

UNIDAD: 02024400 ==> CENTRO INV EN TECNL. DE INF. Y COMUNICACIÓN

Nom.: Proyecto Pry01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad Subactividad Código inscripción
Investigación Investigación Aplicada

Vigencia del Proyecto: 01/07/2015 al 30/06/2017

Descripción:

Los niños que nacieron en la Era de la Información son nativos digitales. Esta característica puede ser aprovechada para mejorar el proceso de aprendizaje en las disciplinas STEM (Science, Technology, Engineering, and Math) a través del uso de la tecnología. En este proyecto se aborda el desarrollo de un entorno de programación colaborativo para fomentar el desarrollo del pensamiento lógico-matemático y la solución de problemas en niños entre 4 y 6 años. Con esta propuesta se pretende contribuir y estimular la integración de la tecnología en la educación, para apoyar el aprendizaje de disciplinas STEM mediante el aprendizaje lúdico con la utilización de un entorno colaborativo que permite a los niños de la primera infancia crear programas y ejecutarlos utilizando robots remotos. La propuesta incluye la construcción y la evaluación del entorno, proporcionando un ambiente que permita trabajar con sistemas físicos reales a través de Internet por medio de dispositivos. La arquitectura propuesta se basa en los recursos de hardware y software de código abierto, con el fin de favorecer la replicación del proyecto, y la integración de éstos en un entorno distribuido de enseñanza y aprendizaje.

Observaciones:

El sistema no permitió ingresar el período y la cantidad de horas por semana de cada participante del proyecto. Por lo que, se detalla a continuación:

Investigador Principal

110330144 Kryscia Ramírez Benavides. Periodo: Del 01/07/2015 al 30/06/2017, 13 horas por semestre.

Asociados o Colaboradores

204180223 Luis Guerrero Blanco. Periodo: Del 01/07/2015 al 30/06/2017, 4 horas por semestre.

Unidades participantes en el proyecto:

01060309 ESCUELA DE CIENCIAS COMPUT. E INFORMATICA

¿El proyecto tiene porroga? No

¿El proyecto tiene renovación? No

Relación con otros proyectos:

Proyectos:	Periodo	Estado
2714 Creación de un entorno de programación en dispositivos móviles orientado a niños entre 4 y 6 años que permita programar aplicaciones para robots	2014	Editable

Adscripciones con programas inscritos en las Vicerrectorías

Costo Total del Proyecto:

Monto estimado UCR	Entes externos	Total
0,00	0,00	0,00

UNIDAD: 02024400 ==> CENTRO INV EN TECNL. DE INF. Y COMUNICACIÓN

Nom.: Proyecto Pry01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

Entes externos:

Información acerca de los encargados del proyecto

Investigador principal / Responsable

Tipo	Identificación	1er apellido	2do apellido	Nombre	Grado	Unidad / Institución a la que pertenece	Estado en régimen
CÉDULA NACIONAL	110330144	RAMIREZ	BENAVIDES	KRYSCIA DAVIANA	Maestría		INTERINO LICENCIADO(A)

Fecha de inicio y fin del nombramiento

al

Horas por semanas	
Propia	Adicional

Colaborador / Asociado

Tipo	Identificación	1er apellido	2do apellido	Nombre	Grado	Unidad / Institución a la que pertenece	Estado en régimen
CÉDULA NACIONAL	204180223	GUERRERO	BLANCO	LUIS	Doctorado Académico	ESCUELA DE CIENCIAS DE LA COMPUTACION E INFORMATICA	INSTRUCTOR(A)

Fecha de inicio y fin del nombramiento

al

Horas por semanas	
Propia	Adicional

UNIDAD: 02024400 ==> CENTRO INV EN TECNL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

Impacto del Proyecto

Área(s) de impacto del proyecto:

¿En qué consiste el impacto?.

El impacto consiste en un beneficio para los niños de preescolar, las docentes y la sociedad en general. Se beneficiarán de la plataforma los estudiantes y los profesores de escuelas públicas. Se promoverán habilidades como: pensamiento crítico y solución de problemas; colaboración y liderazgo; agilidad y adaptabilidad; iniciativa y emprendedurismo; acceso y análisis de información; curiosidad e imaginación. Eventualmente los niños desarrollarán las competencias definidas como claves para alcanzar el éxito en la vida profesional y personal, según el modelo de las habilidades para el Siglo 21. Además, se utilizará el acercamiento lúdico a través del entorno de programación colaborativo para transmitir al estudiante diferentes conceptos matemáticos. Por lo que, viendo como los diferentes robots cobran vida y ejecutan las soluciones que ellos proponen, los estudiantes aprenden a ver la tecnología, la robótica y la programación como herramientas creativas y experimentan las recompensas de comprensión y realización. Por otra parte, toda la teoría y las prácticas utilizadas durante la investigación podrían formar parte de contenidos de un curso de HCI (Human Computer Interaction). Este curso se enfocaría en el proceso de desarrollo de aplicaciones utilizando la metodología DCU (Diseño Centrado en el Usuario), para dispositivos convencionales y móviles, donde se tomen en cuenta los principios, las guías y las heurísticas asociadas, así como analizar las características necesarias de los dispositivos donde se ejecutarán dichas aplicaciones. Además, se verían diferentes métodos de evaluación de usabilidad y funcionalidad, que los estudiantes podrían en práctica según las características de las aplicaciones que realicen. Los diferentes contenidos de este curso se impartirían a través de la realización de un proyecto a lo largo del semestre, el cual consistiría en proponer y desarrollar una aplicación que resuelva algún problema de algún público en particular.

Población Beneficiada Directa

Quien o quienes:

Beneficios para la población:

Beneficios para la Universidad:

Antecedentes del proyecto

En la década de 1970, Papert y sus estudiantes del MIT iniciaron la investigación de métodos para introducir a la niñez en la programación. Esta investigación inicial llevó a la creación del lenguaje de programación Logo y la conocida tortuga que se movía por el suelo en respuesta a los comandos simples de este lenguaje, tales como adelante, atrás, izquierda y derecha. Papert y sus discípulos lograron que Logo quedara relativamente libre de las reglas sintácticas que hacen difícil el aprendizaje de la programación en niños [1], [2].

Como parte del lenguaje Logo, se tenía una tortuga de piso (robot), que era una estructura con una pluma para dibujar figuras en una hoja grande de papel colocada en el suelo. Los niños aprendieron a enseñarle a la tortuga a dibujar formas básicas, como triángulos, cuadrados, círculos, e incluso letras. La tortuga también podría recibir instrucciones para dibujar formas complejas dibujando varias veces formas más simples, girando un poco antes de cada repetición [2], [3]. Logo adquirió gran popularidad a principios de 1980, cuando las computadoras personales (especialmente el Apple II) se hicieron comunes en las escuelas. Sin embargo, pocas tortugas robóticas llegaron a la escuela ya que los robots demostraron ser caros y poco fiables. Los investigadores del MIT temporalmente se apartaron de este enfoque, y desarrollaron la tortuga de la pantalla: una representación de la tortuga de suelo, pero en la pantalla del computador [1], [2], [4]. Logo llegó a ser un lenguaje basado en escritura con instrucciones muy fáciles de aprender. El resultado de

Fecha de impresión: 08/05/2015

Página: 3 de 20

Hora de impresión: 10.19 AM

Información general por proyecto

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

cada instrucción se veía plasmado en el recorrido efectuado por un "robot virtual" (popularmente una tortuga virtual).

En 1985 Logo Computer Systems Inc. (LCSI) introdujo el programa LogoWriter, que incluía la capacidad de procesamiento de textos (de allí el nombre). La interfaz de usuario era simple y más intuitiva [1]. LogoWriter se implementó en muchos idiomas, debido a la popularidad que alcanzó en todo el mundo.

Otra innovación de la década de los ochenta (1988) fue LEGO Logo. Mitchel Resnick y Steve Ocko, trabajando en el MIT Media Lab, desarrollaron un sistema que interconectaba Logo con motores, luces y sensores que se incorporaron en las máquinas construidas con ladrillos LEGO y otros elementos. Los sistemas de robótica con Logo no eran nuevos, pero LEGO TC Logo fue un éxito comercial que llegó a miles de maestros y estudiantes [5].

