



**Creación de un entorno de programación en dispositivos móviles orientado a niños entre 4 y 6 años que permita programar aplicaciones para robots**

**16/10/2013**



**Especificación de**

**Requerimientos**

|  |  |
| --- | --- |
| Nombre del Proyecto: | *Creación de un entorno de programación en dispositivos móviles orientado a niños entre 4 y 6 años que permita programar aplicaciones para robots* |
| Cliente: | *Fundación Omar Dengo* |
| Fecha: | *09/10/2013* |
| Preparado por: | *Kryscia Ramírez Benavides* |

1. Introducción

La era digital también ha provocado que los dispositivos móviles en la actualidad se encuentran en todas partes; ya que son pequeños, accesibles, potentes y personalizables; pueden comunicarse por diversas infraestructuras de red y proporcionar aplicaciones integradas utilizadas en la vida cotidiana.

Todo esto ha llevado a diferentes investigadores a estudiar la forma en que estos dispositivos pueden ser integrados en los sistemas de educación y sus efectos sobre los procesos de aprendizaje. Ya que la actitud que los nativos digitales tienen hacia estas tecnologías móviles es: interesado, competitivo, cooperativo, orientado a los resultados, búsqueda activa de información y soluciones (Prensky 2003). Y es la actitud que se desearía en los salones de clases.

Lo que atrae a los nativos digitales a estas tecnologías, en opinión de Prensky, es el aprendizaje que ofrecen los juegos de vídeo y la computadora. Ya que a los niños les encanta aprender constantemente, sobre todo cuando se les impone retos, y las herramientas de la era digital ofrecen oportunidades de aprendizaje a cada segundo (Prensky 2003).

Prensky propone cambiar el proceso de enseñanza-aprendizaje; reconsiderando la metodología y el contenido. En primer lugar, la metodología debe cambiar en la manera de interactuar con los estudiantes y de transmitir los conocimientos: ir más rápido, menos paso a paso, más en paralelo, con acceso aleatorio, entre otras cosas. En segundo lugar, el contenido que se debe enseñar, existiendo dos tipos: contenido antiguo (de siempre: matemática, español, ciencias, etc.) y el contenido futuro (lógico, digital y tecnológico, como robótica). Además, se debe pensar en cómo enseñar ambos tipos de contenido a los nativos digitales; donde el primero implica un cambio de la metodología y, el segundo consiste en agregar nuevos contenidos (Prensky 2001).

Además, a medida que se avanza hacia un mundo donde la tecnología controla cada vez más nuestras vidas, el incorporar la habilidad de programar en el proceso de enseñanza-aprendizaje desde muy temprana edad se vuelve muy importante. Ya que ayuda a expandir la mente y a pensar de una forma más ordenada. La programación es una plataforma para mostrar la creatividad, especialmente en la solución de problemas.

Papert describe la importancia de la programación como una herramienta para reflexionar sobre el propio pensamiento, ya que desarrolla un amplio conjunto de capacidades interconectadas como articulación del problema, trabajo en equipo, persistencia, y otras habilidades esenciales que se necesitan en la vida (Papert 1980). Además, explica cómo las computadoras pueden ayudar a los niños a aprender activamente y crear conocimiento.

Además, Papert introduce el concepto de construccionismo, el cual lo define como una extensión del concepto de constructivismo dado por Jean Piaget (teoría del aprendizaje atribuido principalmente a la investigación psicológica del desarrollo). El Construccionismo se centra en cómo las interacciones con las tecnologías pueden promover el desarrollo social y cognoscitivo, proponiendo que la tecnología se combine con el constructivismo (Papert 1980). Esto permitirá a los estudiantes oportunidades para diseñar, construir y programar proyectos de significado personal, lo que añade un componente afectivo.

