

Aplicaciones educativas accesibles para alumnado con discapacidad visual que usa revisor de pantalla¹

Accessible educational applications for visually impaired students using a screen reader

G. Cambronero Serentill, M. T. Corbella Roqueta

Resumen

Presentamos una propuesta de aplicaciones web *app* accesibles para el alumnado con discapacidad visual que utiliza revisor de pantalla. Una actividad de animales y una actividad de números. Estas aplicaciones, desarrolladas con el entorno de programación de Ionic, están acompañadas de una guía que anima al profesorado y al alumnado con nociones de programación a realizar modificaciones en el código fuente de la aplicación para obtener ligeras variaciones en las *apps*. Estas aplicaciones también pretenden ser una plantilla para la obtención de nuevas *apps* accesibles y así ampliar los recursos disponibles para el alumnado con discapacidad visual. Estas aplicaciones también son una alternativa accesible a los alumnos que se inician en la programación en sus centros escolares y que no pueden usar App Inventor u otros recursos no accesibles.

Palabras clave

Aplicaciones web. Accesibilidad. Recursos accesibles. Programación accesible. Juegos educativos. HTML, CSS, TypeScript. Discapacidad visual.

¹ Trabajo galardonado con el tercer premio del 7.º Concurso de Experiencias de Innovación y Buenas Prácticas en Servicios Sociales de la ONCE, presentado bajo el título *Aplicaciones educativas accesibles*.

Abstract

We present a proposal for web apps accessible to visually impaired students who use screen readers, comprising an animal activity and a number activity. These applications, developed using the Ionic programming environment, are accompanied by a guide that encourages teachers and students with programming skills to make changes to the app source to achieve slight variations in the apps. These apps are also intended as a template for the development of new accessible apps to expand the resources available to visually impaired learners. These apps are also an accessible alternative for learners who are new to programming at school and who are unable to use App Inventor or other non-accessible resources.

Key words

Web applications. Accessibility. Accessible resources. Accessible programming. Educational games. HTML, CSS, TypeScript. Visual impairment.

1. Presentación y justificación

Inicialmente, el proyecto surge de la necesidad de dar solución a un problema que se plantea en algunas aulas de Bachillerato y ciclos formativos de informática. En el área de Tecnología o Informática se propone al alumnado el diseño de *apps* utilizando un *software* no accesible con revisor de pantalla,² en concreto MIT App Inventor. Así pues, nos planteamos el reto de ofrecer un recurso accesible al alumnado con discapacidad visual que utiliza el revisor de pantalla, de modo que, a partir de un prediseño de aplicación accesible con código de programación, con nulos o muy pocos conocimientos previos, pudiese diseñar también una *app* al igual que sus compañeros.

Con el resultado de esta primera *app*, decidimos ir más allá y transformar esta idea de creación de *apps* mediante plantillas modificables en un método para obtener aplicaciones accesibles para el alumnado con discapacidad visual.

El alumnado con ceguera, al igual que sus compañeros videntes, necesita utilizar aplicaciones web y juegos educativos para aprender conceptos de forma lúdica o simplemente para sus ratos de ocio. Es evidente que hay una falta de aplicaciones accesibles en

² Un revisor de pantalla es una aplicación *software* que trata de identificar e interpretar aquello que aparece en pantalla. Esta interpretación se muestra al usuario mediante sintetizadores de texto a voz o una salida braille.

Google Play, Apple Store y en la web; por este motivo, nos planteamos ampliar el reto inicial y crear aplicaciones educativas accesibles.

2. Objetivos

A continuación, se detallan los principales objetivos del proyecto de creación de web *apps*³ accesibles:

- Crear un entorno simple y accesible de diseño de *apps* para el alumnado que trabaja con revisor de pantalla como alternativa al programa MIT App Inventor.
- Diseñar una guía para que el alumnado que usa un revisor de pantalla sea capaz de desarrollar una *app* sin muchos conocimientos previos de programación y que sirva de soporte para el profesorado del aula.
- Crear algunas web *apps* accesibles para alumnos con discapacidad visual que se inician en el uso del revisor de pantalla en entorno web o dispositivos móviles.
- Sensibilizar al profesorado desarrollador de *apps* sobre la importancia de la accesibilidad.

