UNS – DCIC

Arquitectura y Diseño de Sistemas

2019

Trabajo Práctico 2: Clean Code

- Para los incisos 1 y 2, evaluar el código fuente en el repositorio: https://github.com/elagarrigue/AyDS-TP-Clean-Code.
- Listar los smells de código no limpio. Hacer refactoring para que cumplan con los principios SOLID y Clean Code.

1. Crear url de servicio web: El objetivo de la porción de código es crear la url para la llamada de un servicio de creación de usuario. Utilizar el test implementado para referencia y pruebas.
2. Datos remotos y locales: El objetivo de la porción de código es obtener información de discos musicales a través de un servicio remoto, y guardarlos localmente. Los datos se piden al servicio sólo si no existen localmente, o si la copia local es demasiada antigua. Utilizar el test implementado para referencia y pruebas.
3. https://github.com/elagarrigue/AyDS-TP-Clean-Code-Ej3

   La aplicación "Sapena" realiza el cálculo del costo de estacionar en una playa de estacionamiento. El precio depende del tiempo que estuvo el vehículo estacionado y de las tarifas vigentes, que son por fracción de tiempo. Además, los días de semana se realiza un descuento del 10 %. Por ejemplo, dada la tarifa:

   - 15 minutos - $20
   - 1 hora - $60
   - 12 horas - $600
   - 24 horas - $800

   El costo de una estadía de 2hs 10min en un día de semana sería de $126, y de $140 en el fin de semana.

   Analizar el código fuente y resolver los siguentes puntos.

   a) Enumerar los principios SOLID que no se cumplen y explicar cómo se solucionarían. Tener en cuenta que en el futuro se van a agregar descuentos a algunas tarifas (por ejemplo, 10 % de descuento a jubilados para las fracciones de 15 minutos). Los descuentos no serán acumulativos con el descuento de día de semana.

   b) Hacer refactor del paquete `rate` para que cumpla con los principios SOLID y clean code.