



# **ESCUELA SUPERIOR DE INGENIERÍA**

## **INGENIERÍA INFORMÁTICA**

### **RobotUI**

**Asistente de diseño de interfaz de control, seguimiento  
y sharing de robots en tiempo real**

Manuel López Urbina  
Director: Arturo Morgado Estévez

Cádiz, 26 de junio de 2017



# Índice general

Índice general	I
Índice de figuras	III
<b>1. Guía de Usuario</b>	<b>1</b>
1.0.1. Objetivo de esta guía . . . . .	2
1.0.2. Dirigido a . . . . .	2
1.0.3. Obtener RobotUI . . . . .	2
1.1. Uso de RobotUI . . . . .	2
1.1.1. Acceso a la aplicación . . . . .	2
1.1.2. Registro de usuario . . . . .	3
1.1.3. Ingreso al sistema . . . . .	5
1.1.4. Registro de un Robot . . . . .	5
1.1.5. Gestión de permisos . . . . .	7
1.1.6. Configuración de la interfaz . . . . .	8
1.1.7. Ejemplo . . . . .	11
1.1.8. Programa tu robot . . . . .	12
1.1.9. Control de robots . . . . .	16
1.1.10. Seguimiento de robots . . . . .	18
1.1.11. Mensajes . . . . .	20
1.1.12. Panel de administración . . . . .	21



# Índice de figuras

1.1. Página principal RobotUI. . . . .	3
1.2. Vista de la barra superior en modalidad usuario no identificado en el sistema. . . . .	3
1.3. Vista de la barra superior en modalidad usuario identificado en el sistema. . . . .	3
1.4. Formulario de registro de usuario. . . . .	4
1.5. Información de un usuario. . . . .	4
1.6. Formulario de creación de un robot. . . . .	6
1.7. Vista informativa de un robot. . . . .	6
1.8. Panel de configuración de los permisos de un robot. . . . .	8
1.9. Panel informativo de un robot privado frente a uno público. . . . .	8
1.10. Panel de configuración de la interfaz sin elementos añadidos. . . . .	9
1.11. Panel de elemento (izquierda) y panel de acciones (derecha) para la configuración de la interfaz. . . . .	10
1.12. Formulario para la creación de un botón. . . . .	11
1.13. Vista tipo de una interfaz configurada. . . . .	12
1.14. Panel informativo de un robot donde se aprecia su identificador. . . . .	13
1.15. Robot a la espera de conexión entrante. . . . .	16
1.16. Índice robots públicos. . . . .	17
1.17. Vista del panel de control de un robot junto con el listado de usuarios realizando el seguimiento. . . . .	17
1.18. Panel informativo de las características del robot de la interfaz. . . . .	18
1.19. Histórico de las órdenes enviadas al robot junto con una entrada de comandos para enviar nuevas órdenes directamente junto con el listado de usuarios presentes en la sesión de control. . . . .	18
1.20. Panel informativo de un robot en su modalidad <i>ocupado</i> . . . . .	19
1.21. Vista del panel de seguimiento de un robot en tiempo real. . . . .	19
1.22. Vista del panel de control de un robot junto con el listado de usuarios realizando el seguimiento. . . . .	20
1.23. Formulario de redacción de un mensaje. . . . .	21
1.24. Paneles de administración para robots y usuarios. . . . .	22



# Capítulo 1

## Guía de Usuario

La aplicación RobotUI ha sido creada por Manuel López Urbina para el proyecto fin de carrera de la titulación de Ingeniería Informática de la Universidad de Cádiz.

Resulta de vital importancia consultar esta guía antes y/o durante la utilización de los diferentes elementos tanto hardware, robot de pruebas desarrollado, como software del presente proyecto ya que le proporcionará una guía paso a paso en el manejo correcto de la aplicación.

La resolución recomendada para la aplicación debe ser superior o igual a 1024x768 (Estándar XGA), aunque es adaptable a cualquier resolución debido a su diseño web responsive.

RobotUI ha sido elaborado con un claro propósito; el de proporcionar a los usuarios un medio donde compartir sus dispositivos robóticos con el resto de usuarios. Esto es posible gracias a una serie de herramientas desarrolladas para que, sin necesidad de tener grandes conocimientos en programación, puedan configurar un entorno para el manejo de sus proyectos robóticos y tener posibilidad de compartir sus dispositivos y experiencias con el resto de usuarios.

La particularidad de RobotUI es que el usuario propietario del robot tiene la posibilidad de permitir el manejo de sus dispositivos robóticos al resto de usuarios que él mismo considere de una manera controlada o, por otra parte, permitir que otros usuarios visualicen, como si de espectadores se tratase, el control que un determinado usuario realiza de un determinado robot. Todo ello en tiempo real.

Por tanto, tras esta breve introducción en el ámbito de la aplicación, en este manual se describen los diferentes pasos a realizar para configurar sus dispositivos correctamente en el sistema y abrirlo a toda una comunidad de usuarios. Además de tener abierto el acceso a otros muchos dispositivos de otras personas.

### 1.0.1. Objetivo de esta guía

Esta guía tiene como objetivo proporcionar al usuario un soporte de ayuda e iniciación a la utilización de RobotUI.

Esta sección comprende:

- Introducción.
- Guía de acceso al código fuente de la aplicación.
- Guía de uso de la aplicación.
- Guía para la puesta en marcha y programación de un robot.

### 1.0.2. Dirigido a

Esta guía esta dirigida al usuario final del proyecto RobotUI. Tiene la finalidad de proporcionar una guía descriptiva de los procedimientos de creación, configuración y utilización de los diferentes dispositivos robóticos en sus dos modalidades disponibles, la de control y la de visualización.

### 1.0.3. Obtener RobotUI

El código fuente junto con la presente memoria se encuentra disponible en el repositorio GitHub en el enlace <https://github.com/lopi87/SAILS-RobotUI> o usando la herramienta Git, escribiendo en la consola el siguiente comando:

```
1 git clone git@github.com:lopi87/SAILS-RobotUI.git
```

## 1.1. Uso de RobotUI

### 1.1.1. Acceso a la aplicación

Para acceder a la aplicación, el usuario deberá acceder al siguiente enlace: [www.robotui.es](http://www.robotui.es).

Al acceder podrá ver el portal de entrada a la aplicación. En él puede acceder al resto de funcionalidades identificándose con sus credenciales y acceder a los formularios de registro de usuario.



