**Team Software Process**

**Ciclo 1**



**Nombre Proyecto:** Sistema TSP

**Fecha:** Marzo 9 de 2011

**Realizado por:**

|  |  |
| --- | --- |
| **Persona** | **Código** |
| Carlos Ernesto González Vargas | 200819123 |
| Willian Alejandro Idrobo Luna | 201110544 |
| Erik Fernando Arcos Franco | 201110856 |
| David Pérez Chibuque | 201117818 |
| Andrés Mauricio Erazo Benavides | 201110949 |
| Sandra Milena Gómez Ríos | 201110951 |

**Control de versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Autor** | **Descripción del Cambio** |
| 1.00 | Marzo 9 de 2011 | Ingenium | Creación del documento |
| 1.01 | Marzo 10 de 2011 | Líder de Desarrollo | Definición del diseño conceptual |
| 1.02 | Marzo 12 de 2011 | Lider de Planeacion | Definicion del plan de trabajo |
| 1.03 | Marzo 14 de 2011 | Lider de Desarrollo | Se define análisis y diseño |
| 1.04 | Marzo 14 de 2011 | Lider de Soporte | Adicion del manual de soporte |
| 1.05 | Marzo 15 de 2011 | Lider del Grupo | Postmortem |
|  |  |  |  |

 

**Contenido**

[**1.** **Documento de Lanzamiento** 2](#_Toc287952167)

[**1.1.** **Objetivos y Métricas** 2](#_Toc287952168)

[**1.1.1.** **Objetivos del Grupo** 2](#_Toc287952169)

[**1.1.2.** **Objetivos de los Miembros del Grupo** 2](#_Toc287952170)

[**1.1.3.** **Objetivos del Proyecto y del Proceso** 2](#_Toc287952171)

[**1.2.** **Definición de los roles del equipo** 3](#_Toc287952172)

[**1.3.** **Reglas** 4](#_Toc287952173)

[**1.4.** **Horarios de Reunión** 5](#_Toc287952174)

[**1.5.** **Información que se reporta en cada ciclo** 5](#_Toc287952175)

[**2.** **Delimitación del Proyecto** 6](#_Toc287952176)

[**2.1.** **Definición del requerimiento** 6](#_Toc287952177)

[**2.2.** **Diseño Conceptual** 9](#_Toc287952178)

[**2.3.** **Estimación de Tiempo y Esfuerzo** 10](#_Toc287952179)

[**2.3.1.** **Regresión Lineal** 12](#_Toc287952180)

[**2.3.2.** **Estimación de Tiempo** 14](#_Toc287952181)

[**2.4.** **Plan de trabajo** 14](#_Toc287952182)

[**2.5.** **Requerimientos planeados por ciclo** 15](#_Toc287952183)

[**3.** **Definición Primer ciclo** 18](#_Toc287952184)

[**3.1.** **Objetivo** 18](#_Toc287952185)

[**3.2.** **Plan de Trabajo** 18](#_Toc287952186)

[**4.** **Artefactos Primer ciclo** 20](#_Toc287952187)

[**4.1.** **Arquitectura de la Solución** 20](#_Toc287952188)

[**4.2.** **Herramientas de Desarrollo** 20](#_Toc287952189)

[**4.2.1.** **Herramientas** 20](#_Toc287952190)

[**4.2.2.** **Control de Versiones** 21](#_Toc287952191)

[**4.2.3.** **Referencias** 22](#_Toc287952192)

[**4.3.** **Guía de Implementación** 23](#_Toc287952193)

[**4.4.** **Aplicativo Desarrollado** 24](#_Toc287952194)

[**5.** **PostMortem Primer Ciclo** 26](#_Toc287952195)

**Team Software Process**

**Ciclo 1**

1. **Documento de Lanzamiento**
   1. **Objetivos y Métricas**
      1. **Objetivos del Grupo**

* O1: Producir un producto de Buena Calidad
  + M1: 100% Requerimientos incluidos en el producto final
  + M2: Porcentaje de defectos encontrados en cada etapa mayor a 65%
* O2: Realizar un proyecto bien administrado y productivo
  + M3: Porcentaje de error máximo permitido en la estimación de tamaño del producto menor a 30%
  + M4: Porcentaje de error máximo permitido en la estimación de cantidad de horas menor a 30%
    1. **Objetivos de los Miembros del Grupo**
* O3: Ser un miembro efectivo y cooperativo
* O4: Hacer el trabajo personal de manera disciplinada consistentemente
  + M5: Promedio  de evaluación dentro del grupo superior a 3
* O5: Planear y hacer seguimiento al trabajo personal
  + M6: Se registro el 90% de las actividades  y resumen de calidad
  + M7: Porcentaje de tareas planeadas y completadas, promedio semanal mayor a 80%
  + M8: Ser puntuales a todas las reuniones planeadas (puntualidad> 80%)
    1. **Objetivos del Proyecto y del Proceso**
* O6:Objetivos del proyecto por ciclo
  + M9:Redefinir objetivos del documento por ciclo
* O7: Tiempo definido para cada tarea
  + M10: El tiempo empleado por tarea no debe superar el 20% del tiempo planeado
* O8: Definir documento de análisis de requerimientos para cada ciclo
  + M11: Identificar el 95% de los requerimientos
* O9: Cumplir los requerimientos definidos en el ciclo
  + M12 Finalizar a tiempo
  + M13 Implementar el 80% de los requerimientos
* O10:Construir codigo mantenible y bajo estandares
  + M14: Cubrimiento de métodos y atributos con Javadoc en 70%
  + M15: Componentes nombrados de acuerdo al estándar en un 70%
  + M16: Código con formato de indentación en un 100%.
* O11:Desarrollar pruebas para los requermimientos funcionales
  + M17: Desarrollar 1 prueba automatizadas para cada requerimiento establecido del ciclo
* O12:El tiempo de los miscelaneos   debe ser bajo respecto al tiempo  del proyecto
  + M18: Los miscelaneos no debe superar el 10% por ciclo
* O13:Realizar diseño de interfaz antes de la codificación
  + M19: Diagrama de flujo por requerimiento = 1
  + M20: Prototipo de interfaz por requerimiento = 1
* O14:Realizar diagrama de clases
  + M21: Realizar un diagrama de clases para el producto
* O15:Documentos y formatos
  + M22: Definir una plantilla para los reportes de defectos = 1
  + M23: Porcentaje de defectos encontrados con “Defect Report” =  70 %
  + M24: Porcentaje de defectos aprobados para solución efectivamente solucionados (Antes de la prueba de sistema) = 90%
  + M25: Errores encontrados en pruebas de sistema no mayor al 10%
  1. **Definición de los roles del equipo**

