

**PORTALES WEB PARA LAS LÍNEAS DE NEGOCIO DE  
FAMILIA/SCA, PARA COSAS DE FAMILIA, FAMILIA  
INSTITUCIONAL Y NOSOTRAS ONLINE**

**JUAN CAMILO ATEHORTUA ZULUAGA**

**CORPORACIÓN UNIVERSITARIA LASALLISTA  
FACULTAD DE INGENIERÍAS  
INGENIERÍA INFORMÁTICA  
CALDAS- ANTIOQUIA  
2011**

**PORTALES WEB PARA LAS LÍNEAS DE NEGOCIO DE  
FAMILIA/SCA, PARA COSAS DE FAMILIA, FAMILIA  
INSTITUCIONAL Y NOSOTRAS ONLINE**

**JUAN CAMILO ATEHORTUA ZULUAGA**

**Trabajo de grado para optar el título de Ingeniería Informática**

**ASESOR  
ALEJANDRO DIAZ BETANCOURT  
INGENIERÍA INFORMÁTICA**

**CORPORACIÓN UNIVERSITARIA LASALLISTA  
FACULTAD DE INGENIERÍAS  
INGENIERÍA INFORMÁTICA  
CALDAS- ANTIOQUIA  
2011**

**Nota de Aceptación**

---

---

---

---

---

---

**Firma del presidente del jurado**

---

**Firma del jurado**

---

**Firma del jurado**

---

**Ciudad y fecha (día, mes, año)**

---

## **AGRADECIMIENTOS**

En primer lugar agradezco a Dios, porque Él me dio las fuerzas para poder cumplir con todos mis tareas y realizarlas con éxito. A mis padres porque me incentivaron a lograr mi mayor meta "Ser ingeniero Informático". Quiero agradecer a todas aquellas personas que me apoyaron en las dificultades durante todo el proceso de práctica empresarial como Mauricio bedoya, un gran profesor, profesional y sobretodo una gran persona, a mi primer jefe que tuve en mi experiencia laboral gloria Henao una persona muy calmada y muy buena consejera.

También quisiera agradecer a Alejandro Díaz por la colaboración que me brindo para la creación del proyecto de práctica y en algunas situaciones laborales las cuales me hicieron crecer como profesional.

## TABLA DE CONTENIDO

1. JUSTIFICACIÓN .....	16
2. OBJETIVOS.....	17
2.1 OBJETIVO GENERAL.....	17
2.2 OBJETIVOS ESPECÍFICOS.....	17
3. RESEÑA.....	18
3.1 HISTORIA.....	18
3.2 OBJETIVOS PRAGMA .....	19
3.3 ALCANCE.....	19
3.4 POLITICA DE CALIDAD .....	20
3.5 VALORES.....	20
Figura 3: Estructura organizacional .....	22
3.6 PORTALES.....	23
3.6.1 Algunos de los clientes.....	23
3.6.2 Típicamente un portal apoya el cumplimiento de los siguientes objetivos.....	23
3.6.3 Objetivos de los portales .....	24
3.6.4 Objetivo: Comunicación.....	25
3.6.5 Objetivo: Posicionamiento .....	25
3.6.6 Objetivo: Fidelización .....	25
3.6.7 Objetivo: Ventas .....	25
3.6.8 Objetivo: Servicio al cliente.....	26
3.6.9 Objetivo: Soporte de Operaciones.....	26
3.6.10 ¿Cómo apoya Pragma el logro de los objetivos de sus clientes a través de un Portal?.....	26
3.7 LINEAS DE NEGOCIO .....	28
3.7.1 Fábrica de software .....	28
3.7.2 SOA.....	29
3.7.3 BPM.....	29
3.7.4 Algunos clientes para las líneas de negocio software, SOA y BPM .....	29
3.7.5 Agencia Digital.....	29
3.7.6 CERO .....	30
4. MARCO TEÓRICO .....	31
4.1 ESTADO DEL ARTE.....	31
4.1.1 Portales .....	31

4.2	.NET.....	33
4.2.1	Componentes .....	33
4.2.2	Common Language Runtime.....	34
4.2.3	Características.....	34
4.3	Microsoft SQL Server.....	35
4.3.1	Características de Microsoft SQL Server .....	35
4.3.2	Programación .....	35
4.4	ORACLE DATABASE .....	37
4.5	ASP.NET .....	37
4.5.1	Características.....	37
4.6	EPISERVER CMS .....	39
4.7	FIREBUG.....	40
5.	DISEÑO METODOLOGICO.....	41
5.1	METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	41
5.1.1	Documentar la toma de requerimientos.....	41
5.1.2	Documentar las Herramientas .....	43
5.1.3	Documentar los desarrollos específicos .....	45
5.2	ADMINISTRACIÓN DE LAS COMUNICACIONES.....	45
5.2.1	Participantes en el proyecto .....	45
5.3	TSP\PSP.....	48
5.3.1	Metodología TSP / PSP.....	48
5.3.2	Estos datos los utilizan para: .....	48
5.4	ESTANDAR CODIFICACION C#.....	56
5.4.1	Nombres .....	56
6.	CONCLUSIONES .....	58
	BIBLIOGRAFIA.....	59
	ANEXOS.....	60

## INDICE DE FIGURAS

Figura 1: Objetivos .....	19
Figura 2: Mapa de procesos .....	21
Figura 3: Estructura organizacional.....	22
Figura 4: Clientes .....	23
Figura 5: Metodología .....	27
Figura 6: Algunos clientes Software, SOA Y BPM .....	29
Figura 7: Servicios Estrategicos y creativos.....	30
Figura 8: Clientes CERO .....	30
Figura 9: Plantillas .....	43
Figura 10: Team Software Process .....	49
Figura 11: Metodología TSP/PSP .....	49
Figura 12: Curso de PSP .....	51
Figura 13: Tiempos de estimacion de errores.....	51
Figura 14: Defectos eliminados en compilacion y pruebas.....	52
Figura 15: Productividad.....	52

## INDICE DE TABLAS

Tabla 1. Proceso de levantamiento de Requisitos .....	54
Tabla 2. Estandares de codificacion C# .....	56

## INDICE ANEXOS

Anexo A. Detalle de las funcionalidades .....	60
Anexo B. Restricciones técnicas .....	61
Anexo C. Atributos de calidad .....	62
Anexo D. Listado de pendientes .....	63
Anexo E. Checklist diseño de alto nivel .....	64
Anexo F. Checklist inspeccion DLD .....	65
Anexo G. Checklist inspección de codificación .....	67

## GLOSARIO

**Microsoft Visual Studio 2008:** es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

**Microsoft .NET Framework:** es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

**Microsoft SQL Server:** Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.

**OracleDataAccess ODP.Net:** Oracle Data Access Components (ODAC) ofrece cuatro componentes que simplifican el desarrollo de .NET con la base de datos Oracle. Herramientas de desarrollo de Oracle para Visual Studio, Oracle Data Provider para .NET, proveedores de Oracle para ASP.NET y .NET en procedimientos almacenados. Estos proporcionan un amplio apoyo para las tareas básicas, tales como arrastrar y soltar datos en .NET para la generación de código de acceso y operaciones avanzadas, como integrada de PL / SQL con la depuración de aplicaciones .NET.

**ASP.NET:** Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

**VisualSVN:** es Subversion (SVN) del cliente, implementado como un add-in para Microsoft Visual Studio, que proporciona una interfaz para realizar las operaciones más comunes de revisión de control directamente desde dentro del IDE de Visual Studio. Se utiliza TortoiseSVN para ejecutar los comandos de Subversion.

**TortoiseSVN:** es un cliente Subversion, implementado como una extensión al shell de Windows. Es software libre liberado bajo la licencia GNU GPL.

**EPiServer Relate:** es un módulo add-on para EPiServer CMS que en combinación, le da una plataforma social completa para su website. Esta le permite convertir un sitio estático en una comunidad próspera.

**filter60:** Utilizado por el Buscador de EPiServer

**FilterPackx86:** Utilizado por el Buscador de EPiServer

**UML:** Lenguaje Unificado de Modelado (por sus siglas en inglés, Unified Modeling

Language) es el lenguaje de modelado de sistemas de software más conocido y

utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

**StarUML:** es una herramienta UML de código abierto, licenciado bajo una versión modificada de la licencia GNU GPL. StarUML soporta la mayoría de los tipos de diagramas especificados en UML 2.0. En la actualidad faltan los diagramas de objeto, paquete, el caso de uso y descripción de la interacción (aunque los dos primeros pueden ser asociados al modelo a través del editor de diagramas de clase).

**Open Office:** Editor de textos.

**Lotus Symphony:** Editor de textos.

**MICROSOFT EXPLORER 7:** Explorador web.

**MOZILLA FIREFOX 3.5:** Explorador web.

**GOOGLE CHROME:** Explorador web.

**FIREBUG:** es una extensión de Firefox creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga, estructura DOM), editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea e online.

**PSP:** El Personal Software Process (PSP) es un proceso de desarrollo de software estructurado que tiene como objetivo ayudar a los ingenieros de software a entender y mejorar su desempeño, mediante el uso de un "disciplinado, impulsado por el procedimiento datos".

**Process Dashboard:** es una iniciativa de código abierto para crear una PSP (SM) / TSP (SM), herramienta de apoyo.

**Microsoft Line of code counter:** es una métrica de software utilizado para medir el tamaño de un programa de software contando el número de líneas en

el texto del código fuente del programa. SLOC se suele utilizar para predecir la cantidad de esfuerzo que se requiere para desarrollar un programa, así como para estimar la productividad de programación o mantenimiento una vez que el software que se produce.

## **RESUMEN**

Con este documento busco principalmente evidenciar el proceso de prácticas realizado en la organización Pragma S.A, empresa de desarrollo de software en la cual he desarrollado la práctica empresarial.

Todo el proceso de desarrollo que se llevó a cabo para la ejecución de los proyectos para la líneas de negocio de familia/SCA, se planteó por medio de la metodología TSP/PSP. Este proceso tiene como objetivo tener un cronograma creíble y aceptado por los ingenieros de proyectos a cargo, se considera muy importante dentro del proceso de desarrollo TSP/PSP la excelente administración del tiempo de los ingenieros, para no tener contratiempos a la entrega del sistema.

## **ABSTRACT**

With this document I seek mainly show the process of practical work to organize Pragma SA, software development company in which I have developed business practice.

The whole process of development that took place for the implementation of projects for the family business lines / SCA, was raised by the methodology TSP / PSP. This process is designed to have a credible schedule and accepted by the project engineers in charge, is considered very important in the process of developing TSP / PSP excellent time management of engineers, to avoid setbacks to system delivery.

## INTRODUCCIÓN

Pragma S.A. es una empresa con 15 años de experiencia en la creación y desarrollo de soluciones de negocio basadas en Internet y medios relacionados, es una empresa de clase mundial, que compite a nivel nacional e internacional, con una serie de productos y servicios enfocados con la mejor tecnología , como Integración y automatización de procesos a través de la aplicación de los paradigmas de Service Oriented Architecture (SOA) y Business Process Management (BPM); Portales a través de CMS (Drupal), Apache, PHP, Flash; Publicidad y mercadeo web; Desarrollo de Software apalancada en la aplicación constante de los lineamientos de TSP™, el Proceso Unificado y el Project Management Institute.

Para el requerimiento tecnológico que solicita cada uno de los proyectos, Pragma utiliza todos los roles necesarios y conocimientos para garantizar a sus clientes la mejor calidad.

El trabajo a realizar en la compañía es el desarrollo de un proyecto de portales para el cliente Familia, en el cual se cumplirán ciertas funciones como análisis de diseño, desarrollo software basados en las tecnologías para calidad de software y enfoque de procesos PSP y TSP.

