



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ARQUITECTURA DE SOFTWARE



Integrantes: Jenniffer Baque 6765

Andrés Alcoser 6760

Curso: 7mo 1

Fecha: 19/10/2022

Atributos de calidad - Tipos de relaciones

Preparar una tabla en donde se compare los distintos pares de atributos (dos primeras columnas), su tipo de relación (positiva o negativa) y justificación de la relación.

Atributo 1	Atributo 2	Relación	Justificación
Disponibilidad	Fiabilidad	+	Si el sistema tiene una buena disponibilidad refiriéndose al control de acceso y manipulación a ciertos datos sin que estos se pierdan o borren se relaciona directamente con la fiabilidad ya que es un sistema construido eficientemente y cumple con sus requerimientos.
Eficiencia	Flexibilidad	-	Como arquitectos si tomamos como prioridad la eficiencia sería algo contraproducente relacionarlo con la flexibilidad ya que el sistema opera ofreciendo varios usos y puede fallar a la petición de los usuarios.
Eficiencia	Interoperabilidad	-	El sistema debe ser eficiente, pero al momento de introducir la capacidad de interoperabilidad pierde eficiencia ya que hay mucho mas trafico de datos y puede que surjan ciertos errores.

Eficiencia	Mantenibilidad	-	Como arquitecto si garantizo la eficiencia del producto influye negativamente con el mantenimiento ya que me estaría contradiciendo ya que garantizo su uso sin errores.
Eficiencia	Portabilidad	-	Tomamos la eficiencia como principal objetivo ya que a los usuarios no les interesa la portabilidad además que a mayor número de plataformas no se garantiza su eficiencia.
Eficiencia	Fiabilidad	-	Si el arquitecto software tiene como prioridad la eficiencia del producto sería un poco difícil garantizar su eficiencia a lo largo del tiempo en el que esta previsto funcionar.
Eficiencia	Testabilidad	-	Si la eficiencia del sistema es alta, seria contradictoria realizar pruebas para ver si cumple con los criterios establecidos.
Eficiencia	Usabilidad	-	Como arquitecto doy prioridad a la eficiencia, pero ponemos en riesgo la usabilidad ya que no se toma en cuenta el control del usuario con el sistema por que nos centramos en lo que el sistema debe y tiene que hacer.
Flexibilidad	Integridad	-	Como arquitecto software el objetivo principal del sistema es poder modificarlo para los usos que demanden los usuarios, pero puede ocurrir que en los cambios se puedan modificar inintencionalmente la información.
Flexibilidad	Mantenibilidad	+	El momento en que el sistema se modifique para un nuevo requerimiento se puede realizar su mantenibilidad y aumentar su monitorización y control de este.
Flexibilidad	Portabilidad	+	La facilidad que tiene un sistema de ser modificado según los usuarios lo requieran y además de su capacidad de adaptabilidad en distintas plataformas.
Flexibilidad	Fiabilidad	+	El sistema permite cambios según los requerimientos de usuario según este lo vaya solicitando en un largo tiempo.
Integridad	Eficiencia	-	Si el sistema no permite la modificación de su información puede en ocasiones perder eficiencia ya que puede haber sido construido para un equipo establecido.
Integridad	Interoperabilidad	-	Si el arquitecto como principal objetivo la integridad del sistema no se permitiría la interoperabilidad ya que al momento de intercambiar información los datos se ven afectados como por ejemplo se cambia su tipo de formato.

