

Desarrollo de aplicación web progresiva de realidad aumentada mediante ar.js como medio de soporte para la educación básica

Rogelio Robledo Moreno, Raúl Sánchez Vázquez

Departamento de Ciencias de la Computación, Benemérita Universidad Autónoma de Aguascalientes,

Aguascalientes, México

al220099@edu.uaa.mx

al148837@edu.uaa.mx

Abstract — El presente trabajo desarrolla una línea de investigación dentro de unos de los campos más pujantes hoy en día, las aplicaciones web progresivas, dicho esfuerzo se nutrido por la integración de funciones de realidad aumentada, todo esto en un marco de tecnologías web capaces de aprovechar funcionalidades offline con el fin de apoyar a los procesos educativos en la educación básica.

Palabras clave: realidad aumentada, aplicación web progresiva, educación básica, modelos 3D, escenas 3D.

INTRODUCCIÓN

Problemática

En la actualidad las aplicaciones de las ciencias de la computación han rebasado toda proporción, posibilitando el desarrollo de un sin número de solución en diversos campos del conocimiento. Sin embargo, dichas aplicaciones a la par de verse robustecidas demandan recursos en muchos casos estratosféricos.

Si bien es cierto que las capacidades computacionales han aumentado en proporción que su precio ha bajado, dicha relación aun no es proporcional, lo cual pone en desventaja a todos aquellos sectores que carecen de recursos monetarios e inclusive técnicos, pues las aplicaciones no solo presentan una complejidad computacional sino también una usabilidad precaria.

En alguna época esto no importaba pues los sectores con mayor acercamiento tecnológico eran los relacionados con el sector académico o industrias de vanguardia, no

obstante, con la alta difusión de las tecnologías de la información, el público en general ha visto la necesidad del uso de la informática, quizá no en un alto nivel, mas no en el nivel que requieren no suele ser fácil.

Siendo específicos es posible poner de manifiesto la presente contingencia sanitaria (COVID-19), la cual obbligo miles o hasta millones de estudiantes de educación básica a suspender sus actividades escolares de forma presencial. Esto en países altamente desarrollados o de primer mundo no representa un problema gravoso, pero en un país como México es muy complejo implantar un modelo a distancia sobre todo en educación básica publica, donde los recursos de la mayoría se ven limitados.

Justificación

Afortunadamente la gran mayoría tiene acceso a dispositivos inteligentes¹ y conocen de forma suficiente el manejo de redes sociales.

A partir de la problemática presentada, es posible resumir que los mecanismos de enseñanza a distancia aunado a la precaria infraestructura tecnológica son una limitante para el óptimo aprendizaje del alumno promedio, por lo cual es necesario buscar una solución que bifurque los recursos existentes con lo último en tecnología a fin de eliminar la brecha tecnológica.

Suena un tanto antitético, sin embargo, el postulado lo que plantea es que es posible emplear técnicas y tecnologías de última generación donde la complejidad resida del lado del desarrollador y no del lado de usuario de tal forma que este último se olvide de las capas de abstracción y simplemente se ocupe de usar recursos básicos con los conocimientos técnicos que tenga por más precarios que sean.

¹ Cuando menos con las características mínimas.

Propuesta de solución

Basados en la problemática y considerando el uso de la realidad aumentada se propone desarrollar una línea de investigación para la generación de una aplicación que cumpla los siguientes requisitos.

- Multiplataforma
- Basada en la web
- Que se pueda ejecutar en dispositivo cuyos requisitos sean:
 - Cámara
 - Navegador web
 - Acceso a internet
 - Pantalla táctil
- Con una experiencia de usuario semejante a WhatsApp o Facebook

Con estas ideas en mente se pretende desarrollar una PWA de realidad aumentada que emplee marcadores o imágenes embebidos en libros de texto o láminas de apoyo, cuyo contenido sirva de apoyo en el entendimiento de temas y conceptos difíciles de abstraer por medios tradicionales.

MARCO TEORICO

Antecedentes

La realidad aumentada es un paradigma que ha llegado a reinterpretar la realidad de forma garrafal, tal fenómeno se pensaría tuvo su Genesis en la evolución de la telefonía móvil con la inclusión de cámaras y pantallas táctiles, no obstante, es una visión demasiado simplista ya que la realidad aumentada no solo se evoca a la presentación de gráficos, sino que incluye un gran número de recursos multimedia capaces de transmitir un mensaje o favorecer al desarrollo de alguna actividad.

En los albores de la tecnología *low cost*, los años 80, se desarrollaron tecnologías basadas en la electrónica, de la cual se desprenden grandes ingenios computacionales, entre ellos la optimización de los dispositivos de realidad virtual, los cuales impulsados por la industria de los videojuegos dieron muestra del gran potencial, sin embargo, para ese momento fueron pocos los sectores en explotarlo, destacando al entretenimiento y al aeroespacial.

