**EJE:** ACCESIBILIDAD Y TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

**PROYECTO DE INVESTIGACIÓN**

**“*Scratch: ¿programar para todos?***

***Análisis de adecuación a pautas de accesibilidad web WCAG 2.0”***

- Natalia Monjelat. DNI 30431603. Dra. en Comunicación, Educación y Sociedad (UAH, Madrid, España). Investigadora Asistente CONICET. Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE: CONICET-UNR). Contacto: monjelat@irice-conicet.gov.ar

- Marisa Cenacchi. DNI 26724472. Profesora en Educación Musical (UNR. Rosario, Argentina). Becaria Doctoral CONICET Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE: CONICET-UNR). Contacto: cenacchi@irice-conicet.gov.ar

**Palabras claves**: Accesibilidad web; Scratch; Pautas de Accesibilidad para el Contenido Web (WCAG) 2.0.

# Introducción

La implementación de proyectos educativos destinados a incluir actividades de programación computacional en los distintos niveles y modalidades del sistema educativo resulta una temática ponderada en la agenda educativa actual. En el contexto educativo formal argentino, se ha lanzado en el año 2014 el proyecto “*Program.ar”* ([www.programar.gob.ar/de-que-se-trata/](http://www.programar.gob.ar/de-que-se-trata/)), cuyo objetivo es acercar a niños y jóvenes al aprendizaje de las Ciencias de la Computación en las escuelas y concientizar a la sociedad en general sobre la importancia de conocer estos conceptos (Factorovich & Sawady O´Connor, 2015). Desde la página *web* del proyecto se señalan diferentes herramientas a utilizar en la enseñanza de la programación, dentro de las cuales se encuentra *Scratch* ([www.scratch.mit.edu](http://www.scratch.mit.edu)) (Resnick et al., 2009) que actualmente se utiliza para enseñar programación a personas de todas las edades en más de 30 países en diversas modalidades educativas, a través de proyectos en educación formal e informal. Esta plataforma ha sido creada siguiendo ideas construccionistas de Logo (Kafai & Resnick, 1996) y de *Etoys* (Steinmetz, 2002) y propone un lenguaje de programación basado en bloques. Su diseño original surgió en torno a las necesidades e intereses de un grupo de jóvenes que participaban en clubes extra-escolares y ofrece la posibilidad de programar a través de la exploración y la colaboración entre pares. Primeramente, la herramienta se descargaba pero desde el 2013, cuenta con una versión *online*, desde la cual se pueden crear proyectos, compartirlos, comentarlos, etc., accediendo desde un navegador *web* (Ricarose, Rusk & Resnick, 2016).

Según estadísticas recientes del MIT se han creado, a la fecha, más de 13 millones de proyectos, y sólo en Argentina hay registrados más de 37 mil usuarios. Estas cifran señalan el impacto de esta herramienta y justifican su inclusión dentro de programas destinados a la enseñanza de la programación. Asimismo, Scratch pone a disposición de los usuarios espacios de debate, foros para docentes, documentación explicativa y demás materiales que pueden facilitar su utilización.

Sin embargo, aunque esta tecnología resulta innovadora, un primer análisis observacional muestra que pese al alto grado de impacto de Scratch, la herramienta presenta barreras para la participación autónoma de quienes no pueden interactuar con las TIC mediante el uso del *mouse*, requisito que es exclusivo y excluyente para operar con dicha aplicación. Esta situación también aparece reflejada en foros de debate *online* en torno a Scratch, donde diferentes usuarios demandan una versión accesible, destacando el potencial que tiene esta herramienta para aprender programación.

Aunque algunos estudios señalan posibles tecnologías asistivas que podrían facilitar su uso (Lopez & Sanchez, 2012; Wagner, Rudraraju, Datla, Banerjee, Sudame & Gray, 2012), una revisión de antecedentes identificó una ausencia de estudios centrados en la accesibilidad *web* de la herramienta.