Integridad	Reutilización	-	Tomamos como objetivo principal la integridad del sistema, pero si se baso en otro sistema no se puede garantizar la integridad ya que puede existir alguna función que modifique los datos sin previo aviso.
Integridad	Testabilidad	-	Si el sistema tiene como objetivo la integridad asegurando que sus datos no se verán alterados seria un problema realizar pruebas continuamente ya que pueden burlar la seguridad y acceder a los datos y modificarlos.
Integridad	Usabilidad	-	Como arquitecto software mi prioridad es la integridad del sistema, pero si el sistema es fácil de manipular los usuarios podrían cambiar ciertas configuraciones afectando indirectamente a la integridad.
Interoperabilidad	Flexibilidad	+	El momento en que se comunican distintos sistemas compartiendo datos implicara nuevas modificaciones dependiendo del tipo de sistema con el cual queramos comunicarnos.
Interoperabilidad	Portabilidad	+	Si el sistema es interoperable eso quiere decir que su implementación ofrece maneras de intercambiar información con otros equipos sin ningún tipo de problema.
Mantenibilidad	Disponibilidad	+	Como arquitecto software la mantenibilidad de mi producto me permite modificarlo y agregar nuevas funcionalidades de acuerdo a los requisitos del usuario y esto conlleva a una buena disponibilidad ya que funcionara correctamente mientras el usuario lo necesite.
Mantenibilidad	Fiabilidad	+	Como arquitecto tomo la mantenibilidad como objetivo principal ya que podremos monitorear y observar los fallos que se produzcan en el sistema mientras funciona en un tiempo determinado.
Mantenibilidad	Testabilidad	+	Tomamos como prioridad la mantenibilidad del sistema y se aprovecha para realizar pruebas y ver si existen problemas y solucionarlos rápidamente.
Portabilidad	Mantenibilidad	-	Si la portabilidad de un software es alta, será un poco complejo someterlo a reparación y evolución, debido a su difícil conservación o restauración a un estado en el que pueda realizar las funciones requeridas.

Portabilidad	Reutilización	+	Las funcionalidades que se utilizaron para conseguir que un software sea portable, podrían beneficiar al arquitecto ya que estas se podrían reutilizar en la construcción de otro producto software.
Portabilidad	Testabilidad	+	Si en el software se prioriza la portabilidad, este garantizará que se han realizado diferentes test y pruebas lo que facilita el establecimiento de criterios de prueba que deben cumplirse.
Portabilidad	Usabilidad	-	Si en el producto se toma como prioridad la portabilidad, este podría no contar con la facilidad con la que un usuario puede aprender a operar el sistema, preparar entradas e interpretar salidas.
Fiabilidad	Testabilidad	+	Si un producto es fiable, implica la capacidad de que el sistema se desarrolló para que sea más fácil de verificar en su totalidad, ya que sus funciones se cumplen cada cierto tiempo y en esos intervalos se pueden realizar test o pruebas.
Fiabilidad	Usabilidad	+	Si el sistema cumple una determinada función durante un tiempo específico, significa que el arquitecto lo diseñó para que los usuarios cuenten con la capacidad de usarlos con facilidad.
Reutilización	Eficiencia	-	Si el arquitecto desarrolla un sistema basándose en otro, es posible que el código no podría adaptarse correctamente al producto a construir lo que no garantizará la eficiencia del mismo.
Reutilización	Flexibilidad	+	Cuando un software se construye con reutilización, podría tener muchos módulos los cuales se puedan modificar para que realizar otras acciones que no estaban previstas desde el inicio de la fase de desarrollo.
Reutilización	Fiabilidad	-	Cuando se utilizan módulos de otro producto en el que se está desarrollando, es probable que el software no sea fiable, ya que las funcionalidades utilizada no garantizarían que se use a lo largo del tiempo.
Reutilización	Testabilidad	+	Si el arquitecto construye un software con reutilización, podrían aparecer módulos complicados que dificultarán que el sistema se desarrolle con la capacidad de ser más fácil de verificar.
Testabilidad	Disponibilidad	+	Cuando el arquitecto de software se encarga de que el sistema sea más fácil de verificar, aumenta la disponibilidad ya que se asocia con fallos, y la testabilidad se

			apoya de ella para remover los defectos en el desarrollo y al terminar la fase, el producto dispondrá de disponibilidad.
Testabilidad	Flexibilidad	+	Si el producto es desarrollado para que se pueda probar en su completitud, presentaría beneficios para la flexibilidad pues será más fácil encontrar partes del sistema que deben cambiar la base de las actividades las cuales fueron implementadas desde el inicio.
Testabilidad	Usabilidad	+	Si el sistema se desarrolla de tal manera que sea más fácil de verificar, la usabilidad aumentará ya que el sistema contará con pruebas las mismas que garanticen que el usuario sabrá cómo gestionar el software.