Para la ejecución del proyecto se van a tener los siguientes roles los cuales deben ser asignados dentro de los integrantes del grupo:

* ***Líder del grupo:***Es quien conduce el grupo y se asegura de que todos los integrantes reportan sus datos del proceso y terminan su trabajo como se planeó.
* ***Líder de desarrollo:***Se encarga de liderar y guiar el grupo en la definición, diseño, desarrollo y pruebas del producto.
* ***Líder de planeación:***Su función es dar soporte y guía al grupo en las tareas deplaneación y seguimiento del proyecto.
* ***Líder de calidad:***Se encarga de dar soporte en definir las necesidades del proceso, en hacer el plan de calidad y en hacer seguimiento al proceso y a la calidad del producto.
* ***Líder de soporte:***Es quien da soporte al grupo en la determinación, obtención y administración de las herramientas necesarias para desarrollar el producto.

Basados en el experiencias e intereses de los integrantes se realizó una votación en donde cada uno indicó qué rol debería desempeñar y cuál debería desempeñar cada uno de sus compañeros, basados en esto se seleccionó para cada rol la persona con mayor votación, quedando la asignación de la siguiente forma:

Tabla 1. Definición de Roles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Líder de** | | | | |
| **Integrante** | **Grupo** | **Desarrollo** | **Planeación** | **Calidad** | **Soporte** |
| Carlos Ernesto González | 4 | 2 |  |  |  |
| Willian Alejandro Idrobo | 1 | 4 | 1 |  |  |
| Erik Fernando Arcos | 1 | 4 |  | 1 |  |
| David Pérez |  | 1 |  | 4 |  |
| Andrés Mauricio Erazo |  |  |  |  | 5 |
| Sandra Milena Gómez |  |  | 5 |  |  |

Con los resultados de la anterior tabla se tiene que a criterio del grupo los más indicados para cada uno de los roles definidos son:

**Líder del Grupo:** Carlos Ernesto González

**Líder de Desarrollo:** Erik Fernando Arcos

**Líder de Desarrollo:** Willian Alejandro Idrobo

**Líder de Planeación:** Sandra Milena Gómez

**Líder de Calidad:** David Pérez

**Líder de Soporte:** Andrés Mauricio Erazo

* 1. **Reglas**

• Llegar a la reunión a la hora acordada. Cuando uno de los integrantes se presente más de 15 minutos tarde a una reunión se le impondrá una multa de $2.000 por cada 10 minutos de retraso.

• Estar preparado para la reunión. Cada integrante del grupo deberá estar contextualizado con el tema a tratar, asegurarse de haber leído todas las diapositivas, PDFs, mensajes, visitados todos los links y llevar copias de los documentos.

• Responsabilidad con el grupo. Cada integrante será honesto sobre el tiempo, problemas y elaboración de la parte de trabajo que tiene a su cargo.

• Todo el grupo estará abierto y escuchará con respeto los comentarios, aportes, inquietudes y problemas de cada integrante.

• Cuando se presenten problemas, se compartirán opiniones siempre con el fin de buscar una solución y no de formar discordias.

• Para la toma de decisiones en donde se presenten desacuerdos, se realizará por votación. En caso de que exista empate entre las opciones, se presentará la inquietud al profesor teniendo como tiempo máximo de respuesta 2 días, si no se cuenta con ese tiempo o no se recibió repuesta la decisión se realizará por medio de una rifa.

• Si uno de los integrantes incumple con alguna actividad, el resto del grupo decidirá el tiempo adicional que se le dará para terminar el compromiso teniendo en cuenta el tiempo disponible del infractor.

• Cuando se presenten problemas personales entre integrantes que afecten el trabajo del grupo, se les dará un tiempo prudencial para que arreglen sus diferencias, si después de este tiempo el problema persiste en la siguiente reunión del grupo se tratará de solucionar entre todos los integrantes.

• Cuando se presente una duda sobre algún tema y la respuesta generada por el grupo no satisfaga a quien formula la inquietud, se procederá a consultar al profesor.

• Si se desea realizar una reunión o algo que influya en las decisiones del grupo, se debe informar por medio de un correo a todos los integrantes, y siempre que se envíe una inquietud al profesor, tanto la información de la inquietud como la respuesta deberá ser replicada a los demás integrantes del grupo.

• Al realizarse una presentación a terceros sólo se expondrán las ideas concertadas por el grupo. En caso que el exponente se equivoque en la presentación de la idea no se mostrarán desacuerdos ante el público. En la siguiente reunión de trabajo se discutirán los desacuerdos presentados.

• Todos los integrantes deberán trabajar sobre las mismas herramientas seleccionadas por el líder de soporte. En caso de presentarse problemas con el ambiente de alguno de los integrantes y se deba seleccionar otra herramienta para este, se tratará de afectar en lo mínimo la configuración del resto del equipo.

• Si uno de los integrantes no se puede presentar a alguna de las reuniones, deberá avisar mínimo con 2 horas de anticipación, de lo contrario se le cobrará una multa de $10.000 siempre y cuando no tenga una excusa valedera y que pueda ser verificada por el resto de los integrantes.

* 1. **Horarios de Reunión**

De acuerdo a la reunión de lanzamiento realizada el día 9 de marzo de 2011, se definió el siguiente horario de reuniones:

Tabla 2. Horarios de Reunion

|  |  |  |
| --- | --- | --- |
| Días de la semana | Horario | Tipo |
| Lunes | 09:30 pm | Internet |
| Miércoles | 09:30 pm | Internet |
| Sábado | 09:00 am | Presencial |
| Domingo | 09:00 am | Internet y/o presencial |

* 1. **Información que se reporta en cada ciclo**

Para realizar el postmortem y tener una visión sobre el estado actual y futuro del proyecto, cada uno de los integrantes del grupo debe reportar la siguiente información al finalizar cada ciclo:

* Análisis personal

En esta primera sección de información que se debe reportar está relacionada a la percepción de su propio trabajo de acuerdo al rol que desempeño y los objetivos que se planteo.

* + Objetivos definidos por rol

Qué objetivos se planteo de acuerdo a su rol.

