



CONSEJERÍA DE EDUCACIÓN
Comunidad de Madrid

IES ENRIQUE TIERNO GALVAN

Parla

**CFGS DESARROLLO DE APLICACIONES
WEB**

Curso 2024/2025

Proyecto DAW

TITULO: JustTrack

Alumno: Eddyson Bueno Soto

Tutor: Francisco Isidoro Jiménez Caro

Junio 2025

Contenido

Introducción.....	4
Justificación del proyecto y su viabilidad	4
Dificultades encontradas.....	5
Tipo de proyecto.....	5
Descripción del proyecto	5
Alcance	5
Objetivos.....	6
Requisitos.....	6
Funcionales.....	6
Técnicos	6
Riesgos.....	6
Calendario	7
Presupuesto	7
Recursos.....	7
Estado actual del proyecto.....	7
Próximos pasos	8
Planificación	9
Análisis: Requisitos funcionales.....	9
Diseño: Modelo Entidad-Relación (MER)	10
Modelo conceptual	10
Modelo lógico	14
Modelo físico.....	20
Despliegue	20
Configuración del servidor	20
Instalación de requisitos	20
Configuración de instalaciones	22

Creación de base de datos e inserción de datos de pruebas	25
Manual de usuario	28
Apartado Inicio	28
Cabecera.....	30
Apartado Nuevo	31
Apartado “Popular”.....	32
Detalle de película	33
Apartado “Listas”	34
Barra de búsqueda.....	37
Inicio de sesión de usuario.....	37
Registro de usuario	38
Cerrar sesión	39

Introducción

En la era digital, el consumo de contenido audiovisual se ha diversificado a través de múltiples plataformas de streaming, de lectura o de videojuegos, lo que dificulta la organización y seguimiento de las películas y series vistas o pendientes de visualizar, libros que ya has leído o pendientes de leer o videojuegos jugados o pendientes de jugar. "JustTrack" nace como una solución para centralizar y gestionar el contenido audiovisual de manera centralizada, eficiente y personalizada.

El objetivo principal de este proyecto es desarrollar una plataforma web intuitiva y moderna que permita a los usuarios marcar el contenido que han visto, leído o jugado, calificarlos y organizar su lista de contenido pendiente de consumir, el consumido y el contenido que les gusta o no les gusta. En una versión inicial la plataforma permitirá marcar películas, pero está planificado que en un futuro el proyecto se expanda para abarcar otros tipos de contenido, como series, libros, videojuegos o incluso música.

Justificación del proyecto y su viabilidad

Actualmente, existen plataformas como JustWatch o Letterboxd, que permiten buscar dónde ver películas y series en streaming, pero se limitan a películas y series, "JustTrack" ofrecería una experiencia enfocada en la organización personalizada y centralizada de más contenido audiovisual como puedan ser libros, videojuegos, etc (esta amplia variedad estaría disponible en futuras versiones, no en la que se presentará en el TFG que vendría siendo una primera versión que presentaría las bases de lo que se haría en un futuro).

Desde el punto de vista técnico, la aplicación se desarrollará como un sistema web accesible desde cualquier navegador, con un backend alojado en un hosting y una interfaz gráfica responsiva. El desarrollo a medida garantiza flexibilidad, escalabilidad y un rendimiento optimizado, y en este caso específico, un coste reducido, dado que en sistemas de CMS nos encontraremos ciertos costes algo elevados para personas como yo si queremos hacer uso de ciertas funciones o incluso para el uso de muchos plugins.

La viabilidad del proyecto se sustenta en su potencial de crecimiento y en la creciente demanda de herramientas de organización para consumidores de contenido digital. Además, su implementación como Trabajo Final de Grado me permitirá adquirir y demostrar habilidades clave en desarrollo web.

Dificultades encontradas

Una de las cosas que planteaba para la aplicación es la posibilidad de directamente utilizar alguna API de películas que haya en algún servicio/plataforma como IMDB o la propia API de JustWatch, pero éstas opciones requieren de que contactes con la empresa y consultes qué precios específicos te podrían brindar o requieren de cierto tiempo extra para poder utilizarlas además de cierto código extra, de forma que si tengo tiempo, implementaré de alguna forma utilizar alguna API de biblioteca de contenido multimedia. Mientras tanto estaré usando una base de datos propia con datos suficientes como para demostrar el funcionamiento de la plataforma con películas.

Tipo de proyecto

Este proyecto es de tipo gestión, ya que su objetivo es desarrollar una plataforma que permita organizar y administrar información relacionada con el consumo de contenido audiovisual.

Descripción del proyecto

Alcance

La versión inicial de "JustTrack" se centrará en la gestión de películas, permitiendo:

- Marcar películas vistas.
- Marcar películas que te gustan.
- Marcar películas que no te gustan.
- Marcar películas que quieres ver introduciéndolas en una "watchlist" de contenido pendiente.

Objetivos

- Desarrollar una plataforma web funcional accesible desde cualquier navegador.
- Implementar un sistema de usuarios con perfiles personalizados.
- Diseñar una interfaz intuitiva y responsive.
- Garantizar la escalabilidad del proyecto para futuras expansiones.

Requisitos

Funcionales

- Registro e inicio de sesión de usuarios.
- Búsqueda de contenido.
- Posibilidad de marcar y calificar contenido audiovisual.
- Interfaz responsiva para distintos dispositivos.

Técnicos

- Backend desarrollado en PHP con Laravel y base de datos MySQL.
- Frontend desarrollado en HTML, CSS y JavaScript y el uso de Vite para optimización.
- Hosting en Clouding con Apache.

Riesgos

Limitaciones de tiempo: Se debe gestionar adecuadamente la carga de trabajo para completar el proyecto a tiempo, al menos su primera fase que incluye la gestión de películas y series.

Complejidad técnica: Implementar un sistema de gestión de contenido.

Disponibilidad de recursos: Se necesita asegurar la estabilidad del hosting y la base de datos.

Calendario

Abril: Análisis, diseño del sistema, diseño de base de datos y configuración del servidor.

Mayo: Desarrollo del frontend, implementación de la base de datos, desarrollo del backend, integración.

Junio: Desarrollo del backend, pruebas, optimización y despliegue.

Presupuesto

Hosting Clouding: 5,91€ al mes, 0,0082€ por hora, 720 horas al mes 24/7.

Dominio: 12,61€ comprando el dominio justtrack.xyz durante 1 año con protección del dominio.

Desarrollo: Gratis, programaré yo todo empleando herramientas gratuitas.

Recursos

- Ordenador personal con entornos de desarrollo.
- Documentación de HTML, CSS, MySQL, Apache y JavaScript.
- Navegador web para búsquedas de ayuda.
- Comunidades y foros de desarrollo.

Estado actual del proyecto

De momento tengo el proyecto con la GUI a comenzada, tengo algún elemento como la cabecera desarrollado y ya están decididas las funciones que permitirá la aplicación cuando esté al completo: Buscar películas por su nombre, consultar información detallada de estas películas como por ejemplo ver en qué plataforma se encuentra y visualizar algún tráiler de éstas o consultar su sinopsis, añadir una película a una watchlist en la que tienes el contenido que quieres ver próximamente y finalmente el poder añadir películas a la lista de contenido ya visualizado.

Próximos pasos

Los pasos a realizar consisten en:

1. Planificar el sistema: Tengo que decidir que se podrá hacer en la plataforma y planificarme.
2. Configuración del servidor: Debo configurar tanto el servidor en el que alojaré la aplicación en producción, como el servidor que usaré en mi entorno de desarrollo. En mi caso utilizaré XAMPP en el entorno de desarrollo, y en producción utilizaré un servidor de Clouding que servirá la aplicación web usando Apache en el dominio que he adquirido por 1 año.
3. Crear modelos de la base de datos: Debo crear el modelo conceptual, lógico y finalmente el modelo físico de la base de datos para poder trabajar. En mi caso la base de datos la haré en MySQL.
4. Implementar la base de datos: Debo hacer que la base de datos esté funcional y utilizable en XAMPP (la herramienta de desarrollo que utilizaré para el proyecto) y posteriormente, deberé crearla en el servidor.
5. Inserción de datos de pruebas: Debo insertar datos en la base de datos para ir programando cada apartado de la aplicación y verla funcionando y probar su correcto funcionamiento.
6. Creación de la interfaz gráfica: Debo diseñar una interfaz gráfica para cada apartado de la aplicación para posteriormente hacer que esa interfaz gráfica sea dinámica. Crearé la interfaz gráfica empleando HTML, CSS y JavaScript.
7. Programación de backend: Debo hacer que las interfaces gráficas que he hecho anteriormente con HTML, CSS y JavaScript sean dinámicas y que interactúen con la base de datos en la que anteriormente debo haber introducido datos de prueba. En mi caso emplearé PHP con Laravel para dinamizar toda la aplicación web.
8. Testeo de la aplicación web: Una vez tenemos la web lista, tocará testearla para probar su correcto funcionamiento, su estabilidad y robustez ante el uso malintencionado de la misma.
9. Despliegue de la aplicación: Una vez que la aplicación es completamente funcional y se ha probado cada apartado a ser publicado, pondré en el entorno de producción la aplicación funcionando.

Planificación

Lo primero que se debe hacer al comenzar un proyecto es planificarlo. Iremos paso a paso planificando el proyecto más en detalle.

Análisis: Requisitos funcionales

Dividiré los requisitos en dos partes, los requisitos funcionales y los no funcionales.

Requisitos funcionales:

- El usuario podrá ver un listado con las películas de la base de datos.
- El usuario podrá buscar una película de la base de datos.
- El usuario podrá registrarse proporcionando su nombre, correo y contraseña.
- El usuario podrá iniciar sesión empleando su correo y contraseña.
- El usuario podrá cerrar sesión.
- El usuario podrá marcar las películas que ha visto.
- El usuario podrá marcar las películas que le han gustado.
- El usuario podrá marcar las películas que no le han gustado.
- El usuario podrá marcar las películas que quiere ver.
- El usuario podrá ver el detalle de cada película (sinopsis, año, género...).

