

Proyecto

Desarrollo de Aplicaciones Web

Entrega 3 - Parte 1

Aspectos de interés. Documentación.
Implementación. Problemas y soluciones.
Mejoras de futuro.
Guía de configuración e instalación.
Despliegue en local y en línea.

ÍNDICE

<u>Aspectos de interés</u>	<u>3</u>
<u>Documentación</u>	<u>3</u>
<u>Implementación</u>	<u>4</u>
<u>Problemas y soluciones</u>	<u>4</u>
<u>Mejoras de futuro</u>	<u>5</u>
<u>Despliegue local</u>	<u>5</u>
<u>Despliegue en línea</u>	<u>7</u>
<u>Documentación, aclaraciones y anotaciones sobre carpetas y archivos relevantes</u>	<u>8</u>

Aspectos de interés durante el desarrollo:

A lo largo del desarrollo de mi aplicación he aprendido mucho y considero que he incorporado herramientas que antes no tenía.

He renombrado el proyecto como Overdubs, ya que el nombre anterior, como indiqué en la memoria correspondiente, era provisional. Overdubs es la técnica de grabación en la que se superponen pistas que recrean las mismas notas, para conseguir un efecto de mayor amplitud. Vi una relación simbólica entre esta técnica y el hecho de que los usuarios de la aplicación escriban cada uno sobre su punto de vista sobre un mismo disco, de manera que las opiniones y los puntos de vista se comparten y superponen.

He realizado cambios estéticos en el proyecto, de manera que predominan los colores oscuros y sobre ellos el texto claro. Después de muchas pruebas he considerado que facilita una lectura casual y relajada, produciendo menos cansancio a la vista.

El funcionamiento de la aplicación es el esperado, y se adapta al descrito en las memorias previas sin problemas. De hecho creo que potencialmente se amplía, ya que conforme iba avanzando he ido descubriendo las posibilidades del proyecto hacia la interacción entre usuarios, de manera que he añadido algunas pequeñas funcionalidades, como son la posibilidad de seguirse entre ellos y enviar y recibir mensajes. De esta manera pueden compartir gustos, hablar sobre ellos, hacerse recomendaciones que podrán nutrir la base de datos, y todo ello sin la parte de exposición que pueden tener las redes sociales.

Uno de los aspectos fundamentales para mí es la posibilidad de que, en el futuro, se puedan analizar los datos almacenados para conocer tendencias y públicos objetivos. Creo que esto haría que el valor del proyecto incrementara.

Finalmente, quisiera destacar el aspecto colaborativo y colectivista de Overdubs, ya que todo el mundo añade información al proyecto, de manera que con el paso del tiempo y la actividad, pueda tener una magnitud mayor.

Documentación:

La documentación se ha escrito y generado valiéndome de la herramienta Phpdocumentor. Dicha documentación se encuentra en la carpeta `.phpdoc` dentro del proyecto.

En la ruta `.phpdoc/build/index.html` se encuentra el archivo principal para acceder a la documentación.

Implementación:

Para desarrollar el proyecto y llevar a cabo su implementación, lo primero fue tener claras las herramientas a utilizar. Opté por utilizar Laravel como framework, ya que lo utilicé durante el curso pasado en algunas tareas. Si bien no profundicé mucho en su momento, he visto en la realización de este proyecto una buena oportunidad para ello, ya que me pareció muy robusto.

El proyecto se basa en la arquitectura MVC, así que lo siguiente fue definir los modelos, las vistas y los controladores. No obstante, no los integré todos de golpe, sino que los fui incorporando conforme avanzaba paulatinamente en el proyecto e iba necesitando dichos elementos.

La mayor parte del proyecto se basa en PHP como lenguaje principal, pero también se apoya en Javascript. El uso de HTML y CSS también está por todas partes, como es habitual. Laravel favorece el uso de Bootstrap, pero en mi caso decidí pasarlo por alto y centrarme en las herramientas que he utilizado al menos una vez. Por supuesto también se ha utilizado Javascript y AJAX en el front end.

Para la base de datos he optado por usar MySQL como sistema de gestión, ya que es el que hemos trabajado durante estos dos años en el ciclo, y es un estándar. Siendo más exacto, he utilizado MariaDB, ya que es el que utilizo en mi configuración local. Se trata de un sistema derivado, pero se adapta a un despliegue con MySQL sin necesidad de cambios de ningún tipo.

