Programacion III

Alejandro León Marín

October 11, 2024

Contents

1 Elementos esenciales del desarrollo de aplicaciones

El estándar ISO/IEC 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para un producto de software. Este estándar es una simplificación del Modelo de McCall (Losavio et al., 2003), e identifica siete características básicas de calidad que pueden estar presentes en cualquier producto de software. Estas características son:

- **Funcionalidad:** Conjunto de atributos que relacionan la existencia de un conjunto de funciones con sus propiedades especificadas. Las funciones satisfacen necesidades especificadas o implícitas.

  Adecuación: Atributos que determinan si el conjunto de funciones son apropiadas para las tareas especificadas.

  Exactitud: Atributos que determinan que los efectos sean los correctos o los esperados.

  Interoperabilidad: Atributos que miden la habilidad de interactuar con sistemas especificados.

  Seguridad: Atributos que miden la habilidad para prevenir accesos no autorizados, ya sea accidentales o deliberados, tanto a programas como a datos.

- **Fiabilidad:**Conjunto de atributos que se relacionan con la capacidad del software de mantener su nivel de performance bajo las condiciones establecidas por un período de tiempo.

  Madurez: Atributos que se relacionan con la frecuencia de fallas por defectos en el software.

  Tolerancia a fallos: Atributos que miden la habilidad de mantener el nivel especificado de performance en caso de fallas del software.

  Recuperabilidad: Atributos que miden la capacidad de reestablecer el nivel de performance y recuperar datos en caso de falla, y el tiempo y esfuerzo necesario para ello.

  Cumplimiento: Atributos que hacen que el software se adhiera a estándares relacionados con la aplicación, y convenciones o regulaciones legales.

- **Usabilidad:** Conjunto de atributos que se relacionan con el esfuerzo necesario para usar, y en la evaluación individual de tal uso, por parte de un conjunto especificado o implícito de usuarios.

  Entendimiento: Atributos que miden el esfuerzo del usuario en reconocer el concepto lógico del software y su aplicabilidad.

  Aprendizaje: Atributos que miden el esfuerzo del usuario en aprender la aplicación (control, operación, entrada, salida).

Operabilidad: Atributos que miden el esfuerzo del usuario al operar y controlar el sistema.

- **Eficiencia:** Conjunto de atributos que se relacionan con el nivel de performance del software y la cantidad de recursos usados, bajo las condiciones establecidas.

  Comportamiento en tiempo: Atributos que miden la respuesta y tiempos de procesamiento de las funciones.

  Comportamiento en recursos: Atributos que miden la cantidad de recursos usados y la duración de tal uso en la ejecución de las funciones.

- **Mantenibilidad:** Conjunto de atributos que se relacionan con el esfuerzo en realizar modificaciones.

  Facilidad de análisis: Atributos que miden el esfuerzo necesario para el diagnóstico de deficiencias o causas de fallas, o para identificación de las partes que deben ser modificadas.

  Facilidad para el cambio: Atributos que miden el esfuerzo necesario para realizar modificaciones, remoción de fallas o cambios en el contexto.

  Estabilidad: Atributos que se relacionan con el riesgo de efectos no esperados en las modificaciones.

  Facilidad de pruebas: Atributos que miden el esfuerzo necesario para validar el software modificado.

- **Portabilidad:** Conjunto de atributos que se relacionan con la habilidad del software para ser transferido de un ambiente a otro.

  Adaptabilidad: Atributos que miden la oportunidad de adaptación a diferentes ambientes sin aplicar otras acciones que no sean las provistas para el propósito del software.

  Capacidad de instalación: Atributos que miden el esfuerzo necesario para instalar el software en el ambiente especificado.

  Conformidad: Atributos que miden si el software se adhiere a estándares o convenciones relacionados con portabilidad.

  Reemplazo: Atributos que se relacionan con la oportunidad y esfuerzo de usar el software en lugar de otro software en su ambiente.

- **Calidad de uso:** Conjunto de atributos relacionados con la aceptación por parte del usuario final y Seguridad.

  Eficacia: Capacidad de ayudar al usuario a realizar sus objetivos con exactitud y completitud.

  Productividad: Atributos relacionados con el rendimiento en las tareas cotidiana realizadas por el usuario final.

Satisfacción: Capacidad de satisfacer un usuario en un dado contexto de uso.

Seguridad: Capacidad de lograr aceptables niveles de riesgo para las personas, el ambiente de trabajo, y la actividad, en un dado contexto de uso.

2 Interfaces

2.1 Interfaces Funcionales

Las interfaces funcionales permiten pasar funciones como argumentos a métodos, algo que ya existía en Java (por ejemplo, con la interfaz Comparator), pero que fue ampliado en Java 8 con nuevas interfaces y aplicaciones. Estas interfaces se utilizan para definir comportamientos específicos. mediante métodos abstractos únicos, que pueden implementarse de manera más simple y concisa con expresiones lambda o referencias a métodos.

2.1.1 Expresiones Lambda

Una de las principales mejoras de Java 8 es la introducción de las lambda expresiones, que permiten una forma más compacta y legible de definir el comportamiento funcional. Las lambda expresiones se utilizan para simplificar la definición de funciones en lugar de crear clases o instancias anónimas.

2.1.2 Interfaces Funcionales Comunes

- **Predicate:** Evalúa una condición sobre un objeto de tipo T y devuelve un valor booleano. Ejemplo: determinar si un vuelo está completo.

```
Predicate<Vuelo> vueloCompleto =
x-> x.getNumPasajeros().equals(x.getNumPlazas());
```

2.2 Interfaces Comparator y Comparable

2.3 Metodos default en Interfaces

3 Java versions

4 Arquitectura de aplicaciones

5 Web Service

6 Bases de datos