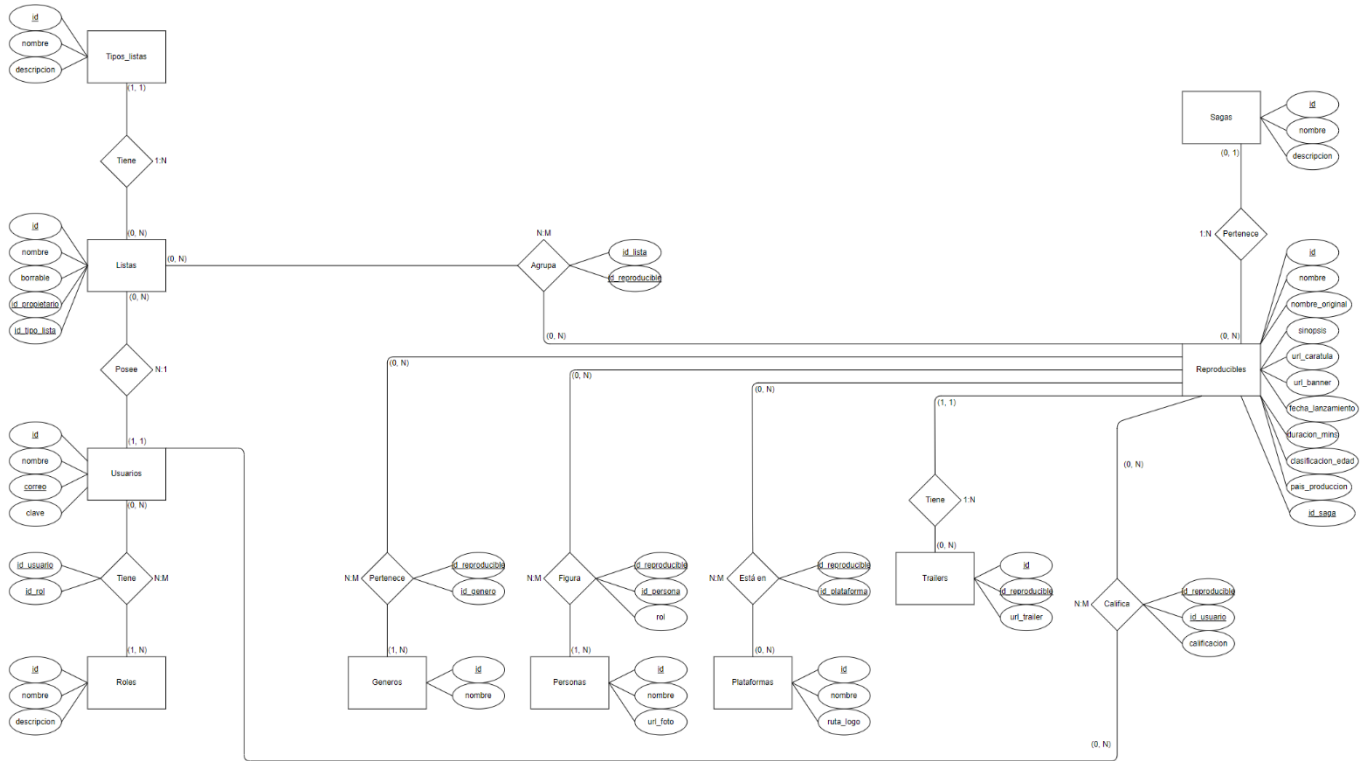
Requisitos no funcionales:

- La aplicación web debe tener una interfaz gráfica moderna, minimalista y atractiva visualmente.
- La interfaz gráfica debe ser intuitiva y sencilla de utilizar.
- La aplicación web no debe tardar más de 3 segundos en cargar cualquier página.
- La aplicación web debe cumplir con una alta disponibilidad.
- La aplicación web debe ser responsive.
- La aplicación web debe ser mantenible y actualizable a largo plazo (aplicando MVC por ejemplo).
- El código debe estar documentado y versionado en GitHub.

Diseño: Modelo Entidad-Relación (MER)

Ahora mostraré los diagramas que he hecho para la base de datos de la aplicación.

Modelo conceptual



En el modelo conceptual muestro los datos que existen y las relaciones que tienen los datos almacenados en la aplicación. Como actualmente sólo se almacenan películas, todos estos datos figuran en torno a la gestión de películas pero sigue siendo flexible para cuando haga falta añadir otro tipo de contenidos como series, libros, videojuegos, etc. Comienzo con la explicación de cada tabla y relación.

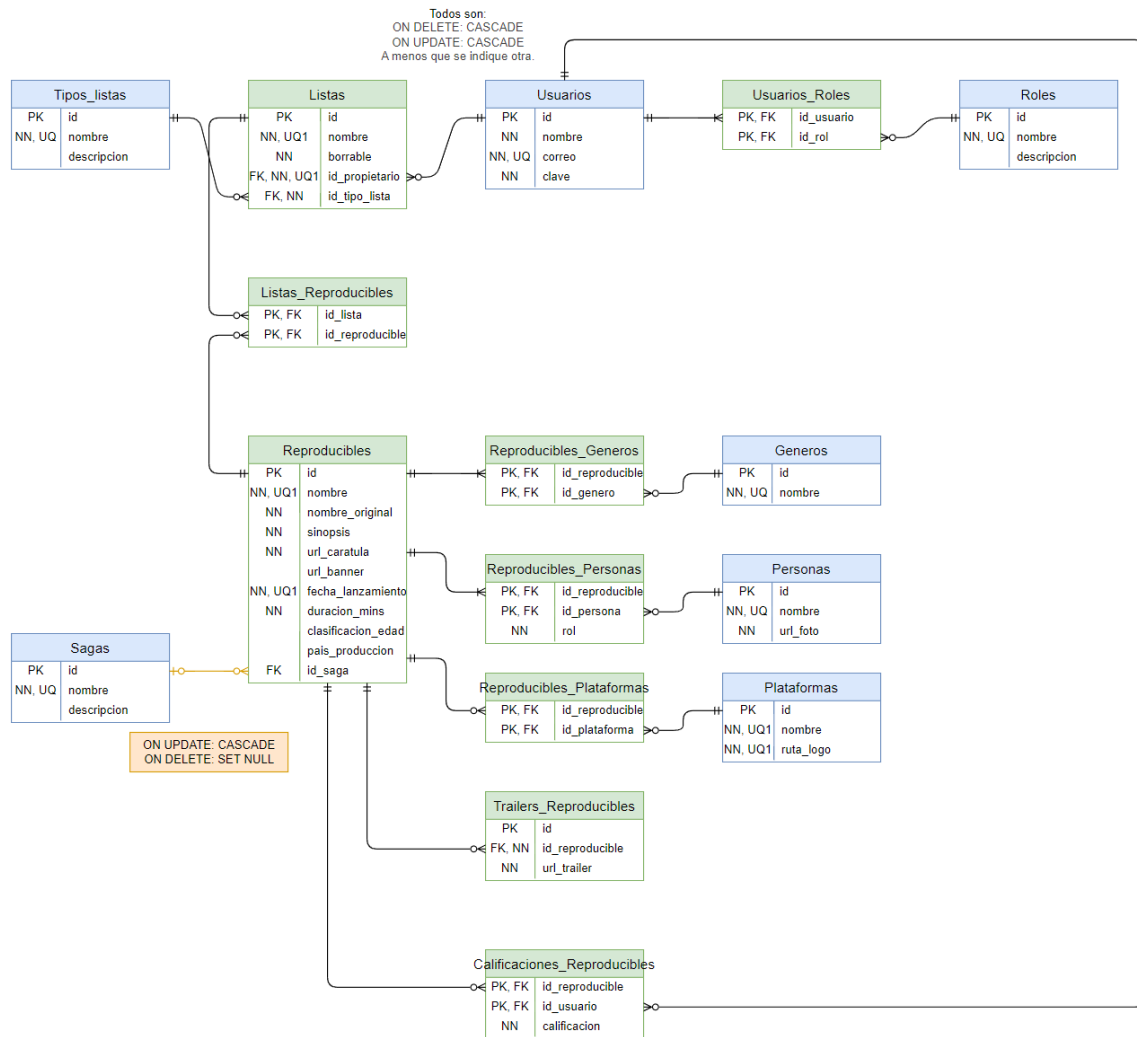
1. La entidad “Tipos_listas” almacena el tipo de elementos que se podrá almacenar en una lista (esto se gestionará empleando la lógica de programación en el backend de la aplicación para cuando hayan más tipos de contenido a parte de películas), es decir, si en una lista se podrán guardar películas, series, libros o incluso varios de estos tipos (esta tabla existe planificando todo de cara al futuro cuando se expanda más la aplicación). Esta entidad tiene los atributos id que es la clave primaria, además almacena el nombre del tipo de lista y una descripción.

2. La entidad “Tipos_listas” se relaciona con la entidad “Listas” siendo una relación 1:N; un tipo de lista puede estar presente en varias listas, pero una lista sólo tiene un tipo de lista.
3. La entidad “Listas” es la que almacena las listas de cada usuario. Cada lista tiene un id que es su clave primaria, un nombre para que el usuario la pueda identificar por su nombre, y un valor que indica si esta es borrrable o no; este valor ha sido creado ya que como hemos planteado antes, el usuario tendrá varias listas que serán creadas por defecto para cada usuario, es decir, listas que no deberían de ser borradas/borrables, pero a parte tengo pensado que el usuario pueda crear en alguna futura actualización sus propias listas, y esas listas creadas por el usuario sí que podrán ser borradas por el usuario, de forma que este atributo sirve para distinguir si el usuario propietario de esa lista la puede borrar o no. Además de los valores ya mencionados anteriormente en la entidad “Lista” se almacenan dos claves foráneas, el id del usuario propietario de esa lista, y el id del tipo de lista.
4. La entidad “Listas” se relaciona con la entidad “Usuarios” siendo una relación 1:N; un usuario posee varias listas, pero una lista sólo le pertenece a un usuario.
5. La entidad “Usuarios” es la que almacena los datos de cada usuario. En esta entidad tenemos un id que es su clave primaria y además almacenamos el nombre del usuario, su correo electrónico y su clave (su contraseña).
6. La entidad “Usuarios” se relaciona con la entidad “Roles” siendo una relación N:M; un usuario tiene al menos un rol pero pudiendo tener varios, y un rol lo puede tener ninguno o varios usuarios. Para relacionarlos debemos tener el id del usuario y el id del rol.
7. La entidad “Listas” se relaciona con la entidad “Reproducibles” siendo una relación N:M; una lista almacena ninguno o varios reproducibles (películas), y una película puede estar en ninguna o varias listas.
8. La entidad “Reproducibles” es en la que se almacenan las películas que se puede consultar en la base de datos. En esta entidad tenemos los atributos id, que es su clave primaria, el nombre de la película en España, el nombre original de la película, su sinopsis, la url de su carátula, la url del banner de la película, su fecha de estreno/lanzamiento, la duración de la película en minutos la clasificación de edad (la restricción de edad que tienen las películas, como por ejemplo que una película no se recomienda para menores de 16 años o para menores de 18 años...), el país de producción y el id de la saga a la que pertenece esa película (si se aplica).

