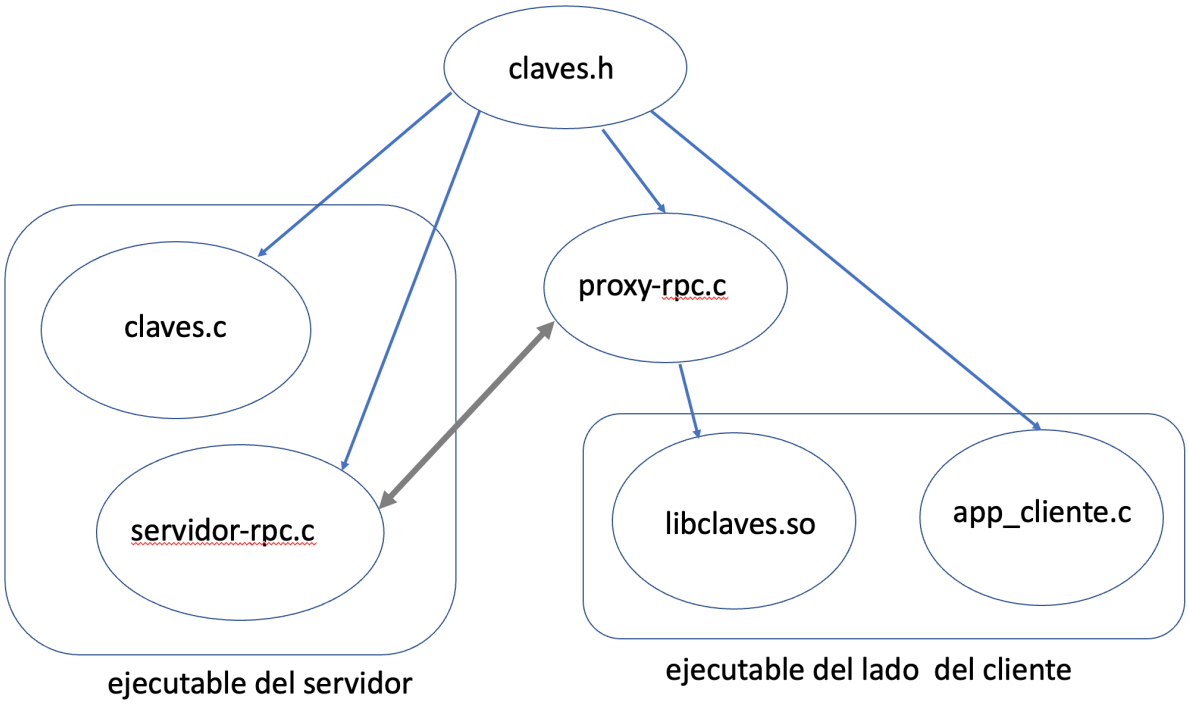
Sistemas Distribuidos.
Ejercicio Evaluable 3: RPC

Se desea diseñar e implementar el mismo servicio realizado en el ejercicio evaluable 1 y 2, pero en este caso utilizando ONC RPC.

En este ejercicio la aplicación tendrá la estructura de la siguiente figura:



En esta aplicación los ficheros `claves.h`, `claves.c` y `app-cliente.c` son los mismos que los implementados en el primer ejercicio. Para este ejercicio solo han de implementarse los archivos `proxy-rpc.c` y `servidor-rpc.c`, en este caso utilizando RPC y el lenguaje de programación C. Como parte de este ejercicio tendrá que definirse la interfaz de servicio (archivo con extensión .x que permite dar soporte a la funcionalidad de este ejercicio. Asimismo, harán falta los archivos que obtiene de forma automática el programa `rpcgen`.

Aclaraciones adicionales:

Tenga en cuenta, para las llamadas de la interfaz de tuplas, que también se considera error, por ejemplo, que se produzca un error en el sistema de RPCs.

En este ejercicio, el cliente y la biblioteca a desarrollar tiene que conocer la dirección IP del servidor que ofrece el servicio de tuplas (para las RPC no es necesario conocer el puerto). Para evitar tener una dirección físicamente programada en el código y no tener que pasar la IP en la línea de mandatos del programa cliente, se va a considerar que la dirección IP se va a pasar utilizando la siguiente variable de entorno:

- `IP_TUPLAS`: variable de entorno que define la IP donde ejecuta el servidor RPC.

Esta variable de entorno habrá de definirse en cada terminal donde se ejecute el cliente.

En cuanto al servidor, en este caso no será necesario pasarle en la línea de argumentos ningún puerto y este se ejecutará de la siguiente forma:

```
./servidor
```

Para el desarrollo de este ejercicio es importante diseñar y especificar la interfaz de servicio (archivo con extensión `.x`) que permite dar soporte a esta funcionalidad de claves.

Un aspeto importante a tener en cuenta en el desarrollo de este ejercicio es descubrir cómo integrar los archivos que se generan con el programa `rpcgen` para dar soporte a la funcionalidad descrita anteriormente. Por tanto, forma parte de este ejercicio el obtener la forma de integrar estos archivos en la solución final.

Material a entregar: Se deberá entregar el siguiente material:

Fichero **ejercicio_evaluable3.zip**, que incluirá, además de los archivos necesarios para la compilación del cliente y del servidor, los siguientes elementos:

- Un archivo `Makefile`, que permite compilar todos los archivos necesarios y generar la biblioteca `libclaves.so`. Este `Makefile` debe generar además dos ejecutables: el ejecutable del servidor, que implementa el servicio y el ejecutable de cliente o clientes utilizados para probar el sistema, obtenido a partir del archivo `cliente.c` (u otros clientes de prueba) y la biblioteca `libclaves.so`. Este `Makefile` se tendrán que incluir aquellos ficheros obtenidos con `rpcgen` que sean necesarios. Es necesario que el Makefile incluya una regla para invocar al programa rpcpgen, de forma que se creen lo archivos necesarios para el desarrollo de la aplicación final.
- Una pequeña memoria en **PDF** (no más de cinco páginas, incluida la portada), indicando el diseño realizado, la estructura de archivos utilizado en la compilación del servidor, de la biblioteca `libclaves.so` y del cliente y la forma de compilar y generar el ejecutable del cliente y del servidor.
- Dentro del fichero comprimido `ejercicio_evaluable3.zip`, también se incluirán todos aquellos archivos adicionales que necesite para el desarrollo del servicio. Por ejemplo, ficheros que gestionan las estructuras de datos elegidas, etc.

Para el almacenamiento de los elementos `key-value1-value2-value3` puede hacer uso de la estructura de datos o mecanismo de almacenamiento que considere más adecuado (listas, ficheros, etc.), el cual describirá en la memoria entregada. La estructura elegida **no** debe fijar un límite en el número de elementos que se pueden almacenar.

Se recomienda probar el servidor con varios clientes de forma concurrente.

Tenga en cuenta que el prototipo de las funciones (claves.h) `destroy`, `set_value`, `get_value`, `delete_key`, `modify_value y exist` no puede ser modificado.

Todo el código desarrollado en el ejercicio 1 y 2 sigue siendo válido en este ejercicio a excepción de los archivos `proxy-rpc.c` y `servidor-rpc.c`, que son los únicos que hay que implementar en este.

La entrega se realizará mediante Aula Global en el entregador habilitado. La fecha límite de entrega es: 27/04/2025. (23:55 horas).