

UNIVERSIDAD NACIONAL DEL ESTE
FACULTAD POLITÉCNICA

TESIS DE
MAESTRÍA EN INFORMÁTICA Y COMPUTACIÓN

**“DESARROLLO DE UNA NUEVA
METODOLOGÍA DERIVADA DEL
COCOMO II AGREGANDO PAUTAS DE
CALIDAD”**

Elaborada por:

Obdulia Lorena Franco Araujo

Julio/2014

Franco Araujo, O. L. (2014). DESARROLLO DE UNA NUEVA
METODOLOGÍA DERIVADA DEL COCOMO II AGREGANDO
PAUTAS DE CALIDAD.

Lorena Franco. 151 páginas.

Tutor: Magister Oscar Adolfo Vallejos.

Co-Tutor: Dr. Daniel Romero.

Tesis de Maestría en Informática y Computación

Universidad Nacional del Este, 2014

Obdulia Lorena Franco Araujo

**“DESARROLLO DE UNA NUEVA METODOLOGÍA
DERIVADA DEL COCOMO II AGREGANDO PAUTAS
DE CALIDAD”**

Esta tesis fue evaluada y aprobada para la obtención del título de Máster en Informática y Computación por la Universidad Nacional del Este.

Mesa Examinadora:

Dedicatoria:

Esta tesis la dedico a:

A mi hijo Félix Gabriel por ser la luz de mis ojos.

A Félix, por el apoyo incondicional y su comprensión.

A mi Mamá, por haberme dado todo lo necesario para llegar hasta aquí, y mucho más.

Agradecimientos:

Principalmente a JESÚS por darme salud y sabiduría para concluir la tesis.

Al Dr. David La Red Martínez, por sus aportes en los lineamientos, sus comentarios y el tiempo dedicado en la lectura de versiones preliminares.

Al Dr. Daniel Romero, en su carácter de Co-Tutor del presente trabajo, por su buena disposición, aportes y comentarios al mismo.

A la Fundación Parque Tecnológico de Itaipú, en especial a los compañeros de trabajo y al Lic. Luis Benítez - Coordinador del Área de Datacenter por la disponibilidad de los datos para estudio y consultas realizadas.

Al Magister Oscar Vallejos, en su carácter de Tutor, por sus ideas y recomendaciones.

A los miembros de la Dirección de Investigación de la Facultad Politécnica por el apoyo constante y la fuerza necesaria para lograr el objetivo.

A la Conacyt por el apoyo económico necesario para cursar la maestría.

Lorena Franco

Tabla de contenido

1. Capítulo 1	13
1.1. Introducción.....	13
1.2. Descripción de la Problemática	14
1.1.1. Enunciado Preciso del Eje de la Situación Problemática	15
1.2. Hipótesis y Objetivos	16
1.2.1. Hipótesis	16
1.2.2. Objetivo General:.....	16
1.2.3. Objetivos Específicos:.....	16
1.3. Marco Teórico	17
1.3.1. Concepto de Metodología	17
1.3.2. Estimación de Costo para Desarrollo de Software	17
1.3.3. Proceso de Estimación	20
1.3.4. Medidas y Métricas del Software.....	22
1.3.5. Métodos de Estimación de Costos Existentes.....	24
1.3.6. Modelos Existentes.....	26
1.4. Metodología	28
1.4.1. Dimensión de la Estrategia General.....	28
1.5. Estructura de la Tesis.....	30
1.6. Discusiones y Comentarios.....	30
2. Capítulo 2	32
2.1. COCOMO II.....	32
2.2. Estimación con COCOMO II	32
2.3. Modelo de Composición de Aplicaciones.....	34
2.3.1. Introducción a los Puntos Objeto	35
2.4. Modelos de Diseño Temprano y Arquitectura Tardía	37
2.4.1. Componentes de los Modelos.....	38
2.4.2. Modelo de Mantenimiento de las Aplicaciones.....	42

2.5.	Discusiones y Comentarios.....	44
3.	Capítulo 3	45
3.1.	Metodologías Ágiles	45
3.2.	El Manifiesto Ágil.....	45
3.3.	Metodologías Ágiles de Desarrollo más Exitosas.....	46
3.3.1.	Scrum.....	46
3.3.2.	XP (Extreme Programming).....	50
3.3.3.	Kanban	58
3.3.4.	Scrumban.....	59
3.3.5.	Otras Metodologías Ágiles.....	61
3.4.	Comentarios y Discusiones.....	62
4.	Capítulo 4 Propuesta Metodológica	63
4.1.	Introducción.....	63
4.2.	Planificación.....	64
4.2.1.	Etapa 1 – Análisis de Parámetros.....	65
4.2.2.	Etapa 2 – SCRUM, como Gestión del Proyecto.....	65
4.2.3.	Etapa 3 - COCOMO II, como Pauta de Calidad.....	72
4.2.4.	Etapa 4 – Definición de la Propuesta	74
4.2.5.	Etapa 5 – Formulas y Cálculos Resultantes	75
4.2.1.	Etapa 6 – Evaluación de la Propuesta.....	78
4.3.	Discusiones y Comentarios.....	80
5.	Capítulo 5 Pruebas Experimentales.....	81
5.1.	Introducción.....	81
5.2.	Casa Práctico 1	82
5.3.	Caso Práctico 2.....	82
5.4.	Resultados Obtenidos	82
5.4.1.	Caso Práctico 1	82
5.4.2.	Caso Práctico 2.....	83
5.5.	Lecciones Aprendidas.....	85
5.6.	Ventajas Obtenidas con la Nueva Metodología.....	86

6. Capítulo 6 Conclusiones y Recomendaciones y Trabajos Futuros.....	87
6.1. Conclusiones	87
6.2. Recomendaciones	88
6.3. Trabajos Futuros	89
7. Bibliografía	90
8. Anexo	92