Por otra parte, Bers afirma que el uso de la robótica puede ser una poderosa herramienta práctica en niños pequeños, para aprender sobre ciencia, tecnología, ingeniería y matemáticas (STEM: *Science, Technology, Engineering, and Math*). Bers explica un paralelismo entre el uso de la robótica durante los primeros años de escolaridad y el desarrollo de las habilidades que los estudiantes necesitan en el Siglo XXI. Estas habilidades serán críticas para el éxito en el futuro de los nativos digitales: pensamiento creativo, claridad en la comunicación, análisis sistemático, colaboración efectiva, diseño reiterativo y aprendizaje continuo (Bers 2008).

El uso de la robótica permite a los niños explorar conceptos complejos de una manera concreta y divertida, involucra habilidades sensomotoras y socioemocionales fundamentales para el sano desarrollo de los niños pequeños, proporciona el ambiente de aprendizaje libre de ideales y el uso principal de la curiosidad natural de los niños para aprender a construir y programar (Bers 2012; Bers 2008).

Por otra parte, diversos autores coinciden en que las habilidades básicas que se desarrollan con la participación en cursos o talleres de robótica tales como el diseño, la construcción, y optimización de modelos robóticos permiten adquirir habilidades fundamentales de ingeniería, que a su vez inciden en el logro de habilidades de otras disciplinas como lenguajes matemáticos, análisis de datos y trabajo en equipo (Verner and Ahlgren 2004).

Asimismo, en diversas latitudes, particularmente en países desarrollados, se ha utilizado la robótica y la programación para el logro de estándares de aprendizaje en ciencias, ingeniería, tecnología y matemáticas (STEM), áreas identificadas cómo claves para lograr el desarrollo tecnológico y educativo de los países (Caron 2010; Colelli 2009; Ivey and Quam 2009). En este sentido, también se ha evidenciado que muchos estudiantes que terminan programas en robótica continúan con una carrera en ingeniería (Caron 2010).

Algunas de las habilidades que promueve la participación en programas de robótica refieren a: pensamiento crítico y resolución de problemas; colaboración y liderazgo; agilidad y adaptabilidad; iniciativa y emprendedurismo; acceso y análisis de información; curiosidad e imaginación (Ivey and Quam 2009).

En resumen, la robótica combina las posibilidades físicas y creativas de construir y manipular objetos con las experiencias de resolución de problemas y la colaboración, para aprender a programar los objetos y moverlos como se deseé. Todo lo anterior redunda en la posibilidad de que los jóvenes puedan desarrollar las competencias definidas como claves para alcanzar el éxito en la vida profesional y personal, según el modelo de las habilidades para el Siglo XXI, esas habilidades refieren a pensamiento superior o sofisticado, resolución flexible de problemas, habilidades de comunicación y colaboración, todo ello a través del uso y apropiación de las tecnologías (Binkley et al. 2010).

En Costa Rica se ha observado que niños, niñas y adolescentes que participan en programas de robótica adquieren habilidades referidas a la resolución de problemas, construcción de sistemas mecánicos y la construcción y control de prototipos robóticos de invención propia. Asimismo, desarrollan habilidades en áreas sociales como el trabajo en equipo, la comunicación de procesos de pensamiento y la divulgación de resultados todas habilidades que buscan superar la brecha digital y generar intereses e inquietudes que en el mediano y largo plazo puedan materializarse en el desarrollo profesional de nuevas generaciones (Castro-Rojas and Acuña-Zúñiga 2012).

Es por todo esto que en esta investigación se propone la creación de un entorno de programación para LEGO Mindstorms NXT en dispositivos móviles, con el fin de permitir programar a niños entre 4 y 6 años. Este entorno de programación será simple, constituido por símbolos iconográficos y sonidos, debido al público meta al que está dirigido; y esto es uno de los principales retos de este proyecto de investigación. Además, el entorno de programación será utilizado por el proyecto implementado por el Instituto INNOV@ de la Fundación Omar Dengo (FOD) y el Ministerios de Educación Pública: Robótica Educativa y Aprendizaje por Diseño de Programas.

