

Ejercicios Solemne 1

Marcelo A. Yáñez
1er semestre, 2023

Preguntas Teóricas

1. El gerente general de la empresa en que usted trabaja ha escuchado el término “Arquitectura de Software” y como él no es técnico le ha pedido que le explique en que consiste, pero sin utilizar terminología técnica. ¿Qué haría usted para responder a este requerimiento?

Respuesta:

La arquitectura de software se refiere a la forma en que un programa de computadora está diseñado y organizado. Es como el plano de una casa, donde se establecen los cimientos y la estructura para que todo funcione de manera eficiente y se puedan agregar características adicionales. Al igual que un arquitecto diseña una casa para que sea funcional y atractiva, un arquitecto de software diseña el programa para que sea fácil de entender, modificar y mantener.

2. Describa 4 competencias que debe tener el arquitecto de software.

Respuesta:

- Administración del riesgo.
- Excelentes habilidades de diseño.
- Interfaz entre el cliente y el equipo técnico.
- Puente de comunicación entre los equipos de desarrollo.

3. Explique cuales factores son utilizados para medir la escalabilidad de un sistema de software.

Respuesta:

- Carga: Cantidad de transacciones por segundo que puede soportar el sistema.
- Conexiones simultáneas
- Volumen de datos

4. Describa las diferencias que tienen entre si los siguientes requerimientos no funcionales:

- Mantenibilidad
- Soportabilidad
- Verificabilidad

Respuesta:

La mantenibilidad está enfocada en la facilidad de modificación del sistema, mientras que la soportabilidad se enfoca en poder diagnosticar y resolver problemas en el sistema en ejecución. Finalmente, la verificabilidad corresponde a la capacidad del sistema de comprobar su propio funcionamiento.

5. Explique qué se debe considerar al especificar una arquitectura de un sistema.

Respuesta:

Al especificar una arquitectura se deben considerar los siguientes factores:

- Requerimientos funcionales: Que es lo que debe hacer el sistema.
- Requerimientos no funcionales: Que atributos de calidad son requeridos.

- Ambiente operacional: Dónde se utilizará el sistema (Hardware y Software involucrados).
 - Restricciones: Las restricciones pueden ser tecnológicas, temporales, de personal, etc.
6. Analice cinco requerimientos no funcionales que podrían ser considerados al especificar una arquitectura para un sistema que deba desarrollar.

Respuesta:

- Escalabilidad: Algunas arquitecturas facilitan la escalabilidad del sistema (e.g. Cliente / Servidor) mientras que otras arquitecturas son más difíciles de escalar.
 - Mantenibilidad: La especificación de una arquitectura de software puede aumentar la mantenibilidad de un sistema. Debido a que si el sistema se diseña bien (y no es monolítico), se puede alterar o añadir componentes sin afectar el resto del sistema.
 - Performance: Algunas arquitecturas requieren mayor comunicación entre componentes que otras (e.g. SOA), lo que genera un deterioro en la performance en comparación a un sistema monolítico.
 - Seguridad: La arquitectura especificada puede facilitar que el sistema sea seguro (e.g. añadir una capa de seguridad en el modelo de capas) mientras que en otras (e.g. Cliente / Servidor) es más complicado.
 - Integrabilidad: Algunas arquitecturas facilitan el uso de sistemas ya construidos anteriormente (e.g. SOA), por lo que debe considerarse si se requiere interoperar con otros sistemas.
7. Indique las características de las siguientes arquitecturas genéricas y analice sus ventajas y desventajas:
- Modelo de Repositorio.
 - Cliente / Servidor.
 - Arquitectura Orientada a Servicios (SOA).

