

# API Laravel y proyecto Nuxt sobre Harry Potter

## Objetivo

Desarrollar una API con la última versión de Laravel y un proyecto web en Nuxt que se alimente de ella, con temática de Harry Potter.

## Funcionalidades

- Zona de administración:
  - Gestión de alumnos:
    - Alta, baja y modificación de alumnos.
    - Visualización de información detallada de cada alumno (casa, año, etc.).
    - Poder cambiar de casa.
  - Gestión de profesores:
    - Alta, baja y modificación de profesores.
    - Visualización de información detallada de cada profesor (asignaturas, etc.).
    - Poder cambiar de casa.
  - Gestión de casas:
    - Alta, baja y modificación de casas.
    - Visualización de información detallada de cada casa (puntos, miembros, etc.).
- Importación de datos:
  - Importación de datos de alumnos, profesores y casas desde las APIs gratuitas de Harry Potter:
    - Estudiantes: <https://hp-api.onrender.com/api/characters/students>
    - Profesores: <https://hp-api.onrender.com/api/characters/staff>
    - Casas:
      - Gryffindor: <https://hp-api.onrender.com/api/characters/house/gryffindor>
      - Hufflepuff: <https://hp-api.onrender.com/api/characters/house/hufflepuff>
      - Ravenclaw: <https://hp-api.onrender.com/api/characters/house/ravenclaw>
      - Slytherin: <https://hp-api.onrender.com/api/characters/house/slytherin>

## Requisitos técnicos

### API

- Desarrollada con la última versión de Laravel.
- Seguir las buenas prácticas de desarrollo de Laravel.
- Implementar la autenticación y autorización de usuarios.
- Exponer endpoints RESTful para la gestión de alumnos, profesores y casas.
- Implementar la paginación y el filtrado de datos.
- Documentar la API con Swagger.

### Proyecto Nuxt

- Desarrollado con Nuxt.js.
- Utilizar Tailwind CSS para el diseño.
- Implementar un diseño atractivo y responsive inspirado en el universo de Harry Potter.
- Consumir la API Laravel para mostrar la información de alumnos, profesores y casas.
- Permitir la búsqueda y filtrado de datos.
- Implementar una experiencia de usuario intuitiva y fluida.

## Recursos adicionales

- Documentación de Laravel: <https://laravel.com/docs/>
- Documentación de Nuxt: <https://nuxtjs.org/docs/>
- Tailwind CSS: <https://tailwindcss.com/>

## Evaluación

Se evaluará la calidad del código, la arquitectura del proyecto, la eficiencia de la API, la usabilidad del proyecto Nuxt y el diseño general.

## Recomendaciones

- Se recomienda dividir el proyecto en tareas más pequeñas y manejables.
- Implementar un sistema de control de versiones como Git.
- Realizar pruebas unitarias y de integración para asegurar la calidad del código.
- Documentar el proyecto de forma clara y concisa.