

¿Programación para Todos? Herramientas y Accesibilidad: Un Estudio de Caso

Programming for All? Tools and Accessibility: A Case Study

Natalia G. Monjelat ^{1*}
Marisa A. Cenacchi ¹²
Patricia S. San Martín ¹²

¹Instituto Rosario de Investigaciones en Ciencias de la Educación (CONICET-UNR), Argentina
² Universidad Nacional de Rosario, Argentina

Los entornos en línea de programación con bloques han introducido la programación en múltiples contextos educativos. Sin embargo, para garantizar a los estudiantes el derecho de acceso a estos contenidos novedosos en igualdad de condiciones, es importante identificar la adecuación de estos espacios virtuales a las pautas de accesibilidad web. A partir de esta problemática, se realizó un estudio de caso en torno a la plataforma en línea Scratch, donde se la evalúa siguiendo las pautas de accesibilidad web 2.0. Los resultados evidenciaron que la web de creación de proyectos no resulta accesible a múltiples colectivos de personas. A partir de esto, se advierte sobre la importancia de contar con tecnologías que permitan aprender a programar desde un enfoque no excluyente y complejo, reconociendo las barreras que presentan las herramientas en los procesos de diseño de programas educativos, trayectos de formación docente y en la elaboración de políticas vinculadas con estas temáticas.

Descriptores: Programación; Educación ciudadana; Educación inclusiva.

Online programming environments with blocks have introduced programming in multiple educational contexts. However, in order to guarantee students, the right of access to these novel contents on an equal footing, it is important to identify the adequacy of these virtual spaces to web accessibility guidelines. Based on this issue, a case study was carried out to evaluate the online Scratch platform according to web 2.0 accessibility guidelines. The results showed that the web for projects creation is not accessible to multiple groups of people. From this, it is noticed the importance of having technologies that allow learning to program from a non-exclusive and complex approach, recognizing the barriers presented by the tools in the design process of educational programs, trajectories of teacher training and in the development of policies related to these issues.

Keywords: Computer programming; Civic education; Inclusive education.

Introducción

En los últimos años, la enseñanza y el aprendizaje de programación informática en contextos escolares es una temática ampliamente estudiada a nivel internacional. Aunque estas ideas ya fueron planteadas previamente, en la actualidad están siendo reconsideradas a través de propuestas innovadoras. En este marco se destacan las interfaces visuales con bloques, que al presentar la programación de una manera más sencilla e intuitiva apuntan a alcanzar a un colectivo masivo de personas (Resnick et al., 2009). Por ello, la mayoría de los recursos para enseñar y aprender programación se encuentran disponibles en páginas

*Contacto: monjelat@irice-conicet.gov.ar

de internet, pero que, en general, no siguen las pautas de accesibilidad para el contenido web (WCAG). Aunque la legislación de cada país es diferente en cuanto al cumplimiento de estas pautas (Coleman, 2017; Luján-Mora, 2013; Peñafiel y Luján Mora, 2014) no cabe duda que resultan fundamentales para garantizar a los estudiantes el derecho de acceso a estos contenidos novedosos en igualdad de condiciones.

Por otra parte, para programar con bloques suele ser necesario utilizar un periférico de entrada como el *mouse* o ratón, tanto sea para seleccionar los mismos, construir los programas o ejecutar el programa creado. Por ejemplo, en las plataformas Scratch o Blockly no se utiliza el teclado para ubicar, seleccionar y posicionar un bloque en sus espacios de trabajo, ya que fueron diseñadas para ser utilizadas con el ratón (Ludi, 2015). Asimismo, el producto que se programa suele presentarse en formato visual, al igual que ciertos mensajes que indican errores o menús emergentes. Estas cuestiones implican que los estudiantes con dificultades, por ejemplo, de movilidad o visuales, no puedan utilizar este tipo de herramientas, reduciendo así sus posibilidades de participación en las actividades educativas propuestas (Wagner et al., 2016). Sin embargo, las propuestas y programas que abogan por el acceso a la programación desde contextos escolares, continúan privilegiando la programación con bloques y los entornos visuales a través de cursos, manuales para docentes y alumnos y demás materiales.

Considerando lo expuesto, el presente trabajo tiene por objetivo visualizar esta problemática a partir del análisis de una reconocida plataforma de programación, tomando como referencia su adecuación a las pautas de accesibilidad web (WCAG2.0). Con este estudio, se pretende aportar datos concretos que puedan guiar futuros diseños de prácticas de programación con tecnologías de la información y la comunicación que habiliten y promuevan una educación inclusiva desde un marco crítico y reflexivo (Booth y Ainscow, 2015; Duk y Murillo, 2016; Monjelat y San Martín, 2016).

1. Programación: Múltiples herramientas, múltiples iniciativas

La enseñanza de la programación en el ámbito escolar data de 1960 cuando el programa Logo (Papert, 1980) fue planteado como un marco para enseñar matemáticas. No obstante, ese entusiasmo inicial decayó debido a que la sintaxis de los lenguajes utilizados resultaba compleja y la programación era generalmente presentada con actividades desconectadas de los intereses de los estudiantes (generación de números primos, líneas simples de dibujo). Además, al trabajar en estos entornos no se ofrecía guía frente a los errores ni se promovía mayor exploración cuando se la requería (Brennan, 2013).

Frente a esta situación se desarrollaron herramientas que intentaban superar esas barreras utilizando, por ejemplo, entornos visuales de programación con bloques. Entre las más destacadas se señalan Scratch, Alice, Blockly y Kodu, plataformas en las cuales es necesario arrastrar y unir distintos bloques para generar código (Kaleliog, 2015). Asimismo, en este marco se destacan iniciativas como “Code.org” que ofrecen actividades en más de 20 idiomas dentro de la propuesta la “Hora del código”, donde han participado más de trescientos millones de personas (Alvarado, 2014).

