

## ANEXOS

### Anexo A. Manual de usuario

En este manual se explica cómo utilizar el modelo de EV3 en el simulador V-REP y la toolbox de Matlab para controlarlo. En las primeras secciones se enumeran los pasos básicos para la instalación. A continuación, en las siguientes secciones, se describirá cómo poner en marcha una simulación, los modelos de EV3 y escenas para V-REP disponibles en el paquete de instalación, y como añadir vistas flotantes para las cámaras proporcionadas en el modelo.

#### A.1. Instalación de V-REP

Para poder simular nuestro robot, se utiliza el entorno de simulación V-REP. En primer lugar hemos de instalar V-REP en nuestro equipo. La versión de V-REP usada en nuestro trabajo es la 3.3.0 para Windows y Linux, ya que es la versión estable que los equipos han utilizado. Para ello basta con instalar los paquetes que aparecen en su web:

<http://www.coppeliarobotics.com/downloads.html>

Con ello, nos descargaremos el archivo de instalación al ordenador para posteriormente extraerlo (Linux) o instalarlo (Windows). Una vez concluida la instalación ya podemos empezar a utilizar el entorno, cuya apariencia se muestra en la Figura 100.

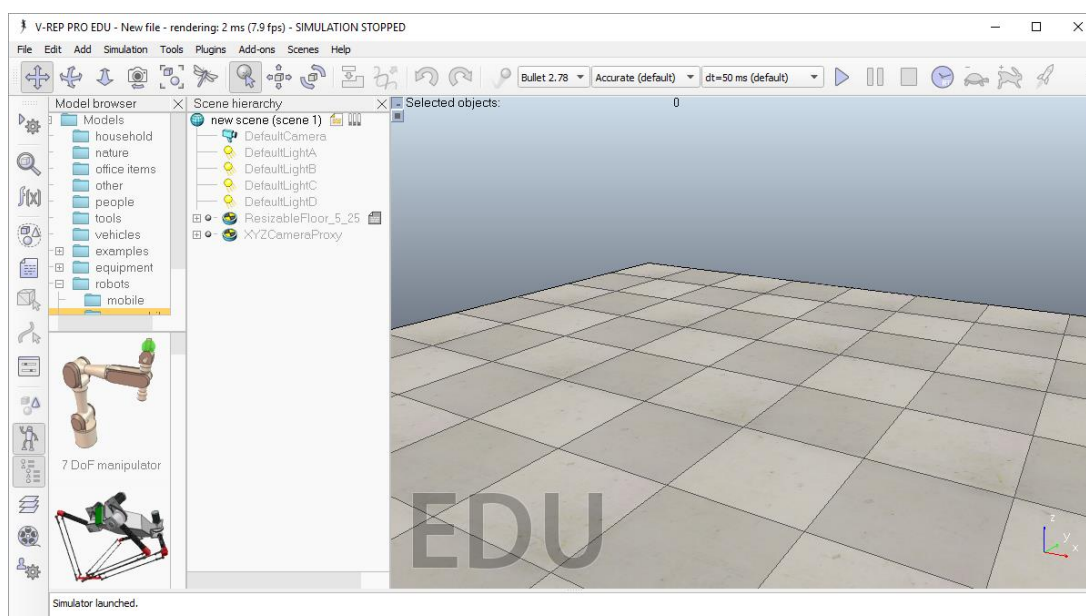


Figura 100. Interfaz del entorno de simulación de V-REP

## A.2. Instalación de Matlab

La versión de Matlab que se ha usado para la realización del trabajo es la 2016a. Para poder obtener Matlab, hace falta adquirir una licencia, la cual se puede comprar desde su sitio web; no obstante, la Universidad de Málaga puede distribuir licencias para estudiantes y docentes de manera gratuita desde el Servicio Central de Informática en la siguiente dirección:

[https://software.uma.es/index.php?option=com\\_content&view=article&id=412:matlab-estudiante&catid=198&Itemid=3070&showall=&limitstart=8](https://software.uma.es/index.php?option=com_content&view=article&id=412:matlab-estudiante&catid=198&Itemid=3070&showall=&limitstart=8)

Para poder instalar Matlab, basta con seguir los pasos que aparecen en dicho enlace. Una vez instalado y con la licencia activada, ya podemos utilizar Matlab, cuya interfaz aparece en la Figura 101.