Índices de Ilustraciones

Ilustración 1 Estadística de uso de estimación de costos de desarrollo de software en CDE (fuente propia)	15
Ilustración 2 Arquitectura COCOMO II (Boehm B., 2000).....	33
Ilustración 3 Gráfica burndown (Cockburn, 2007).....	50
Ilustración 4 Ciclos XP (Wells, 2013).....	51
Ilustración 5 Proyecto XP (Wells, 2013).....	56
Ilustración 6 Muro Kanban (Mendes Calo, 2010).....	58
Ilustración 7 Historia de Usuarios (Schwaber K., 2001)	67
Ilustración 8 Planning poker (Cockburn, 2007).....	68
Ilustración 9 Pila de Producto & Sprint 1 (fuente propia).....	72
Ilustración 10 Nueva Arquitectura (fuente propia).....	74
Ilustración 11 Factor TEAM (fuente propia).....	84
Ilustración 12 Factor MATURITY (fuente propia).....	84
Ilustración 13 Multiplicador de Esfuerzo PLEX (fuente propia).....	85
Ilustración 14 PF - Variable de Conteo del Módulo de Bienes (fuente propia).....	97
Ilustración 15 PF - Variable de Conteo del Módulo de Compras (fuente propia).....	104
Ilustración 16 PF - Variable de Conteo del Módulo de Contabilidad (fuente propia).....	109
Ilustración 17 PF - Variable de Conteo del Módulo de Documentos (fuente propia).....	117
Ilustración 18 PF - Variable de Conteo del Módulo de Presupuesto (fuente propia).....	121
Ilustración 19 PF - Variable de Conteo del Módulo de RRHH (fuente propia).....	132
Ilustración 20 Variable de Conteo del Módulo de Tesorería (fuente propia).....	142
Ilustración 21 Resultado COCOMO II.2000.4 (Boehm B., 2000).....	143
Ilustración 22 Plantilla1 - Datos Iniciales (fuente propia).....	144
Ilustración 23 Plantilla 2 - Datos del Equipo (fuente propia).....	144
Ilustración 24 Plantilla 3 - Pila de Producto (fuente propia).....	148
Ilustración 25 Plantilla 5 - Factores de Escala (fuente propia).....	149
Ilustración 26 Plantilla 5 - Multiplicadores de Esfuerzo (fuente propia).....	150
Ilustración 27 Plantilla 6 - Calculo de Costos (fuente propia).....	151

Índices de Tablas

Tabla 1 Descripción de modelos de estimación del tamaño del software (Clemmons, 2006).....	27
Tabla 2 Descripción de modelos de gestión de costos (Clemmons, 2006).	27
Tabla 3 Descripción de modelos de gestión de riesgos (Clemmons, 2006).....	28
Tabla 4 Complejidad asociada a instancias de objetos (Longstreet, 2004).....	36
Tabla 5 Tipo Objeto / Pesos Asociados (Longstreet, 2004).....	36
Tabla 6 Ratio de Productividad (Longstreet, 2004).....	37
Tabla 7 Puntos de Función sin ajustar al equivalente SLOC (Banerjee, 2001).....	39
Tabla 8 Conceptos de los Factores de escala para el Modelo de Cocomo II de Diseño Temprano pertenecientes a la versión USC-COCOMOII.2000 (B. Boehm, 2000)	41
Tabla 9 Valores de los Factores de escala para el Modelo de Cocomo II de Diseño Temprano pertenecientes a la versión USC-COCOMOII.2000 (B. Boehm, 2000)	41
Tabla 10 Drivers de coste (B. Boehm, 2000)	42
Tabla 11 Scrum vs Scrumban (Mendes Calo, 2010).	60
Tabla 12 Parámetros de COCOMO & SCRUM (fuente propia).	65
Tabla 13 Pila de Producto (fuente propia).	66
Tabla 14 Velocidad del Equipo (fuente propia).....	71
Tabla 15 Características evaluadas (fuente propia).	79
Tabla 16 Tabla de valores de factores y multiplicadores (fuente propia).	83

Resumen

El objetivo de este trabajo es ofrecer una metodología de estimación de costos basada en la metodología COCOMO II (Boehm B., 2000) que sea aplicable a proyectos de desarrollo de software que utilizan métodos ágiles de desarrollo (Kamaruddin, 2012). Este trabajo permitirá aportar mejoras en la estimación de costos de proyectos ágiles considerando la productividad del equipo y ofreciendo técnicas flexibles en el desarrollo de un producto de software.

COCOMO II (Milicic, 2004) es un modelo de estimación de costos referente en el mercado, pero la aplicación del mismo es poco efectiva en ambientes de desarrollo caóticos (Ahmed E. Hassan, 2003) compuestos por equipos de desarrollo inmaduros así como en proyectos que siguen una metodología ágil de desarrollo.

En este trabajo se espera establecer un modelo práctico de estimación de costos que incorpore las características consideradas en los proyectos ágiles (Pressman R., 2005).

En la actualidad se observa la importancia de la estimación de costos de proyectos ágiles debido al auge que estas metodologías están teniendo en el mercado nacional e internacional.

Abstract

The aim of this paper is to provide an estimation methodology based on the COCOMO II (Boehm B., 2000) cost methodology applicable to software development project using agile methods development. This item will bring improvements in the estimation of considering costs agile projects and team productivity providing flexible techniques for the development of product software.

COCOMO II is a model of cost estimating reference in the market, but its application is very effective in environments chaotic development (Ahmed E. Hassan, 2003) consisting of development teams immature as well as projects that follow agile methodology development .

This paper hopes to establish a practical model estimation cost to incorporate the features seen in the agile projects (Pressman R., 2005).

Today we see the importance of cost estimation agile projects due to the boom that these methodologies are having in the national and international market.

Capítulo 1

1.1. Introducción

La creciente complejidad de los desarrollos software que provocó la denominada "crisis del software" se ha tratado de abordar mediante el planteamiento de nuevos métodos, metodologías, técnicas y paradigmas para minimizar su impacto.

El alcance de dichas propuestas no se limita exclusivamente a actividades relacionadas con el desarrollo en sí de los sistemas, sino que abarca también las actividades de gestión de los mismos. Una de estas actividades es la de la estimación de los proyectos software.

A pesar de que la estimación de proyectos continúa siendo una tarea muy compleja, en muchas ocasiones dejada al albur de la pericia del experto estimador, en las últimas décadas se han desarrollado algunas técnicas para la estimación del esfuerzo de proyectos software completos, tales como casos de usos (Banerjee, 2001), puntos de función (Longstreet, 2004) y COCOMO II (Boehm B., 2000).