9. La entidad “Reproducibles” se relaciona con la entidad “Sagas” siendo una relación 1:N; una película puede pertenecer a ninguna o como máximo una saga, pero a una saga le puede pertenecer ninguna o múltiples películas.
10. La entidad “Sagas” es la que almacena las sagas que existen en la base de datos, una saga puede ser por ejemplo “Star Wars” o “Spider-man”. Esto nos serviría para agrupar posteriormente las películas por sagas (simplemente esto es para crear algún filtro en un futuro). La entidad “Sagas” tiene los atributos id que es su clave primaria, el nombre de la saga, y una descripción.
11. La entidad “Reproducibles” se relaciona con la entidad “Generos” siendo una relación N:M; un reproducible (o película) pertenece/tiene 1 o más géneros, mientras que un género tiene ninguna o más películas que se identifican con ese género. Para relacionarlas empleamos el id del reproducible y el id de género.
12. La entidad “Generos” almacena los géneros cinematográficos con los que se puede catalogar cada reproducible/película/serie/libro/videojuego... Esta entidad tiene los atributos id que es su clave principal, y el nombre del género. De cara a futuro esta entidad puede almacenar los géneros con los que identificar incluso canciones, de forma que aseguramos que la base de datos sea más flexible aún.
13. La entidad “Reproducibles” se relaciona con la entidad “Personas” siendo una relación N:M; en un reproducible figura al menos una persona y pueden figurar varias (a parte del director podemos tener el reparto), y una persona puede figurar en ningún o varios reproducibles. Para relacionar estas entidades utilizamos el id del reproducible, el id de la persona a relacionar y el rol que ha desempeñado esta persona en la película (director, actor...).
14. La entidad “Personas” almacena todas las personas que se relacionan con alguna de las películas que tenemos almacenadas en la base de datos. Esta entidad tiene los atributos id que es su clave primaria, el atributo nombre que es el nombre de la persona y una url_foto que es un enlace de una fotografía del actor/actriz/director... De cara a futuro esta entidad es completamente reutilizable para almacenar cantantes, productores o escritores entre otros, asegurando que la base de datos sea muy flexible.
15. La entidad “Reproducibles” se relaciona con la entidad “Plataformas” siendo una relación N:M; un reproducible se encuentra en ninguna o múltiples plataformas de la base de datos, y una plataforma tiene ningún o múltiples de los reproducibles de la base de datos. Para relacionar estas tablas utilizamos el id de reproducible y el id de plataforma.

16. La entidad “Plataformas” almacena todas las plataformas de streaming en las que puedes ver las películas, e incluso almacena los cines en los que puedes ir a ver la película en caso de ser muy reciente. Esta entidad tiene los atributos id que es su clave primaria, el nombre de la plataforma o medio y la ruta hasta el logo de la plataforma descargado en el servidor que hostea la página. De cara a futuro esta entidad se puede reutilizar para almacenar las plataformas en las que podamos encontrar determinados libros, series, etc, con lo que aseguramos que la base de datos sea flexible.
17. La entidad “Reproducibles” se relaciona con la entidad “Trailers” siendo una relación 1:N; un reproducible tiene ningún o múltiples tráilers, mientras que un tráiler únicamente pertenece a un reproducible.
18. La entidad “Trailers” almacena los tráilers de las películas. Esta entidad tiene los atributos id que es su clave primaria, el id del reproducible al que pertenece y la url del tráiler, es decir, un link/enlace que lleva hasta el vídeo que es el tráiler de la película.
19. La entidad “Reproducibles” se relaciona con la entidad “Usuarios” siendo una relación N:M; un usuario califica ningún o múltiples reproducibles y un reproducible es calificado por ninguno o múltiples usuarios. Para esta relación empleamos el id del reproducible calificado, el id del usuario que lo califica y la calificación/valoración del usuario, con la nota con la que lo puntúa.

Modelo lógico



En el modelo lógico muestro una estructura más cercana a la base de datos real mostrando las tablas a crear, sus campos y restricciones en cada campo en vez de seguir hablando de entidades y sus atributos. Como es de esperar, este modelo viene partiendo del modelo conceptual que había en el punto anterior. Como se indica en la captura de pantalla, todas las relaciones tienen “ON DELETE CASCADE” y “ON UPDATE CASCADE”, a menos que se indique otra. Procedo con la explicación de este modelo relacional.

1. La tabla “Tipos_listas” almacena los tipos de contenidos que puede llegar a almacenar una lista; tiene los campos:
 - a. id: Es la clave primaria
 - b. nombre: No puede ser nulo, ya que sino no tendría sentido crear un registro de esta tabla, además debe ser único, para que un administrador al ver el nombre pueda distinguirlos fácilmente y unívocamente.
 - c. descripcion: Campo para describimos ese tipo de lista.

2. La tabla “Usuarios” almacena los datos de los usuarios registrados en la aplicación; tiene los campos:
 - a. id: Es la clave primaria.
 - b. nombre: No puede ser nulo, ya que es el nombre por el que nombrar al usuario registrado.
 - c. correo: Es el correo electrónico del usuario con el que se puede identificar, de forma que no puede ser nulo y además no puede ser repetido, por lo que debe ser único.
 - d. clave: Es la contraseña del usuario, la cual esperablemente no puede ser nula.
3. La tabla “Roles” almacena los roles que puede tener un usuario; tiene los campos:
 - a. id: Es la clave primaria
 - b. nombre: No puede ser nulo y debe ser único, para que cada rol sea identificable a nivel de interfaz gráfica, ya que identificar cada rol por su id, para nosotros como humanos nos resultaría muy ineficiente.
 - c. descripcion: Campo para poder describir cada rol.
4. La tabla “Listas” almacena todas las listas de cada usuario; tiene los campos:
 - a. id: Es la clave primaria.
 - b. nombre: No puede ser nulo para que un usuario pueda identificar a nivel de interfaz gráfica cada lista.
 - c. borrrable: Como comenté en el modelo conceptual, es para indicar si una lista podría llegar a ser borrada por un usuario (este campo es para futuras actualizaciones, no específicamente para lo que llegaré a presentar en este TFG, aunque si me da tiempo lo implementaré también).
 - d. id_propietario: Es una clave foránea que viene de la tabla “Usuarios” y no puede ser nula ya que sino una lista podría no pertenecerle a nadie.
 - e. id_tipo_lista: Es una clave foránea que viene de la tabla “Tipos_listas” y no puede ser nulo ya que sino una lista podría no tener un tipo, y busco que justamente tenga un tipo para poder limitar el tipo de contenido que puede almacenar esa lista específica.
 - f. El conjunto de los campos “nombre” e “id_propietario” deben ser únicos, es decir, un mismo usuario no puede tener dos listas con el mismo nombre.

5. La tabla “Usuarios_Roles” relaciona a cada usuario con sus roles; tiene los campos:
 - a. id_usuario: Es una clave foránea que viene de la tabla “Usuarios”.
 - b. id_rol: Es una clave foránea que viene de la tabla “Roles”.
 - c. El conjunto de ambos campos forman la clave primaria.
6. La tabla “Listas_Reproducibles” relaciona cada lista con los reproducibles que contienen éstas; tiene los campos:
 - a. id_lista: Es una clave foránea que viene de la tabla “Listas”.
 - b. id_reproducible: Es una clave foránea que viene de la tabla “Reproducibles”.
 - c. El conjunto de ambos campos forman la clave primaria.
7. La tabla “Reproducibles” almacena las películas de la base de datos; tiene los campos:
 - a. id: Es la clave primaria.
 - b. nombre: No puede ser nulo, ya que sino estaríamos guardando una película que no tiene nombre en España.
 - c. nombre_original: No puede ser nulo, ya que sino estaríamos almacenando una película que no tiene nombre en su país de producción.
 - d. sinopsis: No puede ser nulo ya que para una aplicación que consiste en la consulta de información relativa a películas la sinopsis es uno de los campos clave a revisar por un usuario.
 - e. url_caratula: No puede ser nulo ya que la carátula nos permite reconocer instantáneamente una película y distinguirla de otras o incluso nos permite saber información acerca de qué tipo de película se trata con tan solo ver la portada, algo clave para una aplicación que quiere permitirte descubrir películas que no has visto antes y que podrían interesarte.
 - f. url_banner: Puede ser nulo, ya que no es información vital, sino más bien estética para cuando se vaya a representar la película en la interfaz gráfica de la aplicación web.
 - g. fecha_lanzamiento: No puede ser nulo ya que hay algunos casos en los que hay películas que se llaman con el mismo nombre y la forma de distinguirlas es por la fecha en la que salen al mercado.
 - h. duracion_mins: No puede ser nulo ya que muchas veces cuando un usuario busca una película quiere saber cuánto tiempo durará ésta, y esta información debería ser siempre accesible en todas las películas para esas veces en las que el usuario tiene tiempo algo más limitado para ocio.

- i. `clasificacion_edad`: Puede ser nulo ya que honestamente no todo el mundo se fija en el límite de edad para una película y no lo considero uno de los datos más relevantes, además de que con películas de estreno a veces la clasificación de edad suele estar indefinida hasta que finalmente determinan para qué edad corresponde esa película.
 - j. `pais_produccion`: Puede ser nulo ya que el país de donde venga una película no suele ser un dato tremendamente relevante, sino que viene a ser un dato más bien curioso.
 - k. `id_saga`: Es una clave foránea que viene de la tabla “Sagas”.
 - l. El conjunto de los campos “nombre” y “fecha_lanzamiento” deben ser únicos, es decir, no puede haber dos películas con el mismo nombre que salgan en la misma fecha, ya que literalmente sería la misma película duplicada en la tabla, y eso es algo a evitar a toda costa.
8. La tabla “Sagas” identifica una saga de películas; tiene los campos:
- a. `id`: Es la clave primaria.
 - b. `nombre`: No puede ser nulo ya que no tiene sentido almacenar una saga sin nombre, no tendría sentido la existencia de una saga así, y debe ser único, ya que no tendría sentido almacenar dos sagas con el mismo nombre, es decir, almacenar dos veces la misma saga.
 - c. `descripcion`: Puede ser nulo; es un campo para describir una saga.
9. La tabla “Generos” almacena todos los géneros a los que puede pertenecer una película; tiene los campos:
- a. `id`: Es la clave primaria.
 - b. `nombre`: No puede ser nulo, ya que no tiene sentido un género sin nombre, y debe ser único ya que no tiene sentido almacenar dos géneros con el mismo nombre (es decir, el mismo género pero con distinto id).
10. La tabla “Reproducibles_Generos” relaciona cada película con los géneros a los que pertenece; tiene los campos:
- a. `id_reproducible`: Es una clave foránea que viene de la tabla “Reproducibles”.
 - b. `id_genero`: Es una clave foránea que viene de la tabla “Generos”.
 - c. El conjunto de ambos campos forman la clave primaria.

