

# **CFGS**

# **DESARROLLO DE**

# **APLICACIONES WEB**

# **ONLINE**

**Proyecto Final de Ciclo**

**Apellidos y nombre del autor/a: Guillem Sales, Rubén**

**Fecha de entrega: 20-05-2025 Entrega final**

DevPlace

2024-2025

**ÍNDICE**

1. **Portada**
2. **Índice**
3. **Resumen del proyecto, que se propone y de que está formado**
4. **Justificación y objetivos**
5. **Desarrollo del proyecto**
   1. **Análisis del mercado**
   2. **Metodologías**
   3. **Componentes y tecnologías usadas en el proyecto**
   4. **Problemas o dificultades en el desarrollo del TFG**
6. **Conclusiones tras la realización del TFG (sin realizar)**
7. **Líneas futuras de trabajo (sin realizar)**

**3.Resumen del proyecto.**

**ESPAÑOL**

DevPlace es un proyecto que consistirá en una aplicación web la cual permitirá a los usuarios que se registren poder compartir sus scripts y recursos con otros desarrolladores, es el lugar perfecto para compartir conocimiento, los desarrolladores junior podrán aprender de los que tienen más experiencia, en DevPlace el objetivo es ayudarnos entre todos a ser mejores desarrolladores y fomentar así una comunidad colaborativa.

Se podría decir que es muy parecido a aplicaciones como GitHub, por ejemplo. Donde cada persona puede subir su código en un repositorio ya sea público o privado, y el resto de los usuarios usar ese código según le convenga. Pero con una gran ventaja, DevPlace es mucho más intuitivo. ¿A quién no le ha resultado difícil en sus inicios encontrar información en GitHub? Con el diseño minimalista de DevPlace conseguimos que no de tanto ‘’miedo’’ aprender a programar cuando eres principiante, encontrando la información que necesitas con mucha más facilidad.

Básicamente, en la versión actual de DevPlace, podrás:

-Registrarte e iniciar sesión con tu cuenta ingresando un email y una contraseña. Es obvio, pero al igual que te logueas, cuando quieras salir de la app también podrás cerrar sesión o desloguearte

-Una vez hayas iniciado sesión, tendrás acceso a todos los scripts que hayan subido los demás usuarios, podrás simplemente visualizarlos o si lo crees conveniente, incluso descargarlos.

-Podrás subir tus propios scripts para compartirlos con todo el mundo

La aplicación combina tecnologías como Angular que nos permite hacer toda nuestra parte front, manejar el token Bearer.. Con Symfony que nos permite elaborar toda la parte del back que incluye todo el CRUD de scripts, subida y descarga de ficheros…

Todo esto usando una BBDD MySql para qu ellos usuarios puedan guardar y nutrirse de los datos que hay dentro de la app.

**INGLÉS**

DevPlace is a project that will consist of a web application that will allow registered users to share their scripts and resources with other developers. It's the perfect place to share knowledge. Junior developers can learn from those with more experience. The goal of DevPlace is to help each other become better developers and thus foster a collaborative community.

You could say it's very similar to applications like GitHub, for example. Where each person can upload their code to a repository, whether public or private, and the rest of the users can use that code as they see fit. But with one major advantage, DevPlace is much more intuitive. Who hasn't found it difficult to find information on GitHub in the beginning? With DevPlace's minimalist design, we've made learning to program less "scary" for beginners, making it much easier to find the information you need.

Basically, in the current version of DevPlace, you can:

-Register and log in to your account by entering an email address and password. It's obvious, but just like logging in, you can also log out or log out when you want to exit the app.

- Once you're logged in, you'll have access to all the scripts uploaded by other users. You can simply view them or, if you see fit, even download them.

- You can upload your own scripts to share with everyone.

The application combines technologies like Angular, which allows us to manage our entire front-end, manage the Bearer token, and Symfony, which allows us to develop the entire back-end, including all CRUD scripts, file uploads, and downloads.

All of this using a MySQL database so that users can save and access the data within the app.

**4. Justificación y objetivos del proyecto**

DevPlace es una aplicación que tiene como objetivo formar una comunidad de desarrolladores la cual sea accesible para todo el mundo. Hoy en día, en el mundo del desarrollo de software, compartir conocimiento se ha vuelto en una práctica fundamental para el aprendizaje. Sin embargo, las principales plataformas actuales tienen un diseño complejo o poco intuitivo, lo cual puede suponer una barrera para los principiantes. Esto genera dos problemas principales:  
 -Curva de aprendizaje elevada: los desarrolladores junior no están habituados y les puede resultar de gran dificultad buscar y asimilar la información.

-Perdida de tiempo: interfaces complejas, hacen que principiantes pierdan mucho tiempo buscando la información, tiempo que podrían aprovecharlo aprendiendo si se les facilita esa búsqueda de información.

La aplicación nace con el objetivo de solucionar dicha necesidad, hacer el aprendizaje más fácil para los principiantes, consiguiendo que tanto compartir como buscar scripts y recursos de programación se haga de una forma más sencilla y clara, ahorrando mucho tiempo en encontrar lo que necesitas. Yo, personalmente al principio de interesarme en este mundo, exceptuando videos, me era muy difícil encontrar información ‘entendible’ las plataformas que encontraba me resultaban muy difíciles de entender, navegar por dentro de ellas, etc.

Por eso, esta app va dirigida sobre todo a gente que acaba de empezar, para ayudarles a poder encontrar información sin perder mucho tiempo buscando en interfaces ‘para expertos’.

Objetivos específicos:  
 1-Usabilidad e interfaz:

* 1. Diseñar una interfaz de usuario con rutas claramente identificadas y con una navegación sencilla para el usuario.
  2. Conseguir que las funciones básicas de la app se hagan en los menores pasos posibles, para no complicar así a los usuarios principiantes.

2-Funcionalidad básica:

2.1- Permitir el crud completo de scripts, incluyendo campos como título, descripción, precio, fecha(aunque la ‘escondemos’ nos la quedamos para nosotros) y sobre todo el fichero/archivo asociado.

2.2- Facilitar la descarga y edición de scripts, solo el autor puede modificar eliminar sus scripts.

3-Seguridad y autenticación:

3.1-Proteger las rutas mediante JWT y validación de token Bearer.

3.2-Asegurar la integridad de las peticiones de subida y descarga de ficheros con formData y comprobaciones en nuestro back.

4-Futura escalabilidad:

4.1- Filtrado: Busquedas avanzadas por palabra clave y metadatos,

4.2-Sistema de etiquetas y categorías de scripts.

4.3-Valoraciones y comentarios: Implementaremos un sistema de ‘me gusta’ y comentarios, para que los usuarios puedan agradecer a otros.

4.4-Integracion de la pasarela de pago: añadiremos un sistema de pago para que los ‘scripts premium’ tengan recompensa para el usuario que los sube y que el usuario que quiera beneficiarse de ellos lo ‘valore’ más.

4.5-Un punto muy importante es el tiempo de carga, actualmente tenemos un problema de comunicación entre back y front que hace que cualquier llamada al back tarde mas de lo deseado, solucionaremos esto lo más pronto posible.

Con estos objetivos, queremos no solo satisfacer las necesidades esenciales en un mercado de scripts, sino también sentar las bases de lo que el día de mañana será unn ecosistema colaborativo y evolutivo, centrándonos siempre en que los más principiantes no ‘tengan miedo’ pero sin olvidarnos de los más expertos que son los que realmente nutren nuestra app.

**5. Desarrollo del proyecto**

**5.1 Análisis del mercado**

En el ecosistema actual existen varias plataformas que, aunque no compiten de forma directa con DevPlace, comparten funcionalidades relacionadas con el intercambio de código y la colaboración:

-GitHub, GitLab y Bitbucket, que ofrecen funcionalidades para gestionar los repositorios de tus proyectos, con un control de versiones (git) y permiten también la colaboración del equipo de desarrollo, estas plataformas están más orientadas a desarrolladores y equipos de desarrollo avanzados.

-Otras plataformas como, por ejemplo, StackOverflow podrían ser más parecidas a DevPlace, StackOverflow es una plataforma enfocada más concretamente en compartir conocimiento o soluciones a problemas concretos a través de preguntas y respuestas entre los usuarios.

-Codester por ejemplo, es un mercado de recursos de diseño web, la cual sería más parecida aun a lo que ofrece DevPlace.