Aun así, estas técnicas si bien se postulan como independientes de la tecnología final de desarrollo fueron concebidas para su aplicación en sistemas basados en el paradigma estructurado con un ciclo de vida clásico o en cascada, y aún es difícil emplearlas en metodologías ágiles de desarrollos y ciclos de vida iterativo-incrementales utilizados en los últimos años. Incluso, parece interesante que estas técnicas de estimación exploten para sus propósitos la información proporcionada por prácticas muy extendidas últimamente, como, por ejemplo, la de los casos de uso (Clemmons, 2006).

Como se ve en un ambiente de que todo debe ser para ayer, se torna caótico el desarrollo de software y de ahí surgen las metodologías ágiles para poder cubrirlo, sin embargo encontrar una combinación de metodología de desarrollo, método de

estimación de esfuerzo y costo todavía es un trabajo minucioso y de ahí que surge el área de interés para la elaboración de esta tesis.

1.2. Descripción de la Problemática

Según las estadísticas publicadas por el Standish Group (Standish Group), las cuales revelan que sólo el 12% de todos los proyectos de software a nivel mundial son entregados a tiempo y dentro del presupuesto planificado.

Las técnicas tradicionales de ingeniería se han adaptado al campo de desarrollo de software precisamente para proveer herramientas que permitan gestionar los proyectos, administrar los requerimientos y riesgos, así como crear métricas confiables para hacer estimaciones acertadas de tiempo y costo; sin embargo, ha creado gran asombro ver las estadísticas antes mencionadas a pesar de que la ingeniería del software existe hace varias décadas.

En la actualidad, los requerimientos del software cambian constantemente. Esto así, porque las empresas cambian, y junto con ellas debe de cambiar su tecnología. Si no se hace una eficiente administración de los requerimientos puede convertirse el proyecto en un bucle sin final hasta que finalmente éste es cancelado.

Las estimaciones de costo y el horario en proyectos de software se basan en una predicción del tamaño del sistema futuro. Por desgracia, en el área del software es notoriamente inexacta la estimación de costos y programación.

Las estimaciones preliminares de esfuerzo siempre incluyen muchos elementos de inseguridad. Las primeras estimaciones fiables son difíciles de obtener debido a la falta de información detallada sobre el futuro sistema en una etapa temprana. Sin embargo, se requiere estimaciones iniciales al hacer una oferta para un contrato o determinar si un proyecto es factible en los términos de un análisis de costo-beneficio, como proceso de predicción y guías de toma de decisiones. Una predicción sólo es útil si es razonablemente exacta (Dekkers, 2003).

1.1.1. Enunciado Preciso del Eje de la Situación Problemática

La estimación de lo que costará el desarrollo de software es una de las actividades de planeación más importantes y se debe saber si el desarrollo del producto de software será beneficioso y se necesita velocidad y agilidad a la hora de elaborar el cálculo de estimación.

Los modelos de costos como COCOMO y Puntos de Función (FPA) (Longstreet, 2004) son bien conocidos y de uso generalizado en la ingeniería de software. Sin embargo, con las actuales metodologías ágiles ninguno de estos enfoques es adecuado para el dimensionamiento del desarrollo de software por el uso de líneas de código como medida de tamaño de software, pues existen un sinnúmero de paradojas en cuanto a este tema (Jones C. , 2007).

En nuestro país existe un alto promedio de sobrecostos en los que incurren grandes empresas y la experiencia ha demostrado que este alto índice de fracaso en los proyectos de software está relacionado a fechas de entrega inalcanzables dadas sin utilizar métrica alguna para llegar a una estimación eficiente aproximada.

En Ciudad del Este (CDE) se puede decir que sólo el 5% de las empresas que desarrollan software utilizan algún método de estimación de costos como indica la Ilustración 1.

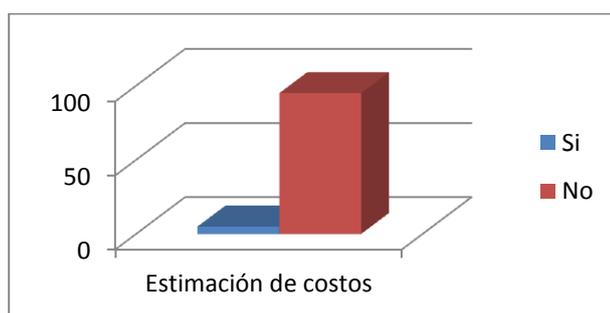


Ilustración 1 Estadística de uso de estimación de costos de desarrollo de software en CDE (fuente propia).

De ahí los problemas de atrasos en los cronogramas y sobrecostos en los proyectos de desarrollo de software. Estos problemas ocurren en grandes, medianas y pequeñas empresas.

1.2. Hipótesis y Objetivos

1.2.1. Hipótesis

Es posible estimar el costo de un equipo de desarrollo de software de manera sencilla y rápida aplicando metodologías de desarrollo ágiles conforme a estándares de calidad.

1.2.2. Objetivo General:

- ✓ Desarrollo de una propuesta metodológica de estimación de costos incorporando pautas de calidad.

1.2.3. Objetivos Específicos:

- ✓ Simplificar y disminuir los parámetros o criterios utilizados en COCOMO II para medición del desarrollo de software.
- ✓ Ofrecer métodos sencillos y prácticos para obtener calidad en la estimación de un desarrollo de software.
- ✓ Desarrollar una metodología de estimación de costos de software para la verificación del nuevo modelo propuesto en este trabajo de investigación en base al COCOMO II.

1.3. Marco Teórico

1.3.1. Concepto de Metodología

Como se ve a continuación, tanto en los objetivos planteados en esta tesis como en la hipótesis que se pretende demostrar, se hace uso de la palabra "metodología", término cuyo significado tal vez resulte demasiado amplio y demasiado vago. Por esta razón, algunos autores la definen así: "*Una metodología debe disponer del soporte necesario para cubrir el ciclo de vida completo desde los puntos de vista técnico y de gestión*" (Dekkers, 2003) (en este caso, el modelo de procesos del ciclo de vida se limitaría a la provisión de un modelo para la estimación de costos en la etapa de implementación del desarrollo de software).

