

---

# Capítulo 4

---

## PROYECTO ANS

---

Este capítulo trata sobre el proyecto ANS(*ATM Name System*), el cual es una implementación del sistema ANS para máquinas UNIX(en concreto para sistema operativo Solaris). Explica cómo se ha ido estructurando el proyecto en distintos bloques dando lugar a un sistema “*complejo*” como es el sistema ANS.

### INTRODUCCIÓN

El proyecto ANS nació como una necesidad de dar servicio directorio de nombres para una red ATM de ordenadores, tal y como ya existía en la red INTERNET. Las especificaciones del sistema estaban recogidas de antemano en el documento *ATM Name System Specification Version 1.0(af-saa-0069.000)* del *ATM FORUM Technical Committee* . Estas especificaciones están basadas en el sistema DNS con las extensiones necesarias para poder implementarse dentro de una red ATM.

El sistema ANS implementado en este proyecto consiste en una aplicación desarrollada en lenguaje C++ para máquinas UNIX. Se ha pretendido dar al proyecto la forma más estructurada posible, desarrollando por líneas separadas los diferentes elementos que componen el sistema, además de utilizar la programación orientada a objetos y el uso de métodos genéricos virtuales que abstraen del tipo de objeto utilizado. Hay dos bloques claramente diferenciados, uno es la parte *servidora* de la aplicación(implementa al servidor de nombres) y el otro es la parte *cliente* de la aplicación(implementa al cliente del sistema ANS). Estos dos bloques estarán a su vez formados por bloques más pequeños que implementarán funciones más específicas(gestión de las comunicaciones, gestión de los datos, manejo de la memoria caché... etc.).

---

## ENTORNO DE DESARROLLO

El proyecto ANS se ha desarrollado en sistema operativo UNIX, utilizando lenguaje de programación C++. Aunque en un principio fue pensado únicamente para el sistema operativo *Solaris*, se ha conseguido compatibilidad con otro sistema operativo UNIX como es *Linux*.

Desde un primer momento el cuerpo del proyecto ha sido desarrollado y depurado en el sistema operativo Linux, utilizando el sistema operativo Solaris para el desarrollo de la librería de comunicaciones a través de la red ATM y la prueba final de funcionamiento del sistema ANS.

Las librerías utilizadas han sido las genéricas que acompañan al sistema operativo *Solaris 2.5*. Una de ellas que cabe destacar es el uso de las IPC(*Inter Process Communication*) para el desarrollo de la memoria caché dentro del sistema, tal y como se explicará más adelante. Esta librería provee de herramientas para la comunicación de los distintos procesos del sistema dentro de una misma máquina.

El compilador de C++ utilizado ha sido el *gcc*. Otra herramienta utilizada ha sido la utilidad de UNIX *flex*. Esta herramienta es un analizador léxico con el que se ha implementado un analizador de los ficheros de datos, los cuales poseen una sintaxis particular, tal y como permite las recomendaciones del protocolo.

Las comunicaciones a través de la red ATM del sistema ANS se han desarrollado utilizando la API *XTI* que acompaña al software de la tarjeta SBA-200. Esta librería nos permite el establecimiento y liberación de conexiones dentro de la red ATM, así como la transferencia de datos.

---

## BLOQUES DEL SISTEMA ANS

El sistema ANS se ha dividido en una serie de bloques, cada uno con una funcionalidad bien definida y, en lo posible, independiente de los demás bloques, todo ello con el objetivo de facilitar el desarrollo del sistema ANS. Los bloques en los que se ha dividido el sistema son los siguientes:

- *ANALIZADOR FICHEROS.*  
Es el encargado de interpretar la sintaxis de los distintos ficheros de datos correspondientes a las distintas zonas del espacio de nombres de dominio.
- *COMUNICACIÓN EN LA RED ATM.*  
Es una librería que ofrece funciones para el establecimiento y liberación de conexiones conmutadas(SVC) a través de la red ATM, así como la transferencia de datos.
- *MENSAJE PROTOCOLO ANS.*  
Los mensajes del protocolo ANS son tratados a través de una clase *MENSAJE*. Esta clase sirve para manejar los distintos datos que contiene un mensaje ANS. La clase *MENSAJE* provee métodos con los que facilitar las diferentes operaciones que se pueden realizar con un mensaje del protocolo ANS: codificación/decodificación formato red, inserción/extracción de datos, etc...
- *MEMORIA CACHE.*  
La clase *CACHE* provee todas las operaciones necesarias para el uso de una memoria caché por parte de algún/os elemento/s del sistema ANS. Estas operaciones básicas son: creación/destrucción de la zona de memoria caché, inserción/extracción de datos, mantenimiento, etc....
- *ACTUALIZACIÓN DE ZONAS.*  
Es una librería que ofrece funciones para la actualización automática de los ficheros de datos de zonas del espacio de nombres de dominio. Estas funciones implementan el protocolo básico de transferencia de mensajes tipo AXFR tal y como contempla las recomendaciones del protocolo.
- *MAQUINA DE ESTADOS DEL SERVIDOR DE NOMBRES.*  
El núcleo del servidor de nombres ANS ha sido desarrollado basándose en una máquina de estados. Esta máquina de estados contempla todas las posibles situaciones en la que se puede encontrar el servicio en la resolución de peticiones, realizando las acciones oportunas según la situación.
- *MAQUINA DE ESTADOS DEL CLIENTE.*  
El cliente del sistema ANS(*RESOLVER*) ha sido implementado como una máquina de estados. Esta máquina de estados contempla todas las posibles situaciones en la que se puede encontrar un *RESOLVER*(cliente) durante la resolución de una determinada petición.

