Documentación LearnWeb

Índice

Introducción

Objetivos

Arquitectura

Sigue el modelo MVC implementado de manera semi-automática gracias a Symfony, que conforma nuestro backend.

```
src/
|-- Controller/
|-- Entity/
|-- Repository/
```

Base de datos

Base de datos relacional mysql, llamada learnweb y funcionando en el puerto 3306.

Modelo E/R

Esquema de Tablas SQL

Tipos de Usuarios

Administradores

ROLE_ADMIN

Usuarios que pueden gestionar el estado de baneo a otros usuarios. Editar cualquier tutorial o curso, mostrar/ocultar tutoriales o cursos. Registrar nuevos administradores.

Estudiante

ROLE_LEARNER

Usuarios corrientes que pueden crear tutoriales/cursos y editar los suyos propios, mostrar/ocultar estos. Puntuar tutoriales/cursos, ver tutoriales/cursos. Los administradores incluyen estas funciones.

Mapa de Navegación

```
/register
/login
|-- /
|---- courses/
|---- tutorials/
|---- my-courses/
|---- my-tutorials/
|---- new-course/
|---- new-tutorial/
|---- admin-panel/
|------ admin-users/
```

Manual de instalación

Instalar MySQL

Debian / Ubuntu

1. Lo instalamos e iniciamos

```
sudo apt update && sudo apt install mysql-server
systemctl start mysql
```

2. Accedemos a la BBDD

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

Contraseña: `root`

3. Ya podemos ejecutar cualquier consulta, para salir basta con escribir `exit`

En Windows

1. Lo descargamos desde esta URL: MySQL
2. Lo añadimos al PATH de `Variables de entorno del sistema`
3. Accedemos a la BBDD

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

Contraseña: `root`

4. Ya podemos ejecutar cualquier consulta, para salir basta con escribir `exit`

Configurar Symfony

Instalamos PHP

Ubuntu / Debian

1. Instalación

```
sudo apt update
sudo apt install php php-cli php-fpm php-mysql php-curl php-zip php-xml php-
mbstring php-intl
```

2. Verificamos la instalación

```
php -v
```

Windows

1. En Windows accedemos a este enlace y elegimos la versión Non-Thread Safe.
2. Luego extraemos el ZIP en C:\php
3. Configuramos el PATH en "Variables de entorno del sistema", puedes buscarlo en la barra de búsqueda de Windows, y en PATH agregamos C:\php
4. Verificamos la instalación

```
php -v
```

Instalamos Composer

Linux

1. Instalamos

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

2. Movemos composer a un directorio global

```
mv composer.phar /usr/local/bin/composer
```

3. Verificamos la instalación

```
composer -v
```

Windows

1. Descargamos el instalador desde este enlace en manual download
2. Ejecutamos el instalador y nos aseguramos de marcar la opción de "Add to PATH"
3. Verificamos la instalación

```
composer -v
```

Seguimos con Symfony

1. Nos vamos a la raíz del backend (LearnWebApi) e instalamos las dependencias

```
composer install
```

2. Ejecutamos las migraciones para crear las tablas

```
php bin/console doctrine:migrations:migrate
```

En caso de que no estuvieran creadas las migraciones, las creamos

```
php bin/console make:migration
```

Y ahora si las migraríamos con el comando del paso 2

Claves PEM para la gestión de los JWT Token

Linux Debian

1. Instalamos openssl

```
sudo apt install openssl -y
```

2. Comprobamos que se ha instalado

```
openssl version
```

3. Creamos la carpeta config/jwt desde la raíz del backend

```
mkdir -p config/jwt
```

4. Generamos las claves

```
openssl genpkey -out config/jwt/private.pem -aes256 -algorithm rsa -pkeyopt
rsa_keygen_bits:4096
```

5. Nos pedirá un PEM Pass Phrase, introducimos la que tenemos en .env

6. Terminamos de generar las claves

```
openssl pkey -in config/jwt/private.pem -out config/jwt/public.pem -pubout
```

7. Nos aseguramos de tener permisos para acceder a las claves

```
chmod 600 config/jwt/private.pem
chmod 644 config/jwt/public.pem
```

Windows

1. Lo descargamos desde el siguiente enlace (que sea la versión lite preferiblemente)

2. Añadimos C:\Program Files\OpenSSL-Win64\bin al PATH de variables de entorno del sistema

3. Comprobamos que ya está correctamente instalado y usable

```
openssl version
```

4. Creamos la carpeta config/jwt desde la raíz del backend

```
mkdir -p config/jwt
```

5. Generamos las claves

```
openssl genpkey -out config/jwt/private.pem -aes256 -algorithm rsa -pkeyopt
rsa_keygen_bits:4096
```

6. Nos pedirá un PEM Pass Phrase, introducimos la que tenemos en .env

7. Terminamos de generar las claves

```
openssl pkey -in config/jwt/private.pem -out config/jwt/public.pem -pubout
```

8. Nos aseguramos de cambiar los permisos para poder acceder con nuestro usuario de Windows a los archivos

Terminamos con Symfony

1. Iniciamos el servidor Symfony

```
symfony server:start
```

El backend estará en http://localhost:8000 y http://127.0.0.1:8000, puedes acceder a la api con un /api al final

2. Comprobamos que está funcionando

```
symfony server:status
```

Para parar el servidor:

```
symfony server:stop
```

Configuramos Astro

Instalar Node.js

Ubuntu / Debian

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
```

Verificamos la instalación