Este tipo de herramientas y espacios web son empleados en diferentes iniciativas y proyectos a nivel mundial, que se enmarcan en políticas educativas que promueven el desarrollo del pensamiento computacional a partir del aprendizaje de la programación. En

el contexto educativo latinoamericano la enseñanza de la programación se enmarca en distintos proyectos y políticas educativas (Rodríguez y Carnota, 2015). Aunque no se revelan estudios comparativos, sí se reportan experiencias pioneras en diferentes niveles educativos en varios países (por ejemplo: Barreto, L'erario y Fabri, 2015; Galvis, 2014; Mota y Adamatti, 2015; Kereki y Fonseca de Oliveira, 2013; Muñoz et al., 2014).

Asimismo, es posible relevar documentos y materiales donde se señala cómo utilizar herramientas de programación con bloques dentro del contexto escolar (Brennan, Balch y Chung, 2014; Factorovich y Sawady, 2017), así como múltiples cursos de formación docente, tanto en educación primaria, secundaria como superior (Grover y Pea, 2013; Hubwieser, Armoni y Giannakos, 2015; Monjelat, 2017).

En síntesis, programar utilizando bloques en entornos predominantemente visuales, resulta una estrategia ampliamente utilizada para aprender programación y desarrollar el pensamiento computacional. Como se ha mostrado, este tipo de herramientas son utilizadas en diversos contextos, avalados por políticas educativas y programas internacionales. Por ello resulta relevante estudiar cómo estos recursos se adecúan a pautas internacionales que posibiliten prácticas educativas inclusivas como, por ejemplo, las pautas de accesibilidad para el contenido web, en el caso de recursos que se presenten en línea.

2. La importancia de la accesibilidad web para los contenidos educativos

Al ser la programación una instancia que permite elaborar y utilizar contenidos digitales, resulta fundamental eliminar las barreras de acceso a estas herramientas. En este sentido, la organización internacional World Wide World Consortium (W3C), se ha esforzado en elaborar pautas y estándares que guíen y permitan la elaboración de contenidos digitales accesibles. Estos estándares son normativos y diversas leyes nacionales e internacionales, como el estándar ISO/IEC40500, los han tomado como bases que deben cumplir los sitios web para ser accesibles y garantizar el derecho de acceso a la información de todas las personas en igualdad de condiciones. La W3C se refiere a la accesibilidad web, como un acceso universal a la web independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los participantes.

En este marco, dicha asociación ha elaborado las Pautas de Accesibilidad para el Contenido Web (WCAG) 2.0 (WC3, 2008), que cubren un amplio rango de recomendaciones para crear contenido web más accesible. Seguir estas pautas habilita la creación de un contenido que no presente barreras para un mayor número de personas, incluyendo aquellas que presentan diversas problemáticas funcionales como ceguera y baja visión, sordera y deficiencias auditivas, deficiencias del aprendizaje, limitaciones cognitivas o de la movilidad, dificultades del habla, fotosensibilidad y/o combinaciones de las anteriores. Asimismo, se sostiene internacionalmente que cumplir con estas pautas puede ayudar a que el contenido sea más usable para cualquier persona y bajo diferentes circunstancias tecnológicas: conectividad, hardware, software, etc. (Cenacchi, 2015).

3. Metodología

Considerando el bajo porcentaje de estudios sobre la temática de la accesibilidad web de plataformas de programación por bloques, se realizó un estudio de caso exploratorio y descriptivo (Yin, 2003) con el objetivo de realizar una investigación inicial que aporte datos validados en relación a la problemática. Se escogió como caso la herramienta Scratch, por las siguientes razones:

- Permite crear y desarrollar proyectos desde una plataforma en línea.
- Existen múltiples guías, manuales, tutoriales, foros, programas educativos, documentos institucionales y gubernamentales donde se la menciona y destaca como herramienta idónea para la enseñanza de la programación.
- Según estadísticas recientes del MIT con esta herramienta se han creado, a la fecha, más de 19 millones de proyectos. Asimismo, hay registrados más de 16 millones de personas en su comunidad en línea, donde se ha manifestado la necesidad de contar con una interfaz más accesible (puede consultarse más información en <https://scratch.mit.edu/discuss/topic/20058/>).
- El acceso a la herramienta no implica registro (excepto si se quiere guardar lo realizado) por lo que se puede acceder de forma directa a la interfaz para programar.
- Desde el equipo de Scratch se han desarrollado diferentes estrategias de mejora de la herramienta por lo cual es necesario aportar evidencias que sustenten posibles cambios futuros que consideren la problemática planteada.

Para estudiar la accesibilidad de la interfaz se siguieron las recomendaciones publicadas por la WAI: “Metodología de Conformidad de Accesibilidad del Sitio Web 1.0” (WCAG-EM), cuya última actualización es de julio del 2014 (WC3, 2014). La WCAG-EM1.0 plantea 5 pasos como muestra la figura 1.

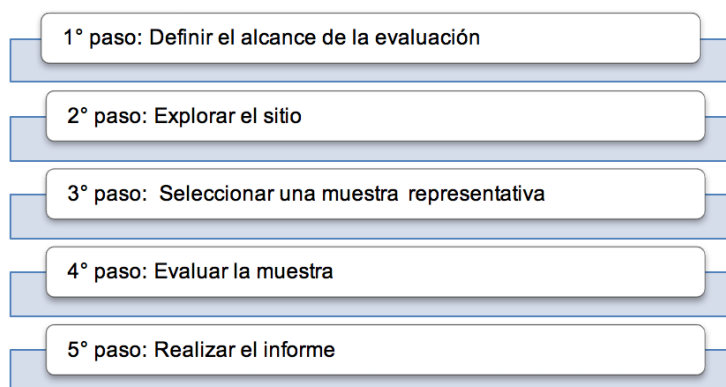


Figura 1. Pasos en la metodología WCAG-EM1.0

Fuente: Elaboración propia.

Estos pasos tienen por objetivo proporcionar una metodología fiable que se pueda utilizar internacionalmente para la evaluación de sitios web tomando como referencia las WCAG 2.0, previamente mencionadas. La metodología WCAG-EM1.0, además de contar con el respaldo de una entidad internacionalmente reconocida en el tema como es la WC3, ha sido

empleada en múltiples investigaciones debido a su validez y confiabilidad como instrumento de análisis (Acosta-Vargas et al., 2018; Cennachi, 2015; Laitano, 2015; Rodríguez et al., 2017).

4. Resultados

A los fines del presente artículo, cabe señalar que los resultados se presentan considerando como destinatarios principales personas no expertas en programación web y/o accesibilidad, por lo cual se proporcionan algunas informaciones en referencia a la accesibilidad web, señalando ejemplos concretos relacionados con la herramienta estudiada. A continuación se desglosa el análisis realizado siguiendo la secuencia descrita por la WCAG-EM1.0. Los pasos 2 y 3 se presentan juntos con el objetivo de aportar mayor claridad en cuanto al proceso realizado.

4.1. Alcance de la evaluación: conociendo Scratch

Tomando como objeto de estudio la herramienta de programación por bloques Scratch, se evaluará la plataforma en línea que ofrece, alojada en <https://scratch.mit.edu/>. Este sitio cuenta con un espacio para crear proyectos multimedia programando diferentes objetos mediante bloques. Los proyectos creados pueden ser comentados y valorados e incluso compartidos con los demás participantes. De esta manera se construye un espacio de colaboración e intercambio, ya que éstos pueden ser reinventados o continuados por otros, generando versiones remix.

Entonces, la evaluación se realizará sobre el sitio de creación de proyectos de la plataforma ubicada en <https://scratch.mit.edu/projects/editor/>. A éste se accede desde la página inicial ingresando en la opción *Crear*.

Como muestra la figura 2, la interfaz presenta secciones donde se crean los objetos y los escenarios (B), una parte donde se visualizan todos los bloques que pueden ser utilizados para programar esos objetos y escenarios (C) y un espacio destinado a ordenar y agrupar los bloques que permitirán construir la programación (D). También contempla un espacio con ayudas (E). El resultado de la programación realizada puede observarse en otra área, que se activa al pulsar la bandera verde (A).

Por otra parte, este paso implica definir el nivel de adecuación o conformidad con las pautas WCAG 2.0. Cabe señalar que cada pauta recoge un conjunto de criterios de conformidad, redactados en forma de enunciados verificables sobre el contenido web. Para atender a las necesidades de los distintos grupos de personas y sus circunstancias, cada uno de los criterios de conformidad está asociado a un nivel. Los niveles de conformidad son: A (menos exigente), AA y AAA (más exigente). En este caso se ha escogido el nivel de adecuación intermedio AA como referencia para valorar el funcionamiento de la página web en su versión en idioma español en los navegadores Mozilla Firefox y Google Chrome, utilizando una tecnología asistida de lector de pantalla, que en este caso ha sido NVDA (NonVisual Desktop Access).

4.2. Explorando el sitio para seleccionar una muestra representativa

Al explorar el sitio fue posible observar que las tecnologías principales de base son Adobe Flash y JavaScript. De allí que sea necesario tener instalado Adobe Flash Player para poder acceder al contenido web. Como indica la metodología, no suele ser posible evaluar el sitio completo y se debe seleccionar una muestra de páginas que lo represente de manera

que asegure que los resultados de la evaluación reflejen la accesibilidad de todo el sitio con suficiente fiabilidad.

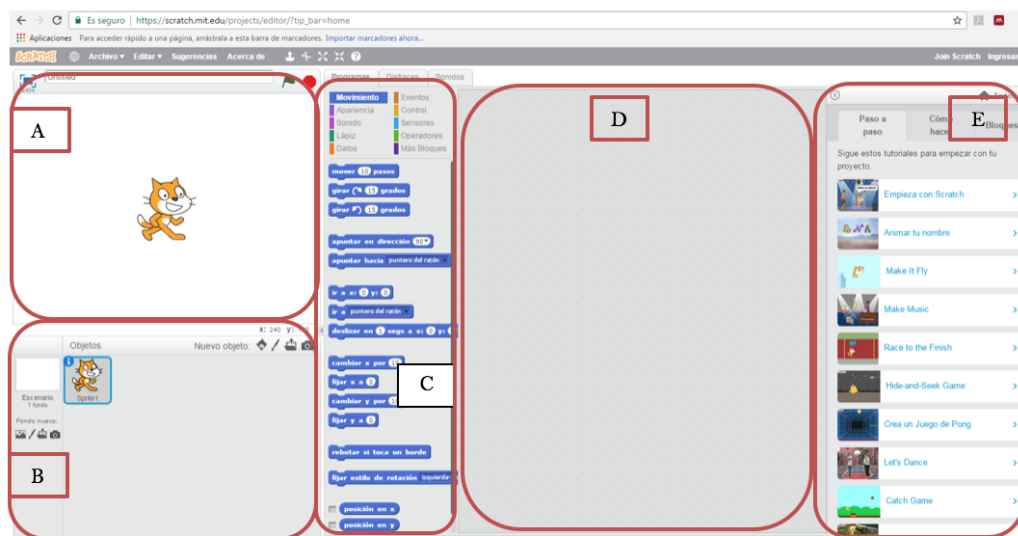


Figura 2. Interfaz para la creación de proyectos con Scratch
Fuente: Elaboración propia.

En este caso, la muestra ha sido la página de creación de proyectos, presentada en la figura 2. Esta elección se ha realizado siguiendo los criterios que la misma metodología sugiere, la cantidad de páginas que contiene el sitio (si consideramos que cada proyecto es una página, por ejemplo, y que hay millones de ellos) y la complejidad de la misma, ya que el contenido puede ser editado por los participantes, subido por ellos a distintos tipos de páginas (foros de discusión, estudios, proyectos realizados, páginas de usuarios donde éstos muestran sus creaciones). Dicha complejidad surge y es posible a partir de la creación de proyectos que habilita la página web donde está la interfaz de creación. Estos aspectos fundamentan la importancia de su evaluación, ya que nada de lo que se presenta en las demás páginas del sitio de Scratch sería posible si no existiera esta opción. De esta forma, se cumple con el criterio de buscar representatividad en cuanto a la selección de la muestra.