Durante esa misma época (1988), la Fundación Omar Dengo (FOD) inició en Costa Rica el Programa de Informática Educativa, apoyado por el Ministerio de Educación Pública de Costa Rica e IBM América Latina. Este proyecto puso a Logo en manos de la mayoría de los estudiantes y profesores de primaria de Costa Rica, y un proyecto posterior hizo lo mismo con los estudiantes de secundaria [6].

Una nueva versión de Logo, llamado MicroMundos, fue lanzado en 1993 por LCSI. Esta versión incorporó muchos cambios en el entorno y en el lenguaje de Logo. Se incluyeron muchas características adicionales como herramientas de dibujo, editores, fabricación de melodías, y posibilidad de importar gráficos y sonidos. Estas nuevas características, junto con Logo, permitieron apoyar la creación de proyectos multimediales, juegos y simulaciones [7]. Desde entonces, MicroMundos se ha actualizado varias veces. Actualmente se ofrece MicroMundos EX, MicroMundos JR y MicroMundos EX Robotics, que incorpora la posibilidad de programación para el robot Cricket y el ladrillo LEGO RCX.

Otra innovación que presenciamos en esa época fue el ladrillo programable LEGO, un proyecto de investigación del MIT encabezado por Fred Martin. Este ladrillo programable tenía un pequeño computador en su interior, al que se le descargaban programas escritos en un computador de escritorio. Traía su propio entorno de programación, y los programas se ejecutan de forma autónoma para crear un robot inteligente [5]. LEGO comercializó en 1994 el ladrillo programable RCX que se programaba por medio del software RoboLab. Este entorno estaba basado en íconos y permitía crear diagramas (eran los programas) que controlaban el RCX. Ofrecía modos diferentes de programación adaptados al nivel de aprendizaje de los alumnos, como Pilot e Inventor. Además, ofrecía el modo Investigator orientado a su uso en el laboratorio de ciencias [5]. En el año 2006 se comienza a comercializar el ladrillo NXT de LEGO Mindstorms. El LEGO Mindstorms es atractivo para todas las edades, y se puede utilizar para diferentes propósitos dependiendo de las habilidades de sus usuarios. Permite comunicarse con diferentes dispositivos a través de bluetooth. Además, su ambiente de programación visual permite a un programador principiante crear fácilmente un nuevo programa. El software de NXT se basa en National Instruments' Lab VIEW, que permite arrastrar íconos y soltarlos en la pantalla principal del programa para utilizarlo [8].

También se han desarrollado versiones más pequeñas del ladrillo programable, llamados crickets. Estos ladrillos se desarrollaron comercialmente con el nombre Pico-Cricket con su entorno de programación PicoBlocks [9]. Sin embargo, este ladrillo se encuentra discontinuado desde el año 2010.

Como parte del proyecto del ladrillo programable, el Media Lab del MIT creó una nueva versión de Logo llamada LogoBlocks. En lugar de escribir líneas de código, en este lenguaje los programas se construyen mediante el ajuste de piezas, como si fuera un rompecabezas [5].

Un Logo radicalmente diferente llamado StarLogo se introdujo en 1994. Fue una versión paralela desarrollada por Mitchel Resnick en el MIT. StarLogo es un entorno de programación de videojuegos en tres dimensiones, cuya sintaxis se basa en enlazar virtualmente varias cajas o bloques creando diferentes secuencias que definen el comportamiento de los objetos en el juego. Este tipo de lenguajes tienen la ventaja de que están libres de errores de sintaxis y semántica ya que las diferentes cajas representan las instrucciones (sintaxis) y cada caja solo se puede enlazar con un tipo de cajas predeterminado (metáfora de rompecabezas) solucionando así errores semánticos [10].

Aproximadamente 10 años después, aparece StarLogo TNG, la siguiente generación de StarLogo. Entre sus mejoras estaban los gráficos 3D, el sonido y una interfaz de programación basada en bloques. Permitía la entrada de datos por teclado, y esto lo convirtió en una gran herramienta para la programación de videojuegos educativos [11].

En 1999 aparece el entorno de programación Alice, el cual es un entorno innovador y libre que hace fácil crear una animación para contar una historia jugando un juego interactivo, o un video para compartir en la Web [12], [13]. Alice está programado en Java y utiliza un entorno sencillo basado en "arrastrar y soltar" para crear animaciones mediante modelos 3D [14], [15]. Este software fue desarrollado por los investigadores de la Carnegie Mellon University, entre los que destaca Randy Pausch, y fue creado para aprender programación orientada a objetos [16].

En el 2004 apareció Etoys, un entorno informático amigable para niños, basado en un lenguaje de programación orientado a objetos, el cual es utilizado para la educación. Etoys es un entorno de creación de medios con un modelo de objetos de secuencias de comandos para los diferentes objetos en que se ejecuta [17], [18]. Es código libre y abierto.

El set LEGO WeDo, diseñado por LEGO en 2005, consistía en un conjunto de piezas y partes mecánicas usadas para construir y diseñar modelos de LEGO. Traía un software fácil de usar, basado en íconos, que proporcionaba un entorno de programación intuitivo con instrucciones de montaje, ejemplos de programación y sugerencias de actividades [19]–[21].

En el verano del 2007 aparece un actor importante: Scratch. Scratch es un entorno de programación que permite a principiantes obtener resultados sin tener que aprender a escribir de manera sintácticamente correcta, facilitando el aprendizaje autónomo. Fue desarrollado por un equipo del Media Lab del MIT dirigido por Mitchel Resnick. Actualmente Scratch se utiliza en casi todo el mundo en muchos entornos diferentes: escuelas, museos, centros comunitarios y hogares.

Scratch está diseñado especialmente para personas entre 8 y 16 años de edad. Scratch es un entorno de programación simple, constituido por símbolos iconográficos denominados "bloques" [22], [23]. Este entorno aprovecha los avances en diseño de interfaces para hacer que la programación sea más atractiva y más accesible para todo aquel que se enfrente por primera vez al reto

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

de aprender a programar. Debido a la popularidad de Scratch, la programación por bloques se ha generalizado y se utiliza en muchas aplicaciones. En ese mismo año (2007) comienza la creación del software libre de código abierto miniBloq, un entorno de programación gráfico para plataformas Arduino, Multiplo y otras [24]. Creado por Julián da Silva Gillig, este entorno es utilizado con los kits educativos de robótica RobotGroup. La adopción de miniBloq en los robots educativos de RobotGroup comenzó en 2008, cuando se sumó como socio Da Silva Gillig, junto a Paves y el ingeniero Lucio Mercado. Este software, basado en C++, utiliza símbolos, permite visualizar en tiempo real los posibles errores de código y permite agregar código textual o incluso programar sin los bloques (pasa un proyecto de programación gráfica a textual, y viceversa con el proyecto ya comenzado) [24].

En el año 2010 aparece Enchanting, desarrollado por Clinton Blackmore, quien pertenece a la asociación Southern Alberta Robotics Enthusiasts (SABRE) de Canadá, junto con un grupo de colaboradores. Enchanting es un entorno de programación para LEGO Mindstorms NXT que permite a los niños programar robots LEGO Mindstorms NXT. Se basa en Scratch y BYOB/Snap!, y es soportado por leJOS NXJ (Java para el NXT) [25]. Es un entorno de programación libre y de código abierto.

En el año 2011 comienza el proyecto ScratchJr, que es liderado por Marina Umaschi Bers y su grupo de investigación DevTech en la Tufts University y Mitch Resnick del Lifelong Kindergarten del Media Lab del MIT. Este proyecto pretende desarrollar y estudiar la próxima generación de tecnologías innovadoras y materiales curriculares para la educación de la primera infancia (orientadas a niños de más de 5 años) [26]. A finales de agosto del 2014 publicaron una primera versión.

