



## Especial DotNetNuke

**PEP LLUÍS, Miembro del Equipo de Mercadeo en INETA-Latam, Presidente de SpainNet y Microsoft MVP – Visual Developer**

Los pasados 19 y 20 de noviembre, Microsoft organizó en El Escorial el primer CodeCamp, un evento enfocado totalmente a desarrolladores y donde un gran número de nosotros pudimos disfrutar durante dos días del mejor código en directo.

Empezamos hablando de DotNetNuke con Vicenç Masanas, para continuar con una estupenda presentación de WPF (Windows Presentation Foundation) por David Salgado. Con Chema pudimos asistir a una espectacular puesta en escena sobre temas de seguridad, viendo como trabajan los "rootkits" (troyanos actuales) y demostrando un alto nivel de maestría con el uso de técnicas hiperexpertas en la captura de llamadas a direcciones memoria, codificando en assembler y, cómo no, en C++. También tuvimos la oportunidad de escuchar a Ricardo Varela hablándonos de los sistemas de posicionamiento global y dispositivos móviles, con su ya clásica escenografía, esta vez ambientada en *El señor de los anillos*. Para finalizar, y si os parece poco, Alejandro Amescua y Unai Zorilla nos hablaron de desarrollo de aplicaciones para Windows Mobile, David Carmona de cómo atreverse con el ShareWare, Ilitia con una excelente exposición sobre cómo montar una empresa, terminando con Miguel Jiménez y Diego Muñoz programando con DirectX.

Está claro que esta interesante agenda daría para llenar más de un artículo. Sin embargo, de todos estos temas he escogido hablarlos de DotNetNuke.

Os preguntareis ¿por qué?, y la respuesta es simple. Cuando Vicenç empezó a esgrimir las palabras "open source" entremezcladas entre "windows" y "framework", creí que me había equivocado de charla. Me pareció más que exótico pues, a menudo, para la mayoría de nosotros Open Source viene asociado a otro

tipo de proyectos. Así pues, me complace presentaros un estupendo proyecto Open Source para la plataforma .NET de la mano de uno de los principales conocedores del mismo en España y miembro del DotNetNuke INNER TEAM, Vicenç Masanas.

**Pep.** Sabemos de DotNetNuke que es un sistema de administración para crear y manejar sitios web. También sabemos que podemos utilizarlo como un punto integrador entre comunidades distribuidas, donde la prioridad es compartir un contenido digital además de mantener una vital colaboración en línea. Danos exactamente tu definición de DotNetNuke.

**Vicenç.** DotNetNuke (DNN) puede verse desde varios puntos de vista dependiendo de las necesidades específicas de cada usuario. Como muy bien has comentado, puede utilizarse como un sistema de creación y administración de sitios web con funcionalidades muy avanzadas, sin por eso renunciar a la facilidad de uso para usuarios no técnicos. Cualquier usuario sin previos conocimientos de edición de páginas web puede gestionar una web con unos mínimos conceptos de funcionamiento del sistema.

Un segundo aspecto de interés en DotNetNuke es la facilidad de creación de sitios de colaboración para grupos de usuarios que necesiten trabajar en un entorno virtual que pueda proveerles las herramientas básicas que se requieren en este entorno: foros de mensajes, compartición de documentos, gestión y edición de contenidos, agenda, etc. Todo esto, unido a un potente control de permisos basado en roles y usuarios, ofrece a los administradores de la web una enorme granularidad de acceso a las funciones del sistema. Aquí vemos claramente su potencial en entornos de intranet.

Finalmente y quizás, desde mi punto de vista, lo más importante, DNN puede verse como un entorno de creación de soluciones a medida basado en la web (Web Application Framework) para cualquier necesidad de la empresa. DNN ofrece de base un sistema (API) con unas funcionalidades muy interesantes sobre las cuales satisfacer toda necesidad de negocio que se presente. De hecho



el núcleo (core) de DotNetNuke no deja de ser una herramienta sobre la que se construye todo el resto: diseño de la web, contenidos, módulos personalizados, etc. La clave es: extensibilidad.

