

B o l e t í n UNAMente Robótica



Año 1 N° 4 Publicación Bimestral Enero de 2013

¿Por qué enseñamos robótica en el bachillerato?

“El aprendizaje experiencial a través de la programación y el diseño de robots impulsa la formación y la vocación científica de los alumnos de nuestro bachillerato, uno de los objetivos principales del Colegio de Ciencias y Humanidades de la UNAM”, así se planteó en el Plan de Trabajo 2011-2012 de esta institución educativa.

Con dicho objetivo en mente, el Club de Robótica e Informática del CCH, CREI, ha elaborado un conjunto de estrategias de enseñanza y experiencias de aprendizaje que, mediante ejercicios de programación en Karel y “C” complementados con el diseño y la construcción de robots, pretenden provocar un aprendizaje significativo en el que nuestros alumnos adquieren nuevos conocimientos y los relacionan con los conceptos, experiencias e ideas que ya poseen. Es entonces que a través de dichas actividades de aprendizaje experiencial nuestros alumnos se enfrentan, de forma activa y constante, a nuevos retos dando solución a los problemas propuestos. Al enfrentarse a nuevas situaciones se rompe el equilibrio entre lo que saben y lo que no saben, provocando un conflicto cognitivo. De esta manera, las experiencias personales se vuelven punto focal del aprendizaje, ya que a través de un proceso de reflexión y ajuste a la nueva información, se produce un proceso de asimilación en el que los alumnos construyen, o reconstruyen, significados.

Asimismo, el aprendizaje experiencial que se genera en la programación y la construcción de robots es reforzado mediante la conformación de equipos de trabajo y la participación en eventos académicos, que se convierten en importantes fijadores del aprendizaje, pues es a través de ellos y el trabajo colaborativo que los alumnos viven nuevas situaciones, sensaciones y emociones fortaleciendo el desarrollo de sus valores y de sus competencias personales.

Alberto Ávila Ramos

Juventino Ávila Ramos

Flor Clara Cubillas Hernández

Victor Hugo Leyva García

Norberto Alejandro Pérez Colín

Profesores del Colegio de Ciencias y Humanidades, UNAM

La increíble y triste historia de la cándida experimentación básica en la FI y de su abuela la desalmada *buroacademia*.

Segunda parte

No cabe duda que lo más hermoso después de las fiestas decembrinas es que se termina la falsa ternura de buenos propósitos académicos y todo mundo recupera la añorada neurosis, feliz año nuevo chamacones.

Como lo prometimos en el artículo pasado, comentaremos en este espacio el origen y la importancia de los talleres de inicio a la tecnología y a la robótica, sus vicisitudes y anécdotas, pero también el impacto que tienen en la formación básica tecnológica de los alumnos.

Continuando con el breve contexto histórico de los escenarios académicos del Anexo de Ingeniería, se tiene que a finales del periodo de la década de los 70, la configuración académica de esta entidad prácticamente quedó consolidada. Sus laboratorios y prácticas, las asignaturas y sus programas, las ediciones de apuntes y libros, sentarían las bases para apoyar el reposado modelo pedagógico con vías a instaurarse durante los siguientes años.

Después de la década de los 80, se empieza a observar que la plantilla de profesores de tiempo completo se incrementa con la incorporación de profesores jóvenes recientemente egresados, los concursos de oposición están de moda y se contrata mucho personal administrativo. Hay que prepararse para el futuro globalizador que se vislumbra generoso.

En esta misma década de los 80 hasta entrados los 90's, también se observa que los funcionarios académicos se aferran patológicamente a sus cargos, muchos de ellos, profesores de carrera, acumulan irracionalmente sus periodos sabáticos dejando de lado la razón de ser de este derecho y a la vez obligación académica, que es la de actualización y superación. Estos aspectos, entre otros, contribuyen al inicio del relajamiento académico en la entidad. Los académicos medianamente jóvenes (de los viejos para qué les cuento), empiezan a ser invadidos por el *síndrome del vampiro académico*, es decir la pasmosa actitud inercial que consiste en que el saber del mañana iba a ser, en todo lo esencialmente académico, igual al de hoy; que el progreso en lo referente a educación básica científica y tecnológica, como decía Ortega y Gasset, sólo iba a avanzar de manera lapidaria por todos los *siempre*, sobre un camino elástico, idéntico al eternamente caminado. A la par de este problema surge otro: los nuevos programas de las asignaturas



IISUE/AHUNAM/Colección Universidad, doc. 4672.

del Anexo, al incrementar su número, pero manteniendo sus contenidos prácticamente constantes, fragmentan la continuidad conceptual de su enseñanza y resquebrajan irracionalmente su relación con otras disciplinas, lo que ocasiona graves dificultades y confusiones en la enseñanza experimental.

Esta situación, que también acontecía en otras escuelas de enseñanza superior, ya lo habían detectado tres talentosos universitarios: Arturo Azuela, Jaime Labastida y Hugo Padilla, ya que en 1979 habían publicado un librito llamado *Educación por la ciencia*, aunque esta obra de divulgación se dio a conocer algunos años después en la década de los 80. Se critica la mentada *hiperespecialización* de la formación en las áreas científicas, como un fenómeno educativo que presenta graves retrocesos a nivel básico: el principal de ellos, la separación artificial de conocimientos, aspecto que impacta de manera negativa en la formación de científicos, tecnólogos e ingenieros que se necesitan para enfrentar los elementales retos tecnológicos de nuestro país en supuestas vías de desarrollo.

En otro apartado se expone de manera sencilla por qué la enseñanza de la actividad científico-experimental, más o menos seria y racional, se debe guiar de dos formas: la primera, anteponiendo la duda en la novedad que se estudia, es decir, partiendo de lo conocido hacia lo que no se conoce; y la otra, planteando teóricamente lo posible en las extrañezas observadas, es decir partiendo de lo desconocido hacia lo que ya se conoce.



Los autores señalan que para posibilitar la actividad creadora de los alumnos, la ciencia no debe ser enseñada como una burda asimilación mecánica del conocimiento válido que resulta de la investigación científica, ya que el quehacer científico y tecnológico que de ella se deriva, se nutre de la duda, de una actitud inquisitiva, el descubrimiento, la creación de nuevos conocimientos, teorías y leyes; pero también de cierta componente inevitable de memorización. Los autores dejan ver con honesta claridad que vale la pena y es preferible formar alumnos que sean capaces de formular problemas, de establecer conexiones entre fenómenos diversos, de encontrar la similitud en la diferencia, de señalar la semejanza en la unidad, aunque también se tenga una dosis mínima y moderada de tradición medieval. Los autores aclaran que las ideas vertidas en la obra se deben considerar como parte de un proyecto eventualmente realizable.

Concluyen que la educación científico-tecnológica de nuestros alumnos debe formar parte del mundo contemporáneo; tener contacto con todos los objetos que constituyen el espacio urbano del estudiante, histórica lección, no me cabe la menor duda. A la distancia de 34 años de esta propuesta, la misma tesis resurge con mayor intensidad en el libro *Innovación en la educación superior, hacia las sociedades del conocimiento*, recientemente editado el año pasado, ahora en su versión femenina, por tres talentosas universitarias: Rosaura Ruiz, Rina Martínez y Liliana Valladares. La obra contiene propuestas muy audaces e inteligentes acordes con la tecnología prevaeciente: El teléfono celular, el uso de la Internet, la nube, la educación virtual, entre otros.

Por desgracia no es posible ampliar la discusión de este tema en este espacio,

sin embargo, es un secreto a voces, que nuestra voluntad luce relajada para contribuir a que nuestro país se integre a las sociedades del conocimiento, por la sencilla y pesimista razón de que la formación de nuestros jóvenes, que va desde nivel maternal lactante hasta doctorado, es tan banal y modesta (¿?) que nos impide insertarnos de sopetón y por dictado en los ajuares tecnológicos del primer mundo.

Una posibilidad para dar cabida a la voluntad y al optimismo, necesarios para potenciar la formación temprana de actitudes y hábitos hacia la formación tecnológica, está en poner en marcha a los parientes incómodos de los laboratorios, es decir, apostar a la opción pedagógica de los talleres libres de inicio a la actividad tecnológica.

El primero de estos talleres en el Anexo, llamado Taller de Modelismo, asociado a la desaparecida asignatura Introducción a la Ingeniería, tuvo sus momentos brillantes hace más de 35 años; las características de estos espacios de enseñanza las explicaré a partir de una anécdota entrañable y posteriormente comentaré las diferencias que tiene con un laboratorio. En el siguiente número les contaré sus pasajes sin cortes ni censuras.

Hugo Serrano Miranda
Profesor de carrera de la DCB

Robótica y Android: descripción de la plataforma Android y de su utilización para crear aplicaciones de robótica

Es posible considerar a los teléfonos inteligentes como un conjunto de sensores. Dichos sensores pueden ser integrados en un robot, el cual será capaz de reconocer su ambiente (utilizando la cámara), comunicarse con otros robots (mediante comunicación Wi-Fi o Bluetooth) e incluso conocer su posición en el espacio (utilizando el sensor GPS y los acelerómetros).



Figura 1 Tanque controlado por un teléfono inteligente.

Imagen obtenida de <http://es.paperblog.com/android-impulsando-a-un-robot-tanque-642492/>

Los teléfonos inteligentes pueden ser utilizados como computadoras a bordo para robots pequeños. Además, son dispositivos que contienen gran cantidad de sensores y pueden ser utilizados como tales. Android es una plataforma de software especialmente diseñada para dispositivos móviles. La plataforma Android incluye, entre otras cosas: un sistema operativo basado en Linux, aplicaciones predeterminadas para ejecutar las tareas más comunes, bibliotecas de código para desarrollo de nuevas aplicaciones, y por supuesto, la funcionalidad de un teléfono. A continuación se hará una breve descripción de los componentes de la plataforma Android y de los aspectos a considerar al momento de desarrollar aplicaciones para robótica.

Los elementos en la plataforma Android

El sistema operativo Android gestiona, entre otras cosas: los procesos que controlan el hardware del dispositivo; los servicios encargados de realizar las tareas más comunes de un sistema operativo, el manejo de la memoria y el manejo del sistema de archivos. Por lo tanto, el sistema operativo es el encargado de comunicarse directamente con todos los sensores y tarjetas de comunicación del dispositivo. La mayoría de los componentes del sistema operativo están programados en C y C++. Sin embargo, las aplicaciones para los usuarios son desarrolladas utilizando el lenguaje Java mediante el kit de desarrollo de software de Android (SDK por sus siglas en inglés).

El kit de desarrollo de software de Android. El SDK provee todas las herramientas y bibliotecas necesarias para desarrollar aplicaciones Android. Para el caso de desarrollo de aplicaciones para robótica es posible mencionar las siguientes:

- 1) bibliotecas para manejo de bases de datos (diversos algoritmos de programación trabajan con gran cantidad de datos, por ejemplo, el reconocimiento de rostros, en donde éstos son caracterizados mediante un vector);
- 2) bibliotecas para manejo de gráficos 2D y 3D (es posible desarrollar aplicaciones utilizando OpenGL para aplicaciones de visión artificial o realidad aumentada, por mencionar algunas);
- 3) bibliotecas para manejo de audio y vídeo (es posible implementar algoritmos de visión por computadora, así como algoritmos de reconocimiento y síntesis de voz);
- 4) bibliotecas para controlar los sensores del dispositivo (es posible desarrollar aplicaciones capaces de controlar cada uno de los sensores de los dispositivos: cámara, acelerómetros, etc.);
- 5) bibliotecas para comunicación inalámbrica (es posible dotar a las aplicaciones con capacidad de comunicación Wi-Fi, Bluetooth, 3G, SMS, o incluso aplicaciones que realicen llamadas telefónicas).

El entorno de ejecución de Android. Las aplicaciones en Android desarrolladas con el SDK, no son ejecutadas directamente por el sistema operativo, sino que son ejecutadas mediante la Máquina Virtual Dalvik, la cual es una versión

optimizada de la máquina virtual de Java. El que las aplicaciones sean ejecutadas por la máquina virtual es un punto a considerar al momento de crear aplicaciones para robótica. Una aplicación Java tiene menor rendimiento que una aplicación desarrollada en C o C++. Sin embargo, es posible desarrollar aplicaciones nativas en Android mediante el kit de desarrollo nativo de Android (NDK por sus siglas en inglés), aunque, al hacer esto último, el desarrollador sacrifica todas las ventajas de utilizar el SDK.

La capa de servicios. El sistema provee procesos que se encargan de gestionar: el sistema de ventanas, las aplicaciones que están siendo ejecutadas, los recursos del dispositivo, las funciones de telefonía, los servicios de localización, etc. Es importante entender el funcionamiento de la capa de servicios de Android. Muchos servicios son ejecutados por el sistema de manera automática, cada uno de ellos consumen memoria y procesamiento del dispositivo. Esta es una desventaja evidente puesto que dichos procesos, en ocasiones no necesarios, son ejecutados automáticamente por el sistema disminuyendo la memoria y el procesamiento para nuestras aplicaciones.

Android en robots, ventajas y desventajas

Ventajas. 1) Los dispositivos Android tienen integrados gran cantidad de sensores útiles para robótica. Es posible dotar al robot de las siguientes habilidades: comunicación inalámbrica (Wi-Fi, Bluetooth, SMS, Red 3G), localización (acelerómetros, sensores GPS, giroscopios), visión (utilizando la cámara del dispositivo), audio (utilizando el micrófono y las bocinas del dispositivo). 2) El lenguaje para el desarrollo de aplicaciones es Java. La curva de aprendizaje de Java es baja, comparada con otras como la de C y C++. Además, ofrece los beneficios de la programación orientada a objetos y del manejo automático de la memoria. 3) Las dimensiones de un móvil y las capacidades que éstos ofrecen es un punto a favor. Existen diversos dispositivos utilizando Android, cada uno con especificaciones técnicas diferentes. Es posible conseguir un dispositivo según las operaciones computacionales que la aplicación demande. 4) Los dispositivos Android pueden comunicarse con otros dispositivos (gracias a su capacidad de comunicación inalámbrica es posible establecer comunicación con otros elementos tales como computadoras, otros celulares, o incluso con tarjetas de hardware como el Arduino).

Desventajas. 1) A pesar de que en los últimos años la tecnología de los móviles ha avanzado continuamente, es importante recordar que la capacidad de memoria y de procesamiento de un dispositivo móvil es menor a la de una computadora portátil o de escritorio. 2) La capacidad de la batería es limitada, el nivel se reduce considerablemente cuando el celular se encuentra ejecutando procesos computacionales costosos. 3) Es muy común que la memoria de los

dispositivos se encuentre en tarjetas externas, este hecho implica aumento en el tiempo de lectura y escritura de los datos. 4) Es importante recordar que las aplicaciones en Android son ejecutadas por la Máquina Virtual Dalvik. El desempeño de estas aplicaciones es menor que el de una aplicación creada con C o C++. Sin embargo, como se indicó anteriormente, es posible utilizar el NDK. Terminaremos el presente artículo, no sin antes mencionar el gran potencial que ofrecen los teléfonos inteligentes Android: podemos pensar que estos dispositivos son como pequeñas computadoras con sensores integrados, justo lo necesario para un robot.

Luis Mateos Ortiz

Alumno de Ingeniería Mecatrónica

Arduino y Lego, universos complementarios en la Robótica.

Segunda Parte.

El código de programación para trabajar el NXShield de Arduino con Lego, será el siguiente para que un sensor IRSeeker de HT siga la pelota:

```
/*
Este programa lee el sensor IR de HighTechnic por medio del NXShield
*/
//Las bibliotecas que se ocupan son las siguientes
#include <Wire.h>
#include <NXShield.h>
#include <NXTLight.h>
#include <SumoEyes.h>
#include <ACCLNx.h>
#include <NXTUS.h>
#include <HTIRS_driver.h>
//Se declara el NXShield conectado a Arduino
NXShield nxshield;
// Se declara el Seeker
ClassHTIRS HTIRSeeker;
// Esta cadena contendrá los mensajes a la consola
char strseek[256];
void setup()
{
  char str[256];
  Serial.begin(115200);
  delay(500);
  Serial.println ("Initializing the devices ...");
  nxshield.init( SH_SoftwareI2C );
  nxshield.init( SH_HardwareI2C );
  HTIRSeeker.init( &nxshield, SH_BAS1 );
//Espera a que se presione el botón GO
  Serial.println ("Press GO button to continue");
  nxshield.waitForButtonPress(BTN_GO);
//Se reinician los motores
  nxshield.bank_a.motorReset();
  nxshield.bank_b.motorReset();
}
```

```
// Estas variables almacenan los valores de los sensores
byte Direction, s1, s2, s3, s4, s5;
void loop()
{
  char aa[80];
  int a;
  char str[256];
  boolean last_status;
  while (true) {
    int ba_v1, ba_v2, bb_v1, bb_v2, bb_us;
    ba_v1 = nxshield.bank_a.sensorReadRaw(SH_S1);
    ba_v2 = nxshield.bank_a.sensorReadRaw(SH_S2);
    bb_v1 = nxshield.bank_b.sensorReadRaw(SH_S1);
    bb_v2 = nxshield.bank_b.sensorReadRaw(SH_S2);
    //Se lee la dirección donde está la pelota y se despliega en la pantalla
    Direction = HTIRSeeker.ReadDirection();
    sprintf (strseek, "HiTechnic IR Seeker ReadDirection: %3.d", Direction);
    Serial.print(strseek);
    //Se leen los valores de los sensores de IR y se despliegan en la pantalla (pero no se
    utilizan en este programa)
    Direction = HTIRSeeker.ReadStrength(s1, s2, s3, s4, s5);
    sprintf (strseek, " ReadStrength Direction: %3.d Strengths S1: %3.d S2: %3.d
    S3: %3.d S4: %3.d S5: %3.d", Direction, s1, s2, s3, s4, s5);
    Serial.println(strseek);
    //Si la pelota está enfrente, el robot camina hacia adelante
    if ( Direction == 5 ) {
      Serial.println( "La pelota está enfrente del robot" );
      nxshield.bank_b.motorRunUnlimited(SH_Motor_1, SH_Direction_Forward, 100);
      nxshield.bank_a.motorRunUnlimited(SH_Motor_1, SH_Direction_Forward, 100);
      delay(10);
    }
    //Si la pelota está a la derecha el robot gira hacia ese lado
    if ( Direction > 5 ) {
      Serial.println( "La pelota está a la derecha" );
      nxshield.bank_b.motorRunUnlimited(SH_Motor_1, SH_Direction_Forward, 50);
      nxshield.bank_a.motorRunUnlimited(SH_Motor_1, SH_Direction_Forward, -10);
      delay(10);
    }
    //Si la pelota está a la izquierda el robot gira hacia ese lado
    if ( Direction < 5 ) {
      Serial.println( "La pelota está a la izquierda" );
      nxshield.bank_b.motorRunUnlimited(SH_Motor_1, SH_Direction_Forward, -10);
      nxshield.bank_a.motorRunUnlimited(SH_Motor_1, SH_Direction_Forward, 50);
      delay(10);
    }
  }
}
```

*Norma A. González Sandoval y Sergio López Luna
Profesores de la Escuela Nacional Preparatoria.*