

**NOMBRE: ESTEBAN ANDRES URBINA BRAVO**  
**CARGO: DESARROLLADOR**  
**PROFESION: INGENIERO DE SISTEMAS**

## **Cuestionario**

**¿Qué es Laravel y cuáles son sus principales características?**

Laravel es un framework desarrollado bajo criterio de php para el desarrollo web, es uno de los mas populares y usados en aplicaciones como Facebook, el objetivo es simplificar el desarrollo en aplicaciones complejas. Algunas de sus características son: arquitectura MVC, Eloquent para la integración de bases de datos sql y permitir de manera segura, rutas y controladores lo que ayuda direccionar url de manera clara y a su vez tener un código limpio y seguro, plantillas blade que permite crear plantillas sencillas y claras, artisan que es una línea de comandos que simplifica tareas, en ella podemos levantar servidor, crear migraciones y levantar migraciones. Estas son las principales características que conozco de laravel

**Explique la diferencia entre Eloquent ORM y el Query Builder en Laravel.**

La principal diferencia entre Eloquent ORM y el Query Builder es que Eloquent ORM proporciona una capa de abstracción que hace que sea más fácil trabajar con los datos de la base de datos. En cambio, el Query Builder proporciona un acceso más directo a la base de datos, lo que puede ser útil para realizar consultas complejas, es decir utiliza métodos correspondientes a cláusulas SQL SELECT, FROM, WHERE, JOIN

**¿Qué es la inyección de dependencias y cómo se implementa en Laravel?**

Es un patrón de diseño que por medio sus objetos reciban dependencias de otro objeto, el objetivo es que el código sea mas modular y se pueda reutilizar. Se implementa a través del contenedor de inyección de dependencias. El contenedor de inyección de dependencias almacena todas las dependencias de la aplicación. Cuando se crea un nuevo objeto, el contenedor de inyección de dependencias se utiliza para resolver la dependencia

**¿Cuál es el propósito de los Middleware en una aplicación Laravel?**

Los middleware en Laravel son funciones que se ejecutan antes de que una solicitud llegue a un controlador. Estos son :

**Autenticación:** verificar si el usuario está autenticado.

**Autorización:** verificar si el usuario tiene permiso para realizar una acción.

**Validación:** validar los datos de la solicitud.

**Caching:** almacenar la respuesta de una solicitud en caché para mejorar el rendimiento.

**Registro:** registrar la solicitud para fines de auditoría.

**¿Qué es la autenticación de Passport en Laravel y cuándo podría ser útil?**

La autenticación de Passport en Laravel es un paquete que proporciona una implementación completa de OAuth 2.0 para aplicaciones web. OAuth 2.0 es un protocolo de autorización que permite a los usuarios autorizar aplicaciones de terceros a acceder a sus datos sin tener que compartir su contraseña.

Puede ser útil para :

**Aplicaciones API**→ es una buena opción para permitir que otras aplicaciones accedan a sus datos.

**Aplicaciones móviles** → puede ser útil para permitir que los usuarios inicien sesión con sus cuentas existentes.

## **Flutter**

**¿Qué es Flutter y cuáles son las ventajas clave de usarlo para el desarrollo de aplicaciones móviles?**

Es un marco de desarrollo de interfaz de usuario (UI) de código abierto creado por Google. Está diseñado para crear aplicaciones móviles, web y de escritorio. Utiliza un motor de renderizado propio, este permite crear aplicaciones de calidad y rendimiento nativo

**¿Explique la diferencia entre un StatefulWidget y un StatelessWidget en Flutter?**

la diferencia fundamental entre StatefulWidget y StatelessWidget en Flutter es la capacidad de cambiar su estado interno. StatefulWidget puede cambiar su estado y actualizar su interfaz de usuario en consecuencia, mientras que StatelessWidget representa una parte inmutable de la interfaz de usuario que no cambia su estado después de ser construida

**¿Qué es el "hot reload" en Flutter y cómo beneficia el proceso de desarrollo?**

El hot reload en Flutter es una función que permite a los desarrolladores ver los cambios en su código en tiempo real. Esto significa que se pueden realizar cambios en su código y ver los resultados en ejecución de la aplicación, como se beneficia:

- ➔ Agiliza el desarrollo
- ➔ Fácil depuración
- ➔ Productividad mejorada
- ➔ Exploración de diseño y estilos

## **Angular**

**¿Qué es Angular y cuáles son sus componentes clave?**

Es framework de JavaScript de código abierto creado por Google. Está diseñado para crear aplicaciones web de una sola página. Angular utiliza una arquitectura Modelo-Vista-Controlador (MVC) para separar la lógica de la aplicación de la presentación

Algunos de sus componentes claves son:

- ➔ Componentes -> bloques de construcción de la aplicación
- ➔ Módulos-> son contenedores para agrupar componentes
- ➔ Directivas-> son atributos especiales que se usan en los componentes (ngIf,ngFor)
- ➔ Enrutamiento->permiten navegar entre vistas
- ➔ Servicios->encapsulan la lógica y los datos que no están directamente relacionados con la interfaz de usuario

**Explique el concepto de "binding" en Angular y mencione los tipos de binding disponibles?**

Es el proceso de conectar la lógica de la aplicación con la presentación. El binding permite que los datos de la aplicación se muestren en la interfaz de usuario y que las acciones del usuario se envíen a la lógica de la aplicación.