```
node -v
```

Windows

Descargamos Node.js (versión LTS) desde aquí

Al ejecutar el instalador nos aseguramos de marcar la opción "Add to PATH"

Verificamos la instalación

```
node -v
```

Terminamos con Astro

En la raíz del frontend, instalamos las dependecias del proyecto

```
npm install
```

E iniciamos el frontend

```
npm run dev
```

Accesible desde http://localhost:4321

Supuestamente ya debería de estar todo en funcionamiento, sería el turno de rellenar la BBDD con datos si se desea

Comprobaciones finales

Comprobamos que todos los servicios estén funcionando en sus respectivos puertos

Ubuntu / Debian

```
sudo lsof -i :4321,3306,8000
```

Deberíamos ver mysql, Symfony y node

Windows

```
netstat -ano | findstr ":4321 :3306 :8000"
```

Manual de usuario

Manual de administración

Fase de pruebas

Datos de prueba para la BBDD

1. Los datos los tendremos en SQL directamente para introducirlos en la BBDD, así que nos metemos en la consola de MySQL

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

Contraseña: root

2. Una vez dentro, solo queda copiar y pegar por orden (importante) para no tener problemas de relaciones entre tablas

Usuarios

Primero dejaré por aquí los datos de inicio de sesión de cada usuario por si queremos utilizarlos para iniciar sesión luego

Nombre de usuario -> contraseña -> rol

1. SuperAdmin -> AdminPass123! -> Admin
2. Aprendiz1 -> UserPass456! -> Aprendiz
3. AdminHelper -> Secure789! -> Admin
4. JuanP -> JuanPerez123 -> Aprendiz
5. MariaG -> MariaG#2023 -> Aprendiz
6. AlexS -> AlexSmith! -> Aprendiz
7. SysAdmin -> SysAdmin#456 -> Admin
8. LauraB -> LauraBlue$ -> Aprendiz
9. DavidK -> KingDavid2023 -> Aprendiz

10. SaraC -> SaraConor! -> Aprendiz

Las contraseñas tienen cifrado bcrypt en la BBDD

La sentencia SQL para introducirlos:

```
INSERT INTO user (user_type, email, passwd, nickname, banned) VALUES
('ROLE_ADMIN', 'admin1@gmail.com',
'$2a$10$6khzLWyE6SxgJifuv8rK4uD38hGSAZfFUIulISrApoFUCm5coofH2', 'SuperAdmin', 0),
('ROLE_LEARNER', 'learner1@gmail.com',
'$2a$10$dvpf9gk55QkYDWWmWJEkJuO4NZzpFECPVAFuUk2qhI3bTGGD48Mn6', 'Aprendiz1', 0),
('ROLE_ADMIN', 'admin2@hotmail.com',
'$2a$10$ZWl.ZeES6Wx8tz/5Dn6deuJQIS168Lccq2mooC0jY7q.Lh4OU9eRa', 'AdminHelper', 0),
('ROLE_LEARNER', 'juan.perez@gmail.com',
'$2a$10$adm953ytOgLaSHO96UIWb.b5eAAL8FI91ODX9rMfsnUEw8WZ2.Eq2', 'JuanP', 1),
('ROLE_LEARNER', 'maria.garcia@hotmail.com',
'$2a$10$Q/kK2la4XrlB/Qwji8jclu3Km5pFfTCyHmqt0ewKnpBgComvdqIjS', 'MariaG', 0),
('ROLE_LEARNER', 'alex.smith@gmail.com',
'$2a$10$LsDxPMgktocr1AP.9nagO.mIo7NsbVxLDpZvVrOIqt9my6zmButci', 'AlexS', 0),
('ROLE_ADMIN', 'sys.admin@hotmail.com',
'$2a$10$GeUN3DCNIUoWsJPMJNsL0uCMc78Lfe7BTgrupuXcdu.MyRCts.s/m', 'SysAdmin', 0),
('ROLE_LEARNER', 'laura.blue@gmail.com',
'$2a$10$CDY9z3Nn1ohDZFjonbEW4O46I90rOT6IGzeEoTuWq09PrdLn0yQ02', 'LauraB', 0),
('ROLE_LEARNER', 'david.king@hotmail.com',
'$2a$10$QS39pp3l1V8w58ePp3bpQut0uuIVED6T75.m0Gfi2sJ8aJAPbpkZ.', 'DavidK', 1),
('ROLE_LEARNER', 'sara.conor@gmail.com',
'$2a$10$MTY.XCoH8/o74BzuxRzjPO/vqWyQFOQg0eK2Hxm2QS00/hnJAguTu', 'SaraC', 0);
```

Tutoriales