* 1. Alcance

Crear un entorno de programación en dispositivos móviles para LEGO Mindstorm NXT v2.0, que permita a niños de 4 a 6 años programar robots construidos por estos kits. La creación contempla el diseño, la implementación y la evaluación (en relación a la usabilidad y la funcionalidad) del entorno. Además, se definirá un protocolo para establecer comunicación y enviar comandos entre el dispositivo móvil y el robot.

El objetivo de la creación de este entorno es enseñar a los niños de edad preescolar a programar, antes de que incluso hayan aprendido a leer, y con ello lograr el desarrollo del pensamiento lógico-matemático. Así los niños, desde muy temprana edad, puedan desarrollar habilidades para poder resolver problemas simples de forma organizada, a veces utilizando el método de ensayo y error.

El entorno de programación propuesto debe tener una interfaz de usuario muy simple, constituido por símbolos iconográficos y sonidos, debido al público meta al que está dirigido. Este es el principal reto en este proyecto de investigación.

El proyecto plantea que los niños puedan resolver varios problemas previamente definidos, y que utilizando el entorno, programen las soluciones. Cada problema tiene un respectivo robot previamente construido.

Esta investigación cuenta con el apoyo de la Fundación Omar Dengo (FOD), que ha desarrollado el programa de Informática Educativa durante los últimos 26 años en escuelas públicas de Costa Rica. Los expertos de la FOD son los encargados de dar los requerimientos del entorno de programación y de evaluarlo.

* 1. Definiciones, siglas y abreviaturas.

**Android OS** = *Android Operating System*, en español Sistema Operativo Android.

**Archivo APK =** Un archivo con extensión .apk (*Application PacKage File*) es un paquete para el sistema operativo Android. Este formato es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android para *smartphones* y *tablets*.

**Archivo NXJ =** Un archivo con extensión .nxj es un archivo binario que une todos los archivos de clases Java que se quieran correr en el NXT y luego este debe ser subido al bloque NXT.

**Bluetooth** = Especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.

**FOD** = Fundación Omar Dengo

**LEGO MINDSTORMS NXT** = Kit programable de robótica de LEGO, lanzado a finales de julio de 2006. El kit tiene dos versiones: la versión comercial (set # 8527) y la versión educacional (set # 9797). Viene con el software de programación NXT-G, o LabVIEW opcionalmente para LEGO MINDSTORMS. Existen una variedad de lenguajes no oficiales, como NXC, NBC, leJOS NXJ y RobotC.

**leJOS NXJ** = Entorno de programación Java para el LEGO MINDSTORMS NXT, que permite programar robots LEGO en Java. leJOS es una pequeña máquina virtual de Java para ladrillo LEGO NXT, portada en el 2006. leJOS NXJ es un reemplazo del firmware original que trae el ladrillo inteligente del NXT, el nuevo firmware leJOS NXJ debe ser *flasheado* en el NXT, y reemplazará el firmware del LEGO MINDSTORMS.

**Protocolo NFC** **(*Near field communication*)** = Tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.

***Text To Speech*** = Es un tipo de aplicación de síntesis de voz que se utiliza para crear una versión de sonido hablado del texto en un documento de la computadora.

**TTS** = *Text To Speech*.

* 1. Referencias

Bers MU (2008) Blocks to Robots: Learning with Technology in the Early Childhood Classroom. 154.

Bers MU (2012) Designing Digital Experiences for Positive Youth Development, 1st ed. 216.

Binkley M, Erstad O, Herman J, Raizen S, Ripley M, Rumble M (2010) Defining 21st century skills. 71.

Caron D (2010) Competitive Robotics Brings Out the Best in Students. Tech Dir 69:21–23.

Castro-Rojas MD, Acuña-Zúñiga AL (2012) Propuesta comunitaria con robótica educativa: valoración y resultados de aprendizaje. Rev Teoría la Educ Educ y Cult en la Soc la Inf 13:91–118.

Colelli R (2009) Model Program: Southern Lehigh High School, Center Valley, PA. Technol Teach 68:27–32.

Ivey D, Quam G (2009) 4-H and Tech Ed Partnership Gets Students Geeked about STEM. Techdirections 69:19–21.

