

Expertis Engine 5.0. Documentación Técnica.

Presentación

Expertis Engine (EE) es un marco de trabajo para la construcción de aplicaciones empresariales, que aporta productividad durante el desarrollo de soluciones altamente escalables.

EE es el resultado de unir las directrices derivadas de las últimas propuestas de arquitecturas de aplicación modernas y de alta escalabilidad, con la experiencia de Solmicro en la construcción de aplicaciones empresariales.

EE consigue estos altos niveles de productividad permitiendo que el equipo de desarrollo se centre en la implementación de la funcionalidad que aporta valor a la empresa y automatizando todos los aspectos del desarrollo que no lo hacen.

Arquitectura

EE implementa los principios de una arquitectura de aplicación distribuida. Este tipo de arquitecturas nacen con el objetivo de solucionar los problemas de escalabilidad que se presentaban en las arquitecturas existentes previamente, como la denominada "Cliente/Servidor".

Definimos la escalabilidad como la capacidad de la aplicación de crecer en número de usuarios, aportando más máquinas cuando sea preciso, si tener que rediseñar ninguno de los elementos que componen la aplicación.

Aunque se define como objetivo principal el de la consecución de una alta escalabilidad, otros objetivos importantes de este tipo de arquitecturas son:

- **Seguridad:** Que tenga en consideración la seguridad desde el principio, teniendo en cuenta los mecanismos adecuados de autenticación, la lógica de autorización y la comunicación segura.
- **Rendimiento:** Que proporcione un alto rendimiento y esté optimizada para operaciones frecuentes entre patrones de implementación.
- **Confiabilidad:** Que se ejecute de una manera predecible.
- **Disponibilidad:** Que esté disponible y sea resistente, capaz de implementarse en centros de datos de alta disponibilidad y redundantes.
- **Escalabilidad:** Que permita la escalabilidad para cumplir las expectativas de la demanda y admita un gran número de actividades y usuarios con el mínimo uso de recursos.
- **Administrable:** Que se pueda administrar, permitiendo a los operadores implementar, supervisar y resolver los problemas de la aplicación en función del escenario.

- **Mantenible:** Que se pueda mantener. Cada parte de funcionalidad debería tener una ubicación y diseño predecibles teniendo en cuenta distintos tamaños de aplicaciones, equipos con conjuntos de habilidades variadas y requisitos técnicos y cambios empresariales.
- **Flexible.** Que funcione en los distintos escenarios de aplicaciones y patrones de implementación

La estructura de aplicación propuesta por este tipo de arquitecturas, define una división en capas de la aplicación, que tradicionalmente son tres, presentación, negocio y datos, aunque puedan añadirse más capas según necesidades de aplicación o infraestructura, lo que hace que generalmente se llamen como aplicaciones en N capas.

Como decíamos, de forma tradicional se habla de tres capas:

- **Capa de presentación:** Contiene todos los elementos necesario para que el usuario interactúe con la aplicación. Depende del dispositivo empleado para dicha interacción. Pueden existir varias implementaciones de capa de presentación para la misma aplicación, habilitando así la interacción con la aplicación mediante diferentes dispositivos o tecnologías. Debe permanecer vacía de lógica de negocio.
- **Capa de negocio:** Contiene la lógica de la aplicación y por lo tanto puede considerarse la parte más importante de la aplicación. La lógica de la aplicación incluye aspectos tan importantes dentro la funcionalidad empresarial como la correcta ejecución de los procesos de negocio, la seguridad, la gestión de transacciones, ...
- **Capa de datos:** Contiene la lógica de interacción con el sistema de almacenamiento de datos, lo que permite que la capa de negocio se abstraiga del sistema de almacenamiento usado, consiguiendo así esta ser mas portable. La capa de datos se encarga de realizar trabajos de bajo nivel como la creación de conexiones a base de datos, la construcción de sentencias de lectura y escritura para el sistema de almacenamiento, ...

La división en capas tiene las siguientes ventajas:

- Cada capa sólo se comunica con las capas adyacentes, con lo que se simplifican y limitan los interfaces y tecnologías a usar por cada capa.
- Permite la distribución de capas en diferentes máquinas, lo que da acceso a escenarios con conjuntos de máquinas con diferentes roles y diferentes necesidades de configuración, administración, disponibilidad y seguridad.
- Aumenta la capacidad de la aplicación para ser usada desde diferentes escenarios, mediante diferentes capas de presentación o capas de integración con terceras aplicaciones.
- Permite la especialización de desarrolladores, de forma que exista un conjunto de desarrolladores con mas habilidades en el diseño para la capa de presentación y otros con más habilidades para la implementación de procesos de negocio seguros y con alto rendimiento.