## 1. JUSTIFICACIÒN

La empresa PRAGMA S.A realiza proyectos y actividades que desarrollan los profesionales y practicantes, siendo este proceso muy provechoso para personas en formación profesional, con lo cual se mejora conocimientos tecnológicos y académicos, ayudando al desarrollo con calidad, brindando capacidades necesarias para resolver problemas y adaptarse al medio con gran facilidad y buen desempeño.

La empresa PRAGMA S.A enfoca sus productos en las siguientes líneas: integración y automatización de procesos, desarrollo de portales, publicidad y mercadeo web, con clientes que quieren ser los mejores y no se conforman en su medio. Por ello requieren de personas que aspiren a tener mejores conocimientos y habilidades para aplicarlos en sus líneas de negocio, personas competentes en lo técnico y lo humano, honestas, respetuosas con los clientes, compañeros y el entorno de trabajo, con sentido común para encontrar soluciones, prácticas que lleven a Pragma a ser cada vez mejor y a lograr sus objetivos.

El principal activo sobre el que está construido la compañía es el conocimiento, para ello utiliza el 10% del tiempo total de los empleados en enriquecimiento, para el avance técnico y tecnológico. De esta manera busca que el personal sea más competente, y estén relacionadas con las actividades del negocio.

Otro factor importante en el que tiene foco, es un entorno laboral satisfactorio, en el cual inciden dos factores, en primer lugar un clima laboral agradable en donde las relaciones de amistad y respeto sean mejores, y en segundo lugar proveer una infraestructura adecuada con un entorno idóneo para el aprendizaje, productividad, innovación, conocimiento, socialización.

Con nuestra participación activa en Pragma, la Corporación Universitaria Lasallista podrá establecer un vínculo empresarial, poniendo a disposición practicantes íntegros e idóneos, reconocidos por su excelencia ética, académica y laboral.

## **2. OBJETIVOS**

### **2.1 OBJETIVO GENERAL**

- ✓ Realizar portal web para las líneas de negocio de Familia/SCA en Cosas de Familia, Familia Institucional y Nosotras Online.

### **2.2 OBJETIVOS ESPECÍFICOS**

- ✓ Ampliar conocimiento en la Herramientas: ASP.Net, EPiServer.
- ✓ Implementar TSP en la organización.
- ✓ Desarrollar software de calidad.
- ✓ Utilizar las herramientas PSP/TSP.
- ✓ Tener cronograma al día con las tareas especificadas.

### **3. RESEÑA**

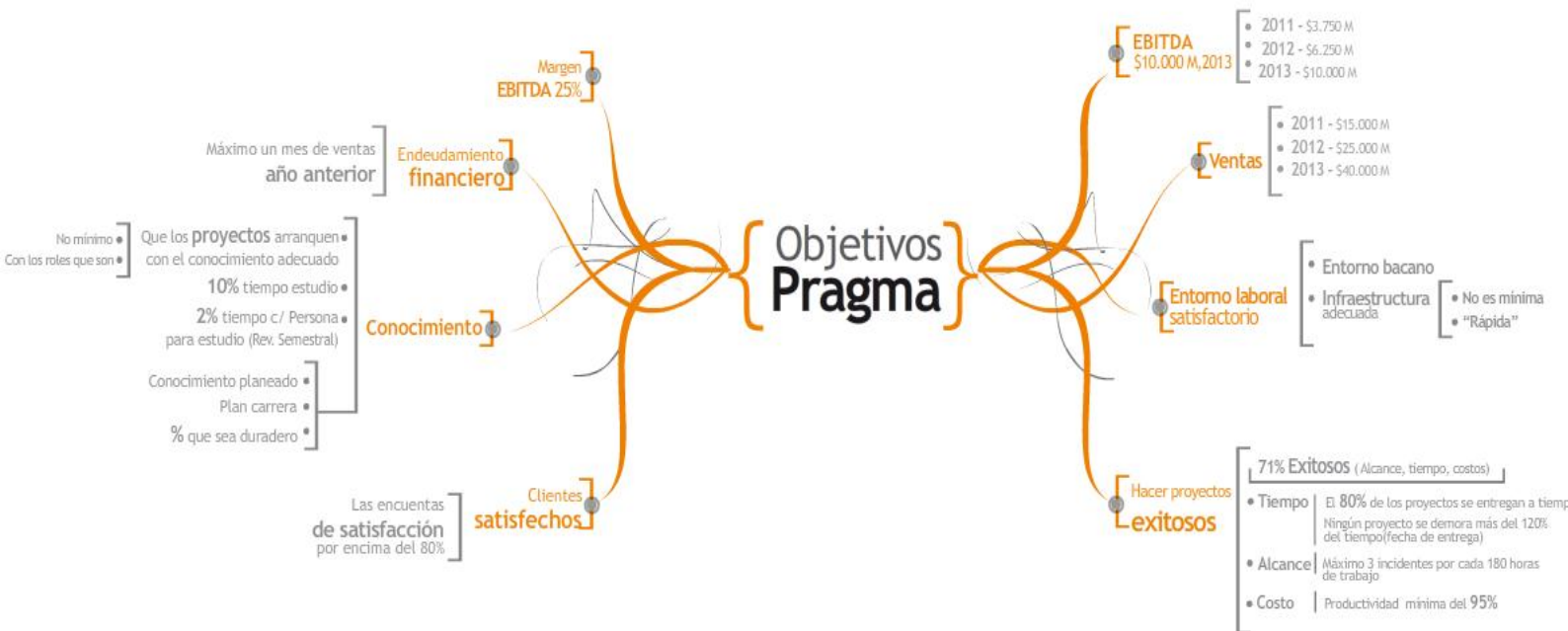
**NOMBRE:** Pragma S.A

#### **3.1 HISTORIA**

Es una organización fundada en 1996 que nació con el objetivo de asesorar a las grandes empresas colombianas en el uso y aprovechamiento de Internet y sus tecnologías relacionadas dentro sus procesos productivos y comerciales. Durante estos años, Pragma se ha consolidado como una de las principales empresas en la prestación de servicios relacionados con Desarrollo de Software de Aplicaciones de Misión Crítica, Marketing Digital y Comercio Electrónico, Automatización de Procesos y la Integración de Sistemas de Información. Actualmente, Pragma es el principal proveedor de soluciones y consultoría alrededor de la Gestión de Riesgo con su producto "Cero" (Control Estratégico del Riesgo), con el cual llega a los mercados de Perú y México.

### 3.2 OBJETIVOS PRAGMA

Figura 1: Objetivos



### 3.3 ALCANCE

Se ha desarrollado un sistema de gestión de calidad bajo los lineamientos establecidos por la norma ISO9001:2008, cuyo objetivo principal es:

Requisitos, Análisis y diseño, Construcción, Pruebas, Despliegue y gestión de proyectos de soluciones informáticas.

### **3.4 POLITICA DE CALIDAD**

La Política de Calidad de PRAGMA se fundamenta en:

- Somos una empresa de clase mundial: competimos aquí y en cualquier parte del mundo, con la mejor oferta de servicios.
- Nuestra empresa es de conocimiento, le garantizamos a nuestros clientes que siempre que comenzamos un proyecto, lo hacemos con todo el conocimiento requerido y todos los roles que el proyecto necesita.
- Garantizamos proyectos exitosos, más del 71% de nuestros proyectos se entregan en la fecha pactada, sin incidentes e invirtiendo sólo los recursos planeados desde el principio. Desagregados estos factores:
  - Más del 80% de nuestros proyectos se entregan a tiempo, ninguno de los que se entregan por fuera de las fechas pactadas supera el 20% de retraso.
  - Nuestros clientes no encuentran más de 3 incidentes por cada 180 horas de trabajo y nuestra productividad siempre supera el 95%.
  - Nuestros clientes siempre quedan muy satisfechos (calificados sobre el 80%).

### **3.5 VALORES**

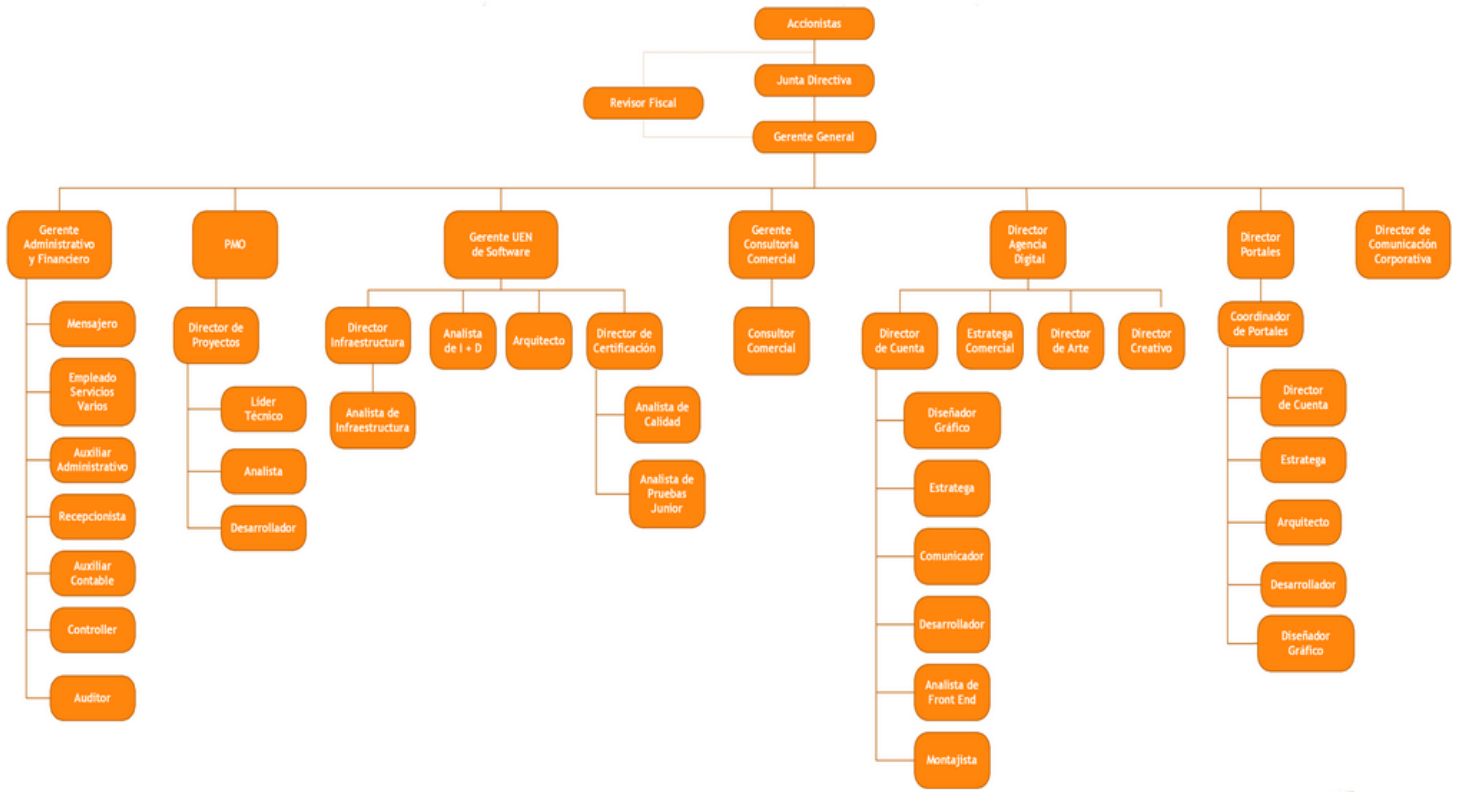
- ✓ Ser honesto
- ✓ Ser competente

**Figura 2: Mapa de procesos**

Proceso	Actividad	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
Planeación corporativa	Definir plan pragma		x		x								
	Planes por dirección				x								
	Presupuesto						x					x	x
Gestión comercial	Meta de ventas												x
Satisfacción cliente	Encuesta satisfacción cliente	x	x	x	x	x	x	x	x	x	x	x	x
Gestión oficina de proyectos PMO	Gestionar seguimiento facturación	x	x	x	x	x	x	x	x	x	x	x	x
	Gestionar productividad	x	x	x	x	x	x	x	x	x	x	x	x
Gestión infraestructura	Gestionar continuidad	x	x	x	x	x	x	x	x	x	x	x	x
Compras	Evaluación de proveedores		x										
Gestión documental	Control de obsoletos			x									
Salud ocupacional	Evaluación de panorama de factores de riesgo									x			
Gestión talento humano	Evaluación de clima laboral		x										
Gestión talento humano	Evaluación de desempeño			x	x								
Auditoría interna	Auditorías			x									
Revisión gerencial	Reunión revisión gerencial				x								
Medición corporativa	Indicadores	x	x	x	x	x	x	x	x	x	x	x	x
Mejora continua	Acciones	x	x	x	x	x	x	x	x	x	x	x	x

**Figura 3: Estructura organizacional**

CÓDIGO: D.DOC.007



### 3.6 PORTALES

Comunicación interactiva, asertiva y estratégica.