```
INSERT INTO tutorial (name, description, author_id, hidden, add_date, mod_date)
VALUES ('Introducción a Python', 'Aprende los fundamentos de Python', 1, 0, NOW(),
NOW()), ('JavaScript Básico', 'Conceptos esenciales de JavaScript', 2, 0, NOW(),
NOW()), ('HTML para principiantes', 'Guía básica de estructura web con HTML', 3,
0, NOW(), NOW()), ('CSS desde cero', 'Estiliza tus páginas web con CSS', 4, 0,
NOW(), NOW()), ('SQL Fundamentos', 'Aprende a manejar bases de datos', 5, 0,
NOW(), NOW()), ('Java OOP', 'Programación Orientada a Objetos en Java', 6, 0,
NOW(), NOW()), ('React Intro', 'Primeros pasos con React', 7, 0, NOW(), NOW()),
('Node.js Básico', 'Introducción a Node.js para backend', 8, 0, NOW(), NOW()),
('Git y GitHub', 'Control de versiones con Git', 9, 0, NOW(), NOW()), ('C++
Moderno', 'Conceptos actuales de C++', 10, 0, NOW(), NOW()), ('Python Avanzado',
'Técnicas avanzadas en Python', 1, 0, NOW(), NOW()), ('JavaScript ES6+',
'Características modernas de JavaScript', 2, 0, NOW(), NOW()), ('Responsive
Design', 'Diseño web adaptable con CSS', 3, 0, NOW(), NOW()), ('SQL Avanzado',
'Consultas complejas y optimización', 4, 0, NOW(), NOW()), ('Spring Framework',
'Desarrollo con Spring en Java', 5, 0, NOW(), NOW()), ('React Hooks', 'Uso
avanzado de Hooks en React', 6, 0, NOW(), NOW()), ('Express.js', 'Creación de APIs
con Express', 7, 0, NOW(), NOW()), ('Git Avanzado', 'Flujos de trabajo
profesionales con Git', 8, 0, NOW(), NOW()), ('C++ STL', 'Biblioteca estándar de
plantillas en C++', 9, 0, NOW(), NOW()), ('Python Data Science', 'Introducción a
```

```
pandas y numpy', 10, 0, NOW(), NOW()), ('TypeScript', 'JavaScript tipado para
aplicaciones robustas', 1, 0, NOW(), NOW()), ('Vue.js', 'Framework progresivo de
JavaScript', 2, 0, NOW(), NOW()), ('SASS/SCSS', 'Preprocesadores CSS para estilos
avanzados', 3, 0, NOW(), NOW()), ('NoSQL con MongoDB', 'Bases de datos no
relacionales', 4, 0, NOW(), NOW()), ('Kotlin', 'Lenguaje moderno para Android', 5,
0, NOW(), NOW()), ('Angular', 'Framework frontend de Google', 6, 0, NOW(), NOW()),
('Docker', 'Contenedores para desarrollo', 7, 0, NOW(), NOW()), ('GitLab CI/CD',
'Integración y despliegue continuo', 8, 0, NOW(), NOW()), ('Rust', 'Lenguaje
seguro y concurrente', 9, 0, NOW(), NOW()), ('Python Web Scraping', 'Extracción de
datos web con Python', 10, 1, NOW(), NOW());
```

Páginas

```
INSERT INTO page (tutorial_id, title, description, order_number) VALUES (1,
'Instalación de Python', 'Cómo instalar Python en tu sistema', 1), (1, 'Hola Mundo
en Python', 'Primer programa en Python', 2), (1, 'Variables y tipos', 'Manejo de
variables en Python', 3), (2, 'Configuración inicial JS', 'Preparar entorno para
JavaScript', 1), (2, 'Sintaxis básica', 'Fundamentos de sintaxis en JavaScript',
2), (3, 'Estructura HTML', 'Etiquetas básicas de HTML', 1), (3, 'Formularios
HTML', 'Creación de formularios web', 2), (4, 'Selectores CSS', 'Cómo seleccionar
elementos', 1), (4, 'Box Model', 'Modelo de caja en CSS', 2), (5, 'Consultas
SELECT', 'Búsqueda de datos en SQL', 1), (5, 'Cláusulas WHERE', 'Filtrado de
resultados', 2), (6, 'Clases en Java', 'Conceptos de POO en Java', 1), (6,
'Herencia', 'Relaciones entre clases', 2), (7, 'Componentes React', 'Creación de
componentes', 1), (7, 'Estado y props', 'Manejo de datos en React', 2), (8,
'Módulos Node.js', 'Sistema de módulos', 1), (8, 'Servidor HTTP', 'Crear servidor
básico', 2), (9, 'Comandos Git', 'Comandos esenciales', 1), (9, 'Ramificaciones',
'Trabajo con branches', 2), (10, 'Punteros en C++', 'Manejo de memoria', 1), (10,
'Templates', 'Plantillas en C++', 2), (11, 'Decoradores Python', 'Funciones
avanzadas', 1), (11, 'Generadores', 'Creación de generadores', 2), (12, 'Arrow
functions', 'Funciones flecha en ES6', 1), (12, 'Desestructuración', 'Asignación
desestructurada', 2), (13, 'Media Queries', 'Adaptabilidad en CSS', 1), (13,
'Flexbox', 'Diseño con Flexbox', 2), (14, 'JOINs SQL', 'Unión de tablas', 1), (14,
'Subconsultas', 'Consultas anidadas', 2), (15, 'Inyección Spring', 'Dependencias
en Spring', 1), (15, 'Spring Boot', 'Configuración rápida', 2), (16, 'useState',
'Manejo de estado', 1), (16, 'useEffect', 'Efectos secundarios', 2), (17, 'Rutas
Express', 'Sistema de enrutamiento', 1), (17, 'Middleware', 'Funciones
intermedias', 2), (18, 'Rebase', 'Reorganización de commits', 1), (18, 'Cherry-
pick', 'Selección de commits', 2), (19, 'Vectores STL', 'Contenedores
secuenciales', 1), (19, 'Algoritmos STL', 'Funciones predefinidas', 2), (20,
'pandas básico', 'DataFrames en Python', 1), (20, 'numpy arrays', 'Arrays
multidimensionales', 2), (21, 'Tipos en TS', 'Sistema de tipos', 1), (21,
'Interfaces', 'Definición de contratos', 2), (22, 'Directivas Vue',
'Funcionalidades template', 1), (22, 'Vuex', 'Manejo de estado', 2), (23,
'Variables SASS', 'Reutilización de valores', 1), (23, 'Mixins', 'Funciones
reutilizables', 2), (24, 'Consultas MongoDB', 'Búsqueda documentos', 1), (24,
'Agregaciones', 'Pipeline de datos', 2), (25, 'Null safety', 'Manejo de nulos',
1);
```

Bloques