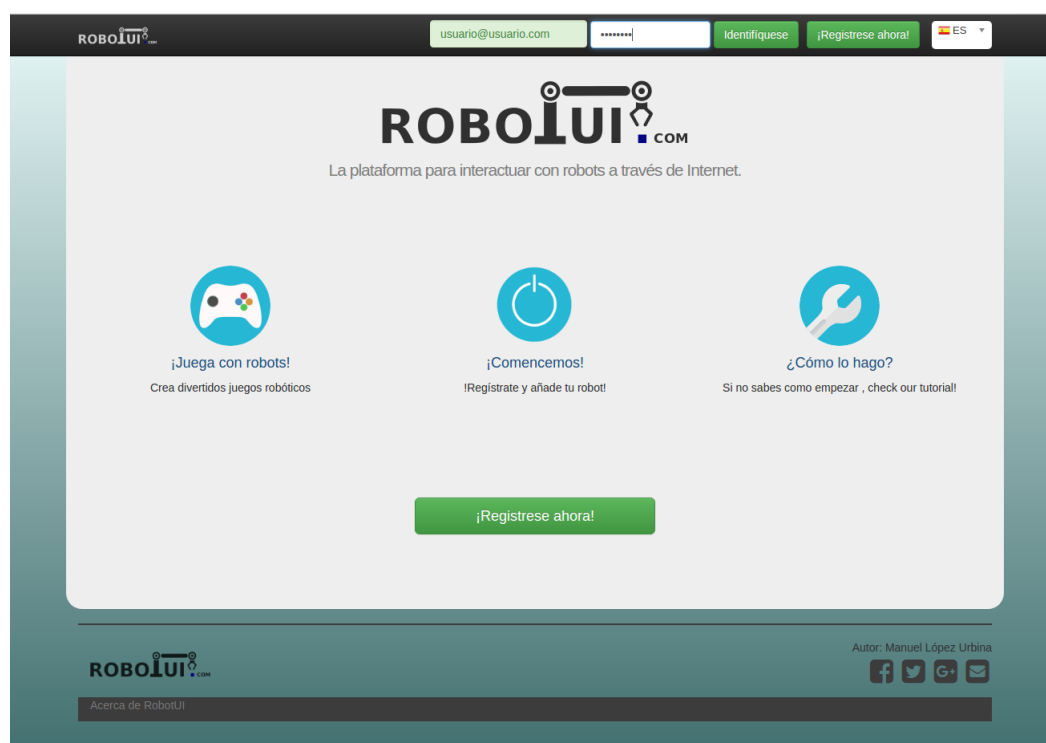


Figura 1.1: Página principal RobotUI.

### 1.1.2. Registro de usuario

Una vez en la página principal, figura 1.18, podemos observar la barra superior, la cual posee dos estados, la de usuario sin identificar y la de usuario identificado.

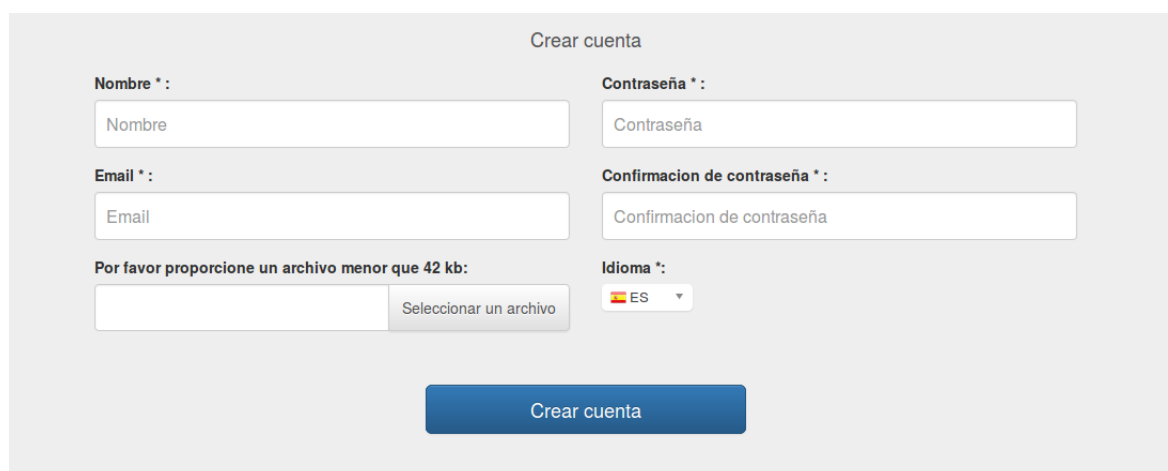


Figura 1.2: Vista de la barra superior en modalidad usuario no identificado en el sistema.



Figura 1.3: Vista de la barra superior en modalidad usuario identificado en el sistema.

Para iniciar el proceso de registro de usuario hacemos clic sobre el botón de la barra superior [¡Regístrese ahora!](#) mediante el cual accedemos al formulario 1.4.



Crear cuenta

Nombre \* :

Contraseña \* :

Email \* :

Confirmacion de contraseña \* :

Por favor proporcione un archivo menor que 42 kb:  [Seleccionar un archivo](#)

Idioma \* : ES

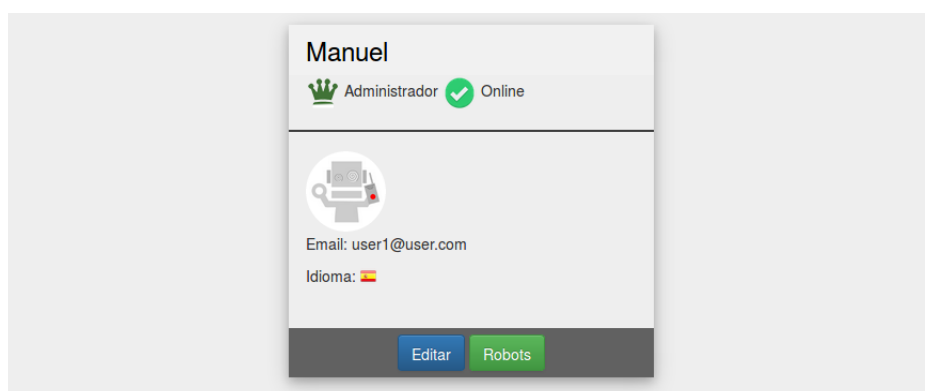
[Crear cuenta](#)

Figura 1.4: Formulario de registro de usuario.



Para proceder con el registro necesitaremos los siguientes datos:


1. Nombre: nombre de usuario.
2. Email: dirección de correo electrónico utilizada para acceder al sistema.
3. Contraseña: deberá proporcionar una contraseña para poder acceder al sistema.
4. Confirmación de contraseña: campo para comprobar que la contraseña es la deseada por el usuario.
5. Imagen: imagen avatar del usuario.
6. Idioma: selección del idioma por defecto del usuario.

Una vez introducidos los datos correctamente podremos hacer clic en *crear cuenta*. Si los datos introducidos son correctos dispondremos de un nuevo usuario en la aplicación rediriéndonos a la página de información del nuevo usuario [1.5](#).



Manuel

 Administrador  Online




Email: user1@user.com

Idioma: ES

[Editar](#) [Robots](#)

Figura 1.5: Información de un usuario.

### 1.1.3. Ingreso al sistema

Una vez registrados, procedimiento descrito en 1.1.2, podremos acceder al sistema mediante la introducción de nuestro usuario y contraseña en los campos disponibles en la barra superior localizado en la aplicación y haciendo clic en  localizados en la barra superior.

### 1.1.4. Registro de un Robot

Para el iniciar el proceso de creación de un robot haremos clic en *Robots* → *Mis robots* del menú superior accediendo al formulario de creación, figura 1.8.

