

**ESCUELA POLITÉCNICA  
SUPERIOR DE CÓRDOBA**  
Universidad de Córdoba



**TRABAJO FIN DE GRADO**  
*Grado en Ingeniería Informática*  
**PLATAFORMA WEB PARA EL  
SEGUIMIENTO DEL APRENDIZAJE DE  
TEORÍA MUSICAL A TRAVÉS DE  
JUEGOS ONLINE**

**Manual de Código**

**Autor:** Javier Ramírez Quintero  
**Director:** Antonio Aráuzo Azofra  
María Luque Rodríguez

Córdoba, septiembre de 2019



UNIVERSIDAD DE CÓRDOBA



## Índice de contenido

1.	INTRODUCCIÓN.....	1
2.	INSTALACIÓN .....	3
3.	ESTRUCTURA.....	5
4.	CÓDIGO FUENTE .....	7
4.1.	Módulo [alumns] .....	8
4.2.	Módulo [teachers].....	13
4.3.	Módulo [games].....	15
4.4.	Módulo [grupos] .....	16
4.5.	Módulo [main].....	20
4.6.	Fichero [static].....	22
4.6.1.	Fichero [css] .....	22
4.6.2.	Fichero [img].....	28
4.7.	Fichero [templates] .....	28
4.7.1.	Módulo [base] .....	29
4.7.2.	Módulo [alumn].....	30
4.7.3.	Módulo [teacher] .....	37
4.7.4.	Módulo [grupo] .....	41
4.7.5.	Módulo [main].....	44
4.8.	Fichero [game].....	51
4.9.	API.....	52



## Índice de códigos

Archivo 1. run.py.....	7
Archivo 2. __init__.py.....	7
Archivo 3. views.py – alumni.....	13
Archivo 4. views.py – teachers.....	15
Archivo 5. views.py – games.....	16
Archivo 6. views.py – grupos.....	20
Archivo 7. views.py – main.....	21
Archivo 8. avatar.css – [static] .....	22
Archivo 9. calendar.css – [static] .....	23
Archivo 10. style.css – [static].....	25
Archivo 11. main.css – [static] .....	26
Archivo 12. Propiedad body actualizada para “main_t.css” .....	26
Archivo 13. resp.css – [static] .....	28
Archivo 14. base.html – [templates-base] .....	30
Archivo 15. avatar_a.html – [templates-alumn].....	31
Archivo 16. avatares.html – [templates-alumn] .....	33
Archivo 17. index_a.html – [templates-alumn].....	34
Archivo 18. profile_a.html – [templates-alumn] .....	34
Archivo 19. edit_a.html – [templates-alumn].....	35
Archivo 20. register_a.html – [templates-alumn].....	35
Archivo 21. login_a.html – [templates-alumn] .....	36
Archivo 22. games_a.html – [templates-alumn] .....	36
Archivo 23. results_a.html – [templates-alumn] .....	37
Archivo 24. logros_a.html – [templates-alumn].....	37
Archivo 25. index_t.html –[templates-teacher].....	38
Archivo 26. register_t.html – [templates-teacher].....	39
Archivo 27. login_t.html – [templates-teacher].....	39
Archivo 28. profile_t.html – [templates-teacher] .....	40
Archivo 29. edit_t.html – [templates-teacher].....	40
Archivo 30. results_alumn.html – [templates-teacher] .....	41
Archivo 31. manage.html – [templates-grupo].....	41
Archivo 32. grupo.html – [templates-grupo].....	42
Archivo 33. edit_g.html – [templates-grupo] .....	42
Archivo 34. delete_g.html – [templates-grupo].....	42
Archivo 35. create.html – [templates-grupo].....	43
Archivo 36. confirm.html – [templates-grupo] .....	43
Archivo 37. choose.html – [templates-grupo] .....	43
Archivo 38. see.html – [templates-grupo] .....	44
Archivo 39. grupo_teacher.html – [templates-grupo] .....	44
Archivo 40. about_us.html – [templates-main] .....	44
Archivo 41. about.html – [templates-main].....	45
Archivo 42. calendar.html – [templates-main].....	45
Archivo 43. calendario.html – [templates-main].....	47
Archivo 44. contact.html – [templates-main] .....	47
Archivo 45. contacto.html – [templates-main].....	47
Archivo 46. academy.html – [templates-main] .....	48
Archivo 47. denegado.html – [templates-main] .....	49
Archivo 48. index.html – [templates-main].....	49
Archivo 49. inicio.html – [templates-main] .....	51

Archivo 50. games.html .....	51
Archivo 51. API en views.py del módulo [alumni] .....	52
Archivo 52. API en script.js del código del juego: Notas Musicales .....	52

# 1. INTRODUCCIÓN

Este manual de código pertenece al proyecto, “Plataforma Web para el seguimiento del aprendizaje de teoría musical a través de juegos online”.

Los lenguajes de programación utilizados son:

- ✓ Python a través del framework Flask, utilizando las siguientes librerías:
  - flask-MySQL (para conectar con la base de datos)
  - Flask-Flash (para mandar mensajes desde el servidor)
- ✓ MySQL para la base de datos
- ✓ HTML, CSS y JavaScript para el diseño web

A continuación, se pasa a detallar todo lo necesario para instalar el framework Flask así como sus dependencias (librerías).





## 2. INSTALACIÓN

Se ha trabajado en todo momento bajo el sistema operativo (SO) Linux, por lo que se indicarán los pasos a seguir para su instalación bajo este SO.

- 1) Abrimos un terminal y comprobamos si tenemos instalado Python

```
python -version
```

- 2) Si no está instalado Python 2.7, instale Python con el administrador de paquetes de su distribución

```
sudo apt-get install python3
```

- 3) Para verificar que hemos instalado correctamente Python, escribimos la instrucción del paso 1)

- 4) A continuación, creamos una carpeta en nuestro escritorio, a la que llamaremos por ejemplo “flask\_web” y nos situaremos en dicha carpeta

```
mkdir flask_web
```

```
cd flask_web
```

- 5) Una vez realizado el proceso anterior, instalamos Flask

```
pip install flask
```

- 6) Por último, instalamos todas las dependencias necesarias para que nuestro proyecto funcione correctamente, tal como se indican en el archivo “requirements.txt” de nuestro proyecto:

```
pip install flask-mysql
```

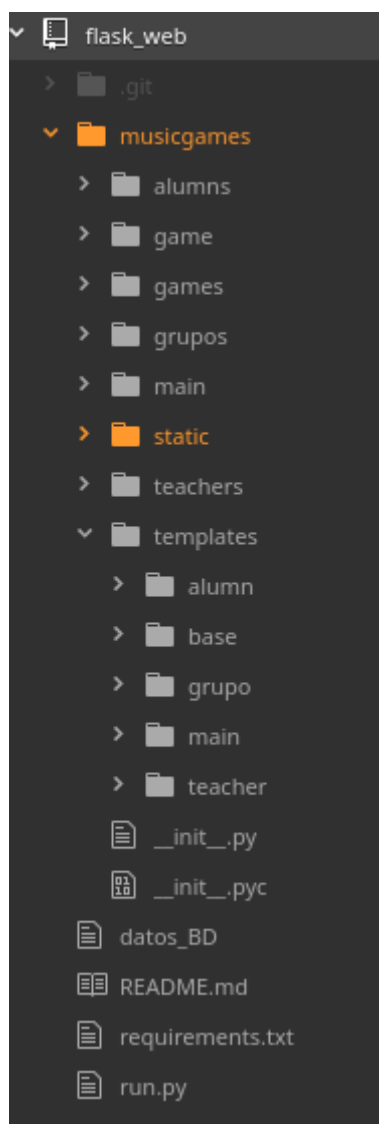
```
pip install Flask-flash
```

Con estos pasos tendremos todo listo para empezar a trabajar con nuestro proyecto, a continuación, se pasa a detallar la estructura del proyecto, así como, el código de cada módulo.



### 3. ESTRUCTURA

Nuestro proyecto sigue un patrón de arquitectura Modelo-Vista-Controlador (MVC) el cual trabaja con el framework Flask, nuestra estructura es la siguiente:



En un proyecto Flask, el archivo “requirements.txt”, indica al desarrollador las dependencias Python que debe instalar para que el proyecto funcione correctamente. El archivo “run.py” es el encargado de ejecutar la aplicación Flask.

El fichero [musicgames] contiene la aplicación Flask en sí. Y dentro de éste:

- ✓ [templates]  
Contiene las plantillas web de nuestra aplicación Flask, divididas en [alumn], [base], [grupo], [main], [teacher].
- ✓ [static]  
Contiene los archivos y ficheros estáticos de nuestra aplicación:
  - [css]  
Contiene los archivos de estilos y de responsive.
  - [img]  
Contiene los archivos imagen usados.

- [js]  
Contiene los archivos JavaScript usados
- ✓ [alumnos], [grupos], [games], [main], [teachers]  
Distintos módulos de la aplicación Flask.
- “\_\_init\_\_.py”  
Sirve para inicializar cada módulo como paquete de Python.
- “views.py”  
Contiene todas las vistas accesibles.
- “\_\_init\_\_.py”  
Conecta con la base de datos (MySQL), crea la aplicación Flask y registra los objetos alumnos, games, grupos, main y teachers.

A continuación, pasaremos a detallar el código por cada módulo que refleja nuestra estructura de la plataforma web.

## 4. CÓDIGO FUENTE

En la raíz de nuestro proyecto nos encontramos con el fichero [musicgames], un archivo “README.md” con información del proyecto; un archivo “datos\_BD” que incluye todas las tablas de nuestra base de datos, y la cual está explicada en el capítulo 1 del Manual de Usuario de este proyecto. También tenemos el archivo “requirements.txt” explicado en el capítulo de instalación y el archivo “run.py”, sin él, sería imposible arrancar nuestro proyecto.

```
from musicgames import create_app

app = create_app()

if __name__ == '__main__':
    app.run(debug=True)
```

*Archivo 1. run.py*

Con ‘debug=True’, indicamos a nuestro terminal que estamos trabajando en modo desarrollador por lo que cualquier cambio que realicemos, actualizará el servidor.

Cuando subimos un nivel más, nos encontramos con todos los módulos descritos anteriormente, con el fichero “static”, “templates” y con el archivo “\_\_init\_\_.py”. Este archivo es el encargado de realizar la conexión con la base de datos, crear la app Flask y conectar con los módulos ya citados, a través de los Blueprint.

```
from flask import Flask
from flask_mysql import MySQL

app = Flask(__name__)

app.secret_key = 'mysecretkey'

app.config['MYSQL_HOST']='127.0.0.1'
app.config['MYSQL_USER']='xabl0'
app.config['MYSQL_PASSWORD']='121087'
app.config['MYSQL_DB']='academia'

mysql = MySQL()

def create_app():

    mysql.init_app(app)

    from musicgames.teachers.views import teachers
    from musicgames.alumnos.views import alumnos
    from musicgames.grupos.views import grupos
    from musicgames.main.views import main
    from musicgames.games.views import juego

    app.register_blueprint(teachers)
    app.register_blueprint(alumnos)
    app.register_blueprint(grupos)
    app.register_blueprint(main)
    app.register_blueprint(juego)

    return app
```

*Archivo 2. \_\_init\_\_.py*

## 4.1. Módulo [alumns]

Este fichero esta dividido a su vez en dos archivos, estos archivos son comunes al resto de módulos. El archivo “\_\_init\_\_.py”, es necesario para que todos los módulos funcionen correctamente, aunque se encuentra vacío.