Entre los elementos con que debe contar una metodología, los autores citan:

- ✓ Un modelo de procesos de ciclo de vida.
- ✓ Un conjunto de técnicas.
- ✓ Un conjunto de entregables.
- ✓ Un conjunto de métricas.
- ✓ Guías para la gestión del proyecto, incluyendo roles y estructura del equipo.
- ✓ Herramientas.

Con esta definición, el concepto de metodología continúa siendo amplio pero encierra un significado mucho más preciso. Se ha hecho la siguiente distinción dentro de las categorías de técnicas y métricas:

- ✓ Técnicas de estimación de recursos, de estimación de esfuerzo y de identificación y estimación de riesgos.
- ✓ Métricas de producto y métricas de proceso.

1.3.2. Estimación de Costo para Desarrollo de Software

El producto de software ha tenido una muy alta frecuencia de exceso de calendarización, costos, problemas de calidad e indiscutibles cancelaciones. Mientras esta mala reputación a menudo es merecida, es importante notar que algunos grandes proyectos de software finalizan a tiempo y con el presupuesto estimado, además funcionan satisfactoriamente cuando éstos son desplegados.

Los grandes proyectos exitosos difieren en muchos sentidos de los fracasos y desastres (Capers, 2004). Una diferencia importante es, cómo los proyectos exitosos

concluyen a tiempo, dentro de los costos, recursos y estimación de calidad prevista en un primer término. De un análisis de resultados de las herramientas de estimación usadas, publicado en (Jones C. , 2008), el uso de instrumentos de estimación automatizados conduce a estimaciones más exactas. A la inversa, los métodos informales o manuales de llegar en estimaciones iniciales son generalmente inexactos y a menudo excesivamente optimistas.

Uno de los problemas encontrados es el hecho de que muchos proyectos grandes con estimaciones imprecisas fueron cancelados sin haber culminado. De ese modo, para que los proyectos estuviesen incluidos en todo, ellos tendrían que haber finalizado. Este criterio eliminó muchos proyectos que usaron tanto estimación manual como automatizada.

De modo interesante, las estimaciones manuales y las automatizadas eran equitativamente cercanas en términos de predicción del esfuerzo de programación o codificación. Pero las estimaciones fueron muy optimistas cuando predecían el crecimiento de los requerimientos, esfuerzo del diseño, esfuerzo de documentación, esfuerzo de administración, esfuerzo de evaluación y esfuerzo de revisión y reparación. La conclusión de la comparación fue, que tanto las estimaciones manuales como las automatizadas eran equivalentes para la programación real, pero las estimaciones automatizadas eran mejores para predecir actividades de no codificación.

Las estimaciones de tiempo y costo deberían ser exactas, naturalmente. Pero si ellas difieren de los resultados reales, es más seguro ser ligeramente conservador que ser optimista. Una de las principales quejas sobre los proyectos de software se refiere a su tendencia alarmante de exceder gastos y calendarios planificados. Desafortunadamente, tanto clientes como ejecutivos superiores tienden a ejercer presiones considerables en los administradores de proyectos y en el personal encargado de realizar las estimaciones. Por consiguiente un corolario oculto de estimación acertada es aquél en donde éstas deben ser defendibles. La mejor defensa es una buena colección de datos históricos de proyectos similares.

Debido a que el crecimiento de la estimación de costos es una actividad compleja, existe un crecimiento industrial de compañías dedicadas a ofrecer diferentes marcas comerciales de herramientas de estimación de costos en el mercado. A partir del 2005, algunas de esas herramientas de estimación son: COCOMO II, CoStar, CostModeler, CostXpert, KnowledgePlan®, PRICE S, SEER, SLIM y SoftCost. Algunas de las herramientas de estimación de costos más antiguas, no están activamente en el

mercado pero todavía son utilizadas, tales como: CheckPoint, COCOMO, ESTIMACS, REVIC y SPQR/20 (Jones C. , 2008), ya que su uso no es apoyado por los vendedores, por lo que su utilización está en declive.

Mientras estos instrumentos de estimación fueron desarrollados por empresas diferentes y no son idénticos, ellos realmente tienden a proporcionar un núcleo de funciones comunes. Los rasgos principales de instrumentos de estimación de software comerciales incluyen estos atributos:

- ✓ Lógica de dimensionamiento para especificaciones, código fuente y casos de evaluación.
- ✓ Nivel de fase, nivel de actividad, y estimación de nivel de tarea.
- ✓ Ajustes para períodos de trabajo específicos, vacaciones y horas extraordinarias.
- ✓ Ajustes para salarios locales e índices de carga.
- ✓ Ajustes para varios proyectos de software como militares, sistemas, comerciales, etc.
- ✓ Apoyo para métricas de puntos de función, métricas de líneas de códigos o ambas.
- ✓ Apoyo para “backfiring”, es decir, convertir líneas de códigos a puntos de función.
- ✓ Apoyo tanto para nuevos proyectos como a proyectos de mejora y mantenimiento.

Algunas herramientas de estimación también incluyen funciones más avanzadas como:

- ✓ Estimación de fiabilidad y calidad.
- ✓ Análisis de valor y riesgo.
- ✓ Retorno de Inversión.
- ✓ Posibilidad de compartir datos con herramientas de administración de proyectos.
- ✓ Medios de medición para reunir datos históricos.
- ✓ Costo y tiempo para completar estimaciones que combinan datos históricos con datos proyectados.
- ✓ Apoyo para evaluaciones de procesos de software.

- ✓ Análisis estadístico de múltiples proyectos y análisis de cartera.
- ✓ Conversión monetaria para acordar proyectos en el exterior.

1.3.3. Proceso de Estimación

En la industria en general, es necesario calcular y estimar el esfuerzo y el tamaño del proyecto en etapas muy tempranas del desarrollo del mismo. Sin embargo, si en el ámbito software se hacen las estimaciones en estas fases iniciales, dichas previsiones pueden estar basadas en unos requerimientos erróneos o incompletos, por lo que disminuye mucho su fiabilidad.

El proceso de estimación del coste de un producto software está formado por un conjunto de técnicas y procedimientos que se usan en la organización para poder llegar a una predicción fiable. Éste es un proceso continuo, que debe ser usado y consultado a lo largo de todo el ciclo de vida del proyecto. Se divide en los siguientes pasos (Agarwal, 2001).