En la línea de Portales, se desarrolla óptimas estrategias orientadas y diseñadas para que en el corto, mediano y largo plazo, los clientes puedan alcanzar sus objetivos con un excelente retorno de inversión.

#### 3.6.1 Algunos de los clientes

Figura 4: Clientes



#### 3.6.2 Típicamente un portal apoya el cumplimiento de los siguientes objetivos

Ventas

Aumenta el interés y la intención de compra.

Posicionamiento

Integra acciones de interacción con los usuarios (Clientes potenciales).

Fidelización

Resalta la marca y su promesa de valor.

Comunicación

Genera una conexión emocional y/o racional.

Servicio al cliente

Genera consultas y/o transacciones.

Soporte de Operaciones

Ofrece atención directa y concisa a los usuarios, clientes y proveedores.

### **3.6.3 Objetivos de los portales**

Ventajas de optar por una línea especializada de portales.

Velocidad de desarrollo (Menor Time to Market).

Reducción de costos de desarrollo (reusó).

Mantenimiento a futuro al utilizar módulos estándar.

Disminución de costos en el ciclo de rediseño de portales. (se reúsa entre el 60% y el 70% del esfuerzo invertido).

Los nuevos proyectos siempre se basan en la experiencia de implementación de los anteriores (mayor calidad y funcionalidad).

Uso de plataformas estudiadas y aplicadas, y con soporte a nivel mundial.

### **3.6.4 Objetivo: Comunicación**

Portal enfocado a ofrecer información corporativa y de alto valor para los usuarios.

Busca integrar espacios de interacción y participación con los clientes, y abrir otros para escuchar y establecer una comunicación con los usuarios.

### **3.6.5 Objetivo: Posicionamiento**

Portales enfocados a aumentar las ventas o la intención de compra de los productos o servicios.

En este tipo de portales se busca exhibir los productos resaltando sus características, propiedades y precios.

Se incluyen funcionalidades como: catálogo de productos, salas de chat, galerías de videos e imágenes.

### **3.6.6 Objetivo: Fidelización**

Portal enfocado a ofrecer información y servicios de valor agregado para clientes de la compañía.

Se busca recrear y generar experiencias positivas para los usuarios.

Cuidar, escalar la relación con los clientes, integrar servicios y funcionalidades de interacción y participación directa con los usuarios, son otras de las características de este tipo de portal.

### **3.6.7 Objetivo: Ventas**

Portales enfocados a aumentar las ventas o la intención de compra de los productos o servicios.

En este tipo de portales se busca exhibir los productos resaltando sus características, propiedades y precios.

Se incluyen funcionalidades como: catálogo de productos, salas de chat, galerías de videos e imágenes.

### **3.6.8 Objetivo: Servicio al cliente**

Portal enfocado a ofrecer atención directa, oportuna, certera y concisa para los usuarios.

Se ocupa de integrar espacios de atención y participación con los clientes, y de generar movimientos, consultas y/o transacciones a través del portal.

### **3.6.9 Objetivo: Soporte de Operaciones**

Busca automatizar procesos y movimientos.

A través de un portal de este tipo es posible generar movimientos, consultas y/o transacciones.

Se caracteriza por ofrecer un alto valor agregado a los usuarios (facilidades, accesibilidad, rapidez, agilidad)

### **3.6.10 ¿Cómo apoya Pragma el logro de los objetivos de sus clientes a través de un Portal?**

#### **3.6.10.1 Metodología**

La metodología de Pragma está compuesta por 4 fases diseñadas y planteadas de tal manera que los proyectos puedan realizarse de forma: eficiente, programada, ajustada a los tiempos y expectativas de los clientes.

**Figura 5: Metodología**



### **3.6.10.2 Planeación**

Definición de roles y normas.

Cronogramas, en el que se establece un plan de fases, entregables al cliente y stepping stones o iteraciones (permite la construcción paso a paso del proyecto, permitiéndonos tener una retroalimentación temprana, oportuna y permanente de nuestros clientes.)

### **3.6.10.3 Definición Estratégica**

Llegamos a ella a través de:

- Brief.
- Entendimiento de la marca.
- Identificación del problema.
- Lluvia de ideas con grupo creativo.
- Validación con terceros.
- Definición de idea clave.

### **3.6.10.4 Implementación**

Consiste en llevar a cabo etapas como:

- Corte HTML optimizado para CMS teniendo en cuenta SEO.

- Arquitectura de software (módulos e integración) (diseño técnico.)
- Ejecución Plan de stepping stones teniendo en cuenta SEO. (implementación.)
- Desarrollo.
- Plan de pruebas.
- Pruebas y certificación.
- 

#### **3.6.10.5 Puesta en marcha**

Consiste en:

- Instalación.
- Capacitación.
- Cargue de contenido.
- Campaña de lanzamiento.
- Hosting.
- Soporte de estrategia.
- Fee – mantenimiento técnico.

#### **3.6.10.6 Beneficios de esa metodología**

Reducción de riesgos del proyecto.  
Involucramiento constante del cliente.  
Entrega oportuna.

### **3.7 LINEAS DE NEGOCIO**

#### **3.7.1 Fábrica de software**

Soluciones de Software a la Medida  
Aplicaciones de Misión Crítica  
Mejores Prácticas

Utilizando los marcos metodológicos del Project Management Institute (PMI), Unified Process, y de las mejores prácticas dictadas por el Team Software Process (TSP) y Personal Software Process (PSP), desarrollamos y evolucionamos soluciones de software con alta calidad y arquitectura.

### 3.7.2 SOA

Integración de Aplicaciones  
Reducción de Costos de Mantenimiento SW  
Optimización Flujo de Información

A través del paradigma de Service Oriented Architecture, facilitamos el flujo de información entre sistemas, en tiempo real, soportando la rápida toma de decisiones y la disminución de costos asociados a la manipulación manual de información.

### 3.7.3 BPM

Automatización de Procesos  
Optimización de Costos de Operación /  
Visibilidad y Control /Time to Market

Asesoramos e implementamos soluciones de negocio que permiten la automatización y mejoramiento continuo de los procesos de negocio, a través del aprovechamiento de la disciplina de Business Process Management y de las herramientas de IBM.

### 3.7.4 Algunos clientes para las líneas de negocio software, SOA y BPM

Figura 6: Algunos clientes Software, SOA Y BPM



### 3.7.5 Agencia Digital

Un equipo capaz de identificar las necesidades de comunicación y mercadeo de los clientes, construir su estrategia y realizar el despliegue táctico de las interacciones en entornos digitales.

Los servicios son Estratégicos y creativos

**Figura 7: Servicios Estratégicos y creativos**



### 3.7.6 CERO

Es una herramienta para apoyar el proceso de gestión de riesgos de manera estructurada y sistematizada. Es una solución altamente configurable, segura, flexible y de rápida implementación.

**Figura 8: Clientes CERO**



## **4. MARCO TEÓRICO**

### **4.1 ESTADO DEL ARTE**

En esta sección se mostrará como los diferentes portales se están rediseñando para las líneas de negocio de familia. Se dará a conocer que herramientas, procesos, desarrollos y personal se necesitan para la creación de un portal.

#### **4.1.1 Portales**

##### **4.1.1.1 Rediseño CDF**

Familia Sancela S.A, es una compañía que tuvo su comienzo en la ciudad de Medellín, importando papel higiénico desde los EEUU; con una visión de mercado y una misión clara: mejorar la vida de las personas.

Grupo Familia, es una compañía colombiana con capital sueco. Es una compañía líder en el diseño, la innovación, la producción y la distribución de productos de aseo personal en Colombia con una fuerte y determinante filosofía internacional que se ve reflejada en las operaciones internacionales que se realizan en diversos países alrededor del mundo donde sus marcas generan directamente confianza, estabilidad y compromiso con sus consumidores.

Familia Sancela S.A., con sus diferentes líneas, marcas y variedad de productos llega a más de 20 países alrededor del mundo con quienes ha logrado importantes nexos comerciales.

##### **4.1.1.1.1. Situación a Mejorar**

Desde el nacimiento de las estrategias Web (lo que se conoce ahora como Web 1.0) hasta hoy CDF y PRAGMA han aprendido y recorrido el camino más conveniente para la marca y ha pasado de simplemente publicar información y mostrar a establecer comunicaciones de 2 vías, reforzar y estrechar lazos con los usuarios, generar valor agregado, crear elementos diferenciadores y llegar a especializarse en el segmento del mercado, conociendo y relacionándose con los usuarios a un nivel personal e individual.

Se propone un rediseño del sitio que busca evolucionar la comunicación actual en el vía (la marca le habla a los usuarios) y pasar a disponer un canal de comunicación en 2 vías e incluso permitir el relacionamiento entre los usuarios.

##### **4.1.1.1.2. Objetivos**

Nueva Imagen:Renovar el sitio Web de Cosas de Familia con un look más moderno, más atractivo y que ofrezca a su público mejores contenidos.

Fortalecer el vínculo emocional entre la marca y su público, posibilitando la construcción de comunidad a través de herramientas que permitan la interacción entre ellos mismos. Creando conocimientos colectivos.

Ampliar la base de datos de usuarios con la inclusión participativa de otros segmentos de público hasta ahora no tenidos en cuenta, y poder así diseñar estrategias futuras de mercadeo a un público más amplio.

Alinear la página Web de acuerdo con las nuevas tendencias de marketing online y tecnología disponibles, para evolucionar en el medio y destacarse de la competencia.

#### **4.1.1.1.3. Dimensionamiento de recursos**

Estratega: 1 persona  
Diseñador: 1 persona  
Arquitecto: 1 persona  
Ingeniero de CMS: 2 persona  
Ingeniero .NET: 1 persona  
Analista de Pruebas: 1 persona

#### **4.1.1.2 Rediseño NOL**

##### **4.1.1.2.1. Situación a Mejorar**

El sitio actual de NosotrasOnLine fue rediseñado en el 2005, aunque el sitio siempre ha estado a la vanguardia, se vio en la necesidad de ofrecerle a su público objetivo que son las niñas entre los 13 y los 18 años un avance tecnológico que les proporcione herramientas de interacción y en las tendencias de comunicación online actuales.

##### **4.1.1.2.2. Objetivos**

Incrementar la interactividad del sitio, mejorar la experiencia del usuario y fortalecer la relación con el público.

##### **4.1.1.2.3. Dimensionamiento de recursos**

Estratega: 1 persona  
Diseñador: 1 persona  
Arquitecto: 1 persona  
Ingeniero de CMS: 2 persona  
Ingeniero .NET: 1 persona  
Analista de Pruebas: 1 persona

## 4.2 .NET

La plataforma .NET de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows.

.NET Framework se incluye en Windows Server 2008, Windows Vista y Windows 7. De igual manera, la versión actual de dicho componente puede ser instalada en Windows XP, y en la familia de sistemas operativos Windows Server 2003. Una versión "reducida" de .NET Framework está disponible para la plataforma Windows Mobile, incluyendo teléfonos inteligentes.

La norma (incluido en ECMA-335, ISO/IEC 23271) que define el conjunto de funciones que debe implementar la biblioteca de clases base (BCL por sus siglas en inglés, tal vez el más importante de los componentes de la plataforma), define un conjunto funcional mínimo que debe implementarse para que el marco de trabajo sea soportado por un sistema operativo. Aunque Microsoft implementó esta norma para su sistema operativo Windows, la publicación de la norma abre la posibilidad de que sea implementada para cualquier otro sistema operativo existente o futuro, permitiendo que las aplicaciones corran sobre la plataforma independientemente del sistema operativo para el cual haya sido implementada. El Proyecto Mono emprendido por Ximian pretende realizar la implementación de la norma para varios sistemas operativos adicionales bajo el marco del código abierto.

### 4.2.1 Componentes

Los principales componentes del marco de trabajo son:

El conjunto de lenguajes de programación.

La biblioteca de clases base o BCL.

El entorno común de ejecución para lenguajes, o CLR por sus siglas en inglés.

Debido a la publicación de la norma para la infraestructura común de lenguajes (CLI por sus siglas en inglés), el desarrollo de lenguajes se facilita, por lo que el marco de trabajo .NET soporta ya más de 20 lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos, lo que elimina las diferencias que existían entre lo que era posible hacer con uno u otro lenguaje.

Algunos de los lenguajes desarrollados para el marco de trabajo .NET son: C#, Visual Basic .NET, Delphi (Object Pascal), C++, F#, J#, Perl, Python, Fortran, Prolog (existen al menos dos implementaciones, el P#1 y el Prolog.NET2 ), Cobol y PowerBuilder.

[editar]Common Language Runtime

## 4.2.2 Common Language Runtime

Estructura interna del entorno de ejecución en lenguaje común.

El CLR es el verdadero núcleo del framework de .NET, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo (W2k y W2003). Permite integrar proyectos en distintos lenguajes soportados por la plataforma .Net, como C++, Visual Basic, C#, entre otros.

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio, el CIL (Common Intermediate Language) antes conocido como MSIL (Microsoft Intermediate Language), similar al BYTECODE de Java. Para generarlo, el compilador se basa en la especificación CLS (Common Language Specification) que determina las reglas necesarias para crear el código MSIL compatible con el CLR.

Para ejecutarse se necesita un segundo paso, un compilador JIT (Just-In-Time) es el que genera el código máquina real que se ejecuta en la plataforma del cliente. De esta forma se consigue con .NET independencia de la plataforma de hardware. La compilación JIT la realiza el CLR a medida que el programa invoca métodos. El código ejecutable obtenido se almacena en la memoria caché del ordenador, siendo recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente.

## 4.2.3 Características

Es el encargado de proveer lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta. Los servicios son variados:

Cargador de clases: permite cargar en memoria las clases.

Compilador MSIL a nativo: transforma código intermedio de alto nivel independiente del hardware que lo ejecuta a código de máquina propio del dispositivo que lo ejecuta.

Administrador de código: coordina toda la operación de los distintos subsistemas del Common Language Runtime.

Recolector de basura: elimina de memoria objetos no utilizados automáticamente.

Motor de seguridad: administra la seguridad del código que se ejecuta.

Motor de depuración: permite hacer un seguimiento de la ejecución del código aún cuando se utilicen lenguajes distintos.

Verificador de tipos: controla que las variables de la aplicación usen el área de memoria que tienen asignado.

Administrador de excepciones: maneja los errores que se producen durante la ejecución del código.

Soporte de multiproceso (hilos): permite desarrollar aplicaciones que ejecuten código en forma paralela.

Empaquetador de COM: coordina la comunicación con los componentes COM para que puedan ser usados por el .NET Framework.  
Biblioteca de Clases Base que incluye soporte para muchas funcionalidades comunes en las aplicaciones.

### **4.3 Microsoft SQL Server**

Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.

#### **4.3.1 Características de Microsoft SQL Server**

Soporte de transacciones.

Escalabilidad, estabilidad y seguridad.

Soporta procedimientos almacenados.

Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.

Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa la base de datos (Microsoft SQL Server), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

En el manejo de SQL mediante líneas de comando se utiliza el SQLCMD

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos

#### **4.3.2 Programación**

##### **4.3.2.1 Transact-SQL**

Es el principal medio Server. Expone claves para las operaciones realizarse en SQL Server, incluyendo creación y modificación de esquemas de la base de datos, introducir y editar datos en la base de datos, así como datos o administren el servidor, aprovechan la funcionalidad de SQL Server mediante el

envío de consultas de T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente. SQL Server que sean administrados mediante T-SQL.

#### **4.3.2.2 Cliente Nativo de SQL**

Cliente Nativo de SQL es la biblioteca de acceso a datos para los clientes de Microsoft SQL Server versión 2005 en adelante. Implementa nativamente soporte para las características de SQL Server, incluyendo la ejecución de la secuencia de datos tabular, soporte para bases de datos en espejo de SQL Server, soporte completo para todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asincrónicas, las notificaciones de consulta, soporte para cifrado, así como recibir varios conjuntos de resultados en una sola sesión de base de datos. Cliente Nativo de SQL se utiliza como extensión de SQL Server plug-ins para otras tecnologías de acceso de datos, incluyendo ADO u OLE DB. Cliente Nativo de SQL puede también usarse directamente, pasando por alto las capas de acceso de datos.

## **4.4 ORACLE DATABASE**

Oracle es un sistema de gestión de base de datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando: Soporte de transacciones, estabilidad, escalabilidad y Soporte multiplataforma.

Su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

Para desarrollar en Oracle utilizamos PL/SQL un lenguaje de 5ª generación, bastante potente para tratar y gestionar la base de datos, también por norma general se suele utilizar SQL al crear un formulario.

## **4.5 ASP.NET**

Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Cualquier persona que está familiarizada con el desarrollo de aplicaciones web sabrá que el desarrollo web no es una tarea simple. Ya que mientras que un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo, la programación web es una mezcla de varios lenguajes de etiquetas, un gran uso de lenguajes de script y plataformas de servidor. Por desgracia para el programador de nivel intermedio, el conocimiento y habilidades que se necesitan para desarrollar aplicaciones web tienen muy poco en común con las que son necesarias en el desarrollo tradicional de aplicaciones.

### **4.5.1 Características**

#### **4.5.1.1 Páginas**

Las páginas de ASP.NET, conocidas oficialmente como "web forms" (formularios web), son el principal medio de construcción para el desarrollo de aplicaciones web. Los formularios web están contenidos en archivos con una extensión ASPX; en jerga de programación, estos archivos típicamente contienen etiquetas HTML o XHTML estático, y también etiquetas definiendo

Controles Web que se procesan del lado del servidor y Controles de Usuario donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web. Adicionalmente, el código dinámico que se ejecuta en el servidor puede ser colocado en una página dentro de un bloque `<% -- código dinámico -- %>` que es muy similar a otras tecnologías de desarrollo como PHP, JSP y ASP, pero esta práctica es, generalmente, desaconsejada excepto para propósitos de enlace de datos pues requiere más llamadas cuando se genera la página.

ASP.NET sólo funciona sobre el servidor de Microsoft IIS, lo que supone una desventaja respecto a otros lenguajes del lado de servidor, ejecutables sobre otros servidores más populares como Apache. Ejemplos de esto son PHP, Perl o Python.

## 4.6 EPISERVER CMS

EPiServer CMS da todo lo que se necesita para construir una presencia en línea que atrae a los clientes - todo lo que necesita y fácil de usar, basado en plataforma web, lo que proporciona acceso en cualquier momento y en cualquier lugar.

los sitios web juegan un papel importante en el marketing en línea. Se debe apoyar a la marca, atraer a los visitantes específicos y convertir las oportunidades en customers. Su presencia en línea es la primera interacción entre marcas y consumidores. Su contenido debe ser relevante, atractivo y coherente para construir la confianza y el retorno de la inversión en línea. También tiene que ser fácil de usar - sin depender de el desarrollador.

EPiServer CMS no requiere ningún conocimiento de programación para crear, administrar y publicar nuevos contenidos. EPiServer CMS es más simple la publicación de contenido.

WYSIWYG EPiServer del entorno de edición, las capacidades de la página de vista previa, estructura de árbol para la organización del contenido y alcance de las normas de gestión de flujo de trabajo significa que cada miembro de su equipo de marketing puede crear fácil y rápidamente nuevos contenidos, revisar y tomar en vivo sin la ayuda de un desarrollador.

## 4.7 FIREBUG

Firebug es una extensión de Firefox creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga, estructura DOM), editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea e inline.

Firebug no es un simple inspector como DOM Inspector, además edita y permite guardar los cambios, un paso por delante del conocido Web Developer. Su atractiva e intuitiva interfaz, con solapas específicas para el análisis de cada tipo de elemento (consola, HTML, CSS, Script, DOM y red), permite al usuario un manejo fácil y rápido. Firebug está encapsulado en forma de plug-in o complemento de Mozilla, es Open Source, libre y de distribución gratuita. Con Firebug es posible editar el código HTML de una página y comprobar el resultado al instante, sin tener que recargar.

Esa misma acción puede realizarse sobre las hojas de cascada en estilo (CSS), sin tener que editar el fichero que las contiene. Los cambios también se verán en vivo.

Firebug incluye una pequeña regla que te ayudará en las tareas de diseño, para que no yerres ningún pixel.

Otras características interesantes de FireBug son:

- Supervisar la actividad de red.
- Depurar el código Javascript.
- Explorar el DOM (Document Object Model).
- Y mucho más.

## **5. DISEÑO METODOLOGICO**

### **5.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE**

#### **5.1.1 Documentar la toma de requerimientos**

A la reunión de toma de requisitos con el cliente, se debe hacer énfasis primero, en lo que él quiera mantener del sitio actual, en caso de ser un rediseño, tales como secciones, subsecciones y herramientas, porque así el encargado del proyecto por parte de PRAGMA S.A. podrá tener un punto para comenzar con la distribución de elementos.

Si no es un rediseño y el portal saldrá por primera vez después de esa toma de requerimientos con el cliente, se le debe preguntar por las secciones y subsecciones que debe tener el sitio web, para luego seguir con las herramientas.

Si es posible, se le debe preguntar al cliente acerca de cómo se imagina cada una de esas secciones y subsecciones, para empezar a visualizar las plantillas que compondrán el sistema, del mismo modo que con las herramientas.

También se debe hacer lo mismo con los elementos que aparecerán en el home, que mientras el cliente va mencionando el contenido del sitio, se le debe preguntar cuáles de esos contenidos o servicios deben tener un espacio en el home.

Cada una de las plantillas debe tener los siguientes elementos en el documento:

##### **5.1.1.1 Nombre de la plantilla**

Será el nombre que identifica la plantilla y su uso, es decir, si es una plantilla que muestra sólo contenido, podría llamarse Plantilla Contenido, o si muestra un formulario, llamarse Plantilla Formulario. Así suene evidente, el nombre debe hacer alusión al contenido de esa plantilla, por lo que este nombre no está tampoco estandarizado.

##### **5.1.1.2 Codificación de la plantilla**

Será el código que se le asignará a cada una de las plantillas que compongan el portal. Este código puede dárselo quien esté haciendo el documento, o el ingeniero que intervendrá en la implementación de las plantillas. El código será del tipo: PTL\_01NOMBRE DE LA PLANTILLA, donde PTL es un genérico que indica que es "plantilla", 01 indica que es la primera plantilla planteada en el

sistema y el NOMBRE DE LA PLANTILLA, es el mismo nombre que se le da a la plantilla en el punto anteriormente explicado.

#### **5.1.1.3 Esquema de la plantilla**

Será un esquema, realizado en Visio preferiblemente, donde se esquematicen los elementos que deben tener cada una de las plantillas que complementarán el sistema.

#### **5.1.1.4 Componentes de la plantilla**

Se trata de una tabla con dos columnas, Elemento y Descripción, en las cuales se deben listar cada uno de los elementos que aparecen en ese esquema de la plantilla, con el mismo nombre que se le haya puesto en el esquema a cada elemento. En la columna de descripción se debe detallar el alcance de ese elemento, lo que hace, para qué sirve, si es obligatorio en la plantilla, su extensión, el número de caracteres, etc. Toda aquella información que ayude a delimitar cada uno de esos elementos.

#### **5.1.1.5 Descripción de la plantilla**

Desde aquí se puede contar para qué sirve esta plantilla, o qué tipo de contenidos se mostrarán en ella.

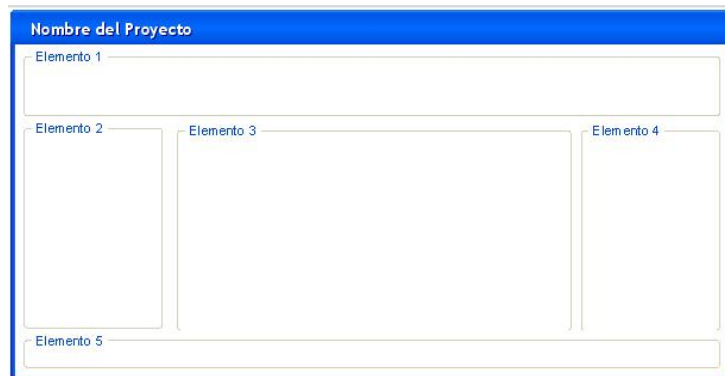
#### **5.1.1.6 Secciones para las que aplica esta plantilla**

Será una tabla de dos columnas, donde se listarán las secciones y las subsecciones en las que debería aparecer cada una de las plantillas usadas en el portal.

La primera plantilla que se debe contemplar en la construcción de este documento, es aquella que sea común a todo el sitio, es decir, la metaplantilla, teniendo en cuenta los pasos anteriormente mencionados.

A continuación se presenta un esquema que describe la manera en que se debe especificar una plantilla:

**Figura 9: Plantillas**



Es importante aclarar que los elementos que se ilustran en la plantilla deben corresponder con un nombre del elemento, por ejemplo Cabezote, Menú.

Algunas de estas plantillas, que no son las únicas que podrían intervenir en la construcción de un portal, sino que son las más comunes, son:

#### **5.1.1.7 Plantilla de Contenido**

Es la plantilla que permite mostrar contenidos, acompañados de una o varias fotografías alusivas al tema, o con un espacio para un histórico que muestre los últimos XX artículos anteriores por ejemplo. Es una plantilla que permite mostrar contenidos principalmente, con títulos, entradillas, subtítulos, entretítulos, etc.

#### **5.1.1.8 Plantilla Tabla**

Es una plantilla que permite mostrar una tabla, con un número determinado de columnas y de filas, con unos datos específicos de interés para el usuario final.

#### **5.1.1.9 Plantilla Gráficos o Ilustraciones**

Será una plantilla que permitirá mostrar una gráfica, una imagen, o un video, es decir, que a través de la administración se permita subir al sistema un archivo de este tipo, que se podría o no mostrar con algo de contenido que lo acompañe.

#### **5.1.2 Documentar las Herramientas**

Se trata de todas aquellas funcionalidades que acompañan el menú principal de navegación del sitio, que le dan un valor agregado al portal y que complementan la navegación. Generalmente son aquellos enlaces que se

encuentran por fuera del menú principal de navegación, pero que se mantienen a la vista durante toda la navegación del usuario.

Algunas herramientas comunes que hacen parte de un portal son:

#### **5.1.2.1 Contáctenos**

Es un formulario a través del cual los usuarios, sin necesidad de estar registrados en el sitio, pueden enviar sus comentarios a la empresa propietaria del portal. Éstos pueden ser almacenados en base de datos o llegarles a un correo predeterminado dentro de la Compañía, que será el encargado de darle una respuesta.

#### **5.1.2.2 Suscríbete**

Es un formulario, que puede o no estar ligado al registro de un sitio, a través del cual se le pregunta al usuario si desea suscribirse al contenido que ofrece el portal, que se cambia con cierta periodicidad para que sea llamativo. Este suscríbete puede permitir que el usuario seleccione a qué secciones de información quiere suscribirse, o de manera global, aceptar recibir información en su mail.

Esta suscripción está ligada necesariamente a un newsletter, que es el mail que le llegará al usuario con el contenido al que se suscribió, generalmente con un diseño predeterminado, al que únicamente le cambia el subject, el contenido, las fotos y la fecha de publicación de la información.

En caso de que el suscríbete no esté relacionado con el registro del sitio, se deberán solicitar ciertos datos básicos del usuario, como su nombre y su mail, para poder hacerle el envío periódico de la información.

#### **5.1.2.3 Buscador**

Es una herramienta a través de la cual, se puede ingresar una palabra clave en un campo de texto, y encontrar contenidos dentro del portal que estén relacionadas con ella.

Esta búsqueda también podrá ser avanzada, donde se le permita al usuario incluir más información, tal como fechas de publicación, áreas de la compañía dentro de las que desea buscar, tipos de archivos, etc., en resumen, es una herramienta que le permite al usuario segmentar más su búsqueda.

#### **5.1.2.4 Registro / Login**

Es una herramienta a través de la cual, se le solicita al usuario una cierta cantidad de datos personales, para almacenarlos en base de datos, y generalmente, es a esos usuarios registrados con los que se crea comunidad, y

a los que se dirigen las comunicaciones especiales de la marca o producto protagonista del portal.

Una vez el usuario se encuentre registrado en el sitio, deberá ver un nuevo acceso llamado "login", a través del cual el usuario registrado pueda ingresar una clave y un nombre de usuario, para poder acceder a los servicios o secciones especiales que se crearon para él.

### **5.1.3 Documentar los desarrollos específicos**

Para el caso de los desarrollos específicos, se debe tener en cuenta su magnitud dentro del proyecto, para poder saber si harán parte del documento que contiene todas las especificaciones de la construcción del portal, o conformarán un documento independiente.

Los desarrollos específicos son aquellos que tienen en cuenta casos de uso, o que su desarrollo implique un módulo fundamental en la construcción y lanzamiento del portal, tales como módulos de consulta de saldos o estados de transacciones, consultorios, integraciones con aplicativos ya existentes, presentación de información traída desde bases de datos, entre otros desarrollos.

## **5.2 ADMINISTRACIÓN DE LAS COMUNICACIONES**

### **5.2.1 Participantes en el proyecto**

A continuación se describen los roles y responsables generales, estos se pueden omitir o se deben adicionar otros dependiendo de las necesidades de cada proyecto.

#### **5.2.1.1 Director del proyecto**

Este rol debe comprender plenamente los objetivos del proyecto y saber cómo alcanzarlos. El Director del proyecto es el responsable de la planeación del proyecto, de conseguir los recursos necesarios para ejecutar el plan de trabajo y de monitorear y reportar el progreso del proyecto contra dicho plan. Es el responsable del cumplimiento del alcance pactado. Algunas tareas que debe realizar el Director del proyecto, son:

Establecer y mantener el Plan de Calidad, los estándares del proyecto y sus procedimientos.

Establecer y mantener el Plan de Trabajo y el Plan Financiero.

Procurar la minimización de los riesgos del proyecto.

Desarrollar funciones de comunicación y coordinación dentro de la organización del proyecto.

Preparar los reportes de avance del proyecto.  
Generar información del estado del proyecto a partir de los registros del mismo.

#### **5.2.1.2 Líder Técnico**

Es la persona responsable de todas las decisiones técnicas que se deben tomar durante el proyecto por parte de Pragma. Entre sus funciones principales están:

Realizar la revisión al Análisis de Requisitos y Diseño del Sistema.

Asesorar a los Ingenieros de Pragma en cuestiones técnicas (en caso de tener un equipo de trabajo).

Definir la tecnología a utilizar. (Cuando aplique).

Es el directo responsable de los temas técnicos globales al desarrollo.

Auditar el trabajo técnico en general.

Planear la integración de subsistemas.

#### **5.2.1.3 Ingeniero de Proyecto**

Es el responsable de implementar el sistema y llevarlo a un nivel funcional. Dentro de sus principales funciones están:

Implementar clases.

Arreglar defectos.

Hacer pruebas por módulos.

Integrar subsistemas.

Instalar el sistema en el ambiente definido. (Cuando aplique).

Detectar cambios al alcance en el proyecto.

#### **5.2.1.4 Analista de Calidad**

Es la persona del área de calidad en Pragma S.A. y se encarga de verificar que el producto final entregado al cliente esté dentro de los estándares de desarrollo.

Dentro de sus principales funciones están:

Planear las pruebas funcionales que debe realizar el área de calidad.

Realizar las pruebas programadas para cada proyecto.

Entregar un informe de las incidencias generadas a partir de las pruebas.

Crear los proyectos en la herramienta Scarab, para el reporte de incidencias por parte del Cliente

#### **5.2.1.5 Diseñador Gráfico**

Es el responsable de diseñar la interfaz del sistema desde el punto de vista gráfico. (Parte del sistema que tiene interacción con el usuario). Dentro de sus principales funciones están:

Definir la presentación del sistema y/o aplicación.

Respetar la imagen del cliente y proponer un lenguaje gráfico acorde con la naturaleza del negocio.

Procurar mantener el balance entre tecnología y comunicación visual, junto con el ingeniero de proyecto.

## **5.3 TSP\PSP**

### **5.3.1 Metodología TSP / PSP**

TSP/PSP, es un proceso de desarrollo para ingenieros de software, que permiten ser más acertivos en la entrega de productos de Software de Alta Calidad.

El Team Software Process (TSP) atiende las principales preocupaciones del negocio:

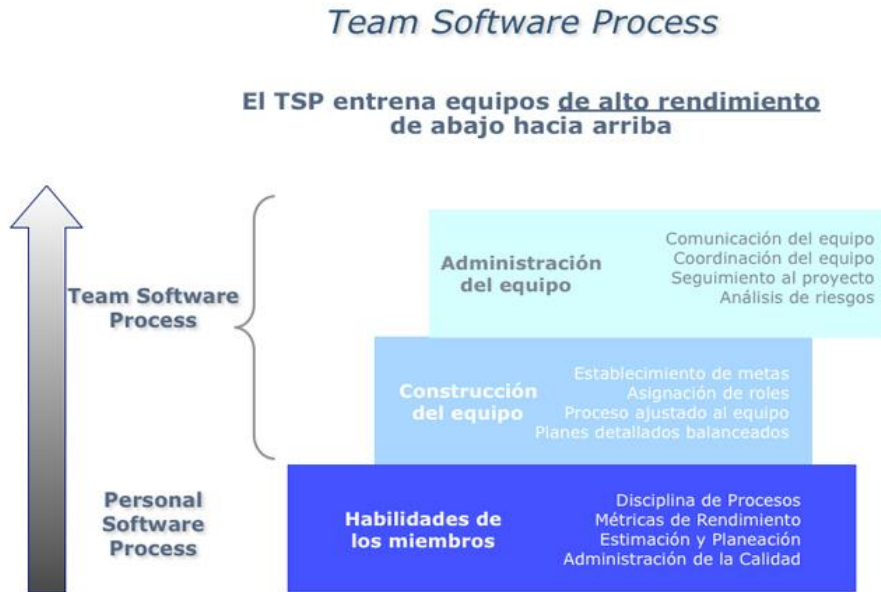
- Predecibilidad en costo y cronograma
- Productividad y tiempo del ciclo de desarrollo
- Mejora de calidad en los productos

El Personal Software Process (PSP) es un proceso diseñado para los individuos que usan y aplican tareas personales estructuradas. Con PSP los desarrolladores definen y miden sus procesos personales para desarrollo de software. En el entrenamiento de PSP aprenden a registrar datos de tamaño, tiempo y defectos.

### **5.3.2 Estos datos los utilizan para:**

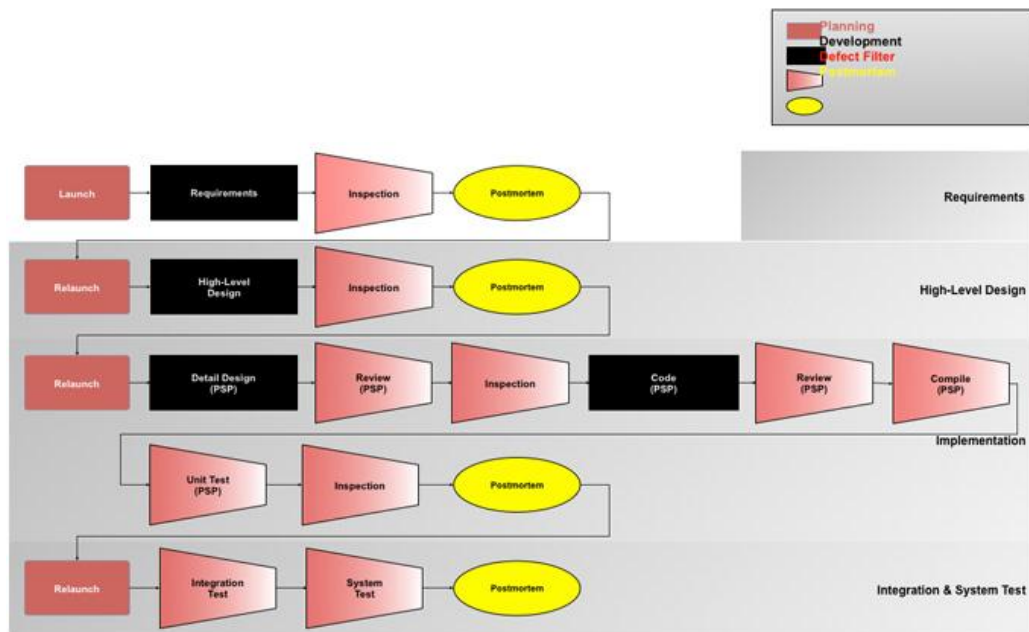
- ✓ Planear y dar seguimiento a su trabajo
- ✓ Administrar la calidad de los productos que generan
- ✓ Medir y mejorar su desempeño

**Figura 10: Team Software Process**



En el diagrama abajo se pueden observar las actividades principales en la ejecución de un proyecto utilizando la Metodología TSP/PSP.

**Figura 11: Metodología TSP/PSP**



El PSP es un modelo desarrollado por Watts Huphrey en el Software Engineering Institute, con el fin de desarrollar competencias específicas en los desarrolladores de software que les permita implementar las mejores prácticas

en el desarrollo de software y por consecuencia puedan realizar mejores estimaciones y generar productos de software de alta calidad).

El PSP establece la base para integrar las competencias de desarrollo en un equipo mediante el TSP.

El TSP, mediante un coach certificado, coordina actividades que llevan a establecer un plan balanceado, considerando actividades del modelo PSP, incluyendo revisiones e inspecciones de productos de trabajo.

El PSP/TSP establece prácticas de nivel 5 para el proceso de desarrollo de software y si bien no contempla todas las áreas propuestas por CMMi, incluye los procesos fundamentales para asegurar que se realice adecuadamente el proceso de desarrollo de software.

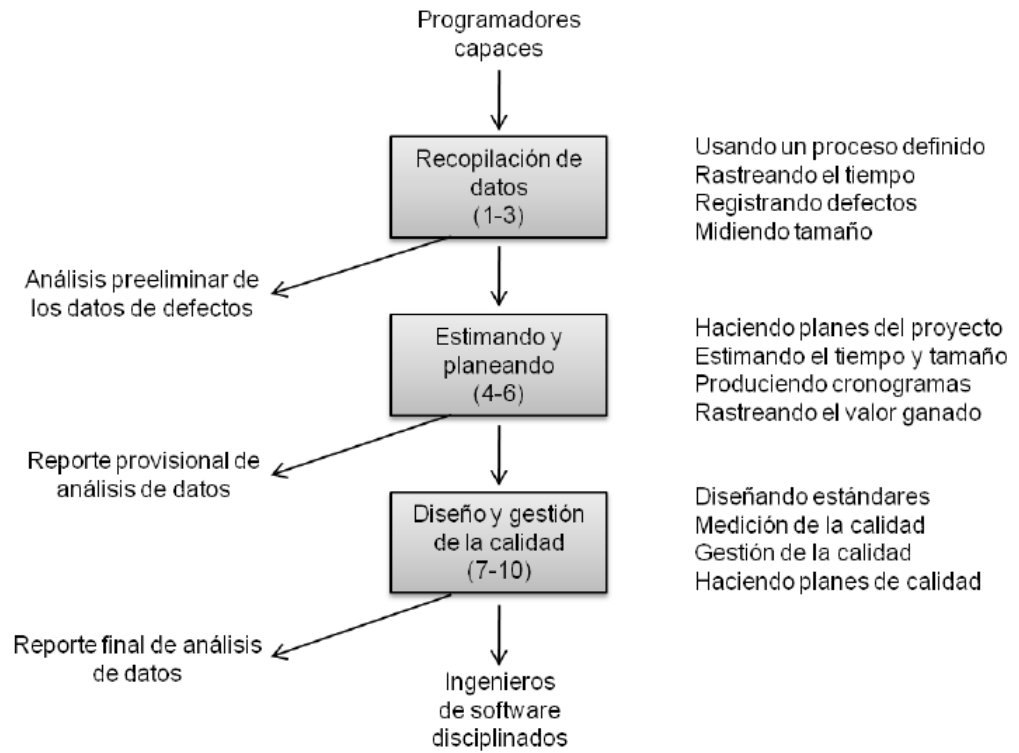
Ejecutando la Metodología TSP/PSP en los desarrollos de software las estimaciones tienen menos desviaciones y los errores que observa el usuario final se minimiza. Mejora el desempeño tanto de equipos como individuos, es disciplinado, ágil proporcionando beneficios inmediatos y medibles acelerando las iniciativas de mejora de procesos en las organizaciones.

Los principios básicos del curso PSP se muestran en la figura 5. Debido a que el propósito de este curso es convencer a los ingenieros de la utilidad del PSP, ellos deben usarlo para escribir programas y reunir datos. Para estar realmente convencidos, sin embargo, ellos deben escribir los suficientes programas para aprender los métodos y reunir la suficiente información para mostrar que su trabajo ha mejorado. Esto requiere al menos nueve o diez programas de ejercicio.

Antes de tomar el curso de PSP, los ingenieros deben ser programadores capaces. Entonces usan el PSP para planear cada programa, usan un proceso definido para escribirlo, y reúnen información precisa. Ellos aprenden a planear, usar prácticas de diseño firmes, y a medir y gestionar la calidad. Además ven por sus propios datos que los métodos disciplinados de PSP les ayudan a hacer un mejor trabajo.

Los resultados del curso de PSP de 298 ingenieros experimentados han sido publicados por el IIS [4]. Los ejemplos se muestran en las figuras 6, 7, y 8. En estas figuras, el punto 1 a la izquierda muestra los datos para el programa 1 al principio del curso. Ya que los ingenieros aún no habían aprendido el PSP, este primer punto representa su desempeño con sus prácticas de software actuales. El punto 10 a la derecha muestra sus datos cuando estaban usando el PSP. La diferencia es la mejoría originada por el PSP.

