Repaso

Drivers Arquitectónicos

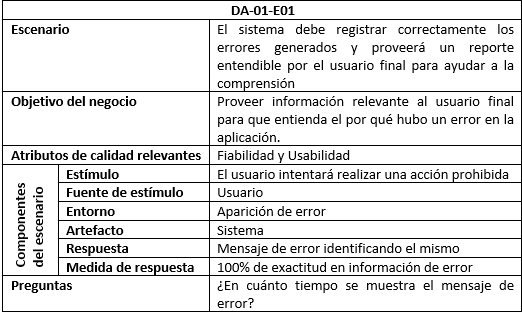
* Es el subconjunto de requisitos (funcionales, atributos de calidad, restricciones) que tienen una gran importancia para la arquitectura del sistema .
* No a todos los requisitos se le da la misma importancia al momento de elaborar la arquitectura de software.
* Debe enfocarse en los requisitos de mayor influencia respecto a la forma que tomaran estos elementos al momento de diseñar la arquitectura.

Atributos de calidad

1. Disponibilidad
   1. Probabilidad que tiene el sistema de estar operativo cuando se necesite
2. Interoperabilidad
   1. Es la capacidad de dos sistemas puedan intercambiar información a través de una interfaz
3. Modificabilidad
   1. Relacionada con los costos de los cambios
   2. Que se puede cambiar
4. Performance
   1. Relacionada al tiempo de ejecución de un proceso o evento que utilicen los usuarios
5. Seguridad
   1. Habilidad del sistema para resistir usos no autorizados y seguir brindando servicios a usuarios
6. Testeabilidad
   1. Facilidad que se presenta a un sistema para hacer testing
7. Usabilidad
   1. Facilidad con las que el usuario puede cumplir una tarea del sistema.

Escenario de Atributos de Calidad

|  |  |  |
| --- | --- | --- |
| Entradas | Fuente de estimulo | Es la entidad(humano ,sistema, etc) |
| Estimulo | Es la condición que se da para utilizar el sistema |
| Condiciones | Entorno | Condiciones en las que se encuentran el sistema al presentarse el estimulo |
| Artefacto | Componente que recibe el estimulo (puede ser sistema u otro) |
| Resultados | Respuesta | Define la acción que se realiza como respuesta al estimulo |
| Medida de respuesta | Métrica que define al estimulo |



Tácticas

Disponibilidad

* Detección
  + Ping
  + Hearthbeat
  + Exception
* Recuperación
  + Votación
  + Redundancia Activa
  + Redundancia Pasiva
  + De repuesto
  + Operación en la sombra
  + Resincronización de estado
  + Rollback
* Prevencion
  + Eliminacion del servicio
  + Transacciones
  + Monitor de procesos

Inteoperabilidad

* Descubrimiento de servicios. - Es usada por el sistema que necesita interoperar para descubrir los tiempos de ejecución
* Orquestacion.-Mecanismo de control para coordinar y administrar la secuencia de invocaciones entre los servicios
* Acondicionamiento de interfaces. -es una táctica que permite agregar o retirar capacidades a las interfaces.

Modificabilidad

Perfomance

Seguridad

Testeabilidad

Usabilidad

Preocupaciones de diseño

* Allocation Responsabilities.- Identificar las actividades mas importantes
  1. Funcionamiento básico del sistema
  2. Infraestructura
  3. Cumplimiento de atributos de calidad
* Cordination Model.-
  1. Identificar los elementos del sistema que deben y no deben coordinar
  2. Determinar las propiedades de la coordinación
  3. Seleccionar los mecanismos de coordinación
* Data model
  + Selecciona las mayores abstracciones de los datos,operaciones y propiedades
  + Determina la metadata que describen la información
  + Selecciona la categoría con la que se organizan los datos.
* Managment of resources
  + Balancear el uso de recursos compartidos
  + Determinar los recursos manejados
  + Determinar las estrategias cuando haya contención de recursos
  + Determinar los elementos a ser afectados cuando haya saturación de recursos.
* Mapping Among Arquitetural Elements
  + Mapeo de los elementos en diferentes tipos de estructuras de arquitectura
  + Mapeo de elementos de software en su entorno
* Binding time desicions
  + Introduce un rango de variación
  + La variación puede ser cincuscrita en las diferentes etapas del ciclo de la vida del software para diferentes entidades
  + Considera los costos de implementar una decisión y los costos de hacer una modificación después de que se tiene la implementación
* Choice of tecnology
  + Si la tecnología es seleccionada por otros, se convierte en una restricción
  + El arquitecto debe seleccionar un conjunto de tecnología para comptemplar los atributos de calidad