La baja explotación de la realidad virtual tenía por precedente el costo, que, aunque reducido para la época no era asequible para el consumidor promedio. Y por otro lado la complejidad en el desarrollo de aplicaciones y sistemas adecuados a la tecnología.

No fue hasta que la industria de la computación le dio una significativa importancia al desarrollo de gráficos por computadora que la realidad virtual comenzó a ser accesible para algunos iniciados, con lo cual se abrió el camino a una pujante industria y disciplina que dotaría

de grandes avances a la posteridad. En este espíritu de innovación los sistemas fueron expandiendo sus campos de aplicación llegando hasta uno que sin duda alguna sería un nicho al cual apostarle fervientemente, la educación.

La educación ha tenido grandes transformaciones desde que el hombre vio la necesidad de estructurar el conocimiento, lo cual supone retos enormes, pues no todo el conocimiento es fácil de asimilar, ya que existen conceptos que al no poder ser palpados o percibidos bajo un modelo o esquema son difíciles de absorber, pues no es lo mismo leer cómo se comporta un volcán que ver un modelo pictórico animado capaz de transmitir los detalles precisos para su comprensión.

En este sentido se han desarrollado proyectos libres y comerciales que pretenden abordar problemas en el ámbito educativo, desde problemas cognitivos hasta la asimilación de temas, y aunque el primero suena atractivo, es el segundo el que más campo de acción tiene y más inconvenientes representa. Dichos inconvenientes parten de la definición de los procesos de aprendizaje y como aplicarlo en un contexto determinado, ya que no importan las clasificaciones y técnicas pedagógicas sino se es capaz de aterrizar un concepto de forma generalizada sin importar las características socioculturales del estudiante.

Han surgido propuestas diversas y por demás interesantes que coadyuban a la asimilación de temas y conceptos, sin embargo, las que atienden a la problemática carecen de optimalidad tecnológica y viceversa. Esta dualidad problemática busca ser balanceada mediante el presente proyecto, sin embargo, para ello es necesario sumergirse en el estado del arte y ver que se ha pretendido hacer al respecto.

Trabajos relacionados

El proyecto a cuál nos evocamos busca dos objetivos principales, en primer lugar, desarrollar escenas tridimensionales de alta calidad cuya representación sea fiel al fenómeno que emula y que desarrolle animaciones fluidas, vanguardistas y llenas de información relevante.

En segundo lugar, es preciso soportar las escenas en un medio de alto rendimiento con un bajo costo computacional capaz de ejecutar en múltiples plataformas y con el mínimo de requisitos. Para ello se extendió una búsqueda exhaustiva en las técnicas de modelado tridimensional, las aplicaciones web progresivas, y la realidad aumentada bajo un entorno de ejecución web.

Ambos objetivos han sido ampliamente estudiados por el gremio académico de tal forma que existe una gran gama de trabajos al respecto. Estos trabajos se pueden abordar desde dos vertientes: la didáctica y la parte técnica.

En la parte didáctica existe el caso particular de una herramienta, la cual es un framework que permite

particular con el plano de un juego que contiene casilleros y se encuentra desplegado en un tablero físico, los contenidos de realidad aumentada están previamente configurados, seleccionados y almacenados en función de la temática.[1]

La mecánica invita a que cada participante (alumno) del juego toca el dado virtual para que su ficha virtual avance a través de los casilleros del tablero físico. En cada casillero aparecerá un video aleatorio correspondiente a la temática seleccionada, sobre el cual se hará una pregunta. [1]

Es posible ver, que el funcionamiento es sencillo, con una mecánica por demás conocida, ya que los juegos de mesa son una constante en el ideario popular, lo destacable es el hecho de usar la interfaz tecnológica para simplificar los recursos, evitando la necesidad de demasiados objetos, como dados, tokens o fichas, por lo cual la abstracción realizada permite desarrollar un esquema de portabilidad e inclusión destacable. [1]

Del lado técnico es posible encontrar dos subdivisiones, el desarrollo de aplicaciones y el modelado 3D. En este punto las investigaciones, pero sobre todo las herramientas y marcos de desarrollo son notables, y la evidencia más palpable son los paradigmas derivados.

Para tal entender esto se desarrollarán los conceptos necesarios.

Navegador web

Es un software, aplicación o programa que permite el acceso a la web, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser vistos.[2]

Aplicación web

Una aplicación web es un tipo de software cuya ejecución es realizada por un navegador web, con una estructura backend y frontend.[3]

Aplicación web progresiva

Las aplicaciones web progresivas son aplicaciones web que se han diseñado para que sean capaces, confiables e instalables.[4]

librerías WebGL

Las librerías WebGL son implementaciones cuya utilidad es facilitar el uso de WebGL, facilitando la codificación.

Existen diversas implementaciones, sin embargo, existen dos destacables aframe.js y ar.js. Aframe es una librería cuyo objetivo es renderizar objetos en 2 y 3 dimensiones. Por otro lado ar.js es una implementación basada en

aframe que permite la integración de periféricos móviles para la generación de escenas de realidad aumentada.[5]

Archivos GLTF

GLTF es un formato de archivo para almacenar escenas² y modelos 3D, mediante una estructura JSON.[6]

Este formato comprime bien las escenas tridimensionales y minimiza el procesamiento en tiempo de ejecución de las aplicaciones que utilizan WebGL. [6]

DESARROLLO

La realidad aumentada demanda una serie de requisitos para lograr una implementación eficaz, donde se destaca la utilización de un dispositivo móvil como interfaz de interacción y procesamiento.

Tomando como base el supuesto de que es necesario trabajar con un dispositivo móvil, se infiere la necesidad de desarrollar una aplicación móvil, lo cual desde punto de vista técnico no es complejo, sin embargo, hay que tener en cuenta que hoy en día existen una variedad de sistemas operativos los cuales requieren la manipulación de distintos lenguajes de programación, como se muestra en la Fig. 1.[7]



Fig. 1. Lenguajes móviles

Para subsanar esa deficiencia es necesario desarrollar la aplicación de tal forma que con un solo desarrollo sea posible implementarlo en múltiples dispositivos.

Afortunadamente existen un sinnúmero de posibilidades, cuyo principal impulsor son las tecnologías web, donde encontramos frameworks capaces de interactuar de forma casi nativa con los sistemas operativos. No obstante, existe un punto interesante a considerar y es el hecho de una integración “casi nativa”, pues marca una

² Son modelos 3D con animaciones.

línea de deficiencia, ya que los procesos intermedios de transpilación y compilación disminuyen el rendimiento.

Es aquí donde las aplicaciones web progresivas (Fig. 2) se vuelve la opción ideal para lograr un desarrollo multiplataforma de alto rendimiento, al cual se la posibilidad de ser ejecutadas bajo requisitos mínimos sin pérdida de rendimiento.[4]

Una vez que se ha consolidado la tecnología para el desarrollo de la aplicación se procederá con su implementación.

Una aplicación web progresiva se puede desarrollar bajo uno de tres tipos de esquemas.[8]

- **Vanilla JavaScript:** Este primer esquema permite desarrollar una PWA utilizando únicamente HTML, CSS, JavaScript del lado del cliente.



Fig. 2. Aplicación Web Progresiva

- **Framework JavaScript:** Utiliza frameworks y entornos JavaScript, como vue.js y node.js
- **Framework web:** Se basa en la utilización de frameworks web como flask o .NET.

Todas las opciones ofrecen las ventajas de una PWA, mas solo la primera posibilita la concentración total de la estructura, diseño y lógica dentro del cliente, lo cual es ideal para lograr un rendimiento favorable y eliminar casi en su totalidad la dependencia de una conexión a internet.

Para el desarrollo se utilizó la plataforma *repl.it*, la cual permite el desarrollo de aplicaciones de diversos lenguajes mediante la nube.

Una vez creado el proyecto se definió una estructura de directorios y ficheros obedeciendo a las reglas de diseño de una PWA, dicha estructura se muestra en la Fig. 3.[9]

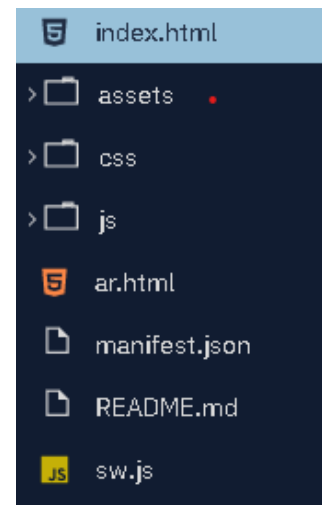


Fig. 3. Árbol de directorios y ficheros

Cada fichero en el árbol tiene una razón de ser, descrita a continuación

- **index.html:** Este fichero define el inicio y definición de la estructura general de la aplicación, aquí se llaman los scripts y archivos principales, además de que se define la estructura de la pantalla principal.
- **assets:** Se incluyen todos los archivos gráficos.
 - **icons:** Almacena las imágenes que la aplicación usara para la interfaz y para los menús en cada sistema operativo.
 - **markers:** Almacena las imágenes de los marcadores de realidad aumentada.
 - **models:** Almacena los modelos y escenas 3D en formato gltf.
- **css:** Directorio donde residen todos los archivos que dan estilo y diseño a la aplicación.
 - **style.css:** Fichero principal de la interfaz, para este caso particular toma como base el framework skeleton.
- **js:** En este directorio se encuentran los archivos referentes a la lógica de la aplicación.
 - **lib:** Almacena las librerías de realidad aumentada de la aplicación (ar.js y aframe)
 - **index.js:** Define la lógica, para el caso particular el reconocimiento de sistema operativo y el activador del Service Worker.
- **ar.html:** En este archivo se encuentra la estructura de cómo se representará la escena

3D, los mecanismos de interacción, los marcadores a usar y el modelo a representar.