DevPlace ofrece una alternativa a estas plataformas, más intuitiva y amigable centrada en la comunidad y sobre todo en ayudar a los principiantes. Con un diseño minimalista e intuitivo conseguimos que los desarrolladores que están empezando puedan aprender con la menor dificultad añadida.

Modelo de negocio de devplace:  
 1-Comision por venta: Nuestra empresa se llevara un porcentaje de cada transacción.

2-Planes premium: suscripción para autores con funcionalidades extra (destacar sus scripts, analíticas de descargas y visitas, etc.)

3-Publicidad: espacio para que otras plataformas puedan publicitarse en nuestra web

**5.2 Metodología**

En el proyecto se sigue la metodología de trabajo Scrum, donde cada 2 semanas aproximadamente se realizan Sprints, donde se analiza y organiza el trabajo realizado y a realizar. Estos sprints son reuniones con el tutor realizadas con Microsoft Teams.

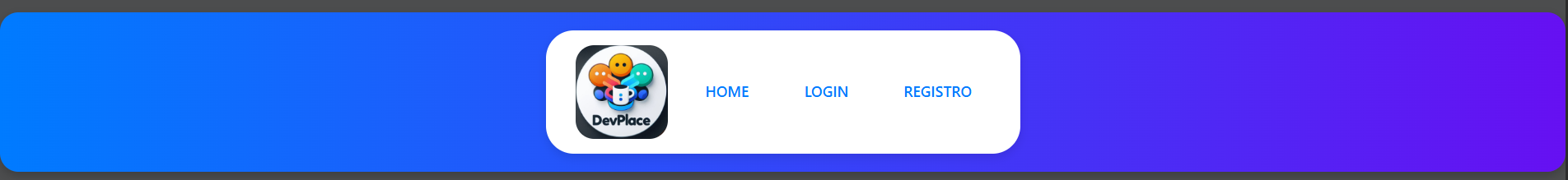
Sprint1.XXDe marzode2025  
Sprint 2. 16 de abril de 2025

**5.3 Componentes de la aplicación**

DevPlace se estructurará en diferentes paginas las cuales cada una aportará funcionalidades diferentes a la aplicación:

**Home:**

Dispondrá de un menú de navegación siempre visible (el mismo para todas las páginas) en el cual podremos navegar a las diferentes páginas de la aplicación (Home, scripts, registro, logueo),



Como contenido, tendrá una descripción de lo que ofrece DevPlace con varios botones que redirigirán a la página de registro para que el cliente no pierda el tiempo y empiece a aprender cuanto antes. Si se trata de un usuario ya logueado el que visita la página de home, será prácticamente igual cambiando los botones que estos llevaran a la pagina Scripts y a la de Mis scripts.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

La home para usuarios logueados y sus pequeños cambios (incluyo el menú de navegación)

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto. :  
Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Scripts:**  
Aquí aparecerán todos los scripts subidos por los usuarios, donde los mismos podrán descargarlos si lo creen necesario

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Mis scripts:**

Pagina donde cada usuario podrá ver los scripts que ha subido el a lo largo del tiempo, así como editarlos, borrarlos o descargarlos.

En el momento de la carga aparece esto:  
Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Una vez cargados los scripts, tenemos la misma interfaz que en la pagina de scripts, pero añadiéndole las funcionalidades de editar y borrar el script. Esto solo puedo hacerlo el ‘dueño’ del script

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Los scripts de pago salen con un botón de comprar y en un color diferente, mas destacado:  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Subir script:**

Formulario sencillo donde los usuarios preparan su publicación, añadirán un título una descripción, un precio y lo mas importante, u archivo de cualquier tipo o una carpeta comprimida

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Editar script:**

en esta pagina que se accede desde mis scripts-> script en concretp -> botón editar, los usuarios pueden modificar lo que quieran de sus scripts, como si quieren cambiar el archivo por si hay algún problema:  
Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Logueo:**  
Interfaz de logueo sencilla

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Registro:**

Pagina donde los usuarios podrán registrarse como usuarios para tener acceso completo a la aplicación

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

**Arquitectura general**

Cliente–Servidor desacoplado:

Frontend: Single-Page Application en Angular.

Backend: API RESTful en Symfony.

Comunicación: peticiones HTTP con cabecera Authorization: Bearer token.

Despliegue: contenedores Docker orquestados con Docker Compose.

* **Frontend (Angular)**

Standalone Components para cada página y componente reutilizable.