3. Población destinataria

Este recurso está dirigido al alumnado de Bachillerato o ciclos formativos del área de informática con discapacidad visual que utilizan revisor de pantalla y precisan programar una *app*. Al mismo tiempo, los profesores del aula ordinaria que imparten estas materias de programación podrán conocer y utilizar un recurso alternativo a MIT App Inventor.

También se dirige al alumnado con discapacidad visual grave de Educación Primaria que, en sus centros educativos, empiezan a utilizar la *tablet* como dispositivo de acceso a aplicaciones educativas. Este recurso les permitirá iniciarse en el uso de estos dispositivos con *apps* educativas accesibles y usables.

³ Las web *apps* no necesitan descargar e instalar ninguna aplicación para acceder a ellas. Se accede a través del navegador web.

4. Temporalización y fechas de realización

El proyecto para el desarrollo de una aplicación educativa accesible empezó a gestarse durante el curso 2020-2021. Esta aplicación debía reunir unos requisitos básicos:

- Entorno de programación accesible.
- Crear una aplicación accesible.
- Crear una aplicación multiplataforma.
- Poderse utilizar como plantilla para iniciar en la programación al alumnado con discapacidad visual grave.
- Lenguaje sencillo y universal.

Tras localizar un entorno y lenguaje de programación se empezó a desarrollar la aplicación y a ponerla en práctica con el alumnado destinatario.

El siguiente curso, 2021-2022, con una aplicación inicial ya creada, se crea un repositorio abierto acompañado de una página web. En el repositorio se sube el código fuente de la aplicación, así como la aplicación ya creada, con pequeñas anotaciones para realizar modificaciones por parte de los alumnos.

Estos comentarios se complementan con la página web donde se muestran un manual con ejemplos para realizar las modificaciones y otro en el que se explican los conceptos básicos de programación que hay detrás del código fuente de la aplicación.

Tras los resultados obtenidos en el curso 2021-2022, se decidió dar más difusión a la aplicación para hacerla llegar a más profesores y que pudiera ser utilizada por más alumnos durante su formación en programación.

Otra de las líneas futuras es que la aplicación sea multilinguaje, con diferentes niveles de dificultad, y hacer que el código sea más sencillo y manipulable, si es posible.

5. Metodología y desarrollo del proyecto

Para la realización de este proyecto se han seguido los siguientes pasos:

1. Investigación de la accesibilidad del *software* MIT App Inventor con revisor de pantalla.

Cambronero Serentill, G., y Corbella Roqueta, M. T. (2024). Aplicaciones educativas accesibles para alumnado con discapacidad visual que usa revisor de pantalla. *RED Visual: Revista Especializada en Discapacidad Visual*, 84, 74-89. <https://doi.org/10.53094/ILCK9875>.

2. Búsqueda de alternativas accesibles para la programación de *apps* con alumnado que usa revisor de pantalla.
3. Creación de una plantilla inicial de *app* accesible y de guías de programación con ejercicios.
4. Creación de *apps* educativas accesibles para el alumnado a partir de Educación Primaria.

A continuación, se describen cada uno de los pasos.

5.1. Investigación de la accesibilidad del *software* MIT App Inventor con revisor de pantalla

MIT App Inventor es un entorno de programación visual e intuitivo que permite crear aplicaciones completamente funcionales para teléfonos Android, iPhone y tabletas Android/iOS. Esta herramienta, basada en bloques, facilita la creación de aplicaciones complejas, pero no es accesible para el alumnado con discapacidad visual que utiliza revisor de pantalla.

Aunque hay zonas del entorno accesibles con el revisor, no es usable, ya que deben arrastrarse los bloques hacia la zona central de trabajo para realizar el diseño de la *app* y no dispone de ninguna entrada por código que permita diseñar aplicaciones a partir de un lenguaje de programación textual accesible.