**Pep.** A pesar de que muchos conocemos la filosofía Open Source, se nos hace extraño entender las condiciones exactas de uso para un proyecto como DotNetNuke dentro de un marco totalmente comercial. ¿Podrías explicarnos qué implica para cualquiera de nosotros el uso de este tipo de programario?

**Vicenç.** La licencia de DotNetNuke se basa en el estándar BSD (<http://www.opensource.org/licenses/bd-license.php>) y es una de las licencias menos restrictivas que existen. La concreción es muy fácil: la licencia nos da derecho a utilizar DNN en cualquier entorno (comercial o no) y con total libertad de redistribución (preservando el copyright existente). La única restricción es que no puede utilizarse el nombre DotNetNuke en posibles derivaciones del código fuente. Esto es así básicamente para permitir a todos los usuarios reconocer claramente la fuente original de DotNetNuke e identificarlo con la calidad que intentamos mantener en todas nuestras versiones.

Esta licencia permite construir aplicaciones verticales sobre DotNetNuke para su distribución comercial sin ninguna restricción. Otro modelo de negocio se encuentra en la comercialización de módulos genéricos: de comercio electrónico, workflow, gestión documental...

**Pep.** Si es Open Source, argumentanos por qué nos obliga a registrarnos para poder descargar los instalables.

**Vicenç.** Este es el único peaje que tenemos que pagar para utilizar DotNetNuke, y ¡no es mucho si lo comparamos con todo lo que nos ofrece! Este registro obligatorio nos permite (al Core Team) ver qué nivel de aceptación tiene nuestra herramienta. También es una opción añadida para poder comunicarnos con todos los usuarios registrados en caso de detectarse algún problema grave de seguridad (que yo recuerde solamente se ha hecho uso de este

canal en una única ocasión en las primeras versiones del sistema). Y, últimamente, también ha servido como base para la distribución de la Newsletter oficial del proyecto, sumando otro canal de comunicación para usuarios interesados en DNN.

**Pep.** Es lógico que, al ser muy genérico, DotNetNuke cubra muchas necesidades y se adapte a multitud de escenarios. Sin embargo, por la realimentación que recibes, nos gustaría conocer a qué tipos de audiencia va dirigido, qué necesidades cubre y los principales escenarios donde se desenvuelve DotNetNuke.

**Vicenç.** La gran aportación de DotNetNuke y su principal activo es precisamente su adaptabilidad a multitud de entornos. Los desarrolladores de aplicaciones web lo ven como una herramienta sobre la que basar su trabajo (tanto para soluciones para clientes finales como para la creación de módulos estándar); los usuarios finales como una opción de gestión de webs de muy fácil utilización; grupos departamentales como una herramienta de colaboración en entorno intranet, y cada vez más empresas lo ven como una apuesta de futuro sobre la que basar su infraestructura web (tanto pública como privada). Últimamente también los hosters están muy interesados en DotNetNuke como alternativa a otros entornos basados en LAMP (Linux-Apache-MySQL-PHP). Esperamos que muy pronto haya una buena oferta de servicios de hosting para desplegar DotNetNuke.

**Pep.** Estás convencido de que DotNetNuke me evita reinventar la rueda y que realmente puedo crear aplicaciones de alto nivel y profesionalidad. A simple vista, un producto gratuito debe estar enfocado a estudiantes o aficionados.

**Vicenç.** Cualquier persona que haya trabajado mínimamente con cualquier producto Open Source sabe que la calidad de estos productos es comparable a sus competidores de código cerrado, si no mayor. Todos los que trabajamos en DotNetNuke lo hacemos para tener la

herramienta más potente posible al alcance de nuestra mano. Nosotros somos los primeros interesados en que DNN sea una opción viable para entornos de alto nivel y con requerimientos estrictamente empresariales.