El archivo “views.py”, es el encargado de gestionar todas las vistas/rutas que son accesibles por los alumnos.

```
import os
from flask import Flask, render_template, request, redirect, url_for, flash, session,
Blueprint
from flask import jsonify
from musicgames import mysql
import MySQLdb.cursors
import re

alumns = Blueprint('alumns', __name__, template_folder='../templates/alumn/templates')

# Registrar Alumnado
@alumns.route('/register_a', methods=['GET', 'POST'])
def register_a():
    if request.method == 'POST' and 'username' in request.form and 'password' in
request.form and 'fullname' in request.form and 'birth_date' in request.form and 'email'
in request.form and 'avatar' in request.form:
        # Crear un acceso facil para las variables
        username = request.form['username']
        password = request.form['password']
        fullname = request.form['fullname']
        birth_date = request.form['birth_date']
        email = request.form['email']
        avatar = request.form['avatar']
        # Comprueba si la cuenta existe ya
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM alumn WHERE username = %s", [username])
        account = cursor.fetchone()
        # Si la cuenta existe muestra un error de validacion
        if account:
            flash('Cuenta ya existente, elija otro nombre de usuario', 'danger')
        elif not re.match(r'^[^\s@]+@[^\s@]+\.[^\s@]+', email):
            flash('Email no valido', 'danger')
        elif not re.match(r'[A-Za-z0-9]+', username):
            flash('Nombre de usuario, debe contener numeros y/o letras', 'danger')
        elif not username or not password or not fullname or not email or not
birth_date:
            flash('Por favor, rellene el formulario', 'danger')
        else:
            # Si la cuenta no existe y el form es valido, se agrega el nuevo alumno
            cursor.execute("""
INSERT INTO alumn (username, password, fullname, birth_date, email, avatar)
VALUES (%s, %s, %s, %s, %s, %s)""",
            (username, password, fullname, birth_date, email, avatar))
            mysql.connection.commit()
            flash('Tu cuenta ha sido creada. Ahora puedes iniciar sesion', 'success')
            return redirect(url_for('alumns.login_a'))
    # Si falta algun dato
    elif request.method == 'POST':
        flash('Por favor, rellene todo el formulario', 'danger')
    return render_template('register_a.html', title='Registro')

# Acceso Alumnado
@alumns.route('/login_a/', methods=['GET', 'POST'])
def login_a():
    if request.method == 'POST' and 'username' in request.form and 'password' in
request.form:
        username = request.form['username']
        password = request.form['password']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM alumn WHERE username = %s AND password = %s',
(username, password))
        account = cursor.fetchone()
        if account:
            # Crear sesion dato, podemos acceder a este dato por otras rutas
```

```

        session['loggedin'] = True
        session['alumn_id'] = account['alumn_id']
        session['username'] = account['username']
        # Se actualiza la fecha cuando inicia sesion
        cursor.execute("""
        UPDATE alumn
        SET login_date = curdate()
        WHERE username = %s
        """, [username])
        mysql.connection.commit()
        # Redireccionamos al inicio
        return redirect(url_for('main.home'))
    else:
        # Si la cuenta no existe o usuario/clave son erroneos
        flash('Inicio de sesion fallido. Por favor, compruebe su usuario y clave',
'danger')
        return redirect(url_for('alumni.login_a'))
    return render_template('login_a.html', title='Acceso')

# Perfil Alumnado
@alumni.route('/alumn/profile')
def alumn_profile():
    if 'loggedin' in session:
        # Necesitamos toda la informacion de la cuenta del alumno para poder mostrarla
        en la pagina de perfil
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM alumn WHERE alumn_id = %s', [session['alumn_id']])
        account = cursor.fetchone()
        # Muestra el perfil si esta logueado
        # Se comprueba el avatar elegido
        if account['avatar'] == "1":
            avatar = url_for('static', filename='img/avatar/1.png')
        elif account['avatar'] == "2":
            avatar = url_for('static', filename='img/avatar/2.png')
        elif account['avatar'] == "3":
            avatar = url_for('static', filename='img/avatar/3.png')
        elif account['avatar'] == "4":
            avatar = url_for('static', filename='img/avatar/4.png')
        elif account['avatar'] == "5":
            avatar = url_for('static', filename='img/avatar/5.png')
        elif account['avatar'] == "6":
            avatar = url_for('static', filename='img/avatar/6.png')
        elif account['avatar'] == "7":
            avatar = url_for('static', filename='img/avatar/7.png')
        elif account['avatar'] == "8":
            avatar = url_for('static', filename='img/avatar/8.png')
        elif account['avatar'] == "9":
            avatar = url_for('static', filename='img/avatar/9.png')
        elif account['avatar'] == "10":
            avatar = url_for('static', filename='img/avatar/10.png')
        elif account['avatar'] == "11":
            avatar = url_for('static', filename='img/avatar/11.png')
        elif account['avatar'] == "12":
            avatar = url_for('static', filename='img/avatar/12.png')
        elif account['avatar'] == "13":
            avatar = url_for('static', filename='img/avatar/13.png')
        elif account['avatar'] == "14":
            avatar = url_for('static', filename='img/avatar/14.png')
        elif account['avatar'] == "15":
            avatar = url_for('static', filename='img/avatar/15.png')
        elif account['avatar'] == "16":
            avatar = url_for('static', filename='img/avatar/16.png')
        else:
            avatar = url_for('static', filename='img/avatar/default.png')
        return render_template('profile_a.html', account=account, foto=avatar, title='Mi
Perfil')
    # Si no esta logueado, redirige al acceso
    return redirect(url_for('alumni.login_a'))

```

```
# Editar Alumnado
@alumns.route('/alumn/edit/<alumn_id>')
def edit_alumn(alumn_id):
    if 'loggedin' in session:
        # Buscamos al alumno por su id
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM alumn WHERE alumn_id = %s", [alumn_id])
        account = cursor.fetchone()
        return render_template('edit_a.html', alumn=account, title='Modificar Datos')
    return redirect(url_for('alumns.login_a'))

# Actualizar Alumnado
@alumns.route('/update/<alumn_id>', methods=['POST'])
def update_alumn(alumn_id):
    if request.method == 'POST':
        username = request.form['username']
        fullname = request.form['fullname']
        email = request.form['email']
        # El alumno actualiza sus datos
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
        UPDATE alumn
        SET username = %s,
            fullname = %s,
            email = %s
        WHERE alumn_id = %s
        """, (username, fullname, email, alumn_id))
        mysql.connection.commit()
        flash('Tus datos han sido actualizados correctamente.', 'success')
        return redirect(url_for('alumns.alumn_profile'))
    return render_template('edit_a.html', title='Modificar Datos')

# Selecciono elegir avatar
@alumns.route('/alumn/update/<alumn_id>')
def edit_avatar(alumn_id):
    if 'loggedin' in session:
        # Buscamos al alumno por su id
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM alumn WHERE alumn_id = %s", [alumn_id])
        account = cursor.fetchone()
        return render_template('avatar_a.html', alumn=account, title='Elegir Avatar')
    return redirect(url_for('alumns.login_a'))

# Subir Avatar
@alumns.route('/update/avatar/<alumn_id>', methods=['GET', 'POST'])
def update_avatar(alumn_id):
    if request.method == 'POST':
        avatar = request.form['avatar']
        # Selecciona el avatar nuevo y lo actualiza
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('UPDATE alumn SET avatar = %s WHERE alumn_id = %s', (avatar,
alumn_id))
        mysql.connection.commit()
        flash('Tu avatar ha sido actualizado correctamente.', 'success')
        return redirect(url_for('alumns.alumn_profile'))
    return render_template('avatar_a.html', title='Elegir Avatar')

# -----
# Juegos Alumnado
@alumns.route('/alumn/games')
def alumn_games():
    if 'loggedin' in session:
        # Se muestran todos los juegos existentes
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM game")
        account = cursor.fetchall()
        return render_template('games_a.html', title='Juegos', juegos=account)
    return redirect(url_for('alumns.login_a'))
```



```
# Jugar Juego
@alumns.route('/alumn/game/<game_id>')
def play_gameA(game_id):
    if 'loggedin' in session:
        # Se pulsa en jugar, en funcion del id del juego, sale un juego u otro
        cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cur.execute("SELECT * FROM alumn WHERE username = %s", [session['username']])
        acc = cur.fetchone()
        # Se pasa el nombre de usuario
        if game_id == "1":
            return render_template('game1/alumn/index.html', alumn=acc)
        elif game_id == "2":
            return render_template('game2/alumn/PInicio.html', alumn=acc)
        elif game_id == "3":
            return render_template('game3/alumn/PInicio.html', alumn=acc)

# Recibir Puntuaciones (API)
@alumns.route('/alumn/api', methods=["POST", "GET"])
def createResult():
    if 'loggedin' in session:
        # Muestra el array de valores recibidos de la API
        print(request.json)
        # Recoge los valores recibidos por la API
        _name_game = request.json["name_game"]
        _level = request.json["level"]
        _points = request.json["score"]
        # Se insertan estos, en la tabla result
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
INSERT INTO result (name_game, level, points, user_name)
VALUES (%s, %s, %s, %s)""",
        (_name_game, _level, _points, [session['username']]))
        mysql.connection.commit()
        # Si todo ha ido ok, se muestra por consola
        return jsonify('Resultados recibidos correctamente')

@alumns.route('/alumn/game/2')
def index_game2():
    # Redireccionamos a la pag de inicio
    return render_template('game2/alumn/PInicio.html')

@alumns.route('/alumn/game/3')
def index_game3():
    # Redireccionamos a la pag de inicio
    return render_template('game3/alumn/PInicio.html')

@alumns.route('/alumn/game/2/levels')
def level_game2():
    # Redireccionamos a la pag de niveles
    return render_template('game2/alumn/Niveles.html')

@alumns.route('/alumn/game/3/levels')
def level_game3():
    # Redireccionamos a la pag de niveles
    return render_template('game3/alumn/Niveles.html')

@alumns.route('/alumn/game/2/level')
def lvl_game2():
    return render_template('game2/alumn/nivelgene.html')

@alumns.route('/alumn/game/3/level')
def lvl_game3():
    return render_template('game3/alumn/nivelgene.html')

@alumns.route('/alumn/game/2/gameover')
def gameover2():
    # Redireccionamos a la pag de fin de juego
    return render_template('game2/alumn/gameover.html')

@alumns.route('/alumn/game/3/gameover')
def gameover3():
    # Redireccionamos a la pag de fin de juego
    return render_template('game3/alumn/gameover.html')

# -----
```

```
# Resultados de Alumnado
@alumns.route('/alumn/results')
def alumn_results():
    if 'loggedin' in session: # Si no esta logueado no hace nada
        usuario = session['username']
        # Puntuaciones totales Game 1
        cursor1 = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        game1 = "Notas Musicales"
        cursor1.execute("""
        SELECT SUM(IF(name_game = %s AND user_name = %s, points, 0)) total
        FROM result
        """, (game1, usuario))
        account1 = cursor1.fetchone()

        # Puntuaciones totales Game 2
        cursor2 = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        game2 = "Equivalencias Musicales"
        cursor2.execute("""
        SELECT SUM(IF(name_game = %s AND user_name = %s, points, 0)) total
        FROM result
        """, (game2, usuario))
        account2 = cursor2.fetchone()

        # Puntuaciones totales Game 3
        cursor3 = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        game3 = "Atrapa al Intruso"
        cursor3.execute("""
        SELECT SUM(IF(name_game = %s AND user_name = %s, points, 0)) total
        FROM result
        """, (game3, usuario))
        account3 = cursor3.fetchone()

    return render_template('results_a.html', title='Mis Puntuaciones', puntos1=account1,
        puntos2=account2, puntos3=account3)
# -----
# Logros Alumnado
@alumns.route('/alumn/logros')
def alumn_logros():
    if 'loggedin' in session:
        # Actualizar Puntuaciones Totales del Alumno
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
        SELECT SUM(IF(user_name = %s, points, 0)) total
        FROM result
        """, [session['username']])
        acc = cursor.fetchone()
        # Dependiendo del total de puntos muestra un logro u otro (acumulables)
        if acc['total'] == 0:
            logro = url_for('static', filename='img/logros/0.png')
        elif acc['total'] > 0 and acc['total'] < 51:
            logro = url_for('static', filename='img/logros/1.png')
        elif acc['total'] > 50 and acc['total'] < 201:
            logro = url_for('static', filename='img/logros/2.png')
        elif acc['total'] > 200 and acc['total'] < 501:
            logro = url_for('static', filename='img/logros/3.png')
        elif acc['total'] > 500 and acc['total'] < 1001:
            logro = url_for('static', filename='img/logros/4.png')
        elif acc['total'] > 1000 and acc['total'] < 2001:
            logro = url_for('static', filename='img/logros/5.png')
        elif acc['total'] > 2000 and acc['total'] < 5001:
            logro = url_for('static', filename='img/logros/6.png')
        elif acc['total'] > 5000 and acc['total'] < 10001:
            logro = url_for('static', filename='img/logros/7.png')
        elif acc['total'] > 10000:
            logro = url_for('static', filename='img/logros/8.png')

    return render_template('logros_a.html', title='Mis Logros', logro=logro)
# -----
```

```
# Salida Alumnado
@alumnos.route('/logout')
def logout():
    # Eliminamos el dato de sesion, para que el alumno salga
    session.pop('logged_in', None)
    session.pop('id', None)
    session.pop('username', None)
    # Redireccionamos al inicio
    return redirect(url_for('main.inicio'))
```

*Archivo 3. views.py – alumnos*

## 4.2. Módulo [teachers]

El archivo “views.py”, es el encargado de gestionar todas las vistas/rutas que son accesibles por los profesores.

```
import os
from flask import Flask, render_template, request, redirect, url_for, flash, session,
Blueprint
from musicgames import mysql
import MySQLdb.cursors
import re

teachers = Blueprint('teachers', __name__,
template_folder='../templates/teacher/templates')

# Registro Porfesorado
@teachers.route('/register_t', methods=['GET', 'POST'])
def register_t():
    if request.method == 'POST' and 'fullname' in request.form and 'email' in
request.form and 'username' in request.form and 'password' in request.form and 'dni' in
request.form and 'phone' in request.form and 'gender' in request.form:
        # Crear un acceso facil para las variables
        username = request.form['username']
        password = request.form['password']
        fullname = request.form['fullname']
        dni = request.form['dni']
        email = request.form['email']
        phone = request.form['phone']
        gender = request.form['gender']
        # Chequea si la cuenta existe ya
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM teacher WHERE username = %s", [username])
        account = cursor.fetchone()
        # Si la cuenta existe muestra un error de validacion
        if account:
            flash('Cuenta ya existente, elija otro nombre de usuario', 'danger')
        elif not re.match(r'^[^\s]+@[^\s]+\.[^\s]+', email):
            flash('Email no valido.', 'danger')
        elif not re.match(r'[A-Za-z0-9]+', username):
            flash('El numero de telefono solo debe contener numeros.', 'danger')
        elif not re.match(r'[0-9]+', phone):
            flash('El numero de telefono solo debe contener numeros.', 'danger')
        elif not username or not password or not fullname or not dni or not email or not
phone or not gender:
            flash('Por favor, rellene el formulario.', 'danger')
        else:
            # Si la cuenta no existe y el form es valido, se agrega el nuevo profesor
            cursor.execute('INSERT INTO teacher (username, password, fullname, dni,
email, phone, gender) VALUES (%s, %s, %s, %s, %s, %s, %s)', (username, password,
fullname, dni, email, phone, gender))
            mysql.connection.commit()
            flash('Tu cuenta ha sido creada. Ahora puedes iniciar sesion', 'success')
            return redirect(url_for('teachers.login_t'))
    elif request.method == 'POST':
        flash('Por favor, rellene el formulario.', 'danger')
    return render_template('register_t.html', title='Registro')
```

```
# Acceso Profesorado
@teachers.route('/login_t/', methods=['GET', 'POST'])
def login_t():
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM teacher WHERE username = %s AND password = %s',
            (username, password))
        account = cursor.fetchone()
        if account:
            # Crear sesion dato, podemos acceder a este dato por otras rutas
            session['logged_in'] = True
            session['teacher_id'] = account['teacher_id']
            session['username'] = account['username']
            # Se actualiza la fecha de cuando inicia sesion
            cursor.execute("""
                UPDATE teacher
                SET login_date = curdate()
                WHERE username = %s
            """, [username])
            mysql.connection.commit()
            # Redireccionamos al inicio
            return redirect(url_for('main.home_t'))
        else:
            # Si la cuenta no existe o usuario/clave son erroneos
            flash('Inicio de sesion fallido. Por favor, compruebe su usuario y clave',
                'danger')
            return redirect(url_for('teachers.login_t'))
    return render_template('login_t.html', title='Acceso')

# Perfil Profesorado
@teachers.route('/teacher/profile')
def teacher_profile():
    # Chequeamos si esta logueado
    if 'logged_in' in session:
        # Necesitamos toda la informacion de la cuenta del profesor para poder mostrarla
        # en la pagina de perfil
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM teacher WHERE teacher_id = %s',
            [session['teacher_id']])
        account = cursor.fetchone()
        # Profesorado logueado redirigir al perfil
        if account['gender'] == 'F':
            photo = url_for('static', filename='img/profile/teacher_female.png')
        else:
            photo = url_for('static', filename='img/profile/teacher_male.jpg')
        return render_template('profile_t.html', account=account, photo=photo, title="Mi Perfil")
    # Profesorado no esta logueado, redirigir al login
    return redirect(url_for('teachers.login_t'))

# Editar Profesorado
@teachers.route('/teacher/edit/<teacher_id>')
def edit_teacher(teacher_id):
    if 'logged_in' in session:
        # Buscamos al profesor por su id
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM teacher WHERE teacher_id = %s", [teacher_id])
        account = cursor.fetchone()
        return render_template('edit_t.html', teacher=account, title='Modificar datos')
    return redirect(url_for('teachers.login_t'))

# Actualizar Profesorado
@teachers.route('/update_t/<teacher_id>', methods=['POST'])
def update_teacher(teacher_id):
    if request.method == 'POST':
        username = request.form['username']
        fullname = request.form['fullname']
        email = request.form['email']
        phone = request.form['phone']
        # El profesor actualiza sus datos
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
            UPDATE teacher
            SET username = %s,
```

```

        fullname = %s,
        email = %s,
        phone = %s
        WHERE teacher_id = %s
        """, (username, fullname, email, phone, teacher_id))
    mysql.connection.commit()
    flash('Tus datos han sido actualizados correctamente.', 'success')
    return redirect(url_for('teachers.teacher_profile'))
    return render_template('edit_t.html', title='Modificar Datos')

# -----
# Salida Profesorado
@teachers.route('/teacher/logout')
def logout_t():
    # Eliminamos el dato de sesion, para que el profesor salga
    session.pop('logged_in', None)
    session.pop('id', None)
    session.pop('username', None)
    # Redireccionamos al inicio
    return redirect(url_for('main.inicio'))