11. La tabla “Personas” almacena todas las personas que tienen relación con las películas de la base de datos; tiene los campos:
- a. id: Es la clave primaria.
 - b. nombre: No puede ser nulo, ya que no tiene sentido almacenar una persona sin nombre por el que identificarla, además, debe ser único ya que si intento introducir un nombre y apellidos que ya existe podré ver que aparece un error de que ya existe en la base de datos un usuario con ese nombre y apellidos, de forma que puedo comprobar si esa persona que ya está almacenada es la misma que justamente estoy tratando de almacenar ahora en la base de datos, y en caso de que sean distintas personas con el mismo nombre y apellidos simplemente añadido algún número o coletilla para distinguirlos.
 - c. url_foto: No puede ser nulo ya que la idea es ofrecer en la interfaz gráfica el nombre de la persona junto con su retrato.
12. La tabla “Reproducibles_Personas” relaciona cada película con todas las personas que trabajaron en ella; tiene los campos:
- a. id_reproducible: Es una clave foránea que viene de la tabla “Reproducibles”.
 - b. id_persona: Es una clave foránea que viene de la tabla “Personas”.
 - c. rol: No puede ser nulo ya que la idea de este campo es que indique qué rol cumplía la persona que se está relacionando con la película, es decir, si esta persona era por ejemplo actor o director en la película.
 - d. El conjunto de “id_reproducible” y “id_persona” forman la clave primaria.
13. La tabla “Plataformas” almacena las plataformas de streaming y cines en los que se pueden reproducir películas de la base de datos; tiene los campos:
- a. id: Es la clave primaria.
 - b. nombre: No puede ser nulo ya que no tiene sentido almacenar una plataforma que no tiene nombre.
 - c. ruta_logo: No puede ser nulo ya que la idea es mostrar en la interfaz gráfica de la aplicación el logo de la plataforma o cine en la que se encuentra. En este caso lo que se almacena es la ruta hasta la imagen del logo en vez de una url, y es que los logos de las plataformas directamente los he descargado para que se representen en la aplicación.
 - d. El conjunto del campo “nombre” y el campo “ruta_logo” deben ser únicos, es decir, no pueden aparecer dos plataformas con el mismo nombre y logo. He decidido que esta restricción dado que hay algún caso en el que me conviene

que se repita el logo de una plataforma, pero con distinto logo, y es que hay casos como el de plataformas como Amazon Prime Video, en la que tienes sub suscripciones dentro de la suscripción que te brindan acceso a más contenido de otras productoras pero que en sí se encuentra dentro de Prime Video.

14. La tabla “Reproducibles_Plataformas” relaciona las películas con las plataformas en las que se encuentra disponible para visualizarla; tiene los campos:
 - a. id_reproducible: Es una clave foránea que viene de la tabla “Reproducibles”.
 - b. id_plataforma: Es una clave foránea que viene de la tabla “Plataformas”.
 - c. El conjunto de ambos campos forman la clave primaria.
15. La tabla “Trailers_Reproducibles” almacena y relaciona cada película con sus tráilers; tiene los campos:
 - a. id: Es la clave primaria.
 - b. id_reproducible: Es una clave foránea que viene de la tabla “Reproducibles”.
 - c. url_trailer: Es el link/enlace de YouTube que lleva al tráiler de la película a la que se le está añadiendo un tráiler. No puede ser nulo porque este es el campo por el que se crea un registro de esta tabla, es decir, no tiene sentido registrar un tráiler vacío.
16. La tabla “Calificaciones_Reproducibles” almacena y relaciona las películas con las calificaciones que le dan los usuarios; tiene los campos:
 - a. id_reproducible: Es una clave foránea que viene de la tabla “Reproducible”.
 - b. id_usuario: Es una clave foránea que viene de la tabla “Usuarios”.
 - c. calificacion: No puede ser nulo dado que se trata de la calificación que le da el usuario, por lo que no tendría sentido que este campo pudiese ser nulo, no nos serviría almacenar una calificación nula.
 - d. El conjunto de los campos “id_reproducible” e “id_usuario” forman la clave principal.

Modelo físico

El modelo físico lo entrego junto al código del proyecto; es el fichero modelo_fisico.sql.

Adjunto el modelo_fisico.sql también por el aula virtual.

Despliegue

Ahora hablaré un poco acerca de la implementación de la aplicación en el servidor en el que ésta está hosteada, de forma que si queremos replicar la configuración, podríamos hacerlo en una máquina virtual de Linux; en el servidor estoy utilizando específicamente Ubuntu (sin interfaz gráfica) y lo estoy configurando utilizando en el servidor un servidor SSH y en mi máquina un cliente SSH para conectarme al servidor. Podríamos utilizar XAMPP perfectamente para desplegar esta aplicación (de hecho es la herramienta que utilizo en el entorno de desarrollo para hacer los cambios y las pruebas antes de subirlas al servidor) pero los pasos a seguir serían ligeramente distintos.

Configuración del servidor

En el caso de estarlo haciendo en una máquina virtual yo optaría por configurar los puertos de red en modo adaptador puente, ya que así podríamos acceder a los servicios que provee nuestro servidor desde otros dispositivos de la red.

Una vez que tenemos un servidor Ubuntu funcionando, lo primero que haremos será instalar todo lo que necesitaremos para la aplicación web.

Instalación de requisitos

1. Lo primero que haremos será actualizar el sistema. Para esto emplearemos el siguiente comando:

- a. `sudo apt update && sudo apt upgrade -y`

2. Instalaremos PHP junto a sus extensiones: Para instalar PHP junto a sus extensiones simplemente tendremos que ejecutar el siguiente comando:
 - a. `sudo apt install php php-cli php-mbstring php-xml php-bcmath php-curl php-mysql php-zip unzip -y`
 - b. `php -v` # Para comprobar la versión instalada, si no aparece error sabremos que se ha instalado correctamente.
3. Instalaremos composer, que es un gestor de paquetes de PHP con los siguientes comando:
 - a. `sudo cd ~`
 - b. `sudo php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"`
 - c. `sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer`
 - d. `composer --version` # Para comprobar qué versión se ha instalado; si no aparece ningún error entonces habremos instalado correctamente composer.
4. Instalamos Node.js y npm (lo necesitamos para Vite) con los siguientes comandos:
 - a. `curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -`
 - b. `sudo apt install -y nodejs`
 - c. `node -v && npm -v` # Comprobamos qué versiones se nos han instalado; si no da ninguna de las dos error, habremos instalado correctamente node con npm.
5. Instalamos GIT, para poder clonar el repositorio de github en el que he subido toda la aplicación; empleamos los siguientes comandos:
 - a. `sudo apt install git -y`
 - b. `git --version` # Para comprobar la versión instalada; si no da error, la instalación habrá sido exitosa.
6. Instalamos Apache para publicar la aplicación web empleando el siguiente comando:
 - a. `sudo apt install apache2 libapache2-mod-php -y`
 - b. Accediendo desde un navegador a “localhost” deberemos ver una página de bienvenida a Apache, eso quiere decir que la instalación ha sido un éxito.
7. Ahora instalaremos MySQL para gestionar los datos de la base de datos con el siguiente comando:
 - a. `sudo apt install mysql-server`

Con todos estos pasos ya tendremos todos los requisitos instalados, ahora simplemente tendremos que ponerlo todo a funcionar configurando lo que nos haga falta.

Configuración de instalaciones

Lo primero que haremos será limpiar la carpeta en la que se encuentra la página por defecto de Apache ya que no los usaremos ni los necesitaremos más. Simplemente la vaciaremos empleando el siguiente comando:

```
sudo rm -R /var/www/html/*
```

Una vez hecho esto, vamos a clonar el repositorio de GitHub en el que tenemos la aplicación y todos los datos necesarios para hacerla funcional completamente.

```
cd /var/www/html/
```

```
sudo git clone https://github.com/Edsonbs/justtrack.git
```

Ahora tenemos nuestro proyecto de GitHub clonado en la carpeta de sitios web de Apache. El siguiente paso es configurar Apache para que pueda mostrar correctamente la aplicación que ahora hay en la ruta `/var/www/html/justtrack`. Lo que haremos es crear un archivo de configuración para este sitio web:

```
sudo cp -R /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/justtrack.conf
```

Con esto ya tenemos un archivo de configuración que de momento tiene los valores por defecto. Lo que tendremos que hacer ahora es cambiar la configuración de ese archivo por defecto para que ahora apunte correctamente al sitio web correspondiente.

```
sudo nano /etc/apache2/sites-available/justtrack.conf
```

Ahora, dentro de esta configuración tendremos que aplicar los siguientes cambios:

1. "DocumentRoot" ahora debe apuntar a `"/var/www/html/justtrack/public"`.
2. Debemos añadir esta línea para que el servidor busque los ficheros `index.html` e `index.php`: `DirectoryIndex index.html index.php`
3. Debemos añadir estas líneas para configurar los permisos sobre la carpeta:

```
<Directory /var/www/html/justtrack/public>
```

```
Options Indexes FollowSymLinks
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

Ahora tendremos que deshabilitar la configuración del sitio por defecto para habilitar la configuración que acabamos de crear, además que deberemos habilitar varios módulos de Apache para que funcione correctamente nuestro proyecto Laravel. Finalmente tendremos que reiniciar Apache para aplicar estos cambios:

```
sudo a2dissite 000-default.conf
```

```
sudo a2ensite justtrack.conf
```

```
sudo a2enmod rewrite headers env
```

```
sudo service apache2 restart
```

Ahora hacemos que Apache sea el propietario de la carpeta del proyecto y todo lo que contiene y damos permisos de escritura para Apache.