**Pep.** ¿Qué necesitamos para instalarnos DotNetNuke?

**Vicenç.** DotNetNuke funciona sobre la plataforma ASP.NET de Microsoft. Lo primero que vamos a necesitar es un servidor IIS y el framework ASP.NET 1.1 ([msdn.microsoft.com/asp.net](http://msdn.microsoft.com/asp.net)). Todo el contenido se guarda en base de datos, por lo que vamos a necesitar también un servidor de base de datos SQLServer 2000 o en su defecto la versión gratuita MSDE ([www.microsoft.com/sql/msde](http://www.microsoft.com/sql/msde)). Existe también una versión de DotNetNuke (4.0) para la ASP.NET 2.0. En este caso, necesitaremos la última versión de ASP.NET y podemos utilizar el servidor web (Cassini) integrado en Visual Web Developer ([www.asp.net](http://www.asp.net)). También podemos utilizar la nueva versión del servidor de base de datos SQLServer 2005. Recomiendo encarecidamente probar las versiones Express de estos productos: son gratuitas y perfectas para empezar a jugar con DNN.

**Pep.** Danos, a grandes trazos, los principales pasos para su instalación.

**Vicenç.** La instalación varía un poco (en los detalles) dependiendo de si vamos a utilizar la versión para ASP.NET 1.1 (DotNetNuke 3.x) o la versión para ASP.NET 2.0 (DotNetNuke 4.x). Son unos pocos pasos necesarios: crear una base de datos y un usuario, crear un directorio virtual en IIS que apunte al directorio donde tengamos DNN instalado y modificar los parámetros de conexión de la base de datos de DNN para que conecte a la base de datos que hemos creado. Una vez hecho esto nos conectamos con cualquier navegador a la web y DNN hará el resto: inicialización de la base de datos, creación de un sitio por defecto y configuración de los usuarios necesarios para empezar a trabajar. Además del paquete de instalación de DotNetNuke hay también una gran colección de documentos de ayuda de las diferentes partes del sistema. La



guía de instalación detalla claramente estos pasos y puede solventar cualquier problema que pueda surgir. Estamos muy orgullosos de nuestra documentación y dedicamos mucho esfuerzo a crear ayudas y guías para la comprensión del proyecto. Os recomiendo una lectura previa para asentar el funcionamiento básico de DotNetNuke antes de instalarlo.

**Pep.** ¿Podemos personalizar el entorno? Háblanos de módulos, skins, data providers, los language pack y de los templates.

**Vicenç.** DNN es personalizable a muchos niveles. Los módulos son las entidades básicas de extensibilidad de DNN. Hay módulos de fóruns, galerías de imágenes, blogs, gestión de documentos, comercio electrónico, etc. En el paquete base de DNN vienen unos cuantos módulos pero hay muchos más disponibles en numerosas webs dedicadas a DotNetNuke (para una lista de módulos gratis ver en [www.cathal.co.uk](http://www.cathal.co.uk); para ver los módulos comerciales ir a [www.snowcovered.com](http://www.snowcovered.com)). El diseño de la web se controla mediante lo que llamamos skins. Un skin es un diseño específico que cumple unos requisitos básicos para utilizarse en el sistema, la única limitación es la imaginación.

DotNetNuke está basado en el concepto de proveedores ("providers"). Un proveedor es una interficie de acceso común que puede ser intercambiada por otra que cumpla la misma interficie. Esto facilita el cambio de partes del sistema de forma totalmente transparente a su funcionamiento. Por ejemplo, hay providers de base de datos para Oracle para quien lo quiera utilizar en este entorno. Si no nos gusta alguna parte del sistema (por ejemplo el editor de textos) siempre podemos crear un provider para nuestro editor preferido e integrarlo en el sistema sin afectar para nada otras partes de DNN.