* + Cumplimiento de los objetivos durante el ciclo

Evalué si los compromisos fueron alcanzados y porque.

* + Inconvenientes

Problemas para alcanzar los objetivos.

* + Aspectos a mejorar

Mencione lo que cree debe mejorar individualmente

* Reflexión

Esta segunda sección se refiere a las percepciones que tuvo del grupo.

* + ¿Qué nos faltó como grupo en este ciclo?

Mencione los aspectos faltantes o débiles dentro del grupo.

* + ¿Cómo debería ser el proceso en el próximo ciclo?

Mencione las mejoras para el próximo ciclo.

* + ¿Qué etapas fueron las más difíciles? Porqué?

De las etapas del Proceso de software cuales considera que requirieron mayor esfuerzo.

* + ¿Qué no me gustó del ciclo?

Mencione los aspectos con los que no está de acuerdo del ciclo del proceso TSP.

1. **Delimitación del Proyecto**
   1. **Definición del requerimiento**

El producto que se desarrollará corresponde al software encargado de medir y presentar la productividad del equipo de trabajo y la de cada uno de sus integrantes a través del conteo y recuperación de anotaciones que se han ingresado en los archivos java que conforman el proyecto. El producto debe presentar una interfaz gráfica.

Se han identificado las siguientes funcionalidades del producto:

* Permitir al usuario registrar el grupo de trabajo dentro de la aplicación
* Permitir al usuario registrar las tareas planeadas por cada ciclo y asignarlas a un responsable.
* Permitir al usuario registrar el plan de calidad
* Permitir al usuario generar el reporte de productividad del grupo, así como también la productividad de cada integrante del grupo.

A continuación se definen los requerimientos en detalle:

|  |  |
| --- | --- |
| **Identificador del Requerimiento** | **RF1.0** |
| **Nombre del Requerimiento** | Administración de miembros del equipo |
| **Actores** | Usuario |
| **Propósito** | Este requisito comprende la creación, modificación y consulta de los miembros del equipo en la aplicación. Todos los miembros registrados se deben persistir en un archivo de propiedades. |
| **Principales Parámetros** | **Resultados** |
| 1. Identificador del integrante del grupo. | El integrante ha sido creado, modificado o eliminado del archivo de propiedades |
| 2. Nombre del integrante del grupo |  |
| 3. Rol del integrante del grupo |  |

|  |  |
| --- | --- |
| Identificador del Requerimiento | RF2.0 |
| **Nombre del Requerimiento** | Administración de planeación de tareas |
| **Actores** | Usuario |
| **Propósito** | Este requisito comprende la creación, modificación y eliminación de las tareas del plan.  Este plan debe persistir en un archivo de propiedades. |
| **Principales Parámetros** | **Resultados** |
| 1. Identificador de la tarea del plan | Actividad del plan creada, modificada o eliminada |
| 2. Nombre de la tarea |  |
| 3. Duración de la tarea |  |
| 4. Responsable de la tarea |  |
| 5. % Valor ganado |  |

|  |  |
| --- | --- |
| **Identificador del Requerimiento** | **RF3.0** |
| **Nombre del Requerimiento** | Administración del plan de calidad |
| **Actores** | Usuario |
| **Propósito** | Este requisito comprende la creación y modificación del plan de calidad. Este plan de calidad debe persistir en un archivo de propiedades. |
| **Principales Parámetros** | **Resultados** |
| 1. Identificador del plan de calidad | Plan de calidad creado, modificado y persistido en el archivo de propiedades |
| 2. Cantidad de errores inyectados Planeacion |  |
| 3. Cantidad de errores inyectados Diseño |  |
| 4. Cantidad de errores inyectados Análisis |  |
| 5. Cantidad de errores inyectados Pruebas |  |
| 6. Cantidad de errores inyectados Inspección |  |
| 7. Cantidad de errores inyectados Postmorten |  |

|  |  |
| --- | --- |
| **Identificador del Requerimiento** | **RF4.0** |
| **Nombre del Requerimiento** | Generación Reporte Productividad |
| **Actores** | Usuario |
| **Propósito** | Este requisito comprende la visualización de las estadísticas grupales e individuales. |
| **Principales Parámetros Presentados** | **Resultados** |
| 1. LOC reales vs LOC estimadas por cada grupo | Reporte con los parámetros presentados. |
| 2. LOC reales vs LOC estimadas por cada integrante del grupo. |  |
| 3. Productividad del grupo |  |
| 4. Productividad de cada integrante del grupo |  |

|  |  |
| --- | --- |
| **Identificador del Requerimiento** | **RF5.0** |
| **Nombre del Requerimiento** | Generación Reporte de Ciclos |
| **Actores** | Usuario |
| **Propósito** | Este requisito consiste en la presentación del estado de cada ciclo. Cuando se selecciona una actividad de un ciclo se presenta el estado de esta actividad |
| **Principales Parámetros Presentados** | **Resultados** |
| 1. Identificador de la tarea | Reporte con los parámetros presentados. |
| 2. Descripción de la tarea |  |
| 3. Tiempo planeado |  |
| 4. % valor ganado planeado |  |
| 5. % valor ganado real |  |
| 6. Responsable y rol |  |

* 1. **Diseño Conceptual**

Con base a la definición del requerimiento total del sistema se realiza el siguiente diagrama de clases que sirve como proxy para poder identificar el tamaño y esfuerzo necesario para construir la aplicación deseada.



Ilustración 1. Diseño Conceptual

Donde se tiene que cada una de las clases definidas cumple con las siguientes funcionalidades:

Tabla 3. Definición de Roles

|  |  |  |
| --- | --- | --- |
| **Clase** | **Tipo** | **Funcionalidad** |
| **FramePrincipal** | Presentación | Permite controlar todos los paneles de la aplicación |
| **PanelPersonas** | Presentación | Panel que permite mostrar la información de los integrantes del grupo |
| **PanelPlanCalidad** | Presentación | Panel que permite administrar el plan de calidad del proyecto |
| **PanelPlan** | Presentación | Panel que permite administrar el plan para cada ciclo |
| **PanelReporte** | Presentación | Panel que permite generar el reporte de productividad para cada ciclo |
| **Analizador** | Lógica | Analiza la información almacenada en las anotaciones realizadas para que sean mostradas al usuario |
| **Persona** | Lógica | Clase que permite gestionar la información de cada integrante del grupo |
| **PlanCalidad** | Lógica | Clase que permite gestionar el plan de calidad del sistema |
| **PlanProductoXCiclo** | Lógica | Clase que permite gestionar el plan de producto para cada plan |
| **Proyecto** | Lógica | Clase que permite gestionar la información del proyecto |
| **Interrupción** | Lógica | Clase que permite gestionar las interrupciones encontradas |
| **GestorArchivoPropiedades** | Lógica | Clase abstracta que controla las operaciones básicas del acceso a los archivos de propiedades que almacenan la información del sistema |
| **LogT** | Datos | Anotación para las tareas |
| **LogD** | Datos | Anotación para los defectos |
| **LogInt** | Datos | Anotación para las interrupciones |
| **Loc** | Datos | Anotación para contar el total de líneas de código |

* 1. **Estimación de Tiempo y Esfuerzo**

Tomando como referencia la estimación para desarrollo en C++ de Humphrey[[1]](#footnote-1), mostrada en la siguiente tabla, se realiza la estimación del total de tiempo requerido para el sistema.

Tabla 4. Estimación de LOC en C++

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Categoría | Muy Pequeño | Pequeño | Mediana | Grande | Muy Grande |
| Cálculo | 2,3 | 5,1 | 11,3 | 24,7 | 54,0 |
| Datos | 2,6 | 4,8 | 8,8 | 16,3 | 30,1 |
| I/O | 9,0 | 12,1 | 16,2 | 21,6 | 28,9 |
| Lógica | 7,6 | 11,0 | 16,0 | 23,3 | 33,8 |
| Configuración | 3,8 | 5,0 | 6,6 | 8,5 | 11,1 |
| Texto | 3,8 | 8,0 | 17,1 | 36,4 | 77,7 |

Con base a la tabla anterior y al diseño realizado se realiza la siguiente tabla que permite estimar el tamaño en LOC del sistema:

Tabla 5. Definición de LOC estimados con base al proxy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLASE** | **MÉTODO** | **CATEGORIA** | **TAMAÑO** | **LOC** |
| **FramePrincipal** | MostrarPantalla | I/O | Mediano | 16,2 |
| InicializarComponentes | Lógica | Mediano | 16 |
| **PanelPersonas** | InicializarComponentes | Lógica | Mediano | 16 |
| GestionarPersona | Lógica | Mediano | 15,98 |
| **PanelPlanCalidad** | InicializarComponentes | Lógica | Mediano | 16 |
| GestionarPlanCalidad | Lógica | Mediano | 15,98 |
| **PanelPlan** | InicializarComponentes | Lógica | Mediano | 16 |
| GestionarPlan | Lógica | Mediano | 15,98 |
| **PanelReporte** | MostrarReportePersonas | I/O | Mediano | 16,2 |
| MostrarReporteGrupo | I/O | Mediano | 16,2 |
| **Analizador** | AnalizarLogT | Cálculo | Grande | 21,6 |
| AnalizarLogD | Cálculo | Grande | 21,6 |
| AnalizarLogInt | Cálculo | Grande | 21,6 |
| AnalizarLoc | Cálculo | Grande | 21,6 |
| BuscarArchivos | Data | Mediano | 8,8 |
| GenerarReportePersonas | Cálculo | Muy Grande | 54 |
| GenerarReporteGrupo | Cálculo | Muy Grande | 54 |
| **Persona** | GestionarRol | Lógica | Mediano | 16 |
| MapearDatos | Lógica | Grande | 23,3 |
| **PlanCalidad** | MapearDatos | Lógica | Grande | 23,3 |
| **PlanProductoXCiclo** | MapearDatos | Lógica | Grande | 23,3 |
| **Proyecto** | ListarCiclos | I/O | Mediano | 16,2 |
| MapearDatos | Lógica | Grande | 23,3 |
| **Interrupción** | MapearDatos | Lógica | Grande | 23,3 |
| **GestorArchivoPropiedades** | LeerPropiedad | Data | Grande | 16,3 |
| EliminarPropiedad | Data | Grande | 16,3 |
| InsertarPropiedad | Data | Grande | 16,3 |
| ModificarPropiedad | Data | Grande | 16,3 |
| **LogT** |  |  |  |  |
| **LogD** |  |  |  |  |
| **LogInt** |  |  |  |  |
| **Loc** |  |  |  |  |
|  |  |  | **Total LOC** | **577,64** |

* + 1. **Regresión Lineal**

Ya contamos con un valor estimado para el tamaño del proyecto, pero este valor se puede ajustar mas a la realidad si se toman como referencia valores comparativos de LOC estimados contra LOC reales.

Gracias al primer y segundo ciclo del PSP realizado de manera individual por cada uno de los integrantes del grupo se cuenta con los datos comparativos que nos permiten realizar un mejor acercamiento a la realidad, estaos datos son los siguientes:

Tabla 6. Datos iníciales del grupo de trabajo

|  |  |  |  |
| --- | --- | --- | --- |
|  | **LOC**  **Estimados** | **LOC**  **Real** | **Productividad**  **LOC/Hora** |
|  | 110 | 182 | 19,77 |
|  | 150 | 289 | 16,00 |
|  | 170 | 271 | 18,00 |
|  | 180 | 332 | 18,60 |
|  | 300 | 492 | 35,00 |
| **Total** | 910 | 1566 | 88,77 |
| **Promedio** | 182 | 313,2 | 22,19 |

Se aplica una regresión lineal teniendo en cuenta las siguientes formulas:



Tomando como referencia los datos de la tabla anterior tenemos:

 

Por tanto despejando los valores de  se obtiene:



Y al despejar  se obtiene:



Finalmente con la formula de la regresión lineal se obtiene un dato más cercano a la realidad:



Esta información se puede ver en la siguiente gráfica:

Ilustración 2. Regresión Lineal

En la gráfica anterior se puede observar que los datos históricos permiten generar la regresión lineal que indica que si el estimado es de 577,64 LOC lo más posible es que realmente se empleen 933.34 LOC, es decir aproximadamente 934 LOC

* + 1. **Estimación de Tiempo**

Con base a los resultados obtenidos en la sección anterior tenemos que para el equipo de trabajo se puede esperar lo siguiente:

**LOC Proyectadas:** 934 LOC

**Productividad Promedio:** 22,19 LOC/Hora

Por tanto tendríamos que:

**Tiempo total estimado = LOC Proyectadas / Productividad Promedio**

**Tiempo total estimado =** (934/LOC) / (22,19 LOC/Hora)

**Tiempo total estimado =** 42,09 Horas

* 1. **Plan de trabajo**

El plan de trabajo se realizará en tres ciclos con duración de una semana cada uno, empezando el día miércoles a las 10 am y terminando el siguiente miércoles a las 6 am. Cada ciclo estará basado en las fases de TSP:

* Lanzamiento
* Estrategia
* Plan
* Requisitos
* Diseño
* Implementación
* Pruebas
* Postmortem

Al finalizar cada ciclo no solo se entregará las líneas codificadas, sino también los documentos asociados y solicitados, entre ellos el postmortem. Más adelante se especifica claramente los entregables necesarios en cada ciclo.

