Tema 3:

Servicios de nombres

Un **servicio de nombres** es aquel que provee a los clientes información sobre elementos de un sistema distribuido de manera legible para el ser humano, con el fin de identificar los mismos. Es usado para referencia recursos y usuarios, así como comunicar y compartir recursos. Este servicio almacena colecciones de pares <nombre, atributo> y busca atributos a partir de nombres. Símil con las páginas blancas.

Características de los servicios de nombres:

- Utilizan el paradigma cliente/servidor.
- Es un servicio independiente fácilmente escalable.
- Independencia de su ubicación.
- Alta disponibilidad.
- La información se almacena jerárquicamente.
- Débil consistencia de replicación.
- Flexibilidad.
- BD optimizada: orientada a la lectura de información, datos de una entrada en un único registro, no necesita transacciones y tampoco bloqueos.

Un **servicio de directorio** es aquel que provee a los clientes información sobre objetos que satisfacen una determinada descripción. Es similar al servicio de nombres pero éste nos permite buscar nombres a través de atributos. Símil con las páginas amarillas.

Características de los servicios de directorio (además de las de los servicios de nombres:

- Información acerca de objetos relacionados (recursos de red, personas…).
- Refuerza la seguridad para proteger a los objetos de intrusos.
    - Servicios de nombre (páginas blancas): DNS (específico).
- DNS es un servicio de nombre ya que nos permite buscar un atributo a través de un nombre, y específico debido a que nos permite crear un espacio y extender la funcionalidad, guardando la información en la estructura que se crea.
    - Servicio de directorio (páginas amarillas): UDDI (específico) y LDAP (general).
- UDDI es un servicio de desarrollo ya que nos permite realizar búsquedas por una descripción dada, además es específico por lo mismo que lo es el DNS.
- LDAP es un servicio de directorio ya que nos permite realizar búsquedas por atributo y/o distinta información, además es de propósito general ya que sólo nos permite crear un espacio de nombres.

DNS

DNS (*Domain Name System*) es un servicio de nombre ya que nos permite buscar un atributo a través de un nombre, y específico debido a que nos permite crear un espacio y extender la funcionalidad, guardando la información en la estructura que se crea.

Definición:

- Establece una jerarquía de nombres para nodos en redes TCP/IP.
- Asocia cada nombre con una dirección IP.
- Todo un sistema basado en BD distribuida que permite resolver (directa e inversamente) nombre DNS a dirección IP y viceversa.

Elementos:

- **Espacio de nombres**: Jerarquía estructurada de dominios para organizar los nombres.
- **Registros de recursos**: Asignan nombres a un tipo específico de información de recurso (utilizada para resolver el nombre en el espacio de nombres).
- **Servidores DNS**: Almacenan y responden a las consultas de nombres.
- **Clientes DNS**: Consultan a los servidores para buscar y resolver nombres de un tipo de registro de recursos.

Funcionamiento básico:

1. Petición del cliente.
2. El cliente DNS solicita la resolución de un nombre.
3. El servidor DNS revuelve la IP asociada al nombre.
4. El cliente puede realizar su petición.

Espacio de nombres:

- **Dominio**: Agrupación lógica del espacio de nombres. Un subárbol del espacio de nombres de dominio => ua.es
- **Subdominio**: Otros dominios dentro de un dominio =>desarrollo.ua.es
- **Zona**: Unidad más pequeña y manejable, creada por delegación. Relacionada con la gestión y resolución de recursos. Normalmente es un archivo físico que gestiona un conjunto de recursos y puede que de varios dominios.

LDAP (*Lightewight Directory Access Protocol*)

LDAP es un servicio de directorio ya que nos permite realizar búsquedas por atributo y/o distinta información, además es de propósito general ya que sólo nos permite crear un espacio de nombres.

Características del LDAP

- Puede utilizar bases de datos recuperar información de forma rápida o hacer una consulta de los mismos.
- Puede dividir el árbol de directorios en subárboles gestionados por diferentes servidores LDAP. Distribuye en toda la red información lista para ser usada por todas las aplicaciones. Todo esto sin afectar ningún acceso externo a estos datos.
- Independencia de la plataforma: como LDAP es un protocolo estándar, que proporciona un método de acceso a datos remoto y local, es posible intercambiar completamente la implementación del LDAP sin afectar la forma en que se podrá acceder a los datos. El cliente y servidor se pueden ejecutar en SO diferentes.
- Seguridad integrada en el repositorio: La información de acceso se almacena en el mismo repositorio.
- Implementación fácil del cliente: la disponibilidad de interfaces de programación (API) del LDAP para casi cualquier lenguaje de programación facilita la compatibilidad del LDAP con prácticamente todas las aplicaciones.
- Gran difusión.
- Bajo coste ⇒ Sin pagar licencias por su uso ni para disponer de clientes o servidores.

Funcionamiento del LDAP

En el modelo cliente-servidor de LDAP, ante una consulta concreta de un cliente, el servidor contesta con la información solicitada. La respuesta puede ser, de forma alternativa (o además de la información que se había solicitado), un puntero que indica donde conseguir esta información o datos adicionales.

