

Especificaciones Técnicas del Sistema Integral de Lenguas Extranjeras

1. Información General

Repositorio oficial: <https://github.com/DevEliezerMartinez/Sistema-Integral-de-Lenguas-Extranjeras> **Institución:** Tecnológico Nacional de México, Campus San Marcos **Propósito del sistema:** Proveer una plataforma integral para la gestión administrativa y académica de la Coordinación de Lenguas Extranjeras, facilitando la administración de cursos, grupos, estudiantes y procesos de evaluación.

2. Resumen Ejecutivo

El Sistema Integral de Lenguas Extranjeras es una aplicación web contemporánea, desarrollada mediante una arquitectura desacoplada basada en una SPA (Single Page Application). El frontend y el backend se comunican mediante una API RESTful que intercambia información en formato JSON. El sistema optimiza los procesos internos de la Coordinación de Lenguas Extranjeras, fortaleciendo la eficiencia operativa, la trazabilidad de información y la automatización de procedimientos.

3. Arquitectura del Sistema

3.1 Patrón Arquitectónico

- Tipo:** Single Page Application (SPA)
- Comunicación:** API RESTful
- Formato de datos:** JSON
- Autenticación:** Cookies HTTP-only mediante Laravel Sanctum

3.2 Stack Tecnológico

Backend

- Framework: Laravel 11
- Lenguaje: PHP 8.2
- Base de datos: SQLite (entorno local)
- ORM: Eloquent
- Servidor web en producción: Nginx + PHP-FPM
- Autenticación: Laravel Sanctum

Nota: Para entornos de producción se recomienda utilizar PostgreSQL o MySQL debido a sus capacidades superiores de concurrencia.

Frontend

- Biblioteca: React 18
- Empaquetador: Vite
- Estilos: TailwindCSS
- Enrutador: React Router
- Cliente HTTP: Axios
- Servidor web en producción: Nginx

Infraestructura

- Containerización: Docker
- Orquestación: Docker Compose
- Proxy reverso: Nginx

4. Funcionalidades Principales

4.1 Gestión de usuarios

- Registrar y administrar cuentas de usuario.
- Autenticar sesiones mediante cookies seguras.
- Control de acceso basado en roles y permisos.
- Mantener perfiles con información académica y administrativa.

4.2 Gestión académica

- Administrar cursos de lenguas extranjeras.

- Registrar y dar seguimiento a estudiantes.
- Crear y gestionar grupos y horarios.
- Registrar calificaciones y evaluaciones.

4.3 Gestión administrativa

- Gestionar procesos de inscripción.
- Administrar documentación y expedientes.
- Controlar trámites y procesos internos.

4.4 Sistema de notificaciones

- Emitir alertas administrativas.
- Generar recordatorios de eventos clave.
- Gestionar notificaciones internas.

5. Características Técnicas

5.1 Seguridad

- Autenticación mediante cookies HTTP-only.
- Protección CSRF mediante tokens administrados por Sanctum.
- Validación estricta de datos.
- Sanitización de entradas.
- Encriptación de contraseñas con bcrypt.
- Configuración de CORS para SPA.

5.2 Rendimiento

- Lazy loading de componentes del frontend.
- Caché de configuración y rutas en producción.
- Compilación y minificación de recursos.
- Compresión de respuestas HTTP.
- Optimización de consultas mediante Eloquent.

5.3 Escalabilidad

- Arquitectura desacoplada.
- API stateless (excepto autenticación).
- Containerización reproducible mediante Docker.
- Fácil migración a bases de datos más robustas.

6. Estructura General del Proyecto

```
Sistema-Integral-de-Lenguas-Extranjeras/  
├── Back/                               # Backend Laravel  
│   ├── app/                           # Lógica de aplicación  
│   ├── config/                        # Archivos de configuración  
│   ├── database/                      # Migraciones y seeders  
│   ├── routes/                       # Rutas API  
│   └── storage/                       # Logs y archivos generados  
│  
├── Frontend/                          # Frontend React  
│   ├── src/                          # Código fuente principal  
│   ├── public/                       # Recursos estáticos  
│   └── dist/                          # Compilación para producción  
│  
├── nginx/                             # Configuración de Nginx  
├── docker-compose.yml                 # Entorno de desarrollo  
└── docker-compose.prod.yml           # Entorno de producción
```

7. Despliegue

7.1 Flujo de despliegue recomendado

1. Clonar repositorio.
2. Configurar archivos `.env`.
3. Construir contenedores.
4. Ejecutar migraciones.
5. Poblar la base de datos.
6. Validar funcionamiento del frontend y backend.

7.2 Entorno de desarrollo

```
git clone https://github.com/DevEliezerMartinez/Sistema-Integral-de-Lenguas-Extranjeras
cd Sistema-Integral-de-Lenguas-Extranjeras

docker-compose up --build
```

7.3 Entorno de producción

```
# Configurar IP y variables en docker-compose.prod.yml
# Crear archivo database.sqlite en Back/database/

docker-compose -f docker-compose.prod.yml up --build -d

docker-compose -f docker-compose.prod.yml exec backend php artisan migrate --force
docker-compose -f docker-compose.prod.yml exec backend php artisan db:seed --force
```

8. Requisitos del Sistema

8.1 Desarrollo

- Docker Desktop
- Git
- Editor de código (VS Code recomendado)

8.2 Producción

- Docker Engine
- 2 GB de RAM (mínimo)
- 5 GB de almacenamiento libre
- CPU de dos núcleos (recomendado)
- Conectividad de red estable

9. Mantenimiento y Soporte

9.1 Ubicación de logs

- Backend: `Back/storage/logs/laravel.log`
- Acceso web: logs del contenedor Nginx

9.2 Comandos útiles

```
docker-compose -f docker-compose.prod.yml logs -f          # Ver logs
docker-compose -f docker-compose.prod.yml restart          # Reiniciar servicios
docker-compose -f docker-compose.prod.yml exec backend php artisan cache:clear
docker-compose -f docker-compose.prod.yml exec backend php artisan config:clear
```

10. Licenciamiento

Este proyecto se encuentra disponible bajo la licencia MIT, lo que permite su uso, distribución y modificación con fines académicos y administrativos.

11. Contacto

Desarrollador: Eliezer Solano Martínez **Institución:** Tecnológico Nacional de México, Campus San Marcos **Repositorio:**
<https://github.com/DevEliezerMartinez/Sistema-Integral-de-Lenguas-Extranjeras>
