



GRADO EN INGENIERÍA TELEMÁTICA

Curso Académico 2019/2020

Trabajo Fin de Grado

Mejoras en entorno de robótica educativa para niños

Autor : Rubén Álvarez Martín

Tutor : Dr. José María Cañas Plaza



# Índice general

1. Introducción	1
1.1. Tecnologías web . . . . .	1
1.2. Robótica . . . . .	1
1.3. Robótica educativa . . . . .	1
2. Objetivos	3
3. Herramientas	5
3.1. JavaScript . . . . .	5
3.1.1. Características . . . . .	5
3.2. A-Frame . . . . .	6
3.2.1. HTML y primitivas . . . . .	6
3.2.2. Entidad, Componente y Sistema . . . . .	8
3.3. Blockly . . . . .	9
3.3.1. Traductor de código . . . . .	9
3.3.2. Bloques personalizados . . . . .	10
3.4. Blender . . . . .	11
3.4.1. Modelos 3D . . . . .	12
3.5. Gestores de paquetes . . . . .	12
3.5.1. NPM . . . . .	12
3.5.2. WebPack . . . . .	12
3.6. WebSim . . . . .	12
3.6.1. Descripción . . . . .	12
3.6.2. Drivers . . . . .	12

4. Mejoras a WebSim	13
---------------------	----

Bibliografía	15
--------------	----

# Capítulo 1

## Introducción

1.1. Tecnologías web

1.2. Robótica

1.3. Robótica educativa



## Capítulo 2

### Objetivos





# Capítulo 3

## Herramientas

En este capítulo se van a detallar las herramientas utilizadas en el desarrollo de este proyecto. Algunas se han elegido por facilidad de uso y otras por necesidad del entorno desarrollado.

### 3.1. JavaScript

JavaScript es un lenguaje interpretado de alto nivel que se encuentra bajo el estándar ECMAScript<sup>1</sup> y está basado en otros lenguajes de programación como Java o C.

En su principio fue concebido como lenguaje para el lado cliente implementado en un navegador web, permitiendo mejorar la interfaz de usuario y realizar páginas web dinámicas. Actualmente, JavaScript se ha ido integrando en el lado servidor y es por ello que es el lenguaje más utilizado para desarrollo web y todos los navegadores interpretan el código integrado en las páginas web.

#### 3.1.1. Características

Las siguientes características tienen en común que todas se ajustan al estándar de ECMAScript:

- Es un lenguaje estructurado, tiene gran similitud con C y comparte gran parte de su estructura (bucles, condicionales, sentencias...) a excepción del alcance de

---

<sup>1</sup>Especificación de lenguaje de programación el cual define tipos dinámicos y soporta características de programación orientada a objetos.

sus variables. En C su ámbito es el bloque en el que fue definida y, en su origen, JavaScript tenía un alcance global en las variables definidas. Es en ECMAScript 2015 cuando se añade la palabra clave `let`, que incorpora compatibilidad con block scoping (alcance de la variable en el bloque en la que es definida).

- Tipado débil, por el cual el tipo de datos está asociado al valor, no a la variable. Esto significa que una variable puede ser `number` o `string` en distintos momentos de ejecución.
- Formado en su totalidad por objetos, en los cuales los nombres de sus propiedades son claves de tipo cadena siendo `objeto.a = 1` y `objeto['a'] = 1` equivalentes.
- Lenguaje interpretado, es por esto que no requiere un compilador ni crear un fichero binario del código; cada navegador tiene su intérprete que se encarga de ejecutarlo.
- Evaluación en tiempo de ejecución gracias a la función `eval`, la cual evalúa un código en JavaScript representado como una cadena de caracteres.

## 3.2. A-Frame

A-Frame es un framework de código abierto destinado a crear experiencias de realidad virtual a partir de HTML de forma que sea sencillo de leer y comprender. De esta manera es accesible para crear una gran comunidad.

Además tiene compatibilidad con Vive, Rift, Windows Mixed Reality, Daydream, GearVR y CardBoard así como soporte para todos los controladores respectivos. También ofrece soporte para ordenadores de escritorio y para la mayoría de teléfonos inteligentes.

### 3.2.1. HTML y primitivas

A-Frame se basa en HTML y el DOM usando un polyfill<sup>2</sup> para elementos personalizados. HTML es un componente básico para Web y como tal, tiene una gran accesibilidad

---

<sup>2</sup>Fragmento de código en JavaScript utilizado para proporcionar una funcionalidad moderna en navegadores antiguos.

como lenguaje. Para crear una escena de realidad virtual con A-Frame no se requiere ninguna instalación y simplemente con la creación del HTML se puede abrir en el navegador. La mayoría de herramientas existentes (como React, Vue.js, d3.js y jQuery) funcionan en este framework.

HTML y DOM son solo la capa más externa del framework, debajo se encuentra el componente three.js en el que está basado A-Frame gracias al cual un componente puede ser utilizado en distintas entidades. De esta manera hace posible seguir el principio de programación Don't Repeat Yourself ya que, una vez registrada una primitiva, se puede hacer referencia al elemento todas las veces que sea necesario.