```

*Archivo 4. views.py – teachers*

## 4.3. Módulo [games]

El archivo “views.py”, es el encargado de gestionar todas las vistas/rutas que son necesarias para el acceso de los juegos.

```

import os
from flask import Flask, render_template, request, redirect, url_for, flash, session,
Blueprint
from musicgames import mysql
import MySQLdb.cursors

juego = Blueprint('juego', __name__, template_folder='../game')

# Juegos
@juego.route('/game')
def index_game():
    # Se muestran todos los juegos disponibles
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute("SELECT * FROM game")
    account = cursor.fetchall()
    return render_template('games.html', title='Juegos', games=account)

# Jugar Juego
@juego.route('/game/<game_id>')
def play_game(game_id):
    # Se accede a cada juego por su id
    if game_id == "1":
        return render_template('game1/user/index.html')
    elif game_id == "2":
        return render_template('game2/user/PInicio.html')
    elif game_id == "3":
        return render_template('game3/user/PInicio.html')

@juego.route('/game/2')
def index_game2():
    # Redireccionamos a la pag de inicio
    return render_template('game2/user/PInicio.html')

@juego.route('/game/3')
def index_game3():
    # Redireccionamos a la pag de inicio
    return render_template('game3/user/PInicio.html')

@juego.route('/game/2/levels')
def level_game2():
    # Redireccionamos a la pag de niveles
    return render_template('game2/user/Niveles.html')

```

```
@juego.route('/game/3/levels')
def level_game3():
    # Redireccionamos a la pag de niveles
    return render_template('game3/user/Niveles.html')

@juego.route('/game/2/level')
def lvl_game2():
    return render_template('game2/user/nivelgene.html')

@juego.route('/game/3/level')
def lvl_game3():
    return render_template('game3/user/nivelgene.html')

@juego.route('/game/2/gameover')
def gameover2():
    # Redireccionamos a la pag de fin de juego
    return render_template('game2/user/gameover.html')

@juego.route('/game/3/gameover')
def gameover3():
    # Redireccionamos a la pag de fin de juego
    return render_template('game3/user/gameover.html')
```

*Archivo 5. views.py – games*

## 4.4. Módulo [grupos]

El archivo “views.py”, es el encargado de gestionar todas las vistas/rutas que son necesarias para el acceso a los grupos.

```
import os
from flask import Flask, render_template, request, redirect, url_for, flash, session,
Blueprint
from musicgames import mysql
import MySQLdb.cursors

grupos = Blueprint('grupos', __name__, template_folder='../templates/grupo/templates')

# Selecciono Crear Grupo (Teacher)
@grupos.route('/group/create')
def create_grupo():
    if 'loggedin' in session:
        # Buscamos al profesor por su id
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM teacher WHERE teacher_id = %s",
[session['teacher_id']])
        account = cursor.fetchone()
        return render_template('create.html', teacher=account, title='Crear Grupo')
    return redirect(url_for('teachers.login_t'))

# Creo Grupo (Teacher)
@grupos.route('/group/create/add', methods=['GET', 'POST'])
def add_grupo():
    if 'loggedin' in session:
        cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cur.execute("SELECT * FROM teacher WHERE teacher_id = %s",
[session['teacher_id']])
        acc = cur.fetchone()
        if request.method == 'POST' and 'name' in request.form and 'classroom' in
request.form:
            # Se crea un acceso facil para las variables
            name = request.form['name']
            classroom = request.form['classroom']
            # Comprueba si el grupo existe
            cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
            cursor.execute("SELECT * FROM grupo WHERE name = %s", [name])
            account = cursor.fetchone()
            # Si el grupo existe muestra un error de validacion
            if account:
                flash('Grupo ya existente!', 'danger')
                return redirect(url_for('teachers.teacher_profile'))
            elif not name or not classroom:
                flash('Por favor, rellene el formulario.', 'danger')
```

```

        else:
            # Si el grupo no existe, pasamos a crearlo
            cursor.execute('INSERT INTO grupo (name, classroom) VALUES (%s, %s)',
                (name, classroom))
            mysql.connection.commit()
            flash('Confirme su grupo, por favor', 'primary')
            return render_template('confirm.html', teacher=acc, title='Confirmar
Grupo')
        elif request.method == 'POST':
            flash('Por favor, rellene el formulario.', 'danger')
            return render_template('create.html', teacher=acc, title='Crear Grupo')

# Confirmar grupo (Teacher)
@grupos.route('/group/create/confirm', methods=['GET', 'POST'])
def confirm_group():
    if 'loggedin' in session:
        if request.method == 'POST' and 'id_teacher' in request.form and 'name' in
request.form:
            id_teacher = request.form['id_teacher']
            name = request.form['name']
            cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
            cursor.execute("SELECT * FROM grupoteacher WHERE name_grupo = %s", [name])
            account = cursor.fetchone()
            # Si el grupo ya esta asociado muestra un error de validacion
            if account:
                flash('El grupo ya pertenece a otro/a profesor/a.', 'danger')
                return redirect(url_for('teachers.teacher_profile'))
            else:
                # Al no estar asociado, lo asociamos al profesor
                cursor.execute('INSERT INTO grupoteacher (id_teacher, name_grupo) VALUES
(%s, %s)', (id_teacher, name))
                mysql.connection.commit()
                flash('Se ha asociado correctamente a su grupo.', 'success')
                return redirect(url_for('teachers.teacher_profile'))
        elif request.method == 'POST':
            flash('Por favor, rellene el formulario.', 'danger')
            return render_template('confirm.html', title='Confirmar Grupo')

# Selecciono Gestionar Grupo (Teacher)
@grupos.route('/teacher/group')
def manage_grupo():
    if 'loggedin' in session:
        # Se buscan todos los grupos pertenecientes a cada profesor
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
SELECT grupo.name, grupo.classroom, teacher.fullname, teacher.teacher_id
FROM teacher
INNER JOIN grupoteacher ON teacher.teacher_id = grupoteacher.id_teacher
INNER JOIN grupo ON grupoteacher.name_grupo = grupo.name""")
        account = cursor.fetchall()
        return render_template('manage.html', manages=account,
teacher=session['teacher_id'], title='Mis grupos')
    return redirect(url_for('teachers.login_t'))

# Ver Grupo (Teacher)
@grupos.route('/group/<name>')
def show_grupo(name):
    if 'loggedin' in session:
        cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cur.execute("SELECT * FROM grupo WHERE name = %s", [name])
        acc = cur.fetchone()
        # Buscamos todos los alumnos pertenecientes al grupo dirigido por el profesor
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
SELECT alumn.fullname, alumn.username, grupo.name,
grupo.grupo_id, grupo.classroom
FROM grupo
INNER JOIN grupoalumn ON grupo.name = grupoalumn.name_grupo
INNER JOIN alumn ON grupoalumn.id_alumn = alumn.alumn_id""")
        account = cursor.fetchall()
        return render_template('grupo.html', manages=account, grupo=acc, title='Info
Grupo')
    return redirect(url_for('teachers.login_t'))

```

```
# Ver puntuaciones alumnos (Teacher)
@grupos.route('/group/alumn/<username>')
def show_results(username):
    # Resultados Game 1
    cursor1 = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    game1 = "Notas Musicales"
    cursor1.execute("""
    SELECT SUM(IF(name_game = %s AND user_name = %s, points, 0)) total
    FROM result
    """, (game1, username))
    account1 = cursor1.fetchone()

    # Resultados Game 2
    cursor2 = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    game2 = "Equivalencias Musicales"
    cursor2.execute("""
    SELECT SUM(IF(name_game = %s AND user_name = %s, points, 0)) total
    FROM result
    """, (game2, username))
    account2 = cursor2.fetchone()

    # Resultados Game 3
    cursor3 = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    game3 = "Atrapa al Intruso"
    cursor3.execute("""
    SELECT SUM(IF(name_game = %s AND user_name = %s, points, 0)) total
    FROM result
    """, (game3, username))
    account3 = cursor3.fetchone()

    return render_template('results_alumn.html', title='Puntuaciones',
    username=username, puntos1=account1, puntos2=account2, puntos3=account3)

# Editar aula del grupo (Teacher)
@grupos.route('/group/<name>/edit')
def edit_group(name):
    if 'loggedin' in session:
        # Buscamos al grupo por su nombre
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM grupo WHERE name = %s", [name])
        account = cursor.fetchone()
        return render_template('edit_g.html', grupo=account, title='Modificar Aula')
    return redirect(url_for('teachers.login_t'))

# Actualizar aula (Teacher)
@grupos.route('/group/update/<name>', methods=['POST'])
def update_group(name):
    if request.method == 'POST':
        classroom = request.form['classroom']
        # Se actualiza el aula del grupo
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("UPDATE grupo SET classroom = %s WHERE name = %s", (classroom,
name))
        mysql.connection.commit()
        flash('Aula actualizada correctamente.', 'success')
        return redirect(url_for('grupos.manage_grupo'))
    return render_template('edit_g.html', title='Modificar Aula')

# Eliminación del grupo (Teacher)
@grupos.route('/group/<name>/select')
def supr_group(name):
    if 'loggedin' in session:
        # Seleccionamos el grupo por su nombre
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM grupo WHERE name = %s", [name])
        account = cursor.fetchone()
        return render_template('delete_g.html', grupo=account, title='Eliminar Grupo')
    return redirect(url_for('teachers.login_t'))

# Eliminar Grupo (Teacher)
@grupos.route('/group/delete/<name>', methods=['POST'])
def delete_group(name):
    if request.method == 'POST':
        name = request.form['name']
        # Eliminamos el grupo por su nombre
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("DELETE FROM grupo WHERE name = %s", [name])
```



```

mysql.connection.commit()
flash('Grupo eliminado correctamente.', 'success')
return redirect(url_for('grupos.manage_grupo'))
return render_template('delete_g.html', title='Eliminar Grupo')

# -----
# Selecciono Elegir Grupo (Alumn)
@grupos.route('/group/choise')
def select_choise_grupo():
    if 'loggedin' in session:
        # Se busca al alumno por su id
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM alumn WHERE alumn_id = %s", [session['alumn_id']])
        account = cursor.fetchone()
        return render_template('choise.html', alumn=session['alumn_id'], title='Elegir
grupo')
    return redirect(url_for('alumni.login_a'))

# Asociar grupo a alumno/a (Alumn)
@grupos.route('/group/choise/mygroup', methods=['GET', 'POST'])
def choise_grupo():
    if 'loggedin' in session:
        if request.method == 'POST' and 'id_alumn' in request.form and 'name_grupo' in
request.form:
            id_alumn = request.form['id_alumn']
            name_grupo = request.form['name_grupo']
            # Con el id del alumno y el nombre del grupo
            cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
            cursor.execute("SELECT * FROM grupoalumn WHERE id_alumn = %s", [id_alumn])
            account = cursor.fetchone()
            # Si el alumnado ya ha elegido muestra un error de validacion
            if account:
                flash('Ya pertenece a un grupo.', 'danger')
                return redirect(url_for('alumni.alumn_profile'))
            else:
                # Si el alumno no ha elegido grupo, se le asocia el grupo ingresado
                cursor.execute('INSERT INTO grupoalumn (id_alumn, name_grupo) VALUES
(%, %s)', (id_alumn, name_grupo))
                mysql.connection.commit()
                flash('Se ha asociado correctamente a su grupo.', 'success')
                return redirect(url_for('alumni.alumn_profile'))
            elif request.method == 'POST':
                flash('Por favor, rellene el formulario.', 'danger')
                return render_template('choise.html', title='Elegir Grupo')

# Selecciono Mi Grupo (Alumn)
@grupos.route('/alumn/group')
def see_grupo():
    if 'loggedin' in session:
        # Se busca el grupo del alumno
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
SELECT grupo.name, grupo.classroom, alumn.fullname, alumn.alumn_id
FROM alumn
INNER JOIN grupoalumn ON alumn.alumn_id = grupoalumn.id_alumn
INNER JOIN grupo ON grupoalumn.name_grupo = grupo.name""")
        account = cursor.fetchall()
        return render_template('see.html', manages=account, alumn=session['alumn_id'],
title='Mi grupo')
    return redirect(url_for('alumni.login_a'))

# Ver Profesor (Alumn)
@grupos.route('/alumn/group/<name>/teacher')
def grupo_teacher(name):
    if 'loggedin' in session:
        # Se busca por el nombre del grupo, el profesor que lo imparte
        cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cur.execute("SELECT * FROM grupo WHERE name = %s", [name])
        acc = cur.fetchone()
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
SELECT teacher.fullname, teacher.email, grupo.name,
grupo.grupo_id, grupo.classroom
FROM grupo
INNER JOIN grupoteacher ON grupo.name = grupoteacher.name_grupo
INNER JOIN teacher ON grupoteacher.id_teacher = teacher.teacher_id""")
        account = cursor.fetchall()