Rutas protegidas mediante AuthGuard que valida existencia y caducidad del JWT.

Servicios:

AuthService: gestión de login/logout y almacenamiento de token.

ScriptService: llamadas CRUD (GET /scripts, POST /scripts, etc.), usando HttpClient y FormData para subir ficheros.

UI: Bootstrap para diseño responsivo, feedback visual en formularios y listas.

* **Backend (Symfony API)**

Bundles:

lexik/jwt-authentication-bundle para emisión y validación de tokens.

doctrine/orm para mapeo de entidades User y Script.

Controladores:

AuthController: registro (POST /api/register), login (POST /api/login).

ScriptController: rutas CRUD, aplicando restricciones de propietario con Voter.

Upload: recepción de FormData, validación de tamaño y tipo de archivo, guardado en carpeta public/uploads.

* **Base de datos y despliegue**

MySQL: esquema con tablas user (id, email, password, roles) y script (id, title, description, price, date, filename, owner\_id).

Migraciones gestionadas con doctrine:migrations.

Docker Compose: servicios para Angular, PHP-FPM, Nginx (reverso proxy) y MySQL, con volúmenes para persistencia y variables de entorno.

**5.4 Problemas/dificultades al desarrollar la aplicación**

-A día 20 de abril, me ha resultado un problema el unir el Frontend con angular y el backend con php ya que nunca lo había hecho, siempre había desarrollado la aplicación entera en frontend o backend solo. Otro problema menor ha sido el control de versiones, había empezado con una estructura de carpetas del proyecto TFG> Devplace, en devplace (Solo contenía el front) había iniciado el repositorio de github y luego me di cuenta que tenía que iniciar el repositorio en la carpeta raíz, entonces me toco perder el control de versiones hasta el momento, iniciar el repositorio en la carpeta TFG con la siguiente estructura: TFG como raíz, con dos hijos al mismo nivel: DevPlace (donde desarrollo el front) y DevPlace-Back (donde desarrollo el back) creándome dos ramas ‘ramafront’ y ‘ramaback’ donde ir subiendo los cambios.

-A 29 de abril sigo mas o menos atascado en lo mismo, buscando la manera de unir correctamente back y front (siempre me falla una cosa u otra al final y no lo consigo) En estos puentes de pascua he adelantado antes de subsanar ese problema, la parte de back que la tenía muy escasa, están creadas las entidades etc. a falta de hacer que tengan funcionalidad de verdad y se muestren los datos del back en el front.

-Tuve un problema que tarde casi una semana en resolverlo y es que no me dejaba subir scripts, pensaba que era un problema con el tamaño de los archivos que subía y investigando pensé que la solución era añadir un archivo php.ini junto a una línea en el dokerfile, hice Docker-compose down, build y up y dejo de comunicarse el front con el back, borré el archivo y la línea de dockerfile, volví a reconstruir el dockerfile y seguía sin irme, me frustró bastante

-A pocos días ates de la entrega, he tenido muchos problemas con git, desde el principio del proyecto me ‘molestó’ creé por error la siguiente estructura.

TFG->

Devplace-> todo el proyecto front

Devplace-back-> Devplace-back->todo el proyecto back

Cuando el directorio subrayado no hace falta para nada, pero por si acaso lo he dejado ahí.

Otra cosa que me ha generado problemas es mi ordenador y su sincronización con onedrive y la falta de almacenamiento que tengo, de repente hay archivos que se me eliminan del ordenador.

Los merge, fetch, etc. Se me dan muy mal, creo que por fin he conseguido subir todo el proyecto final correctamente en la rama master, pero me ha generado mucha confusión las ramas que había creado y como juntarlas o ir subiendo commits en paralelo entre ellas.

**5.5 Resultados obtenidos**

Tras la implementación de los componentes descritos y la ejecución de las pruebas oportunas, se han alcanzado los siguientes hitos:

**Resultados funcionales**

* **Autenticación y autorización**
  + Registro y login de usuarios con JWT, con almacenamiento del token en localStorage y protección de rutas en Angular (AuthGuard) y Symfony (Voters).
* **Gestión de scripts**
  + **CRUD completo**: creación, visualización, edición y eliminación de scripts, incluyendo metadatos (título, descripción, precio, fecha) y fichero adjunto.
  + **Subida y descarga de archivos**: uso de FormData en el frontend y validaciones de tamaño/tipo en el backend.
* **Listados**
  + **Catálogo general**: paginación básica y ordenación por fecha.
  + **“Mis scripts”**: listado filtrado por autor ligado al usuario autenticado.