A-Frame proporciona elementos como a-box o a-sky llamados primitivas. Podemos crear un escenario a través de estas primitivas como el mostrado en la figura 3.1 con el siguiente código:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Escenario primitivas</title>
6     <script src="https://aframe.io/releases/0.9.2/aframe.min.js"></script>
7   </head>
8   <body>
9     <a-scene background="color: #FAFAFA">
10       <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9" shadow
11         ></a-box>
12       <a-sphere position="0 1.25 -5" radius="1.25" color="#ff0000" shadow
13         ></a-sphere>
14       <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#
15         FFC65D" shadow></a-cylinder>
16       <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4"
17         color=" #1cde83" shadow></a-plane>
18       <a-sky color="#e7e1e0"></a-sky>
19     </a-scene>
20   </body>
21 </html>
```

Listing 3.1: Código con primitivas que representa un escenario

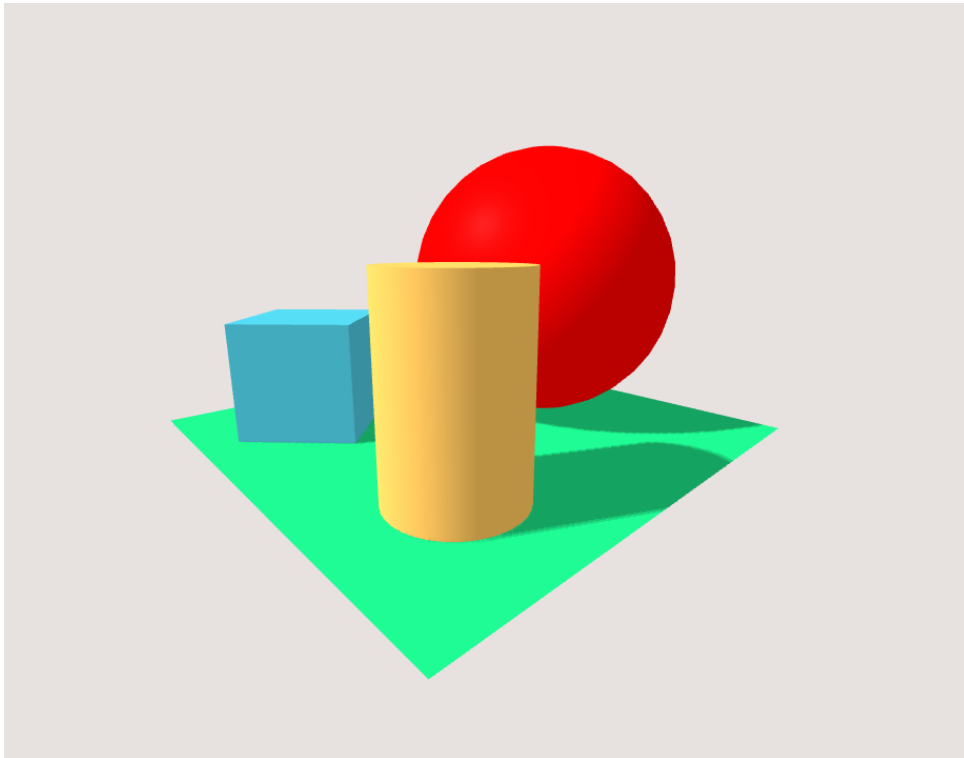


Figura 3.1: Escenario a-frame

A-Frame, además de disponer primitivas como las mostradas, hace posible la creación de primitivas para poder elaborar escenas lo más completas posible. También se pueden incluir entidades más complejas a partir de modelos 3D en formatos como gltf, obj o collada de los cuales se hablará en siguientes apartados.

### 3.2.2. Entidad, Componente y Sistema

Como ya se ha comentado, A-Frame es un framework three.js con una arquitectura de entidad-componente-sistema (ECS). Es un patrón común en 3D y desarrollo de juegos que siguen la composición sobre el principio de herencia y jerarquía. Algunos beneficios que ECS aporta son mayor flexibilidad al definir objetos, gran escalabilidad o eliminación de problemas de largas cadenas de herencia. Existe un API que representa cada pieza de ECS:

- Una entidad se representa con la etiqueta a-entity.

```
1 <a-entity geometry="primitive: box" material="color: red">
```

En este ejemplo se hace uso de `a-entity` para crear una caja de color rojo.

- Un componente se representa como un atributo de HTML. Cada componente es un objeto que tiene un esquema, manejadores y métodos. Para registrar componentes se utiliza el método de A-Frame `registerComponent`.