En este caso registraremos nuestro robot de pruebas descrito en el capítulo ??. Para ello introducimos los siguientes datos:

1. **Nombre:** nombre de nuestro robot.
2. **Dirección IP:** dirección donde se encontrará accesible nuestro robot.
3. **Puerto:** puerto donde se encontrará nuestro robot a la escucha de conexiones.
4. **Descripción:** campo opcional en el que podemos añadir una descripción de nuestro robot y que será visible al resto de usuarios.
5. **Imagen:** campo opcional en el que podremos subir una imagen de nuestro robot.
6. **Usuarios espectadores:** selector en el que podemos especificar qué usuarios del sistema tendrán permisos para visualizar el funcionamiento de nuestro robot.
7. **Usuarios controladores:** selector para especificar qué usuarios podrán tomar control de nuestro robot.
8. **Público:** checkboxes para indicar si por el contrario, el robot que abierto a todos los usuarios para su control o abierto para su visualización si marcamos el primero o el segundo checkbox respectivamente y haciendo por tanto inválidos los parámetros de los selectores anteriores.

Una vez introducidos los datos para el nuevo usuario pulsamos en .

The screenshot shows the 'Robot' creation form in the ROBOTUI interface. The form is titled 'Robot' and contains several input fields and buttons. The fields are: 'Nombre \*' (Name) with the value 'RC Car', 'Dirección IP\*' (IP Address) with the value '80.30.240.XXX', 'Puerto\*' (Port) with the value '4040', and 'Descripción:' (Description) with the value 'Vehiculo de pruebas RobotUI'. There is also a field for 'Por favor proporcione un archivo menor que 42 kb:' (Please provide a file smaller than 42 kb) with the value 'electric-buggy.jpg' and buttons 'Cambiar' (Change) and 'Eliminar' (Delete). Below this are two dropdown menus for 'Usuarios espectadores:' (Spectator users) and 'Usuarios controladores:' (Controller users), both with the placeholder 'Seleccione usuarios'. At the bottom right, there is a 'Público \*' (Public) section with 'Control:' and 'Ver:' (View) checkboxes, both of which are checked. A large blue 'Crear' (Create) button is at the bottom center. The footer of the page includes the ROBOTUI logo, the text 'Acerca de RobotUI', and the author's name 'Autor: Manuel López Urbina' along with social media icons for Facebook, Twitter, Google+, and Email.

Figura 1.6: Formulario de creación de un robot.

Si los campos introducidos son correctos la aplicación nos redireccionará a la página informativa de nuestro nuevo robot:

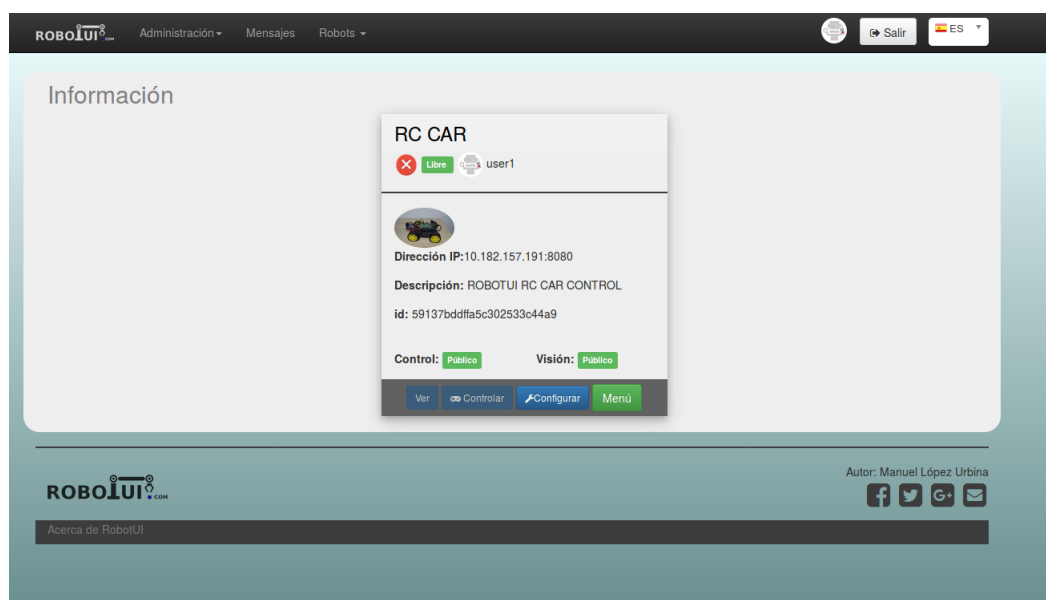


Figura 1.7: Vista informativa de un robot.

En la ventana informativa de un robot, figura 1.7, podemos apreciar una serie de elementos representativos de los diferentes estados en los que se puede encontrar el dis-

positivo. La tabla 1.1 describe cada uno de los elementos:









Elemento	Significado
	Imagen indicativa de estado <i>offline</i> . El robot se encuentra sin conexión.
	Imagen indicativa de estado <i>online</i> . El robot se encuentra conectado, reemplazará a la imagen de estado <i>offline</i> .
Control: 	Imagen representativa de robot de control público.
Control: 	Imagen representativa de robot de control privado.
Seguimiento: 	Imagen representativa de robot de seguimiento público.
Seguimiento: 	Imagen representativa de robot de seguimiento privado.
	Imagen representativa de robot en disponible.
	Imagen representativa de robot en uso.

Tabla 1.1: Elementos representativos de los estados de un robot junto con su descripción.

### 1.1.5. Gestión de permisos

Como hemos mencionado en el procedimiento de registro de un robot, éste se puede crear con una serie de permisos. Existen dos modalidades en RobotUI, como sabemos, la de visualización o seguimiento y la de control. Funcionalidades que podemos hacer públicas o privadas, y, en el caso de que optemos por alguna modalidad privada, definir para qué usuarios estará disponible.






La gestión y visualización de los permisos configurados estarán disponibles en todo momento en la ventana de edición de un robot. En ella podremos visualizar los diferentes usuarios añadidos con los permisos de cada uno de ellos. Podremos eliminar usuarios o añadir otros nuevos estableciendo el acceso a alguna de las modalidades existentes como pública o privada.

Si el robot es establecido como público en alguna de sus dos modalidades (control o seguimiento) entonces se ignorará toda la configuración de usuarios establecida para esa modalidad aunque permanecerán registradas en el sistema por si se desea volver al modo privado tan solo haciendo clic en el checkbox correspondiente.

**Permisos**


**Público \* :**


Control: ☐ Seguimiento: ☐

Usuario	Nombre	Control	Seguimiento	Acciones
	user2			<button>Eliminar</button>
	user3			<button>Eliminar</button>

---

**Usuarios \* :**



 user2

 user4


Añadir

Figura 1.8: Panel de configuración de los permisos de un robot.

**RC CAR**

 Libre  user1

---





Dirección IP: 10.182.157.191:8080


Descripción: ROBOTUI RC CAR CONTROL

id: 59137bddffa5c302533c44a9

Control: Privado Seguimiento: Privado



 user2

 user2


 user3

Seguir
Controlar
Configurar
Menú

**RC CAR**

 Libre  user1

---



Dirección IP: 10.182.157.191:8080

Descripción: ROBOTUI RC CAR CONTROL

id: 59137bddffa5c302533c44a9

Control: Público Seguimiento: Público

Seguir
Controlar
Configurar
Menú

Figura 1.9: Panel informativo de un robot privado frente a uno público.