- ✓ Estimación del tamaño.
- ✓ Estimación del costo y del esfuerzo.
- ✓ Estimación de la programación temporal.
- ✓ Estimación de la cantidad de recursos computacionales.
- ✓ Asunción de riesgos.
- ✓ Inspección y aprobación.
- ✓ Redacción de informes de estimación.
- ✓ Medición y perfeccionamiento del proceso.

De una manera general se puede afirmar que existen dos maneras diferentes de estimar el presupuesto y el tiempo para un proyecto software: usando modelos de costo y usando razonamiento basado en similitud. En ambas opciones es necesario recurrir a información histórica y de proyectos anteriores previamente almacenados en bases de datos.

Existen cuatro puntos fundamentales sobre los que se apoya la estimación:

- ✓ Las consideraciones y opiniones de los profesionales de la materia, basada en la experiencia y la madurez de los gestores de proyecto, los cuales tendrán que adivinar y predecir el tiempo de realización del proyecto o su coste.
- ✓ La participación de expertos, cuyas opiniones no deben ser consideradas y abordadas como las de los profesionales y gestores de proyecto, ya que los expertos

no pertenecen a la organización y pueden estar no familiarizados con las prácticas propias de la organización.

- ✓ La utilización de factores estándar de tiempos, calculados y establecidos a partir de proyectos anteriores.
- ✓ Por último el empleo de fórmulas y funciones, que implica la existencia de datos cuantitativos que representen una buena aproximación a la estimación. Además no deben existir dudas acerca de la fiabilidad y seguridad del producto usado.

Por lo tanto, con el fin de gestionar presupuesto y calendario de los proyectos de software, se han desarrollado varios modelos de estimación de costos de software. La estimación de costos de software exactos es fundamental para los desarrolladores y los clientes (Jones C. , 2007). Pueden ser utilizados para la generación de solicitud de propuestas, negociaciones de contratos, la programación, el seguimiento y control. Estimación de costes precisa es importante debido a las siguientes razones (Leungh, 2001):

- ✓ Puede ayudar a clasificar y priorizar los proyectos de desarrollo con respecto a un plan de negocios general.
- ✓ Se puede utilizar para determinar cuáles son los recursos que se comprometan con el proyecto y lo bien que se puede utilizar estos recursos.
- ✓ Se puede utilizar para evaluar el impacto de los cambios y el apoyo planificación previa.
- ✓ Los proyectos pueden ser más fáciles de gestionar y controlar que los recursos están mejor adaptados a las necesidades reales.
- ✓ Los clientes esperan que los costos de desarrollo reales sean acordes con los costos estimados.

Estimación de costos de software ha sido históricamente una de las principales dificultades en el desarrollo de software. Varias razones han sido identificado que afecta el proceso de estimación, tales como:

- ✓ Es muy difícil estimar el costo de desarrollo de software. Uno de los primeros pasos en cualquier estimación es entender y definir el sistema a ser estimado.
- ✓ Una estimación de costos de software hecho al principio del ciclo de vida del proyecto se basa generalmente en las entradas menos precisas y menos especificaciones de diseño detalladas.

- ✓ El desarrollo de software que implica muchos factores interrelacionados que afectan a actividades de desarrollo y la productividad, y cuyas relaciones no son bien comprendidas.
- ✓ La falta de una base de datos histórica de medición de costos que significa que los datos históricos son a veces incompletos, inconsistentes o incorrectos.
- ✓ La falta de peritos capacitados y estimadores con la experiencia necesaria.
- ✓ Sin embargo, el software es intangible, invisible, y de difícil solución por lo que es más difícil de entender y estimar un producto o proceso que no se pueden ver y tocar.
- ✓ Si bien las estimaciones demasiado poco esfuerzo pueden llevar a problemas de gestión de proyectos, entregas retrasadas, excesos presupuestarios y la baja calidad del software, las estimaciones demasiado altas esfuerzo puede dar lugar a oportunidades de negocio perdidas y poco eficientes uso de los recursos.

Otros factores que afectan el costo son la capacidad del programador, la experiencia de la superficie del desarrollador, la complejidad del proyecto y los requisitos de fiabilidad, etc. La estimación se ajusta de acuerdo a una serie de factores de coste (por ejemplo, la experiencia del personal y la complejidad de sistema) para llegar al estado final. Las restricciones financieras limitan la cantidad de dinero que puede ser presupuestado para el proyecto.

Para limitaciones de calendario, se debe especificar una fecha de entrega que se debe cumplir, las restricciones de mano de obra limita el número de personas que puede ser asignado al proyecto. La carga es el número de personal de ingeniería y de gestión atribuido al proyecto como una función de tiempo. El esfuerzo se define como la ingeniería y la gestión del esfuerzo requerido para completar un proyecto. Por lo general se mide en meses-persona. La duración es la cantidad de tiempo requerido para completar el proyecto. El estimador puede también cuantificar un conjunto de factores de costos (Leungh, 2001).

1.3.4. Medidas y Métricas del Software

En el campo de la ingeniería del software, se suele hablar indistintamente de “métricas” y de “medidas”, pero sin embargo existen diferencias entre estos términos. Una medida indica cuantitativamente algún atributo de proceso o de producto (extensión, cantidad, dimensiones, capacidad, tamaño, etc.). Una métrica es definida por el Glosario de Estándares del IEEE (Institute of Electrical and Electronics

Engineers) como una “medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo determinado” (Agarwal, 2001).

Si se recopila un sólo tipo de datos, como por ejemplo el número de errores dentro de un componente, se ha establecido una medida. Una métrica de software relaciona de alguna manera las medidas individuales. Siguiendo nuestro ejemplo, podríamos tratar el número de errores encontrados en cada revisión o prueba.

Los ingenieros de software, a partir de las medidas, elaboran métricas que les proporcionan información para poder controlar el proceso o el desarrollo de software. Aquí radica la diferencia entre estos términos.

