

**USO DE TECNOLOGÍAS Y METODOLOGÍAS DE DESARROLLO MANEJADOS  
EN PRAGMA S.A, PARA LA CONSTRUCCIÓN DE PORTALES WEB**

**ROY STEEVEN YARCE DAVID**

**INFORME DE PRÁCTICA EMPRESARIAL**

**ASESOR  
MAURICIO BEDOYA LONDOÑO  
INGENIERO DE SISTEMAS**

**CORPORACIÓN UNIVERSITARIA LASALLISTA  
FACULTAD DE INGENIERÍAS  
INGENIERÍA INFORMÁTICA  
CALDAS, ANTIOQUIA  
2012**

## **DEDICATORIA**

A mis padres, mi madre Marta Inés David y mi padre Alfonso Yarce, quienes me apoyaron y estuvieron presentes en toda mi etapa de formación profesional.

A la Familia Pérez Calle conformada por: Marta Calle, Ramiro Pérez y Felipe Pérez, quienes me acogieron como un miembro más de su familia y apoyaron en todo momento.

A Sara Soto Castrillón, quien me apoyó incondicionalmente y estuvo conmigo en mi proceso de formación profesional.

A los profesores de la Corporación Universitaria Lasallista por aportarme de su conocimiento y experiencia al desarrollo de mi formación profesional.

## CONTENIDO

<b>1</b>	<b>OBJETIVOS</b> .....	<b>11</b>
1.1	OBJETIVO GENERAL .....	11
1.2	OBJETIVOS ESPECIFICOS.....	11
<b>2</b>	<b>JUSTIFICACIÓN</b> .....	<b>12</b>
<b>3</b>	<b>MARCO TEÓRICO</b> .....	<b>13</b>
3.1	<b>MICROSOFT VISUAL STUDIO</b> .....	<b>13</b>
3.2	MICROSOFT VISUAL STUDIO 2008 .....	13
3.3	SQL SERVER 2008 R2 .....	14
3.4	EPISERVER CMS .....	15
3.4.1	Arquitectura.....	15
3.5	ORACLE .....	15
3.6	STARTUML.....	16
3.7	PSP.....	17
3.7.1	Proceso PSP.....	18
3.8	TSP.....	20
3.8.1	La estructura del PSP .....	21
3.8.2	Lanzamiento TPS.....	22
3.9	PROCESS DASHBOARD.....	23
<b>4</b>	<b>METODOLOGÍA</b> .....	<b>25</b>
4.1	CAPACITACIÓN PSP FUNDAMENTALS.....	25
4.2	INICIO DE LOS PROYECTOS PILOTO .....	26
4.2.1	Lanzamiento del Proyecto.....	26

4.2.2	Ejecución del Proyecto.....	34
4.2.2.1	Requisitos .....	34
4.2.2.2	Diseño de alto nivel.....	35
4.2.2.3	Construcción .....	35
4.2.2.4	Pruebas.....	38
4.2.2.5	Despliegue y Post Mortem .....	39
4.2.2.6	Junta Semanal .....	40
<b>5</b>	<b>CONCLUSIONES .....</b>	<b>43</b>
<b>6</b>	<b>RECOMENDACIONES .....</b>	<b>44</b>
	<b>BIBLIOGRAFÍA .....</b>	<b>45</b>

## LISTA DE FIGURAS

<b>Figura 1. Proceso PSP .....</b>	<b>19</b>
<b>Figura 2. Métodos de mejora de proceso .....</b>	<b>21</b>
<b>Figura 3. Construcción de un equipo TSP.....</b>	<b>22</b>
<b>Figura 4. Proceso de Lanzamiento TSP.....</b>	<b>23</b>
<b>Figura 5. Creación de miembros del equipo en Process Dashboard.....</b>	<b>27</b>
<b>Figura 6. Creación de proceso de desarrollo en Process Dashboard.....</b>	<b>28</b>
<b>Figura 7. Elaboración descendente del plan en Process Dashboard.....</b>	<b>32</b>
<b>Figura 8. Balance de tareas en Process Dashboard.....</b>	<b>32</b>
<b>Figura 9. Cronograma de actividades personal – Process Dashboard.....</b>	<b>33</b>
<b>Figura 10. Cronograma de actividades personalizado – Process Dashboard. .....</b>	<b>34</b>
<b>Figura 11. Diagrama de Clases en StartUML.....</b>	<b>36</b>
<b>Figura 12. Diagrama de Secuencia en StartUML.....</b>	<b>37</b>
<b>Figura 13. Proceso Junta Semanal TSP.....</b>	<b>41</b>

## GLOSARIO

**FRAMEWORK:** En programación, framework es una estructura conceptual y tecnológica de soporte definido, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Se trata de una colección de librerías de software que proporciona una interfaz de programación de aplicaciones (API).

**ESCALABILIDAD:** Es la propiedad que tiene un sistema, red o proceso, para extender el margen de operaciones sin perder calidad, manejar el crecimiento continuo de una manera fluida o hacerse más grande.

**DDL:** Es un Lenguaje de definición de datos (por sus siglas en ingles, *Data Definition Language*), que permite a los usuarios de un sistema de base de datos la definición de las estructuras que almacenaran los datos, así como procedimientos o funciones que permitan consultarlos.

**DML** Es un Lenguaje de Manipulación de Datos (por sus siglas en ingles *Data Manipulation Language*), permite a los usuarios de un sistema de base de datos llevar a cabo las tareas de consulta o manipulación de datos.

**WIN32:** Win32 significa "Windows 32 bits". En otras palabras, hace referencia a todas las plataformas de 32 bits del sistema operativo Windows: Windows NT, Windows 95, Windows 98, Windows CE.

**DEFECTO:** En PSP un defecto es todo lo que implica cambio en las diferentes etapas del desarrollo.

**INTRANET:** Es una red de computadores privada que utiliza el internet para compartir dentro de una empresa parte de sus sistemas de información y sistemas operacionales.

**EXTRANET:** Es una red privada que utiliza protocolo de internet, protocolos de comunicación y probablemente infraestructura pública de comunicación para compartir de forma segura parte la información u operación propia de una empresa, proveedores, compradores, socios, clientes o cualquier otro negocio u organización.

**PROBE:** Por sus siglas en ingles PROxy – Based Estimating, es un proceso de estimación usado en PSP para estimar tamaño y esfuerzo.

**COACH:** Es la persona entrenada para dirigir los equipos TSP en los lanzamientos y relanzamientos, reconocer y tratar eficazmente los problemas más comunes que se producen en TSP y transmitir los principios de TSP en la organización.

**CMMI:** Integración de modelos de madurez de capacidades (por sus siglas en inglés, *Capability maturity model integration*) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

## RESUMEN

El presente trabajo ilustra el proceso que se llevo a cabo durante 6 meses de práctica empresarial en la empresa PRAGMA incluyendo las metodologías y herramientas informáticas utilizadas para el proceso de desarrollo de los diferentes proyectos de software en los que se trabajó.

PRAGMA es una empresa con 15 años de experiencia en la creación y desarrollo de soluciones de negocio basadas en Internet y medios relacionados. Por lo cual se contó durante el proceso de aprendizaje y práctica empresarial con herramientas informáticas como: ASP.NET, C#, SQL SERVER, EPISERVER CMS, y las metodologías TSP (Team Software Process) - PSP (Personal Software Process), utilizados y aplicados para la creación de dichas soluciones.

Se participó activamente en todo el ciclo de desarrollo de software utilizado en PRAGMA específicamente en las etapas de Levantamiento de requisitos, Análisis y Diseño, Construcción, Pruebas e Integración. Además de ser parte del equipo piloto de la compañía en la implementación de las metodologías antes mencionadas: PSP – TSP.

Como especialistas en soluciones de negocio basadas en internet se contó con proyectos de software destinados para facilitar y promover la comunicación y el contacto de las entidades corporativas con su público objetivo y su grupo de trabajo.

## **ABSTRACT**

This paper illustrates the process that took place during 6 months of business practice in the company PRAGMA SA including the methodologies and tools used for the process of development of different software projects I worked on.

PRAGMA SA is a company with 15 years of experience in the creation and development of business solutions based on Internet and related media for this reason, I use during the learning process and business practice tools such as: ASP.NET, C #, SQL SERVER, EPiServer CMS, and methodologies PSP (Personal software Process)-TSP (Team software Process), used and applied for the creation of software projects.

I participated actively in the entire software development cycle used in PRAGMA SA specifically in the early stages of lifting requirements, construction and testing.

PRAGMA SA is specialized in internet-based business, it had software projects to facilitate and promote communication and contact corporate entities with their target audience and its working group.

## INTRODUCCIÓN

El presente trabajo pretende exponer el proceso de aprendizaje y las actividades realizadas durante el ciclo de desarrollo de los proyectos en los cuales fui partícipe en PRAGMA.

Se ilustra además los pasos esenciales para la implementación de la metodología PSP-TSP, con la cual se pretende llevar a la organización a niveles más altos de madurez CMMI y convertirse en una empresa de clase mundial, proyecto en el que se participo activamente como parte del equipo piloto, encargado de implementar esta metodología por primera vez en la compañía. Desarrollando así actividades como: Levantamiento de requisitos, Análisis y Diseño, Construcción, Pruebas e Integración, para la creación de Portales Web de alta calidad y bajo costo.