```

```

        return render_template('grupo_teacher.html', manages=account, grupo=acc,
title='Profesor Grupo')
        return redirect(url_for('alumni.login_a'))

```

*Archivo 6. views.py – grupos*

## 4.5. Módulo [main]

El archivo “views.py”, es el encargado de gestionar todas las vistas/rutas que son necesarias para el resto de las opciones.

```

import os
from flask import Flask, render_template, request, redirect, url_for, flash, session,
Blueprint
from musicgames import mysql
import MySQLdb.cursors

main = Blueprint('main', __name__, template_folder='../templates/main/templates')

# Inicio
@main.route('/')
def inicio():
    # Pagina inicial
    return render_template('inicio.html', title='Inicio')

# Inicio No Registrados
@main.route('/index')
def index():
    return render_template('index.html', title='Inicio')

# Inicio Alumnado
@main.route('/alumn')
def home():
    # Chequear si esta logueado
    if 'loggedin' in session:
        # Alumno logueado redirigir al inicio
        return render_template('index_a.html', username=session['username'],
title=session['username'])
    # Alumno no esta logueado, redirigir al login
    return redirect(url_for('alumni.login_a'))

# Acceso Privado Profesorado
@main.route('/academy')
def teacher_pass():
    return render_template('academy.html', title='Profesorado')

# Permiso Denegado
@main.route('/not_permit')
def not_permit():
    return render_template('denegado.html', title='No Permitido')

# Inicio Profesorado
@main.route('/teacher')
def home_t():
    # Chequear si esta logueado
    if 'loggedin' in session:
        # Consulta para mostrar los grupos de todos los profesores
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
SELECT grupo.name, grupo.grupo_id, grupo.classroom, teacher.fullname
FROM teacher
INNER JOIN grupoteacher ON teacher.teacher_id = grupoteacher.id_teacher
INNER JOIN grupo ON grupoteacher.name_grupo = grupo.name""")
        account = cursor.fetchall()
        return render_template('index_t.html', manages=account,
title=session['username'])
    # Profesorado no esta logueado, redirigir al login
    return redirect(url_for('teachers.login_t'))

# Calendario -----
@main.route('/teacher/calendar')
def calendar_t():
    return render_template('calendar_t.html', title='Calendario')

```

```
@main.route('/alumn/calendar')
def calendar_a():
    return render_template('calendar_a.html', title='Calendario')

@main.route('/calendar')
def calendar():
    return render_template('calendar.html', title='Calendario')

# Quienes somos -----
@main.route('/about')
def about_us():
    return render_template('about_us.html', title='Equipo')

@main.route('/alumn/about')
def about_us_a():
    return render_template('about_us_a.html', title='Equipo')

@main.route('/teacher/about')
def about_us_t():
    return render_template('about_us_t.html', title='Equipo')

# Contacto -----
@main.route('/contact')
def contact():
    return render_template('contact.html', title='Contacto')

@main.route('/alumn/contact')
def contact_a():
    return render_template('contact_a.html', title='Contacto')

@main.route('/teacher/contact')
def contact_t():
    return render_template('contact_t.html', title='Contacto')

# Noticias -----
"""@main.route('/news')
def news():
    return render_template('news.html')"""
```

*Archivo 7. views.py – main*

Como podemos observar existen tres rutas y definición de función, prácticamente iguales cada separación, esto se hace para las distintas sesiones en las que se divide nuestro proyecto. Por ej., para el calendario, una ruta es para “Usuarios No Registrados”, otra para el “Alumnado” y otra para el “Profesorado”.

## 4.6. Fichero [static]

En este fichero nos encontramos con otros dos grandes ficheros como son [css] e [img].

### 4.6.1. Fichero [css]

En este apartado tenemos seis archivos de estilos CSS, los cuáles paso a citar:

```
#avatar1, #avatar2, #avatar3, #avatar4, #avatar5, #avatar6, #avatar7, #avatar8 {
    display: none;
}

#avatar9, #avatar10, #avatar11, #avatar12, #avatar13, #avatar14, #avatar15, #avatar16 {
    display: none;
}

label img.oculta {
    display: none;
}

input[name='avatar']:checked + label img {
    display: none;
}

input[name='avatar']:checked + label img.oculta {
    display: inline;
}
```

*Archivo 8. avatar.css – [static]*

Este archivo, se encarga de dar estilo a los avatares del alumnado.

```
div.title {
    font: x-large Verdana, Arial, Helvetica, sans-serif;
    text-align: center;
    height: 40px;
    background-color: white;
    color: #444444;
}

/* Styling for the footer */
div.footer {
    font: small Verdana, Arial, Helvetica, sans-serif;
    text-align: center;
}

/* Styling for the overall table */
table {
    font: 100% Verdana, Arial, Helvetica, sans-serif;
    table-layout: fixed;
    border-collapse: collapse;
    width: 100%;
}

/* Styling for the column headers (days of the week) */
th {
    padding: 0 0.5em;
    text-align: center;
    background-color: gray;
    color: white;
}

/* Styling for the individual cells (days) */
td {
    font-size: medium;
    padding: 0.25em 0.25em;
    width: 14%;
    height: 80px;
    text-align: left;
    vertical-align: top;
}

/* Styling for the date numbers */
.date {
    font-size: large;
    padding: 0.25em 0.25em;
    text-align: left;
    vertical-align: top;
}
```

```
/* Class for individual days (coming in future release) */
.sun {
    color:red;
}
/* Hide the month element (coming in future release) */
th.month {
    visibility: hidden;
    display:none;
}
Archivo 9. calendar.css – [static]
```

Este archivo, se encarga de dar estilo al calendario que será visible para todos los usuarios.

```
* {
    margin: 0;
    padding: 0;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

a {
    text-decoration: none;
}

.contenedor {
    max-width: 1200px;
    margin: 50px auto;
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
    justify-content: center;
}

.contenedor_tarjeta {
    margin: 20px;
}

.contenedor_tarjeta a {
    display: inline-block;
}

.contenedor_tarjeta:hover figure {
    transform: perspective(600px) rotateY(180deg);
    -webkit-box-shadow: 0px 2px 10px 2px rgba(0,0,0,0.25);
    -moz-box-shadow: 0px 2px 10px 2px rgba(0,0,0,0.25);
    box-shadow: 0px 2px 10px 2px rgba(0,0,0,0.25);
}

figure {
    width: 300px;
    height: 350px;
    margin: 0;
    position: relative;
    transition: all ease .5s;
    transform-style: preserve-3d;
    /* Nota:
        Establecemos que la imagen tendra una rotacion de 0grados al inicio
        porque si no
        ponemos esta propiedad nos da un poco de problemas en algunos
        navegadores al pasar el cursor.
    */
    transform: perspective(600px) rotateY(0deg);
}

figure .frontal,
figure .trasera {
    width: 100%;
    height: 100%;
    border-radius: 5px;
    transition: all ease .5s;
}

figure .frontal {
    display: block;
    background: #000;
```

```

/* Podemos ocultar la imagen al da la vuelta si lo queremos */
/*backface-visibility: hidden;*/
}

figure .trasera {
    position: absolute;
    top: 0;
    padding: 20px;
    color: #000;
    transform: perspective(600px) rotateY(180deg);
    backface-visibility: hidden;
    overflow: auto;
}

.contenedor_tarjeta:nth-child(1) figure .trasera {
    background: rgba(246,246,246,0.75);
    background: -moz-linear-gradient(top, rgba(246,246,246,0.75) 0%,
    rgba(242,207,153,0.75) 0%, rgba(207,172,51,1) 100%);
    background: -webkit-gradient(left top, left bottom, color-stop(0%,
    rgba(246,246,246,0.75)), color-stop(0%, rgba(242,207,153,0.75)), color-stop(100%,
    rgba(207,172,51,1)));
    background: -webkit-linear-gradient(top, rgba(246,246,246,0.75) 0%,
    rgba(242,207,153,0.75) 0%, rgba(207,172,51,1) 100%);
    background: -o-linear-gradient(top, rgba(246,246,246,0.75) 0%, rgba(242,207,153,0.75)
    0%, rgba(207,172,51,1) 100%);
    background: -ms-linear-gradient(top, rgba(246,246,246,0.75) 0%, rgba(242,207,153,0.75)
    0%, rgba(207,172,51,1) 100%);
    background: linear-gradient(to bottom, rgba(246,246,246,0.75) 0%,
    rgba(242,207,153,0.75) 0%, rgba(207,172,51,1) 100%);
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#f6f6f6',
    endColorstr='#cfac33', GradientType=0 );
}

.contenedor_tarjeta:nth-child(2) figure .trasera {
    background: rgba(246,246,246,0.75);
    background: -moz-linear-gradient(top, rgba(246,246,246,0.75) 0%,
    rgba(242,207,153,0.75) 0%, rgba(207,172,51,1) 100%);
    background: -webkit-gradient(left top, left bottom, color-stop(0%,
    rgba(246,246,246,0.75)), color-stop(0%, rgba(242,207,153,0.75)), color-stop(100%,
    rgba(207,172,51,1)));
    background: -webkit-linear-gradient(top, rgba(246,246,246,0.75) 0%,
    rgba(242,207,153,0.75) 0%, rgba(207,172,51,1) 100%);
    background: -o-linear-gradient(top, rgba(246,246,246,0.75) 0%, rgba(242,207,153,0.75)
    0%, rgba(207,172,51,1) 100%);
    background: -ms-linear-gradient(top, rgba(246,246,246,0.75) 0%, rgba(242,207,153,0.75)
    0%, rgba(207,172,51,1) 100%);
    background: linear-gradient(to bottom, rgba(246,246,246,0.75) 0%,
    rgba(242,207,153,0.75) 0%, rgba(207,172,51,1) 100%);
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#f6f6f6',
    endColorstr='#cfac33', GradientType=0 );
}

.contenedor_tarjeta:nth-child(3) figure .trasera {
    background: rgba(255,255,255,0.75);
    background: -moz-linear-gradient(top, rgba(255,255,255,0.75) 0%,
    rgba(242,207,153,0.75) 0%, rgba(240,204,170,1) 100%);
    background: -webkit-gradient(left top, left bottom, color-stop(0%,
    rgba(255,255,255,0.75)), color-stop(0%, rgba(242,207,153,0.75)), color-stop(100%,
    rgba(240,204,170,1)));
    background: -webkit-linear-gradient(top, rgba(255,255,255,0.75) 0%,
    rgba(242,207,153,0.75) 0%, rgba(240,204,170,1) 100%);
    background: -o-linear-gradient(top, rgba(255,255,255,0.75) 0%, rgba(242,207,153,0.75)
    0%, rgba(240,204,170,1) 100%);
    background: -ms-linear-gradient(top, rgba(255,255,255,0.75) 0%, rgba(242,207,153,0.75)
    0%, rgba(240,204,170,1) 100%);
    background: linear-gradient(to bottom, rgba(255,255,255,0.75) 0%,
    rgba(242,207,153,0.75) 0%, rgba(240,204,170,1) 100%);
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#ffffff',
    endColorstr='#f0ccaa', GradientType=0 );
}

figure .trasera .titulo {
    color: #000;
    font-weight: normal;
    margin-bottom: 20px;
    font-family: sans-serif;
}

```

```

    font-size: 24px;
}

figure .trasera hr {
    height: 2px;
    background: #000;
    border: none;
    margin-bottom: 20px;
    opacity: .5;
}

figure .trasera p {
    font-family: 'Open Sans', sans-serif;
    line-height: 22px;
    font-size: 14px;
}

@media screen and (max-width: 992px) {
    figure .trasera {
        backface-visibility: visible;
        transform: perspective(600px) rotateY(0deg);
    }

    .contenedor_tarjeta:hover figure {
        transform: perspective(600px) rotateY(0deg);
    }
}

```

*Archivo 10. style.css – [static]*

Este archivo, se encarga de dar estilo a la pantalla inicial que será visible para todos los usuarios.

```

body {
    /*background: #fafafa;*/
    background-image: url(../img/background/1.jpg);
    background-size: 100vw 100vh;
    background-attachment: fixed;
    color: #333333;
    margin-top: 5rem;
}

h1, h2, h3, h4, h5, h6 {
    color: #444444;
}

.bg-steel {
    background-color: #5f788a;
}

.site-header .navbar-nav .nav-link {
    color: #cbd5db;
}

.site-header .navbar-nav .nav-link:hover {
    color: #ffffff;
}

.site-header .navbar-nav .nav-link.active {
    font-weight: 500;
}

.content-section {
    background: #ffffff;
    padding: 10px 20px;
    border: 1px solid #dddddd;
    border-radius: 3px;
    margin-bottom: 20px;
}

.article-title {
    color: #444444;
}

a.article-title:hover {
    color: #428bca;
}

```

```

    text-decoration: none;
}

.article-content {
    white-space: pre-line;
}

.article-img {
    height: 65px;
    width: 65px;
    margin-right: 16px;
}

.article-metadata {
    padding-bottom: 1px;
    margin-bottom: 4px;
    border-bottom: 1px solid #e3e3e3
}

.article-metadata a:hover {
    color: #333;
    text-decoration: none;
}

.article-svg {
    width: 25px;
    height: 25px;
    vertical-align: middle;
}

.account-img {
    height: 125px;
    width: 125px;
    margin-right: 20px;
    margin-bottom: 16px;
}

.account-heading {
    font-size: 2.5rem;
}

```

*Archivo 11. main.css – [static]*

Este archivo, se encarga de dar estilo a todas las templates o plantillas que utilizan los “Usuarios No Registrados” y el “Alumnado”.

