

## Preguntas

1. ¿Qué frameworks PHP has usado? ¿Qué destacarías de cada uno de ellos?

Durante los últimos años he utilizado Laravel tanto en la versión 4.2 como 5.6. Principalmente, destacaría la facilidad de realizar consultas gracias al ORM, su integración con vuejs y el hecho de que escribir test es muy sencillo.

2. ¿Qué es la inyección de dependencias? ¿Cuáles son los principales beneficios de usarla?

Es un patrón de diseño que pretende abstraer y reducir el acoplamiento al diseñar clases y objetos. Su objetivo es que el objeto sea lo más independiente posible de sus propiedades, facilitando la reutilización del código, su escalabilidad y mantenimiento.

3. Nombra algún patrón de diseño usado en aplicaciones web.

Estoy familiarizado con patrones MVC y en mis últimos desarrollos con react native he usado redux que cambia la manera en la que se gestiona el estado de una manera global.

4. ¿Qué frameworks Javascript has usado? ¿Qué destacarías de cada uno de ellos?

He usado vuejs para el desarrollo de aplicaciones web y react native para aplicaciones nativas. Lo que más me gusta de vuejs es que fue creado y diseñado desde un punto de vista del desarrollo web, la estructura de cada componente y los estilos, pudiendo elegir utilizar CSS o SASS de manera sencilla, me parece una maravilla.

También, me gusta mucho como gestiona vuejs los eventos para pasar información de hijos a padres y entre varios componentes.

De react native, destacaría el rendimiento que se obtiene, siendo prácticamente igual a una aplicación escrita putamente con lenguaje nativo y su curva de aprendizaje es muy suave, al igual que vuejs, en cuestión de horas puedes estar desarrollando aplicaciones medianamente complejas.

5. ¿Qué es un test unitario? ¿Qué diferencia hay con un test funcional?

Un test unitario se realiza sobre un componente o funcionalidad concreta, se suele testear con datos de pruebas y de forma automatizada. Existen librerías tipo jest para vuejs o react native y phpUnit en laravel. Para un test funcional se tiene de disponer de un “patrón” o una planificación

y que sigan cada uno de los pasos, para después evaluar dependiendo del tipo de prueba, si es válida o no.

6. ¿Qué es un mock?

Son objetos y funciones preconfigurados para simular el comportamiento específico en un test unitario. Se utilizan básicamente para comprobar si una llamada a una API, o un proceso específico funciona como se espera que se haga.

7. ¿Podrías explicar qué es SQL injection?

Consiste en un fallo de seguridad en una petición sql, básicamente un error de programación que permite realizar una query no esperada pudiendo afectar la integridad de los datos.

8. En una aplicación tenemos lo siguiente:

- a. Empresas que pueden tener clientes.
- b. Los clientes son otras empresas.
- c. Guardamos la misma información para empresas que para clientes.
- d. Las empresas tienen trabajadores
- e. Cada trabajador tiene un rol dentro de la empresa.

¿Qué tablas en una base de datos crearías para guardar la información y por qué? Sólo pedimos el nombre de las tablas, no los campos en cada una.

En un diseño de bases de datos relacional, una posible solución sería crear una tabla **Empresa** con una relación 1 a muchos, de forma que una empresa esté relacionada con muchas. Para Ello, hay que crear otra tabla (**es\_cliente**), que relacione una empresa con todas sus clientes.

Para los trabajadores, una tabla con la información que se quiera almacenar de ellos (sin tener en cuenta la normalización de la tabla) y una relación 1 a muchos con la tabla **Empresa**. En este caso, un trabajador solo puede pertenecer a una empresa por lo que el campo ID de la empresa pasaría a ser un atributo de la tabla **Trabajador**. Otro atributo de la tabla **Trabajador** sería el rol de la empresa.

9. ¿Cuál es el último libro técnico que has leído?

Actualmente estoy leyendo “Software Configuration Management Patterns” de Steve Berczuk donde habla sobre cómo se debe gestionar el trabajo en equipos IT, gestiones de repositorios, etc..

10. ¿Qué te gustaría aprender en este trabajo?

Me gustaría mejorar y ampliar mi experiencia, junto con el desarrollo en despliegues y automatizaciones de servicios clouds, CI/CD, etc...

11. ¿Podrías explicar brevemente qué es blockchain?

Es una tecnología que mantiene la integridad de los datos de forma descentralizada y segura, de manera que cada transacción es anotada y verificada por el resto de los nodos que forman parte.

12. ¿Conoces alguna blockchain en la que se puedan implementar contratos inteligentes?

No tengo mucho conocimiento sobre blockchain, pero tengo entendido que Ethereum puede ser usado para contratos inteligentes.