Desde la línea de investigación sobre "Dispositivos Hipermediales Dinámicos" se están llevando a cabo diferentes proyectos[[1]](#footnote-2) que estudian los criterios teóricos-metodológicos y pedagógicos que habiliten prácticas educativas inclusivas con el objetivo de construir redes socio-técnicas, físico-virtuales, accesibles y plurales donde se aborda, entre otras problemáticas, la enseñanza y aprendizaje de nociones de programación. En el marco de estos proyectos y considerando el alto impacto de Scratch, identificar el estado de esta herramienta en cuanto a su accesibilidad resulta una tarea ineludible. A partir de esta problemática, el presente trabajo presenta una primera evaluación de la herramienta, focalizada en la interfaz de creación de proyectos del sitio *web* de Scratch en referencia a las “Pautas de Accesibilidad para el Contenido *Web”* - WCAG 2.0.

 A continuación se presentará el marco teórico-metodológico desde el cual fue abordada la evaluación de accesibilidad de dicha herramienta, para luego reseñar brevemente las características fundamentales de Scratch, su evaluación y un primer análisis de resultados. Finalmente, se señalarán las conclusiones y prospectiva de trabajo.

***Marco teórico-metodológico***

En el presente trabajo se considera a la discapacidad como un concepto complejo, artificial, construido socialmente, que se ha modificado de acuerdo a diversos paradigmas y modelos históricos (Aguado Diaz, 1995; Pfeirrer, 2002; Rosato & Angelino, 2009; Skliar, 2002). Actualmente, los marcos legales, especialmente la Convención sobre los Derechos de las Personas con Discapacidad (CDPD) (ONU, 2006) han adoptado el llamado “modelo Social de la Discapacidad” (Palacios, 2008; Palacios & Bariffi, 2007). Desde esta perspectiva el problema se desplaza del sujeto hacia el entorno, construido de modo excluyente, diseñado de manera tal, que las problemáticas que poseen las personas con discapacidad se hacen visibles y presentes, obstaculizando su participación y/o autonomía. De esta forma, las dificultades no se ubican en las personas, sino más bien en la interacción de ellas con determinados contextos, definidos desde dimensiones físicas, sociales y culturales, en los que han de desenvolverse (Cuomo, 1994).

En esta línea, cobra sentido la noción de “diversidad funcional”, propuesta por Romanach y Lobato en el “Foro de Vida Independiente” en el año 2005, ya que no conlleva un carácter negativo ni médico, poniendo énfasis en la diferencia o diversidad, como valores que enriquecen al mundo en que vivimos (Romanach & Lobato, 2005). No obstante, se observa que en muchas oportunidades el término “diversidad funcional” se utiliza como reemplazo directo del término “discapacidad”, manteniendo su uso para la identificación de un colectivo determinado, distinguiendo personas con diversidad funcional de las personas sin diversidad funcional, lo que deriva nuevamente en una demarcación binaria que remite al concepto de déficit tergiversando los alcances de los postulados teóricos. Una comprensión profunda de la génesis de lo propuesto puede habilitar cambios constructivos en los discursos y en las prácticas si se considera a la diversidad funcional como una característica intrínseca a todas las personas, y no sólo de determinados colectivos. En efecto, el concepto no remite a una falta en relación con una supuesta completud o normalidad sino a lo que, en mayor o menor grado, las diferencias culturales, etarias, preferencias cognitivas, gustos, situaciones circunstanciales, etc., dan consistencia a lo diverso como un rasgo multidimiensional propio de lo humano.

En un intento de atender a esta diversidad, surge la “Iniciativa de Accesibilidad Web”, conocida como WAI (Web Accessibility Initiative), actividad desarrollada por el “World Wide World Consortium” (W3C). El W3C es una organización internacional que se ha esforzado en elaborar pautas y estándares, que guíen y permitan la elaboración de contenidos digitales accesibles, es decir, contenidos que no presenten barreras de acceso a cualquier persona, incluyendo y atendiendo especialmente a las dificultades que enfrentan quienes poseen alguna diversidad funcional específica. Aunque en esta propuesta se emplea el término “discapacidad”, las pautas que se proponen favorecen no solo a ese colectivo sino también a la sociedad en general. En este sentido, consideramos que concuerda con la concepción de diversidad funcional que se señaló previamente.

Estos estándares son normativos y diversas leyes nacionales e internacionales, como el estándar ISO/IEC 40500:2012, 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. En este sentido la W3C señala que hablar de accesibilidad *web*, por lo tanto, es hablar de un acceso universal a la *Web*, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios.