En el trabajo describen los objetivos que se pretenden cumplir con la realización de la práctica empresarial, los antecedentes y la razón fundamental de porque hacer esta entrega. Se hace una descripción de todos los instrumentos o herramientas utilizados y la metodología implementada en el transcurso de la práctica. Se generan unas conclusiones y comentarios sobre los resultados obtenidos con el fin de expresar los resultados obtenidos.

# 1 OBJETIVOS

## 1.1 OBJETIVO GENERAL

Participar en el proceso de implementación de la metodología PSP (*Personal Software Process*) – TSP (*Team Software Process*) en PRAGMA S.A, aplicando los conocimientos del curso PSP Fundamentals en el desarrollo de Portales Web, con el fin de llevar a la organización a niveles más altos de madurez CMMI que le permitan a la empresa la reducción de costos y el aumento en la calidad de los proyectos de software.

## 1.2 OBJETIVOS ESPECIFICOS

- Desarrollar habilidades en la construcción de portales web con EPiServer CMS, ASP.NET y C#.
- Utilizar los estándares de codificación, análisis y arquitectura que sigue PRAGMA para el desarrollo de software
- Aprender la metodología PSP-TSP por medio del curso PSP Fundamentals.
- Intervenir en el proceso de lanzamiento TSP de los proyectos asignados por PRAGMA.
- Aplicar con éxito el proceso de PSP en las actividades principales del ciclo de vida del software.

## 2 JUSTIFICACIÓN

Este trabajo se hizo con el fin de explicar cómo se aplicarán los conceptos del proceso de desarrollo de software desde el levantamiento de requerimientos hasta las pruebas de desarrollador y puesta en producción, basado en la metodología TSP/PSP que permite mejorar la calidad de los productos desarrollados, haciendo énfasis en la mejora continua de las practicas individuales de la ingeniería de software.

Con la realización de la práctica empresarial se está entregando a la sociedad un nuevo ingeniero con la capacidad necesaria para hacer parte de la productividad, desarrollo y crecimiento industrial brindando nuevos conocimientos y experiencias que estimulen el mejoramiento tecnológico.

En cuanto a la parte económica PRAGMA recibirá ingresos por los productos desarrollados y ajustados a las necesidades del cliente, y estos a su vez se beneficiarán incrementando la interactividad de sus sitios o portales web, mejorando y fortaleciendo la relación con el usuario. Ejecutar la metodología TSP/PSP implica cambios en el modelo tradicional de desarrollo de software, incrementando la calidad y predicción de costos. Lo que puede permitir a las empresas del país, elevar la madurez de la industria de software hasta alcanzar niveles los más populares como el CMMI, para desempeñarse mejor en competencias internacionales.

## 3 MARCO TEÓRICO

### 3.1 MICROSOFT VISUAL STUDIO

Visual Studio Es un entorno de desarrollo integrado (IDE, por sus siglas en ingles - *integrated development environment*) cuenta con un conjunto de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles.

Es compatible con diferentes lenguajes de programación como: Visual Basic, Visual C++, Visual C# y Visual J#, lo que permite a estos lenguajes compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan la biblioteca de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles.

Visual Studio incluye un editor de código de apoyo y un diseñador de formularios para la construcción de interfaces gráficas de usuario y diseños de páginas web.

### 3.2 MICROSOFT VISUAL STUDIO 2008

Es una versión de Visual Studio que trabaja con la versión 3.5 del framework de .NET, para poder programar para las versiones anteriores (2.0, 3.0). Microsoft Visual Studio 2008 viene con muchas mejoras y funcionalidades como:

- **XAML** (del inglés *eXtensible Application Markup Language*), Es el Lenguaje de Marcado Estructurado para Aplicaciones, en pocas palabras su utilidad más usual es para pasar un diseño de interfaz a la aplicación para poder trabajar sin problemas.
- **IntelliSense para JavaScript**,. La tecnología IntelliSense es la que se encarga de detectar qué es lo que el usuario está tecleando para darle la opción de seleccionar en una lista las posibles palabras que el programador va a escribir.
- **LINQ** (del inglés *Language Integrated Query*), LINQ define operadores de consulta estándar que permiten a lenguajes habilitados con LINQ filtrar, enumerar y crear proyecciones de varios tipos de colecciones usando la misma sintaxis. Tales colecciones pueden incluir arreglos, clases enumerables, XML, conjuntos de datos desde Bases de Datos relacionales y orígenes de datos de terceros.

- **ASP .NET**

Es un Framework de aplicaciones web, desarrollado y comercializado por Microsoft para permitir a los programadores crear sitios web dinámicos, aplicaciones Web y servicios Web.

- **C#**

Es un lenguaje de programación orientado a objetos desarrollado por Microsoft, diseñado para generar programas sobre .NET Framework. Su sintaxis deriva de C/C++ e incluye mejoras derivadas de otros programas como; Java, Visual Basic o Delphi.

### **3.3 SQL SERVER 2008 R2**

Es un sistema de base de datos relacional (*RDBMS*) de Microsoft, su principal función es la de almacenar y recuperar datos, Sus lenguajes para consultas son T-SQL y ANSI SQL, entre sus principales características podemos encontrar:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Entorno grafico de administración, que permite el uso de comandos DDL y DML de manera grafica.
- Control de Excepciones y manejo de errores.
- Soporte de almacenamientos de datos de diferentes variedades: XML, correo electrónico, hora /calendario, archivos y documentos.

### **3.4 EPISERVER CMS**

EPiServer CMS es un Sistema Administrador de contenido (por sus siglas en ingles *Content Management System*), utilizado para desarrollar grandes sitios Web, sistemas Intranet y Extranet.

EPiServer CMS está basado en la plataforma .NET, desarrollado por la empresa sueca EPiServer AB.

Es necesario estar familiarizado con los conceptos sobre desarrollo web en ASP .NET para la creación de soluciones.

Los proyectos de EPiServer pueden ser creados directamente dentro de Visual Studio 2008. También se pueden crear plantillas de páginas EPiServer CMS, propiedades personalizadas, paginas maestras y mucho más directamente desde la interfaz de Visual Studio.

#### **3.4.1 Arquitectura**

La arquitectura de EPiServer es una arquitectura abierta y flexible, diseñado para ser ampliado funcionalmente. La ampliación de EPiServer es posible de muchas maneras. Plug-in's son a menudo agregados o creados para ampliar las funciones de edición y modo de administración. También es posible personalizar los tipos de datos de las propiedades, propiedades personalizadas permiten diferentes formas en la información que puede ser accedida por el usuario del sitio web.

### **3.5 ORACLE**

Es un sistema de gestión de base de datos objeto-relacional ó ORDBMS (por sus siglas en ingles, *Object – Relational DataBase Management System*), desarrollado por Oracle Corporation, es una potente herramienta basada en la arquitectura Cliente/Servidor para la gestión de Bases de Datos Relacionales desarrollada por Oracle Corporation. Ofrece una interfaz intuitiva basada en el explorador, que es capaz de administrar las bases de datos, crear tablas, vistas y otros objetos de bases de datos, importar, exportar y visualizar datos de tablas, ejecutar scripts de SQL y generar informes. Además, soporta transacciones, es estable, escalable y multiplataforma.

Para desarrollar en Oracle se utiliza PL/SQL, el cual es un lenguaje de quinta generación, bastante potente para tratar y gestionar la base de datos. Oracle Designer y Oracle Developer son las herramientas de programación que se usan en este motor. Permite entonces: Manipular datos de una Base de Datos Oracle, usar técnicas de control (bucles) y condicionales, controlar las filas de una consulta (una a una), controlar errores (excepciones) definidas por el usuario o propios de Oracle (predefinidos), no diferencia las minúsculas de las mayúsculas

Su arquitectura Objeto – Relacional es un punto medio entre las base de datos relacionales que incorporan una estructura estática de datos (tablas) y las base de datos orientada a objetos donde el elemento de trabajo en lugar de una tabla es el objeto, que incorpora datos y procedimientos, sin embargo este ultimo enfoque está en estudio y no es clara su ventaja frente al enfoque relacional, entonces, siendo Oracle punto medio entre estos dos enfoques.

### 3.6 STARTUML

Es un proyecto Open Source rápido, flexible, extensible, con muchas características y de acceso libre para el desarrollo de UML. Es una plataforma que se ejecuta sobre Win32. El objetivo del proyecto StarUML es construir una herramienta de modelado de software.

- **UML**

Lenguaje Unificado de Modelado, es una herramienta que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas y reducir el proceso de desarrollo.

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como *modelo*. El modelo UML describe lo que supuestamente hará un sistema, pero no dice como implementar dicho sistema.

Los diagramas más usados en PRAGMA:

- **Diagrama de Clases**, un conjunto de clases con atributos (propiedades) y acciones (funciones), una clase es una categoría o grupo de cosas que tienen atributos y acciones similares.
- **Diagrama de secuencia**, los diagramas de clases representan una información estática. En un sistema funcional los objetos interactúan entre si y tales interacciones suceden con el tiempo. El diagrama de

secuencia muestra la mecánica de la interacción con base en tiempos.