**Resultados técnicos**

* **Contenerización con Docker Compose**
  + Entorno reproducible con servicios para Angular, PHP-FPM, Nginx y MySQL, facilitando la puesta en marcha local y futura orquestación en produccion.
* **Arquitectura modular**
  + Separación clara de responsabilidades: servicios HTTP en Angular, controladores y Voters en Symfony, entidades Doctrine y migraciones de base de datos.
* **Pruebas**
  + Pruebas manuales de flujo end-to-end (registro → subida → descarga → edición → borrado).

**Valoración global:**  
La plataforma funciona de manera estable, cumple con los objetivos de usabilidad, seguridad y funcionalidad definidos al inicio, y ofrece un punto de partida sólido para las futuras mejoras (buscador avanzado, sistema de etiquetas y valoraciones, pasarela de pago). La experiencia de usuario es directa, tal como se había planteado, y la arquitectura permite escalar y mantener el código con facilidad.

Lo que debemos mejorar cuanto antes son los tiempos de carga.

**6. Conclusiones tras la realización del TFG**

La realización del TFG para mí, ha sido un sufrimiento a la vez que una satisfacción al acabarlo, me ha costado muchas horas por culpa, en gran parte, de que me he atascado varias veces y he estado varios días sin avanzar intentando solucionar x problema. He aprendido mucho haciéndolo, he intentado aplicar las cosas que he aprendido en clase junto a las cosas que estoy aprendiendo en las practicas, que aun que en las practicas es todo bastante diferente, creo que me han servido muchísimo para poder hacer este proyecto.

He podido realizar todas las funciones básicas de la app, me he dejado, por ejemplo, el implementar una plataforma de pago que no me ha dado tiempo, pero creo que podía haber estado muy bien haberlo implementado

**7**. **Líneas futuras de trabajo**

* **Buscador avanzado**
* Implementar filtros por palabras clave, rango de precios, fecha de publicación y etiquetas, además de ordenación dinámica.
* Integración de sugerencias automáticas (autocomplete) basadas en los scripts más populares y en el historial de búsqueda del usuario.
* **Sistema de etiquetas y categorías**
* Permitir que los autores asignen etiquetas libres y categorías predefinidas a cada script.
* Pantalla de exploración por etiqueta/categoría, con contadores de uso e índices de popularidad.
* **Valoraciones y comentarios**
* Añadir un módulo de valoraciones (estrellas, “me gusta”) y comentarios en cada script.
* Moderación básica: reportes de contenido inapropiado y sistema de flags para revisión.
* **Perfiles de usuario**
* Página “Mi perfil” donde cada desarrollador pueda:
* Editar sus datos (nombre, email, foto).
* Ver estadísticas de descargas y valoraciones de sus scripts.
* Gestionar suscripciones (planes gratuitos/premium).
* **Pasarela de pago**
* Integración con un proveedor (Stripe, PayPal…) para que los autores puedan monetizar sus scripts.
* Gestión de comisiones y facturación básica en el panel de administrador.
* **Internacionalización (i18n)**
* Localizar la aplicación a varios idiomas (español, inglés y otros), incluyendo soporte para formatos de fecha, moneda y textos.
* **Notificaciones y correo electrónico**
* Sistema de notificaciones in-app y envío de emails para eventos clave: confirmación de registro, subida de script, comentarios recibidos, etc.
* **Optimización y despliegue en producción**
* Configurar CI/CD para pruebas automáticas y despliegue en un entorno de producción (por ejemplo, Kubernetes o un PaaS).

**8.Bibliografia**

Angular. (Angular Documentation. <https://angular.dev/overview> y Diferentes foros

Bootstra. Bootstrap Documentation. <https://getbootstrap.com/docs/4.1/getting-started/introduction/> y Diferentes foros

Docker Docker Documentation https://docs.docker.com/ y Diferentes foros

Git.Diferentes foros

Chatgpt

Ayuda de los compañeros de las practicas

Symfony <https://symfony.com/doc>

Stack overflow <https://stackoverflow.com/tour>