La WAI ha desarrollado dos versiones de “Pautas de Accesibilidad al Contenido en la Web” (WCAG). La primera, WCAG 1.0, publicadas en 1999, fueron desarrolladas exclusivamente para tecnologías HTML y CSS, consisten en 14 pautas que proporcionan soluciones de diseño a situaciones comunes que pueden producir problemas de acceso a la información. A su vez, las pautas contienen una serie de puntos de verificación que ayudan a detectar posibles errores. Las WCAG 2.0, publicadas por la WAI en diciembre de 2008, mejoran las WCAG 1.0 y se consideran los referentes definitivos en accesibilidad *Web*. Se organizan a partir de cuatro principios fundamentales: perceptibilidad, operabilidad, comprensibilidad y robustez, que encierran cada uno una serie de pautas generales con los objetivos básicos que se deben lograr para crear un contenido accesible. Además, se incluyen criterios de éxito verificables y técnicas recomendables para cada uno de ellos. De este modo, las WCAG 2.0 se complementan con documentos asociados que proporcionan información importante para poder entenderlas e implementarlas, pero que no son normativos, como por ejemplo: “Comprender las WCAG 2.0” (<https://www.w3.org/TR/UNDERSTANDING-WCAG20/>) o “Técnicas para las WCAG 2.0” (<https://www.w3.org/TR/WCAG20-TECHS/>).

Para esta ponencia, se siguió la metodología de evaluación de accesibilidad siguiendo las recomendaciones publicadas por la WAI en el artículo “Website Accessibility Confromance Methodology (WCAG-EM) 1.0” (<https://www.w3.org/TR/WCAG-EM/>) cuya última actualización es de julio del 2014. En la misma se pretende proporcionar una metodología fiable que se pueda utilizar internacionalmente para la evaluación de cualquier tipo de sitio *web* tomando como referencia las pautas WCAG 2.0. Esta metodología establece como requisito que el evaluador o el equipo de evaluadores tenga un profundo conocimiento de dichas pautas, del diseño accesible, de las tecnologías de apoyo y de cómo utilizan las personas con alguna diversidad funcional la *Web*, realizando en primer lugar una revisión de la muestra aplicando evaluadores automáticos, y posteriormente controles manuales detallados.

A continuación, se procederá a la descripción de la herramienta para luego presentar el análisis focalizado de la interfaz online de creación de proyectos, siguiendo la metodología descripta previamente.

# Scratch: una herramienta para la enseñanza de la programación

El proyecto Scratch, desarrollado por el grupo “Lifelong Kindergarten” perteneciente al MIT (Instituto tecnológico de Massachussets) Media Lab, comenzó en 2003 (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010). La primera versión del año 2007 incluía un software descargable que permitía elaborar las producciones y una comunidad online diseñada como fuente de ideas, para proporcionar una audiencia para las creaciones y para fomentar la colaboración entre los miembros (Monroy-Hernandez, 2012) a través de componentes similares a los que se encuentran en otras redes sociales (Me gusta, favoritos, etc.). Una segunda versión de Scratch (Scratch 2.0) fue lanzada en 2013 y es la que se utiliza actualmente. La misma ha sido creada con Adobe Flash, usando JavaScript para los elementos interactivos[[2]](#footnote-3). De esta forma, esta versión simula una aplicación de escritorio, pero opera sobre un navegador. En esta versión, los usuarios pueden crear una cuenta y guardar sus proyectos en la “nube”. Asimismo, incluye la posibilidad de observar por dentro cualquier proyecto, teniendo acceso a toda la programación realizada por otros usuarios. Esto ha dado lugar a una cultura en torno al “remix” (manteniendo la autoría del original) que es fomentada por la propia plataforma.

El lenguaje de programación de Scratch cuenta con alrededor de 100 bloques de programación agrupados en categorías (movimiento, apariencia, sonido, lápiz, datos, eventos, control, sensores, operadores, más bloques). En los proyectos de Scratch los bloques se utilizan para manipular los atributos de los objetos (sprites).