El archivo “main\_t.css” es exactamente igual que “main.css” tan solo quitamos en la propiedad de body, aquellas subpropiedades que hacen referencia al fondo de pantalla, quedando como:

```

body {
    background: #fafafa;
    /*background-image: url(../img/background/1.jpg);
    background-size: 100vw 100vh;
    background-attachment: fixed;*/
    color: #333333;
    margin-top: 5rem;
}

```

*Archivo 12. Propiedad body actualizada para “main\_t.css”*



Y por último se muestra el código que hace que nuestro proyecto sea responsive o adaptativo. Se ha aplicado para los principales dispositivos móviles del mercado.

```
/* APPLE */
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px) { /*iPad*/
    #navegador {
        float: left;
        width: 25%;
    }
}

@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px)
and (-webkit-min-device-pixel-ratio: 2) { /*iPad 3y4*/
    #navegador {
        float: left;
        width: 25%;
    }
}

@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px)
and (-webkit-min-device-pixel-ratio: 1) { /*iPad Mini*/
    #navegador {
        float: left;
        width: 25%;
    }
}

@media only screen
and (min-device-width : 375px)
and (max-device-width : 812px)
and (-webkit-device-pixel-ratio : 3) { /*iPhone X-Xs*/
    #navegador {
        float: left;
        width: 25%;
    }
}

@media only screen
and (device-width : 414px)
and (device-height : 896px)
and (-webkit-device-pixel-ratio : 2) { /*iPhone Xr*/
    #navegador {
        float: left;
        width: 25%;
    }
}

@media only screen
and (device-width : 414px)
and (device-height : 896px)
and (-webkit-device-pixel-ratio : 3) { /*iPhone XsMAX*/
    #navegador {
        float: left;
        width: 25%;
    }
}

@media only screen
and (min-device-width : 375px)
and (max-device-width : 667px) { /* iPhone */
    #navegador {
        float: left;
        width: 25%;
    }
}

@media only screen
and (min-device-width : 414px)
and (max-device-width : 736px) { /* iPhone Plus */
```

```
#navegador {
    float: left;
    width: 25%;
}

/* OTROS */
@media only screen
and (min-device-width: 320px)
and (max-device-width: 736px) {
    #navegador {
        float: left;
        width: 25%;
    }
}

@media only screen
and (min-device-width: 320px)
and (max-device-width: 480px) {
    #navegador {
        float: left;
        width: 25%;
    }
}

@media only screen
and (min-device-width: 481px)
and (max-device-width: 767px) {
    #navegador {
        float: left;
        width: 25%;
    }
}
```

*Archivo 13. resp.css – [statitc]*

#### 4.6.2. Fichero [img]

En este apartado se contienen todos los archivos imagen necesarios para nuestro proyecto.

### 4.7. Fichero [templates]

Contiene todas las templates o plantillas que dispone nuestro proyecto, en otras palabras, todo el código HTML, se encuentra en este fichero, a excepción de la template encargada de los juegos, que para simplicidad está separada de este fichero.

Cuando trabajamos con Flask, nuestras plantillas incorporan ya el motor de plantillas web, Jinja2, el cual se representa como:

{{ <expresión> }} → Cualquier expresión que necesitemos expresar

{% <declaración> %}

...

...

{% end<tipo de declaración> %} → Para realizar if, for, ...

{# <comentario> #} → Cualquier comentario que se quiera realizar

Una vez, explicado esto pasamos a mostrar cada una de las templates por sus módulos.

### 4.7.1. Módulo [base]

En este fichero encontramos nuestra base HTML, dividido en cuatro archivos. En primer lugar, mostramos el archivo base para los “Usuarios No Registrados”.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/calendar.css') }}">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/avatar.css') }}">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/main.css') }}">

    {% if title %}
      <title>MusicGames - {{ title }}</title>
    {% else %}
      <title>MusicGames</title>
    {% endif %}
  </head>
  <body>
    <header class="site-header">
      <nav class="navbar navbar-expand-md navbar-dark bg-steel fixed-top">
        <div class="container">
          <a class="navbar-brand mr-4" href="/">MusicGames</a>
          <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarToggle" aria-controls="navbarToggle" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarToggle">
            <div class="navbar-nav mr-auto">
              <a class="nav-item nav-link" href="{{ url_for('main.index')
}}>Inicio</a>
              <a class="nav-item nav-link" href="{{
url_for('juego.index_game') }}">Juegos</a>
            </div>
            <!-- Navbar Right Side -->
            <div class="navbar-nav">
              <a class="nav-item nav-link" href="{{ url_for('main.about_us')
}}>¿Quiénes somos?</a>
              <a class="nav-item nav-link" href="{{ url_for('alumnos.register_a')
}}>Registro</a>
              <a class="nav-item nav-link" href="{{ url_for('alumnos.login_a')
}}>Acceso</a>
            </div>
          </div>
        </nav>
      </header>
      <main role="main" class="container">
        <div class="row">
          <div class="col-md-8">
            {% with messages = get_flashed_messages(with_categories=true) %}
              {% if messages %}
                {% for category, message in messages %}
                  <div class="alert alert-{{ category }}">
                    {{ message }}
                  </div>
                {% endfor %}
              {% endif %}
            {% endwith %}
            {% block content %}{% endblock %}
          </div>
          <div class="col-md-4">
```

```

<div class="content-section">
  <h3>Más Opciones</h3>
  <p class='text-muted'>
    <ul class="list-group">
      <li class="list-group-item list-group-item-light">
        <a href="#">Noticias</a>
      </li>
      <li class="list-group-item list-group-item-light">
        <a href="{{ url_for('main.calendar') }}">Calendario</a>
      </li>
      <li class="list-group-item list-group-item-light">
        <a href="{{ url_for('main.contact') }}">Contacto</a>
      </li>
    </ul>
  </p>
</div>
</div>
</div>
</main>

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTElPi6jizo"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEeFF/nJGzIxFDs4fx0xIM+B07jRM"
crossorigin="anonymous"></script>
</body>
</html>

```

*Archivo 14. base.html – [templates-base]*

En head, añadimos nuestros archivos de estilo, en el header nuestra barra de navegación, y en el main añadimos nuestros mensajes flash, así como, el contenido de las demás templates con {% block content %}{% endblock %}. Finalmente, aparecen los scripts “\*.js” por si hiciera falta utilizarlos.

El resto de los archivos base, son utilizados para los demás accesos de usuarios, y su código es exactamente el mismo a excepción de la barra de navegación, que por ejemplo es distinta la de un alumno (“base\_a.html”) que la de un profesor (“base\_t.html”), lo único que cambian son las vistas a las que son redirigidos.

#### 4.7.2. Módulo [alumn]

Nos encontramos en primera instancia con el fichero [templates] que tiene todas las plantillas usadas por el alumnado y el archivo “avatares.html” el cual es llamado por el archivo “avatar\_a.html”, usado para editar el avatar.

```

{% extends "base/base_a.html" %}
{% block content %}
  <div class="content-section">
    <form method="post" action="{{ url_for('alumni.update_avatar',
alumn_id=alumno['alumn_id']) }}" enctype="multipart/form-data" autocomplete="off">
      <div class="form-row">
        <legend class="border-bottom mb-4"><b>Modificar Avatar</b></legend>
        {% include "alumn/avatares.html" %}
      </div>
      <button type="submit" class="btn btn-outline-success">Confirmar</button>
      <a class="btn btn-outline-secondary btn mt-1 mb-1" href="javascript:history.back(-
1);" title="Ir la página anterior">Volver</a>
    </form>
  </div>
{% endblock %}

```

*Archivo 15. avatar\_a.html – [templates-alumn]*

```
<fieldset class="form-group">
  <div class="row">
    <legend class="col-form-label col-sm-12 pt-0">Avatares Disponibles</legend><br><br>
    <div class="col-sm-12">
      <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar1"
value="1">
        <label class="form-check-label" for="avatar1">
          
          
        </label>
      </div>
      <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar2"
value="2">
        <label class="form-check-label" for="avatar2">
          
          
        </label>
      </div>
      <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar3"
value="3">
        <label class="form-check-label" for="avatar3">
          
          
        </label>
      </div>
      <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar4"
value="4">
        <label class="form-check-label" for="avatar4">
          
          
        </label>
      </div>
      <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar5"
value="5">
        <label class="form-check-label" for="avatar5">
          
          
        </label>
      </div>
      <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar6"
value="6">
        <label class="form-check-label" for="avatar6">
          
          
        </label>
      </div>
      <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar7"
value="7">
        <label class="form-check-label" for="avatar7">
          
          
        </label>
      </div>
      <div class="form-check form-check-inline">
```

```

        <input class="form-check-input" type="radio" name="avatar" id="avatar8"
value="8">
        <label class="form-check-label" for="avatar8">
            
            
        </label>
    </div>
    <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar9"
value="9">
        <label class="form-check-label" for="avatar9">
            
            
        </label>
    </div>
    <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar10"
value="10">
        <label class="form-check-label" for="avatar10">
            
            
        </label>
    </div>
    <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar11"
value="11">
        <label class="form-check-label" for="avatar11">
            
            
        </label>
    </div>
    <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar12"
value="12">
        <label class="form-check-label" for="avatar12">
            
            
        </label>
    </div>
    <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar13"
value="13">
        <label class="form-check-label" for="avatar13">
            
            
        </label>
    </div>
    <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar14"
value="14">
        <label class="form-check-label" for="avatar14">
            
            
        </label>
    </div>
    <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar15"
value="15">
        <label class="form-check-label" for="avatar15">
            
            
    
```

```

        </label>
    </div>
    <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="avatar" id="avatar16"
value="16">
        <label class="form-check-label" for="avatar16">
            
            
        </label>
    </div>
</div>
</div>
</div>
</fieldset>

```

*Archivo 16. avatares.html – [templates-alumn]*

A continuación, se observa el código para el inicio del alumnado.

```

{% extends "base/base_a.html" %}
{% block content %}
<div class="content-section">
    <div class="media">
        <div class="media-body">
            <legend class="border-bottom mb-4">¿Qué debes saber antes de jugar?</legend>
            <ol>
                <li>Si es la primera vez que entras en la plataforma, deberás esperar a que
el/la profesor/a te explique el funcionamiento</li><br>
                <li>Una vez sepas el correcto funcionamiento, podrás acceder a las distintas
opciones de la barra menú</li>
            </ol>
        </div>
    </div>
</div>
<div class="content-section">
    <div class="media">
        <div class="media-body">
            <legend class="border-bottom mb-4">Perfil</legend>
            <ol>
                <li>En "Detalles sobre tu cuenta", encontraras toda la información de registro a
excepción de la contraseña, que por motivos de seguridad no se mostrará</li><br>
                <li>Si deseas cambiar algún dato que no esté correcto, pulsa en modificar y
accederas al formulario para que modifiques los datos correctamente</li><br>
                <li>Si deseas cambiar la contraseña, pulsa en modificar y accederas al
formulario para que modifiques la contraseña correctamente</li>
            </ol>
        </div>
    </div>
</div>
<div class="content-section">
    <div class="media">
        <div class="media-body">
            <legend class="border-bottom mb-4">Juegos</legend>
            <ol>
                <li>Aquí podrás ver todos los juegos disponibles</li><br>
                <li>Si deseas jugar a un juego, pulsa en "Jugar"</li><br>
                <ul>
                    <li type="square">Si es la primera vez que juegas, entraras directamente al
juego</li><br>
                    <li type="square">Si ya has jugado, te dará opción de elegir el nivel</li>
                </ul>
            </ol>
        </div>
    </div>
</div>
<div class="content-section">
    <div class="media">
        <div class="media-body">
            <legend class="border-bottom mb-4">Puntuaciones</legend>
            <ol>
                <li>Si no has jugado a ningún juego no aparecerá ningún resultado</li><br>
                <li>Si has jugado a algún juego, las puntuaciones aparecerán por cada juego
jugado</li>
            </ol>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
</div>
<div class="content-section">
    <div class="media">
        <div class="media-body">
            <legend class="border-bottom mb-4">Logros</legend>
            <ol>
                <li>Si no has jugado a ningún juego no tendrás ningún logro activado</li><br>
                <li>Si has jugado a al menos a un juego, tendrás activado el logro "Primera
Vez"</li><br>
                <li>Conforme vayas jugando más, se irán activando los distintos logros</li>
            </ol>
        </div>
    </div>
</div>
{% endblock %}

```

*Archivo 17. index\_a.html – [templates-alumn]*

El siguiente código, muestra el perfil del alumnado.

```

{% extends "base/base_a.html" %}
{% block content %}
<div class="content-section">
    <div class="media">
        
        <div class="media-body">
            <legend class="border-bottom mb-4">Mis datos</legend>
            <h2 class="account-heading">{{ account['fullname'] }}</h2>
            <p class="text-secondary"><b>Usuario: </b>{{ account['username'] }}</p>
            <p class="text-secondary"><b>Año de Nacimiento: </b>{{ account['birth_date']
}}</p>
            <p class="text-secondary"><b>Email: </b>{{ account['email'] }}</p>
            <legend class="border-bottom mb-4">Modificar</legend>
            <a class="btn btn-outline-primary btn-sm mt-1 mb-1" href="{{
url_for('alums.edit_alumn', alumn_id=account['alumn_id']) }}">Datos</a>
            <a class="btn btn-outline-warning btn-sm mt-1 mb-1" href="{{
url_for('alums.edit_avatar', alumn_id=account['alumn_id']) }}">Avatar</a>
        </div>
    </div>
</div>
<div class="content-section">
    <div class="media">
        <div class="media-body">
            <legend class="border-bottom mb-4">Grupos</legend>
            <a class="btn btn-outline-primary btn-sm mt-1 mb-1" href="{{
url_for('grupos.select_choise_grupo') }}">Elegir Grupo</a>
            <a class="btn btn-outline-success btn-sm mt-1 mb-1" href="{{
url_for('grupos.see_grupo') }}">Mi Grupo</a>
        </div>
    </div>
</div>
{% endblock %}

```

*Archivo 18. profile\_a.html – [templates-alumn]*

El siguiente código muestra como editar los datos del alumnado.