```
INSERT INTO block (page_id, type, content, order_number) VALUES (1, 'title',
'Instalación de Python', 1), (1, 'text', 'Python es un lenguaje interpretado, por
lo que necesitamos instalarlo antes de usarlo.', 2), (1, 'code', 'sudo apt-get
install python3', 3), (2, 'title', 'Hola Mundo en Python', 1), (2, 'text', 'El
tradicional primer programa en cualquier lenguaje.', 2), (2, 'code', 'print("Hola
Mundo")', 3), (3, 'title', 'Variables en Python', 1), (3, 'text', 'Python es de
tipado dinámico, no necesitas declarar el tipo.', 2), (3, 'code', 'nombre =
"Juan"\nedad = 25', 3), (4, 'title', 'Configuración de JavaScript', 1), (4,
'text', 'Puedes usar JS directamente en el navegador o con Node.js', 2), (4,
'code', 'console.log("JavaScript listo");', 3), (5, 'title', 'Sintaxis básica JS',
1), (5, 'text', 'JavaScript usa sintaxis similar a C/Java', 2), (5, 'code', 'let x
= 5;\nconst y = 10;', 3), (6, 'title', 'Estructura HTML', 1), (6, 'text', 'HTML se
compone de etiquetas que definen elementos', 2), (6, 'code', '<html>\n<head>
</head>\n<body></body>\n</html>', 3), (7, 'title', 'Formularios HTML', 1), (7,
'text', 'Los formularios permiten entrada de datos', 2), (7, 'code',
'<form>\n<input type="text">\n</form>', 3), (8, 'title', 'Selectores CSS', 1), (8,
'text', 'Los selectores apuntan a elementos HTML', 2), (8, 'code', 'p {\n  color:
red;\n}', 3), (9, 'title', 'Box Model', 1), (9, 'text', 'Todo elemento es una caja
con márgenes y padding', 2), (9, 'code', 'div {\n  margin: 10px;\n  padding:
20px;\n}', 3), (10, 'title', 'SELECT en SQL', 1), (10, 'text', 'Consulta básica
para recuperar datos', 2), (10, 'code', 'SELECT * FROM usuarios;', 3), (11,
'title', 'WHERE en SQL', 1), (11, 'text', 'Filtra resultados según condiciones',
2), (11, 'code', 'SELECT * FROM productos WHERE precio > 100;', 3), (12, 'title',
'Clases en Java', 1), (12, 'text', 'Las clases son plantillas para objetos', 2),
(12, 'code', 'public class Persona {\n  String nombre;\n}', 3), (13, 'title',
'Herencia en Java', 1), (13, 'text', 'Una clase puede heredar de otra', 2), (13,
'code', 'public class Empleado extends Persona {}', 3), (14, 'title', 'Componentes
React', 1), (14, 'text', 'Bloques fundamentales de una app React', 2), (14,
'code', 'function MiComponente() {\n  return <div>Hola</div>;\n}', 3), (15,
'title', 'Estado en React', 1), (15, 'text', 'El estado almacena datos del
componente', 2), (15, 'code', 'const [contador, setContador] = useState(0);', 3),
(16, 'title', 'Módulos Node', 1), (16, 'text', 'Node.js usa módulos para organizar
código', 2), (16, 'code', 'const fs = require("fs");', 3), (17, 'title', 'Servidor
HTTP', 1), (17, 'text', 'Crear un servidor básico en Node', 2), (17, 'code',
'http.createServer((req, res) => {\n  res.end("Hola");\n});', 3), (18, 'title',
'Comandos Git', 1), (18, 'text', 'Comandos básicos para empezar', 2), (18, 'code',
'git init\ngit add .\ngit commit -m "Mensaje"', 3), (19, 'title', 'Ramas en Git',
1), (19, 'text', 'Las ramas permiten trabajo paralelo', 2), (19, 'code', 'git
branch nueva-rama\ngit checkout nueva-rama', 3), (20, 'title', 'Punteros C++', 1),
(20, 'text', 'Los punteros almacenan direcciones de memoria', 2), (20, 'code',
'int* ptr = &variable;', 3), (21, 'title', 'Templates C++', 1), (21, 'text',
'Plantillas para funciones genéricas', 2), (21, 'code', 'template <typename T>\nT
max(T a, T b) { return a > b ? a : b; }', 3), (22, 'title', 'Decoradores Python',
1), (22, 'text', 'Funciones que modifican otras funciones', 2), (22, 'code',
'@decorador\ndef funcion(): pass', 3), (23, 'title', 'Generadores Python', 1),
(23, 'text', 'Producen valores sobre la marcha', 2), (23, 'code', 'def
generador():\n  yield 1\n  yield 2', 3), (24, 'title', 'Arrow Functions', 1), (24,
'text', 'Sintaxis compacta para funciones', 2), (24, 'code', 'const suma = (a, b)
=> a + b;', 3), (25, 'title', 'Desestructuración', 1), (25, 'text', 'Extraer
valores de objetos/arrays', 2), (25, 'code', 'const { nombre, edad } = persona;',
3), (26, 'title', 'Media Queries', 1), (26, 'text', 'Adaptar estilos a diferentes
pantallas', 2), (26, 'code', '@media (max-width: 600px) {\n  body { font-size:
```