En la figura 1, se presenta la interfaz desde la cual se realizan los proyectos, creando objetos, escenarios y programando los mismos a partir de los bloques antes mencionados, que se encuentran agrupados por colores que facilitan el reconocimiento de la categoría a la que pertenecen. Es esta la interfaz online que ha sido analizada en el presente trabajo.

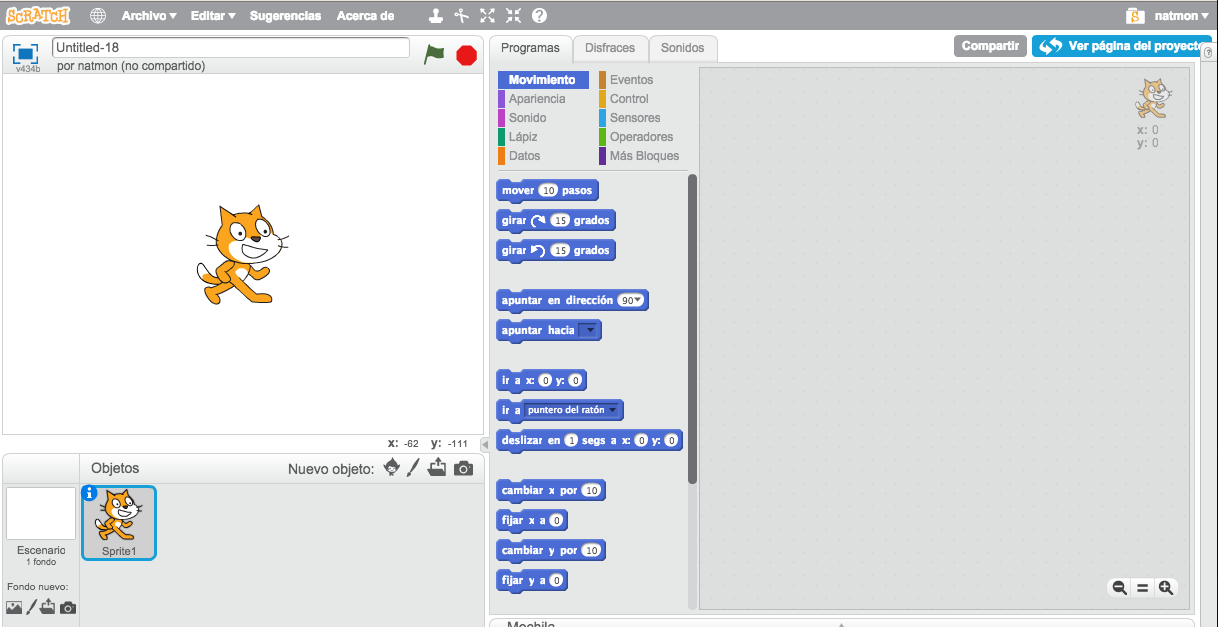


Figura 1. Interfaz para la creación de proyectos con Scratch

Por su parte, la figura 2 muestra un proyecto avanzado, que cuenta ya con varios objetos programados y escenarios creados. Se puede observar entonces como los bloques se organizan dando lugar a diferentes acciones que son llevadas a cabo por los objetos. Como se observa en las imágenes, la aproximación visual a la programación lo diferencia de otros lenguajes de programación, como C, Java o Python, que requieren que el programador escriba las instrucciones (Monroy-Hernández, 2012), o donde aparecen mensajes con errores. En Scratch, la gramática visual de las formas de los bloques y las reglas de combinación (no todos los bloques se combinan con todos) tienen el mismo rol que la sintaxis en los lenguajes basados en texto (Maloney et al., 2010).

