Laser Library

**Laser Library** es la interfaz de programación en C++ para el uso cómodo del láser Hokuyo-URG-04LX con la librería de gráficos OpenGL.

Para implementar las funciones interesantes para el desarrollo de las diferentes aplicaciones de los equipos se ha optado por desarrollar la clase *LaserManager* que tiene las siguientes funcionalidades:

* Iniciar la conexión con el puerto serie y delimitar el espacio de lectura del laser entre los 90 grados y los -90 grados.
* Establecer los vértices que delimitan el espacio de trabajo interesante.
* Devolver el punto de interés dentro del espacio de trabajo.

Para implementar las funcionalidades descritas arriba se desarrollan los siguientes métodos estático:

* *Bool Open(void)*: que abre la conexión con el driver Ugr del dispositivo. Establece el ángulo de trabajo entre -90 y 90 grados e inicia la medición de distancias. Si la conexión no se puede abrir entonces se devuelve *false* y queda a criterio del programador lo que hacer en caso de que no se establezca la conexión.
* *setWorkSpaceVertex(int right, int top, int left, int bottom)* : establece los metros que hay de separación hasta cada uno de los vértices teniendo en cuenta la posición del laser. Cada uno de los parámetros representan las siguientes coordenadas:
  + *RIGHT*: milimetros a la derecha del láser.
  + *TOP*: milimetros hacia el frente del láser.
  + *LEFT*: milimetros hacia la izquierda del láser.
  + *BOTTOM:* milímetros mínimos a los que hay que estar para dar como válido un punto.
* *readLaser(void)*: devuelve un objeto del tipo *PlanePoint* que contiene el punto más cercano de los leídos por el láser. Esta función también se encarga de gestionar la apertura y cierre de conexión cuando no se puede leer nada del láser por causas externas a la aplicación.

La intención de esta clase es que se llame a todas las funciones en el siguiente orden:

Durante la fase de inicialización en el *main*:

1. *Open*
2. *setWorkSpaceVertex*

Dentro del bucle de atención de eventos o del gestor del temporizador a criterio del programador:

1. *readLaser*

Tenemos de ejemplo el siguiente código:

