

Introducción a la realidad aumentada

Esquema resumen de A-Frame y AR.js

Índice

	pág.
1. ¿Qué son A-Frame y AR.js?	2
2. A-Frame resumido.	
2.1. Ejemplo de escena simple.	3
2.2. Formas básicas.	4
2.3. Atributos color, posición, rotación y escala.	5
2.4. Texturas.	5
2.5. Modelos prediseñados.	5
2.6. Animaciones sobre los objetos (movimientos y rotaciones).	6
3. AR.js resumido.	
3.1 Creación de una escena básica con un modelo.	7
3.2. Marcadores personalizados.	8
Apéndice: Instalación de Web server for Chrome.	9

1.- ¿Qué son A-Frame y AR.js?.

A-Frame (<https://aframe.io/>) es un framework web de código abierto para crear experiencias de realidad virtual. Los escenarios 3D son creados por medio de etiquetas HTML y la interacción con los mismos se realiza usando javascript.

AR.js (<https://github.com/jeromeetienne/AR.js/blob/master/README.md>) es una biblioteca programada por Jerome Etienne que apoyándose en A-Frame permite crear de forma sencilla aplicaciones de realidad aumentada usando solo HTML.

2. A-Frame resumido

2.1. Ejemplo de escena simple.

```
<html>
<head>

  <!--Biblioteca A-Frame -->
  <script src="https://aframe.io/releases/1.0.3/aframe.min.js"></script>

</head>

<body>
  <!-- Definimos la escena -->
  <a-scene>

    <!-- Representamos una caja de color rojo en posición (-1,0.5, -3) y rotada 45° respecto el eje Y -->
    <a-box position="-1 0.5 -3" rotation="0 45 0" color="red"></a-box>

  </a-scene>

</body>
</html>
```

Pasos para crear una escena en 3D con A-Frame:

- Cargamos la biblioteca de A-Frame con `<script src="https://aframe.io/releases/1.0.3/aframe.min.js"></script>`
- Creamos una escena añadiendo en el cuerpo de la página la etiqueta `<a-scene> </a-scene>`
- Entre ambas etiquetas (`<a-scene>...</a-scene>`) copiamos los modelos u objetos que deseamos que aparezcan. En la siguiente página aparece un listado con los objetos o formas más comunes.

2.2. Formas básicas:

Instrucción	Descripción	Atributos	Ejemplo
<a-box>	Crea una caja	Anchura: width Altura: height	<a-box width="2" height="1" color="red"></a-box>
<a-sphere>	Crea una esfera	Radio: radius	<a-sphere radius="1" color="red"></a-sphere>
<a-cylinder>	Crea un cilindro	Altura: height Radio: radius	<a-cylinder height="2" radius="1" color="blue"></a-cylinder>
<a-tube>	Crea un cilindro abierto (tubo)	Altura: height Radio: radius Abierto: openEnded	<a-tube height="2" radius="1" openEnded="true" color="blue"></a-tube>
<a-cone>	Crea un cono	Radio inferior: radiusBottom Radio superior: radiusTop	<a-cone radiusBottom="2" radiusTop="0.1" color="red" position="0 3 - -3"></a-cone>
<a-plane>	Crea un plano	Anchura (eje X): width Altura (eje Y): height	<a-plane width="2" height="1" color="#FFC65D"></a-plane>
<a-sky>	Crea el cielo		<a-sky color="#6EBAA7"></a-sky>
<a-image>	Inserta una imagen	Anchura: width Altura: height Ruta de la img.: src	<a-image src="unaimagen.png" width="2" height="2" position="0 2 -3"></a-image>
<a-video>	Inserta un vídeo	Anchura: width Altura: height Repetir: loop	<a-video autoplay loop="true" src="mivideo.mp4" width="2" height="2" ></a-video>
<a-text>	Inserta un texto	Texto: value Alineación: align	<a-text value="Hola" width="20" height="20" align="center" position="0 4 -5" color="blue"></a-text>

2.3. Atributos color, posición, rotación y escala (atributos comunes para cada una de las formas básicas o modelos).

Color:

El color de un objeto se puede definir por medio de un código hexadecimal siguiendo el esquema rojo-verde-azul (RGB) o con su equivalente escrito en inglés.