### **3.7 PSP**

Proceso Personal de Software (por sus siglas en ingles Personal Software Process), es un proceso de superación personal que ayuda a controlar, administrar y mejorar la forma de trabajar de cada desarrollador. Es un Framework estructurado de formas, guías y procedimientos para desarrollar software. Creado por Watts Humphrey del Software Engineering Institute (SEI).

Los ingenieros utilizan PSP para desarrollar software siguiendo los procesos definidos y recolectando métricas detalladas sobre el tiempo requerido para producir un producto, los defectos que se inyectan y se retiran en las distintas etapas del desarrollo, y el tamaño del producto terminado.

Estos indicadores son analizados con métodos estadísticos, permitiendo a los ingenieros producir estimaciones muy precisas sobre históricos, seguir el progreso y la calidad de un proyecto en curso, predecir el impacto del calendario y predecir la calidad de un producto de software terminado. PSP enseña a los ingenieros a determinar cuantitativamente la forma de mejorar sus procesos.

Usado en forma adecuada PSP provee la información necesaria para cumplir los compromisos, y hace a los elementos rutinarios de trabajo más predecibles y eficientes.

El propósito de PSP es ayudar a mejorar las habilidades del ingeniero de software, es una herramienta que puede ser utilizada de muchas formas, por ejemplo: ayudará a administrar su trabajo, evaluar sus talentos y construir sus habilidades. Puede ayudarle a hacer mejores planes, seguir de manera precisa su rendimiento y medir la calidad de sus productos. Ya sea para diseñar programas, levantar requisitos, documentar o mantener software existente, PSP puede ayudarle a hacer mejor el trabajo.

PSP provee la información y técnicas de análisis que necesitas para determinar cual tecnología y métodos de trabajo son mejores para usted. También le ayuda a entender porque comete errores y como encontrarlos, repararlos y prevenir que vuelva a hacerlos.

### 3.7.1 Proceso PSP

El principal objetivo del proceso PSP, es proveer un Framework para escribir los programas y reunir información sobre tu trabajo, el proceso PSP provee algunos beneficios como:

- Una estructura conveniente para la realización de tareas a pequeña escala.
- Un Framework para medir estas tareas.
- Una base para la mejora de procesos.

El proceso se divide en tres partes principales Planeación, Desarrollos y Post Mortem. A continuación describo los principales objetivos de estas etapas, las cuales se ilustran en la Figura 1:

- **Planeación:**

- Producir u obtener los requisitos del programa.
- Asegurar que los requerimientos son claros y sin ambigüedades.
- Resolver cualquier inquietud.
- Estimar el tiempo de desarrollo necesario.

- **Desarrollo:**

- **Diseño**

Diseñar el programa de acuerdo con los requerimientos.

Ingresar en el log de defectos cualquier defecto encontrado con respecto a los requerimientos, mientras se hace el diseño.

Ingresar en el log de tiempos, el tiempo total gastado en el diseño.

- **Código**

Implementar el diseño.

Ingresar en el log de defectos cualquier defecto encontrado con respecto al diseño.

Ingresar en el log de tiempos, el tiempo total gastado en la codificación.

- **Compilación**

Compilar el programa hasta que no haya errores de compilación.

Reparar todos los defectos encontrados.

Ingresar en el log de defectos cualquier defecto encontrado con respecto al código.

Ingresar en el log de tiempos, el tiempo total gastado en compilación.

- **Pruebas**

Probar hasta que no haya errores.

Reparar todos los defectos encontrados.

Ingresar en el log de defectos cualquier defecto encontrado con respecto al código.

Ingresar en el log de tiempos, el tiempo total gastado en pruebas.

- **Post Mortem**

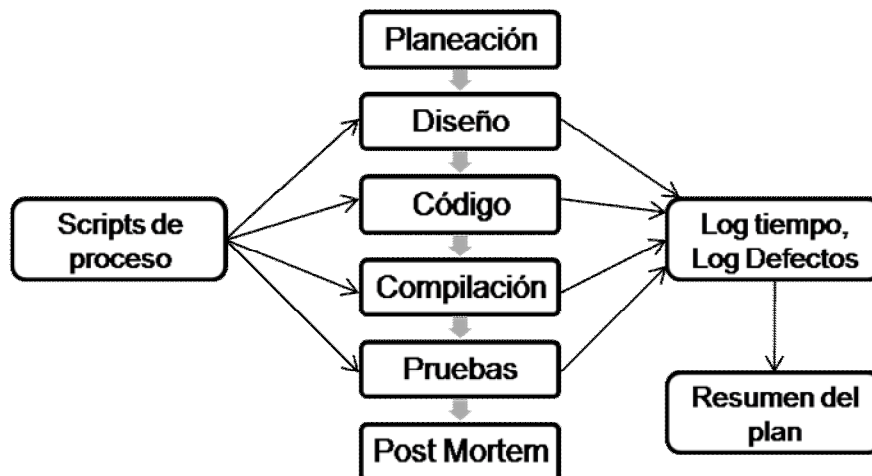
Revisar el resumen del plan del proyecto, con los tiempos reales, defectos y tamaño de los datos, que se registraron en el log de cada etapa para:

Recordar y guardar en el log cualquier defecto que fuese omitido

Corregir la información guardada sobre los defectos y corregir.

Corregir cualquier error en la recopilación de tiempos.

**Figura 1. Proceso PSP**

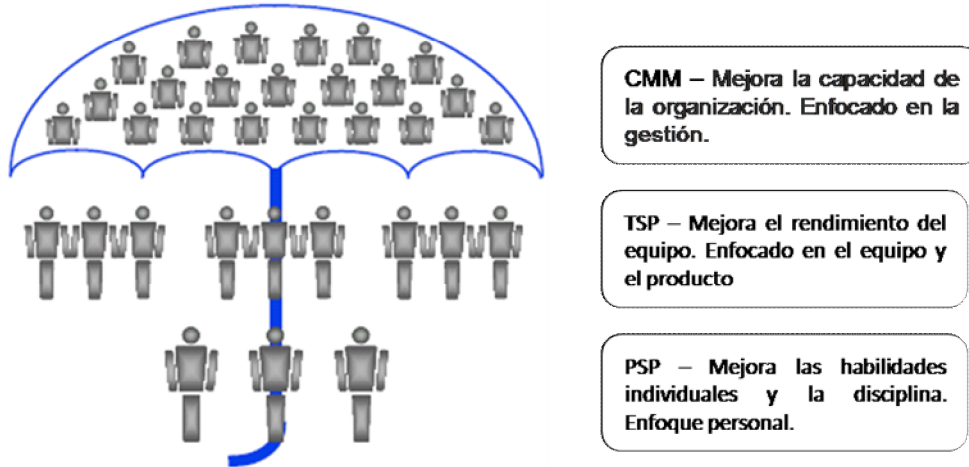


### 3.8 TSP

TSP proporciona un proceso operación definido para guiar a los ingenieros y administradores a través de los para los pasos de trabajo en equipo. Creado por Watts Humphrey del SEI. Este proceso explica los pasos necesarios para establecer un eficiente ambiente de trabajo. Con una serie de métodos que ayudan a los equipos de ingeniería, así de una manera más eficaz desarrollar y dar soporte a los sistemas de software. El objetivo del TSP es mejorar los niveles de calidad y productividad de un proyecto de desarrollo de software de un equipo, con el fin de ayudarlos a alcanzar los acuerdos de costos y tiempos en dicho desarrollo.

La Figura 2 muestra como TSP agrupa los principios de los equipos PSP y Métodos de CMM para producir equipos eficaces. En esencia, CMM y PSP proporcionan el contexto y habilidades para una ingeniería efectiva mientras el TSP guía a los ingenieros en cómo hacer el trabajo. Así, el TSP aprovecha la preparación proporcionada por el PSP y CMM, mientras que también proporciona una orientación explícita sobre la forma de hacer el trabajo.

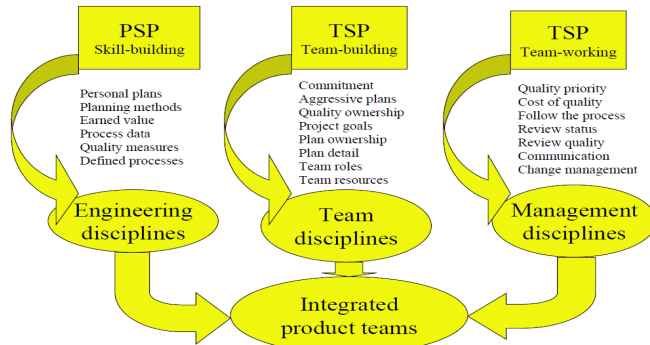
**Figura 2. Métodos de mejora de proceso**



### **3.8.1 La estructura del PSP**

Los principales elementos del proceso TSP son mostrados en la Figura 3. Antes que los miembros puedan participar en un equipo TSP, deben saber cómo hacer un trabajo disciplinado. Entrenarse en PSP es necesario para proporcionar a los ingenieros conocimiento y habilidades para usar TSP. El entrenamiento en PSP consiste en aprender cómo hacer planes detallados, reunir y usar el proceso de información, desarrollar planes de valor ganado, usar el valor ganado para realizar seguimiento del proyecto, medir y administrar la calidad de un producto, y definir y usar procesos operacionales.