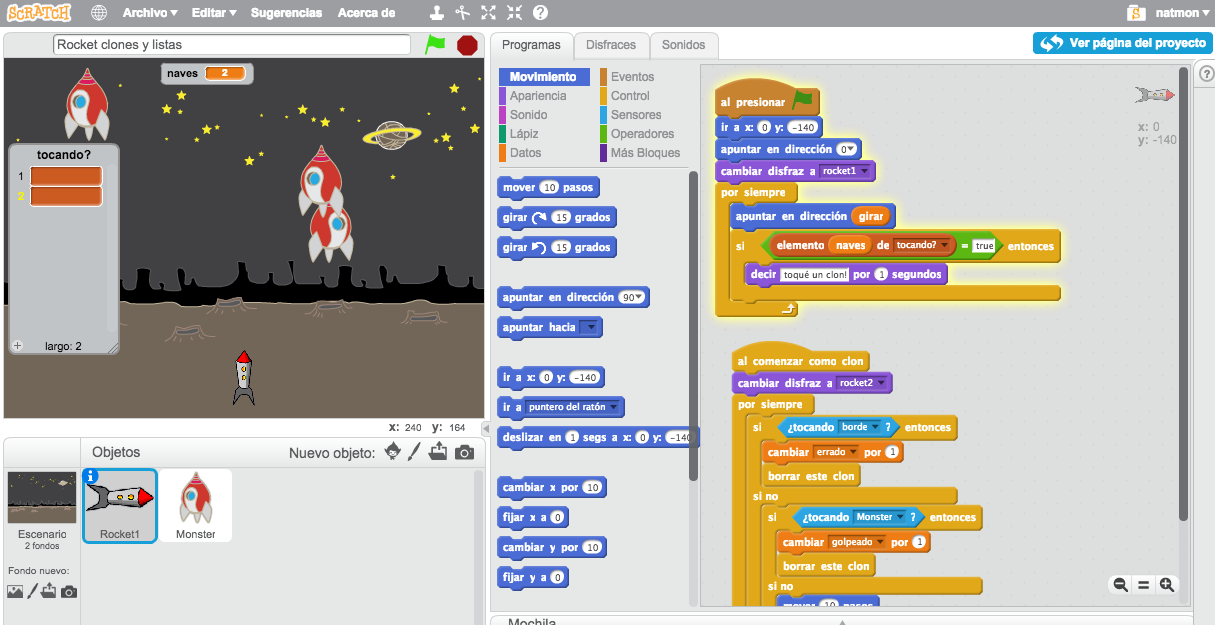


Figura 2. Proyecto avanzado realizado con Scratch

Por otra parte, la comunidad online de Scratch facilita el trabajo colaborativo de jóvenes ubicados en diferentes puntos del globo (Aragon, Poon, Monroy-Hernández, & Aragon, 2009). Las personas son consideradas creadoras activas formando parte de la denominada “cultura participativa” (Jenkins, 2009) y dejando de ser solo consumidoras de medios, y en este caso de medios interactivos.

Como se puede observar, las posibilidades que ofrece Scratch son diversas, permitiendo no solo el acceso a la programación, sino también, a que, una vez creado un proyecto, el mismo pueda ser puesto a disposición de la comunidad, generando un intercambio entre los usuarios que enriquece a todos. Como podemos observar, la interfaz de creación es el punto clave de toda la iniciativa, la participación en la misma es el fruto de años de investigaciones y mejoras, aunque la accesibilidad puede que sea aún un tema pendiente.

# Evaluación de Accesibilidad

Como se señaló previamente, un primer análisis observacional permitió identificar algunos obstáculos en el acceso a la programación de proyectos con Scratch, lo cual motivó este estudio. A partir de esa observación, y siguiendo los pasos propuestos en la Metodología WCAG-EM 1.0 adoptada, para este trabajo se definió la muestra en un recorte del sitio que se corresponde con la página *Web* que contiene la aplicación de creación de proyectos, ya que, como señalábamos, supone la esencia misma de Scratch en su propuesta de enseñanza y aprendizaje de programación a través de bloques. 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 usuarios, subido por ellos y organizado según sus propios criterios y los distintos tipos de páginas (foros de discusión, estudios, proyectos realizados, páginas de usuarios donde éstos muestran sus creaciones). Sin embargo, esta complejidad surge y es posible a partir de la creación de proyectos que habilita la página *web* donde está la interfaz de creación. De ahí que resulte importante 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 las distintas páginas que se presentan y la funcionalidad.

La creación de proyectos se realiza a través de una interfaz que está alojada en un sitio *web*. Se ha escogido el nivel de adecuación AA como referencia para valorar el funcionamiento de la página *web* en su versión en idioma español en los navegador Mozilla Firefox y Google Chrome, utilizando la tecnología asistiva NVDA.