```
14px; }\n}', 3), (27, 'title', 'Flexbox', 1), (27, 'text', 'Modelo de diseño
flexible', 2), (27, 'code', '.contenedor {\n  display: flex;\n}', 3), (28,
'title', 'JOINs SQL', 1), (28, 'text', 'Combinar datos de múltiples tablas', 2),
(28, 'code', 'SELECT * FROM tabla1 JOIN tabla2 ON tabla1.id = tabla2.id;', 3),
(29, 'title', 'Subconsultas SQL', 1), (29, 'text', 'Consultas dentro de otras
consultas', 2), (29, 'code', 'SELECT nombre FROM usuarios WHERE id IN (SELECT
user_id FROM pedidos);', 3), (30, 'title', 'Inyección Spring', 1), (30, 'text',
'Spring maneja las dependencias automáticamente', 2), (30, 'code',
'@Autowired\nprivate Servicio servicio;', 3), (31, 'title', 'Spring Boot', 1),
(31, 'text', 'Configuración automática para Spring', 2), (31, 'code',
'@SpringBootApplication\npublic class MiApp {}', 3), (32, 'title', 'useState
Hook', 1), (32, 'text', 'Manejar estado en componentes funcionales', 2), (32,
'code', 'const [valor, setValor] = useState(inicial);', 3), (33, 'title',
'useEffect Hook', 1), (33, 'text', 'Efectos secundarios en componentes', 2), (33,
'code', 'useEffect(() => {\n  // efecto\n}, [dependencias]);', 3), (34, 'title',
'Rutas Express', 1), (34, 'text', 'Definir endpoints de la API', 2), (34, 'code',
'app.get("/ruta", (req, res) => res.send("Hola"));', 3), (35, 'title', 'Middleware
Express', 1), (35, 'text', 'Funciones que procesan peticiones', 2), (35, 'code',
'app.use((req, res, next) => {\n  // lógica\n  next();\n});', 3), (36, 'title',
'Rebase en Git', 1), (36, 'text', 'Reorganizar el historial de commits', 2), (36,
'code', 'git rebase -i HEAD~3', 3), (37, 'title', 'Cherry-pick', 1), (37, 'text',
'Aplicar commits específicos', 2), (37, 'code', 'git cherry-pick abc123', 3), (38,
'title', 'Vectores STL', 1), (38, 'text', 'Contenedores dinámicos de C++', 2),
(38, 'code', 'std::vector<int> numeros = {1, 2, 3};', 3), (39, 'title',
'Algoritmos STL', 1), (39, 'text', 'Funciones predefinidas para colecciones', 2),
(39, 'code', 'std::sort(vector.begin(), vector.end());', 3), (40, 'title', 'pandas
DataFrame', 1), (40, 'text', 'Estructura principal de pandas', 2), (40, 'code',
'import pandas as pd\ndf = pd.DataFrame(datos);', 3), (41, 'title', 'numpy
arrays', 1), (41, 'text', 'Arrays multidimensionales eficientes', 2), (41, 'code',
'import numpy as np\narr = np.array([1, 2, 3]);', 3), (42, 'title', 'Tipos en
TypeScript', 1), (42, 'text', 'TypeScript añade tipos a JavaScript', 2), (42,
'code', 'let nombre: string = "Juan";', 3), (43, 'title', 'Interfaces TS', 1),
(43, 'text', 'Definen la forma de un objeto', 2), (43, 'code', 'interface Usuario
{\n  id: number;\n  nombre: string;\n}', 3), (44, 'title', 'Directivas Vue', 1),
(44, 'text', 'Atributos especiales en templates', 2), (44, 'code', '<p v-
if="visible">Texto</p>', 3), (45, 'title', 'Vuex Store', 1), (45, 'text', 'Manejo
de estado centralizado', 2), (45, 'code', 'const store = new Vuex.Store({\n
state: { contador: 0 }\n});', 3), (46, 'title', 'Variables SASS', 1), (46, 'text',
'Almacenan valores reutilizables', 2), (46, 'code', '$color-primario: #3498db;',
3), (47, 'title', 'Mixins SASS', 1), (47, 'text', 'Bloques reutilizables de
estilos', 2), (47, 'code', '@mixin centrar {\n  display: flex;\n  justify-content:
center;\n}', 3), (48, 'title', 'Consultas MongoDB', 1), (48, 'text', 'Buscar
documentos en colecciones', 2), (48, 'code', 'db.usuarios.find({ edad: { $gt: 18 }
});', 3), (49, 'title', 'Agregaciones MongoDB', 1), (49, 'text', 'Procesamiento de
datos complejo', 2), (49, 'code', 'db.ventas.aggregate([\n  { $group: { _id:
"$producto", total: { $sum: "$cantidad" } } }\n]);', 3), (50, 'title', 'Null
Safety', 1), (50, 'text', 'Manejo seguro de valores nulos', 2), (50, 'code', 'var
nombre: String? = null', 3);
```

Puntuaciones a tutoriales