Se puede decidir separar el directorio entre varios servidores por motivos organizativos o para facilitar su gestión. El servidor está configurado para devolver referencias a otros servidores LDAP en caso de que se le pida información de la que no dispone, pero que sabe dónde conseguir.

Se puede aumentar la disponibilidad y la fiabilidad del directorio utilizando más de un servidor LDAP para mantener la información.

Usos empresariales

- Directorios de información.
- Sistemas de Autenticación/Autorización.
- Sistemas de información de cuentas de correo electrónico.
- Grandes sistemas de autenticación basados en RADIUS (Remote Access Dial-In User Server -con control de consumo-).
- Servidores de certificados públicos y llaves de seguridad.
- Perfiles de usuarios centralizados.

Modelos LDAP

- **Modelo de información**: Define qué tipo de información se puede almacenar en el directorio y las unidades básicas en que el LDAP estructura la información. Describe la estructura de la información almacenada en el directorio LDAP.

  Estructura y tipos de datos (esquemas, entradas, atributos). Utiliza ficheros ASCII para entradas LDAP: formato LDIF. => Diapositivas

- **Modelo de asignación de nombres o nomenclatura**: Define cómo se organiza y se referencia la información. Las entradas de directorio se disponen en una estructura de árbol jerárquica. El nombre de una entrada debe ser único en cada servidor LDAP. Describe cómo se organiza e identifica la información en el directorio LDAP

  Define cómo referenciar de forma única las entradas y los datos en el árbol de directorios =>**RDN** (Nuevo Nombre completo Relativo) y **DN** (Nombre Distintivo). => Diapositivas

- **Modelo funcional**: Define cómo se recupera y modifica la información del directorio, describiendo las operaciones que se pueden realizar en el acceso, el mantenimiento y la gestión del directorio. Describe qué operaciones pueden ser realizadas con la información almacenada en el directorio LDAP.

  Operaciones para acceder al árbol de directorio: autenticación, solicitudes y actualizaciones. => Diapositivas

- **Modelo de seguridad**: Muestra cómo se controla el acceso a la información contenida en el directorio. Antes de que un cliente pueda acceder a los datos de un servidor LDAP, se llevan a cabo dos procesos: **autenticación** (para asegurar que las identidades de los usuarios y máquinas están validadas) y **autorización** (para controlar el acceso a los recursos del directorio a las personas o entidades que intentan acceder a ellos).

Justifica la gestión distribuida de LDAP, define las ventajas sobre otros sistemas de información tradicionales y enumera y describe los motivos que pueden llevar a esta gestión distribuida. ¿Qué técnica es utilizada para relacionar las diferentes partes del espacio de nombres cuando es distribuido?

- La justificación de porqué la gestión es distribuida es que de esa forma puede dividirse en subárboles por motivos de rendimiento, localización geográfica (q ayuda al rendimiento, porque está más cerca y a los contenidos) y por cuestiones administrativas.
A diferencia del sistema tradicional (x500):

  - LDAP utiliza TCP/IP en lugar de protocolos OSI
  - El modelo funcional de LDAP es más simple y ha eliminado opciones raramente utilizadas en X.500. LDAP es más fácil de comprender e implementar.
  - LDAP representa la información mediante cadenas de caracteres en lugar de complicadas estructuras ASN.1.

- La técnica para referenciar las diferentes partes del espacio de nombres es el uso del objeto ObjectClass:referal. Atributo obligatorio para almacenar la url de acceso a los subárboles.

JNDI

La arquiectura JNDI consiste en un API y un "service provider interface (SPI)". Las aplicaciones Java usan el API JNDI para acceder a una gran variedad de servicios de nombres y directorios. El SPI permite conectar de forma transparente una gran variedad de servicios de nombres y directorios, por lo tanto permite a las aplicaciones Java usar el API JNDI para acceder a sus servicios.

Características:

- JNDI ≈ JDBC
- Unificar el acceso a SN y SD: Acceso transparente
- API de acceso a
    - Servicios de nombre
    - Servicios de directorio
- SPI: Interfaz de proveedor de servicios
- Arquitectura de plugin: conexión dinámica de diferentes implementaciones
- Federación: Comunicación entre Proveedores de servicios (LDAP ⇒ DNS)

Funciones JNDI:

- Interfaz Context (javax.naming)
    - Inicializar el contexto
    - Buscar (lookup)
    - Unir y desunir (bind y unbind)
    - Renombrar objetos (rename)
    - Crear y eliminar subcontextos (createSubcontext y destroySubcontext)
    - Enumerar enlaces (listBindings)

- Interfaz DirContext (javax.naming.directory)
    - Extiende javax.naming
    - Acceso a directorios además de nombres
    - Trabaja con atributos
    - Obtener y modificar atributos (getAttributes y modifyAttributes)
    - Búsqueda por filtros (search)

- JNDI => rmiregistry