5.2. Búsqueda de alternativas accesibles para la programación de *apps* con alumnado que usa revisor de pantalla

Después del análisis de diferentes alternativas de programación accesible (Eclipse, Python, etc.), nos decidimos por el marco de trabajo Ionic Framework. Se trata de un *framework* multiplataforma que permite crear aplicaciones para dispositivos móviles, ya sean Android o iOS, mediante tecnología web. Si sabemos crear una página web, podemos fácilmente utilizar Ionic. Con Ionic las aplicaciones se crean, fundamentalmente, a través de la línea de comandos.

En primer lugar, configuramos el ordenador del alumno realizando la instalación del ecosistema (Node.js y NPM (Node Package Manager)), para después poder instalar

el entorno Ionic CLI (Command-Line Interface) y Cordova. Por último, se instala el IDE Visual Code, que es accesible con los lectores de pantalla y contiene múltiples comandos. A partir de aquí, usaremos la línea de comando de Ionic CLI para crear el esqueleto de la aplicación, sus módulos y componentes.

Para la instalación de los paquetes, se puede consultar la breve explicación y los enlaces de descarga del Apéndice para Node.js, NPM, Ionic CLI, Cordova y Visual Studio.

5.3. Creación de una plantilla inicial de *app* accesible y de guías de programación con ejercicios

Con el ordenador preparado, podemos iniciar la formación del alumno a distintos niveles, en función del tiempo disponible y de sus conocimientos.

Para facilitar esta labor, se han creado dos guías de introducción a la programación en Ionic que permiten la creación de dos *apps* accesibles:

- *Guía de iniciación a Ionic*. Siguiendo las instrucciones de esta guía, seremos capaces de desarrollar una *app* de matemáticas. Esta *app*, posteriormente, se convertirá en la que se presenta en este trabajo como «Números».
- *Guía de creación de sonidos*. Siguiendo las instrucciones de esta guía, seremos capaces de desarrollar una *app* de sonidos. Esta *app*, posteriormente, se convertirá en la que se presenta en este trabajo como «Sonidos».

Estas guías están alojadas en un repositorio web y tienen asociado el código fuente de la aplicación que se espera obtener. Destacamos la importancia de los ejercicios que se van desarrollando a lo largo de las guías, así como los comentarios o apartados en los que se habla sobre la accesibilidad de las aplicaciones. Se trata, pues, no solo de un guion para el alumnado, sino también para el profesor de aula y su sensibilización sobre la importancia de la accesibilidad digital.

Para poder utilizar las diferentes guías creadas, deben realizarse las siguientes acciones:

1. Descargamos e instalamos la aplicación de ejemplo «Sonidos» o «Números».
2. Se realiza una primera lectura de la guía para entender las instrucciones.

Cambronero Serentill, G., y Corbella Roqueta, M. T. (2024). Aplicaciones educativas accesibles para alumnado con discapacidad visual que usa revisor de pantalla. *RED Visual: Revista Especializada en Discapacidad Visual*, 84, 74-89. <https://doi.org/10.53094/ILCK9875>.

3. Partimos de cero con el alumno y, siguiendo las instrucciones y ejercicios de la guía de iniciación a Ionic, vamos diseñando y construyendo una *app* similar a «Números».
4. Siguiendo las instrucciones de la guía para «Sonidos», iniciamos la creación de una *app* similar a «Sonidos».

5.4. Creación de web *apps* educativas accesibles para el alumnado a partir de Educación Primaria

A raíz de todo el trabajo desarrollado, se presentan dos aplicaciones educativas accesibles con el revisor de pantalla: la aplicación «Sonidos» y la aplicación «Números», para trabajar con los alumnos con discapacidad visual a partir de Educación Primaria.

Estas dos aplicaciones se encuentran disponibles en un espacio web gratuito y se puede acceder de una aplicación a la otra.

Pueden utilizarse en entorno web con un ordenador. Este entorno nos permite iniciarnos en el uso de los revisores de pantalla JAWS⁴ o NVDA⁵ de forma lúdica y con actividades del entorno educativo.

También pueden descargarse en un dispositivo móvil o tableta Android, lo que nos da la oportunidad de jugar con dos *apps* accesibles usando el revisor de pantalla TalkBack⁶ y, al mismo tiempo, iniciar al alumnado en el uso de dispositivos móviles aprendiendo los gestos específicos.