- *manifest.json*: Define el comportamiento que tendrá la aplicación al instalarse en el dispositivo móvil, así como la referencia a los iconos necesarios para los lanzadores y banners.[10]
- *README.md*: Archivo de definición del proyecto (No necesario para el funcionamiento de la aplicación).
- *sw.js*: Archivo que define la interacción de la aplicación con el navegador de forma interna, esto implica su instalación, manejo de archivos, notificaciones, etc.[11]

Donde es importante destacar la relevancia del archivo *manifest.json* y *sw.js*, ya que estos son los que proveen la cualidad de PWA a la implementación.

En principio el archivo *manifest* es un conjunto de especificaciones descritas en formato json (Fig. 4), las cuales serán interpretadas por el navegador. Es importante distinguir el soporte, ya que en los navegadores base Chromium y Firefox el manifiesto es válido, sin embargo, para la ejecución en iOS es necesario incluir una serie de tags en el archivo *index.html* (Fig. 5) (ya que esta será la ruta de acceso), puesto que en dicho sistema solo safari soporta las PWA.

```
{
  "name": "Parentesis Cultural",
  "short_name": "PCultural",
  "theme_color": "#aald1d",
  "background_color": "#aald1d",
  "display": "standalone",
  "orientation": "portrait",
  "Scope": "/",
  "start_url": "/",
  "icons": [],
  "splash_pages": null
}
```

Fig. 4. Archivo *manifest.json*

```
<meta name="theme-color" content="#aald1d" />
<meta name="description" content="Aplicación AR educativa">
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="white">
<meta name="apple-mobile-web-app-title" content="Paréntesis Cultural">
<link rel="apple-touch-icon" href="assets/icons/icon-152x152.png">
<link rel="apple-touch-icon" href="assets/icons/icon-192x192.png">
```

Fig. 5. Meta tags iOS

Por otro lado, el archivo *sw.js* define un *Service Worker*, el cual contiene una serie de scripts que el navegador ejecuta en segundo plano, los cuales controlan la carga de los archivos, las bases de datos internos de tal forma que el navegador funja como entorno de ejecución para la aplicación. El *Service Worker* considera 5 etapas básicas³. [12]

1. **Inicialización del Service Worker:** Se ejecuta cuando carga la página, y le dice al navegador que hay un service worker listo para instalarse (Fig. 6). [12]

```
let sw = null;
if ("serviceWorker" in navigator) {
  navigator.serviceWorker.register("sw.js").then(swRegistered => {
    console.log("[ServiceWorker] - Registered");
    sw = swRegistered;
  });
}
```

Fig. 6. Inicialización de Service Worker.

2. **Definición de cache:** En un vector se listan los archivos que se guardaran en cache para su disponibilidad sin conexión (Fig. 7). [12]

```
const cacheName = "p-cultural";
const filesToCache = [
  "/",
  "/index.html",
  "/ar.html",
  "/js/index.js",
  "/js/lib/aframe-ar.min.js",
  "/js/lib/aframe-extras.loaders.min.js",
  "/js/lib/aframe.min.js",
  "/css/style.css",
  "/assets/icons/android.svg",
  "/assets/icons/ios.svg",
  "/assets/icons/logo.svg",
  "/assets/icons/info.svg",
  "/assets/icons/team.svg",
  "/assets/icons/github.svg",
  "/assets/models/space.glTF"
];
```

Fig. 7. Archivos en cache

3. **Instalación la aplicación:** El navegador crear un espacio de memoria (cache), donde se almacenarán los archivos de la aplicación (Fig. 8). [12]

³ Existen más, sin embargo, para el correcto funcionamiento y ejecución básica se requieren únicamente estas.

```

self.addEventListener("install", event => {
  console.log("[ServiceWorker**] - Install");
  event.waitUntil(
    caches.open(cacheName)
      .then(cache => {
        console.log("[ServiceWorker**] - Caching app shell");
        return cache.addAll(filesToCache);
      })
  );
});

```

Fig. 8. Instalación de service worker

4. **Solicitud de cache:** Cuando el usuario solicita la aplicación web el navegador busca los archivos en cache, en caso de no existir los solicita al servidor (Fig. 9). [12]

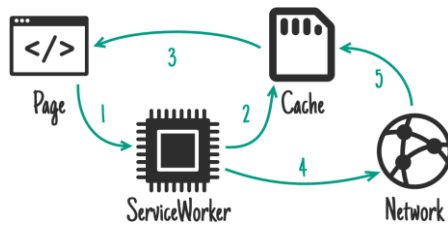


Fig. 9. Estrategia primero cache.