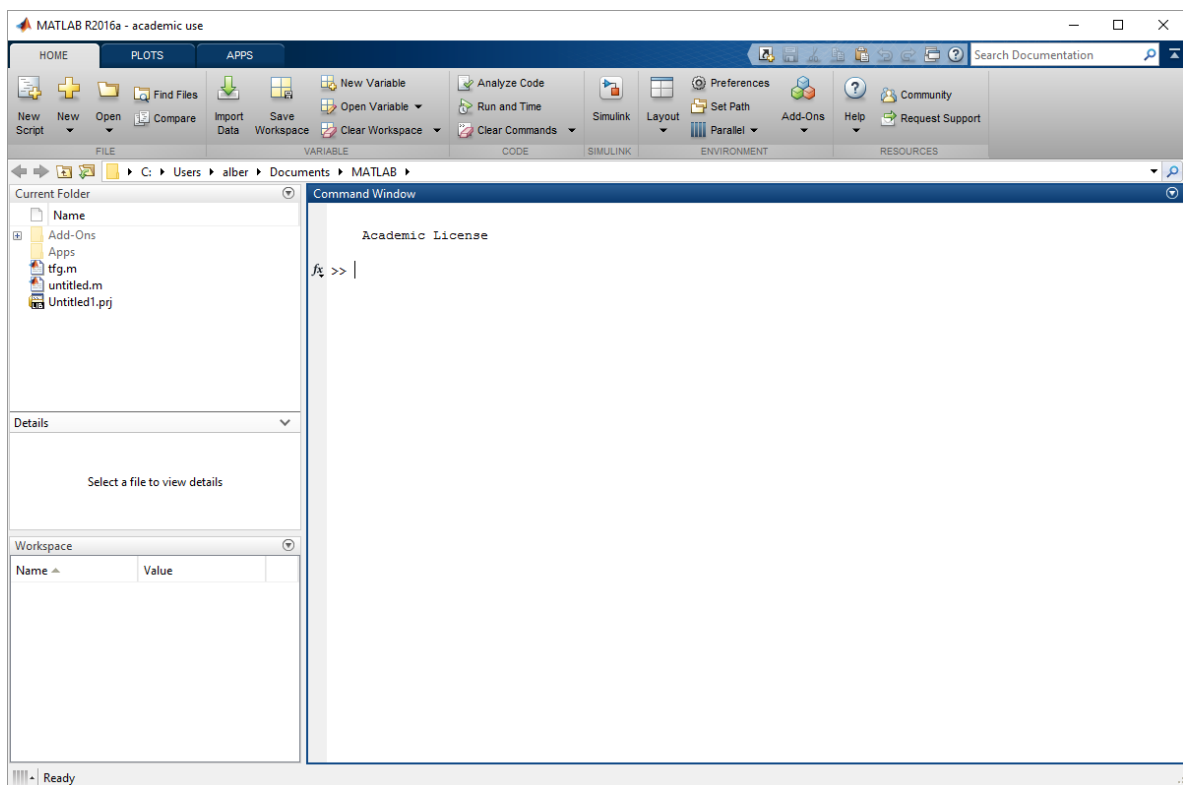


Figura 101. Interfaz de Matlab

## A.3. Instalación de la toolbox en Matlab

Para poder instalar la toolbox desarrollada, para todos los sistemas operativos (Windows, Linux y, previsiblemente, MacOS), hay que seguir los siguientes pasos:

1. Descargar el archivo EV-R3P\_1.3.1.zip del siguiente enlace:  
<https://drive.google.com/open?id=0B2mnuwTFyL-7ZkNFa1JlcmpXeWM>

O del repositorio de GitHub:

<https://github.com/albmardom/EV-R3P>

2. Al descomprimir el archivo, deben encontrarse los siguientes directorios (Figura 102):

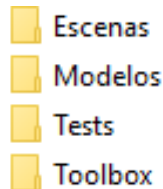


Figura 102. Directorios del paquete de instalación

3. Instalar la toolbox en Matlab; para ello, abriremos previamente Matlab y buscaremos el archivo *LEGO EV3 V-REP toolbox x64 (NXC versión).mltbx*, para procesadores x64, o *LEGO EV3 V-REP toolbox x86 (NXC versión).mltbx*, para procesadores x86, en el directorio descomprimido, concretamente, en la carpeta *Toolbox*. Al hacer doble clic nos aparecerá un cuadro de diálogo para confirmar que vamos a instalarla (Figura 103), y pulsaremos el botón “install”.

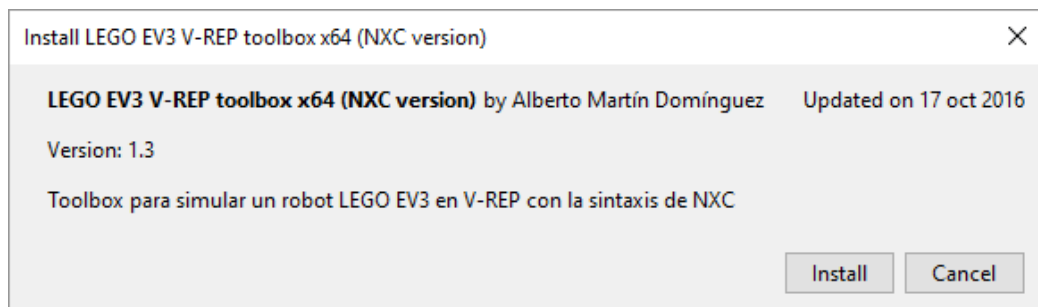


Figura 103. Cuadro de diálogo de instalación de toolbox

Para comprobar si se ha instalado correctamente la toolbox, escribir en la consola de Matlab un comando de ayuda con algunas de las funciones como argumento, como, por ejemplo: “help OnFwd” (Figura 104).

```
>> help OnFwd
OnFwd proporciona una velocidad al motor indicado
OnFwd(motor, velocity) activa el motor "motor" a una velocidad
"velocity" indicada entre -100 y 100 hacia adelante (siendo el porcentaje de velocidad a
partir de la velocidad máxima). Las constantes de motores que se pueden
usar son las siguientes:

* OUT_A: puerto A de actuadores (motor A)
* OUT_C: puerto C de actuadores (motor C)
* OUT_AC: puertos A y C de actuadores (ambos motores)

See also OnRev, Off, MotorRotationCount, ResetRotationCount
```

Figura 104. Comando “help” a la función “OnFwd”

En el caso de que no funcione, podemos abrir con un programa de extracción de archivos el fichero de toolbox (“.mltbx”), extrayendo los códigos de este del directorio *fsroot*. Luego, añadiremos al path de Matlab el directorio donde hemos descomprimido los ficheros fuente, además de la subcarpeta Funciones, pero, sin añadir la subcarpeta *private* (Figura 105).