Una vez descargada el APK (Android Application Package) en el dispositivo móvil desde Google Drive, podemos instalarla. Para ello, debe permitirse, desde «Ajustes», la instalación de aplicaciones desconocidas.

En el siguiente epígrafe se describen en qué consisten estas aplicaciones.

4 JAWS (acrónimo de Job Access With Speech) es un *software* lector de pantalla para personas ciegas o con visión reducida. Es un producto del Blind and Low Vision Group de la compañía Freedom Scientific de San Petersburgo, Florida, Estados Unidos.

5 NVDA (Non Visual Desktop Access) es un lector de pantalla libre y gratuito desarrollado por NV Access que permite a las personas ciegas y con discapacidad visual usar ordenadores. Para ello, lee el texto que se muestra en pantalla mediante una voz sintética.

6 TalkBack es una función de accesibilidad que ayuda a las personas ciegas o con baja visión a interactuar con sus dispositivos Android mediante mensajes de voz y opciones táctiles.

6. Aplicaciones desarrolladas

6.1. Aplicación «Sonidos»

Esta aplicación permite conocer diferentes sonidos de animales con la opción «Sonidos de animales». En la opción «Adivina el animal», podemos jugar a adivinar el animal escondido. En este último juego, el alumnado podrá solicitar pistas de ayuda, que aparecen de forma textual, hasta adivinar el nombre del animal.

6.1.1. Competencias

Además de trabajar la competencia digital, ya que el alumnado con discapacidad visual aprende a manejar aplicaciones y programas informáticos (navegar por la web, usar una *app*, conocimiento de la herramienta tiflotécnica, etc.) y a acceder a la información auditiva y textual, se trabajan las competencias personal, social y de aprender a aprender. El alumnado conseguirá un aprendizaje más autónomo y eficaz utilizando el sistema de pistas, navegando por la aplicación y controlando su propio proceso de aprendizaje con el sistema de puntuación. La competencia lingüística también se refuerza, ya que el alumnado debe teclear el nombre del animal de forma correcta. En caso de trabajar con el lector de pantalla, recomendamos usar la línea braille.⁷

6.1.2. Manejo de la aplicación

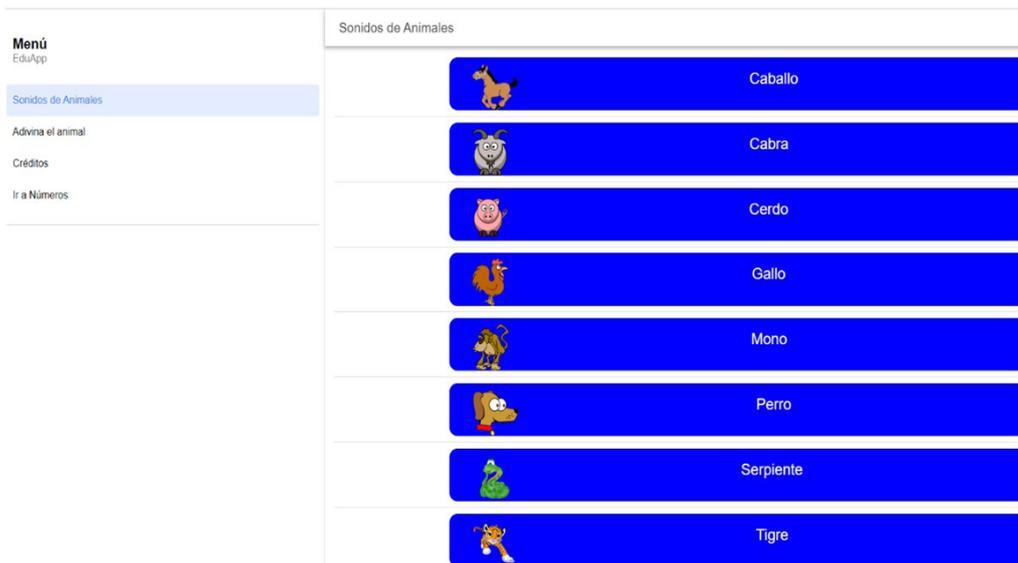
En un menú lateral, podemos seleccionar la opción de juego: «Sonidos de animales», «Adivina el animal», «Créditos» o «Ir a Números».