* 1. **Requerimientos planeados por ciclo**

De acuerdo a los requerimientos definidos y al diseño conceptual obtenido se definen las siguientes actividades por cada ciclo:

Tabla 7. Plan de trabajo para el projecto

|  |  |
| --- | --- |
| 1 | AnalizadorTSP |
| 1.1 | Inicio |
| 1.2 | Lanzamiento. |
| 1.2.1 | Reunión lanzamiento |
| 1.2.2 | Definición objetivos |
| 1.2.3 | Elaboración documento |
| 1.2.4 | Corrección defectos |
| 1.3 | Ciclo 1 |
| 1.3.1 | Estrategia |
| 1.3.1.1 | Elaborar diseño conceptual |
| 1.3.1.2 | Elaborar estándar de anotaciones |
| 1.3.1.3 | Selección de herramientas |
| 1.3.1.4 | Definición de estrategia |
| 1.3.1.5 | Corrección defectos |
| 1.3.2 | Plan |
| 1.3.2.1 | Estimación tamaño |
| 1.3.2.2 | Estimación tiempo |
| 1.3.2.3 | Elaborar cronograma |
| 1.3.2.4 | Elaborar plan de calidad |
| 1.3.2.5 | Corrección defectos |
| 1.3.3 | Requisitos |
| 1.3.3.1 | Definir requerimientos funcionales y no funcionales |
| 1.3.3.2 | Corrección defectos |
| 1.3.4 | Diseño |
| 1.3.4.1 | Elaborar diagrama de clases |
| 1.3.4.2 | Definir arquitectura de solución |
| 1.3.4.3 | Diseñar prototipos interfaces graficas |
| 1.3.4.4 | Corrección defectos |
| 1.3.5 | Implementación |
| 1.3.5.1 | Ajustar estructura de anotaciones existentes |
| 1.3.5.2 | Crear anotaciones |
| 1.3.5.3 | Corrección defectos |
| 1.3.6 | Pruebas |
| 1.3.7 | Postmortem |
| 1.3.7.1 | Definir información |
| 1.3.7.2 | Crear presentación |
| 1.3.7.3 | Corrección defectos |
| 1.3.8 | Fin Ciclo 1 |
| 1.4 | Ciclo 2 |
| 1.4.1 | Estrategia |
| 1.4.1.1 | Retroalimentación Ciclo 1 |
| 1.4.1.2 | Definición de estrategia |
| 1.4.1.3 | Corrección defectos |
| 1.4.2 | Plan |
| 1.4.2.1 | Ajustes cronograma |
| 1.4.2.2 | Ajustes plan de calidad |
| 1.4.2.3 | Corrección defectos |
| 1.4.3 | Requisitos |
| 1.4.3.1 | Detallar requerimientos funcionales |
| 1.4.3.2 | Corrección defectos |
| 1.4.4 | Diseño |
| 1.4.4.1 | Ajustes diagrama de clases |
| 1.4.4.2 | Corrección defectos |
| 1.4.5 | Implementación |
| 1.4.5.1 | Implementar reporte productividad |
| 1.4.5.2 | Implementar reporte ciclos |
| 1.4.5.3 | Inspección de código |
| 1.4.5.4 | Corrección defectos |
| 1.4.6 | Pruebas |
| 1.4.6.1 | Diseño de casos de pruebas |
| 1.4.6.2 | Implementar y ejecutar pruebas junit modulo reporte productividad |
| 1.4.6.3 | Implementar y ejecutar pruebas junit modulo reporte ciclos |
| 1.4.6.4 | Corrección defectos |
| 1.4.7 | Postmortem |
| 1.4.7.1 | Definir información |
| 1.4.7.2 | Modificar presentación |
| 1.4.7.3 | Corrección defectos |
| 1.4.8 | Fin Ciclo 2 |
| 1.5 | Ciclo 3 |
| 1.5.1 | Estrategia |
| 1.5.1.1 | Retroalimentación Ciclo 2 |
| 1.5.1.2 | Definición de estrategia |
| 1.5.1.3 | Corrección defectos |
| 1.5.2 | Plan |
| 1.5.2.1 | Ajustes cronograma |
| 1.5.2.2 | Ajustes plan de calidad |
| 1.5.2.3 | Corrección defectos |
| 1.5.3 | Requisitos |
| 1.5.3.1 | Corrección defectos |
| 1.5.4 | Diseño |
| 1.5.4.1 | Corrección defectos |
| 1.5.5 | Implementación |
| 1.5.5.1 | Implementar reporte productividad |
| 1.5.5.2 | Implementar reporte ciclos |
| 1.5.5.3 | Integración de módulos |
| 1.5.5.4 | Inspección de código |
| 1.5.5.5 | Corrección defectos |
| 1.5.6 | Pruebas |
| 1.5.6.1 | Ajustes de casos de pruebas |
| 1.5.6.2 | Implementar y ejecutar pruebas junit modulo reporte productividad |
| 1.5.6.3 | Implementar y ejecutar pruebas junit modulo reporte ciclos |
| 1.5.6.4 | Implementar y ejecutar pruebas funcionales |
| 1.5.6.5 | Corrección defectos |
| 1.5.7 | Postmortem |
| 1.5.7.1 | Definir información y conclusiones |
| 1.5.7.2 | Modificar presentación |
| 1.5.7.3 | Corrección defectos |
| 1.5.8 | Fin ciclo 3 |

1. **Definición Primer ciclo**

El primer ciclo se pretende enfocar en tener un mayor entendimiento de los requerimientos del proyecto por parte de todos los integrantes del equipo, implementando el diseño conceptual y la planeación general tanto del primer ciclo como también una planeación preliminar de los ciclos restantes.