---

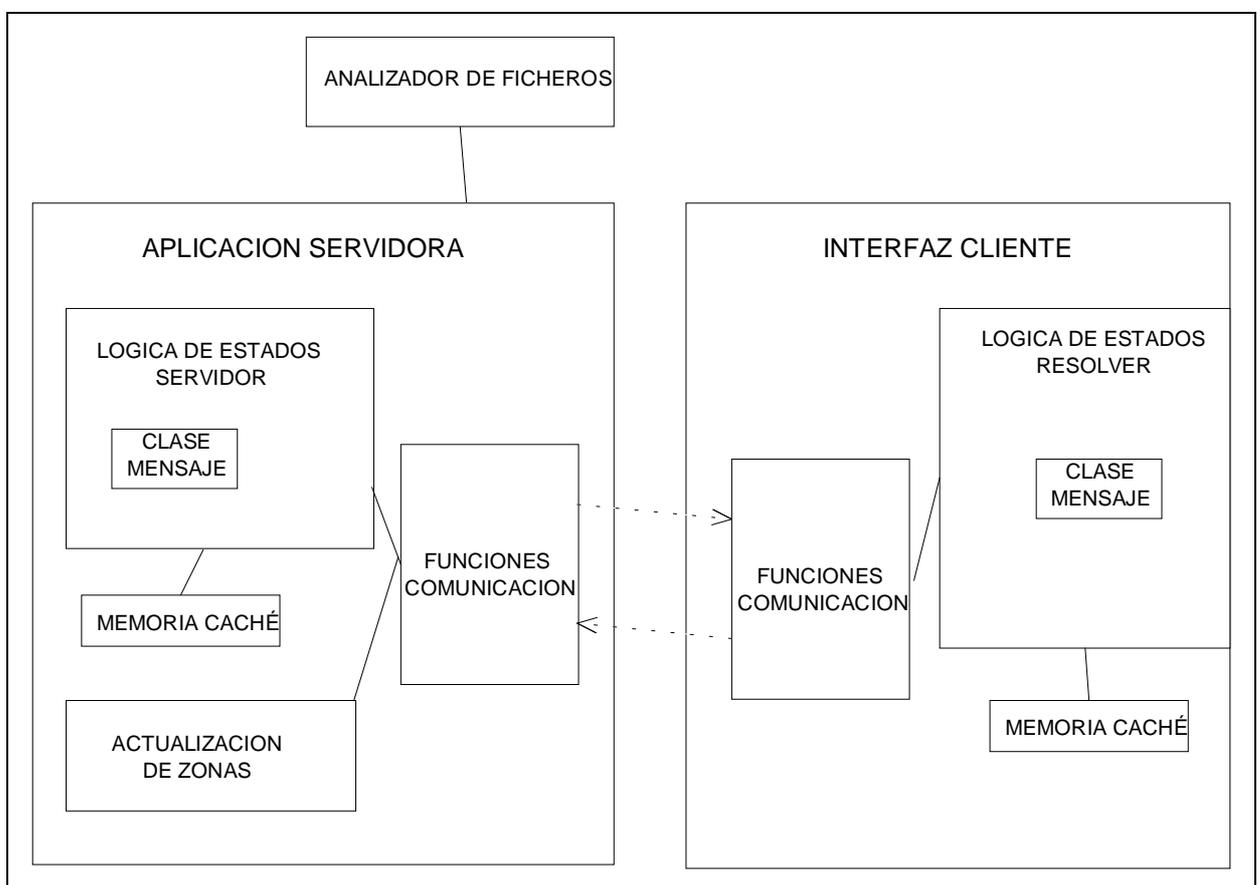
- **APLICACIÓN SERVIDORA.**

La aplicación servidora de nombres del sistema ANS es el proceso demonio que continuamente estará ejecutándose para dar servicio. Este proceso se encarga de las tareas diversas del servidor de nombres como pueden ser: mantenimiento de la memoria caché, recepción de peticiones, creación de nuevos procesos para dar respuesta a las peticiones que le sean cursadas.

- **INTERFAZ CLIENTE ANS.**

Es una librería que ofrece una serie de funciones y estructuras de datos para que cualquier programa usuario del sistema ANS pueda hacer peticiones al mismo, obteniendo en cada caso su respuesta.

A continuación se muestra un diagrama funcional que da idea de las relaciones existentes entre los distintos bloques que componen un servidor ANS con configuración básica:



En los siguientes capítulos de esta memoria se explica cada bloque del proyecto de una forma más detallada.