En la primera opción, «Sonidos de animales», podemos realizar, usando el tabulador, un recorrido por un total de ocho animales y escuchar su sonido al seleccionarlos con «Enter».

En la Figura 1 se observa la pantalla inicial de la aplicación en su opción «Sonidos de animales». En una columna a la izquierda se eligen las opciones del menú EduApp. En este caso, «Sonidos de animales». A la derecha, en la zona central, aparecen en columna los botones con los animales caballo, cabra, cerdo, gallo, mono, perro, serpiente y tigre. Botones con una imagen del animal y su nombre.

⁷ Una línea braille es un dispositivo que permite que un usuario de lector de pantalla pueda leer el contenido de la pantalla utilizando la yema de sus dedos mediante el alfabeto braille. Esta línea va mostrando en braille el contenido de la pantalla según se va desplazando el usuario por ella.

Figura 1. Pantalla inicial de la web app «Sonidos de animales»



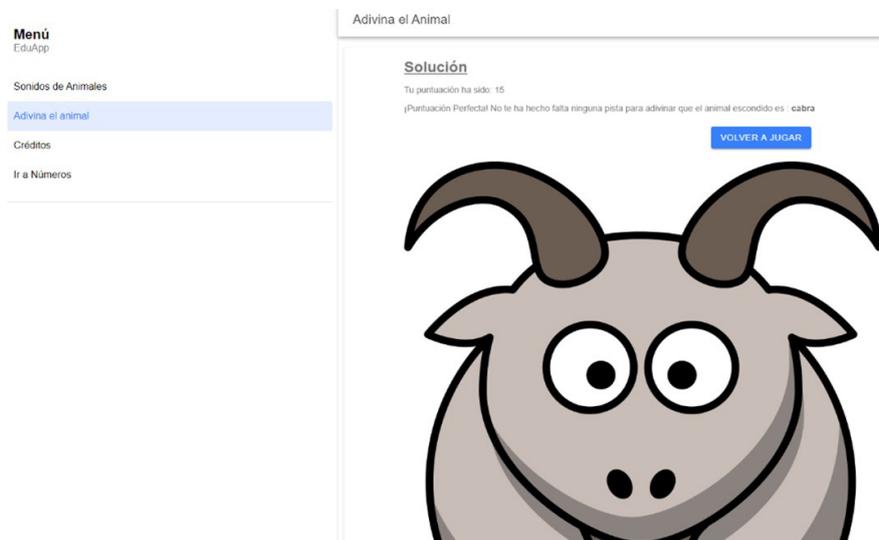
En la opción «Adivina el animal», nos movemos con el teclado hasta el botón de «Reproducir» y, al pulsar «Enter», escuchamos el sonido de un animal. En el cuadro de edición «Nombre del animal», podemos teclear su nombre. Seguidamente, debemos pulsar el botón «Respuesta» para verificar el resultado. Inmediatamente, en caso de acierto, nos aparecerá la puntuación y un mensaje de felicitación con la imagen del animal y el botón «Volver a jugar». En caso de error, nos indica el número de intentos. En este caso, debemos borrar la respuesta con el botón «Borrar» y teclear una nueva respuesta. Así hasta acertar. También podemos solicitar una pista de ayuda con el botón «Mostrar pista» y aparecerá una pista en formato texto. Cada vez que pulsamos el botón de «Respuesta» se contabiliza un intento.

En la Figura 2 se observa la pantalla inicial de la aplicación en su opción «Adivina el animal». En una columna a la izquierda se eligen las opciones del menú EduApp. En este caso: «Adivina el animal». A la derecha, en la zona central, aparece la solución con la puntuación obtenida, 15 puntos, y el nombre y la imagen del animal: la cabra. También, el botón para volver a jugar.

En el apartado «Créditos» se encuentra el enlace al código fuente y los enlaces a las fuentes libres de imágenes y sonidos utilizados en la aplicación.

El apartado «Ir a Números» nos enlaza con la web de la aplicación «Números».