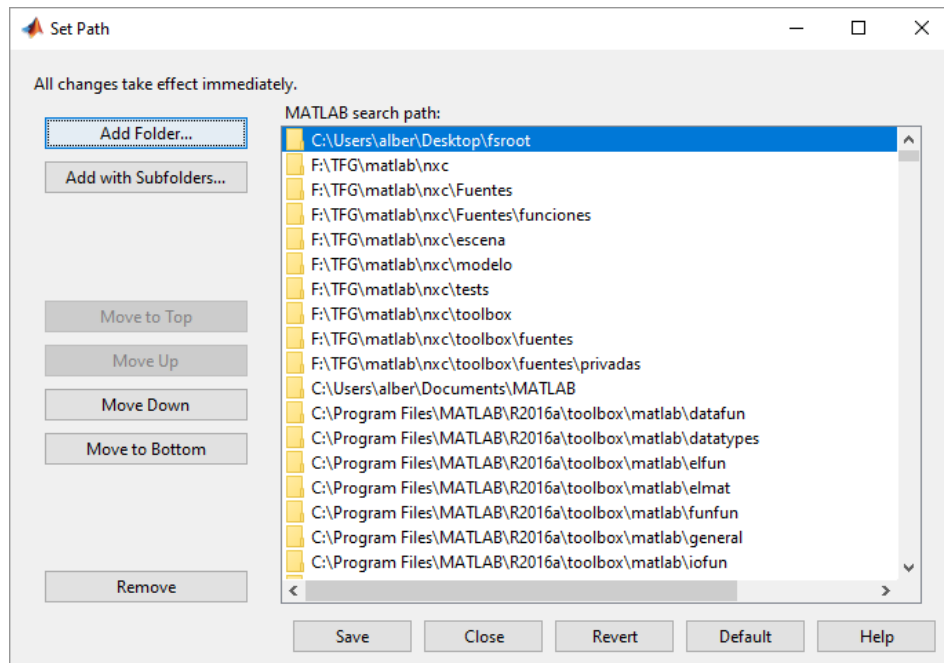


Figura 105. Path de Matlab con la carpeta importada de la toolbox

Con esto, ya solo falta probar el comando “help” mencionado anteriormente.

#### A.4. Carga de los modelos de EV3 en V-REP

En el simulador, para cargar el modelo solo tendremos que irnos, en la barra de menús, a “File → Load model...”, y buscar, en el directorio donde se han extraído los ficheros descargados, en la carpeta *Modelo*, los dos modelos disponibles de EV3:

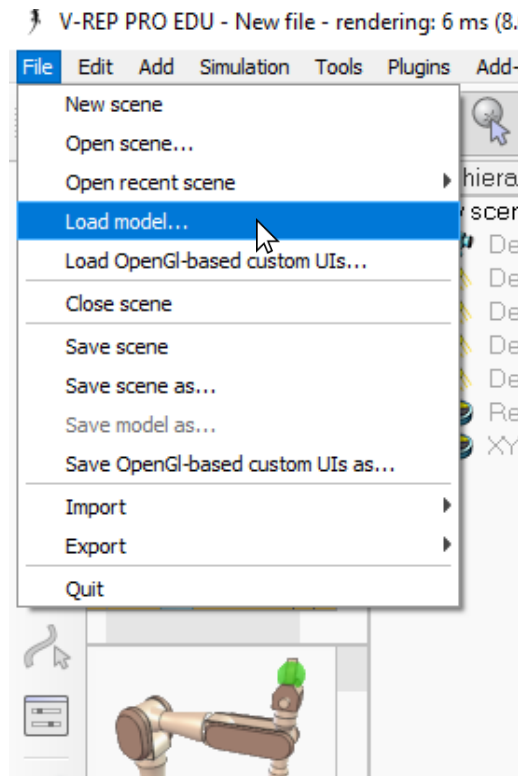


Figura 106. Situación de la opción “Load model...” en V-REP

- **“LEGO\_EV3\_GROUP.ttm”**. Este modelo es el más detallado, pero, por el contrario, es el más pesado para renderizar; no recomendable para equipos de bajo rendimiento.
- **“LEGO\_EV3\_MERGE.ttm”**. Este modelo es menos detallado que el anterior, pero en cambio es más eficiente en la renderización; recomendado para ordenadores que no puedan utilizar fluidamente el modelo anterior.

Si se quiere que el modelo cargado aparezca en el Explorador de Modelos del simulador, basta con guardar el modelo en “File → Save model as...” (teniendo el objeto “LEGO\_EV3” seleccionado en la escena), en el siguiente directorio: *<DIRECTORIO\_INSTALACIÓN\_V-REP>/Models/robots/mobile*.