```
1  AFRAME.registerComponent( 'example', {  
2    init: function () {  
3      var el = this.el;  
4      el.setObject3D( 'mesh', new THREE.Mesh() );  
5      el.getObject3D( 'mesh' ); // Returns THREE.Mesh that was  
                                just created.  
6    }  
7  });
```

Listing 3.2: Código para registrar un componente

- Un sistema es el representado por atributos HTML mediante la etiqueta `a-scene`. Se registran de manera similar a un componente; gracias al método `registerSystem`.

## 3.3. Blockly

Blockly es una librería que añade un editor de código visual a aplicaciones web y móviles. Utiliza bloques gráficos para representar conceptos complejos de código de manera más sencilla. De esta manera permite a los usuarios aplicar principios de programación sin tener que preocuparse por la sintaxis y ayuda a iniciarse y a aprender a programar a estudiantes de temprana edad. Esta librería está diseñada por las personas que están detrás de Scratch<sup>3</sup> del MIT y construido sobre su base de código.

### 3.3.1. Traductor de código

Blockly es para desarrolladores y las aplicaciones Blockly son pensadas para estudiantes. Desde la perspectiva de usuario, Blockly es una forma visual e intuitiva de crear código y, desde la de desarrollador, es una interfaz de usuario preparada para crear un

---

<sup>3</sup><https://scratch.mit.edu/>

lenguaje visual que emite código y se puede exportar a otros lenguajes de programación como JavaScript, Python, PHP, Lua o Dart.

Estos generadores de código aportan las herramientas para crear funciones, condicionales, bucles, etc. El principal problema de esto es que en ocasiones se requiere del uso de APIs de otras dependencias. Blockly aporta una solución; un generador de bloques personalizados que traduce al código que deseemos. Esto aporta mucha flexibilidad y da la funcionalidad deseada para el desarrollo de este proyecto.

### 3.3.2. Bloques personalizados

Como se ha explicado, Blockly dispone de una gran cantidad de bloques predefinidos; desde funciones matemáticas hasta estructuras en bucle. Sin embargo, para interactuar con una aplicación externa, se deben crear bloques personalizados para formar una API. Según la documentación ofrecida por Google<sup>4</sup>, la mejor forma de crear un bloque es buscar un bloque existente con una funcionalidad similar y modificarlo según se necesite. De otra manera, para generar un bloque personalizado, hay varios aspectos a tener en cuenta:

- Primero se define el bloque para determinar su aspecto gráfico y su comportamiento. Esto incluye el texto, color, forma o cómo conectarlos con otros bloques. La configuración de estos parámetros se puede realizar mediante JSON o JavaScript. El bloque mostrado en la figura 3.2 se puede configurar con los dos métodos de la siguiente manera:

```
1  {  
2    "type": "custom_block",  
3    "message0": "%1 %2",  
4    "args0": [  
5      {  
6        "type": "field_input",  
7        "name": "TEXT",  
8        "text": "code_test"  
9      },  
10     {  
11       "type": "input_value",
```

<sup>4</sup><https://developers.google.com/blockly/guides/create-custom-blocks/overview>

```

12     "name": "NAME",
13     "check": "Boolean",
14     "align": "RIGHT"
15   }
16 ],
17 "output": "String",
18 "colour": 300,
19 "tooltip": "Hello",
20 "helpUrl": ""
21 }

```

Listing 3.3: Código en JSON para configurar un bloque personalizado

```

1  Blockly.JavaScript['custom_block'] = function(block) {
2    var text_text = block.getFieldValue('TEXT');
3    var value_name = Blockly.JavaScript.valueToCode(block, 'NAME',
4      Blockly.JavaScript.ORDER_ATOMIC);
5    var code = '...';
6    return [code, Blockly.JavaScript.ORDER_NONE];
7  };

```

Listing 3.4: Código en javascript para configurar un bloque personalizado

- Configurar la traducción del bloque a la instrucción deseada en los distintos lenguajes necesarios.
- Inicializar el bloque para que sea visible en el editor visual de código.

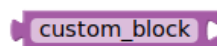


Figura 3.2: Bloque personalizazdo

## 3.4. Blender

Blender es un programa libre dedicado al diseño y animación 3D.

### 3.4.1. Modelos 3D

## 3.5. Gestores de paquetes

### 3.5.1. NPM

### 3.5.2. WebPack

## 3.6. WebSim

Websim es un simulador robótico diseñado para enseñar conceptos básicos de tecnología e iniciar a niños en robótica y programación de robots.

### 3.6.1. Descripción

El simulador hace uso del entorno A-Frame y permite conectar un editor de texto o un editor de bloques para poder programar en JavaScript o Blockly y conectar este código con el robot simulado.

### 3.6.2. Drivers

Cuadro 3.1: Métodos (HAL API) de los motores del robot.

Método	Descripción
<code>.setV(integer)</code>	Mueve hacia delante o atrás el robot.
<code>.setW(integer)</code>	Hace girar al robot.
<code>.move(integer, integer)</code>	Mueve el robot hacia delante/atrás y gira al mismo tiempo.
<code>.getV()</code>	Obtener la velocidad lineal configurada en el robot.
<code>.getW()</code>	Obtener la velocidad angular configurada en el robot.



## Capítulo 4

### Mejoras a WebSim



# Bibliografía

- [1] Documentación JavaScript: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [2] Documentación A-Frame: <https://aframe.io/>