Respuesta:

- Modelo de Repositorio
Estructurado en base a una gran cantidad de datos que debe ser compartida. El almacenamiento de los datos puede ser centralizado o distribuido.
Ventajas:
 - Eficiente en la compartición de datos.
 - Administración centralizada de los datos.
 - Escalable y fácil de mantener.**Desventajas:**
 - Fuerza un modelo de datos.
 - La base de datos es un punto único de falla.
 - Dificulta la modificación del modelo de datos.
- Cliente / Servidor
Servidores que ofrecen servicios específicos y Clientes quienes requieren servicios, conectados por una red de comunicación. El modelo es parcialmente distribuido.
Ventajas:
 - Procesamiento distribuido.
 - Almacenamiento distribuido.
 - Fácil de escalar.**Desventajas:**
 - Datos no compartidos.
 - Baja performance.
 - No hay registro centralizado de servicios.

- Arquitectura Orientada a Servicios (SOA)

Ambiente de intergración basado en estándares, con servicios administrados centralizadamente. Evolución del modelo Cliente/Servidor.

Ventajas:

- Reutiliza sistemas existentes.
- Bajo nivel de acoplamiento.
- Escalable, mantenible e integrable.

Desventajas:

- Alta dependencia a estándares.
- Baja performance.

8. Explique en qué contexto y para que tipo de sistemas es recomendable utilizar el modelo de repositorio.

Respuesta:

El modelo de repositorio es recomendable cuando todos los componentes de un sistema tienen el mismo modelo de datos y se requiere administración centralizada de estos.

Por ejemplo, un tipo de sistema para el que sería recomendable este modelo es un e-commerce. El sistema necesita acceder a grandes volúmenes de datos tanto por el lado del cliente, como por el lado de la administración del negocio.

Caso 1

A usted le han solicitado el desarrollo urgente de un sistema, cuya primera versión, con el 30% de los requerimientos funcionales implementados, debe estar operativa en dos semanas. Luego, tiene 4 semanas para finalizar la implementación de los restantes.

En este contexto, analice tres requerimientos no funcionales que le pueden ayudar a cumplir con el plazo establecido y tres que lo retrasarían.

Solución

Requerimientos no funcionales que ayudan a cumplir con el plazo establecido:

- **Mantenibilidad:** Si el sistema es mantenible, es más fácil implementar el resto de los requerimientos funcionales.
- **Integrabilidad:** Dependiendo del contexto, permitir que el sistema funcione en conjunto con sistemas ya existentes puede disminuir la cantidad de trabajo que debe hacerse.
- **Escalabilidad:** Si el sistema es escalable, se pueden añadir nuevos requerimientos sin problemas y el sistema podrá crecer a medida que se implanta el resto de los requerimientos.

Requerimientos no funcionales que retrasarían el desarrollo:

- **Confiabilidad:** Para lograr confiabilidad se deben tener en cuenta muchos factores que pueden afectar el funcionamiento del sistema.
- **Verificabilidad:** La verificabilidad implica que el sistema pueda probarse a sí mismo, por lo que se requeriría de trabajo adicional para cada componente del sistema.
- **Seguridad:** Si se requiere un alto nivel de seguridad, puede ser necesario dedicar una cantidad significativa de tiempo para lograrlo. Esto puede retrasar el desarrollo del sistema.

Caso 2

Las empresas pertenecientes a un holding de empresas (del tipo de Cencosud, Falabella o Walmart) han desarrollado, independientemente unas de otras, sus propios sistemas de atención a clientes, mediante sitios web y sistemas internos. El directorio del holding ha planteado la necesidad de integrar en un solo portal los diversos sistemas de atención, de manera que los clientes tengan una experiencia única e integrada, independientemente de la empresa con la que estén interactuando.

Asimismo, requiere que los sistemas internos conversen entre sí, de manera de aprovechar la información que cada uno administra. Así, por ejemplo, en lugar de tener varios sistemas que manejen la cartera de clientes, se plantea tener uno solo, de manera tal que pueda ser accedido por los sistemas que lo requieran. Cada empresa ha gastado una gran cantidad de dinero en el desarrollo independiente de sus sistemas y se requiere no perder esa inversión hecha. En definitiva, no se está planteando hacer nuevos sistemas, sino que adaptar los actuales sistemas para que funcionen mancomunadamente.