* 1. **Objetivo**

Realizar la planeación del proyecto basado en el diseño conceptual y la estimación obtenida de acuerdo a los requerimientos definidos.

Para el ciclo 1, se busca alcanzar los siguientes objetivos específicos:

* Realizar la configuración inicial de la solución, esto es, adaptar un producto existente que ya realiza la lectura de anotaciones.
* Modificar las anotaciones para se pueda registrar además el ciclo del desarrollo y el responsable de la tarea.

El desarrollo de la nueva funcionalidad del producto se considera en ciclos posteriores.

Los artefactos generados en esta fase son:

* Arquitectura de la solución
* Guía de configuración del ambiente
* Guía de implementación
* Prototipos
  1. **Plan de Trabajo**

El plan de trabajo para el primer ciclo es el siguiente:

Tabla 8. Plan de trabajo para el primer ciclo

|  |  |  |
| --- | --- | --- |
| 1 | AnalizadorTSP | Tiempo |
| 1.1 | Inicio |  |
| 1.2 | Lanzamiento. |  |
| 1.2.1 | Reunión lanzamiento | 30 |
| 1.2.2 | Definición objetivos | 20 |
| 1.2.3 | Elaboración documento | 60 |
| 1.2.4 | Corrección defectos | 5 |
| 1.3 | Ciclo 1 |  |
| 1.3.1 | Estrategia |  |
| 1.3.1.1 | Elaborar diseño conceptual | 40 |
| 1.3.1.2 | Elaborar estándar de anotaciones | 15 |
| 1.3.1.3 | Selección de herramientas | 20 |
| 1.3.1.4 | Definición de estrategia | 20 |
| 1.3.1.5 | Corrección defectos | 5 |
| 1.3.2 | Plan |  |
| 1.3.2.1 | Estimación tamaño | 40 |
| 1.3.2.2 | Estimación tiempo | 30 |
| 1.3.2.3 | Elaborar cronograma | 30 |
| 1.3.2.4 | Elaborar plan de calidad | 20 |
| 1.3.2.5 | Corrección defectos | 5 |
| 1.3.3 | Requisitos |  |
| 1.3.3.1 | Definir requerimientos funcionales y no funcionales | 10 |
| 1.3.3.2 | Corrección defectos | 5 |
| 1.3.4 | Diseño |  |
| 1.3.4.1 | Elaborar diagrama de clases | 20 |
| 1.3.4.2 | Definir arquitectura de solución | 30 |
| 1.3.4.3 | Diseñar prototipos interfaces graficas | 20 |
| 1.3.4.4 | Corrección defectos | 5 |
| 1.3.5 | Implementación |  |
| 1.3.5.1 | Ajustar estructura de anotaciones existentes | 10 |
| 1.3.5.2 | Crear anotaciones | 20 |
| 1.3.5.3 | Corrección defectos |  |
| 1.3.6 | Pruebas |  |
| 1.3.7 | Postmortem |  |
| 1.3.7.1 | Definir información | 10 |
| 1.3.7.2 | Crear presentación | 30 |
| 1.3.7.3 | Corrección defectos | 5 |
| 1.3.8 | Fin Ciclo 1 |  |

1. **Artefactos Primer ciclo**
   1. **Arquitectura de la Solución**

Para el proceso de construcción del producto se organizará la solución por paquetes de la siguiente manera:



Ilustración 3. Arquitectura de solución

Donde cada modulo cumple con las siguientes funciones:

* **Paquete de presentación:** Contiene las clases que se encargan de capturar y/o presentar la información al usuario.
* **Paquete de Lógica de Negocio:** Contiene las clases que modelan el negocio.
* **Paquete de Pruebas Unitarias:** Contiene las clases necesarias para realizar las pruebas unitarias de toda la aplicación.
  1. **Herramientas de Desarrollo**

El objetivo de este artefacto es presentar al equipo de trabajo actual una breve descripción de las herramientas que se usarán a lo largo de la ejecución del proyecto Sistema TSP.

Igualmente, el documento servirá de referencia para futuras iteraciones del proyecto, así como a futuros miembros del equipo, y estará abierto a la inclusión de nuevas herramientas a medida que la evolución del proyecto lo necesite.

* + 1. **Herramientas**

Para asegurar la correcta ejecución del proyecto se utilizaran diversas herramientas las cuales se usaran de manera colectiva y simultánea por los miembros del equipo.

Estas herramientas incluyen:

***Hosting del Proyecto***

El hosting del proyecto se realizara a través de la plataforma de código abierto de Google, **Google Code**.

Se decidió hacer uso de esta plataforma, siguiendo la iniciativa de código abierto de Google, la cual nos permite interactuar entre los miembros del equipo de una manera activa, sin la restricción de buscar un servidor privado en el cual podamos configurar nuestro repositorio de datos.

El proyecto se encuentra creado bajo el nombre de IngeniunManagment, y se puede acceder a el mismo desde la URL <http://code.google.com/p/ingenium-managment/>, y desde ahí se realizar toda la gestión y administración del proyecto.

Algunos de los valores agregados de usar Google Code, son, el versionamiento de la información, el cual se detallara más adelante en este documento; así como la wiki, el seguimiento de incidentes y la generación de paquetes de descargas.

***Entorno de Desarrollo***

El desarrollo de la aplicación se realizara en maquinas que usen sistemas operativos Microsoft Windows (XP, Vista, 7)

El desarrollo se realizara en la plataforma de Java Standard Edition Java Runtime Enviroment Version 6 (JRE1.6), la cual debe ser descargada e instalada (ver referencias)

El entorno de desarrollo será Eclipse Helios (3.6.2), el cual nos provee un entorno de desarrollo libre, robusto, amigable, con una librería de complementos (plugins) bastante amplia, las cuales nos pueden ayudar a mejorar nuestros niveles de productividad en el equipo.

El entorno de desarrollo debe ser descargado (ver referencias) según el sistema operativo (32 o 64 bits), e instalado. La instalación del entorno de desarrollo no representa mayores complicaciones ni configuraciones adicionales. Únicamente se debe tener previamente instalado la JRE 1.6. Después de la instalación se deberá configurar el WorkSpace de Eclipse la primera vez que se ejecuta el programa, el cual es la carpeta de trabajo donde estarán nuestros desarrollos.