- Aumenta la capacidad de la aplicación para ser mantenida, acotando los elementos a reemplazar ante modificaciones y llegando incluso a poderse reemplazar elementos sin que se perciba parada en los servicios que presta la aplicación.
- Permite el desarrollo de cada capa con los lenguajes y las tecnologías más adecuadas para conseguir los objetivos de cada una.

La división en capas lleva también asociada la necesidad de implementar los mecanismos de comunicación adecuados entre ellas. Esto impone la restricción de que las tecnologías de implementación de las capas deben ser interoperables.

Tecnología

EE Esta desarrollado completamente con tecnología .Net de Microsoft. Se ha utilizado para su implementación todo el conjunto de herramientas y tecnologías asociadas al .Net Framework 3.5:

- Visual Studio y C# para el desarrollo, compilación y depuración del código.
- XAML WPF para la construcción de parte del interfaz gráfico de EE.
- Sql Server como sistema principal de almacenamiento de datos, aunque también se ha construido una capa de acceso a datos específica para trabajar con Oracle.
- Crystal Reports XI R2 como motor de construcción de informes.
- Componentes gráficos de Janus System e Infragistics para desarrollar un interface gráfico atractivo y altamente usable.
- Remoting, una tecnología perteneciente al .Net Framework, para habilitar la comunicación de la capa de presentación con la capa de negocio. Se sigue prefiriendo Remoting frente a WCF, una tecnología de comunicación desarrollada más recientemente, por la sencillez de su uso y manutención, a la vez que por sus capacidades de configuración y extensión.

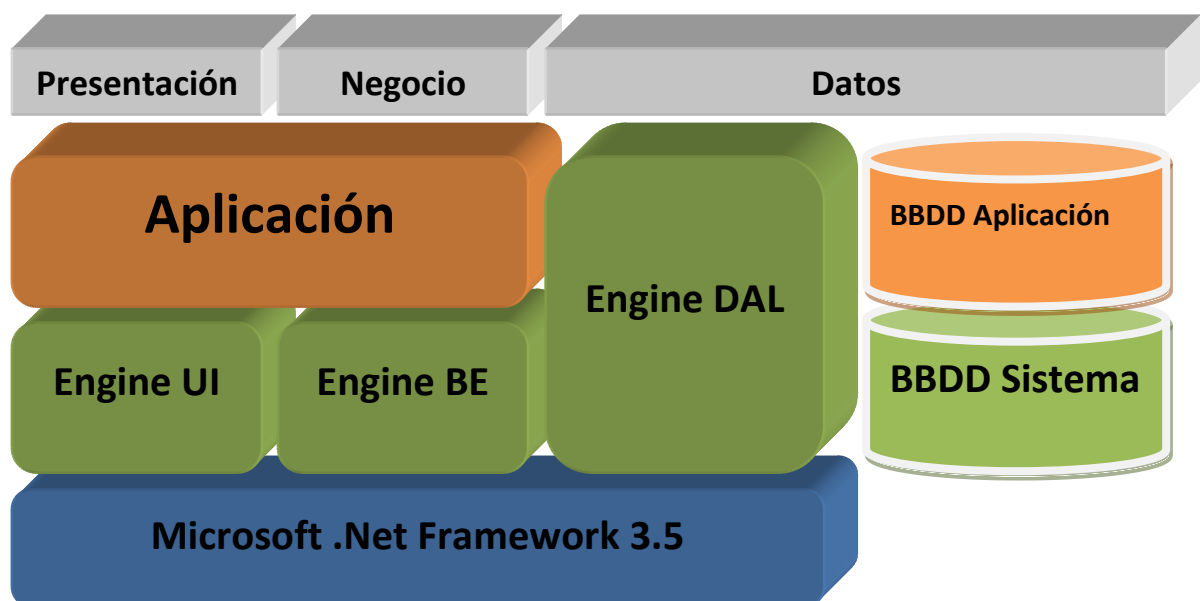


Fig. 1

Características

Multi Base de datos

Se ha incorporado a EE 5.0 la capacidad de trabajar con diferentes sistema gestores de bases de datos (SGBD).

Para ello se ha dividido en dos la lógica de acceso datos, por un lado una interface independiente del SGBD, que implementa la fachada visible desde la capa de negocio, y por otro lado, un proveedor de datos específico de cada SGDB.

