IndignaFramework

# Grupo

Tecnólogo en Informática: **tsi.net04**

# Integrantes

Vanessa Revetria - [lavarece@hotmail.com](mailto:lavarece@hotmail.com) - 4.472.280-0

Maximiliano Silvera - [maxi.silveraf@gmail.com](mailto:maxi.silveraf@gmail.com) - 4.106.938-6

Andrés Aldao - [andrew5900@hotmail.com](mailto:andrew5900@hotmail.com) - 4.770.435-0

Juan Manuel Miraballes - [jmg216@hotmail.com](mailto:jmg216@hotmail.com) - 4.115.494-3

Gonzalo Castro - elgonzacastro@hotmail.com - 4.607.434-0

Índice

[Grupo 1](#_Toc327918701)

[Integrantes 1](#_Toc327918702)

[Resumen 3](#_Toc327918703)

[Palabras Claves 4](#_Toc327918704)

[Introducción 4](#_Toc327918705)

[Marco Conceptual 5](#_Toc327918706)

[Multi Tenant 5](#_Toc327918707)

[Web Service 5](#_Toc327918708)

[WCF 5](#_Toc327918709)

[Arquitectura en capas 6](#_Toc327918710)

[Descripción Del Problema 6](#_Toc327918711)

[Solución Planteada 7](#_Toc327918712)

[Front Office 8](#_Toc327918718)

[Back Office 9](#_Toc327918719)

[Arquitectura Del Sistema 10](#_Toc327918720)

[Presentation Layer 10](#_Toc327918721)

[Business Layer 10](#_Toc327918722)

[Persistence Layer 10](#_Toc327918723)

[Entity Layer (capa lateral) 11](#_Toc327918724)

[Implementación 11](#_Toc327918725)

[Productos y Herramientas 11](#_Toc327918726)

[Tabla - Evaluación de Productos y Herramientas 11](#_Toc327918727)

[Problemas Encontrados 12](#_Toc327918728)

[Evaluación De La Solución 12](#_Toc327918729)

[Desarrollo del Proyecto 12](#_Toc327918731)

[Conclusiones y Trabajo a Futuro 12](#_Toc327918732)

[Referencias 12](#_Toc327918733)

# Resumen

En los últimos años se ha incrementado la cantidad de grupos de protesta, basados inicialmente en el movimiento Occupy Wall Street, el cual, realiza actividades de protesta contra instituciones financieras en New York. Actualmente, estos grupos abarcan diferentes temáticas como por ejemplo protestas sobre el medio ambiente, reivindicaciones sociales y políticas, etc.; y a su vez, se han expandido por muchas partes del mundo.

En este artículo se presenta el diseño e implementación de una plataforma informática que de soporte para este tipo de movimientos donde se permita la gestión, coordinación y organización de las diferentes actividades que estos grupos pueden llevar a cabo.

Para ello, se realizará un sistema desarrollado con tecnologías .NET que constará de dos módulos. Uno de ellos front office, de acceso público y utilizando MVC 3 con Razor; el segundo, un back office para la administración del sistema, de acceso restringido y utilizando Windows Presentation Fundation (WPF).

## Palabras Claves

SQL, .NET, Windows Comunication Fundation (WCF), Windows Presentation Fundation (WPF), MVC 3, Razor, Movimientos de protesta, Multi-Tenant, ADO (Data Access Object).

# Introducción

Tal como se ha mencionado anteriormente, en el último tiempo se han incrementado la cantidad de grupos que sienten la necesidad de protestar sobre temas que resultan de importancia para la sociedad actual. Estos grupos han crecido en número y se han ido extendiendo a lo largo del planeta. Debido a esto, es de suma importancia generar un espacio en la web para todas aquellas personas interesadas, en donde se pueda compartir información de interés común y conocer las diferentes actividades que se organizan por parte de sus integrantes.

El objetivo es desarrollar una plataforma de soporte para estos movimientos de protesta, con la finalidad de centralizar toda la información relevante para los grupos. A su vez, se busca que el sistema pueda servir como infraestructura para la gestión y coordinación de varios de estos movimientos. Cada grupo tendrá a su disposición un único portal web, el cual deberá actuar como punto de integración de toda la información relevante al mismo. El contenido de esta información variará dependiendo del foco temático del grupo como por ejemplo protestas medio ambientales, reivindicaciones estudiantiles, lucha por la igualdad social, etc.