Como se mencionó previamente las tecnologías principales de base son Adobe Flash y JavaScript. Es por esto que es necesario tener instalado Adobe Flash Player para poder acceder al contenido *web*.

Respecto a la funcionalidad de la *web*, se identifican las siguientes:

- 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 el usuario.

- Edición y programación de los objetos y escenarios a partir de diferentes pestañas. Cada una de estas pestañas presenta una serie de opciones. 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.

- Nombrar al proyecto utilizando un cuadro de texto donde. En caso que se lo quiera guardar, es necesario tener creada una cuenta de usuario.

- Iniciar y parar el proyecto a través de dos iconos: una bandera verde y un octágono rojo.

Por otro lado, la página contiene distinto tipo de contenido: imágenes, sonidos, herramientas de edición.

Para la revisión de la muestra, en primer lugar se utilizaron evaluadores automáticos (Achecker, TAW y HERA) para posteriormente realizar controles manuales detallados. Estos resultados se presentan en el siguiente apartado.

# Presentación de resultados

Siguiendo la metodología descripta se relevaron las siguientes problemáticas, que se presentan a continuación en relación a cada uno de los principios:

1) Perceptible:

* Imágenes sin alternativa textual (atributo *alt*)
* Contenidos multimedia tempodependientes (animaciones gráficas y videos) sin descripción textual o contenidos textuales equivalentes. (Ayudas)
* Controles de formularios sin etiquetar
* Opciones de los menús de selección agrupadas en formularios
* Inexistencia del elemento “H1”
* 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. (ver)
* Utilización de tamaños de fuente absolutos y de medidas absolutas en elementos de bloque.

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 ataques, 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

3) Comprensible:

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

4) Robusto:

* El documento no está “bien formado”
* 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 de usuario, así como los estados, propiedades y valores asignados a los mismos, ni los cambios en estos elementos.

Este listado pone en evidencia grandes 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 *mouse*. 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 asistiva; solamente se puede recoger la información disponible en el apartado de “ayuda” y de modo parcial, ya que las imágenes, animaciones y videos no contienen alternativas textuales ni mecanismos para regular su funcionamiento.

Además, se ha observado al configurar el navegador “sin imágenes”, que la interfaz no realizaba ningún cambio, es decir, se mantiene invariable en su visualización. Ello se debe a que, la aplicación *web*, emula visualmente a una aplicación de escritorio, sumando procesos en segundo plano mediante Javascript, y algunos objetos gráficos no están dispuestos para ser reconocidos como imágenes.

# Discusión y conclusiones

El presente trabajo ha presentado una primera evaluación de accesibilidad sobre una aplicación *Web* de gran impacto, utilizada por múltiples instituciones y organismos en iniciativas tanto formales como informales para la enseñanza de la programación informática. A partir de la misma, se han relevado importantes barreras de accesibilidad, que corroboraron el análisis observacional inicial.

Las evaluaciones realizadas siguiendo la metodología WCAG-EM 1.0 identificaron que la *web* de creación de proyectos online de Scratch no cumple con múltiples indicadores correspondientes a las pautas de accesibilidad *web* WCAG 2.0. En consecuencia, el contenido no resulta accesible a todos los usuarios.

Es necesario tomar conciencia de las posibilidades reales de participación de todos y cada uno de los estudiantes en los actuales programas que se impulsan, incluyendo aquellos que pretenden enseñar a programar, donde herramientas como Scratch son ampliamente utilizadas y difundidas. En este sentido, al diseñar y generar prácticas educativas de programación informática, no se trata sólo de pensar en los artefactos tecnológicos y su potencial educativo, sino también de atender a la diversidad y garantizar el derecho a la educación a todos los ciudadanos.

Es por ello, que visibilizar la no-accesibilidad resulta una tarea imprescindible para no caer en la generación de prácticas que resulten excluyentes para determinados colectivos sociales, limitando de este modo el cumplimiento de sus derechos socio-técnicos.