```
INSERT INTO tutorial_score (tutorial_id, user_id, score) VALUES (1, 1, 5), (2, 2,
4), (3, 3, 3), (4, 4, 5), (5, 5, 4), (6, 6, 5), (7, 7, 3), (8, 8, 4), (9, 9, 5),
(10, 10, 4), (11, 1, 5), (12, 2, 3), (13, 3, 4), (14, 4, 5), (15, 5, 2), (16, 6,
4), (17, 7, 5), (18, 8, 3), (19, 9, 4), (20, 10, 5), (21, 1, 4), (22, 2, 3), (23,
3, 5), (24, 4, 4), (25, 5, 3), (1, 2, 4), (2, 3, 5), (3, 4, 3), (4, 5, 4), (5, 6,
5), (6, 7, 4), (7, 8, 3), (8, 9, 5), (9, 10, 4), (10, 1, 3), (11, 2, 5), (12, 3,
4), (13, 4, 3), (14, 5, 5), (15, 6, 4), (16, 7, 2), (17, 8, 4), (18, 9, 5), (19,
10, 3), (20, 1, 4), (21, 2, 5), (22, 3, 4), (23, 4, 3), (24, 5, 5), (25, 6, 4);
```

Cursos

```
INSERT INTO course (author_id, name, description, difficulty, hidden, add_date,
mod_date) VALUES (1, 'Python Completo', 'De básico a avanzado en Python',
'beginner', 0, NOW(), NOW()), (2, 'JavaScript Moderno', 'ES6+ y frameworks
modernos', 'intermediate', 0, NOW(), NOW()), (3, 'Desarrollo Web Full Stack',
'HTML, CSS y JavaScript completo', 'beginner', 0, NOW(), NOW()), (4, 'Bases de
Datos Profesional', 'SQL y NoSQL para aplicaciones reales', 'advanced', 0, NOW(),
NOW()), (5, 'Java Enterprise', 'Desarrollo de aplicaciones empresariales',
'advanced', 0, NOW(), NOW()), (6, 'React Profesional', 'Patrones avanzados en
React', 'intermediate', 0, NOW(), NOW()), (7, 'Node.js Avanzado', 'Backend con
Node y Express', 'intermediate', 0, NOW(), NOW()), (8, 'DevOps Essentials', 'Git,
CI/CD y despliegues', 'intermediate', 0, NOW(), NOW()), (9, 'C++ para Sistemas',
'Programación de bajo nivel', 'advanced', 0, NOW(), NOW()), (10, 'Data Science con
Python', 'Análisis de datos y machine learning', 'advanced', 0, NOW(), NOW()), (1,
'Python para Automatización', 'Automatiza tareas con Python', 'beginner', 0,
NOW(), NOW()), (3, 'CSS Avanzado', 'Animaciones y diseños complejos',
'intermediate', 1, NOW(), NOW()), (5, 'Spring Boot', 'Microservicios con Spring',
'advanced', 0, NOW(), NOW()), (7, 'APIs REST', 'Diseño de APIs profesionales',
'intermediate', 0, NOW(), NOW()), (9, 'Algoritmos en C++', 'Estructuras de datos y
algoritmos', 'advanced', 0, NOW(), NOW());
```

Tutorial dentro de curso

```
INSERT INTO tutorial_in_course (course_id, tutorial_id, order_number) VALUES (1,
1, 1), (1, 11, 2), (1, 20, 3), (2, 2, 1), (2, 12, 2), (2, 21, 3), (3, 3, 1), (3,
4, 2), (3, 13, 3), (4, 5, 1), (4, 14, 2), (4, 24, 3), (5, 6, 1), (5, 15, 2), (6,
7, 1), (6, 16, 2), (6, 22, 3), (7, 8, 1), (7, 17, 2), (8, 9, 1), (8, 18, 2), (9,
10, 1), (9, 19, 2), (10, 11, 1), (10, 20, 2), (11, 1, 1), (11, 11, 2), (12, 3, 1),
(12, 13, 2), (13, 6, 1), (13, 15, 2), (14, 8, 1), (14, 17, 2), (15, 10, 1), (15,
19, 2), (15, 29, 3);
```

Puntuación a curso

```
INSERT INTO course_score (course_id, user_id, score) VALUES (1, 2, 5), (1, 5, 4),
(2, 3, 4), (2, 10, 5), (3, 1, 3), (3, 8, 4), (4, 6, 5), (4, 9, 4), (5, 2, 5), (5,
```

```
7, 3), (6, 4, 4), (6, 10, 5), (7, 3, 4), (7, 9, 3), (8, 5, 5), (8, 10, 4), (9, 2,
3), (9, 8, 5), (10, 4, 4), (10, 7, 5), (11, 3, 3), (11, 9, 4), (12, 6, 5), (12,
10, 3), (13, 4, 4), (13, 8, 5), (14, 2, 3), (14, 9, 4), (15, 3, 5), (15, 7, 4);
```

Ya podemos salir de la consola

```
exit
```

Docker

La comunicación entre contenedores docker no ha funcionado correctamente.

En este apartado veremos mi configuración de Docker en el proyecto.

Instalación

1. Dependencias

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
```

2. Comprobamos que funciona

```
sudo docker run hello-world
```

3. Le damos permisos a nuestro usuario para utilizar docker

```
sudo usermod -aG docker miguelarroyo
```

Y aplicamos los cambios sin necesidad de reiniciar

```
newgrp docker
```

4. Comprobamos la versión de docker

```
docker compose version
```

5. Ejecutamos los contenedores, debemos estar situados en la raíz del proyecto

```
docker compose up -d --build
```

Si algún puerto está en uso, podemos ver el servicio que lo ocupa y pararlo si es necesario