En el año 2012 surge MoWay, un pequeño robot autónomo, rápido y provisto de varios sensores. MoWay cuenta con su propio lenguaje de programación, el cual es adecuado para niños mayores de 8 años [27]. Este robot no puede cambiar su forma, lo que lo hace poco flexible, y se considera una desventaja. En abril del 2012 se ofrece en el App Store de Apple la aplicación Move The Turtle de la empresa Next is Great, es una aplicación para iPhone y iPad que enseña a los niños los fundamentos de la creación de programas informáticos, usando comandos gráficos intuitivos [28]. Esta aplicación es una versión de LOGO para dispositivos móviles de Apple. Está orientada a niños de 5 o más años para la programación básica, y a niños entre 9 y 12 años para la programación avanzada.

En junio del 2012 se lanza Kodable, una aplicación para iPad que pretende enseñar un nuevo lenguaje a niños de más de 5 años, siendo éste un lenguaje de programación [29], [30]. Los creadores de esta aplicación son Grechen Huebner y Jon Mattingly de la empresa SurfScore, Smeeborgian Central Intelligence. Esta aplicación trata de enseñar los conceptos fundamentales de programación y lógica, mediante comandos de arrastrar y colocar para llevar a un personaje de un punto inicial a un punto final [30], [31].

En agosto del 2013 se ofrece en el AppStore de Apple la aplicación KineScript, es una aplicación para iPad que proporciona un escenario vacío, y actores que se arrastran a la pantalla, a quienes se les dan comportamientos y lógica con el guión, para luego disfrutar de los resultados [32]. Es un lenguaje de programación visual en que los niños pueden aprender a programar y luego compartir sus resultados. KineScript está inspirado en Scratch del MIT.

En setiembre del 2013 se comercializa el ladrillo EV3 de LEGO Mindstorms. El LEGO Mindstorms EV3 combina la versatilidad del sistema de construcción LEGO con la más avanzada tecnología, ofreciendo una interfaz de programación fácil, intuitiva y basada en íconos [33], [34]. Además, en esta versión se puede controlar el ladrillo mediante un control remoto (incluido en el kit) y una aplicación móvil llamada "Robot Commander" (disponible para dispositivos iOS y Android) [34].

En el mismo mes de setiembre, Dan Shapiro da a conocer un juego de mesa (inspirado en Logo) que enseña de forma lúdica los fundamentos de programación a niños entre 4-8 años [35], [36]. Este juego trata, mediante tarjetas de comandos y un tablero, de llevar a las tortugas a través de un laberinto, donde el adulto es el computador quien ejecuta los comandos. Es decir, los niños dan las órdenes a los adultos [36]. Este juego está disponible desde finales de junio.

En octubre del 2013, aparece Play-i, cuyos fundadores y creadores del proyecto son Vikas Gupta, Saurabh Gupta y Mikal Greaves. En setiembre del 2014 cambiaron el nombre a Wonder Workshop. El objetivo de Play-i es que sus dos pequeños robots, Dash y Dot (inicialmente se llamaban Bo y Yana), enseñen a los niños de más de 5 años el lenguaje y los conceptos básicos de la programación a través de juegos interactivos [37]. Los dos robots actúan y se ven como juguetes, pero en realidad son máquinas de aprendizaje, que guían a los niños a través de códigos muy básicos aunados a una historia musical. Los niños simplemente deben coreografiar una secuencia de acciones desde su iPad o tableta Android, y los robots los imitarán. Actualmente están en la etapa de producción y se espera que se ofrezca comercialmente a finales de noviembre del 2014.

Aproximadamente un mes después, en noviembre del 2013, aparece el proyecto llamado Primo de Filippo Yacob, Matteo Loglio, Danilo Di Cuia, Valeria Leonardi, Lucia Rabago y Josh Valman. Primo es una interfaz de programación tangible, para niños entre 4-7 años, diseñada para enseñar a programar [38]. Cuenta con un robot que utiliza la tecnología de Arduino. Actualmente están en la etapa de producción y se ofrecerán comercialmente aproximadamente a finales del 2014.

A finales de enero del 2014, Linda Liukas propone el proyecto llamado Hello Ruby. Su idea es crear un libro de cuentos (físico y digital) que mediante dibujos y textos sencillos los niños entre 4-7 puedan aprender a programar [39], [40]. También, Linda escribirá una guía para padres y madres donde puedan aprender a programar a la vez que sus hijos y con ello les puedan orientar. Todavía no se ofrece una fecha para su publicación.

En junio del 2014 se propone el proyecto KIBO. KIBO es un kit robótico diseñado específicamente para niños de 4 a 7 años. Los niños construyen sus propios robots con KIBO, los programan para que hagan lo que ellos quieren, y los decoran. KIBO es una interfaz de programación tangible,

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

donde los niños juegan con bloques de madera que traen imágenes de comandos de programación, estos bloques pueden ser compartidos de una forma fácil [41], [42]. KIBO es el resultado de investigación de Marina Umaschi Bers y su grupo de investigación DevTech de la Tufts University's Eliot Pearson Department of Child Development. Esta investigación es la unión del software CHERP (2009) [43], [44] y el robot KIWI (2012) [41], [45], ambos son proyectos del grupo investigación DevTech.

Es importante señalar que también se han creado juegos encubriendo lenguajes de programación. Un ejemplo de esto es el juego Little Big Planet (juego de Playstation 3), un potente emulador de física donde los jugadores pueden definir sus propios escenarios mediante la definición de características físicas de materiales programando de esta manera diferentes objetos.

Estos lenguajes de programación orientados a la niñez han adquirido gran importancia en las últimas décadas en el proceso de enseñanza-aprendizaje debido a la era digital, ya que esta era, según Prensky, ha provocado la creación de dos categorías generacionales: los Nativos Digitales y los Inmigrantes Digitales. Los primeros han nacido y crecido en la era digital, rodeados de todas las herramientas tecnológicas, mientras que los segundos no nacieron en la era digital, pero adoptan las nuevas tecnologías a sus vidas [46]. Esto ha provocado que los patrones de pensamiento de las nuevas generaciones hayan cambiado y el sistema educativo tradicional no sea el más adecuado para ellos [46], [47].

Por su parte, la era digital ha llevado a crear juguetes más útiles y más funcionales gracias a la tecnología. LEGO Mindstorms NXT es un ejemplo de ello, ya que es un juguete que tiene un ladrillo inteligente que se puede programar desde una computadora, con su propio entorno de programación, para crear un robot inteligente. El LEGO Mindstorms es atractivo para todas las edades, y se puede utilizar para diferentes propósitos dependiendo de las habilidades de sus usuarios. También puede comunicarse con diferentes dispositivos a través de Bluetooth. Además, su entorno de programación visual permite a un programador principiante crear fácilmente un programa. El software de NXT se basa en National Instruments' Lab VIEW, que consiste en arrastrar íconos y soltarlos en la pantalla principal del programa [8], [48].

La mayoría de los entornos de programación discutidos en esta sección tienen la desventaja que no pueden ser utilizados por la niñez que todavía no sabe leer, o sea, diseñados para niños mayores de 6 años, dejando por fuera a los niños que no han adquirido esta habilidad. Son pocos los entornos que se han centrado en niños de preescolar.

