

# REALIDAD VIRTUAL



# INTRODUCCIÓN Y OBJETIVOS

- El objetivo de este taller, es enseñar a crear una experiencia de realidad virtual con A-Frame. Además se añadirá el objeto previamente creado con BeetleBlocks en el primer taller, incorporándole animación al objeto y visualizándolo en un escenario de A-Frame con un smartphone, siguiendo los pasos que se indicarán.
- A-frame es la herramienta elegida, se trata de un framework que permite crear experiencias de realidad virtual y se compone de elementos propios basados en HTML y de la importación de archivos en Javascript, para complementar y añadir funcionalidades.
- Los pasos en los que se divide este taller son los siguientes:
  1. Introducción y manejo de A-Frame.
  2. Uso de *primitives* e imágenes.
  3. Exportación del objeto de Beetle Blocks y textura.
  4. Módulos Javascript y animación.
  5. Visualización en el smartphone.

# PASOS PREVIOS

- Para crear la experiencia de realidad virtual con A-Frame, el primer paso, es crear un documento index.html y un repositorio en el que se van a incluir los archivos necesarios. Todos los comandos de terminal descritos en este taller, se basan en un terminal Linux, desde la carpeta Home ejecuta:

```
mkdir A-Frame
cd A-Frame
mkdir img js models
makefile index.html
```

- El contenido del index.html debe ser el siguiente:

```
1  <!--DOCTYPE html-->
2  <html>
3  <head>
4      <title>Cabecera</title>
5  </head>
6  <body>
7      <div>Cuerpo</div>
8  </body>
9  </html>
```

# INTRODUCCIÓN Y MANEJO DE A-FRAME

- El método de instalación más sencillo que ofrece A-Frame, consiste en su importación a un documento plano de HTML. Para empezar a usarlo se debe importar en la cabecera <head> de index.html de esta manera:

```
<head>
```

```
  <script src="https://aframe.io/releases/0.9.0/aframe.min.js">  </script>
```

- Dentro del `<body>` del documento se debe crear la escena de A-Frame, identificada como <a-scene>.

```
<body>
```

```
  <a-scene> </a-scene>
```

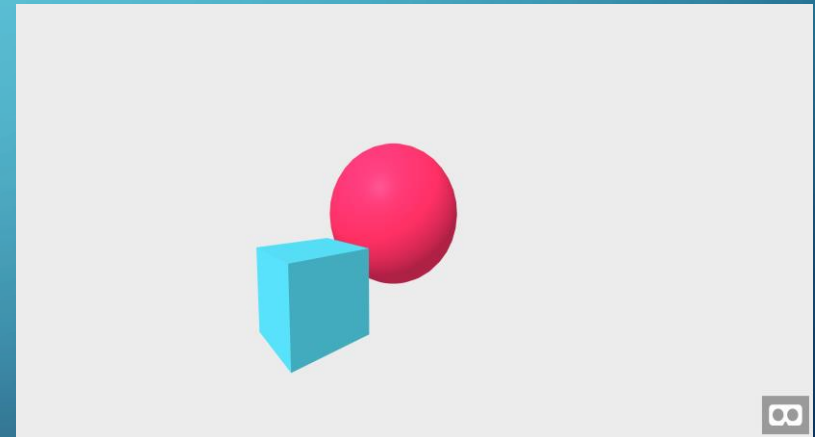
```
</body>
```

- La escena supone el núcleo de A\_Frame y es donde colocaremos los objetos, identificados como <a-entity>.

# USO DE *PRIMITIVES*

- Un ejemplo sencillo para aprender a usar A-Frame, consiste en cargar plantillas de elementos que están dentro del código de A-Frame conocidas como *primitives*, gracias a ellos se puede crear un cubo , un cilindro, una esfera .... de una forma muy sencilla. A la hora de usarlos en el index.html basta con nombrarlos como si se tratara de una entidad, además se le pueden dar diferentes propiedades. Dentro del HTML esta sería la manera de crear un cubo y una esfera, con unas propiedades de posición rotación y color específicas.

```
<body>
  <a-scene>
    <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"> </a-box>
    <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"> </a-sphere>
```



## ✓ Ejercicio 1:

- ❖ Explora la [biblioteca](#) de primitives de A-Frame y carga 3 de ellos con colores y posiciones diferentes.

# IMÁGENES

- Exportar imágenes

- A-Frame permite la exportación de imágenes propias dentro de la escena, a éstas se les puede dar diversas propiedades.
- Para cargar una imagen, primero hay que declararla dentro de la zona de *assets*. Todos los elementos externos que se introducen en A-Frame deben declararse primero en esta zona, a la cual A-Frame identifica como `<a-assets>` `</assets>`.
- Cuando se crea un *asset* se le debe dar asignar un identificador, para posteriormente poder invocarlo en la parte de la escena que se desee.
- En el siguiente ejemplo, se muestra como declarar la imagen e invocarla posteriormente con unos valores de tamaño y posición concretos:

```
<a-assets>
  
</a-assets>
[...]

<a-image src="#city" width="70" height="50" position="1.5 1 -6.5"> </a-image>
```

✓ **Ejercicio 2:**

- ❖ Carga una imagen de tal forma que sea el fondo de los dos primitivos cargados anteriormente, para ayudarte, prueba a usar el primitive `<a-sky>`. Para que la imagen quede uniforme debe ser equirectangular (360°), puedes encontrar varias en [Flickr](#).