Solmicro, sólo ha desarrollado los proveedores de datos específicos para Sql Server y Oracle, pero esta estructura de la capa de acceso a datos hace factible la construcción de otros proveedores de datos para otros SGBD, como por ejemplo DB2, y su integración en EE de forma sencilla.

Las dos partes mencionadas en las que se ha dividido la capa de acceso a datos, se han implementado en los siguientes ensamblados:

- **Expertis.Engine.DAL.dll** contiene la fachada independiente del SGBD para la capa de negocio.
- **Expertis.Engine.SqlDAL.dll** contiene la implementación del proveedor de datos específico para Sql Server.
- **Expertis.Engine.OraDAL.dll** contiene la implementación del proveedor de datos específico para Oracle. Cabe subrayar que este proveedor de acceso a datos permite el uso de tipos de datos GUID y booleanos no admitidos específicamente por Oracle, pero ampliamente utilizados por EE.

Lan/Wan

Gracias a la tecnología Remoting, de .Net, EE puede trabajar en diferentes configuraciones en función de la ubicación de los servicios necesarios para su correcto funcionamiento.

Se pueden diferenciar tres escenarios diferentes:

- **Monolítico.** EE se ejecuta bajo un único proceso en cada máquina. Es un modo de ejecución poco escalable y recomendado únicamente para escenarios de demostración o formación, o bien en instalaciones de un único usuario sin infraestructura de servidores adicional.

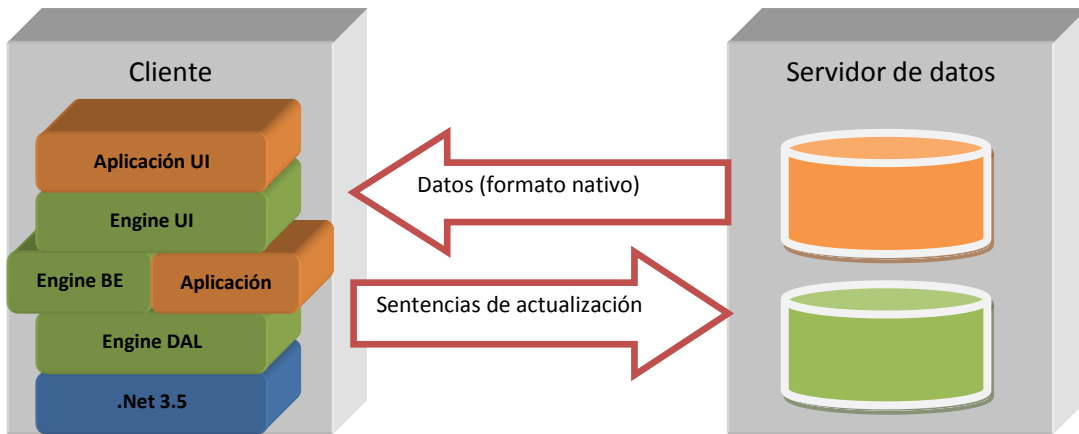


Fig. 2

- **LAN.** EE ejecuta la capa de presentación en cada equipo cliente, mientras que las capa de negocio y acceso a datos se ejecutan en servidores destinados a tal efecto bajo un servicio de Windows llamado **ExpertisHost**. Las comunicaciones en esta configuración usan infraestructura privada de alto rendimiento.

