

La utopía del conocimiento

Despertarum

Septiembre de 2008

PROGRAMACIÓN

SHELL, LA HERRAMIENTA MÁS PODEROSA

Continúa con la segunda parte de este útil tutorial para administrar adecuadamente tu sistema.

Ruby

CSS

¿Aún no sabes qué es?, ya no pierdas más tiempo, mejora el aspecto y la organización del código de tus sitios web.

CONSOLAS DE VIDEOJUEGOS

Da un interesante paseo a lo largo de la historia de los videojuegos

PLATAFORMAS DE PROGRAMACIÓN MÓVILES

La importancia del desarrollo de aplicaciones para dispositivos móviles.

Editorial

Este mes nos enfocamos en un tema que muchas veces llega a ser controversial para los alumnos de la carrera, debido a que muchos consideran que es la parte de la Informática más difícil de desempeñar, tratándola de evadir hasta el momento que se dan cuenta de que es el origen de casi todo lo que nos rodea. Por esto, en la actualidad no es raro llegar a ver un dispositivo eléctrico que no tenga algún software que controle el funcionamiento del mismo, logrando dar más utilidad al dispositivo, esto ha ocasionado que en cada innovación tecnológica sea una parte indispensable e importante en su desarrollo, y a pesar de que en el presente vivimos un desequilibrio entre el hardware que avanza a pasos agigantados y el software que por razones como la comercialización, el error humano y sobre todo los intereses personales, se ha visto estancada. Es importante reflexionar sobre la importancia que le damos a la programación como futuros licenciados en Informática.

Sin más rodeos damos inicio al número 9 de su ya tan conocida revista Despertarum que viene plagada de temas interesantes.

Equipo Despertarum

Cualquier duda, comentario o sugerencia, escríbenos a:

despertarum@fesc.unam.mx

Emmanuel Ortiz Cruz



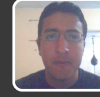
...cuando aquello que en teoría no puede, haya sucedido
psycho.emorc@gmail.com

Julio C. Rivera López



Nunca es suficiente
jcrivera.mx@gmail.com

Oscar Ramírez Hernández



El talento sólo es superado por el esfuerzo
gallo1oscar@gmail.com

Gustavo Carrillo Romero



La clave del conocimiento es mirar más allá de lo que crees lo es todo
gustavocr.info@gmail.com

Iván Lagunas Salazar



Condenados a derribar muros
ivan_lags@hotmail.com

Gabriela Aguilar Mendoza



Lo pasajero, es lo inevitable. Lo definitivo, las lecciones de lo inevitable
gaby_aguilar_mendoza@yahoo.com.mx

Carlos Pineda Muñoz



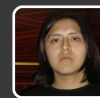
El conocimiento es libre
carpin@servidor.unam.com

Sonia A. Morales Salazar



Lean, lean lo que sea, leer siempre nutre
sonia_1258@hotmail.com

Eduardo de la cruz flores



El hombre le teme a lo que no tiene, y la mujer tiene lo que no entiende
volcom_9999@hotmail.com

Ana G. Villa Trujillo



Después de escalar una montaña muy alta, descubrimos que hay muchas otras por escalar
anaabbsb@gmail.com

04

Consolas de videojuegos

No te pierdas este interesante artículo con la historia de los videojuegos.
Por: Julio C. Rivera López

07

Programación

Un tema controversial pero parte importante para nosotros, Lic. en Informática.
Por: Oscar Ramírez Hernández, Carlos Pineda Muñoz, Emmanuel Ortiz Cruz

16

Plataformas de programación móviles

Como programadores, no debemos olvidar el desarrollo de software para móviles.
Por: Gustavo Carrillo

18

Dell comienza la cuenta atrás para preinstalar Linux en sus computadoras

Ya es un hecho, confirmado como el Sistema Operativo por defecto.
Por: Eduardo de la cruz flores

19

Shell, la herramienta más poderosa

Continúa con la segunda parte que te traemos de este interesante tema.
Por: Oscar Ramírez Hernández

24

Open Office.org Base

Conoce esta aplicación -Software libre- que permite la creación y administración de bases de datos (Primera parte).
Por: Gabriela Aguilar Mendoza

28

CSS

¿Aún no sabes qué es?, ya no pierdas más tiempo, mejora el aspecto y la organización del código de tus sitios web.
Por: Sonia A. Morales Salazar

31

Particiones

Una interesante aportación para la administración de tus discos.
Por: Ismael Mayo, alumno de 5o. semestre de la carrera

33

Domótica

Más comodidades para tu vida diaria con sistemas informáticos.
Por: Gabriela Aguilar mendoza

35

¡Ayudaj, nuestro planeta se enferma cade vez más

¿Sabes a dónde irá a parar ese celular o computadora que acabas de desechar?
Por: Ana G. Villa Trujillo

Consolas de videojuegos

Por: Julio Cesar Rivera López



Hablar de videojuegos hace que viajemos a los inicios de la década de los setenta, cuando Nolan Bushnell y Al Alcorn desarrollaron “Pong”, un juego que visualmente tenía dos pequeñas barras en los extremos de la pantalla (las cuales eran las raquetas) y un círculo que para esa época era más un cuadrado, el cual viajaba de izquierda a derecha simulando el viaje de una pelota de tenis, así fue creada una máquina traga-monedas que se llevó a los bares de California, con un éxito rotundo.

Lo anterior sin duda sería el paso fundamental para establecer los cimientos de lo que llegaría a ser Atari, pues el éxito de Pong en los bares llevó a los directivos de Atari a crear una versión casera denominada Home Pong, la cual para la época fue realmente innovadora, pues esta cajita mágica se conectaba al televisor; así para la navidad de 1975 en conjunto con la cadena de tiendas Sears, Atari y el Home Pong saltaron a la fama al vender 100,000 unidades en un mes.



Para finales de la década de los setenta el Pong pasaba por uno de sus peores momentos pues la competencia era cada vez más reñida, por lo que era inminente una actualización en el Hardware y el Software, por lo cual fue presentada la Atari CX-2600, ésta ya no sólo se conectaba a la computadora sino que también venia con un joystick y la posibilidad de usar cartuchos que contenían toda la información sobre el juego dentro de ellos, estos cartuchos complementaron de gran forma la experiencia pues la creación de títulos fue descomunal, entre los más famosos están el Ateroids y el Space Invaders en la fase inicial y Pac-Man unos años después, por mencionar algunos.



Para mediados de la década de los ochenta justo cuando Atari se encontraba tratando de dominar al mercado, surgió de la nada una compañía que cambiaría la industria de los videojuegos; Nintendo. Una compañía con raíces similares a las de Atari pues se dedicaba al diseño y distribución de máquinas traga-monedas sumamente exitosas, por lo cual decidieron lanzar al mercado Japonés en 1983 el Family Computer

mejor conocido como Famicom, el cual incluía el famoso Donkey Kong.

La consola de la gran “N” se convirtió en la más vendida de Japón por lo cual se decidió presentarla en el CES (Consumer Electronic Show) de 1984, con la finalidad de conquistar el mundo occidental, pero antes de su lanzamiento en América su nombre paso de Famicom a NES (Nintendo Entertainment System); para su lanzamiento la consola incluía dos controles, una pistola (Zapper) y dos juegos en un sólo cartucho, el Super Mario Bros. y el Duck Hunt, lo cual bastó para llevar a esta consola de ocho bits a vender más de 60 millones de unidades en todo el mundo.

La industria continuaba desarrollándose y en el mapa apareció Sega, con su SEGA Master System, la cual era una consola de 8 bits (como el NES) diseñada especialmente para competir con la consola de Nintendo, pero a pesar de poseer tecnología superior no logró consolidarse en el mercado mundial. Así la década de los ochenta continuó con una lucha férrea entre SEGA y Nintendo por el dominio del mercado mundial.



Los 16 bits se hicieron presentes con la aparición del Sega Mega Drive en 1988, y venía dispuesta a terminar con el reinado del NES con títulos deportivos y un personaje denominado Sonic, esta consola puso a temblar al mercado, lo que motivó a Nintendo a liberar la Super Nintendo Entertainment System (SNES) también llamada Super Famicom en oriente, la cual al ser de 16 bits compitió cara a cara con la consola de SEGA, fue realmente difícil para Nintendo que tuvo que aumentar sus esfuerzos para poder ser la consola con mejores títulos y mejor vendida de su generación.

El declive de SEGA después del lanzamiento del Sega CD debido a que no obtuvo la aceptación que se esperaba del público, el dominio por parte de Nintendo y la gestación de una próxima generación de consolas llevó a directivos de Sony y de la gran N a iniciar pláticas para desarrollar e integrar una unidad de CD-ROM a la Super NES, pláticas que fracasaron; por lo cual Sony decidió lanzar por su cuenta una nueva consola denominada; PlayStation.



Una vez más la industria de los videojuegos dio un giro de 180 grados entre diciembre de 1994 y septiembre de 1995 pues la PlayStation no utilizaba cartuchos, en su lugar utilizó el CD-ROM, además tenía dos ranuras que serían utilizadas para insertar tarjetas de memoria en donde se grabarían los avances de los juegos.

La PlayStation le arrebató la corona a Nintendo pues dominó el mercado gracias a la combinación de tecnología, vanguardia y grandes títulos como Metal Gear Solid, Winning Eleven, Tekken, Resident Evil, Gran Turismo, Final Fantasy, Silent Hill, entre otros, que claramente estaban enfocados a un grupo de gamers adultos que ya tenían un ingreso constante y que deseaban jugar con un mayor grado de dificultad, lo que llevó a la consola a colocar un estimado de 100 millones de unidades en todo el mundo durante su existencia.

Sony era el rey para finales de los 90 pero ya no bastaba con tener buenas gráficas, utilizar tarjetas de memoria y tener un decente nivel de dificultad, el usuario deseaba más y dicho capricho se le habría de conceder, la primera compañía en tomar el riesgo fue SEGA que con la colaboración de Hitachi y Microsoft presentó la que sería la última consola que produciría; el DreamCast, para muchos el Rey o Reina (como más les agrade) sin corona, la cual tenía como característica innovadora la integración de un modem para poder jugar on-line, el DreamCast fue sin duda un éxito rotundo pues colocó más de medio millón de consolas en los hogares en menos de un mes, ayudado por títulos como Soul Calibur y Sonic Adventure.



El reinado duró poco pues tras la incursión del GameCube, el aviso por parte de Sony del lanzamiento de una nueva consola, y los constantes problemas en la estructura y en la economía de la compañía, SEGA anunció que se retiraba definitivamente del mercado de consolas caseras convirtiendo así a esta consola en una leyenda de esta generación que hasta nuestros días sigue dando de que hablar.

Después de la salida del DreamCast y del lanzamiento del PlayStation 2 Nintendo presentó el GameCube, en donde por fin se dejaron de utilizar cartuchos para utilizar el GameCube Optical Disc, integraban procesadores PowerPC y una placa gráfica desarrollada por una compañía que pertenece a ATI por lo cual en cuanto a gráficos era superior al PlayStation 2, pero todo lo anterior no bastó para ser la mejor, lo que llevó a esta consola a quedarse relegada una vez más por la competencia



El reinado debía continuar, de esta forma con unidad de DVD, ranuras para memoria, puertos USB y nuevos controles (mandos), en el 2000 se lanzó la siguiente generación de la consola de Sony la cual se agotó el día de su lanzamiento, esto llevó a que por primera vez se vendiera una consola a precios exorbitantes, los cuales rondaban los mil dólares americanos, el lanzamiento fue un verdadero caos pues no habían consolas ni grandes títulos disponibles por lo cual la reputación de la primera consola y la retro-compatibilidad de la misma dieron un respiro para esperar la llegada de nuevas consolas y grandes títulos.

Sony logró llevar a su Play Station 2 al primer lugar una vez más, logrando vender cerca de 120 millones de consolas en todo el mundo y colocando una cantidad impresionante de títulos en el mercado, pero a pesar del éxito títulos como Resident Evil 4 dieron muestras del pobre poder gráfico de la consola y la incursión del Xbox al mercado hizo parecer por momentos viejo al PS2.

Una nueva batalla estaba por comenzar, lo que inminentemente nos llevaría a nuevos horizontes pues para finales de 2001 y el otoño de 2002 con la ayuda de Intel y Nvidia Microsoft presentó un contendiente más, el XBOX, que posee un procesador Pentium III, DVD, buenos gráficos, disco duro (para almacenar avances de juego y música) en lugar de tarjetas de memoria, y la posibilidad de jugar on-line gracias al servicio Live el cual fue sin duda un parteaguas en la manera en la que disfrutamos los juegos gracias a su excelente planeación y funcionamiento en términos generales, y que combinado con juegos como; Halo, FIFA Soccer, Dead or Alive, Project Gotham, Soul Calibur y Ninja Gaiden han dado un lugar a la caja "X" en nuestros hogares.



Hoy vemos el pasado de los videojuegos como el causante del presente, lo señalamos como la base del futuro, pensamos en las horas de diversión que nos brinda, deseamos llevarlo siempre en nuestro corazón e irónicamente lo llevaremos al olvido debido a que ya no nos satisface del todo, debido al vertiginoso desarrollo de tecnologías, pero aún así estamos expectantes al siguiente paso, a la siguiente innovación, al siguiente enemigo, al siguiente reto a vencer, a la siguiente aventura, al siguiente nivel...

Nos vemos en la siguiente generación.

GAME OVER

QUE MARCARON LA HISTORIA.

Por: Oscar Ramírez Hernández

Existen demasiados lenguajes de programación y cada uno dentro de la historia ha sido desarrollado pensando en tareas específicas, algunos muriendo en el intento, sin embargo otros han dejado una huella marcando un antes y un después en la historia de la programación, por eso dedicamos un pequeño espacio a estos lenguajes. Para ello los presentaremos de acuerdo a su origen.

Ensamblador

No podemos empezar sin mencionar uno de los lenguajes que revolucionó la historia de la programación, nos referimos al lenguaje ensamblador, por el simple hecho de que antes de que existiera este lenguaje la programación de procesos se hacía en base al lenguaje máquina, es decir, se interpretaba a partir de unos y ceros, causando mucha dificultad en el desarrollo de procesos y aún más en el depurado de ellos, por tal motivo cuando surge el lenguaje ensamblador no sólo trajo un lenguaje más entendible para el desarrollador, sino también una gran automatización en los sistemas, debido a que el lenguaje permitía crear procesos más complejos e incluso rutinas automáticas causando menos trabajo al operador de la máquina.

Hoy en día el lenguaje ensamblador es muy poco usado debido a que es considerado un lenguaje de bajo nivel, siendo uno de los más difíciles de entender o escribir, e incluso también depende directamente de la arquitectura que usemos, pero aún así, este lenguaje es muy recomendable cuando se quiere optimizar algún proceso principalmente de hardware, que no sería posible con un lenguaje de medio o alto nivel.

FORTTRAN

Uno de los lenguajes que no podemos dejar pasar es precisamente FORTRAN, que viene a ser el primer lenguaje de programación de alto nivel, el cual fue diseñado para la resolución de problemas científicos, éste genera una gran utilidad debido a que surge con una gran biblioteca de funciones matemáticas directamente entendibles con el lenguaje máquina, además de quitarle la carga de trabajo al operador de realizar más de un proceso para la obtención de un resultado complejo, debido a que FORTRAN hacía subprocesos elementales sin necesidad de que el operador interviniera, que esto a su vez permitía una gran velocidad en la obtención de resultados en esa época, dándole al operador la facilidad de leer solo el resultado final.

Hoy en día sigue siendo uno de los lenguajes de programación predilectos en el área de las matemáticas y generalmente más en la física, debido a su gran procesamiento de datos en sus versiones más modernas, incluso se han hecho pruebas donde realizan un determinado proceso complejo matemático en un lenguaje de programación llamado "Visual Basic", tardando 80 horas para arrojar un resultado, en cambio FORTRAN ejecutando el mismo proceso llega a tardar 10 segundos.

COBOL

Otro de los lenguajes dentro de nuestra lista es COBOL, este lenguaje surge mediante la necesidad que se tenía de que el software de esa época llegara a ser legible y sobre todo portátil, junto al gran mercado comercial en las empresas conscientes de la importancia de automatizar sus operaciones, el resultado fue un lenguaje con una sintaxis simple que generaba programas fáciles de trasladar de computadora a computadora, siendo de los primeros lenguajes enfocados netamente hacia actividades administrativas.

El impacto de este lenguaje fue tan grande que hoy en día todavía existen grandes sistemas corriendo con versiones muy antiguas de COBOL, esto debido más que nada, a que todavía les son útiles los sistemas y prefieren conservar el código que desarrollar nuevas aplicaciones.

C/C++

Bueno, no podría haber una lista sin mencionar al llamado “Inmortal” de los lenguajes de programación, nos referimos a C, que cuando se desarrolló este lenguaje fue tanto su impacto que en vez de que surgiera otro lenguaje para sustituirlo, como normalmente pasa, se optó por ampliar su versión, a lo que conocemos como C++, dejando una enorme huella dentro de la historia de los lenguajes de programación, debido a que es un lenguaje de medio nivel portátil y estándar en todas las plataformas, logrando desarrollar enormes sistemas complejos, con gran estabilidad y muy livianos.

Este lenguaje desde sus inicios en los años 70s hasta la actualidad ha sido utilizado para diversas actividades, desde pequeñas rutinas hasta Sistemas Operativos, lo cual advierte que todavía queda mucha pila para este lenguaje de programación.

JAVA

Este lenguaje a pesar que desde sus inicios ha recibido infinidad de críticas por su robustez y que sus actualizaciones no son muy estables, entre otras, ha sabido sobrevivir más por mérito de los programadores que de los desarrolladores del lenguaje. La importancia de este lenguaje fue el impacto que tuvo en sistemas basados en internet, a pesar de que no fue desarrollado para eso, sino para sistemas atractivos para calculadoras, televisiones, etc. supo adaptarse a las circunstancias y a la nueva ola de internet.

En la actualidad JAVA ha sido desfasado poco a poco en el ámbito de sistemas basados en internet, sin embargo la importancia que está teniendo sobre sistemas portátiles, sobre todo celulares, sugiere un segundo aire para este lenguaje de programación.

Estos 5 lenguajes de programación a mi parecer han sido los que han trascendido a través de la historia dejando su huella, y aunque han existido muchos que realmente han sido buenos y algunos que prometen demasiado, necesitarán buenos argumentos para desplazar a estos 5.

Por: Carlos Pineda Muñoz

Los lenguajes de programación pueden clasificarse empleando distintos métodos y puntos de vista. Esta clasificación se basa en el paradigma que utilizan. Se debe aclarar que todavía hay conflictos en las definiciones y alcances de ciertos paradigmas.

¿Qué es un paradigma de programación?

Un paradigma de programación determina la visión y métodos de un programador en la construcción de un programa o subprograma. De los diferentes paradigmas resultan diferentes estilos de programación y en diferentes formas de plantear la solución de problemas.

Los lenguajes de programación están basados en uno o más paradigmas. Por ejemplo: C++, Smalltalk y Java son lenguajes basados en el paradigma orientado a objetos. El lenguaje de programación Scheme, en cambio, soporta sólo programación funcional. En cambio Python, soporta múltiples paradigmas, PHP y C ANSI bajo el paradigma estructurado.

Descripción por paradigmas de programación

Paradigma Imperativo: describe la programación como una secuencia de instrucciones o comandos que cambian el estado de un programa. El código máquina en general está basado en el paradigma imperativo.

Características

Los datos. Cada dato tiene una representación y un número de propiedades conocidas como tipo.

El estado de los programas. Una actividad importante de un programa en un lenguaje imperativo es el cambio del estado interno de la computadora, representado por los valores de las variables y del estado externo, representado por las entradas y salidas a dispositivos.

El flujo de control. Los lenguajes imperativos dan al programador un control extensivo sobre el orden en el cual se ejecutan los enunciados en el programa. Los mecanismos usuales son: secuencia, selección, repetición e invocación de rutinas.

La composición de los programas. Se pueden distinguir cuatro niveles de jerarquía: bloques, rutinas, paquetes (módulo o tareas) y programas.

Efectos de una asignación: Una propiedad característica de una asignación es que cambia el valor almacenado (estado actual) en una dirección (localidad) de memoria en una computadora. Los conceptos de "estado" y "localidad" vienen directamente del modelo de la computadora para el lenguaje imperativo.

Ejemplo del paradigma imperativo con el Cálculo del módulo

```

función mod (x, y: enteros) : entero
inicio
z := x
mientras z >= y hacer
z := z - y
finmientras
devolver z
fin
    
```

Paradigma Declarativo: No se basa en el cómo se hace algo (cómo se logra un objetivo paso a paso), sino que describe (declara) cómo es algo. En otras palabras, se enfoca en describir las propiedades de la solución buscada, dejando indeterminado el algoritmo (conjunto de instrucciones) usado para encontrar esa solución. Es más complicado de implementar que el paradigma imperativo, tiene desventajas en la eficiencia, pero ventajas en la solución de determinados problemas.

Paradigma Estructurado: la programación se divide en bloques (procedimientos y funciones) que pueden o no comunicarse entre sí. Además la programación se controla con secuencia, selección e iteración. Permite reutilizar código programado y otorga una mejor comprensión de la programación. Es contrario al paradigma no estructurado, de poco uso, que no tiene ninguna estructura, es simplemente un “bloque”, como por ejemplo, los archivos batch (.bat).

Paradigma Orientado a Objetos: está basado en la idea de encapsular estado y operaciones en objetos. En general, la programación se resuelve comunicando dichos objetos a través de mensajes (programación orientada a mensajes). Se puede incluir -aunque no formalmente- dentro de este paradigma, el paradigma basado en objetos, que además posee herencia y subtipos entre objetos. Ej.: Simula, Smalltalk, C++, Java, Visual Basic .NET, etc.

Su principal ventaja es la reutilización de códigos y su facilidad para pensar soluciones a determinados problemas.

Los conceptos básicos del Modelo de Objetos se originaron con la definición de lenguajes que incorporan conceptos tales como:

- Encapsulamiento
- Modularidad
- Abstracción
- Polimorfismo

Características de la programación orientada a objetos

1. La programación consiste en el envío de mensajes.
2. El resultado del envío de un mensaje a un objeto es un objeto.
3. La solución del problema consiste en:
 - a) identificar los objetos en el enunciado del problema
 - b) identificar los mensajes asociados a cada objeto
 - c) desarrollar una secuencia de mensajes a objetos que resuelvan el problema
4. El lenguaje orientado a objetos debe soportar encapsulamiento, abstracción, herencia y polimorfismo.
5. La unidad de encapsulamiento es el objeto, el cual incluye datos privados, compartidos y descripciones de mensajes.

6. La POO soporta la abstracción por medio de una jerarquía de clases, que representan diferentes clases de objetos.
7. Un protocolo de descripción de clase define las propiedades de cada abstracción. Nuevas abstracciones son agregadas por la adición de nuevas clases.
8. Las subclases heredan las propiedades de sus superclases incluyendo los datos privados, los compartidos y los mensajes.
9. Los mensajes heredados pueden ser polimórficamente redefinidos por las subclases.
10. La POO soporta la definición polimórfica del mismo mensaje en diferentes clases.
11. En POO los datos son almacenados como objetos, los cuales son eliminados automáticamente de la memoria cuando no son necesarios.
12. La POO provee el soporte completo para la resolución incremental del problema.
13. Los objetos del sistema y los objetos definidos por el usuario son de la misma categoría. No existe la diferencia entre tipos predefinidos y dDefinidos por el usuario.
14. Se soporta completamente la extensibilidad.
15. El uso de estructuras iterativas y el indexado multinivel está minimizado y localizado en detalles de implementación de mensajes simples.

Paradigma Funcional: este paradigma hace hincapié en la aplicación de las funciones y composición entre ellas, más que en los cambios de estados y la ejecución secuencial de comandos (como lo hace el paradigma procedimental). Permite resolver ciertos problemas de forma elegante y los lenguajes puramente funcionales evitan los efectos secundarios comunes en otro tipo de programaciones.

Uno de los lenguajes que se desarrollaron bajo este paradigma, es LISP un lenguaje interactivo, a continuación algunos ejemplos:

```
(setq A 0)
(setq B 1.0)
(setq C '(Hola mundo))
(+ A B)
(* (+ A 1) B)
A
```

Manejo de cadenas, listas y pilas

```
(car '(Este es un buen dia))
(cdr '(Este es un buen dia))
(append '(hola) '(mundo) '(azul))
(list '(1) '(2) '(3))
(setq A '(1))
(push 2 A)
(push 3 A)
```

Explicación:

La línea 1: (car '(Este es un buen dia)) realiza la función car que obtiene la primer palabra de una cadena.

En el ejemplo, el resultado es "Este".

La línea 2: (cdr '(Este es un buen dia)) realiza la función cdr que elimina el primer elemento de una cadena.

En el ejemplo, el resultado es "es un buen día"

La línea 3: (append '(hola) '(mundo) '(azul)) usa la función append que concatena o une cadenas y las regresa como una sola.

La línea 4: (list '(1) '(2) '(3)) forma una lista con los elementos 1 2 3 con el comando lisp.

La línea 5: (setq A '(1)) crea una variable A y le asigna el elemento 1. En las últimas dos líneas le agrega los elementos 2 y 3 con push.

Los elementos serán agregados al inicio de la pila.

Paradigma lógico: se basa en la definición de reglas lógicas para luego, a través de un motor de inferencias lógicas, responder preguntas planteadas al sistema y así resolver los problemas. Ej.: prolog.

Ejemplos típicos de problemas para el paradigma lógico.

•Indicar cuales de las siguientes cadenas son fórmulas bien conformadas, considerando el conjunto de variables x, y, z ; las constantes a, b , las funciones f de aridad 1 y g de aridad 2; y el conjunto de predicados P, Q, R donde $r(P) = 2, r(Q) = 1$ y $r(R) = 3$

- a) $P(a, f(a))$
- b) $\forall x P(a, f(a))$
- c) $\exists z g(z, z)$
- d) $(P(x,y) \rightarrow Q(x))$
- e) $f(f(a))$
- f) $(P(x) \wedge Q(x))$
- g) $Q(P(x))$
- h) $Q(f(f(a)))$
- i) $R(f, a, b)$
- j) $(P(x, f(g(y,y))) \leftrightarrow Q(x))$
- k) $\forall x (P(a, f(a)) \rightarrow x)$

•Utilizando el predicado elegir

$elegir(X, [X|X_s], X_s).$
 $elegir(X, [Y|Y_s], [Y|Z_s]) :- elegir(X, Y_s, Z_s).$

Construya el árbol de búsqueda Prolog que le permite hallar todas las soluciones posibles para la siguiente consulta:

$elegir(T?, [a1, a2, a3], P?)$

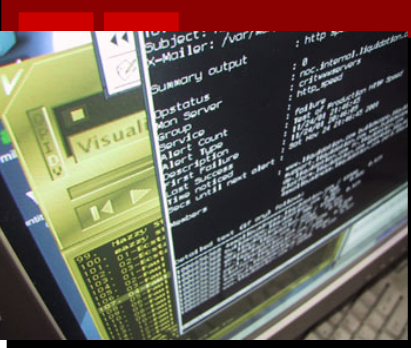
e indique cuales son las mismas.

Como se dijo al inicio del texto, en ocasiones no hay acuerdos para la clasificación de algunos lenguajes, pero se consideran los más aceptados y como se puede observar hay lenguajes para todo tipo de aplicaciones, y es decisión del programador elegir el que resuelva de mejor forma el problema a enfrentar.

Pequeños que caminan a pasos AGIGANTADOS

Programación

Por: Emmanuel Ortiz Cruz



En el mundo de la programación actual nada está escrito, el que un nuevo lenguaje salga y se apodere del corazón de los programadores es algo que prácticamente es imposible, para que un lenguaje gane adeptos generalmente pasan años antes de que se pueda consolidar y considerarse maduro. En los años recientes algunos lenguajes de programación se han ganado poco a poco gran cantidad de seguidores debido a su simpleza, lo poderoso de sus resultados o simplemente a la gran difusión y publicidad que han tenido.

Existen 5 lenguajes que en lo particular me han llamado la atención porque poco a poco están ganándole terreno a los clásicos como C/C++ y Java.

PHP (PHP: Hypertext Pre-processor):

Este lenguaje fue inicialmente desarrollado por Rasmus Lerdorf en 1994 para proveer ciertas funcionalidades extras a su página web personal, con el nombre de Personal Home Page Tools, y ahora se ha convertido en uno de los lenguajes de programación de scripts del lado servidor más utilizados, la ligereza a la hora de desempeñar su labor en añadir dinamismo a las páginas web aunado a la sencillez a la hora de aprenderlo han hecho que su crecimiento sea enorme.

PHP tiene entre sus ventajas que fácilmente puede ser utilizado para interactuar con MySQL y es totalmente compatible con el servidor web Apache lo que hace de estos tres más el uso del Sistema Operativo Linux, el medio más utilizado para crear páginas web.

Algunas características más son:

- Multiplataforma
- De uso libre para cualquiera
- Soporta programación orientada a objetos

PHP sin duda es uno de los lenguajes que poco a poco han ido haciéndose de una fama propia y debido a las extensiones creadas para QT (PHP-QT) y GTK (PHP-GTK) ha dado el salto hacia el desarrollo de aplicaciones de escritorio.



Python:

Es un lenguaje que fue creado por Guido van Rossum en 1990 y actualmente se mantiene desarrollado por la Python Software Foundation, la utilización de este lenguaje se da principalmente en ambientes Unix, en Linux principalmente ya que es una de las herramientas que se instala automáticamente y como base de todas las distribuciones Linux, es un lenguaje sencillo de aprender y debido a su filosofía el código que se genera haciendo uso de éste es bastante limpio, legible y ordenado, no está enfocado a un paradigma en particular sino que se pueden hacer uso de varios estilos ya sea programación estructurada, orientada a objetos o funcional, existen diversas implementaciones creadas para ser utilizadas con otros lenguajes de programación:

CPython para C

IronPython para .NET

Jython para Java

Es lenguaje multiplataforma y poco a poco va ganando más y más seguidores.



Ruby:

Como ya se había comentado en el número 3 de Despertarum Ruby fue creado por Yukihiro Matsumoto, el principal objetivo de este lenguaje es la productividad y la diversión del desarrollador, por lo tanto el lenguaje en sí es bastante sencillo de aprender, el crecimiento en los últimos años de este lenguaje hasta hace poco desconocido se sitúa en el desarrollo de sitios web, debido a la gran difusión que tiene Ruby on Rails.

Aunque es un lenguaje principalmente desarrollado para aplicaciones de escritorio la simpleza de la sintaxis y la facilidad de distribuir los paquetes mediante Ruby Gem hizo que el framework Ruby on Rails se considere una alternativa seria a PHP y ASP.

La hasta hace poco principal desventaja era que la mayoría de la documentación existente sobre este lenguaje estaba en japonés pero debido a su crecimiento en la actualidad es mucho más fácil encontrarla en diferentes idiomas.



Ruby
A Programmer's Best Friend

Javascript:

Sí, leyeron bien, Javascript es uno de los lenguajes que en el futuro próximo serán indispensables a la hora de desarrollar aplicaciones incluso de escritorio. El crecimiento acelerado de la internet ha traído consigo el renacimiento de este lenguaje que parecía desaparecería; durante la guerra de los navegadores este lenguaje fue odiado y amado a la vez ya que la incompatibilidad entre navegadores obligaba a los desarrolladores crear versiones específicas para cada navegador, y aunque en la actualidad se sigue haciendo de la misma forma, la estandarización y que Microsoft por fin haya decidido apearse a las reglas han resucitado a este poderoso lenguaje de scripts del lado cliente.

En los últimos 5 años aproximadamente el crecimiento que ha tenido ha sido asombroso, primero con la capacidad de utilizar AJAX en navegadores distintos a Internet Explorer y después con el surgimiento de los llamados frameworks o librerías Javascript que ahorran trabajo a los desarrolladores. En algunos ámbitos se rumora que próximamente Javascript dará un salto hacia el terreno de las aplicaciones de escritorio, versión que aún no se puede confirmar.

```
var x = 2; var y = 10; var z = "5"
document.write(eval("x * y + z + 1"))
```

Se obtiene: 2051

Como pueden ver, la de eval() y parece a la que se utiliza por el

objeto.toString(). La finalidad de este método, común

C#

Pronunciado como “si sharp” es un lenguaje creado por Microsoft incluido dentro de su plataforma de desarrollo .NET, la forma en que este nuevo lenguaje (surgió en 2001) ha ganado seguidores es debido a que la plataforma .NET es exclusiva para el sistema operativo Windows el cual sigue siendo (y seguirá por mucho tiempo) el más utilizado, su sintaxis es parecida a la de C++ y de hecho el símbolo # hace alusión a su descendencia de C++ ya que se forma de “++” sobre otro “++” eliminando separaciones.

Actualmente existen dos proyectos libres (muy discutidos) que intentan que los programas creados con (C# principalmente) la plataforma .NET puedan ser ejecutados en plataformas diferentes a Windows, los proyectos son DotGNU y Mono.



Quizás para muchas personas existan lenguajes que omití pero como señalé al principio estos lenguajes a mi punto de vista están creciendo y no será extraño que en los próximos años sean básicos para cualquier programador.

Plataformas de programación móviles

Como estudiantes de la licenciatura en Informática es indispensable abordar el tema de programación, a lo largo de nuestra estancia universitaria manejamos varias herramientas y técnicas de programación, siempre empezando con los conceptos básicos y programando bajo plataformas muy conocidas como Windows o Linux, pero es necesario que ampliemos nuestros horizontes para poder tener un panorama más extenso y poder elegir el área a la que nos vamos a dedicar por una buena parte de nuestros años.

Muchas plataformas de desarrollo están siendo creadas o están evolucionando, en este caso expondré una alternativa al cómputo tradicional y que trata sobre movilidad, el tema que muchos expertos dicen que se está convirtiendo en una parte fundamental del desarrollo de las empresas.

Actualmente los usuarios no se conforman con tener una computadora de escritorio o una notebook, ahora desean llevar muchos de sus datos en el bolsillo; y es precisamente bajo este concepto que muchas empresas se han puesto a trabajar en el desarrollo de teléfonos inteligentes (Smartphones); en la actualidad hay muchas plataformas como Symbian, Windows Mobile, RIM, etc., pero últimamente dos están haciendo mucho ruido, se trata de iPhone y Android, dos plataformas recientes que buscan acaparar el mercado de las aplicaciones móviles.

Al hablar de Smartphones es necesario incluir un Sistema Operativo, Aplicaciones y un Kit de Desarrollo de Software (SDK) para crear una plataforma robusta que sea capaz de resolver muchos de los problemas de un usuario convencional o empresarial.

En tanto al iPhone, se basa en que un par de desarrolladores pueden descargar su SDK, crear una aplicación, someterla a evaluación y después venderla a través de una tienda especializada, lo que permitiría crear nuevos modelos de negocio y generar nuevas empresas que desarrollen software para móviles. Como habría de esperarse de la marca, hay varios controles de calidad, derechos de autor y restricciones que aunque no son malas, cierran un poco el panorama de opciones.



Por otra parte, el sistema operativo Android de Google para Smartphones que acaba de ser anunciado el 23 de Septiembre de 2008, es una plataforma abierta, libre de pago que contempla la creación de aplicaciones gratuitas o de pago, sólo hay que bajar el SDK, programar una aplicación y publicarla en Android Market Place, un lugar donde estas aplicaciones pueden ser descargadas por un comunidad de usuarios que irá creciendo poco a poco.

Estas dos plataformas de desarrollo han hecho campañas para difundir sus capacidades, cada una tiene sus beneficios y sus problemas, sin embargo nos dejan un panorama más claro, donde nos damos cuenta que hay un gran mercado que cada vez se extenderá más y que es sin duda la evolución de las computadoras, hoy en día no nos podemos conformar con aprender ciertos lenguajes, debemos de contemplar el futuro de las aplicaciones y así enfocar nuestros objetivos, para que el día

que queramos entrar a un mercado laboral tengamos las bases necesarias para hacerlo.



ANDROID
Google

Seguramente en un futuro cercano podremos tener una verdadera experiencia web y multimedia en nuestras manos, para esto hace falta hardware poderoso que todos los días se renueva y mejora, sin embargo lo que realmente se necesita son programas capaces de llevar el cómputo tradicional a la vida móvil, es en este punto donde nosotros como estudiantes podemos ver la posibilidad de no sólo enfocarnos a lo que actualmente se usa, sino a lo que en un futuro será lo que más se utilice.

Links:

Android SDK

<http://code.google.com/android/>

iPhone SDK

<http://developer.apple.com/iphone/>

Dell comienza la “cuenta atrás” para preinstalar Linux en sus computadoras

Por: Eduardo de la cruz flores

Dell confirmó que para este año (2008) su intención principal es incluir Linux por defecto, y sin costo alguno, y que se establezcan en algunas de sus computadoras de mesa y portátiles de su amplia gama existente en el mercado.

Esta decisión, sobre la que se especula desde hace unas cuantas semanas, da como consecuencia lógica a la tormenta de ideas que la compañía lanzó a mediados de febrero, y en la que preguntaba a sus clientes qué nuevas herramientas querrían tener en su PC.

La confirmación se ha efectuado mediante un mensaje en la web oficial de la compañía, en la que Dell afirma lo siguiente: “facilitaremos más información en las próximas semanas, incluyendo detalles sobre los sistemas que ofreceremos y la distribución o distribuciones que estarán disponibles. La cuenta atrás comienza hoy”.

Es por eso que la decisión de Dell es revolucionaria, por cuanto supondrá una gran evolución para la expansión de los sistemas Linux, y en especial de aquellas distribuciones (o versiones) que resulten ser elegidas, ya que con esta selección de la distintas distribuciones de linux, será más fácil de meter alguna distribución de Linux en algún equipo de Linux, en base a las características que pida la distribución que vaya a ser agregada, para que ésta pueda ser agregada al equipo. Y más aún si se tiene en cuenta que muy probablemente otras compañías sigan el paso que la marca Dell.

Hasta la fecha, Linux no era considerado una alternativa seria a los sistemas tradicionales, especialmente para Windows de Microsoft, debido a las dificultades de configuración, problemas de compatibilidad y sobre todo, un entorno gráfico que hacía muy complicado su uso. Pero los últimos avances de los grupos sin ánimo de lucro y de las empresas que desarrollan las diversas distribuciones de Linux, están cambiando rápidamente el panorama, haciendo que Linux sea más rápidamente aceptado y que Windows se vaya desplazando más rápido del mercado de las tecnologías de la información.

Por lo que en conclusión, no falta mucho para que Linux se vuelva líder y desbanque a otros sistemas operativos, y bueno, que además su uso se vuelva más generalizado entre las marcas de computadoras más comunes y que además el tipo de tecnologías que usan estos equipos de cómputo sean más aprovechables usando alguna de las tantas distribuciones que tiene Linux y también que satisfaga las necesidades de los usuarios finales.



¡Hola a todos!, esta es la segunda entrega de la sección dedicada a la Shell dentro de su revista Despertarum, esperando que la primera entrega halla sido de su utilidad. Para comenzar seguiremos hablando sobre la manipulación de archivos y directorios.

Normalmente cuando estamos navegando por los directorios de nuestro sistema, existen archivos de texto, los cuales pueden contener información útil para la configuración de algún programa o sobre nuestro sistema, para ello existen dos instrucciones útiles, una de ellas es “cat”.

Sintaxis: `cat [Archivo]`

Mostrará todo el contenido del archivo, el problema de la instrucción consiste en que desplegará todo el contenido hasta el final, sin tomar en cuenta si nos da tiempo de leerlo, debido a que la consola solo puede visualizar una cierta cantidad de caracteres, mostrando solo los últimos, no pudiendo visualizar una parte del documento si es muy grande, podemos pausar el flujo de información cuando oprimimos la tecla “Ctrl” + “s”, pero si queremos abortar el despliegue de la información tendremos que teclear “Ctrl” + “c”.

Ejemplo:

```
$ cat xorg.conf
```

Existe otra instrucción llamada “more” que también nos servirá para visualizar archivos de texto, pero la diferencia es que nos mostrará la información por página, si tecleamos la barra espaciadora nos mostrará la siguiente página, pero si presionamos la tecla “Enter” nos mostrara la siguiente línea del archivo, incluso al final de la pantalla nos muestra el porcentaje de lo que se ha desplegado del archivo; para cerrar el archivo utilizaremos la tecla “q”.

Sintaxis: `more [Archivo]`

Ejemplo:

```
$ more respaldodb.sql
```

Para poder crear links o también conocidos como accesos directos existe la instrucción “ln”, nos permitirá crear enlaces hacia los archivos para mejor uso de los recursos.

Sintaxis: `ln [Archivo] [Destino]`

Ejemplo

```
$ ln prueba.avi /home/gallo/videos/editando/version1.avi
```

Existen también instrucciones muy útiles para buscar comandos o programas ejecutables dentro de nuestro sistema, por ejemplo, si lo que deseamos es buscar un comando dentro de todo nuestro sistema utilizaremos la orden “`whereis`”, que nos mostrará las direcciones absolutas de donde se encuentra el comando, aunque la orden “`whereis`” no solo busca el comando, también buscará el código fuente y la página del manual donde encontraremos información del comando, las opciones serían para filtrar la información, “`-b`” si deseamos que sean comandos, “`-m`” si deseamos la ruta del manual o “`-s`” si queremos la dirección absoluta del código fuente.

Sintaxis: `whereis` [Opciones] [Comando]

Ejemplo:

```
$ whereis pidgin
pidgin: /usr/bin/pidgin /usr/lib/pidgin /usr/local/bin/pidgin
/usr/local/lib/pidgin /usr/include/pidgin /usr/share/man/man1/pidgin.1.gz
```

Otra instrucción que nos va a ser de ayuda será “`which`”, que a diferencia de `whereis`, sólo muestra la dirección absoluta de los archivos, pero sólo de los que se encuentren dentro del PATH del usuario.

Sintaxis: `which` [Comando] [Archivo]

Ejemplo:

```
$ which pidgin
/usr/local/bin/pidgin
```

Uno de los comandos que su uso llega a ser más frecuente es “`chmod`”, el cual nos va a permitir cambiar los permisos de lectura, escritura y ejecución de los archivos o carpetas de nuestro sistema. Como habíamos visto en el comando “`ls`”, mostraba los permisos del archivo o carpeta en una cadena de 9 caracteres en donde los primeros tres pertenecían al propietario, los siguientes 3 caracteres al grupo y los últimos 3 pertenecerían al resto de los usuarios, en donde el carácter “`r`” significaba permiso de lectura, el carácter “`w`” significaba permiso de escritura y el carácter “`x`” permiso de ejecución, ahora bien, ya teniendo esto presente, la forma más fácil de entender cómo funciona el cambio de permisos es por medio de código binario.

Tomando como ejemplo un archivo que contenga los siguientes permisos:

```
rwxr--r--
```

El ejemplo pasado puede describirse que el propietario tiene permiso de lectura, escritura y ejecución, los que pertenezcan al grupo del propietario sólo van a tener permiso de lectura, y todos los demás usuarios también permiso de lectura, ahora en base del mismo ejemplo si sustituimos la cadena de caracteres por un “`1`” donde el permiso esté activo y un “`0`” donde no, quedaría de la siguiente manera:

```
111100100
```


Ahora a la nueva cadena la separamos en subcadenas de 3 caracteres y hagamos la conversión de binario a decimal, quedaría de la siguiente manera:

744

De esta manera podemos cambiar todos los permisos de ese archivo o directorio con el comando “chmod”.

Sintaxis: `chmod [Parametros] [Archivos o directorios]`

Ejemplo:

```
$ ls -l script.sh
-rwxr--r-- 1 gallo gallo 807 feb  6  2008 script.sh
$ chmod 765 script.sh
$ ls -l script.sh
-rwxrw-r-x 1 gallo gallo 807 feb  6  2008 script.sh
```

Existe otra manera de eliminar o agregar un permiso, en donde por medio de caracteres, como el “u” que significa el propietario, el carácter “g” que representa el grupo, y el carácter “o” que anuncia a los demás usuarios, además de los caracteres “+” y “-” que denota si vamos a añadir o eliminar algún permiso.

Ejemplo:

```
$ ls -l script.sh
-rwxrw-r-x 1 gallo gallo 807 feb  6  2008 script.sh
$ chmod o+x
$ ls -l script.sh
-rwxrwxr-x 1 gallo gallo 807 feb  6  2008 script.sh
```

Como podemos ver, en el ejemplo añadimos el permiso de ejecución para el grupo del propietario. También es importante mencionar que para poder cambiar los permisos de un archivo o directorio, necesitamos ser los propietarios de los mismos o en su defecto ser el administrador del sistema.

Hasta este punto hemos aprendido todo lo necesario en cuanto a manipulación de archivos y directorios, y estamos listos para ingresar hacia lo que nos interesa, que es la administración del sistema, siendo lo que nos va a diferenciar de los usuarios finales.

Antes que nada necesitamos saber cuáles son las responsabilidades que tiene un usuario como administrador del sistema, sin importar que sea el único usuario dentro del sistema o que el sistema cuente con cientos de usuarios, la responsabilidad va ser la misma “mantener en óptimo funcionamiento el sistema”. El administrador del sistema o también llamado “root”, va a tener la libertad absoluta de poder modificar archivos o eliminarlos afectando directamente al sistema, lo que un usuario común no puede hacer sin que se lo permita el root, es importante cuando hemos iniciado sesión como root saber claramente lo que queremos y vamos a hacer porque un error podría llegar a ser perjudicial para todos los usuarios que dependan del sistema, para ello el administrador del sistema tiene que cumplir con tareas como:

La administración del hardware como lo es el perfecto funcionamiento de los periféricos de entrada y salida, así como si existiera un fallo detectarlo inmediatamente y corregirlo, además de comprobar la correcta instalación o actualización de un dispositivo.

La administración del software consistirá en controlar que el sistema operativo se mantenga estable al ejecutar acciones como la instalación, actualización y desinstalación de software, así como también debe cuidar aspectos de seguridad y rendimiento a la hora de realizarlo, incluso el administrador va a ser responsable de realizar respaldos del sistema e información de los usuarios por si sucede algún imprevisto.

La administración de los usuarios es una de las tareas más importantes porque en ella se van a controlar aspectos como el añadir y eliminar usuarios, así como si es necesario modificar la cuenta de alguno que ya exista, también se controlarán los recursos físicos y lógicos del sistema en los que un usuario tendrá acceso dependiendo de sus necesidades y privilegios, a su vez estará encargado de proporcionar ayuda a todos los usuarios del sistema, además de mantenerlos informados de actualizaciones, modificaciones, etc.

Ya teniendo claro cuáles son las responsabilidades del administrador del sistema, nos enfocaremos antes que nada a la administración de los usuarios, para empezar es necesario conocer la estructura de dos archivos claves en los que guardaremos la información de los usuarios que registremos en nuestro sistema, para ello accederemos al directorio `“./etc”` este directorio sólo puede ser modificado por el administrador del sistema, dentro de este directorio se encuentra el archivo `“passwd”` (`./etc/passwd`), en el cual cada línea del archivo estará agrupada por 7 parámetros separados por el carácter `“:”`, en la cual tendrá información de un usuario por línea, este archivo debe tener permisos del tipo `“444”`, y su sintaxis es la siguiente:

Sintaxis: `login:clave:id_login:id_grupo:comentarios:directorio:programa`

Parámetros

login: será el nombre con el cual accederá al sistema el usuario.

clave: contiene la clave del usuario encriptada.

id_login: contiene un número con el cual el usuario es identificado en varias partes del sistema.

id_grupo: es el número del grupo al que pertenece.

Comentarios: es información adicional del usuario.

Directorio: es el directorio dentro de la carpeta `“./home”` que pertenece al usuario.

Programa: sera el programa que por default se ejecuta cuando el usuario se loguea.

Existe otro archivo dentro del directorio `“./etc”`, el cual contendrá información de los grupos dentro del sistema, este archivo se llama `“group”` (`./etc/group`), este archivo estará conformado por líneas en las cuales cada una describirá información de un grupo y estará formada por 4 parámetros separados por el carácter `“:”`, la sintaxis es la siguiente:

Sintaxis: `grupo:clave:id_grupo:listado`

Parámetros

grupo: contiene el nombre del grupo.

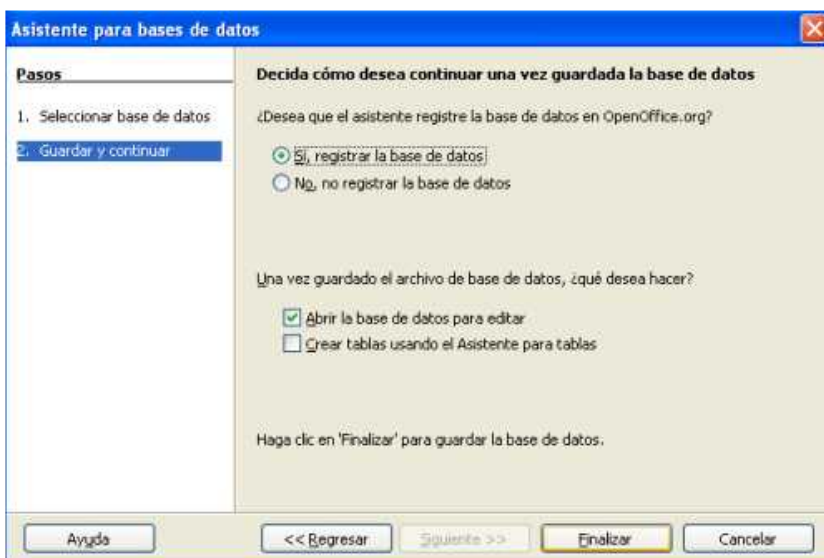
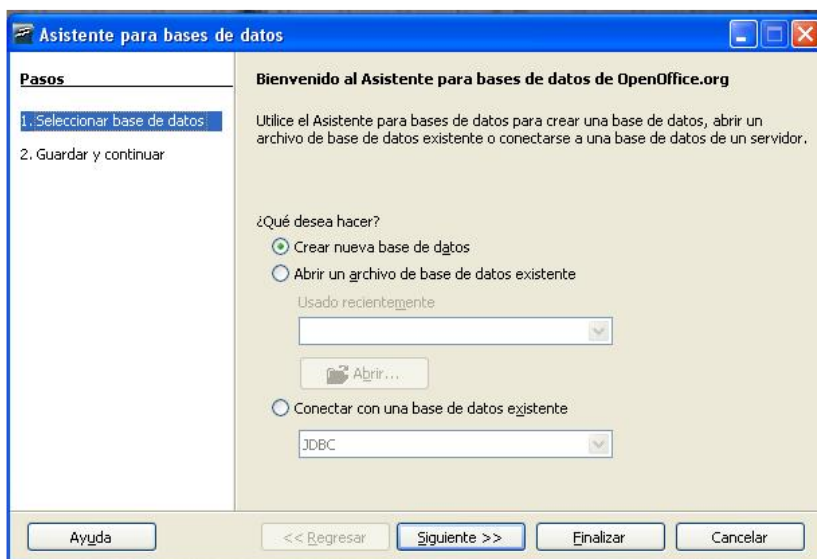
Clave: es la clave del grupo encriptada.

id_grupo: es el número del grupo por el cual es reconocido por el sistema.

Listado: contiene los nombres de los usuarios de los que pueden llegar a pertenecer al grupo aunque no lo sean actualmente y estarán separados por una coma.

Estos dos archivos son clave en la administración de los usuarios, para poder manipularlo solo es necesario un editor de textos y ser obviamente el administrador del sistema, para ello nos enfocaremos en un potente editor de textos denominado “vi”; desgraciadamente el espacio para esta sección se ha agotado, pero en el siguiente número aprenderemos a usarlo, y seguiremos hablando sobre la administración de los usuarios, también es importante que nos hagan llegar sus dudas y comentarios a través del foro o directamente a nuestros correos electrónicos, en la que podremos resolver sus dudas o ampliar más los temas tratados.

Al entrar en OpenOffice.org-Base nos encontraremos la ventana “Asistente para base de datos” donde se nos da a elegir entre crear una nueva base de datos, abrir una base de datos existente o conectar con una base de datos existente.



Al seleccionar "Crear una base de datos" nos aparecerá una ventana donde debemos registrar la base y también seleccionar "Abrir la base para editar".

Debemos registrar la base de datos si queremos que los datos sean localizables desde otras aplicaciones como Writer.

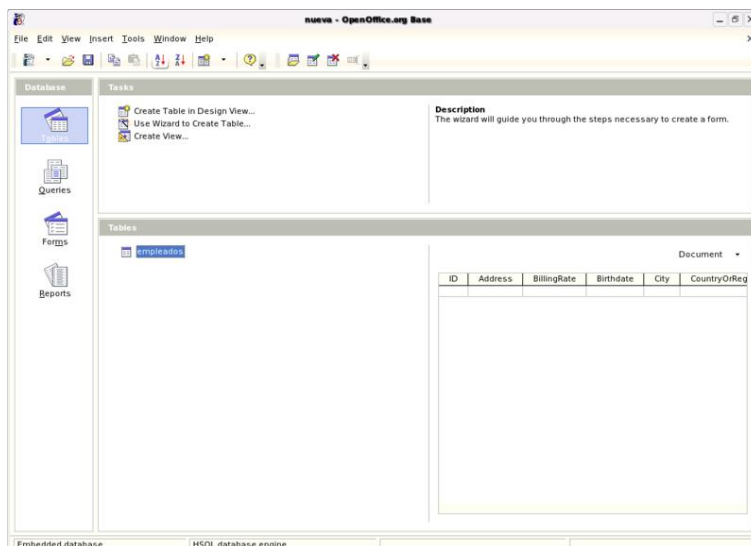
“Abrir la base de datos para editar” es obligatoria, ya que de lo contrario después de crear la base de datos se cerraría el programa.

“Crear tablas usando el Asistente para tablas” se marca si queremos que un asistente nos ayude a crear una tabla, aunque el Asistente limita bastante las posibilidades de creación de tablas. Por último debemos mencionar el directorio donde se guarda la base.

Entorno de Base

Una vez que tenemos creada nuestra base, nos aparece el entorno de trabajo de Base que se divide en 4 partes:

En la zona de la izquierda tenemos para elegir el tipo de objeto de la base. En una base hay cuatro tipos principales de objetos: tablas, consultas, formularios e informes.



En la zona central están las tareas que podemos realizar con el tipo de objeto seleccionado.

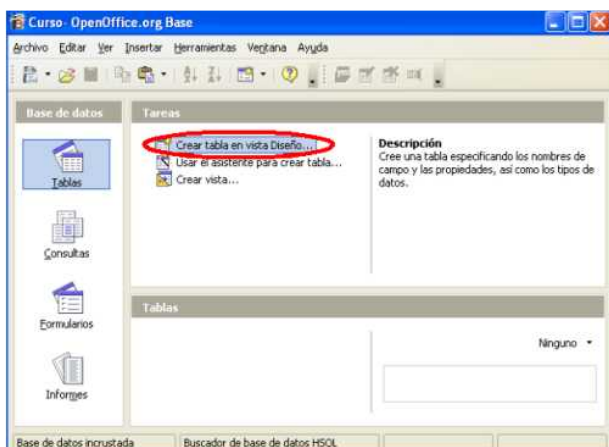
En la zona inferior, aparecen los objetos creados del tipo seleccionado. En nuestro caso, todavía aparece en blanco porque no tenemos creado ningún formulario.

En la zona superior se encuentran los distintos menús de opciones junto con los iconos más importantes de acceso directo.

En la siguiente sección vamos a ver el contenido de los distintos menús.

TABLAS

Se puede definir una base de datos como una colección de datos relacionados entre sí. Cada colección de datos almacena en forma de tablas, donde guardan información sobre un grupo de cosas con las mismas características o atributos.



Creación de tablas: En el menú de "Tablas" y, dentro de las tareas, pulsamos sobre "Crear tabla en vista Diseño", nos aparece una pantalla para definir las columnas; por cada una podemos indicar el nombre, el tipo de dato y una breve descripción sobre lo que almacenará dicha columna.

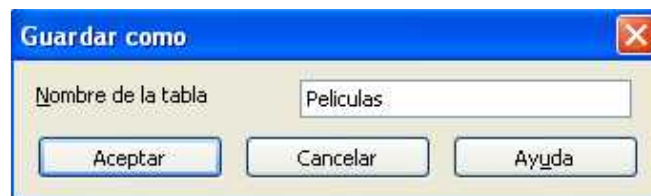


Para este tipo de tablas se suele definir un campo de tipo numérico (Id_pelicula) que debe cumplir dos condiciones: la primera, es que ese campo siempre tiene que tomar un valor para cada fila de la tabla, y la segunda, que dicho valor no puede repetirse en ninguna fila. Cuando un campo cumple estas dos propiedades (sin nulos y sin repetidos) se le llama llave primaria, y toda tabla debe tener una.

Para indicar esta llave nos situamos en la parte izquierda del campo, pulsamos botón derecho del ratón y seleccionamos la opción Llave primaria

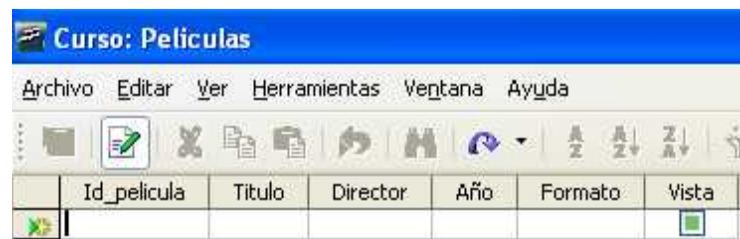


Una vez insertado este primer campo, insertaríamos el resto de los campos, una vez creados todos los campos y sus propiedades, debemos guardar la tabla. A continuación nos aparece una ventana que nos pide que introduzcamos el nombre y damos en Aceptar



Edición de datos

Inserción de datos: Tenemos que situarnos en la pantalla principal y hacer doble clic con el botón izquierdo del ratón sobre la tabla Películas o hacer un clic con el botón derecho del ratón y seleccionar la opción Abrir; nos aparece entonces la siguiente pantalla.



Podemos ver también que la columna “Vista” tiene un cuadrado, este nos indica que es un campo del tipo Sí/No y que aparezca con fondo verde significa que no hay ninguna opción elegida por el momento.

También se puede ver que al principio de la fila aparece una pequeña estrella amarilla, dicha estrella señala el lugar donde está la nueva fila a insertar.

Modificación y eliminación de datos

Para modificar el valor para algún campo de alguna fila en particular, nos situamos con el cursor en el valor a cambiar y lo reescribimos. En el caso de los campos Sí/No, basta con marcar o desmarcar la casilla. Para borrar una fila entera, basta con situarse con el cursor al inicio de la fila y seleccionar eliminar fila.

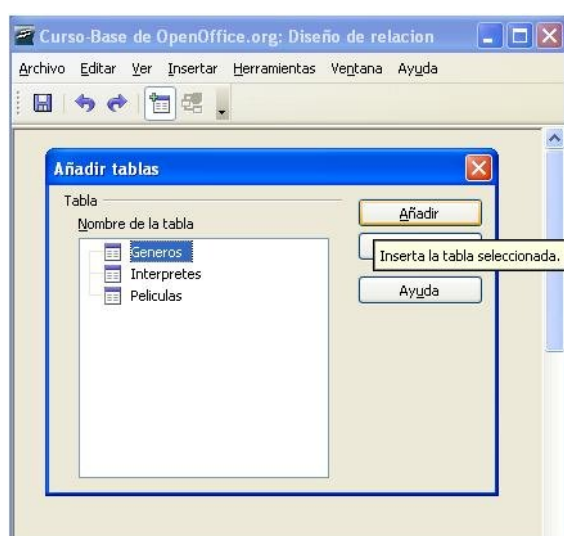
Relaciones entre tablas

Tipos de relaciones entre tablas

Uno a muchos. Este tipo se da cuando una fila de la primera tabla puede estar relacionada con muchas filas de la segunda tabla, pero una fila de la segunda sólo está relacionada con una de la primera.

Muchos a muchos. Cuando una fila de la primera tabla puede estar relacionada con muchas filas de la segunda tabla y una fila de la segunda tabla puede estarlo con muchas filas de la primera.

Uno a uno. Cuando una fila de la primera tabla sólo puede estar relacionada con una fila de la segunda y una fila de la segunda tabla sólo puede estar relacionada con una de la primera.



Establecer relaciones entre tablas

Una vez que tenemos creada la tabla nos falta indicar las relaciones existentes, para ello en la ventana principal nos posicionamos en el menú "Herramientas – Relaciones", que nos despliega una ventana donde debemos seleccionar las tablas sobre las que vamos a establecer la relación.

A continuación ya podemos establecer la relación que queremos indicar. Para ello damos clic en el icono que aparece a la derecha en la ventana.

Con las relaciones muchos a muchos primero se crea una tabla intermedia para relacionar ambas tablas y luego la relación que pasan a tener las dos tablas originales con la recién creada es de uno a muchos, respectivamente. Además, en este caso, es recomendable fijar las opciones de actualización y borrado a “en cascada”.



¿Qué es CSS?

CSS, siglas de Cascade Style Sheet (Hojas de estilo en cascada), es un lenguaje en el cual se puede tener el mismo aspecto o presentación en todos nuestros documentos HTML y XHTML. Con esto podemos evitar tener código de presentación en todos nuestros documentos, pudiendo así separar los contenidos de las presentaciones.

Antecedentes

El gran impulso de los lenguajes de las hojas de estilo se dio con el boom del Internet, era una situación complicada ya que no había un estándar, esto hacía difícil la creación de documentos que tuvieran la misma apariencia en diferentes navegadores.

En 1995 el W3C (World Wide Web Consortium) propuso la creación de un lenguaje de hojas de estilo para HTML, se presentaron 9 respuestas de las cuales se aceptaron 2, CHSS(Cascading HTML Style Sheets) y SSP(Stream-based Style Sheet Proposal), de las cuales se sacó lo mejor y se creó CSS, así todo se da de forma oficial en 1996.

¿Cuándo se usa el lenguaje CSS?

Una vez que están creados los documentos, es importante marcar los contenidos que se desean sean iguales, posteriormente se designan las funciones a cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc., las cuales nos ayudarán a definir el aspecto que se desea tener en todos nuestros documentos, por ejemplo, el tipo, color y tamaño de letra, los márgenes, separaciones, fondo, etc.

¿Cómo incluir CSS en un documento?

Los CSS se pueden incluir de tres formas distintas:

- En el mismo documento HTML. Los estilos se definen en una zona específica del documento, empleando la etiqueta <style>, sólo se pueden incluir en la cabecera del documento (o sea, en <head>).

```
<html >
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
    <title>Ejemplo de estilos CSS en el propio documento</title>
    <style type="text/css">
      p { color: black; font-family: Verdana; }
    </style>
  </head>
```

```
<body>
  <p>Un párrafo de texto.</p>
</body>
</html>
```

- Definir CSS en un archivo interno. Para mí este es el más recomendable, todos los estilos CSS se incluyen en un archivo de tipo CSS, donde la extensión es .css, aquí vamos a enlazar todos nuestros documentos al css mediante un enlace. Ejemplo:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <title>Ejemplo de estilos CSS en un archivo externo</title>
    <link rel="stylesheet" type="text/css" href="C:\Users\sonia\Documents\estilo.css" media="screen" />
  </head>
  <body>
    <p>Un párrafo de texto.</p>
  </body>
</html>
```

El estilo es:

p { color: red; font-family: Verdana; }

Como se mencionó se guarda con extensión .css en un bloc de notas distinto.

- Incluir CSS en los elementos de HTML. El peor y por lo mismo menos utilizado, ya que tiene los mismos problemas que usar etiquetas . Ejemplo:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <title>Ejemplo de estilos CSS en el propio documento</title>
  </head>
  <body>
    <p style="color: black; font-family: Verdana;">Un párrafo de texto.</p>
  </body>
</html>
```

Ventajas

Control del diseño: Separa el contenido del diseño, esto es útil cuando se quiere cambiar un aspecto del diseño de un sitio web, ya que sin hojas de estilo se tendría que cambiar página por página

página dicho aspecto, situación que cuando se usa una CSS sólo se cambia en la hoja de estilo, dándonos como resultado una uniformidad en el diseño.

Redefinición de etiquetas: Se puede redefinir el aspecto de visualización de una etiqueta, con lo que se puede lograr una sólo línea de código que todos los encabezamientos <h1> tengan un color o un tamaño determinado.

Uso de etiquetas para su misión: Gracias a poder maquetar mediante el uso de hojas de estilo, se pueden usar cada etiqueta para lo que realmente sirve.

Personalización: Nos permite definir aspectos concretos de un documento, facilitando el diseño de los documentos.

Respeto a los estándares: Será más funcional en los diferentes navegadores, el código será más sencillo y se tendrá documentación más precisa para trabajar los documentos.

Tamaño: Se logra reducir el peso de los archivos, disminuyendo el ancho de banda que se consume.

Usabilidad: Esta se da ya que el diseñador permite a los usuarios definir hojas de estilo locales, alcanzando así una mejor visualización que se adapte mejor a sus características.

Visibilidad: Se le ahorra trabajo al buscador ya que podrá indexar los documentos con mayor rapidez, reduce la utilización de etiquetas.

Desventajas

Ya que la hoja de estilo está en un archivo aparte, al visitar la página por primera vez en las conexiones lentas comenzará a cargarse el sitio sin su estructura gráfica.

Ya estoy harto de tener problemas con mi discos, ya sean duros internos, duros externos o grandes USB's. Si tú como yo has tenido problemas de espacio en tu Laptop o Desktop, he aquí un par de ideas que te ayudarán a solucionar algunos problemas.

Antes de iniciar hay que dejar claros un par de conceptos básicos:

Disco duro: Es un dispositivo de almacenamiento no volátil, es decir conserva la información que le ha sido almacenada de forma correcta aun con la pérdida de energía.

Partición: Es el nombre de cualquier división de un disco. Esta técnica consiste en dividir un disco en una o varias partes, las cuales actúan y son tratadas por el sistema como discos independientes.

Existen tres tipos de particiones; primarias, extendidas y lógicas.

Las **primarias** son las más importantes, generalmente son usadas para la instalación del sistema operativo, cualquier dispositivo de almacenamiento tiene que tener por lo menos una.

Extendidas; son particiones generalmente usadas para la preservación de archivos (música, videos, fotos, etc.), en este tipo de particiones no es posible albergar un sistema operativo ya que estas particiones no son “booteables”.

Para describir a las particiones **lógicas** es necesario decir que existe la limitación de que los dispositivos sólo pueden tener 3 o 4 particiones primarias y una extendida. Como en todas las limitaciones computacionales, debe de haber uno o más caminos para superarlas; para eso se idearon las particiones lógicas ya que estas se alojan dentro de las particiones extendidas. De este modo, dentro de un dispositivo de almacenamiento se pueden tener 3 o 4 particiones primarias, 1 extendida y dentro de ésta un número casi infinito de particiones lógicas, el cual sólo estará limitado por el espacio disponible en éste.

RAID: (Redundant Array of Inexpensive Disks, “conjunto redundante de discos baratos” o Redundant Array of Independent Disks, “conjunto redundante de discos independientes”). Este concepto es “prácticamente” aunque no siempre, el proceso inverso a la partición; es decir, tomar varios discos de poco espacio (generalmente) y unirlos para que se manejen como uno sólo. Con esto podemos hacer combinaciones interesantes, por ejemplo lectura y escritura simultanea, fragmentación de archivos para su lectura simultanea, etc.

Algo que no hay que perder de vista es que cada partición sin importar su tipo debe tener un sistema de archivos (**FAT**, **REISER**, **NTFS**, **ext3**, etc). Éste dependerá de nuestras necesidades o nuestros gustos.

Una vez terminado el breviarío cultural iniciemos con los problemas más comunes y algunos consejos para solucionarlos:

¿Quieres tener más de un sistema operativo en tu computadora ?

Por ejemplo, si quieres tener un “Windows” y un Ubuntu, en tu disco duro necesitas por lo menos tres particiones:

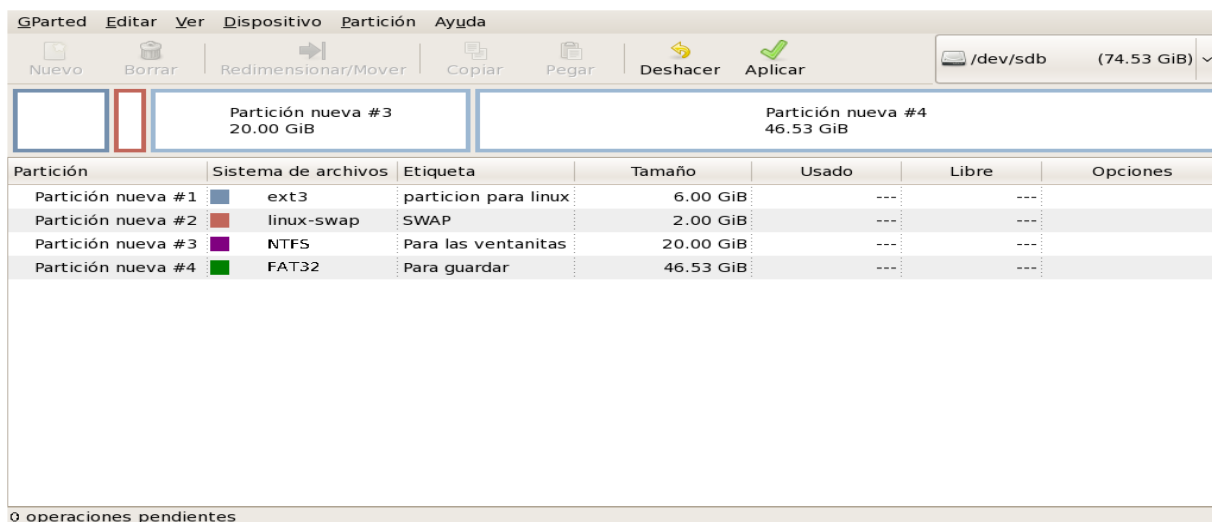
1. Una FAT32 o NTFS
2. Una nativa de Linux, ext3, ext2, REISER, etc
3. Una de intercambio "SWAP"

La **FAT** o **NTFS** para el sistema de las ventanitas, una para el sistema Linux, y el SWAP para el intercambio.

En mi opinión además de éstas, se debe incluir una partición lógica en formato FAT (como la mayoría de las memorias USB), para el respaldo de los archivos importantes o personales, de ese modo en caso de alguna falla del sistema los archivos estarían intactos.

Para hacer esto, yo recomiendo ampliamente una herramienta llamada Gparted la cual es generalmente incluida en las distribuciones live de Linux con escritorio Gnome, también la puedes descargar desde su sitio <http://gparted.sourceforge.net/>, en versión para que la cargues en USB o CD y la puedas usar en cualquier computadora (siempre y cuando sea compatible, claro está).

De vuelta a nuestro caso, tenemos que hacer la partición para las ventanitas (NTFS), las de Linux (ext3 y SWAP), y la FAT utilizando Gparted se haría mas o menos así en un disco de 80GB:



El tamaño de las particiones depende de gran modo del tamaño de nuestro espacio de disco, lo mínimo recomendado para el sistema Linux oscila entre los 4GB y los 6GB, mientras que la partición SWAP lo ideal, es que sea exactamente el doble de nuestra memoria física, es decir; si tenemos 512MB de RAM la SWAP debe de ser de 1GB, Si nuestra RAM es de 1GB nuestra SWAP debe de ser de 2GB, y sucesivamente.

No fue tan difícil ¿verdad?, bueno; con esta herramienta y un poco de paciencia podremos realizar todas las tareas de administración de discos.

Una vez terminada la tarea de particionado podemos pasar a instalación de los SO's. Como recomendación les comento que se debe de instalar primero el MS Win y después el Linux ya que el primero coloca su registro de MBR(Master Boot Record) y no permite el arranque de otros SO's, a diferencia de Linux. Si ya instalaste el Linux y te topaste con éste problema, no te preocupes, existen varios caminos para corregir tal problema; pero ese tema lo trataremos en otra ocasión.

Espero que ésta pequeña guía les haya sido útil, ¡¡saludos!!.

... the world is open source...

Domótica

La domótica se podría definir como el conjunto de servicios proporcionados por sistemas informáticos integrados en nuestras casas o en otros lugares (oficinas, hoteles, jardines, etc.), que nos ayudan en nuestras tareas diarias, mejoran nuestra calidad de vida y automatizan la vivienda, la cual debe estar al servicio del usuario.

La Domótica aporta servicios de administración energética, seguridad, bienestar y comunicación, estos pueden estar integrados por medio de redes interiores y exteriores, ya sea cableadas o inalámbricas.



La Domótica tiene como principal objetivo satisfacer todas las necesidades del hombre y su entorno.

La Domótica tiene las siguientes aplicaciones:

Iluminación: Encendido y apagado de luces.

Seguridad: Alarmas de incendio y antirrobo.

Climatización: Control de calefacción y estufas.

Cámaras IP: Cámaras de vigilancia para seguridad del hogar que se activan con la detección de calor o movimiento.

Control de Acceso: Portones, puertas, ventanas, etc.

Entretenimiento: Televisores, videojuegos, etc.

Electrodomésticos: Programación para activar microondas, lavadora, secadora, persianas y motores.

Riego: Activación automática a una hora predeterminada por el usuario.

Detección de incendios

Fugas de gas

Escapes de agua.

Alerta médica.

Teleasistencia.

Y cualquier dispositivo adaptado mediante una interfaz.

Ventajas:

- El incremento en el confort
- Ahorro energético
- Protección patrimonial
- Mejora la calidad de vida

La instalación de estos dispositivos puede tener diferentes arquitecturas como:

Arquitectura Centralizada: Cuenta con un controlador centralizado que recibe información de múltiples sensores y una vez procesada, genera las órdenes oportunas para los actuadores.



Arquitectura Distribuida: Todo el control del sistema está distribuido por todos los módulos, ya sean sensores o actuadores. Suele ser típico de los sistemas de cableado en bus, o redes inalámbricas.

Arquitectura mixta: son sistemas con arquitectura descentralizada pues disponen de varios dispositivos capaces de adquirir y procesar la información de múltiples sensores y transmitirlos al resto de los dispositivos distribuidos por la vivienda; como aquellos sistemas basados en Zigbee totalmente inalámbricos.

Las principales conexiones de la Domótica son:

Conexiones Alámbricas:

- Conexiones Inalámbricas: xDSL
- Fibra óptica
- Cable (coaxial y par trenzado)

Conexiones Inalámbricas:

- Wifi
- GPRS
- Bluetooth
- Radiofrecuencia
- Infrarrojos
- ZigBee

La Domótica está ligada al desarrollo de la computadora, inicialmente surge en la segunda mitad de los años 70 cuando sale al mercado el sistema X-10, posteriormente a comienzo de los 80, sólo tenía aplicaciones en cosas básicas como la gestión integral de la calefacción y el aire acondicionado. Con la creciente de Internet, ésta ha tomado un gran desarrollo, desde la implementación de los llamados “Edificios Inteligentes”, hasta sistemas básicos de seguridad, lo cierto es que actualmente la Domótica ya no es un lujo sino una necesidad.



¡Ayuda!, nuestro planeta se enferma cada vez más

Por: Ana Guadalupe Villa Trujillo

La tecnología avanza a grandes pasos y con ello contribuimos a que sea la misma situación con la contaminación en nuestro planeta, y obviamente de todos nosotros junto con él.



Apenas hace unos días me encontré con una publicación en enter@te en línea de la UNAM, en donde se expone un problema que me interesó bastante, así que decidí buscar más del asunto. Encontré que los desechos electrónicos que se han estado generando durante los últimos años han crecido de una manera muy considerable, y es que, ¿alguna vez te has preguntado dónde terminará tu compu, tu celular, o esa televisión viejita que estás pensando reemplazar por una digital de alta definición?

La nueva amenaza

Resulta ser que todos estos aparatos tienen materiales como: acero, cromo, cadmio, mercurio, berilio, níquel, oro, plata, paladio, plásticos, cables, baterías, etc., donde la mayoría de ellos son altamente contaminantes (principalmente por gases tóxicos) y son causantes de enfermedades serias como cáncer o problemas en el desarrollo fetal.

Según los datos de la Agencia de Protección Ambiental (Environmental Protection Agency) de EE.UU, debido a estos rápidos avances y la gran popularidad que han tenido recientemente los celulares, computadoras y demás aparatos electrónicos, de 30 a 40 millones de computadoras son reemplazadas anualmente, lo mismo que los celulares con unos 200 millones según cifras de del Programa de Naciones Unidas para el Medio Ambiente, y se estima que habrán unas 250 millones de computadoras obsoletas tan sólo en los próximos 5 años, eso sin contar el resto de aparatos.

¿A dónde va a parar?

El problema más grave es que de todos estos aparatos desechados, el 80% terminará en tiraderos principalmente de países de tercer mundo.



Durante mucho tiempo China fue el gran tiradero mundial de países desarrollados, ahora ya ha implementado políticas más

estrictas donde ya es un poco más difícil ingresar todos estos desechos, sin embargo eso no es impedimento, el problema sigue y el destino es lo único que cambia, pues según leía, ahora las grandes concentraciones de basura se encuentran en otros países asiáticos y en África oriental principalmente.

Lo irónico

Lo que llamó un tanto la atención fue el caso de EE.UU., uno de los países que más exporta esta basura a China, y siendo éste último el principal centro manufacturero del mundo, pues encontraron que el país americano por el contrario adquiere bisutería procedente del país asiático y tras realizar varias pruebas descubrieron que contienen altas cantidades de plomo, estaño y cobre; irónicamente hechas con soldadura de plomo, el mismo material para la fabricación de circuitos electrónicos.

La situación en México

Pero sin ir más lejos, aunque aún no se cuenta con cifras oficiales, en México (el segundo lugar en producción de aparatos electrónicos de América Latina) se sabe que la mayoría de las ciudades están teniendo un incremento muy importante de esta basura en sus tiraderos, y esto es porque nuestro país no cuenta con alguna política de reciclaje o para la importación de estos productos de desecho originarios de países de primer mundo.

Actualmente se cree que estos desechos crecen con más rapidez que el resto de la basura generada, más aún cuando los mexicanos cambiamos cada vez con mayor frecuencia de celular, computadora, televisión, equipos de audio, impresoras, etc.

Por otra parte, las pocas empresas que dicen tener métodos de reciclaje no operan adecuadamente, ya que lo único que hacen es quitar el material que les sirve, lo revenden y finalmente tiran el resto nuevamente.

La solución

Algunas empresas de países en desarrollo se sienten comprometidas por hacer algún cambio y se han dedicado a realizar programas para la recolección y reciclaje de basura dentro de esos países, y otras más poderosas con mercado internacional como HP, Dell, IBM, Motorola y Nokia han trabajado con campañas puestas en marcha en distintos países del mundo, que incluyen equipos no sólo de su misma producción sino otros más. La mala noticia es que aún son muy pocos los países los que abarcan y donde hay, el conocimiento de la gente es insuficiente.

En 1989 se creó la convención de Basilea haciéndose más rígida años más tarde, en ella participan 170 países y se obliga a los países desarrollados a informar a aquellos a donde se exporten los desechos sobre el peligro de contaminación que representan debido a los pobres e inseguros métodos de reciclaje que utilizan, y a las empresas generadoras a realizar productos que faciliten el reciclado, así como el uso de materiales poco contaminantes.

Por otro lado, resulta un tanto difícil combatir el problema pues el reciclaje es muy costoso y las ganancias por la reventa de los productos son considerablemente altas, aunque de entrada se piense lo contrario. La mayoría de las empresas recicladoras se encuentran en Europa y una importante en Florida, EE.UU., la Creative Recycling Systems que ha invertido 3 millones de dólares en una máquina que realiza el proceso de una forma segura, y con el reciclaje y venta de los metales preciosos se consiguen buenas ganancias.

En la misma publicación de la UNAM, encontré que ésta podría ser un factor importante para liderar estos proyectos en México. Hace un tiempo, el Dr. Fernando González Farías, investigador del Instituto de Ciencias del Mar y Limnología planteaba un proyecto para desmantelar equipos informáticos de la Universidad, todo esto mediante la implementación de manuales y procedimientos para que los centros de cómputo de la UNAM recuperaran algunos de los materiales más valiosos de las computadoras.

Por otra parte, Greenpeace demanda a las compañías creadoras de electrónicos en México, para que se responsabilicen de sus productos, incluyendo aquellas extranjeras que sólo venden dentro del territorio nacional.

¿Qué tal?, ¿habían pensado en algo de esto?, y si es el caso, ¿imaginaron lo rápido que está creciendo?, realmente yo no, y estoy sorprendida por lo efectos que pueden surgir por todo esto. Sólo esperemos que realmente todas esas medidas continúen, que los países desarrollados y grandes empresas se interesen cada vez más y que el problema tenga la suficiente difusión para que realmente se tengan buenos resultados.

Por último, aquí les dejo un link con un artículo de Chris Carroll de National Geographic que narra un poco cómo se ven las cosas desde un país con grandes tiraderos de basura electrónica, es un poco largo, pero me gustó mucho y creo que vale la pena leerlo.

<http://news.solliclima.com/index.php?seccio=noticies&accio=veure&id=2099>



La última y nos vamos

Sin dinero y sin vida



Despertarum fue realizada con:



Scribus 1.3.4



GIMP 2.4.6