* + 1. **Control de Versiones**

Nuestro proyecto cuenta con un sistema de manejo de versiones, proveído por la misma plataforma de Google Code. Cada miembro del equipo podrá acceder al repositorio través de un cliente el cual permitirá obtener las últimas versiones de los archivos (para cualquier usuario) y, actualizar los cambios correspondientes (solo usuarios autenticados)

***Plataforma***

La plataforma permite usar el versionado en base a **Mercurial** o en base a **Subversion.** Para nuestro proyecto de decidió usar Subversion como sistema de control de versiones, puesto que este sistema es ampliamente usado en proyectos globales, por ende, es un producto bastante robusto y estable, se encuentra bien documentado y ya es conocido y manejado por el equipo de trabajo, lo cual nos reduce la carga generada por la curva de aprendizaje necesaria para adaptar una herramienta desconocida al equipo de trabajo.

***Cliente***

Para acceder al repositorio, cada miembro usara como cliente del sistema Subversion, la aplicación **TortoiseSVN.**

Se usara esta aplicación puesto que es una la misma integra completamente con el sistema operativo Windows. Igualmente se encuentra ampliamente documentada y su modo de uso es bastante sencillo e intuitivo.

La aplicación se debe descargar (ver referencias) en su versión adecuada (32 o 64 bits). Una vez instalada se debe proceder a obtener una copia local del repositorio, después de reiniciar le equipo, paso absolutamente necesario para finalizar la instalación del cliente y lograr su integración con el sistema operativo

Para acceder al repositorio a través del cliente de TortoiseSVN, se debe realizar lo siguiente

Crear un directorio en el cual se creara la copia local del repositorio

Click derecho sobre el repositorio y en el menú contextual, click sobre la opción **SVN Checkout**

En el formulario de checkout ingresar la dirección del repositorio, <https://ingenium-managment.googlecode.com/svn/trunk> y Aceptar

Al final del proceso se creara una copia local de nuestro repositorio, el cual podremos manipular según se requiera y una vez se finalizado el trabajo se realizara la actualización en el repositorio

Para actualizar las copias locales, se debe dar click derecho sobre el directorio en el cual hemos configurado nuestro repositorio y dar click en la opción **SVN Update**

Para subir nuestros cambios al repositorio, se debe dar click derecho sobre el archivo o directorio que se enviará al repositorio y dar click en la opción **SVN Commit.** Al hacer Commit en nuestro repositorio, es probable que el cliente solicite el usuario y la contraseña, en caso de que estos datos no se hayan ingresado previamente.

Si no se tienen estos datos se debe ingresar a la administración del proyecto en Google Code, autenticarse con el usuario de Google Accounts, y posteriormente dirigirse a la pestaña **Source** del proyecto. Una vez ahí, vamos a la opción [*googlecode.com password*](http://code.google.com/hosting/settings), la cual nos presentar la contraseña autogenerada y también nos da la opción de generar una nueva contraseña individual en caso de que sea requerido*.*

* + 1. **Referencias**

A continuación se encuentran las referencias para la descarga o información general de las herramientas que serán utilizadas

* URL del proyecto: <http://code.google.com/p/ingenium-managment/>
* JRE 1.6:

<https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_Developer-Site/en_US/-/USD/ViewProductDetail-Start?ProductRef=jre-6u24-oth-JPR@CDS-CDS_Developer>

* Entorno de desarrollo Eclipse: <http://www.eclipse.org/downloads/>
* Cliente de Subversion TortoiseSVN: <http://tortoisesvn.net/downloads.html>
* Documentación del cliente de TortoiseSVN:

<http://cdnetworks-us-1.dl.sourceforge.net/project/tortoisesvn/1.6.13/Documentation/TortoiseSVN-1.6.13-en.pdf>

* 1. **Guía de Implementación**

Para el desarrollo del producto, los ingenieros de desarrollo involucrados deberán seguir un estándar común. A continuación se presentan algunos ejemplos tipo.

1. Ejemplo definición de una clase

/\*\*

\* Es la clase que representa a un perro. <br>

\*/

**public** **class** Perro

{

}

1. Ejemplo definición de un método con anotaciones

/\*\*

\* Compara dos perros según el nombre. <br>

\* **@param** p es el perro contra el que se está comparando - p !=null

\* **@return** Retorna 0 si los perros tienen el mismo nombre. <br>

\* Retorna -1 si el perro p tiene una valor "MAYOR" para el nombre. <br>

\* Retorna 1 si el perro p tiene una valor "MENOR" para el nombre. <br>

\*/

@LogTList({

@LogT(taskId = "30", date = "21/02/2011 00:45", min = 10, cicloId=1, responsableId=3)})

@LogIntList({

LogInt(date=”21/02/2011 00:52”, min=2, cicloId=1, responsableId=3, intId=1 )})

@LocList({

@Loc(size=”20”, cicloId=1, responsableId=3)})

@LogDList({

@LogD(date=”21/02/2011 01:45”, stage=”codificación”, cicloId=1, responsableId=3)})

**public** **int** compararPorNombre( Perro p )

{

**//Definición del método**

}

Cada una de las anotaciones cumple con la siguiente funcionalidad:

* **Anotación LogT:** Registra el tiempo tomado para ejecutar una tarea determinada. Se registra la fecha, tiempo en minutos, identificador del ciclo, identificador de la tarea realizada y id del responsable que ejecuta la tarea.
* **Anotación LogInt:** Registra el tiempo de las interrupciones que se generaron durante la realización de una tarea. Requiere fecha, tiempo en minutos, identificador del ciclo y responsable de la interrupción.
* **Anotación LogD:** Registra los defectos encontrados en el producto. Registra fecha, fase, identificador del ciclo e identificador del responsable.
* **Anotación Loc:** Registra el número de líneas de código que una persona generó durante la ejecución de una tarea.
  1. **Aplicativo Desarrollado**

El objetivo de este ciclo era realizar las modificaciones necesarias en el aplicativo de PSP para que pudiera soportar el manejo de la aplicación de TSP, adicionando información sobre el ciclo y el responsable.

La siguiente información es la que se muestra por consola una vez se ejecuta el aplicativo para analizar el proyecto de TSP.