```
sudo chown -R www-data:www-data /var/www/html/justtrack
```

```
sudo chmod -R 775 /var/www/html/justtrack
```

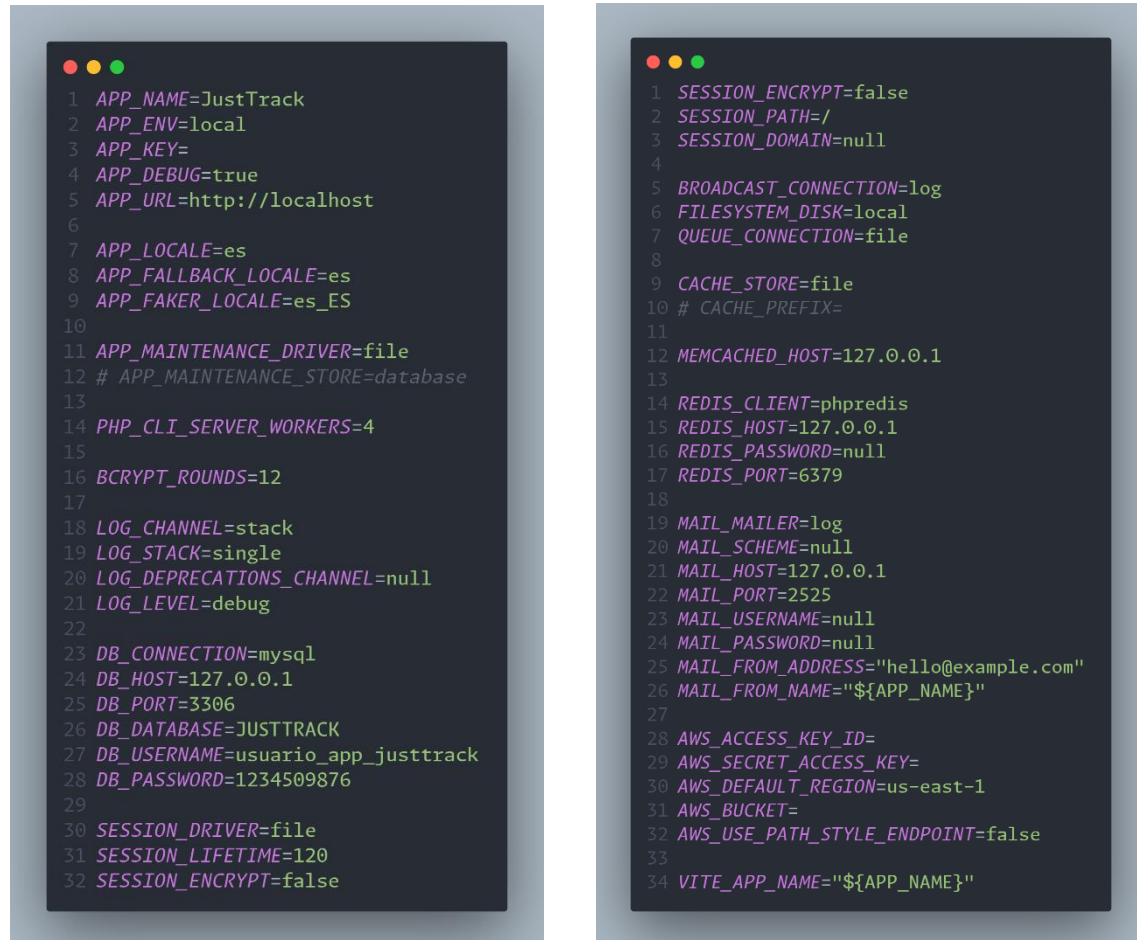
Ahora tendremos que crear el `.env` del proyecto.

```
sudo cp /var/www/html/justtrack/.env.ejemplo /var/www/html/justtrack/.env
```

Una vez creado el .env lo tendremos que modificar para ajustarlo a nuestra configuración:

```
sudo nano /var/www/html/justtrack/.env
```

Nos debe quedar esta configuración:



Una vez tenemos esta configuración en nuestro fichero “.env”, tendremos que instalar todas las dependencias de PHP del proyecto; accedemos a la carpeta del proyecto y ejecutamos el comando de composer:

```
cd /var/www/html/justtrack/
```

```
sudo composer install
```

Ahora tendremos que generar una clave para nuestra aplicación ejecutando el siguiente comando:

```
sudo php artisan key:generate
```

Ahora tendremos que instalar todas las dependencias JavaScript del proyecto:


```
cd /var/www/html/justtrack/
```

```
sudo npm install
```

Ahora crearemos un enlace a los elementos almacenados en la carpeta “resources/storage/app/public”, para que la aplicación tenga acceso a los elementos visuales descargados localmente en el proyecto:

```
sudo php artisan storage:link
```

Finalmente compilaremos con Vite nuestros ficheros CSS y JS:

```
sudo npm run build
```

Con todos estos pasos finalmente tendremos la aplicación ya “funcional”, lo pongo entre comillas porque como aún tengo que crear la parte de base de datos la aplicación aún no puede hacer nada, de forma que ese es el próximo punto a tratar, la gestión de datos.

Creación de base de datos e inserción de datos de pruebas

Comenzaremos con la creación de la base de datos de la aplicación web. En el repositorio de GitHub que hemos clonado en el paso anterior se encuentran varios ficheros relativos a la base de datos, éstos se encuentran en “documentación_proyecto/base_datos”; en esa carpeta tenemos tanto el diagrama del modelo conceptual como el diagrama del modelo lógico, y finalmente el modelo_fisico.sql, que contiene todas las sentencias necesarias para crear toda la base de datos con todas sus tablas. Finalmente tenemos el fichero eliminar_base_datos.sql que como su nombre indica, es para eliminar la base de datos. Realmente la podríamos eliminar sólo con el comando “DROP DATABASE JUSTTRACK”, pero ahí tenemos en ese fichero en qué orden deben eliminarse las tablas si por algún motivo necesitamos eliminar alguna tabla.

Ahora comenzamos con el sencillo proceso para crear la base de datos e insertarle todos los valores de prueba que tenemos.

Lo primero que haremos es crear una contraseña para el usuario root de la base de datos (como MySQL está recién instalado, aún no tiene configuración alguna). Para esto primero accederemos al sistema gestor de base de datos MySQL:

```
sudo mysql
```

Una vez aquí dentro, ejecutamos la siguiente línea (reemplazamos “1234” con la contraseña que deseamos para ese usuario, en mi caso como es un ejemplo mostrando cómo desplegar la app, usará 1234 mismamente):

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '1234';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

Una vez que tenemos esto configurado vamos a crear la base de datos con todas sus tablas y el usuario con su contraseña y permisos para la aplicación simplemente empleando el script que he comentado anteriormente:

```
mysql -u root -p <
```

```
/var/www/html/justtrack/documentacion_proyecto/base_datos/modelo_fisico.sql
```

Se nos solicitará la contraseña del usuario root al que acabamos de asignarle una contraseña. La introducimos y ya tendremos la base de datos creada. Podemos comprobar que se ha creado tanto la base de datos como todas las tablas ejecutando los siguientes comandos:

```
mysql -u root -p1234
```

```
SHOW DATABASES;
```

```
USE JUSTTRACK;
```

```
SHOW TABLES;
```

Ahora procederemos a insertar los datos de prueba de la base de datos:

```
mysql -u root -p <
```

```
/var/www/html/justtrack/documentacion_proyecto/base_datos/insercion_datos.sql
```

Se nos solicitará nuevamente la contraseña del usuario root; tras introducir la contraseña esperaremos un par de segundos y ya tendremos nuestra base de datos nutrida de datos de pruebas. Podremos comprobar que ahora las tablas tienen datos empleando los siguientes comandos:

```
mysql -u root -p1234
```

```
USE JUSTTRACK;
```

```
SELECT * FROM REPRODUCIBLES;
```

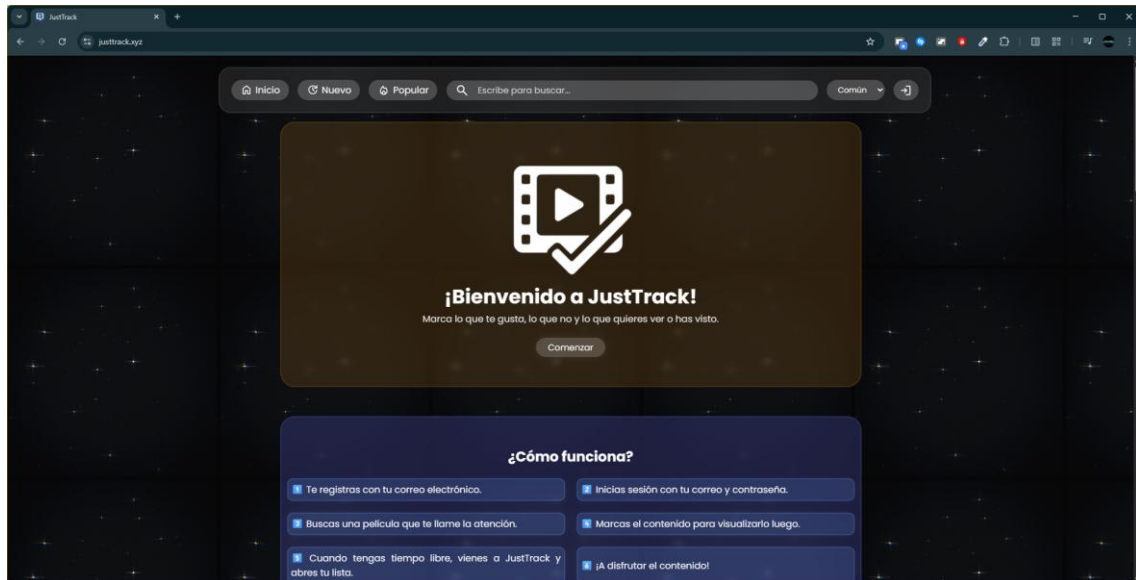
A partir de aquí simplemente ejecutamos esa última línea de SELECT con todas las tablas que queramos comprobar.

Tras haber seguido todas estas líneas explicadas más arriba ya tendremos la aplicación web completamente funcional. Ya podremos interactuar con absolutamente todo y ya la aplicación web será funcional.

Manual de usuario

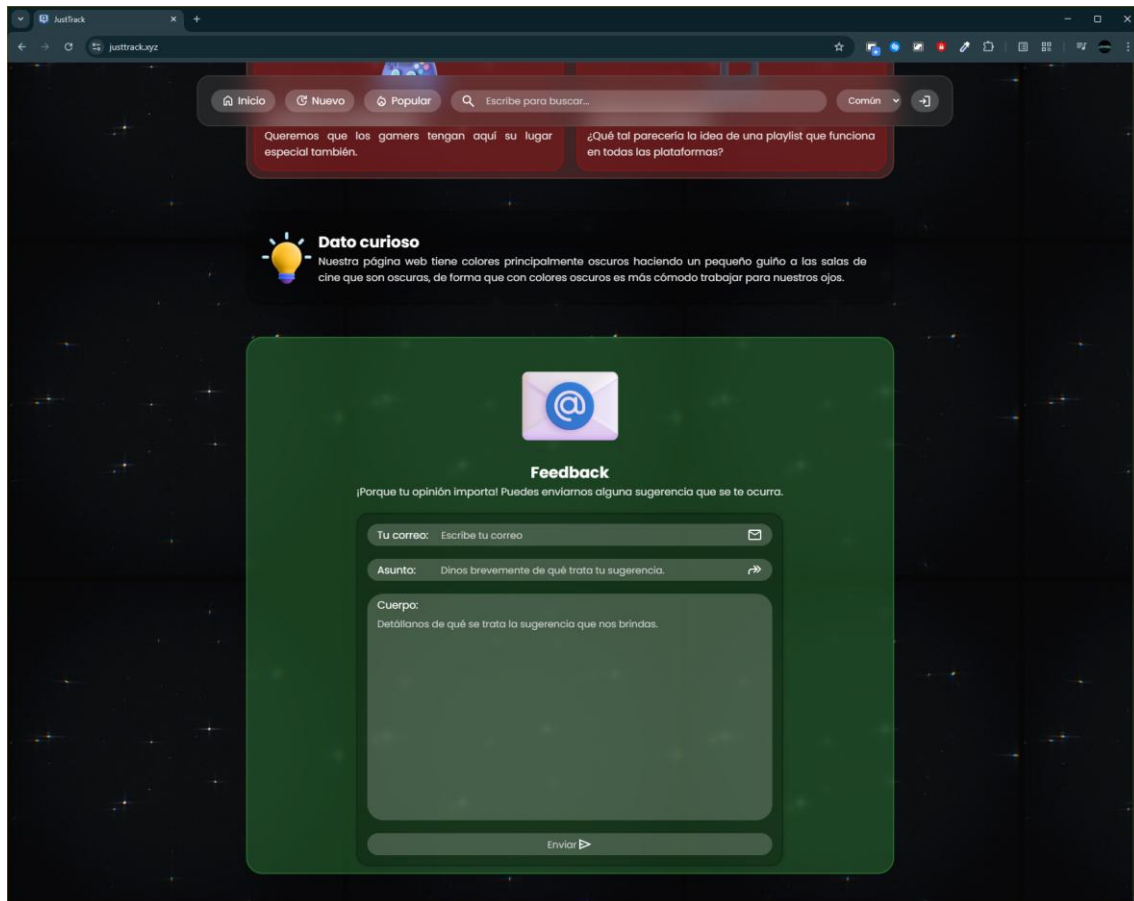
Ahora comenzaré con lo que es la explicación cómo utilizar mi aplicación web y qué es cada apartado.

Apartado Inicio



Cuando abrimos la aplicación web lo primero que veremos es lo que se ubica en la captura de pantalla que se ubica justo encima de este texto (la interfaz gráfica del sitio web es susceptible a cambios, pero la estructura será muy similar sino idéntica). Como se puede apreciar a la pantalla que da la bienvenida a la plataforma. Esta parte se nos explica un poco qué es la plataforma y nos indica cómo funciona y qué es cada parte. Hay además un apartado en el que se nos explican los planes a futuro de la aplicación web.

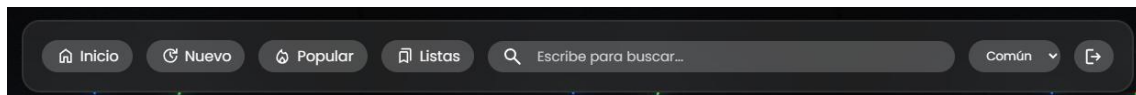
Se trata simplemente de una página de introducción para que el usuario se pueda ubicar. Es una página que simplemente pretende que el usuario se sienta cómodo con el sitio web, ya que generalmente cuando conocemos bien las herramientas que utilizamos y se nos hace sencillo de utilizar somos propensos a usarlas más, y eso es lo que yo buscaría con esto.



En la parte final de este apartado de inicio veremos un formulario como el que se muestra en la captura de pantalla que se ubica justo encima de este texto en el que el usuario podrá transmitirme alguna idea u opinión que tenga acerca de la plataforma, lo que me permitiría mejorar/cambiar la plataforma según las sugerencias de los usuarios según lo vea conveniente.

Como podemos ver en la captura de pantalla, el usuario tendría que introducir su correo electrónico o el correo con el que quiere comunicarse conmigo/con el soporte de la aplicación, posteriormente introducirá un asunto para el correo y finalmente introducirá un cuerpo, que es donde irá todo el texto que quiera redactar.

Cabecera

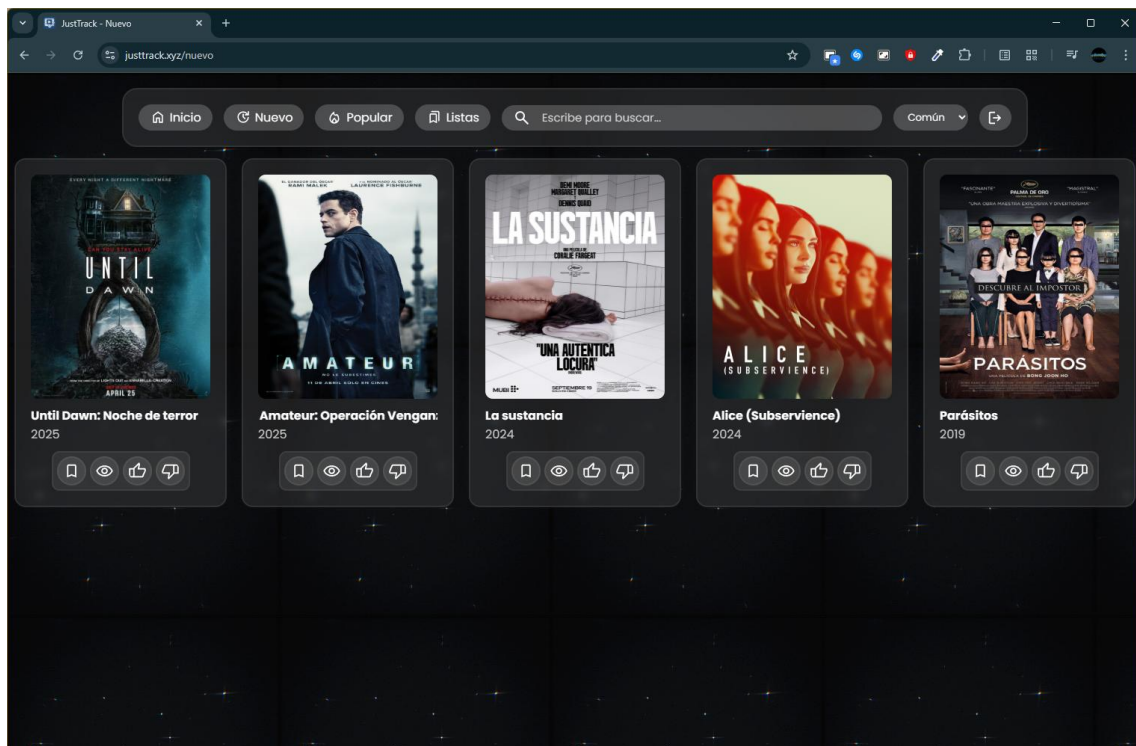


En la captura de pantalla que se encuentra justo encima de este texto podemos visualizar la cabecera que veríamos cuando iniciamos sesión con un usuario (hay elementos que no se pueden ver si no tenemos sesión iniciada, así que la cabecera con sesión iniciada es más completa que sin sesión iniciada). Veremos de izquierda a derecha todos los elementos que tenemos en la cabecera:

- **Enlace a la página “Inicio”:** La página que como comentaba anteriormente, es la primera página que verá el usuario al abrir la aplicación web. Es la página que explica lo que es la plataforma, qué es cada apartado y cómo utilizarla.
- **Enlace a la página “Nuevo”:** La página “Nuevo” es en donde tendremos todas las películas que hay en la plataforma con la peculiaridad de que estarán ordenadas por fecha de lanzamiento.
- **Enlace a la página “Popular”:** La página “Popular” muestra todas las películas que hay en la plataforma con la peculiaridad de estar ordenadas según cuántos me gusta hayan recibido las películas (también se toma en cuentas cuantos no me gusta hayan recibido).
- **Enlace a la página “Listas”:** La página “Listas” muestra las 4 listas básicas que tiene un usuario y le permite ver qué películas tiene en esa lista e incluso eliminarlas de esa lista o cambiarla a otra de sus 4 listas básicas “Visto”, “Me gusta”, “No me gusta” y “Watchlist”.
- **Barra de búsqueda:** En esta barra de búsqueda será en donde el usuario podrá introducir el nombre de una película (ya sea el nombre original o el nombre que recibe la película en España) y podrá pulsar la tecla “Enter” o hacer click sobre el logo de una lupa que aparece a la izquierda en la barra de búsqueda para realizar la búsqueda y que le aparezcan las películas que coincidan con lo que el usuario ha buscado.
- **Apartado de rol:** En este apartado el usuario puede ver el rol que tiene (en caso de que tengas más de 1 rol) y seleccionando uno de los roles de acceso que tienes podrás ver ciertas páginas a las que no todo el mundo puede acceder y que te permitirán hacer unas u otras cosas.

- **Botón de “Registrarse” y “Cerrar sesión”:** Es el botón con el que podremos acceder a la página de registrarnos; desde ahí, si ya tenemos una cuenta podremos ir a la página de inicio de sesión. Una vez iniciamos sesión el botón que nos llevaba a la página de registro se cambiará para mostrarnos un botón para cerrar sesión. Para saber si inicias o cierras sesión es tan simple como ver el icono, ya que empleo el icono “universal” de iniciar sesión o cerrar sesión.

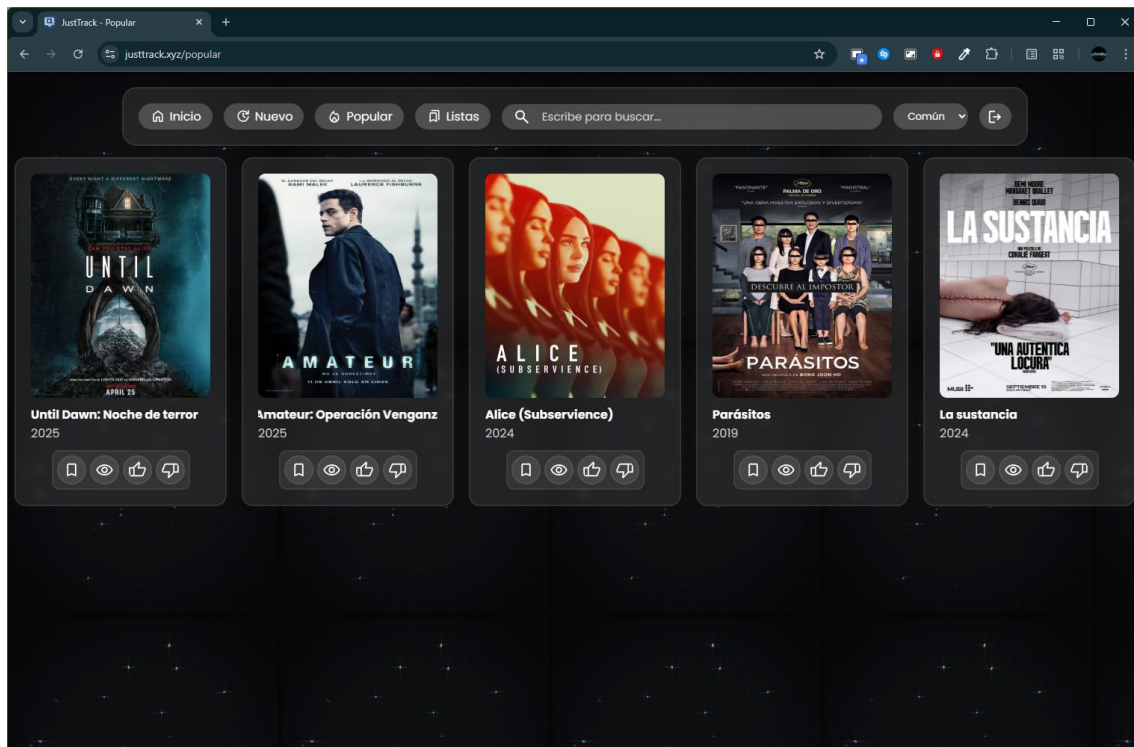
Apartado Nuevo



En este apartado de la aplicación web podremos ver todas las películas de la base de datos en orden de fecha de lanzamiento; veremos primero las películas que han salido en fechas más recientes.

Como podemos ver en la captura de pantalla que se encuentra justo encima de este texto, en cada película tenemos en la parte inferior una fila de botones. Con estos botones podremos marcar la película en varias de nuestras listas. El primer botón guardaría la película en nuestra lista “Watchlist”, es decir, películas que queremos ver, el segundo botón guardaría la película en nuestra lista “Visto”, es decir, la lista de películas que hemos visto, el tercer botón es para guardar la película en la lista de “Me gusta”, y el cuarto en la lista de “No me gusta”, es decir, la lista de películas que nos gusta o no. No se pueden pulsar estos dos últimos a la vez, ya que no tiene sentido.