### 1.1.6. Configuración de la interfaz

Una vez creado nuestro robot en el sistema ya podremos configurar su interfaz. En la figura 1.10 podemos ver el canvas sobre el que iremos añadiendo los diferentes ele-

mentos que conformarán nuestro panel de control.

El diseño resultante de esta ventana servirá tanto para panel de control del robot como ventana de seguimiento a los usuarios que sigan el funcionamiento del dispositivo.

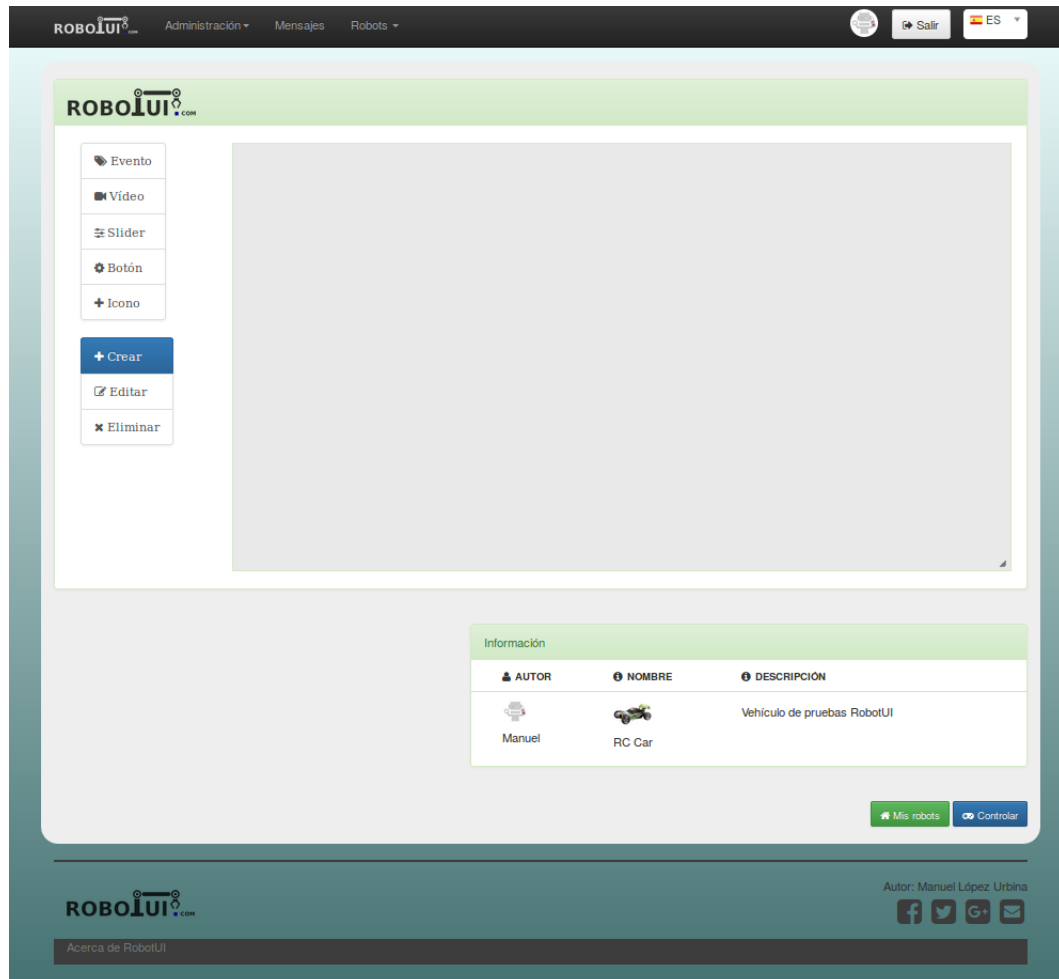


Figura 1.10: Panel de configuración de la interfaz sin elementos añadidos.

La figura 1.10 correspondiente al panel de configuración de una interfaz queda dividida en los siguientes elementos:

1. Panel de elementos disponibles (Evento, Vídeo, Slider, Botón e Icono).
2. Panel de acciones sobre elementos (Crear, editar y eliminar).
3. Panel canvas sobre el que posicionar los elementos.
4. Panel informativo de las características del robot.

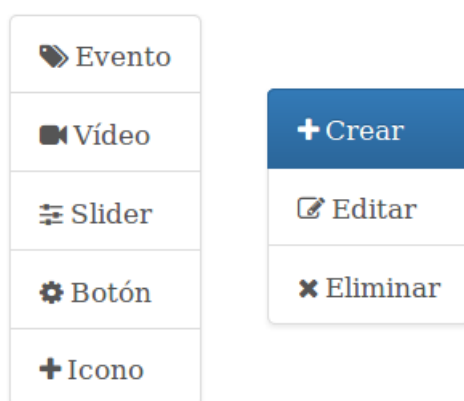


Figura 1.11: Panel de elemento (izquierda) y panel de acciones (derecha) para la configuración de la interfaz.

En los siguientes subapartados se describen los procedimientos para la creación, edición y borrados de elementos:

### Nuevo elemento

Para la creación de elementos debemos tener seleccionado el modo creación en el panel de acciones y pulsar sobre el tipo de nuevo elemento deseado. La siguiente tabla muestra las diferentes combinaciones:











Botón	Modo	Acción
 Botón		Apertura del formulario para la creación de un botón.
 Vídeo		Apertura del formulario para la creación de un canvas de vídeo.
 Slider		Apertura del formulario para la creación de un slider.
 Evento		Apertura del formulario para la creación de una etiqueta.
 Icono		Apertura del formulario para la subida de un icono.

Tabla 1.2: Elementos añadibles a una interfaz.

Una vez completado el formulario y creado el nuevo elemento tan solo debemos arrastrar el nuevo elemento a la posición deseada.



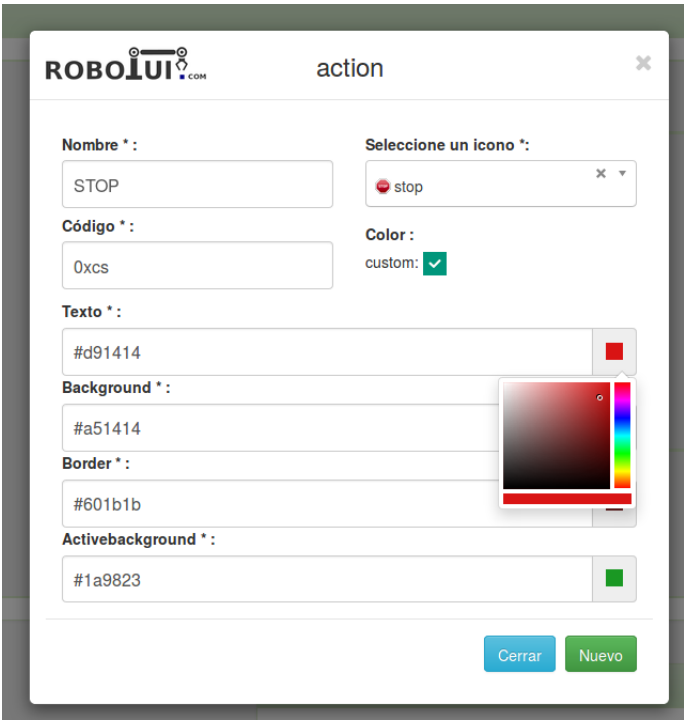

The image shows a web form titled 'action' with the ROBOTUI logo. It contains several input fields: 'Nombre \*' with the value 'STOP', 'Código \*' with the value '0xcs', 'Texto \*' with the value '#d91414', 'Background \*' with the value '#a51414', 'Border \*' with the value '#601b1b', and 'Activebackground \*' with the value '#1a9823'. There is a 'Seleccione un icono \*' dropdown menu showing a red stop icon and the text 'stop'. A 'Color:' section has a 'custom:' checkbox checked. A color picker is open over the 'Background' field, showing a red color. At the bottom, there are two buttons: 'Cerrar' (blue) and 'Nuevo' (green).