**Figura 3. Construcción de un equipo TSP.**



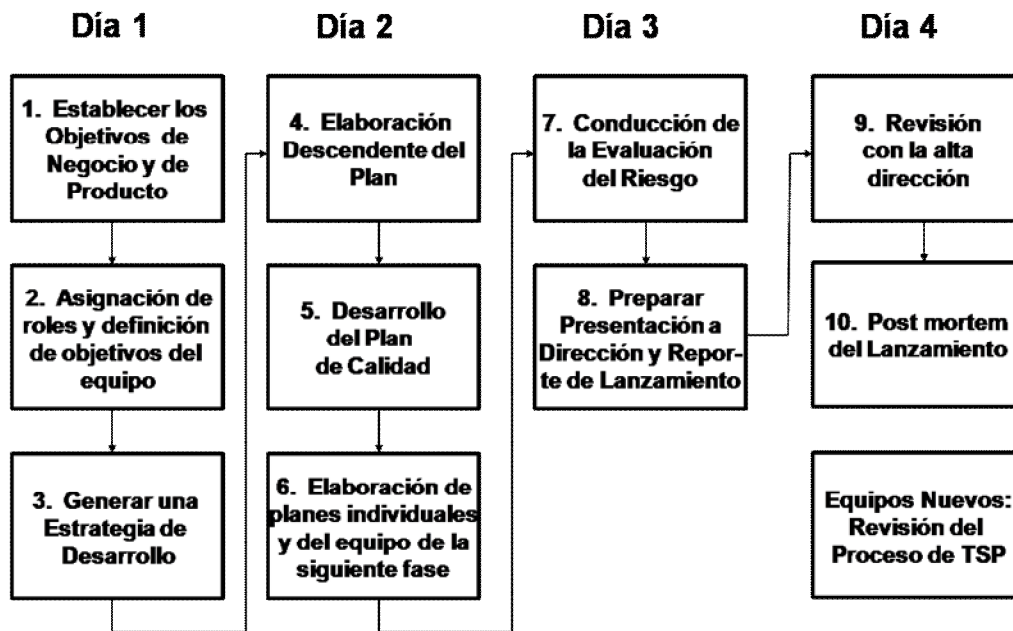
### 3.8.2 Lanzamiento TPS

El lanzamiento establece un entendimiento común del equipo acerca del proyecto, sobre temas como:

- Metas de la dirección para el proyecto.
- Las metas del equipo y metas de los miembros del equipo.
- Procesos que el equipo utilizará.
- Roles que los miembros del equipo realizarán.
- El trabajo de desarrollo a ser realizado.
- Plan para hacer el trabajo.
- Sistema de reporte a la dirección y al cliente.
- Proceso de comunicación del equipo funcionando.

La Figura 4 muestra el proceso que se lleva en el lanzamiento TSP.

**Figura 4. Proceso de Lanzamiento TSP.**



### 3.9 PROCESS DASHBOARD

Process Dashboard es una iniciativa de código abierto para crear una herramienta de apoyo para PSP - TSP. Fue desarrollado por la Fuerza Aérea de los Estados Unidos y ha seguido evolucionando con el modelo de código abierto. Disponible para descargar con la licencia pública GNU. Process Dashboard permite:

- Recolección de datos: tiempo, defectos, tamaño; plan vs información actual.
- Planeación: scripts integrados, plantillas, formas y resúmenes, PROBE (*PROxy - Based Estimating*)
- Seguimiento: soporte del Valor Ganado.
- Análisis de Datos: gráficos e informes sobre la tendencia de los datos históricos.
- Exportar Datos: exporta datos a Excel o exporta datos en formato de texto para el uso con herramientas externas.

Las principales fortalezas del Process Dashboard son:

- **Facilidad de uso**
  - Optimiza la recolección de los indicadores más comunes (tiempo y defectos).
  - Las tareas se organizan jerárquicamente.
- **Flexibilidad / Extensibilidad**
  - Nuevos procesos y nuevos tipos de datos se pueden agregar sin programación.
  - Los Scripts, plantillas, formularios y resúmenes son HTML, por lo que se puede ejecutar en cualquier navegador.
- **Independencia de la plataforma**
  - Implementado 100% en Java se puede ejecutar en Windows, Unix, Linux, Macintosh y otros.
- **Precio**
  - Es una herramienta de código abierto que se distribuye sin costo, no depende de ningún software con licencia, por lo que no es necesario comprar ningún software para usar el Process Dashboard.

## 4 METODOLOGÍA

PRAGMA es una empresa con 15 años de experiencia en la creación y desarrollo de soluciones de negocio basadas en Internet y medios relacionados.

Uno de sus principales propósitos es convertirse en una empresa de clase mundial, reconocida por sus proyectos y la calidad de los mismos. Por esta razón PRAGMA ha optado por la implementación de la Metodología PSP – TSP; que le permita llegar a niveles más altos de madurez CMMI.

Para cumplir este propósito PRAGMA puso en marcha El proyecto PSP-TSP uno de los proyectos más importantes que PRAGMA ha emprendido, el proyecto consta de 4 partes:

- Capacitación de las directivas de la compañía (*TSP Executive Strategy Seminar*): al cual asisten todos los gerentes de la compañía.
- Capacitación a los "líderes" de equipo (*Leading a Development Team*): en el caso de PRAGMA a esta capacitación asisten los directores de proyectos.
- Capacitación a los ingenieros (*Personal Software Process (PSP) Fundamentals*): al cual asisten los ingenieros de los diferentes grupos de trabajo.
- Inicio de los proyectos piloto.

Siendo parte de uno de los equipos pilotos encargados de poner en marcha PSP-TSP en la compañía, fue necesario que asistiera al curso o capacitación PSP Fundamentals.

A continuación se describe el proceso llevado durante la capacitación PSP Fundamentals y el inicio del proyecto.

### 4.1 CAPACITACIÓN PSP FUNDAMENTALS.

Este curso de 5 días enseña a los ingenieros los principios, conceptos y beneficios que tiene PSP. El objetivo del curso es que sus estudiantes sean capaces de aplicar los métodos de PSP a su propio proceso de trabajo y participar en un TSP.

El proceso de aprendizaje del curso se compone de 4 programas, los cuales pueden ser desarrollados en cualquier lenguaje de programación, para mi caso C#. Estos programas nos enseñan cómo medir y analizar nuestro proceso, gestionar y reducir los defectos de los programas desarrollados, para mejorar el

rendimiento personal. A demás proporciona un mecanismo para el cambio cultural en el equipo de desarrollo a través de su programa de formación.

## 4.2 INICIO DE LOS PROYECTOS PILOTO

Después de que los desarrolladores han completado el curso de PSP Fundamentals están preparados para ser parte de un TSP y el primer paso para esto es participar en un *Lanzamiento*.

### 4.2.1 Lanzamiento del Proyecto.

El lanzamiento de TSP está organizado como un conjunto de juntas, en cada junta se definen unos roles, se sigue una agenda y se realiza un reporte, como roles de la junta se tienen:

- Moderador: Concerta y encabeza la junta.
- Cronometrista: Ayuda al equipo a seguir la agenda y el guión, sigue y anota los tiempos gastados en los puntos de la agenda.
- Anotador: Registra todas las decisiones importantes (qué y quién) y acciones planeadas (qué, quién, cuándo). Verifica las acciones y decisiones con los asistentes de la junta al final de la misma.

Las junta son dirigidas por un Coach TSP de la organización o externo, esta persona esta a cargo de solucionar por medio de su experiencia: dudas o problemas que se vayan presentando durante el transcurso del lanzamiento.

A continuación describo el proceso que se llevo durante los 4 días de lanzamiento.

- **Día 1**
  - **Junta 1. Establecer los objetivos de negocio y producto**, en esta junta la gerencia expuso y argumento lo que desea que el equipo desarrolle, cuando necesita el producto, los recursos con que el equipo cuenta, la flexibilidad del equipo y porque el trabajo es tan importante

para la compañía. El equipo entonces da opiniones sobre los objetivos expresados por la administración y si está de acuerdo o no.

- Junta 2. Asignación de roles y definición de objetivos del equipo**, en esta junta el equipo revisa las metas que estableció la dirección, y adopta estas metas como parte del proyecto, agrega metas específicas al equipo, obteniendo un acuerdo sobre las metas del equipo. El equipo divide los roles de administración del equipo de tal forma que cada miembro tenga una responsabilidad de rol, entre los roles se encuentran: Interfaz con el cliente, diseño, implementación, pruebas, planeación, proceso, calidad, soporte. En la Figura 5. Se muestra la creación de miembros del equipo en el Process Dashboard y el tiempo disponible para el proyecto. Con respecto al tiempo invertido en el trabajo PSP – TSP propone que los ingenieros trabajen la mitad de la jornada laboral en actividades directas del proyecto, que generen valor, y la otra mitad del tiempo para invertirlo en actividades extras al proyecto.