Ejemplo con mysql:

```
sudo lsof -i :3306
sudo systemctl stop mysql
```

Y volvemos a ejecutar los contenedores

6. Una vez iniciados, podemos ver los contendores en activo

```
docker ps
```

Mi output:

```
CONTAINER ID   IMAGE              COMMAND                  CREATED
STATUS          PORTS                  NAMES
61a6be608650   learnweb-nginx     "/docker-entrypoint.…"   20 minutes ago   Up 20
minutes   0.0.0.0:8888->80/tcp   learnweb-nginx-1
69131e0d368d   learnweb-backend   "docker-php-entrypoi…"   20 minutes ago   Up 20
minutes   8000/tcp, 9000/tcp     learnweb-backend-1
62817037b396   mysql:8            "docker-entrypoint.s…"   20 minutes ago   Up 20
minutes   3306/tcp, 33060/tcp    learnweb-db-1
3b6c45c40330   learnweb-frontend  "docker-entrypoint.s…"   20 minutes ago   Up 20
minutes   4321/tcp               learnweb-frontend-1
```

7. Vamos a ejecutar las migraciones para crear la base de datos

```
docker exect -it learnweb-backend-1 bash
```

Una vez dentro de bash

```
php bin/console doctrine:migrations:migrate
```

Si pide confirmación, escribe `yes` y pulsa `ENTER`

8. Una vez creadas, accedemos a la consola

```
docker exec -it learnweb-db-1 mysql -u root -p LearnWeb
```

9. Dentro de la consola, ya podemos ejecutar cualquier sentencia SQL, como por ejemplo:

```
SHOW TABLES;
```

Deberemos meternos en la consola para introducir los datos de prueba mediante sentencias SQL.

Podemos salirnos de la consola con `exit`.

Para reiniciar los contenedores, usamos:

```
docker compose down && docker compose up -d --build
```

Deberíamos de poder acceder desde el navegador a http://localhost:8888/login

Fallos

Básicamente estos son los fallos de comunicación entre contenedores:

- Podemos iniciar sesión -> fetch (/api/login/nickname/passwd), pero no podemos obtener los tutoriales disponibles (/api/tutorialsavaidable)

El la petición del login no está protegido por token, pero los tutoriales disponibles si.

El error dice `Failed to parse URL from /api/tutorialsavaidable`

Según los logs del contenedor backend, ni siquiera llega la petición

Toda la configuración relacionada con Docker y Nginx

docker-compose.yaml:

```
version: "3.8"

services:
    nginx:
        build: ./nginx
        ports:
        - "8888:80"
        depends_on:
        - frontend
        - backend
        networks:
        - app-network
```

```
    frontend:
        build: ./LearnWebAstro
        command: npm run dev -- --host 0.0.0.0
        #ports:
        #  - "4321:4321"
        environment:
        - PUBLIC_API=/api
        volumes:
        - ./LearnWebAstro:/app
        - /app/node_modules
        #depends_on:
        #  - backend
        networks:
        - app-network

    backend:
        build: ./LearnWebApi
        #ports:
        #  - "8000:8000"
        environment:
        - DATABASE_URL=mysql://root:root@db:3306/LearnWeb
        - APP_ENV=dev
        volumes:
        - ./LearnWebApi:/var/www/symfony
        depends_on:
        - db
        networks:
        - app-network

    db:
        image: mysql:8
        environment:
        MYSQL_ROOT_PASSWORD: root
        MYSQL_DATABASE: LearnWeb
        #ports:
        #  - "3306:3306"
        volumes:
        - mysql_data:/var/lib/mysql
        networks:
        - app-network

networks:
app-network:
    driver: bridge

volumes:
mysql_data:
```

nginx/Dockerfile:

```
FROM nginx:alpine

COPY nginx.conf /etc/nginx/conf.d/default.conf

EXPOSE 80
```

nginx/nginx.conf

```
server {
    listen 80;
    server_name localhost;

    # Frontend (Astro)
    location / {
        proxy_pass http://frontend:4321;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # HMR support
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }

    # Backend (Symfony) - Cambiado a /api/
    location /api/ {
        proxy_pass http://backend:8000/;  # Barra final importante
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

}
```

LearnWebAstro/Dockerfile

```
FROM node:lts

WORKDIR /app

COPY package\*.json ./
RUN npm install
```

```
COPY . .

ENV HOST=0.0.0.0
ENV PORT=4321

EXPOSE 4321

CMD ["npm", "run", "dev", "--", "--host", "0.0.0.0"]
```