REFERENCIAS

- [1] M. Tempel, "Logo: A Language for All Ages," *Comput. Sci. K-8 Build. a Strong Found.*, no. Special, pp. 16-17, 2012.
- [2] T. S. McNerney, "From turtles to Tangible Programming Bricks: explorations in physical language design," *Pers. Ubiquitous Comput.*, vol. 8, no. 5, pp. 326-337, Jul. 2004.
- [3] S. Papert, *Mindstorms: children, computers, and powerful ideas*. Basic Books, Inc., 1980.
- [4] T. S. McNerney, "Tesis_Tangible Programming in Early Childhood_Revisiting Developmental Assumptions through New Technologies," 2000.
- [5] M. Resnick, F. Martin, R. Sargent, and B. Silverman, "Programmable Bricks: Toys to think with," *IBM Syst. J.*, vol. 35, no. 3.4, pp. 443-452, 1996.
- [6] M. Zúñiga, "Aprendizaje mediado por tecnologías digitales: la experiencia de Costa Rica," in *Educación y Nuevas Tecnologías*, Buenos Aires, Argentina: International Institute for Educational Planning. UNESCO, 2003, pp. 99-114.
- [7] S. Einhorn, "MicroWorlds, Computational Thinking, and 21st Century Learning," *LCSI White Paper*, pp. 1-10, 2011.
- [8] S. H. Kim and J. W. Jeon, "Programming LEGO mindstorms NXT with visual programming," in *International Conference on Control, Automation and Systems*, 2007. ICCAS '07., 2007, pp. 2468-2472.
- [9] D. Nam and T. Lee, "The Effect of Robot Programming Education by Pico Cricket on Creative Problem-Solving Skills," in *Proceedings of the 19th International Conference on Computers in Education*, 2011, pp. 1-6.
- [10] M. Resnick, "StarLogo: an environment for decentralized modeling and decentralized thinking," in *Conference companion on Human factors in computing systems common ground - CHI '96*, 1996, pp. 11-12.
- [11] A. Begel and E. Klopfer, "Starlogo TNG: An introduction to game development," *J. E-Learning*, 2007.
- [12] S. Campe, L. Werner, and J. Denner, "Game Programming with Alice: A Series of Graduated Challenges," *Comput. Sci. K-8 Build. a Strong Found.*, no. Special, pp. 13-14, 2012.
- [13] P. Mullins, D. Whitfield, and M. Conlon, "Using Alice 2.0 as a first language," *J. Comput. Sci. Coll.*, vol. 24, no. 3, pp. 136-143, 2009.
- [14] S. Cooper, W. Dann, and R. Pausch, "Alice: a 3D Tool For Introductory Programming Concepts," *J. Comput. Sci. Coll.*, 2010.
- [15] R. Pausch, T. Burnette, A. C. Capeheart, M. Conway, D. Cosgrove, R. DeLine, J. Durbin, R. Gossweiler, S. Koga, and J. White, "Alice: Rapid Prototyping System for Virtual Reality," *IEEE Comput. Graph. Appl.*, vol. 15, no. 3, pp. 8-11, 1995.
- [16] M. Conway, R. Pausch, R. Gossweiler, and T. Burnette, "Alice: a rapid prototyping system for building virtual environments," in *Conference Companion on Human Factors in Computing Systems*, 1994, pp. 295-296.
- [17] C. James, "An 8-Day Plan with Etoys," *Comput. Sci. K-8 Build. a Strong Found.*, no. Special, pp. 23-24, 2012.
- [18] Y.-J. Lee, "Empowering teachers to create educational software: A constructivist approach utilizing Etoys, pair programming and cognitive apprenticeship," *Comput. Educ.*, vol. 56, no. 2, pp. 527-538, Feb. 2011.
- [19] S. Coxon, "Design to Succeed in LEGO WeDo Robotics Challenges," 2010.
- [20] S. Coxon and K. Chandler, "LEGO WeDo," 2010.
- [21] K. Mayerová, "Pilot Activities: LEGO WeDo at Primary School," in *Proceedings of 3rd International Workshop Teaching Robotics, Teaching with Robotics: Integrating Robotics in School Curriculum*, 2012, pp. 32-39.
- [22] J. H. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk, "Programming by Choice Urban Youth Learning Programming with Scratch," *ACM SIGCSE Bull.*, vol. 40, no. 1, p. 367, Feb. 2008.

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pry01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

- [23] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The Scratch Programming Language and Environment," ACM Trans. Comput. Educ., vol. 10, no. 4, pp. 1–15, 2010.
- [24] G. Tomoyose, "miniBlok, el lenguaje de programación argentino para robots que llega a todo el mundo," LA NACION, Argentina, p. 1, 09-May-2014.
- [25] A. Yera-Gil, "Iniciación a la programación visual e interactiva y desarrollo de robótica educativa con Scratch y Enchanting," Univerisdad Pública de Navarra, 2010.
- [26] L. P. Flannery, E. R. Kazakoff, P. Bontá, B. Silverman, M. U. Bers, and M. Resnick, "Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming," in Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13), 2013, pp. 1–10.
- [27] S. Romero, I. Angulo, I. Ruíz, and J. M. Angulo, "Competencias y Habilidades con el Robot 'MOWAY,'" in TAAE: Tecnologías Aplicadas a la Enseñanza de la Electrónica, 2008, p. 8.
- [28] Next is Great, "Move The Turtle: Programming for Kids," Next is Great, 2012. [Online]. Available: <http://movetheturtle.com/>.
- [29] L. Orsini, "Kodable Teaches Kids To Code Before They Learn To Read," readwrite, 2013. [Online]. Available: <http://readwrite.com/2013/04/23/kodable-teaches-kids-to-code-before-they-learn-to-read#awesm=-oDmckub1ryv8TN>.
- [30] edshelf, "Kodable," edshelf, 2013. [Online]. Available: <https://edshelf.com/tool/kodable>.
- [31] G. Huebner and J. Mattingly, "Kodable," Kodable - Site Web, 2012. [Online]. Available: <http://www.kodable.com/>.
- [32] D. Thomas, "KineScript: Visual Programming," Graphite, 2013. [Online]. Available: <http://www.graphite.org/app/kinescript-visual-programming>.
- [33] M. Rollins, Beginning LEGO MINDSTORMS EV3. Apress, 2014.
- [34] LEGO, "LEGO Mindstorms EV3," LEGO, 2013. [Online]. Available: <http://www.lego.com/en-us/mindstorms/?domainredir=mindstorms.lego.com>.
- [35] D. Shapiro, "Robot Turtles: The Board Game for Little Programmers," Robot Turtles - Site Web, 2013. [Online]. Available: <http://www.robotturtles.com/>.
- [36] D. Shapiro, "Robot Turtles: The Board Game for Little Programmers," Kickstarter, Inc, 2013. [Online]. Available: <https://www.kickstarter.com/projects/danshapiro/robot-turtles-the-board-game-for-little-programmer>.
- [37] V. Gupta, S. Gupta, and M. Greaves, "Play-i," Madrona Venture Group and Charles River Ventures, 2013. [Online]. Available: <https://www.play-i.com/>.
- [38] F. Yacob, M. Loglio, D. Di Cuia, V. Leonardi, L. Rabago, and J. Valman, "PRIMO," Solid Labs, 2013. [Online]. Available: <http://primo.io/>.
- [39] L. Liukas, "Hello Ruby," Kickstarter, Inc, 2014. [Online]. Available: <https://www.kickstarter.com/projects/lindaliukas/hello-ruby>.
- [40] L. Liukas, "Hello Ruby," Hello Ruby - Site Web, 2014. [Online]. Available: <http://konichiwaruby.tumblr.com/>.
- [41] T. A. Lentz, "Kids, Robotics, and Gender: a pilot study," Tufts University, 2014.
- [42] M. U. Bers, "Programming in Kindergarten: A Playground Experience," Comput. Sci. K-8 Build. a Strong Found., no. Special, pp. 7–8, 2012.
- [43] M. U. Bers, "The TangibleK Robotics Program: Applied Computational Thinking for Young Children," Early Child. Res. Pract., vol. 12, no. 2, 2010.
- [44] M. U. Bers, "The TangibleK Robotics Program: Applied Computational Thinking for Young Children," Early Child. Parent. Collab., vol. 12, no. 2, 2010.
- [45] M. U. Bers, S. Seddighin, and A. Sullivan, "Ready for Robotics: Bringing Together the Y and E of STEM in Early Childhood Teacher Education," J. Technolgy Teach. Educ., vol. 21, no. 3, pp. 355–377, 2013.
- [46] M. Prensky, "Digital Natives, Digital Immigrants Part 1," Horiz., vol. 9, no. 5, pp. 1–6, 2001.
- [47] M. Prensky, "Digital Game-Based Learning," Comput. Entertain., vol. 1, no. 1, p. 21, Oct. 2003.
- [48] M. Hamada and S. Sato, "Lego NXT as a learning tool," Proc. fifteenth Annu. Conf. Innov. Technol. Comput. Sci. Educ. - ITiCSE '10, p. 321, 2010.