Por otra parte, la metodología indica la necesidad de señalar las funcionalidades del sitio-muestra, que se detallan a continuación:

Creación de objetos programables y escenarios a partir de cuatro formas diferentes (cada una implica abrir un nuevo menú, aunque no una nueva página): desde archivo, desde biblioteca de Scratch, desde una captura de pantalla o a partir de un dibujo realizado por la propia persona. En la figura 3 se muestra el proceso realizado para crear un nuevo objeto desde la biblioteca, ya que es similar al utilizado para las demás opciones. Como se observa en la barra de navegación, la página sigue siendo la misma, pero estos menús superpuestos no son reconocidos por el lector de pantalla.

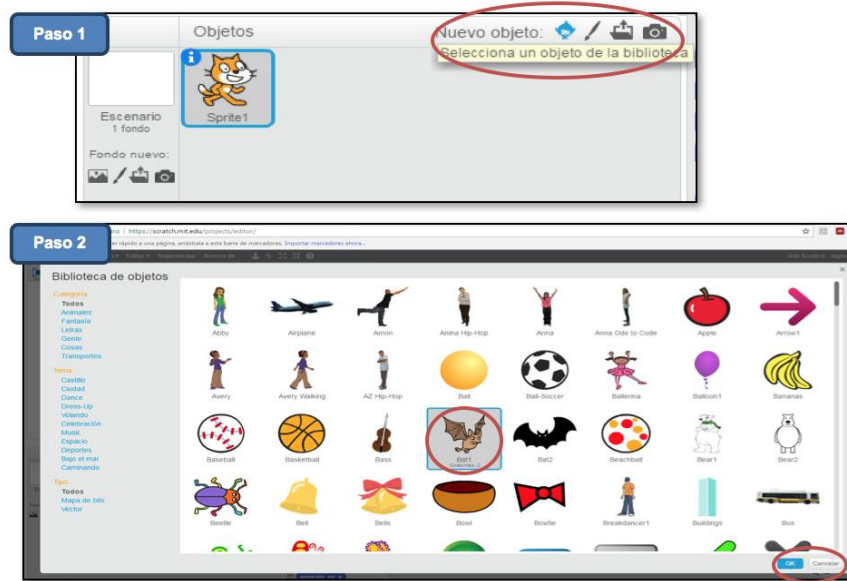


Figura 3. Creación de objetos desde biblioteca
Fuente: Elaboración propia.

Edición y programación de objetos y escenarios a partir de diferentes pestañas: una vez creado el objeto o el escenario, éste puede programarse a partir de bloques. Al entrar a la página, se muestra la pestaña de los bloques de movimiento. Es necesario entrar en otras pestañas para ver otros tipos de bloques que de entrada no aparecen abiertos, como por ejemplo los de apariencia, como se muestra en la figura 4.

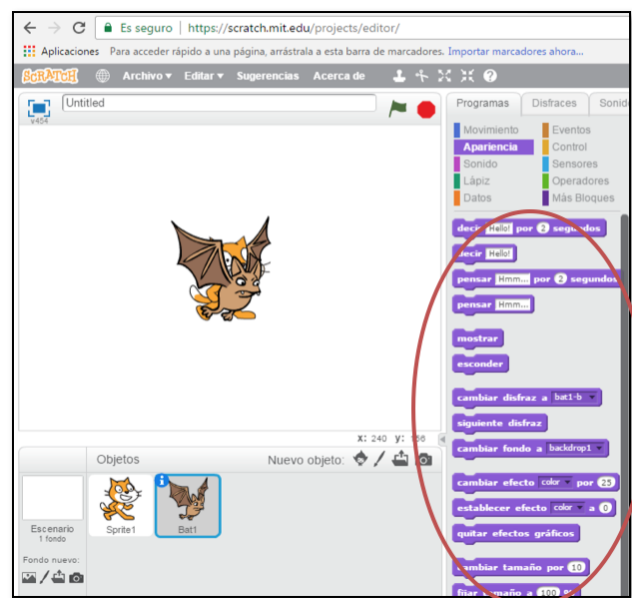


Figura 4. Bloques de Apariencia
Fuente: Elaboración propia.

Nombrar al proyecto utilizando un cuadro de texto y, en caso que se lo quiera guardar, crear una cuenta de usuario o ingresar los datos si se la ha creado previamente. Iniciar y parar el proyecto a través de dos iconos: una bandera verde y un octógono rojo. Estas características pueden observarse en la figura 5. En la imagen de la izquierda se observa la barra para escribir el nombre del proyecto (por defecto se titula "Untitled"), así como la bandera verde de inicio y el ícono para detener. En la imagen de la derecha se muestra el menú para crear un nuevo usuario, que aparece por encima de la interfaz, como una pantalla emergente.

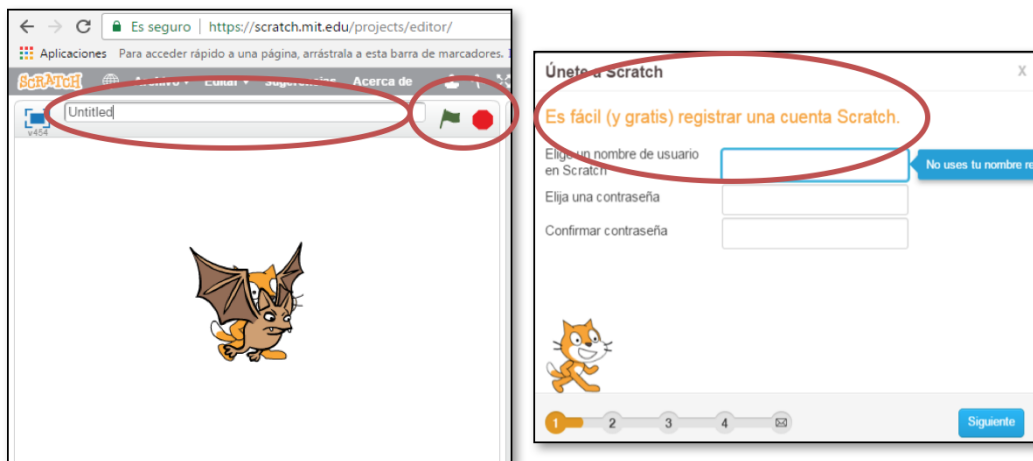


Figura 5. Cuadros de texto e iconos de inicio y parada
Fuente: Elaboración propia.

A continuación, el siguiente paso permitirá establecer el nivel de adecuación a las pautas con mayor precisión, señalando concretamente las barreras.

4.3. Evaluación de la muestra

Siguiendo la metodología descrita se presentan a continuación las diferentes problemáticas identificadas en relación a cada uno de los principios que establecen las pautas WCAG2.0. Estas pautas se organizan a partir de cuatro principios fundamentales: perceptibilidad, operabilidad, comprensibilidad y robustez. A su vez cada uno plantea una serie de pautas generales con los objetivos básicos que se deben lograr para crear un contenido accesible.

Para la evaluación se utilizaron evaluadores automáticos (Achecker, TAWy HERA) y se realizaron controles manuales detallados. En la evaluación de cada principio se incluyen ejemplos concretos que reflejan las barreras identificadas para el caso estudiado.

El primer principio plantea que la página debe ser perceptible. Por lo tanto, la información y los componentes de la interfaz deben ser presentados de manera que los destinatarios puedan percibirlos. Para cumplir con esta pauta la página debería proporcionar alternativas textuales para todo contenido no textual, de modo que se puedan convertir a otros formatos que las personas necesiten, como textos ampliados, braille, voz, símbolos o un lenguaje más simple. A su vez, tendría que proporcionar alternativas para los medios tiempo-dependientes (animaciones gráficas, audios y videos), crear contenido que pueda presentarse de diferentes formas sin perder información o estructura (por ejemplo, con una disposición más simple) y facilitar a los destinatarios ver y oír el contenido, incluyendo

la separación entre el primer plano y el fondo. El cuadro 1 muestra las dificultades arrojadas por las evaluaciones en relación con este primer principio.

Cuadro 1. Problemáticas respecto al principio 1: Perceptible

Imágenes sin alternativa textual (atributo <i>alt</i>).
Contenidos multimedia tempo-dependientes sin descripción textual o contenidos textuales equivalentes.
Controles de formularios sin etiquetar.
Opciones de los menús de selección agrupadas en formularios.
Inexistencia del elemento “H1”, que se refiere a encabezados de primer nivel, en cuanto a títulos que definen secciones dentro del documento.
Encabezados del mismo nivel seguidos sin contenido entre ellos.
La información, estructura y relaciones comunicadas a través de los bloques no están disponibles como texto.
Generación de contenido desde las hojas de estilo.
Posicionamiento de elementos mediante flotado que afecta la lectura.
La presentación visual de textos e imágenes no tiene las relaciones de contraste mínimas.
Información determinada por color, al cambiar de bloques de herramientas.
Utilización de tamaños de fuente absolutos y de medidas absolutas en elementos de bloque.

Fuente: Elaboración propia.

Concretamente, se observa que todos los objetos a programar son imágenes, por lo cual al no ofrecer alternativa textual, sólo pueden ser utilizados por personas que cuenten con buena visión. Lo mismo ocurre con los bloques de programación y con diversos iconos que forman parte de la plataforma (como el botón de comenzar y parar, o para achicar o agrandar objetos). Asimismo, los objetos pueden ser editados dentro de la propia interfaz. Scratch ofrece un menú relativamente amplio de opciones para cambiar el color, tamaño e incluso dibujar objetos originales. Las dificultades se observan también en esa parte de la interfaz. Aunque se pueden escoger objetos de la biblioteca, esta opción también presenta las mismas dificultades.

Por otra parte, presenta contenidos multimedia tempo-dependientes y sincronizados, sonidos y videos, estos últimos dentro de un menú de ayudas. En esta herramienta, los sonidos constituyen un aspecto importante en la creación de proyectos. Los objetos pueden contar con diferentes sonidos que pueden exportarse, editarse e incluso ser grabados desde la propia interfaz, como se observa en la figura 6.

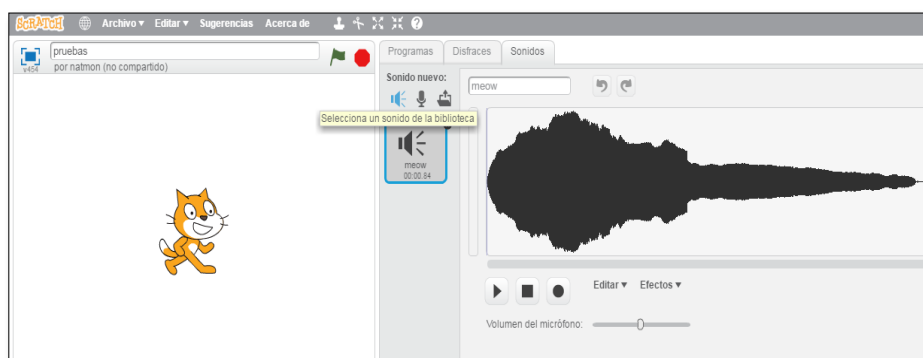


Figura 6. Menú de sonidos en la interfaz

Fuente: Elaboración propia.

La evaluación mostró que este tipo de funcionalidades no ofrece descripciones textuales o contenidos textuales equivalentes, dificultando nuevamente la accesibilidad a la

posibilidad de programación que ofrece la plataforma. Sin embargo, se observó en los chequeos manuales que al pasar con el ratón por sobre algunos iconos (figura 6), correspondientes a seleccionar sonido de la biblioteca, los mismos revelan una breve descripción textual, aunque no fueron reconocidos por el lector de pantalla. Esta situación se registró también en otros casos, pero no en todos los presentes en la interfaz, es decir, que sólo algunos tienen su descripción textual.

Los demás puntos recogidos en la evaluación del principio 1, son relativamente sencillos de identificar en la propia interfaz, como por ejemplo las relaciones mínimas en cuanto a contraste de colores, o el posicionamiento de elementos mediante flotado, como los menús emergentes que se mencionaron previamente para crear el usuario.

Por su parte, el principio 2 plantea que los componentes de la interfaz y la navegación deben ser operables. Las pautas incluidas en este principio apuntan a proporcionar acceso a toda funcionalidad mediante el teclado, proporcionar a los destinatarios el tiempo suficiente para leer y usar el contenido, no diseñar contenido que podría provocar ataques, espasmos o convulsiones y proporcionar medios para ayudarlos a navegar, encontrar contenido y determinar dónde se encuentran. En este caso, la evaluación arrojó las siguientes problemáticas, recogidas en el cuadro 2.

Cuadro 2. Problemáticas respecto al principio 2: Operable

No se proporciona acceso a la mayoría de las funcionalidades mediante teclado.
Trampas para el foco del teclado.
Foco prácticamente no visible; la mayoría de los objetos y menú de opciones que se utilizan para programar no reciben el foco.
En la sección de ayudas hay elementos dinámicos que no pueden detenerse ni ajustar la velocidad.
Se supera el umbral de destellos, lo que puede provocar espasmos o convulsiones.
Páginas sin título que describan su propósito.
Enlaces sin texto significativo.
No hay mapa del sitio.
El contenido de los encabezados y etiquetas no es adecuado.

Fuente: Elaboración propia.

A partir de lo expuesto, la evaluación aporta evidencia respecto a la falta de funcionalidad por teclado, mostrando datos concretos sobre esta temática. Es por ello que no es posible mover el foco mediante el teclado y navegar por la página. Las demás barreras señaladas se relacionan con estas cuestiones, explorándolas en mayor profundidad. Asimismo, se observan nuevamente barreras relacionadas con el menú de ayudas, donde se incluyen elementos dinámicos, pequeños videos, que no resultan accesibles. Esta sección de ayudas se presenta en la sección derecha de la interfaz, al pulsar sobre un pequeño icono de signo de interrogación. Es posible que en este espacio se detectaron los destellos que también deberían ser revisados.

Con respecto al principio 3, se pretende que la información y el manejo de la interfaz sean comprensibles. Con ello se pretende que los contenidos textuales resulten legibles y comprensibles, que las páginas web aparezcan y operen de manera predecible y que se ofrezcan ayudas que permitan a los participantes evitar y corregir errores. El cuadro 3 señala las dificultades detectadas en relación con este principio.

Cuadro 3. Problemáticas respecto al principio 3: Comprensible

- No se puede identificar el idioma del documento.
- No se presentan etiquetas de texto.

Fuente: Elaboración propia.

Aunque Scratch ofrece la posibilidad de cambiar el idioma de la interfaz, contando con más de 30 opciones, no es posible identificar el idioma del documento y sólo se puede acceder al menú desde el cual se realiza el cambio a través del uso del ratón, como se muestra en la figura 7. De esta forma, el potencial de alcance que ofrece Scratch para que las personas cuenten con todos los menús y los bloques en sus propios idiomas, no resulta un atributo posible de ser utilizado por un amplio abanico de participantes. Finalmente, en el principio 4, se señala que el contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de aplicaciones, incluyendo las ayudas técnicas como los lectores de pantalla.

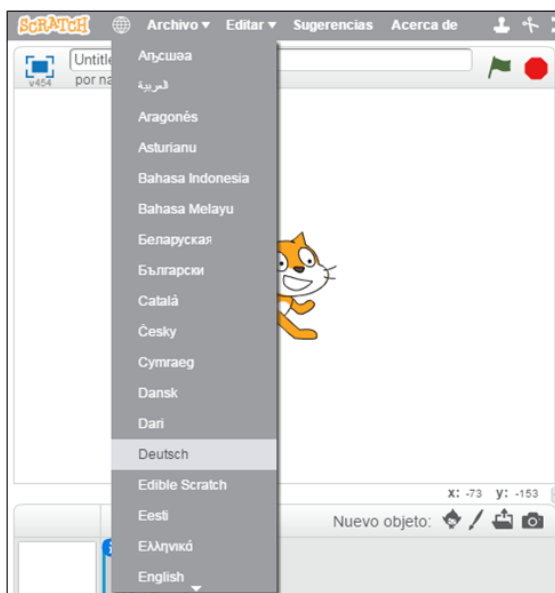


Figura 7. Menú de selección de idioma

Fuente: Elaboración propia.

Como se señaló previamente, y se comprobó con los chequeos manuales, no es posible realizar una lectura del contenido o navegar por la página a través del lector de pantalla ni utilizar sus funcionalidades. En el cuadro 4 se exponen algunas cuestiones más específicas relacionadas con esta situación.

Cuadro 4. Problemáticas en relación al principio 4: Robusto

- El documento no está “bien formado”, presentando errores en la página relacionados con la estructura sintáctica de la misma, considerando el lenguaje en que se ha creado.
- Las hojas de estilo tienen errores gramaticales.
- Controles de formularios sin etiquetar.
- Marcos sin etiqueta.
- No pueden determinarse por software (incluyendo ayudas técnicas) el nombre y/o la función de todos los componentes de la interfaz, así como los estados, propiedades y valores asignados a los mismos, ni los cambios en estos elementos.

Fuente: Elaboración propia.

El análisis expuesto pone en evidencia importantes barreras de accesibilidad, especialmente para usuarios de lectores de pantalla, devenidas principalmente de la invisibilidad de los componentes principales de la aplicación para el foco del teclado, lo que genera una imposibilidad de operarla utilizando el mismo. Además, no se registran atajos de teclado u otro tipo de control que permitan crear una programación sin requerir del uso de ratón. Más allá de las limitaciones en su operatividad, tampoco es posible acceder a la lectura de la totalidad de elementos que se disponen en la pantalla de la interfaz utilizando dicha tecnología asistida; solamente se puede recoger la información disponible en el apartado de “ayuda” y de modo parcial, ya que como se señaló previamente, las imágenes, videos y animaciones no contienen alternativas textuales ni mecanismos para regular su funcionamiento. Asimismo, se ha observado que al configurar el navegador “sin imágenes”, la interfaz no realiza ningún cambio, es decir, se mantiene invariable en su visualización, de modo que algunos objetos gráficos no están dispuestos para ser reconocidos como imágenes.

