>
53                         <th style="padding: 8px; text-align: right; border-bottom: 2px solid #333;">Resultado</th>
54                     </tr>
55                 </thead>
56                 <tbody>
57                     <tr>
58                         <td style="padding: 8px; border-bottom: 1px solid #eee;">Estado del encuentro</td>
59                         <td style="padding: 8px; text-align: right; border-bottom: 1px solid #eee;">
60                             <span class="badge badge-secondary" style="text-transform: uppercase;">Finalizado</span>
61                         </td>
62                     </tr>
63                     <tr>
64                         <td style="padding: 8px; border-bottom: 1px solid #eee;">Diferencia de goles</td>
65                         <td style="padding: 8px; text-align: right; border-bottom: 1px solid #eee; font-weight: bold;">
66                             <t t-esc="abs(doc.goles_casa - doc.goles_fuera)" />
67                         </td>
68                     </tr>
69                 </tbody>
70             </table>
71         </div>

```

Figura 26: Esta parte mostrará los detalles del partido

Por último falta añadir el botón en la vista *liga_partido.xml* tal como se ve en la siguiente captura:



```

addons > EJ07-LigaFutbol > views > liga_partido.xml
2   <odoo>
14     <!-- VISTA DE FORMULARIO -->
15     <record id="liga_partido_view_form" model="ir.ui.view">
16         <field name="name">Formulario Partidos</field>
17         <field name="model">liga.partido</field>
18         <field name="arch" type="xml">
19             <form>
20                 <button name="%(%(EJ07-LigaFutbol.action_report_liga_partido)d" string="Generar PDF" type="action" class="btn-primary"/>
21             <group>
22                 <group>
23                     <field name="equipo_casa" />
24                     <field name="goles_casa" />
25                 </group>
26
27                 <group>
28                     <field name="equipo_fuera" />
29                     <field name="goles_fuera" />
30                 </group>
31             </form>
32         </field>
33     </record>
34

```

ESTE FORMATO DE CADENA "%(..)d" PRIMERO INDICA QUE ES UNA ID DE LA BASE DE DATOS Y SEGUNDO PRECISA QUE ES UN NÚMERO ENTERO. ¡¡ES ESTRÍCTAMENTE NECESARIO DAR ESTE FORMATO!!

Figura 27: Esta será nuestro botón, se llamará “Generar PDF”

Sólo nos falta comprobar si funciona. Cabe recordar que será necesario reiniciar el contenedor y actualizar el módulo desde Odoo. Veamos los resultados en las siguientes capturas:

Gestión de liga

Nuevo +2 goles equipos locales +2 goles equipos visitantes Partidos de la liga Añadir equipo

1-5 / 5 < >

Buscar...

- Resultado -	- Resultado -
Barcelona : 8 R. Madrid : 15	Celta : 10 Atlético : 8
- Resultado -	- Resultado -
Celta : 11 Espanyol : 9	Osasuna : 12 Celta : 10
- Resultado -	
Espanyol : 11 Atlético : 8	

Figura 28: Clicamos en un partido

Gestión de liga

Nuevo Partidos de la liga liga.partido.9

2 / 5 < >

Generar PDF	Equipo local Celta
	Goles Casa 10
	Equipo visitante Atlético
	Goles Fueras 8

Figura 29: Clicamos en el nuevo botón

Acta Oficial de Partido

Acta Oficial de Partido

Celta **10** VS **Atlético** **8**

LOCAL VISITANTE

Detalle	Resultado
Estado del encuentro	FINALIZADO
Diferencia de goles	2

Figura 30: Tras unos segundos se generará el siguiente archivo PDF

Si quisiésemos imprimir todos los partidos en un único archivo deberemos cargar la vista Lista, seleccionar todos los partidos y darle a imprimir.

The screenshot shows the Odoo 'Gestión de liga' (League Management) module. At the top, there are tabs for 'Gestión de liga', 'Equipos', 'Clasificación', 'Partidos de la liga', and 'Añadir equipo'. Below the tabs, there are buttons for 'Nuevo', '+2 goles equipos locales', '+2 goles equipos visitantes', and 'Partidos de la liga'. A status bar indicates '5 seleccionado' (5 selected). An 'Imprimir' (Print) button is highlighted with a red arrow. To its right is an 'Acciones' (Actions) button. The main area displays a table of matches:

Equipo local	Goles Casa	Equipo visitante	Goles Fu...
Barcelona	8	R. Madrid	15
Celta	10	Atlético	8
Celta	11	Espanyol	9
Osasuna	12	Celta	10
Espanyol	11	Atlético	8

The screenshot shows a PDF viewer displaying a generated PDF document. The document is titled 'Acta Oficial de Partido'. It features a header with the teams: 'Barcelona' and 'R. Madrid'. The score is listed as '8 VS 15'. Below the score, it says 'LOCAL' under Barcelona and 'VISITANTE' under R. Madrid. A table at the bottom provides details: 'Estado del encuentro' (FINALIZADO) and 'Diferencia de goles' (7). On the left side of the PDF viewer, a vertical sidebar shows a preview of the generated PDF pages, with the first page highlighted by a red box.

Figura 31: ¿El resultado? Un PDF con todos los partidos

Es un quebradero de cabeza muy elevado crear una plantilla decente si no se tienen en cuenta todas las dificultades existentes actualmente entre Odoo y la herramienta externa *wkhtmltopdf*.

2.5. Wizard para crear nuevos partidos

05

2.6. Vista Graph

06

3. Actividad 02 – Pruebas *EJ08-API-REST_Socio*

Para la realización de esta actividad he empleado las siguientes herramientas:

- **Kdenlive:** es un potente editor de video no lineal, libre y de código abierto, basado en el framework MLT. Funciona en Linux, Windows y macOS, permitiendo la edición multipista de audio y video con múltiples efectos, transiciones y herramientas de corrección de color, ideal tanto para usuarios principiantes como profesionales.
- **Luvvoice:** es una plataforma de inteligencia artificial diseñada para convertir texto en voz (Text-to-Speech) de forma realista y natural. Es una herramienta muy utilizada por creadores de contenido para narrar videos de YouTube o TikTok, así como por educadores para crear audiolibros o material didáctico.

El vídeo creado se divide en dos partes bien diferenciadas. En la primera parte se muestra el código de la API donde se puede apreciar la url en la que realizar las peticiones y el formato en el que deben enviarse los datos. En la segunda parte se muestran las peticiones http (POST, GET, PUT y DELETE) hechas con el programa **Postman**.

La duración del vídeo no debe exceder de los 3 minutos (en este caso dura aproximadamente 2 minutos y medio).

El archivo *Actividad 2.mp4* está ubicado en la carpeta *practica5* del proyecto.

4. Actividad 03 – Bot Telegram API REST

03

5. Actividad 04 – Generación imágenes aleatorias Web Controller

```
docker exec -u 0 -it d-odoo /bin/bash exit
```

docker exec Indica a Docker que quieres ejecutar un nuevo proceso dentro de un contenedor que ya está en marcha.

-u 0 Define el usuario con el que vas a entrar. El 0 es el ID del usuario root. Esto te da permisos totales de administrador dentro del contenedor.

-it Es la combinación de dos banderas:

-i (interactive): Mantiene el flujo de entrada abierto.

-t (tty): Asigna una terminal virtual para que se vea y actúe como una consola normal.

<nombre_o_id> El identificador único del contenedor donde quieres entrar.

/bin/bash El programa que quieres ejecutar. En este caso, abres la consola Bash. Si el contenedor es muy ligero (como Alpine), podrías necesitar usar /bin/sh en su lugar.