Justificación del proyecto

El conocimiento matemático es una herramienta básica para la comprensión y manejo de la realidad en que se vive. Su aprendizaje, además de durar toda la vida, debe comenzar lo antes posible para que el niño se familiarice con su lenguaje, su manera de razonar y de deducir [1]–[3]. Se sabe que la matemática es la disciplina donde se presenta una actitud social persistente de rechazo y temor, lo que se consigna como una "Matefobia". Esta condición de naturaleza cultural, que trasciende sin duda el sistema educativo, conspira para limitar condiciones socio afectivas apropiadas en los estudiantes, padres y madres de familia e incluso docentes a la hora de realizar una construcción de aprendizajes matemáticos significativos y edificantes. No es sólo una realidad nacional, es una condición que se presenta en muchos países [4]. Muchos reconocidos neurocientíficos en todo el mundo afirman que las habilidades matemáticas están impresas genéticamente en nuestro cerebro. Por lo que, aconsejan enseñar matemática mediante el desarrollo del razonamiento intuitivo, la manipulación de materiales y la realización de actividades lúdicas [5], [6]. Las frases: "la matemática desarrolla el razonamiento lógico" y "aprender a resolver problemas es fundamental para vivir el día a día" son a menudo el vocabulario popular, pero para la mayoría de las instituciones de educación en nuestro país

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

es una realidad distante. En general, debido a la falta de recursos e infraestructura, entre otros, se busca enseñar matemática a partir de la memorización de los procedimientos y la aplicación de ejercicios de repetición. Donde realmente, la práctica matemática implica razonar, imaginar, descubrir, conocer, probar, generalizar, utilizar técnicas, aplicar habilidades, estimar, comprobar resultados, entre otros [4], [7].

Las habilidades asociadas con el desarrollo de la enseñanza de la matemática son esenciales para los niños, ya que puede promover el desarrollo de formas de pensar, actitudes y valores a través de actividades en las que los estudiantes resuelven problemas y situaciones en las que poner en acción todos sus conocimientos [6]–[8]. Por lo que, es necesario que las actividades sean significativas y útiles para los niños, y nunca estar “separadas” de la realidad. Así, el desarrollo del pensamiento lógico-matemático estará ligado a la vida cotidiana de los niños y será un elemento decisivo para la comprensión de la realidad. Por lo tanto, en la educación de la matemática se debe ir evolucionando a través de distintos medios, buscando el planteo de preguntas, otros enfoques imaginativos y permitiendo el desarrollo de ideas. Es necesario que se aplique la matemática a la vida cotidiana, así el aprenderla se hace más dinámico, interesante, comprensible, y lo más importante, útil [2], [9], [10].

En la etapa de la Educación Preescolar, el conocimiento se construye de manera global, y ésta disciplina no es una excepción. Cualquier situación puede aprovecharse para el desarrollo de los conceptos matemáticos. Al introducirse en la práctica de un juego, se adquiere cierta familiarización con sus reglas, relacionando unas piezas con otras, del mismo modo, el novato en matemática compara y hace interactuar los primeros elementos de la teoría unos con otros. El gran beneficio de este acercamiento lúdico consiste, en su potencia para transmitir al estudiante la forma correcta de colocarse en su enfrentamiento con problemas matemáticos. Por su parte, el propósito principal de la programación y la robótica es fomentar la resolución de problemas, el pensamiento lógico-matemático, la abstracción, la creatividad y el aprendizaje a través de exploraciones lúdicas. Por lo que, los estudiantes aprenden a ver la tecnología, la robótica y la programación como herramientas creativas y experimentan las recompensas de comprensión y realización.

A medida que se avanza hacia un mundo donde la tecnología se inserta cada vez más en nuestras vidas, el incorporar la habilidad de programar en el proceso de enseñanza-aprendizaje desde muy temprana edad se vuelve muy importante. Esto porque que ayuda a desarrollar el pensamiento lógico-matemático, la abstracción, y el pensar de una forma más ordenada [11], [12]. La programación se convierte en una plataforma para mostrar la creatividad, especialmente en la resolución de problemas, lo que ayudará a los aprendices, consecuentemente, a desarrollar otras habilidades.

Entre más temprano los niños aprendan a programar, desarrollan competencias para diseñar y resolver problemas que son fundamentales para el éxito académico futuro, y comienzan a usar conceptos matemáticos y destrezas del lenguaje en un contexto significativo.

Mitchel Resnick, del Media Lab del MIT, dice que la programación es la nueva alfabetización, y agrega que al igual que la escritura, la programación ayuda a organizar el pensamiento y a expresar las ideas. En el pasado se pensaba que la programación era muy difícil para la mayoría de las personas, pero gracias al avance de la tecnología es posible realizar entornos de programación que sean accesibles para todas las personas, igual que con la escritura [13].

Por otra parte, existen varios investigadores [14]–[17] que han estudiado la utilización de la robótica en el proceso de enseñanza-aprendizaje de la niñez, logrando que los niños programen y construyan robots para alcanzar contenidos curriculares específicos. Dichos estudios han generado buenos resultados. Temas de áreas como ciencia, tecnología, ingeniería y matemáticas (conceptos STEM) resultan ser complejos. Sin embargo la enseñanza de estas materias durante la primera infancia (periodo que va del nacimiento hasta los ocho años de edad) puede ser interesante y gratificante para los jóvenes estudiantes, especialmente si los temas se abordan a través de la robótica y la programación básica [18]. Además, las habilidades obtenidas con la programación y la robótica son un aspecto clave en el desarrollo de la niñez y su futuro profesional [19].

Asimismo, el aprendizaje en los niños de edad preescolar debe darse de forma concreta y divertida para obtener su atención. Es claro que la robótica es una buena herramienta de aprendizaje para niños pequeños, porque los robots les permiten a los niños tener un aprendizaje partiendo desde lo concreto hacia lo abstracto de forma lúdica.

Por otro lado, es importante incorporar la habilidad de colaborar desde edades tempranas. Los niños, por la etapa de desarrollo de las primeras edades, son egocéntricos, pero cuando empieza a realizar acciones a favor de los demás, su egocentrismo inicial cede el paso a la conducta cooperadora [20]. Esto se puede lograr incorporando actividades de aprendizaje colaborativo en el salón de clases, y con ello se consiguen ventajas en dos importantes áreas: la cognitiva y la socio-afectiva (citadas en la sección anterior). Además, favorece la capacidad de resolver problemas de forma creativa, a partir de estrategias de negociación, la mediación y la búsqueda cooperativa de alternativas.

Referencias

- [1] R. Montalbán, “Infancia y Educación Infantil,” CEAPA: Confederación Española de Padres y Madres de Alumnos, 2005. [Online]. Available: <https://www.ceapa.es/sites/default/files/Documentos/tema9.pdf>. [Accessed: 11-Mar-2015].
- [2] P. González Rodríguez, “Iniciación al número y al cálculo en Educación Infantil,” Universidad de Valladolid, 2013.
- [3] T. L. Arribas and M. Antón, La educación infantil 0-6 años, 6th ed. Editorial Paidotribo, 1990.
- [4] A. Ruiz, “La reforma de la Educación Matemática en Costa Rica - Perspectiva de la praxis,” Cuad. Investig. y Form. en Educ. Matemática Reforma la Educ. Matemática en Costa Rica, vol. 8, no. Especial, pp. 1–111, 2013.
- [5] B. Butterworth, The Mathematical Brain. MacMillan Publishing Company, 1999.
- [6] D. A. Sousa, How the Brain Learns Mathematics, 2nd ed. Corwin Press - Sage Publications Ltd., 2007.