Apartado “Popular”



En este apartado de la aplicación web podremos ver todas las películas de la base de datos en orden de la puntuación que ha recibido la película por los usuarios en el último mes; veremos primero las películas que han recibido las mejores puntuaciones en los últimos 30 días. En el caso de las películas que se encuentran sin datos de puntuaciones/opiniones, se representarán cronológicamente, al igual que en el apartado “Nuevo”.

Como podemos ver en la captura de pantalla que se encuentra justo encima de este texto, en cada película tenemos en la parte inferior una fila de botones (al igual que en el apartado “Nuevo”). Con estos botones podremos marcar la película en varias de nuestras listas. El primer botón guardaría la película en nuestra lista “Watchlist”, es decir, películas que queremos ver, el segundo botón guardaría la película en nuestra lista “Visto”, es decir, la lista de películas que hemos visto, el tercer botón es para guardar la película en la lista de “Me gusta”, y el cuarto en la lista de “No me gusta”, es decir, la lista de películas que nos gusta o no. No se pueden pulsar estos dos últimos a la vez, ya que no tiene sentido.

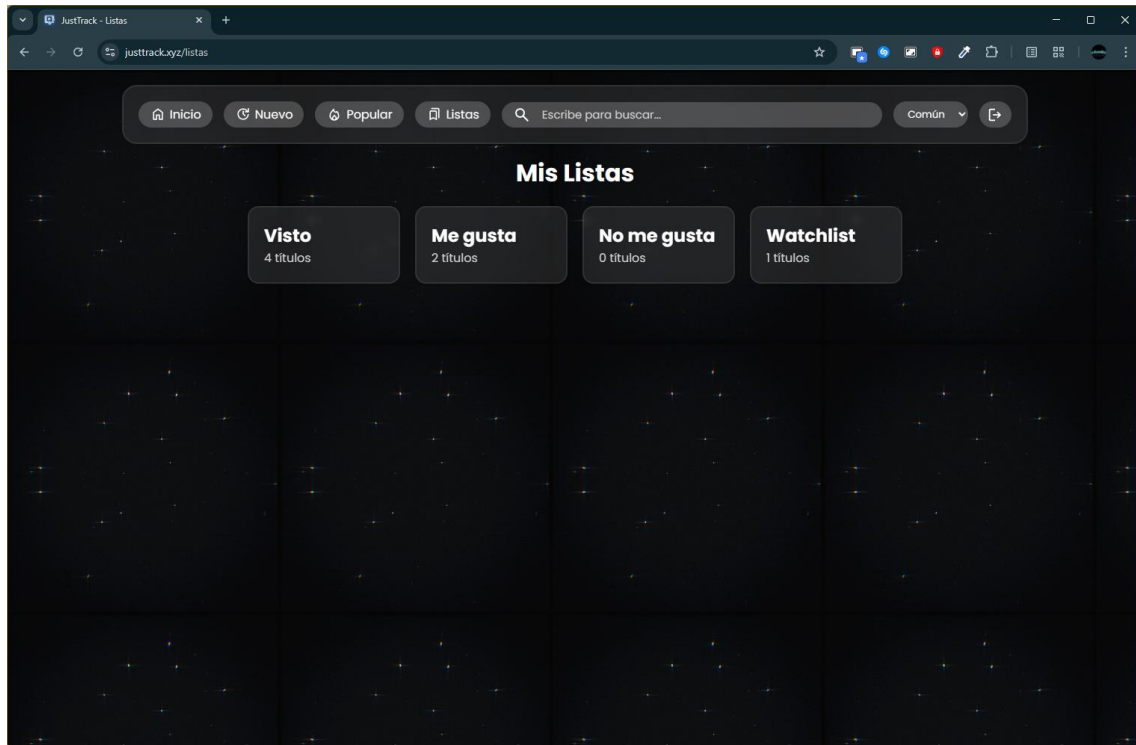
Detalle de película

Al hacer click en cualquier película que nos apareciese en alguno de los apartados anteriores (o en cualquier otro) se nos abrirá la vista de del detalle de una película, el cual se ve como lo que se muestra en esta captura de pantalla:



Como podemos ver en esta captura de pantalla, aquí podremos visualizar toda la información disponible acerca de la película que hemos elegido. Podemos ver el título que tiene en España, su año de lanzamiento, su título original, su sinopsis, su carátula, un banner, su restricción de edad, en qué plataformas se encuentra, sus tráilers, el reparto, si director (o directores), su duración, su país o países de producción, sus géneros y su calificación del 0 al 10 (si es que la tiene) según las listas en la que lo hayan guardado los usuarios (esto lo explico en el siguiente punto).

Apartado “Listas”



En este apartado podremos ver las 4 listas básicas que tiene un usuario además de poder ver cuántas películas hay en cada una de las listas.

Ahora vamos a describir cada una de las 4 listas:

- **Lista “Visto”:** Es la lista en la que aparecen todas las películas que has marcado que has visto con el usuario con el que tienes la sesión iniciada en este momento.
- **Lista “Me gusta”:** Es la lista en la que aparecen todas las películas a las que le has dado al botón de “Me gusta” con el usuario con el que tienes la sesión iniciada actualmente.
- **Lista “No me gusta”:** Es la lista en la que aparecen todas las películas a las que le has dado al botón de “No me gusta” con el usuario con el que tienes la sesión iniciada actualmente.
- **Lista “Watchlist”:** Es la lista en la que aparecen todas las películas que has marcado que quieres ver. Pueden ser películas que ya has visto y que simplemente te apetece volver a verla porque ya se te ha olvidado de qué iba o simplemente porque te gustó mucho esa película y la quieres volver a ver.

Para las películas quiero que el usuario le dé una nota a cada película; esta nota debe ser una puntuación del 0 al 10. La forma en que se me ha ocurrido que sería bastante buena forma de puntuar una película es automatizarlo usando las propias listas, de forma que cuando un usuario añade una película a una lista, le estaría dando una puntuación sin saberlo; no todas las listas son válidas para añadir una puntuación; tiene que ser una de las 3 listas que explicaré debajo:

- Si incluyes la película en la lista “Visto”, pero no has incluido esa película en la lista “Me gusta” o “No me gusta”, la película recibe por tu parte una valoración de un 5/10, ya que asumo que si no te ha gustado lo suficiente como para darle a “Me gusta”, es que la puntuación no será muy alta, además, si no el has dado a “No me gusta”, asumo que la película no te ha disgustado tanto como para suspender la película, de forma que el valor que considero ideal para esta situación es darle un 5 de valoración.
- Si incluyes la película en la lista “Me gusta”, la película recibe por tu parte una valoración de un 10/10; sé que te puede haber gustado sin necesariamente darle un 10 de nota, pero estos datos están planteados para hacer media con las valoraciones que reciba la película por parte de otros usuarios que la hayan añadido también a sus respectivas listas.
- Si incluyes la película en la lista “No me gusta”, la película recibe por tu parte una valoración de un 0/10; con esta valoración ocurre lo mismo que con la anterior, sé que una película puede no gustarte sin necesariamente ser un 0/10 para ti, pero estas valoraciones, como comenté anteriormente, están planteadas para hacer un contraste con la puntuación que le hayan dado otros usuarios al añadirla a sus distintas listas también.

Si añades la película a “Me gusta” o “No me gusta”, ésta se encontrará obligatoriamente en la lista “Visto”, puesto que una película no puede gustarte o no gustarte si no la has visto.