```

{% extends "base/base_a.html" %}
{% block content %}
    <div class="content-section">
        <form method="post" action="{{ url_for('alums.update_alumn',
alumn_id=alumno['alumn_id']) }}" enctype="multipart/form-data" autocomplete="off">
            <div class="form-row">
                <legend class="border-bottom mb-4">Modificar Datos</legend>
                <div class="form-group col-md-6">
                    <label for="username">Usuario</label>
                    <input type="text" class="form-control" name="username" id="username"
value="{{ alumno['username'] }}" required>
                </div>
                <div class="form-group col-md-6">
                    <label for="fullname">Nombre Completo</label>

```



```

        <input type="text" class="form-control"
name="fullname" id="fullname" value="{{ alumno['fullname'] }}" required>
    </div>
    <div class="form-group col-md-6">
        <label for="email">Email</label>
        <input type="email" class="form-control"
name="email" id="email" value="{{ alumno['email'] }}" required>
    </div>
    </div>
    <button type="submit" class="btn btn-outline-success">Actualizar</button>
    <a class="btn btn-outline-secondary btn mt-1 mb-1" href="javascript:history.back(-
1);" title="Ir la página anterior">Volver</a>
    </form>
</div>
{% endblock %}

```

*Archivo 19. edit\_a.html – [templates-alumn]*

Los siguientes códigos muestran las pantallas de registro y de acceso al alumnado.

```

{% extends "base/base.html" %}
{% block content %}
    <div class="content-section">
        <form method="POST" action="{{ url_for('alums.register_a') }}"
autocomplete="off">
            <div class="form-row">
                <legend class="border-bottom mb-4"><b>Registro</b></legend>
                <div class="form-group col-md-6">
                    <label for="username"><b>Usuario</b></label>
                    <input type="text" class="form-control" name="username"
placeholder="Usuario" id="username" required>
                </div>
                <div class="form-group col-md-6">
                    <label for="password"><b>Contraseña</b></label>
                    <input type="password" class="form-control" name="password"
placeholder="Clave" id="password" required>
                </div>
                <div class="form-group col-md-6">
                    <label for="fullname"><b>Nombre Completo</b></label>
                    <input type="text" class="form-control"
name="fullname" placeholder="Nombre Completo" id="fullname" required>
                </div>
                <div class="form-group col-md-6">
                    <label for="birth_date"><b>Año de Nacimiento</b></label>
                    <input type="text" class="form-control" name="birth_date" placeholder="Año
de Nacimiento" id="birth_date" required>
                </div>
                <div class="form-group col-md-12">
                    <label for="email"><b>Email</b></label>
                    <input type="email" class="form-control"
name="email" placeholder="Email" id="email" required>
                </div>
                {% include "alumn/avatares.html" %}
            </div>
            <button type="submit" class="btn btn-outline-success">Registrar</button><br>
            <small>¿Ya tienes cuenta?, inicia sesión <a href="{{ url_for('alums.login_a') }}">aquí</a></small>
        </form>
    </div>
{% endblock %}

```

*Archivo 20. register\_a.html – [templates-alumn]*

```

{% extends "base/base.html" %}
{% block content %}
    <div class="content-section">
        <form action="{{ url_for('alums.login_a') }}"
method="post" autocomplete="off">
            <div class="form-row">
                <legend class="border-bottom mb-4">Acceso</legend>
                <div class="form-group col-md-6">
                    <label for="username">Usuario</label>
                    <input type="text" class="form-control"
name="username" placeholder="Usuario" id="username" required>
                </div>

```

```

        <div class="form-group col-md-6">
            <label for="password">Contraseña</label>
            <input type="password" class="form-control"
name="password" placeholder="Clave" id="password" required>
        </div>
        </div>
        <button type="submit" class="btn btn-outline-success">Acceder</button><br>
        <small>¿Aún no tienes cuenta?, pulsa <a href="{{ url_for('alumni.register_a')
}}">aquí</a></small>
        </form>
    </div>
{% endblock %}

```

*Archivo 21. login\_a.html – [templates-alumni]*

El siguiente archivo, muestra la pantalla donde aparecen todos los juegos disponibles.

```

{% extends "base/base_a.html" %}
{% block content %}
{% for juego in juegos %}
    <div class="content-section">
        <div class="media">
            <div class="media-body">
                <legend class="border-bottom mb-4">{{ juego['name'] }}</legend>
                <p class="text-secondary" align="justify"><b>Descripción: </b><br>{{
juego['description'] }}</p>
                <p class="text-secondary"><b>Niveles: </b>{{ juego['levels'] }}</p>
                <a class="btn btn-outline-primary btn-sm mt-1 mb-1" href="{{
url_for('alumni.play_gameA', game_id=juego['game_id']) }}">Jugar</a>
            </div>
        </div>
    </div>
{% endfor %}
{% endblock %}

```

*Archivo 22. games\_a.html – [templates-alumni]*

En los dos restantes códigos, primero observamos la pantalla de los resultados y posteriormente la pantalla de logros.

```

{% extends "base/base_a.html" %}

{% block title %}Puntuaciones - Alumno/a{% endblock %}

{% block content %}
<div class="content-section">
    <div class="media">
        <div class="media-body">
            <legend class="border-bottom mb-4"><b>Mis Puntuaciones</b></legend>
            <p class="text-secondary">Juegos Disponibles</p>
            <ul class="nav nav-pills mb-3" id="pills-tab" role="tablist">
                <li class="nav-item">
                    <a class="nav-link active" id="pills-home-tab" data-toggle="pill"
href="#pills-home" role="tab" aria-controls="pills-home" aria-selected="true">Notas
Musicales</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" id="pills-profile-tab" data-toggle="pill" href="#pills-
profile" role="tab" aria-controls="pills-profile" aria-selected="false">Equivalencias
Musicales</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" id="pills-contact-tab" data-toggle="pill" href="#pills-
contact" role="tab" aria-controls="pills-contact" aria-selected="false">Atrapa al
Intruso</a>
                </li>
            </ul>
            <div class="tab-content" id="pills-tabContent">
                <div class="tab-pane fade show active" id="pills-home" role="tabpanel" aria-
labelledby="pills-home-tab">
                    <b>Puntuación Total: </b>{{ puntos['total'] }} puntos
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="tab-pane fade" id="pills-profile" role="tabpanel" aria-
labelledby="pills-profile-tab">
            <b>Puntuación Total: </b>{{ puntos2['total'] }} puntos
        </div>
        <div class="tab-pane fade" id="pills-contact" role="tabpanel" aria-
labelledby="pills-contact-tab">
            <b>Puntuación Total: </b>{{ puntos3['total'] }} puntos
        </div>
    </div>
</div>
</div>
</div>
{% endblock %}

```

#### Archivo 23. results\_a.html – [templates-alumn]

```

{% extends "base/base_a.html" %}

{% block title %}Logros - Alumno/a{% endblock %}

{% block content %}
<div class="content-section">
    <div class="media">
        <div class="media-body">
            <legend class="border-bottom mb-4"><b>Mis Logros</b></legend>
            <center></center>
            <legend class="col-form-label col-sm-12 pt-0"><b>Leyenda</b> (Puntos
Totales)</legend><br>
            
            
            
            
            
            
            
            
            
        </div>
    </div>
</div>
{% endblock %}

```

#### Archivo 24. logros\_a.html – [templates-alumn]

Las imágenes que aparecen como ley0.png, ley1.png... se trata de la leyenda de los logros.

### 4.7.3. Módulo [teacher]

Nos encontramos en primera instancia con el fichero [templates] que tiene todas las plantillas usadas por el profesorado. A continuación, se observa el código para el inicio del profesorado.

```

{% extends "base/base_t.html" %}
{% block content %}
{% for manage in manages %}
    <div class="content-section">
        <div class="media">
            <div class="media-body">
                <legend class="border-bottom mb-4">Grupo: {{ manage['name'] }}</legend>
                <p class="text-secondary"><b>Aula: </b>{{ manage['classroom'] }}</p>
                {% if manage['fullname'] == 'NULL' %}
                    <p class="text-secondary"><b>Profesor/a: </b>No asociado</p>
                {% else %}

```

```

        <p class="text-secondary"><b>Profesor/a: </b>{{ manage['fullname'] }}</p>
    {% endif %}
</div>
</div>
</div>
{% endfor %}
{% endblock %}

```

*Archivo 25. index\_t.html –[templates-teacher]*

Los siguientes códigos, muestran en primera instancia la pantalla de registros del profesorado, y en segunda instancia la pantalla de acceso del profesorado.

```

{% extends "base/base_especial.html" %}
{% block content %}
<main role="main" class="container">
    <div class="row">
        <div class="col-md-8">
            {% with messages = get_flashed_messages(with_categories=true) %}
                {% if messages %}
                    {% for category, message in messages %}
                        <div class="alert alert-{{ category }}">
                            {{ message }}
                        </div>
                    {% endfor %}
                {% endif %}
            {% endwith %}
            <div class="content-section">
                <form method="post" action="{{ url_for('teachers.register_t') }}"
                    autocomplete="off">
                    <div class="form-row">
                        <legend class="border-bottom mb-4">Registro</legend>
                        <div class="form-group col-md-6">
                            <label for="username">Usuario</label>
                            <input type="text" class="form-control" name="username"
                                placeholder="Usuario" id="username" required>
                        </div>
                        <div class="form-group col-md-6">
                            <label for="password">Contraseña</label>
                            <input type="password" class="form-control" name="password"
                                placeholder="Clave" id="password" required>
                        </div>
                        <div class="form-group col-md-6">
                            <label for="fullname">Nombre Completo</label>
                            <input type="text" class="form-control" name="fullname"
                                placeholder="Nombre Completo" id="fullname" required>
                        </div>
                        <div class="form-group col-md-6">
                            <label for="email">Email</label>
                            <input type="email" class="form-control"
                                name="email" placeholder="Email" id="email" required>
                        </div>
                        <div class="form-group col-md-4">
                            <label for="dni">DNI</label>
                            <input type="text" class="form-control" name="dni"
                                placeholder="DNI" id="dni" required>
                        </div>
                        <div class="form-group col-md-4">
                            <label for="phone">Teléfono</label>
                            <input type="tel" class="form-control"
                                name="phone" placeholder="Teléfono" id="phone" required>
                        </div>
                        <div class="form-group col-md-4">
                            <label for="gender">Sexo - M (Masc) o F (Fem)</label>
                            <select class="form-control" name="gender" id="gender">
                                <option>M</option>
                                <option>F</option>
                            </select>
                        </div>
                        <div>
                            <button type="submit" class="btn btn-outline-
                                success">Registrar</button><br>
                            <small>¿Ya tienes cuenta?, inicia sesión <a href="{{
                                url_for('teachers.login_t') }}">aquí</a></small>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</main>
{% endblock %}

```

```

    </div>
  </div>
</div>
</main>
{% endblock %}

```

#### Archivo 26. register\_t.html – [templates-teacher]

```

{% extends "base/base_especial.html" %}
{% block content %}
<main role="main" class="container">
  <div class="row">
    <div class="col-md-8">
      {% with messages = get_flashed_messages(with_categories=true) %}
      {% if messages %}
        {% for category, message in messages %}
          <div class="alert alert-{{ category }}">
            {{ message }}
          </div>
        {% endfor %}
      {% endif %}
      {% endwith %}
      <div class="content-section">
        <form action="{{ url_for('teachers.login_t') }}"
method="post" autocomplete="off">
          <div class="form-row">
            <legend class="border-bottom mb-4">Acceso</legend>
            <div class="form-group col-md-6">
              <label for="username">Usuario</label>
              <input type="text" class="form-control"
name="username" placeholder="Usuario" id="username" required>
            </div>
            <div class="form-group col-md-6">
              <label for="password">Contraseña</label>
              <input type="password" class="form-control"
name="password" placeholder="Clave" id="password" required>
            </div>
          </div>
          <button type="submit" class="btn btn-outline-success">Acceder</button><br>
          <small>¿Aún no tienes cuenta?, pulsa <a href="{{
url_for('teachers.register_t') }}">aquí</a></small>
        </form>
      </div>
    </div>
  </div>
</main>
{% endblock %}

```

#### Archivo 27. login\_t.html – [templates-teacher]

Los siguientes códigos, nos muestran primeramente la pantalla del perfil del profesorado y a continuación, la pantalla de modificación de los datos del profesor.

```

{% extends "base/base_t.html" %}
{% block content %}
<div class="content-section">
  <div class="media">
    
    <div class="media-body">
      <legend class="border-bottom mb-4">Mis datos</legend>
      <h2 class="account-heading">{{ account['fullname'] }}</h2>
      <p class="text-secondary"><b>Usuario:</b> {{ account['username'] }}</p>
      <p class="text-secondary"><b>DNI:</b> {{ account['dni'] }}</p>
      <p class="text-secondary"><b>Email:</b> {{ account['email'] }}</p>
      <p class="text-secondary"><b>Teléfono:</b> {{ account['phone'] }}</p>
      <p class="text-secondary"><b>Último acceso:</b> {{ account['login_date'] }}</p>
      <a class="btn btn-outline-dark btn-sm mt-1 mb-1" href="{{
url_for('teachers.edit_teacher', teacher_id=account['teacher_id']) }}">Modificar</a>
    </div>
  </div>
</div>
<div class="content-section">
  <div class="media">
    <div class="media-body">

```

```

        <legend class="border-bottom mb-4">Grupos</legend>
        <a class="btn btn-outline-primary btn-sm mt-1 mb-1" href="{{
url_for('grupos.create_grupo', teacher_id=account['teacher_id']) }}">Crear</a>
        <a class="btn btn-outline-success btn-sm mt-1 mb-1" href="{{
url_for('grupos.manage_grupo', teacher_id=account['teacher_id']) }}">Gestionar</a>
    </div>
</div>
</div>
{% endblock %}

```

#### Archivo 28. profile\_t.html – [templates-teacher]

```

{% extends "base/base_t.html" %}
{% block content %}
    <div class="content-section">
        <div class="media">
            
            <div class="media-body">
                <legend class="border-bottom mb-4">Mis datos</legend>
                <h2 class="account-heading">{{ account['fullname'] }}</h2>
                <p class="text-secondary"><b>Usuario: </b>{{ account['username'] }}</p>
                <p class="text-secondary"><b>DNI: </b>{{ account['dni'] }}</p>
                <p class="text-secondary"><b>Email: </b>{{ account['email'] }}</p>
                <p class="text-secondary"><b>Teléfono: </b>{{ account['phone'] }}</p>
                <p class="text-secondary"><b>Último acceso: </b>{{ account['login_date'] }}</p>
                <a class="btn btn-outline-dark btn-sm mt-1 mb-1" href="{{
url_for('teachers.edit_teacher', teacher_id=account['teacher_id']) }}">Modificar</a>
            </div>
        </div>
    </div>
    <div class="content-section">
        <div class="media">
            <div class="media-body">
                <legend class="border-bottom mb-4">Grupos</legend>
                <a class="btn btn-outline-primary btn-sm mt-1 mb-1" href="{{
url_for('grupos.create_grupo', teacher_id=account['teacher_id']) }}">Crear</a>
                <a class="btn btn-outline-success btn-sm mt-1 mb-1" href="{{
url_for('grupos.manage_grupo', teacher_id=account['teacher_id']) }}">Gestionar</a>
            </div>
        </div>
    </div>
{% endblock %}

```

#### Archivo 29. edit\_t.html – [templates-teacher]

Y, por último, se muestra el código de la pantalla que muestra al profesorado los resultados del alumnado.

