

Cursos y Tutoriales

Guía para la programación de Vídeo Juegos

(Esta guía forma parte de la colección CURSOS y TUTORIALES)

Me gusta jugar, ¿y ahora que hago?

Programar un Vídeo Juego no es un juego. Quítatelo de la cabeza. Puedes intentarlo con algún lenguaje sencillo por sólo sacarte el gusto, pero en pocos días lo abandonarás. Programar un Vídeo Juego requiere disciplina, estudio, ingenio y, por sobre todas las cosas, ambición.

Dentro de las áreas en que se divide la programación (Internet, aplicaciones, sistemas) la creación de Vídeo Juegos es la más divertida, esto hay que admitirlo, pero no quiere decir que vayamos a pasar las horas de trabajo jugando. Es más, la mayoría de los programadores, según sé, nunca juegan con sus obras. Están tan cansados de haber pasado meses de pruebas de rutinas individuales, corrección de algoritmos, aplicación de inteligencia artificial y de ver siempre en sus pantallas los mismos gráficos, algunos durante más de un año, que ya nada quieren saber con él. Pero también es cierto, no lo podemos negar, que un programador debe ser, casi necesariamente podríamos decir, un experto jugador y un fanático de los Vídeo Juegos.

Ahora tal vez esta descripción encaje más con tu persona, ¿verdad? No obstante, para precisar definitivamente el perfil ideal de un programador de Vídeo Juegos debemos concluir que el mismo ha de ser amante, conocedor y jugador asiduo. Si no amas los juegos esta actividad no es para ti, por más buen programador que seas y por más que se trate de un negocio excelente.

Pues bien, dirás, yo soy todo lo que pides y estoy dispuesto a hacer el esfuerzo necesario para lograr disciplina, sentarme a estudiar y abusar lo más posible de mi desmedido ingenio, pero ¿por dónde empiezo?

En este punto debería ser yo el que pregunte. ¿Qué conocimientos tienes? No es lo mismo iniciarse en la programación de Vídeo Juegos cuando ya se han programado aplicaciones y se tiene una base sólida de los lenguajes más utilizados en esta área que comenzar sabiendo solamente cómo conectarse con el Counter-Strike para una partida multijugador. Para estos últimos les recomiendo en primer lugar una lectura minuciosa de la "Guía para el programador independiente". Este es otro de los archivos introductorios incluidos en el disco 1 de CURSOS y TUTORIALES, llamado **comoestudio.doc**, y que puedes obtener también gratuitamente de nuestra página www.macrosftwab.com en la sección **Download**. Si tú eres un neófito (burro para los más ignorantes) esta guía te indicará qué hacer y cuánto tiempo empeñar en ello para luego retomar en este punto y continuar avanzando hacia la creación de tu primer Vídeo Juego. Necesariamente tienes que aprender programación para programar, obvio. El echo de que la mayoría de los cursos de programación no estén orientados hacia la construcción de Vídeo Juegos no debe desanimarte. Este arte es uno de los más complejos y extenuantes que existe. Realizar un Vídeo Juego implica conocer todas las técnicas de programación existentes. Tienes que aprender a manipular datos, almacenarlos en una base de datos, administrarla, generar gráficos, desplazarlos aplicando leyes físicas, cálculos complejos, sonido, tienes que tener estudios sobre inteligencia y vida artificial, establecer cientos de controles, manejar periféricos, teclado, mouse, y encima tener buen gusto.

Pero en definitiva, todo esto lo harás programando, por lo que el primer paso, si aún no lo has dado, es aprender a programar. Como se explica en la Guía para el programador independiente, existen infinidad de lenguajes, cada uno con una aplicación específica, y algunos de ellos de carácter multiuso. En estos últimos vas a encontrar una variedad muy interesante y tentadora, especialmente por su facilidad de uso. Por el otro lado, entre los lenguajes específicos se cuentan aquellos que están destinados pura y exclusivamente a la

programación de Vídeo Juegos, como DarkBasic o algunos que se incluyen dentro de completos entornos de desarrollo como es el caso del DIV Game Studio, que actualmente se encuentra en su segunda versión. Es aquí, después de conocer la oferta existente, que debemos optar por la opción que mejor se adapte a nuestras necesidades. Seguramente estarás pensando que si vas a programar Vídeo Juegos y si se encuentran en el mercado lenguajes específicamente diseñados para el desarrollo de los mismos ésta debería ser tu elección. Error. Si has decidido dedicarte a esta actividad, olvida lo simple, por más tentador que resulte y ve a lo complejo. El lenguaje para programar Vídeo Juegos es, definitivamente el C++. ¿Qué tienen de malo los demás? Pues simplemente no son sistemas estándar. Para llevar a cabo un proyecto a nivel profesional necesitarás utilizar herramientas, librerías y sistemas adicionales que no son compatibles con estos lenguajes. Todos conocemos el vertiginoso desarrollo del hardware y la carrera que lleva con el software persiguiendo un punto de sofisticación ideal. Este avance nos obliga a amoldarnos a estándares establecidos que nos permitan crecer a la misma velocidad y no quedar rezagados. Trabajar en C++ nos ayudará a conseguir esto y, además, a perfeccionar motores y soportes para la creación de Vídeo Juegos en serie. Es decir, se desarrolla una base y de esa base soporte se construyen una serie de juegos que comparten ciertas similitudes pero que se comercializan como productos diferentes, algo que los lenguajes dedicados a Vídeo Juegos también permiten pero a una escala muy inferior. Por ejemplo, los juegos en primera persona (Quake, Doom) poseen un motor que fue utilizado por otros juegos que trataban diferentes temáticas. Así, con un mínimo esfuerzo y menos de la mitad del tiempo de desarrollo requerido, se producen una variedad de productos que le permiten al creador explotar muy diferentes nichos de mercado. Con la misma base se puede hacer un juego de acción para adultos como de exploración para niños.

El lenguaje a aprender es C++, pero al igual que en "Guía para el programador independiente" cabe aclarar que la dificultad implícita complica su estudio para el novato. Lo ideal entonces es comenzar por C y así una vez adquirida la base de conocimientos necesarias volcarse lentamente hacia C++, la programación visual y la utilización de librerías gráficas.

Al comenzar por C utilizarás, preferentemente, el sistema DOS. Por lo menos los primeros dos o tres meses del aprendizaje inicial realizarás pequeñas rutinas en este entorno. Al igual que aconsejo en otros artículos, sólo debes inclinarte hacia Linux si eres usuario del sistema desde tus comienzos. Si nada conoces de DOS o Windows y sabes todo sobre Linux, pues puedes trabajar sobre este último. Algunas de las librerías que posteriormente usarás son multiplataforma, por lo que en definitiva no importa mucho dónde lo hagas. Mi consejo es que elijas lo que más simple te resulte a ti, por más que sea lo que no te recomienden, para complicaciones ya tienes el lenguaje C y todas sus novedades.

¿Por dónde empiezo?

Empieza a programar en C bajo un entorno tipo consola de texto como el DOS, que te facilitará enormemente el estudio. Puedes instalar el compilador **DJGPP** o el IDE **DevC++** que trae incluido el compilador mingw32. Este último es el más recomendable ya que opera desde entorno Windows y te permite programar en C y C++ para este sistema o DOS. Ambos son gratuitos y de libre uso, y funcionan muy bien con algunas librerías gráficas como **Allegro** y **SDL**. Los encontrarás en la carpeta **Compiladores e Interpretes** → **CyC++** en el Disco 1 de CURSOS y TUTORIALES. El DevC++ es simple de instalar, sólo tienes que hacer doble clic sobre el archivo **Devcpp4960.exe** y seguir las instrucciones. Para instalar el **DJGPP** abre el archivo **Instalación de DJGPP.doc** y sigue los pasos que se indican. Consulta el archivo **links.doc** del Disco 1 de CURSOS y TUTORIALES y busca la dirección del sitio donde se alojan para encontrar nuevas versiones o más información del mismo. Junto con el DJGPP ya viene incluida la librería Allegro. Ambos son buenos compiladores, gratuitos, y tienen todo lo necesario para empezar, pero si debo recomendarte uno me inclino por el **DevC++**, trabajarás bajo Windows y podrás hacer tus primeras aplicaciones para DOS sin problema, además de ser fácil de instalar y aceptar las librerías más usadas. En CURSOS y TUTORIALES, disco 2, carpeta **Herramientas** → **Librerías** → **Directx** → **Directx8forDev-c** encontrarás los archivos necesarios para ejecutar aplicaciones con Directx 8 dentro de este compilador.

Revisa, lee y ejércitate con cada curso de C que encuentres. En el disco 1 de CURSOS y TUTORIALES en la carpeta Cursos->CyC++ hay muchos que te pueden ayudar. Visita las páginas de sus creadores para encontrar más información o una actualización de los mismos.

Sigue los pasos especificados en la Guía para el programador Independiente que se encuentra en el archivo **comoestudio.doc**. Lo necesario en esta primer etapa es que aprendas a programar como cualquier otro programador. Tienes que conocer el manejo de variables, punteros, generar sin ningún tipo de problemas funciones, transferir datos, crear y operar con estructuras, etc. Nada es específico de la programación de Vídeo Juegos, pero todo, absolutamente todo es necesario. Todo lo antes nombrado es fundamental para el funcionamiento de cualquier programa incluyendo, por supuesto, nuestros juegos. Te será de mucha utilidad ser ágil en este aspecto para encarar el aprendizaje de una librería gráfica y del diseño de aplicaciones bajo Windows sin trabas. Practica y ejercita mucho con los ejemplos que los cursos te proveen. No tengas miedo. Prueba cambios y experimenta.

Continúa con el aprendizaje del lenguaje C sin preocuparte por obtener resultados importantes, lo harás cuando empieces a trabajar con librerías. Los primeros meses practica y asimila las particularidades del lenguaje. Es bueno avanzar rápido, pero muchas veces es necesario vestirse despacio si se tiene prisa :-).

El C es un lenguaje que se utiliza para crear algunos Vídeo Juegos, pero cuando el código se torna complejo se requieren las técnicas que aporta C++ para obtener archivos fuente ordenados que sean de fácil interpretación. Es por esto que la mayoría de los juegos y aplicaciones profesionales se programan en C++ mientras que los menos son hechos en C. Pero C++ es complicado, mientras que el lenguaje C es accesible al programador novato. Aprende C, sabrás cuando sea el momento de pasar a C++.

Dedica los primeros meses a afianzar los conocimientos que tienes sobre este lenguaje. Recuerda: lo primero es saber programar. Sé constante. Programa todos los días, ejércitate y lee mucho. Una de las cosas a las que tiene que acostumbrarse un programador de Vídeo Juegos es a estar actualizado, y para esto tienes que leer toda noticia que encuentres y todo artículo relacionado con la materia. No puedes permitirte ni por un segundo estar desactualizado. Revisa el archivo **links.doc** e ingresa periódicamente a los sitios que reúnen a desarrolladores de Vídeo Juegos. En ellos encontrarás diariamente nuevos artículos sobre técnicas viejas o innovadoras que te ayudarán a obtener los mejores resultados y te ahorrarán siempre mucho tiempo de investigación y pruebas.

Al contrario de lo que puedan llegar a pensar algunos, un programador de Vídeo Juegos nunca deja de jugar. Baja de Internet todas las demos que puedas sobre los últimos juegos en el mercado. Conoce lo que hace la competencia. Piensa siempre que tu producto va en algún momento a competir directamente con ellos, y que, además, el día de mañana incluso podrías formar parte del plantel de estas enormes empresas. Así que no dejes de jugar y analizar la actualidad del desarrollo.

Otro de los puntos importantes en ésta y en etapas sucesivas son los estudios paralelos. No sólo dediques tu tiempo a la programación, empieza poco a poco a estudiar sobre márketing, e-commerce, idiomas y economía. Aprende cómo se maneja una empresa, cómo vende sus productos y cuáles son las técnicas que existen para promocionarlos. Y fundamentalmente, aprende idiomas. Saber inglés es tan necesario como saber programar. Si no conoces inglés en el mundo de los Vídeo Juegos estás perdido. Consigue cursos en libros y cassettes, CD Roms o lo que sea. Si te es posible acude a una academia, un curso privado o cómo se llame en tu país. Esto te dará, además de conocimientos del idioma, un horario y te impondrá un poco de disciplina. También, en los buenos institutos que contienen la carrera de programador de Vídeo Juegos, se enseña Japonés. Quizás no sea una prioridad, pero en el futuro tenlo en cuenta.

Organízate. Si a cada momento tomas un libro sobre cualquier cosa, lo dejas y sigues con otro será muy difícil avanzar rápido. Dedica el tiempo suficiente a cada materia y concéntrate lo más que puedas en ella. Para repartir tu tiempo podríamos estimar un 60% de estudio de programación, un 30% de inglés y un 10% para el

resto. Todo depende de los conocimientos que ya poseas y del nivel en que te encuentres en cada uno de ellos. Si has adquirido la experiencia suficiente programando en C deberías concentrar ya al menos la mitad de ese 60% en el estudio de alguna librería gráfica. Si ya has hecho tus primeros programas con una librería empieza a aplicar lo aprendido en una plataforma visual como Windows y bajo C++.

Avanza combinando las técnicas y lo aprendido.

Las Librerías

Para incluir gráficos en tus juegos y también incluir sonido y sistemas de control efectivos contamos con librerías pre-programadas. Sería una locura de nuestra parte y un suicidio profesional rechazarlas y hacer nuestras propias rutinas. Como lo que quieres es seguir viviendo y triunfar lo más rápido posible, vamos a estudiarlas y a conocerlas para saber cuál de ellas te conviene. Existen cuatro librerías principales, dos de ellas consideradas Amateur y otras dos consideradas Profesionales. Las primeras son Allegro y SDL, las segundas DirectX y OpenGL.

Allegro es una librería bastante simple de utilizar y suficientemente potente para una primera incursión en el mundo gráfico, pero sigue manteniendo el perfil de una librería no profesional y es utilizada por programadores novatos. Se han hecho infinidad de juegos con ella y existe mucha información en español, cursos y tutoriales. Es multiplataforma, lo que quiere decir que se puede generar código para Windows, Linux y otros sistemas sin modificar línea alguna, pero tiene la desventaja de no tener soporte para aceleración gráfica. Se han creado agregados como el AllegroGL pero no es la intención de su creador incorporar esta clase de soporte como nativo, por lo que deberemos recurrir a alguna otra librería y usarlas en conjunto, lo que la hace poco recomendable para programadores con grandes aspiraciones. Ubicación: disco 2, carpeta: Herramientas->librerias->allegro.

SDL también es considerada una librería no profesional a pesar de que con ella se han portado juegos comerciales desde Windows hacia Linux como el Civilización III. No obstante lo que digan, esta es una muy recomendable librería para el que recién empieza porque además de ser simple de programar es rápida y tiene la posibilidad de trabajar con OpenGL para el manejo de aceleración gráfica. Algo importante a tener en cuenta es que es también multiplataforma y como dato curioso se está trabajando para poder crear juegos para Playstation 2. Todo un aliciente y un motivo más para adoptarla como tu librería de batalla. Ubicación: disco 2, carpeta: Herramientas->librerias->sdl.

DirectX es la librería de Microsoft por lo que únicamente puedes hacer Juegos para Windows con ella (así es Microsoft :-)). Esa no es la única desventaja, también es difícil de programar (obvio si viene de Microsoft ;-)). Se requieren más líneas de código que en otras librerías para una misma acción. Pero esto no es problema para el programador avanzado que ya crea sus propias rutinas y motores 3D. Para que quede más claro, no es recomendable comenzar por DirectX a menos que te decidas a hacer Juegos para Windows exclusivamente y tengas un sólido respaldo de conocimientos sobre programación general. Te recomiendo dejar la dificultad de DirectX para una segunda etapa, cuando ya sepas manejar gráficos y demás elementos multimedia. Tarde o temprano tendrás que aprender a trabajar con ella, pero es mejor si te introduces en el mundo de las DirectX después de haber programado ya en otras librerías. Ubicación: disco 2, carpeta: Herramientas->librerias->directx.

OpenGL es una librería profesional multiplataforma, y esa es su mayor ventaja. Fue la primer librería en permitir crear juegos en 3D. En la actualidad muchas personas utilizan Linux u otros sistemas ajenos a Microsoft lo que abre nuevos e importantes nichos de mercado. Aprender a programar en OpenGL nos facilita el ingreso a estos mercados y amplía nuestros horizontes. Será fundamental conocerla e implementarla con SDL para hacer tus primeros trabajos. Ubicación: disco 2, carpeta: Herramientas->librerias->opengl.

Actualmente la mayoría de las placas de vídeo toleran todas las librerías nombradas, elegir se trata

simplemente de optar por la que más nos convenga. Te aconsejo que leas un poco más sobre cada una en los archivos y tutoriales que encontrarás en el disco 1 de CURSOS y TUTORIALES y en los respectivos manuales y documentos que acompañan a cada una de ellas.

Iníciate con SDL o Allegro. Son librerías prácticas, simples y efectivas. Con SDL, especialmente, podrás construir juegos a un nivel profesional sin tener que incursionar de inmediato por las complicaciones de DirectX, y a medida que aprendas a usarla aprenderás también OpenGL. Cualquiera de las dos son un buen comienzo y al final podrás volcarte a DirectX con mayor facilidad.

Tienes que tener en cuenta lo siguiente: un programador de Vídeo Juegos conoce todas las librerías de uso corriente. Siempre terminará especializándose en la construcción de proyectos sobre alguna plataforma específica, pero tendrá un completo conocimiento de todas las técnicas de programación.

Si lo prefieres, sigue el camino siguiente: instala **DevC++** que encontrarás en el disco 1 de CURSOS y TUTORIALES o en la página oficial (encontrarás la dirección en el archivo **links.doc**). Lee y sigue las instrucciones de instalación de SDL que figuran en el capítulo 1 del curso ubicado en la carpeta Cursos->Programaciondejuegos->SDL->Curso1 del Disco 1 de CURSOS y TUTORIALES. Puedes buscar la dirección de la página en el archivo **links.doc** para bajar nuevos capítulos o pedir ayuda o consejos a su creador.

Lee toda la información que encuentres, tanto en CURSOS y TUTORIALES como en Internet sobre esta librería y su implementación. Copia los ejemplos que vienen con la librería y trabaja sobre ellos. Busca código fuente y estúdialo. Aprenderás mucho sobre cómo estructurar un Vídeo Juego y cómo aplicar lo que ya sabes de programación en función a lo que deseas obtener.

Genera diferentes estructuras de programa hasta que encuentres aquella que se adapta mejor al trabajo que quieres hacer. No te preocupes si no tienes mucha idea de cómo hacerlo al principio, pronto sabrás, después de algo de práctica, cuál es la correcta y más funcional. Para que tengas una referencia te diré que al menos en los primeros trabajos debes crear una función central con una opción de salida (por ejemplo la presión de la tecla ESCAPE) y luego otras funciones aledañas específicas (detección de choque, control de teclado o mouse y volcado en pantalla de los gráficos, por ejemplo). Si algo ocurre será llamada la función correcta, si no continuará ejecutándose el bucle central infinitas veces hasta que sea presionada la tecla ESCAPE. Esto es a grandes rasgos una estructura básica para el correcto funcionamiento de un Vídeo Juego. Estudia otras similares leyendo los artículos que se encuentran en la carpeta Cursos->Programaciondejuegos del CD 1 de CURSOS y TUTORIALES. Encontrarás algunos para Allegro y otros para SDL. Cualquiera te puede servir si lo que quieres es conocer la construcción básica de un Juego. Además, el modo de utilización de las dos librerías, en lo fundamental, es muy similar, por lo que no te creará confusión estudiar ejemplos de una u otra.

Recuerda que en definitiva un programador de Vídeo Juegos que se precie conoce todas las librerías y sabe trabajar con cada una de ellas. Lo importante, repito, es que en un principio asimiles muy bien los conocimientos de programación fundamentales. Tienes que conocer lo mejor posible el lenguaje C y sus funciones principales antes de avanzar a C++ y a los gráficos. Practica permanentemente. Ejecuta los ejemplos y haz cambios. Lee diferentes cursos aunque hablen de los mismo, eso te va a permitir trabajar con variables, punteros y funciones de forma natural. No te apresures. Dedícale el tiempo que haga falta. Todos los días estudia un poco. No pierdas el ritmo. Si mantienes una rutina te sorprenderás luego de unos meses cuánto has avanzado.

Sigue así, y, como siempre digo, trabaja mientras aprendes y aprende mientras trabajas. No dejes de probar cada cosa que leas.

Haciendo dinero

Contrario a lo que te digan, tú tienes que hacer dinero. Tienes que ganar dinero con tus trabajos. Si no lo haces nunca podrás convertirte en un programador profesional. El paso obvio para esto es vender, y la forma obvia de vender es a través de Internet. Es increíble el incremento en los volúmenes de negocios que se operan en la red. Debes aprovechar esto al máximo. Recuerda que Internet te acerca al mundo, y el mundo es tan inmenso, existen tantos potenciales compradores, que con sólo captar una diminuta porción puedes hacerte millonario. Ve por tu porción.

Construye tu sitio en la red. Aprende lo necesario para hacerlo atractivo y funcional. Te recomendamos la lectura de nuestra Guía para la programación de páginas web que puedes descargar gratuitamente de nuestro sitio: www.macrosoftwab.com.ar. Con ella conocerás los pormenores de un buen desarrollo web. Lo importante es tener presencia en Internet y empezar a comercializar tus productos.

Los casos en que un programador de Vídeo Juegos o aplicaciones vive gracias a la venta de sus productos a través de la red son infinitos. Algunos ganan lo suficiente, otros acumulan buenos dividendos y otros afortunados se han hecho millonarios. Pienses lo que pienses, estos resultados no se deben a la calidad o la sofisticación de los productos vendidos, ni tampoco a su atractivo visual, ni mucho menos a la suerte, el nivel de rentabilidad de cualquiera de estos productos se obtiene como resultado del proceso de venta aplicado. Es por esto que una de las cosas más importantes que debes aprender es a vender tus juegos.

Lee cuanto curso o artículo sobre marketing encuentres. En CURSOS y TUTORIALES, disco 1, carpeta: Cursos->Varios->Marketing encontrarás algunos buenos consejos. En el quiosco de tu barrio seguramente podrás comprar revistas especializadas en marketing o negocios de la nueva economía. Todo lo que puedas aprender se reflejará en los resultados de tu sitio.

Como principal medida debes tener en cuenta los siguientes puntos:

- El sitio en Internet es tu vidriera. Cualquier interesado en tus juegos se debe llevar una buena impresión de ti con solo ver la página.
- Si no das buenas opciones de pago, por más bueno que sea el juego y por más bien presentado que esté, nadie te va a comprar. Contrata un servicio de cobro con tarjeta de crédito y utiliza al máximo también los sistemas de cobro de la empresa de correo.
- Las recomendaciones de boca en boca (de e-mail a e-mail para ser más exactos) también venden. Entrega un producto acabado, sin fallas y presentado en un estuche con impresión de calidad. Los juegos son un producto principalmente visual, nunca desagrades a tus clientes con malos diseños de tapa.
- Vende cuando tengas todo listo. No te apresures. No vendas algo que no está lo suficientemente probado. Si vas a regalar el producto como una estrategia de marketing o vas a entregarlo bajo la modalidad shareware, también debes entregar un producto sin fallas. Nadie compra algo fallado.

Todos los sistemas de venta están permitidos y todos producen buenos resultados si se explotan adecuadamente. La venta de un juego como shareware puede producir grandes ganancias, pero tienes que lograr equilibrios adecuados dentro de lo que entregas. De nada sirve entregar un buen producto si cansas a todo el que lo usa con fatídicas ventanas de registro o demoras en el arranque, ni tampoco si les das una demostración tan extensa que terminan cansándose de jugarla y pierden el interés. Lo que debes hacer es asegurarte de lograr que el usuario sienta deseos de más, y cuando lo has obtenido cortar el juego e indicarle cómo adquirir la versión completa. Es la técnica de la necesidad. Haces que el usuario te necesite, que necesite lo que tienes. Esto, que parece simple es muy complejo. Si no sabes o no consigues crear este equilibrio la modalidad shareware no es para ti. Prueba con amigos o parientes. Además de comprobar errores comunes en el diseño podrás descubrir qué tanto puede enganchar a la gente tu juego y cuánto más necesitas que lo haga. Utiliza a todos los que conozcas como conejillos de indias. Estudia concienzudamente los resultados y realiza las modificaciones necesarias en tu código para mejorarlos. No estés satisfecho hasta haber provocado un interés en algunos de los testers. Busca atraparlos, que no dejen de jugar, y encuentra el

límite justo para detener la acción, cuando ellos están más entusiasmados. Si produces este efecto en al menos una de las personas probadas puedes estar satisfecho.

El shareware tiene esa particularidad, pero no todos los negocios son así. Puedes decidir vender tus juegos a una distribuidora, y en este caso ya no tendrás que preocuparte por atrapar potenciales clientes, ahora deberás convencer a un negociante de que tu juego es un buen negocio. Este puede ser un buen camino cuando el producto terminado cumple con ciertos requisitos, por ejemplo: la temática del juego es la que impera en el mercado, el juego posee una calidad y una jugabilidad fuera de lo común, o trata algún tema de actualidad (como historias de moda). Después del éxito de la película Rescatando al soldado Rian, por ejemplo, aparecieron en el mercado muchos juegos de guerra. Muchos de ellos habían sido postergados en un cajón por no ser absolutamente comerciales, pero debido a este éxito repentino de las historias de guerra, vieron la luz y se convirtieron también en productos rentables. Busca un tema de moda y explótalo con tu juego. Para vender a un distribuidor esta será tu única opción.

Sea como sea que decidas trabajar, siempre caerás en la cuenta de que en solitario nunca lograrás nada importante. Trata de unirse a otros programadores. Participa de proyectos colectivos de código abierto y cuando tengas ciertos conocimientos acumulados ofrece organizar la construcción de un juego comercial. Crea tu propia empresa (que no es más que un grupo de personas persiguiendo un objetivo en común) y trata de convencer a la mayor cantidad de personas posibles para que participen en ella. Une esfuerzos y lograrás grandes resultados.

Estudia y trabaja, trabaja y estudia. Nunca dejes de aprender.

Todos los archivos, cursos y herramientas nombrados en esta guía se encuentran en los dos discos que conforman la recopilación CURSOS y TUTORIALES. Si deseas adquirirlos ingresa a www.macrosoftwab.com.ar y sigue los enlaces.

Esta guía se incluye en CURSOS y TUTORIALES conjuntamente con los archivos Guía para el programador independiente y Guía para la programación de Páginas WEB. Ambos archivos están disponibles en forma gratuita en nuestro sitio.

Si necesitas contratar espacio en un servidor para alojar tu página web puedes utilizar nuestro servicio de hosting económico. Tendrás la posibilidad de subir tu página, lanzar tu primer negocio virtual y experimentar con las herramientas más utilizadas del mercado (PHP, MySQL, ASP, Java, etc...) al más bajo costo. Ingresa a www.macrosoftwab.com.ar y elige la opción Hosting.

Si quieres recibir novedades o noticias del mundo de la programación inscríbete en www.macrosoftwab.com.ar, es gratis. Recibirás la información que necesitas en tu casilla de correo todos los meses y actualizaciones de esta guía o nuevas ediciones.

Un saludo

Comunidad Macro

Macrosoft WAB

www.macrosoftwab.com.ar

Cursos y Tutoriales

Guía para el Programador Independiente

(Esta guía forma parte de la colección CURSOS y TUTORIALES)

Introducción

Esta pretende ser una guía para que cualquier persona, en cualquier nivel de estudios, pueda comenzar a aprovechar el material contenido en los CD de la selección CURSOS y TUTORIALES. Aquí le daremos algunos consejos sobre programación en general (cursos recomendados, instalación de aplicaciones necesarias, etc...) y principalmente le indicaremos cómo empezar a estudiar programación y convertirse en un programador independiente.

Nota: Si usted ya se ha decidido por la programación de Páginas Web o Video Juegos, le recomendamos que posterior a este artículo lea los archivos: **paginasweb.doc** y **juegos.doc**, en los que se trata con profundidad aspectos más específicos de estos temas.

Dividimos estos consejos en tres grandes áreas:

- **Principiante**
- **Medio**
- **Avanzado**

Principiante

Empezar de una vez por todas

¿Cómo empezar cuando no se sabe nada y se tienen todas las ganas? Simple, por el principio. Entonces, ¿cuál es el principio?...

Se puede elegir un lenguaje sencillo para dar los primeros pasos, pero también es bueno definir desde el comienzo nuestras metas como futuros programadores y optar por lo que mejor cumpla estas expectativas aunque sea un camino más difícil de lo esperado.

La oferta en estos momentos es tan grande que cualquiera podría confundirse y terminar desilusionado al primer intento. La decepción es habitual en los programadores novatos cuando no cuentan con la guía adecuada.

Si seguimos una línea respetando la dificultad implícita en cada lenguaje de programación, tendríamos que comenzar, por ejemplo, por BASIC, seguir con PASCAL, C, VISUAL BASIC, C++, y profundizar aquí lo relacionado con la programación visual, el desarrollo de aplicaciones multimedia, manejo de bases de datos o gráficos en 3D de acuerdo a cuál es nuestro interés principal. Esto a grandes rasgos y obviando muchos lenguajes y muchas técnicas específicas, incluyendo la programación de páginas web.

Pero hay que elegir porque el tiempo es oro, y el tiempo que nos llevaría respetar esta línea o cualquier otra que se plantee podría ocupar muchos años y producir muchas decepciones. Hay que elegir y URGENTE!!!!

Construir una aplicación utilizable, desarrollar las bases de un sólido y confiable sistema administrativo o finalizar nuestro primer motor 3D para los futuros Video Juegos que estamos dispuestos a dar a luz nos puede costar algunos meses o hasta años de esfuerzo, no podemos, por lo tanto, perder demasiado de este precioso tiempo en aprender cientos o miles de cosas que en definitiva vamos a descartar. Podrías, por ejemplo, empezar a programar en el lenguaje BASIC utilizando el viejo compilador QBasic que se incluía en la instalación del sistema operativo DOS. Aprender con este lenguaje es relativamente sencillo, con algunos ejercicios empezamos a familiarizarnos con las variables, rutinas y comandos. Es fácil habituarse a su estructura ya que cada línea va numerada y es muy difícil perderse. ¿Cuál es la razón por la que no debas

empezar con BASIC? Simple, porque no sirve para nada. Lo nombro simplemente porque muchas personas me consultan sobre este lenguaje que según referencias es el más simple para dar los primeros pasos. El BASIC es un lenguaje caduco, con el que estaremos desperdiciando, siendo optimistas, tres o cuatro meses preciosos en los que habrías aprendido lo mismo en un lenguaje afín a lo que usarás en el futuro para aplicaciones reales y útiles, y únicamente hubiese bastado poner un poco más de buena voluntad de tu parte. Aprender BASIC es más simple que un lenguaje estructurado en funciones, pero esta última es la técnica utilizada por todos los sistemas de programación modernos y acostumbrarte a trabajar con ella desde el principio evitará malos vicios y te impulsará rápidamente hacia prácticas más avanzadas como la construcción de Clases en programación orientada a objetos.

Vamos a ser claros: la mayoría de las aplicaciones comerciales se hacen en C++. Los programas más exitosos y útiles fueron desarrollados en este lenguaje. El C++ es y, al menos por el momento, será el lenguaje a dominar para un desarrollador de aplicaciones o Video Juegos. El resto de los lenguajes cumplen funciones específicas, como el diseño de páginas web, pero el programador profesional, dedicado al desarrollo de aplicaciones o Video Juegos que no conozca C++ no podrá nunca obtener resultados realmente profesionales, y, lo que es más importante, programas comercializables.

Entonces, te preguntarán, ¿por qué no aprendo C++ y listo? Bueno, la respuesta no es muy complicada: porque es difícil. ¿Y entonces, qué hago? Pues, como dijimos anteriormente, empezar desde el principio, y ese principio es C.

El lenguaje C te proveerá de los conocimientos necesarios para trabajar en C++, te dará una introducción a las "Clases" por medio de las funciones (las "Clases" son una particularidad del C++, así como de todos los lenguajes de programación orientada a objetos) y conocerás a través de él todas las características que tiene que tener un programa, su funcionamiento, su estructura, los medios de control, variables, punteros y más. Todo esto absolutamente reutilizable cuando te vuelques de lleno al C++. Lo que aprendamos en C queda y lo seguirás utilizando en el futuro. Además, muchas aplicaciones operativas pueden hacerse con él. Dependiendo del nivel al que quieras llegar, C podría ser, incluso, el único lenguaje que necesites.

Ahora que ya sabes que el C es el lenguaje a aprender, ¿cuál es el primer paso que debes dar? Bien, el paso es leer todo lo que se te cruce sobre el tema. Especialmente los cursos que puedes encontrar en CURSOS y TUTORIALES, la mayoría de los cuales fueron seleccionados especialmente para el programador novato, y también toda la información que pueda obtenerse en las páginas web de donde provienen estos cursos. Las direcciones las puedes obtener del archivo **links.doc** o de los mismos cursos.

Descarga de estos sitios actualizaciones o información adicional. Consulta con la comunidad de programadores existente en ellos o en cualquier otra página en Internet y quítate de inmediato las dudas. Adhiérete a las listas de correo. No es bueno pasar días enteros estancado en algo que alguien con un poco más de experiencia y en pocos segundos nos puede enseñar. Ayudemos y dejémonos ayudar. Hoy día aprender a programar y desarrollar aplicaciones comerciales no es tarea de una sola persona. Y no solo en cuanto a conocimientos hablamos, también el código provisto gentilmente por otros nos puede facilitar nuestra tarea. Por ejemplo, para la programación de Video Juegos se utilizan librerías gratuitas pre-programadas que nos permiten concentrarnos en las tareas de desarrollo y no en puntos menores y complicados como el manejo de la placa de video o la reproducción de sonido. Si queremos construir un visualizador de imágenes encontraremos cientos de funciones gratuitas y de uso libre que decodifican los diferentes formatos ahorrándonos mucho esfuerzo. Varias de estas librerías están incluidas en los CD de CURSOS y TUTORIALES junto a ejemplos y código útil. Gracias a estos aportes, con una simple línea de código podemos enviar un dibujo a la pantalla o reproducir un tema musical completo sin dedicarnos al estudio de complejos formatos y reglas estándar.

Todo bien, ¿pero en qué sistema operativo trabajo? El entorno de trabajo para el lenguaje C es el sistema operativo DOS (aunque puedes programar también bajo Linux, lo que no recomiendo a menos que éste sea tu

sistema nativo, es decir, que TU hallas nacido con él :-)). Para la programación visual (aplicaciones para Windows) se utiliza normalmente el C++. Es por esto que mientras aprendas C trabajarás bajo DOS y poco a poco aprenderás lo relacionado con Windows y todo su entorno junto a las innovaciones de C++. DOS, a pesar de ser un sistema operativo antiguo, se continúa utilizando y es muy recomendable como punto de partida para evitar las extremas complicaciones que nos impone Windows. Linux es el mejor sistema operativo en el mercado de PC's pero le falta camino por recorrer para convertirse en el ideal. Vamos a ser realistas y sinceros: nos guste o no, la mayoría de los cursos están destinados a la programación bajo Windows, las aplicaciones necesarias para programar, las más útiles y fáciles de conseguir son para Windows, y lo más más más IMPORTANTE de todo es que Windows tiene entre el 80 y el 90 % del mercado de PC's del mundo y programar para este sistema te permitirá acceder a él sin trabas.

Cabe aclarar que muchos programadores aprenden programación visual con Visual Basic, un Basic visual que prácticamente nada tiene que ver con el viejo QBasic antes nombrado. Si debo aconsejarte al respecto te diré que olvides Visual Basic y de C pases a C++ sin escalas. La relación entre C y C++ es casi total mientras que Visual Basic difiere una enormidad y será como empezar todo desde cero otra vez. Las verdaderas aplicaciones se programan en C++ y algunas, incluso, en C. Recuerda, un programador no puede perder el tiempo. TU TIEMPO ES ORO!!!!

Entonces, ya definimos el camino a seguir. Debes estudiar C y C++. ¿Qué herramientas usas para trabajar? Los compiladores (el programa que traduce el código C o C++ en un lenguaje que la computadora puede entender) más utilizados entre los de uso gratuito son el **DJGPP** y el **DevC++** (este es un IDE, un entorno de desarrollo, que trabaja junto al compilador mingw32).

El DevC++ es un entorno amigable que opera bajo Windows. Es muy recomendable su instalación y utilización. Muchos cursos y muchos desarrolladores trabajan con él. Te permitirá diseñar programas tanto para DOS como para Windows. Búscalo en la carpeta Compiladores e Interpretes->CyC++->Dev CPP en el Disco 1 de CURSOS y TUTORIALES. Haz doble clic sobre el archivo y sigue las indicaciones.

En la carpeta Djgpp ubicada también dentro de Compiladores e Interpretes->CyC++ en el Disco 1 de CURSOS y TUTORIALES encontrarás todos los archivos de instalación de este otro compilador junto a un archivo en formato Word con las indicaciones necesarias para su correcto funcionamiento. Este paquete incluye un editor (programa donde se escribe nuestro código) llamado Rhide.

Una vez instalado el compilador y después de haberlo explorado un poco, es aconsejable crear una carpeta de trabajo (puedes llamarla "trabajos" :-)) donde almacenar todos tus proyectos. Todo lo que generes a partir de aquí deberás grabarlos en esta carpeta o en subcarpetas dentro de ella. ¿Por qué aclaro esto? Bueno, porque una de las cosas más importantes si quieres avanzar rápido es SER ORDENADO. No importa si la ropa en tu habitación está desparramada y sucia tirada por todo el piso, tú no vas a ser modisto y mucho menos modelo, tú vas a ser programador. Los programas, utilidades, herramientas y tus propios trabajos deben estar ordenados y ser de fácil acceso.

Consulta el archivo **indice.doc** del CD 1 de CURSOS y TUTORIALES y comienza por leer los cursos de C que son específicamente para principiantes. En el archivo **links.doc** encontrarás también enlaces otros sitios importantes donde podrás bajar útiles recursos y actualizaciones. No te desanimes si al principio no entiendes, nadie lo hace. Todos, incluso los programadores más avanzados, tenemos que leer y releer muchos cursos sobre lo mismo antes de entender de qué se trata. Es bueno, en este momento, leer también información sobre la estructura interna de un ordenador para saber qué hace un programa cuando se ejecuta, dónde se aloja, de qué manera se comunica con los periféricos, cómo administra la memoria, etc... Busca en la carpeta Manuales->Hardware del CD 1 de CURSOS y TUTORIALES para encontrar artículos que hablan al respecto.

Dedícale al lenguaje C todo el tiempo y esfuerzo que puedas durante algunos meses. Plantéate objetivos a

corto plazo. Metas realizables. Por ejemplo, es muy aconsejable obligarse a estudiar al menos dos horas por días (cuanto más mejor) lo que nos permitiría tener una buena base en algo menos de seis meses. Seis meses es un buen plazo para dedicar a nuestro primer lenguaje y a nuestra primer experiencia en programación. No intentes adelantar el plazo o avanzar precipitadamente hacia C++ o la programación visual. Durante este tiempo estará muy bien que te concentres en el C y en el entorno DOS, aunque parezca anticuado (y por supuesto que lo es). Pero es el paso correcto. La programación visual no es pura y no comparto la opinión de aquellos que afirman que hay que iniciarse en la programación visual. Es como querer construir una nave espacial sin saber cómo funciona un automóvil. Puedes quedarte tranquilo, todo lo que aprendas durante estos primeros meses formará una base sólida y los conocimientos adquiridos no deberán ser descartados ya que estás trabajando sobre un mismo lenguaje. C y C++ tienen múltiples similitudes. Tú sigue así que vas bien aunque te parezca que tus primeras rutinas no tienen una utilidad real.

El lenguaje C, incluso, sienta las bases para trabajar con otros lenguajes y otros sistemas, como es el caso del PHP, un lenguaje de programación de aplicaciones para Internet que te permitirá fácilmente trabajar con bases de datos SQL y otorgarle interactividad a tus páginas web. El C te ayudará, también, a iniciarte en PHP.

¿Ves qué importante es saber elegir la ruta adecuada?

El mejor camino a seguir, y por ende el más rápido, es el que está poblado de relaciones. Si tuvieras que hacer un viaje, y de tener la posibilidad de optar, ¿cuál camino elegirías, el que habitan personas que hablan otro idioma o el que está repleto de gente que puedes entender? Siempre será más fácil si eliges estudiar aquello de lo que ya posees una base de conocimientos sólida. No vas a pasar de reparar camiones a preparar comidas rápidas. El camino a seguir, si lo que deseas es crecer rápido, será dirigir o trabajar en una empresa que construya vehículos, por ejemplo. Siempre debes elegir la ruta más accesible para ti. Es por esto que tienes que optar por un sistema operativo que te sea familiar, si toda tu vida trabajaste en Windows, no lo rechaces porque muchos te digan que es un sistema impuesto por el mercado, ni tampoco rechaces Linux por lo mal que puedan hablar de él si éste es tu territorio. De igual manera, busca los lenguajes que naturalmente se relacionan. C, C++, PHP, Java, son lenguajes que están íntimamente conectados. No te adaptes a otras tecnologías porque ellos opinen que es lo nuevo o lo revolucionario o lo que sea. Ellos no saben lo que tú necesitas, sólo tú puedes hacerlo. Sigue este camino y crecerás a la velocidad de la luz.

Practica mucho. Apenas empieces a estudiar comprenderás que es la práctica lo que afirma los conocimientos en tu mente y te descubre nuevas técnicas o formas de hacer las cosas. Prueba cada ejemplo que te de el curso que estás siguiendo. Hazlo incluso cuando ya sabes cómo funciona. Esto te ayudará a recordar la sintaxis del lenguaje (cuándo va un punto y coma, cuándo hay que cerrar o abrir llaves, etc). Prueba y modifica. Hacer pequeños cambios también ayuda y enseña. **NO TENGAS MIEDO DE PROBAR!!!!** A lo sumo tendrás que reiniciar el equipo después de un tilde. Pero la mayoría de las veces aprenderás más de lo que crees.

Experimentar, probar y practicar. **NO LO OLVIDES!!!**

Si algo no te sale no te detengas ahí. Continúa con la siguiente lección, quizás en ella se te aclare un poco más el panorama y puedas volver atrás después. Si aún así siguen las dudas consulta otro curso que trate el mismo tema, seguramente una explicación diferente te ayude a comprenderlo. En computación las cosas se pueden hacer de diferentes maneras y siempre habrá alguien que lo hace mejor y de un modo más comprensible. Encuentra estas mejores formas de hacer las cosas y aplícalas. Pero lo importante de todo esto: **NO TE DETENGAS**, continúa aunque no entiendas del todo lo que lees. Relee cuantas veces sea necesario cada capítulo hasta comprender. En las horas designadas al estudio dedícate a pleno a él. Muchas veces es necesario leer tres o cuatro libros completos sobre el mismo tema para conseguir asimilarlo. Utiliza todas las técnicas que creas necesarias. Subraya, remarca, toma apuntes. Utiliza un cuaderno especial para anotar lo que consideres más importante (funciones más utilizadas, caracteres especiales, operadores matemáticos, etc). Lo único que importa es aprender y adquirir práctica. **NO IMPORTA COMO LO HAGAS!!!!** Aprende todo y en el menor tiempo posible. Recuerda, tú no tienes que dedicar mucho tiempo a aprender, tienes que dedicar tu

tiempo a trabajar, a programar y crear. Tú eres un programador, no un estudiante, el tiempo que dediques a aprender es tiempo que no puedes dedicar a tus proyectos. Entonces, aprende trabajando. Trabaja mientras aprendes. Programa constantemente. El código que escribas durante tu aprendizaje seguramente te servirá después para tus aplicaciones.

Mientras avances en este lenguaje y bajo este entorno (DOS) sufrirás la tentación de hacer alguna aplicación gráfica o agregar gráficos a tus ejercicios. Olvídalo. No lo hagas. No te preocupes demasiado por la parte gráfica, trabaja únicamente con texto (caracteres ASCII) y has tus programas rápidamente, sin detenerte en el aspecto visual. **OLVIDA POR EL MOMENTO LO VISUAL.** Lo que importa en esta etapa es la programación, aprender cómo se hacen los programas y no lo lindos que queden. Ni siquiera tienen que ser útiles. Tú, durante estos meses no harás más que rutinas inútiles que sólo servirán para ejemplificar lo que aprendes y nada más. No tiene que interesarte en este momento construir una aplicación completa y tampoco tienes que dejarte vencer por tus grandes aspiraciones. Una vez más te lo digo: **LO IMPORTANTE ES APRENDER.** Los gráficos y la vistosidad no justifican el tiempo perdido si de todos modos nuestro objetivo final es hacer aplicaciones para Windows que ya cuenta con su propia interfase gráfica y su propia forma de programarla. Inclusive, si nuestro interés son los Vídeo Juegos o aplicaciones multimedia, para la generación de gráficos aprenderemos posteriormente el uso de librerías que se encargan por completo del manejo de los mismos.

Pasado el periodo inicial, deberás comenzar a plantearte la posibilidad de encarar cosas más serias como la programación visual. Tendrás que empezar a trabajar en C++ y esto te llevará tiempo. Pero si has cumplido la primer etapa correctamente y en los plazos debidos tendrás los conocimientos necesarios para afrontar este nuevo y mayor desafío. Sigue los mismos pasos que al principio. Estudia los cursos para principiantes en C++ que encuentres en CURSOS y TUTORIALES o en Internet. Consulta el archivo **links.doc** para conocer direcciones importantes que visitar. Lee los artículos de sus creadores, intercambia opiniones con programadores ya experimentados a través de listas de correos o foros y, lo más importante, practica y practica mucho.

Debes estudiar C++ directamente sobre el entorno Windows. No trabajes en DOS. Aprende a programar en este entorno visual con C o con C++. Muchas veces, de acuerdo a los cursos que sigas, empezarás directamente en C++, cambiando de una sola vez lenguaje y plataforma. Está bien, aunque si te cuesta un poco busca otros cursos de programación visual que trabajen en C para no complicarte demasiado con las clases y todo lo relacionado con la programación orientada a objetos. De todos modos, el objetivo final es programar en C++ bajo Windows, no importa cómo llegues a hacerlo.

Un buen consejo quizás en esta etapa sea adquirir un compilador profesional de C++. Puedes continuar utilizando **DJGPP** u otro compilador de uso libre y gratuito como el mingw32 que viene junto al entorno de desarrollo **Dev-C++**, pero llegado a este momento sabrás, con toda seguridad, si la programación realmente te interesa o si lo tuyo es freír papas en un puesto en la plaza. Si estás decidido a continuar te recomiendo adquirir un compilador que te brinde todo y sea práctico y fácil de utilizar. Microsoft Visual C++ o Borland Builder C++, por ejemplo, son paquetes completos de aplicaciones diseñadas para facilitar la tarea del programador. Si te interesa la programación de aplicaciones pueden servirte los sistemas de Borland, para Vídeo Juegos lo recomendable es el de Microsoft y para aplicaciones administrativas, como programas de stock, ventas, etc, lo ideal es el Borland Builder. Prácticamente todos los paquetes profesionales incluyen administradores de bases de datos, objetos predefinidos, editores de recursos e importantes complementos. Son fáciles de programar y la mayoría de los cursos o artículos escritos para este lenguaje están dirigidos a estos compiladores y traen ejemplos para ejecutar en ellos.

Por último recuerda: aprender es lo importante, pero el tiempo es oro. Combina los dos procesos, el aprendizaje y el trabajo. Aprende mientras trabajas y trabaja mientras aprendes.

Medio

Tomar la decisión fundamental

Lo importante es establecer objetivos claros y realizables. Has llegado a una etapa en la que has adquirido ciertos conocimientos pero aún no sabes de qué modo aplicarlos o crees que debes aprender más para poder empezar a trabajar en ello. Pues bien, si este es tu caso, manos a la obra. Busca los cursos que necesites leer para fortalecer lo que sabes. Dentro del CD número 1 de CURSOS y TUTORIALES encontrarás artículos específicos sobre algunas materias de nivel medio y avanzado. Busca los sitios a los que pertenecen estos cursos y asesórate con sus creadores y demás expertos. Consulta otros programadores y descarga también cualquier artículo, actualización o nuevo curso que encuentres que consideres pueda ayudarte. No dejes nada de lado. Lee todo lo que creas necesario. Échale un vistazo rápido a lo que ya conoces. Ten esto en claro: lo principal es que fortalezcas lo que ya sabes y lo complementes. Este trabajo no debería llevarte más de un mes. De inmediato debes comenzar a trabajar. No esperes más. Busca ideas. Internet es una fuente de ellas. No te desanimes si descubres que algo que pensabas hacer ya está hecho. NO IMPORTA!!! Windows se hizo cuando ya había otros sistemas operativos y no hace más que generar millones y millones de dólares. No importa si algo ya está hecho. Tú no tienes que hacer nada revolucionario. Tú tienes que hacer algo que se pueda vender, un programa comercializable. Incluso no importa si es más feo que el que ya está, la mayoría de las veces tu éxito no dependerá de lo que vendes sino de cómo lo vendes.

Trabaja mientras aprendes, aprende mientras trabajas.

Tu objetivo principal deberá ser siempre el DINERO. Tú tienes que hacer dinero con tu trabajo. ¿Materialista? ¡Ja! que piensen lo que quieran, mientras tú ganes dinero con tus programas podrás seguir programando, seguir aprendiendo, mejorar tu técnica, vivir de lo que te gusta e incluso vivir bien, y te encontrarás realizando una actividad que en cualquier momento puede hacerte millonario. Tú nunca serás millonario como oficinista, ni como chofer, ni como profesor, pero si eres programador y te lo propones puedes empezar a trabajar en un garage y al año siguiente dirigir una gran empresa o emplearte en ella con sueldos y beneficios fuera de lo normal. Siempre depende de ti, de lo que quieras para ti.

Concéntrate en tu actividad pero siempre da un vistazo a lo que la rodea. Por ejemplo, puedes hacer aplicaciones en C++ y esto no involucra nada relacionado con Internet, pero debes aprender HTML, PHP y MySQL para tener una base de lo que es necesario a la hora de presentar y vender un producto en la red. Conoce sobre Marketing. Todo producto ha de publicitarse para poderse vender. Nada se vende sin el proceso de marketing adecuado. Hay miles de cosas necesarias que un programador no puede obviar a la hora de buscar el éxito.

Construye tu página web. No importa si ya has terminado tu trabajo, no tienes porqué tener lista una aplicación y un juego para tener presencia en Internet. Tampoco hace falta que ya conozcas HTML, puedes construirla con cualquier editor HTML sin programar una sola línea. Ni siquiera necesitas una conexión permanente, únete a cualquier proveedor de Internet gratuito o contrata un servicio mínimo. Lo único que necesitas es unos minutos para subir tu página terminada y algunos segundos para revisar tu e-mail todos los días. Dale una visión lo más profesional que puedas. Visita páginas de empresas dedicadas al desarrollo de aplicaciones para saber cómo hacerlo. Un buen ejemplo lo encontrarás en la página de Gabriele Scibilia http://www.geocities.com/gabriele_scibilia/index-i.html (en italiano). Publica en ella tus primeros trabajos. No vendas nada hasta que no consideres que esté realmente listo para salir al mercado. Vende únicamente aquellos programas que han sido probados por ti y por todo el que se te cruce enfrente. Pero muestra lo que sabes hacer. En el sitio coloca un enlace hacia una página con información exclusivamente sobre ti. Has un breve currículum para que cualquiera pueda conocerte. No pienses que los visitantes son personas comunes, siempre escribe los textos como si lo hicieras para un gran empresario a punto de contratarte. Aprende a venderte. Lee cuanto artículo encuentres sobre e-commerce. Busca en el CD 1 de CURSOS y TUTORIALES, carpeta Cursos->Varios->Marketing o consulta en el archivo **links.doc** del CD 1 los enlaces relacionados con Marketing y con estos mismos artículos para ampliar o conseguir nueva información.

IMPORTANTE: No sólo programes, descubre cómo vender tus programas.

Quizás me digas que el marketing y todo eso a ti no te interesa, pues te interesará cuando en lugar de una docena de copias vendas miles. El marketing hace la diferencia, es lo que vende. Si tu producto es un lujo de programa pero no sabes venderlo nadie lo comprará. Tienes que esforzarte por hacer un buen producto y luego hacer el doble de esfuerzo por venderlo. Recuerda: no importa lo bueno o malo que sea, con un poco de marketing se vende. La diferencia entre ofrecer un producto malo o bueno está en la reputación que te crearás, factor decisivo a la hora de atrapar clientes. Lanza a la venta tus programas cuando ya los consideres aptos para operar. No te apresures, ten en mente siempre que mejorar una mala reputación puede costarte años de enorme esfuerzo.

Lanza tu página lo más rápido que puedas. Busca servidores gratuitos o de bajo costo. Preferiblemente sin banners publicitarios. Existen servicios muy económicos que te brindan la capacidad suficiente para empezar. Revisa los enlaces sobre hosting del archivo **links.com**. Macrosoft WAB te ofrece hosting por muy poco dinero y con abonos anuales. No gastes demasiado en esto. Lo importante es tener presencia en Internet, no construir un super portal y conseguir tráfico. Más adelante podrás concentrarte en otorgarle interactividad y en colocarle más opciones y un buscador y lo que sea. Lo que importa en esta etapa es tener presencia en Internet, poder darle tu e-mail y la dirección de tu página a cualquiera que lo pida (empresario, gerente, colega, etc). En cuanto a la dirección (el dominio) intenta registrar uno simple de recordar. Esto no es demasiado importante ahora que la gente acostumbra usar buscadores pero sí queda pintoresco en una tarjeta de presentación o al pie de un e-mail. Puedes registrar tu dominio nacional. En la mayoría de los países esto es gratuito. Por ejemplo, los dominios .ar (www.tudominio.com.ar) se registran en www.nic.ar y no cuestan nada. Averigua en tu país cuáles son los requisitos y procura encontrar una dirección interesante, fácil de recordar y sobretodo pegadiza para tu página.

No dediques meses a esto último, en una semana, más tardar 10 días, deberías tener todo listo. Una vez instalada podrás ingresar cada vez que mejoras alguno de tus programas y publicarlo. **ATENCIÓN:** no hagas público aquello que consideras original o forma parte de algún proyecto importante. Lo único que tienes que hacer es mostrar lo que sabes hacer, **no** lo que haces. Aunque consideres la posibilidad de ofrecer el producto de forma gratuita en el futuro, espera hasta estar seguro. Todo se debe vender, de una u otra forma. La estrategia de brindar un producto gratuito es para captar clientes a los que luego se les venderá una actualización del mismo producto u otra cosa, no se trata de tirar todo tu esfuerzo a la basura. Muchas empresas lo hacen, regalan algo para luego cobrar por otra cosa relacionada. Los que publican software en versión shareware luego te cobran para registrarte, los que dan algo freeware te cobran una nueva versión, te venden el soporte o consiguen patrocinio. Nadie puede sobrevivir sin dinero. No importa cuál sea tu estrategia, siempre busca la forma de obtener un beneficio. No regales nada.

Ya estás en Internet, tienes un e-mail donde cualquiera puede contactarte desde cualquier lugar del mundo, tienes los conocimientos necesarios para empezar a trabajar. Ya eres un programador. ¿Y con eso qué? No importa lo que sepas ni lo que tengas, tienes que decidirte. Si aún no lo has hecho, decídetelo ya. Tienes que decidir qué vas a hacer, a qué vas a dedicarte en primer lugar. Siempre a corto plazo. No puedes pensar que porque eres programador y ya estás capacitado puedes hacer cualquier tipo de programa en cualquier momento. Si te gusta la música y el procesamiento de sonido, busca construir aplicaciones relacionadas (reproductores de MP3, editores de audio, compresores, descompresores, etc). Si te gusta el procesamiento de imágenes trabaja sobre visores, programas de retoque, morphing, etc. No te desanimes si ya está hecho. ¡TODO ESTA HECHO! No dejes de hacer un reproductor de MP3 porque ya existe WinAmp. Si te fijas, este último es uno de los reproductores más populares y a la vez más simples existentes. Cualquiera podría fabricar un producto mejor. Ellos regalan el programa pero crearon a su alrededor toda una comunidad de usuarios interesados por la música a los que pueden venderles lo que quieran relacionado con este rubro. A veces (muchas veces) no es tu programa lo que se vende, sino que tu programa es un medio para vender otros programas o publicitar otros negocios. Amplía tus horizontes. Toma caminos paralelos y descubre nuevas posibilidades de explotar tus trabajos y hacer dinero. Por ejemplo, puedes crear un visor de archivos de

imágenes y ofrecerlo gratuitamente porque en su ventana publicitarás una página que se dedica a vender otros programas hechos por ti sobre retoque fotográfico o colecciones de fotografías, texturas, gráficos o cualquier otra cosa relacionada con esto. Hacer el visor no te costará mucho esfuerzo. Dentro del archivo **links.doc** encontrarás enlaces a páginas donde se ofrecen librerías gratuitas y de código abierto que se encargan del procesamiento de imágenes. El que se interese por tu visor seguramente comprará la versión completa o confiará en la calidad de los demás productos que le ofreces y se convertirá en cliente en el futuro próximo. Incluso puedes no explotar el proyecto tú mismo sino venderlo completo a alguien que esté dispuesto a explotarlo por su cuenta. Las posibilidades son infinitas.

Cualquiera sea el rubro al que te dediques concéntrate en eso. Sé lo más específico que puedas. Por ejemplo, si lo tuyo son los Video Juegos, trabaja sobre sonido, o procesamiento de imágenes 3D, o generación y tratamiento de escenarios, creación de niveles. No trates de hacer todo porque no llegarás a nada. **Lo necesario es saber un poco de todo pero especializarse en algo.**

Decídete y pronto empezarás a ver los resultados. Si la decisión no te resulta fácil busca un nicho del mercado poco explotado. Existen muchos. Aunque parezca mentira, la mayoría de las veces los mercados no son explotados como se cree. Siempre existe lugar para uno más. A veces con cambiar sólo algunas cosas de lo que ya existe se puede conquistar un sector importante y generar una comunidad de usuarios adecuada para obtener rentabilidad.

Concéntrate en lo que elijas y luego sigue estudiando. Trabaja y estudia. Lee todo lo relacionado con lo que estás haciendo. Aprende lo indispensable para empezar a trabajar en eso hoy, para complementar los conocimientos que ya tienes y terminar las aplicaciones que están en etapa de desarrollo. No sigas los pasos que puedan parecer adecuados. Por ejemplo, si debes construir unas pequeñas aplicaciones para administrar tu nueva empresa, pues trabaja en Visual Basic aunque antes no te lo haya recomendado. Fortalecerá tus conocimientos sobre programación visual y podrás hacer tu trabajo rápidamente. Solo tú sabes cuales son los pasos a seguir. Nada está establecido. Recuerda que esto es una guía a seguir, no un mapa de ruta. Yo jamás podría saber lo que tú necesitas, sólo sé lo que tu puedes conseguir. Si alguien te pide un trabajo o descubres una brecha en el mercado que puedes aprovechar no dejes de hacerlo porque debes utilizar un lenguaje no recomendado o una técnica considerada ambigua. Recuerda: **NO IMPORTA COMO LO HAGAS**, lo que importa es que esté hecho, funcione correctamente y se pueda vender.

Si lo que necesitas es crear páginas web, empieza, por supuesto, con HTML y sigue con lenguajes que te permitan otorgarle al sitio interactividad. PHP es el más recomendado porque trae una interfase incorporada para operar con el administrador de bases de datos MySQL y además es muy similar a C. Busca en CURSOS y TUTORIALES, en el CD 1, el archivo **paginasweb.doc** que trata específicamente este tema.

En el CD 2 de CURSOS y TUTORIALES también encontrarás herramientas muy útiles para el desarrollo de aplicaciones (colecciones de íconos, texturas, aplicaciones para páginas web, applets, etc) que pueden ayudarte en este nivel. Algunas son de libre uso y otras tienen restricciones, consulta los archivos que acompañan estos recursos para asesorarte al respecto.

Avanzado

No basta con saber, hay que demostrar que se sabe

Vender tus aplicaciones debe ser el objetivo primordial. Debes pensar en función de esto: hacer DINERO. ¿Por qué? ¿Por qué debo pensar en hacer dinero si lo único que quiero es aprender? ¿Por qué buscar una ganancia si lo mío es un hobby? Porque lo necesitas. Necesitas hacer dinero con tus aplicaciones porque si no deberás trabajar en otra cosa para mantenerte y dejarás a la programación para las horas libres, lo que no te dará el tiempo suficiente para convertirte en un verdadero programador ni, seguramente, hacer nada de lo que tenías planeado. No tendrás el tiempo suficiente para aprender. Si quieres ser programador debes pensar en

vivir de eso. Si no, te lo aseguro, tarde o temprano dejarás la programación de lado. No creas lo que dicen. No dejes que te inhiban. Hacer dinero es lo que hacen todos. Tú puedes ganar dinero programando. Puedes HACER MILLONES si te lo propones y estás dispuesto a hacer el esfuerzo necesario. ¿Cómo crees que empezó Bill Gates? ¿Cómo crees que empezaron la mayoría de los programadores y dueños de enormes empresas? Aprende de ellos. Aprende de todos los que están delante de ti. Programar es un negocio de miles de millones de dólares. Entonces, ¡aprende a hacer de tu trabajo un negocio!

Olvídate de los que se ríen de ti. Aléjate de ellos y de las personas negativas que te rodean. Únete a todos los que sabemos que podemos conseguir enormes logros y obtener infinitos beneficios. Si tú estás en un nivel avanzado de estudios y aún no has podido rentabilizar tu trabajo o no has logrado obtener los beneficios que buscas es simplemente porque no te has vendido lo suficientemente bien. Tienes que aprender a vender y a venderte. No sólo comercializa tus productos sino también a ti mismo y a tu empresa. Estudia todo lo relacionado con Marketing, comercio electrónico e e-bussines. Mejora tu página en Internet, publica tus productos y empieza a venderlos a través del sitio. Si aún no sabes como hacerlo aprende HTML, PHP y MySQL. Lee los cursos ubicados en la carpeta Cursos→Internet→PHP del CD 1 de CURSOS y TUTORIALES. Especialmente los que relacionan PHP y MySQL. Con esto podrás construir tu primer carro de compras y procesar los formularios que necesites. Contrata algún servicio de cobranza electrónica que procese pedidos con tarjeta de crédito. Algunos de estos también te brindan un carro de compras y otras facilidades que te ahorrarán trabajo. En el archivo **links.doc** del CD 1 de CURSOS y TUTORIALES encontrarás enlaces a estos servicios. Ofrece múltiples opciones para el pago de tus productos. Agiliza el sistema. Ingresa a un sitio de venta de software, copia lo bueno del proceso de venta y anula o mejora lo malo. Regionaliza el sistema de cobro. Ofrece envíos por correo y cobro revertido en los países que puedas hacerlo. Enfoca tus esfuerzos hacia la comunidad internacional. Trata a los usuarios de cada país como si tú pertenecieras al suyo. No ofrezcas una imagen extranjera. Dedicar una página al menos a cada país. Ofrece tus productos como si tuvieras tu oficina al lado de la casa de tus clientes. Que ellos sientan que pueden confiar en ti al pagar lo que pides y que recibirán pronto y sin preocupaciones ni contratiempos lo que están a punto de comprarte.

Escribe textos interesantes y atrayentes sobre el final de las descripciones de tus productos y junto a los botones de compra. La última decisión que toma el cliente es la que importa. De nada vale que le agrade el producto si decide comprarlo después. Tienes que convencerlo de que lo haga **ya**. Frases como oferta de la semana o descuento por Navidad, aunque parezcan retrógradas, venden. Le están diciendo al subconsciente del cliente que a lo mejor cuando regresen ya no encontrarán el producto o habrá aumentado. Utiliza todas las artimañas que conozcas para definir una compra.

Construye también dentro del sitio formularios para consultas o sugerencias. No coloques simplemente un e-mail. Por regla general son muchas más las personas que rellenan un formulario que las que envían la información por e-mail. El formulario debe ser simple y te debe asegurar obtener la dirección de e-mail del cliente. Por todos los medios posibles asegúrate de que la dirección de e-mail sea la correcta. No importa si olvidó su nombre o colocó iniciales o si no especificó su dirección, tú no vas a enviarle nada por correo sino hasta que compre, por ahora lo que necesitas es su e-mail porque se trata de un potencial cliente que debes contactar.

Contesta de inmediato todos los mensajes que recibas. Si lo crees necesario, programa un sistema de autorespuesta para que el cliente reciba un e-mail segundos después de haber enviado la consulta. En el mensaje hazle saber que estás trabajando para responderle o evacuar sus dudas, aunque tú aún no sepas nada de la existencia de esta consulta. No importa. Lo que importa es que el cliente se sienta satisfecho y atendido.

Apenas tengas la respuesta envíala. Trata de ser personal. De manejarte como si además de un cliente fuese un amigo. Bríndale toda la información que puedas. Si te piden mejoras a tus programas o te brindan consejos para ampliar sus capacidades y tu no crees estar capacitado para hacerlo o no piensas hacerlo nunca, responde de todos modos que sus sugerencias serán tenidas en cuenta. No niegues nunca. Porque en definitiva, ni

siquiera tú mismo sabes si en el futuro no harás lo que él te aconseja. Las cosas cambian permanentemente y quizás termines trabajando sobre lo que nunca creíste ser capaz.

Lo importante: RESPONDE SIEMPRE. Sea cual sea la consulta y su origen. El que el cliente viva en Rusia no impide que compre nuestros productos. De echo existen infinidad de programas hechos en India, Rusia y países que no son de habla hispana y apenas conocen el inglés que se han hecho terriblemente populares, como el compresor/descompresor WinRar, que además es un claro ejemplo de cómo regalar un producto para después venderlo. Eugene Roshal, su creador, lo viene haciendo por años, cuando Windows 95 aún se estaba programando.

Piensa globalmente. Sé cosmopolita.

Y vende. No tengas miedo. Tus programas se pueden comercializar al igual que cualquier otro. Vende programas o servicios o proyectos. Todo se puede convertir en dinero. Estudia y encuentra la manera de rentabilizar lo que haces. El dinero te ayudará a continuar con tus estudios, porque por más avanzado que estés siempre necesitarás estudiar. Este es el sentido de todo lo dicho desde un principio. Estudia mientras trabaja y trabaja mientras estudias.

Como dijo Thomas H. Huxley, Trate de aprender algo acerca de todo y todo acerca de algo.

Todos los archivos, cursos y herramientas nombrados en esta guía se encuentran en los dos discos que conforman la recopilación CURSOS y TUTORIALES. Si deseas adquirirlos ingresa a www.macrosoftwab.com.ar y sigue los enlaces.

Esta guía se incluye en CURSOS y TUTORIALES conjuntamente con los archivos Guía para la programación de páginas web y Guía para la programación de Vídeo Juegos. Ambos archivos están disponibles en forma gratuita en nuestro sitio.

Si necesitas contratar espacio en un servidor para alojar tu página web puedes utilizar nuestro servicio de hosting económico. Tendrás la posibilidad de subir tu página, lanzar tu primer negocio virtual y experimentar con las herramientas más utilizadas del mercado (PHP, MySQL, ASP, Java, etc...) al más bajo costo. Ingresa a www.macrosoftwab.com.ar y elige la opción Hosting.

Si quieres recibir novedades o noticias del mundo de la programación inscríbete en www.macrosoftwab.com.ar, es gratis. Recibirás la información que necesitas en tu casilla de correo todos los meses y actualizaciones de estas guía o nuevas ediciones.

Un saludo

Comunidad Macro

Macrosoft WAB

www.macrosoftwab.com.ar

Cursos y Tutoriales

Guía introductoria a la programación de Páginas Web

(Esta guía forma parte de la colección CURSOS y TUTORIALES)

Introducción

Internet ya no sólo involucra a programadores y gente del ambiente informático. Hace varios años ya que la gran red afecta a todo el mundo, sin excepción. El cambio en la economía mundial a sido significativo, pero en especial este efecto se aplica a la micro economía, aquella que abarca los pequeños negocios y nuevos emprendimientos.

La vieja frase el mundo es un pañuelo se aplica ahora perfectamente sobre este mundo globalizado. Antes, quizás, el concepto o la imagen que tenían las personas del mundo, no se extendía más allá de su región e incluso su ciudad. Hablar de la pequeñez del mundo tenía un significado simbólico. Hoy, en cambio, este mundo realmente se ajusta a tal definición. Los nuevos sistemas de comunicación han hecho posible semejante hazaña. Unir y acercar a todos los habitantes de este planeta ha sido una tarea que Internet a desempeñado efectivamente.

Pero este prodigio, así como las redes internacionales de computación, son un fenómeno relativamente nuevo. Hablamos de algo que apenas hace diez años comenzó a hacerse popular en el mundo entero, y el cambio producido recién por estos días empieza a percibirse. Muchas personas, muchísimas personas, ignoran estos efectos y permanecen aisladas del mundo, tanto comercial como afectivamente. Internet a generado nuevos lazos comerciales y afectivos que aún muchos ignoran, aunque para quién conozca sobre el tema resulte difícil de aceptar.

Literalmente, hoy día puedes vender una manzana a un cliente en Moscú cuando tú vives en Quito. Los costos, la delicadeza de una manzana y las restricciones aduaneras de una economía mundial decadente harán que esto sea aún impracticable, pero el hecho es que si tu producto cumple con estas restricciones y los cargos por transporte no incrementan su valor excesivamente, Internet te brindará la posibilidad de venderle a quién sea, dónde sea que esté, cualquier cosa que se te ocurra.

Las barreras impuestas por la distancia han sido eliminadas. Ya no importa en qué lugar te encuentres, incluso tu empresa puede estar únicamente alojada en Internet, sin oficinas físicas más que tu propio hogar, instalada en un servidor en Estados Unidos o Europa, mientras los productos o servicios que ofreces son adquiridos desde Shanghai hasta España.

Los límites de esta diversificación los establecen las características del producto que la empresa fabrica o comercializa. Y conocer cómo vencer estos límites es de vital importancia para el éxito en lo que ahora llaman nueva nueva economía. Sí, como lo lees. Es que los negocios ahora se adaptan rápidamente a Internet, a las nuevas tecnologías y al avance de los sistemas de comunicación, y crecen al mismo ritmo. Lo que hace apenas un año daban en llamar sólo nueva economía ahora denominan, un tanto confusos, nueva nueva economía.

Nunca los negocios estuvieron tan relacionados con la evolución tecnológica. Nunca se vieron afectados de tal manera. Hoy un negocio puede fracasar de manera estrepitosa por el simple, y hasta sorprendente echo de no aplicar nuevas tecnologías o no saber aprovecharlas.

Es por esto fundamental que aquella persona decidida a programar sitios web y sistemas de redes conozca de negocios, y hombres de negocios se relacionen con las innovaciones, nuevos desarrollos, aplicaciones y códigos del software y hardware que ahora rigen sus actividades.

Nicholas Negroponte llamó a su libro Being digital, ser digital, lo que no fue en realidad más que un buen consejo. La necesidad de ser digital, de convertirse en un conocedor de las tecnologías actuales y su aplicación es imperiosa. Estés del lado de la calle en que estés, siempre tu objetivo será llevar el negocio adelante (sea tuyo o de quien sea) y para esto deberás tener bien presente esta importante relación, y satisfacer la nueva necesidad de información y conocimientos sin excusas. Tanto en los negocios como en la vida, debes ser digital o perecer en el intento.

No obstante, más allá de lo que esto pueda llevar a pensar, los negocios siguen operando del mismo modo que

antes. Y esto también hay que tenerlo bien en claro. Muchos proyectos fracasaron por abusar de lo nuevo. Internet no es más que una herramienta que te brinda agilidad, alcance y capacidad de comunicación. Una nueva vidriera para tus negocios. Pero las reglas fundamentales que los rigen siguen siendo las mismas de antes. Incluso los sistemas de pago más utilizados son los mismos que existían antes que Internet viera la luz: tarjetas de crédito, transferencias bancarias, pago efectivo, cheque o envío de fondos a través de empresas de correo.

Por eso, tienes que pensar en todas las posibilidades que tu negocio o el negocio para el que trabajas posee y luego explotarlas al máximo aprovechando esa extensión y nuevo alcance que te asegura la red. Ubícate dentro de tu región pero piensa globalmente. Recuerda permanentemente que ya no te cuesta nada tratar con clientes distantes. Busca las posibilidades que el producto te ofrece, estúdialas a fondo y aplica las técnicas que consideres convenientes para hacer rendir al máximo tu empresa o el sitio que estás construyendo por encargo.

A veces, al visitar la página web de una empresa, nos sorprende su desarrollo simple, incluso sin grandes despliegues visuales. Es que precisamente, el diseño gráfico no representa la mayor dificultad a la hora de tener presencia en Internet, las páginas web de las empresas más exitosas de la red no son un alarde artístico. Aunque sí son importantes los detalles gráficos, y hay que prestarles atención, la dificultad principal en el manejo de nuestra empresa o pequeño negocio de forma virtual se presenta por todo lo que estar en la red significa. A través de nuestro sitio en Internet deberemos atender al cliente, hacerlo sentir comfortable, mostrarle nuestros productos, captar su interés, arrastrarlo hacia la decisión de compra, venderle el producto y cobrarle. Todos los mecanismos de venta actúan sin que nosotros conozcamos cara a cara a nuestro cliente, incluso, la mayoría de las veces, sin que siquiera le enviemos un mensaje por e-mail. Nuestro sitio en Internet es nuestra imagen y lo que nos llevará hacia el triunfo o nos hará caer en el peor de los fracasos. Deberemos saber y aprender a aplicar todas las técnicas que conocemos en el mundo real en un negocio virtual, y en las dosis justas. Tarea no siempre fácil y que muchas veces, por inexperiencia, a destruido proyectos completos.

Entonces no sólo se trata de aprender a programar una página en Internet, sino, también, conocer los mecanismos de venta y marketing que nos permitan que esa página, ya sea para nosotros o la empresa para la cual estemos trabajando, funcione y sea rentable.

Estos mecanismos son muchos, pasando desde la atracción visual hasta el convencimiento textual. Pero la meta principal de todo webmaster será otorgar a sus sitios interactividad. Esta palabra, nombrada infinidad de veces en estos días, es la clave para el éxito de tu negocio o de los sitios que diseñes y publiques en la red.

Para que tus páginas sean interactivas es necesario utilizar lenguajes especiales que te permitan manejar bases de datos y aplicaciones gráficas.

Para empezar a aclarar un poco el panorama y referirnos específicamente al tema en el que se centra esta guía, vamos a estudiar estos lenguajes.

Programación de una página web

El navegador es el programa por el que todo usuario de Internet se comunica y visualiza las páginas web (Microsoft Internet Explorer, Netscape, Opera, etc). Este programa reconoce un único lenguaje llamado HTML. El HTML es el lenguaje con el que se programan todas las páginas de Internet, pero tiene severas limitaciones. Fue creado con el propósito de generar aplicaciones de tamaño reducido que pudieran ser rápidamente transportables a través de la red. Con el crecimiento de la utilización de los servicios electrónicos, también creció la demanda de nuevos y mejores sistemas, por lo que surgieron otros lenguajes que suplieron las falencias técnicas del HTML. Estos nuevos lenguajes se utilizan interna o externamente. Desde HTML se puede llamar a rutinas o programas completos escritos en otros lenguajes o por medio de instrucciones especiales se pueden incluir sentencias dentro del mismo HTML y extender así sus

posibilidades.

Estos nuevos lenguajes incorporan principalmente la posibilidad de manejar bases de datos, obtener información del usuario y desplegar resultados. Los más populares son ASP, PHP, Perl y Python. También existen otros pero más inclinados hacia aplicaciones gráficas como Java, Javascript o Flash, por citar los más conocidos.

El panorama es mucho más amplio del que se presenta y los pormenores de la aplicación de estos lenguajes son demasiados para envolvernos en su estudio, pero afortunadamente podemos concentrarnos en pocas herramientas manteniendo la misma productividad.

ASP, PHP, Perl o Python cumplen prácticamente la misma función y están destinados al mismo propósito. Es por esto que optar por solamente uno de ellos es suficiente y muchos programadores profesionales desconocen el funcionamiento del resto. Debido a la similitud con C, el lenguaje PHP es uno de los más utilizados y la mejor opción que tienes. Si ya conoces programación en lenguaje C o C++ pasar a PHP prácticamente no requiere esfuerzos. De igual modo, por más que desconozcas estos lenguajes, PHP es uno de los más simples de aplicar y con pequeñas líneas puedes otorgarle interactividad a tus páginas, además de aprender un lenguaje relacionado con los más populares y poderosos existentes.

En PHP se crean funciones, clases, y se utilizan variables exactamente de la misma manera que en C o C++. Estos tres lenguajes están íntimamente relacionados. Trabajar sobre ellos te ayudará a crecer a pasos agigantados.

Para que la interactividad sea completa es necesario almacenar datos y manipularlos, para esto se utilizan bases de datos. El administrador de bases de datos más utilizado es MySQL y, como era de esperarse, PHP incorpora una interfase para operar con este sistema sin ninguna clase de librería adicional y todo listo para funcionar, ya configurado.

El primer paso para comenzar a trabajar es instalar un programa editor de HTML. Las páginas que diseñes, a pesar de contener código en otros lenguajes o hacer llamadas a rutinas externas siempre estarán escritas en HTML. Existen dos clases de editores para este lenguaje. Los de código y los visuales, también llamados editores WYSIWYG (iniciales de what you see is what you get, lo que ves es lo que obtienes). Los primeros están destinados a programadores más profesionales ya que todo lo que se quiere construir se hace programando, mientras que los últimos pueden ser utilizados por personas sin conocimiento de programación y trabajan de manera similar a cualquier editor de texto. Lo ideal es uno que convine ambos aspectos para poder empezar a trabajar mientras, por medio de los resultados obtenidos, aprender los fundamentos del lenguaje y empezar a hacer modificaciones a nivel código, algo que todo programador que se precie sabe y debe efectuar permanentemente.

En CURSOS y TUTORIALES, disco 2, carpeta Herramientas->Editores->EditoresHTML se incluyen algunos con los que puedes empezar. Prueba algunos y descubre con cuál te sientes más cómodo. El mejor editor de los existentes en el mercado y de uso profesional es Dreamweaver de Macromedia. La empresa provee una versión de prueba que puede obtenerse de su sitio en Internet: www.macromedia.com. Lamentablemente no es un sistema gratuito y esta demostración tiene fecha de caducidad, por lo que sería bueno pensar en realizar un esfuerzo y adquirirlo. Piensa en que podría ser quizás la única herramienta paga que necesites.

Practica generando páginas de toda clase. Enlázalas por medio de hipervínculos para empezar a tener una idea de los pormenores del desarrollo de un sitio completo. En principio descubrirás que mucho se trata de imitar y del ejercicio de la prueba y el error. Investiga cómo otros webmasters organizan sus sitios y trata de imitarlos. Esto te dará la pauta de la dificultad implícita en el desarrollo de un sitio interactivo y ágil.

No te preocupes en principio del funcionamiento de los formularios o las respuestas automáticas por e-mail, son elementos que pronto aprenderás trabajando con PHP y MySQL. Tú no necesitas que nadie complete y envíe sus datos ahora. Lo que tú necesitas es conocer cómo se construye una página y cómo se estructura un sitio completo. Haz distintos sitios con diferentes temáticas. Coloca artículos completos, aunque no sean tuyos. La idea no es publicar sino saber cuáles son las reacciones de la página cuando hacemos inserciones de diferente extensión a la esperada y controlar estos efectos.

Practica mucho y revisa el código que el programa editor va generando automáticamente. En CURSOS y TUTORIALES, Disco 1, carpeta Cursos->Internet->HTML encontrarás algunas referencias básicas y cursos sobre este lenguaje. Estudia qué función cumple cada línea y haz modificaciones. Examina los resultados y busca adaptarlos a tus necesidades.

HTML no es un lenguaje exacto (en realidad no es un lenguaje de programación), tiene muchas deficiencias y notarás que al exhibir tus páginas en un explorador de Internet se producen cambios inesperados. No te alteres, todo tiene solución, aunque lleve mucho tiempo encontrarlas.

Conéctate a Internet y visita las páginas incluidas en el fichero **links.doc** que forma parte del disco 1 de CURSOS y TUTORIALES. Ingresa a los foros temáticos y deja un mensaje con tus dudas. Inscríbete en las listas de correo que encuentres sobre este tema y otros que sean de tu interés y da a conocer a los integrantes tus inquietudes. Baja de estos sitios cuanta información adicional encuentres. Nunca pierdas más de dos o tres horas en un problema. Si no lo pudiste resolver en ese período de tiempo muy difícilmente lo hagas tú solo. Consulta con quienes saben, un simple e-mail puede salvarte de horas perdidas y un gran dolor de cabeza.

Pasado el tiempo de práctica en HTML, del cuál sólo tú puedes juzgar su duración, y cuando ya conozcas y manipules lo fundamental de este lenguaje, instala PHP y un servidor en tu computadora. Elige uno de los paquetes de instalación que se encuentran en la carpeta Compiladoreseinterpretes->PHP del disco 1 de CURSOS y TUTORIALES. El más recomendable es **PHPTriad**, que instala PHP, MySQL y el servidor Apache, y los deja listos para funcionar.

La mayoría de los cursos que se encuentran en el disco 1, carpeta Cursos->Internet->PHP, incluyen instrucciones de instalación para cualquier sistema operativo. Siempre recomendamos trabajar en Windows. No porque este sea el mejor sistema operativo, ni siquiera se acerca a eso, sino porque es el más popular, y eso es una ventaja a la hora de rentabilizar nuestro trabajo.

Comienza por leer los cursos básicos de PHP que encuentres en el CD 1 de CURSOS y TUTORIALES, en la carpeta Cursos->Internet->PHP. Practica los ejercicios. Estos no deberían producirte demasiados problemas ya que por lo general son básicos y están muy bien explicados. Lee diferentes cursos aunque hablen de lo mismo, esto te ayudará a afianzar los conocimientos. Visita las páginas de sus creadores para encontrar actualizaciones o nuevos artículos y para evacuar tus dudas. Nunca dejes de consultar con aquellos que conocen más que tú, aunque te avergüence. Entra en contacto con la comunidad de programadores existente en el sitio por medio de sus listas de correo o sus foros (todo sitio los tiene). Si después de algunas horas de estudio y esfuerzo no puedes resolver un problema no te quedes estancado, envía de inmediato un e-mail solicitando amablemente una respuesta. Siempre habrá algún alma generosa que evacuará tus dudas y te permitirá dar el siguiente paso.

Aplica todo lo que aprendas de inmediato. Siempre es bueno trabajar sobre un pequeño proyecto que no tenga demasiada importancia para hacer pruebas y experimentar.

Después de algunos días, o algunos meses (tómate el tiempo suficiente para adquirir una buena base en el manejo de PHP), empieza a estudiar MySQL (podrías también haberlo hecho en forma conjunta, eso queda a tu criterio). Lee los cursos que explican cómo aplicar PHP con MySQL. En CURSOS y TUTORIALES, disco 1, carpeta Cursos->Internet->PHP encontrarás varios que enseñan las funciones fundamentales para trabajar

con bases de datos. En estos cursos se explica lo fundamental de MySQL. Aunque esto será suficiente para tus primeros proyectos, es conveniente que profundices algo más, pero hazlo a medida que trabajas, cuando lo necesites. Lo primero será aprender a crear una base de datos, generar un nuevo registro, hacer consultas y eliminar datos. Estos movimientos básicos de datos son suficientes para cualquier aplicación inicial y no representan complicación alguna.

Puedes instalar programas como MySQL Front ubicado en el Disco 2 de CURSOS y TUTORIALES, en la carpeta Herramientas->Editores, que te permiten generar bases de datos en MySQL por medio de una interfase gráfica amigable.

Para que este programa funcione, así como cualquier aplicación que trabaje con MySQL, necesitas estar ejecutando el servidor. Con PHPTriad instalaste un servidor Apache y uno para MySQL. Desde el menú programa ejecuta los dos y ya podrás trabajar sin problema con estos sistemas.

Busca en la carpeta Codigos->PHP del disco 2 de CURSOS y TUTORIALES los archivos de ejemplo de scripts programados en PHP, o lee el archivo **links.doc** para conocer más sitios de PHP que te ofrezcan ejemplos que puedas aplicar parcial o totalmente.

Siempre prueba y ejercita. No pierdas el ritmo. Establece metas claras desde un principio y organízate. Necesitas saber hacia dónde vas y cómo piensas hacerlo. Si lo que quieres, por ejemplo, es aprender a construir páginas web para vender otros productos de tu creación, estudia conjuntamente marketing, técnicas de e-commerce y analiza los sistemas de cobro existentes. Si quieres diseñar páginas para terceros entonces estudia idiomas e intenta conocer un poco de cada lenguaje existente (ASP, Java, Flash). Concentra tus estudios en PHP y MySQL, pero arma una pequeña colección de scripts Java y aprende a animar gráficos con Flash para poder satisfacer lo mejor posible las demandas de quienes te contratan. Nunca sabes lo que pueden llegar a exigirte. También es bueno estudiar el diseño gráfico de diferentes sitios para tener idea de las tendencias y conocer los proyectos más exitosos que puedas imitar.

Sigue los pasos de los que están delante de ti. La experiencia y los conocimientos acumulados de los demás te servirá para evitar frustraciones. Ya llegará el momento de innovar. Por ahora dedícate a aprender, y, lo más importante, a trabajar. Nunca dejes de crear y trabajar sobre tus proyectos. Aprende mientras trabajas y trabaja mientras aprendes.

Una vez adquirida la práctica inicial estarás preparado para lanzar tu primer sitio. Como en todo, lanza un proyecto experimental y adquiere experiencia con esto. Deja los trabajos más importantes para una segunda etapa.

Para publicar tu página web deberás contratar espacio en un servidor. Busca proveedores de hosting económicos o incluso gratuitos. Macrosoft WAB ofrece 30 megabytes de espacio a un precio muy rentable. Infórmate en nuestro sitio: www.macrosoftwab.com.ar o en cualquier otro que conozcas. Otras opciones son servidores gratuitos, pero tienen la desventaja de que colocan publicidad en todas tus páginas y muchos de ellos no activan las opciones PHP o MySQL por lo que sólo podrás publicar páginas HTML. En el archivo **links.doc** se listan algunos de ellos.

Una vez contratado el espacio, comienza a hacer pruebas. Una cosa es tener funcionando la página en nuestra computadora y otra muy distinta es ejecutarla en el nuevo servidor. Lo que debes tener en cuenta es el tamaño de los archivos que utilizas. La importancia del tamaño radica en la velocidad a la que nuestros clientes accederán al sitio y en la cantidad de ancho de banda que consumimos. Los sitios web además de consumir espacio en disco dentro del servidor consumen ancho de banda. Este ancho es limitado, por lo que si excedemos el límite de transferencia nuestros visitantes recibirán un desagradable mensaje de error. Para acelerar el proceso y asegurarse del correcto funcionamiento es necesario disminuir el tamaño de los archivos al máximo sin quitarle valor a nuestro trabajo. Este equilibrio es el que todo diseñador y programador de

páginas web persigue.

Registra tu primer sitio en diferentes buscadores, publica mensajes haciendo alusión al mismo en foros y salas de chat. Necesitas adquirir experiencia con visitas reales. Controla estas visitas por medio de contadores y sistemas de estadísticas. Puedes encontrar enlaces a empresas que proveen de estas herramientas en forma gratuita dentro del archivo **links.doc**. Si la experiencia que has adquirido al llegar aquí es la adecuada, seguramente tú mismo podrás construir un contador y un sistema de información en PHP.

Crea también foros o libros de visitas para que los navegantes te den su impresión y dejen consejos. Esta primera experiencia podría ser tu página oficial, con fotos de tu familia o artículos sobre temas de tu interés. Asuntos irrelevantes pero que puedan resultar de interés para algunas personas que al menos la visitarán para saludarte.

No te estanques en este proyecto ni te preocupes por hacer mejoras en él. Lo importante es la experiencia y nada más que eso. Sigue trabajando en proyectos más importantes y continúa estudiando.

En esta etapa lo importante será afianzar los conocimientos adquiridos sobre PHP y MySQL, además de todo lo que puede enseñarte tratar con visitantes reales.

Es bueno conocer un poco de todo, pero siempre debes especializarte en algo. Lee los cursos avanzados de bases de datos ubicados en el disco 1 de CURSOS y TUTORIALES, carpeta Cursos->Basesdedatos->MySQL para profundizar lo más que puedas en este aspecto.

Mi negocio

Existen tres grupos bien diferenciados de programadores de páginas web: aquellos que lo hacen para su empresa, ya sean dueños o empleados, aquellos que lo hacen para terceros, por encargo, y quienes lo hacen porque desean llevar a cabo negocios en la red. Lo explicado hasta ahora se aplica a todos ellos. El lenguaje PHP y el sistema de bases de datos MySQL son las mejores herramientas para ayudar a todos a construir sitios web de calidad, y cualquiera necesita conocer al menos los detalles básicos para emprender un proyecto exitoso. Pero no todo se termina aquí. Aquellos que desean hacer negocios con la red y no solo cobrar por un buen diseño, deberán abarcar más terreno. Explorar los sistemas de compra y venta, conocer los pormenores sobre el concepto de comunidad, ejecutar aplicaciones y juegos en línea, etc...

Nunca es simple poner en marcha un negocio, y cualquier proyecto de Internet debe ser encarado del mismo modo y con el mismo esfuerzo que su equivalente en el mundo real. No porque sea más económico y accesible vamos a construir un negocio virtual en dos días, sin pruebas previas, y sin aplicar las normas generales de cualquier negocio. Para vender en Internet hay que tratar con el cliente, demostrarle las cualidades de nuestros productos o servicios, atenderlo lo mejor posible y ofrecerle todas las facilidades para que se decida a hacer su compra y efectúe el pago correspondiente. Un negocio virtual exige incluso más esmero que uno convencional.

Pero el esmero no significa necesariamente extensos períodos de prelanzamiento. El sitio debe ser puesto en marcha lo más pronto posible. El esmero hay que ponerlo después, una vez en marcha y con mucho que actualizar, ampliar o mejorar. Se deben construir los elementos esenciales y funcionales con rapidez y eficiencia, para luego preocuparse por extender el alcance y las posibilidades de nuestro negocio virtual en una etapa posterior de plena actividad.

Si esperamos a tener todo listo nunca lo haremos. Los sitios en Internet son algo en constante crecimiento, dejemos que el nuestro crezca naturalmente.

Dar los pasos correctos nunca es garantía de éxito en este nuevo medio para hacer negocios. Explora y

practica técnicas originales, pero nunca expongas la integridad de tu negocio. Dicen los expertos que Internet es un universo estupendo en el que se pueden probar las más descabelladas estrategias de venta y los más innovadores sistemas de marketing y publicidad. Aprovecha al igual que ellos este nuevo medio y deja correr tu creatividad, pero nunca expongas tu sitio al fracaso. Siempre deja la base operativa intacta, al menos hasta que lo experimentado haya sido adoptado e impuesto por tus clientes y pueda ser asimilado sin riesgos en tus páginas.

Busca cada oportunidad de negocio existente en tu nicho de mercado. La mentira más grande que he escuchado en mi vida es que las oportunidades se presentan. ¡Tú tienes que ir tras las oportunidades! ¿Piensas que sentado en tu casa, mirando televisión o programando una página web sin publicarla en Internet, o publicando una página web sin promocionarla adecuadamente vas a tener una oportunidad de triunfar? Las oportunidades hay que buscarlas. Ingresa al mundo de los triunfadores siguiendo el camino correcto. Plantéate objetivos a corto, mediano y largo plazo y trabaja para cumplirlos sistemáticamente. Y no regales nada, a menos que se trate de una estrategia de venta o de acaparamiento de clientes y tráfico.

Recuerda siempre: trabaja mientras estudias y estudia mientras trabajas.

Todos los archivos, cursos y herramientas nombrados en esta guía se encuentran en los dos discos que conforman la recopilación CURSOS y TUTORIALES. Si deseas adquirirlos ingresa a www.macrosoftwab.com.ar y sigue los enlaces.

Esta guía se incluye en CURSOS y TUTORIALES conjuntamente con los archivos Guía para el programador independiente y Guía para la programación de Vídeo Juegos. Ambos archivos están disponibles en forma gratuita en nuestro sitio.

Si necesitas contratar espacio en un servidor para alojar tu página web puedes utilizar nuestro servicio de hosting económico. Tendrás la posibilidad de subir tu página, lanzar tu primer negocio virtual y experimentar con las herramientas más utilizadas del mercado (PHP, MySQL, ASP, Java, etc...) al más bajo costo. Ingresa a www.macrosoftwab.com.ar y elige la opción Hosting.

Si quieres recibir novedades o noticias del mundo de la programación inscribete en www.macrosoftwab.com.ar, es gratis. Recibirás la información que necesitas en tu casilla de correo todos los meses y actualizaciones de estas guía o nuevas ediciones.

Un saludo

Comunidad Macro

Macrosoft WAB

www.macrosoftwab.com.ar