Ejemplo: color="red" o color="#FF0000"

Nota: En Google si busca "hexadecimal color picker" le aparecerá un selector de colores que le permite saber el equivalente en hexadecimal.

Posición:

position="coordX coordY coordZ"

Ejemplo: <a-box width="2" height="1" color="red" position="-1 2 -3.5"></a-box>

Rotación:

rotation="ejeX ejeY ejeZ"

Ejemplo: Rotamos 90° respecto al eje Y

<a-box width="2" height="1" color="red" rotation="0 90 0"></a-box>

Escala:

scale="ejeX ejeY ejeZ"

Ejemplo: Escalamos un objeto en un factor 0.5 respecto a los tres ejes

<a-entity scale="0.5 0.5 0.5" position="0 2 -1" rotation="0 180 0" gltf-model="src: url(/assets/mimodelo.gltf)">

2.4. Texturas.

Para poner una textura sobre un objeto basta con cargar la imagen de la textura con el atributo **src**.

Ejemplo: Si la imagen correspondiente a la textura se encuentra en la carpeta assets y se llama mitextura.jpg

<a-box width="2" height="1" src="/assets/mitextura.jpg"></a-box>

2.5 Modelos prediseñados.

Se pueden cargar modelos ya creados con herramientas como Blender o 3D-Max. Aunque A-Frame admite diferentes formatos es aconsejable usar el formato gltf dado que también admiten animaciones.

Ejemplo: Mostramos un modelo previamente descargado desde Sketchfab (<https://sketchfab.com/>) en formato gltf que se encuentra en la carpeta assets, lo escalamos, lo posicionamos en las coordenadas (0,2,-1) y lo rotamos 180° respecto al eje Y

```
<a-entity scale="0.1 0.1 0.1" position="0 2 -1" rotation="0 180 0" gltf-model="src: url(/assets/mimodelo.gltf)">
```

2.6. Animaciones sobre los objetos (movimientos y rotaciones).

Dentro de la etiqueta que define un objeto añadimos el atributo **animation** con las siguientes propiedades:

property: Propiedad del objeto (coordenadas o eje de rotación) sobre el que se realiza la animación.

Por ejemplo: Rotación sobre el eje X: rotation.x. Desplazamiento en el eje Z: position.z

to: Valor final de la propiedad seleccionada.

dur: Duración de la animación en milisegundos.

loop: Si la animación se repite de forma continua.

easing: Facilita la transiciones en la animación para evitar saltos.

Ejemplo:

Rotamos una esfera de forma continua (loop: true) respecto al eje Y tardando 4 segundos (4000 milisegundos) en realizar cada vuelta.

```
<a-sphere position="0 2 -3" radius="1" animation="property: rotation.y; to: 360; dur: 4000; loop: true; easing: linear">
```

3. AR.js resumido

3.1 Creación de una escena básica.

Todas las primitivas anteriores de A-Frame para añadir objetos o modificarlos son válidas con AR.js:

```
<html>
<script src="https://aframe.io/releases/1.0.3/aframe.min.js"></script>
<script src="https://raw.githubusercontent.com/jeromeetienne/AR.js/2.1.4/aframe/build/aframe-ar.js"></script>
<body >
  <a-scene embedded arjs>
    <!-- Marcador por defecto (hiro) -->
    <a-marker preset="hiro">
      <a-box position="0 0.5 0" color="yellow"></a-box>
    </a-marker>

    <!-- Cargamos la cámara del móvil o webcam -->
    <a-entity camera></a-entity>

  </a-scene>
</body>
</html>
```

Los pasos para convertir cualquier escenario de A-frame en una escena de realidad aumentada son:

- Añadir a scene los atributos **embedded arjs**
- Usar la etiqueta <a-marker> con el atributo **preset="hiro"** si vamos a usar el marcador por defecto.
- Copiar todos los objetos o modelos que deseamos que aparezcan al enfocar la cámara sobre el marcador entre las etiquetas **<a-marker preset="hiro"> </a-marker>**
- Poner la instrucción **<a-entity camera></a-entity>** para usar la cámara (cámara del móvil o webcam).