Hecha esta aclaración, se puede pasar a tratar algunas métricas de proyecto usadas por los ingenieros del software, todas ellas de vital importancia durante los procesos de estimación. Suelen ser recopiladas de proyectos anteriores, de manera que se comparan sucesivamente con los valores actuales de esfuerzo y tiempo, y así se puede supervisar la corrección de las estimaciones realizadas.

Otras métricas de gran utilidad pueden ser el número de páginas de documentación, las horas de revisión, el número de errores detectados, etc. (Dekkers, 2003).

Referente a las medidas del software, también vitales en la estimación, podemos clasificarlas en directas o indirectas: entre las directas se encuentran las líneas de código, velocidad de ejecución, tamaño de la memoria, etc.

1.3.5. Métodos de Estimación de Costos Existentes

Los métodos de estimación de costos son básicamente de dos tipos: algorítmicos y no algorítmicos. Los métodos algorítmicos utilizan una fórmula para calcular la estimación de costos de software. La fórmula se desarrolló a partir de los modelos que se crean mediante la combinación relacionada de los factores de costo. Además, el método estadístico se utiliza para la construcción del modelo. Los métodos no algorítmicos no utilizan una fórmula para calcular la estimación de costos de software.

A. *Métodos de estimación no algorítmicos*

1. Juicio de expertos

Las técnicas de la opinión de expertos implican consultar con un profesional experto en estimación de costos o un grupo de expertos para utilizar su experiencia y conocimiento en la propuesta de proyecto, para llegar a una estimación de su costo. Se trata de los métodos más útiles para la estimación de costos de software. Sobre todo las empresas utilizan este método para generar el costo del producto. Se trata de una técnica de consenso del grupo del tipo Delphi, que proporciona una comunicación satisfactoria entre los expertos e intercambiar el volumen de información necesaria se utiliza esta técnica (Capers, 2004).

Los pasos de estimación utilizados en este método son los siguientes:

- a. Un coordinador presenta a cada experto una especificación y un formulario de valoración.
- b. Los expertos rellenan formularios anónimos.
- c. El coordinador convoca una reunión de grupo, en la que los expertos discuten temas de estimación con el coordinador.
- d. El coordinador prepara y distribuye un resumen de la estimación en un formulario de iteración.
- e. Los expertos llenan los formularios, una vez más anónimamente, y los pasos c y d se reitera durante tantas rondas como sea apropiado.

La técnica Delphi posteriormente se ha utilizado en una serie de estudios y en las actividades de estimación de costes. Ha tenido un gran éxito en la combinación de las ventajas de discutir libremente en las reuniones de grupos (Capers, 2004).

2. Estimación por analogía

La estimación por analogía significa comparar el proyecto propuesto con un proyecto similar realizado anteriormente en el desarrollo de identificación de las informaciones conocidas. Los datos reales de los proyectos terminados son extrapolados para estimar la propuesta proyecto. Este método se puede utilizar ya sea a nivel de sistema o en el nivel de componente (Agarwal, 2001). Los pasos de estimación utilizando éste método son las siguientes:

- a. Descubrir las características del proyecto propuesto.
- b. Seleccionar los proyectos terminados más similares, cuyas características han sido almacenados en la base de datos histórica.
- c. Averiguar la estimación para el proyecto propuesto considerando la analogía con el proyecto similar terminado.

3. Método Top-Down

Este método de estimación de arriba hacia abajo se llama también un modelo macroeconómico. Utilizando el método de estimación de arriba a abajo, una estimación general de los costos para el proyecto se deriva de las propiedades globales del proyecto de software y, a continuación, el proyecto se divide en varios mecanismos o componentes de bajo nivel. El método principal de este enfoque es el modelo de Putnam. Este método es más aplicable a la estimación de costos temprano, cuando sólo se conocen las propiedades globales del producto. En la fase temprana del software desarrollo, es muy útil porque no hay información detallada disponible (Capers, 2004).

4. Método Bottom-Up

Este método de estimación de abajo a arriba, es la construcción de la estimación de un sistema a partir del conocimiento acumulado sobre los componentes de software pequeños y sus interacciones (Capers, 2004).

5. Ley de Parkinson

El uso del principio "El trabajo se expande hasta llenar todo el tiempo disponible para su realización" (Agarwal, 2001), determina el costo por los recursos disponibles, más que sobre la base de una evaluación objetiva. Si el software tiene que ser entregado en 12 meses y 5 personas están disponibles, se estima que el esfuerzo de esta es 60 meses-persona. Aunque a veces da buena estimación.

No se recomienda el método, ya que puede proporcionar estimaciones no muy realistas. Además, este método no promueve las buenas prácticas de ingeniería de software.

6. Price-to-win

El costo del software se estima que es el mejor precio para ganar el proyecto. La estimación se basa en el presupuesto del cliente en lugar de la funcionalidad del software. Por ejemplo, si una estimación razonable para un proyecto cuesta 100 persona-mes pero el cliente sólo puede permitirse 60 meses-persona, es común que al estimador se le pida modificar la estimación para ajustar al planteo; de nuevo, esto no es una buena práctica ya que es muy probable que cause una demora de entrega o fuerza al equipo de desarrollo a trabajar horas extras.

B. Métodos de estimación algorítmicos

El método algorítmico está diseñado para proporcionar algunas ecuaciones matemáticas para llevar a cabo la estimación de software. Estas ecuaciones matemáticas se basan en la investigación histórica y datos y cifras de uso, tales como líneas de código fuente (SLOC), número de funciones a realizar, y otros indicadores de costos, tales como el lenguaje, la metodología de diseño, los niveles de habilidad, riesgo de evaluaciones, etc. Los métodos algorítmicos se han estudiado en gran medida y muchos modelos han sido desarrollados, tales como modelos COCOMO, modelo de Putnam, y los puntos de función (Milicic, 2004).

1.3.6. Modelos Existentes

A continuación se listan y describen los modelos utilizados en la actualidad para la estimación del tamaño del software.

A. Estimación del tamaño del software

Metodología	Descripción
Conteo de Líneas de Código	Este método toma las líneas de código necesarias para la construcción de un sistema como medida de su tamaño.
Conteo por Número de Bloques	Este método toma como medida del tamaño de un sistema, el número de funciones que éste contiene.
Estimación basada en la estadística	Este método divide el sistema en componentes, para así realizar las estimaciones sobre cada uno de ellos.
Estimación por Puntos de Función	Se basa en la funcionalidad del sistema, no en el producto del desarrollo, para realizar su estimación se deben determinar los componentes de puntos de función para el sistema y clasificarlos según su dificultad.