Papert S (1980) Mindstorms: children, computers, and powerful ideas. 244.

Prensky M (2003) Digital Game-Based Learning. Comput Entertain 1:21. doi: 10.1145/950566.950596

Prensky M (2001) Digital Natives, Digital Immigrants Part 1. Horiz 9:1–6.

Verner IM, Ahlgren DJ (2004) Conceptualizing Educational Approaches in Introductory Robotics. Int J Electr Eng Educ 41:183–201.

1. Descripción general

El entorno debe tener una interfaz con una metáfora ambigua (sin género) y simple, que logre que el concepto de “programar” quede claro a los niños. Es importante enfatizar que el núcleo de la programación son los algoritmos, los cuales son una serie de instrucciones (pasos) para que se lleve a cabo una tarea. Por tanto, se quiere lograr que los niños creen un conjunto de instrucciones ordenadas para resolver un problema específico, y puedan ver al robot ejecutando las instrucciones dadas.

La metáfora que se definió para el entorno es de un libro para la creación de cuentos, incorporando el uso de la flora y la fauna costarricense, así como su patrimonio cultural. Por lo que, se utilizará como personaje (tanto físico como digital) el mono titi (mono ardilla) y el ambiente será el bosque tropical de Costa Rica, donde se deberán manipular las esferas de piedad.

Con este sistema se pretende independizarse de las computadoras de escritorio, y realizar el entorno en dispositivos móviles, para dar mayor comodidad y flexibilidad a los niños. El diseño e implementación del entorno de programación es un proceso incremental, y se han establecido diferentes fases para la implementación de acuerdo a los requerimientos y problemas planteados por la FOD.

* 1. Perspectiva del producto

Este entorno de programación tendrá una interfaz de usuario muy simple, constituida por símbolos iconográficos y sonidos, debido al público meta al que está dirigido. La interfaz será intuitiva y fácil de usar para que los niños puedan crear una secuencia ordenada de acciones (programar) para resolver un problema dado. Además, se definirá un protocolo para establecer comunicación y enviar comandos entre el dispositivo móvil y el robot.

* + 1. Interfaces de usuario

La interfaz de usuario estará constituida por símbolos iconográficos y sonidos para interactuar con las niñas y los niños. En relación a este módulo deberá contar con las siguientes etapas:

* Pantalla donde se seleccione el actor que interpretará al personaje: tipo de robot, mediante símbolos iconográficos y sonidos (se utilizará la característica *text to speech*).
* Pantalla donde se seleccione el personaje: robot personalizado, mediante símbolos iconográficos y sonidos (se utilizará la característica *text to speech*).
* Pantalla donde se cree el cuento: creación del programa (selección de una secuencia de comandos), mediante símbolos iconográficos y sonidos (se utilizará la característica *text to speech*). En esta misma pantalla habrá un botón para enviar los comandos al robot para que actúe el cuento: ejecución del programa creado, mediante símbolos iconográficos y sonidos (se utilizará la característica *text to speech*).
	+ 1. Interfaces con hardware

Los dispositivos móviles que se utilizarán deben tener como mínimo las siguientes características:

* Tableta de 7” u 8”
* Android OS, v4.1.2 (Jelly Bean) o superior
* *Text To Speech* (TTS)
* Conexión *bluetooth*
* *Multitouch*

Los robots de LEGO Mindstorm NXT usados serán de la versión 2.0 o superior, y deberán tener en los ladrillos inteligentes (*bricks*) el firmware de leJOS NXJ versión 0.9.1 o superior.

* + 1. Interfaces con software

Los dispositivos móviles deberán tener el sistema operativo Android Jelly Bean (v4.1.2) o superior.

Se utilizarán aplicaciones (algunas propias) del dispositivo móvil, como TTS, entre otros.

* + 1. Interfaces de comunicación

La manera que se comunicará el entorno de programación con los robots será uno a uno por medio de la conexión *bluetooth*.

* + 1. Restricciones de memoria

El entorno de programación necesitará aproximadamente 6MB, más los tamaños de los archivos con los programas que se guardan localmente.