5. **Actualización de cache:** Periódicamente el navegador pregunta al servidor si existen actualizaciones de los archivos que se encuentran en cache, de ser así los actualiza (Fig. 10). [12]

```

self.addEventListener("activate", event => {
  caches.keys().then(keyList => {
    return Promise.all(
      keyList.map(key => {
        if (key !== cacheName) {
          console.log("[ServiceWorker] - Removing old cache", key);
          return caches.delete(key);
        }
      })
    );
  });
  event.waitUntil(self.clients.claim());
});

```

Fig. 10. Actualización de cache.

Una vez que el sw.js y el manifest.json se han desarrollado se debe verificar su correcto funcionamiento mediante la consola del navegador (Fig. 11).

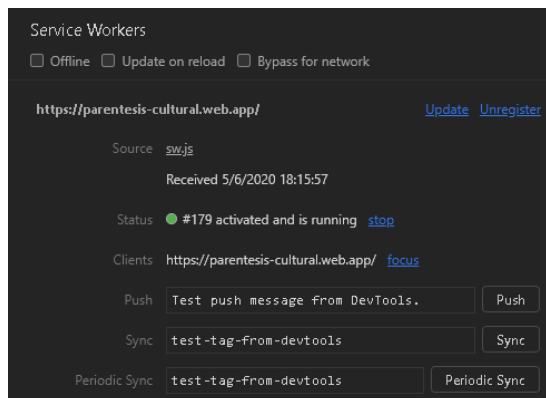


Fig. 11. Application Console sw.js en Google Chrome

Posteriormente se procede al diseño del modelo 3D, basado en los modelos WebXR, que para nuestro caso será un modelo del sistema solar. Para ello se utilizó la herramienta blender, obteniendo el modelo de la Figura 12 como base.[13]

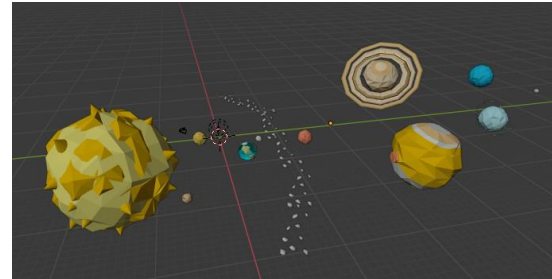


Fig. 12. Modelo del sistema solar

El desarrollo de este modelo obedece al hecho de que tanto el espacio como los cuerpos celestes son un tema que se apega de forma favorable a las representaciones tridimensionales, además de que la explicación de estos ofrece un gran cumulo de información.

Gracias a la riqueza temática del modelo es posible desarrollar animaciones sobre el movimiento de los cuerpos celestes, y exponerlos de tal forma que se ofrezca una idea del potencial de la aplicación, y en un futuro agregar mas modelos sobre diversos temas.

Ahora que el modelo fue desarrollado, se presenta la necesidad de crear la estructura donde enajara la escena para ello se utilizaran las librerías de la Figura 13.

```

<script src="./js/lib/aframe.min.js"></script>
<script src="./js/lib/aframe-ar.min.js"></script>
<script src="./js/lib/aframe-extras.loaders.min.js"></script>

```

Fig. 13. Librerías AR y VR

- **aframe.min.js:** Librería JavaScript para el renderizado de objetos tridimensionales, es una simplificación de WebGL.
- **aframe-ar.min.js:** Librería JavaScript para el renderizado de escenas mediante realidad aumentada.
- **aframe-extras.loaders.min.js:** Librería que funge como interfaz para la carga de modelos tridimensionales.

Lo siguientes es maquetar la escena mediante HTML como se muestra en Figura 14.


```

<body style="margin: 0px; overflow: hidden;">
  <script mtraced vr-mode-ui="enabled: false; arjs'sourcetype: webcam; debugUIEnabled: false;
  detectionModes: none_and_matrix; matrixCoccyos: 3d;">
  </script>
  <asset-item id="animated-asset" src="/assets/models/space.glb"/></asset-item>
  </asset>
  <marker id="animated-marker" type="barcode" value="6">
    <entity animation-mixer gltf-model="#animated-asset" scale="1 1 1"/></entity>
  </marker>
  <entity camera></entity>
</body>

```

Fig. 14. ar.html

En este documento se destacan dos elementos, la carga del modelo space.glb que es el modelo del sistema solar animado y la selección del marcador, que para este caso será el barcode-6.

Como último paso se desarrollará la interfaz de inicio, ya que se accederá a la aplicación bajo dos mecánicas:

- Acceso mediante QR: Con el acceso mediante QR, el usuario escaneará el código QR en la página del libro y como acto seguido se abrirá una página con acceso a la cámara del dispositivo y sin tener que moverse se cargará el modelo 3D.[14]
- Acceso mediante URL: Al escribir la URL en el navegador se cargará una interfaz con generales de la aplicación, un botón de inicio, y la posibilidad de instalar la aplicación.