Tabla 1 Descripción de modelos de estimación del tamaño del software (Clemmons, 2006).

B. Gestión de costos

Metodología	Descripción
Costos por Analogía	Se estima el costo del proyecto basándose en el costo de proyectos similares ya realizados.
Precio a Ganar	Se ajusta el precio del proyecto para mejorar la propuesta más económica realizada, con el fin de ganar el proyecto.
COCOMO	Modelo empírico para la estimación del esfuerzo y costo del desarrollo de un sistema de software, se basa en el uso de multiplicadores de esfuerzo.
SLIM	Se basa en la distribución de poder hombre, se usa la ecuación de software, en donde se relaciona, el tiempo de entrega, factores ambientales, en los cuales se refleja la capacidad de desarrollo de la compañía

Tabla 2 Descripción de modelos de gestión de costos (Clemmons, 2006).

C. Gestión de riesgos

Proceso	Descripción
PMBok® 2000	Estándar que utiliza el conocimiento, herramientas y técnicas para resolver posibles problemas del proyecto.
SEI-CRM, Método Continuos Risk Management	Proporciona una guía compuesta por principios, conceptos y funciones para la toma de decisiones entorno a riesgos que deben ser evaluados continuamente.
IEEE	Establece una norma para el desarrollo de planes de gestión del riesgo constituidos por el uso de formatos. Esta norma no establece técnicas exactas para ser usadas en los planes de proyecto.

Tabla 3 Descripción de modelos de gestión de riesgos (Clemmons, 2006).

1.4. Metodología

1.4.1. Dimensión de la Estrategia General

A. Tipo de estrategia y fundamentación

El marco metodológico de este trabajo incluye investigación, técnicas y procedimientos. Es el cómo se realizará el estudio para responder al problema planteado.

Los elementos constitutivos del marco metodológico en el proyecto de investigación serán:

- ✓ Tipo de investigación.
- ✓ El diseño de investigación.
- ✓ Población o universo de estudio.
- ✓ La muestra.
- ✓ Los instrumentos de recolección de la información.
- ✓ Descripción de los instrumentos y técnicas de recolección de datos.

- ✓ Medición.
- ✓ Niveles de medición de las variables.
- ✓ La prueba piloto.
- ✓ El análisis e interpretación de los resultados.
- ✓ Codificación y tabulación de los datos.
- ✓ Técnicas de presentación de los datos.
- ✓ El análisis estadístico de los datos.
- ✓ Técnicas para la formulación del modelo operativo o sistema propuesto.

B. Universo

Se limita a empresas de la región de Ciudad del Este dedicadas al desarrollo de software en ambientes caóticos compuestos por treinta o más profesionales de poca experiencia y presionados a un resultado rápido.

C. Unidad de análisis

La unidad de análisis será el área Datacenter de la fundación del Parque Tecnológico Itaipú dedicada al desarrollo de software de la gestión de dicha fundación.

D. Selección de casos

El caso seleccionado será el sistema de gestión SGPTI (Sistema de Gestión del Parque Tecnológico Itaipú) de la fundación del Parque Tecnológico de Itaipú.

E. Énfasis en el proceso lineal

Conforme a las características de la investigación propuesta, se ha optado por la lógica cuantitativa, basada en la búsqueda y verificación de relaciones causa-efecto, que se puedan generalizar; es decir, hay un proceso lineal de relación teoría-empiría, con diferentes espacios y momentos para la definición de las hipótesis, la obtención de los datos, su análisis y su posterior interpretación.

F. Rol del investigador

En una *lógica cuantitativa* como la adoptada para la presente investigación, el investigador intenta descubrir una realidad preexistente a la investigación, formulando hipótesis que luego habrá de confrontar con los datos obtenidos durante la investigación, como resultado de lo cual eventualmente podrían tener que reformularse las hipótesis originales.

1.5. Estructura de la Tesis

Esta tesis está estructurada del siguiente modo:

En el capítulo 2 se presenta a COCOMO II en el campo de la medición y estimación del tamaño funcional de los artefactos generados en etapas tempranas del desarrollo de software. Esta técnica es evaluada de acuerdo a un conjunto de criterios previamente establecidos.

En el capítulo 3 se estudia a las metodologías ágiles de desarrollo de software sus parámetros de estimación.

En el capítulo 4 se presentan la especificación y evaluación de una nueva metodología propuesta. Se definen las reglas y procedimientos que permitirán estimar el costo del software con las nuevas tendencias tecnológicas.

En el capítulo 5 se presentan dos casos de estudio en la estimación de software, una estimando con COCOMO II y otra con la nueva metodología propuesta.

En el capítulo 6 se presentan las conclusiones y contribuciones de la propuesta presentada y los trabajos futuros.

1.6. Discusiones y Comentarios

En este capítulo se ha elaborado la metodología a ser aplicada para la ejecución de esta tesis, también se ha detallado el marco teórico en el cual se basará la implementación de la hipótesis.

Se han encontrado artículos que ayudan a cumplir los objetivos, además de indicar un breve resumen de lo que tratará el trabajo a realizar.

En este trabajo, se ofrece una visión general de los diferentes tipos de métodos de estimación de costos de software, las ventajas y desventajas de estos métodos.

Este documento también presenta algunas de las razones por las cuales las estimaciones de costos de desarrollo son habitualmente inexactas. Para producir una estimación significativa y confiable, se debe mejorar la comprensión de atributos del proyecto de software y sus relaciones causales, desarrollar formas efectivas de la complejidad del software de medición, se necesita ser cuidadoso siguiendo el proceso de estimación de costos. Se ha visto que todos los métodos de estimación son específicos para algún tipo específico de proyectos.

Es muy difícil decidir qué método es mejor que otros métodos porque cada método o modelo tiene un propio significado o importancia. Entender sus ventajas y desventajas es muy importante cuando se quiere calcular el costo de desarrollo de software. En investigaciones de años recientes, se trabajó con otro campo, junto con la ingeniería de software como la minería de datos y técnicas de aprendizaje automático para la mejora de la precisión de proceso de estimación de costos de software.