**Figura 5. Creación de miembros del equipo en Process Dashboard.**

Name	Initials	Color	Hrs/Week	14/11/11	21/11	28/11	5/12	12/12	
Carlos Ospina Morales	CO	<span style="background-color: #FF69B4; border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></span>	20	START→	16	20	20	16	12
Luis Velez	LV	<span style="background-color: #0000FF; border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></span>	20	START→	16	20	20	16	20
Sebastian Holguin	SH	<span style="background-color: #00BFFF; border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></span>	20	START→	16	20	20	16	20
Mauricio Bedoya	MB	<span style="background-color: #9ACD32; border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></span>	20	START→	16	20	20	16	12
Roy Yarce	RY	<span style="background-color: #FFFF00; border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></span>	20	START→	16	20	20	16	20
Juan Camilo Atehortua	JA	<span style="background-color: #32CD32; border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></span>	20	START→	16	20	20	16	20
Alejandro Díaz	AD	<span style="background-color: #8A2BE2; border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></span>	2	START→	0	0	6	8	10
		<span style="background-color: #FFA500; border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></span>	20	START→	20	20	20	20	20

- Junta 3. Generar una estrategia de desarrollo**, en esta junta el equipo visualiza el proyecto desde varias perspectivas, definiendo el producto que construirán y como lo realizarán. El equipo procede a generar una lista de todos los productos que deben ser generados, una estrategia para cumplir las metas del equipo y un proceso para realizar el trabajo y el porcentaje que será invertido en cada fase del proceso, en este caso el equipo adapta el proceso TSP al proceso PRAGMA; combinando en gran medida las fases de *Análisis y Pruebas*. También genera una lista de cualquier actividad que deba ser hecha por el equipo. Con esto el equipo realiza una estimación de tamaño lo que permite visualizar

cuánto tiempo tomará el proyecto. En la Figura 6. Se muestra como quedaron los procesos de TSP y PRAGMA combinados para el equipo los procesos personalizados de desarrollo que se implementará en el equipo, con la herramienta Process Dashboard.

- **Día 2**
  - **Junta 4. Elaboración descendente del plan,** con ayuda del Process Dashboard como se muestra en la Figura 7. se definen las tareas en secuencia de inicio a final del proyecto, las cuales siguen el proceso del equipo, incluyen todos los productos y están detalladas para la siguiente fase. El tiempo necesario para cada tarea se estima en base al tamaño y los datos de productividad o directamente sobre las experiencias anteriores. Las tareas son programadas basándose en las horas disponibles del equipo. El equipo puede crear diferentes alternativas en el caso de que la primera no cumpla o satisfaga las metas de la dirección.
  - **Junta 5. Desarrollo de plan de calidad,**  
  
El equipo construye un plan de calidad del producto. Este plan muestra como el equipo llegará a su meta de calidad. En el TSP, la calidad del software durante el desarrollo del producto es medida contando los defectos normalizándolos por la medida apropiada de tamaño.

**Figura 6. Creación de proceso de desarrollo en Process Dashboard.**

Portal de familia - Common Team Workflows	
Name	%
Common Workflows	
Requisitos	
REQ Validación técnica	5%
REQ Documento de Alcance Detallado	45%
REQR DAD Review	20%
TD Casos de Prueba Sistema	15%
REQINSP DAD Inspect	15%
HLD	
HLD	50%
HLDR HLD Review	20%
TD Casos de Prueba Integracion	15%
HLDINSP HLD Inspect	15%
Assembly Funcional	
DLD Diseño detallado	24,1%
DLDR Revision diseño detallado	12%
TD Casos de prueba unitarias	3,7%
DLDinsp Inspeccion diseño detallado	9,7%
MISC Prueba conceptual	5,7%
CODE Codificacion	16,4%
CR Revision de la codificacion	8,2%
CODEINSP Inspección de la codificacion	7,3%
UT Pruebas unitarias	12,9%
Assembly Visual	
CODE Codificacion Visual	36,9%
CR Revision Visual	18,5%
CODEINSP Inspeccion Visual	15,9%
UT Pruebas unitarias Visual	28,7%

- Junta 6. Construcción de los Planes Detallados para la Siguiente Fase,** se asignan a los desarrolladores las tareas para la siguiente fase (desarrollo), los desarrolladores construyen sus propios planes. El equipo balancea sus tareas de la siguiente fase, de manera que todos terminen aproximadamente al mismo tiempo, en la Figura 8. Se muestra como el Process Dashboard indica el balance actual de las tareas del proyecto.

- **Día 3**
  - **Junta 7. Conducción de la evaluación de riesgo,** El equipo identifica y evalúa los riesgos para su plan. cada riesgo es evaluado con respecto a impacto y probabilidad. Se definen planes de mitigación para los riesgos de alta y media prioridad. Cada riesgo se asigna a un miembro del equipo para su investigación y seguimiento.

- **Junta 8. Preparar presentación a la dirección y reporte de lanzamiento,** El equipo prepara una presentación del plan a la dirección. Si el plan del equipo no cumple las metas de la dirección, el equipo incluye planes alternativos que pueden agregar más recursos, entregar una versión con funcionalidad reducida, iniciar con una versión pequeña y seguir posteriormente con una versión con funcionalidad completa.

- **Día 4**

- **Junta 9. Revisión con la alta dirección,** El equipo presenta su plan y si es necesario, las alternativas para cumplir las necesidades de negocio. La dirección entonces; prueba el plan del equipo para evaluar la calidad del trabajo del equipo, decide si el plan del equipo (o uno alternativo) es aceptable, si es necesario, pide al equipo examinar otras alternativas.
- **Junta 10. Post Mortem del Lanzamiento,** El administrador de la planeación revisa los datos de lanzamiento para asegurar que: todos los datos requeridos del lanzamiento son recolectados y registrados, los datos son capturados apropiadamente en las formas apropiadas y bases de datos del proyecto.

**Figura 7. Elaboración descendente del plan en Process Dashboard.**

Launch Size	Launch Size Accounting	Plan Size	Actual Size	Planned Time	Task Time	Task Details	Actual Time	DLD Lines
Size	Units	LOC	LOC	Text Pages	Req Pages	HLD Pages		
24,067.5	LOC		24,067.5					
24,067.5	LOC		24,067.5					
24,067.5	LOC		24,067.5					
8,792.5	LOC		8,792.5					
2,155	LOC		2,155					
6,637.5	LOC		6,637.5					
9,787.5	LOC		9,787.5					
5,487.5	LOC		5,487.5					
0	LOC		0					
0	Req Pages		0		0			
0	Req Pages		0		0			
0	Req Pages		0		0			
0	Req Pages		0		0			
0	LOC		0					
0	HLD Pages		0			0		
0	LOC		0					
0	Text Pages		0		0			
0	Text Pages		0		0			
0	Text Pages		0		0			
0	Text Pages		0		0			
5,487.5	LOC		5,487.5					
1,525	LOC		1,525					
150	LOC		150					
75	LOC		75					
75	LOC		75					
50	LOC		50					
150	LOC		150					
50	LOC		50					
75	LOC		75					
50	LOC		50					
75	LOC		75					
50	LOC		50					
75	LOC		75					
50	LOC		50					
75	LOC		75					
75	LOC		75					
0	LOC		0					
0	LOC		0					
125	LOC		125					
75	LOC		75					
75	LOC		75					
50	LOC		50					
100	LOC		100					
75	LOC		75					

**Figura 8. Balance de tareas en Process Dashboard.**

Launch Size	Launch Size Accounting	Plan Size	Actual Size	Planned Time	Task Time	Task Details	Actual Time	DLD Lines
Size	Units	LOC	LOC	Text Pages	Req Pages	HLD Pages		
24,067.5	LOC		24,067.5					
24,067.5	LOC		24,067.5					
24,067.5	LOC		24,067.5					
8,792.5	LOC		8,792.5					
2,155	LOC		2,155					
6,637.5	LOC		6,637.5					
9,787.5	LOC		9,787.5					
5,487.5	LOC		5,487.5					
0	LOC		0					
0	Req Pages		0		0			
0	Req Pages		0		0			
0	Req Pages		0		0			
0	Req Pages		0		0			
0	LOC		0					
0	HLD Pages		0			0		
0	LOC		0					
0	Text Pages		0		0			
0	Text Pages		0		0			
0	Text Pages		0		0			
0	Text Pages		0		0			
5,487.5	LOC		5,487.5					
1,525	LOC		1,525					
150	LOC		150					
75	LOC		75					
75	LOC		75					
50	LOC		50					
150	LOC		150					
50	LOC		50					
75	LOC		75					
50	LOC		50					
75	LOC		75					
50	LOC		50					
75	LOC		75					
75	LOC		75					
0	LOC		0					
0	LOC		0					
125	LOC		125					
75	LOC		75					
75	LOC		75					
50	LOC		50					
100	LOC		100					
75	LOC		75					

Carlos Ospina Morales  
 Luis Velez  
 Sebastian Holguin  
 Mauricio Bedoya  
 Roy Yance  
 Juan Camilo Atehortua  
 Alejandro Diaz

El segundo paso para la implementación de la metodología PSP – TSP, consiste en ejecutar las tareas asignadas en el plan. Cada miembro del equipo tiene su propio plan y debe ejecutar las tareas de acuerdo al orden de elaboración descendente del plan, que se realizó en la junta 4 del lanzamiento. Las tareas de muestran en el cronograma personal, integrado con el Process Dashboard, como lo ilustra la Figura 9. El Process Dashboard también permite personalizar el orden de ejecución de las tareas, como se muestra en la Figura 10. Este cambio es necesario cuando el ingeniero siente que no puede cumplir con alguna tarea por falta de recursos o tiempo.