- ➔ Property binding: Se utiliza para conectar el valor de una propiedad de un componente con el valor de una propiedad de un elemento HTML.
- ➔ Event binding: se utiliza para conectar un evento de un elemento HTML con un método de un componente.

- ➔ **Attribute binding:** se utiliza para establecer el valor de un atributo de un elemento HTML.
- ➔ **Interpolation:** La interpolación se utiliza para insertar datos en el contenido de un elemento HTML.

¿Qué son los servicios en Angular y cómo se comunican con los componentes?

los servicios son clases que proporcionan funcionalidad compartida a los componentes de una aplicación. Los servicios pueden utilizarse para realizar tareas como acceder a datos, realizar solicitudes HTTP y manejar eventos.

Los servicios se comunican con los componentes a través de la inyección de dependencias. La dependencia es un patrón de diseño que permite a los componentes depender de objetos externos.

¿Hable sobre las ventajas de Lazy Loading en una aplicación Angular y cuándo se debería utilizar?

Lazy Loading es una técnica que permite cargar los módulos de una aplicación Angular solo cuando son necesarios. Ayuda a mejorar el rendimiento de la aplicación, ya que los módulos innecesarios no se cargan en el inicio de la aplicación.

- ➔ **Mejor rendimiento**→mejora al cargar modulos necesarios en una aplicación compleja
- ➔ **Experiencia de usuario**→ ayuda a usuario a que la app se bloquee al cargar modulos grandes
- ➔ **Mayor flexibilidad**→ ayuda al desarrollador a cargar módulos dinámicamente

## SQL

¿Qué significa SQL y cuál es su función principal en el desarrollo de bases de datos?

Es un lenguaje de programación que se utiliza para acceder, manipular y administrar BD relacionales, sus funciones son :

- ➔ **Crear y modificar BD**
- ➔ **Crear y modificar tablas**
- ➔ **Insertar, actualizar y eliminar datos**
- ➔ **Consulta de datos**
- ➔ **Reporte de informes**

¿Explique la diferencia entre INNER JOIN y LEFT JOIN en una consulta SQL?

La principal diferencia entre INNER JOIN y LEFT JOIN es que INNER JOIN devuelve solo las filas que coinciden en ambas tablas, mientras que LEFT JOIN devuelve todas las filas de la tabla de la izquierda, incluso si no hay una coincidencia en la tabla de la derecha.

¿Qué es la normalización de bases de datos y por qué es importante?

La normalización se basa en un conjunto de reglas que se aplican a las tablas de una base de datos. Estas reglas tienen como objetivo eliminar la redundancia de datos, garantizar la integridad referencial y simplificar el diseño de la base de datos. Además, es importante porque mejora el rendimiento de los datos evitando el rendimiento en consultas, mejora la integridad evitando inconsistencias en los datos, fácil mantenimiento al reducir el número de tablas y columnas innecesarias

¿Qué son las transacciones en SQL y cuál es su propósito en una base de datos?

En SQL, una transacción es un conjunto de operaciones de datos que se ejecutan como una sola unidad. Las transacciones proporcionan un mecanismo para garantizar la integridad de los datos en una base de datos.

El propósito de las transacciones en una base de datos es garantizar que todas las operaciones de datos se realicen correctamente o que ninguna se realice. Esto se conoce como el principio de atomicidad

¿Cuál es la diferencia entre GROUP BY y HAVING en una consulta SQL y cómo se utilizan?

La principal diferencia entre GROUP BY y HAVING es que GROUP BY se utiliza para agrupar los datos, mientras que HAVING se utiliza para filtrar los grupos de datos

GROUP BY se utiliza para agrupar los datos de una consulta según los valores de una o más columnas

```
Select region, COUNT(*) AS total_ven  
From ventas  
Group by region
```

HAVING se utiliza para aplicar filtros a los grupos de datos que se han creado con la cláusula GROUP BY. Por ejemplo, quieres encontrar las regiones donde el total de ventas sea mayor o igual a \$10,000

```
Select region, COUNT(*) AS total_ven  
From ventas  
Group by region  
HAVING SUM(monto) >= 10000;
```