

Prueba técnica Senior PHP Symfony/Laravel

Necesitamos una prueba piloto de un blog para uno de nuestros productos (es mentira, es un caso para esta prueba 😊). Lo queremos mantener separado, así que debemos empezar un nuevo proyecto.

Empezaremos con algo sencillo, sólo queremos autores y posts (no vamos a gestionar comentarios, categorías, multimedia ni nada de eso). Vamos a necesitar acceder al contenido del blog desde otras aplicaciones, lo que nos requiere exponer una API.

Como modelo de datos te proponemos usar el de <https://jsonplaceholder.typicode.com/>. Puedes usar esa API como sustituto de una base de datos, aunque si lo prefieres puedes usarla como guía e incluso ampliar los campos disponibles creando un servicio tipo *stub*.

Por tanto, este proyecto debe tener 2 secciones:

- **Blog público**
 - Lista de posts
 - Página de post, donde se mostrará una pequeña ficha del autor
- **API**
 - Endpoint GET para la obtención de posts (y en cada post incluir la información del autor)
 - Endpoint POST para la publicación de un post. No es necesario crear una base de datos ni hacer realmente persistencia. En este punto nos interesa que realices la validación de los datos enviados y simules una inserción en base de datos.

El **objetivo** de la prueba es la demostración de conocimientos en Symfony o Laravel (tú eliges) y PHP, por lo que se primará el cuidado de la estructura backend: aplicación de principios SOLID, la correcta separación de servicios y responsabilidades, introducción de interfaces, tolerancia ante fallos, etc. Api Plattform ofrece ciertos atajos que nos impiden ver cómo un desarrollador senior podría resolver ciertas casuísticas, por lo que preferimos evitar su uso.

Para la parte front no requerimos una gran labor de maquetación. Puedes usar librerías adicionales a las del framework si lo necesitas. También en la parte de front, puedes usar Bootstrap, Bulma, Tailwind u otro framework CSS.

Requisitos:

- PHP 7.4 ó 8.0
- Composer y estructura PSR-4 en el proyecto
- La última versión estable de Symfony/Laravel
- Testing unitario
- La API debe devolver y consumir los datos en formato JSON
- Programar en el idioma inglés

- Uso de herramientas de análisis estático (por ejemplo PHPStan en modo máximo) y de estilo de código (por ejemplo PHP CS Fixer en modo @Symfony)
- Uso de SCSS y Webpack, ya sea usando Webpack directamente o mediante Symfony Encore / Laravel Mix o similar
- Ofrecer un Swagger/OpenAPI para la parte de la API

Para la entrega, puedes mandarnos un enlace a un repositorio público (GitHub, GitLab u otro sitio donde tengas cuenta), aunque también nos la puedes mandar un zip al correo (en este caso procura incluir el repositorio git en el fichero comprimido).

Lo que sí nos gustaría que incluyeras es un README explicando la prueba y cualquier cosa de la que quieras dejar constancia, como las decisiones a la hora de organizar el código, las librerías extra que hayas añadido para facilitar el trabajo, problemas que te hayas encontrado, soluciones alternativas que te gustaría haber probado, etc.

Nos gusta que nos den este tipo de feedback ya que nos ayuda a valorar mejor y de forma objetiva la prueba técnica.