Figura 2. Pantalla «Adivina el animal». La cabra. Sin error



6.2. Aplicación «Números»

La aplicación «Números» presenta diferentes juegos matemáticos en los que el alumnado deberá adivinar el número escondido, calcular una suma o realizar la multiplicación de dos números aleatorios.

6.2.1. Competencias

Aparte de la competencia matemática del cálculo, se trabaja el recorrido por la recta de números, ya que se va informando al usuario si el resultado que ha introducido es mayor o menor al real. Además de la competencia matemática, desde el punto de vista de un estudiante con discapacidad visual, se trabaja la competencia digital específica de navegación por una página web con una herramienta tiflotécnica,⁸ y el relleno y envío de un formulario, tan abundantes en las páginas web actuales, utilizando el revisor de pantalla y la línea braille.

Otras competencias que se trabajan son la personal, la social y la de aprender a aprender. El alumno realizará un aprendizaje más autónomo y eficaz al recibir diferentes estímulos durante el uso de la aplicación. El alumno, a su vez, puede realizar una autoevaluación de su respuesta al saber cuántos intentos ha necesitado hasta calcular

⁸ Las herramientas tiflotécnicas son el conjunto de conocimientos, técnicas y recursos encaminados a procurar a los ciegos y deficientes visuales los medios oportunos para la correcta utilización de la tecnología.

el resultado correcto, y realizar, de forma autónoma, tantos cálculos como estime oportuno hasta conseguir no equivocarse.

En caso de utilizar un lector de pantalla, se aconseja el uso de una línea braille.

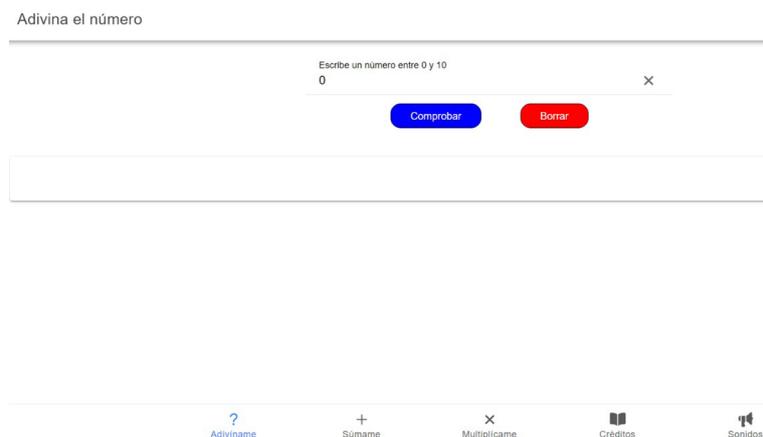
6.2.2. Manejo de la aplicación

La aplicación «Números» ofrece un menú con cinco opciones o pestañas: «Adivíname», «Súmame», «Multiplícame», «Créditos» y «Sonidos».

En la primera opción, «Adivíname», nos solicita un número entre 0 y 10. Con el teclado, podemos situarnos en el cuadro de edición y teclear nuestra propuesta. Seguidamente, pulsamos «Enter» o hacemos clic en el botón «Comprobar» y nos dirá si hemos acertado, si nos hemos pasado o si nos hemos quedado cortos. En caso de acierto, aparece el botón de «Volver a jugar». En caso de error, leyendo los mensajes, deberemos ajustar nuestra respuesta hasta adivinar el número propuesto por la aplicación. Antes de teclear nuestra siguiente opción, debemos borrar el número escrito pulsando la cruz del cuadro de edición (botón «Reset»). Disponemos de un total de 5 intentos.