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

- [7] J. A. Fernández-Bravo, "Neurociencias y Enseñanza de la Matemática - Prólogo de algunos retos educativos," Rev. Iberoam. Educ., vol. 51, no. 3, pp. 1–12, 2010.
- [8] P. D. Zoltan and E. G. W., Los primeros pasos en matemáticas. 1: lógica y juegos lógicos. Barcelona: Teide, 1967.
- [9] B. Villabrille, "El Juego en la Enseñanza de las Matemáticas," Buenos Aires, Argentina, 2000.
- [10] P. D. Zoltan and M. Holt, Zoo: juegos matemáticos para educación preescolar. Barcelona: Teide, 1973.
- [11] S. Papert, Mindstorms: children, computers, and powerful ideas. Basic Books, Inc., 1980.
- [12] I. M. Verner and D. J. Ahlgren, "Conceptualizing Educational Approaches in Introductory Robotics," Int. J. Electr. Eng. Educ., vol. 41, no. 3, pp. 183–201, 2004.
- [13] M. Resnick, "Technologies for Lifelong Kindergarten," Educ. Technol. Res. Dev., vol. 46, no. 4, pp. 1–18, 1998.
- [14] A. L. Acuña-Zúñiga, "Proyectos de robótica educativa: motores para la innovación," in Current Development in Technology- Assisted Education, 2006, pp. 951–956.
- [15] A. L. Acuña-Zúñiga, "Robótica: para el desarrollo de habilidades en diseño con niños, niñas y jóvenes en América Latina," in Current Development in Technology- Assisted Education, 2006, p. 5.
- [16] M. Bers, C. Rogers, L. Beals, M. Portsmouth, K. Staszowski, E. Cejka, A. Carberry, B. Gravel, J. Anderson, and M. Barnett, "Innovative session: early childhood robotics for learning," in ICLS '06 Proceedings of the 7th international conference on Learning sciences, 2006, pp. 1036–1042.
- [17] E. R. Kazakoff, A. Sullivan, and M. U. Bers, "The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood," Early Child. Educ. J., vol. 41, no. 4, pp. 245–255, Oct. 2012.
- [18] M. U. Bers, Blocks to Robots: Learning with Technology in the Early Childhood Classroom. Teachers College Press, 2008.
- [19] M. Binkley, O. Erstad, J. Herman, S. Raizen, M. Ripley, and M. Rumble, "Defining 21st century skills," 2010.
- [20] J. L. Zubano Díaz de Cerio, Bases de una Educación para la Paz y la Convivencia. Fondo de Publicaciones del Gobierno de Navarra - Departamento de Presidencia, 1998.

Descriptores

Colaboración
Entorno de programación
Robótica
Niños de la primera infancia
Disciplinas STEM
Pensamiento lógico-matemático
Abstracción
Solución de problemas

Objetivo general

Crear un entorno de programación colaborativo para niños entre 4 y 6 años, sin experiencia previa de programación, que fomente el pensamiento lógico-matemático y la solución de problemas

Objetivo específico: 1 Tipo: Investigación

Diseñar los elementos gráficos (interfaz e interacción) del entorno de programación colaborativo, orientado a niños entre 4 y 6 años

Metas: 1 Tipo: Cuantitativa Cantidad 80

Diseño de los elementos de la interfaz de usuario del entorno de programación colaborativo que satisfaga al menos el 80% de la especificación de los requerimientos

Fecha de impresión: 08/05/2015

Página: 9 de 20

Hora de impresión: 10.19 AM

Información general por proyecto

UNIDAD: 02024400 ==> CENTRO INV EN TECNL. DE INF. Y COMUNICACIÓN

PROYECTO: Pry01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad Subactividad Código inscripción
Investigación Investigación Aplicada

Indicador: 1

Porcentaje de elementos de la interfaz diseñados de acuerdo al documento de especificación de requerimientos

Objetivo específico: 2 Tipo: Investigación

Seleccionar los robots más adecuados para el entorno de programación y realizar su configuración e implementación

Metas: 1 Tipo: Cuantitativa Cantidad 1

Elección del robot más adecuado para el entorno de programación

Indicador: 1

Robot seleccionado para el entorno de programación

Metas: 2 Tipo: Cuantitativa Cantidad 80

Configuración e implementación del robot seleccionado en al menos 80% de los comandos definidos para ejecutar

Indicador: 1

Porcentaje de los comandos implementados para ejecutar con el robot

Objetivo específico: 3 Tipo: Investigación

Diseñar un protocolo de comunicación entre los robots y los dispositivos convencionales o móviles

Metas: 1 Tipo: Cuantitativa Cantidad 80

Utilización del protocolo de comunicación en un 80% en la implementación del entorno de programación

Indicador: 1

Porcentaje de la definición del protocolo de comunicación que se implementará en el entorno de programación

Objetivo específico: 4 Tipo: Investigación

Implementar el entorno de programación colaborativo basado en los diseños propuestos: elementos gráficos (interfaz e interacción) y el protocolo de comunicación, utilizando el robot seleccionado

Metas: 1 Tipo: Cuantitativa Cantidad 80

Prototipo del entorno de programación que implemente al menos 80% de los elementos de la interfaz

Indicador: 1

Porcentaje de elementos de la interfaz implementados de acuerdo al diseño realizado

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Prr01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	
<u>Objetivo específico:</u>	5	Tipo: Investigación
Evaluar la funcionalidad y la usabilidad del entorno de programación		

Metas: 1 Tipo: Cuantitativa Cantidad 3

Evaluación del entorno de programación en relación a su usabilidad y su funcionalidad en al menos 3 tareas específicas

Indicador: 1

Cantidad de tareas evaluadas de acuerdo a la implementación realizada

Objetivo específico: 6 Tipo: Investigación

Divulgar los resultados de la investigación

Metas: 1 Tipo: Cuantitativa Cantidad 2

Al menos 2 artículos publicados en una conferencia, simposio, o revista con los resultados de la investigación

Indicador: 1

Cantidad de artículos publicados

Metas: 2 Tipo: Cuantitativa Cantidad 1

Al menos 1 presentación pública sobre la investigación

Indicador: 1

Cantidad de presentaciones realizadas

Ubicación geográfica del Proyecto

COSTA RICA

Provincia	Cantón	Distrito	Región
TODAS	TODOS	TODOS	TODAS

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

Metodología:

En primer lugar es importante entender y especificar el contexto de uso del entorno de programación, donde se debe identificar las características y limitaciones de los usuarios finales, el propósito de crear el entorno, y bajo qué condiciones se debe crear. Esto se logrará mediante un estudio de literatura y entrevistas a docentes de preescolar y expertos de la FOD.

Se utilizará y se adaptará la especificación de requerimientos y la definición de tareas específicas que se obtuvo como producto del Proyecto N° 834-B3-260, donde se realizó entrevistas con docentes de preescolar y expertos de la FOD y visitas de campo a grupos de preescolar y a la misma FOD para conseguir tal fin.

El proyecto se realizará para dispositivos convencionales o móviles, por lo que se utilizarán y se adaptarán los resultados del estudio comparativo sobre los sistemas operativos móviles y dispositivos móviles existentes del mercado del Proyecto N° 834-B3-260, para utilizar el sistema operativo móvil y el dispositivo móvil más adecuado a utilizar para desarrollar el entorno de programación. Aunque

Paralelamente, se estudiarán diversos diseños de interfaces orientadas a niños entre 4 y 6 años, mediante la búsqueda y el análisis de sistemas existentes orientados a la niñez de estas edades. También se tomará en cuenta la literatura, las visitas de campo y los criterios de los expertos entrevistados. A partir de todo esto se comenzará a extraer características de la interfaz de un entorno de programación adecuado para niños.

Posteriormente, se comenzará a diseñar los elementos del entorno de programación basados en las heurísticas de usabilidad, las características del dispositivo seleccionado, la especificación de requerimientos, las tareas especificadas, los aspectos colaborativos y las características extraídas del estudio de las interfaces de las aplicaciones orientadas a niños entre 4 y 6 años en dispositivos convencionales o móviles.

En esta etapa se diseñarán soluciones acordes al entorno. Como resultado se deberá especificar la arquitectura de información, el diseño conceptual, el diseño de interacción y el diseño gráfico. Estos diseños se deberán validar con los expertos de la FOD, con los docentes de preescolar y por supuesto con niños entre 4 y 6 años, hasta obtener una propuesta final de diseño.

Luego, se diseñará el protocolo de comunicación evaluando diferentes medios de comunicación (como WiFi, Bluetooth u otros disponibles) para comunicar los dispositivos y el robot en una red de área local con acceso limitado a Internet, con el fin de iniciar la implementación del entorno.

