

TEMA 5. INGENIERÍA DE SERVICIOS

Sistemas Distribuidos

Un sistema distribuido es un conjunto de ordenadores independientes que aparecen ante un usuario o una aplicación como un sistema único.

- **Transparencia:** Capacidad de aparecer como un sistema único frente a un usuario o una aplicación.
- **Apertura:** Capacidad para ofrecer servicios contruidos en base a unas reglas estandarizadas que definen la sintaxis de dichos servicios.
- **Escalabilidad:** Capacidad de crecimiento, sin pérdidas, en cuanto a tamaño.

Arquitectura Orientado a Servicios (SOA)

Estilo de arquitectura software basada en el uso de servicios que exponen una funcionalidad de negocio para el desarrollo de aplicaciones distribuidas.

- **Aplicación distribuida:** Sistema único aunque se ejecute en distintos sistemas.
- **Servicio:** Posee una interfaz procesable y una implementación reemplazable.
- **Negocio:** El nivel de granularidad de un servicio es mayor que en una clase.
- **Estilo arquitectónico:** Forma de exponer e invocar servicios.

Esta formado por las siguientes entidades:

- **Proveedor:** Dueño de los servicios.
- **Registro:** Encargado de gestionar los servicios.
- **Consumidor:** Busca servicios y los invoca.

Servicios

Los servicios web son cualquier sistema accesible desde la web. Están basados en un enfoque SOA. Pueden tener dos enfoques de implementación:

- **Llamada a procedimiento:** Expone un procedimiento remoto invocado por un consumidor y que recibe una respuesta tras su ejecución.
- **Orientado a recursos:** Expone una interfaz para la gestión de los recursos.

Hay dos tipos de servicios:

- **Servicios Web SOAP:** También llamados servicios web. La interacción entre entidades se realiza mediante XML.
 - A. **SOAP:** Define un mecanismo para el paso de instrucciones y parámetros entre cliente y servidor. Basado en HTTP y XML.
 - B. **WSDL:** Lenguaje que define las interacciones entre cliente y proveedor de Servicios Web SOAP.
 - C. **UDDI:** Interfaz y estructura de datos que permite a los proveedores registrar el lenguaje WSDL.

Existen dos tipos de desarrollo de servicios SOAP:

1. **Top-Down:** Primero se define la interfaz y luego la implementación del sistema.
 2. **Bottom-Up:** Primero se define la implementación del sistema y luego la interfaz.
- **Servicios Web REST:** Basado en el protocolo HTTP, solo es necesario definir los permisos de los clientes y la presentación de los recursos. Se utiliza WADL.

XML

Lenguaje para almacenamiento e intercambio de información en un formato textual. Se basa en el uso de etiquetas o marcas, que pueden tener atributos y que pueden incluir contenido, y que se pueden anidar para crear una estructura jerárquica en forma de árbol.

XSD

Permite describir la estructura de un documento XML, definiendo los tipos de datos (etiquetas) que puede haber, sus atributos, contenido, y relaciones.

JSON

Lenguaje para almacenamiento e intercambio de información en un formato textual. Se basa en el formato de representación de objetos de JavaScript, y es más simple y rápido de procesar que XML.

Composición de servicios

El proceso de diseño implica utilizar una estrategia de reducción de la complejidad por partición y descomposición. Así se reduce el esfuerzo, mejora la reutilización. SOA facilita la composición mediante la definición de servicios y su integración en flujos de trabajos.

- **Orquestación:** Una entidad central controla el orden de ejecución de los servicios. Al uso de la orquestación se le denomina Business Process Management. Está centrado en la definición de workflow, reutiliza servicios y sigue una arquitectura SOA.
- **Coreografía:** Todos los servicios se ejecutan de manera independiente siguiendo unas normas preestablecidas.



TEMA 6. PRUEBA DE SISTEMAS Y SERVICIOS

Calidad del software

La calidad del software se define como la conformidad con los requisitos que se esperan.