Para la búsqueda de discos me he valido de la API de Discogs, una conocida plataforma en el mundo del coleccionismo de CDs y vinilos. Está muy especializada, es muy completa y detallada y su uso es relativamente sencillo. En el futuro me gustaría ampliar este aspecto para que Overdubs beba de más APIs, como last.fm, Spotify, o Bandcamp.

Problemas y soluciones:

Uno de los primeros problemas que encontré fue la gestión de registro de usuarios y login, así como la encriptación de las contraseñas y demás cuestiones relacionadas. Afortunadamente, Laravel provee todas las herramientas necesarias para ello.

Encontrar la API adecuada con la que poder «nutrir» poco a poco, y de manera colectiva, mi base de datos fue otro de los aspectos más conflictivos, ya que tuve que documentarme sobre varias y practicar con ellas, con el consumo de tiempo que conlleva. Algunas, como la de Spotify, me dieron bastantes problemas y no me parecieron intuitivas. A raíz de seguir explorando di con la anteriormente mencionada, Discogs, y tras semanas utilizándola estoy más que satisfecho con el resultado.

Algunas queries más complejas a la base de datos me llevaron algún tiempo de dominar, sobre todo teniendo en cuenta que Laravel tiene su sintaxis propia para ello, pero la solución fue practicar hasta poco a poco llegar a dominarlas con soltura.

El layout de la página en ocasiones ha sido complejo de definir, en parte por falta de planificación en ese aspecto. Además de ello, CSS en ocasiones se comportaba de manera un poco tosca y se producían algunos conflictos, pero nuevamente los achaco a la improvisación y a usar demasiados nombres para clases e identificadores de elementos. La única manera de salir de esa situación fue ir haciendo poco a poco varios barridos a lo largo de todo el código del front end hasta minimizar el uso de clases y así reducir el número de líneas de código CSS. Puede parecer una herramienta muy simple, pero la posibilidad de reemplazar determinado texto por otro se convierte en una ayuda enorme en estas situaciones.

El despliegue online también ha sido algo más dificultoso de lo que esperaba, pero he aprendido mucho y ha sido fructífero.

Funciones sin implementar y mejoras de futuro:

De cara al futuro me gustaría ahondar en la manera en que a un usuario se le pueden hacer recomendaciones en base a sus gustos, esto es, en función de los discos que más ha puntuado o que ha guardado en sus listas de favoritos. De esta manera, se le podrían sugerir álbumes, listas de otras personas e incluso usuarios con los que compartir gustos. También sería muy interesante hacerle un resumen privado a cada usuario sobre sus gustos, al estilo del famoso Wrapped de Spotify. En la misma línea se podría ahondar en el análisis de datos y por tanto en el informe de los mismos que se muestra a los administradores.

Otra de las implementaciones que me gustaría llevar a cabo, ya dejada entrever previamente, es la utilización de varias APIs simultáneamente, de manera que no se dependa solo de una para recabar información.

Actualmente me encuentro realizando un curso de React. Es más que evidente que se trata de un framework muy potente y creo que podría darle al frontend un plus muy interesante de cara a la experiencia del usuario. No obstante, aún no tengo experiencia como para aventurarme a usarlo en el proyecto actualmente, pero me encantaría poder aplicar los conocimientos adquiridos en una futura versión de Overdubs.

Despliegue local. Guía de instalación y configuración:

Para poder poner en marcha la aplicación en otro equipo en local son necesarios varios pasos previos.

1. Instalación de PHP en su versión más reciente.
2. Instalar Composer para la gestión de las dependencias.
3. Instalar un servidor web o entorno local como Apache. Este paso no es imprescindible, ya que podremos usar el servidor de desarrollo de Laravel, pero sí es necesario dado el enunciado de la tarea.
4. Instalar MySQL y crear una base de datos con el nombre **overdubs** y otra con nombre **overdubs_test**.

Si el usuario tiene ya instalado un paquete como XAMPP solo será necesario instalar Composer, ya que ya dispondrá de PHP, MySQL/MariaDB y PHP.

5. El siguiente paso será copiar la carpeta del proyecto en la ubicación correspondiente. En mi caso, utilizando XAMPP, lo he hecho en la carpeta /htdocs.

6. A continuación, estando ubicados en la carpeta del proyecto, procederemos a instalar las dependencias de PHP con Composer mediante el comando:

composer install

Este proceso puede tardar unos instantes.

