

# Aprendizaje móvil de lenguajes informáticos en educación primaria

Enrique Ruiz-Velasco Sánchez  
Instituto de Investigaciones sobre la Universidad y la Educación  
Universidad Nacional Autónoma de México

## Resumen

Una de las principales formas de expresión de la inteligencia humana se hace a través de los lenguajes naturales y, ahora también, de los lenguajes artificiales. El aprendizaje de lenguajes artificiales o informáticos es factible gracias a nuestro lenguaje natural o materno. Los lenguajes naturales y artificiales nos permiten solucionar problemas. Los lenguajes artificiales nos conceden ahora, resolver una gran cantidad de problemas de manera eficaz, clara, precisa y sobre todo, automatizada. La solución de problemas consiste en las respuestas que ofrecemos para dejarlos atrás. Las respuestas o soluciones a problemas vía los lenguajes artificiales, pueden ser comprendidas como expresiones mínimas y precisas de las respuestas que ofrecemos. Estas expresiones mínimas que contienen la solución al problema se traducen en secuencias, decisiones y/o repeticiones. Lo anterior conforma la programación informática. Aprender la filosofía de un lenguaje de programación informática resulta hoy día, una competencia o habilidad de base para enfrentar con éxito y tener una mejor comprensión del mundo tecnologizado en que vivimos. Es por ello, que mostraremos en esta comunicación una experiencia para volver factible el aprendizaje de la filosofía de los lenguajes de programación informática, vía el aprendizaje móvil, con estudiantes del nivel básico de educación primaria teniendo como argumento la concepción, puesta en marcha y programación de un robot pedagógico desde un laboratorio cibertrónico 3.0<sup>1</sup>.

## Palabras clave:

Aprendizaje móvil, lenguajes naturales, lenguajes artificiales, programación informática, filosofía de lenguajes de programación, robot pedagógico.

## *Mobile Learning of computer languages in primary education*

### **Abstract**

*One of the main forms of expression of human intelligence is through natural languages and now also of artificial languages . Learning artificial or computer languages is feasible thanks to our natural or native language . Natural and artificial languages allow us to solve problems. Artificial languages grant us now solve a lot of problems effectively , clear, accurate and above all automated . Problem solving is the answers we offer to leave them behind. The answers or solutions to problems via artificial languages , can be understood as minimum and precise expressions of the answers we provide. These minimum expressions containing the solution to translate into sequences , decisions and / or repetitions. This conforms computer programming. Learn the philosophy of a computer programming language is today a basic competence or ability to deal with success and have a better understanding of the technologized world we live. Therefore , it will show in*

---

<sup>1</sup> Es un entorno rico de aprendizaje, -espacio de experimentación, investigación, exploración- que favorece la **construcción de conocimientos** de manera colaborativa y entre pares; privilegia una **constante interacción del usuario con el mundo real** desde cualquier posición geográfica (acceso a Internet, bibliotecas, museos, archivos, etc.).

*this communication an experience to make feasible learning philosophy of computer programming languages via mobile learning , with students from the basic level of primary education having as argument the design, implementation and programming an educational robot from a cybertronic 3.0 laboratory.*

**Keywords:**

*Mobile learning, natural languages, artificial languages, computer programming, philosophy of programming languages, pedagogical robot.*

**Introducción**

El problema de la adquisición de los lenguajes naturales no es más que una parte del problema de la adquisición del conocimiento. Existen dos enfoques generales relativos a la adquisición de conocimientos. El conductista y el cognitivista.

El enfoque conductista, enseña que el aprendizaje del lenguaje es una construcción por condicionamiento o por enseñanza de procedimientos elementales del proceso de tratamiento de datos lingüísticos. El aprendizaje del lenguaje sería independiente en su estructuración de alguna facultad mental innata. Este enfoque presume que la estructura del dispositivo de adquisición de conocimientos está limitada a ciertos mecanismos de tratamiento periféricos tales como las funciones reflejas, ciertas funciones espacio-temporales y más precisamente en el caso del lenguaje, habilidades perceptivas relativas al oído. Este enfoque considera también, que este dispositivo posee mecanismos analíticos de tratamiento de datos y de principios inductivos elementales tales como los mecanismos de asociación mentales, los principios taxonómicos y los mecanismos de segmentación y de clasificación de dantos lingüísticos. En el caso del aprendizaje del lenguaje, un análisis preliminar de la experiencia es provisto por esos mecanismos periféricos. Los conceptos y los conocimientos son adquiridos una vez que el principio inductivo es aplicado a la experiencia lingüística vivida. Es así que según este enfoque, el lenguaje se aprende por condicionamiento, por ejercicios, por construcción de procedimientos, en fin, por tratamiento de datos elementales que son independientes de las estructuras innatas. Entre sus más ardientes defensores de este enfoque se encuentra Skinner (1957) quien propuso estímulos que vienen a controlar la elaboración del lenguaje por intermediación de un condicionamiento instrumental tal como las palabras escritas, el entorno, etcétera.

El enfoque cognitivista, supone que existen ideas y principios innatos de tipos variados que determinan la manera de adquisición del conocimiento de nuestro lenguaje. Este enfoque afirma que es necesario ser estimulado de manera adecuada para que los mecanismos de aprendizaje sean puestos en movimiento. Entre los defensores de esta teoría está Piaget (1970) quien dijo que el desarrollo intelectual del niño, incluyendo el lenguaje se hace en función del desarrollo de las estructuras cognitivas. Chomsky (1967) lanzó la hipótesis de que el desarrollo del lenguaje se hace en función del desarrollo de las estructuras cognitivas heredadas. Es decir, que un mecanismo mental innato volvería evidente la adquisición del lenguaje. Chomsky estudió el lenguaje como un objeto con propiedades formales (matemáticas) sin interesarse en la forma en que es utilizado. Estas gramáticas son muy complejas para poder ser aprendidas a partir de principios generales (gramáticas universales) que él considera como innatas, es decir, codificadas genéticamente. Mientras que para Piaget, existe una relación entre la inteligencia sensorio-motriz y la formación del lenguaje y no está de acuerdo con Chomsky sobre los aspectos innatos que relaciona con las estructuras sujeto, predicado, relaciones, etcétera.

Si combinamos las teorías de Chomsky y Piaget, se puede admitir la existencia de estructuras genéticamente predeterminadas y la existencia de estructuras aprendidas en un entorno externo para el aprendizaje del lenguaje. Esto es, podemos decir que las ópticas de Chomsky y Piaget no son contradictorias, sino que las podemos apreciar como complementarias.

En relación al aprendizaje de un lenguaje de programación, de su filosofía o de ciertas estructuras de programación, nos damos cuenta que éste no debería ser difícil, dado que se trata de un lenguaje restringido, finito, con una estructura que existe al interior del lenguaje materno o natural, lógico, fácil de comprender y sobre todo, coherente.

### **Lenguaje natural**

En el caso de los lenguajes naturales, el lenguaje es concebido para transmitir la información de una persona a otra. A nivel de pensamiento, la información dada es recibida y representada por una red de estructuras interrelacionadas, complejas y multidimensionales. Es así que lo que se transmitirá a través del lenguaje es una gran parte de su mensaje por vía de convenciones sociales, de sobre entendidos y de la confianza que acuerda el emisor al receptor. Este último, utilizará sus medios de tratamiento de información para traducir y comprender el mensaje. Es decir, muchas más cosas son transmitidas que dichas. Los lenguajes naturales también llamados maternos, los aprendemos desde nuestra infancia. Es por ello, que aprendemos a hablar antes de escribir y formalizamos nuestros lenguajes maternos en las escuelas. Esto es, aprendemos la gramática de las lenguas naturales a lo largo de mucho tiempo y las llegamos a dominar, gracias a su práctica tanto hablada como escrita.

### **Lenguaje artificial**

Un lenguaje artificial es concebido como un conjunto finito de series donde cada una es un programa (serie de símbolos en un alfabeto determinado). El lenguaje de programación tiene una gramática que caracteriza con precisión el alfabeto y el conjunto de técnicas que permiten construir programas. Esto quiere decir que un lenguaje de programación debe poseer una gramática no ambigua. En caso contrario, el programador puede construir un programa que él espera que la computadora lo interprete de cierta manera, pero la computadora lo interpretará de otra forma totalmente distinta. Los lenguajes artificiales, han sido creados para la programación informática; la puesta en marcha de algoritmos para la solución de problemas mediante la utilización de palabras reservadas, secuenciación de instrucciones, toma de decisiones y repeticiones sobre la gramática artificial del propio lenguaje de programación. Es por ello, que un lenguaje de programación informática cuenta con un número finito de palabras reservadas, treinta o cuarenta, mismas que son utilizadas de manera secuencial y/o recursiva para solucionar los problemas planteados. En la actualidad, a estos lenguajes de programación se les llama de alto nivel y pueden y deben ser utilizados por todo mundo, para desarrollar una de las principales habilidades o capacidades del siglo XXI, esto es, saber programar desde el punto de vista informático. Es decir, solucionar un problema y expresarlo en términos sucintos, mismos que pueden ser traducidos en instrucciones o comandos de cualquier lenguaje de programación informática. Esto significa a grandes rasgos, plantear la solución del problema en pseudocódigo y traducirlo a un lenguaje de programación informática.

### **Diferencia entre lenguajes naturales y artificiales**

Los lenguajes naturales tales como el español, el inglés, el francés, el chino, etcétera, tienen una capacidad de extensión y su forma actual es el resultado de una evolución

importante. En cuanto a los lenguajes artificiales éstos, han sido creados en un momento dado y no tienen posibilidad de evolucionar como los lenguajes naturales. Al parecer, algunos autores coinciden en que se trata de volver los lenguajes naturales lo más parecido a los lenguajes de programación. Esto podría admitirse si construyeran interfaces hombre-máquina suficientemente inteligentes.

### **La solución de problemas vía lenguajes de programación informática**

La solución de problemas puede ser realizada de manera muy eficiente o no tan eficiente. Independientemente de la solución propuesta, ésta puede ser traducida en un lenguaje previo al lenguaje informático de programación. A este lenguaje previo se le llama pseudocódigo.

### **El pseudocódigo**

El pseudocódigo es la delineación de la solución del problema, utilizando el lenguaje natural pero incluyendo algunas de las convenciones estructurales propias de los lenguajes de programación informática. Esto es, por ejemplo: "Hacer el cálculo del cambio y moverlo hacia el registro 1", que está escrito en lenguaje natural, puede ser traducido en pseudocódigo como "dar cambio y *move to record* 1.

### **La expresión mínima de la solución de problemas**

Cuando se logra traducir la solución del problema en expresiones mínimas de lenguaje natural, estas son traducidas al pseudocódigo y posteriormente se buscan las instrucciones o comandos propios del lenguaje que resuelvan la tarea indicada en el pseudocódigo. El pseudocódigo es una expresión mínima de la solución del problema. De aquí a la programación informática, queda tan sólo un paso: traducir el pseudocódigo en instrucciones propias del lenguaje artificial en que se está programando.

### **Filosofía del los lenguajes de programación informática**

Cualquier lenguaje de programación informática tiene estructuras secuenciales, de decisión y de repetición. Estas estructuras pueden ser secuenciales, de decisión, repetitivas tantas veces como sean necesarias o recursivas. Así, tenemos que la solución de un problema puede ser planteada en términos de programación informática como una secuencia de instrucciones, una a una; una después de la otra. Por otra parte, la mayoría de los lenguajes de programación, cuentan con estructuras de decisión, mismas que permiten hacer más inteligente y expedita la solución al problema. Si se da esta condición, entonces realiza este procedimiento, si no se da la condición, entonces realiza otro procedimiento. Algunos problemas, tendrán instrucciones o procedimientos que requieren ser repetidos una o varias veces. Para esto, los lenguajes cuentan con estructuras de repetición. Las repeticiones pueden ser bien definidas, es decir, repetirla un número determinado de veces, o pueden repetirse tantas veces como sean necesarias hasta que se cumpla una cierta condición. Si la condición es cierta, entonces el proceso se para. Si la condición no es cierta, entonces el proceso continúa. Finalmente, algunos lenguajes de programación cuentan con estructuras recursivas, estas estructuras recursivas permiten llamar o especificar un proceso basado en su propia definición. Así pues, cualquier lenguaje de programación cuenta con estas estructuras, mismas que pueden ser utilizadas, combinadas o llamadas de distintas maneras, dependiendo de la solución propuesta al problema informático.

### **La programación informática**

Una vez contando con la expresión mínima de la solución del problema (pseudocódigo), ésta puede ser expresada en términos de un lenguaje de programación, porque el paso

se vuelve inmediato. Tenemos como expresión mínima el verbo en infinitivo, y de ahí traducirlo al comando o instrucción, se vuelve natural. De esta manera la programación informática se vuelve inmediata.

### **¿Por qué es importante desarrollar habilidades de programación informática?**

Porque permite:

- ✚ Plantear y resolver problemas de manera diferente: resolución de problemas en modo abstracto.
- ✚ El conocimiento de lenguajes artificiales.
- ✚ Tener una visión holista del problema a resolver.
- ✚ Estructurar y resolver problemas de manera algorítmica.
- ✚ Permite resolver problemas utilizando un lenguaje artificial, restringido pero poderoso.
- ✚ Modelar y formalizar problemas informáticos. Habitualmente, los profesores de informática aportan algoritmos para aplicarlos, exponen la situación, formalizan el problema y después lo resuelven.
- ✚ Memorizar ciertos elementos tales como la sintaxis del lenguaje y ciertas instrucciones, y una vez que éstas son dominadas, aplicarlas.

### **¿Por qué es importante desarrollar habilidades de programación informática vía el aprendizaje móvil (teléfono celular)?**

El aprendizaje y práctica de de los lenguajes de programación informática vía el aprendizaje móvil (su filosofía<sup>2</sup>, independientemente del lenguaje en uso), permiten entre otras cosas:

- ✚ Manipular de manera concreta, implica desde el punto de vista didáctico que las estructuras formales sobre las cuales descansa nuestra acción educativa se vuelven objetos controlables.
- ✚ Expresar con precisión e introducir lo “concreto”. Cuando el alumno hace manipulaciones, podrá jugar con las representaciones de donde extraerá más fácilmente los conceptos, concibiendo y realizando con placer las tareas de construcción de modelos simples. Lo concreto se introduce inicialmente en la manipulación de palabras, de ideas, de reflexiones previas al análisis.
- ✚ Maniobrar directamente para construir un sistema de representaciones visuales y *kinestésicas* más estructuradas, necesarias para la construcción de conocimientos informáticos, esto es, se vuelve realizable, gracias a la manipulación de objetos concretos.
- ✚ Generar varias representaciones de las variables en juego. Con la representación concreta, ellos tendrán una vista global y sintética de la realidad y de los conceptos en interacción (al generar diversas representaciones, éstas son verificadas por la situación concreta). La manipulación concreta permitirá unificar y visualizar en interacción las variables para descomponerlas de manera más precisa. Es así, que la precisión del razonamiento se aprenderá mejor y más

---

<sup>2</sup> La filosofía de un lenguaje de programación nos permite el aprendizaje de éste, independientemente del lenguaje en uso. Esto quiere decir, que en su aprendizaje se pone el acento en su sintaxis y en las estructuras que conforman los lenguajes de programación a saber: secuenciales, de decisión, de repetición y recursividad y su posible generación y combinación. Lo anterior, nos asegura que podemos programar en cualquier lenguaje de programación informática observando únicamente las definiciones, palabras reservadas y formatos propios del lenguaje en uso.

fácilmente (sí existe la manipulación concreta). Sola, la presentación intuitiva no es suficiente. Es necesario que los jóvenes estudiantes operen y manipulen por ellos mismos, esto porque nosotros creemos que es la operación manual la que prepara la operación mental. Esta organización didáctica coincide con el modelo piagetiano.

- ✚ Reorganizar, producir, constituir, transmitir, crear, modificar, proponer, planificar, proyectar, derivar, desarrollar, combinar, organizar y formular en una nueva estructura de conjuntos, elementos provenientes de distintas fuentes, gracias a la manipulación y a la experimentación. En lugar de buscar los elementos explícitos e implícitos de un mensaje o consigna o de obtener las relaciones que existen entre estos elementos.
- ✚ Presentar bajo la forma de una comunicación original, el instrumento producido o construido de un plan o estrategia, privilegiando de esta forma, la creatividad.
- ✚ Resolver problemas de su vida cotidiana.
- ✚ Partir de lo concreto hacia lo abstracto, resulta propicio para elaborar poco a poco un pensamiento formal, estructurado y algorítmico.
- ✚ Planificar directamente movimientos de objetos concretos y manipulables (robots pedagógicos).
- ✚ Organizar secuencialmente el conjunto de acciones para facilitar así la transición entre la planificación de un número de desplazamientos en una serie de instrucciones.
- ✚ Codificar las acciones, para facilitar la adquisición de un lenguaje simbólico.
- ✚ Sustituir las actividades de planificación de un conjunto de acciones concretas por la programación de un conjunto de instrucciones.

Con esta propuesta, la transición es más gradual puesto que existe un entorno concreto en donde el estudiante planifica, describe operaciones lingüísticas, comete errores, recomienza su proceso de solución del problema, ayudándose de heurísticas que le permitirán pasar a un entorno informático de una manera mucho más natural y sencilla. Es a través de esta actividad que nosotros pensamos que los estudiantes encontrarán una fuerte motivación.

Para ello, se desarrolló un programa llamado ROMPI<sup>3</sup> para ser utilizado en un teléfono celular de muy bajo costo<sup>4</sup>:  
Este programa contiene:

- a) Una función de diálogo entre el alumno y la el teléfono celular (software);
- b) Una función interactiva (control de procedimientos) entre el teléfono móvil y el dispositivo experimental vía una interfaz material (hardware); y
- c) Una función de representación gráfica construida a través del teléfono celular a partir de ejercicios (variables) experimentales que se representa en la pantalla del teléfono celular (software).

El programa es interactivo y pretende hacer la liga o puente entre el alumno y el robot, esta opción del menú termina cuando se aprehenden las nociones: programación,

---

<sup>3</sup> Ruiz-Velasco, E. Cibertrónica. Aprendiendo con tecnologías de la inteligencia en la Web semántica. UNAM-Díaz de Santos: México. <http://www.educatronica.mx/content/rompi>

<sup>4</sup> Esto quiere decir que podemos programar robots pedagógicos mediante un teléfono celular de muy bajo costo. Lo que significa que las posibilidades del aprendizaje de lenguajes de programación informática, ahora puede hacerse también con un teléfono celular, sin necesidad de contar con una computadora. Los costos económicos ser reducen enormemente y también las ventajas cognitivas.

secuenciación, instrucción, ejecución de programa, compilación, ejecución, tablero, programa, sistema de comando, control, robot, trabajo.

**ROMPI** es una aplicación para dispositivos móviles (teléfonos celulares) que permite la creación, edición, compilación y ejecución de programas escritos en lenguaje **LCS**.

A partir de este lenguaje, se creó un conjunto restringido de instrucciones y palabras reservadas, en donde:

Instrucción	Modo abreviado	Descripción
<b>Der</b>	D	Indica un giro a la derecha
<b>Izq</b>	Z	Indica un giro a la izquierda
<b>Pausa</b>	P	Indica un alto temporal

Palabra reservada	Modo abreviado	Descripción
<b>Inicio</b>	I	Indica el inicio de un programa
<b>Fin</b>	F	Indica el final de un programa

La sintaxis de Rompi no distingue entre minúsculas y mayúsculas por lo que **DER** y **Der** representan la misma instrucción.

### Estructura básica de un programa

La estructura básica de un programa está constituida por la palabra reservada **inicio** al comienzo del programa una serie de instrucciones válidas y la palabra reservada **fin** al final del programa.

Una instrucción válida para un programa tiene la siguiente forma: Instrucción Duración.

Una instrucción indica la acción a llevar a cabo.

La duración indica el tiempo en segundos que durará la instrucción, éste puede ser un número comprendido entre 1 y 10.

Un programa típico tendría una estructura parecida al siguiente programa:

#### **Inicio**

**Der 5**

**Izq 3**

**Pausa 2**

#### **Fin**

La aplicación se encuentra dividida en dos secciones:

El **Menú principal** es una pantalla en la que se muestran las principales acciones que se pueden llevar a cabo.

#### Menú principal

##### Opciones del menú

- **Abrir**
- **Nuevo**
- **Eliminar**
- **Ejecutar**
- **Ayuda**
- **Salir**



En esta pantalla se visualizarán las opciones que están disponibles en ese momento, pudiendo ser que:

- a) En caso de no haber programas guardados en la memoria del teléfono no se mostrarán las opciones: **Abrir, Ejecutar y Eliminar**
- b) En caso de no haber programas compilados no se mostrará la opción **Ejecutar**

Para salir de la aplicación seleccionar la **opción salir del menú principal**.

El **editor**: Es una pantalla que permite la captura y edición de los programas, la cual cuenta con un menú que permite realizar diferentes acciones con el programa que se encuentre abierto

Menú del editor



Pantalla de captura del editor



Opciones del **menú del editor**

- **Compilar**
- **Ejecutar**
- **Guardar**
- **Guardar Como**
- **Menú principal**
- **Salir**

Para regresar al **Menú principal** desde el editor seleccionar la **opción Menú principal** del menú, o bien para **salir de la aplicación** seleccionar la **opción Salir**.

**Abrir un archivo**

En el **menú principal** se desplegarán los programas guardados en la memoria del teléfono hasta ese momento, junto con la **opción Abrir**.

Para abrir un programa:

- i) Seleccionar la **opción Abrir**
- ii) Seleccionar el nombre del programa que se desee abrir del listado de programas para visualizarlo en el **Editor**.
- iii) En caso de no querer abrir un programa elegir la **opción Cancelar** para regresar al **Menú principal**.

Menú abrir



### Crear un programa

Para crear un programa desde el **Menú principal**, seleccionar la **opción Nuevo**, a continuación se mostrará el **Editor**, en éste se podrá capturar el contenido del nuevo programa.

Dentro del **Editor** seleccionar la **opción Nuevo** del **menú**.

#### Editor



### Eliminar un programa

En el **Menú principal** se desplegarán los programas guardados en la memoria del teléfono hasta ese momento, junto con la **opción Eliminar**.

Para eliminar un programa:

- 1) Seleccionar la **opción Eliminar**.
- 2) Seleccionar el nombre del programa que desea eliminar del listado.
- 3) En caso de seleccionar un programa se mostrará un mensaje para confirmar que se desea eliminar el programa, teniendo las siguientes dos opciones:
  - a) **Aceptar**, el programa será eliminado.
  - b) **Cancelar**, el programa no será eliminado.

Cualquiera que sea la elección la aplicación regresará a **Menú principal**.

#### Menú eliminar



#### Confirmación Eliminar



### Ejecutar un programa

Para ejecutar un programa desde el **Menú principal**:

## Menú ejecutar



- 1) Estando en el **Menú principal** seleccionar la **opción Ejecutar**
- 2) Seleccionar la **opción Ejecutar Programa**

3) Seleccionar el nombre del programa que se desea ejecutar, del listado mostrado

4) Seleccionar la **opción ejecutar** para comenzar la ejecución.

Durante la ejecución de un programa se despliega un mensaje indicando que el programa se encuentra en ejecución con las siguientes opciones:

a) **Cancelar.** Permite cancelar la ejecución del programa.

b) **Detener.** Detiene la ejecución del programa

c) **Reanudar.** Permite continuar con la ejecución del programa.

Para **ejecutar un programa desde el Editor**, dentro del programa que se encuentra abierto seleccionar la **opción Ejecutar del menú**.

## **Modificar el volumen de ejecución**

### Modificación del volumen

Para modificar el volumen de ejecución desde el **Menú principal**:

1) Seleccionar la **opción Ejecutar** y después la **opción Volumen**.

2) Dentro del control gráfico seleccionar el nivel de volumen deseado, el cual puede variar entre 0 y 100%; seleccionar la **opción Aceptar** para regresar al menú principal.



## **Ayuda**

Para visualizar la ayuda desde el **Menú principal** seleccionar la **opción Ayuda**

Se mostrará un listado con las siguientes opciones:

## Opciones del *menú del ayuda*

**Ayuda Rompi:** Muestra una descripción breve de cada una de las opciones del menú principal.

**Palabras reservadas:** Muestra un listado con las palabras reservadas que componen al lenguaje.

**Acerca Rompi:** Muestra una breve descripción de la funcionalidad del programa.

## Menú ayuda



## Guardar un programa

Para guardar un programa dentro del **Editor** según sea el caso realizar:

### Guardar un programa por primera vez:

- 1) Seleccionar la **opción Guardar** del menú del **Editor**
- 2) Introducir el nombre del programa en el cuadro de diálogo
- 3) Seleccionar la **opción Aceptar** para guardar el programa o bien seleccionar la **opción cancelar** para regresar al editor.

### Guardar un programa con otro nombre

- 1) Seleccionar la **opción Guardar Como**
- 2) Introducir el nombre del programa en el cuadro de diálogo
- 3 Seleccionar la **opción Aceptar** para guardar el programa o bien seleccionar la **opción Cancelar** para regresar al **Editor**.

### Guardar cambios realizados en un programa

- 1) Seleccionar la **opción Guardar**
- 2)

#### Confirmación guardar



#### Diálogo guardar como



## Compilar un programa

Dentro del **Editor** para compilar un programa realizar:

- 1) Seleccionar la **opción Compilar** del menú
- 2) En caso de que el programa no haya sido guardado o haya sido modificado se desplegará un mensaje indicando que se deben guardar los cambios (ver guardar un programa), posteriormente se desplegará un mensaje en el cual se indica si la compilación ha tenido éxito, en caso contrario mostrará el error que se ha encontrado.

Mensaje compilado



Mensaje de error



En cualquier caso seleccionar la **opción Aceptar** para regresar al **Editor**

### Mensajes de error al compilar un programa

En caso de que el programa compilado no respete la estructura o sintaxis de un programa en lenguaje **LCS** al compilarse se pueden mostrar los siguientes mensajes de error:

id	Mensaje	Descripción
1	Programa vacío	Cuando el programa compilado no contiene ninguna instrucción
2	Palabra desconocida	Cuando se ha detectado una palabra que no corresponde a ninguna instrucción válida
3	Se esperaba Inicio	Cuando la primera palabra que se encuentra no es Inicio
4	Fin debe ser último	Cuando la última palabra no es la palabra fin
5	Se esperaba instrucción o Fin	Cuando se ha detectado que después de una instrucción válida sigue una palabra que no es una instrucción o un número.
6	Se esperaba número entre 1 y 10	Cuando el número que indica la duración de la instrucción tiene un valor menor a 1 o mayor a 10

La dirección URL desde donde pueden descargar Rompi, se accesa mediante la siguiente liga: <http://www.educatronica.mx> Véase la figura 1.



Fig. 1 Pantalla de inicio para acceder a Rompi

Enseguida, en la figura 2, aparece en la parte media superior una ventana que dice ROMPI, como se muestra en la siguiente pantalla. Esta es la ventana que permite el acceso al lenguaje de programación.



Fig. 2 Ventana de acceso a Rompi

En la figura 3, en el ángulo superior izquierdo se muestra con letras anaranjadas, la opción de descarga: Descarga **Rompi. Zip**. Dando clic en el letrero de descarga, se descargará automáticamente en el teléfono celular el programa y estará listo para desarrollar programas en lenguaje Rompi y programar los robots pedagógicos.

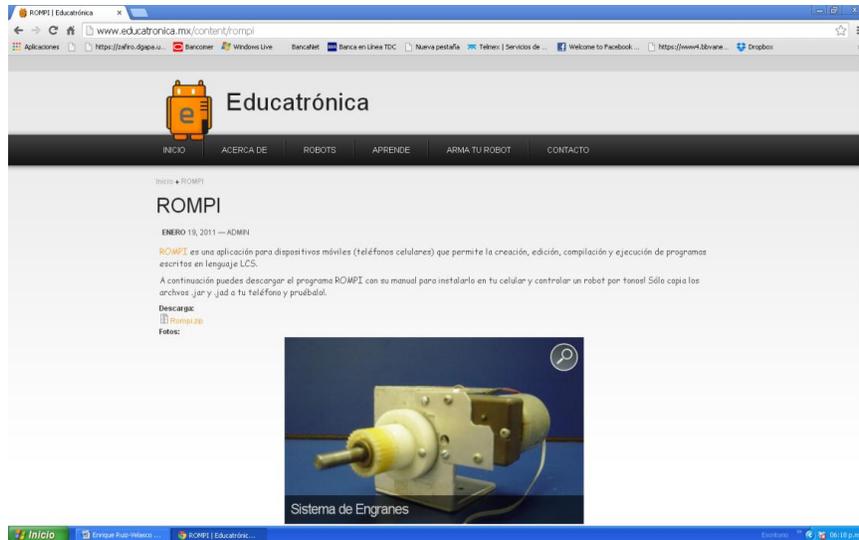


Fig. 3 Descarga de Rompi.zip en el teléfono celular

Una vez hecha la descarga correspondiente, la figura 4, muestra la página desde donde se puede seguir el diseño pedagógico para trabajar en los laboratorios cibertrónicos 3.0, independientemente de la posición geográfica, y aprender de manera colaborativa y entre pares.

<http://roboticapedagogicamovil.blogspot.mx>



Fig. 4 Situación didáctica constructorista para aprender a programar

En esta dirección, se encuentran el conjunto de instrucciones que llevaron paso a paso, a estudiantes de escuelas primarias, a trabajar en el Laboratorio Cibertrónico 3.0, instalado el 9 de noviembre de 2013, en el evento de la Noche de las Estrellas; en la explanada de Ciudad Universitaria, D.F. con el objetivo de aprender la filosofía de los lenguajes de programación informática, armando y programando un robot pedagógico con el lenguaje ROMPI con sus teléfonos celulares.

La figura 5, muestra más detalles de construcción del prototipo pedagógico “El helicóptero”.

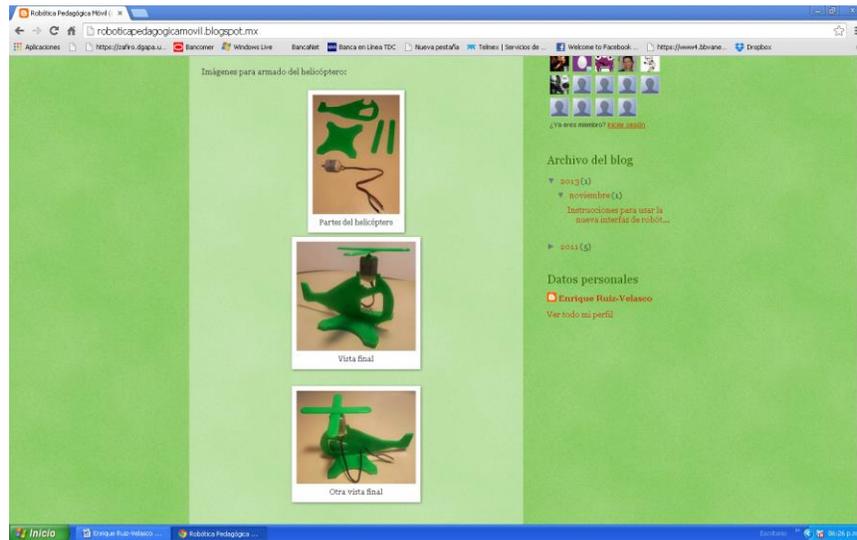


Fig. 5 Etapas de construcción del prototipo robótico pedagógico “El helicóptero”

La figura 6 muestra una pantalla conteniendo la dirección: <http://cibertronica.mx> misma que ofrece acceso a formar parte de una red de aprendizaje de robótica pedagógica móvil.

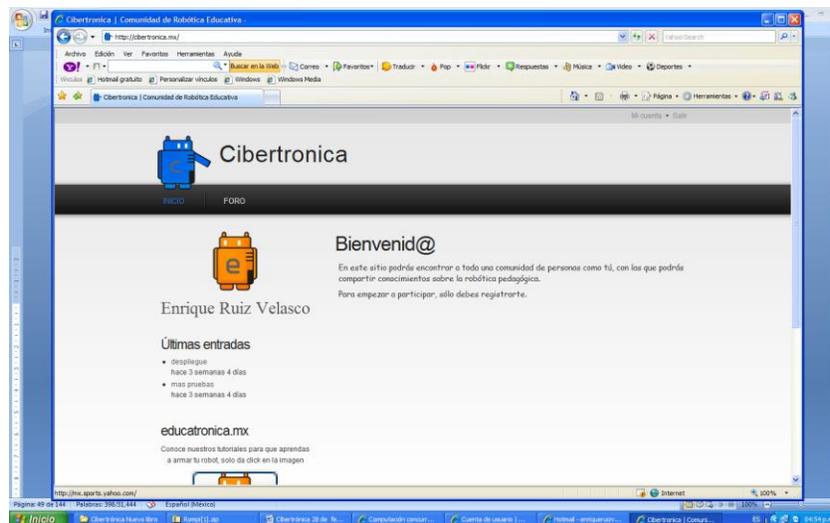


Fig. 6 Página de Cibertrónica

## Desarrollo de la experiencia

La experiencia se llevó a cabo en un laboratorio cibertrónico 3.0 Este laboratorio, forma parte de una red de laboratorios cibertrónicos 3.0, dispuestos en diversas posiciones geográficas. Un laboratorio cibertrónico 3.0, plantea que es posible desarrollar entornos ricos de aprendizaje en y desde la distancia. Este laboratorio se implantó en la

Universidad Nacional Autónoma de México, en la explanada central de Ciudad Universitaria, en el mes de noviembre de 2013. Se ofreció un curso de construcción y control de robots pedagógicos, utilizando el lenguaje ROMPI para que pudieran **controlar y programar** sus robots pedagógicos, vía sus teléfonos celulares. Los robots pedagógicos ejecutaron movimientos en función de las instrucciones dadas en el teléfono celular, programadas con Rompi.

## Discusión

Cabe mencionar que la concepción, diseño y desarrollo del lenguaje ROMPI se hizo teniendo en cuenta que los niños tienen que aprender a programar un lenguaje informático a partir de su lenguaje natural o materno, para que la transición sea más amigable. Es decir, que a partir de su lenguaje materno, ellos puedan controlar, dar órdenes y programar sus dispositivos robóticos, sin necesidad de que sean violentados por un lenguaje distinto a su lenguaje materno, tal como puede ser el inglés o el mismo lenguaje de programación informática escrito en una lengua distinta a su lengua materna.

También es importante señalar que se atendieron alrededor de 700 estudiantes de escuelas primarias de diversos lugares del Distrito Federal, Área Metropolitana, Estado de México y Oaxaca en el Laboratorio Cibertrónico 3.0. Se logró que los estudiantes programaran sus prototipos robóticos. Asimismo, las situaciones didácticas constructivistas-construccionistas que se desarrollaron, privilegiaron todo el tiempo el aprendizaje entre pares y el aprendizaje basado en proyectos. Esto es, tenían que echar a andar y programar sus proyectos de robots didácticos en equipos. Lo que dio lugar a este tipo de aprendizajes. Es evidente que para lograr aprendizajes significativos y evaluarlos, se vuelve necesario, repetir esta experiencia y sobre todo pensar en alargar los tiempos de experimentación, exploración e investigación en los laboratorios cibertrónicos 3.0.



## A manera de conclusión

Creemos que es factible desarrollar las condiciones para favorecer y privilegiar la enseñanza-aprendizaje de los lenguajes de programación informática (filosofía de la programación informática) de manera física. Es decir, trabajar con sustratos tecnológicos (robots pedagógicos) y generar las condiciones abstractas de la programación informática. Esta, es una manera más lúdica, amigable y sencilla para el aprendizaje de lenguajes de programación informática utilizando como base, el lenguaje natural (materno) de los niños.

Resulta mucho más económico en términos presupuestales y cognitivos, la utilización de un teléfono celular para hacer la programación informática de los robots pedagógicos. Con teléfonos de muy bajo costo, se puede practicar esta programación de lenguajes artificiales. Planteamos como hipótesis que el aprendizaje, dominio y utilización de lenguajes artificiales (conocimientos de base de lenguajes informáticos), facilita la comprensión tecnológica del mundo (automatización, robotización, programación, etcétera).

## **Bibliografía**

- Chomsky, N. (1967). *Linguistic Contributions to the Study of Mind*. Academic Publishing Co. Berkeley, California.
- Chomsky, N. (1985). *Sintáctica y Semántica en la Gramática Generativa*. Siglo XXI. 3era. Ed. México.
- Piaget, J. (1970). *Piaget's Theory*. En Mussen, Carmichael's Child Psychology. Wiley: New York.
- Ruiz-Velasco, E. (2007). *Educatrónica. Innovación en el aprendizaje de las ciencias y la tecnología*. Madrid: UNAM-Díaz de Santos.
- Ruiz-Velasco, E. (2012). *Cibertrónica. Aprendiendo con tecnologías de la inteligencia en la Web Semántica*. México: UNAM-Díaz de Santos.
- Skinner, B.F. (1957). *Verbal Behavior*: Appleton-Century-Crofts. New York.

## **Webgrafía**

Educatrónica

<http://www.educatronica.mx>

Cibertrónica

<http://www.cibertronica.mx>

Robótica Pedagógica Móvil

<http://roboticapedagogicamovil.blogspot.mx>