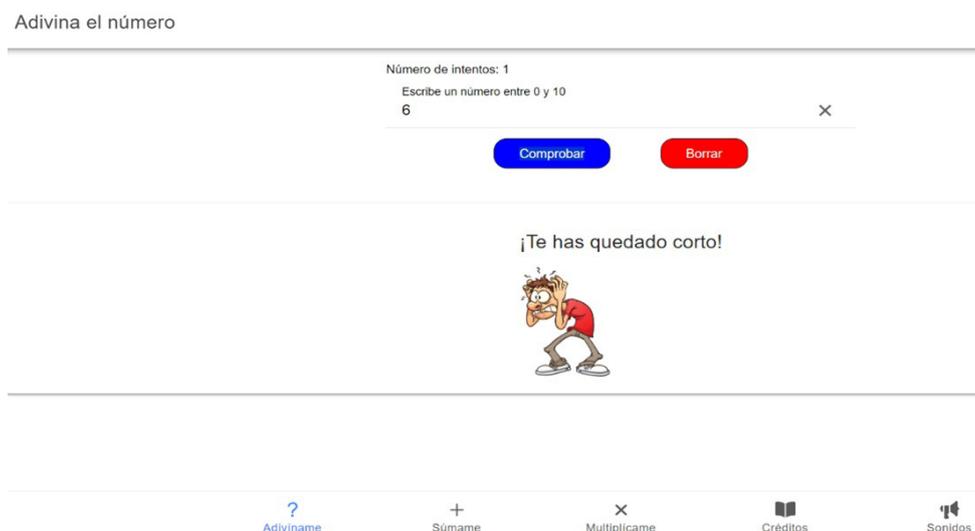
En la Figura 3 se observa la pantalla inicial de la aplicación en su opción «Adivina el número». En la parte inferior de la pantalla se eligen las opciones «Adivíname», «Súmame», «Multiplícame», «Créditos», «Sonidos». En la zona central de la aplicación, aparece un cuadro de edición para escribir un número entre 0 y 10. También los botones «Comprobar» y «Borrar».

Figura 3. Web app «Números». Pantalla «Adivina el número». Entre 0 y 10



En la Figura 4 se ha tecleado un 6, pero aparece la pista «¡Te has quedado corto!» y la imagen de muñeco que se lleva las manos a la cabeza. Lo leemos con flecha de cursor.

Figura 4. Tecleamos un 6, pero nos hemos quedado cortos



Borramos con el botón «Borrar» y vamos probando suerte hasta encontrar la respuesta correcta. Tenemos un total de 5 intentos. Podemos continuar el juego pulsando en el botón «Volver a empezar».

En la opción «Súmame», el funcionamiento es similar. Leemos la propuesta de dos números aleatorios y debemos calcular el valor de su suma. La tecleamos en el cuadro de edición y comprobamos el resultado pulsando «Enter» o haciendo clic en el botón «Comprobar». En caso de acierto, podemos volver a jugar pulsando en el botón «Reiniciar». En caso de error, disponemos de más intentos para volver a teclear el resultado, pero antes de teclear un nuevo número debemos borrar haciendo clic en el botón «Borrar».

En la Figura 5 se observa la pantalla de la opción Suma. Se proponen dos números, el primero 28 y el segundo 29 y en el cuadro de edición Valor de la suma se ha escrito el número 47.

En la opción «Multiplícame» el funcionamiento es el mismo, pero, en este caso, nos propone calcular el producto de dos números aleatorios.

Figura 5. Opción «¡Suma!». Botón «Comprobar» y botón «Borrar»



Si se quiere obtener el código fuente de la aplicación, puede accederse a la pestaña «Créditos», en la que se muestra un enlace al repositorio web, así como a los sonidos y las imágenes utilizados en la aplicación.

7. Conclusiones

Nuestra experiencia de introducción de un entorno de programación accesible a nivel de ciclos formativos de informática ha sido positiva. El alumnado ha podido revisar y comprender el código y avanzar realizando los ejercicios de la guía propuesta en el desarrollo de la *app* de ejemplo.

Las dos aplicaciones web *app*, «Sonidos» y «Números», se han usado con alumnado de Primaria con discapacidad visual, tanto baja visión como ceguera, obteniendo un resultado satisfactorio.

Un reto futuro consistirá en, a partir de estas plantillas de web *app*, conseguir, con algunas variaciones, más aplicaciones similares, tales como sonidos de instrumentos musicales, sonidos de vehículos, frases célebres, refranes conocidos o cálculos por niveles educativos.

Otro reto que queda pendiente es la accesibilidad de los programas que se utilizan con el alumnado de 4.º de la ESO para el diseño de *apps* basados en bloques de

programación (MIT App Inventor), ya que, para este colectivo, de momento todavía es pronto para introducirlo en un entorno de programación como Ionic.

Gabriel Cambronero Serentill. Instructor de tflotecnología y braille (ITB). Delegación Territorial de la ONCE en Cataluña. Calle Sepúlveda, 1; 08015 Barcelona (España). Correo electrónico: gacs@once.es.

María Teresa Corbella Roqueta. Profesora. Centro de Recursos Educativos (CRE) de la ONCE en Barcelona. Gran Vía de les Corts Catalanes, 394; 08015 Barcelona (España). Correo electrónico: mtrc@once.es.

Cambronero Serentill, G., y Corbella Roqueta, M. T. (2024). Aplicaciones educativas accesibles para alumnado con discapacidad visual que usa revisor de pantalla. *RED Visual: Revista Especializada en Discapacidad Visual*, 84, 74-89. <https://doi.org/10.53094/ILCK9875>.

Apéndice. Enlaces

Enlaces a aplicaciones y guías para la creación de plantillas iniciales de *apps* accesibles

- MIT App Inventor. Enlace a la página principal: <https://appinventor.mit.edu/>.
- Guía para la instalación de Ionic: <https://docs.google.com/document/d/1cFBnJand-qEZohNkBVrYmA9qUT772c3vN/edit>.
- Enlace a la página principal: <https://ionicframework.com/>.
- Enlace para la instalación de Ionic CLI: <https://ionicframework.com/docs/cli>.
- Ionic CLI. Es un entorno de desarrollo gratuito y de código abierto para crear aplicaciones web multiplataforma con diferentes lenguajes de programación. Para instalarlo, se debe ejecutar el siguiente comando en un terminal:⁹

```
npm install @ionic/cli -g
```
- Cordova es un paquete de Ionic que permite utilizar configuraciones nativas de los sistemas operativos iOS y Android.
 - Enlace para la instalación de Cordova: <https://ionicframework.com/docs/cli/commands/cordova-build>.
 - Para instalarse, debe ejecutarse el comando:

```
npm install -g cordova
```
- Guía de creación de «Sonidos»: <https://docs.google.com/document/d/1pzXsqEwhEI-qutEe2h5PCfoD4ws40E00P/edit>.
- Node.js y NPM. Es un entorno de ejecución de JavaScript gratuito, de código abierto y multiplataforma que permite, entre otras cosas, crear aplicaciones web. Dirección de descarga: <https://nodejs.org/en/download/prebuilt-installer/current>.

⁹ Debe tenerse instalado Node.js para poderse ejecutar el comando.

NPM (Node Package Manager) es un sistema de gestión de paquetes predeterminado para tiempo de ejecución de JavaScript Node.js. Este paquete se instala al instalar Node.js.

- Enlace para la instalación de Node.js y NPM: <https://nodejs.org/en/>.
- Visual Studio Code.¹⁰ Es un entorno de desarrollo interactivo para múltiples lenguajes de programación. Dirección de descarga: <https://code.visualstudio.com/download>.

Enlaces a aplicaciones creadas

- Aplicación «Sonidos»:
 - Enlace de acceso a la aplicación: <https://sounds-f5696.web.app>.
 - Enlace de descarga de la aplicación (APK para Android): https://drive.google.com/file/d/14vEx_pxFiA3XbOsSgB3LnX6ACCmK8zoh/view.
 - Enlace al repositorio del código fuente: <https://github.com/BCNITB/EduApp/tree/main/sounds>.
- Aplicación «Números»:
 - Enlace de acceso a la aplicación: <https://numbers-54c2d.web.app>.
 - Enlace de descarga de la aplicación (APK para Android): https://drive.google.com/file/d/1_yCDG1IX_IAU-zjhwSze-7J4tkOdaGsP/view.
 - Repositorio del código fuente de la aplicación «Números»: <https://github.com/BCNITB/EduApp/tree/main/numbers>.

Enlaces a fuentes gratuitas

- Fuente de imágenes: <http://cliparts.co/>.
- Fuente de «Sonidos»: <http://www.sonidosmp3gratis.com/>.

¹⁰ Puede utilizarse el bloc de notas para la edición y visualización del código.