5. Conclusiones

Al implementar los pasos expuestos en la metodología y sistematizar la evaluación de accesibilidad, fue posible identificar las funcionalidades básicas de la herramienta, contextualizando el posterior análisis según las pautas. De esta forma, no quedan dudas del potencial que tiene esta interfaz en cuanto a las posibilidades que ofrece para la creación de proyectos de programación, lo intuitiva de la misma desde lo visual, y la sencillez de la programación facilitada por los bloques y por la disposición de los elementos dentro de la página. Sin embargo, el estudio detallado siguiendo la metodología propuesta pone en evidencia que programar con Scratch aún no facilita la inclusión de un amplio colectivo de posibles usuarios.

El análisis realizado evidenció que el sitio de creación de proyectos en línea de Scratch no cumple con múltiples indicadores correspondientes a las pautas de accesibilidad WCAG 2.0. En los cuatro principios revisados se identificaron barreras que impiden a un sector de la población la percepción, operación y comprensión de la interfaz, no siendo a su vez suficientemente robusta. A partir de los antecedentes expuestos y los marcos legales internacionales acerca de los derechos de la ciudadanía respecto de la accesibilidad web y la educación inclusiva, la persistencia de estas problemáticas plantea profundas interrogantes sobre cómo acortar distancias entre los discursos promocionales sobre el acceso que brindan este tipo de herramientas de programación y la realidad de los contextos de las diversas prácticas educativas.

Ante esta situación y a los fines de construir un camino válido hacia la toma de conciencia sobre las problemáticas de accesibilidad que aún persisten en la web, es importante mencionar que algunas de las opciones que ofrece la herramienta y que se presentan como barreras, pueden a su vez ser consideradas como potencialidades para diseñar proyectos accesibles. Por ejemplo, la posibilidad de utilizar sonidos abre nuevas opciones a la hora de crear juegos, animaciones y otras producciones audiovisuales para personas con diferentes diversidades funcionales. En este sentido, presentar las instrucciones o describir las situaciones que tienen lugar a lo largo del proyecto en formato sonoro, permitiría el acceso a dichos contenidos. De la misma forma, la herramienta permite incluir diálogos a partir de cuadros de texto, facilitando también el acceso por esa vía. Estas opciones se constituyen en posibles caminos a transitar si se considera la accesibilidad en el proceso

de diseño de las propuestas educativas, en el marco de una educación que asegure la sostenibilidad de prácticas inclusivas. Para ello resulta imprescindible identificar las barreras que presentan las herramientas, ya que esto posibilita la investigación y búsqueda de soluciones creativas sobre las formas de superarlas, permitiendo plantear producciones de diversa índole que promuevan el diseño de dichas prácticas. De esta forma será posible abrir el abanico de recursos hacia otras herramientas que, aunque menos populares, ofrecen alternativas válidas para la enseñanza de la programación y el desarrollo del pensamiento computacional.

En consecuencia, resulta indispensable que quienes diseñen e implementen políticas educativas y proyectos que buscan introducir la programación y desarrollar el pensamiento computacional, tengan en consideración las posibilidades reales de participación que permite este tipo de interfaz. En este sentido, es necesario considerar a los colectivos institucionales, particularmente cuando se elaboren materiales docentes, propuestas de formación o prácticas escolares, independientemente del nivel educativo donde se efectúen. Asimismo, estas cuestiones deben ser abordadas dentro de los procesos de formación y capacitación docente, aportando los recursos y conocimientos necesarios para diseñar, implementar y evaluar prácticas accesibles que respondan a las singularidades de los participantes.

Finalmente, el sentido último de la evaluación expuesta en este trabajo advierte sobre la necesidad de contar con tecnologías accesibles que permitan aprender a programar desde un enfoque no excluyente y complejo, en sintonía con el ideario de la educación inclusiva. Por tanto, al diseñar e impulsar políticas y prácticas educativas innovadoras integrando TICS, no se trata sólo de pensar en los artefactos tecnológicos y su potencial educativo. Es prioritario desarrollar una metodología de trabajo interdisciplinar que posibilite garantizar el derecho a la educación respetando al otro en su singularidad. Sólo así será posible la construcción de un compromiso ético responsable y plural que promueva la equidad en el acceso al aprendizaje y la información para la ciudadanía.

Referencias

- Acosta-Vargas P., Luján-Mora S., Acosta T. y Salvador-Ullauri L. (2018). Toward a combined method for evaluation of web accessibility. En A. Rocha, T. Guarda (Eds.), *Proceedings of the international conference on information technology & systems*. Cham: Springer.
- Alvarado, C. (2014). CSEd Week 2013: The hour of code. *ACM SIGCSE Bulletin*, 46(1), 2-4. <https://doi.org/10.1145/2583781.2583782>
- Barreto, V. B., L'Erario, A. y Fabri, J. A. (2015, abril). Teaching programming for high school students using the lego mindstorms robot. Comunicación presentada en *10th Iberian Conference on Information Systems and Technologies*. Aveiro, Portugal.
- Booth, T. y Ainscow. M. (2015). *Guía para la educación inclusiva. Desarrollando el aprendizaje y la participación en los centros escolares*. Madrid: OEI/FUHEM.
- Brennan, K. (2013). *Best of both worlds: Issues of structure and agency in computational creation, in and out of the school*. Tesis Doctoral. Boston, MA: Massachusetts Institute of Technology.
- Brennan, K., Balch, C. y Chung, M. (2014). *Creative computing*. Boston, MA: Harvard Graduate School of Education.