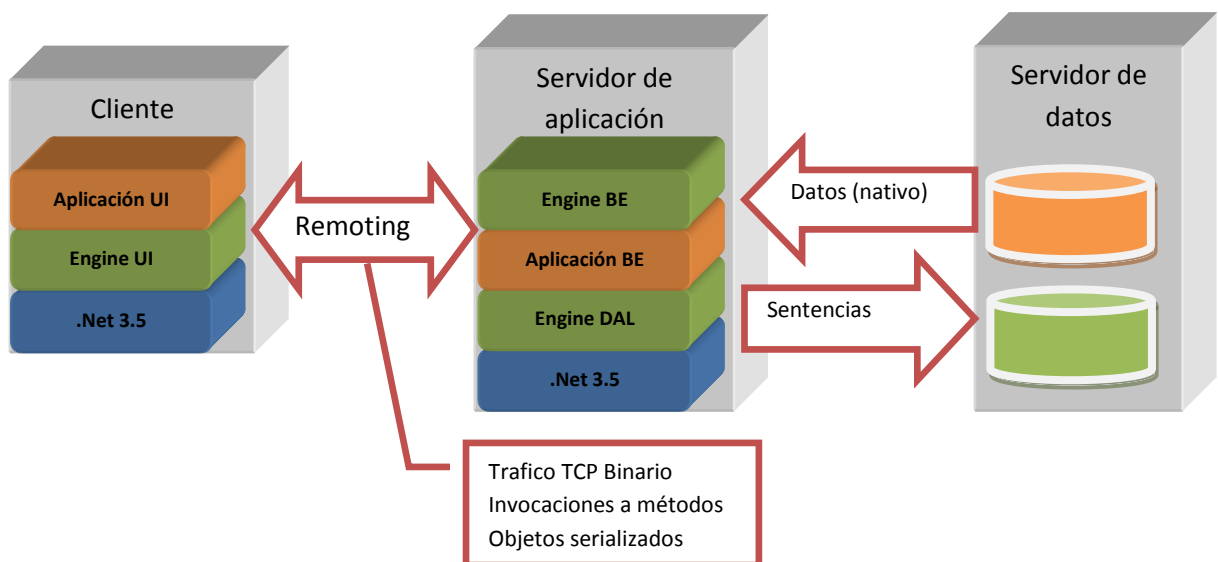


Fig. 3

- **WAN.** La ubicación y ejecución de cada capa de la aplicación es la misma que en el caso anterior, pero se usa infraestructura pública para las comunicaciones y se puede usar **Internet Information Services (IIS)** como hospedaje para los servicios de EE. En esta configuración es útil usar la compresión de datos que permite EE.

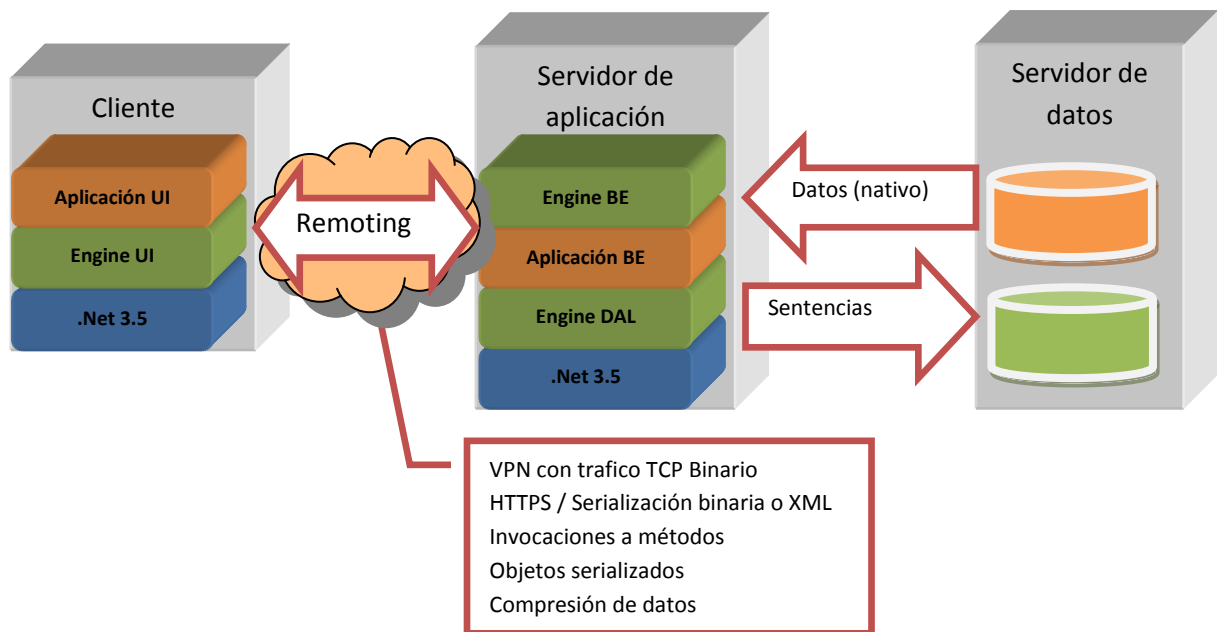


Fig. 4

Se puede configurar fácilmente EE para trabajar en cualquiera de estos modos sin más que distribuyendo adecuadamente los ensamblados que componen las diferentes capas de la aplicación y modificando el fichero de configuración de cliente.

Extensibilidad de la lógica de negocio

EE 5.0 implementa un modelo de extensibilidad de la capa de negocio, basado en la segmentación de los procesos de negocio en pequeñas fragmentos gestionados llamados "tareas".

Estas tareas se pueden agrupar posteriormente en estructuras más grandes, formando una lista ordenada de tareas a realizar que se denominan "procesos", o ser invocadas de forma individual en el código de negocio.

Que las tareas sean elementos gestionados, permite realizar ciertas configuraciones sobre ellas que finalmente proporcionan la capacidad de modificar y extender la funcionalidad contenida en la capa de negocio de la aplicación.

Las configuraciones realizables sobre una tarea son:

- **Deshabilitar.** Se puede deshabilitar una tarea de forma que, aun siendo invocada por el código contenido en la capa de negocio de la aplicación, no se ejecute el código contenido en la tarea. De esta forma se puede evitar la ejecución de tareas que pueden carecer de importancia en ciertos escenarios.
- **Reemplazar.** Se puede redireccionar una tarea hacia código desarrollado por una tercera empresa, en vez de que se ejecute el código desarrollado por Solmicro. De esta forma se puede modificar el comportamiento de las tareas definidas por Solmicro para adecuar mas la solución final a las necesidades del cliente.
- **Configurar tareas anteriores y/o posteriores.** Una de la forma de extensión de la funcionalidad más potentes y sencillas, es la de configurar las tareas definidas por

Solmicro de forma tal que se ejecute una tarea de un tercero antes y/o después de la ejecución de aquella. De esta forma se pueden alterar fácilmente los datos de entrada o salida de una tarea dada, modificando así también el comportamiento de la aplicación.

Todas estas configuraciones se tendrán en cuenta cuando la tarea sea invocada individualmente o dentro de un proceso.

En el caso de los procesos, que son elementos gestionados también, las configuraciones posibles son las siguientes:

- **Reemplazar.** Se puede redireccionar un proceso completo, de forma que la lista de tareas a ejecutar en ese proceso esté definida en el código de un proceso implementado por un distribuidor.
- **Código anterior y posterior.** Se pueden sobrescribir los procesos implementados por Solmicro de forma hereden de estos pero incorporen código a ejecutar al comienzo o al final de la ejecución del aquellos.
- **Agregar tareas.** Se pueden agregar nuevas tareas a la lista de tareas a ejecutar dentro de un proceso sin tener que modificar el código del proceso. Esto posibilita la inyección de código de distribuidor en el código de negocio de la aplicación sin tener que modificar el código proporcionado por Solmicro.
- **Deshabilitar.** Se puede deshabilitar un proceso de forma que no se ejecute su lista de tareas cuando sea invocado por cualquier tipo de código.
- **Deshabilitar tareas de proceso.** Se pueden deshabilitar tareas concretas dentro de la lista de tareas de un proceso, de forma que no se lleguen a ejecutar las tareas dentro del proceso en cuestión, pero continúen ejecutándose dentro de otros procesos o cuando se invocan de forma individual.

Los beneficios derivados de este modelo de extensibilidad afectan los procesos de modificación, actualización y mantenimiento de la aplicación:

- Se aporta un modelo de adecuación de la solución a las necesidades del cliente final en el que no se requiere la modificación del código estándar de la aplicación.
- Se facilita el despliegue correcciones o incrementos funcionales provenientes del producto estándar, ya que se eliminan las colisiones con el código de negocio modificado por terceros.
- Se facilita el versionado del producto, ya que las modificaciones a la funcionalidad para una instalación en concreto permanecen separadas del código estándar del producto.
- Se mejora el rendimiento de la aplicación, ya que el modelo de extensibilidad lleva asociado un mecanismo de cacheo y compartición de datos entre los procesos y tareas ejecutados en el contexto de una invocación a un proceso de negocio.

Gestor de transacciones

EE incorpora un mecanismo de gestión de transacciones propio. La creación de un sistema tal obedece a la necesidad de proporcionar un modelo de gestión de transacciones que simplifique el uso de este tipo de recursos por parte de los desarrolladores de lógica de

negocio, así como de garantizar la correcta finalización de las transacciones ante cualquier situación que se produzca en esta capa de la aplicación.

Este mecanismo de gestión de transacciones asegura que cualquier transacción iniciada por la lógica de negocio, se mantenga encerrada únicamente en el ámbito de la lógica de negocio, y que no se puedan iniciar o gestionar transacciones desde la capa de presentación.

De esta forma se consigue una vez más, aportar escalabilidad a la aplicación, ya que la contención de las transacciones dentro exclusivamente del ámbito de la lógica de negocio, hace que estas se prolonguen menos en el tiempo, y por lo tanto también sean más cortos los bloqueos que conllevan, con lo cual aumenta la capacidad para dar servicio a usuarios concurrentes.

En términos generales, lo que aporta este mecanismo es la capacidad para que un desarrollador de lógica de negocio, determine en su código donde arranca una transacción y, que, si quiere, omita indicar cuándo termina. Una vez definido el comienzo de una nueva transacción, cualquier actualización de la base de datos estará amparada por la transacción en curso, y si no existe un punto explícito en el que finalizar la transacción, EE le dará el final adecuado cuando el hilo que ha ejecutado la invocación a la lógica de negocio, salga del ámbito de negocio y retorne a la capa de presentación. En este punto se asume que si la salida del ámbito de negocio lleva asociada una excepción, un error, se desharán todos los cambios contenidos en la transacción, o se confirmarán si la salida se hace sin errores.

Este mecanismo de gestión de transacciones está basado en características de .Net Framework relacionadas con la tecnología de **Remoting**, pero que siguen funcionando de la misma forma independientemente del modelo de ejecución de la aplicación. O sea, el mecanismo de gestión de transacciones funciona exactamente igual independientemente de que la aplicación se ejecute en modo monolítico, LAN o WAN.

Despliegue automatizado

EE dispone de mecanismos para automatizar el despliegue de modificaciones y nuevas funcionalidades en cualquier parte de la aplicación.

Básicamente este mecanismo compara la versión de los ficheros locales de la máquina cliente con sus homónimos en una carpeta específica del servidor, provocando la descarga de aquellos cuya versión en el servidor sea más reciente que la del equipo cliente.

Este mecanismo de despliegue atiende a tres tipos diferentes de ensamblados actualizables:

- Ensamblados de presentación. Este tipo de ensamblados son los que contienen todos los elementos mediante los cuales, el usuario realizará su interacción con la aplicación, generalmente formularios que son inventariables en el sistema de menús de EE, y cuyos accesos directos se muestran en la pantalla principal de la aplicación. Cuando el usuario intenta acceder a la funcionalidad contenida en este tipo de ensamblados a través de esos accesos directos, se hace la comprobación de versión del ensamblado y se produce la descarga. En este caso se produce una actualización bajo demanda implícita en la acción del usuario.

- Ensamblados de negocio o referenciados. Este tipo de ensamblados, cuyo contenido está destinado a ejecutarse en el servidor, también han de distribuirse en las máquina cliente. Es un requisito de [Remoting](#) para habilitar la invocación remota. Este tipo de ensamblados se actualizan cuando se inicia la aplicación construida con EE.
- Ensamblados de EE. Los ensamblados de los que se compone EE también son susceptibles de actualizarse de forma automática. En este caso el proceso de actualización consta de dos partes y se ejecuta también durante las primeras fases del arranque de la aplicación.

Balanceo de carga

EE es totalmente compatible con la creación de granjas de servidores de aplicación con balanceo de carga. Este tipo de configuraciones se pueden implementar ante instalaciones con requerimientos de alta disponibilidad y de una alta carga trabajo.

Ante escenarios que requieran únicamente alta disponibilidad, EE puede instalarse en un cluster de conmutación por error, sin necesidad de usar balanceo de carga

Hay que diferenciar dos tipos configuraciones de balanceo de carga, aquellas que tienen afinidad, entendiéndose por afinidad la característica de configuración que hace que un equipo cliente sea atendido siempre por el mismo nodo de la granja de servidores, de aquellas que no la tienen.

En una granja de servidores configurada con afinidad, no hay que hacer ninguna configuración adicional en EE para que la aplicación funcione con balanceo de carga. Toda la responsabilidad del balanceo recae entonces en el componente hardware o software de balanceo.

En el caso de que la granja esté configurada sin afinidad, lo que hace que las peticiones de un cliente puedan ser atendidas cada vez por un nodo diferente, se puede configurar EE para que sincronice la información de sesión entre todos los nodos de la granja, asegurando así la que el nodo tendrá la información necesaria para procesar la petición. La configuración de este escenario es tan sencilla como inventariar las direcciones IP internas de todos los nodos de la granja mediante la consola de Administración de EE.

Reemplazo en caliente de librerías de negocio

Otra característica implementada en EE y que tiene como objetivo proporcionar escenarios en los que las aplicaciones construidas con EE se ejecuten en modos de alta disponibilidad, es la de la actualización de ensamblados de negocio en caliente.

Esta característica permite que se envíen, a todos los servidores que están ejecutando la capa de negocio de la aplicación, nuevas versiones de los ensamblados de negocio, que comienzan a ejecutarse en cuanto son recibidas, sin que sea necesario reiniciar los servicios bajo los cuales se ejecuta el código de negocio.

De esta forma se pueden hacer labores de mantenimiento e incremento de la funcionalidad de la aplicación sin que los usuarios de esta perciban parada alguna en la prestación de los servicios que proporciona.

LOPD

EE proporciona las características necesarias para crear aplicaciones que cumplen con los requisitos definidos en la LOPD:

- Se puede definir el número máximo de intentos de conexión a la aplicación fallidos. Al sobrepasar este umbral, la cuenta de usuario queda automáticamente bloqueada, pudiendo ser únicamente desbloqueada por un usuario del grupo de administradores.
- Se puede habilitar el cambio de contraseñas por parte de los usuarios, de forma que este pueda cambiarla en el momento que quiera.
- Se puede establecer un plazo máximo de validez de las contraseñas, de forma que el usuario se vea obligado a cambiarla una vez vencido dicho plazo.
- Se puede establecer el número de contraseñas de usuario recordadas por el sistema, de forma que el usuario se vea obligado a crear una contraseña no contenida en la lista de contraseñas recordadas.
- Se puede establecer, mediante una expresión regular, una política de complejidad de la contraseña.
- EE lleva automáticamente un registro de los intentos de acceso a la aplicación, consultable mediante la consola de administración de EE.
- Se puede configurar mediante la consola de administración de EE, qué entidades de la aplicación contienen información amparada por la LOPD, de forma que se registren los intentos de acceso de los usuarios a la información de dichas entidades. Esta información es también consultable mediante la consola de administración.
- En escenarios que requieran mas nivel de auditoría y usando características de Sql Server 2008 Enterprise, es posible registrar las modificaciones que los usuarios realizan sobre los registros de las entidades amparadas por la LOPD y consultar esta información mediante una utilidad integrable en EE.

Crecimiento funcional no traumático

EE proporciona un modelo de crecimiento funcional de aplicaciones empresariales no traumático. En el terreno práctico esto significa que el ejecutable principal de la aplicación, **Expertis.Engine.UI.Shell.exe**, se ha concebido como un contenedor en el que se muestran contenidos externos. Estos contenidos son básicamente, formularios de aplicación construidos con Visual Studio, informes de Crystal Reports y páginas Web. Una vez desarrollados mediante las correspondientes herramientas, se pueden inventariar en EE mediante la consola de administración y agrupar y organizar para formar una estructura que facilite al usuario el acceso a los mismos.

Esta facilidad para moldear los perfiles de usuario, unido a la capacidad de autodespliegue de EE, hace que las nuevas funcionalidades de la aplicación, se presenten de una forma sencilla al usuario y este pueda utilizarlas sin ninguna molestia por su parte.

Cabe mencionar que EE incorpora un complejo sistema de construcción de menús de usuario, con la capacidad de crear menús básicos que van siendo completados con funcionalidades añadidas en nuevos menús que derivan de aquellos a través de un mecanismo de herencia. Esto permite administrar fácilmente las aplicaciones construidas para empresas con un

elevado número de usuarios, pudiendo elegir a qué nivel de la jerarquía de menús se insertan nuevas funcionalidades y, por tanto, que grupos de usuarios podrán acceder a ellas.

EE también permite incorporar fácilmente nuevos informes y nuevas acciones relacionables con los formularios o programas existentes

Seguridad

EE permite la definición de grupos de seguridad y los recursos de la aplicación a los que cada grupo tiene acceso.

Esta definición de seguridad se superpone sobre las configuraciones de menús, de forma que el alcance funcional accesible finalmente por el usuario, puede llegar a ser un subconjunto de la funcionalidad propuesta en el menú que se le ha asignado.

Los elementos sobre los que se pueden establecer restricciones de seguridad a grupos de usuarios son:

- Los programas o formularios de la aplicación.
- Los informes invocables desde los programas.
- Las acciones globales invocables desde los programas
- La visibilidad de la entidades, pudiendo definir si se pueden ver registros de una determinada entidad, si se pueden modificar, crear o eliminar.
- La visibilidad de los atributos de una entidad, pudiendo definir, si se puede ver un atributo determinado y si se puede modificar.

Multidioma

EE incorpora un mecanismo de traducción automática de los textos mostrados en la aplicación, de forma que puede mostrar toda su interfaz en el idioma configurado para cada usuario.

Hay que diferenciar dos tipo diferentes de elementos a traducir. Por una parte están los literales de elementos inventariables a través de la consola de administración de EE, como son los programas, informes, menús, etc, cuyos textos a mostrar son directamente traducibles en la consola de administración, y por otro lado, están los literales empleados por los desarrolladores en la construcción de los programas que conforman la aplicación. Estos últimos también son traducibles mediante la consola de administración, pero han de ser extraídos previamente del código compilado usando una utilidad proporcionada por Solmicro.

Una vez extraídos y traducidos, EE distribuye de forma automática un fichero a cada cliente, con los textos traducidos al idioma configurado para el usuario, fichero que será usado en tiempo de ejecución para traducir los literales de cada formulario.

Este fichero de diccionario se actualizará también de forma automática cada vez que se detecte que la versión del fichero ubicada en el cliente, es más vieja que la última fecha de actualización del diccionario, fecha que puede verse y modificarse por medio de la consola de administración.

Sistema de mensajería integrado

EE proporciona un sistema de mensajería integrado, que permite definir alertas personalizadas y cuyas características principales son las siguientes:

- Permite definir alertas detonadas por modificaciones en datos o ejecutadas de una forma programada.
- Permite la entrega de mensajes vía correo electrónico o mediante mensajes de aplicación, que se visualizan en un área específica de la aplicación. Esta última forma de entrega permite además cierta gestión de notificaciones y más interacción con la aplicación, pudiendo acceder directamente a los elementos que han originado la alerta desde el mensaje.
- Se ejecuta en un hilo separado del hilo de ejecución que realiza las modificaciones en los datos, por lo que apenas tiene repercusión sobre el tiempo de ejecución de los procesos de negocio.
- Permite la definición de las condiciones en las que se debe realizar la notificación, mediante la identificación del tipo de modificación: inserción, modificación, eliminación.
- Permite la definición de las condiciones en las que se debe realizar la notificación mediante un sistema de generación de filtros sobre los datos, guiado por las relaciones entre las entidades de la aplicación.
- Dispone de un sistema de extensión que permite resolver los casos de notificaciones más complejas, que no pueden resolverse mediante el diseñador de alertas integrado en la consola de administración.

Presentación de informes en formato PDF

Una de las nuevas características de EE, es que permite presentar en formato pdf en el equipo cliente, los informes generados en el servidor mediante el motor de generación de informes de Crystal Reports. Esta característica evita la necesidad de tener que instalar dicho motor de generación de informes en las máquinas cliente.

Integración "Búsqueda Federada" de Windows 7

Windows 7 introduce una característica denominada Búsqueda Federada, que permite integrar información procedente de fuentes externas, en el explorador de Windows. De esta forma el usuario puede acceder a toda la información de su interés desde una única herramienta.

EE proporciona los conectores y servicios necesarios para poder buscar información contenida en la aplicación mediante la búsqueda federada de Windows 7, permitiendo disponer de un resumen del elemento seleccionado en el propio Explorador de Windows y abrir el Shell de la aplicación, localizando automáticamente el elemento seleccionado dentro de la aplicación.

Integración con "Smart Tags" de Office 20xx

A partir de la versión 2003 de Office se incluye una característica denominada SmartTags, que básicamente consiste en la habilidad, por parte de Office, de reconocer etiquetas configurables y poder asociar acciones a dichas etiquetas.

EE proporciona los elementos necesarios para que se pueda acceder a información de la aplicación desde Word a través de SmartTags y el servicio proveedor búsqueda para Búsqueda federada.

Herramientas de desarrollo integradas en el IDE de Visual Studio

EE dispone de un complemento que se integra en el entorno de desarrollo de Visual Studio 2008, llamado Expertis.Tools.Designer, cuyo objetivo es agilizar el desarrollo de formularios según el modelo de desarrollo propuesto por EE. Para ello proporciona una interfaz conectada a la base de datos de metadatos, o base de datos de sistema, para permitir, mediante operaciones de arrastrar y soltar (Drag&Drop), crear controles correctamente enlazados a los orígenes de datos subyacentes en los formularios, y que permitirán, en tiempo de ejecución, la visualización y edición de dichos datos.

Estructura