Se analizaron los siguientes archivos:

com.ingenium.tsp.annotations.Constants

com.ingenium.tsp.annotations.Loc

com.ingenium.tsp.annotations.LocList

com.ingenium.tsp.annotations.LogD

com.ingenium.tsp.annotations.LogDList

com.ingenium.tsp.annotations.LogInt

com.ingenium.tsp.annotations.LogIntList

com.ingenium.tsp.annotations.LogT

com.ingenium.tsp.annotations.LogTList

com.ingenium.tsp.annotations.Plan

com.ingenium.tsp.annotations.PlanQ

com.ingenium.tsp.control.Analizer

com.ingenium.tsp.control.IngeniumTSP

com.ingenium.tsp.control.Report

com.ingenium.tsp.model.Person

com.ingenium.tsp.report.LocRecord

com.ingenium.tsp.report.LogIntRecord

com.ingenium.tsp.report.LogTRecord

com.ingenium.tsp.report.Record

Interrupciones

----------------------------------------------------------------------------------

| Ciclo | Responsable | Interrupcion | Fecha | Duracion |

----------------------------------------------------------------------------------

| c1 | Erik | Skype | 14/03/2011 | 25 |

| c1 | Willian | Internet | 09/03/2011 | 10 |

| c1 | David | Internet | 09/03/2011 | 15 |

| c1 | David | Cena | 10/03/2011 | 50 |

| c1 | David | Cena | 10/03/2011 | 50 |

| c1 | David | Internet | 11/03/2011 | 40 |

| c1 | David | Cena | 14/03/2011 | 35 |

| c1 | Carlos | Baño | 09/03/2011 | 10 |

| c1 | Mauricio | LlamadaTelefonica | 14/03/2011 | 30 |

| c1 | Mauricio | LlamadaTelefonica | 13/03/2011 | 40 |

| c1 | Carlos | PedirComida | 13/03/2011 | 10 |

| c1 | Carlos | PedirComida | 14/03/2011 | 10 |

----------------------------------------------------------------------------------

| | | | | 325 |

----------------------------------------------------------------------------------

Total de tareas realizadas

------------------------------------------------------------------------

| Ciclo | Responsable | Tarea | Fecha | Duracion |

------------------------------------------------------------------------

| c1 | David | 1.2.2 | 11/03/2011 | 30 |

| c1 | David | 1.2.3 | 14/03/2011 | 20 |

| c1 | David | 1.2.1 | 09/03/2011 | 120 |

| c1 | David | 1.3.2.4 | 14/03/2011 | 139 |

| c1 | David | 9999 | 14/03/2011 | 60 |

| c1 | Erik | 1.2.1 | 09/03/2011 | 30 |

| c1 | Erik | 1.3.1.1 | 12/03/2011 | 25 |

| c1 | Erik | 1.3.2.1 | 09/03/2011 | 20 |

| c1 | Erik | 1.3.2.2 | 13/03/2011 | 60 |

| c1 | Erik | 1.3.5.1 | 14/03/2011 | 90 |

| c1 | Erik | 1.3.5.2 | 14/03/2011 | 15 |

| c1 | Mauricio | 1.2.1 | 09/03/2011 | 120 |

| c1 | Mauricio | 1.2.3 | 12/03/2011 | 60 |

| c1 | Mauricio | 1.3.1.3 | 14/03/2011 | 180 |

| c1 | Mauricio | 1.3.1.4 | 12/03/2011 | 60 |

| c1 | Mauricio | 1.3.1.5 | 12/03/2011 | 30 |

| c1 | Mauricio | 1.3.2.5 | 12/03/2011 | 30 |

| c1 | Mauricio | 1.3.3.2 | 12/03/2011 | 15 |

| c1 | Mauricio | 1.3.3.4 | 12/03/2011 | 60 |

| c1 | Mauricio | 1.3.7.1 | 12/03/2011 | 15 |

| c1 | Sandra | 1.2.1 | 09/03/2011 | 150 |

| c1 | Sandra | 1.2.3 | 14/03/2011 | 145 |

| c1 | Sandra | 1.3.1.4 | 12/03/2011 | 20 |

| c1 | Sandra | 1.3.2.3 | 13/03/2011 | 30 |

| c1 | Willian | 1.2.1 | 12/03/2010 | 50 |

| c1 | Willian | 1.2.3 | 12/03/2010 | 30 |

| c1 | Willian | 1.3.1.1 | 09/03/2011 | 180 |

| c1 | Willian | 1.3.1.4 | 12/03/2011 | 30 |

| c1 | Willian | 1.3.1.5 | 12/03/2011 | 15 |

| c1 | Willian | 1.3.2.1 | 09/03/2011 | 20 |

| c1 | Willian | 1.3.3.1 | 11/03/2011 | 150 |

| c1 | Willian | 1.3.3.2 | 12/03/2011 | 15 |

| c1 | Willian | 1.3.4.1 | 12/03/2011 | 20 |

| c1 | Willian | 1.3.4.2 | 11/03/2011 | 30 |

------------------------------------------------------------------------

| | | | | 2064 |

------------------------------------------------------------------------

Total de lineas de codigo

-------------------------------------------------------------------------------------

| Clase | Metodo | LOC |

-------------------------------------------------------------------------------------

| com.ingenium.tsp.control.Analizer | checkAnnotation | 15 |

| com.ingenium.tsp.control.Analizer | getAccessibleObjectName | 4 |

| com.ingenium.tsp.control.IngeniumTSP | IngeniumTSP() | 1 |

| com.ingenium.tsp.control.IngeniumTSP | checkProject | 20 |

| com.ingenium.tsp.control.IngeniumTSP | loadFileList | 7 |

| com.ingenium.tsp.control.IngeniumTSP | main | 2 |

| com.ingenium.tsp.control.Report | printLogInt | 10 |

| com.ingenium.tsp.control.Report | printLogT | 15 |

| com.ingenium.tsp.control.Report | printLoc | 10 |

-------------------------------------------------------------------------------------

| | | 84 |

-------------------------------------------------------------------------------------

Productividad Real: TotalLOC = 84, Total Tiempo = 2064 por tanto: 2,44 LOC/Hora

1. **PostMortem Primer Ciclo**
   1. **Producto**
   2. **Reporte del Ciclo**
   3. **Proceso de TSP**
   4. **Equipo**
   5. **Herramientas**
   6. **Evaluación**
   7. **Propuestas de mejoramiento**

1. A Discipline for Software Engineering, page 117, by Watts S. Humphrey, Addison Wesley Publishing Co., 1995. [↑](#footnote-ref-1)