Como parte de este proceso se estudiarán diferentes opciones de robots para seleccionar el más adecuado de acuerdo a la especificación de requerimientos, las tareas especificadas y los elementos colaborativos que se diseñen. Una vez con el robot seleccionado, se procederá a realizar la configuración e implementación correspondiente.

Al mismo tiempo, se plantearán estrategias didácticas colaborativas para niños de edad preescolar. Estas estrategias se podrán en práctica en los últimos experimentos que se lleven a cabo con niños entre 4 y 6 años, contando con la ayuda de los expertos de la FOD y los docentes de preescolar para la realización, la revisión y la validación de dichas estrategias didácticas.

En este punto se procederá a redactar un primer artículo para publicar los resultados obtenidos hasta el momento, y generar el informe parcial del proyecto. Debido a que la investigación será de carácter académico, se procurará aprovechar al máximo las posibilidades que ofrecen las herramientas gratuitas (Open Source) disponibles a través de Internet.

Se utilizarán los resultados de las validaciones de las soluciones de diseño realizadas con expertos y los niños, las heurísticas, las características y los aspectos que se deben tomar en cuenta en la implementación de una aplicación móvil obtenidos del Proyecto N° 834-B3-260. Con ellos y el diseño del protocolo de comunicación se llevará a cabo la implementación del prototipo del entorno de programación.

En esta etapa se estarán utilizados los siguientes métodos:

- Prototipado ágil, donde se realizará con las personas implicadas (expertos de la FOD) un conceso del prototipo, tomando en cuenta las recomendaciones de los consultores (docentes de preescolar).
- Pilotos del entorno de programación con los niños entre 4 y 6 años, donde se estará evaluando la usabilidad y la funcionalidad del mismo, capturando las métricas respectivas.

Con los resultados obtenidos de los métodos anteriores, se procederá a la implementación de una versión final del entorno, con el consenso de las personas implicadas, las recomendaciones de los consultores y los cambios resultantes de las evaluaciones de los pilotos. A continuación se describe el proceso de evaluación del entorno.

La fase de evaluación es la más importante del proceso del DCU, ya que es la fase en la que se validan las soluciones de diseño (el sistema satisface los requisitos), o por el contrario, se detectan problemas de usabilidad.

En las etapas de diseño e implementación se crearán documentos que describan la funcionalidad y los requisitos especificados, y se diseñará un plan de pruebas que cubra todos los casos de uso de la aplicación, la simulación de entornos de pruebas, la ejecución de los casos de prueba y la gestión de los defectos encontrados. Dichos documentos serán utilizados en esta etapa de evaluación por los distintos actores involucrados. Además, se utilizarán los resultados de las evaluaciones realizadas en el Proyecto N° 834-B3-260.

La evaluación de usabilidad se llevará a cabo siguiendo la metodología de "test de usabilidad" utilizando el método de pruebas con usuarios, siguiendo los pasos definidos por Nielsen (Nielsen, 1994).

La evaluación de funcionalidad está contemplada en la evaluación de usabilidad, ya que en estas pruebas el usuario deberá ejecutar las tareas especificadas en los requisitos del entorno (casos de uso), garantizando que el entorno hace lo que debe (casos positivos), que no falla (casos negativos) y que cumple con lo que se ha solicitado (satisface los requisitos). Cada usuario realizará las acciones indicadas en cada caso de

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad Subactividad Código inscripción

Investigación Investigación Aplicada

prueba, comprobando que se cumpla el resultado esperado. Si el resultado es distinto al esperado, se reportará un defecto con los detalles: descripción, datos utilizados, capturas de pantalla, etc., para facilitar la solución del defecto.

Aunque las pruebas con usuarios son pruebas de evaluación, no se deben llevar a cabo al finalizar el proceso de diseño e implementación de la aplicación, ya que el DCU es una filosofía de diseño iterativa, basada en la mejora incremental del producto. Por tanto, cuanto más se espere para realizar la primera de las pruebas, más costoso resultará la corrección de los errores de diseño que se detecten.

Algo importante, es que cada prueba debe contextualizarse bajo un objetivo o motivación. Además, cada prueba debe ser razonable, específica, factible y debe tener una duración razonable (al trabajar con niños de estas edades se debe trabajar en la planificación de diversas actividades en periodos de tiempo de 45 minutos como máximo).

Los expertos de FOD, los docentes de preescolar y los niños utilizarán y evaluarán los diferentes prototipos del entorno, para probar y validar lo que se ha especificado con ellos, buscando problemas de usabilidad y funcionalidad que deban ser corregidos. Además, varias de estas pruebas piloto serán evaluadas con los niños.

En relación con la etapa de implementación, al mismo tiempo que se llevan a cabo las validaciones de diseño, se realizará el prototipado ágil. Luego, con una solución de diseño final y un prototipo implementado se realizarán varios pilotos con los niños de preescolar para probar la usabilidad y la funcionalidad del entorno de programación. Estos pilotos serán realizados utilizando robots y el entorno implementado en el dispositivo móvil.

Para cada uno de estos pilotos se confeccionarán guías didácticas de trabajo, guías de observación y guías de evaluación respectivas, que utilizarán como apoyo los miembros del equipo de trabajo. Además, cada piloto deberá ser grabado y deberá contar con un grupo de facilitadores y observadores que tomarán notas de todo el proceso.

El número de participantes necesarios para detectar los problemas más importantes de usabilidad de un diseño es, según Nielsen, alrededor 15 (Nielsen, 1994). El objetivo de estas pruebas es mejorar de forma iterativa la usabilidad de la aplicación, por lo que cada prueba ofrecerá suficiente información para mejorar la solución de diseño, aun cuando no se detecte el 100% de los problemas de usabilidad.

En general, los primeros pilotos que se llevarán a cabo serán pruebas informales del entorno, para detectar posibles errores de usabilidad y funcionalidad, evaluar que el entorno apoye el aprendizaje de la programación de forma colaborativa del niño (que no interfiera negativamente en el proceso), y comprobar las hipótesis planteadas en la investigación. Cada prueba será realizada con al menos 12 niños y en sesiones de aproximadamente 1,5 horas.

Para el último piloto se llevará a cabo una prueba más rigurosa utilizando un mayor número de usuarios (aproximadamente 21 niños), y en sesiones de aproximadamente 1 hora cada una, en un taller de programación aproximadamente de 10 sesiones. El objetivo de este último piloto es comprobar el objetivo principal.

Las métricas que se utilizarán para llevar a cabo la evaluación de cada piloto son las que establece Nielsen (Nielsen, 1994) para el método "test de usabilidad". Estas métricas se medirán de la siguiente manera:

- **Facilidad de aprendizaje.** Cuando un niño utilice por primera vez el entorno, se observará si la interacción con éste es sencilla, intuitiva y fácil de realizar, es decir, si es fácil aprender a utilizarla.
- **Eficiencia.** Cuando los niños hayan aprendido el funcionamiento básico del diseño, se contabilizará: (1) el número de errores cometidos por los niños al realizar las tareas durante la prueba y si dichos errores fueron recuperables, es decir, si el niño, puede continuar la prueba y realizar el procedimiento adecuado; y (2) el tiempo empleado por los niños en la realización de las tareas.
- **Cualidad de ser recordado.** Cuando los niños hayan utilizado varias veces el entorno, se recurrirá al recuerdo (la memoria en el uso de dicha aplicación), es decir, qué tanto recuerda el niño después de un período sin utilizar dicha aplicación. Se realizará por medio de entrevistas a los niños, preguntándoles aspectos relacionados al entorno y pidiéndoles que dibujen (describiendo) lo que recuerden del mismo.
- **Efectividad.** Se medirá en función de los éxitos y fracasos en la ejecución de las tareas asignadas durante la prueba: ¿cuántas tareas logró realizar cada niño?
- **Satisfacción.** Medidas subjetivas, provenientes de comentarios, opiniones, o bien a través encuestas a los niños, obteniendo respuestas emocionales de los niños de tal manera que permita conocer cómo se siente el niño al terminar la tarea y/o la prueba (si le gusta el entorno, si le ha resultado fácil utilizar el entorno, si le fue fácil realizar las tareas, entre otros). Se realizará escuchando lo que comentan en voz alta mientras realizan la prueba, mediante encuestas post-prueba, las observaciones del personal que realiza la prueba (los facilitadores y/o los observadores presentes) y la grabación de dicha prueba.

