

Prueba de conocimientos “PHP Developer”

A continuación, se realiza un ejercicio con el objetivo de tener una noción del dominio y conocimiento en ciertas tecnologías que serán críticas en la ejecución de proyectos del equipo de TI.

1. ¿Laravel tiene algún tipo de ventaja sobre otros Frameworks? Fundamente su respuesta
 - a. Laravel cuenta con el sistema de plantillas Blade el cual se usa para crear las vistas e interactuar con ellas de una manera más fácil.
 - b. Laravel cuenta con Eloquent, que se usa para interactuar con la base de datos de una manera más fácil
 - c. Comunidad y documentación, esto por que laravel es uno de los frameworks más usados a nivel mundial, cuenta con su propia documentación oficial y una comunidad de programadores inmensa
2. ¿Laravel tiene algún tipo de debilidades? En caso positivo indique cómo se pueden superar dichas debilidades.
 - a. Yo diría que una de sus desventajas es que sólo se puede usar a partir de la versión 5.4 de php y que su curva de aprendizaje es un poco difícil
3. ¿Conoce Eloquent ORM? En caso afirmativo, brinde una pequeña explicación cómo funciona.

ORM significa en inglés Object relational mapping, esto hace que cada tabla de la base de datos tiene un modelo correspondiente, y esto permite interactuar con la BD de una manera más fácil, se interactúa de una forma orientada a objetos.

Por ejemplo para crear una nuevo registro en Laravel haría lo siguiente

```
$book = New Book($request->all());  
$book->save();
```

Estoy creando una instancia del modelo Book y después accediendo al método save para guardarlo.

4. ¿Bajo qué arquitectura trabaja Laravel?
 - a. Trabaja bajo la arquitectura MVC, es decir Modelo - Vista - Controlador
5. ¿PHP en la versión 8 tiene ventajas sobre las versiones anteriores? En caso afirmativo, enumere al menos 4.

Una de las mayores ventajas de laravel 8 es que incluyó la compilación JIT y mejoró el rendimiento y el uso de memoria.

Se incluyeron los tipos de unión(Union types) y lo que hace es crear uniones de 2 o más datos según se vaya necesitando.

Los errores de tipo y la interfaz encadenable son nuevas características de php v8

6. Se debe elaborar una pequeña aplicación en Laravel 9, la cual debe permitir realizar lo siguiente:
- Realizar autenticación web y API (debe tener la vista de autenticación y lo requerido para una autenticación desde el API), tanto para login como para logout en ambas (WEB y API).
 - Tener al menos un mantenimiento web funcional con el CRUD completo de usuarios (como mínimo los datos que se requieren para dicho login).
 - Crear un mantenimiento que debe ser funcional con el CRUD completo, para el cual puede elegir una temática de libre elección (por ejemplo: libros, películas, etc.).
El tema elegido debe tener una base de datos pública que cuente con un API para que pueda obtener información extra, por ejemplo: imdb). Debe implementar dicho mantenimiento en el código de Laravel, no puede ser consumido desde la vista (por ejemplo, si eligió películas, se tendrá un campo de título de película con un botón de buscar, el cual se conectará con la base de datos pública y obtendrá los datos que esta ofrezca, al menos 3 campos).
Se deberá almacenar al menos 6 campos (3 propios y 3 de la base de datos pública).
Este mantenimiento solo podrá visualizar el listado de forma pública y permitir visualizar el detalle de la selección que realice el usuario. Las demás acciones solo las podrán visualizar y ejecutar un usuario autenticado.
 - El mantenimiento anterior, deberá tener a través del API la posibilidad de realizar el CRUD completo. En este caso, al igual que el web, la consulta del listado y el detalle será pública, el resto será solo para usuarios autenticados.

Dicho desarrollo debe tener al menos lo siguiente:

- Las validaciones, las cuales deben ser funcionales en WEB y API, mostrando los mensajes de error de forma clara al usuario, donde se detalle el error.
- Las Migraciones y los Seeders que considere necesarios.
- Los controladores, modelos, vistas, rutas y demás componentes que considere necesarios.
- La colección en Postman para poder consumir los servicios API.

El desarrollo debe ser funcional y cumplir con lo solicitado, pero no está limitado a esto.

Puede utilizar las dependencias, componentes, etc. Que considere necesarios.

Las vistas no requieren de un diseño específico, pero si se requiere una presentación adecuada.

Debe aplicar en todo momento las mejores prácticas.

Fecha límite de entrega: 13 de junio de 2022, 10:00 p.m.

Se asignarán puntos extra por pronta entrega.

Se debe enviar el proyecto comprimido, incluyendo este documento a la dirección electrónica: jelizondo@nidux.com.

El resultado será evaluado por nuestro equipo de desarrollo.