El programa que interpreta el protocolo de comunicación utilizado entre el dispositivo y el robot necesita aproximadamente 32KB, este programa estará en cada uno de los robots.

* + 1. Requerimientos de adecuación al entorno

[En esta sección se deben especificar los requerimientos de datos o secuencias de inicialización que son específicas a un sitio dado, misión, o modo operacional (por ejemplo, valores posibles, límites de seguridad, etc.); se debe especificar el sitio o las características relacionadas a la misión que deben modificarse para adaptar el software a una instalación particular.]

* 1. Funciones del producto

Las funciones generales del entorno de programación son:

* 1. Realizar la conexión uno a uno entre el dispositivo móvil y el robot.
	2. Realizar secuencias de acciones.
	3. Implementar estructuras de control: condicional y ciclo.
	4. Implementar actuadores: motores y lámparas.
	5. Implementar sensores: distancia, tacto y sonido.
	6. Incorporar niveles en el entorno, para ir enseñando los comandos poco a poco por parejas.
	7. Tener un proceso de programación correctivo y progresivo.
	8. Guardar localmente el último programa realizado.

Para describir de forma más precisa las funcionalidades identificadas para el sistema, se han organizado de la siguiente manera:

|  |  |
| --- | --- |
| **Actores y características del sistema** | **Funciones** |
| Mediador | * Seleccionar tipos de robots y robots personalizados
* Asignar robot personalizado con su respectiva dirección de *bluetooth*
* Pasar de nivel a cada niño, de acuerdo a los logros alcanzados
 |
| Preescolar | * Crear programas que solucionen problemas planteados
* Enviar a ejecución programas creados
 |
| Otras | * Guardar de forma automática el último programa creado
* Tener un proceso de programación correctivo y progresivo
 |

Los expertos de la FOD definieron un conjunto de tareas específicas que se deberán de resolver con el entorno de programación propuesto:

* Realizar una secuencia de acciones de movimiento.
	+ Llevar al robot de un punto a otro.
* Realizar una secuencia de acciones de movimiento, prendiendo y/o apagando luces.
* Crear un robot (móvil o estático) con una garra para mover objetos de un lugar a otro.
* Realizar la repetición de una secuencia de acciones.
* Realizar determinado conjunto de acciones si se da una condición específica.
	1. Características de los usuarios

El entorno de programación está pensado para ser utilizado bajo mediación, por lo que está dirigido a docentes de preescolar con experiencia en el uso de tecnologías en ambientes educacionales para enseñar programación y robótica.

El público meta son las niñas y los niños entre 4 y 6 años, ya que ellos serán quienes utilicen el entorno bajo la mediación de los docentes. De acuerdo a nuestro sistema de educación costarricense, se comprende los ciclos Materno Infantil y el de Transición. El Ciclo Materno Infantil comprende a las niñas y a los niños desde los cuatro años y tres meses de edad, hasta los cinco años y dos meses. Este grupo es conocido como grupo de edad Interactivo II. El Ciclo de Transición comprende a las niñas y a los niños desde los cinco años y tres meses, hasta los seis años y dos meses.

Para el acceso al sistema, se definen dos perfiles de usuario: docente y estudiante de preescolar. Dependiendo de su perfil, e usuario podrá ejercer las funciones descritas en la sección 2.2.

* 1. Restricciones de diseño

El entorno de programación tendrá restricciones de acuerdo a las características de los dispositivos móviles que se utilicen, pero al menos los dispositivos deben contar con las especificaciones de la sección 2.1.2. Por ejemplo, alguna de las restricciones, el tipo de voz del TTS; ya que se usarán las suministradas por la aplicación.

Por otro lado, se tiene que tomar en cuenta las restricciones de memoria de los ladrillos NXT a la hora de definir el programa que interpreta el protocolo de comunicación y los programas que se crean a través del entorno; ya que el ladrillo cuenta con una memoria flash de 113KB y una memoria RAM de 16KB.