Figura 1.12: Formulario para la creación de un botón.


Del mismo modo para la creación del resto de tipos de elementos.

### Editar elemento

Para la edición de elementos pulsaremos seleccionaremos el modo edición  y posteriormente haremos clic sobre el elemento que deseemos editar.

Tras ello se abrirá el formulario correspondiente, modificamos los parámetros deseados y pulsamos guardar.

### Eliminar elemento

Del mismo modo que en la edición de elementos, para borrar un elemento seleccionaremos el modo eliminar  del panel de acciones y posteriormente haremos clic sobre el elemento que deseemos borrar.

### 1.1.7. Ejemplo

La figura 1.13 muestra una posible interfaz de control para un dispositivo robótico.



Figura 1.13: Vista tipo de una interfaz configurada.

### 1.1.8. Programa tu robot

NOTA: Esta guía comprende una serie de pasos para realizar la programación de un dispositivo robótico para la aplicación RobotUI. En este caso particular, el robot empleado posee como base una placa Raspberry Pi, la cual emplea para la conexión de sensores y motores haciendo uso de su sistema de Entrada/Salio Gpio.

El sistema es compatible con cualquier dispositivo, no solo limitado a las placas Raspberry. La única diferencia radica en que se deberán emplear a nivel de programación las bibliotecas adecuadas para la activación o lectura de las Entradas/Salidas correspondientes al modelo de placa utilizado.

Antes de proceder con la programación de nuestro robot debemos de realizar su creación en la aplicación RobotUI como queda descrito en el punto 1.1.4 y tomar nota del identificador único que la aplicación proporciona para dicho robot y que posteriormente necesitaremos.

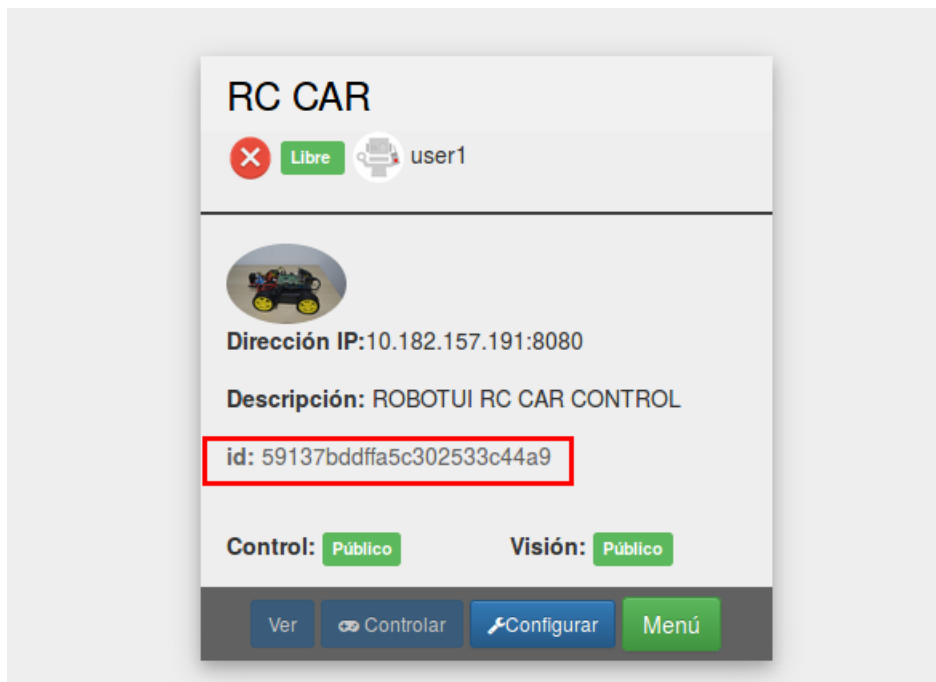


Figura 1.14: Panel informativo de un robot donde se aprecia su identificador.

Una vez creado el robot y configurada su interfaz, añadiendo los diferentes elementos disponibles como acciones, vídeo, etiquetas, etcétera, recogidos en el punto 1.1.6: Configuración de la interfaz. Podemos comenzar con la programación del robot.

Para ello debemos seguir una serie de pasos:

1. Generar un archivo de extensión .js, con el código base mostrado en el punto 1.1.8.
2. Reemplazar en el código la palabra *IDENTIFICADOR* por el identificador único de nuestro robot obtenido en la vista informativa del mismo. Por ejemplo: 59188631c8e94ba54f7a4bdc.
3. Indicar el puerto en el que el robot permanecerá a la escucha. Para ello debemos reemplazar la palabra *PUERTO* del código inferior por el puerto deseado. Por ejemplo 8085.
4. El siguiente paso es determinar qué puertos GPIO necesitaremos y si van a ser utilizados en modo Entrada, Salida o Entrada/Salida dentro de la sección *PINES* de la plantilla. En este caso se ha empleado la biblioteca *pigpio* cuya documentación se encuentra disponible en el siguiente enlace: <https://www.npmjs.com/package/pigpio>.
5. Definir el funcionamiento de los diferentes eventos dentro de la sección *EVENTOS*.

**código base**

```
1
2 var io_client = require('./node_modules/socket.io-client');
3 var sails_client = require('./node_modules/sails.io.js');
4 var io_server = sails_client(io_client);
5 io_server.sails.url = 'http://46.101.102.33:80';
6 io_server.socket.get('/robot/changetoonline/', {robot:
  'IDENTIFICADOR', online: true});
7
8 // Inicia servidor socket.io en el puerto PUERTO.
9 var io =io_client.listen( PUERTO, { log: false });
10
11 var ffmpeg_command, running_camera = false, child_process =
  require('child_process');
12
13 var Gpio = require('pigpio').Gpio;
14
15
16 // SECCION PINES
17
18 // EJEMPLO:
19 //     var gpio2 = new Gpio(2, {mode: Gpio.OUTPUT}),
20 //     gpio3 = new Gpio(3, {mode: Gpio.OUTPUT}),
21 //     gpio17 = new Gpio(17, {mode: Gpio.OUTPUT}),
22 //     gpio27 = new Gpio(27, {mode: Gpio.OUTPUT});
23
24
25 // FIN SECCION PINES
26
27
28 console.log('Esperando conexi n...');
29
30 var sockets = {};
31
32 io.sockets.on('connection', function (socket)
33 {
34
35     // SECCION EVENTOS
36
37     // FIN SECCION EVENTOS
38 });
39
40 function stopStreaming(socket) {
41     delete sockets[socket.id];
42     // no more sockets, kill the stream
43     if (Object.keys(sockets).length == 0) {
44         if (ffmpeg_command){
45             ffmpeg_command.kill();
46             running_camera = false;
47             console.log('Stop streaming');
48         }
49     }
50 }
51
```

```

52 function startStreaming(socket) {
53   if (running_camera == false){
54     console.log('Starting streaming...');
55     var args = ["-f", "video4linux2", "-i", "/dev/video0", "-s",
56               "300x150", "-f", "mjpeg", "pipe:1", "-b:v 28k", "-bufsize 28k"]
57     ffmpeg_command = child_process.spawn("ffmpeg", args);
58     running_camera = true
59   }
60   ffmpeg_command.on('error', function(err, stdout, stderr) {
61     console.log("ffmpeg stdout:\n" + stdout);
62     console.log("ffmpeg stderr:\n" + stderr);
63     running_camera = false
64   });
65
66
67   ffmpeg_command.on('close', function (code) {
68     console.log('ffmpeg exited' + code );
69     running_camera = false
70   });
71
72
73   ffmpeg_command.stderr.on('data', function (data) {
74     //console.log('stderr: ' + data);
75   });
76
77   ffmpeg_command.on('end', function() {
78     console.log('Fin');
79     running_camera = false
80   });
81
82   ffmpeg_command.stdout.on('data', function (data) {
83     //console.log('stdout: ' + data);
84     var frame = new Buffer(data).toString('base64');
85     socket.emit('canvas', frame);
86   });
87
88 }