Como señalamos previamente, Scratch en su concepto visual y modular, de fácil manejo mediante el *mouse*, simula una aplicación de escritorio sin dejar de operar sobre un navegador. Esta emulación que proporciona una experiencia más rica para el usuario, genera barreras si no se tienen en cuenta disposiciones especiales que visibilicen la funcionalidad, estados, propiedades y actualizaciones de sus componentes a las tecnologías asistivas.

Asumir la responsabilidad ética de responder a y ante las diversidades funcionales es un compromiso que debe asumirse día a día en el desafío de construir una sociedad plural, más justa y equitativa. En este sentido,  Argentina suscribe a la “Convención sobre los Derechos de las Personas con Discapacidad” (ONU, 2006) mediante la aprobación de la Ley 26.378 en mayo de 2008, que señala que las personas con discapacidad tienen derecho a acceder a la información en igualdad de condiciones con los demás. Asimismo, se dio un paso adelante en esta temática con la “Ley de accesibilidad de la información en las páginas web” (Ley 26.653. Año 2010) que reconociendo la necesidad de facilitar el acceso a los contenidos de las páginas Web a todas las personas con discapacidad, determina que todos los entes pertenecientes al Estado Nacional deberán respetar en la elaboración de sus contenidos digitales las normas y requisitos de accesibilidad Web que establezca la ONTI (Oficina Nacional de Tecnologías de la Información), que en la actualidad se corresponden con las Web Content Accesibility Guideline (WCAG) 1.0, con un Nivel de Conformidad AA, elaboradas por la WAI (Web Accessibility Initiative), iniciativa del W3C (World Wide Web Consortium). Por consiguiente, es necesario avanzar para que la oferta de aplicaciones web que se realiza desde los programas nacionales contemple esta legislación, promoviendo el uso de herramientas online accesibles, o al menos, visibilizando las posibles limitaciones de las mismas y ofreciendo alternativas.

Por ello, evaluaciones como la expuesta advierten sobre la importancia del diálogo en torno a la necesidad de contar con tecnologías accesibles que permitan aprender a programar desde un enfoque no exclusivo y complejo. Avanzar en esa dirección hacia un reconocimiento en profundidad de las herramientas que se promueven para aprender programación, permitirá proponer alternativas para una participación real de todas las personas en las propuestas pedagógicas de programación  que se desarrollen a futuro con Scratch. Considerando que el código fuente de Scratch 2.0 está oficialmente disponible en acceso abierto[[3]](#footnote-4), y que existen actualmente además de las pautas de accesibilidad *web*, las indicaciones WAI - ARIA (Accessible Rich Internet Applications)[[4]](#footnote-5) sobre cómo hacer accesibles los contenidos dinámicos, sería posible señalar como prospectiva de trabajo, la generación de alianzas socio-técnicas que posibiliten espacios de reflexión y desarrollo sobre los posibles cambios que podrían realizarse desde este marco, hacia una versión de Scrath accesible.

# Referencias bibliográficas

Aguado Díaz, A.L., (1995). *Historia de las deficiencias*. Escuela Libre EditorialFundación ONCE: Madrid.

Aragon, C., Poon, S., Monroy-Hernández, A., & Aragon, D. (2009). A tale of two online communities: fostering collaboration and creativity in scientists and children. In *Proceeding of the seventh ACM conference on Creativity and cognition* (pp. 9–18). Berkeley, California: ACM Press.

Cuomo, N. (1994). *La integración escolar¿Dificultades de aprendizaje o dificultades de enseñanza?* Madrid: Visor.

Factorovich, P., & Sawady O´Connor, F. (2015). *Actividades para aprender a Program.AR: Segundo Ciclo de la Educación Primaria y Primero de la Secundaria*. (I. Miller, Ed.). Buenos Aires: Presidencia de la Nación. Retrieved from <http://programar.gob.ar/descargas/manual-docente-descarga-web.pdf>

ISO/IEC (2012). ISO/IEC 40500. Information technology. W3C Web Content Accessibility Guidelines (WCAG) 2.0. Disponible en <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=58625>

Jenkins, H. (2009). Confronting the Challenges of Participatory Culture: Media Education for the 21st century. MacArthur Foundation.

Kafai, Y. B., & Resnick, M. (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Hillsdale, NJ: Erlbaum.