El marcador que se usa en el ejemplo anterior es el que hay por defecto y se puede descargar para imprimir desde este enlace:

<https://jeromeetienne.github.io/AR.js/data/images/HIRO.jpg>

3.2 Marcadores personalizados:

Es necesario dos archivos:

- Imagen del marcador para imprimir.
- Pattern. Fichero donde se almacenan las características del marcador para que AR.js sea capaz de reconocerlo cuando lo enfoque la cámara.

Los dos archivos anteriores se pueden crear usando la herramienta online AR.js Marker Training:

<https://jeromeetienne.github.io/AR.js/three.js/examples/marker-training/examples/generator.html>

Ejemplo:

Cargamos el patrón (pattern) que con anterioridad hemos generado con la herramienta AR.js Marker Training desde la carpeta assets (patron.patt) y mostramos el modelo mimodelo.glTF. Debemos también imprimir nuestro marcador personalizado.

En este ejemplo mostramos además como cargar las animaciones del modelo si dispone de ellas. Para ello cargamos la biblioteca **aframe-extras** y añadimos el atributo **animation-mixer** a la entidad.

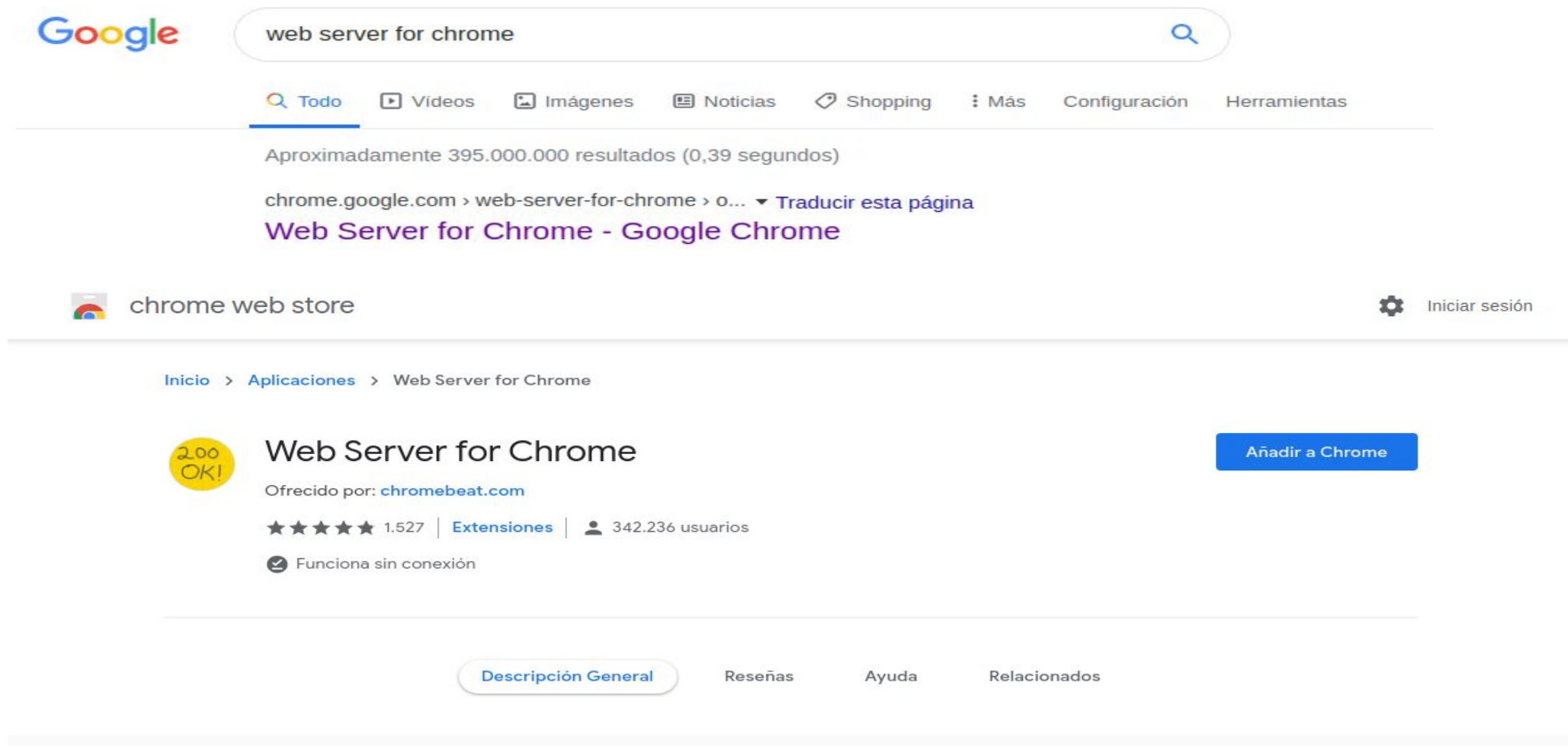
```
<html>
  <script src="https://aframe.io/releases/1.0.3/aframe.min.js"></script>
  <script src="https://raw.githack.com/jeromeetienne/AR.js/2.1.4/aframe/build/aframe-ar.js"></script>

  <!-- Biblioteca para mostrar las animaciones del modelo si dispone de ellas-->
  <script src="https://cdn.jsdelivr.net/gh/donmccurdy/aframe-extras@v6.0.1/dist/aframe-extras.min.js"></script>

  <body >
    <a-scene embedded arjs>
      <a-marker type="pattern" preset="custom" url="/assets/patron.patt">
        <a-entity scale="0.007 0.007 0.007" gltf-model="url(/assets/mimodelo.glTF)" animation-mixer></a-entity>
      </a-marker>
      <a-entity camera></a-entity>
    </a-scene>
  </body>
</html>
```