Cada uno de los participantes realizará la prueba, y durante cada prueba se registrará toda información relevante (encuestas, grabaciones y notas tomadas por los facilitadores y los observadores) para el posterior análisis del comportamiento de los niños.

Una vez que los participantes finalicen cada prueba y se haya registrado toda la información pertinente se procederá a analizar los resultados y sintetizarlos en un informe final, concluyendo qué mejoras necesita el diseño y la aplicación en base a estos resultados.

Con los resultados obtenidos se procederá a la implementación de un piloto final, cuya evaluación se realizará en un contexto real.

Por otra parte, este trabajo de investigación cuenta con la colaboración de los expertos:

- Unidad de Aprendizaje Lógico, Científico y Robótica de la FOD, quienes coordinan y llevan a cabo, junto al MEP, el Programa Nacional de Informática Educativa MEP-FOD (PRONIE). Ellos serán los encargados de la especificación de requerimientos y tareas específicas del entorno, y los responsables en la realización de las evaluaciones con los niños, entre otras actividades.
- Laboratório de Experimentação Remota (REXLab) de la Universidade Federal de Santa Catarina (UFSC), Campus Araranguá, en Brasil, donde se

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

estará trabajando con el Dr. Juarez Bento da Silva y su equipo de investigación. Ellos aportarán su conocimiento en la realización de entornos virtuales y remotos en aplicaciones para la enseñanza y aprendizaje de diferentes disciplinas de la Educación Pública. El fin de esta colaboración es realizar un robot que se pueda acceder de forma remota, y así utilizarse en aquellas Escuelas que no cuenten con los recursos económicos para obtener robots. También se cuenta con el apoyo del Programa de Tecnologías Educativas Avanzadas (PROTEA) de la Universidad de Costa Rica. Los expertos de PROTEA brindarán el contacto con docentes y estudiantes de la carrera de Educación Preescolar de la Escuela de Formación Docente de la Universidad de Costa Rica. Al finalizar el piloto público, se procederá a redactar el segundo artículo para publicar los resultados obtenidos del proyecto, y generar el informe final del mismo.

Recursos con que cuenta el proyecto:

- Tiempo de la profesora e investigadora Kryscia Ramírez (aportado por la ECCI).
- Tiempo del profesor e investigador Luis Guerrero (aportado por la ECCI).
- Infraestructura tecnológica (de hardware y software) de la ECCI; 10 tabletas de 7", 1 tableta de 10", 7 celulares.
- Infraestructura tecnológica de la ECCI, que incluye la computadora de la profesora Kryscia Ramírez.
- Espacio físico aportado por la ECCI (oficinas) de los profesores Kryscia Ramírez y Luis Guerrero.
- Material de oficina aportado por el CITIC como por la ECCI.

Evaluación del proyecto por parte de los participantes (mecanismos o herramientas que se van a tomar en cuenta para evaluar el proyecto)

Evaluación del impacto: mecanismos o herramientas que se van a tomar en cuenta para evaluar el proyecto

Se medirá por medio de la satisfacción de los niños y expertos de la FOD del entorno de programación implementado y la intención que muestren de utilizarlo en el futuro para el proceso de enseñanza-aprendizaje de la programación (esto se evaluará mediante encuestas y/o entrevistas)

Evaluación de la propuesta: mecanismos o herramientas que se van a tomar en cuenta para evaluar el proyecto

Disciplina:

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pry01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

Cargas totales por actividad sustantiva

Actividad	Subactividad	Total de horas propias	Total horas adicionales
Investigación	Investigación Aplicada	null	null

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pry01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad
Investigación

Subactividad
Investigación Aplicada

Código inscripción

Período: null

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pry01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad
Investigación	Investigación Aplicada

Código inscripción

Período: null

Partida:

Monto:

Descripción del Financiamiento del Monto Solicitado

Justificación:

UNIDAD: 02024400 ==> CENTRO INV EN TECNL. DE INF. Y COMUNICACIÓN

PROYECTO: Pry01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad	Subactividad	Código inscripción
Investigación	Investigación Aplicada	

Tipo de horas: Asistente

<u>Cantidad de Horas:</u>	<u>Meses:</u>	<u>Solicitado a:</u>
10	12	Investigación

Justificación:

El asistente se necesita para ayudar en las diferentes etapas de la investigación, ya que realizará labores de apoyo a los investigadores.

UNIDAD: 02024400 ==> CENTRO INV EN TECNL. DE INF. Y COMUNICACIÓN

PROYECTO: Pxy01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Estado: Editable

Actividad Subactividad Código inscripción
Investigación Investigación Aplicada

ACTIVIDADES	FECHA INICIAL	FECHA FINAL
Recopilar y estudiar la literatura	01/07/2015	30/08/2015
Utilizar y adaptar la especificación de las tareas específicas y los requerimientos del entorno de programación obtenida del Proyecto N° 834-B3-260	01/07/2015	30/09/2015
Utilizar y adaptar la tabla comparativa de ventajas y desventajas de los dispositivos móviles de acuerdo a sus características del Proyecto N° 834-B3-260 para elegir el sistema operativo móvil y el dispositivo móvil más adecuado a utilizar	01/07/2015	30/09/2015
Especificar las heurísticas de usabilidad para aplicaciones	01/10/2015	31/10/2015
Estudiar diseños de interfaces y aplicaciones orientadas a niños entre 4 y 6 años	01/10/2015	31/10/2015
Diseñar los elementos del entorno de programación	01/11/2015	31/12/2015
Diseñar los elementos que fomenten la colaboración en el entorno de programación	01/11/2015	31/12/2015
Diseñar el protocolo de comunicación para la conectividad entre los dispositivos (convencionales o móviles) y el robot	01/01/2016	31/03/2016
Evaluar diferentes medios de comunicación (como WiFi, Bluetooth u otros disponibles) para comunicar los dispositivos y el robot en una red de área local con acceso limitado a Internet	01/01/2016	31/03/2016
Definir estrategias didácticas colaborativas para el aprendizaje de la programación en niños de edad preescolar	01/03/2016	31/05/2016
Estudiar diferentes opciones de robots para el entorno de programación	01/03/2016	31/05/2016
Realizar la configuración y la implementación del robot de acuerdo a la elección realizada	01/03/2016	31/05/2016
Seleccionar el robot para el entorno de programación	01/03/2016	31/05/2016
Elaborar el primer artículo que resuma el trabajo realizado hasta el momento	01/05/2016	30/06/2016
Elaborar el informe parcial del proyecto de investigación	01/06/2016	30/06/2016
Implementar el entorno de programación mediante prototipado ágil, generando diferentes prototipos	01/06/2016	31/12/2016
Evaluar la usabilidad y la funcionalidad del entorno de programación planificando y realizando diferentes pilotos con los expertos de la FOD, los docentes de preescolar y los niños entre 4 y 6 años, siguiendo la metodología de test de usabilidad	01/08/2016	28/02/2017
Realizar ajustes al prototipo, con base en los resultados de cada evaluación	01/08/2016	28/02/2017
Realizar un piloto público con la versión final del entorno de programación en un contexto real	01/03/2017	31/05/2017

Fecha de impresión: 08/05/2015

Página: 19 de 20

Hora de impresión: 10.19 AM

Información general por proyecto

UNIDAD: 02024400 ==> CENTRO INV EN TECNOL. DE INF. Y COMUNICACIÓN

PROYECTO: Pry01 - 537 - 2016 Creación de un entorno de programación colaborativo orientado a niños entre 4 y 6 años que fomente el pensamiento lógico-matemático y la solución de problemas

Actividad	Subactividad	Estado:	Código inscripción
Investigación	Investigación Aplicada	Editable	
Elaborar el informe final del proyecto de investigación			30/06/2017
Elaborar el segundo artículo que resume el trabajo realizado			30/06/2017