Plantilla auditoría TP Final

Objetivo general

El objetivo general de este trabajo de auditoría es que los alumnos desarrollen razonamiento crítico frente a oportunidades de mejora, comprendan la estructura actual de un sistema y evalúen los costos de incorporar cambios en un proyecto real de software. Para esto, deberán aplicar los contenidos vistos en clase (Patrones de diseño, GRASP y principios SOLID).

Para la creación de este documento, los equipos de trabajo pueden optar por una API desarrollada por ellos mismos en la materia de Programación / Laboratorio 3, o podrán elegir un proyecto a elección.

Conformación de equipos

Los alumnos deberán formar grupos de trabajo de no más de cuatro miembros para el equipo. Cada miembro del equipo deberá tener participación demostrable en el documento, ya sea en secciones de análisis, propuestas o redacción. Al mismo tiempo, se deben defender los contenidos frente a los docentes de la materia.

Actividades a desarrollar

Para la aprobación de este trabajo, los grupos de trabajo deberán realizar las siguientes actividades:

1. Breve descripción del dominio

Descripción general del alcance del proyecto, describiendo brevemente los requerimientos generales planteados.

2. Composición del equipo

Listado de los miembros participantes en el proyecto y si participaron en la autoría del proyecto en cuestión.

3. Detalle de arquitectura

En este apartado se debe describir la arquitectura seleccionada para la confección del proyecto. Explicando el lenguaje utilizado, frameworks utilizados, repositorios y arquitectura Web.

4. Estado actual del código

Esta sección detalla el estado actual del código creado previamente. Se deberá detectar los bloques de código donde haya lugar para la incorporación de buenas prácticas de código y mejoren la calidad del mismo.

Se busca que el equipo de trabajo detecte bloques de código donde por ejemplo:

i. Se encuentran clases saturadas de responsabilidades.
ii. Grandes jerarquías de clases definidas donde existan diversas subclases que realicen diversas funciones o los atributos sean muy diferentes.
iii. Alto acoplamiento a servicios/módulos/controllers, etc.
iv. Oportunidades de implementación de interfaces.
v. Oportunidades de implementación de patrones de diseño.

Cada problema o mejora deberá ser planteada utilizando la siguiente estructura:

- **Módulo / paquete**: Describir el scope general donde se detectó la oportunidad de mejora.
- **Descripción del dominio**: Describir cómo el dominio afecta a este módulo.
- **Solución a implementar**: Detallar el patrón seleccionado o la modificación en código necesaria.
- **Ventajas de la implementación**
- **Posibles desventajas**: Detallar, por ejemplo, ¿es viable la implementación del cambio?, ¿afectará demasiado a servicios o clases relacionadas?, ¿es tolerante a futuros cambios?

Para la aprobación de esta sección, se deberán plantear al menos cinco propuestas de mejora.

5. Diagramas UML

Se deberá incorporar a la exposición oral una presentación que acompañe la explicación de cada diagrama indicado a continuación:

- 1 Diagrama de caso de uso
- 1 Diagrama de secuencia
- 1 Diagrama de estado
- 1 Diagrama de actividades
- 1 Diseño de condiciones de prueba aplicando técnicas de caja negra.

Revisión de TP: 4 de noviembre.
Fecha límite de entrega de documentación: 10 de noviembre - 22:00hs.
Fecha de exposición: 11 de noviembre.