* 1. Supuestos y dependencias

Algunos factores que pueden afectar los requerimientos del sistema son:

* Los dispositivos móviles utilizados deben contar con al menos las especificaciones de la sección 2.1.2; en caso contrario alguna de las funciones del entorno de programación no podrán ejecutarse.
* El ladrillo NXT debe tener el firmware de leJOS NXJ con la versión 0.9.1 para que funcione adecuadamente el programa de interpretación del protocolo de comunicación.
1. Requerimientos

El entorno de programación propuesto tiene solamente dos actores: mediador y estudiante de preescolar.

* **Mediador:** Docente de preescolar con experiencia en el uso de tecnologías en ambientes educacionales para enseñar programación y robótica.
* **Preescolar:** Niñas y niños entre 4 y 6 años que utilizarán el entorno bajo la mediación de los docentes para programar robots. De acuerdo a nuestro sistema de educación costarricense, se comprende los ciclos Materno Infantil y el de Transición

De acuerdo a esto, se presentan los casos de uso para las funciones llevadas a cabo por cada actor del sistema.

**Seleccionar tipos de robots**

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Selección de los tipos de robots del entorno de programación |
| **Descripción** | Selecciona los tipos de robots que podrán ser utilizados por los niños en el entorno de programación, si desean agregar uno nuevo podrán hacerlo. |
| **Actores** | Mediador |
| **Precondiciones** | Ser un usuario mediador y haber entrado al módulo de configuración. |
| **Poscondiciones** | El entorno de programación mostrará los tipos de robots seleccionados. |
| **Flujo principal** | 1. Este caso de uso empieza cuando el mediador selecciona la opción de seleccionar tipos de robots.
2. El sistema le muestra la lista de tipos de robots que contiene (fotos de los robots) agregados.
3. El mediador selecciona los tipos de robot que desea.
4. En caso que quiere agregar otros tipos debe dar clic al botón de agregar tipo de robot y se ejecuta el subflujo S.1.
5. El caso de uso termina cuando el mediador sale de la opción.
 |
| **Subflujos** | S.1. El sistema activa la cámara del dispositivo para que el mediador tome la foto del tipo de robot, luego debe dar clic en el botón agregar. Se continúa en el punto 2. |

**Seleccionar robots personalizados**

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Selección de los robots personalizados del entorno de programación |
| **Descripción** | Selecciona los robots personalizados que podrán ser utilizados por los niños en el entorno de programación, si desean agregar uno nuevo podrán hacerlo. |
| **Actores** | Mediador |
| **Precondiciones** | Ser un usuario mediador y haber entrado al módulo de configuración. |
| **Poscondiciones** | El entorno de programación mostrará los robots personalizados seleccionados. |
| **Flujo principal** | 1. Este caso de uso empieza cuando el mediador selecciona la opción de seleccionar robots personalizados.
2. El sistema le muestra la lista de robots personalizados que contiene agregados.
3. El mediador selecciona los robots personalizados que desea.
4. En caso que quiere agregar otros robots personalizados debe dar clic al botón de agregar problema a resolver y se ejecuta el subflujo S.1.
5. El mediador debe asignar cada robot personalizado con su respectiva dirección de *bluetooth* (ver Caso de uso: Asignación de conexión uno a uno).
6. El caso de uso termina cuando el mediador sale de la opción.
 |
| **Subflujos** | S.1. El sistema activa la cámara del dispositivo para que el mediador tome la foto del robot personalizado, luego debe dar clic en el botón agregar. Se continúa en el punto 2. |