```

{% extends "base/base_t.html" %}

{% block title %}Puntuaciones - Alumno/a{% endblock %}

{% block content %}
    <div class="content-section">
        <div class="media">
            <div class="media-body">
                <legend class="border-bottom mb-4"><b>Puntuaciones: </b>{{ username }}</legend>
                <p class="text-secondary">Juegos Disponibles</p>
                <ul class="nav nav-tabs" id="myTab" role="tablist">
                    <li class="nav-item">
                        <a class="nav-link active" id="home-tab" data-toggle="tab" href="#home"
role="tab" aria-controls="home" aria-selected="true">Notas Musicales</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" id="profile-tab" data-toggle="tab" href="#profile"
role="tab" aria-controls="profile" aria-selected="false">Equivalencias Musicales</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" id="contact-tab" data-toggle="tab" href="#contact"
role="tab" aria-controls="contact" aria-selected="false">Atrapa al Intruso</a>
                    </li>
                </ul>
                <div class="tab-content" id="myTabContent">

```

```

        <div class="tab-pane fade show active" id="home" role="tabpanel" aria-
labelledby="home-tab">
            <br><b>Puntuación Total: </b>{{ puntos1['total'] }} puntos
        </div>
        <div class="tab-pane fade" id="profile" role="tabpanel" aria-
labelledby="profile-tab">
            <br><b>Puntuación Total: </b>{{ puntos2['total'] }} puntos
        </div>
        <div class="tab-pane fade" id="contact" role="tabpanel" aria-
labelledby="contact-tab">
            <br><b>Puntuación Total: </b>{{ puntos3['total'] }} puntos
        </div>
        <br><a class="btn btn-outline-secondary btn-sm mt-1 mb-1"
href="javascript:history.back(-1);" title="Ir la página anterior">Volver</a>
    </div>
</div>
</div>
</div>
{% endblock %}

```

*Archivo 30. results\_alumn.html – [templates-teacher]*

#### 4.7.4. Módulo [grupo]

Nos encontramos en primera instancia con el fichero [templates] que tiene todas las plantillas usadas para los grupos. A continuación, se observa el código para la gestión de grupos.

```

{% extends "base/base_t.html" %}
{% block content %}
{% for manage in manages %}
    {% if manage['teacher_id'] == teacher %}
        <div class="content-section">
            <div class="media">
                <div class="media-body">
                    <legend class="border-bottom mb-4">
                        <a href="{{ url_for('grupos.show_grupo', name=manage['name']) }}">{{
manage['name'] }}</a>
                    </legend>
                    <p class="text-secondary"><b>Aula: </b>{{ manage['classroom'] }}</p>
                </div>
            </div>
        </div>
    {% endif %}
{% endfor %}
<a class="btn btn-outline-secondary btn-sm mt-1 mb-1" href="javascript:history.back(-
1);" title="Ir la página anterior">Volver</a>
{% endblock %}

```

*Archivo 31. manage.html – [templates-grupo]*

El siguiente código, muestra la pantalla donde se gestiona el grupo en sí, se observan los alumnos disponibles, etc.

```

{% extends "base/base_t.html" %}
{% block content %}
<legend class="border-bottom mb-4"><b>{{ grupo['name'] }}</b></legend>
<p class="text-secondary"><b>Aula: </b>{{ grupo['classroom'] }}<br>
    <a class="btn btn-outline-success btn-sm mt-1 mb-1" href="{{
url_for('grupos.edit_grupo', name=grupo['name']) }}">Actualizar</a>
    <a class="btn btn-outline-danger btn-sm mt-1 mb-1" href="{{
url_for('grupos.supr_grupo', name=grupo['name']) }}">Eliminar</a>
    <a class="btn btn-outline-secondary btn-sm mt-1 mb-1" href="javascript:history.back(-
1);" title="Ir la página anterior">Volver</a>
</p>
<legend class="border-bottom mb-4">Alumnado</legend>
{% for manage in manages %}
    {% if manage['name'] == grupo['name'] %}
        <div class="content-section">
            <div class="media">
                <div class="media-body">

```

```

        <a href="{{ url_for('grupos.show_results', username=manage['username']) }}">{{
manage['fullname'] }}</a>
        <p class="text-secondary"><b>Usuario: </b>{{ manage['username'] }}</p>
    </div>
</div>
</div>
{% endif %}
{% endfor %}
{% endblock %}

```

*Archivo 32. grupo.html – [templates-grupo]*

Los siguientes códigos, muestran las pantallas para modificar aula y para eliminar grupo.

```

{% extends "base/base_t.html" %}
{% block content %}
    <div class="content-section">
        <form method="post" action="{{ url_for('grupos.update_group',
name=grupo['name']) }}" autocomplete="off">
            <div class="form-row">
                <legend class="border-bottom mb-4">Modificar Aula</legend>
                <div class="form-group col-md-6">
                    <label for="username">Aula</label>
                    <input type="text" class="form-control" name="classroom" id="classroom"
value="{{ grupo['classroom'] }}" required>
                </div>
            </div>
            <button type="submit" class="btn btn-outline-success">Actualizar</button>
            <a class="btn btn-outline-secondary btn mt-1 mb-1" href="javascript:history.back(-
1);" title="Ir la página anterior">Volver</a>
        </form>
    </div>
{% endblock %}

```

*Archivo 33. edit\_g.html – [templates-grupo]*

```

{% extends "base/base_t.html" %}
{% block content %}
    <div class="content-section">
        <form method="post" action="{{ url_for('grupos.delete_group',
name=grupo['name']) }}" autocomplete="off">
            <div class="form-row">
                <legend class="border-bottom mb-4">Eliminar Grupo</legend>
                <div class="form-group col-md-6">
                    <label for="name">Grupo</label>
                    <input type="text" class="form-control" name="name" id="name" value="{{
grupo['name'] }}" required readonly>
                </div>
            </div>
            <button type="submit" class="btn btn-outline-success">Eliminar</button>
            <a class="btn btn-outline-secondary btn mt-1 mb-1" href="javascript:history.back(-
1);" title="Ir la página anterior">Volver</a>
        </form>
    </div>
{% endblock %}

```

*Archivo 34. delete\_g.html – [templates-grupo]*

Los siguientes códigos, nos muestran las pantallas de creación y confirmación de un nuevo grupo, gestionado por el profesorado.

```

{% extends "base/base_t.html" %}
{% block content %}
    <div class="content-section">
        <form method="post" action="{{ url_for('grupos.add_grupo',
teacher_id=teacher['teacher_id']) }}" autocomplete="off">
            <div class="form-row">
                <legend class="border-bottom mb-4">Crear Nuevo Grupo</legend>
                <div class="form-group col-md-3">
                    <label for="name"><b>Nombre</b></label>
                    <input type="text" class="form-control" name="name" placeholder="Nombre Grupo"
id="name" required>
                </div>
            </div>

```



```

        <div class="form-group col-md-3">
            <label for="classroom"><b>Aula</b></label>
            <input type="text" class="form-control" name="classroom" placeholder="Aula"
id="classroom" required>
        </div>
    </div>
    <button type="submit" class="btn btn-outline-success">Siguiente</button>
    <a class="btn btn-outline-secondary btn mt-1 mb-1" href="javascript:history.back(-
1);" title="Ir la página anterior">Volver</a>
    </form>
</div>
{% endblock %}

```

#### Archivo 35. create.html – [templates-grupo]

```

{% extends "base/base_t.html" %}
{% block content %}
    <div class="content-section">
        <form method="POST" action="{% url_for('grupos.confirm_group') %}"
autocomplete="off">
            <div class="form-row">
                <legend class="border-bottom mb-4"><b>Confirmar grupo</b></legend>
                <div class="form-group col-md-3">
                    <label for="id_teacher"><b>Profesor (ID)</b></label>
                    <input type="text" class="form-control plaintext" name="id_teacher"
id="id_teacher" value="{% teacher['teacher_id'] %}" required readonly>
                </div>
                <div class="form-group col-md-6">
                    <label for="name"><b>Grupo (Nombre)</b></label>
                    <input type="text" class="form-control" name="name" id="name"
placeholder="Vuelva a escribir su grupo" required>
                </div>
            </div>
            <button type="submit" class="btn btn-outline-success">Confirmar</button>
            <small>Recuerda!, sino confirmas tu grupo, no podrás volver atrás.</small>
        </form>
    </div>
{% endblock %}

```

#### Archivo 36. confirm.html – [templates-grupo]

Por último, se muestran tres códigos, pertenecientes a la pantalla de elección de grupo del alumno, de ver el grupo del alumno y de ver el profesor del grupo de dicho alumno.

```

{% extends "base/base_a.html" %}
{% block content %}
    <div class="content-section">
        <form method="POST" action="{% url_for('grupos.choise_grupo') %}"
autocomplete="off">
            <div class="form-row">
                <legend class="border-bottom mb-4"><b>Elección de Grupo</b></legend>
                <div class="form-group col-md-3">
                    <label for="id_alumn"><b>Alumno (ID)</b></label>
                    <input type="text" class="form-control plaintext" name="id_alumn"
id="id_alumn" value="{% alumn %}" required readonly>
                </div>
                <div class="form-group col-md-3">
                    <label for="name_grupo"><b>Grupo (Nombre)</b></label>
                    <input type="text" class="form-control" name="name_grupo" id="name_grupo"
required>
                </div>
            </div>
            <button type="submit" class="btn btn-outline-success">Confirmar</button>
            <a class="btn btn-outline-secondary btn mt-1 mb-1" href="javascript:history.back(-
1);" title="Ir la página anterior">Volver</a>
        </form>
    </div>
{% endblock %}

```

#### Archivo 37. choise.html – [templates-grupo]

```
{% extends "base/base_a.html" %}
{% block content %}
{% for manage in manages %}
  {% if manage['alumn_id'] == alumn %}
    <div class="content-section">
      <div class="media">
        <div class="media-body">
          <legend class="border-bottom mb-4">
            <h2 class="account-heading">{{ manage['name'] }}</h2>
          </legend>
          <p class="text-secondary"><b>Aula: </b>{{ manage['classroom'] }}</p>
          <a class="btn btn-outline-danger btn-sm mt-1 mb-1" href="{{
url_for('grupos.grupo_teacher', name=manage['name']) }}">Ver Profesor</a>
          <a class="btn btn-outline-secondary btn-sm mt-1 mb-1"
href="javascript:history.back(-1);" title="Ir la página anterior">Volver</a>
        </div>
      </div>
    </div>
  {% endif %}
{% endfor %}
{% endblock %}
```

*Archivo 38. see.html – [templates-grupo]*

```
{% extends "base/base_a.html" %}
{% block content %}
{% for manage in manages %}
  {% if manage['name'] == grupo['name'] %}
    <div class="content-section">
      <div class="media">
        <div class="media-body">
          <legend class="border-bottom mb-4">
            <h2 class="account-heading">{{ manage['fullname'] }}</h2>
          </legend>
          <p class="text-secondary"><b>Email: </b>{{ manage['email'] }}</p>
        </div>
      </div>
    </div>
  {% endif %}
{% endfor %}
<a class="btn btn-outline-secondary btn-sm mt-1 mb-1" href="javascript:history.back(-1);" title="Ir la página anterior">Volver</a>
{% endblock %}
```

*Archivo 39. grupo\_teacher.html – [templates-grupo]*

## 4.7.5. Módulo [main]

Nos encontramos en primera instancia con el fichero [templates] que tiene todas las plantillas restantes, y tres archivos, uno sobre el calendario, otro de contacto y otro sobre quien ha realizado el proyecto.

A continuación, se observa el código para la sesión sobre quien realizo el proyecto.

```
{% extends "base/base.html" %}
{% block content %}
{% include "main/about.html" %}
{% endblock content %}
```

*Archivo 40. about\_us.html – [templates-main]*

El resto de los archivos “about\_us\_a.html” y “about\_us\_t.html”, son iguales al archivo anterior, solo que extienden a su base. Y los tres archivos incluyen a la siguiente plantilla:

```
<h3><b>Proyecto llevado a cabo por</b></h3>
<div class="content-section">
  <div class="media">
    <div>
      
      <br><a href="https://www.linkedin.com/in/xabl0/"></a>
    </div>
    <div class="media-body">
      <legend class="border-bottom mb-4"><b>Alumno</b></legend>
      <h2 class="account-heading">Javier Ramírez Quintero</h2><br>
      <p class="text-secondary"><b>Grado: </b>Ingeniería en Informática (Mención
Software)</p>
      <p class="text-secondary"><b>Universidad: </b>Escuela Politécnica Superior de
Córdoba (EPSC) - Universidad de Córdoba</p>
    </div>
  </div>
</div>
```

Archivo 41. about.html – [templates-main]

A continuación, se observa el código para la sesión sobre el calendario.

