**PRACTIA DOS**

La Cabro-Hada

**Fecha de entrega:** Examen 1er. Cuatrimestre:

1. **Requisitos:**

Para realizar esta práctica es necesario conocer los factores de calidad del software

**Factores de calidad del software**:

1. **Corrección:**

Es la capacidad para realizar con exactitud sus tareas, tal y como se definen en las especificaciones.

1. **Robustez:**

Es la capacidad de los sistemas de software de reaccionar apropiadamente ante condiciones excepcionales.

1. **Extensibilidad:**

Es la facilidad de adaptar los productos de software a los cambios de especificación.

1. **Reutilización:**

Es la capacidad de los elementos de software de servir para la construcción de muchas aplicaciones diferentes.

1. **Compatibilidad:**

Es la facilidad de combinar unos elementos de software con otros.

1. **Eficiencia**

Es la capacidad de un sistema software para exigir la menor cantidad posible de recursos hardware.

1. **Portabilidad (transportabilidad)**

Es la facilidad de transferir los productos software a diferentes entornos hardware y software.

1. **Facilidad de uso:**

Es la facilidad con la cual personas con diferentes formaciones y aptitudes pueden aprender a usar los productos software y aplicarlos a la resolución de problemas.

1. **Integridad:**

Es la capacidad de un sistema para protegerse de elementos que no tengan derecho de acceso.

1. **Verificabilidad:**

Que la aplicación sea fácilmente verificable.

1. **Mantenibilidad:**

La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse.

1. **Legibilidad:**

Se refieres al modo de en qué se estructura la información.

1. **Objetivos**

**General**:

El objetivo de la pratica es comprobar lo costoso que es comprender y mantener un código sin ningún comentario y sin variables autodocumentadas. Por eso se pide que se modifique el código para que cumpla los factores de calidad del sofware, en esta práctica se pide en particular dos factores:

-**Mantenibilidad**: La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las existentes, depurar errores, o mejorar el rendimiento.

-**Legibilidad**: La legibilidad se refieres al modo de en que se estructura la información (flujo de datos). En otras palabras, que esté claramente documentado, sangrado, sin errores, y que facilite su rápido entendimiento.

**Particulares**:

Crear una batería de pruebas que testeen que el código funcione correctamente.

1. **Resumen:**

En esta práctica se pide que se modifique un código hecho, de mala manera, para que cumpla los factores de calidad del software. Para eso contamos con algunas herramientas como el PMD o el CheckStyle que comprueban que nuestra aplicación cumpla una serie de reglas. También se pide que se haga una batería de pruebas para comprobar que el código funcione debidamente.

1. **Abstract**
2. **Observaciones**

La practica requiere mucho tiempo para poder entender que es lo que hace cada parte del código ya que esta escrito de una manera antinatural y sin ninguna legibilidad, además el nombre de las variables no ayudan en nada, y que se halla modificado algunas palabras insertando espacios entre medias solo provoca un estado de exasperación hacia la persona que lo halla hecho.

1. **Sugerencias**

Pocas sugerencias se pueden hacer, ya que el objetivo de la práctica es exasperar al alumno, y eso lo consigue perfectamente.

Como única sugerencia serie que se adjuntara el fichero que usa la práctica.

1. **Introducción**

Dado un codigo aportado por el profesor, el cual no contiene ningun factor de calidad del software, y en caso de contenerlo, es en muy poca extension. Se pretende acondicionar lo maximo posible este codigo, aplicandole los factores de calidad del software para que pueda ser llevadero y comprensible para el proximo usuario que tenga que mantenerlo o ampliarlo.

1. **Objetivos que se alcanzarán**

* Comprobar por nosotros mismos lo difícil y molesto que es mantener un código sin legibilidad.
* Aprender métodos y herramientas que se utilizan para aumentar la legibilidad y mantenibilidad de un código.

1. **Relación con la docencia cursada**

* Esta práctica pone a prueba nuestros conocimientos de factores de calidad del software.
* También se tiene que conocer las formas que utilizamos para aumentar la legibilidad de un código.

1. **Viabilidad**

Esta práctica resulta viable dado que su envergadura y complejidad es menor, si bien todos los conceptos y conocimientos necesarios para realizarla eran nuevos y ha sido necesario que comprenderlos y conseguir utilizarlos correctamente.

1. **Estado del arte y fundamentación teórica**
2. **Propuestas de solución**:
   1. Primera solución:

Se ha procedido primero a analizar el código e intentar conocer lo que este hace, para así poder cambiar poco a poco el nombre de las variables y métodos por nombres mas significativos.

Según se ha ido comprendiendo el código, se ha podido dividir este, en clases mas concretas, con sus respectivos métodos y atributos.

El resultado final contiene:

- Un código sangrado, con mayor legibilidad que el anterior.

- Se han reutilizado clases de practicas anteriores para esta.

- Han sido añadidos comentarios.

- Han sido añadidos paquetes y clases nuevas.

- Aún quedan variables sin renombrar, puesto que no se sabe su uso.

* 1. Segunda solución:
* **Desestimaciones**
* **Solución escogida**

1. **Conclusión**

Al realizar esta practica hemos comprendido mejor la importancia de los factores de calidad del software, y el porque es necesario llevarlos a cabo correctamente.

También se ha visto lo frustrante que es llevar a cabo el mantenimiento de un código con una baja legibilidad y porque es básico que se lleven acabo sencillos pasos como son el sangrado o los comentarios.

1. **Líneas Futuras**

De cara al futuro se tendrá muy en cuenta los factores de calidad del software, y la importancia de ellos, así como la gran importancia que tiene legibilidad del código.

1. **Bibliografía**

**Estructura del documento actual:**

* Transparencias sobre la estructura de las prácticas, del profesor Francisco Javier Crespo.

**Programación orientada a objetos:**

* http://www.elguille.info/colabora/NET2005/Percynet\_ConstruyendoSoftCalidad.htm