LearnWebAstro/.dockerignore

```
.DS_Store
node_modules
dist
.env
```

LearnWebAstro/src/api/api.js

```
// Prefijo para las rutas (uso en Docker + proxy inverso Nginx)
const API_PREFIX = "/api";

/**
* Construye una URL completa para los endpoints de la API, eliminando /api
duplicados, solo útil realmente cuando usamos Docker + proxy inverso Nginx
* @param {string} endpoint - Ruta del endpoint
* @returns {string} URL completa formateada
*/
const buildUrl = (endpoint) => {
    // Elimina cualquier /api duplicado
    const cleanEndpoint = endpoint.replace(/^\/?api\//, "");
    return `${API_PREFIX}/${cleanEndpoint}`;
};

/**
* Realiza una petición GET a la API, sin token, usado principalmente para iniciar
sesión con un usuario cuando aún no se ha guardado token, o para comprobar si
existe un usuario dado el nickname y email (en registro)
* @param {string} endpoint - Ruta del endpoint
* @returns {Promise<Object>} Respuesta de la API parseada como JSON
* @throws {Error} Si la respuesta no es exitosa
*/
const get = async (endpoint) => {
const url = buildUrl(endpoint);
const response = await fetch(`api/${url}`, {
    method: "GET",
```

```
    headers: {
        accept: "application/json",
        "Content-Type": "application/json",
    },
});
if (!response.ok) {
    throw new Error(
        `Error en la petición GET al endpoint: ${endpoint} (${response.status}
${response.statusText})`
    );
}
const data = await response.json();
    return data;
};


/**
* Realiza una petición GET autenticada con JWT
* @param {string} endpoint - Ruta del endpoint
* @param {string} token - Token JWT de autenticación
* @returns {Promise<Object>} Respuesta de la API parseada como JSON
* @throws {Error} Si la respuesta no es exitosa
*/
const get_jwt = async (endpoint, token) => {
    const url = buildUrl(endpoint);
    console.log(`${url}`);
    const response = await fetch(url, {
        method: "GET",
        headers: {
        Authorization: `Bearer ${token}`,
            accept: "application/json",
            "Content-Type": "application/json",
    },
});
if (!response.ok) {
    throw new Error(
        `Error en la petición GET al endpoint: ${endpoint} (${response.status}
${response.statusText})`
    );
}
const data = await response.json();
    return data;
};
```

LearnWebAstro/src/api/useCases.js

```
/**
* Obtiene un usuario por credenciales de acceso
* @param {string} nickname - Nombre de usuario
```

```
* @param {string} passwd - Contraseña
* @returns {Promise<Object>} Datos del usuario
*/
const getUserByLogin = async (nickname, passwd) => {
    const endpoint = `users/login/${nickname}/${passwd}`;
    const data = await get(endpoint);
    return data;
};

/**
* Obtiene tutoriales disponibles (requiere autenticación)
* @param {string} token - Token JWT
* @returns {Promise<Array>} Lista de tutoriales
*/
const getTutorialsAvaidable = async (token) => {
    const endpoint = "tutorialsavaidable";
    const data = await get_jwt(endpoint, token);
    return data;
};
```

LearnWebApi/Dockerfile

```
# Usamos una imagen base de PHP con Composer y Nginx
FROM php:8.2-fpm

# Instalar dependencias
RUN apt-get update && apt-get install -y \
    libicu-dev \
    libpq-dev \
    libzip-dev \
    git \
    unzip \
    libcurl4-openssl-dev \
    libpng-dev \
    libjpeg-dev \
    libfreetype6-dev \
    && docker-php-ext-configure gd --with-freetype --with-jpeg \
    && docker-php-ext-install intl pdo pdo_mysql zip gd curl

# Instalar Symfony CLI
RUN curl -sS https://get.symfony.com/cli/installer | bash \
    && mv /root/.symfony*/bin/symfony /usr/local/bin/symfony

# Instalar Composer
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer

# Definir el directorio de trabajo
WORKDIR /var/www/symfony
```

```
# Copiar los archivos de la aplicación Symfony
COPY . .

# Instalar dependencias de Symfony (Composer)
RUN composer install --no-interaction --prefer-dist

# Exponer el puerto 8000
EXPOSE 8000

# Comando para ejecutar Symfony en modo de desarrollo
CMD ["symfony", "server:start", "--no-tls", "--allow-http", "--allow-all-ip", "--
port=8000"]
```

LearnWebApi/docker-compose.yaml

```
services:
    backend:
        ports:
            - "8000:8000"
```

LearnWebApi/.env

```
CORS_ALLOW_ORIGIN='^https?://(localhost|127\.0\.0\.1|frontend|backend)(:[0-9]+)?$'
```

LearnWebApi/config/packages/nelmio_cors.yaml

```
nelmio_cors:
    defaults:
        allow_credentials: true
        allow_origin: ["*"] # Cambiar a ['http://frontend:4321'] o
['http://localhost:4321'] si deseas restringir
        allow_headers: ["Content-Type", "Authorization"]
        allow_methods: ["GET", "POST", "PUT", "DELETE", "OPTIONS"]
        max_age: 3600
    paths:
        "^/api/":
        allow_origin: ["*"]
        allow_methods: ["*"]
        allow_headers: ["Content-Type", "Authorization"]
```

LearnWebApi/config/packages/security.yaml

```
security:
    password_hashers:
        App\Entity\User:
        algorithm: "auto"
        cost: 15

    providers:
        app_user_provider:
        entity:
            class: App\Entity\User
            property: id

    firewalls:
        login:
        pattern: ^/api/users/login
        stateless: true
        json_login:
            check_path: /api/users/login
            username_path: nickname
            password_path: passwd
            success_handler:
lexik_jwt_authentication.handler.authentication_success
            failure_handler:
lexik_jwt_authentication.handler.authentication_failure

        api:
        pattern: ^/api
        stateless: true
        jwt: ~

    access_control:
        - { path: ^/api/users/login, roles: PUBLIC_ACCESS }
        - { path: ^/api/users/register, roles: PUBLIC_ACCESS }
        - { path: ^/api, roles: ROLE_LEARNER }
        - { path: ^/admin, roles: ROLE_ADMIN }
```

Conclusiones

Justificación tecnológica

Mejoras implementables