

Mecanismo programable para niños

Aldama Pérez CristopherZavala Ventura Miguel Ángel

2015

1. Resumen

Mecanismo programable para niños es un sistema que consta de un lenguaje gráfico de programación y un bloque microcontrolador al que se le pueden conectar sensores (contacto, luz y temperatura) y actuadores (motores), que sirve como apoyo a la enseñanza de programación y robótica a niños de primaria de entre 7 y 11 años.

Palabras clave: Lenguaje, Sensor, Actuador, Robótica, Programación, Educación.

2. Advertencia

“Este trabajo contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional a partir de datos y documentos con derecho de propiedad y por lo tanto su uso queda restringido a las aplicaciones que explícitamente se convengan.”

Índice

1. Resumen	1
2. Advertencia	2
3. Objetivo	4
4. Introducción	5
5. Análisis	6
5.1. Planteamiento del problema	6
5.2. Objetivo	6
5.2.1. Objetivo General	6
5.2.2. Objetivos Específicos	6
5.3. Justificación	7
5.4. Productos o Resultados Esperados	8
6. Marco Teórico	8
6.1. Metodología Pedagógica	9
6.2. Perfiles infantiles	9
6.2.1. Segundo grado de primaria	9
6.2.2. Tercer y cuarto año de Primaria	10
6.2.3. Quinto año de primaria	11
6.3. Retroalimentación en escuelas	12
6.4. Intérpretes	12
6.4.1. Ventajas de los intérpretes	13
6.4.2. Desventajas de los intérpretes	13
7. Diseño	13
7.1. El lenguaje está constituido por 4 tipos de elementos	14
7.2. Gramática del Lenguaje	14
7.3. Tipo de Datos	15
7.4. Operaciones	16
7.5. Palabras reservadas	16
7.6. Equivalencia Gráfica	16
7.7. Ejemplo de programa	17
7.8. Requisitos funcionales	17
7.8.1. EDI Olinki	17
7.8.2. Mecanismo Olinki	17
7.8.3. Requisitos no funcionales	18
7.8.4. EDI Olinki	18
7.8.5. Mecanismo Olinki.	19

3. Objetivo

Analizar, implementar y diseñar un sistema de cómputo, tanto en hardware como en software que permita a niños de entre 7 y 11 años crear programas simples usando iconos gráficos, así como su ejecución e interacción con sensores y motores, con la finalidad de ayudar en la enseñanza de programación y uso de computadoras.

4. Introducción

En el campo de la educación básica, tanto en escuelas públicas como privadas, una de las principales preocupaciones es enseñar conceptos relacionados con la tecnología, debido a la exposición que la sociedad tiene con ella es cada vez mayor, y para ello desarrollan competencias en las que motivan el conocimiento, uso y aplicación de la computadora en las tareas de la vida diaria; sin embargo, aunque en el mercado existen diversos materiales para su enseñanza, es difícil encontrar alguno que mantenga el interés de los niños pequeños y se ajuste al ritmo en el que absorben las ideas.

Actualmente los dispositivos en los que se apoya la enseñanza de estos conceptos son circuitos básicos, que están listos para armarse, sin embargo limitan la interacción a la observación de su funcionamiento, lo cual pierde trascendencia e interés al poco tiempo. También existen sistemas más robustos, mecanismos controlados por un programa de computadora, para el que se necesita un nivel de abstracción mayor, pues requiere de la comprensión de conceptos de matemáticas y lógica.

La propuesta de este proyecto consiste en crear un dispositivo cuyo funcionamiento pueda ser aprendido de forma gradual haciendo uso del juego y para ello se debe analizar, diseñar, probar e implementar un sistema mecánico programable que sirva como material auxiliar en la enseñanza de conceptos, en el área de la lógica y la programación de sistemas de cómputo, enfocado en niños de escuelas primarias, con edad de entre 7 y 11 años de edad.

5. Análisis

En este capítulo se plantea el problema en el que este proyecto se enfoca, así como determinar los objetivos específicos, la justificación y se en listan los resultados esperados.

5.1. Planteamiento del problema

La tecnología va adquiriendo día a día un lugar más importante en el desempeño de las tareas diarias, que van desde las compras en el súper mercado, operaciones bancarias, entretenimiento y actividades lúdicas. La computadora y sus aplicaciones tienen un rol central en el desarrollo de la sociedad, es por eso que escuelas, en especial las de educación primaria buscan herramientas que ayuden a sus alumnos a tener un conocimiento adecuado sobre las ciencias de la computación, que sirvan como base en el desarrollo del individuo.

Este trabajo terminal, se presenta como una herramienta para la educación primaria que los maestros pueden aprovechar para facilitar la enseñanza de conceptos elementales sobre el funcionamiento interno de las computadoras así como despertar el interés de los alumnos en la programación de computadoras. Por medio de un lenguaje de programación simple pero que toca los aspectos básicas de programación y hardware en forma de sensores y motores que pueden ser usados para armar mecanismos.

5.2. Objetivo

A continuación se listan el objetivo general del proyecto así como los objetivos específicos.

5.2.1. Objetivo General

Analizar, implementar y diseñar un sistema de cómputo, tanto en hardware como en software que permita a niños de entre 7 y 11 años en escuelas de educación primaria crear programas simples usando iconos gráficos, así como su ejecución e interacción con sensores y motores en una computadora de bajo costo que actúe como controlador principal, con la finalidad de ayudar en la enseñanza de programación y uso de computadoras en general.

5.2.2. Objetivos Específicos

- Crear un lenguaje de programación gráfico, Olinki a partir de ahora.
- Crear un entorno de desarrollo integrado (EDI) con soporte para Olinki.
- Implementar un interprete del lenguaje de programación Olinki.

- Diseñar circuitos electrónicos que den soporte a los sensores de iluminación , contacto y temperatura, así como a los motores eléctricos.
- Motivar al alumno mediante el uso estímulos visuales.
- Definir ejemplos que muestren las capacidades del lenguaje de programación.
- Diseñar una carcasa que proteja los circuitos, así como el controlador principal.
- Realizar pruebas automatizadas que muestren fallas en el diseño del lenguaje de programación.

5.3. Justificación

Con la integración de las computadoras a nuestra vida diaria en forma de teléfonos celulares, relojes inteligentes, consolas de vídeo juegos, tabletas, computadoras personales, etcétera, ha surgido una corriente que propone la enseñanza de programación en escuelas de educación primaria como apoyo en el entendimiento de la manera en que funcionan las computadoras y sus aplicaciones, y que ha sido adoptada en países como Estonia (2012) e Inglaterra (2014), y otros que están haciendo planes para incluirla en su plan de estudios como Finlandia, EUA, Singapur, Dinamarca, Israel y Australia. El objetivo de exponer a los niños al uso de computadoras tan pronto como sea posible y desarrollar habilidades técnicas en ellos, es el de prepararlos en el mundo tecnológico en el que viven inmersos, además de alimentar su curiosidad en el área de ciencias de la computación con la meta de satisfacer la demanda de profesionales en el área.

Por ello se toma en cuenta que el acercamiento a la programación por niños de primaria, requiere de herramientas adecuadas, que simplifiquen el proceso de crear y usar un algoritmo para resolver un problema en específico así pues se buscan lenguajes de programación amigables con los niños pequeños como son: Alice, Scratch, Turtle entre otros. Lenguajes que fueron creados para la educación y que hacen uso de elementos gráficos para la creación de programas simples así como colores llamativos, sentencias simples, animaciones, etcétera, pero que sin embargo solo están disponibles en el idioma inglés o que no tienen manera de interactuar directamente con hardware.

Por otro lado, en nuestro país la Reforma Integral de Educación Básica (RIEB) anima a los docentes a hacer uso de la tecnología con la finalidad de reforzar las clases y consolidar los conocimientos adquiridos, enfocándose en las competencias de los alumnos, las estrategias tomadas por esta reforma son las de capacitar a los docentes en el uso de recursos multimedia, de medios de comunicación, el internet y creación de infraestructura como enciclopedia. Sin embargo no se hace mención de la enseñanza de temas o materias de las ciencias de la computación en las aulas, de tal manera que la tecnología puede ser usada

como apoyo complementario en la enseñanza de las materias y cursos (aprender con tecnología) o como modelo pedagógico (aprender de la tecnología).

Se considera de gran importancia la elaboración de este proyecto ya que propone la realización de un lenguaje de programación simplificado, con elementos gráficos y en español el cual puede ser usado como herramienta en la enseñanza de los conceptos clave de las ciencias de la computación como son: creación de algoritmos y programación de computadoras, además de la experimentación incitando al usuario a diseñar, armar y mejorar sus propios diseños de software y hardware. Que complementa el modelo propuesto por la RIEB, poniendo como actor a la tecnología en este caso la computadora y sus aplicaciones. El juego cobra vida cuando se combina la creación del programa con el ensamblado pieza por pieza del mecanismo, permitiendo llevar a cabo las ideas que se desarrollaron en la imaginación del niño, motivándolo a mejorar y superar lo logrado anteriormente.

5.4. Productos o Resultados Esperados

Al concluir este proyecto, se espera tener un intérprete del lenguaje de programación propuesto (Olinki), un entorno de desarrollo integrado(EDI) con soporte gráfico para el lenguaje de programación Olinki, el cual permita la creación de programas de computadora de manera lúdica y simple mediante el uso de iconos.

Además de una plataforma de hardware basado en la raspberry pi donde se podrá programar y utilizar el EDI Olinki sin necesidad de una computadora externa, así como un sensor de temperatura, luminosidad, y contacto conectables a ésta.

De igual manera, contará con un manual de usuario donde vendrá descrito el lenguaje de programación, el uso del EDI, así como dos programas de ejemplo con su respectiva contra parte en hardware.

6. Marco Teórico

Para fundamentar el desarrollo y la dirección del proyecto fue necesario indagar conceptos teóricos que le diesen forma y contexto funcional, por lo que se realizó investigación para conocer el ambiente de enseñanza en el que se desarrollan los niños. Se produjeron una serie de entrevistas con una especialista en pedagogía para aclarar dudas concretas con la comprensión de términos, ideas y conceptos pedagógicos. Para reafirmar lo anterior con experiencias prácticas se procedió a buscar el contacto e interacción con niños de escuelas para conocer el posible ambiente en el que probar el mecanismo Olinki.

6.1. Metodología Pedagógica

Como apoyo pedagógico se tomaron en cuenta las teorías contructivista y construccionista, ambas centran al aprendizaje como un proceso dinámico que el individuo debe construir activamente, en el caso de la primer teoría, el constructivismo cuyos principales autores fueron Jean Piaget y Lev Vygotski, postula la necesidad de dar herramientas al alumno que le permitan desarrollar una solución a un problema o situación presentada, tiene como base la idea de que el conocimiento se escala, es decir que el conocimiento o experiencia adquirida por el sujeto, servirá como andamio para la construcción de nuevo conocimiento apoyado en la interacción social o con el medio (aprender haciendo), así la persona que aprende queda como actor principal de su propio aprendizaje. Por otro lado, el construccionismo parte de las ideas del constructivismo llamando artefactos a las construcciones mentales del individuo que aprende, así la realización de actividades como en el caso de este proyecto: la escritura de un programa y el armado de circuitos básicos, cumplen la función de facilitadores del aprendizaje, es de notar que el construccionismo desarrollado por Seymour Papert da mayor peso a la parte social, dando gran importancia a las relaciones interpersonales en el uso de material para desarrollar conocimiento.

Los planteamientos anteriormente descritos se tomaron en cuenta en la elección de estas corrientes ya que se basan en un sistema social. Hacen referencia a que en una sociedad la retroalimentación entre sus personajes es obligada porque la comunicación entre sus individuos es la base para su desarrollo; dentro de uno de los conceptos que componen a esta tendencia, está la definición del establecimiento de las reglas de las dinámicas de interacción por el conjunto de sus individuos; otro idea que la compone es que en la realidad de un sujeto, su sistema de acciones es resultado de las interacciones que tiene con la sociedad en la que se desenvuelve. Como vemos ahora, el paradigma socio-construccionismo antecede la relación entre personas a la individualidad y es así como cada sujeto modela su realidad. De este modo podemos analizar el aprendizaje significativo de un sujeto cuando a este se le coloca en un entorno social con el que pueda interactuar con otros individuos, con reglas a consensuar o preestablecidas, para comenzar a experimentar la construcción de conocimiento.

6.2. Perfiles infantiles

6.2.1. Segundo grado de primaria

En el momento en que los niños llegan a esta edad (siente años aproximadamente) buscan la aceptación de los adultos. Algunos demandarán atención de su profesor y se puede ver afectado si no se siente especial. Es importante fortalecer habilidades emocionales como:

- Auto conocimiento
- Control de emociones e identificación de emociones
- Fortaleza emocional

- Determinación
- Auto control
- Auto estima

Mientras desarrollan habilidades intelectuales como:

- Rastreo Visual
- Ubicación espacial
- Clasificación y comparación
- Identificación de patrones y secuencias
- Figuras en espejo
- Ubicación temporal / espacial
- Comparación y asociación de objetos y analogías
- Situaciones de la vida cotidiana
- Transferencia del conocimiento
- Inferencias

6.2.2. Tercer y cuarto año de Primaria

En esta etapa suelen volverse pueden crueles con los comentarios que hacen el uno al otro. Se deben orientara ser objetivos en el tema de las amistades y generar las habilidades necesarias para lograr relaciones de ganar ganar en todo momento. Es importante fortalecer habilidades emocionales como:

- Auto conciencia
- Planeación
- Organización
- Inteligencia emocional
- Autonomía

Mientras desarrollan habilidades intelectuales como:

- Clasificación y comparación
- Descripción
- Interpretación
- Comparación y asociación de objetos y analogías

- Situaciones de la vida cotidiana
- Transferencia del conocimiento
- Inferencias
- Análisis
- Lateralidad
- Observación
- Retención

6.2.3. Quinto año de primaria

Algunos niños pueden sentirse presionados a experimentar. Los niños pasan por una gran cantidad de estrés emocional generado por el entorno, las cuestiones de popularidad y cuestiones personales. Habilidades emocionales:

- Auto conciencia
- Empatía
- Asertividad
- Inteligencia emocional
- Autonomía
- Comunicación
- Solución eficaz de conflictos
- Establecimiento de metas

e intelectuales:

- Razonamiento lógico
- Decodificación e interpretación
- Identificación y selección de información
- Inferencia
- Deducción
- Metacognición (Capacidad del individuo para trascender y re-aplicar su propio conocimiento)

La información de los perfiles de comportamiento de los niños en diferentes edades, en conjunto con la información obtenida en la investigación y la aclaración de conceptos en la retroalimentación pedagógica, sirvió como punto de partida para saber qué comportamientos observar en los niños al momento de visitar escuelas. De manera simultánea, brinda palabras clave para el diseño de la interfaz y la posible retroalimentación que el sistema Olinki le dará al usuario final; así mismo, la aclaración de los conceptos y paradigmas pedagógicos como el construccionismo puede ser usado como guía en el diseño del mecanismo Olinki encajando en las ideas de ésta corriente pedagógica.

6.3. Retroalimentación en escuelas

El proyecto está enfocado principalmente a niños que oscilan entre los 7 y 11 años, lo que implica que se debe conocer cómo es el ambiente en el que se desenvuelven para afinar detalles con el diseño del sistema Olinki. Para esto era necesario convivir con grupos de niños de diferentes edades en un ambiente en el que estén aprendiendo de manera habitual, entonces se buscó un espacio en diferentes escuelas dónde se permitiera poder observar a alumnos de segundo a quinto año.

Las escuelas que se visitaron son el instituto Atenea y el Colegio Alamilla Americano. En ellas se permitió acceso a los salones de computación para presenciar las clases que les imparten a los alumnos de segundo a quinto grado. En esta experiencia se observó el tipo de programas con los que practican los niños el uso de la computadora, cómo son las gráficas de los programas que utilizan para las dinámicas, los conceptos que conocen y ocupan dentro de lo que conocen de la computación; además se conoció el tipo de lenguaje que ocupan las maestras para comunicarse con los alumnos.

6.4. Intérpretes

Se puede considerar a un intérprete como un procesador de lenguaje que analiza un programa escrito en algún lenguaje de programación, y si es correcto, lo ejecuta directamente en el lenguaje nativo de la máquina que lo está ejecutando, para cada ejecución se debe interpretar de nuevo el lenguaje a diferencia de los compiladores, que emiten código nativo una sola vez para ser ejecutado posteriormente. Existen algunas razones por las que algún lenguaje solo pueda ser interpretado pero no compilado, como son:

- El lenguaje tiene operadores difíciles de compilar, por ejemplo si se tratan las cadenas de texto como operaciones ejecutables, no es el caso de Olinki.
- Se ha eliminado la declaración de variables, que pasa a ser implícita como en el caso de Olinki.
- No hay mecanismos explícitos de control de memoria y es el intérprete quien la gestiona, en este caso como se verá más adelante Olinki aprovecha esta característica para simplificar el manejo de variables.

- La presencia del intérprete es por razones de seguridad, no es el caso de Olinki.

El hecho de que los compiladores e intérpretes coexistan, sugieren que ambos tipos de procesadores de lenguajes tienen cada uno ventajas en ciertos aspectos, a continuación se enlistan las ventajas y desventajas de los intérpretes:

6.4.1. Ventajas de los intérpretes

- Flexibilidad: Los lenguajes interpretados suelen tener más flexibilidad y permiten realizar operaciones más complejas, como la ejecución de cadenas de texto, cambiar sobre la marcha el contenido o tipo de símbolos e inclusive descartar la declaración explícita de variables.
- Gestión de Memoria: El intérprete toma por completo la tarea de tomar y liberar memoria, facilitando al programador la depuración y escritura del programa.
- Desarrollo: Como se anotó anteriormente los lenguajes interpretados facilitan la escritura de programas, automatizando las tareas de gestión de memoria, depuración, y ofreciendo mayor expresividad.

6.4.2. Desventajas de los intérpretes

- Velocidad de ejecución: Generalmente los programas interpretados son varias veces más lentos que los compilados, debido a que cada vez que se ejecutan se debe procesar el código fuente para su posterior ejecución.
- Tamaño del programa: Los programas escritos en lenguajes interpretados suelen ser más grandes que los compilados, ya que estos se encuentran en lenguajes intermedios o de programación más generales que el código nativo.

7. Diseño

Se seleccionó como nombre del lenguaje de programación la palabra Nahuatl “Olinki” que significa máquina, como reflejo de la parte técnica del proyecto, es decir el propio lenguaje de programación, el lenguaje que se propone para este proyecto, tiene como objetivo la construcción de programas simples que serán ejecutados por el controlador principal con la finalidad de operar motores, sensores e indicadores, haciendo uso de pictogramas o iconos para representar las estructuras de control y sentencias del lenguaje, así como una representación en forma de texto como referencia.

7.1. El lenguaje está constituido por 4 tipos de elementos

1. Variables de Entrada y Salida: Tienen como propósito alimentar al programa con datos tomados del entorno físico mediante el uso de sensores, así como entregar resultados hacia los actuadores (motores eléctricos, LEDs).
2. Variables Internas: Contienen valores alfanuméricos, usados como pasos intermedios, o valores temporales en el cálculo de expresiones.
3. Expresiones: Expresan relaciones entre los dos tipos de variables que se nombraron anteriormente, dichas expresiones son de naturaleza lógico-matemáticas (Sumas, Restas, Multiplicación, Concatenación, Negación, etcétera).
4. Bloques de Control: Afectan el flujo del programa mediante el uso de operadores lógicos.

7.2. Gramática del Lenguaje

Se propone la siguiente gramática para el lenguaje Olinki, la cual expresa la naturaleza de las expresiones y relaciones entre las variables:

```

S -> I
I -> inicio V
V -> V | F
F -> fin
V -> C
C -> si E { V } Sn
Sn -> sino { V } | epsilon
E ->      var mayor que var |
      var menor que var |
      var mayor igual que var |
      var menor igual que var |
      var igual var |
      var diferente var
E ->      E y E |
      E o E |
      o E o E |
      no E
V -> M
M ->      var + var |
      var - var |
      var * var |
      var / var |
      var % var
C -> A
A -> nombre = M | F | E | S

```

7.3. Tipo de Datos

Internamente las variables pueden ser del tipo: cadena de texto, numérico y lógico, donde cadena representa una secuencia de caracteres, numérico representa enteros y números de coma flotante y los del tipo lógico representan valores que pueden tomar como valor Verdadero o Falso. La mutación de tipos dentro del lenguaje es transparente al programador, estas son las reglas que rigen dicha conversión.

Tipo	Numérico	Cadena	Lógico
Numérico	Valor	"Valor"	Valor ¿= 1? V : F
Cadena	Valor — Longitud("Valor")	Valor	"Verdadero"-¿V, otro =¿F
Lógico	Valor? 1 : 0	Valor? "Verdadero": "Falso"	Valor

7.4. Operaciones

Las operaciones entre las variables quedan definidas en términos del tipo de dato.

Operación	Numérico V1,V2	Cadena V1,V2	Lógico V1,V2
Suma	V1 + V2	Concatenación(V1, V2)	OR(V1, V1)
Resta	V1 - V2	NA	NA
Multipliación	V1 * V2	NA	AND(V1, V2)
División	V1 / V2	NA	NA
Longitud	Valor Absoluto(V1)	Longitud(V1)	V1?1:0

7.5. Palabras reservadas

INICIO: Determina inicio de programa
SI: Primera Condicional
SINO: Segunda Condicional
REPETIR: Inicio de Ciclo de iteración
ESCRIBE: Mostrar cadena en pantalla
LIMPIAR: Limpiar pantalla
MOTOR: Activar motor
SENSOR: Leer valor de sensor
FIN: Terminar Bloque de Iteración, Condicional o Programa
Y: Operador lógico AND
O: Operador lógico OR
NO: Operador lógico de negación

7.6. Equivalencia Gráfica

Las palabras reservadas anteriormente listadas forman reglas gramaticales al principio de este capítulo, estas reglas tienen un equivalente gráfico que facilita la creación de programas mediante el uso de una interfaz gráfica de usuario y acciones drag AND drop. La equivalencia entre reglas gramaticales y elementos gráficos es 1 a 1, es decir que existe un elemento gráfico por cada regla, así estos elementos gráficos son usados como piezas que al ser puestos juntos forman un nuevo programa. Se en lista la equivalencia:

7.7. Ejemplo de programa

```
INICIO
suma = 0
REPETIR 5
suma = suma + 1
FIN
SI suma = 5
ESCRIBE "Hola"
SINO
ESCRIBE "Adiós"
FIN
FIN
```

7.8. Requisitos funcionales

En esta sección conoceremos las acciones que podrá realizar Olinki cuando se manipulado por el usuario. Primero hablaremos del comportamiento que tendrá el EDI Olinki (Entorno de desarrollo integrado Olinki) y posteriormente la variedad de funciones que es posible realizar

7.8.1. EDI Olinki

EDI Olinki es un sistema que puede ser manipulado desde su instalación en un equipo de cómputo o desde la máquina programable; si se programa desde éste último necesitaremos auxiliar nos de una pantalla de televisor para poder visualizar el mismo entorno gráfico y navegar con los botones del mecanismo físico. A continuación conoceremos las funcionalidades de EDI Olinki

- El usuario podrá crear un programa con extensión de formato .oli en EDI Olinki.
- El programa que sea creado en EDI Olinki podrá ser abierto y visualizado para su edición.
- Será posible guardar en el equipo cualquier programa creado o editado en EDI Olinki siempre y cuando no exista algún otro con el mismo nombre.
- Una vez que el usuario haya terminado el programa podrá interpretarlo con EDI Olinki.

7.8.2. Mecanismo Olinki

El mecanismo Olinki es el conjunto de 3 motores y 3 sensores (de contacto, luz y temperatura) que pueden ser ensamblados y conectados en la máquina programable para que realicen alguna acción una vez que es ejecutado el programa creado en EDI Olinki. En seguida haremos una descripción de sus funciones.

1. Uno o más de los sensores podrán ser ensamblados con el cuerpo de la máquina programable Olinki.
2. Uno o más de los motores podrán ser ensamblados con el cuerpo de la máquina programable Olinki.
3. Los sensores podrán ser activados una vez que sea ejecutado el programa cargado en la máquina programable Olinki.
4. Los motores podrán ser activados una vez que sea ejecutado el programa cargado en la máquina programable Olinki.
5. El usuario podrá navegar en la pantalla de televisor con los botones de la máquina programable Olinki.

7.8.3. Requisitos no funcionales

En el siguiente apartado se detallarán características que deberá cumplir el equipo de cómputo en el cual se desee instalar EDI Olinki y los aspectos generales de su interfaz. También se hará una descripción de las características físicas que tendrá el mecanismo para conocer su posible comportamiento.

7.8.4. EDI Olinki

Comenzaremos con los requerimientos básicos para la instalación en un equipo de cómputo y especificaciones de la interfaz del EDI Olinki

1. EDI Olinki podrá ser instalado en cualquier computadora que tenga un procesador superior a 500 MHz, memoria RAM mayor a 512 MB, almacenamiento de disco duro arriba de los 2 GB, periféricos básicos (teclado y mouse).
2. EDI Olinki podrá ser instalado en cualquier sistema operativo Linux, Windows o OSX.
3. Para usar EDI Olinki desde un televisor, éste deberá contar una entrada HDMI o RCA Compuesta.
4. El entorno de programación de EDI Olinki resultará de simple uso en particular para los usuarios de pequeñas edades.
5. Con EDI Olinki el usuario podrá escoger el directorio para ubicar cada programa creado o editado.
6. En EDI Olinki el usuario podrá visualizar los programas que han sido creados.

7.8.5. Mecanismo Olinki.

En seguida presentaremos una descripción de una serie de aspectos que tendrá el mecanismo Olinki en su aspecto físico.

1. El equipo de cómputo deberá contar con uno o más puertos Ethernet.
2. El usuario no tendrá acceso al circuito interno de la máquina programable Olinki ni de sus componentes periféricos (motores y sensores).
3. Los botones de la máquina programable Olinki estarán colocados de tal manera que sea fácil la navegación durante la programación cuando se programe directamente y se quiera hacer uso de una pantalla de tv.
4. Los periféricos de Olinki no estarán conectados permanentemente a la máquina programable.
5. No será necesario que el usuario tenga conocimientos previos para que le resulte fácil el armado de la máquina programable Olinki con sus periféricos.