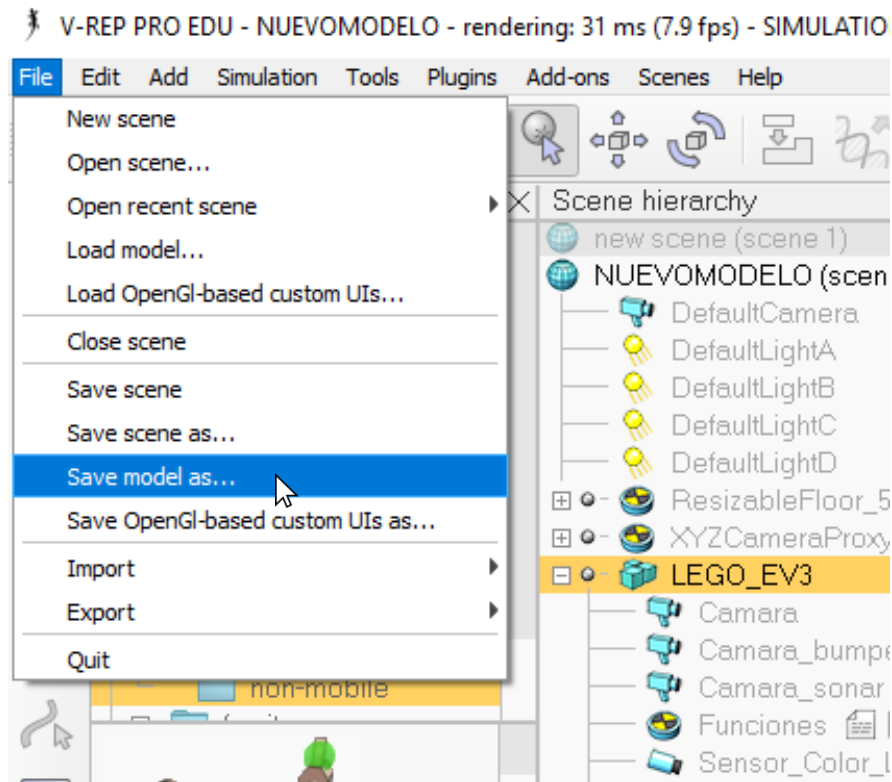


Figura 107. Situación de la opción “Save model as...” con el objeto “LEGO\_EV3” seleccionado en V-REP

Hecho esto, los modelos deberían aparecer en el Explorador de Modelos (Figura 108), situado a la derecha de la pantalla.

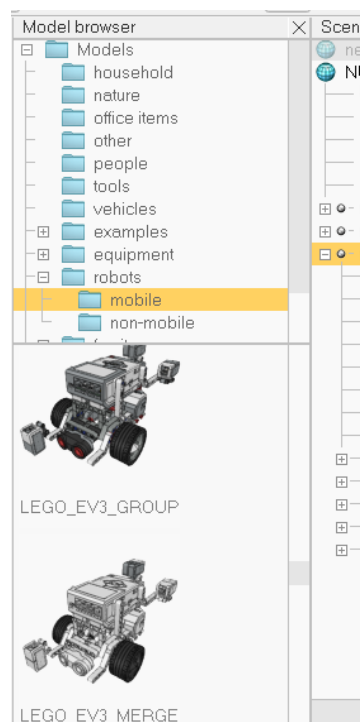


Figura 108. Explorador de modelos de V-REP con los dos tipos de LEGO EV3

Por último, siempre que se cargue el modelo (o una escena que lo contenga), se recomienda iniciar la simulación y luego detenerla, esto es debido a que la simulación tarda mucho al iniciarse la primera vez y, al usar las funciones y los objetos manejadores, se produciría errores en Matlab.

### A.5. Vistas

A ambos modelos se les ha dotado de cámaras para ayudar al desarrollo de programas. Podemos obtener imágenes de ellas si añadimos vistas flotantes; estas se añaden haciendo clic derecho en la escena y luego “add → Floating view” (Figura 109).

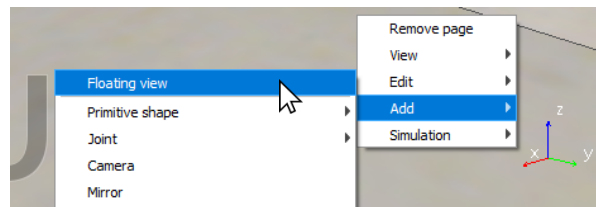


Figura 109. Lugar para añadir vistas flotantes en V-REP

Ahora, haciendo clic derecho dentro de la vista flotante, seleccionamos “view → View selector...” y ahí seleccionaremos la cámara que queramos ver.