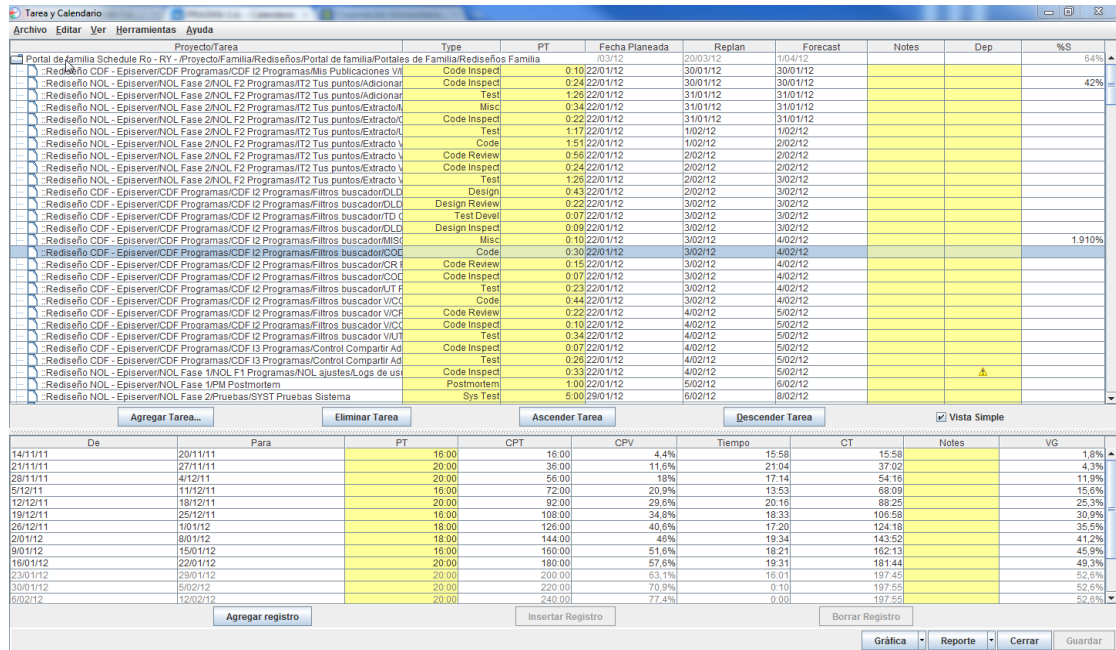
**Figura 9. Cronograma de actividades personal – Process Dashboard.**

The screenshot shows the 'Tareas y Calendario' application window. The top part is a Gantt chart with a tree view of tasks. The bottom part is a table with the following columns: De, Para, PT, GPT, CPV, Tiempo, CT, Notes, and VG. The table contains data for various dates from 14/11/11 to 5/03/12, showing task durations and completion percentages.

De	Para	PT	GPT	CPV	Tiempo	CT	Notes	VG
14/11/11	20/11/11	16.00	16.00	4.4%	15.58	15.58		1.8%
21/11/11	27/11/11	20.00	36.00	11.6%	21.04	37.02		4.3%
28/11/11	4/12/11	20.00	56.00	18%	17.14	54.16		11.9%
5/12/11	11/12/11	16.00	72.00	20.9%	13.53	68.99		15.6%
12/12/11	18/12/11	20.00	92.00	29.6%	20.16	88.25		25.3%
19/12/11	25/12/11	16.00	108.00	34.8%	18.33	106.58		30.9%
26/12/11	1/01/12	18.00	126.00	40.6%	17.20	124.18		35.5%
2/01/12	8/01/12	18.00	144.00	46%	19.34	143.52		41.2%
9/01/12	15/01/12	16.00	160.00	51.6%	18.21	162.13		45.9%
16/01/12	22/01/12	20.00	180.00	57.6%	19.31	181.44		49.3%
23/01/12	29/01/12	20.00	200.00	63.1%	16.01	197.45		52.6%
30/01/12	5/02/12	20.00	220.00	70.9%	0.10	197.55		52.6%
6/02/12	12/02/12	20.00	240.00	77.4%	0.00	197.55		52.6%
13/02/12	19/02/12	20.00	260.00	83.3%	0.00	197.55		52.6%
20/02/12	25/02/12	20.00	280.00	90.3%	0.00	197.55		52.6%
27/02/12	4/03/12	20.00	300.00	96.8%	0.00	197.55		52.6%
5/03/12	11/03/12	9.56	309.56	100%	0.00	197.55		52.6%

El Dashboard facilita las acciones de recolección de datos como la toma tiempos y recolección de defectos, que son las acciones más constantes durante el ciclo de desarrollo.

Figura 10. Cronograma de actividades personalizado – Process Dashboard.



## 4.2.2 Ejecución del Proyecto

Para comenzar el ciclo del proyecto cada ingeniero sigue el proceso generado luego de la junta 3 del lanzamiento y procede a desarrollar cada actividad generada en la junta 4.

### 4.2.2.1 Requisitos

La etapa de levantamiento de requisitos compete 5 etapas:

- **Validación Técnica:** el Líder Técnico y el Estratega de PRAGMA crean una descripción detallada de la funcionalidad del componente a desarrollar, el componente es un sistema que hace parte de un sistema más grande y que a su vez esta dividido en sub sistemas o Tareas.

Luego esta validación es entregada al *Cliente* el cual aprueba el documento o genera cambios que deben ser corregidos por el Líder Técnico y el Estratega, para entregar al ingeniero encargado de la actividad que

consiste en crear el documento de alcance detallado (DAD) la funcionalidad aprobada por el *Cliente*, el ingeniero revisa y comenta acerca de las dudas que tenga y lo devuelve a los validadores, quienes aclaran todas las dudas y le entregan la definición funcional nuevamente al ingeniero.

- **Creación Documento Alcance Detallado:** con la definición funcionalidad ya aprobada y aclarada, el ingeniero procede a crear DAD en el cual se describan las entradas, condiciones, salidas esperadas, imagen y descripción de la imagen.
- **Revisión DAD:** El ingeniero encargado de la creación del DAD sigue una lista de chequeo especial para los DAD que evalúa la pertinencia de la descripción funcional y corrige todos los defectos encontrados con respecto a la lista de chequeo, debe ingresar todos los defectos en la bitácora de defectos del Process Dashboard.
- **Creación de casos de prueba del sistema:** el ingeniero encargado del DAD crea además una definición de las pruebas de rendimiento que se llevaran para la funcionalidad.
- **Inspección DAD:** El ingeniero encargado de esta tarea revisa con respecto a la lista de chequeo si la funcionalidad es pertinente y cumple con los ítems en la lista, genera un documento con todos los defecto encontrados y es devuelta al creador del DAD, este último realiza las correcciones del documento, el inspector debe ingresar todos los defectos en la bitácora de defectos del Process Dashboard.

#### 4.2.2.2 Diseño de alto nivel

El proceso para crear el diseño de alto nivel consta de 4 etapas, las cuales son desarrolladas por el Líder Técnico del equipo, esta etapa no cuenta con participación de otros ingenieros por lo cual no se hace la descripción pertinente de las acciones que se ejecutan.

#### 4.2.2.3 Construcción

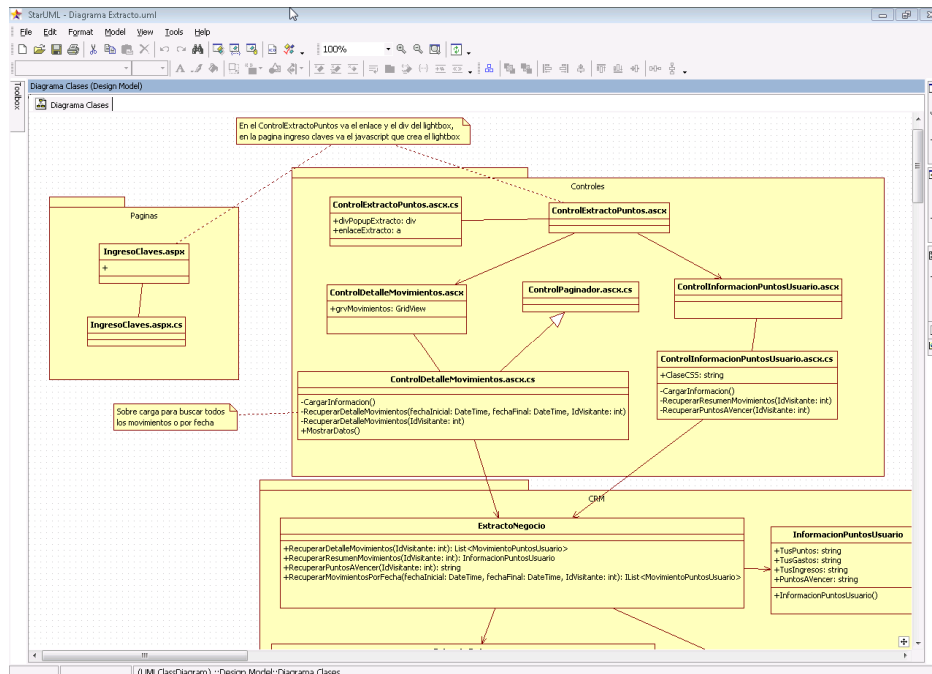
En PRAGMA el proceso de construcción comienza con la creación del Diseño grafico del portal, este diseño es creado por los diseñadores gráficos de PRAGMA, estas personas son ajenas al equipo TSP, este diseño es convertido posteriormente en HTML, parte visual del portal.