**Asignar robot personalizado con su respectiva dirección de *bluetooth***

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Asignación de conexión uno a uno |
| **Descripción** | Asigna cada robot personalizado, que podrán ser utilizados por los niños en el entorno de programación, con su respectiva dirección de *bluetooth.* |
| **Actores** | Mediador |
| **Precondiciones** | Ser un usuario mediador, haber entrado al módulo de configuración y haber seleccionado los robots personalizados a utilizar. |
| **Poscondiciones** | El entorno de programación mostrará los robots personalizados seleccionados con su respectiva dirección de *bluetooth*. |
| **Flujo principal** | 1. Este caso de uso empieza cuando el mediador selecciona la opción de asignar direcciones *bluetooth*.
2. El sistema le muestra la lista de robots personalizados seleccionados.
3. El mediador debe dar clic en cada robot personalizado.
4. El sistema muestra una lista de direcciones *bluetooth*.
5. El mediador debe seleccionar la dirección *bluetooth* respectiva al robot personalizado que le dio clic.
6. El caso de uso termina cuando el mediador ha asignado a cada robot personalizado su respectiva dirección *bluetooth* y sale de la opción.
 |

**Pasar de nivel a los niños**

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Cambio de nivel a cada niño de acuerdo a sus logros |
| **Descripción** | Cambiar de nivel a las niñas y los niños cuando hayan comprendido y asimilado cada par de comandos*.* |
| **Actores** | Mediador |
| **Precondiciones** | Ser un usuario mediador y haber entrado al módulo de cambio de nivel. |
| **Poscondiciones** | El entorno de programación mostrará una pantalla donde se pide con palabras se realice el dibujo de una figura *X* (la solicitud de figura es aleatoria). |
| **Flujo principal** | 1. Este caso de uso empieza cuando el mediador selecciona la opción de cambiar de nivel.
2. El sistema le muestra una pantalla con el nombre de una figura que debe dibujar de una forma dada (por ejemplo: un círculo).
3. El mediador debe dibujar la figura solicitada por el sistema.
4. Si el mediador realizó correctamente la figura se pasa de nivel al niño, en caso contrario se le solicita otra figura.
5. El caso de uso termina cuando el mediador haya dibujado la figura solicitada y se haya pasado de nivel al niño.
 |

**Crear programas que solucionan problemas planteados**

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Creación de programas que solucionan los problemas planteados |
| **Descripción** | Crea los programas que solucionan los problemas planteados, eso lo realizan los niños entre 4 y 6 años de edad*.* |
| **Actores** | Preescolar |
| **Precondiciones** | Ser un usuario preescolar, el entorno de programación debe tener asignado para cada tipo de robot diferentes problemas para solucionar y haber entrado al módulo de creación de programas. |
| **Poscondiciones** | Los programas creados se guardarán localmente y en el Web. |
| **Flujo principal** | 1. Este caso de uso empieza cuando el preescolar da clic al botón de jugar (módulo de creación de programas).
2. El sistema muestra los tipos de robots disponibles.
3. El preescolar selecciona el tipo de robot que desea.
4. El sistema muestra los robots personalizados disponibles.
5. El preescolar selecciona a su robot personalizado para realizar la conexión uno a uno.
6. El sistema muestra los comandos respectivos y el ambiente de trabajo para empezar a resolver el problema seleccionado.
7. El preescolar comienza a programar la solución del problema (crear su cuento).
8. Si el preescolar da en el botón de atrás o manda a ejecutar, el entorno guarda de forma automática el último programa creado.
9. El caso de uso termina cuando el preescolar termina de programar.
 |

**Enviar a ejecución programas creados**

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Ejecución de programas creados |
| **Descripción** | Envía a ejecución los programas creados por los niños entre 4 y 6 años de edad*.* |
| **Actores** | Preescolar |
| **Precondiciones** | Ser un usuario preescolar, haber entrado al módulo de creación de programas y tener programas creados. |
| **Poscondiciones** | Los programas creados se guardarán localmente y en el Web. |
| **Flujo principal** | 1. Este caso de uso empieza cuando el preescolar da clic al botón de ejecutar (módulo de creación de programas).
2. El sistema envía el conjunto de instrucciones al robot para que este las ejecute, al mismo tiempo guarda el programa creado en la ruta especificada en el módulo de configuración por el mediador.
3. El robot recibe el conjunto de instrucciones e interpreta cada una de acuerdo al protocolo de comunicación definido.
4. El sistema muestra paso a paso cómo va ejecutando cada comando el robot (proceso de programación correctivo y progresivo).
5. El caso de uso termina cuando el robot termina de ejecutar el programa creado.
 |

* 1. Requerimientos Suplementarios

El sistema propuesto debe contener una serie de componentes necesarios de un sistema encargado de crear aplicaciones para ser ejecutadas por robots. Por lo cual, se describen los componentes en relación a usabilidad y funcionalidad.