Los "templates" son plantillas de sitios web que pueden utilizarse para modelar una estructura básica que deban tener las nuevas web que creemos en DotNetNuke. Esto es muy importante puesto que DotNetNuke soporta lo que se llama virtualización de sitios web: en

una única instancia de DNN podemos tener múltiples sitios web independientes de forma totalmente transparente e integrada. Los templates son las plantillas que utilizamos para crear nuevos portales.

Los paquetes de lenguaje son las extensiones que permiten traducir la interficie de DNN a cualquier idioma. La traducción y creación de estos paquetes también puede realizarse íntegramente sin otra ayuda que DotNetNuke. En la web de DotNetNuke están ya disponibles traducciones a más de 30 idiomas.

**Pep.** Además de adaptarlo y personalizarlo en nuestros proyectos, ¿podemos añadir funcionalidades que no existan en la base inicial?

**Vicenç.** Esto es perfectamente posible y no solo eso, sino que en sí mismo se ha convertido en la fuente de ampliación más importante de DotNetNuke. Cada día surgen nuevos módulos para cualquier necesidad imaginable. Si estas funciones son muy buenas y suficientemente generales y de amplio interés siempre hay la posibilidad de ofrecer el código al Core Team para su inclusión en próximas versiones. Este ha sido también un activo muy importante en DotNetNuke. Dado que mucha gente lo utiliza para fines muy distintos se han ido añadiendo funcionalidades que actualmente cubren un gran abanico de situaciones.

**Pep.** Finalmente, en la página DotNetNuke hemos podido verte como miembro del INNER Team. ¿Cuál es tu implicación en el proyecto? ¿Qué significa ser miembro del INNER y qué pasos debería hacer para poder colaborar en este proyecto?

**Vicenç.** DotNetNuke es un proyecto Open Source de colaboración voluntaria. El equipo se estructura en cuatro capas: Shaun Walker es el líder del proyecto y quien tiene la última decisión en cualquier discusión (esta figura se llama habitualmente "Benevolent Dictator"). Hay un equipo de dirección ("Board of Directors") que se encarga de la visión y gestión a largo plazo del proyecto. Este grupo está formado por algunos de los miembros más antiguos del equipo y

que han mostrado más empeño y claridad en sus aportaciones. Luego tenemos el "Inner Core Team", que está formado por los miembros más activos y más implicados en la gestión diaria del proyecto. Estos miembros son los que reciben los encargos más importantes y en muchos casos tienen porciones concretas del sistema a su estricto cargo. Y finalmente está el "Outer Core Team", compuesto por los demás miembros de grupo que pueden tener a su cargo funciones de soporte y ayuda para determinadas partes del proyecto.

Actualmente yo tengo a mi cargo el control de la globalización y localización del proyecto. En este área me encargo de verificar que los procesos internos del sistema son compatibles con la estructura multilingüe que queremos ofrecer, así como de explorar posibles mejoras y opciones de futuro en áreas relacionadas.

Si embargo, el hecho de estar a cargo de una determinada área no implica que no trabajemos conjuntamente en muchos otros aspectos. Esta es efectivamente la gran aportación que un equipo bien unido tiene que ofrecer en este entorno altamente distribuido: todos los miembros del equipo tenemos que tener una gran comprensión del sistema en su globalidad para poder tomar las decisiones correctas en todo momento.

Las aportaciones desde fuera del "Core Team" pueden hacerse en los fóruns, aportando ideas, ofreciendo ayuda a otros miembros y, cómo no, solucionando bugs, que también los tenemos :).

**Pep.** Solo me resta finalizar esta entrevista dándote las gracias, Vicenç, por compartir tu experiencia con nosotros. A todos los que después de leer este artículo tengáis la inquietud de saber más, os animo a que os descarguéis el DNN. ¡De verdad que es espectacular! Con un par de clicks tendréis todo un portal listo para usar. Y estad atentos a la próxima edición de *Solo Programadores*, que el mes que viene tratará sobre DotNetDuke con mayor profundidad. Mientras tanto, ¡qué lo disfrutéis! ☺