



DECISIONES EN ESCENARIOS

ANEXO I: CRITERIOS DE EVALUACIÓN

Especificación de Criterios para la Evaluación de Trabajos Finales de Software

Los presentes criterios han sido definidos basados en las especificaciones del estándar internacional para evaluación de software ISO 9126. El mismo, en su modelo de calidad establecido en la primera parte clasifica la calidad de software en un conjunto estructurado de características y subcaracterísticas. Así, cada clasificación agrupa características asociadas a un aspecto en particular de la calidad del producto (por ejemplo, una se centra en las funcionalidades proporcionadas mientras que otra se centra en la eficiencia en el uso de recursos). Se deben definir para las subcaracterísticas un conjunto de atributos (no los define el modelo) cuya presencia debe ser verificable y medible en el software.

Para los propósitos de la materia, se establecen estas clasificaciones como los aspectos fundamentales a ser evaluados de forma independiente, generándose a partir de los puntajes obtenidos una nota general que será la correspondiente al trabajo final. Cabe destacar que por las características del trabajo, el contexto en que se realiza y los objetivos de la cátedra, no todos los aspectos tendrán el mismo peso en la nota final, ni serán evaluados con la misma minuciosidad.

Características a Ser Evaluadas

Funcionalidad: Agrupa a un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones (que satisfacen los requerimientos) y sus propiedades específicas. Las subcaracterísticas correspondientes son: idoneidad, exactitud, interoperabilidad, seguridad, cumplimiento de normas y conformidad.

Esta característica hace referencia fundamentalmente al conjunto de prestaciones que proporciona el software presentado, tanto en requerimientos funcionales básicos (modelos cuantitativos que utiliza, reportes que genera, información proporcionada, gráficos, parámetros que puede modificar el usuario, exactitud de los resultados, etc) como también en requerimientos de infraestructura (seguridad, capacidad de integrarse o intercambiar información con otras aplicaciones). Es importante aclarar que en esta característica se evalúa QUE hace el software, y no como lo hace o como debe usarse.

Fiabilidad: Agrupa a un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestaciones bajo condiciones establecidas durante un período de tiempo definido. Las subcaracterísticas correspondientes son: madurez, recuperación, tolerancia a fallos y conformidad.

Esta característica hace referencia fundamentalmente a la capacidad del software de mantener un funcionamiento aceptable a lo largo del tiempo, y a su capacidad de mantener su servicio frente a fallos. A los propósitos de la evaluación, se evaluará en esta característica aquellos aspectos que tengan que ver con el correcto funcionamiento de la aplicación ante ingreso de datos incorrectos, inconsistentes, inválidos, así como la aparición de errores propios de la aplicación durante la presentación.

Usabilidad: Agrupa a un conjunto de atributos relacionados con el esfuerzo necesitado para el uso, y en la valoración individual de tal uso, por un conjunto de usuarios establecido o implicado. Las subcaracterísticas correspondientes son: aprendizaje, comprensión, operatividad, atraktividad y conformidad.

Esta característica hace referencia fundamentalmente a los aspectos provistos para hacer "amigable" la aplicación al usuario. Se evalúan todos los aspectos referidos a facilidad de uso, tolerancia a errores del usuario (por ejemplo la carga errónea de un número), interactividad, diseño de interfaz, mecanismos para carga de datos, legibilidad y elegancia en la presentación



de resultados, etc. Además, se tendrá en cuenta la existencia de manuales de usuario y de instalación.

Eficiencia: Agrupa a un conjunto de atributos asociados a la relación entre el nivel de desempeño del software y la cantidad de recursos necesarios bajo condiciones establecidas. Las subcaracterísticas correspondientes son: comportamiento en el tiempo, comportamiento de recursos y conformidad.

Esta característica hace referencia fundamentalmente a la cantidad de memoria, procesador, software, y cualquier recurso adicional necesario para el correcto funcionamiento. Principalmente se tendrá en cuenta para la evaluación la velocidad de respuesta del software.