**Figura 12: Curso de PSP**



**Figura 13: Tiempos de estimacion de errores**

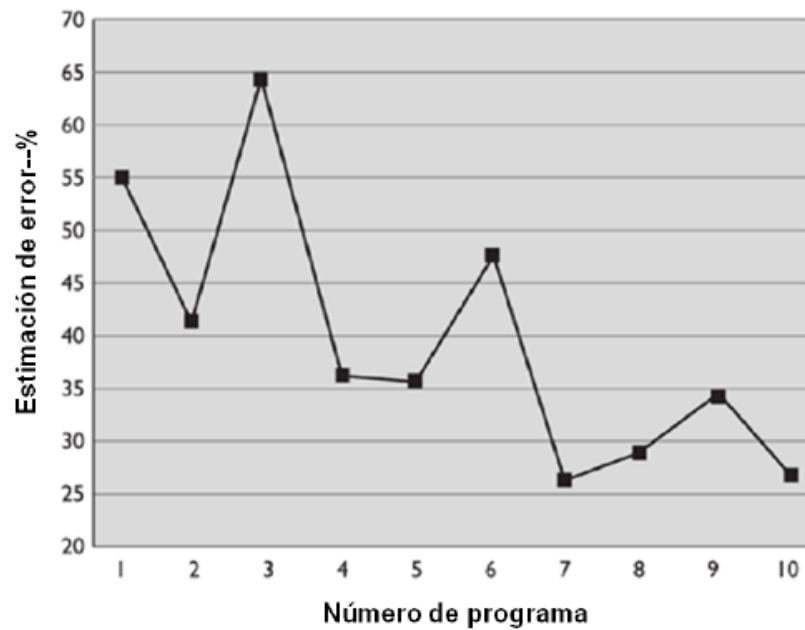


Figura 14: Defectos eliminados en compilación y pruebas

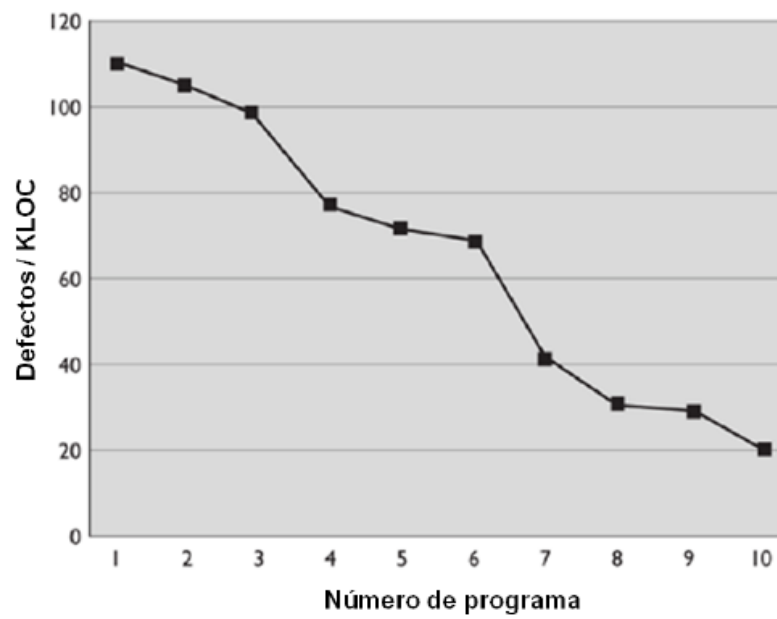
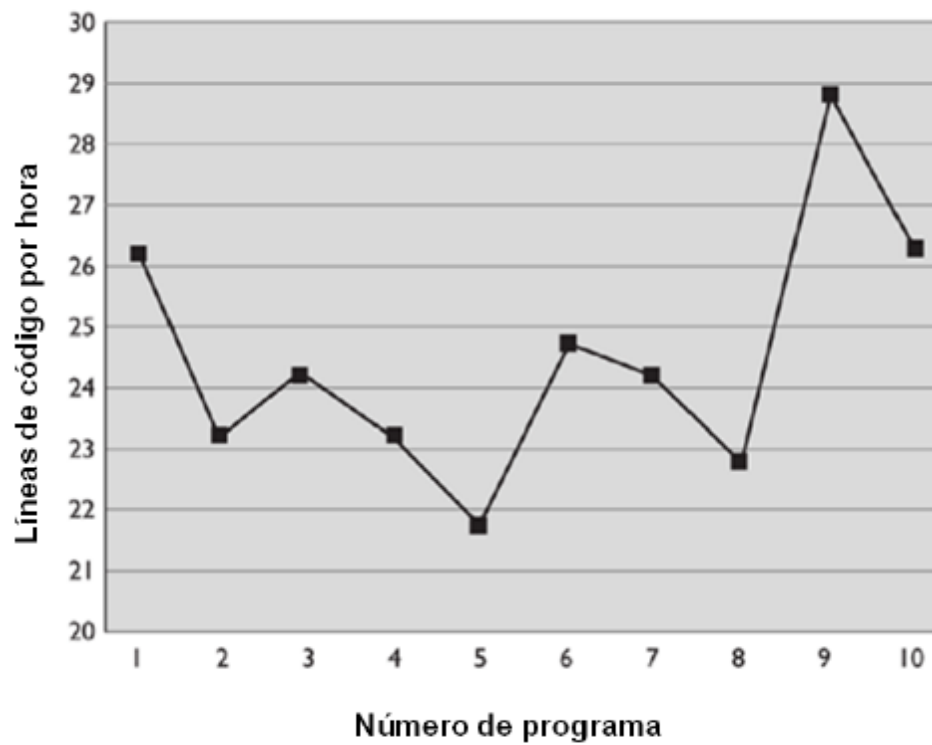


Figura 15: Productividad



La figura 6 muestra como mejora la precisión de la estimación. El punto 1 en la izquierda muestra la precisión de la estimación para el programa 1. Debido a que los ingenieros aún no han aprendido los métodos de estimación y no tienen datos históricos sobre su trabajo, este primer estimado es una suposición. El punto 10 a la izquierda muestra la precisión de la estimación para el programa 10, el programa final del curso. Aquí los ingenieros habían aprendido como usar sus datos históricos para hacer estimados estadísticamente sólidos para el tamaño del programa y el tiempo de desarrollo. Ellos también habían aprendido como estimar el rango probable de errores en estos estimados.

Como queda claro a partir de la figura 6, la precisión de los estimados de los ingenieros mejora hasta el último programa. Esto implica que la precisión de los estimados mejorará incluso más allá, siempre y cuando los ingenieros continúen usando estos métodos y continúen recolectando y usando sus datos personales.

La figura 7 muestra los defectos que estos 298 ingenieros encontraron durante la compilación y pruebas en 10 programas. Los ingenieros de software típicamente escriben programas rápidamente y luego se enfocan en la calidad durante la compilación y pruebas. Con las prácticas típicas de hoy en día, los ingenieros pasan cerca de la mitad de su tiempo compilando y probando. Incluso entonces, los programas resultantes son defectuosos. Con el PSP, los ingenieros se concentran en la calidad desde el comienzo, y encuentran y reparan la mayoría de los errores antes de la compilación. Ellos se sorprenden de ver que al usar el PSP, ellos reducen los defectos de compilación y pruebas en cinco o más veces y reducen el tiempo en estos procesos en tres o más veces.

Para la mayoría de los ingenieros, los resultados en la figura 8 son sorprendentes. Ellos ven por sus datos personales que cuando usan el PSP para hacer planes y seguimiento de su tiempo, tamaño, y datos de defectos, su productividad no se disminuye. De hecho, para muchos, la productividad incrementa. En promedio, los ingenieros son tan productivos personalmente cuando usan métodos disciplinados como lo son hoy. Como veremos más tarde, incluso cuando la productividad personal de los ingenieros no cambia, cuando todos hacen planes personales y miden y gestionan la calidad, el desempeño de su equipo mejora dramáticamente.

Si bien más de 90% de los ingenieros que toman el curso de PSP están convencidos de su valor; unos cuantos no. La mayoría de ellos son como una joven mujer que me dijo, —Yo no creía que el PSP funcionaría, así que no lo usé. De forma ilógica, ella argumentó luego que de acuerdo a sus datos personales los métodos del PSP no eran efectivos. Algunos ingenieros están tan convencidos de que sus métodos actuales son adecuados que ellos no ensayarán nada nuevo. Algunos finalmente verán los beneficios de los métodos disciplinados, particularmente si sus colegas los usan, pero unos cuantos no lo

harán. Ellos tienden a dejar la organización cuando la gerencia insiste en que usen en su trabajo el PSP.

A continuación se describe el guión del proceso de levantamiento de Requisitos.

**Tabla 1. Proceso de levantamiento de Requisitos**

<b>Propósito</b>	Guiar la captura de requisitos
<b>Entradas</b>	<ul style="list-style-type: none"> <li>- Mapa del sitio aprobado por el cliente</li> <li>- Diseño gráfico aprobado por el cliente</li> <li>- Plantilla de documento de requisitos</li> <li>- Lista de verificación de requisitos</li> <li>- Tener un conocimiento general del sistema a desarrollar: Alcance a nivel macro, objetivos del negocio (Cliente) y de la gerencia de Pragma.</li> <li>- Hoja de bitácora de actividades de TSP.</li> </ul>

<b>Paso</b>	<b>Actividad</b>	<b>Descripción</b>
1	Inicie el registro de tiempos en la bitácora	Registrar en la bitácora de actividades el inicio de la actividad.
2	Establecer nombre del proyecto en la plantilla	Modificar el Nombre del Proyecto y Cliente en los lugares de la plantilla marcados como [PROYECTO-CLIENTE]
3	Completar información de control de cambios	Describir las razones de modificación del documento, esto cuando se están haciendo cambios, cuando es la primera vez que se ingresa al documento se debe colocar "Versión inicial" en la descripción
4	Incluir mapa del sitio	Adicionar a la plantilla de funcionalidades el mapa del sitio aprobado.
5	Escribir las funcionalidades.	Escribir las funcionalidades llenando los campos especificados en las plantilla, el detalle de las funcionalidades lo debe determinar la persona responsable por parte del cliente o el responsable de Pragma en el caso en el cual se presta un servicio completo (Pragma define toda la estrategia y funcionalidades).
6	Describir los pendientes general	Llenar los pendientes estableciendo responsable y fecha de seguimiento.
7	Establecer exploradores y software prerrequisito	Establecer los exploradores para los cuales el sitio es certificado por Pragma, no es suficiente con tener el nombre se debe determinar la versión y revisión. (Mozilla Firefox 6.0, Internet

		Explorer 8.0.7) Establecer el software necesario para que el sitio web funcione correctamente: Sistema operativo, Servidor Web, Base de datos, Componentes adicionales.
8	Definir los atributos de calidad	Definir atributos como: seguridad, rendimiento, disponibilidad, modificabilidad, escalabilidad
8	Crear los casos de prueba de sistema	Seguir el proceso "Creación de casos de prueba de sistema para TSP" para definir las pruebas a realizar sobre las funcionalidades.
9	Revisión Personal	Utilizando la lista de verificación se evalúa con respecto a los puntos de la lista y los objetivos generales.
10	Finalizar registro de tiempo	Finalizar el registro de tiempos en la bitácora

<b>Salidas</b>	<ul style="list-style-type: none"> <li>- Documento de funcionalidades</li> <li>- Registro de tiempos para la actividad</li> <li>- Plan De pruebas de sistema.</li> </ul>
----------------	--

## 5.4 ESTANDAR CODIFICACION C#

### 5.4.1 Nombres

Los archivos aspx, es decir, de las páginas dinámicas de .NET deben cumplir con los siguientes estándares:

- Debido a que son páginas web, su contenido es html por lo tanto debe cumplir con los estándares de html.
- Los nombres de las páginas deben corresponder con el estándar de nombre de las clases, pues tienen siempre asociada una clase.
- También incluyen etiquetas propias de aspx las cuales deben cumplir con el mismo estándar xml del documento.

Entre las etiquetas aspx se encuentran las de los controles, las cuales siempre tienen un atributo ID donde se asigna el nombre de la etiqueta y que corresponde con una propiedad asociada a la página aspx para permitir el acceso al control por código, dicho nombre debe cumplir con el mismo estándar de nombrado de las propiedades.

Se recomienda como buena práctica, aunque no necesaria, colocar al principio del nombre de los controles un prefijo de 2 o 3 letras para identificar el tipo de control, por ejemplo:

**Tabla 2. Estándares de codificación C#**

Control	Prefix	Control	Prefix
Label	Lbl	HiddenField	hdf
TextBox	Txt	Literal	lit
DataGrid	dtg	UpdatePanel	upl
Button	btn	RequiredValidator	rqv
ImageButton	imb	RegularExpressionValidator	rev
Hyperlink	Hlk	GridView	grv
DropDownList	ddl	MenuList (Episerver)	mnl

ListBox	Lst	ListPages (Episerver)	ltp
DataList	dtl	LinkButton	Ink
Repeater	rep,rpt		
Checkbox	chk		
CheckBoxList	cbl		
RadioButton	rbl		
RadioButtonList	rbl		
Image	img		
Panel	pnl		
PlaceHolder	phd		
Table	tbl		
Validators	val		

## 6. CONCLUSIONES

Con la realización de las prácticas profesionales en la organización Pragma S.A, se ha adquirido nuevo conocimiento acerca de como se hacen los portales para apoyar las lineas de negocio de una empresa y todo el proceso de desarrollo que debe llevar este sin importar que tan grande sea el proyecto. Además la utilización del conocimiento adquirido en la Universidad, así como las habilidades y aptitudes es demasiado importante para el ciclo de desarrollo de un software o portal.

De esta manera se pueden abrir las ideas ante un mundo organizacional y como en una organización se necesitan miles de personas para realizar una tarea. Cada una con conocimientos diferentes y lo importante es que el grupo de trabajo sea unido y tenga las mejores motivaciones para realizar las tareas..

La empresa Pragma S.A busca ser reconocida en el ambito mundial, por ello busca personas con ganas de adquirir y dar conocimiento. Es un placer hacer parte de un equipo con tantas capacidades y quisiera agradecer a por permitirme formar parte de el. Es muy importante para mi estar involucrado en lo proyectos porque me dieron la oportunidad de demostrar mi trabajo y proponer ideas.