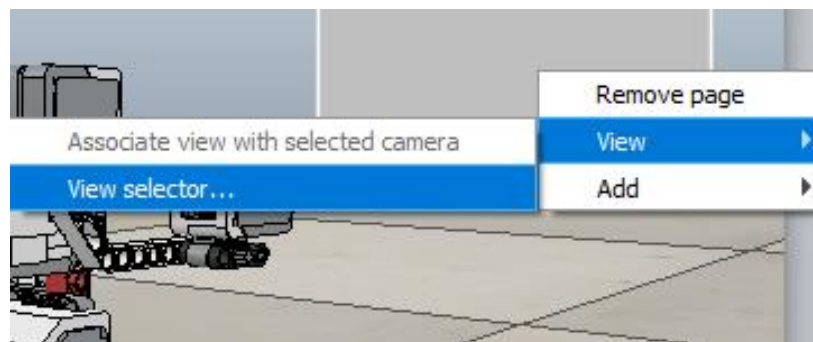


Figura 110. Situación del menú “View selector...” en V-REP para las vistas flotantes

Este procedimiento también sirve para ver las imágenes del sensor de visión, como, por ejemplo, los modos del sensor de color del EV3; en la Figura 111 podemos ver todas las cámaras del robot como vistas flotantes.

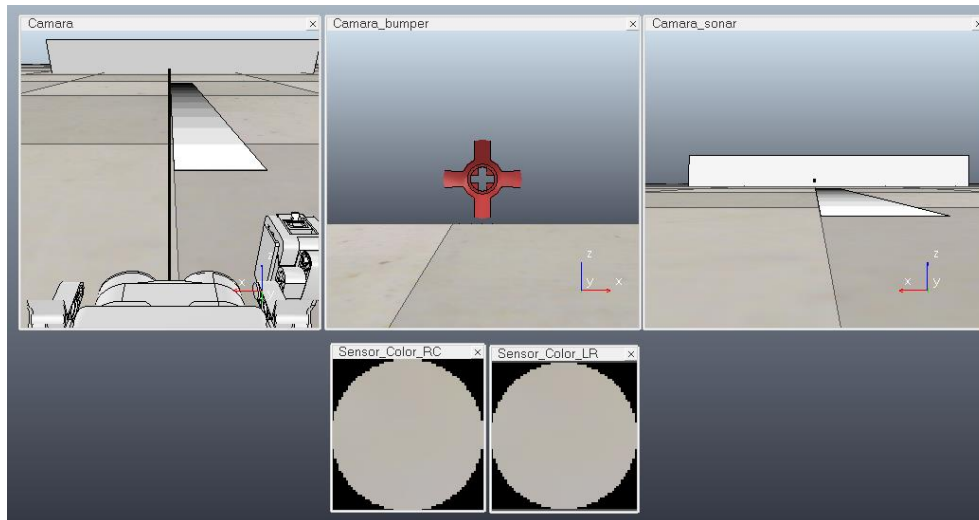


Figura 111. Vistas de las cámaras y sensores de visión en vistas flotantes

## A.6. Escenas

Además, se proporcionan escenas, que pueden servir de apoyo para el alumno y personal docente a la hora de desarrollar las prácticas; estas son:

- **“escenaTest.ttt”**. Esta escena está indicada para probar toda la toolbox mediante los test que se encuentran en el directorio con su mismo nombre del paquete de instalación proporcionado.
- **“sigueLineas.ttt”**. Contiene caminos ya trazados para diseñar un algoritmo de seguimiento de líneas para el robot (esta escena suele tardar en cargar).
- **“localizacionOffline.ttt”**. En esta escena, el robot ya está posicionado para realizar prácticas de localización.

## A.7. Ejecutar un script

Para ejecutar un script ya programado, se dispone de la función “ejecutarCodigoNXC” que se explicará en el anexo posterior (Anexo B) o aplicando el comando “help” en Matlab para dicha función.