```

A continuación mostramos dos posibles eventos para añadir al código base superior a modo orientativo:

El primer fragmento de código se activa al recibir un evento tipo *action* (evento lanzado desde la interfaz al presionar cualquier botón generado por el usuario), captura el comando recibido, y si es igual a *UP*, entonces habilita el pin *gpio1* con el valor 1. Pin inicializado previamente en la sección de pines del código base.

```

1
2 socket.on('action', function (data){
3   if (data == 'UP') {
4     gpio1.digitalWrite(1);
5   }
6 });

```

El segundo fragmento de código devuelve la lectura del pin *gpio2* cada vez que recibe el comando *READ* y mandando al cliente el valor de lectura obtenido:

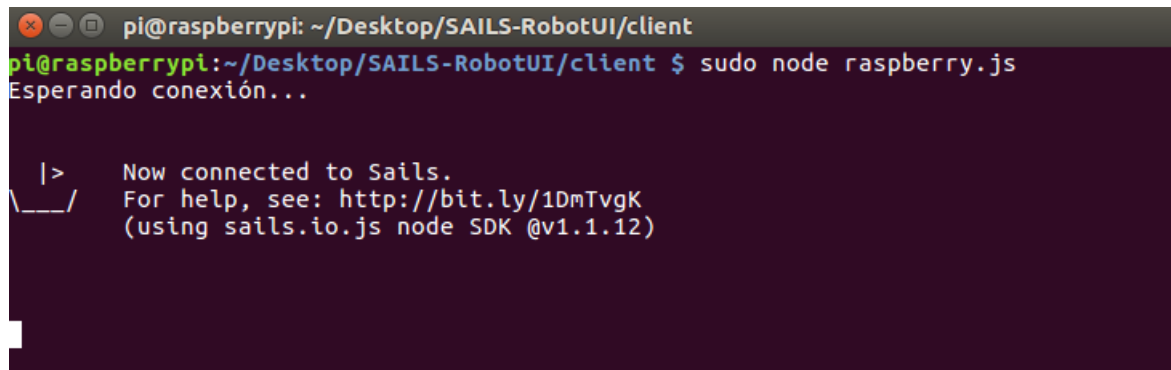
```
1
2 socket.on('action', function (data){
3     if (data == 'READ') {
4         var temp = gpio2.digitalRead(1);
5         socket.emit('robot_temp', {msg: temp});
6     }
7 });
```

En la parte referente a la interfaz de control de la aplicación RobotUI, para lanzar o capturar los comandos correspondientes a los ejemplos superiores, en el primer caso, debemos crear un botón cuyo código a emitir sea *UP* y para el segundo, debemos añadir un botón cuyo código de emisión sea *READ* y una etiqueta cuyo nombre de evento se corresponda con *robot\_temp*.

Una vez generado el código para nuestro robot debemos copiarlo a la placa Raspberry Pi o computador que actuará como Robot para nuestra aplicación y ejecutarlo. Para la ejecución del código introducimos el siguiente comando:

```
1 sudo node raspberry.js
```

Siendo *raspberry.js* el nombre del archivo que contiene nuestro código. Obteniendo el siguiente resultado:




```
pi@raspberrypi: ~/Desktop/SAILS-RobotUI/client
pi@raspberrypi:~/Desktop/SAILS-RobotUI/client $ sudo node raspberry.js
Esperando conexión...

|> Now connected to Sails.
|___/ For help, see: http://bit.ly/1DmTvgK
      (using sails.io.js node SDK @v1.1.12)
```

Figura 1.15: Robot a la espera de conexión entrante.

### 1.1.9. Control de robots

Para el acceso al control de un robot debemos localizar en el índice de robots públicos un robot con el estado *Online* y *Libre*. Ver figura 1.16.

Para acceder al robot pulsaremos sobre .

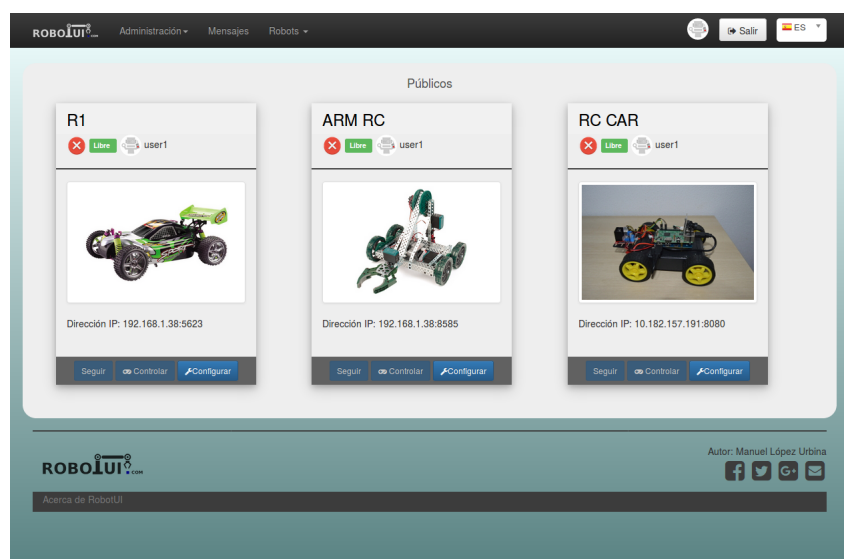


Figura 1.16: Índice robots públicos.

En la ventana de control [1.22](#) pulsamos sobre conectar. En ese momento tendremos plena disponibilidad para el control del robot haciendo uso de todas las acciones previamente configuradas en la interfaz.

