

Proyecto Individual: Aplicación de Valoraciones y Rankings con Django y MongoDB

Sección	Descripción
Título del proyecto	Aplicación web de valoraciones, rankings y estadísticas con Django y MongoDB
Tipo de trabajo	Proyecto individual
Objetivo general	Desarrollar una aplicación web donde los usuarios puedan valorar, puntuar y crear rankings de diferentes elementos, consultando estadísticas globales, con gestión de datos mediante un administrador.
Tecnologías obligatorias	Python 3.x, Django 4.x o superior, MongoDB (base de datos NoSQL)
Conectores o librerías recomendadas	mongoengine , djongo o pymongo
Interfaz recomendada	Django Templates combinado con Bootstrap, Materialize, Bulma, Tailwind o similares.
Roles del sistema	Usuario normal: puede registrarse, valorar, puntuar, crear rankings y consultar estadísticas. Administrador: puede gestionar categorías, elementos y cargar datos desde ficheros CSV o JSON.

Proyecto Integrado II

Profesor: Luis Javier López López

Requisitos del proyecto

Requisito funcional		Descripción / Detalle
RF1	Registro e inicio de sesión	El sistema debe permitir que los usuarios se registren e inicien sesión. Solo los usuarios autenticados pueden valorar o crear rankings.
RF2	Roles diferenciados	Debe haber un rol de administrador y uno de usuario normal , con permisos distintos.
RF3	Gestión de categorías	El administrador puede crear, editar o eliminar categorías (ej. películas, libros, juegos).
RF4	Gestión de elementos	El administrador puede añadir, editar o eliminar elementos dentro de cada categoría.
RF5	Carga de datos desde fichero	El administrador puede cargar categorías y elementos desde ficheros CSV o JSON .
RF6	Visualización de elementos	Los usuarios pueden navegar por categorías y ver los elementos disponibles para puntuar.
RF7	Valoración y puntuación	Cada usuario puede puntuar un elemento y añadir un comentario opcional. Solo puede valorar una vez por elemento, aunque puede editar su valoración.
RF8	Creación de rankings personales	Los usuarios pueden crear rankings personalizados con los elementos (por ejemplo: "Mis libros favoritos").
RF9	Consulta de estadísticas globales	El sistema debe mostrar estadísticas como: elementos más valorados, promedio por categoría, número total de valoraciones, etc.
RF10	Panel de administración	El administrador debe poder consultar las estadísticas globales y supervisar la información del sistema.



Proyecto Integrado II

Profesor: Luis Javier López López

Estructura mínima recomendada de la base de datos

Colección (MongoDB)	Contenido esperado	Ejemplo de documento
usuarios	Datos de los usuarios registrados (nombre, email, contraseña, rol).	{ "usuario": "ana23", "email": "ana@example.com", "rol": "usuario" }
categorías	Categorías creadas por el administrador.	{ "nombre": "Películas", "descripcion": "Valoración de películas populares" }
elementos	Elementos dentro de cada categoría.	{ "categoria": "Películas", "nombre": "Inception", "anio": 2010 }
valoraciones	Valoraciones individuales de los usuarios.	{ "usuario": "ana23", "elemento": "Inception", "puntuacion": 5, "comentario": "Excelente" }
rankings	Rankings personales creados por los usuarios.	{ "usuario": "ana23", "nombre": "Mis películas favoritas", "elementos": ["Inception", "Matrix"] }



Proyecto Integrado II

Profesor: Luis Javier López López

Entregables y fechas

Elemento	Descripción
Código fuente completo	Proyecto Django funcional con todas las características implementadas. Se entregará el enlace al repositorio público en GitHub. Debe de haber un obligatoriamente seguimiento de trabajo diario y semanal.
Documento de definición de proyecto	Un documento en el que se defina cuál es la temática del proyecto y cómo se van a abordar los requisitos.
Video DEMO	Un video mostrando el funcionamiento completo del proyecto.

La entrega final del proyecto tendrá como fecha límite el día 20 de Febrero.