El trabajo realizado ilustra la forma en como se debe realizar un portal web con todas sus fases, pasando desde levantamiento de requerimientos hasta las pruebas. Es muy importante que no se salte ninguno de los pasos y mas con la metodologia de trabajo TSP\PSP, ya que esta exige que todos sus pasos se leven uno a uno con la adquisicion de todos los datos como defectos, tiempos, calendario, cambios, etc.

## BIBLIOGRAFIA

WIKIPEDIA Microsoft . Net [en línea]

[[http://es.wikipedia.org/wiki/Microsoft\\_.NET](http://es.wikipedia.org/wiki/Microsoft_.NET)] [Citado el 16 feb 2012]

WIKIPEDIA Asp.Net [en línea]

[<http://es.wikipedia.org/wiki/ASP.NET>] [Citado el 25 feb 2012]

WIKIPEDIA. Mantis [en línea]

[[http://es.wikipedia.org/wiki/Mantis\\_Bug\\_Tracker](http://es.wikipedia.org/wiki/Mantis_Bug_Tracker)] [Citado el 15 de Marzo del 2011]

WIKIPEDIA. Firebug [en línea]

[<http://es.wikipedia.org/wiki/Firebug>] [Citado el 15 de Marzo del 2011]

Pragma S.A Metodologia de Desarrollo de software [en intranet]

[[https://docs.google.com/a/pragma.com.co/document/d/1nKAzoEyAvVCHcpB\\_VD7qZv55PDkSZCrtLUTioMWrQxA/edit](https://docs.google.com/a/pragma.com.co/document/d/1nKAzoEyAvVCHcpB_VD7qZv55PDkSZCrtLUTioMWrQxA/edit)] [Citado el 30 de Enero del 2012]

Pragma S.A Accesos control fuentes

[[https://docs.google.com/a/pragma.com.co/document/d/1OhS8keegXv\\_pg0yEoPcDONNbCPP9sM150eRmo9tA5tw/edit](https://docs.google.com/a/pragma.com.co/document/d/1OhS8keegXv_pg0yEoPcDONNbCPP9sM150eRmo9tA5tw/edit)] [Citado el 30 de Enero del 2012]

Pragma S.A Acuerdos en servicios entre areas

[<https://docs.google.com/a/pragma.com.co/spreadsheet/ccc?key=0AjFQhNHIJLHfdE5EdG9VbDRIZm9COFFUWTgtbU5GQXc#gid=0>] [Citado el 30 de Enero del 2012]

Pragma S.A Portales

[<https://docs.google.com/a/pragma.com.co/present/view?id=0AQVXmG0ObpDDZGM0cXM4el8yMjVmcHg2NmRncQ>] [Citado el 3 de Febrero de 2012]

Pragma S.A Historia pragma [<http://calidad.pragma.com.co/>] [Citado el 2010]

Watts S. Humphre, Winning of software. En : Boston, 2002 Pearson Education, inc

## ANEXOS

### Anexo A. Detalle de las funcionalidades

A continuación se detalla cada una de las funcionalidades para [PROYECTO-CLIENTE].

NOMBRE DEL MÓDULO		
Nombre de la funcionalidad		
Entradas/Condiciones		
Descripción del funcionamiento		
Entrada de datos		
Nombre del campo	Tipo	Tamaño
Condiciones especiales		
Salidas esperadas		

IMAGEN (Vista previa)	
Listado de contenido necesario (Imágenes, Videos, Texto, Documentos)	
Tipo contenido, descripción de contenido	Características: Imagen (ancho, alto) Video (tipo, tamaño)

## **Anexo B. Restricciones técnicas**

### **Software**

Para la construcción del Sitio se utiliza EPiServer CMS 6

*Pre - requisitos:*

Servidor Web:

Internet Information Services (IIS) 6.0

.Net Framework 3.5

Windows Server 2003 Enterprise R2

Servidor de Base de datos:

SQL Server 2005 SP2

### **Exploradores para los cuales se certifica el sitio**

Internet Explorer 8.0.1

Mozilla Firefox: 5.9.4

### Anexo C. Atributos de calidad

Tipo
Rendimiento – Carga de Home

Descripción
El home debería mostrarse completamente en 4 segundos en una máquina con ancho de banda de 2mbps en un computador con como mínimo 2 Gbs de RAM y procesador core 2 duo E7200 o afines, esto en las horas donde el tráfico es medio, acorde a estadísticas esto se presenta entre las 8am y 5pm. El Servidor web donde se aloja el sitio deberá tener 8Gbs de memoria y un procesador, no deberán existir otros sitios en el mismo servidor.

Medición
Cargar en 4 segundos

## Anexo D. Listado de pendientes

Descripción

Responsable

Fecha de seguimiento

## Anexo E. Checklist diseño de alto nivel

<b>Propósito</b>	Conducir a una efectiva inspección de diseño.
<b>General</b>	Revisar cada uno de los elementos de los cuales se compone el diseño. Para completar cada paso de la revisión, se deben marcar las casillas de la derecha.

<b>Defect Log #</b>	<b>Categoría</b>	<b>Descripción</b>	
	Se cubren todos los requisitos establecidos para el sistema.	Todos los subsistemas identificados son parte de los requisitos. Son claras las relaciones de interfaces entre los subsistemas.	
	Nombre de los Componentes	Los nombres son descriptivos y únicos dentro del diseño. Cumple con el formato establecido.	
	Descripción de los Componentes	Todos los componentes tienen su respectiva descripción. La descripción y comentarios son claros.	
	Gráficos	Las figuras son acordes al estándar de diseño. Los conectores son acordes al estándar de diseño. Las regiones son acordes al estándar de diseño. Los comentarios o anexos son acordes al estándar de diseño.	
	Relación tecnológica	Que los componentes y sus funciones estén acordes con la tecnología usada (EpiServer, Drupal, ASP, etc.).	
	Rendimiento	Es adecuado el acceso a datos según los requisitos.	
	Estándares a Seguir	Se incluyen los estándares necesarios para el desarrollo del sistema (SEO, WebServices).	
	Pruebas	Verificar que se hayan registrado las pruebas del sistema en el ambiente de PRAGMA.	

## Anexo F. Checklist inspeccion DLD

<b>Propósito</b>	Dar los lineamientos generales para dar inspección del diseño detallado de cada una de las tareas a ser desarrolladas.
------------------	--

Defect Log #	Categoría	Descripción
	Compleitud	<p>Verificar que en el diseño se cubran los requisitos de la tarea.</p> <ul style="list-style-type: none"> <li>- Se están produciendo los datos requeridos.</li> <li>- Se proporcionan todos los datos de entrada necesarios.</li> <li>- Se creó con base al diseño de alto nivel.</li> </ul>
	Limites Externos	<ul style="list-style-type: none"> <li>- Se contempla el funcionamiento de la tarea en los valores óptimos y externos.</li> <li>- Que no haga más de lo que tenga que hacer.</li> </ul>
	Lógica	<ul style="list-style-type: none"> <li>- Verificar que la secuencia del diseño es apropiada.</li> <li>- Verificar que en las condiciones se cubran todos los casos.</li> </ul>
	Diagramas	<p>Verificar que estén Completos:</p> <ul style="list-style-type: none"> <li>- Cumplen con el estándar UML.</li> <li>- Estén detallados para poder realizar su codificación.</li> </ul> <p>Verificar que:</p> <ul style="list-style-type: none"> <li>- Nombres claros y únicos.</li> <li>- Los tipos de las variables y los parámetros son definidos apropiadamente.</li> </ul>
	Casos Especiales	<ul style="list-style-type: none"> <li>- Verificar que el diseño cumpla con todos los casos especiales.</li> <li>- Controlar todas las condiciones incorrectas o errores.</li> </ul>
	Uso Funcional	<ul style="list-style-type: none"> <li>- Verificar que todos los métodos y procedimientos son completamente entendibles y apropiadamente usados.</li> </ul>
	Consideraciones del Sistema	<ul style="list-style-type: none"> <li>- Verifique las condiciones de seguridad</li> </ul>

		conforme a las especificaciones.	
	Casos de Prueba	<ul style="list-style-type: none"> <li>- Verificar que existan casos de prueba, que evalúen los requisitos de la tarea.</li> <li>- Verificar que la documentación de los casos cumpla con el estándar de PRAGMA.</li> </ul>	

## Anexo G. Checklist inspección de codificación

<b>Propósito</b>	Dar los lineamientos generales para dar inspección de la codificación de cada una de las tareas desarrolladas.
------------------	--

Defect Log #	Categoría	Descripción
	Compleitud	<ul style="list-style-type: none"> <li>- El código cubre todo el diseño elaborado.</li> <li>- El código implementa más de lo que requiere el diseño.</li> <li>- Existe algún requerimiento que no haya sido tenido en cuenta en la implementación.</li> <li>- Ubicación de archivos en las carpetas adecuadas.</li> </ul>
	Inicialización	<ul style="list-style-type: none"> <li>- Verificar que en la carga de las paginas este el IsPostBack adecuado.</li> <li>- Cada clase se un archivo y no incluya mas clases dentro del mismo.</li> </ul>
	Llamadas	<ul style="list-style-type: none"> <li>- Chequear el formato de la llamada al método y los parámetros.</li> </ul>
	Nombres	<ul style="list-style-type: none"> <li>- Se está utilizando el Pascal Casing (PC) y Camel Casing (CC) adecuadamente para el nombramiento de variables y miembros (CC), clases (PC), métodos (PC), constantes (PC), propiedades (PC) y parámetros (CC).</li> </ul>
	Variables y Miembros	<ul style="list-style-type: none"> <li>- Validación de String, Session, Propiedades con el método IsNullOrEmpty.</li> <li>- Se están utilizando los tipos adecuados para las magnitudes de datos esperados.</li> <li>- Verificar el uso de valores fijos o quemados dentro del código.</li> </ul>
	Lógica	<ul style="list-style-type: none"> <li>- Output correcto de los métodos, con tipo, formato y fuente de información.</li> <li>- Verificar la lógica booleana de las comparaciones (todos los operadores lógicos).</li> <li>- Verificar que un método solo realiza una función específica, si no es así verificar su pertinencia.</li> <li>- Se cumple el diagrama de secuencia</li> </ul>

		<p>diseñado.</p> <ul style="list-style-type: none"> <li>- No actualizar un miembro en cada método, este debe ser cambiado como parámetro del mismo.</li> </ul>	
	Excepciones	<ul style="list-style-type: none"> <li>- Manejo adecuado de excepciones.</li> <li>- Manejo adecuado de excepciones en acceso a base de datos y se incluye finally para cerrar la conexión.</li> </ul>	
	Ciclos	<ul style="list-style-type: none"> <li>- Verificar cada uno de los ciclos para que tenga inicialización y finalización efectiva.</li> <li>- Verificar que cada uno de los ciclos contemple todos los flujos posibles.</li> </ul>	
	Comentarios	<ul style="list-style-type: none"> <li>- Verificar que se comenten las funciones, clases y variables poco descriptivas.</li> <li>- Verificar header (autor y fecha).</li> <li>- Verificar coherencia del comentario.</li> </ul>	
	Vista	<ul style="list-style-type: none"> <li>- Verificar que el código de javascript cumpla con la inspección anterior.</li> <li>- Verificar formato del código ASPX o HTML.</li> <li>- Nombre adecuado de los controles.</li> <li>- Verificar uso de los textos con translate.</li> </ul>	
	Estándar	<ul style="list-style-type: none"> <li>- Verificar que cumpla con los requisitos del estándar.</li> <li>- Verificar estándar externos (SEO, Market View).</li> </ul>	