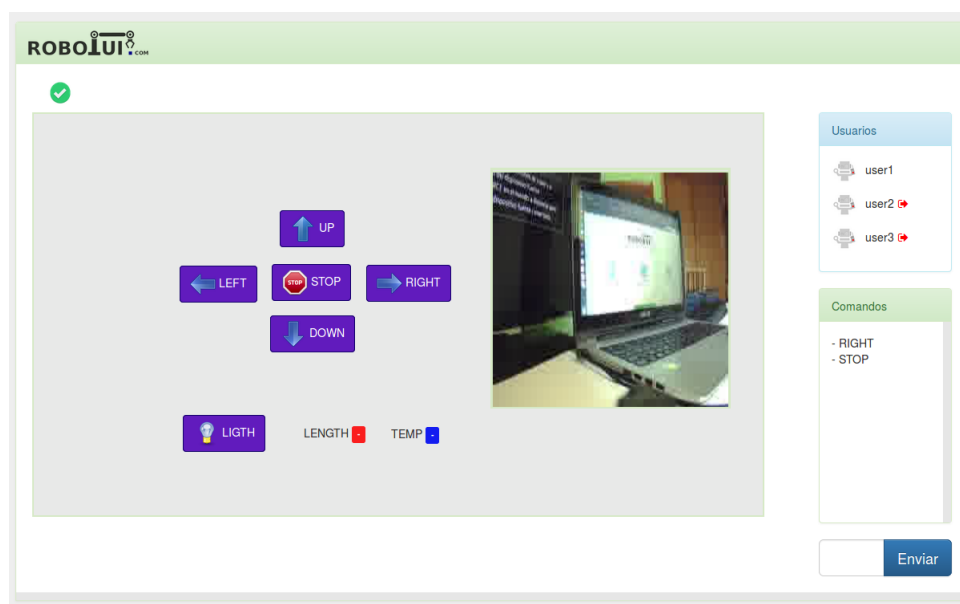


Figura 1.17: Vista del panel de control de un robot junto con el listado de usuarios realizando el seguimiento.

En el panel de control de la interfaz podemos identificar los siguientes elementos:



Información		
AUTOR	NOMBRE	DESCRIPCIÓN
 Manuel	 RC Car	Vehículo de pruebas RobotUI

Figura 1.18: Panel informativo de las características del robot de la interfaz.

Comandos

- STOP
- UP
- STOP
- UP
- STOP
- STOP
- UP
- STOP

Usuarios

-  user1
-  user2 
-  user3 

Enviar

Figura 1.19: Histórico de las órdenes enviadas al robot junto con una entrada de comandos para enviar nuevas órdenes directamente junto con el listado de usuarios presentes en la sesión de control.

### 1.1.10. Seguimiento de robots

Para realizar el seguimiento de un robot accedemos desde el menú superior en el desplegable *Robots* → *Públicos*. En la nueva ventana tendremos acceso a todos los robots públicos existentes pudiendo realizar el seguimiento en aquellos cuyo estado es *ocupado*.



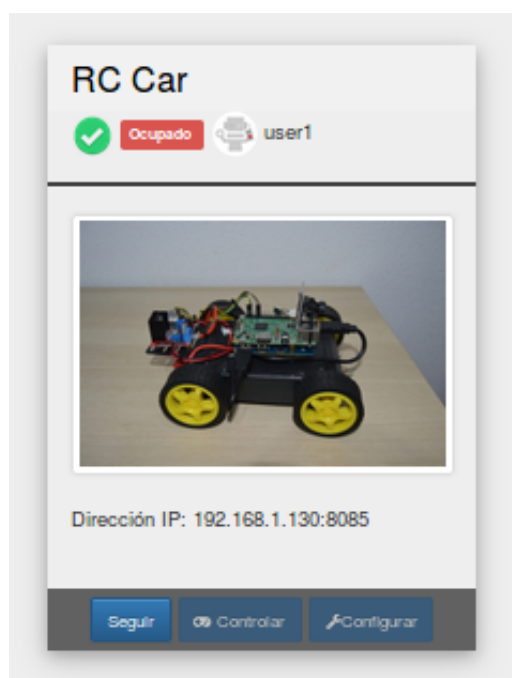


Figura 1.20: Panel informativo de un robot en su modalidad *ocupado*.

Para pulsamos sobre el botón *Seguir* que se encontrará habilitado accediendo a su interfaz en modo espectador. La página mostrada será de aspecto similar a la mostrada en la figura 1.21:

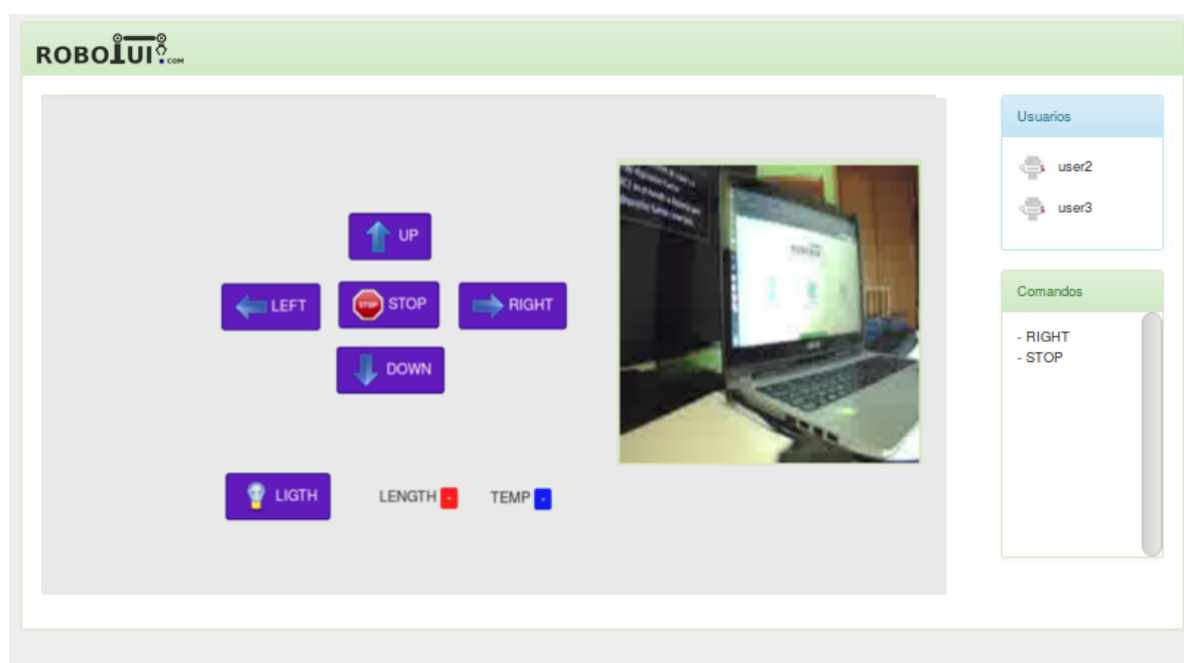


Figura 1.21: Vista del panel de seguimiento de un robot en tiempo real.

Mientras tanto, el usuario que dispone del control del robot será notificado de la entrada de un nuevo espectador. La figura 1.22 muestra la ventana de control de un robot junto con el listado de usuarios espectadores actualizado en todo momento.