Una vez que fue desarrollada la aplicación es necesario desplegarla en un servidor, ya que al ser una aplicación web no podrá ser descargada mediante tiendas de aplicaciones, además, las PWA requieren de un certificado SSL donde la URL sea https.

RESULTADOS

Como resultado se obtuvo una aplicación web cuyo nombre es “Paréntesis Cultural”, este nombre es relevante ya que refleja ese momento en que una persona se detiene a leer las fichas descriptivas en los libros didácticos.

Para el despliegue de la aplicación se utilizó servicio firebase, el cual proporciona hosting y un dominio con certificado SSL (Fig. 15).

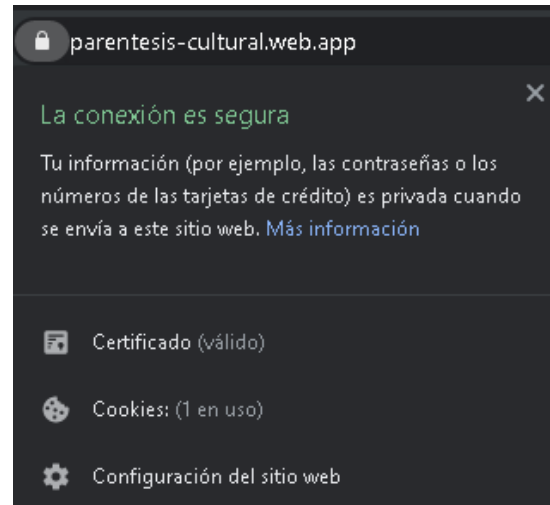


Fig. 15. Dominio con certificado SSL

Una vez desplegada la aplicación se muestra accesible desde escritorio o dispositivo móvil, siendo la versión móvil la que interesa.

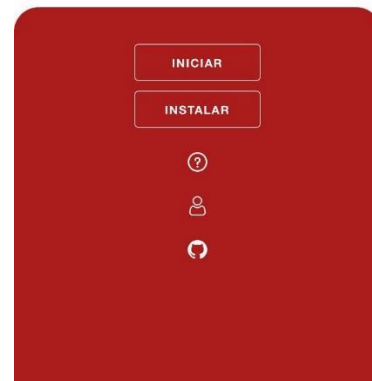
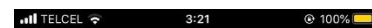


Fig. 16. Aplicación en iOS

Una vez cargada la página se despliega una interfaz de bienvenida, la cual provee información de la aplicación, para comenzar la experiencia de realidad aumentada es necesario pulsar el botón “INICIAR” (Fig. 16).

Una vez que se inició la experiencia de realidad aumentada, se observa la vista de la cámara del dispositivo, con la cual si se apunta al marcador barcode-6 aparecerá la escena del sistema solar (Fig. 17).

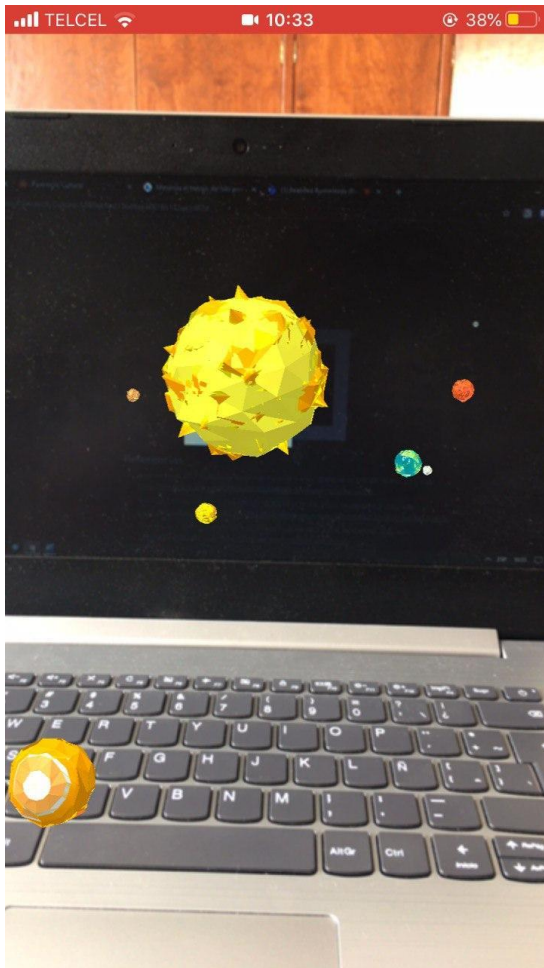


Fig. 17. Escena AR en iOS

Instalación

Como ya se vio la ventaja de desarrollar un PWA es el hecho de que es posible instalar dicha aplicación en el dispositivo de tal forma que función inclusive sin conexión a internet.