Este sistema es incremental, por lo que se irán agregando poco a poco funcionalidades, y en versiones posteriores estará dirigido a poblaciones de niños mayores de 6 años.

Conjunto de Comandos

Al trabajar con niños tan pequeños, es importante tener un pequeño conjunto de comandos específicos, por lo que el entorno debe tener entre 8 y 10 comandos, estos deben agruparse de acuerdo a su función utilizando colores (primarios o secundarios).

Entre el conjunto de comandos especificados por expertos se tienen:

* Inicio y fin.
* Encender y apagar actuadores.
	+ Tiempo específico de encendido o apagado de un motor.
	+ Motor, seleccionar motor (A, B, C). Cambiar la potencia y el giro. Dirección: adelante y atrás.
	+ Lámpara, seleccionar lámpara (A, B, C). Cambiar la intensidad.
	+ Apagado completo (*stop* general).
* Activar y desactivar sensores.
	+ Tacto, apretado y suelto.
	+ Distancia, cerca y lejos.
	+ Sonido, hay ruido o silencio, alto o bajo.
* Espera por: una señal del sensor, tiempo, cantidad de veces, etc.

La mayoría de las funcionalidades de estos comandos se visualizan para versiones posteriores (para poblaciones de niños mayores de 6 años), para la población de niños entre 4 y 6 años se visualiza dichos comandos pero con funcionalidades simples (ya mencionadas en la sección 3).

Estructuras de Control

* Secuencia.
* Ciclo.
	+ Finito.
	+ Infinito.
* Condición.
* Combinación de ambas.
* Llamado a procedimientos, modular.

La estructura de control que se implementará primero para la población de niños entre 4 y 6 años será la secuencia. Luego, se incorporará el ciclo y la condición. El llamado a procedimientos se espera realizar en versiones posteriores.

Otros Requerimientos

* *Multitouch*, esto permite la colaboración.
* Análisis de métricas.
* Conexión uno a uno.
* Mensaje que el *bluetooh* está desactivado.
* Iconos que se arrastre a cualquier parte de la pantalla o dar *clic*, y que se acomode en la secuencia de instrucciones.
* El proceso debe ser correctivo y progresivo, cuando se corre el programa en el robot se debe visualizar en las instrucciones por dónde se va ejecutando.
* Reconocer, sólo si hay un actuador o un sensor conectado, cuál puerto es. Si hay más de dos, se debe diferenciar (implementación en versiones posteriores).
1. Requerimientos de documentación

En esta sección se especifica la documentación del sistema propuesto.

* 1. Manual de Usuario

El manual de usuario del entorno de programación tendrá como propósito especificar las características y funcionalidades del sistema. El manual de usuario estará orientado a los usuarios mediadores del sistema.

* 1. Ayuda en línea

El entorno de programación no tendrá ayuda en línea.

* 1. Guías de instalación, configuración y archivo Léame.

El sistema al ser una aplicación móvil para Android no tendrá una guía de instalación y configuración.

* 1. Etiquetado y empaquetado

El sistema al ser una aplicación móvil para Android será un archivo APK que se subirá a Google Play de forma gratuita, ya que será una aplicación para apoyar el sector educativo preescolar de nuestro país.

Por su parte, el programa instalado en el robot que interpreta los comandos recibidos del dispositivo móvil será un archivo NXJ, este programa se ofrecerá en el sitio Web de forma gratuita.

**Historia de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 16/10/2013 | 001 | Creación del documento | Kryscia Ramírez B. |
| 15/05/2014 | 002 | Adición de requerimientos (incorporación de la metáfora establecida) | Kryscia Ramírez B. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |