
UNIVERSIDAD AUTÓNOMA
"JUAN MISAEL SARACHO"

DiS

DEPARTAMENTO DE
INFORMÁTICA Y SISTEMAS

# bit@bit

Vol. 2 Nº 4 DICIEMBRE 2017
ISSN 2519 - 741X

# Revista Científica del Departamento de Informática y Sistemas

TARIJA - BOLIVIA

# Revista Científica del Departamento de Informática y Sistemas


## CONSEJO EDITORIAL



**Editor: M.Sc. Lic. Octavio Douglas Aguilar Mallea**
Director del Departamento de Informática y Sistemas
Universidad Autónoma "Juan Misael Saracho"
octavioa111@gmail.com

**UNIVERSIDAD AUTÓNOMA JUAN MISAEL SARACHO**

**bit@bit**

Revista Científica del Departamento de Informática y Sistemas
Diciembre, 2017



**Edición**

Departamento de Informática y Sistemas

**Editor**

M.Sc. Lic. Octavio Douglas Aguilar Mallea
Correo electrónico: octavioa111@gmail.com



**Diseño y Diagramación:**

Teófilo Copa Fernández

**Versión Web:** www.uajms.edu.bo/revistas/bitabit

**Impresión:**

Publicación financiada por el proyecto **"Fortalecimiento de la Difusión y Publicación de Revistas Científicas en la Universidad Autónoma Juan Misael Saracho"**

# EDITORIAL

Con gran satisfacción presentamos a ustedes un nuevo número de la revista “bit@bit”, el cual contiene artículos de investigación científica y de revisión. Las temáticas de nuestros artículos son el resultado de la investigación de los docentes y estudiantes de la carrera de Ing. Informática y reflejan la actividad intelectual del Dpto. de Informática y Sistemas. Como podrán apreciar, las áreas temáticas son diversas y relacionadas con las áreas de conocimiento de nuestro departamento y currícula. Nuestra intención es contribuir al desarrollo y a la educación potenciando nuestra área y por ende al desarrollo de la sociedad y a la investigación. Esta revista se constituye en uno de los principales ejes de nuestro compromiso con la calidad. Su aceptación en la comunidad universitaria y sobre todo en los estudiantes y titulados de nuestra carrera nos motiva a continuar y mejorar día a día las publicaciones de “bit@bit”.

Dedicamos este número a la Carrera de Ing. Informática en su XXVII aniversario, consolidando nuestro compromiso con la ciencia y la docencia. La práctica de comunicar los resultados de la investigación nos fortalece y permite una mejora continua en la academia. Esperamos que estos conocimientos lleguen al aula y sean de provecho a la comunidad académica.

No ha sido fácil incorporar la cultura de publicar, por esto, cada entrega de “bit@bit”, nos llena de satisfacción y se constituye en un desafío cumplido.

Agradecemos a los autores de los distintos artículos publicados, valoramos su dedicación y esfuerzo e invitamos a que cada vez seamos más.

M.Sc. Ing. Silvana Paz Ramírez
**COMITÉ EDITORIAL REVISTA “bit@bit”**
**DOCENTE INFORMÁTICA Y SISTEMAS**

# CONTENIDO

# ARTÍCULOS CIENTÍFICOS

# CARACTERÍSTICAS DE LAS PLACAS ARDUINO

**Céspedes Machicao Marcelo**
Universidad Autónoma Juan Misael Saracho
Tarija, Bolivia

**Correo electrónico:** marcelocespedes@yahoo.com

## RESUMEN

Arduino es una placa de hardware open-source, que ha tenido una gran difusión desde su lanzamiento en 2005, su éxito y masificación de uso se debe, en parte, a su bajo costo, facilidad de uso, amplia documentación y gran asequibilidad; en este artículo se describen las características técnicas, desde un punto de vista más accesible al usuario común, sin ingresar en los detalles de las especificaciones técnicas formales.

## PALABRAS CLAVE

Arduino, hardware, microcontroladores, código abierto.

## INTRODUCCIÓN

Arduino es un proyecto de hardware y software que surgió en Italia en 2005 [Hughes, 2016,1] y ha tenido un gran impacto en la comunidad tecnológica debido a que es un proyecto de código abierto, de bajo costo y de múltiples posibilidades de aplicación.

La comunidad que usa esta tecnología y apoya a este proyecto ha tenido un crecimiento exponencial y comprende a personas prácticamente de todas las edades.

Gran parte del éxito de esta tecnología se debe a las extensiones de hardware que se fueron añadiendo a la placa Arduino, como variedad de sensores, actuadores, placas o tarjetas extras (shields), dispositivos de comunicación, conectividad a la internet mediante protocolos estándar y el desarrollo de diferentes placas Arduino con características reducidas y avanzadas, que satisfacen la mayoría de las necesidades y requerimientos para la diversidad de proyectos donde se aplican.

Otro aspecto fundamental de Arduino, es la existencia de extensa documentación técnica disponible en la Internet, y cientos de manuales, revistas y libros disponibles en formato físico y electrónico.

Sin embargo, se ha observado, que la gran documentación existente trata de la descripción de proyectos concretos y particularmente los aspectos relativos a la programación, mientras que la documentación de especificaciones técnicas está más dirigida a personal especializado y profesionales del área; por ello, en este artículo, se realiza una síntesis de las características técnicas, sin entrar en los formalismos de ingeniería.

## EL HARDWARE ARDUINO

Una placa Arduino está construida en base a un microcontrolador, denominado AVR, del fabricante americano ATMEL, que es una versión muy reducida de un microprocesador programable, y que contiene todos los elementos esenciales de una computadora, exceptuando los dispositivos periféricos, como disco duro, tarjeta de video, teclado, etc.

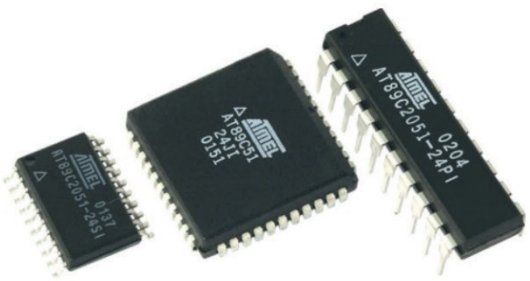


***Figura 1:*** *Microcontroladores ATMEL*

El microcontrolador es el elemento central de la placa Arduino, que le proporciona su carácter altamente versátil, de forma similar al que le otorga un microprocesador a una computadora.

Entre las partes más importantes del microcontrolador se tiene:

- Un procesador programable que contiene una unidad lógica aritmética (ALU) y los registros necesarios para la ejecución de las operaciones, que soporta un conjunto de instrucciones reducido, optimizado y de alto rendimiento.
- Memoria flash (no volátil), para almacenar los programas del usuario.
- Memoria RAM para los datos del usuario.
- Memoria ROM para datos persistentes.
- Puertos de entradas/salidas digitales.
- Puertos de entrada analógicos.
- Salida analógica PWM.
- Temporizadores internos.
- Comunicación serial, I2C y SPI,
- Estado de bajo consumo.

El microcontrolador ejecuta las operaciones en sincronismo con una señal binaria de clock o reloj, a la velocidad de 8 a 32 MHz, dependiendo del modelo, que le provee un cristal de cuarzo.

La placa Arduino contiene diversos componentes para convertirla en una tarjeta autónoma y completamente funcional, como:

1. Puerto USB.
2. Terminales digitales de entrada/salida
3. Terminales para entrada de señales analógicas.
4. Botón RESET de reiniciación.
5. Conector de alimentación con regulador de voltaje.
6. Terminales de alimentación de energía para dispositivos externos.
7. Leds indicadores de transmisión de datos
8. Led indicador de encendido
9. Microcontrolador

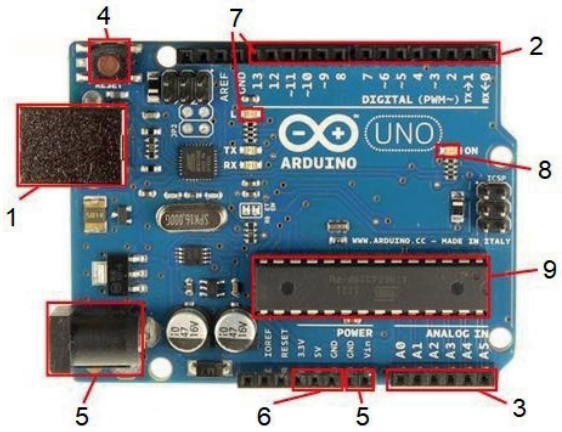


***Figura 2:*** *Tarjeta Arduino UNO*

## PUERTO USB

El puerto USB, cuenta con un conector estándar de tipo B o micro B, según el modelo de placa, para la transmisión serial de datos de forma bidireccional.

Permite la conexión de la placa a una computadora para la transferencia del programa compilado, lo que facilita su programación.

Cuenta con las terminales de energía de 5 V, lo que posibilita la alimentación independiente de la placa por el puerto USB, por lo que no es necesario otra fuente de alimentación.

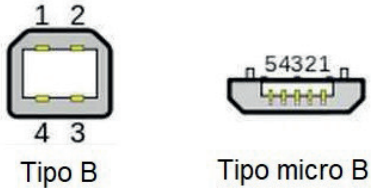


***Figura 3:*** *Conectores USB*

## TERMINALES DIGITALES

Arduino dispone de un conjunto de terminales digitales de entrada/salida para la conexión de cualquier dispositivo externo compatible como: sensores, motores, pantallas, teclados, tarjetas de expansión como microSD, reloj de tiempo real, tarjetas de red, Bluetooth, GSM, etc. El número de dispositivos externos digitales, compatibles con

Arduino es cada vez mayor

Las terminales se numeran a partir de 0 y cada una es configurable mediante software, de manera individual, como de entrada o de salida de datos, constituyendo el medio más empleado para la conexión de dispositivos externos.

Las terminales numeradas como 0 y 1 tienen predeterminadas la función de transmisión serial, sin embargo, también pueden ser reconfiguradas para otros usos, al igual que las demás terminales digitales.

Si en una terminal digital se dispone un 1 lógico, ello se traduce físicamente en una tensión o voltaje de aproximadamente 5V; por el contrario, si la terminal se presenta 0 lógico, ello equivale a una tensión o voltaje de aproximadamente 0V. Esta situación es válida tanto para la entrada de datos como para la salida.

Existe un límite máximo en la corriente que cada terminal puede suministrar al exterior cuando se configura como salida, de alrededor de 40mA, la cual no debe excederse; además se recomienda que la salida de corriente de todas las terminales, en conjunto, no exceda de 200 mA aproximadamente, aunque ello depende del modelo de placa.

## MODULACIÓN POR ANCHO DE PULSO

Algunas terminales digitales tienen una marca o indicación junto a su numeración, lo que indica que, cuando son configuradas como de salida, soportan la modulación por ancho de pulso (PWM).

La modulación por ancho de pulso es una técnica que consiste en controlar el ancho del pulso de una señal digital de alta frecuencia, para obtener un valor promedio equivalente proporcional a la relación de los tiempos de duración del pulso alto entre la duración del pulso bajo, para cada ciclo de la señal digital.

La figura 4, muestra que cuando el ancho de pulso es alto, por ejemplo 90%, el nivel de tensión promedio es de 4,5 V; cuando es de 50% (denominado duty cycle) valor medio que se obtiene es 2,5V y cuando es del 10%, el valor promedio de la tensión es de 0,5 V.

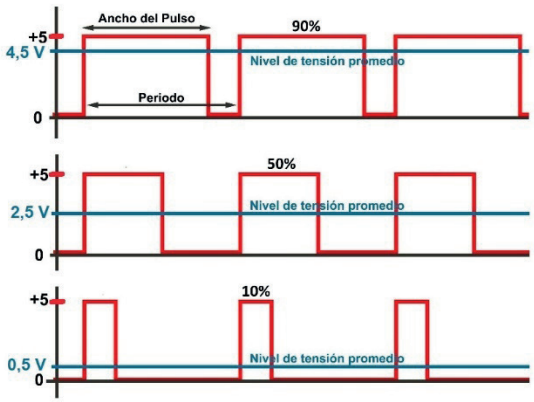


***Figura 4:*** *Modulación por ancho de pulso*

El ancho de pulso puede ser regulado mediante instrucciones de software con muy alta precisión, para obtener cualquier valor comprendido entre 0 y 5 V, lo cual equivale a un conversor de señal digital a señal analógica.

## ENTRADAS ANALÓGICAS

Una característica de Arduino es el soporte que brinda para la adquisición de datos analógicos desde el exterior, para lo que dispone de un conversor analógico digital (CAD) que convierte las señales analógicas en valores binarios equivalentes, para poder procesarlos, transmitirlos o almacenarlos en forma digital, de forma similar a lo que se realiza con un archivo digital de audio o video, como MP3 o MP4.

El CAD incorporado en la placa Arduino tiene una resolución de 10 bits, lo que determina la subdivisión del rango de voltajes de 0 a 5 V en un conjunto de 2^10 = 1024 valores pudiendo discriminarse niveles de voltaje que se diferencian en 5/1024=0,0049 V, que lo hace conveniente y propicio para una variedad de aplicaciones.

El rango de voltajes que se digitaliza es configurable por software, tanto en su valor máximo (no superior a 5 V) y en su valor mínimo.
Arduino dispone también la entrada denominada Aref (referencia analógica), que permite establecer un nivel de referencia externa, la cual servirá de base para la conversión.

Los valores digitales que se obtienen del proceso de conversión se almacenan en variables estándar o arreglos dentro de la placa.

## RESET

La placa Arduino dispone de un botón, tipo pulsador, para el reinicio de la misma, lo que provoca la ejecución del programa almacenado desde el principio.

## LEDs INDICADORES

La placa contiene cuatro LEDs indicadores incorporados en la misma:

- Power ON, indicador de encendido
- TX, transmisión serial
- RX, recepción serial
- LED13, conectado en la salida digital 13.

## COMUNICACIÓN SERIAL

La comunicación serial entre dispositivos es la más empleada actualmente en los sistemas de transmisión de datos digitales y es el sistema básico de comunicación de las placas Arduino. Está soportado por uno o más dispositivos UART o transmisor/receptor asincrónico serial, mediante los cuales se transmite cada byte de manera independiente de los demás y en cualquier instante.

El UART da el soporte básico para la comunicación por el puerto USB y para las terminales digitales denominadas Tx y Rx.

## COMUNICACIÓN SPI

Arduino incluye el sistema de comunicación denominado SPI (Interface serial para periféricos), para la transferencia de datos, full dúplex (bidireccional simultáneo) de alta velocidad (hasta 10 MB) y distancias cortas (hasta 30 cm), entre diferentes dispositivos de hardware, como memorias, sensores, conversores, otras placas Arduino, etc., bajo el esquema: maestro/esclavo.

El dispositivo maestro inicia la comunicación, habilitando a un dispositivo esclavo, pudiendo intercambiar datos de forma sincrónica con el mismo.

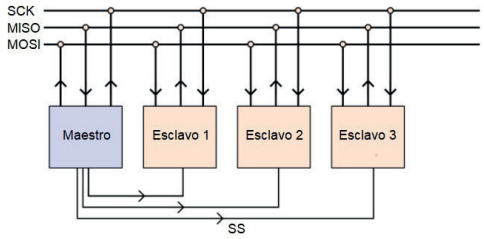


***Figura 5:*** *Comunicación SPI*

El estándar utiliza cuatro cables, denominados:

1. MOSI, master out, slave in, transmisión del maestro al esclavo.
2. MISO, master in, slave out, transmisión del esclavo al maestro.
3. SCK, señal de clock de sincronización enviado por el maestro
4. SS, slave select, permite que el dispositivo maestro habilite a un esclavo, uno a la vez.

## COMUNICACIÓN I²C

Arduino incorpora otro sistema de comunicación para el intercambio de datos, basado en el protocolo I2C que se caracteriza por emplear solo dos líneas denominadas SDA (serial data) y SCL (serial clock), y al igual que el protocolo SPI, usa un esquema maestro/esclavo, una velocidad de hasta 5 MB y un alcance de hasta 30 cm.

A diferencia del protocolo SPI, el I2C permite que cualquier dispositivo conectado sea maestro, lo que se define mediante un sistema de arbitraje, otorgando ese privilegio a un dispositivo a la vez. La transferencia de datos siempre lo inicia el dispositivo maestro, habilitando al esclavo específico mediante una dirección de 7 bits, lo que permite administrar hasta 128 dispositivos esclavos.

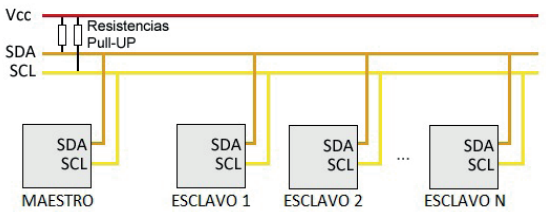


***Figura 6:*** *Comunicación I2C*

## FUENTE DE ALIMENTACIÓN

Los componentes electrónicos de la placa Arduino operan con 5 V y/o 3.3 V, dependiendo del modelo; estas tensiones se consiguen mediante uno o dos reguladores de tensión incorporados en la placa, en consecuencia, la alimentación de energía externa debe tener un voltaje mayor.

Arduino puede ser alimentada a través de tres medios distintos:

Fuente externa de corriente continua, con un valor comprendido entre 7 a 12 V CC, mediante un conector Jack, con la polaridad indicada en la figura 7. La fuente de suministro puede ser un adaptador universal de un mínimo de 500 mA, una batería de 9V, un juego de 4 pilas AA conectadas en serie o cualquier batería recargable que proporciones un voltaje en el rango establecido.


***Figura 7:*** *Conector Jack de alimentación*

Un conector USB con alimentación de 5 V de cc, proporcionado por una computadora o un cargador de teléfono móvil.

Voltaje entre 7 y 9V CC, directamente aplicado a la entrada Vin de la placa y negativo a la terminal GND. (https://www.arduino.cc/en/Reference/Board)

## SHIELDS

Una shield, es una placa de expansión que puede conectarse a la placa Arduino para añadirle funcionalidades extra.

Las placas shield están diseñadas para ser montadas directamente sobre la placa Arduino aprovechando las terminales de la misma y proveyendo nuevas terminales para otros dispositivos.

A la fecha, existe más de un centenar de shields, cuya descripción y conexión puede encontrarse en: http://shieldlist.org/

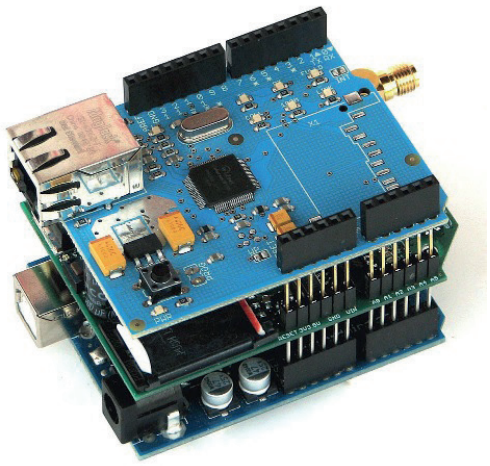

***Figura 8:*** *Montaje de placas Shield*

## SOFTWARE

El software del proyecto Arduino es open-source, desarrollado en Processing (https://processing.org/) y Java; su código fuente está disponible para su descarga en un repositorio de GitHub accesible desde el sitio oficial: (https://www.arduino.cc/en/Main/Software)

El software es un IDE (entorno de desarrollo integrado) libre, con funciones de edición, compilación y transferencia de programas a la placa, disponible en su versión 1.8.5, en el sitio indicado, disponible y actualizado al 13/10/17, para los sistemas operativos Windows, Linux y Mac, para 32 y 64 bits y ahora para Android.

El mismo sitio ofrece un editor Web, permanentemente actualizado con las últimas bibliotecas de los desarrolladores y con almacenamiento de programas en la nube o Internet.

El lenguaje de programación de la placa está basado en C++ estándar, y el compilador denominado avr-gcc,, y las bibliotecas de funciones mínimas necesarias.

El software de Arduino llega con sus propias bibliotecas, algunas de las cuales son:

- Serial, para lectura y escritura por el puerto serie.
- Servo, para controlar servomotores.
- Stepper, para controlar motores paso a paso.
- Wire, para transmisión y recepción de datos I2C.
- LiquidCristal, soporte para pantallas LCD.

La documentación del lenguaje se encuentra en: http://arduino.cc/en/Reference/HomePage

Diversos desarrolladores de hardware y software ponen a disposición bibliotecas para que algunos entornos, plataformas o diversos lenguajes de programación, interactúen con un programa que se ejecute en la placa Arduino, mediante una comunicación serial de datos. Entre los lenguajes que brindan ese soporte se encuentran:

C, C++, C#, Objetive-C, Java, LabView, Mathematica, Matlab, PHP, Physical Etoys, Processing, Pure Data, Python, Ruby, Scratch, VBScript, Visual Basic .NET (http://www.arduino.cc/playground/Main/InterfacingWithSoftware)

Existen además diversas herramientas de desarrollo que dan soporte para la conectividad con Arduino, entre las más destacadas se encuentran:

- PlatformIO IDE
- Visual Studio Code Extension for Arduino
- Aduino for Visual Studio
- Programino IDE
- Deviot, para Sublime Text
- Sloeber, para Eclipse
- Eclipse AVR
- Biicode
- Pluto, para Python
- Embrio
- Zerynth
- Arduino for Atmel Studio (https://playground.arduino.cc/Main/DevelopmentTools)

## CONCLUSIONES

El uso de una placa Arduino requiere un mínimo de conocimientos técnicos básicos, tanto de electrónica, como de programación; una vez superada esta etapa, su uso es simple, pudiéndose abordar proyectos de nivel básico e intermedio. Para encarar proyectos avanzados o más especializados, es necesario considerar otros aspectos técnicos a detalle.

La gran versatilidad de las placas Arduino, permiten aplicarlas en multitud de proyectos en diferentes áreas, por ejemplo: robótica móvil, robótica industrial, domótica, sistemas de vigilancia y seguridad, adquisición de datos, sistemas de control automáticos, inteligencia artificial, publicidad, tableros de control, etc., etc.; estando los campos de aplicación limitados solo por la creatividad de los usuarios.

Las posibilidades de aplicación de las placas Arduino crecen exponencialmente con la gran variedad de sensores, actuadores y placas shields existentes y la gran variedad de recursos de software de programación, plataformas y entornos de desarrollo disponibles.

## REFERENCIAS

Hughes J. M. (2016) Arduino: A Technical Reference. USA: O'Reilly. 1ra Edición. 613 págs.

Evans M., Noble J., Hochenbaum J. (2013) Arduino in Action, Ed. Manning Publications Co.

Wheat D., (2011) Arduino Internals, New York, ed. Apress

OxerJ, Blemings H. (2009) Practical Arduino, Cool projects for open source hardware. New York, ed. Apress.

Página Web oficial de Arduino: https://www.arduino.cc/ [consulta: 17/10/17].

# DESARROLLO DE APLICACIONES MÓVILES HIBRIDAS CON IONIC

**Huanca Churata Luis Fernando**
Universidad Autónoma Juan Misael Saracho
Tarija, Bolivia

**Correo electrónico:** *luishuancafernando@gmail.com*

## Resumen

Este presente artículo explicará lo que es IONIC Framework, algunos módulos con los que trabaja, la ventaja y beneficios de utilizarlo. Además se verá sus principales características, su instalación y herramientas que se necesita para funcionar.

IONIC es un framework gratuito y de open source que sirve de herramienta para el desarrollo de aplicaciones móviles híbridas basadas en HTML5, CSS, JS y optimizado con AngularJS.

"Las aplicaciones son híbridas, ¿Qué quiere decir eso? Que puedes desarrollar una misma aplicación y ejecutarla en Android, iOS y Windows Phone sin tener que desarrollarla en el correspondiente lenguaje nativo de cada plataforma". (Tiago Coelho, Marzo 2016, Desarrollo P.2).

## PALABRAS CLAVE

Framework, S.O.Apps, hibrida, Android, IOS, nativas, open source, plugins, cordova.

## 1. INTRODUCCIÓN

Desde la aparición de celulares inteligentes, el desarrollo de aplicaciones para móviles ha crecido y se convirtió una de las obsesiones de muchos de los programadores, además en el mercado existe una gran demanda por el desarrollo de este tipo de aplicaciones.

Los desarrolladores desde el inicio de los celulares inteligentes comenzaron a desarrollar aplicaciones nativas ya sea para Android o iOS u otros S.O. móviles, tomando distinto lenguaje de programación para cada plataforma.

Con el paso del tiempo se fueron desarrollando y creando herramientas, librerías, paquetes que mejorarían y ayudarían en el desarrollo de las aplicaciones móviles. Pero aun así el aprender a desarrollar aplicaciones móviles era un reto que varios desarrolladores no se animaban a sobrepasarlo.

Es así que nace Ionic framework una herramienta que permite el desarrollo de aplicaciones móviles con un diseño y estructuración de una manera fácil.

## 2. IONIC

Ionic es un framework que está construido con Angular y SASS (SASS es un preprocesador de CSS), que nos provee de una serie de componentes con los que permite crear apps para distintas plataformas móviles. Y lo más interesante, si eres desarrollador web te será muy fácil, porque todo lo que debes saber es HTML, CSS y Angular.


***Figura 1.*** *Logo de Ionic Framework*

## 2.1 VENTAJA DE UTILIZAR IONIC

Ionic permite desarrollar aplicaciones con el lenguaje de HTML, CSS, JavaScript y lo más importante aplicaciones son hibridas, esto quiere decir que una aplicación puede ser exportarla para que pueda correr en distintas plataformas móviles Android, IOS, Windows Phone u otras, esto es una gran ventaja para los desarrolladores web ya que no necesitan aprender otro lenguaje nativo para el desarrollo de aplicaciones móviles.

En cuanto a rendimiento no tiene nada que envidiar a las App nativas, ya que al estar optimizado con AngularJs posee de una gran capacidad de procesar información.

Es fácil de comprender su estructura y su funcionamiento, su estructura está basada en el modelo MVC (Modelo Vista Controlador).

## 2.2 COMO TRABAJA IONIC?

Ionic permite desarrollar y construir en HTML, css y JavaScript, todo esto es interpretado por cordova, que construye la aplicación para la plataforma destino.

### QUE ES CORDOVA?

Es una plataforma de desarrollo de open source, para construir aplicaciones nativas usando HTML, CSS y JavaScript. Prácticamente lo que hace es agarrar el código y transformarlo a lenguaje nativo móvil.

Córdova ofrece muchos plugins de los que se puede disponer, plugins que permite acceder a las propiedades de la cámara, un plugin que permita leer códigos QR, plugins que permitan enviar documentos a imprimir a una impresora y una infinidad de plugin disponibles. En la página oficial se puede ver muchos de los plugin con los que cuenta. http://plugins.cordova.io/

***Figura 2.*** *Dispositivos móviles Inteligentes con sistemas operativos más reconocidos.*

## 2.3 Instalación IONIC

Otra de las ventajas es la fácil instalación del entorno de desarrollo. Para poder instalar Ionic antes se debe tener instalado Node.js. Una vez que se tiene instalado nos vamos a la consola y seguimos los siguientes pasos.

Instalamos ionic y Cordova mediante el siguiente comando en la consola:

```
npm install -g ionic cordova
```

Listo...! de esta manera tan sencilla instalamos Ionic Framework.

Para crear un proyecto debemos ejecutar desde la consola el siguiente comando:

```
ionic start Appionic blank
```

"Appionic" es el nombre del proyecto y aplicación al que se puede poner el nombre que se desee. Este comando crea una app que hereda de la plantilla básica "blank", que es la más sencilla. También se dispone de estas otras dos plantillas: "tabs" y "sidemenu".

En la carpeta de nuestra aplicación "Appionic", vemos que se han añadido muchos directorios y

ficheros que son parte de nuestra aplicación.

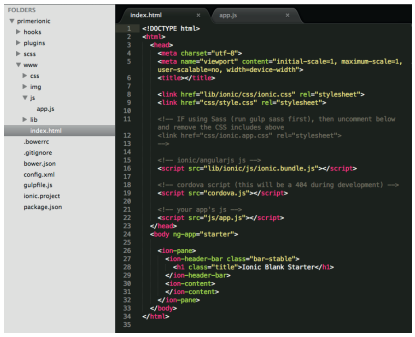

***Figura 3.*** *Directorio de un proyecto Ionic visualizado en Sublime Text (editor de código).*

## CONCLUSIONES

En conclusión, se ha podido ver que esta tecnología es realmente productiva al momento de desarrollar aplicaciones híbridas multiplataforma, ofreciendo todas las facilidades ya que utiliza casi en su totalidad el uso de AngularJS, un framework MVC ultra eficaz.y muy fácil de utilizar.

Si eres desarrollador web y deseas entrar al mundo del desarrollo de aplicaciones móviles, nada mejor que Ionic

## BIBLIOGRAFÍA

[1]. Tiago Coelho. (Marzo 2016). Desarrollo de Aplicaciones Híbridas Móviles con Ionic Framework. Obtenido de: http://equipo.altran.es/desarrollo-aplicaciones-hibridas-moviles-ionic-framework/

[2]. Oscar. (Marzo 2015). Ionic Framework – Primera aplicación. Obtenido de: http://www.aprendeinformaticaconmigo.com/ionic-framework-primera-aplicacion/

[3] Andrea Ardions. (Septiembre 2016). ¿Qué es IONIC? – Todas sus ventajas. Obtenido de: http://www.randyvarela.es/ionic-definicion-ventajas/

[Figura 1]. Logo de Ionic Framework. Recuperada de: https://commons.wikimedia.org/wiki/File:Ionic_Logo.svg

[Figura 2]. Sistemas operativos Móviles. Recuperada de https://www.emaze.com/@AOROFQTQT/expocicion

# CONTROL DE TRÁFICO EN UNA RED DE COMPUTADORA

**Ayarde Ponce Liliana Ximena**
Docente de la Carrera de Ingeniería Informática de la UAJMS
Tarija, Bolivia

**Correo electrónico:** layarde@gmail.com

## RESUMEN.

La amplia presencia redes de computadoras, la necesidad de interconexión entre ellas para el intercambio de información y la exigencia de emplear eficientemente los recursos, llevan a considerar el empleo de aplicaciones para generar soluciones que satisfagan estas necesidades y exigencias, uno de los recursos importantes en las redes de computadoras es el ancho de banda que se requiere para la transmisión de la información, ya que es costoso y limitado, para utilizar eficientemente el canal se necesita controlar de alguna forma el tráfico en la red o entre las redes.

Se hace necesario disponer de soluciones que hagan un uso eficiente del ancho de banda disponible y que además se puedan brindar servicios diferenciados servicios que se ofrecen en diferentes plataformas de sistemas operativos.

Con el enrutamiento avanzado se presentan las herramientas fundamentales para efectuar tareas tales como túneles, firewall, múltiples tablas de enrutamiento, gestión de ancho de banda, multicasting, etc.

## PALABRAS CLAVE.

Control del Tráfico, Probabilidades, Eficiente, Servicio.

## I. INTRODUCCIÓN.

En las redes de computadoras de tráfico de datos que circulan los paquetes que fluyen por la red solicitan ser servidos por los diferentes nodos (Routers, Switches, Bridges, Hubs), se requiere considerar entonces el ancho del canal de transmisión, la velocidad de procesamiento, el tamaño del buffer y la priorización del flujo de estos paquetes en la red. Para el diseño y evaluación de equipos de procesamiento y redes de computadoras se necesita hacer uso de la teoría de Colas, mediante el empleo de modelos de colas y análisis simplificado de estos, soportados en suposiciones que permiten reducir la complejidad del análisis y llevan a resultados satisfactorios.

En general para un modelo de colas se pueden definir tres características básicas: proceso de entrada, mecanismo de servicio y disciplina de cola.

**Proceso de Entrada:** Describe la secuencia de solicitudes de servicio. Generalmente el proceso de entrada es descrito en términos de distribución a lo largo del tiempo entre instantes consecutivos de ingreso de clientes.

**Mecanismo de Servicio:** tiene que ver con el número de servidores y la cantidad de tiempo que el cliente usa los servidores.

**Disciplina de Colas:** específica la disposición de que se tiene para bloquear clientes (clientes que encuentran todos los servidores ocupados),

puede darse que los clientes bloqueados decidan abandonar el sistema inmediatamente o que esperen en la cola por el servicio, los cuales pueden ser atendidos en orden especifico.

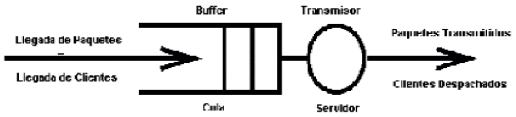


***Figura 1.*** *Modelo de Cola de un Servidor*

La figura 1 presenta un esquema de cola de un solo servidor donde se visualizan en la parte superior los términos de las redes de conmutación de paquetes en un enlace de transmisión y en la parte inferior los términos asociados a la teoría de colas.

En general un sistema de colas se clasifica de acuerdo a sus propiedades, algunas de ellas son:

1. La forma de la distribución de la llegada de los clientes.
2. La forma de la distribución del tiempo de servicio.
3. El número de llagadas en un grupo.
4. El número de Servidores.
5. La disciplina de servicio, es la manera en el cual el servicio es condicionado, si el sistema tiene prioridades.
6. Número de Clientes que se les permite esperar
7. Número de clientes en la población.

## 1.2 ELEMENTOS DE TEORÍA DE PROBABILIDADES Y PROCESOS ESTOCÁSTICOS.

### 1.2.1 Variables Aleatorias.-

El concepto de variable aleatoria dentro de un espacio muestral como una función de este, y lo que nos interesa es que representa mediante un valor numérico un evento al cual se le puede asociar una probabilidad, ejemplos de variables aleatorias en sistemas de colas es el número de clientes en espera, el tiempo de ocupación del servidor, el tiempo de espera.

### 1.2.2 Distribución Exponencial.-

Se tiene una variable aleatoria continua X con una función de distribución.

$$P\{X \leq x\} = F(x)$$

Se dice entonces que es exponencial si cumple:

$$F(x) = \begin{cases} 1 - e^{-ux} (x \geq 0) \\ 0 \qquad (x < 0) \end{cases}$$

Y su función de densidad de probabilidad es: Una distribución exponencial tiene la propiedad de ausencia de memoria o propiedad de Markov, que se define como, una variable aleatoria X sin memoria si y solo si para cada se da:

$$P\{X > \alpha + \beta | X > \beta\} = P\{X > \alpha\} \qquad (3)\ \text{Y}\ [6]$$

### 1.2.3 Distribución de Poisson.-

Una variable aleatoria discreta X sigue una distribución de Poisson con un parámetro.

$$\lambda$$

Si su función de densidad de probabilidad es:

### 1.2.4 Procesos de Poisson.-

Muchos sistemas de colas en el proceso de llegada son caracterizados como tipo Poisson, pues al asumir que el comportamiento de llegada de los clientes en una cola se asemeja a un proceso de Poisson, nos permite reducir la complejidad analítica del problema permitiendo obtener resultados útiles.

Se tiene un proceso A(t) que corresponde al

número de llegadas en un tiempo t donde A(0)=0, y A(t) – A(s) son el número de llegadas en un intervalo (s,t].

El proceso A(t), con

$$t >= 0$$

sigue una distribución de Poisson con Parámetro.

$$\lambda t$$

De acuerdo a:

$$\lambda \tau$$

Donde

Número medio de llegadas y $\lambda$

Cabe destacar dos propiedades importantes de un proceso de Poisson:

1. Mezclado: Se asume que se tienen

$$N_1(t) \ y \ N_2(t)$$

Ambos procesos de Poisson independientes

$$\lambda_1 \ y \ \lambda_2$$

Respectivamente de estos dos procesos se da origen a un nuevo proceso de Poisson.

$$\lambda_1 \ y \ \lambda_2$$

División: Se asume que se tiene un Proceso de Poisson

$$\lambda$$

Y cada llegada está marcada por una probabilidad P independiente de las otras llegadas. Dejando que

$$N_1(t) \ y \ N_2(t)$$

Sean respectivamente los procesos de salida de este, entonces se tiene una distribución de Poisson.

$$p\lambda \ y \ (1-p)\lambda$$

Para cada camino de salida respectivamente.

Se tiene entonces que un proceso de Poisson se mantiene bajo el mezclado o la división.

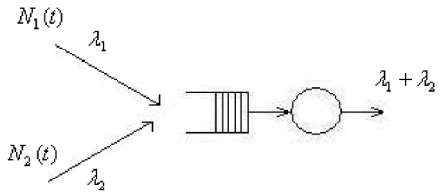


***Figura 2.*** *Propiedad de Mezclado de un Proceso de Poisson.*

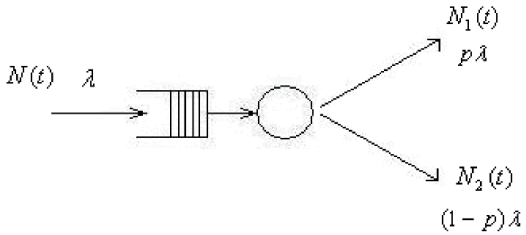


***Figura 3.*** *Propiedad de División de un Proceso de Poisson*

### 1.2.5 Proceso de llegada y tiempo de Servicio.-

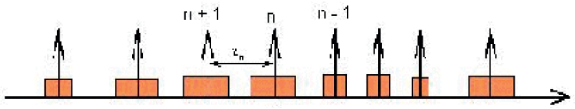


***Figura 4.*** *Proceso de Llegada*

En la figura 2 se tiene

$$\tau_n$$

Tiempo entre llegadas de los clientes n y n+1, el cual es aleatorio y responde a un proceso estocástico.

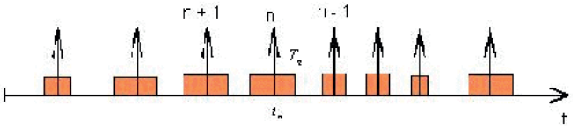


***Figura 5.*** *Proceso de Tiempo de Servicio*

Es el tiempo de servicio del cliente n en el servidor, el cual es un proceso estocástico para n>= 1.

### 1.2.6 Procesos de Markov.-

Un proceso de Markov es un proceso estocástico donde:

$$(x(t), \in T) \qquad X(t) \in E \subset \Re$$

Para el cual se cumple:

$$P(X(t)) \leq x \mid X(t_1) = x_1, \ldots, X(t_n) = x_n) = P(X(i) \leq x \mid X(t_n) = x_n \quad (6)$$

Para todo $x_1, \ldots, x_n \in E$, $t_1, \ldots, t_n \in T$, $con \quad t_1 < t_2 < \ldots < t_n < t$

De manera intuitiva podemos deducir de (6) que un proceso de Markov donde la evolución del futuro probabilístico después de cualquier tiempo t depende únicamente del estado del sistema en el tiempo t y es independiente de la historia del sistema a priori al tiempo t.

Se diferencia las cadenas de Markov en tiempo discreto y las cadenas de Markov en tiempo continuo.

## CONTROL DEL TRÁFICO.

Es el conjunto de sistemas de encolado y mecanismos por los cuales los paquetes son recibidos y transmitidos en un enrutador. Esto incluye la toma de decisión de cuales paquetes aceptar en la entrada de una interface y determinar cuáles paquetes transmitir en qué orden a la salida de una interface. Dado que los enlaces de red transmiten los datos en una forma serial, una cola es requerida para manejar los paquetes de datos salientes.

De una PC de escritorio y un servidor WEB compartiendo el mismo enlace hacia internet, puede ocurrir la siguiente contención de ancho de banda: El servidor WEB puede ser capaz de llenar la cola de salida en el router más rápido de que los datos puedan ser transmitidos a través del enlace.

Cuando se tenga un tipo de problemas que puedan ser solucionados usando estas técnicas y maximizar el uso de una conexión de red.

Limitar el ancho de banda a un valor conocido.

Limitar el ancho de banda a un usuario, servicio o cliente determinado.

Maximizar el troughput en un enlace asimétrico, priorizar la transmisión de paquetes.

Reservar ancho de banda para un usuario o aplicación particular.

Dar prioridad al tráfico sensitivo a la latencia.

Permitir distribución equitativa de ancho de banda no reservado.

Asegurar que un tipo particular de tráfico sea rechazado.

## CONCLUSIONES.

Dentro de las ventajas que ofrece el control del tráfico cuando se emplea apropiadamente, está el lograr tener un uso más predecible de los recursos de la red y una contención menos volátil. La red cumple entonces con las metas de la configuración del control de tráfico.

Además si la configuración de control de tráfico representa una política que ha sido comunicada a los usuarios entonces ellos sabrán que esperar de la red.

Sobre las desventajas, la complejidad es fácilmente la más significativa al usar control de tráfico.

Los recursos computacionales requeridos en un

router para soportar control de tráfico, deben ser capaces de manejar el incremento en el costo de mantener las estructuras de control de tráfico, que afortunadamente es un pequeño costo incremental, pero se puede volver más significante a medida que la configuración crece en tamaño y complejidad.

## 4. BIBLIOGRAFÍA.

[1] Harris Nick et al. Linux Handbook, A Guide to IBM Linux Solutions and Resources. Redbooks, 2003.

[2] COOPER, Robert. Introduction to Queueing Theory. 2 ed. New York: North Holland, 1981.

[3] GELENBE, E; PUJOLLE G. Introduction to Queueing Networks. Great Britain: John Wiley & Sons Ltd. 1987.

[4] Sistema Operativo Linux y Control de tráfico en redes de Computadoras, Juan Carlos Rengifo Salazar, Universidad de Antioquia facultad de Ingeniería.

[5] Calidad de Servicio, Rogelio Montañana Departamento de Informática Universidad de Valencia.

[6] Prado, C. 2009. Implantación de Calidad de Servicio (QOS) en Redes Inalámbricas WI-FI. Tesina. Escuela Superior de Ingeniería Mecánica y Eléctrica. Unidad Culhuacan. México, D.F.

# POTENCIALIDADES Y DEBILIDADES DEL PROCESO UNIFICADO DE DESARROLLO Y WATCH

**Padilla Vedia Carmen Janeth**
Departamento de Informática y Sistemas UAJMS
Tarija, Bolivia

**Correo electrónico:** padillac555@gmail.com

## RESUMEN

Al realizar este artículo se trata de reflejar una panorámica acerca de los conceptos y características de la Ingeniería de Requerimientos (IR), buscando resaltar su relevancia dentro del ciclo de desarrollo de proyectos de software, conocer las diferentes alternativas o técnicas que existen para identificarlos, analizarlos, documentarlos ,así como mostrar la importancia que tienen herramientas automatizadas dentro de este proceso de administración de requerimientos.

## PALABRAS CLAVE:

RUP, WATCH, stakeholders, Fases, Flujos, Actividades, Trabajadores, Procesos, Release.

## PROBLEMA DE INVESTIGACIÓN

Todo proceso de desarrollo de software implica el trabajo coordinado de fases, etapas o como se denominen de acuerdo a la metodología empleada para su desarrollo. Lo cierto es que si tuviésemos que enfrentar un proyecto de desarrollo de software , nos enfrentamos a la interrogante ¿ que metodología o método usar para hacerle frente a esta actividad?, la cual conlleva tareas y responsabilidades dentro de una organización de desarrollo y que debe permitir obtener productos en tiempos considerables con costos adecuados , al igual que los resultados finales cumplan con los atributos de calidad mínimamente exigibles por el cliente. En esta oportunidad se presentara una evaluación comparativa de la metodología RUP y el método WATCH, identificando fortalezas y debilidades que presentan estas alternativas de para hacerle frente al proceso de desarrollo de software.

## 1. INTRODUCCIÓN

Desde hace tiempo se viene enfrentando el desarrollo de productos de software con metodologías tradicionales que permitían llevar adelante este proceso que en principio tenían características mayormente secuenciales, dada esta particularidad se fueron quedando obsoletas y fuimos pasando a otras tendencias como las orientadas a objetos a medida que los lenguajes de programación evolucionaban con estas particularidades.

Las metodologías tradicionales fueron explotadas oportunamente cuando las necesidades de los usuarios eran mínimas comparadas a las exigencias de los usuarios de hoy en dia, sin embargo estas metodologías cubrieron un marco de metodológica el proceso de desarrollo de software y permitir a los desarrolladores entregar productos con calidad en entornos que requerían menos recursos que los entornos actuales.

Hoy en dia los requerimientos son mayores por los tanto hay que sustentarse en modelos que permitan enfrentar el proceso de desarrollo de forma que podamos satisfacer estas necesidades que cada dia son más y mas exigentes por los usuarios. Hoy en dia los usuarios necesitan respuestas

tecnológicamente más complejas que nos hacen pensar en la (s) metodologías a emplear para el desarrollo serán aquellas que cubran de manera exitosa estas necesidades que son evidentes en nuestros entornos actuales.

El desarrollo de software para entornos o dominios empresariales es un proceso complejo que requiere el tratamiento apropiado de aspectos organizacionales, gerenciales y tecnológicos por tanto un de los aspectos importantes es la metodología a emplear para lograr este propósito.

### 1.1. Método WATCH

El método WATCH es un marco metodológico que describe los procesos técnicos, gerenciales y de soporte que deben emplear los equipos de trabajo que tendrán a su cargo el desarrollo de aplicaciones de software empresarial.

Un marco metodológico es un patrón que debe ser instanciado, es decir adaptado cada vez que se use. Cada equipo de trabajo deberá usar el método como un patrón o plantilla metodológica, a partir de la cual dicho equipo debe elaborar el proceso específico de desarrollo de la aplicación que se desea producir.

Se ubica dentro de los métodos disciplinados ya que se centra en los procesos, hace énfasis en los productos y la organización, involucra procesos bien definidos y documentados, requiere de alta formalidad en el proceso de desarrollo, son procesos repetibles, los resultados son predecibles.

Este método incluye, también, una descripción de los procesos de gerencia del proyecto que se aplicarán para garantizar que el proyecto se ejecute en el tiempo previsto, dentro del presupuesto acordado y según los estándares de calidad establecidos.

### 1.2. Proceso Unificado de Desarrollo de Software

Es un proceso que de manera ordenada defina las tareas y quién de los miembros del equipo de desarrollo las hará. Es una guía para usar UML.

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo.

Como RUP es un proceso, en su modelación define como sus principales elementos: Trabajadores, actividades, artefactos, flujo de actividades.

### 1.3. Proceso de Desarrollo de Software según RUP

Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto2. En la fig. 1 se indica que este conjunto de actividades, en el proceso de desarrollo de software que proponen Jacobson, Rumbaugh y Booch, tiene la misión de transformar los requerimientos del usuario en un producto de software; de manera que los integrantes del equipo y todo aquel que pueda estar interesado en el producto final, tenga la misma visión y no ocurra cuando no se aplica un proceso de desarrollo (fig. 2).


*Fig. 1* *Proceso de desarrollo de software*

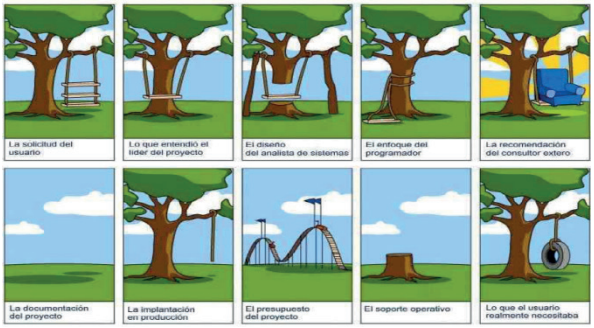


***Fig. 2:*** *Visiones del producto final*

## 1.4. Proceso de desarrollo según WATCH

El método WATCH está fundamentado en las mejores prácticas de la Ingeniería de Software y la Gestión de Proyectos. Cubre todo el ciclo de vida de las aplicaciones; desde el modelado del dominio de la aplicación, pasando por la definición de los requisitos de los usuarios, hasta la puesta en operación de la aplicación.

Este método incluye, también, una descripción de los procesos de gerencia del proyecto que se aplicarán para garantizar que el proyecto se ejecute en el tiempo previsto, dentro del presupuesto acordado y según los estándares de calidad establecidos.

El método WATCH está compuesto por tres modelos fundamentales: (fig. 3 Modelos de WATCH)

- Un modelo de productos
- Un modelo de Actores
- Un modelo de Procesos

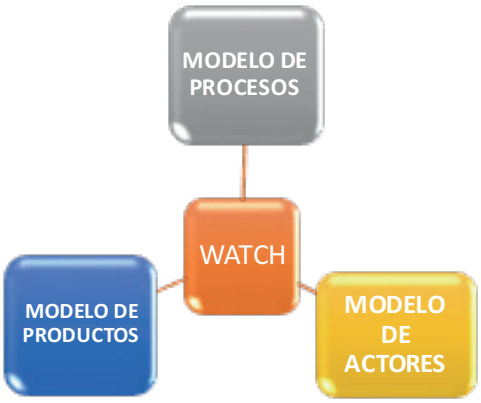


***Fig. 3:*** *Modelos de WATCH*

## 2. CARACTERÍSTICAS DEL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

### a) Las 4 P del proceso de Desarrollo

Se considera piedras angulares del proceso de desarrollo del software a: el proyecto, las personas y el producto; siendo las características del cliente, el entorno de desarrollo y las condiciones del negocio, elementos que influyen en el proceso.

El conocimiento y la experiencia que crea y sostiene la evolución del producto lo tienen las personas. Ellas también financian, se benefician, lo prueban y planifican. Sin un personal competente y experimentado es imposible crear productos competitivos que satisfagan las necesidades de los clientes. Además, el modo en que organiza y gestiona un producto afectan a las personas involucradas en su realización.

Un proyecto es un elemento organizativo a través del cual se gestiona el desarrollo del software.

Un proyecto de desarrollo obtiene un versión de un producto que contiene modelos, código fuente, documentación y un ejecutable. Este producto va evolucionando durante el proceso de desarrollo desde un proyecto inicial o innovador (prototipo

inicial) hasta convertirse en un release del proyecto.

**b) Lenguaje Unificado de Modelado**

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, este lenguaje abarca actividades tales como:

- **Visualizar,** Lenguaje gráfico con una semántica bien definida que estandariza la modelación durante el proceso de desarrollo del software para que sea legible por todo el equipo de proyecto.
- **Especificar**, Se construyen modelos precisos, no ambiguos y completos.
- **Documentar**, Permite describir requerimientos, la arquitectura y modelar las pruebas a través de artefactos que permiten documentar el proceso.
- **Construir**, No es un lenguaje de programación, pero sus modelos pueden transformarse en código fuente, tablas o al almacenamiento de objetos (Generación directa del código).

**c) Fases**

Presenta 4 fases : Inicio, Elaboración, Construcción y Transición

- **Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario . Se obtiene 1 o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.
- **Transición:** El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Estas fases están relacionadas con 7 flujos fundamentales los cuales son:

- Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por loq ue indica con precisión lo que se debe programar.
- Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- Prueba (Testeo): Busca los defectos a los largo del ciclo de vida.

- Instalación: Produce release del producto y realiza actividades (empaque, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

- Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

- Administración de configuración y cambios: Describe cómo controlar los elementos producidod por todos los integrantes del equipo de proyecto en cuanto a: utilización/ actualización concurrente de elementos, control de versiones, etc.

- Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

**d) El ciclo de vida de RUP se caracteriza por:**

1. Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

2. Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.

3. Iterativo e Incremental: Los flujos de trabajo se desarrollan en cascada, RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

Aunque cada iteración tiene que proponerse un incremento en el proceso de desarrollo, todas deben aportar al principal resultado de la fase en la que se desarrolla.

## 3. CARACTERÍSTICAS DEL MODELO WATCH

a)

**1)** Un modelo de productos que describe los productos intermedios y finales que se generan, mediante el uso del método, durante el desarrollo de una aplicación empresarial.

**2)** Un modelo de actores que identifica a los actores interesados (stakeholders) en el desarrollo de una aplicación y describe cómo deben estructurarse los equipos de desarrollo y cuáles deben ser los roles y responsabilidades de sus integrantes.

**3)** Un modelo de procesos que describe detalladamente los procesos técnicos, gerenciales y de soporte que los equipos de desarrollo deberán emplear para elaborar las aplicaciones.

b) El método WATCH emplea el paradigma de desarrollo de software basado en la reutilización

de componentes de software. En base a este paradigma, una aplicación empresarial tiene una arquitectura de software de tres o más capas, en la que cada una de las capas está compuesta de un conjunto de componentes de software interrelacionados.

c) Está sólidamente fundamentado. Posee una base conceptual y metodológica muy bien sustentada. El método descansa en conceptos bien establecidos que se derivan de la Ingeniería de Software y los Sistemas de Información Empresarial. En concreto, el método emplea una arquitectura de dominio de tres capas que define los elementos principales de las aplicaciones empresariales modernas.

Metodológicamente, el modelo ha sido elaborado tomando como referencia modelos de procesos bien conocidos o bien fundamentados, tales como el modelo RUPRational Unified Process (Krutchen, 2000) y versiones anteriores del método WATCH (Montilva y Barrios, 2004b).

d) Es estructurado y modular. Posee una clara estructura que facilita su comprensión y utilización. Esta estructura separa los tres elementos primordiales de un método: el producto que se quiere elaborar, los actores que lo elaboran y el proceso que siguen los actores para elaborar el producto. Estos tres elementos definen los tres componentes del método WATCH: modelo de productos, modelo de actores y modelo de procesos. Cada uno de ellos posee, a su vez, una estructura claramente visible y acorde al elemento que representa.

e) Es de propósito específico. El método está dirigido al desarrollo de aplicaciones de software en entornos empresariales; es decir, al desarrollo de aplicaciones que apoyan uno o más sistemas de negocios de una empresa. Esta orientación concreta y específica resuelve los problemas que tienen la mayoría de los métodos comerciales y académicos existentes, cuya generalidad va en detrimento de su aplicabilidad en software especializado. El método no es apropiado para desarrollar software del sistema (sistemas operativos, utilitarios, middleware, etc.), ni software de programación (compiladores, editores, entornos de programación, etc.).

Tampoco es útil en el desarrollo de software de entretenimiento (videojuegos, herramientas multimedia, etc.). En aplicaciones especializadas, tales como sistemas de información geográfica (GIS), sistemas de control, software educativo y software embebido, el usuario del método debe hacer las adaptaciones pertinentes para ajustar el método al dominio particular de este tipo de aplicaciones.

## 4. ASPECTOS COMPARATIVOS ENTRE EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (RUP) Y EL MARCO METODOLÓGICO WATCH

### 4.1 Potencialidades en ambos modelos

- WATCH considera el desarrollo de software iterativo, incremental y versionado, de forma similar lo hace RUP con la particularidad de estar centrado en la arquitectura , guiado por casos de uso, iterativo e incremental, rescatando una potencialidad de esta metodología como lo es "guiado por los casos de uso" lo que implica de forma puntual que el modelo de caso de uso está presente a lo largo de todo el proceso de desarrollo de software.

- Verificación continúa de la calidad de los productos. WATCH asegura la calidad de la aplicación, a través del uso de procesos bien

definidos de Aseguramiento de la Calidad y Verificación & Validación de software (V&V). Los procesos V&V son aplicados a todos los productos intermedios y finales que se elaboran a lo largo del desarrollo de cada aplicación. En tanto RUP con sus componentes asegura calidad en los productos la misma que se obtiene a lo largo del proceso de desarrollo, aplicando métricas de calidad, permitiendo la toma de decisiones adecuadas, tiempos reales en la planificación y pruebas exitosas que evitan la demora en la aceptación de los usuarios.

- WATCH considera una programación guiada por las pruebas. Para codificar los componentes de software, el método emplea el enfoque de programación guiada por las pruebas, la cual consiste en diseñar y preparar las pruebas de cada componente antes de iniciar su codificación. De esta manera, la codificación se hace con la intención de pasar la prueba, lo cual garantiza una mayor calidad del código producido. La codificación y la prueba unitaria del componente se hacen paralela y coordinadamente usando herramientas de pruebas automatizadas. RUP enfrenta esta actividad por casos de prueba que pueden ser diseñados en etapas tempranas del proceso, ya que su modelo lo permite realizar desde la fase inicial y a medida que avanza el proceso se van mejorando en cada iteración, todo esto gracias al modelo de casos de uso que es el hilo conductor del proceso y nos sirve como base para el diseño de las pruebas tanto iniciales como finales.

- WATCH considera una apropiada gestión de cambios. Los cambios en los requisitos y productos elaborados es una constante en el desarrollo de aplicaciones empresariales. Estos cambios pueden surgir en cualquier fase del desarrollo de una aplicación, por lo que es necesario controlarlos apropiadamente, a fin de evitar que el proyecto se postergue continua o indefinidamente. WATCH emplea procesos bien definidos de Gestión de Requisitos y Gestión de la Configuración de Software (SCM) que se encargan de controlar estos cambios. RUP por su parte es claro y preciso en cuanto a la gestión de cambios de refiere, ya que las iteraciones junto con los hitos de cada fase permiten tener un control efectivo de cambios y así permitir los incrementos efectivos a lo largo del proceso de desarrollo de software.

- Ambos modelos WATCH y RUP Emplean Buenas Prácticas y Procesos de Gestión de Proyectos. Integra los procesos de gestión con los procesos técnicos y de soporte.

  - Ambos modelos contemplan visibilidad al proyecto; pues, permite que el grupo de desarrollo y los usuarios del sistema puedan conocer en qué estado se encuentra el proyecto en cualquier momento.

  - Facilita al líder del proyecto las labores de planificación y control del proyecto.

  - Establecen un marco metodológico único que estandariza el proceso de desarrollo y unifica la documentación que se produce a lo largo del proyecto de desarrollo de una aplicación.

  - Están fundamentados en modelos de procesos de la Ingeniería de Software Basada en Componentes.

  - Emplean las mejores prácticas, técnicas y

notaciones utilizadas regularmente en la Industria del Software.

### 4.2 Debilidades en los modelos

#### 4.2.1 WATCH

- Los aspectos iterativos e incremental no están claramente definidos de forma que permitan una aplicabilidad clara durante el proceso de desarrollo como lo hace RUP con las fases y 9 flujos a realizar en cada fase.
- No es un modelo muy aplicable por su poca popularidad y la presencia de una gran numero de detalles que muchas veces hacen perderse en el proceso de desarrollo.
- La aplicación de procesos, técnicas y prácticas gerenciales es un factor crítico de éxito en el desarrollo de software.
- Considera un estilo de cascada que confunde con lo iterativo, faltando puntualizar que procedimientos seguir, como y cuando de forma que sean verdaderas guías en el proceso de desarrollo.

#### 4.2.2 RUP

- Es un método con un cierto grado de complejidad, implica seguir una visión bidimensional en las que se debe respetar fases y flujos , los mismos que requieren conocimientos bien definidos sobre procesos, lo cual conlleva tiempo que si no se llevan a cabo adecuadamente podría ocasionar retrasos considerables en los proyectos, lo cual implica incremento de costos en el proyecto.
- Es un modelo aplicable muy bien para grandes proyectos de software, sin embargo se puede acomodar a proyectos medianos y pequeños sin mayores problemas, pero aquí surge una interrogante ¿será que los costos de producción son rentables? ,en el caso de los proyectos pequeños puede suceder que los costos que se generen con la planilla del equipo de profesionales necesarios sea alta y no le convenga a la empresa que ofrece el software.
- No cuenta con un plan para la estimación de costos del proyecto, que permitirá la determinación del presupuesto del proyecto. No es claro con técnicas o herramientas adecuadas para la estimación de los costos del proyecto dejando un hueco en este aspecto, el director del proyecto no tiene un artefacto claro que le permita monitorear y controlar los costos del proyecto.

## 5. CONCLUSIONES

- Ambos modelos no cuentan con los procedimientos claros y precisos para la gestión de calidad, gestión de riesgos lo que provoca que los directores de proyecto como equipo de desarrollo sin la experiencia debida no puedan enfrentar fácilmente el proceso de desarrollo con alguno de estos modelos analizados.
- EL modelo de WATCH tiene muchas similitudes con RUP porque rescata elementos básicos de esta última, incluyendo rasgos particulares con un enfoque genérico algo diferente.
- Todo modelo propuesto para enfrentar el proceso de desarrollo es importante frente a producir directamente código sin el modelado respetivo, ocasionando productos sin calidad y difíciles de mantener. Sin embargo a veces quedan huecos o vacíos que los modelos no consideran y que provocan en los analistas y diseñadores situaciones de conflicto debido a

la insuficiente información puntual que debería tener cada proceso implicado en el modelo como también los artefactos claramente definidos.

## REFERENCIAS BIBLIOGRÁFICAS


- Avionics Software Engineering. Requirements ,1998]
- Booch, Grady, Jccobson, Ivar y Rumbauch, James. El Proceso Unificado de Desarrollo de Software, España: Pearson Educación, 2007. 688 p
- Booch, G.: Rumbaugh, J. y Jacobson, I.; "El Lenguaje Unificado de Modelado". 2000. Addison-Wesley.
- Falção, H. y Fontes, J.;"¿En quién se pone el foco?. Identificando stakeholders para la formulación de la misión organizacional". Revista del CLAD Reforma y Democracia". No. 15 (Octubre 1999). Caracas.
- Hans van Vliet, "Software Engineering: Principles and Practice", segunda edición, John Wiley & Sons, 2001.
- Sommerville Ian, 2005, "Ingeniería del Software", Sétima edición, México DF, Editorial Pearson.
- Pressman, Roger S. 2006, "Ingeniería del Software: Un enfoque práctico", Sexta edición, México
- Pressman, Roger; Ingeniería de software. Un enfoque práctico. 2002. McGraw.Hill/ Interamericana de España.
- Sommerville I.Ingenieria del Software,setp. Edición .Madrid 2005
- Von Hallen, B.;"Building a Business.

**Páginas Web**

- IEEE Std 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology",
- rhttp://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html
- Rules". http://www.Kpiusa.com/ReadingRoom ReadingRoom.htm

# ESTANDARIZACIÓN Y REUTILIZACIÓN UNA NECESIDAD EN EL DISEÑO DE COMPONENTES EDUCATIVOS

**Jalil Angulo Raquel Ivonné**
Universidad Autónoma Juan Misael Saracho
Tarija - Bolivia

**Correo electrónico:** jalil.raquel@gmail.com

## RESUMEN

La estandarización en el diseño de componentes educativos es necesaria para su reutilización. Frente al tradicional diseño monolítico y muy específicamente orientado a un objetivo, la tendencia actual en el desarrollo de softwares educativos es crear material educativo reutilizable e interoperable entre sistemas. Sin embargo, se está produciendo la dicotomía de que la mayoría de las herramientas que se basan en estándares educativos no están centradas en el uso, mientras que las que no trabajan con ellos sí lo están. El presente trabajo presenta una introducción a los estándares actuales sobre reutilización de componentes educativos y describe algunas propuestas concretas en este sentido.

## PALABRAS CLAVE

Reutilización, Objetos de aprendizaje.

## REUTILIZACIÓN Y ESTANDARIZACIÓN

La reutilización es un hecho cotidiano en la educación presencial. A nivel personal, los profesores reutilizamos apuntes y esquemas didácticos. A nivel institucional, los libros, y otros recursos docentes, se emplean en una amplia variedad de instituciones educativas, con alumnos y entornos pedagógicos muy distintos, tal como indica Downes (Downes, 2002). Sin embargo, cuando se trata de la educación a distancia o componentes educativos, se presenta una problemática importante a dos niveles. A nivel práctico, la reutilización es difícil de conseguir; a nivel teórico, no hay una terminología clara y aceptada. Un aspecto que contribuye considerablemente a estas dificultades es el debate abierto en torno al análisis del proceso del aprendizaje presencial. Por un lado, la manera informal con la que se habla de la transmisión de información y conocimientos en los procesos comunicativos y educativos; por otro, el consenso en la tradición constructivista, que plantea que la comprensión y los conceptos se construyen en la mente del que aprende [20].

Las tecnologías de aprendizaje representan cualquier ambiente o conjunto definible de actividades que estimulan a los aprendices en la construcción del conocimiento y la toma de decisiones [Jona97]. Los programas computarizados pueden ser usados como un medio para apoyar el aprendizaje y las estructuras de conocimiento en el aprendiz [9].

El rendimiento del sistema educativo actual y los vertiginosos adelantos de la ciencia y la tecnología hacen evidente la urgente necesidad de introducir innovaciones metodológicas, técnicas, empleo de medios y recursos complementarios a la educación tradicional, con el fin de mejorar el aprendizaje en la población estudiantil [6]. Hace falta entender y aplicar teorías de aprendizaje humano que den sustento al diseño de ambientes de aprendizaje efectivos [5].

Se conoce además, que para lograr software educativo con las condiciones deseadas, se deben incorporar dentro de las fases de análisis y diseño, aspectos didácticos y pedagógicos, es decir, el diseño instruccional, de manera que faciliten y garanticen la satisfacción de las necesidades educativas del público al que va dirigido el software. Se deben involucrar también a los usuarios, para conseguir identificar necesidades y/o problemas específicos y se puedan establecer mecanismos de resolución adecuados y apoyar cada una de las fases en sólidos principios educativos, comunicativos y computacionales (Mariño 1998, Galvis 2000) [1]. Al mismo tiempo, ha aumentado la tendencia a crear software abierto, fácilmente ampliable o reutilizable. Por ejemplo, podemos encontrar aplicaciones de modelado que permiten añadir algoritmos o componentes de forma sencilla (Davis, 1996), y librerías de objetos reutilizables para crear simulaciones (Kunze y Steffens, 1996; Neilson y Thomas, 1996). Ya existen trabajos en este sentido, no sólo en el ámbito del software educativo, sino también, por ejemplo en el ámbito de las aplicaciones gráficas (Celes y Corson-Rikert, 1997), del courseware (Harding et al., 1996), o del modelado y la simulación (Odum y Peterson, 1996) [11].

La necesidad de reutilizar los materiales en distintas plataformas y tipos de estudiantes ha provocado la creación de estándares que permitan la documentación, búsqueda y distribución de los contenidos educativos que se generan (Morales, 2002) [17]. La estandarización es un elemento necesario para tratar de reutilizar el material educativo y hacer más rentables los costos de su producción. Frente al tradicional diseño monolítico y muy específicamente orientado a un objetivo, la tendencia actual es crear material educativo reutilizable e interoperable entre sistemas [7]. El logro de un alto nivel de reutilización en el proceso de desarrollo de sistemas representa una meta a conseguir y constituye una señal de madurez en cualquier disciplina de ingeniería [13]. La estandarización en el diseño de componentes educativos es necesaria para su reutilización. Frente al tradicional diseño monolítico y muy específicamente orientado a un objetivo, la tendencia actual en el desarrollo de softwares educativos es crear material educativo reutilizable e interoperable entre sistemas. Sin embargo, se está produciendo la dicotomía de que la mayoría de las herramientas que se basan en estándares educativos no están centradas en el uso, mientras que las que no trabajan con ellos sí lo están.

El paradigma de reutilización generativa, prevalece actualmente en el campo de la educación. Sin embargo existen numerosos estudios y trabajos relacionados con el paradigma de la reutilización composicional que se basa en concebir el desarrollo de aplicaciones como la combinación de componentes reutilizables. Sin lugar a dudas tiene algunas ventajas considerables comparándolo con el paradigma generativo que son: adaptabilidad, flexibilidad, versatilidad, modularidad, mantenibilidad, funcionalidad y lo más importante y lo que a nosotros en este momento nos interesa, reutilizabilidad.

En el contexto de la educación y más concretamente en el diseño de componentes educativos hablar de reutilización lleva de inmediato al concepto de objeto de aprendizaje y se asocia siempre a la reutilización de objetos entre plataformas o entre sistemas educativos, sin embargo las bondades de los objetos digitales permiten la reutilización tanto de los recursos como de sus metadatos, incluso con sistemas no directamente vinculados con la educación pero que sirven como recursos de apoyo

para la enseñanza. Gracias a lenguajes y bases de datos con tecnología abierta y al uso de estándares de metadatos este intercambio es posible.

La característica esencial de un objeto didáctico es la "predisposición al reuso" [15]. Los metadatos, es decir, datos sobre datos, se utilizan con el propósito de poder encontrar, gestionar, reusar y almacenar objetos de aprendizaje de manera efectiva. De esta manera, los objetos de aprendizaje se rodean de una capa de datos que ofrecen información sobre el objeto, para poder ser gestionado por distintos sistemas. Podríamos decir que los metadatos tienen como objetivo último la universalización de los objetos de aprendizaje. Por otra lado, la evolución del e-learning ha sido llevada a cabo por múltiples actores, los cuales han contribuido al desarrollo del mismo mediante la construcción de sistemas propios, basados a su vez en estándares propios. Entre los actores actuales dentro del e-learning podemos encontrar los siguientes, sin pretender ser exhaustivos: IEEE/LTSC/LOM, ARIADNE, IMS, ADL, CanCore, DCMI, AICC, PROMETEUS, etc. [23].

La existencia de tales actores en el campo del e-learning ha impactado de forma especial en el diseño de los metadatos. De manera clara se entiende que la intervención de tantos grupos y organizaciones no contribuye a la unificación que se pretende, provocando que surjan incompatibilidades entre los distintos "estándares". Bien es verdad que existe una cierta labor de unificación en el sentido de promover la colaboración entre ellos. Así, por ejemplo, ARIADNE trabaja en el desarrollo de los estándares junto con IEEE e IMS, al igual que ADL trabaja también con IEEE para su modelo SCORM. [23].

Se hace necesaria, por tanto, una labor de unificación que tienda a crear lenguajes comunes, de tal suerte que los distintos estándares puedan comunicarse entre sí, y se convierta en realidad la posibilidad de manejar y reusar objetos de aprendizaje de manera universal. Un primer paso comprende el entender qué tienen en común cada sistema de metadatos y qué es lo que les diferencia. Si bien se han realizado trabajos comparativos entre distintos estándares, pensamos que debe existir un mayor estudio de análisis de diferencias y similitudes que lleve a la universalidad de los estándares. Ante la multiplicidad de los mismos esta labor de interpretación se torna complicada y debe ser abordada de manera secuencial.

El estándar más utilizado es SCORM [24]; que no sólo es un esquema de metadatos, además dispone de otros elementos orientados específicamente al ámbito educativo, como es, por ejemplo, su sistema de LMS (Learning Management System). SCORM es un conjunto de estándares técnicos que permiten a los sistemas de aprendizaje basados en web, el encontrar, importar, compartir, reusar y exportar contenidos de formación de manera normalizada. El modelo SCORM trabaja con contenidos de formación, formados a su vez por "objetos de aprendizaje", es decir, por unidades independientes, con identidad propia suficiente, que representan de manera virtual elementos de aprendizaje. Desde su nivel más elemental, un objeto que puede ser un texto, una imagen, un sonido, una página de Internet, un video, etc., hasta un conjunto de todos ellos. SCORM se concreta en una serie de libros agrupados en tres grandes bloques: Modelo de Agregación de Contenidos (CAM – Content Aggregation Model); Entorno de ejecución (RTE – Run-Time Enviroment); y Secuencia y Navegación (SN – Sequencing and Navegating). La definición de los metadatos se estudia en el libro Meta-data, que

se encuentra dentro del primer libro anteriormente citado [23].

En lo referente a metadatos, SCORM recomienda seguir el estándar IEEE-LOM 1484.12.1-2002, en una versión reducida; es decir, de los 64 elementos que define LOM, SCORM elige unos pocos sólo como obligatorios. Estos elementos obligatorios no son siempre los mismos. Cada componente tiene un conjunto distinto de metadatos obligatorios. SCORM permite utilizar otros metadatos, como podría ser Dublin Core, pero no garantiza que funcionen correctamente [23].

Diferentes entidades han definido estándares educativos. Mientras Dublin Core constituye una propuesta de metadatos genérica, el LTSC (Learning Technology Standards Committee) de IEEE propuso LOM (Learning Object Metadata), ADL (Advanced Distributed Learning) promueve SCORM (Sharable Content Object Referente Model), IMS Global Consortium promueve diferentes especificaciones. Esto se puede considerar una muestra significativa de esta actividad. Tanto Dublin Core como LOM permiten describir recursos educativos de manera precisa, pero no ofrecen soporte adecuado para los aspectos pedagógicos como por ejemplo la estructura de un curso – actividades, recursos, roles- . Sin embargo, SCORM y EML sí permitirían aproximaciones más pedagógicas. Mientras SCORM está limitado al medio web y a procesos de aprendizaje individuales, EML no está limitado al web, permite interacción entre múltiples estudiantes, y es pedagógicamente neutro mediante un meta-lenguaje que da soporte a diferentes aproximaciones pedagógicas. IMS destaca por su flexibilidad en términos de re-usabilidad e interoperabilidad, integrando estándares previos como IEEE LOM, SCORM y las especificaciones de EML (que se han transformado en IMS Learning Design) [23].

En este contexto se han desarrollado un número de herramientas variadas que se basan en estos estándares. Como ejemplos de aplicaciones comerciales podemos citar el sistema ejecutor de cursos o player EDUBOX, que interpreta y ejecuta cursos descritos en EML; o una extensión de SCORM para Macromedia Dreamweaver. Como ejemplos de software libre o abierto podemos citar la herramienta de autor Colloquia, o Canvas Learning , que es un entorno para ejecutar y de crear preguntas, que trabaja con IMS-QTI (un subconjunto de IMS) [23].

Otro aporte lo encontramos en el desarrollo de QAed que intenta combinar la estandarización con la usabilidad. QAed es una aplicación para la edición de preguntas y cuestionarios, que cumple el estándar de eLearning IMS-QTILite de IMS. QAed es una aplicación para la edición de preguntas y cuestionarios, que cumple el estándar de eLearning IMS-QTILite. Es una aplicación multiplataforma, de código abierto, y desarrollada en Java. La filosofía es que el diseño orientado a la usabilidad permita a los creadores de material educativo utilizar QAed sin tener que conocer los estándares.

QAed permite crear preguntas y cuestionarios siguiendo el estándar IMS-QTILite, que es un subconjunto de IMS-QTI, y que se limita a cuestionarios de preguntas con selección de la respuesta correcta entre varias. Las preguntas pueden utilizar cualquier combinación de texto e imágenes, y son bastante utilizadas para la evaluación (desarrollo de exámenes) en entornos educativos [23].

Un nuevo estándar es también MPEG-7 (Manjunath et al., 2002); de distribución de contenidos multimedia que permite una descripción completa

de la sintaxis del contenido,así como una codificación a nivel semántico. Los anteriores estándares de la familia MPEG, MPEG-1, MPEG-2 y MPEG-4 se basan en la compresión y digitalización de la señal de video y audio. MPEG-7 los complementa ya que su misión es estandardizar la descripción basada en el contenido de diferentes tipos de información audiovisual. La búsqueda, recuperación e indexación de estos contenidos multimedia necesita disponer de una herramienta que describa los documentos y permita reutilizarlos de manera eficiente. El objetivo de MPEG-7 es proporcionar interoperatividad entre sistemas y aplicaciones utilizadas en la generación, gestión, distribución y consumo de contenidos audiovisuales[19].

La característica principal del estándar MPEG-7 es su flexibilidad: el uso de XML (W3C, 2003) (eXtensible Markup Language) como lenguaje de descripción de contenidos y la posibilidad de definir un conjunto de descriptores adaptado a las necesidades de marcado e indexación según la aplicación deseada, permiten la extensión del estándar MPEG-7 para ser usado en diferentes entornos y adaptarlo en cada caso a las necesidades concretas de utilización. El uso de XML permite realizar comprobaciones tanto de la corrección sintáctica como semántica, a través de un DTD que normalmente es un subconjunto del DTD completo que representa todo el estándar MPEG-7, debido al gran tamaño y complejidad del mismo. XML permite un acceso estructurado a la información contenida en el documento MPEG-7 de forma eficiente, y asegura una conexión con otros estándares de descripción de contenidos a través del uso de transformaciones XSLT[19].

Una extensión de IMS, IMS Simple Sequencing permite la definición de recorridos pedagógicos que el usuario puede seguir para atravesar los recursos, con bifurcaciones según unas condiciones simples. El usuario también puede contestar preguntas definidas siguiendo la especificación IMS Question and Test Interoperability, bien como recurso de aprendizaje o de evaluación.

Nuestro análisis indica que un buen número de estas herramientas privilegia el seguir los estándares, en detrimento de la usabilidad. Concebidas como 'etiquetadoras', estas herramientas tienen interfaces muy cercanas a la especificación, y presentan problemas de usabilidad por diversas razones, como son el empleo de un vocabulario poco comprensible para los docentes creadores, o no estar orientadas a tareas educativas reales. En otras herramientas sucede lo contrario: están orientadas al uso, pero hacen caso omiso de la estandarización. Usualmente, las herramientas se concentran en la práctica, dan soporte a un solo modelo pedagógico, y trabajan con formatos propios, con lo que se reduce la interoperabilidad y los usuarios dependen de sistemas propietarios.

En el desarrollo de QAed1 se intenta combinar la estandarización con la usabilidad. QAed es una aplicación para la edición de preguntas y cuestionarios, que cumple el estándar de eLearning IMS-QTILite de IMS. QAed es una aplicación para la edición de preguntas y cuestionarios, que cumple el estándar de eLearning IMS-QTILite. Es una aplicación multiplataforma, de código abierto, y desarrollada en Java2. La filosofía es que el diseño orientado a la usabilidad permita a los creadores de material educativo utilizar QAed sin tener que conocer los estándares. QAed permite crear preguntas y cuestionarios siguiendo el estándar IMS-QTILite, que es un subconjunto de IMS-QTI, y que se limita a cuestionarios de preguntas con selección de la respuesta correcta entre varias. Las preguntas pueden utilizar cualquier combinación de

texto e imágenes, y son bastante utilizadas para la evaluación (desarrollo de exámenes) en entornos educativos.

El foro Dublin Core Metadata Initiative (DCMI) y el IEEE Learning Technology Standards Committee (LTSC) han anunciado un acuerdo de colaboración por el que se comprometen a mantener y reforzar la compatibilidad de sus respectivos estándares de etiquetado y descripción de contenidos. Dichos estándares, basados actualmente, en ambos casos, en HTML, XML y RDF, permiten el intercambio de información entre sistemas y plataformas distintas.

DCMI se ocupa de características genéricas de los recursos de información, como título, autoría, clasificación temática, etc., mientras que LTSC trabaja en un ámbito concreto, el de los contenidos educativos para entornos virtuales de aprendizaje, por lo que el trabajo desarrollado por ambas organizaciones puede considerarse complementario. El acuerdo supone un paso adelante para la estandarización del sector del aprendizaje electrónico.

## CONCLUSIONES

El máximo aprovechamiento de la gran cantidad de recursos disponibles en Internet se logrará cuando la ubicuidad sea su principal características, cuando los sistemas se intercomuniquen sin importar su campo de aplicación y cuando las tecnologías sean transparentes para los usuarios. Hacia esto apunta lo que se conoce como Web Semántica con la que se pretende formar una infraestructura común y de cooperación que permita compartir y reutilizar los datos a través de aplicaciones, empresas y comunidades. A pesar de estas iniciativas tan amplias, actualmente, se distingue que cada sector trabaja en el desarrollo de sus propios estándares para la interoperabilidad de sus sistemas [26].

Diversos autores han reclamado, recientemente, un cambio de paradigma en el desarrollo de software educativo, para responder de forma eficaz a las crecientes necesidades educativas. Este cambio implica aumentar el grado de reutilización actual en el proceso de desarrollo [13].

Actualmente existen algunas aproximaciones de estándares para la reutilización de componentes educativos de software y se espera para el futuro una estandarización más general y que considere en mayor escala tanto la reutilización como la usabilidad de los componentes.

## REFERENCIAS

1. http://www.academia-interactiva.com/ise.pdf. Propuesta de una metodología de desarrollo de Software educativo bajo un enfoque de calidad Sistémica. María Gabriela Díaz-Antón, María Angélica Pérez, Anna C. Grimmán, Luis E. Mendoza. Universidad Simón Bolívar (USB), Caracas, Venezuela.

2. http://cs.uns.edu.ar/jeitics2005/Trabajos/pdf/38.pdf. Tecnología informática aplicada en Educación. Zulema B. Rosanigo; Alicia Paur; Pedro Bramati. Facultad de Ingeniería – Sede Trelew – U.N.P.S.J.B. Te-Fax (02965) 42 84 02.

3. http://www.cudi.edu.mx/primavera_2006/presentaciones/educacion_mr_chan.pdf

4. Reunión de primavera Cudi2006 Mesa educación. Diseño de Programas Educativos por Competencias y Uso de patrones de objetos de aprendizaje. María Elena Chan Núñez, Adriana Margarita Pacheco Cortés , Simón González Flores.

5. http://rapanui.ucv.cl/Modulo7.htm. Trabajo de aula apoyado por software educativo.

6. http://www.inf.udec.cl/revista/ediciones/edicion6/isetm.PDF

7. Ingeniería de software educativo, teorías y metodologías que la sustentan. Pedro Salcedo Lagos.

8. http://www.tise.cl/archivos/tise97/trabajos/trabajo2/index.htm

9. "Desarrollo de un software interactivo para la enseñanza de física de noveno grado de educación basica utilizando la tecnología internet y la tecnología web". Amery Salazar, Pedro Dorta, Ing. René Cabrera. Universidad de Oriente, Núcleo de Anzoátegui, Escuela de Ingeniería y Ciencias Aplicadas, Departamento de Computación y Sistemas. Barcelona, Venezuela.

10. http://www.tecn.upf.es/scope/showcase/documentation/sayago-interaccio2004.pdf. Diseño y evaluación de una aplicación sencilla de eLearning Sergio Sayago Juanjo Martínez, Rocío García, Josep Blat, Dai Griffiths, Francis Casado Departament de Tecnologia, Grup de Tecnologies Interactives, Universitat Pompeu Fabra, Passeig de Circumval·lació 8, E-08003 Barcelona (España).

11. http://www.tise.cl/archivos/tise97/trabajos/trabajo2/index.htm.Taller Internacional de Software Educativo. Un Método de Desarrollo de Aplicaciones Educativas Hipermedia. Directora Grupo de Investigación y Desarrollo de Software Educativo (GIDSE) María Eugenia Valencia.

12. http://www.dcc.uchile.cl/~luguerre/papers/CVEI-01.pdf . Diseño de Software Educativo. César Alberto Collazos O., Luis A. Guerrero B.

13. http://dspace.ou.nl/bitstream/1820/469/1/BURGOS_KOPER_ComunidadesVirtuales_110805.pdf. Comunidades virtuales, grupos y proyectos de investigación sobre IMS Learning Design. Status quo, factores clave y retos inmediatos Virtual communities, research groups and projects on IMS Learning Design. State of the art, key factors and forthcoming challenges Daniel Burgos and Rob Koper Educational Technology Expertise Centre (OTEC) Open University of the Netherlands Valkenburgerweg 177; PO Box 2960 6401 DL Heerlen; The Netherlands {daniel.burgos;rob.koper}@ou.nl August 11th, 2005.

14. http://www.lsi.uvigo.es/lsi/erosello/imo/articulos/OOP-SIIE99.PDF Algunas consideraciones sobre la programación orientada a objetos y la integración en el software educativo. Emilio García Roselló1, Jose B. García Perez-Schofield2 1Departamento de Lenguajes y Sistemas Informáticos, Facultad de Ciencias - Universidad de Vigo, Lagoas-Marcosende.

15. http://www.cudi.edu.mx/primavera_2004/presentaciones/Lourdes_Galeana.pdf. http://www-gist.det.uvigo.es/~ie2002/actas/paper-143.pdf. E. García Roselló, J. González Dacosta, , E. Mandado, V. G. Valdés Pardo, J. Baltasar García, M. Pérez Cota, Una propuesta para la reutilización de componentes en el proceso de desarrollo de software educativo.

16. http://www-gist.det.uvigo.es/~ie2002/actas/paper-010.pdf. Instrumento de evaluación de software educativo bajo un enfoque sistémico Díaz-Antón, G., Pérez, M., Grimán, A., Mendoza, L.

17. http://www.um.es/ead/red/M2/sicilia46.pdf. Reusabilidad y reutilización de objetos didácticos: mitos, realidades y posibilidades Reusability and reuse of learning objects: Myths, realities and posibilitéis Miguel-Angel Sicilia Prof. Titular del Departamento de Ciencias de la Computación, Universidad de Alcalá. Ctra. Barcelona km. 33600 – 28871 Alcalá de Henares (Madrid) ESPAÑA.

18. http://www-gist.det.uvigo.es/~ie2002/actas/paper-010.pdf. Instrumento de evaluación de software educativo bajo un enfoque sistémico Díaz-Antón, G., Pérez, M., Grimán, A., Mendoza, L.

19. http://www.um.es/ead/red/M2/leonel22.pdf. Generación de una biblioteca de objetos de aprendizaje (LO) a partir de contenidos preexistentes Creation of a learning objects (LO) library from pre existing contents Leonel Iriarte Navarro Departamento de Informática, Universidad Agraria de la Habana.

20. http://www3.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_hernandez.htm. Unidades de aprendizaje, una propuesta de complemento a los objetos de aprendizaje. Eduardo Hernández. Consultor y Asesor Independiente en eLearninng. Ingeniero en Computación e Informática.

21. http://www.um.es/ead/red/M5/minguillon37.pdf. Opera-Learning: Integraci´on de est´andares de distribuci´on de contenidos multimedia y learning objects Opera-Learning: Integrating multimedia content distribution and learning objects standards M. Pascual1 y J. Minguillón, Servei daudiovisuals.

22. http://www.um.es/ead/red/M5/griffiths16.pdf. La aportación de IMS Learning Design a la creación de recursos pedagógicos reutilizables The contribution of IMS Learning Design to the creation of reusable learning resources David Griffiths, Josep Blat, Rocío García y Sergio Sayago Grupo de Tecnologías Interactivas.

23. http://www.um.es/ead/red/14/zapata_LO2.pdf. SEQUENCING OF CONTENTS AND LEARNING OBJECTS – part II SECUENCIACION DE CONTENIDOS Y OBJETOS DE APRENDIZAJE (II)) Miguel Zapata Ros.

24. http://www.cic.ipn.mx/revistas/pages/vol08-03/3_art_057_MOSCA.pdf. Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software Prototype of Software Quality Systemic Model (SQSM) Luis E. Mendoza, María A. Pérez y Anna C. Grimán. Artículo recibido en junio 06 de 2002; aceptado en diciembre 16, 2004.

25. http://spdece.uah.es/papers/Rouyet_Final.pdf. A comparative study of the metadata in SCORM and Dublin Core Juan Ignacio Rouyet2 y Víctor Martín1 1 Universidad Pontificia de Salamanca. Campus de Madrid Facultad de Informática Cátedra de Programación Científica. Departamento de Formación José Echegaray.

26. http://www.tise.cl/archivos/tise2005/08.pdf. Empaquetamiento de Objetos de Aprendizaje Bajo elEstandar SCORM Luís Álvarez, Daniela Espinoza Universidad Austral de Chile, Chile {lalvarez, despinoza}@inf.uach.cl Manuel Prieto Universidad de Castilla La Mancha, España manuel.prieto@uclm.es

27. http://www.uib.es/depart/gte/edutec-e/

Revelec13/ims.html. Edutec. Revista Electrónica de Tecnología Educativa Núm. 13. /noviembre 00 . Proyecto PupitreNet. La Especificación del Proyecto IMS: Instructional Management System . Mercé Gisbert mgc@astor.urv.es Patricia Henríquez pmhc@glorieta.fcep.urv.es. Robert Rallo rrallo@etse.urv.es. Proyecto PupitreNet. Subproyecto EduPupitre.

28. http://www.um.es/ead/red/M2/lopez27.pdf Desarrollo de repositorios de objetos de aprendizaje a través de la reutilización de los metadatos de una colección digital: de Dublin Core a IMS Development of learning objects repositories by the reutilization of metadata of a digital collection: from Dublin Core to IMS Clara López Guzmán, Francisco García Peñalvo, Pedro Pernías Peco, Dirección General de Servicios de Cómputo Académico, Universidad Nacional Autónoma de México.

# DESARROLLO DE UN COMPONENTE Y REUTILIZACIÓN EN VARIAS APLICACIONES

**Chambi Gareca Silvia**
Universidad Autónoma Juan Misael Saracho
Tarija - Bolivia

**Correo electrónico:** chambi.gareca.silvia@gmail.com

## RESUMEN

El desarrollo de software basado en componentes es el paso hacia una verdadera ingeniería y en la actualidad se ha convertido en uno de los mecanismos más efectivos para la construcción de aplicaciones software.

Siguiendo la línea de la programación basada en componentes vemos que la reutilización de componentes software permite optimizar el proceso de desarrollo de software.

El presente trabajo es una investigación que pretende mostrar la experiencia en la en la creación de un componente software bajo la plataforma framework.Net mostrando su consumo y reutilización en varias aplicaciones (Sistema de Almacenes Carolina y Sistema de la toma de pedido PIZZA) como elemento de mejoramiento del desarrollo de aplicaciones informáticas.

En la investigación se expone una descripción pormenorizada de las características que debe cumplir un componente como propiedades, atributos, métodos bajo el framework.Net. y se describen los procedimientos de compilación y utilización de componente, finalmente una síntesis de la experiencia realizada.

El objetivo que persigue la investigación es lograr la creación de un componente y consumirlo en varias aplicaciones informáticas, para lo cual se analizaron y sintetizaron la teoría la programación basada en componentes, las características de un componente, plataformas de desarrollo y especificación sobre la creación de un componente.

La implantación del modelo de componentes tiene múltiples beneficios como el desarrollo de aplicaciones es más sencillo y en un tiempo y con un coste menor, la posibilidad de errores es mínima ya que los componentes son sometidos a un riguroso control de calidad antes de ser comercializados, finalmente las empresas de desarrollo especializadas pueden obtener ingresos adicionales vendiendo componentes a otras empresas.

En el ámbito local hay escasa información sobre la creación de componentes software así como en la programación basada en la reutilización de éstos, sin embargo la información de las redes fue determinante para alcanzar el objetivo.

## PALABRAS CLAVE

Programación, software, componentes, TIC, ingeniería de software.

## INTRODUCCIÓN

En el mercado mundial del software, el paradigma de los componentes se ha convirtiendo en un estándar, y en la actualidad está siendo masivamente usado, por lo que el fin de este trabajo fue profundizar en la creación de componentes de software y experimentar su reutilización como elemento

de mejoramiento del desarrollo de aplicaciones informáticas.

La indiferencia hacia este nuevo paradigma trunca el desarrollo de la ingeniería del software, porque se deja de lado una programación que en la actualidad es una realidad, una tendencia, una nueva forma de hacer ingeniería del software.

Generar gente preparada con lo que usa y necesita la industria es lograr un profesional que pueda competir en el mercado, altamente capacitado para desenvolverse en cualquier área.

Por otro lado el desarrollo de la tecnología basada en con componentes es el desarrollo de una ingeniería más madura, en la cual la disponibilidad de soluciones permite al ingeniero elegir entre varias opciones la más adecuada, en vez de perder el tiempo inventando una o dos soluciones; lo cual conduce a grandes ventajas enfocando de mejor manera los problemas de software resultado de un mejor proceso de ingeniería.

Una analogía es como la ingeniería civil alcanzó un grado de desarrollo cuando comenzó a utilizar piezas prefabricadas o fabricadas, por ejemplo cuando se construye una casa se compran las puertas, las aldabas, pero no fabrican desde cero, lo que se pretende con la investigación de programación basada en componentes es aprender a utilizar y manipular fragmentos de código prediseñados , e introducirlos en una aplicación que se construya, así mismo sobre la base de la manipulación de componentes crear un componente y consumirlo.

Para que la ingeniería del software alcance un grado de maduración se debe lograr la generalización de la reutilización de código existente en el desarrollo de software, todavía desarrollar una aplicación sigue requiriendo la escritura de una gran cantidad de código, en cambio, en otras ingenierías como la electrónica el grado de reutilización es altísimo En este ámbito, la construcción de un dispositivo electrónico se reduce a la integración de la manera adecuada de distintos componentes comerciales.

Los usuarios requieren grandes sistemas en tiempos cortos pero el desarrollo implica demasiado tiempo.

El desarrollo basado en componentes se define sobre la construcción de una aplicación a partir de componentes software comercial o gratuito ya existentes, limitando al mínimo necesario el desarrollo de código nuevo, para que este cambio se produzca es necesaria la creación de un mercado amplio de componentes software de calidad y con un ámbito de aplicación general, también es necesaria la adopción de una arquitectura estandarizada que garantice la interoperatividad de los componentes en distintas plataformas, lenguajes de programación y entornos de desarrollo.

Por lo que proponemos la creación de un componente y su consumo en aplicaciones empleando la teoría de componentes que permitirá evaluar en nuestro ámbito la fiabilidad, funcionalidad, y factibilidad del producto, que sustentará el planteamiento de una propuesta de reutilización de componentes en aplicaciones Basadas en Componentes como una opción para mejorar el proceso de desarrollo del software.

## 2. MATERIALES Y MÉTODOS

Para la investigación se utilizó el enfoque de Programación Basada en Componentes considerando la reutilización como elemento fundamental.

Se propone un procedimiento para crear un componente mostrando las ventajas de reutilización del componente en varias aplicaciones

para contribuir al desarrollo de la ingeniería del software.

Las herramientas usadas para la investigación fueron: ENTERPRISE BEANS, Framework .NET

La metodología seguida se describe

**Configuración del proyecto:**

Iniciado el SharpDevelop se elige la opción de nueva solución, luego la plantilla biblioteca de controles de usuario de Windows, finalmente se escribe el nombre del proyecto como también la ruta donde se desea guardar.

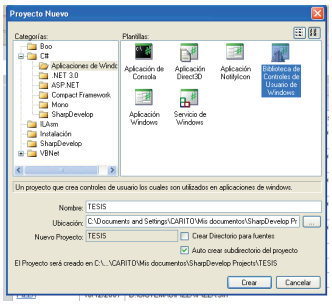


***Fig. 1:*** *Codificación del componente:*

Se realizó la codificación del componente en el entorno de trabajo SharpDevelop 2.1 en el que se visualizan 3 secciones: en la parte izquierda se encuentra la sección de estructura del proyecto o solución, en la parte central la sección de codificación y en la parte derecha la sección de Propiedades como se observa en la figura de abajo.

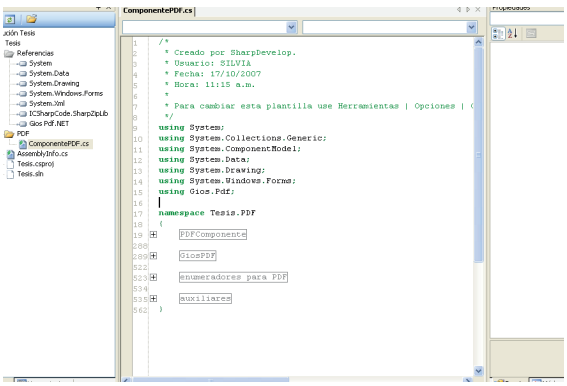


***Fig 2:*** *Compilación del componente*

El componente creado fue establecido como biblioteca de clases por lo que el resultado de la compilación será una DLL (un ensamblado), donde para estar seguros de que el resultado de la compilación sea un DLL se debe verificar las propiedades del proyecto.

En esta figura se señalan dos aspectos que se deben verificar para que el componente sea compilado como una DLL y son el tipo de salida que es Biblioteca de clases y el nombre de salida que debe tener la extensión.dll

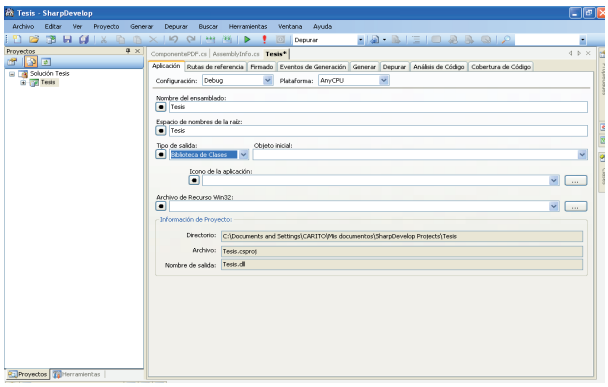


***Fig 3:*** *Reconocimiento del componente por parte del entorno de desarrollo.*

Se debe personalizar la barra de herramientas escogiendo la opción personalizar barra Lateral

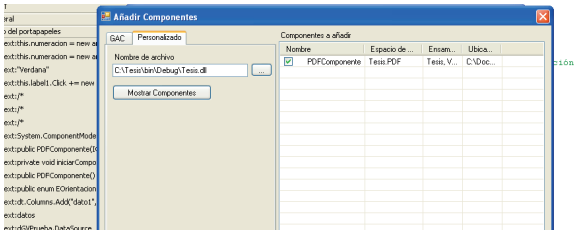


***Fig. 4:*** *Añadir Componentes.*

Si el componente está correctamente definido la ventana de Añadir Componentes debe mostrar al ensamblado de nuestro componente más lista de si existiesen otros componentes.

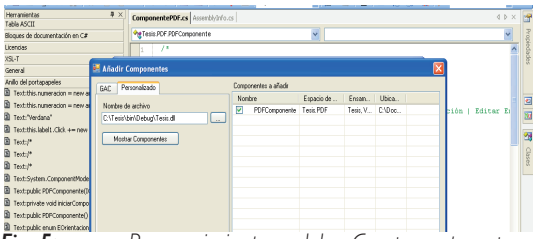

*Fig. 5:* *Reconocimiento del Componente por Microsoft Visual C# 2005 Express Edition*

El componente que exporta archivo a PDF creado por SharpDevelop es reconocido por otro entorno de desarrollo como se muestra en las pantallas siguientes.

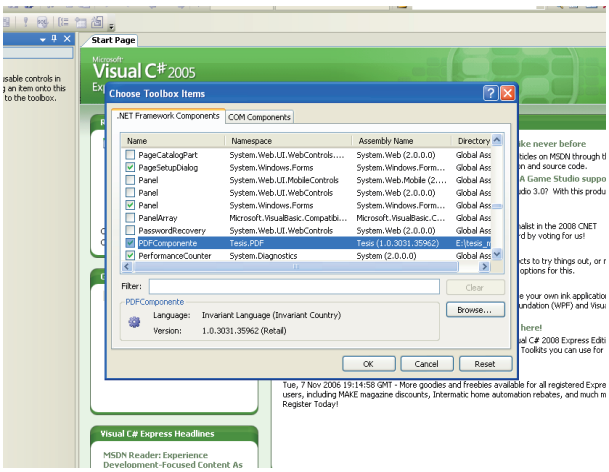

*Fig. 6:* *Prueba de Funcionalidad del componente.*

Para probar la funcionalidad y reutilización del componente creado se describe una de las tres aplicaciones que consumen al componente.

Se describe el consumo del componente por parte de la aplicación informática que es un Sistema de control de pedido que permite la conexión a base de datos y que consta de tres módulos: Toma de Pedido, Administración y Reportes.

El componente es consumido específicamente por el módulo de reportes el que contempla informes de ventas diarias, ventas por pizza, ventas por bebida, ventas por servicios, ventas por clientes, ventas a domicilio, pedidos emitidos, pedidos cancelados, exportación de todos los informes a archivos Excel y pdf.

Esta investigación nace a raíz de que en esta aplicación en una primera versión al usarse las librerías de clases GIOSPDF para la generación de reportes mediante archivos PDF nace la necesidad de convertir a esta funcionalidad en un componente genérico reutilizable.

Es aquí donde se realizaron la mayor cantidad de experimentos para poder reutilizar el componente.

La figura se muestra el consumo del componente PDFComponente con el nombre pdfPizza.

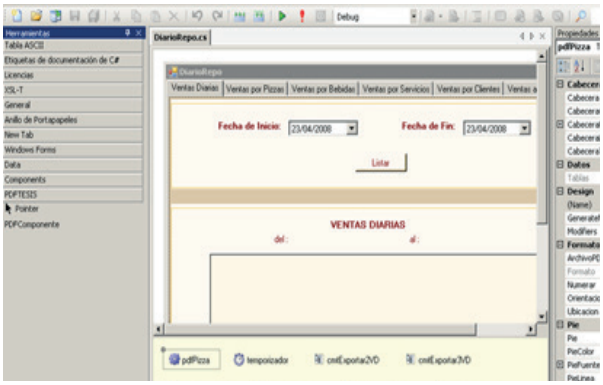

*Fig. 7:* *Se presenta el código para la exportación del archivo a PDF.*

```
void ExportarAPDFToolStripMenuItemClick(object sender, EventArgs
{
        if (dgvVentasDiarias.DataSource != null){
                pdfPizza.Tablas = dgvVentasDiarias.DataSource as DataTable;
                pdfPizza.TituloTexto = "Ventas Diarias";
                pdfPizza.PieTexto = "Del: "+ dtpfecha_inicio.Text+" al: "+
dtpfecha_fin.Text;
                pdfPizza.generarPDF();
                pdfPizza.salvarPDF(true);
        }
        else{
                Mensajes.ventana("No existen datos para exportar", tipoMensaje.Advertencia);
        }
}
```

Se muestra la ejecución de la aplicación y la generación del archivo en formato PDF.

*Fig. 8:* *Vista*



bit@bit Vol. 2 Nº 4 ISSN 2519 - 741X, Pág. 33 - 38. Diciembre 2017

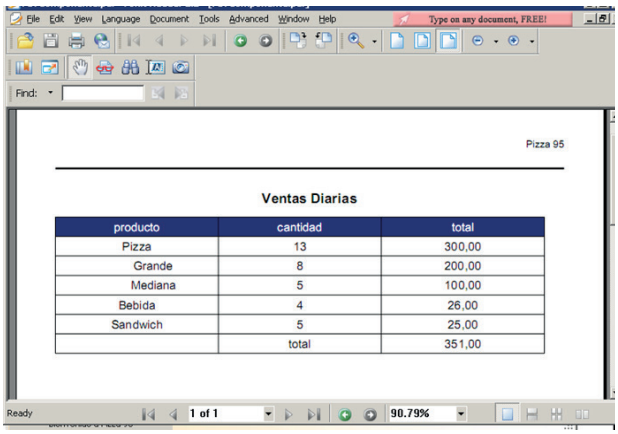


*Fig. 9:* Reporte

## 3. DISCUSIÓN

El paradigma de los componentes es una tendencia sólida y es lo último que utiliza la industria del software, y la utilización de estas tecnologías logra aplicaciones fiables, robustas desarrolladas en tiempos cortos, permitiendo a los profesionales insertarse en proyectos de gran envergadura.

La utilización de plataformas de desarrollo de componentes (framework), permite la elaboración de componentes con un grado adecuado de reutilización por aplicaciones informáticas.

Se desarrolló un componente software generico.

## CONCLUSIONES

- Con la creación de un componente se busca mejorar el desarrollo de aplicaciones informáticas esto se aplica en empresas de desarrollo, en equipos de investigación o grupos de desarrollo.
- Sin la base de la programación Orientada a Objetos programar con componentes es muy complicado y difícilmente se logrará construir sistemas en periodos cortos de tiempo.
- Después de la experiencia en la utilización de componentes propongo se implante este enfoque en los últimos cursos de programación en la carrera de Ing. Informática, donde ya nuestros estudiantes tienen el conocimiento previo de la programación orientada a objetos ya que un componente es una extensión de los objetos y posee características similares a ellos.
- Propongo la creación de un componente avanzado como tema de tesis ya que el componente es una aplicación que necesita análisis, diseño y prueba en las aplicaciones que lleguen a utilizarse.

## BIBLIOGRAFÍA O REFERENCIAS

[1] C. Aranda, M.L Callejo, "Construcción del concepto de dependencia lineal en un contexto de geometría dinámica". Revista Latinoamericana de Investigación en Matemática Educativa, 13(2), 129-158. 2010.

[2] Grady Booch, Ivar Jacobson, James Rumbaugh, Jim Rumbaugh. "The Unified

Modeling Language User Guide. Addison-Wesley", octubre de 1998. ISBN: 0-201-57168-4.

[3] Desmond Francis D'Souza, Alan Cameron Wills. Objects, "Components and Frameworks with UML : The Catalysissm Approach. Object Technology Series". Addison-Wesley, 1999. ISBN: 0-201-31012-0.

[4] Jonás A. Montilva, Nelson Arapé2, Juan Andrés Colmenares. "Desarrollo de Software Basado en Componentes".

[5] Fuentes Lidia, Troya Joséy Vallecillo Antonio Dept. "Lenguajes y Ciencias de la Computación".

Universidad de Málaga. ETSI Informática. Campus Teatinos, s/n. 29071 Málaga, Spain.

[6] Fuentes Lidia, Troya Joséy Vallecillo Antonio Dept. "Lenguajes y Ciencias de la Computación". Universidad de Málaga. ETSI Informática. Campus Teatinos, s/n. 29071 Málaga, Spain.

[7] Zorrilla Santiago, Torres Miguel. "Guía para elaborar la Tesis" Atlacomulco, agosto de 1998. ISBN: 970-10-0139-7

[8] Kraig Brockschmidt. Inside OLE. Second Edition. Redmond, WA: Microsoft

Press, 1993. (Development Library, Books and Periodicals).

[9] Agustín Cernuda del Río "Sistema de verificación de componentes software" Oviedo, Febrero de 2007.

Ejemplo 3: (Cita de una comunicación a un congreso)

Clayton, M. (1998) Computer-assisted, portafolio-nased composition: The next step for

freshman composition at MTSU. Paper presentet at Mid-South Instructional Technology

Conference, Muefreesboro, TN, Abril, 1998

[10]http://nereida.deioc.ull.es/~cleon/doctorado/doc01/lang/poo.html

[11]http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3985.asp/ Desarrollo de Software basado en Componentes. htm

[12] http://computacion.cs.cinvestav.mx/~sgarrido/cursos/ing_soft/Componentes/node55.html

[13] http://java.ciberaula.com/articulo/tecnologia_orientada_objetos/

http://www.geocities.com/eztigma/tendencias.html

[14] http://msdn2.microsoft.com/es-es/library/

[15]http://commanet.blogspot.com/2006/02/java-no-est-orientado-componentes.html (java no esta orientado a componentes)

[16]http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/vbcon/html/vbwlkwalkthr

# ALTERNATIVAS TECNOLÓGICAS APLICADAS AL PROCESO DE CONCIENTIZACIÓN DEL USO RACIONAL DEL AGUA EN LOS HOGARES

**Arancibia Márquez Deysi Beatriz**
Universidad Autónoma Juan Misael Saracho
Tarija - Bolivia

**Correo electrónico:** deysiarancibiam@gmail.com

## RESUMEN

La escasez de agua hoy en día es uno de los grandes retos que afronta la humanidad para el futuro por tal motivo existe la imperiosa necesidad de reducir el consumo de agua en todos los hogares y concientizar a todas las personas a usar racionalmente el agua. En este artículo se presenta una investigación de las diferentes alternativas del uso de la tecnología y su importancia para llegar a los usuarios consumidores de agua en los hogares.

La tecnología permite socializar y concientizar a un uso racional del agua, a través de diferentes herramientas como medios virtuales, multimedia, implementación de controles automáticos y otros, alternativas fáciles de usar y de implementar. Se presentan diversos casos de estudio que se encuentran vigentes al alcance de todos en la web y un estudio de una propuesta para el control automático en los hogares con la finalidad de optimizar y reutilizar el agua de manera óptima.

## PALABRAS CLAVE

Agua, multimedia, control automático, tecnología, virtual, arduino, flujómetro, electroválvula, pantalla.

## 1. INTRODUCCIÓN

El agua es un líquido elemental e indispensable para todos los seres vivos que habitan en el planeta. La vida de toda la humanidad depende del agua. El aumento del consumo de agua se ha multiplicado por seis en un siglo, mientras que la población ha crecido tres veces[1]. Según datos obrantes en la Organización de las Naciones Unidas (ONU), actualmente 80 países del mundo sufren debido a la falta de agua.

Las autoridades de gobierno de los diferentes países, realizan proyectos que involucran el uso eficiente y racional del agua.

El uso racional del agua es tratado en proyectos de ingeniería, arquitectura, urbanismo y agricultura en el marco de la protección y conservación de los recursos naturales.

La tecnología brinda diferentes alternativas para llegar a concientizar al usuario en el uso racional del agua, desde aplicaciones multimedia dirigidos a usuarios de diferentes edades, como aplicaciones de sistemas de control automatizados.

Los principales sistemas de control automatizado son desarrollados con dispositivos electrónicos de aplicación básica acoplados en ocasiones a componentes o sistemas complejos, cuyo costo no debe ser muy alto y ser desarrollado de una forma fácil, para que sea comercializado rápido y fácilmente[2].

## 2. PROBLEMÁTICA

Existe la necesidad de concientizar al usuario en el uso racional del agua, debido a la escasez de este vital elemento. La tecnología a través de diferentes

herramientas permite lograr este objetivo, llegando a cada consumidor de diferentes formas esencialmente con aplicaciones y nuevos diseños tecnológicos que permite ahorrar agua.

En el presente artículo se muestran algunas de las alternativas tecnológicas, tomando como caso de estudio herramientas virtuales, multimedia y diseños de control automatizado que concienticen a los usuarios a cuidar el agua.

## 3. OBJETIVOS

Los objetivos planteados en la investigación son:

- Analizar diferentes técnicas y/o alternativas tecnológicas para concientizar al usuario a un uso racional del agua
- Analizar características de sistemas multimedia que describen la cantidad de agua que se usa en los hogares, para concientizar al uso moderado del agua.
- Analizar el modelo de un control automatizado para cuidar el consumo del agua en los hogares

## 4. METODOLOGÍA

Hoy en día la tecnología permite llegar al usuario para concientizar en el uso racional del agua de diferentes formas, a continuación se describen algunas de ellas.(Los ejemplos que se muestran son el resultado de búsquedas en diferentes fuentes y sitios web).

### 4.1. Herramientas virtuales y productos Multimedia que permiten concientizar el uso racional del agua en los Hogares.

#### 4.1.1. Herramientas virtuales:

En diferentes países las instituciones que se encargan de suministrar el agua a los consumidores ofrecen ayuda de manera virtual para que cada usuario pueda controlar el uso en sus hogares, por ejemplo el enlace: de la oficina virtual ESVAL[3] de Chile permite calcular el consumo del agua a través del enlace http://www.esval.cl/calcula.html.


***Fig. 1 :*** *Calculadora virtual del consumo de agua*

#### 4.1.2: Herramientas Multimedias:

En los siguientes enlaces se encuentra una herramienta multimedia que permite interactuar gráficamente con el usuario para analizar la cantidad de agua que se utiliza en los hogares:

http://www.bbc.co.uk/spanish/flash/swf/water_calculator/water_calculator2.swf https://www.taringa.net/posts/animaciones/13871804/Cuanta-agua-gastas-calculadora-de-consumo-de-agua.html

Las pantallas de interacción son:

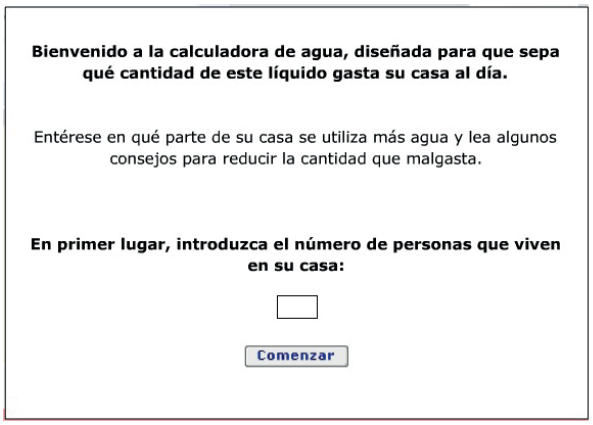


***Figura 2:*** *Inicio del sistema multimedia online*

Inicia con la cantidad de personas que usarán el agua diariamente en la casa, para luego elegir el lugar de la casa:


**Figura 3:** *Menú con Opciones gráficas*

Se selecciona un artefacto, por ejemplo la ducha.

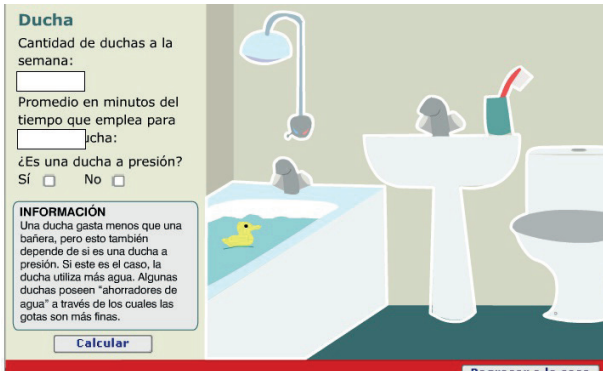


***Figura 5:*** *Interfaces gráficas*


***Figura 6:*** *Introducción de datos*

Luego de recorrer toda la casa o las diferentes opciones, se selecciona calcular gasto de agua.

Posteriormente se obtiene el resultado total por sector en un grafico.


***Figura 7:*** *Resultados finales*

Las herramientas que se describen permitirán que el usuario pueda obtener información oportuna para cuidar el consumo del agua en sus hogares, ahorrar el agua y también cuidar el aspecto económico a la hora de pagar el servicio a la entidad reguladora del agua.

## 4.2: Modelo de un control automatizado para cuidar el uso del agua en los hogares

En los hogares existe una variedad de artefactos

tales como grifos, duchas y bañeras, lavarropas, cocinas y otros que utilizan el agua. Una de las tareas debería ser optimizar el control del tiempo de manera racional en todos estos artefactos y así ahorrar el consumo del agua.

**4.2.1: Caso de Estudio:**

**Proyecto con material reciclado que realiza la simulación del funcionamiento de un cuarto de baño.**

A continuación, se presenta como caso de estudio un proyecto que fue presentado por estudiantes del colegio San Bernardo de Tarija, en las olimpiadas plurinacionales con la temática "cuidando el agua", con la implementación de materiales reciclados y programación en arduino para el control automático.

El objetivo del proyecto es ahorrar el consumo del agua en los hogares, esencialmente en el cuarto de baño, ya que es el sitio donde más agua se consume, en la ducha, lavamos, inodoro y otros.

El proyecto propone una solución para optimizar el uso del agua en la ducha, reutilizando la misma para el uso del inodoro, a la vez de concientizar al consumidor de agua informando la cantidad de agua que usa y el costo del agua consumida. De manera adicional se propone ahorrar el consumo de agua mediante captación de agua de lluvia para ser usado en actividades como lavado de autos, ventanas, regado de jardines y otros.

En resumen las soluciones propuestas son:

- Reutilizar el agua que se consume a diario en lavamanos y la ducha, para ello se debe crear un contenedor de agua bajo el piso, con las instalaciones de tuberías para recolectar el agua, el agua debe ser dirigida posteriormente al inodoro.
- Controlar fugas de agua, con la programación de arduino, se detecta una fuga de agua se activa una alarma y se controla la cantidad de agua que se ha desperdiciado y el costo aproximado de la fuga de agua.
- Implementación de una ducha inteligente, que controla el tiempo de uso del agua en la ducha, la cantidad de agua consumida y el costo del agua utilizada.
- De manera adicional se propone ahorrar agua, captando agua de la lluvia para que sea utilizado en lavado de autos, regado de plantas, etc.

**4.2.1.1. Materiales Utilizados en el proyecto**

- Venesta para la maqueta
- Mangueras para el paso de agua
- Bidones reciclados para recipientes de agua.
- Ducha pequeña
- Material plástico para cortar y formar el baño
- Recipiente transparente para recolectar el agua de la ducha e inodoro.
- Bañera de juguete o plástico para representar la tina de baño.
- Recipiente plástico para representar el uso del inodoro
- Bomba de agua para llevar el agua recolectado al inodoro
- Arduino para controlar el uso en ml del agua y costo.
- Lcd, teclado,valvula,flujómetro.


***Figura 8:*** *Maqueta con material reciclado que realiza la simulación del funcionamiento de un cuarto de baño*

**4.2.1.2. Descripción de la maqueta:**

Como se observa en la maqueta realizada con materiales reciclados, los bidones plásticos representan contenedores de agua, al lado derecho se encuentra un contenedor de agua para representar el lavamanos, el agua que se usa en el lavamanos es dirigida a un contenedor de agua que se encuentra instalado debajo de la tina de agua. Asimismo el Agua que se consume de la ducha también es dirigida al contenedor.

En la parte superior al lado izquierdo de la ducha se encuentra instalado un lcd o pantalla y un teclado, al lado derecho de la ducha se encuentra instalado un flujómetro (que controla la cantidad de agua que se usa) y una electroválvula (que abre y cierra el paso del agua).

En el proyecto cuando el usuario desea utilizar la ducha tiene las opciones de encender la ducha con el teclado o de programar el tiempo que desea utilizar la ducha, si por ejemplo desea utilizar 10 minutos, la pantalla muestra el tiempo transcurrido y al finalizar presenta una alarma que se apagará la ducha. Una vez concluido el uso de la ducha, en la pantalla se observa la cantidad de agua consumida y el costo del consumo de agua.

Si existiese una fuga de agua, se presenta una alarma con luces en rojo y un pitido (sonido del buzer) para informar al usuario que existe fuga de agua, el usuario puede observar en la pantalla la cantidad de agua que se ha desperdiciado y el costo del agua que ha fugado.

El agua utilizada tanto del lavamanos o de la ducha puede ser reutilizada en el inodoro.

De manera adicional el proyecto propone captar agua de la lluvia que llega al techo, dirigirlas por tuberías a un contenedor que realiza una filtración de agua para ser usada de acuerdo a requerimiento como ser lavado de autos, regado de plantas, etc.

**4.2.1.3. Conclusiones:**

Analizando el proyecto, es posible afirmar que la implementación del control automático en los hogares significaría para el usuario una manera de controlar el uso del agua, es posible implementarlo no solo en la ducha, sino también en todas las instalaciones de una casa. Esto permitirá ahorrar el uso del agua y también un bajo costo económico a la hora de pagar la tarifa por el uso del agua.

## 5. BIBLIOGRAFÍA O REFERENCIAS

1. Eloida Almirón, http://www.observatoriomercosur.org.uy/libro/el_agua_como_elemento_vital_en_el_desarrollo_del_hombre_17.php. "El agua como elemento vital en el desarrollo del hombre"
2. http://www.monografias.com/trabajos30/ahorro-agua/ahorro-agua.shtml , Sistema de ahorro de agua doméstico con equipos electrónicos
3. http://portal.esval.cl/oficina-virtual/, ESVAL

4. Felipe Torres B http://reddebibliotecas.org.co/diario/todo-por-el-agua-experiencias-para-la-consciencia-cuidado-proteccion.

5. Todo por el agua: experiencias para la consciencia, el cuidado y la protección del líquido vital.

6. Josep Fernandez Daroca(2012) http://deeea.urv.cat/public/PROPOSTES/pub/pdf/1821pub.pdf "Ejemplo de aplicación con Arduino: medida de caudal".

7. https://es.wikipedia.org/wiki/Uso_racional_del_agua, Uso racional del agua.

# PROGRAMAS PERSISTENTES E HIPERPROGRAMACIÓN

**Jalil Angulo Raquel Ivonné[1], Cortez Michel Fernando Erick[2]**
[1,2]Universidad Autónoma Juan Misael Saracho
Tarija - Bolivia

**Correo electrónico:** jalil.raquel@gmail.com, fecortez@uajms.edu.bo

## RESUMEN

Existen programas que se representan tradicionalmente como secuencias lineales de texto, estos programas se construyen y se almacenan en dispositivos de almacenamiento a largo plazo, tales como un sistema de ficheros, separados del entorno de ejecución que desaparece al final de cada "corrida" del programa. Por el contrario, en sistemas persistentes, los programas se pueden construir y almacenar en el mismo entorno en el que se ejecuten. Esto significa que los objetos a los cuales tiene acceso un programa pueden ya estar disponibles en el momento en que se desarrolle el programa. En este caso los enlaces (links) directos a los objetos se pueden incluir en el programa en vez de descripciones textuales. Un programa que contiene tanto el texto y los enlaces (links) a los objetos, es un hiperprograma.

Este artículo es el resultado de un estudio sobre los hiperprogramas y los lenguajes que los usan en el contexto actual de la tecnología de objetos como son: Napier 88 (ID:2267/napp002), HyperPascal (ID:5108/hyp005), PJama (ID:6912/), Aplicaciones de Hiperprogramación en la Web: HyperWeb, sus ventajas, desventajas y relaciones.

## PALABRAS CLAVE

Hiperprogramación, hiperprograma, persistencia, persistencia ortogonal.

## 1. HIPERPROGRAMACIÓN

En sistemas persistentes, los programas se pueden construir y almacenar en el mismo entorno de ejecución. Esto significa que los objetos a los cuales accede un programa pueden ya estar disponibles en el momento que el programa se desarrolle. En este caso los enlaces (links) directos a los objetos se pueden incluir en el programa en vez de descripciones textuales. Un programa que contiene el texto y enlaces (links) a los objetos es un hiperprograma [1].

Un ejemplo de un hiperprograma [2] se demuestra en la figura 1. Los enlaces incorporados en él son representados por el símbolo no-textual para permitir que se distingan del texto circundante. El primer enlace apunta a un valor del procedimiento de primera clase writeString el cual escribe un aviso al usuario. El programa entonces llama otro procedimiento readString para leer un nombre, y después encuentra una dirección que corresponda a ese nombre. Esta operación se logra llamando a un procedimiento lookup que busca la dirección dentro de una estructura de datos tipo tabla que se encuentra enlazada dentro del hiperprograma. Posteriormente, se escribe la dirección buscada. Observe que los objetos del código (readString, writeString y lookup): están denotados usando exactamente el mismo mecanismo que los objetos de datos (la tabla). Observe también que los nombres del objeto usados en esta descripción

se han asociado a los objetos, simplemente por claridad, y no son parte de la semántica del hiperprograma [4],[5],[6].

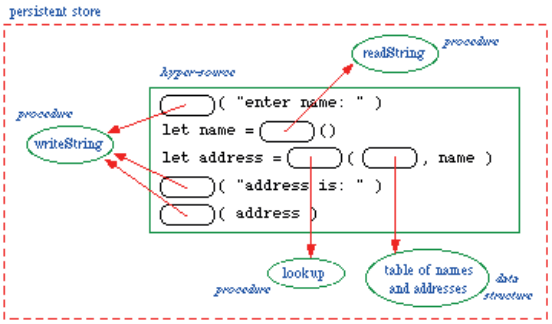


***Figura 1.*** *Ejemplo de un hiperprograma [2]*

## 2. LENGUAJES

El primer lenguaje que soporta hiperprogramación es **Napier88**[7] de tradición ALGOL al igual que su S-ALGOL de los precursores y Picosegundo-ALGOL [8].

En los sistemas de programación persistentes, como es Napier, el programador no precisa crear procedimientos específicos para guardar y recuperar las estructuras de datos que utiliza en sus programas, ya que este proceso se realiza de forma totalmente transparente por el propio sistema. Los sistemas de programación persistentes de la actualidad se basan ampliamente en los tres principios de ortogonalidad, que intentan asegurar que el usuario se abstraiga totalmente del hecho de utilizar objetos persistentes o no persistentes, del proceso de salvaguarda o recuperación de los mismos y de dónde residen esos objetos en cada momento.

Estos principios establecen que un sistema posee Persistencia Ortogonal (Atkinson & Morrison, 1995) si cumple las siguientes condiciones[9], [10]:

a) El trato del programa a objetos persistentes y no persistentes es indistinto. (Ortogonalidad al trato).

b) Cualquier objeto debe poder ser persistente, independientemente de su tipo o clase. (Ortogonalidad al tipo).

c) Todos los objetos relacionados con un objeto persistente deben ser persistentes, cumpliendo éstos a su vez con las condiciones a) y b). (Ortogonalidad de Identificación ó transitividad de identificación persistencia).

Los sistemas de programación persistentes que cumplen los criterios de ortogonalidad, como por ejemplo Napier (Morrison et al., 1999) o Pjama (Jordan & Atkinson, 1998), se basan en su mayor parte en una caché que actúa como intermediaria entre el mecanismo gestor de objetos y el almacenamiento persistente. De esta forma, cuando el programa precisa la utilización de un objeto, el gestor de objetos comprueba primero que no resida ya en la caché, antes de solicitar su traslado desde el almacenamiento persistente. El hecho de no encontrar el objeto buscado en la caché se denomina "fallo de objeto", y obliga a recuperar ese objeto del almacenamiento persistente [11], [12].

El sistema Napier88 se diseña como arquitectura acodada [Bro89] [13] que consiste en un recopilado [14,15,16,17], la máquina abstracta persistente (PAM) [18,19] y la arquitectura persistente del almacenaje [13, 20,21]. Todas las capas arquitectónicas de Napier88 son virtuales, en cualquier puesta en práctica, ellas se pueden poner en ejecución por separado o juntas dependiendo de la eficacia.

La versión actual del lenguaje Napier88 es la versión 2.2.1[22]. El lenguaje tiene solamente

algunos cambios a el del lanzamiento 1,0 [23,24] pero el ambiente persistente se ha enriquecido y se ha reorganizado perceptiblemente. Los cambios al lenguaje son: un modelo abstracto dinámico del testigo para los tipos abstractos, y tipos de operadores.

Los cambios principales al ambiente persistente son la disposición de un browser, un compilador para la programación reflexiva, hilos de rosca y los semáforos, una nueva organización del almacén de objetos para proporcionar una navegación libre en el almacén, almacenes distribuidos con la exploración alejada y un sistema de hiperprogramación. El ambiente también proporciona un mecanismo, a través de Internet, para otros sitios para compartir los programas y los datos que se pueden alcanzar por la exploración y la copia, separada de otros repositorios Napier88.

Por otro lado **HyperPascal** es un nuevo lenguaje visual revolucionario que ha demostrado con éxito la viabilidad de la idea de hiperprogramación. Sus capacidades de manipulación de datos se modelan en PASCAL (ISO), pero el uso de las marcas de sintaxis usan capacidades actuales del GUI (Interfaz Gráfica de Usuario). Sin embargo, como prototipo, tiene limitaciones. Por ejemplo, no proporciona ninguna ayuda de funcionamiento, y el ambiente lingüístico es anticuado (el PASCAL fue elegido como caso de prueba para las ideas de hiperprogramación por su simplicidad, no por su modernidad). En este proyecto, usted desarrollaría en un ambiente completamente compilado para HyperPascal, con la ayuda de un run-time (entorno de ejecución), las instalaciones separadas de la compilación, el acceso a las rutinas externas de la biblioteca, el uso de un instrumento de sistema más actualizado con características lingüísticas más actualizadas, tales como orientación a objetos.

Otro lenguaje que permite hiperprogramación es **PJama** que es un prototipo experimental que pone la persistencia ortogonal en ejecución para la plataforma de Java (OPJ), que es un acercamiento a hacer objetos del uso persistente entre las ejecuciones de programa con el esfuerzo mínimo posible requerido.

Requisitos para proporcionar hiperprogramación en Java:

- un repositorio persistente con root(s), disponibilidad e integridad referencial.
- reflexión lingüística como técnica de programación.
- un browser del repositorio persistente.
- una especificación de hyper-links escrito en Java.
- un mecanismo para preservar hyper-links sobre la compilación.
- un editor de hiperprogramación.
- Una interfaz gráfica de usuario.

La impresión de hiperprogramas y de la transferencia de hiperprogramas a partir de un sistema a otro es obstaculizada por la presencia de hyper-links. Es, sin embargo, posible traducir cada hyperprogram al HTML, representando los hyper-links como URLs. Esto fue hecho para publicar la fuente del compilador Napier88, que es en sí mismo un hiperprograma, y es lo que hace también Java.

**HyperWeb** es una aplicación web para la construcción de aplicaciones web. Las aplicaciones web se construyen vía un interfaz del alto nivel, o la representación del hipercódigo. Por razones pragmáticas, esta representación del hipercódigo hace las páginas web en una manera similar a la que se mostrará en el web browser del usuario. Cada elemento se aumenta con los componentes adicionales que proporcionan el hipercódigo que

corrige funcionalidad.

Las aplicaciones web se construyen con los objetos que se crean y se almacenan en un servidor persistente que proporciona seguridad de tipo y las características de integridad referencial en estos objetos en tiempo de ejecución, y continuamente mantenerlos durante el ciclo de vida de la aplicación. En contraste, la herramienta de lado del cliente no puede asegurarse de que los objetos no sean suprimidos, sean movidos, o no substituidos por los objetos de diverso tipo una vez que la aplicación sea desplegada o en la evolución subsecuente (o concurrente) de la aplicación.

HyperWeb proporciona la primera prueba de la existencia del uso del concepto de hiperpogramación para la construcción de aplicaciones web. La interfaz de HyperWeb se genera enteramente dentro de los límites del HTTP estándar y es por lo tanto accesible vía cualquier web browser. HyperWeb es una aproximación del lado del servidor en el cual las aplicaciones web se construyen fuera de los objetos que se crean y se mantienen en un servidor persistente. Como tal, las garantías estáticas de la seguridad del programa, se proporcionan y se mantienen durante el ciclo de vida de la aplicación.

Los acercamientos más actuales a la seguridad del desarrollo web están basados en el cliente. Pocos acercamientos existentes han adoptado un modelo del lado del servidor, y por lo tanto las ventajas obtenidas de seguridad del lado del servidor no se alcanzan. Los sistemas relevantes con metas similares al de HyperWeb incluyen el < bigwig > y el sistema de WÓbjects.

## 3. CONCLUSIONES

El concepto de hiperprogramación transfiere de un lenguaje persistente polimórfico común a un lenguaje persistente polimórfico orientado a objetos.

La hiperprogramación ofrece varias ventajas referidas al rendimiento y la evolución natural que apunta a un entorno orientado a objetos.

El término "Persistente" se aplica a los sistemas que proporcionan un mecanismo transparente y automático (en mayor o menor medida) de salvaguarda y recuperación de objetos. Entre estos sistemas se encuentran sistemas operativos orientados a objetos, sistemas gestores de bases de datos orientados a objetos, y sistemas de programación orientados a objetos.

La hiperprogramación proporciona un nuevo estilo para enlazar programas a datos persistentes, permitiendo que los acoplamientos directos a los datos sean incorporados en los programas fuente ejecutados en un repositorio persistente. Se destacan ventajas tales como: en detalle permite una composición más apropiada de programas y permite enlaces entre los programas ejecutables y el código fuente, en contraste con los sistemas tradicionales que confían en la convención. La hiperprogramación permite que un código fuente contenga enlaces a los artículos de datos en el repositorio persistente. Esto mejora la confiabilidad del programa. También reduce la cantidad de código escrita permitiendo enlaces directos a los datos en el lugar de descripciones textuales. Ambas técnicas contribuyen a la comprensión del ambiente persistente con el soporte de la puesta en práctica de las herramientas del navegador del almacén y permiten que las representaciones del código fuente sean asociadas a todos los programas

ejecutables en el repositorio persistente.

## BIBLIOGRAFÍA O REFERENCIAS


1. http://www-systems.cs.st-andrews.ac.uk/wiki/Hyper_Programming/.
2. http://www-systems.cs.st-andrews.ac.uk/wiki/Persistence/.
3. Kirby, G.N.C., Connor, R.C.H., Cutts, Q.I., Dearle,A., Farkas,A.M. & Morrison, R. "Persistent Hyper-Programs". In Proc. 5th International Workshop on Persistent ObjectSystems, San Miniato, Italy (1992).
4. Atkinson MP, Morrison R, Pratten GD. Designing a Persistent Information Space Architecture. In: Proc. 10th IFIP World Congress, Dublin, 1986, pp 115-20.
5. Connor RCH. The Napier Type-Checking Module. Universities of Glasgow and St Andrews Report PPRR-58-88, 1988.
6. PS-algol Reference Manual, 4th edition Universities of Glasgow and St Andrews Technical Report PPRR-12-88 (1988).
7. Polimorfismo, reutilización de la persistencia y del software en un ambiente orientado al objeto fuertemente mecanografiado Morrison, R., marrón, A.L., Connor, R.C.H. y Dearle, A. Tecnología de dotación lógica Journal, diciembre (1987) pp 199-204.
8. Tipos, atascamientos y parámetros en un ambiente persistente Atkinson, M.P. y Morrison, R. En tipos y persistencia de datos, Atkinson, M.P., Buneman, O.P. y Morrison, R. (ed), Springer-Verlag (1988) pp 3-20.
9. El rendimiento en Sistemas de Programación Persistentes. J. Baltasar García Perez-Schofield1, Tim B. Cooper2, Emilio García Roselló3, Manuel Pérez Cota3 1Escuela Superior de Ingeniería Informática. Departamento de Informática. Universidad de Vigo [ Bro89 ] El Objeto Persistente Almacena El Marrón, A.L. Ph.D. Tesis, Universidad de St Andrews (1989).
10. En la construcción de los ambientes de programación persistentes Dearle, A. Ph.D. Tesis, universidad de St Andrews (1988).
11. Tipos y polimorfismo en los sistemas de programación persistentes Connor, R.C.H. Ph.D. Tesis, universidad de St Andrews (1990).
12. Entregar las ventajas de la persistencia a la construcción y a la ejecución Cutts del sistema, Q.I. Ph.D. Tesis, universidad de St Andrews (1992).
13. Reflexión e Hiperactivo-Programacio'n en los sistemas de programación persistentes Kirby, G.N.C. Ph.D. Tesis, universidad de St Andrews (1992).
14. El Marrón De la Máquina, el A.L., El Carrick, El R., El Connor, el R.C.H., El Dearle, El A. Y El Morrison Abstractos Persistentes, R. Universidades de Glasgow y del informe técnico Pprr-59-88 (1988) del St Andrews.
15. La Máquina Abstracta Persistente Connor, R.C.H., Marrón, A.L., Carrick, R., Dearle, A. Y Morrison, R. En sistemas persistentes, Rosenberg, el J. y Koch del objeto, D.M. (ed), Springer-Verlag (1990) pp 353-366.
16. On the Integration of Concurrency, Distribution and Persistence Munro, D.S. Ph.D. Thesis, University of St Andrews (1993).
17. The Napier88 Reference Manual (Release 2.0) Morrison, R., Brown, A.L., Connor, R.C.H.,

Cutts, Q.I., Dearle, A., Kirby, G.N.C. & Munro, D.S. University of St Andrews Technical Report CS/94/8 (1994).

18. El Manual De Referencia Napier88 Morrison, R., Marrón, A.L., Connor, R.C.H. Y Dearle, A. Universidades de Glasgow y del informe técnico Pprr-77-89 (1989) del St Andrews.

19. Napier88 Release 1.0 Morrison, R., Brown, A.L., Connor, R.C.H. & Dearle, A. University of St Andrews (1989).

20. http://www.cs.berkeley.edu/~benr/publications/auscc04/papers/fox-auscc04.pdf.

21. http://www.naccq.ac.nz/conference05/proceedings_05/posters/349.pdf.

22. http://hopl.murdoch.edu.au/showlanguage2.prx?exp=5108.

23. http://www.naccq.ac.nz/conference04/proceedings_03/pdf/231.pdf

24. p.lyons@massey.ac.nz (investigadores de hyperpascal).

# TECNOLOGIAS WEB PARA INTERNET DE LAS COSAS (INTERNET OF THINGS)

**Octavio Douglas Aguilar Mallea**
Universidad Autónoma Juan Misael Saracho
Tarija - Bolivia

**Correo electrónico:** octavioa111@gmail.com

## RESUMEN

Una preocupación por parte de los profesionales en informática es la integración de la tecnología digital con artefactos físicos, es decir comunicar la tecnología y los dispositivos ha sido el reto de muchos investigadores. Recientemente han llegado a desarrollar artefactos inteligentes que cada vez es de uso cotidiano en nuestras vidas, como los televisores inteligentes, despertadores, refrigeradores, movilidades con cierta inteligencia, todo esto ha hecho que nuestras actividades sean más sencillo porque a través de estos dispositivos inteligentes podemos mantenernos más informados de ciertos eventos que están ocurriendo en nuestro entorno. Como consecuencia de las investigaciones realizadas sobre Internet de las cosas (Internet of Things (IoT)), se están explorando nuevas formas de conectar dispositivos inteligentes sobre una red de computadoras y para facilitar estas conexiones, en los últimos años han surgido una serie de protocolos de red pero aún siguen siendo difíciles de integrar en aplicaciones compuestas.

Las tecnologías de comunicación web tradicionales, como el protocolo de transferencia de Hipertexto (HTTP), proporcionan un enlace unidireccional y un modelo de intercambio de mensajes de solicitud/respuesta. Esta solución puede ser problemática en aplicaciones basadas en web involucrando una gran cantidad de diferentes dispositivos, como en Internet of Things (IoT).

En este artículo se muestran diferentes modelos adecuados para aplicaciones IoT basadas en las web escalables y de baja latencia. Además se evalúan diferentes tecnologías que permitirán manejar ideas sobre el desarrollo de aplicaciones basados en la web.

## PALABRAS CLAVE

Internet of Things (IoT), Web of Things (WoT), Internet de Todo (IoE), Hipertexto,navegadores web, REST, IPv6.

## INTRODUCCIÓN

Después que la web pasa de la versión 1.0 a 2.0 donde la red como plataforma, abarca todos los dispositivos conectados. A diferencia de la Web 1.0, las aplicaciones se tratan como servicios continuamente actualizados que mejoran a medida que la gente los usa. Esta se realiza mediante el consumo y la fusión de datos de múltiples fuentes, incluidos los usuarios individuales, al tiempo que proporciona sus propios datos y servicios en una forma que permite la fusión por parte de terceros.

Actualmente los navegadores Web se instalan comúnmente en equipos electrónicos inteligentes como computadoras, teléfonos inteligentes, tablets, etc. que se pueden conectar a la Internet. Los navegadores pueden interpretar código HTML presentando textos y multimedia como imágenes, audio y video, siendo que estos últimos se

almacenan en otras computadoras conectadas a Internet y son descargados por el navegador en la computadora local.

El trabajo presentado inicia justo en este punto cuando se analiza como el navegador se comunica directamente con un dispositivo habilitado para Internet y para que el trabajo sea interesante, los dispositivos debe tener una buena interoperabilidad, es decir, deben poder comunicarse con sensores o placas arduino.

La usabilidad es una de las importantes características de la Web 2.0. Las páginas web deben ser sencillas de usar, donde se deben aprovechar el uso de componentes multimedia de manera adecuada de tal forma que los usuarios se sientan cómodo al visitar la página.

Con el Objetivo de presentar contenido enriquecido a los usuarios, un navegador web se comunica con un servidor web para buscar los datos. La interacción entre un navegador y un servidor web se basa tradicionalmente en solicitudes HTTP., es decir, un navegador envía solicitudes HTTP a un servidor web, especificando los datos deseados; el servidor web, responde al navegador con los datos solicitados, si está disponible. En el nivel de aplicación en el lado del navegador, una solicitud HTTP. generalmente se envía mediante el uso de técnicas como por ejemplo AJAX, los datos se recuperan de un servidor remoto enviando una solicitud HTTP de forma asíncrona.

Sería un gran avance conectar todos los dispositivos a la Internet e ingresar a través de la web, sería uno de los pasos importantes que se daría para la Internet de la Cosas. La Web de las Cosas es simplemente el siguiente paso de la evolución usando y adaptando protocolos web para conectar cada cosa o dispositivo y tener acceso a la misma por medio del protocolo HTTP.

## INTERNET DE LAS COSAS (INTERNET OF THINGS (IOT))

Internet de las Cosas (IoT) representa la próxima evolución de Internet, que será un enorme salto en su capacidad para reunir, analizar y distribuir datos que podemos convertir en información, conocimiento y en última instancia, sabiduría.

Internet de la Cosas (IoT) es una extensión de la actual Internet que posibilita las conexiones y comunicaciones entre objetos y dispositivos físicos. Las estimaciones nos indican que para el 2020 habrá 50 mil millones de dispositivos y personas conectadas y potenciando la visión de la tecnología que posibilita la IoT. Un término relacionado que actualmente está de moda es el de La Internet de Todo (IoE Internet of Everything), que se define como "la conexión inteligente de personas, procesos, datos y cosas", este concepto de la IoE es más expansivo que el concepto de la IoT, ya que la IoT además de incluir las comunicaciones de máquina a máquina, incluye también las interacciones de máquina a personas y las de persona a persona asistida por tecnología.

En la tecnología los avances y la evolución marcan hitos representativos. Actualmente la IoT está compuesta por una colección dispersa de redes diferentes y con distintos fines. Por ejemplo, las movilidades actuales tienen múltiples redes para controlar el funcionamiento del motor, las medida de seguridad, los sistemas de comunicación y sucesivamente. De forma similar, los edificios comerciales y residencias tienen distintos sistemas de control para la calefacción, la ventilación y el aire acondicionado, la telefonía, la seguridad y la iluminación. A medida que la IoT evoluciona, estas redes y muchas otras estarán conectadas con la

incorporación de capacidades de seguridad, análisis y administración.

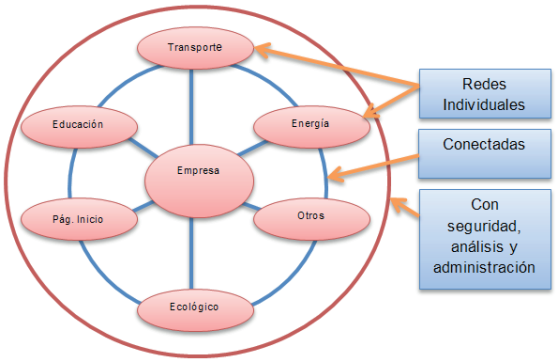


***Figura 1.*** *Internet de la Cosas (IoT)*

## WEB DE LA COSAS (WEB OF THINGS (WOT))

La Web de las Cosas (WoT) es un término que permite describir los objetos que nos rodean pueden recibir y emitir información, relacionares entre ellos, con sistemas informáticos y con las personas, de forma que los datos se puedan agregar y a través del análisis de los mismos, poder realizar tomas de decisiones acertadas y establecer predicciones. Esta tecnología permite el intercambio de información entre diferentes dispositivos por medio del protocolo HTTP.

Los recursos WoT pueden ser ubicuos y quedan a menudo limitados por la potencia disponible, la memoria, la capacidad de procesamiento, el acceso, la disponibilidad, la capacidad de atención y las capacidades de comunicación. El carácter heterogéneo la ubicuidad y la naturaleza dinámica de los recursos junto con el amplio espectro de los datos hacen que la tarea de descubrir, acceder, procesar e integrar e interpretar los datos de la Web sea desafiante. Un ciber componente rico de la WoT incluye datos Web residentes, conocimiento, (como lo son por ejemplo Wikipedia o los Datos Abiertos Enlazados), información intercambiada en medios sociales, (tales como los sitios en donde los pacientes comparten información relacionada con la salud) y mediciones y observaciones del mundo real que son suministradas por los por los usuarios. La integración los recursos físicos, cibernéticos y sociales.

Ejemplos de tales aplicaciones se encuentran desde la salud personalizada, el ejercicio personalizado y el bienestar físico hasta la energía y una amplia variedad de actividades de negocios e industriales.

## COMPARACIÓN ENTRE WEB DE LAS COSAS VS INTERNET DE LAS COSAS)

La Web de las Cosas es un enfoque que esta embebido dentro de la Internet de las Cosas, sin embargo en la Figura 2. se listan algunas principales diferencias:

| Internet de la Cosas | Web de las Cosas |
|---|---|
| Comunicación sobre la capa de Red | Comunicación sobre la capa de Aplicación HTTP |
| RFID, Bluetooth,CoAP | HTTP, REST, HTML5,JS,Web |
| Alto acoplamiento | Bajo acoplamiento |

***Figura 2.*** *Comparación IoT vs WoT.*

En la figura 3. se puede apreciar que en la IoT existen cientos de protocolos incompatibles que coexisten entre ellos y esto lleva a que la integración entre los diferentes dispositivos sea extremadamente complejos y de alto costo. Por otro lado, la WoT puede ser accedido desde el protocolo de la Web, donde la se conectan diferentes dispositivos de una manera más fácil por medio del protocolo HTTP.

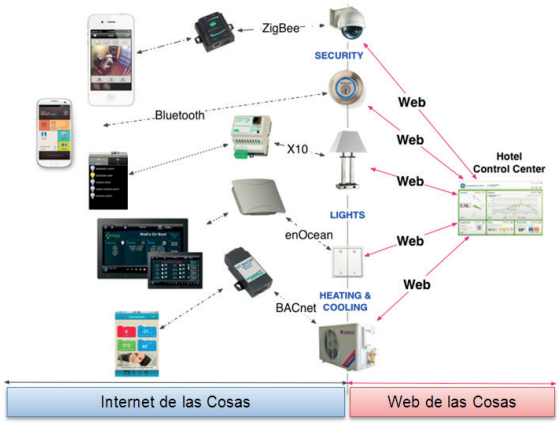


***Figura 3.*** *Diferencias IoT vs WoT.*

## IPV6: EL MOTOR DE LA WEB DE LAS COSAS

Como es de conocimiento de todos, la tecnología avanza a pasos agigantados y que las redes crecen también en cantidad y es por eso que la IPv6 está llamado a convertirse en una parte esencial del paradigma de la "Web de la Cosas".

Si se analiza detenidamente muchas de las soluciones IoT existen actualmente, por lo tanto concluiremos que, en sentido estricto, los dispositivos no son realmente parte de Internet. Por ejemplo, no tienen dirección IP.

Además, los servicios WEB no se ofrecen desde los mismos dispositivos, sino desde pasarelas intermedias (gateways) o plataformas. Estas últimas con las que realmente están presentes en Internet y ofrecen servicios a los usuarios.

Por último, al no ser la comunicación entre los dispositivos y las pasarelas un estándar de Internet, han proliferado los protocolos y soluciones propietarias. En muchos casos, están protegidos por patentes y suponen en definitiva el pago para el desarrollo de nuevas implementaciones.

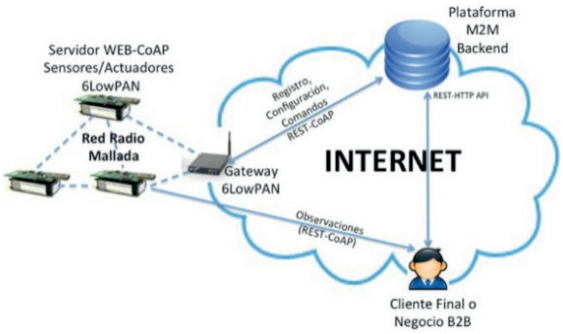


***Figura 4.*** *Uso de la IPv6 en la Internet*

En la Figura 4. Se describe como cierta entidad y alimentados mediante la red eléctrica es posible utilizar TCP/IP sobre redes WiFi o 3G, y que el dispositivo implemente los conocidos servicios WEB de tipo REST para su manejo.

## API REST

REST deriva de "REpresentational State Transfer", que traducido vendría a ser "transferencia de representación de estado", lo que tampoco aclara mucho, pero contiene la clave de lo que significa. Porque la clave de REST es que un servicio REST no tiene estado (es stateless), lo que quiere decir que, entre dos llamadas cualesquiera, el servicio pierde todos sus datos. Esto es, que no se puede llamar a un servicio REST y pasarle unos datos (p. ej. un usuario y una contraseña) y esperar que "nos recuerde" en la siguiente petición. De ahí el nombre: el estado lo mantiene el cliente y por lo tanto es el cliente quien debe pasar el estado en cada llamada. Si quiero que un servicio REST me recuerde, debo pasarle quien soy en cada llamada. Eso puede ser un usuario y una contraseña, un token o cualquier otro tipo de credenciales, pero debo pasarlas en cada llamada. Y lo mismo aplica para el resto de información.

REST es una arquitectura para desarrollo de aplicaciones distribuidas. La esencia de REST es la creación de un bajo acoplamiento de servicios

que pueden ser fácilmente reusados, lo cual es implementado usando URIs, HTTP, etc.

Una de las características principales de los servicios web REST es el uso explícito de los métodos HTTP, siguiendo el protocolo definido por RFC 2616. Por ejemplo, HTTP GET se define como un método productor de datos, cuyo uso está pensado para que las aplicaciones cliente obtengan recursos, busquen datos de un servidor web, o ejecuten una consulta esperando que el servidor web la realice y devuelva un conjunto de recursos.

REST hace que los desarrolladores usen los métodos HTTP explícitamente de manera que resulte consistente con la definición del protocolo. Este principio de diseño básico establece una asociación uno-a-uno entre las operaciones de crear, leer, actualizar y borrar y los métodos HTTP.

De acuerdo a esta asociación:

- Se usa POST para crear un recurso en el servidor
- Se usa GET para obtener un recurso
- Se usa PUT para cambiar el estado de un recurso o actualizarlo
- Se usa DELETE para eleminar un recurso

En la Figura 5. muestra la interacción usando los métodos HTTP.

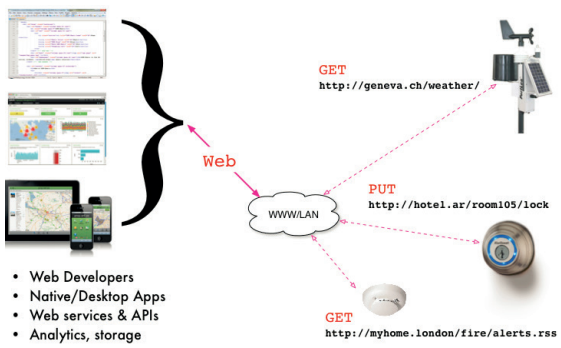


***Figura 5.*** *Métodos HTTP.*

En la Práctica, esto significa que se puede lograr la interacción con cosas por medio de un navegador Web, con solo escribir el Enlace (link) se podrá mantener interacción con las cosas en tiempo real.

En la Figura 5. Se puede observar el uso de los métodos HTTP y como la Web de la Cosas permite desarrollar aplicaciones para el intercambio de datos con dispositivos usando solicitudes HTTP.

## CONCLUSIONES

En conclusión, la Internet de la Cosas está más cerca de ser implementado. La mayoría de los avances tecnológicos necesarios para ello ya se realizaron y algunos fabricantes y agencia ya comenzaron a implementar una versión a pequeña escala de la misma. Las razones principales por las que no se ha implementado realmente es el impacto que tendrá en los campos legal, ético, de seguridad y social. Los trabajadores podrían abusar de él, los hackers podrían acceder a él, las empresas pueden no querer compartir sus datos y puede que las personas no les guste la total ausencia de privacidad. Por estas razones, es muy posible que la Internet de las Cosas retroceda más de lo necesario.

Es importante considerar que la Web de las Cosas tiene algunas ventajas que las hace más simple y flexible en comparación que la Internet de las Cosas, ya que se podrían integrar muchos dispositivos fácilmente por medio de la capa de aplicación.

## REFERENCIAS

Eleonora , B. (2014). The Internet of Things vision: Key features, applications and open issues. SciVerse ScienceDirect, 1-31.

Peter Lubbers, Brian Albers, and Frank Salim. Pro HTML 5 Programming. Apress, March 2010.

Mark Lentczner and Donovan Preston. Reverse HTTP. Technical report, IETF, March 2009.

Roland Kubert, Gregory Katsaros, and Tinghe Wang. A RESTful implementation

of the WS-agreement speci_cation. In Proc. of the Second International Workshop on RESTful Design (WS-REST '11), WS-REST '11, pages 67{72, New York, NY, USA, 2011. ACM.

Erik Wilde. Putting Things to REST. Technical report, School of Information, UC Berkeley, November 2007.

Comer, D. E. Computer networks and internets Prentice Hall Press, 2008.

# NORMAS DE PUBLICACIÓN DE LA REVISTA bit@bit

## Misión y Política Editorial

La Revista bit@bit, es una publicación semestral que realiza la Universidad Autónoma Juan Misael Saracho que tiene como misión, difundir la producción de conocimientos de la comunidad universitaria, académica y científica del ámbito local, nacional e internacional, provenientes de investigaciones que se realiza en las distintas áreas del conocimiento, para contribuir a lograr una apropiación social del conocimiento por parte de la sociedad.

bit@bit es una publicación arbitrada que utiliza el sistema de revisión por al menos de dos pares expertos (académicos internos y externos) de reconocido prestigio, pudiendo ser nacionales y/o internacionales, que en función de las normas de publicación establecidas procederán a la aprobación de los trabajos presentados. Asimismo, la revista se rige por principios de ética y pluralidad, para garantizar la mayor difusión de los trabajos publicados.

La revista bit@bit publica artículos en castellano, buscando fomentar la apropiación social del conocimiento por parte de la población en general.

Tanto los autores, revisores, editores, personal de la revista y académicos de la Universidad Autónoma Juan Misael Saracho, tienen la obligación de declarar cualquier tipo de conflicto de intereses que pudieran sesgar el trabajo.

## Tipo de Artículos y Publicación

La Revista bit@bit, realiza la publicación de distintos artículos de acuerdo a las siguientes características:

Artículos de investigación científica y tecnológica: Documento que presenta, de manera detallada, los resultados originales de investigaciones concluidas. La estructura generalmente utilizada es la siguiente: introducción, metodología, resultados, Discusión, pudiendo también, si así lo desean, presentar conclusiones.
Artículo de reflexión: Documento que presenta resultados de investigación terminada desde una perspectiva analítica, interpretativa o crítica del autor, sobre un tema específico, recurriendo a fuentes originales.

Artículo de revisión: Documento resultado de una investigación terminada donde se analizan, sistematiza e integran los resultados de investigaciones publicadas o no publicadas, sobre un campo en ciencia o tecnología, con el fin de dar cuenta de los avances y las tendencias de desarrollo. Se caracteriza por presentar una cuidadosa revisión bibliográfica de por lo menos 50 referencias.

Artículos académicos: Documentos que muestren los resultados de la revisión crítica de la literatura sobre un tema en particular, o también versan sobre la parte académica de la actividad docente. Son comunicaciones concretas sobre el asunto a tratar por lo cual su extensión mínima es de 5 páginas.

Cartas al editor: Posiciones críticas, analíticas o interpretativas sobre los documentos publicados en la revista, que a juicio del Comité editorial constituyen un aporte importante a la discusión del tema por parte de la comunidad científica de referencia.

## Normas de Envío y Presentación

**a.** La Revista bit@bit, recibe trabajos originales en idioma español. Los mismos deberán ser remitidos en formato electrónico en un archivo de tipo Word compatible con el sistema Windows y también en forma impresa.

**b.** Los textos deben ser enviados en formato de hoja tamaño carta (ancho 21,59 cm.; alto 27,94 cm.) en dos columnas. El tipo de letra debe ser Arial, 10 dpi interlineado simple. Los márgenes de la página deben ser, para el superior, interior e inferior 2 cm. y el exterior de 1 cm.

**c.** La extensión total de los trabajos para los artículos de investigación, científica y tecnológica tendrán una extensión máxima de 15 páginas,

incluyendo la bibliografía consultada.

**d.** Para su publicación los artículos originales de investigación no deben tener una antigüedad mayor a los 5 años, desde la finalización del trabajo de investigación.

**e.** Para los artículos de reflexión y revisión se tiene una extensión de 10 páginas. En el caso de los textos para los artículos académicos se tiene un mínimo de 5 páginas.

**f.** Los trabajos de investigación (artículos originales) deben incluir un resumen en idioma español y en inglés, de 250 palabras.

**g.** En cuanto a los autores, deben figurar en el trabajo todas las personas que han contribuido sustancialmente en la investigación. El orden de aparición debe corresponderse con el orden de contribución al trabajo, reconociéndose al primero como autor principal. Los nombres y apellidos de todos los autores se deben identificar apropiadamente, así como las instituciones de adscripción (nombre completo, organismo, ciudad y país), dirección y correo electrónico.

**h.** La Revista bit@bit, solo recibe trabajos originales e inéditos, que no hayan sido publicados anteriormente y que no estén siendo simultáneamente considerados en otras publicaciones nacionales e internacionales. Por lo tanto, los artículos deberán estar acompañados de una Carta de Originalidad, firmada por todos los autores, donde certifiquen el original del escrito presentado.

## Dirección de Envío de Artículos

Los artículos para su publicación deberán ser presentados en en secretaría del Departamento de Informática y Sistemas, Campus Universitario El Tejar, Tarija – Bolivia, Tel/Fax 591-46640265 o podrán ser envidados a las siguientes direcciones electrónicas: dis@uajms.edu.bo.

También se debe adjuntar una carta de originalidad impresa y firmada o escaneada en formato PDF.

## Formato de Presentación

Para la presentación de los trabajos se debe tomar en cuenta el siguiente formato para los artículos científicos:

## Título del Artículo

El título del artículo debe ser claro, preciso y sintético, con un texto de 20 palabras como máximo.

## Autores

Un aspecto muy importante en la preparación de un artículo científico, es decidir, acerca de los nombres que deben ser incluidos como autores, y en qué orden. Generalmente, está claro que quién aparece en primer lugar es el autor principal, además es quien asume la responsabilidad intelectual del trabajo. Por este motivo, los artículos para ser publicados en la Revista Investigación y Desarrollo, adoptarán el siguiente formato para mencionar las autorías de los trabajos.

Se debe colocar en primer lugar el nombre del autor principal, investigadores, e investigadores junior, posteriormente los asesores y colaboradores si los hubiera. La forma de indicar los nombres es la siguiente: en primer lugar debe ir los apellidos y posteriormente los nombres, finalmente se escribirá la dirección del Centro o Instituto, Carrera a la que pertenece el autor principal. En el caso de que sean más de seis autores, incluir solamente el autor principal, seguido de la palabra latina "et al", que significa "y otros" y finalmente debe indicarse la dirección electrónica (correo electrónico).

## Resumen y Palabras Clave

El resumen debe dar una idea clara y precisa de la totalidad del trabajo, incluirá los resultados más destacados y las principales conclusiones, asimismo, debe ser lo más informativo posible, de manera que permita al lector identificar el contenido básico del artículo y la relevancia, pertinencia y calidad del trabajo realizado.

Se recomienda elaborar el resumen con un máximo de 250 palabras, el mismo que debe expresar de manera clara los objetivos y el alcance del estudio, justificación, metodología y los principales resultados obtenidos.

En el caso de los artículos originales, tanto el título, el resumen y las palabras clave deben también presentarse en idioma inglés.

## Introducción

La introducción del artículo está destinada a expresar con toda claridad el propósito de la comunicación, además resume el fundamento lógico del estudio. Se debe mencionar las referencias estrictamente pertinentes, sin hacer una revisión extensa del tema investigado.

## Materiales y Métodos

Debe mostrar, en forma organizada y precisa, cómo fueron alcanzados cada uno de los objetivos propuestos.

La metodología debe reflejar la estructura lógica y el rigor científico que ha seguido el proceso de investigación desde la elección de un enfoque metodológico específico (preguntas con hipótesis fundamentadas correspondientes, diseños muestrales o experimentales, etc.), hasta la forma como se analizaron, interpretaron y se presentan los resultados. Deben detallarse, los procedimientos, técnicas, actividades y demás estrategias metodológicas utilizadas para la investigación. Deberá indicarse el proceso que se siguió en la recolección de la información, así como en la organización, sistematización y análisis de los datos. Una metodología vaga o imprecisa no brinda elementos necesarios para corroborar la pertinencia y el impacto de los resultados obtenidos.

## Resultados

Los resultados son la expresión precisa y concreta de lo que se ha obtenido efectivamente al finalizar el proyecto, y son coherentes con la metodología empleada. Debe mostrarse claramente los resultados alcanzados, pudiendo emplear para ello cuadros, figuras, etc.

Los resultados relatan, no interpretan, las observaciones efectuadas con el material y métodos empleados. No deben repetirse en el texto datos expuestos en tablas o figuras, resumir o recalcar sólo las observaciones más importantes.

## Discusión

El autor debe ofrecer sus propias opiniones sobre el tema, se dará énfasis en los aspectos novedosos e importantes del estudio y en las conclusiones que pueden extraerse del mismo. No se repetirán aspectos incluidos en las secciones de Introducción o de Resultados. En esta sección se abordarán las repercusiones de los resultados y sus limitaciones, además de las consecuencias para la investigación en el futuro. Se compararán las observaciones con otros estudios pertinentes. Se relacionarán las conclusiones con los objetivos del estudio, evitando afirmaciones poco fundamentadas y conclusiones avaladas insuficientemente por los datos.

## Bibliografía Utilizada

La bibliografía utilizada, es aquella a la que se hace referencia en el texto, debe ordenarse en orden alfabético y de acuerdo a las normas establecidas para las referencias bibliográficas (Punto 5).

## Tablas y Figuras

Todas las tablas o figuras deben ser referidas en el texto y numeradas consecutivamente con números arábigos, por ejemplo: Figura 1, Figura 2, Tabla 1 y Tabla 2. No se debe utilizar la abreviatura (Tab. o Fig.) para las palabras tabla o figura y no las cite entre paréntesis. De ser posible, ubíquelas en el orden mencionado en el texto, lo más cercano posible a la referencia en el mismo y asegúrese que no repitan los datos que se proporcionen en algún otro lugar del artículo.

El texto y los símbolos deben ser claros, legibles y de dimensiones razonables de acuerdo al tamaño de la tabla o figura. En caso de emplearse en el artículo fotografías y figuras de escala gris, estas

deben ser preparadas con una resolución de 250 dpi. Las figuras a color deben ser diseñadas con una resolución de 450 dpi. Cuando se utilicen símbolos, flechas, números o letras para identificar partes de la figura, se debe identificar y explicar claramente el significado de todos ellos en la leyenda.

## Derechos de Autor

Los conceptos y opiniones de los artículos publicados son de exclusiva responsabilidad de los autores. Dicha responsabilidad se asume con la sola publicación del artículo enviado por los autores. La concesión de Derechos de autor significa la autorización para que la Revista bit@bit, pueda hacer uso del artículo, o parte de él, con fines de divulgación y difusión de la actividad científica y tecnológica.

En ningún caso, dichos derechos afectan la propiedad intelectual que es propia de los(as) autores(as). Los autores cuyos artículos se publiquen recibirán un certificado y 1 ejemplar de la revista donde se publica su trabajo.

## Referencias Bibliográficas

Las referencias bibliográficas que se utilicen en la redacción del trabajo; aparecerán al final del documento y se incluirán por orden alfabético. Debiendo adoptar las modalidades que se indican a continuación:

## Referencia de Libro

Apellidos, luego las iníciales del autor en letras mayúsculas. Año de publicación (entre paréntesis). Título del libro en cursiva que para el efecto, las palabras más relevantes las letras iníciales deben ir en mayúscula. Editorial y lugar de edición.

Tamayo y Tamayo, M. (1999). El Proceso de la Investigación Científica, incluye Glosario y Manual de Evaluación de Proyecto. Editorial Limusa. México. Rodríguez, G., Gil, J. y García, E. (1999). Metodología de la Investigación Cualitativa. Ediciones Aljibe. España.

Referencia de Capítulos, Partes y Secciones de Libro

Apellidos, luego las iníciales del autor en letras mayúsculas. Año de publicación (entre paréntesis). Título del capítulo de libro en cursiva que para el efecto, las palabras más relevantes las letras iníciales deben ir en mayúscula. Colocar la palabra, en, luego el nombre del editor (es), título del libro, páginas. Editorial y lugar de edición.

Reyes, C. (2009). Aspectos Epidemiológicos del Delirium. En M. Felipe, y Odun. José (eds). Delirium: un gigante de la geriatría (pp. 37-42). Manizales: Universidad de Caldas.

## Referencia de Revista

Autor (es), año de publicación (entre paréntesis), título del artículo, en: Nombre de la revista, número, volumen, páginas, fecha y editorial.

López, J.H. (2002). Autoformación de Docentes a Tiempo Completo en Ejercicio. En Ventana Científica, N° 2. Volumen 1. pp 26 – 35. Abril de 2002, Editorial Universitaria.

## Referencia de Tesis

Autor(es). Año de publicación (entre paréntesis). Título de la tesis en cursiva y en mayúsculas las palabras más relevantes. Mención de la tesis (indicar el grado al que opta entre paréntesis). Nombre de la Universidad, Facultad o Instituto. Lugar.

Salinas, C. (2003). Revalorización Técnica Parcial de Activos Fijos de la Universidad Autónoma Juan Misael Saracho. Tesis (Licenciado en Auditoria). Universidad Autónoma Juan Misael Saracho, Facultad de Ciencias Económicas y Financieras. Tarija – Bolivia.

## Página Web (World Wide Web)

Autor (es) de la página. (Fecha de publicación o revisión de la página, si está disponible). Título de la página o lugar (en cursiva). Fecha de consulta (Fecha de acceso), de (URL – dirección).

Puente, W. (2001, marzo 3). Técnicas de Investigación. Fecha de consulta, 15 de febrero de 2005, de http://www.rrppnet.com.ar/tecnicasdeinvestigacion.htm

Durán, D. (2004). Educación Ambiental como Contenido Transversal. Fecha de consulta, 18 de febrero de 2005, de http://www.ecoportal.net/content/view/full/37878.

## Libros Electrónicos

Autor (es) del artículo ya sea institución o persona. Fecha de publicación.Título (palabras más relevantes en cursiva). Tipo de medio [entre corchetes]. Edición. Nombre la institución patrocinante (si lo hubiera) Fecha de consulta. Disponibilidad y acceso.

Ortiz, V. (2001). La Evaluación de la Investigación como Función Sustantiva. [Libro en línea]. Serie Investigaciones (ANUIES). Fecha de consulta: 23 febrero 2005. Disponible en: http://www.anuies.mx/index800.html.

Asociación Nacional de Universidades e Instituciones de Educación Superior. (1998). Manual Práctico sobre la Vinculación Universidad – Empresa. [Libro en línea]. ANUIES 1998. Agencia Española de Cooperación (AECI). Fecha de consulta: 23 febrero 2005. Disponible en: http://www.anuies.mx/index800.html.

## Revistas Electrónicas

Autor (es) del artículo ya sea institución o persona. Título del artículo en cursiva. Nombre la revista. Tipo de medio [entre corchetes].Volumen. Número. Edición. Fecha de consulta. Disponibilidad y acceso.

Montobbio, M. La cultura y los Nuevos Espacios Multilaterales. Pensar Iberoamericano. [en línea]. N° 7. Septiembre – diciembre 2004. Fecha de consulta: 12 enero 2005. Disponible en: http://www.campus-oei.org/pensariberoamerica/index.html.

## Referencias de Citas Bibliográficas en el Texto

Para todas las citas bibliográficas que se utilicen y que aparezcan en el texto se podrán asumir las siguientes formas:

a) De acuerdo a Martínez, C. (2010), la capacitación de docentes en investigación es tarea prioritaria para la Universidad..

b) En los cursos de capacitación realizados se pudo constatar que existe una actitud positiva de los docentes hacia la investigación (Fernandez, R. 2012).

c) En el año 2014, Salinas, M. indica que la de capacitación en investigación es fundamental para despertar en los docentes universitarios, la actitud por investigar.


**UNIVERSIDAD AUTÓNOMA**
**"JUAN MISAEL SARACHO"**