Mantenibilidad: Agrupa a un conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software. Las subcaracterísticas correspondientes son: estabilidad, facilidad de análisis, facilidad de cambio, facilidad de pruebas y conformidad.

Esta característica hace referencia fundamentalmente a atributos internos y estáticos del software relacionados a como está construido el mismo y a los artefactos de ingeniería relacionados. En este sentido, escapa a los objetivos de la cátedra y del trabajo práctico final la revisión de código o la capacidad de los estudiantes de modificar la aplicación. Si podrá tenerse en cuenta la existencia de artefactos de ingeniería (código fuente, diagramas, documentos de diseño, etc). También, podrá exigirse la explicación de los módulos y parte del código fuente así como la introducción de pequeños cambios, si a criterio del docente existiesen dudas sobre la participación de los estudiantes en el desarrollo del software presentado.

Portabilidad: Agrupa a un conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra. Las subcaracterísticas correspondientes son: capacidad de instalación, capacidad de reemplazo, adaptabilidad, coexistencia y conformidad.

Esta característica hace referencia fundamentalmente a la capacidad del software de ser utilizado en diferentes plataformas. Si bien se considera deseable, no será exigible dicha característica, debido a las múltiples razones por las que podría querer centrarse en una plataforma en particular que no deberían ser penalizadas (por ejemplo centrarse en un mercado en particular cuya plataforma sea incompatible con las otras, etc). Si podrá sumarse puntos extra de contar con la posibilidad de trabajar con múltiples plataformas.

Listado de Preguntas Clave

Para la evaluación del software, se propone el siguiente listado de preguntas clave que sirven de guía al evaluador para determinar la presencia de los atributos de calidad deseados en el trabajo evaluado. Dicha lista no es taxativa, por lo que podrán sumarse otras preguntas que considere relevantes el docente evaluador, o que resulten sobresalientes en el trabajo presentado y no puedan incluirse dentro de los enumerados. Así mismo, alguna pregunta podría no aplicar, en cuyo caso no debería ser tenida en cuenta.

Además, deberá tenerse presente que si a consideración del docente, se detectan no conformidades o fallos al evaluar el producto en un aspecto o grupo de aspectos considerados claves, el trabajo podrá ser reprobado. Ejemplos de esto son la aparición reiterada de fallas, o la mala implementación del modelo, falta de exactitud, o incapacidad para demostrar la participación en la construcción del producto o desconocimiento de los temas relacionados de la materia. En dichos casos, sin importar la cantidad de puntos obtenidos por las demás características, se considerará aplazo.



TABLA DE PREGUNTAS CLAVE PROPUESTAS POR CARACTERÍSTICA	
Característica	Preguntas clave propuestas:
Funcionalidad	Complejidad del modelo utilizado, o existencia de múltiples modelos.
	¿Se presenta un informe con los resultados?
	¿Se presentan conclusiones o síntesis de los resultados? (ej: alternativa propuesta, etc).
	¿Se presentan gráficos relativos al modelo o al resultado?
	¿Cuántos y cuáles parámetros pueden ser modificados?
	¿Existe alguna restricción respecto a la cantidad de datos, parámetros, variables, alternativas o criterios que puedan utilizarse?
	¿Genera archivos con los resultados que pueda ser abierto después por la aplicación?
	¿Genera archivos con el modelo que pueda ser abierto después por la aplicación?
	¿Genera archivos que puedan ser abiertos por otras aplicaciones como Excel o Acrobat pdf?
	¿Puede abrir archivos provistos por otras aplicaciones para cargar el modelo? (ej: una planilla de cálculo con los valores a utilizar para formular el modelo).
	¿Los resultados son exactos?
	¿Los valores son consistentes con los casos de prueba generados por el docente?
	¿Se identifican valores no consistentes con el modelo? (números negativos donde no corresponde, etc)
Fiabilidad	¿La aplicación presenta mecanismos de validación de los parámetros de entrada?
	¿La aplicación presenta mecanismos de validación de datos de entrada?
	¿La aplicación presenta mecanismos de validación de datos incompletos? (falta de ingreso de parámetros, etc.)
	¿La aplicación presenta mecanismos de validación de consistencia de los datos ingresados?
	¿La aplicación presenta mecanismos de recuperación ante mala utilización por parte del usuario? (por ejemplo seleccionar ver reporte sin haber resuelto el modelo antes).
	Durante la presentación del trabajo, ¿la aplicación presento errores, caídas, mal funcionamiento?
Usabilidad	¿El usuario requiere esfuerzos adicionales o conocimiento técnico para la ejecución del programa? (no se tiene en cuenta la dificultad de instalación)
	¿El usuario requiere esfuerzos adicionales para la utilización de las opciones?
	Se puede acceder a funciones que no tienen sentido en el contexto del uso (por ejemplo: mostrar reporte si no se ha resuelto un modelo antes).