Si le diste “Me gusta” o “No me gusta” a una película y posteriormente le quitas esa valoración, la que se quedará es la puntuación de “Visto”, es decir, un 5/10.

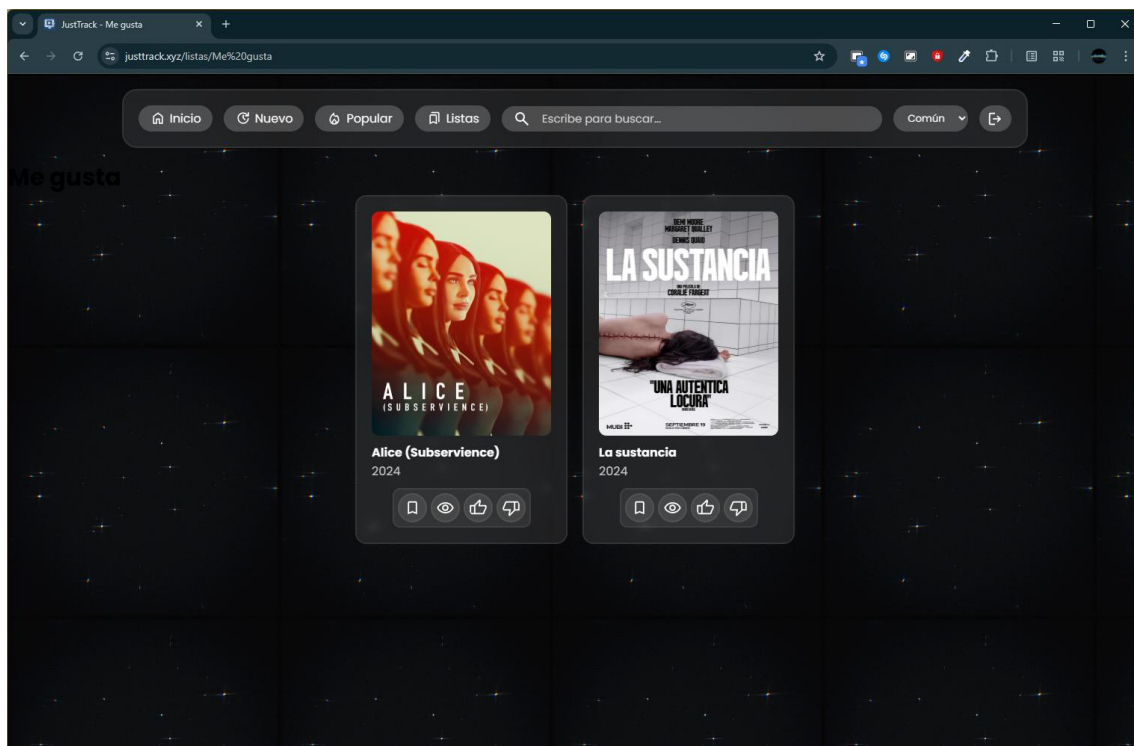
Como ejemplo pongamos que 121 usuarios han guardado una película en su lista “Visto” y además la han añadido a su lista “Me gusta”. Pongamos también que otros 53 usuarios han añadido la misma película a su lista “Visto” y además a “No me gusta”. Finalmente digamos que en total hay 233 usuarios que han añadido la película a su lista “Visto”. Esto

nos dejaría una valoración de un 6,45/10 por parte de los usuarios. La fórmula empleada para calcular esto sería la siguiente:

$$((121*10)+(53*0)+((233-(121+53))*5))/233 = 6,45$$

1. Los 121 usuarios que le dieron a “Me gusta” a la película cuentan como que le han dado una calificación de 10/10 a la película.
2. Los 53 usuarios que le han dado a “No me gusta” a la película cuentan como que le han dado una calificación de 0/10 a la película.
3. Los usuarios que le dieron al botón “Visto” de esa película, pero que no le han dado “Me gusta” o “No me gusta” (like/dislike) cuentan como que le han dado una valoración de 5/10 a la película.
4. Con esa fórmula obtenemos la valoración media de los usuarios a la película.

Volviendo a la parte de la interfaz gráfica, cuando hacemos click sobre alguna de las listas que he mostrado en la captura de pantalla anterior, se nos desplegará una vista en la que podremos ver las películas que se encuentran en esa lista. Por ejemplo, al hacer click sobre la lista “Me gusta” de uno de los usuarios que he creado de ejemplo, veríamos lo siguiente:



Aparecen únicamente las películas a las que ese usuario le ha dado me gusta.

Barra de búsqueda

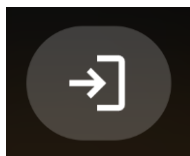
La barra de búsqueda es una barra de búsqueda como cualquier otra; simplemente tendríamos que escribir el título original o el título que recibe en España una película para ver un listado de películas que tienen ese título o esas palabras que escribamos en el buscador dentro de su título, de forma que si recuerdas de forma vaga un título, puedes introducir alguna palabra que sabes que aparece en ese título para encontrarla. Por ejemplo:

- Hay una película que en Español se llama “Until Dawn: Noche de terror”, si en el buscador introducimos sólo la palabra “Until Dawn”, encontraremos esta película, ya que su título original es únicamente “Until Dawn”.
- Hay una película que en Español se llama “El Amateur: Operación Venganza”, pero si introducimos “Amateur” en el buscador, encontraremos esta película, ya que la palabra “Amateur” aparece en el título.

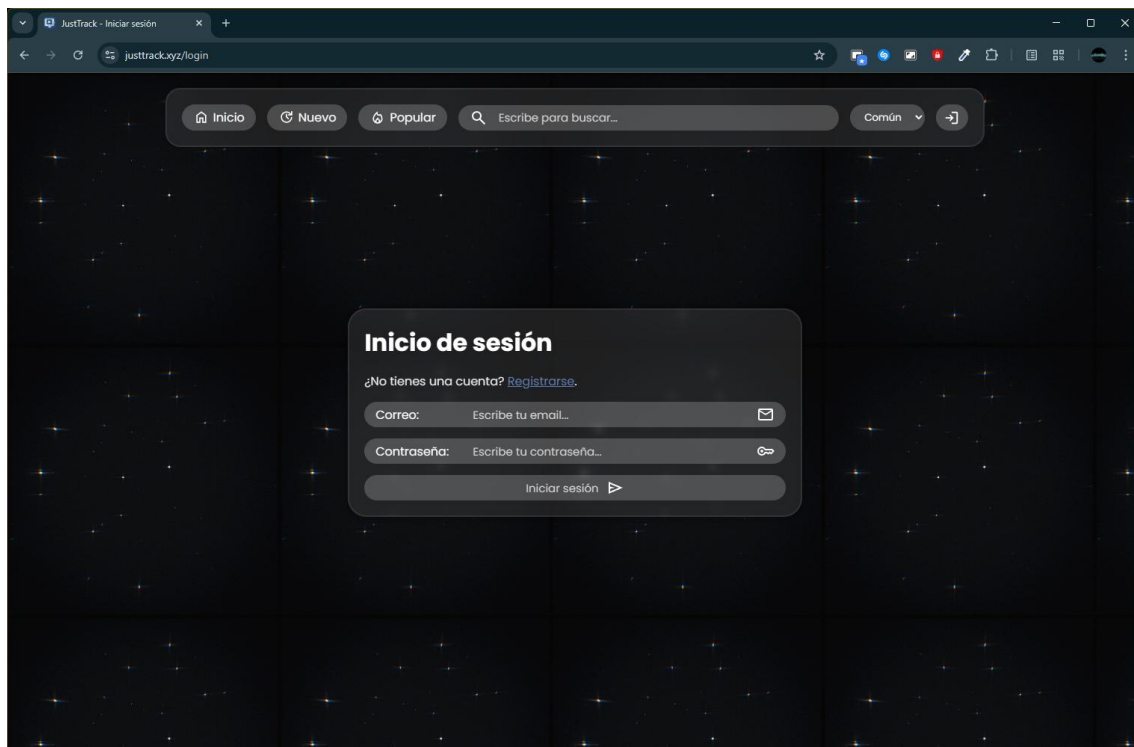
Esto funcionaría sin que tengan que coincidir las mayúsculas y minúsculas, es decir, no es “case sensitive”, de forma que podemos escribir con mayúsculas o minúsculas sin problemas.

Inicio de sesión de usuario

Como podíamos ver en cualquier captura de pantalla, en la cabecera tenemos a la derecha del todo un botón como el siguiente:



Este botón, con este símbolo, como podemos ver muestra una flecha que apunta hacia la derecha apuntando hacia una especie de puerta recortada; este símbolo se emplea universalmente (se emplea en muchísimas páginas web) para indicar que pulsando en ese botón se puede hacer log in, es decir, que puedes iniciar sesión.

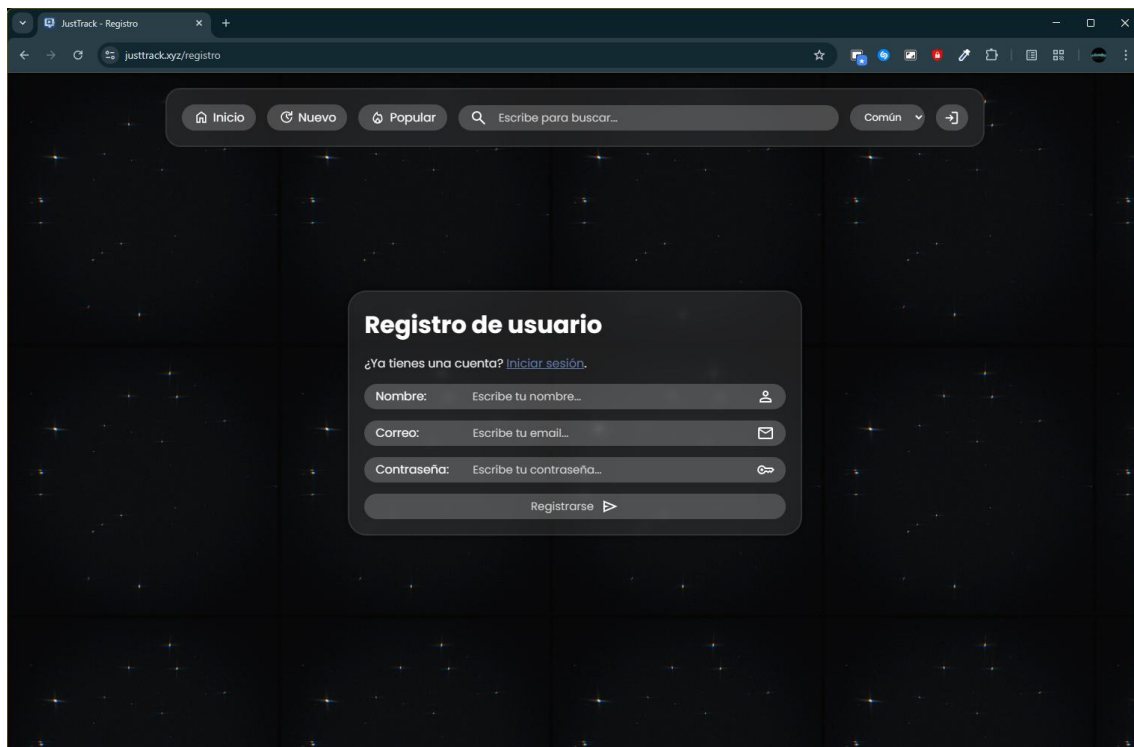


Esto que vemos en la captura de pantalla que está justo encima de este texto es lo que veremos al pulsar en el botón del header que mostraba anteriormente.

Lo primero que vemos es que si no tenemos aún creada ninguna cuenta en la página, simplemente tendremos que pulsar sobre el enlace que se muestra en el texto que se ubica justo debajo del título que pone “Registrarse”. Como podemos ver, tendremos que poner nuestro correo electrónico y la contraseña que hayamos puesto al crear nuestra cuenta en JustTrack para identificarnos y acceder a nuestro usuario.

Registro de usuario

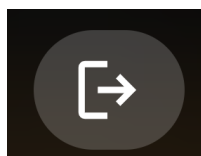
Como comentaba en el punto anterior, tendremos que pulsar sobre el botón “Registrarse” que sale en la pantalla de inicio de sesión para acceder a este panel de registro de usuario. Tras pulsar este botón veremos la siguiente pantalla que muestro en esta captura de pantalla:



Para registrarnos, como podemos ver en la captura de pantalla, se nos solicita nuestro nombre, correo electrónico y una contraseña. Tras introducir los datos con los que nos identificaremos, pulsamos sobre el botón “Registrarse” que está justo debajo de todos los campos para crear nuestro usuario; si todo ha salido bien seremos redirigidos a la pestaña de inicio de sesión para iniciar sesión con el usuario que acabamos de crear, sino, nos quedaremos en esta misma página.

Cerrar sesión

Para cerrar sesión lo primero que tendremos que cumplir es haber iniciado sesión con un usuario válido. Una vez que iniciamos sesión, veremos que el icono que antes se mostraba para iniciar sesión ahora será distinto; ahora veremos el siguiente símbolo:



Este símbolo, al igual que el anterior era un símbolo universal para log in (iniciar sesión), este es un símbolo universal que indica que con ese botón puedes hacer log out (cerrar sesión). Este botón sólo aparecerá si hemos iniciado sesión como comentaba antes, y aparecerá como remplazo del botón de iniciar sesión. Si queremos cerrar sesión

simplemente tendremos que hacer click sobre este botón y nos redirigirá a la página de inicio con la sesión cerrada.