En cada uno de estos sitios web correspondiente a un movimiento de protesta en particular, se podrá compartir información proveniente de diferentes fuentes de datos, como puede ser YouTube, Twitter, BBC, CNN, Wikipedia, etc. Además, se podrá compartir diferentes tipos de recursos como links, imágenes y videos. Desde el portal web de un movimiento se dará la posibilidad de coordinar actividades en donde los integrantes serán informados e invitados a participar y se facilitará la comunicación entre ellos mediante un chat.

Se espera que cada sitio web de los diferentes movimientos de protesta tenga una estética totalmente distinta uno de otro, contenidos específicos y url diferente. Es decir, a pesar de ser una plataforma común a todos los grupos y compartir una misma lógica, su estética deberá ser completamente diferente.

En el marco de la edición 2012 de la asignatura Taller de Sistemas de Información 1, se plantea implementar una solución para dicho interés.

Analizado el problema, surgió la necesidad de implementar dos sistemas, un sistema back office de tipo aplicación de escritorio desarrollada con WPF, donde se manejará la gestión del sistema web por parte de un administrador y un front office de tipo aplicación web que será de acceso público.

# Marco Conceptual

En esta sección se describen varios conceptos sobre los cuales se deben tener conocimiento para la implementación del sistema. Entre estos conceptos podemos encontrar el de Multi-tenant que utilizará para resolver el problema de los multi-sitios para la aplicación web. Además se utilizarán WCF para la comunicación entre la capa de negocio y presentación (para comprender el concepto de WCF se recomendable tener una noción de web services); también debemos tener en cuenta como concepto importante el de Arquitectura en capas que es fundamental para el desarrollo escalable, desacoplable y mantenible de la plataforma.

A continuación detallamos dichos conceptos.

## Multi-Tenant

Multi-Tenant es un enfoque de un arquitectónico de sistemas de software. En un sistema donde se utilice este enfoque, los diferentes clientes consumen un servicio desde una misma plataforma, un mismo modelo de datos y mismos servidores. Este enfoque se refiere a un principio de arquitectura de software donde una misma instancia del software que se ejecuta en un servidor y da servicio a múltiples clientes simultáneamente, y cada cliente trabaja con una instancia de la aplicación virtual personalizada.

Este enfoque es muy útil para aplicaciones web en cuanto a eficiencia en costos y diseño.

## Web Service

Los web services son funcionalidades que pueden ser utilizadas por una aplicación en particular. Estas funcionalidades pueden ser descriptas, publicadas, localizadas e invocadas a través de una red (generalmente Internet). Una de las características principal de los servicios es que éstos se encuentran en una especie de “caja negra”, lo que nos permite reutilizarlos sin importarnos como fueron implementados. La forma de accederlos es utilizando protocolos web como HTTP y XML, esto permite la interoperabilidad entre las aplicaciones, por lo cual, los consumidores de los web services pueden ser implementados en cualquier plataforma y en cualquier lenguaje de programación.

## WCF

WCF es un conjunto de librerías que provee Microsoft en el framework .NET para aquellas aplicaciones orientadas a servicios. WCF es la evolución de los web services de Microsoft, por esta razón provee un rango de funcionalidades superior a su predecesor. Algunas de las ventajas frente a los web services, es que éstos, sólo pueden ser accedidos a través de HTTP en cambio un WCF se puede hospedar en un servidor web, puede ser un servicio de Windows y utiliza una variedad de protocolos más amplia.

## Inversión de Control (IoC)

La inversión de control es un patrón de programación donde se invierte la técnica tradicional del flujo de una aplicación. Comúnmente es que el código que se implementa el que llama a librerías para su ejecución, por el contrario en la inversión de control, son las librerías las que llaman a nuestro código. Con esto se logra delegar a otro componente (un framework por ejemplo) la responsabilidad de crear las instancias necesarias en lugar de crearlas nosotros mismos.

## Inyección de Dependencias

Este concepto está muy ligado al anterior debido a que en un escenario donde utilizamos IoC, delegamos a otro componente no sólo la creación de las instancias sino también la inyección de las dependencias, o sea, evitamos que nuestros objetos creen instancias nuevas de aquellos objetos de los cuales dependen, proveyéndoselos ya listos para operar (por ejemplo pasándoselos como parámetros al constructor).

## Arquitectura en capas

La arquitectura en capas es un estilo de programación donde su objetivo es separar los diferentes aspectos del desarrollo de una aplicación, como por ejemplo cuestiones de la presentación, lógica de negocio y mecanismos de almacenamiento. Comúnmente la comunicación entre las diferentes capas se da mediante interfaces bien definidas que nos permiten la correcta separación de las mismas. Además, este tipo de diseño permite la instalación distribuida de una aplicación, mejorando su escalabilidad, disponibilidad y utilización de recursos.

# Descripción Del Problema

La motivación para la creación de esta plataforma reside en el diseño y construcción del un sistema para aquellos movimientos de protesta que desean tener un espacio para compartir información y organizar sus actividades desde un portal web. Se desea generar un framework para gestionar a todos estos grupos con el fin de centralizar una lógica de negocio que pueda servir de soporte para los todos los sitios de protesta.

Cada movimiento tendrá su propio portal web los cuales serán completamente diferentes desde el punto de vista del diseño, tendrán url distinta y mostrarán contenidos específicos dependiendo de la temática a la cual estén relacionados.

El sistema deberá contar con dos módulos: uno de front office de acceso público que será el encargado de mostrar los sitios al usuario y otro de back office de acceso privado encargado de la administración de los datos de la plataforma. Entre las funcionalidades que debe brindar el back se encuentra, gestión de usuarios, gestión de los grupos de protesta, gestión de variables del sistema (más adelante se explicará con más detalle esta funcionalidad), y creación de reportes. El front deberá permitir el registro de usuarios, la creación de convocatorias (actividades que el movimiento realizará) y visualización de recursos compartidos como videos, imágenes, links y novedades. Este módulo también contará con un chat que permitirá la comunicación entre los usuarios de un grupo.

Además, se plantean las siguientes funcionalidades opcionales: integración con Facebook y Twitter, integración entre servidores, desarrollo de un cliente Windows Phone, detección de dispositivos móviles y deploy en Windows Azure.

Para el desarrollo de la plataforma se exige el framework .NET 4.0, con todas sus tecnologías integradas.

## Solución Planteada

La solución al problema plateado se ha llevado a cabo durante el proceso de análisis al comienzo del proyecto, y se ha ido modificando en el correr del desarrollo debido a que se ha ido adaptando a las diferentes necesidades y a los problemas encontrados durante la implementación.

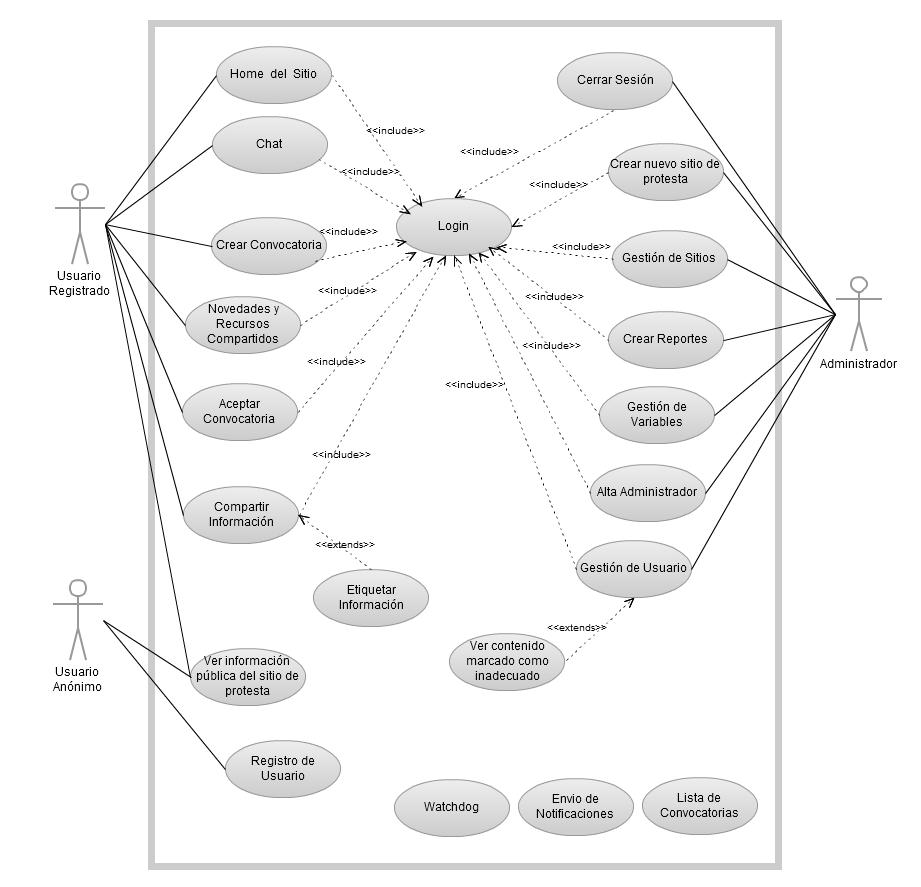
Además de los ya conocidos módulos a implementar (back office y front office), en esta sección se explicará las decisiones que se han tomado para desarrollar el sistema, los casos de uso que se necesitan para cubrir todos los requerimientos y las soluciones que se han tomado al momento de enfrentarse a los diferentes problemas de implementación.

Para desarrollar el back office se ha utilizado la tecnología WPF. Se ha elegido utilizar una aplicación de escritorio bajo el argumento de manejar otro tipo de tecnología diferente a la que se va a utilizar en el front. El objetivo de esto, es adquirir un conocimiento más amplio del framework .NET ya que de la forma planteada inicialmente no se llegaría ver esta tecnología. Desde el punto de vista funcional el back office consta de un menú el cual tiene habilitado las siguientes funcionalidades para usuario específicos (administradores) previo inicio de sesión: Gestión de Administradores, Gestión de Usuarios, Gestión de Grupos, Gestión de Variables, Gestión de Reportes y el Watchdog.

En el caso del módulo front office se ha desarrollado una aplicación web para cumplir con todos los requerimientos. Entre ellos se encuentra el Home del sitio, aquí el usuario no registrado podrá ver información pública, pero para tener acceso a las demás funcionalidades, deberá registrarse para después loguearse y desde ese punto sí, poder acceder a otras secciones de la web y ser parte del movimiento de protesta.

Como anexo a la plataforma se han incluido algunos casos opcionales como la integración entre grupos (en nuestro caso lograda con otra instancia de nuestra misma aplicación), un cliente Windows Phone.

A continuación presentamos los casos de uso requeridos en el sistema.



***Figura 1:*** *Detalle de casos de uso de la solución*

### Front Office

***Home del sitio***: Desplegará todas las opciones detalladas a continuación.

***Login***: Única manera de que un usuario pueda ingresar al sitio, habilitándole todas las opciones.

***Chat*:** Sistema que se realizará para que los usuarios registrados puedan chatear entre ellos.

***Crear Convocatoria***: En esta sección se podrá ingresar todos los datos necesarios para crear una convocatoria. La convocatoria será visible para todos los usuarios registrados, donde ellos podrán marcar su asistencia.

***Novedades y recursos compartidos***: Se detallarán todos los recursos compartidos por los usuarios registrados al sitio. El recurso puede tener la visibilidad habilitada para todos los usuarios o sólo para usuarios registrados.

***Aceptar Convocatoria***: Es la opción donde los usuarios registrados pueden indicar que asistirán a la convocatoria.

***Compartir Información***: En esta sección se podrá subir y compartir la información deseada.

***Etiquetar Información***: Toda información publicada podrá ser etiquetada, la idea es que el usuario tenga una notificación indicando que fue etiquetado.

***Ver información pública del sitio***: Se listará a los usuarios no registrados la información del sitio que fue publicada con visibilidad habilitada para todos los usuarios.

***Cerrar sesión***: Es la opción para que el usuario cierre su sesión.

### Back Office

***Crear nuevo sitio de protesta***: Se creará un nuevo sitio, en una nueva url, con un diseño distinto.

***Gestión de sitios***: Podrá editar otros sitios creados.

**Crear Reportes**: Se podrán crear reportes para obtener diferentes estadísticas.

***Gestión de variables***: Habrá ciertas variables que controlarán el sitio. Las variables podrán ser editadas aquí.

***Alta Administrado***r: Se podrá dar de alta a otros administradores.

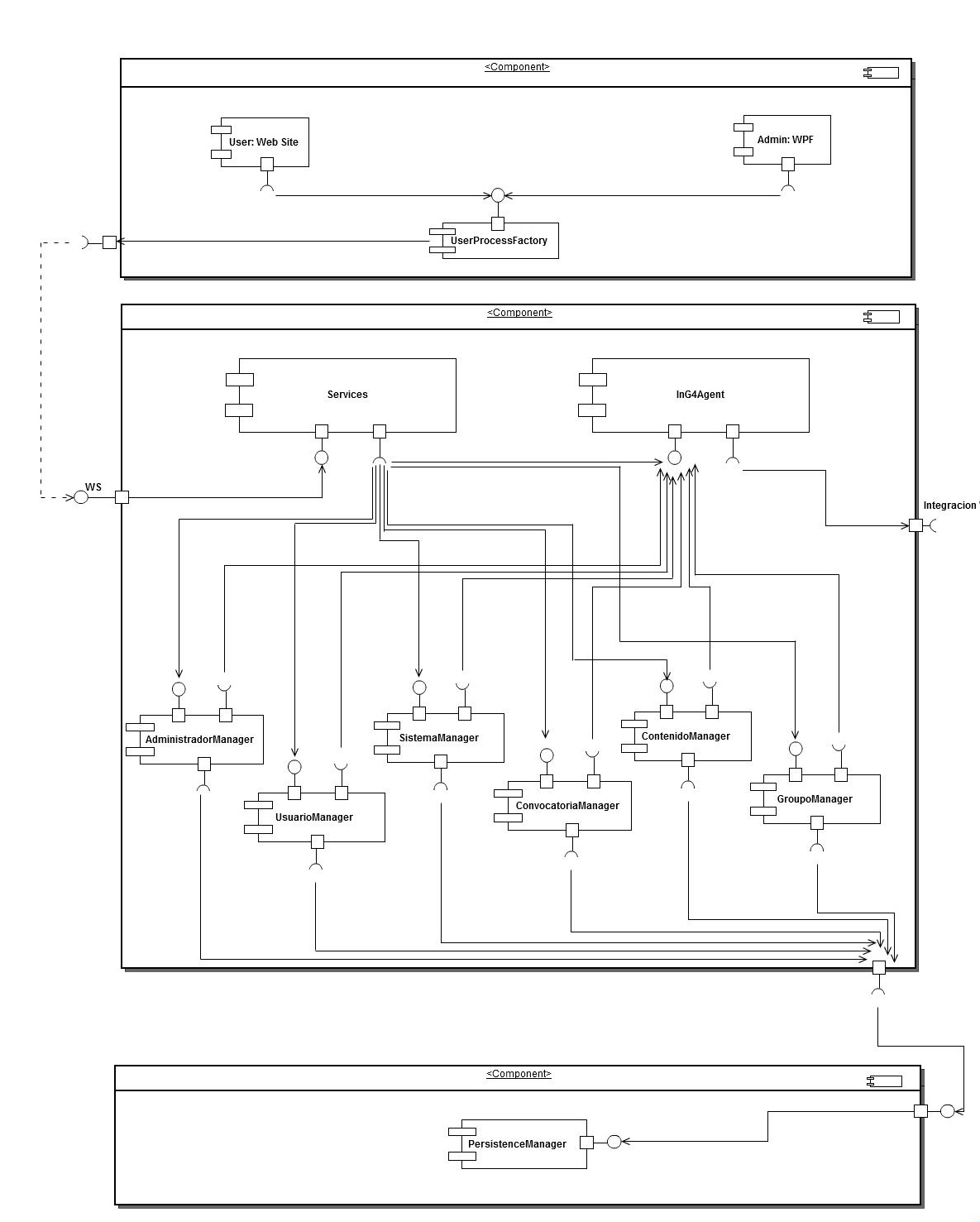
***Gestión de usuarios***: Se controlará la actividad de los usuarios.

***Ver contenido marcado como inadecuado***: Se permitirá ver todo el contenido marcado de manera inadecuada, para evaluar decisiones.

***Watchdog***: Proceso que dará de baja contenidos inadecuados.

***Envío de notificaciones***: Es la notificación que se envía luego de crear una convocatoria.

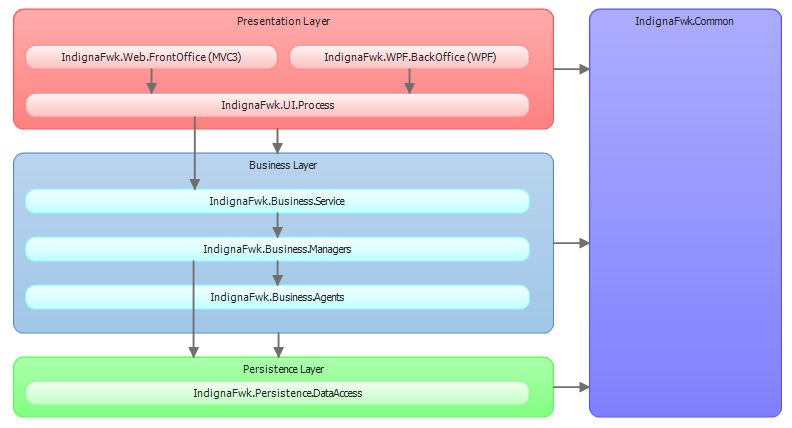
***Lista de convocatorias***: Se listarán todas las convocatorias creadas en el sitio.



***Figura2:*** *Vista de subsistemas*

# 

# Arquitectura Del Sistema

******

***Figura 3:*** *Arquitectura del sistema*

La arquitectura utilizada para el desarrollo del sistema es una arquitectura clásica de 3 capas, dónde las capas superiores consumen de las de más abajo. Además se ha desarrollado una capa lateral que será referenciada en todas las capas horizontales.

Debido al diseño de la arquitectura se ha logrado un sistema mantenible que da la posibilidad de agregar nuevas funcionalidades sin tener que modificar el código ya escrito.

En la ***figura 3*** puede observarse el diseño de la misma.

## Presentation Layer

Capa encargada de mostrar los datos al usuario final. Esta capa consta de otros módulos con el fin de obtener un sistema desacoplado, escalable y que cumpla con las funcionalidades requeridas.

### IndignaFwk.UI.FrontOffice

Es el módulo encargado de mostrar los datos al usuario final y se comunica directamente con el UI.Process. Se ha creado un proyecto web para implementarlo.

### IndignaFwk.UI.BackOffice

Aquí se implementa el sistema para que los administradores gestionen los movimientos de protesta utilizando la tecnología WPF. Este módulo también se comunica directamente con el UI.Process.

### IndignaFwk.UI.Process

Este módulo consume los servicios expuestos en la capa de negocio y es referenciado por IndignaFwk.UI.BackOffice e IndignaFwk.UI.FrontOffice.

## Business Layer

Capa que contiene las reglas de negocio de la aplicación. Se comunica con la capa de persistencia mediante una interfaz definida en los ADO.

### IndignaFwk.Business.Services

Aquí están ubicados los servicios, para los cuales se ha utilizado WCF, que exponen funcionalidades desde la capa de negocio hacia el exterior de la capa. Estos servicios serán consumidos por la capa de presentación, en particular por el por el módulo UI.Process.

### IndignaFwk.Business.Managers

En este módulo se resuelve la parte de la lógica de negocio del framework. Se han creado diferentes managers para una correcta separación de las funcionalidades y una factoryManager para un mejor manejo de los mismos.

### IndignaFwk.Business.Agents

Este módulo se utilizará para la integración con otros grupos y para la integración de fuentes de datos.

## Persistence Layer

Capa de acceso a datos. Esta capa se conecta directamente con la base de datos y es la encargada de crear, editar, borrar y obtener elementos de la base mediate objetos ADO (Data Access Object).

### IndignaFwk.Persistence.DataAccess

Módulo dentro de la capa de acceso a datos con los ADOs y sus correspondientes interfaces para la correcta comunicación con la base de datos.

## Common Layer (capa lateral)

Esta es una capa lateral que será utilizada por todas las demás capas.

### IndignaFwk.Common

Módulo de la capa lateral que contiene las entidades de negocio, enumerados y funcionalidades utilitarias y de seguridad.

## Requerimientos Especiales

### Multi-Tenant

### Utilización API GoogleMaps

### Integración con otros Grupos

### Login con Facebook

# Implementación

* Se utiliza el IDE de Microsoft Visual Studio 2010 profesional SP1, integrado con el sistema de repositorio web Ankhsvn. Para la creación de la base de datos se utiliza Microsoft SQL Server 2008, con el IDE de SQL Server Management Studio.
* Para el acceso a datos se utilizó ADO.NET.
* Para la comunicación entre la capa de presentación y la lógica de negocio se utilizó WCF.
* En el módulo FrontOffice se utilizó el patrón MVC con tecnologías ASP MVC 3 y las vistas Razor. Además, para resolver diferentes problemas y desarrollar funcionalidades específicas en la presentación, se utilizó ajax, jquery, estilos CSS, APIs de GoogleMaps, Youtube y Wikipedia.
* Se utilizó el servidor Web IIS 7.5 para hacer el deploy de la aplicación web y los servicios.
* Para el módulo BackOffice se utilizó Windows Presentation Fundation (WPF) realizando una aplicación de escritorio para la administración de los movimientos de protesta.
* Los clientes SVN utilizados fueron tortoise para la documentación y material en general, y ankhsvn para el control de versiones del sistema IndignadoFramwork.

## Productos y Herramientas

### Tabla - Evaluación de Productos y Herramientas

|  |  |  |  |
| --- | --- | --- | --- |
| **Producto** | **Puntos Fuertes** | **Puntos Débiles** | **Evaluación General** |
| ADO.NET |  |  |  |
| WCF | Utilización de variedad de protocolos.  Puede ser alojado en servidor web (iis). |  |  |
| MVC 3 | Diseño claro. Facilita la implementación. |  |  |
| Razor | Sintaxis compacta, fácil de escribir y minimiza la tipeo de código. |  |  |
| API Google Maps | Disponible públicamente. |  |  |
| WPF | Mejor experiencia de usuario que Windows forms. |  |  |
| CSS | Diseño desacoplado del resto del código.  Posibilidad de armar diferentes presentaciones en una misma página. | Dependiendo del navegado utilizado la página maquetada con CSS puede verse distinta. |  |
| JQuery | Genera contenido dinámico a nuestras páginas de manera sencilla.  Ahorra líneas de código. |  |  |
| Ajax | Mejor experiencia de usuario.  Reducción de tráfico de datos al servidor. |  |  |
| IIS 7.5 | Pools independientes.  Interfaz de administración.  Fácil configuración. | No es multiplataforma. |  |
| SQL Server 2008 |  |  |  |
| SQL Managment | Interfaz que facilita el manejo de datos. |  |  |
| Windows Phone |  |  |  |
| AnkhSvn | Integrado a Visual Studio. |  |  |
| Tortoise | Fácil manejo. | No fue útil como repositorio de código. | Buena: Se utilizó para documentación. |
| Visual Studio 2010 | Rico en la experiencia de usuario, fácil integración con tecnologías compatibles. |  |  |

## Problemas Encontrados

# Evaluación De La Solución

[En esta sección se evalúa críticamente la solución identificando y mencionando brevemente sus puntos fuertes y puntos débiles.]

# Desarrollo del Proyecto

[En esta sección se describe cómo fue el desarrollo del proyecto, porcentaje de tiempo dedicado a cada tarea, desviación con respecto al cronograma inicial y análisis de los posibles motivos que llevaron a que el proyecto se desarrollara de la forma descripta.]

# Conclusiones y Trabajo a Futuro

[En esta sección se presentan las conclusiones y trabajo a futuro.]

# Referencias

Laboratorio de Integración de Sistemas. TSI 1 – Trabajo Obligatorio. Año 2011. <http://www.fing.edu.uy/inco/cursos/tsi/TSI1/2011/laboratorio/Obligatorio2011.pdf>

Microsoft .NET Framework

<http://www.microsoft.com/net/>

ACM Proceedings Templates  
<http://www.acm.org/chapters/policy/toolkit/template.html>