

1. Las malas prácticas de programación que en su criterio son evidenciadas en el código
 - a. Uso de magic string
 - b. Toda la lógica de negocio está fuera de los objetos de objetos de negocio (tell don't ask).

- c. El controlador no cumple con su responsabilidad
 - d. No existe ningún servicio de aplicación o caso de uso
 - e. Aunque activerecord puede denominarse un anti patrón en laravel se pueden aplicar ciertas convenciones, ej: si un modelo activerecord es responsable de mantener su estado, pasele mensajes para que cumpla su responsabilidad

//mal

```
$persona = Persona::find(1);
```

```
$persona->equipoFutbolFavorito = 'Barcelona FC';
```

```
$persona->save();
```

//mejor

```
$persona->miEquipoFavoritoSera('Barcelona FC');
```

- f. alto acoplamiento del dominio de negocio con infraestructura

2. ¿Cómo su refactorización supera las malas prácticas de programación?

- a. Desacoplando dominio de negocio de infraestructura
 - b. evitando tell don't ask
 - c. quitando la lógica de negocio del controlador
 - d. respetando active record
 - e. mudando la lógica de negocio a los objetos de dominio

3. ¿En qué consiste el principio de responsabilidad única? ¿Cuál es su propósito?

Principio de responsabilidad S ((SOLID) consiste en darle una unica razon a un objeto para modificarse, de esta manera reducimos el acoplamiento y escribimos codigo facil de mantener, reutilizar y modificar o escalar.

4. ¿Qué características tiene según tu opinión “buen” código o código limpio?

Partiendo de que desarrollar codigo orientado a objetos, trata sobre aprender de un dominio de problema, para luego ese aprendizaje proyectarlo usando clases y objetos, mientras más se asemeje aprendizaje a la proyección(clases/objetos) más limpio será el código.

Es fácil de mantener, si todo tiene una única responsabilidad, todo es más fácil!

Es Robusto: si todo tiene una única responsabilidad, y necesitamos modificar algo, modificamos solo ese algo.

Es fácil de modificar, si todo objeto tiene una única responsabilidad, podemos intercambiarlos, usando otros objetos que cumplan dicha responsabilidad.