# EXPORTACIÓN DEL OBJETO DE BEETLE BLOCKS Y TEXTURA

- Ahora que ya se ha explicado la metodología para cargar objetos en A-Frame, el siguiente paso es cargar el objeto creado con Beetle Blocks, ya con el formato `.obj`, previamente modificado con Blender. Además de poder editar las propiedades del objeto como su posición o tamaño, se le puede añadir una textura.
- El proceso de carga de cualquier objeto es el siguiente:
  - Se inicializan las variables `.obj`, `.mtl`, y la `textura` en la zona de `assets`, a cada uno se le da un identificador de la misma manera que se mostró anteriormente con las imágenes.

```
<a-assets>
  <a-asset-item id="beetle-obj" src="models/mi_objeto.obj"></a-asset-item>
  <a-asset-item id="beetle-mtl" src="models/mi_objeto.mtl"></a-asset-item>
  
</a-assets>
```

- Se invocan el `.obj`, `.mtl`, y la `textura`. Para el formato de la textura es válido cualquier `.jpg`, aunque es recomendable que se trate de algún tipo de imagen tipo textura, se pueden obtener ejemplos gratuitos en [pixabay](https://pixabay.com). En la propiedad `src` se indica la ruta de cada archivo a importar.

```
<a-entity obj-model="obj: #beetle-obj" mtl="#beetle-mtl" material="src: #beetle-texture"> </a-entity>
```

# MÓDULOS JAVASCRIPT Y ANIMACIÓN

- A-Frame nos permite añadir algo de animación básica a nuestro objeto, gracias a la importación de paquetes en JavaScript. En el ejemplo expuesto, se programa una animación de rotación al hacer click sobre el objeto, para esta funcionalidad se debe importar el módulo [aframe-animation-component](#).
- Se importa en la cabecera <head> del documento de la misma forma que se hace con A-Frame.

```
<head>
```

```
[...]
```

```
<script src="https://unpkg.com/aframe-animation-component@5.1.2/dist/aframe-animation-component.min.js"> </script>
```

- Para añadir una animación al objeto, se debe indicar el tipo entre las propiedades del objeto. Los distintos tipos de animación pueden ser desde cambiar su tamaño (scale) a mover su posición (position) o hacer una rotación.

```
<a-entity obj-model="obj:#cascos-obj" animation="property: rotation; loop: 5; to: 0 360 0"> </a-entity>
```

- Una buena forma de controlar las animaciones, es añadir un evento que de inicio a la animación cuando se interactúe con el objeto, por ejemplo, hacer un click sobre él. Para hacer esto basta añadir el evento de click entre las propiedades del objeto **startEvents: click** y un [cursor](#) para interactuar con él.

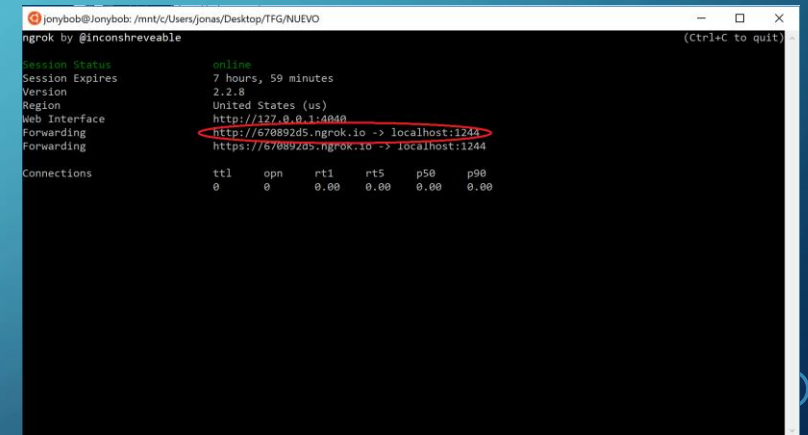
## ✓ Ejercicio 3:

- ❖ Añade una animación en la que el objeto de Beetle Blocks cambie de tamaño cuando se hace click sobre él.



# VISUALIZACIÓN EN EL SMARTPHONE

- En base a lo anteriormente expuesto, se consigue crear un entorno de realidad virtual con A-Frame de forma local. Sin embargo, para visualizar la escena de A-Frame es necesario un visor, la forma más rápida y sencilla de obtener uno, es a través de un smartphone.
- Para exportar el index.html a internet se va a contar con la ayuda de una utilidad llamada ngrok.
- Estos son los pasos necesarios para usar ngrok y exportar la escena de A-Frame:
  1. En una terminal y estando en la carpeta raíz del proyecto, se lanza el servidor de forma local:  
**python -m SimpleHTTPServer 1234**
  2. Para instalar ngrok, simplemente hay que descargar el .zip y descomprimirlo en la carpeta raíz.
  3. Se ejecuta pasando como argumento el puerto del servidor local previamente creado.  
**./ngrok http 1234**
  4. En la pantalla que muestra Ngrok, se encuentra la url pública de la aplicación juntos con otros datos.
  5. Introduciendo esta url en un navegador compatible, por ejemplo Google Chrome, se puede visualizar la experiencia de A-Frame en el teléfono.



```
jorybob@jorybob: /mnt/c/Users/jonas/Desktop/TFG/NUOVO
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Session Expires     7 hours, 59 minutes
Version             2.2.8
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding            http://670892d5.ngrok.io -> localhost:1234
                    https://670892d5.ngrok.io -> localhost:1244

Connections
  ttl  opn  rt1  rt5  p50  p90
   0    0    0.00 0.00 0.00 0.00
```