López, C., & Sánchez, R. S. (2012). Scratch y Necesidades Educativas Especiales: Programación para todos. *Revista de Educación a Distancia*, *34*. Retrieved from http://revistas.um.es/red/article/view/233521

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, *10*(4), 1–15. http://doi.org/10.1145/1868358.1868363

Monroy-Hernández, a. (2012). *Designing for remixing: supporting an Online community of amateur creators*. Massachusetts Institute of Technology. Retrieved from <http://dspace.mit.edu/handle/1721.1/78202>

Organización de Naciones Unidas (2006) *Convención sobre los derechos de las personas con discapacidad*. Sede de las Naciones Unidas. Nueva York.

Palacios, A., (2008). El modelo social de discapacidad: orígenes, caracterización y plasmación. *Proceedings Convención Internacional sobre los Derechos de las Personas con Discapacidad*. CERMI. Madrid.

Palacios, A., & Bariffi, F. (2007). *La discapacidad como una cuestión de derechos humanos. Una aproximación a la Convención Internacional sobre los Derechos de las Personas con Discapacidad*. Telefónica, CERMI, Ediciones Cinca:Madrid.

Pfeiffer, D. (2002). The philosophical foundations of disability studies. *DisabiIity Sdudies Quarterly*. Spring. Vol. 22 (2) 135- 147.

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J. (2009). Scratch: Programming for All. *Communications of the ACM*, *52*(11), 60.

Romañach, J., Lobato, M., (2005). Diversidad funcional, nuevo término para la lucha por la dignidad en la diversidad del ser humano. Foro de Vida Independiente. España. Consultado en Agosto 1, 2014 de http://www.forovidaindependiente.org/files/documentos/pdf/diversidad\_funcional.pdf

Roque, R., Rusk, N., & Resnick, M. (2016). Mass Collaboration and Education. In U. Cress, J. Moskaliuk, & H. Jeong (Eds.), *Mass Collaboration and Education* (pp. 241–256). Springer. <http://doi.org/10.1007/978-3-319-13536-6>

Rosato, A., Angelino, M. A. (2009). *Discapacidad e ideología de la normalidad. Desnaturalizar el déficit*. Buenos Aires: NovEduc.

Skliar C. (2002). *¿Y si el otro no estuviera allí? Notas para una pedagogía (improbable) de la diferencia.* Buenos Aires: Miño y Dávila.

Steinmetz, J. (2002). Computers and squeak as environments for learning. In M. Guzdial & K. Rose (Eds.), *Squeak: Open Personal Computing and Multimedia* (pp. 453–482). Upper Saddle River, NJ: Prentice-Hall, Inc.

Wagner, A., Rudraraju, R., Datla, S., Banerjee, A., Sudame, M., & Gray, J. (2012). Programming by voice: a hands-free approach for motorically challenged children. *Proceedings of the 2012 ACM Annual Conference Extended Abstracts on Human Factors in Computing Systems Extended Abstracts - CHI EA ’12*, (February 2016), 2087. http://doi.org/10.1145/2212776.2223757

1. Proyecto “Construir y sostener un Dispositivo Hipermedial Dinámico para aprender a programar en el nivel primario de escolaridad fundamentado en una perspectiva activa integradora” Investigadora Asistente Dra. Natalia Monjelat , Proyecto “Recursos Educativos Abiertos y Accesibilidad-DHD en la Educación  Musical: hacia la construcción de criterios teórico-metodológicos inclusivos en atención a estudiantes con diversidad funcional por limitación visual“ Doctoranda Marisa Cenacchi y Proyecto Grupal Intedisciplinario de Investigación y Desarrollo “Construir Dispositivos Hipermediales Dinámicos lúdicos, inclusivos y accesibles”; todos ellos dirigidos por Dra. Patricia San Martín. Más información: <http://goo.gl/3Wzdj5> [↑](#footnote-ref-2)
2. Información extraída de la wiki de Scratch: <http://wiki.scratch.mit.edu/wiki/Programming_Language> [↑](#footnote-ref-3)
3. Disponible en el siguiente enlace: https://github.com/LLK/scratch-flash [↑](#footnote-ref-4)
4. Para mayor información: https://www.w3.org/WAI/intro/aria [↑](#footnote-ref-5)