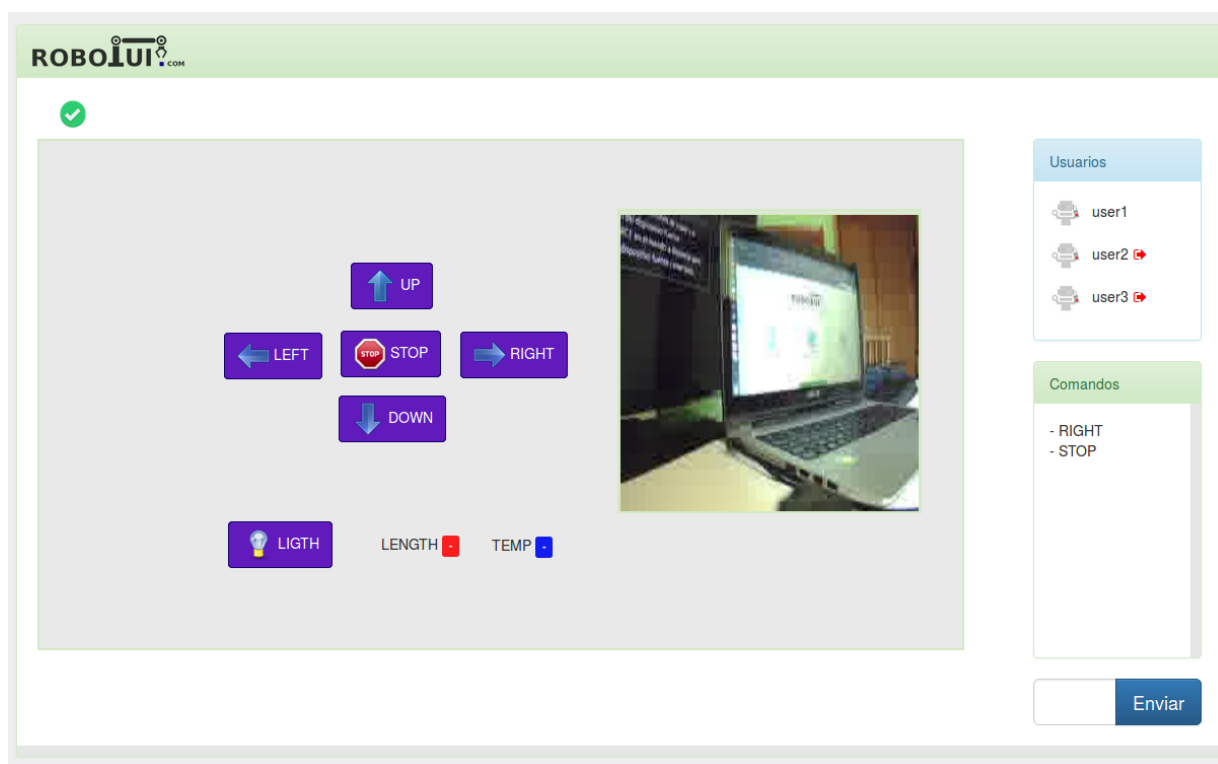


Figura 1.22: Vista del panel de control de un robot junto con el listado de usuarios realizando el seguimiento.

### 1.1.11. Mensajes

RobotUI incorpora un sistema de mensajería donde los usuarios de la aplicación se pueden enviar mensajes a otros usuarios y recibirlos. Para acceder a la página de mensajes pulsamos sobre *Mensajes* de barra superior.

En la página *Mensajes* tenemos acceso a la bandeja de entrada y los mensajes enviados.

Para mandar un mensaje pulsamos sobre **Redactar**, rellenamos el formulario con título, usuario destinatario y contenido del mensaje y pulsamos en **Nuevo**.

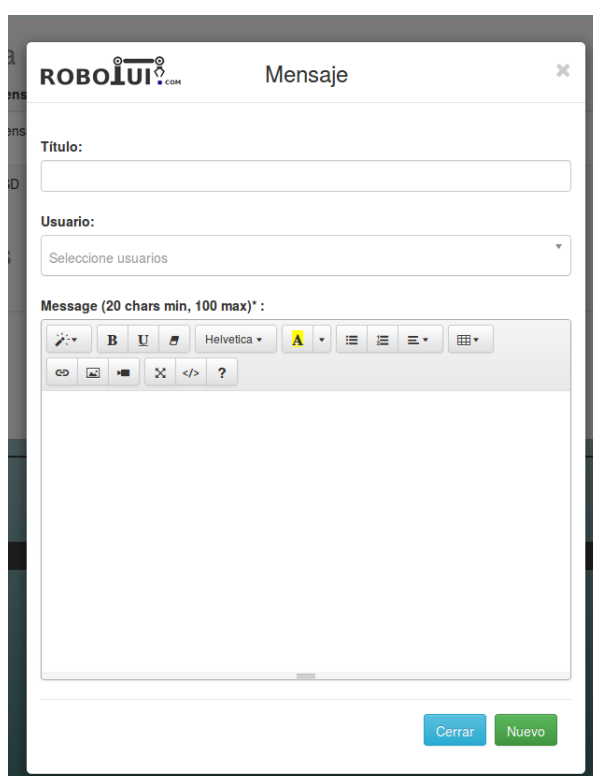


Figura 1.23: Formulario de redacción de un mensaje.

El campo de entrada de texto para la redacción del mensaje incorpora un editor de texto WYSIWYG <sup>1</sup> en el cual podremos dar formato y etilo al mensaje añadiendo imágenes, enlaces, etcétera.













### 1.1.12. Panel de administración

Entradas disponibles únicamente para usuarios administradores. Accesible desde la barra superior *Administración* → *Usuarios* o *Administración* → *Robots*.

En ella podremos gestionar todos los usuarios o robots existentes en el sistema, editarlos, borrarlos o crear nuevos.

---

<sup>1</sup> WYSIWYG es el acrónimo de What You See Is What You Get, (en español, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

Usuarios							
Online	Nombre	Email	Roll	Acciones			
	 user1	user1@user1.com		<a href="#">Añadir robot</a>	<a href="#">Mostrar robots</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>
	 user2	user2@user2.com		<a href="#">Añadir robot</a>	<a href="#">Mostrar robots</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>
	 user3	user3@user3.com		<a href="#">Añadir robot</a>	<a href="#">Mostrar robots</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>
	 user4	user4@user4.com		<a href="#">Añadir robot</a>	<a href="#">Mostrar robots</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>










Robots								
Online	Estado	Nombre	Descripción	direction	Propietario	Conducción	Visión	Acciones
	<a href="#">Libre</a>	 Buggy	Buggy	192.168.1.120:1234	user1 	<a href="#">Publico</a>	<a href="#">Publico</a>	<a href="#">Interfaz</a> <a href="#">Editar</a> <a href="#">Eliminar</a>
	<a href="#">Libre</a>	 ARM RC	ARM	10.182.157.191:4234	user1 	<a href="#">Privado</a>	<a href="#">Privado</a>	<a href="#">Interfaz</a> <a href="#">Editar</a> <a href="#">Eliminar</a>
	<a href="#">Libre</a>	 RC Car	RC ROBOTUI CAR	192.168.1.130:8085	user1 	<a href="#">Publico</a>	<a href="#">Publico</a>	<a href="#">Interfaz</a> <a href="#">Editar</a> <a href="#">Eliminar</a>

Figura 1.24: Paneles de administración para robots y usuarios.