7. Llegados a este punto es conveniente repasar el archivo .env y asegurar que las variables aquí indicadas tienen los valores correctos:

```
APP_NAME=overdubs
APP_ENV=local
APP_DEBUG=true
APP_TIMEZONE=UTC
APP_URL=http://localhost

DB_CONNECTION=mariadb
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=overdubs

DB_USERNAME=root
DB_PASSWORD=
```

8. Generamos una clave de aplicación con el comando:

php artisan key:generate

Y copiamos el valor generado en la variable APP_KEY del archivo .env de manera que quede así:

APP_KEY=[el_valor_generado]

9. Para el informe de datos, que en este caso se consulta en la propia aplicación, he incluido algunas gráficas valiéndome de la librería Chart.js. Así que ejecutaremos el siguiente comando:

npm install

10. Para crear las tablas y poblar la base de datos **overdubs** habrá que ejecutar, sobre esta, las queries encontradas en:

/resources/overdubs_tables en primer lugar para crear las tablas y las claves.

/resources/overdubs_seed para poblar las tablas.

No obstante, todo ello puede hacerse de manera más cómoda simplemente ejecutando el comando:

php artisan migrate:fresh --seed

11. Por último levantamos el servidor. Como se indicó en el paso 3, podemos hacerlo con el servidor de desarrollo de Laravel a través del comando:

php artisan serve

Y accediendo en el navegador a <http://127.0.0.1:8000> o alternativamente a <http://localhost:8000>.

Credenciales de usuario y contraseñas:

Como usuario estándar, se puede acceder mediante:

Email: user_professor@gmail.com
Password: overdubs2025

Como perfil de administrador he registrado la siguiente cuenta:

Email: admin_professor@gmail.com
Password: overdubs2025

Desde este perfil de administrador puede accederse al BackOffice de la web, así como al informe de datos en la sección *Data report*.

Por último está el perfil de artista o banda. Como la idea es que se registren artistas reales y que los discos a buscar existan realmente, he creado un perfil de una banda real, Toundra, como si estuvieran registrados y verificados por los administradores:

Email: toundra@gmail.com
Password: overdubs2025

Despliegue en línea:

Estas credenciales sirven tanto para el despliegue local como para el despliegue en línea. El proyecto se encuentra subido a Koyeb a través de la siguiente URL:

<https://living-dee-felmunpenprojects-5e535f4c.koyeb.app/>

El nombre poco intuitivo se debe a que en la modalidad gratuita es imposible de editar. Es posible que en el primer intento tarde un poco y haya que recargar la página, ya que Koyeb pone el proyecto en estado *Sleeping* si no tiene actividad reciente. Aunque puede tardar un poco en ponerlo en marcha, funciona exactamente igual que el despliegue en local.

Los email y passwords para acceder a la aplicación en línea son los mismos que para el despliegue local. Para más información consultar la página anterior.

Si es requerida algún tipo de ayuda en línea, un usuario registrado podrá enviar reportes o informes con incidencias ocurridas o sugerencias que tenga haciendo clic en el enlace del footer de la web, una vez habiendo iniciado sesión.

Documentación, aclaraciones y anotaciones sobre carpetas y archivos relevantes:

/app/Http/Controllers contiene todos los archivos .php con los controladores del proyecto.

/app/Models contiene los archivos .php con los modelos, en caso de que el desarrollador opte por usar Eloquent para manipular la base de datos.

/config/database.php contiene la configuración para la conexión con la base de datos.

/database/migrations alberga numerosos archivos .php con las tablas, claves foráneas y migraciones necesarias.

/database/seeders/DatabaseSeeder.php llama al archivo .sql que introduce las filas predefinidas en las tablas del proyecto tras una migración.

/public/css/style.css contiene toda la información relativa a la hoja de estilos del proyecto.

/resources/overdubs_seed.sql y **/resources/overdubs_seed.sql** contienen los archivos para la inserción de datos y creación de tablas en formato MySQL, respectivamente.

/resources/views es una de las carpetas más pobladas del proyecto. Dentro se encuentran, divididos en carpetas, todos los archivos que gestiona Blade para las distintas plantillas y vistas.

/routes/auth.php y **/routes/web.php** son los archivos .php que gestionan el enrutamiento correspondiente a la autenticación (registro, login, etc) y navegación de todas las páginas.

/tests/ contiene las carpetas con los archivos .php con las distintas pruebas llevadas a cabo.