Las PWA son soportadas por tres tipos de navegadores web: [15]

1. Navegadores Chromium: Son aquellos navegadores cuya base es chromium:
 - a. Google Chrome
 - b. Edge Chromium
 - c. Opera
 - d. Vivaldi
2. Mozilla Firefox: Soportado por los navegadores Firefox tanto de escritorio como móviles.
3. Safari: Soportado por navegador safari para iOS, iPadOS y MacOS, con limitaciones a diferencia de las otras dos opciones.

Puesto que las pruebas se realizaron en dispositivos Android y iOS, la mecánica de instalación se definió para estos dos tipos de dispositivos.

En Android el proceso es simple, ya que basta con acceder al URL, y después de 30 segundos de haber cargado la página aparece un botón en la parte inferior que permite la instalación (Fig.18).

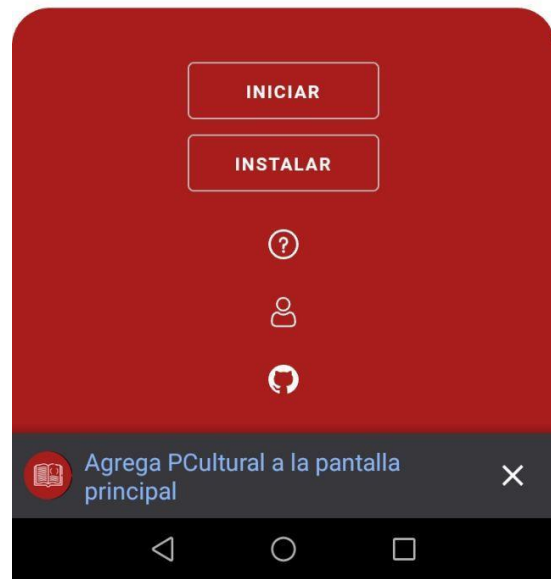
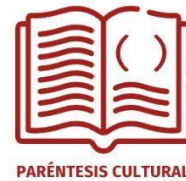
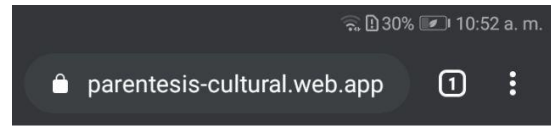


Fig. 18. Método de instalación en Android

En iOS dadas las limitaciones que el sistema presenta hacia las PWA el proceso de instalación se ve limitado y requiere una serie de pasos para lograrlo:

1. Acceder a la URL mediante Safari.
2. Presionar el botón “Compartir” (Fig. 19).



Fig. 19

3. Presionar el botón “Agregar a Inicio” (Fig. 20).

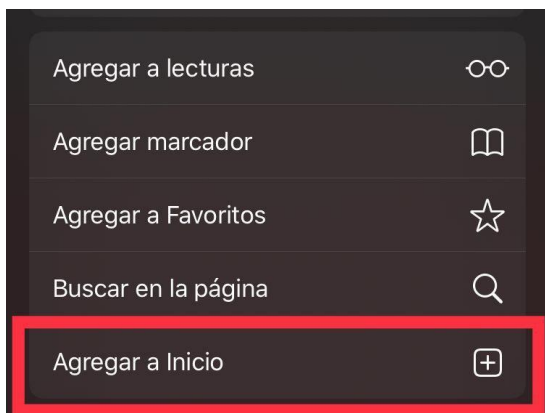


Fig. 20

Rendimiento

Cuando hablamos de una aplicación móvil el rendimiento es uno de los factores más importantes, pues no tenemos la certeza de cuáles serán las capacidades del dispositivo cliente.

Por tanto, es necesario realizar auditorías que provean un norte del estado de la aplicación y además nos ofrezca una guía sobre las correcciones pertinentes.

Para tal fin existe Lighthouse la es una herramienta automatizada de código abierto diseñada para mejorar la calidad de las web apps. Es posible ejecutarla como una extensión de Chrome o desde la línea de comandos. Le proporciona a Lighthouse la URL que quieres auditar, y Lighthouse ejecuta una serie de pruebas contra la página, y luego genera un informe sobre el rendimiento de la página. A partir de aquí, puedes usar las pruebas desaprobadas como indicadores de lo que puedes hacer para mejorar la app.[16]

Una vez desplegada la aplicación en la URL parenthesis-cultural.web.app se realizó la auditoria obteniendo los siguientes resultados de la Figura 21.

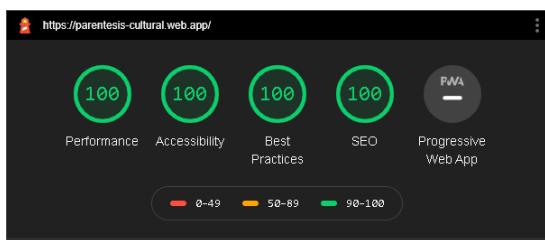


Fig. 21. Auditoria Lighthouse

Considerando los resultados obtenidos es posible establecer que la aplicación tiene un funcionamiento excelente ya que se apega a los estándares y lineamientos propuestos por la web.