Apéndice. Instalación de Web server for Chrome.

Para poder seguir esta guía o crear una aplicación de realidad aumentada usando las herramientas expuestas debemos tener acceso a un servidor web dado que básicamente lo que estamos haciendo son páginas web. Para ello podemos recurrir a la aplicación Web server for Chrome. Para la instalación de esta aplicación desde el navegador Chromium buscamos en Google “Web server for Chrome” y pulsamos sobre el enlace que aparece en la imagen el cual nos llevará a la Chrome Web Store. Ahora solo basta con pulsar en “Añadir a Chrome” para instalarla.



The image shows a Google search for "web server for chrome" and the resulting Chrome Web Store page for the "Web Server for Chrome" extension.

Google Search Results:

- Search query: web server for chrome
- Results: Aproximadamente 395.000.000 resultados (0,39 segundos)
- Link: chrome.google.com > web-server-for-chrome > o... Traducir esta página
- Title: Web Server for Chrome - Google Chrome

Chrome Web Store Page:

- Header: chrome web store
- Navigation: Inicio > Aplicaciones > Web Server for Chrome
- Extension Name: Web Server for Chrome
- Developer: Ofrecido por: chromebeat.com
- Rating: ★★★★★ 1.527 | Extensiones
- Users: 342.236 usuarios
- Status: Funciona sin conexión
- Buttons: Añadir a Chrome, Descripción General, Reseñas, Ayuda, Relacionados

El funcionamiento de esta aplicación es muy sencillo solo debemos seleccionar la carpeta donde se encuentran nuestras páginas web pulsando en "CHOOSE FOLDER". También podemos configurar el puerto en el que escucha el servidor pero es mejor dejar el que viene por defecto (8887). Ahora para poder visualizar las páginas web que hemos creado en el navegador solo hay que dirigirse a la dirección <http://localhost:8887>.