## Anexo B. Manual del desarrollador

En el presente anexo se listarán y clasificarán todas las funciones disponibles para la toolbox, así como los objetos manejadores, scripts y la función para iniciar una simulación. Para saber cómo se utiliza cada función, estas disponen de ayuda escribiendo el comando de Matlab “help” más la función, script o clase a utilizar; por ejemplo: “help OnFwd”.

### B.1. Funciones de control del robot

En esta sección del anexo se listarán, por categorías, todas las funciones que la toolbox posee para el control del robot; estas son:

- Funciones de actuadores:
  - Off
  - OnFwd
  - OnRev
- Funciones de inicio de sensores:
  - SetSensorHtGyro
  - SetSensorLight
  - SetSensorTouch
  - SetSensorUltrasonic
- Funciones de sensores (entre paréntesis la función de inicio de sensor que se tiene que utilizar previamente para definir el puerto de entrada que se va a utilizar para este):
  - MotorRotationCount
  - SENSOR\_1 (SetSensorLight o SetSensorTouch)
  - SENSOR\_2 (SetSensorLight o SetSensorTouch)
  - SENSOR\_3 (SetSensorLight o SetSensorTouch)
  - SENSOR\_4 (SetSensorLight o SetSensorTouch)
  - Sensor (SetSensorLight o SetSensorTouch)
  - SensorHtGyro (SetSensorHtGyro)
  - SensorUS (SetSensorUltrasonic)
  - ResetRotationCount
- Funciones de la interfaz gráfica:
  - ButtonPressed
  - ClearScreen
  - NumOut
  - TextOut
- Funciones de manejo de ficheros:
  - CloseFile
  - CreateFile

- DeleteFile
  - OpenFileRead
  - ReadLnString
  - WriteLnString
- Otras funciones:
  - CurrentTick
  - FreeMemory
  - Stop
  - Wait
- Funciones que no pertenecen al repertorio de funciones de NXC (entre paréntesis una breve descripción de esta):
  - ResetAngle (reseteo del modo de conteo de ángulos en el sensor giroscópico)
  - SensorAngle (devuelve el ángulo del modo conteo de ángulos en el sensor giroscópico)
  - SensorColor (devuelve el código de color leído por el sensor de color)
  - StatusLight (apaga o enciende las luces de estado de la interfaz con el color especificado y si estas parpadean)
- Funciones privadas:
  - privateSensor
  - isFileNXC

### B.2. Clases manejadoras del cliente

Estos objetos se encargan de la comunicación y la ejecución de funciones en V-REP para el robot y el control de la simulación; pueden ser usadas si se quiere desarrollar otra toolbox con otro tipo de estructura de funciones o ampliar la desarrollada. En la Figura 112 se detalla la estructura de las clases que implementan estos objetos y su contenido (atributos y métodos). Para más información de cada función, utilizar el comando “help” o generar toda la documentación del objeto con “doc”, por ejemplo: “doc ev3RemoteApi”.