- **Aseguramiento de la calidad:** Consiste en definir o seleccionar estándares aplicables al proceso de desarrollo.
- **Planificación de la calidad:** Consiste en definir un plan de calidad para cada proyecto que contenga planes, descripciones, objetivos...
- **Control de la calidad:** Consiste en vigilar el proceso de desarrollo para asegurar que se siguen los procedimientos.
 - A. **Revisiones,** Inspecciones, Auditorías.
 - B. **Evaluación mediante métricas:** Utilización de definiciones de medidas numéricas para los atributos.
 - C. **Pruebas.**

Pruebas

Una prueba consiste en la ejecución de un programa para encontrar errores. Las pruebas son una actividad imprescindible y muy costosa.

Las técnicas de pruebas son:

- **Manual:** el diseñador de pruebas parte de documentos informales de requisitos y genera otro documento informal de plan de pruebas. El ingeniero de pruebas de forma manual convierte el plan de pruebas en casos de prueba que ejecuta manualmente.
- **Captura:** el ingeniero de prueba ejecuta manualmente los casos de prueba sobre la aplicación interactiva (web por ejemplo), y se genera una traza de interacción de la prueba con entradas y resultados obtenidos. Esta traza se convierte en una script de pruebas.
- **Basado en scripts:** el diseñador parte de documentos informales y genera otro documento informal de plan de pruebas. El ingeniero de prueba escribe un documento formal procesable (script) que pasa las pruebas y genera un informe.
- **Basado en modelos:** el diseñador de pruebas construye un modelo formal procesable del comportamiento del sistema (por ejemplo un diagrama de estados UML), a partir del cual el ingeniero de pruebas genera una script que pasa las pruebas y genera el informe.

Las pruebas se puede clasificar según:

- **Sus objetivos:**
 - D. **Estáticas:** Basadas en la no ejecución de código. Permiten garantizar el cumplimiento de los estándares de desarrollo.
 - E. **Estructurales:** Basadas en la ejecución de código. Ejecutan sentencias, bucles, condiciones.
 - F. **Funcionales:** Basadas en la ejecución y el análisis de resultados.
 - G. **No funcionales:** Requisitos no funcionales. Permiten garantizar seguridad, funcionalidad, etc.

- **Su alcance:**

- H. **Unitarias:** Se ejecutan sobre un elemento y son necesarias para darlo por bueno.
 - **Caja negra:** Pruebas que ignoran la estructura interna y comprueban el comportamiento mediante entradas y salidas.
 - A. **Clase equivalente:** Valores de entrada que dan lugar a salidas esperadas.
 - B. **Análisis de Valores Límite:** Dominio de la salida para probar valores límite.
 - **Caja blanca:** Consideran la estructura interna del módulo y buscan ejecutar todos los caminos
 - **Caminos mínimos:** Sirve para asegurar que todos los caminos se ejecuta al menos una vez. Nodos: condiciones, Arcos: flujos. Complejidad ciclomática = $n^{\circ} \text{ arcos} - n^{\circ} \text{ nodos} + 2 = n^{\circ} \text{ condiciones} + 1$.
- I. **De integración:** Se realizan a la hora de ensamblar el sistema.
 - Ascendente: Se toman módulos inferiores y se van integrando poniendo módulos superiores.
 - Descendente: Los módulos inferiores son sustituidos. Permiten la visibilidad temprana de los resultados.
- J. **De sistema:** Permiten probar los contenidos, la funcionalidad, fiabilidad, seguridad.
- K. **De aceptación/validación:** Se realizan con criterios de validación. Son las pruebas finales. Adaptan un enfoque de caja negra.

Mantenimiento

El mantenimiento es la modificación del sistema una vez entregado. Los tipos son:

- **Correctivo:** Reparar aquello que no funciona correctamente por defecto, fallo o error.
- **Adaptativo:** Adaptar el sistema a cambios en la plataforma hardware o software.
- **Perfectivo:** Adecuar el sistema a nuevos requisitos.

Selenium

Selenium es un conjunto de herramientas para la automatización de pruebas de aplicaciones web. Está formado por:

- **Selenium IDE:** herramienta para la grabación y reproducción.
- **Selenium WebDriver:** herramienta basada en scripts para la reproducción de pruebas.

Uso: Se trata de hacer un recorrido típico sobre el diagrama de estados grabando con Selenium IDE y generando un caso de prueba JUnit que puede ser modificado y reproducido por Selenium WebDriver.

Diagrama de Flujo TFGs