El proceso de construcción consta de 12 etapas, descritas a continuación:

- **Creación Diseño Detallado (DLD):** en la creación del diseño detallado se utiliza el DAD que contiene los requisitos funcionales de la tarea, el ingeniero encargado de la actividad procede a crear:

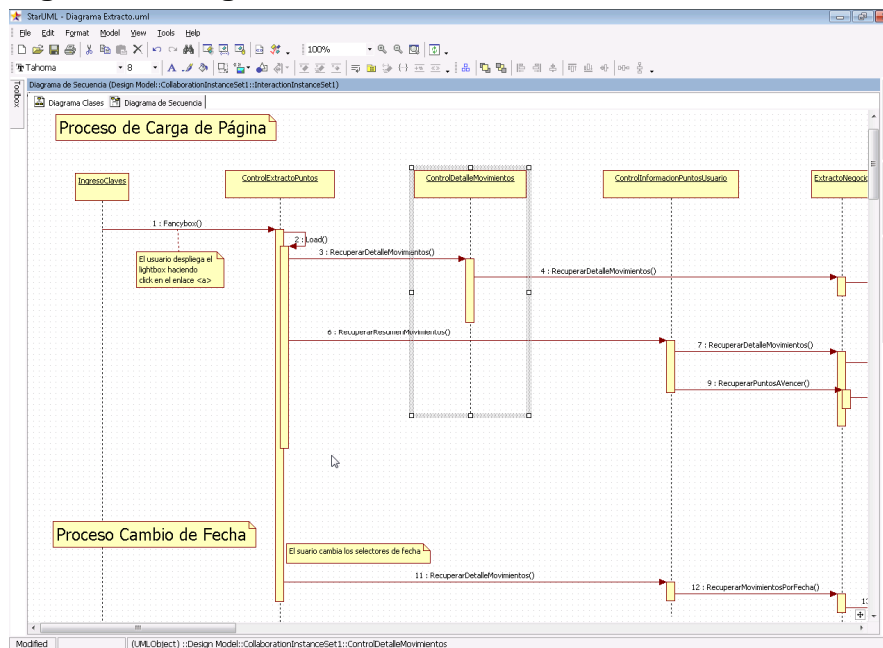
- **Plantilla operacional:** en la cual se describen los posibles escenarios que ocurren en la ejecución del programa, describiendo los pasos que deben ejecutar el usuario y el sistema para que este ocurra.
- **Plantilla funcional:** se describen la clases publicas que serán codificadas, sus propiedades y funciones públicas, junto con una descripción de lo que harán.
- **Plantilla lógica:** se escribe un pseudocódigo de las funciones públicas y privadas que se ejecutaran.
- **Diagrama de clases:** se crea en StartUML como se muestra en la Figura 11.

**Figura 11. Diagrama de Clases en StartUML.**



- **Diagrama de Secuencia:** se crea en StartUML como se muestra en la Figura 12.

**Figura 12. Diagrama de Secuencia en StartUML**



- **Revisión de diseño:** el encargado de crear el diseño, revisa conforme a una lista de chequeo específica para los DLD, encuentra los defectos y los corrige, estos defectos deben ser ingresados en la bitácora de defectos del Process Dashboard.
- **Creación de casos de prueba:** se crean los casos de prueba, para la ejecución de pruebas unitarias, donde se valide toda la funcionalidad de toda la tarea a desarrollar.
- **Inspección de diseño:** el ingeniero encargado revisa el diseño conforme a la lista de chequeo de los DLD, evalúa si cumple con los ítems en la lista y genera un reporte con los defectos encontrados, este reporte es revisado por el creador del DLD y corrige los defectos reportados, estos defectos deben ser ingresados en la bitácora de defectos del Process Dashboard.
- **Codificación:** después de corregir los defectos encontrados en la revisión de diseño, se procede a crear el código del programa con respecto al diseño elaborado, en esta parte solo se desarrolla la parte funcional del Portal web.
- **Revisión de código:** el creador del código se encarga de validar el programa con la lista de chequeo correspondiente a la fase de código y se

verifica que cumpla con cada uno de los elementos de la lista, los defectos encontrados deben ser ingresados en la bitácora de defectos del Process Dashboard.

- **Inspección de código:** el encargado de esta tarea evalúa si el código del programa cumple con los ítems de la lista de chequeo y genera un reporte con los errores encontrados. El creador del código debe ingresar los defectos en la bitácora de defectos del Process Dashboard.
- **Pruebas Unitarias:** el creador del código ejecuta el programa y los casos de prueba creados en la etapa de Creación de casos de prueba, si ocurre algún error en la ejecución se deben reportar los defectos en la bitácora de defectos del Process Dashboard.
- **Codificación visual:** luego de la pruebas se procede a crear la parte visual de la funcionalidad, se revisa el HTML entregado por los diseñadores, esta es la parte con la que el usuario final interactúa y es la parte primordial de un portal web.
- **Revisión de código visual:** el creador del código visual se encarga de validar el programa con la lista de chequeo correspondiente a la fase de código y se verifica que cumpla con cada uno de los elementos de la lista, los defectos encontrados deben ser ingresados en la bitácora de defectos del Process Dashboard.
- **Inspección de código visual:** el encargado de esta tarea evalúa si el código del programa cumple con los ítems de la lista de chequeo y genera un reporte con los errores encontrados. El creador del código debe ingresar los defectos en la bitácora de defectos del Process Dashboard.
- **Pruebas Unitarias:** el creador del código ejecuta el programa y los casos de prueba creados en la etapa de Creación de casos de prueba, si ocurre algún error en la ejecución se deben reportar los defectos en la bitácora de defectos del Process Dashboard.

#### 4.2.2.4 Pruebas

En esta etapa del ciclo se evalúan los resultados de la etapa de construcción, realizando pruebas de sistema, pruebas de certificación y pruebas por el cliente.

Las pruebas se realizan por iteración, en PRAGMA se manejan entregas de componentes en la cuales se muestra al cliente un modulo o componente específico del Portal web, para que este evalúe el estado del proyecto y la calidad

de los productos desarrollados. De igual manera al cliente, PRAGMA posee un equipo de certificación ajeno al equipo TSP, que genera casos de uso con respecto a los DAD desarrollados, este equipo realiza pruebas y genera un reporte de Incidencias o errores, las cuales deben ser corregidas antes de entregar el cliente.

A continuación se mencionan las fases del ciclo de pruebas:

- **Pruebas de sistema:** el equipo ejecuta los casos de prueba del sistema generados en la etapa de requisitos, verifica que cumpla con todas las funcionalidades explícitas en el DAD y genera un reporte final. En caso de encontrar errores, estos errores se reportan a las personas encargadas del componente y estas deben proceder a corregirlos e ingresar los defectos encontrados en la bitácora de defectos del Process Dashboard.
- **Pruebas de certificación:** luego de que el equipo TSP ejecuta las pruebas del sistema, el equipo de certificación ejecuta los casos de uso y genera los reportes de incidencias o defectos, este paso es llamado Ciclo, al terminar el Ciclo, el equipo TSP corrige las incidencias, ingresar los defectos encontrados en la bitácora de defectos del Process Dashboard y reporta al equipo de certificación que todas las incidencias han sido solucionadas. El equipo de certificación pone en marcha un siguiente ciclo, este proceso se repite hasta que no se reporten más incidencias por parte del equipo de certificación.
- **Pruebas del cliente:** el cliente en compañía de un miembro del equipo es guiado en la utilización del componente desarrollado, este verifica que cumpla con las funcionalidades exigidas; si el cliente no se siente conforme con una funcionalidad este puede pedir cambios, el equipo TSP evalúa los riesgos que implica y se procede a crear una nueva tarea de construcción si es necesario.

#### 4.2.2.5 Despliegue y Post Mortem

En la etapa de despliegue el equipo se encarga de instalar el sitio en un ambiente de producción, que cumpla con las especificaciones acordadas entre el Cliente y PRAGMA, este ambiente es suministrado por el Cliente y PRAGMA evalúa si cumple con las especificaciones mínimas requeridas para el buen funcionamiento del portal.

Los ingenieros encargados de la construcción del portal, generan los manuales necesarios para instalación y administración del sitio.