	¿El programa “guía” al usuario para realizar cada una de las actividades?
	Los mecanismos provistos para el ingreso de datos: ¿facilitan la modificación en caso de error en el ingreso de los mismos?
	¿El usuario tiene la visualización de toda la información de forma sencilla, clara y completa?
	¿El usuario puede volver atrás y modificar parámetros y datos sin tener que ingresar todo nuevamente?
	¿La información está formateada de manera adecuada, para facilitar la visualización y la interpretación de la información?
	¿Se proveen mecanismos de alerta para indicar al usuario de errores, inconsistencias, advertencias, etc?
	¿La aplicación tiene un diseño cuidado y agradable para el ingreso de datos?
	¿La aplicación tiene un diseño cuidado y agradable para la presentación de resultados, informes, etc?
	¿La aplicación presenta mecanismos para facilitar el cambio de parámetros y la prueba de nuevos escenarios basados en el modelo ya definido?
	¿El diseño de la interfaz es consistente? (colores, tamaño de botones, imágenes, letras, formato de números, etc)
	¿Se proveen manuales de usuario? (impresos, digitales, en línea)
	¿Se proveen manuales de instalación?
	¿Se considera que el uso de la aplicación es natural e intuitiva?
	¿Se presentan títulos, etiquetas y otros elementos identificatorios de lo que se pide o lo que se presenta en pantalla?
Eficiencia	¿Cuánto tiempo toma iniciar la aplicación? (desde su ejecución hasta que se tiene acceso a las opciones del menú).
	¿Cuánto tiempo toma el cálculo de los resultados? (desde que se terminan de ingresar y se solicita procesar, hasta que se presenta la posibilidad de mostrar los resultados o estos son expuestos).
	¿Cuánto tiempo toma la generación de informes?
	¿La aplicación funciona en un equipamiento estándar o requiere un sistema con características (memoria, procesador) por encima del estándar comercial?
	¿Se necesita algún recurso adicional para su utilización? (existencia de un periférico, software propietario, servidores web desplegados localmente, etc)
Mantenibilidad	¿Se dispone del código fuente de la aplicación?
	¿Existen y son verificables otros artefactos de ingeniería propios de la aplicación fuera del código fuente? (documentos de diseño, diagramas, etc).
	¿Pueden demostrar los integrantes del grupo conocimiento cierto sobre la construcción de la aplicación?
	¿Están en condiciones los integrantes del grupo de realizar modificaciones en la aplicación?
	¿Se indican de alguna manera las dependencias con otras aplicaciones, librerías o productos en algún lugar? (por ejemplo, que el manual de



	instalación indique que es necesario tener Excel instalado, o determinada librería gráfica, o que el instalador indique de una alerta sobre la dependencia).
Portabilidad	¿El software puede ser accedido por el usuario desde distintas plataformas de sistema operativo?
	¿El software puede ser instalado sobre distintas plataformas de sistema operativo?
	¿El software puede ser accedido por el usuario desde distintas plataformas de hardware?
	¿El software puede ser instalado sobre distintas plataformas de hardware?
	¿Qué características tiene la plataforma estándar propuesta para el uso por los integrantes?