* Hacer la aplicación multiplataforma no puntúa
* Hacer interfaz gráfica no puntúa

|  |  |
| --- | --- |
| **OBLIGATORIO** | **OPCIONAL** |
| Comprobar que la MIB tiene todo lo de la página 17 | Incluir la plantilla MODULE-IDENTITY |
| Compilar la MIB | Añadir objetos a la MIB a parte de los básicos |
| Cambiar nombre de proyecto\_SNMP.cpp a ag\_grupo9.cpp | Carga dinámica de la MIB |
| Carga estática | Poder usar otros tipos (OCTET STRING OBJECT IDENTIFIER, IpAddress) |
| Carga de tipos escalares (sin tablas) | Distinguir entre current/obsolete mirando STATUS |
| Distinguir (read-only) (write-only) | Poder hacer get y set de tablas |
| Get de mensajes con 1 solo objeto escalar | Poder hacer get y set de un número indeterminado de objetos ***(3)*** |
| Set de mensajes con 1 solo objeto escalar | Tratamiento de códigos de error |
| GetNext para objetos tipo INTEGER | Poder hacer get y set de enteros de distintas longitudes y no solo 1 byte ***(2)*** |
| Cambiar valores a la MIB | Set para modificar tablas |
|  | Set para creación dinámica de filas (Integer e IpAddress) |
|  | Set con identificación de errores |
|  | GetNext para mensajes diferentes a Integer |
|  | GetNext para más de un objeto |
|  | GetNext para tablas |
|  | Identificación de errores |
|  | Generación de notificación cuando se cumplen las condiciones de la página 26 ***(1)*** |
|  |  |
|  |  |
|  |  |

* DONE
* WORK IN PROGRESS PABLO
* WORK IN PROGRESS ALEX
* TO DO

Notas Alejandro:

* Distinguir entre read-only y read-write debe de ser fácil con la variable int acceso de typedef struct nodo; además dentro de los objetos de la mib se especifíca en

objeto->acceso

* Get y Set y GetNext de un solo objeto INTEGER está hecho ya falta la parte de leer de la MIB, y escribir en la MIB, la lectura del paquete y formar el paquete ya están hechos

Notas Alejandro opcionales:

* Añadir objetos a la MIB como la carga es estática no es difícil, es solo pensar algo que pueda servir con el contexto de para que es la MIB (se explica en la práctica).
* Poder usar otros tipos, en la MIB ya está hecho, y en las operaciones get, set y GetNext yo he hecho ya en la Get como ejemplo INTEGER, OBJECT IDENTIFIER, OCTET STRING e IpAddress, pero porque estoy tratando cualquier cosa que no sea Integer como un OCTET STRING y eso no es así por lo menos con el tipo OBJECT IDENTIFIER pero no creo que me sea complicado hacerlo
* Tratamiento de errores para get, set y getNext no debe de ser muy difícil, es solo mirar si es read-write o no, si el getnext es del último objeto, si el oid existe, etc, y luego poner eso en el paquete que eso es fácil y lo tengo implementado.
* Por último, hacer get, set y GetNext de varios objetos eso si puede ser un poco más lioso pero lo dejo para el final y ya, igualmente todo lo anterior deberíamos de acabarlo fácil