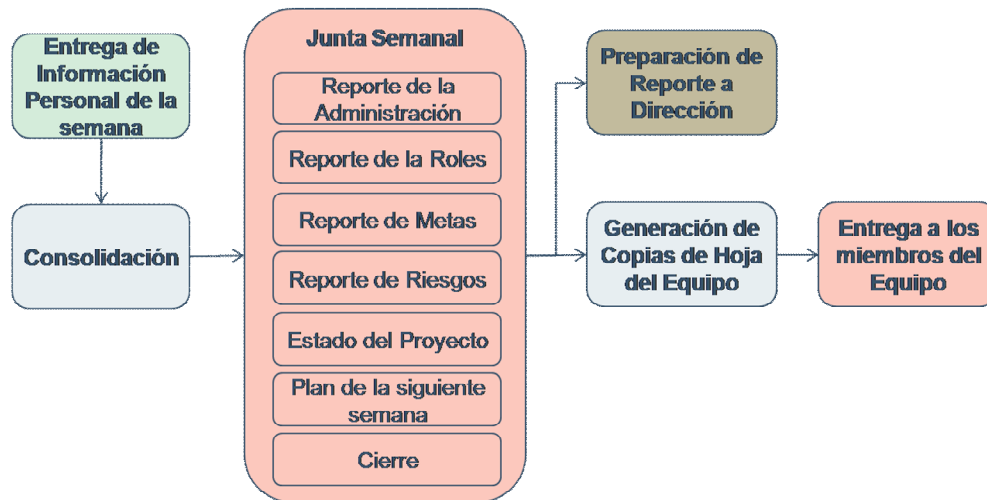
De igual forma se encargan de capacitar a los encargados de administrar el sitio Web, en la utilización de este.

En la etapa de post mortem el equipo genera un reporte con el resumen del proyecto, se evalúa la línea base del proyecto y se compara con la línea actual, se validan las estadísticas de defectos, tiempo y tamaño de todos los componentes desarrollados comparando lo planeado con lo actual, se generan conclusiones de mejora y reporte final del proyecto.

#### **4.2.2.6 Junta Semanal**

Como su nombre lo indica, durante el ciclo del proyecto el equipo realiza una junta cada semana, guiada por un coach y el líder del equipo, en la cual se asegura que todos los miembros del equipo comprenden el estado actual del proyecto y conocen qué es lo que sigue por hacer. En la Figura 13. Se ilustra el proceso llevado a cabo en la junta semanal.

**Figura 13. Proceso Junta Semanal TSP.**



Al inicio de cada junta dos miembros del equipo voluntariamente se asignan los roles de anotador y cronometrista.

Actividades realizadas en la junta:

- **Reporte de la administración:** el líder del equipo abre la junta con un breve resumen de cualquier nuevo desarrollo o situación.
- **Reporte de los roles:** cada miembro del equipo tiene un rol, asignado en la junta 2 del lanzamiento, los miembros del equipo deben asegurar que se cumplan las metas específicas del rol, en esta junta la persona encargada expresa al equipo el estado actual del rol.
  - Interfaz con el cliente, comprender lo que el cliente quiere y necesita.
  - Diseño, encabezar al equipo en la producción de un diseño superior. Utilizar por completo todas las habilidades e ideas del equipo para producir este diseño. Asegurar que el diseño y su documentación sean de alta calidad.
  - Implementación, encabezar al equipo en la producción de una implementación superior. Asegurar que la implementación cumple totalmente el diseño. Producir un producto implementado que sea de alta calidad.
  - Planeación, ayudar al equipo a ejecutar un proyecto bien planeado y seguido. Ayudar al equipo con su planeación y seguimiento de

avance personal. Regularmente seguir y reportar el estado del equipo contra el plan.

- Procesos, asegurar que el equipo tiene disponibles procesos definidos para todas las actividades clave. Asistir a los miembros del equipo en definir y usar procesos. Asegurar que los datos de proceso del equipo son reportados y analizados. Asiste al equipo en la definición y resolución de problemas de procesos.
  - Calidad, encabezar al equipo in la producción y seguimiento de un plan de calidad. Proporcionar análisis oportuno y advertencia de problemas de calidad. Desempeñarse efectivamente como el moderador de inspección del equipo.
  - Soporte, ayudar al equipo a usar las herramientas y métodos apropiados. Maneja la administración de la configuración y el control de cambios del equipo. Actúa como el abogado de reutilización del equipo.
  - Pruebas, encabezar al equipo en desarrollo de planes de prueba amplios. Asegurar que el sistema sea totalmente probado y realiza de manera apropiada todas las funciones más importantes.
- **Reporte de metas y riesgos:** los responsables de metas y riesgos explican al equipo el estado actual de estos aspectos.
  - **Estado del proyecto:** Cada miembro del equipo revisa su avance y estado personal del proyecto; tareas concluidas en la semana anterior actuales contra planeadas, valor generado en la semana anterior actual contra planeado, explica el resultado de sus acciones y las dificultades que tuvo.

El administrador de la planeación resume el avance y estado del equipo, valor generado y horas dedicadas del equipo actual contra planeado, proyección para la conclusión del valor generado actual.

- **Planes de la siguiente semana:** Cada miembro del equipo resume las tareas planeadas para la siguiente semana y cualquier dependencia especial, y se compromete con un valor ganado para esa semana.

El líder del equipo revisa situaciones o acciones esperadas.

## 5 CONCLUSIONES

La practica empresarial ayuda a completar la formación académica de cada estudiante, permitiendo aplicar todos los conceptos aprendidos en la universidad en el mundo laboral y adquirir nuevos conocimientos en un entorno real de trabajo. Reconociendo las fortalezas y debilidades que se alcanzaron durante todos los semestres cursados. Esta experiencia laboral permite hacerse más apto para el futuro profesional y participar en el desarrollo socio económico del país, de igual manera nos prepara psicológicamente para la actividad empresarial, ya que durante el transcurso de la practica pueden presentarse inconvenientes, lo que podría generar actitudes negativas frente al trabajo. También favorece la integración del estudiante en equipos multidisciplinarios de trabajo, generando habilidades de comunicación, autoconfianza e independencia, pues no se siempre se está en contacto con los miembros del equipo de trabajo al que pertenece, en ocasiones, hablar las altas directivas o clientes, requiere de un lenguaje y una actitud diferente.

La implantación de la metodología PSP – TSP permite a las empresas mediante el cambio en su proceso de desarrollo mejorar la calidad de los productos construidos, esta mejora se vio reflejada durante el transcurso de la práctica empresarial, reduciendo los defectos encontrados por el equipo de certificación de PRAGMA, comparados con los productos construidos sin esta metodología.

PSP – TSP también permite mejorar las habilidades de desarrollo de software, esto ocurre mediante el riguroso proceso que se lleva, las revisiones personales y las inspecciones personales, reflejan los errores más comunes y la forma de evitarlos, por lo que se está más consiente a la hora de crear los diseños detallados y programar.

## 6 RECOMENDACIONES

En el proceso de formación académica del programa Ingeniería Informática en la Corporación Universitaria Lasallista, debe incluirse una asignatura que indique los principios de calidad con respecto al desarrollo software, debido a que el mercado laboral es muy exigente, deben incluirse conceptos de trabajo como PSP o CMMI.

La implementación de PSP – TSP en PRAGMA requiere un gran cambio en la cultura organizacional, que permita mejorar la confianza de las altas directivas al equipo, para esto los equipo que vayan a realizar el lanzamiento de un proyecto, deben iniciar con un proyecto que no sea de alto impacto en la compañía, para evitar la presión y el estrés del proyecto.

## BIBLIOGRAFÍA

EPISERVER WORLD. EPiServer CMS and .NET [en línea] <<http://world.episerver.com/Get-Started/EPiServer-CMS/Introduction-to-EPiServer-CMS/EPiServer-CMS-and-NET/>> [Citado el 29 de Enero de 2012]

EPISERVER WORLD. Important Concepts [en línea] <<http://world.episerver.com/Get-Started/EPiServer-CMS/Introduction-to-EPiServer-CMS/Important-Concepts/>> [Citado el 29 de Enero de 2012]

GRUPOTRESS INTERNACIONAL. Microsoft Visual Studio 2008, Boletín Febrero 2008 [en línea] <<http://www.willydev.net/InsiteCreation/v1.0/willycrawler/2008.05.01.articulo.lo%20nuevo%20en%20visual%20studio%202008.pdf>> [Citado el 29 de Enero de 2012]

HUMPHREY, Watts S. PSP A Self-Improvement Process for Software Engineers. 6 ed. Massachusetts. Addison Wesley. 2011.

HUMPHREY, Watts S. The Team Software Process (TSP). 2000. Carnegie Mellon Software Engineering Institute. CMU/SEI-2000-TR-023.

HUMPHREY, Watts S. The Personal Software Process (PSP). 2000. Carnegie Mellon Software Engineering Institute. CMU/SEI-2000-TR-022.

NOTICIA DEL MUNDO. Especial: ¿Qué es Oracle y Para Qué Sirve? [en línea] <<http://noticiadelmundo.com/especial-que-es-oracle-y-para-que-sirve/1277/>> [Citado el 28 de Enero de 2012]

PS&J Software Six Sigma. PSP, TSP and the CMM [en línea] <[http://www.softwaresixsigma.com/Tsp\\_A\\_Cmm.htm](http://www.softwaresixsigma.com/Tsp_A_Cmm.htm)> [Citado el 28 de Enero de 2012]

SOFTWARE ENGINEERING INSTITUTE. Personal Software Process (PSP)  
Fundamentals [en línea] <<http://www.sei.cmu.edu/training/p18b.cfm>> [Citado el 29  
de Enero de 2012]