CONCLUSIONES

Como bien se vio en las primeras etapas del proyecto se definieron una serie de objetivos los cuales es posible resumirlos en dos:

1. Desarrollar una aplicación eficiente y eficaz⁴
2. Desarrollar escenas RA didácticamente eficaces.

Para propósitos del proyecto el primer objetivo es el mas relevante pues al ser desarrollado para una materia introductoria a la realidad aumentada no es posible considerar como propósito de innovación el nicho de implementación, sino que la prioridad es entender y desarrollar la tecnología necesaria para que de soporte a montajes de realidad aumentada.

Por tanto, el primer objetivo se logro a cabalidad, inclusive se logró superar la expectativa ya que se desarrollo una aplicación multiplataforma, ligera y con perfectas calificaciones en las pruebas realizadas.

En cuanto a la parte didáctica, el planteamiento de la aplicación es generar una plataforma a la cual la gente pueda contribuir modelos en diversos ámbitos enriqueciendo la cualidad didáctica. Para la escena desarrollada se tomo en cuenta un tema por demás interesante, cuya complejidad de entendimiento sea alta bajo medios tradicionales, de tal forma que el resulta es una escena minimalista, llena de contexto y de gran atractivo para el publico que se dirige.

En futuros trabajos se plantea la posibilidad de generar una plataforma completa en la cual la gente puede cargar sus modelos, los cuales expliquen diversos temas en el contexto de los conocimientos necesarios para la educación básica e inclusive expandirla a otros niveles.

Dichos modelos serán auditados bajo criterios didácticos probados, con el propósito de asegurar la calidad y buen cause de la aplicación.

REFERENCIAS

- [1] Ierache, J., Igarza, S., Mangiarua, N., Becerra, M., Bevacqua, S., Verdicchio, N., Ortiz, F., Sanz, D., Duarte, N., Sena, M.. 2014. Herramienta de Realidad Aumentada para facilitar la enseñanza en contextos educativos mediante el uso de las TICs Revista Latinoamericana de Ingeniería de Software, 2(6): 365-368, ISSN 2314-2642
- [2] <https://blog.chromium.org/>
- [3] Fourault, S. (2020). How Progressive Web Apps can drive business success. junio 02,

⁴ Resumiendo, los objetivos particulares en el rubro técnico.

- 2020, de Google Sitio web:
<https://web.dev/drive-business-success/>
- [4] Richard, S & LePage. (2020). What are Progressive Web Apps?. junio 01, 2020, de Google Sitio web: <https://web.dev/what-are-pwas/>
 - [5] Etienne, J. (2017). Creating Augmented Reality with AR.js and A-Frame. junio 01, 2020, de Fundacion Mozilla Sitio web: <https://aframe.io/blog/arjs/>
 - [6] Varios. (s.f.). Los fundamentos del formato GLTF y GLB. mayo 28, 2020, de Weekly Geekly ES Sitio web: <https://weekly-geekly-es.github.io/articles/448220/index.html>
 - [7] <https://www.instagram.com/edteamlat/>
 - [8] Jonsson, S. (2019). Progressive Web Applications: Under the Hood (PWA). junio 01, 2020, de DEV Sitio web: <https://dev.to/sofiajonsson/progressive-web-applications-under-the-hood-pwa-pfl>
 - [9] Gene. (2018). Ready to use Progressive Web App template. mayo 31, 2020, de DEV Sitio web: <https://dev.to/genejams/ready-to-use-progressive-web-app-template--18o5>
 - [10] LePage, P; Beaufort, F. (2020). Add a web app manifest. 29 mayo, 2020, de webdev Sitio web: <https://web.dev/add-manifest/>
 - [11] Gauntt, M. (s.f). Introducción a los service workers. junio 02, 2020, de Web Fundamentals Sitio web: <https://developers.google.com/web/fundamentals/primers/service-workers>
 - [12] Pato. (2020). Intro to PWAs and Service Workers. junio 01, 2020, de DEV Sitio web: <https://dev.to/thisdotmedia/intro-to-pwa-and-service-workers-15d4>
 - [13] <https://immersive-web.github.io/webxr/>
 - [14] Carpignoli, N. (2018). How to deliver AR on the web only with a QR Code. mayo 28, 2020, de Chialab Open Source Sitio web: <https://medium.com/chialab-open-source/how-to-deliver-ar-on-the-web-only-with-a-qr-code-e24b7b61f8cb>
 - [15] <https://caniuse.com/#search=pwa>
 - [16] <https://developers.google.com/web/tools/lighthouse>