- Cenacchi, M. (2015). La accesibilidad web en el marco teórico y metodológico del dispositivo hipermedial dinámico: Acerca del caso memoria y experiencia Cossetini. *Revista IRICE*, 28, 37-61.
- Coleman, C.D. (2017). An exploratory analysis of WCAG 2.0 conformance in higher education websites. En P. Resta y S. Smith (Eds.), *Proceedings of society for information technology & teacher education international conference* (pp. 389-393). Austin, TX: Association for the Advancement of Computing in Education (AACE).
- Duk, C. y Murillo, F. J. (2016). La inclusión como dilema. *Revista Latinoamericana de Educación Inclusiva*, 10(1), 11-14.
- Factorovich, P. y Sawady, F. (2015). *Actividades para aprender a Program.AR: Segundo ciclo de la educación primaria y primero de la secundaria*. Buenos Aires: Miller Ed.
- Galvis, A. H. (2014). *Las políticas TIC en los sistemas educativos de América Latina: Caso Colombia*. Buenos Aires: UNICEF.
- Grover, S. y Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
- Hubwieser, P., Armoni, M. y Giannakos, M. (2015). How to implement rigorous computer science education in k-12 schools? Some answers and many questions. *Transactions on Computing Education*, 15(2), 1-22.
- Kaleliog, F. (2015). A new way of teaching programming skills to K-12 students. *Computers in Human Behavior*, 52, 200-210.
- Kereki, I. F. y Oliveira, A. F. (2013). A national experience in training teachers: Scratch and robotics in Uruguay. *Revista de Tecnología*, 12, 15-26.
- Laitano, M.I. (2015). Accesibilidad web en el espacio universitario público argentino. *Revista Española de Documentación Científica*, 38(1), 1-17.
- Ludi, S. (2015). Position paper: Towards making block-based programming accessible for blind users. En F. Turbak (Dir.), *Proceedings - 2015 IEEE Blocks and Beyond Workshop, Blocks and Beyond* (pp. 6769-6781). Atlanta, Georgia, USA: IEEE.
- Luján-Mora, S. (2013). Web accessibility among the countries of the European Union: A comparative study. *Actual Problems of Computer Science*, 1(13), 18-27.
- Monjelat, N. (2017). Programming technologies for social inclusion. En A. Diaz, A. Casali, M. C. Rivas y A. S. Sprock (Eds.), *Twelfth Latin American conference on learning technologies*. La Plata: IEEE.
- Monjelat, N. y San Martín, P. (2016). Programar con scratch en contextos educativos: ¿Asimilar directrices o co-construir tecnologías para la inclusión social? *Praxis Educativa*, 20(1), 61-71.
- Mota, F. P. y Adamatti, D. F. (2015). Programming teaching in high schools: An analysis based on the discourse of collective subject. En *Proceedings -Frontiers in Education Conference FIE 2015* (pp. 86-91). El Paso, TX: IEEE
- Muñoz, L., Brenes, M., Bujanda, M. E., Mora, M., Núñez, O. y Zúñiga, M. (2014). *Las políticas TIC en los sistemas educativos de América Latina: Caso Costa Rica*. Buenos Aires: Unicef.
- Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. Nueva York, NY: Basic Books.
- Peñafiel, M. y Luján Mora, S. (2014). Legislación sobre accesibilidad web: una comparativa de seis países. *Revista Politécnica*, 34(2), 34-45.

- Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N. y Silver, J. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-62.
- Rodríguez, L. y Carnota, R. (2015). *Historias de las TIC en América Latina y el Caribe: Inicios, desarrollos y rupturas*. Madrid: Fundación Telefónica.
- Rodríguez, G., Pérez, J., Cueva, S. y Torres, R. (2017). A framework for improving web accessibility and usability of open course ware sites. *Computers & Education*, 109, 197-215.
- Wagner, A., Gray, J., Marghitu, D. y Stefik, A. (2016). Raising the awareness of accessibility needs in block languages. En VVAA., *Proceedings of the 47th ACM technical symposium on computing science education* (pp. 497-497). Nueva York, NY: ACM Press.
- WC3. (2008). *Web content accessibility guidelines (WCAG) 2.0*. Recuperado de: <https://www.w3.org/TR/WCAG20/>
- WC3. (2014). *Website accessibility conformance evaluation methodology 1.0*. Recuperado de: <http://www.w3.org/TR/WCAG-EM/>
- Yin, R. K. (2003). *Case study research: Design and methods*. Londres: Sage.

Breve CV de las autoras

Natalia Gabriela Monjelat

Licenciada en Psicopedagogía por la Universidad Nacional de San Martín, (Buenos Aires, Argentina). Máster en Comunicación y Aprendizaje en la sociedad digital y Dra. en Comunicación, Educación y Sociedad por la Universidad de Alcalá (Madrid, España). Actualmente es Investigadora en el Instituto Rosario de Ciencias de la Educación (IRICE: CONICET-UNR). Sus intereses investigativos incluyen: procesos de enseñanza y aprendizaje en diferentes niveles educativos mediados por tecnologías de la información y la comunicación, investigación cualitativa, pensamiento computacional y programación en contextos educativos, entre otros. ORCID ID: 0000-0002-5043-8989. Email: monjelat@irice-conicet.gov.ar

Marisa Andrea Cenacchi

Profesora de Música, Especialidad Educación Musical, Universidad Nacional de Rosario (UNR). Doctoranda en Humanidades y Artes (UNR) Mención Ciencias de la Educación. Coordinadora Académica del Campus Virtual UNR. Integrante de la Comisión Universitaria de Discapacidad UNR. Docente en el Plan Vuelvo a Estudiar Virtual, Ministerio de Educación, Prov. de Santa Fe. Becaria CONICET (4-2015/3-2017). Integrante del Programa Dispositivos Hipermediales Dinámicos, Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE: CONICET-UNR). Ha dictado capacitaciones sobre accesibilidad web y TIC aplicados a contextos educativos. ORCID ID: 0000-0002-7672-1993. Email: marixxi@hotmail.com

Patricia Silvana San Martín

Doctora en Humanidades y Artes por la Universidad Nacional de Rosario (UNR), Argentina. Miembro de la Carrera de Investigador Científico y Tecnológico del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina), categoría Independiente. Vicedirectora del Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE: CONICET-UNR). Profesora Titular de la Facultad de Humanidades y Artes (UNR). Docente-Investigadora categoría I (UNR). Su especialidad de Investigación, Desarrollo e Innovación se centra en las Tecnologías de la Información y Comunicación aplicadas a Educación. ORCID ID: 0000-0001-7543-045X. Email: sanmartin@irice-conicet.gov.ar