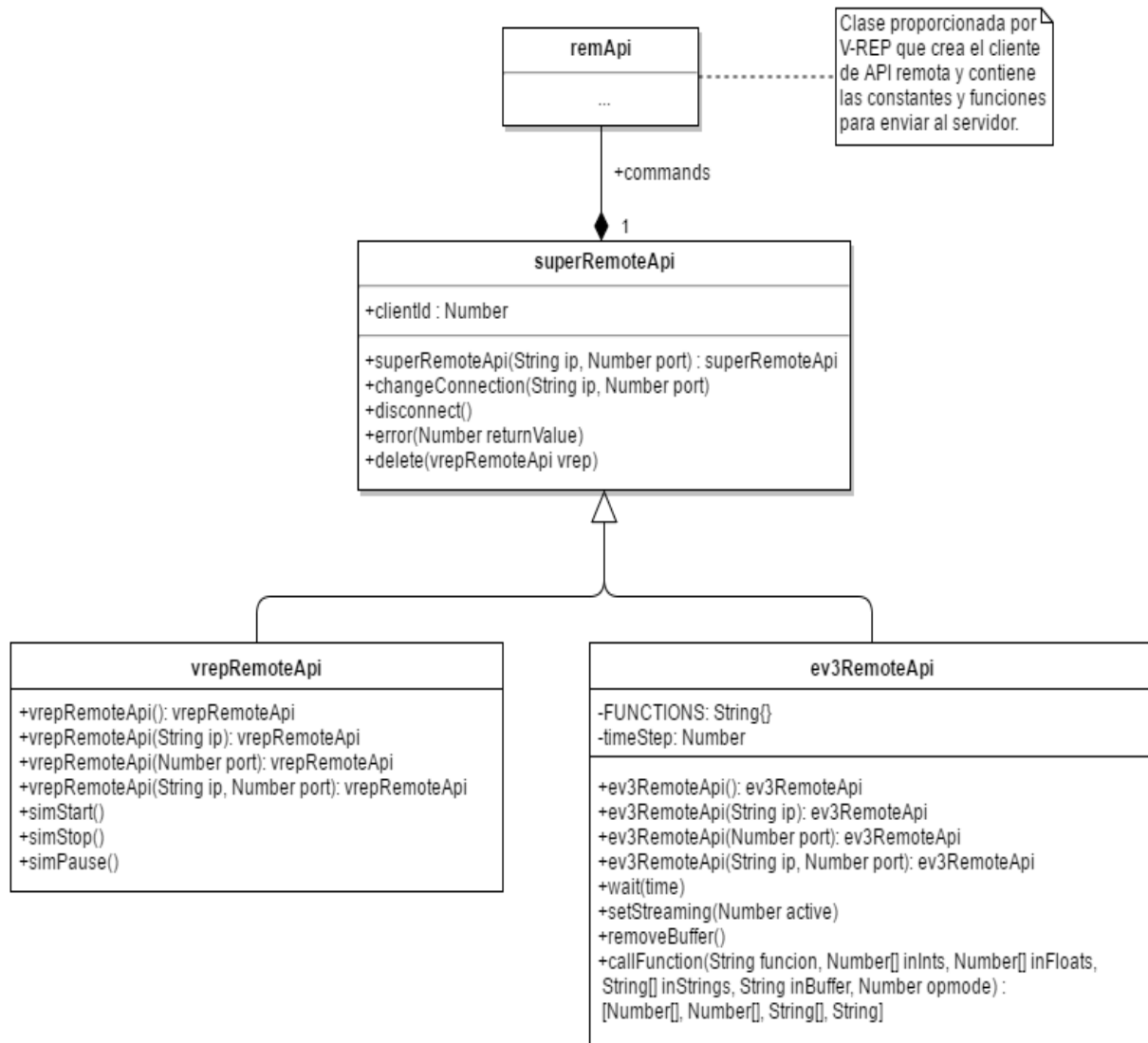


Figura 112. Diagrama de clases de los objetos manejadores de la toolbox

Si se quiere crear una instancia de “vrepRemoteApi”, el simulador debe de estar iniciado; por otro lado, si se desea crear una instancia de “ev3RemoteApi”, además de tener V-REP iniciado, el robot tiene que estar cargado en la escena y la simulación iniciada.

### B.3. Función EjecutarCodigoNXC

Para poder usar toda esta toolbox sin necesidad de preocuparse por la inicialización de las constantes y la conexión con el simulador, se ha creado una función que oculta este proceso y, además, permite el uso de logs para facilitar la labor de depuración. También se encarga de manejar las excepciones para que, en caso de error, la conexión con V-REP no se quede abierta y el espacio de trabajo quede limpio. Para poder usar esta función tenemos las siguientes llamadas:

- **“ejecutarCodigoNXC(String script)”**. Ejecuta el programa de NXC con el nombre “script”. También, se permite la escritura como comando de Matlab: “ejecutarCodigoEV3 script”.
- **“ejecutarCodigoNXC(String script, Bool log)”**. Ejecuta el programa de NXC con el nombre “script” y, si “log” es true o 1, guarda en un fichero de texto los mensajes escritos en la consola de Matlab con el nombre “script”.
- **“ejecutarCodigoNXC(String script, Bool log, String fichero)”**. Igual que el anterior, solo que, el archivo de log se guarda con el nombre “fichero”.