Al respecto, se le pide lo siguiente:

- Explicar la arquitectura que escogería para responder a lo pedido
- Analizar las ventajas y desventajas de la arquitectura elegida
- Justificar tres requerimientos no funcionales que debieran estar contemplados en la solución propuesta

Solución

La arquitectura que escogería para responder a los requerimiento es SOA (Arquitectura Orientada a Servicios). La razón para elegir SOA por sobre otras arquitecturas es que se desea integrar distintos sistemas ya existentes, lo que se puede realizar a través de un bus de servicios.

Ventajas:

- Reutilización de los sistemas existentes.
- Bajo acoplamiento.
- Alta escalabilidad.

Desventajas:

- Dependencia a estándares.
- Baja performance.
- Alta complejidad, debido a la gran cantidad de servicios involucrados.

Requerimientos funcionales a contemplar en la solución:

- Seguridad: Debido a que el sistema manejará administrará una cartera de clientes, es necesario que este sea seguro ya que una vulnerabilidad en el sistema podría afectar gravemente a las empresas del holding.
- Integrabilidad: Es necesario que el sistema presente interoperabilidad entre los distintos subsistemas que lo compondrán para que estos se presenten como un sistema único para el usuario.
- Escalabilidad: Debido a que el sistema a implementar será usado por un holding de empresas este estará sometido a una gran carga, la cual puede variar bajo ciertas circunstancias (e.g. Falabella en un cyber day). Por lo tanto el sistema debe ser escalable para poder manejar estos aumentos de carga.

Caso 3

Se requiere desarrollar un sistema que permita ingresar y reportar la votación obtenida por cada candidato en cada una de las mesas receptoras de sufragios en todo el país. Las mesas receptoras están agrupadas en locales de votación a cargo de un jefe de local.

Entre otras funciones, el jefe de local es el encargado de recolectar las actas de votación de cada mesa receptora e ingresar los resultados al sistema. Para ello, dispone de un notebook con conexión a internet.

Se requiere, además, que el sistema genere un informe con los resultados disponibles hasta ese momento y consolidados por cada candidato, con una periodicidad preestablecida en el sistema. Este reporte debe considerar varios niveles de agrupación de los resultados, tales como, a nivel de la mesa receptora, del local de votación, de la comuna, del distrito (para los diputados), de la región senatorial (para los senadores) y, finalmente, a nivel país (para el presidente).

Considerando que las elecciones son en 23 días más, se pide lo siguiente:

- Analizar y justificar tres requerimientos no funcionales que el sistema debe satisfacer
- Analizar y justificar cuál de las arquitecturas de software genéricas va a usar para desarrollar el sistema
- Describir brevemente la funcionalidad de tres de los componentes que, a su juicio, son los más importantes del sistema a desarrollar.

Solución

Requerimientos no funcionales a satisfacer:

- Seguridad: Debido a que el sistema manejará datos electorales y los enviará por internet este debe ser seguro. Una vulnerabilidad en el sistema podría causar desde reportes incorrectos hasta fraude electoral.
- Verificabilidad: El sistema debe ser capaz de comprobar su propio funcionamiento debido a que será ejecutado en distinto hardware a lo largo de todo el país, y se debe asegurar que cada instancia funcione correctamente.
- Soportabilidad: En caso de alguna incidencia que perjudice el funcionamiento del sistema, deben existir herramientas de diagnóstico con el fin de poder solucionar dichos inconvenientes.

Arquitectura a utilizar:

Debido a que existirán múltiples instancias de un subsistema que enviarán datos y que se debe realizar un análisis de los datos como conjunto se usará el modelo de repositorio. Esta arquitectura es eficiente para compartir datos, y se tiene una administración centralizada de estos.

Componentes principales:

- Componente de seguridad (e.g. un login) que permita autenticar que la persona que intenta ingresar datos es el jefe del local de votación.
- Componente que reciba los datos del local de votación y los envíe hacia la base de datos. Debe utilizar un protocolo de comunicación seguro (e.g. TLSv1.2).
- Componente de análisis de datos que genere el informe con los resultados cada cierto tiempo.