```
{% extends "base/base.html" %}
{% block content %}
{% include "main/calendario.html" %}
{% endblock content %}
```

Archivo 42. calendar.html – [templates-main]

El resto de los archivos “calendar\_a.html” y “calendar\_t.html”, son iguales al archivo anterior, solo que extienden a su base. Y los tres archivos incluyen a la siguiente plantilla:

```
<div class="title">Septiembre 2019</div>
<table border="1">
<tr><th>Lun</th><th>Mar</th><th>Mié</th><th>Jue</th><th>Vie</th><th>Sáb</th><th>Dom</th></tr>
<tr><td><span class="date">&nbsp;</span></td><td><span
class="date">&nbsp;</span></td><td><span class="date">&nbsp;</span></td><td><span
class="date">&nbsp;</span></td><td><span class="date">&nbsp;</span></td><td><span
class="date">&nbsp;</span></td><td><span class="date">1</span></td></tr>
<tr><td><span class="date">2</span></td><td><span class="date">3</span></td><td><span
class="date">4</span></td><td><span class="date">5</span></td><td><span
class="date">6</span></td><td><span class="date">7</span></td><td><span
class="date">8</span></td></tr>
<tr><td><span class="date">9</span></td><td><span class="date">10</span></td><td><span
class="date">11</span></td><td><span class="date">12</span></td><td><span
class="date">13</span></td><td><span class="date">14</span></td><td><span
class="date">15</span></td></tr>
<tr><td><span class="date_sp"><b>16</b></span></td><td><span
class="date">17</span></td><td><span class="date">18</span></td><td><span
class="date">19</span></td><td><span class="date">20</span></td><td><span
class="date">21</span></td><td><span class="date">22</span></td></tr>
<tr><td><span class="date">23</span></td><td><span class="date">24</span></td><td><span
class="date">25</span></td><td><span class="date">26</span></td><td><span
class="date">27</span></td><td><span class="date">28</span></td><td><span
class="date">29</span></td></tr>
<tr><td><span class="date">30</span></td><td><span
class="date">&nbsp;</span></td><td><span class="date">&nbsp;</span></td><td><span
class="date">&nbsp;</span></td><td><span class="date">&nbsp;</span></td><td><span
class="date">&nbsp;</span></td><td><span class="date">&nbsp;</span></td></tr>
</table>
<br>
<div class="content-section">
  <div class="media">
    <div class="media-body">
      <legend class="border-bottom mb-4"><b>Eventos Septiembre</b></legend>
```

```

<ul>
    <li>Apertura académica del curso 19/20 (Lunes 16)</li>
</ul>
</div>
</div>
<p><div class="title">Octubre 2019</div>





```

```
<tr><td><span class="date"><b>23</b></span></td><td><span
class="date">24</span></td><td><span class="date">25</span></td><td><span
class="date">26</span></td><td><span class="date">27</span></td><td><span
class="date">28</span></td><td><span class="date">29</span></td></tr>
<tr><td><span class="date">30</span></td><td><span class="date">31</span></td><td><span
class="date">&nbsp;</span></td><td><span class="date">&nbsp;</span></td><td><span
class="date">&nbsp;</span></td><td><span class="date">&nbsp;</span></td><td><span
class="date">&nbsp;</span></td></tr>
</table>
<br>
<div class="content-section">
  <div class="media">
    <div class="media-body">
      <legend class="border-bottom mb-4"><b>Eventos Diciembre</b></legend>
      <ul>
        <li>Comienzo vacaciones de Navidad (Lunes 23)</li>
      </ul>
    </div>
  </div>
</div>
```

*Archivo 43. calendario.html – [templates-main]*

A continuación, se observa el código para la sesión sobre el contacto.

```
{% extends "base/base.html" %}
{% block content %}
{% include "main/contacto.html" %}
{% endblock content %}
```

*Archivo 44. contact.html – [templates-main]*

El resto de los archivos “contact\_a.html” y “contact\_t.html”, son iguales al archivo anterior, solo que extienden a su base. Y los tres archivos incluyen a la siguiente plantilla:

```
<h2><b>¿Tienes alguna duda, y necesitas más ayuda?</b></h2>
<p>Si tienes cualquier duda más sobre el funcionamiento de la web,
tienes problemas de registro/acceso, o sabes de alguna mejora.
Háznoslo saber, rellena el siguiente formulario y te responderemos
a la mayor brevedad posible.</p>
<div class="content-section">
  <a href="mailto:i52raquj@uco.es">Contactar por correo</a>
</div>
```

*Archivo 45. contacto.html – [templates-main]*

Los siguientes códigos, muestran la pantalla de acceso al profesorado, donde se solicita clave externa, y la pantalla de denegado en caso de introducir otra contraseña.

```
<!DOCTYPE html>
<html>
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/main_t.css') }}">

  {% if title %}
    <title>MusicGames - {{ title }}</title>
  {% else %}
    <title>MusicGames</title>
  {% endif %}
```

```

</head>
<body>
    <header class="site-header">
        <nav class="navbar navbar-expand-md navbar-dark bg-steel
fixed-top">
            <div class="container">
                <a class="navbar-brand mr-4"
href="/">MusicGames</a>
                <button class="navbar-toggler"
type="button" data-toggle="collapse" data-target="#navbarToggle" aria-
controls="navbarToggle" aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-
icon"></span>
                </button>
                <div class="collapse navbar-collapse"
id="navbarToggle">
                    <div class="navbar-nav mr-
auto">
                        </div>
                        <!-- Navbar Right Side -->
                        <div class="navbar-nav">
                            </div>
                        </div>
                    </div>
                </nav>
            </header>
            <script LANGUAGE="JavaScript">
                function Entrar(password) {
                    var passvalido = "1234";
                    if(password == passvalido){
                        window.location="{{ url_for('teachers.login_t') }}"
                    } else {
                        window.location="{{ url_for('main.not_permit') }}"
                    }
                }
            </script>
            <body>
                <main role="main" class="container">
                    <div class="row">
                        <div class="col-md-8">
                            <div class="content-section">
                                <legend class="border-bottom
mb-4"><b>Introduzca la Contraseña facilitada</b></legend>
                                <form>
                                    <p>
                                        <input type="password"
name="password" size="20" placeholder="Contraseña" id="password">
                                        <br><br>
                                        <input type="button" class="btn
btn-outline-success" value="Entrar" onClick="Entrar(this.form.password.value)"
name="button">
                                    </p>
                                    <a class="btn btn-outline-secondary btn mt-1 mb-1"
href="javascript:history.back(-1);" title="Ir la página anterior">Volver</a>
                                </form>
                            </div>
                        </div>
                    </div>
                </main>
            </body>
        </html>

```

#### Archivo 46. academy.html – [templates-main]

```

<!DOCTYPE html>
<html>
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

```

```

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/main_t.css') }}">

    {% if title %}
        <title>MusicGames - {{ title }}</title>
    {% else %}
        <title>MusicGames</title>
    {% endif %}
</head>
    <body>
        <header class="site-header">
            <nav class="navbar navbar-expand-md navbar-dark bg-steel
fixed-top">
                <div class="container">
                    <a class="navbar-brand mr-4" href="{{
url_for('main.teacher_pass') }}">MusicGames - {{ title }}</a>
                    <button class="navbar-toggler"
type="button" data-toggle="collapse" data-target="#navbarToggle" aria-
controls="navbarToggle" aria-expanded="false" aria-label="Toggle navigation">
                        <span class="navbar-toggler-
icon"></span>
                    </button>
                    <div class="collapse navbar-collapse">
                        <div class="navbar-nav mr-
auto">
                            </div>
                        <!-- Navbar Right Side -->
                        <div class="navbar-nav">
                            </div>
                        </div>
                    </div>
                </nav>
            </header>
            <body>
                <main role="main" class="container">
                    <div class="row">
                        <div class="col-md-8">
                            <div class="content-section">
                                <legend class="border-bottom
mb-4"><b>No tienes permiso</b></legend>
                                <a class="btn btn-outline-secondary btn mt-1 mb-1"
href="javascript:history.back(-1);" title="Ir la página anterior">Volver</a>
                            </div>
                        </div>
                    </div>
                </main>
            </body>
        </html>

```

*Archivo 47. denegado.html – [templates-main]*

A continuación, se muestra la pantalla de inicio para los “Usuarios No Registrados”.

```

{% extends "base/base.html" %}
{% block content %}
<div class="content-section">
    <div class="media">
        <div class="media-body">
            <legend class="border-bottom mb-4">¿Qué debes saber antes de jugar?</legend>
            <ol>
                <li>Si es la primera vez que entras en la plataforma, deberás esperar a que
el/la profesor/a te explique el funcionamiento</li><br>
                <li>Una vez sepas el correcto funcionamiento, podrás acceder a las distintas
opciones de la barra menú (Inicio, Perfil, Juegos, Puntuaciones, Logros o Salir)</li>
            </ol>
        </div>
    </div>
</div>
{% endblock %}

```

*Archivo 48. index.html – [templates-main]*

Finalmente, se muestra la pantalla inicial de la plataforma web.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, user-scalable=no, initial-
scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/style.css') }}">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/resp.css') }}">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/main.css') }}">

    {% if title %}
      <title>MusicGames - {{ title }}</title>
    {% else %}
      <title>MusicGames</title>
    {% endif %}
  </head>
  <body>
    <header class="site-header">
      <nav class="navbar navbar-expand-md navbar-dark bg-steel
fixed-top">
        <div class="container">
          <a class="navbar-brand mr-4" href="{{
url_for('main.inicio') }}">MusicGames - {{ title }}</a>
          <button class="navbar-toggler"
type="button" data-toggle="collapse" data-target="#navbarToggle" aria-
controls="navbarToggle" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-
icon"></span>
          </button>
          <div class="collapse navbar-collapse">
            <div class="navbar-nav mr-
auto">
              </div>
            <!-- Navbar Right Side -->
            <div class="navbar-nav">
              </div>
            </div>
          </div>
        </nav>
      </header>
      <br><br>
      <div class="contenedor">
        <div class="contenedor_tarjeta">
          <a href="{{ url_for('juego.index_game') }}">
            <figure id="tarjeta">
              
              <figcaption class="trasera">
                <h2 class="titulo"><b>Usuarios
No Registrados</b></h2>
                <hr>
                <p>Desde aquí puedes entrar
directamente y ver todos los juegos disponibles de nuestra plataforma. <b>Recuerda</b>,
solo podrás jugar un nivel de cada juego.</p>
              </figcaption>
            </figure>
          </a>
        </div>
        <div class="contenedor_tarjeta">
          <a href="{{ url_for('alumni.login_a') }}">
            <figure id="tarjeta">
              
              <figcaption class="trasera">
```



```

class="titulo"><b>Alumnado</b></h2>
<hr>
<p>Espacio dedicado para todo
los alumnos de la Academia. Donde podrás acceder con tu cuenta, además de jugar, podrás
ver todos tus resultados y logros conseguidos.</p>
</figure>
</a>
</div>
<div class="contenedor_tarjeta">
<a href="{{ url_for('main.teacher_pass') }}">
<figure id="tarjeta">

<figcaption class="trasera">
<h2
class="titulo"><b>Profesorado</b></h2>
<hr>
<p>Espacio dedicado para todo
el profesorado de la Academia. Donde podrás llevar un seguimiento de los grupos que
impartes, así como, de los alumnos de éstos.</p>
</figcaption>
</figure>
</a>
</div>
</div>
</body>
</html>

```

*Archivo 49. inicio.html – [templates-main]*

## 4.8. Fichero [game]

En este fichero encontramos todo lo relacionado con los juegos que incorpora nuestro proyecto. Sólo se pasará a mostrar el código del template principal.

```

{% extends "base/base.html" %}
{% block content %}

{% for game in games %}
<div class="content-section">
<div class="media">
<div class="media-body">
<legend class="border-bottom mb-4">{{ game['name'] }}</legend>
<p class="text-secondary" align="justify"><b>Descripción: </b>{{
game['description'] }}</p>
<p class="text-secondary"><b>Niveles: </b>{{ game['levels'] }}</p>
<a class="btn btn-outline-primary btn-sm mt-1 mb-1" href="{{
url_for('juego.play_game', game_id=game['game_id']) }}">Jugar</a>
</div>
</div>
</div>
{% endfor %}

{% endblock %}

```

*Archivo 50. games.html*

## 4.9. API

En este apartado se muestran los códigos que conforman nuestra API, para que se recojan los resultados de los juegos cuando un alumno interactúa con ellos.

```
# Recibir Puntuaciones (API)
@alumns.route('/alumn/api', methods=["POST", "GET"])
def createResult():
    if 'loggedin' in session:
        # Muestra el array de valores recibidos de la API
        print(request.json)
        # Recoge los valores recibidos por la API
        _name_game = request.json["name_game"]
        _level = request.json["level"]
        _points = request.json["score"]
        # Se insertan estos, en la tabla result
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("""
        INSERT INTO result (name_game, level, points, user_name)
        VALUES (%s, %s, %s, %s)""",
        (_name_game, _level, _points, [session['username']]))
        mysql.connection.commit()
    # Si todo ha ido ok, se muestra por consola
    return jsonify('Resultados recibidos correctamente')
```

*Archivo 51. API en views.py del módulo [alumns]*

En este extracto de código, es donde Flask mediante la ruta ‘http://localhost:5000/alumn/api’, recoge los datos que recibe del script en JavaScript que se encuentra en cada juego. A través, de la librería JSON, recogemos los datos, y con la consulta MySQL lo insertamos en la tabla ‘result’.

```
// host
var url = 'http://127.0.0.1:5000/alumn/api';

// post body data
var result = {
  name_game: 'Notas Musicales',
  level: 0,
  score: maxmScore
};

// create request object
let request = new Request(url, {
  method: 'POST',
  body: JSON.stringify(result),
  headers: new Headers({
    'Content-Type': 'application/json'
  })
});

// send POST request
fetch(request)
  .then(res => res.json())
  .then(myJson => console.log(myJson));

const userAction = async () => {
  const response = await fetch(url, {
    method: 'POST',
    body: JSON.stringify(result),
    headers: {
      'Content-Type': 'application/json'
    }
  });
  const myJson = await response.json();
}
```

*Archivo 52. API en script.js del código del juego: Notas Musicales*