SOA

Es el nexo que une las metas de negocio con el sistema de software. Su papel es el de aportar flexibilidad, desde la automatización de las infraestructura y herramientas necesarias consiguiendo, al mismo tiempo, reducir los costes de integración.**SOA** se ocupa del diseño y desarrollo de sistemas distribuidos y es un potente aliado a la hora de llevar a cabo la gestión de grandes volúmenes de datos, datos en la nube y jerarquías de datos.

Caracterisitcas

* Aumenta la eficiencia en los procesos.
* Amortiza la inversión realizada en sistemas.
* Reduce costes de mantenimiento.
* Fomenta la innovación orientada al desarrollo de servicios.
* Simplifica el diseño, optimizando la capacidad de organización

SOUP REST

* Por otra parte, tenemos REST, el chico nuevo de la cuadra. REST ya tiene unos años, pero en realidad tiene poco que se le empezó a dar la importancia que hoy tiene. REST es una tecnología mucho más flexible que transporta datos por medio del protocolo HTTP, pero este permite utilizar los diversos métodos que proporciona HTTP para comunicarse, como lo son GET, POST, PUT, DELETE, PATCH y a la vez, utiliza los códigos de respuesta nativos de HTTP (404,200,204,409). REST es tan flexible que permite transmitir prácticamente *cualquier tipo de datos*, ya que el tipo de datos está definido por el Header Content-Type, lo que nos permite mandar, XML, JSON, Binarios (imágenes, documentos), Text, etc. que contrasta con SOAP que solo permite la transmisión de datos en formato XML. A pesar de la gran variedad de tipos de datos que podemos mandar con REST, la gran mayoría transmite en JSON por un motivo muy importante, *JSON es interpretado de forma natural por JavaScript*, lo que ha hecho que frameworks como Angular y React se aprovechen al máximo, pues pueden enviar peticiones directas al servidor por medio de AJAX y obtener los datos de una forma nativa. Los formularios de HTML pueden ser apuntados a los servicios REST sin ningún problema
* tra de las grandes ventajas que presenta JSON sobre SOAP es el *performance*, ya que los JSON son considerablemente más livianos en peso y mucho más rápido en su procesamiento. Pero como ya vimos, el performance tiene un costo, y es la robustez del mensaje como tal.

C4

C4 Context.-

* Permite ver el paronama general
* Alcance: Un solo sistema de software
* A quien va dirigido: todos

C4 Container

* Muestra la arquitectura del software y distribuye las responsabilidades
* Muestra la conexión y las opciones tecnologías
* Dirigido a : desarrolladores, personal de soporte

C4 Component

Amplia y descompone un contenedor y identifica sus principales componentes

Dirigido a: Desarrolladores y arquitectos

C4 Clase

Muestra como se implementa un componente

Utiliza diagramas uml

Dirigido: desarrolladores y arquitecto

Notación para documentar-vistas

Requisitos

Diseño

Evaluación

Implementación

La documentación sirve para:

Mejorar la información

Preservar la información

Guiar la generación de artefactos

Proveer un lenguaje común entre diversos interesados del sistema

Vistas

Representar un conjunto de elementos y relaciones de un tipo particular

Tipos de vistas

Vista lógica

Vista de Comportamiento

Vista física

Vista modular

Vista runtime

Vista despliegue

Vista modelo de datos

Documentacion

Views and beyond

Vistas Estructura

1. Primary presentation
2. Element catalog
3. Variability Guide
4. Arquitecture Background
5. Related Views

Beyond estructura

1. Hoja de documentación
2. Vision conjunto del sistema
3. Requisitos
4. Vistas
5. Mapeo o relación de las vistas
6. Análisis de arquitectura y racionalidad
7. Mapeo de requisitos de arquitectura
8. Glosario y lista de acronimos

Características de una arquitectura

