

Referencia	Tipo Documento	Actualizado el	Versión	Página
220_(A1)_21_ReglasdeDiseñoProyectosDeDesarrolloInterno.docx	General	07/11/2011	1.0	1/9
<logo >	Reglas de diseño de arquitectura para proyectos de desarrollo interno			

<LA EMPRESA>

Reglas de diseño de arquitectura para proyectos de desarrollo interno

Aprobación del Documento

Nombre	Cargo	Fecha

Referencia 220_(A1)_21_ReglasdeDiseñoProyectosDeDesarrolloInterno.docx	Tipo Documento General	Actualizado el 07/11/2011	Versión 1.0	Página 2/9
<logo >	Reglas de diseño de arquitectura para proyectos de desarrollo interno			

Control de Versión de la Plantilla

Control de Versiones

Historial de Cambios

Versión	Autor	Revisado/Aprobado por:	Fecha

Cambios

Versión	Cambios

Referencias

Ref. No.	Nombre Documento	Ubicación
001	Enterprise Library	Pagina Web
002	MVC .NET	Pagina Web
003	Entity Framework	Pagina Web

Definiciones, Acrónimos y Abreviaturas

Abreviatura / Acrónimo	Definición
MVC	Model view Controller
ORM	Object Relational Mapping

Referencia 220_(A1)_21_ReglasdeDiseñoProyectosDeDesarrolloInterno.docx		Tipo Documento General	Actualizado el 07/11/2011	Versión 1.0	Página 3/9
<logo >		Reglas de diseño de arquitectura para proyectos de desarrollo interno			

ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN	4
1	BACKEND BBDD	5
1.1	HERRAMIENTA ACCESO A DATOS	5
1.2	HERRAMIENTA GESTOR DE BASE DE DATOS.....	5
2	MVC.....	6
3	VALIDACIÓN.....	7
4	LOG	7
5	TRATAMIENTO DE EXCEPCIONES	7
6	SEGURIDAD.....	8
7	CACHE	8
8	SISTEMA DE REPORTING	9

Referencia	Tipo Documento	Actualizado el	Versión	Página
220_(A1)_21_ReglasdeDiseñoProyectosDeDesarrolloInterno.docx	General	07/11/2011	1.0	4/9
<logo> >	Reglas de diseño de arquitectura para proyectos de desarrollo interno			

1 Introducción

En este documento vamos a exponer las diferentes reglas que el proveedor deberá utilizar para el desarrollo de aplicaciones en <LA EMPRESA> en proyectos INTERNOS.

Tecnología principal: Microsoft dotNET, lenguaje C# y sistema SGBD MS-SQL Server.

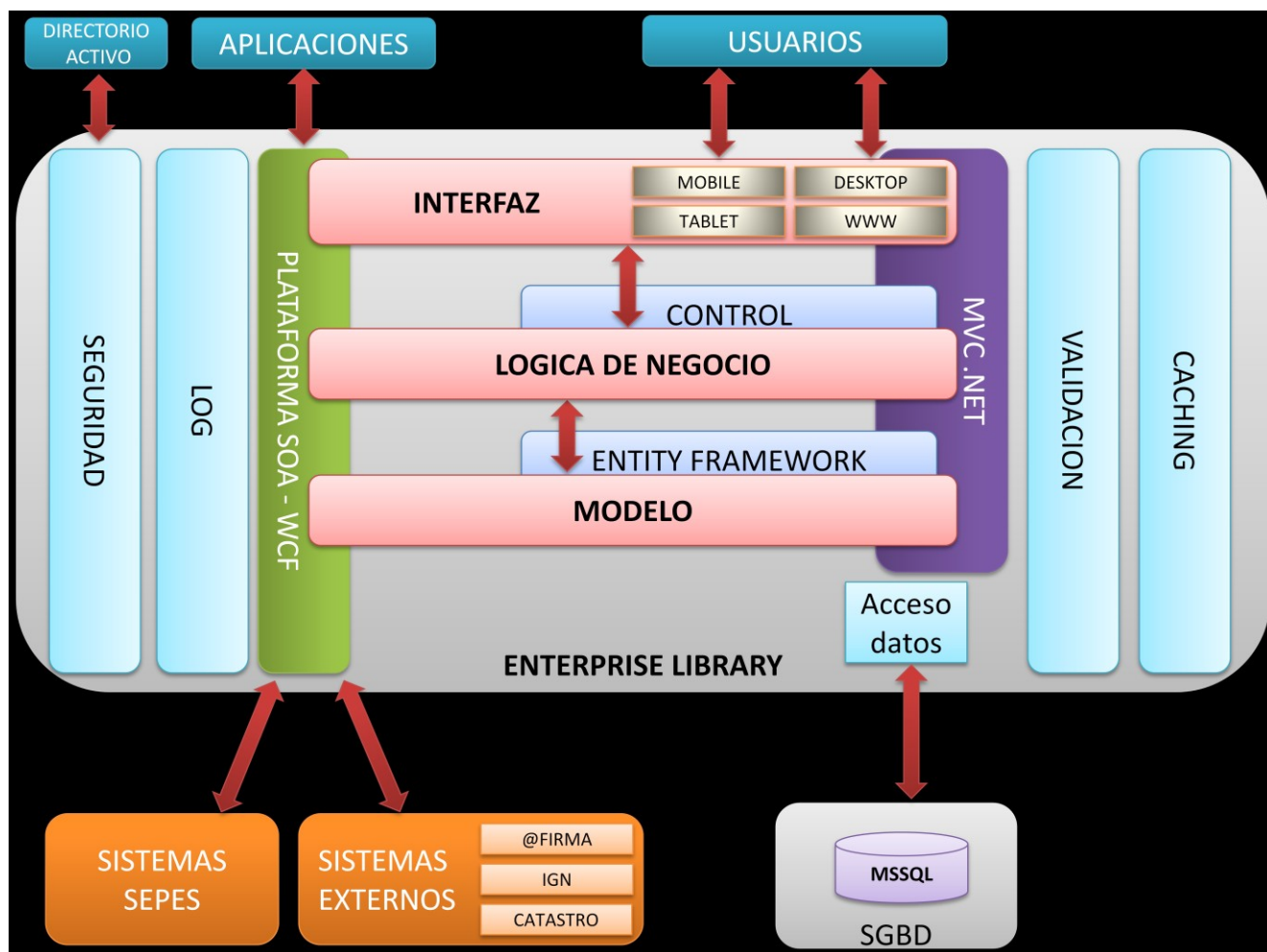


Figura 1

Referencia	Tipo Documento	Actualizado el	Versión	Página
220_(A1)_21_ReglasdeDiseñoProyectosDeDesarrolloInterno.docx	General	07/11/2011	1.0	5/9
<logo >	Reglas de diseño de arquitectura para proyectos de desarrollo interno			

1 BACKEND BBDD

1.1 Herramienta acceso a datos

En la elección de un framework para el acceso a los datos deberán cumplir las características:

- Independencia con el SGBD. Independientemente del proveedor de datos que sea, el framework debe ajustarse sin ningún tipo de esfuerzo adicional y que sea totalmente parametrizable.
- Centralización de la gestión de conexiones totalmente parametrizables ya que tiene que soportar cualquier cambio que pueda realizarse como por ejemplo el servidor, nombre de la base de datos, etc.
- La herramienta tiene que destacar en facilidad y ahorro de tiempo de desarrollo, tanto en la creación de la estructura de la capa de datos como el mantenimiento.
- El framework tiene que tener la capacidad de que coexistan varios sistemas de gestión de base de datos para una aplicación sin cambiar las clases de identidad.
- Permitir la encriptación de la cadena de conexión para evitar la exposición de datos confidenciales.

En el apartado de .NET tenemos el modulo de Data Access que posee el Enterprise Library 5.0 que contiene todas las características que <LA EMPRESA> necesita para el desarrollo de aplicaciones para entornos de Microsoft. Sobre todo resaltar la posibilidad en el Data Access de configurar de una manera sencilla la encriptación de la cadena de conexión a la base de datos.

También valoramos el uso de herramientas ORM dependiendo de la complejidad del proyecto. Su uso será siempre consensuado con los responsables de <LA EMPRESA>, en la plataforma .NET existe una herramienta que se ajusta a lo que demanda <LA EMPRESA> que es el Entity Framework 4.1.

1.2 Herramienta Gestor de base de datos

Para el sistema de gestor de base de datos en el cual nos declinamos es el Microsoft SQL Server 2008 R2. Esta herramienta cumple entre otras con las condiciones que demanda <LA EMPRESA>:

- Permite la creación de esquemas para la estructura de las tablas de la base de datos. Simplifica la concesión de permisos y permite delimitar el acceso a determinadas tablas, procedimientos almacenados, vistas, etc. (Ver figura 1).
- En el apartado de seguridad se caracteriza por la gestión propia de usuarios de acceso a la base de datos y también el acceso a la herramienta por los usuarios de dominio.
- La asignación de roles totalmente personalizados facilita la gestión de usuarios por parte del SQL Server.

Referencia	Tipo Documento	Actualizado el	Versión	Página
220_(A1)_21_ReglasdeDiseñoProyectosDeDesarrolloInterno.docx	General	07/11/2011	1.0	6/9
<logo >	Reglas de diseño de arquitectura para proyectos de desarrollo interno			

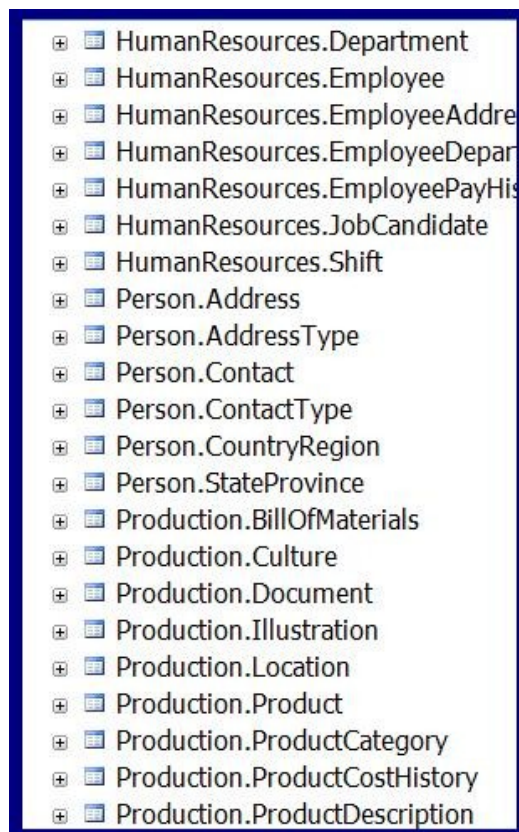


Figura 1

2 MVC

Para este patrón de arquitectura que pretende disociar la lógica de negocio, el acceso a los datos y la interfaz de usuario de una aplicación hemos pensado en el MVC .NET.

Las ventajas que nos presenta dicho patrón:

- La separación del Modelo de la Vista, es decir, separar los datos de la representación visual de los mismos.
- Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema
- Permite el escalamiento de la aplicación en caso de ser requerido.

A priori sería un componente perfectamente válido pero tiene sus desventajas:

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- La curva de aprendizaje del patrón del diseño es más alta que usando otros modelos más sencillos.

Referencia	Tipo Documento	Actualizado el	Versión	Página
220_(A1)_21_ReglasdeDiseñoProyectosDeDesarrolloInterno.docx	General	07/11/2011	1.0	7/9
<logo >	Reglas de diseño de arquitectura para proyectos de desarrollo interno			

Finalmente se usará dicho patrón previo consenso con los responsables de <LA EMPRESA>, ya que la utilización se valorará dependiendo de la complejidad del proyecto que se vaya a realizar, no exime de la separación lógica en tres capas presentada en las reglas de diseño de arquitectura generales.

3 VALIDACIÓN

Para las validaciones que se realicen en el entorno de .NET tiene que contener las siguientes características:

- Su objetivo es proteger la introducción de datos incorrectos y también frente ataques.
- Debe controlar la introducción de datos en la capa de la interfaz, aunque también en otras capas como la de negocios.
- Debería tenerse en cuenta otros elementos además de los datos de usuario como cookies, credenciales o información de contexto.

Para los desarrollos que se hagan en .NET las validaciones que se hagan en la aplicación se va a utilizar el modulo de validación de la herramienta del Enterprise Library 5.0.

4 LOG

En la utilización del modulo de log utilizaremos el que viene integrado en el framework de la Enterprise Library 5.0. Tiene las siguientes características que demanda <LA EMPRESA>:

- Tiene las características de escribir en varias localizaciones:
 - Registro de eventos
 - Correo electrónico
 - Base de datos
 - Cola de mensajes
 - Archivo de texto
 - En el VMI
- Es bastante simplificado para el uso de los desarrolladores ya que proporciona un pequeño conjunto de funciones fáciles de usar.
- Es fácilmente parametrizable para la configuración del destino al cual se van a almacenar los logs.

5 TRATAMIENTO DE EXCEPCIONES

Para las excepciones debe contener las siguientes premisas:

- Debe permitir crear una estrategia para el procesamiento de excepciones que se produzcan en todas las capas de de una aplicación.
- Debe proporcionar funciones de control de excepciones, tales como la capacidad de registrar la información de la excepción, la capacidad de ocultar información sensible mediante la sustitución de la excepción original.
- Debe permitir invocar los controladores de excepciones de una manera consistente. Es decir que los controladores se pueden utilizar en múltiples capas dentro de la aplicación.

Para la utilización de excepciones lo haremos con el modulo de Exception que viene en el framework del Enterprise Library 5.0

Referencia	Tipo Documento	Actualizado el	Versión	Página
220_(A1)_21_ReglasdeDiseñoProyectosDeDesarrolloInterno.docx	General	07/11/2011	1.0	8/9
<logo >	Reglas de diseño de arquitectura para proyectos de desarrollo interno			

6 SEGURIDAD

Para implementar el modulo de seguridad hemos decidido el que viene con la herramienta de la Enterprise Library 5.0 ya que contiene las siguientes características:

- Se reduce la necesidad de escribir código repetitivo para realizar las tareas más comunes en el apartado de seguridad.
- Ayuda a mantener las buenas prácticas de seguridad consistentes, tanto dentro de la aplicación como fuera.
- Facilita la curva de aprendizaje para los desarrolladores.
- Proporciona implementaciones que se pueden utilizar para resolver problemas comunes de seguridad de aplicaciones.
- Es extensible en el modo que soporta implementaciones personalizadas de los proveedores de seguridad.

7 CACHE

La caché se emplea en ciertas aplicaciones donde la obtención de datos es costosa, el objetivo es mantener esos datos en memoria para evitar múltiples recuperaciones. El almacenamiento en caché debe hacerse en función de la memoria disponible, aparte debe establecerse un mecanismo de eliminación de elementos de la caché cuando no sean necesarios.

Para el modulo de caché de las aplicaciones utilizaremos el que trae el framework de la Enterprise Library 5.0. Las características más importantes que nos ofrece el modulo de cache:

- Caching Application Block permite persistencia del cache.
- Es posible mantener múltiples instancias de cache independientes identificables por nombre.
- Es posible configurar el cache usando el archivo de configuración sin necesidad de recompilar la aplicación.
- Es extensible, se pueden implementar nuevas políticas de expiración, medios de almacenamiento, e incluso editar el código fuente para incluir nueva funcionalidad.

Referencia	Tipo Documento	Actualizado el	Versión	Página
220_(A1)_21_ReglasdeDiseñoProyectosDeDesarrolloInterno.docx	General	07/11/2011	1.0	9/9
<logo >	Reglas de diseño de arquitectura para proyectos de desarrollo interno			

8 SISTEMA DE REPORTING

Debe seguirse la arquitectura de reporting presentada en el documento general de reglas de diseño de arquitectura. La arquitectura de informes de <LA EMPRESA> está basada en dos sistemas diferentes. Por un lado SAP-BO y por otro el mundo OLAP con Analysis Services de Microsoft.

Las aplicaciones deberán desarrollar sus informes operativos, si se requiere dentro de la plataforma de BI, se deberán implementar con SAP-BO. En caso de no incluirse en la plataforma general de BI, se puede utilizar el sistema de Reporting Services de MS-SQL Server existente en <LA EMPRESA>. Este tipo de elecciones, se deja en función del tipo de proyecto a desarrollar.

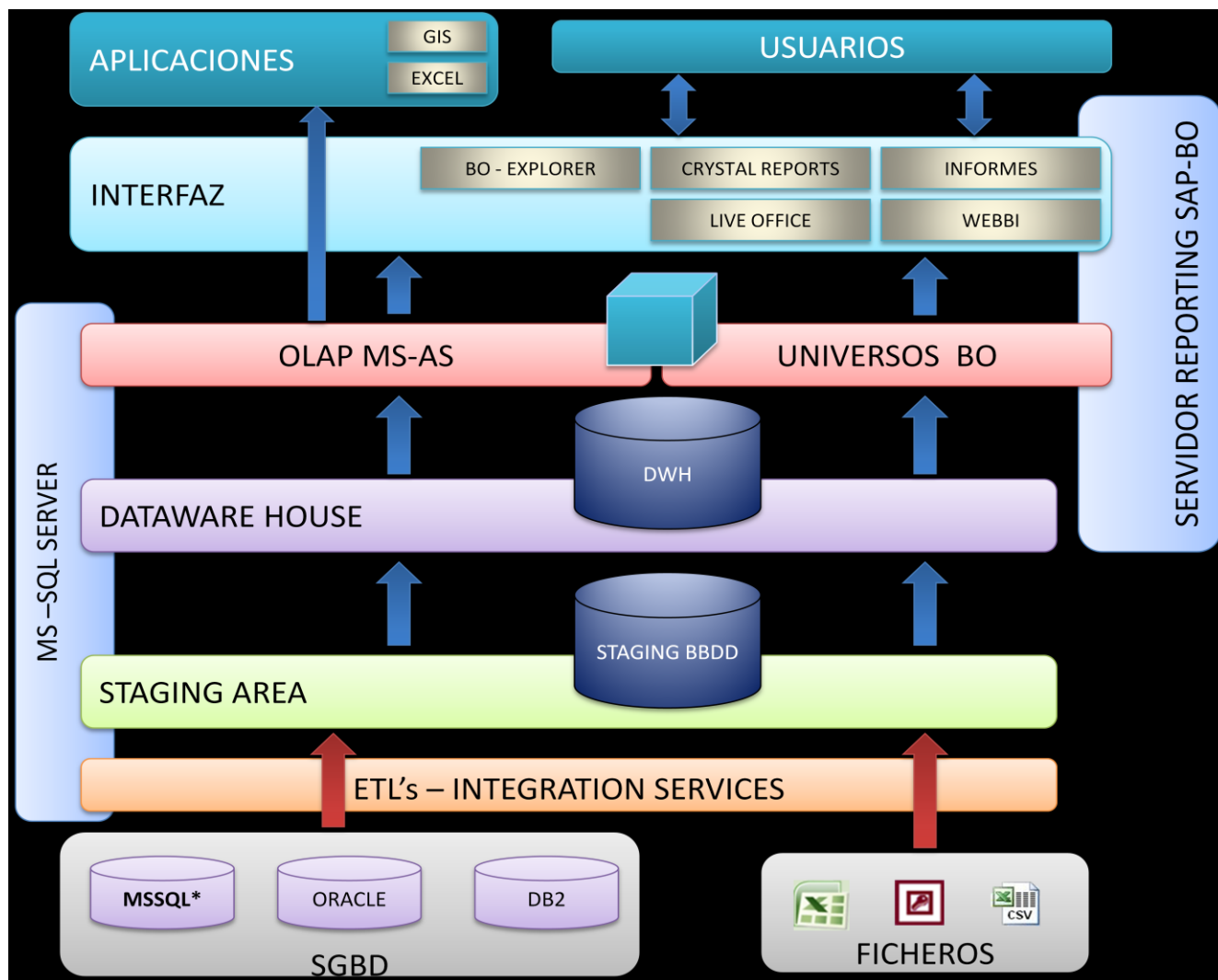


Figura 5