El trabajo consiste en estudiar un nuevo método de estimación de costos de software y un modelo que pueda ayudar a comprender fácilmente el proceso de estimación de costos de software en el desarrollo ágil.

Capítulo 2

2.1. COCOMO II

Hoy en día las compañías de software desarrollan diferentes programas de software en paralelo, que es una tarea muy compleja. Los responsables del proyecto gestionan los diferentes procesos de desarrollo de software basados en restricciones como el tiempo, el costo y el número de personal. Calcular el tiempo, el costo y el número de personal es un trabajo muy tedioso para los directores de proyectos de las empresas de software en las primeras etapas de la planificación y seguimiento. COCOMO (Boehm B., 2000) es uno de los mejores modelos para estimar el costo y el tiempo en meses-persona de un proyecto de software.

Este capítulo presenta una visión general de los modelos COCOMO que incluye extensiones y modelos independientes, y describe las metodologías subyacentes y la lógica detrás de los modelos y cómo pueden ser utilizados en conjunto para apoyar grandes necesidades de estimación del sistema de software. Concluye con una discusión para unificar estos diversos modelos en un proceso más simplificado que derivará en una nueva metodología completa y fácil de usar.

2.2. Estimación con COCOMO II

La estructura de COCOMO II se basa en modelos que asumen que se progresa a lo largo de un desarrollo de tipo espiral para consolidar los requisitos y la arquitectura, reduciendo el riesgo; tales modelos son:

- ✓ Modelo de Composición de Aplicaciones.
- ✓ Modelo de Diseño Temprano.
- ✓ Modelo de Arquitectura Tardía.

La estructura de COCOMO II, como se visualiza en la Ilustración 2, se basa en modelos que asumen que se progresa a lo largo de un desarrollo de tipo espiral para consolidar los requisitos y la arquitectura, reduciendo el riesgo; tales modelos son:

- ✓ Modelo de Composición de Aplicaciones.
- ✓ Modelo de Diseño Temprano.

- ✓ Modelo de Arquitectura Tardía.



Ilustración 2 Arquitectura COCOMO II (Boehm B., 2000).

COCOMO II utiliza variables establecidas en función de una medida de cinco factores de escala:

PREC	Precedencia.
FLEX	Flexibilidad de desarrollo.
RESL	Resolución de Arquitectura / Riesgos.
TEAM	Cohesión de equipo.
PMAT	Madurez del proceso.

Para realizar las estimaciones COCOMO II utiliza como medida puntos de objeto, puntos de función o líneas de código, basándose en el diseño lógico del sistema.

COCOMO II posee 17 multiplicadores de costos, cada uno de los cuales debe ser estimado:

RELY	Fiabilidad.
DATA	Tamaño de la Base de Datos.
CPLX	Complejidad.
RUSE	Reutilización requerida.
DOCU	Documentación.
TIME	Restricción tiempo de ejecución.
STOR	Restricción de almacenamiento principal.
PVOL	Volatilidad plataforma.
ACAP	Capacidad del analista.
PCAP	Capacidad del programador.

AEXP	Experiencia de aplicaciones.
PEXP	Experiencia plataforma.
LTEX	Experiencia del lenguaje y herramienta.
PCON	Continuidad del personal.
TOOL	Uso de herramientas software.
SITE	Desarrollo Multi-lugar.
SCED	Planificación requerida.

Una característica importante a destacar es su modelo de reutilización y sus características de auto calibración.

Como se ve, muchos de sus parámetros de configuración son subjetivos, por lo tanto la exactitud de la estimación depende en gran medida de la experiencia de la persona que la realiza, además es muy importante la cantidad de proyectos anteriores (con características similares al nuevo proyecto) porque ayudaría a obtener datos más precisos y partir de una base sólida.

COCOMO II (Milicic, 2004) pasó a ser una familia de modelos de productividad, estimación y toma de decisiones. Algunos de los modelos se consideran en desarrollo, es decir que requieren todavía estudios para validarlos y calibrarlos.

2.3. Modelo de Composición de Aplicaciones

Este modelo se dirige a aplicaciones que están demasiado diversificadas para crearse rápidamente en una herramienta de dominio específico y que todavía no se conocen suficientemente como para ser compuestas a partir de componentes interoperables. Ejemplos de estos sistemas basados en componentes son los creadores de interfaces gráficas para usuario, bases de datos o gestores de objetos, middleware para proceso distribuido o transaccional, manejadores hipermedia, buscadores de datos pequeños y componentes de dominio específico tales como paquetes de control de procesos financieros, médicos o industriales (Boehm B., 2000).

Dado que el Modelo de Composición de Aplicaciones incluye esfuerzos de prototipado para resolver asuntos potenciales de alto riesgo tales como interfaces de usuario, interacción software/sistema, ejecución o grado de madurez tecnológica, los costes de este tipo de esfuerzo se estiman mejor mediante dicho modelo.

2.3.1. Introducción a los Puntos Objeto

Los Puntos Objeto son el recuento de pantallas, informes y módulos de lenguajes de 3ª generación desarrollados en la aplicación, cada uno ponderado mediante un factor de complejidad de tres niveles (simple, medio y complejo).

La estimación de Puntos Objeto es un enfoque relativamente nuevo de medida del software, pero encaja bien en las prácticas del sector de la Composición de Aplicaciones. También encaja muy bien en los esfuerzos de prototipado asociados, basados en el uso de herramientas ICASE (International Council of Associations for Science Education) que proporcionan constructores de interfaces gráficos de usuario, herramientas de desarrollo software, y en general, infraestructura que puede componerse y componentes de aplicación. En estas áreas se ha comparado bien con la estimación de Puntos de Función en un conjunto de aplicaciones no triviales (pero todavía limitadas) (Kidd, 1994).

Uno de los estudios comparativos entre la estimación de Puntos de Función y Puntos Objeto analizó una muestra de 19 proyectos software sobre inversión bancaria de una misma organización, desarrollado utilizando las utilidades de Composición de Aplicaciones ICASE y con un rango de esfuerzo de 4.7 a 71.9 PM (Persona-Meses). El estudio descubrió que el enfoque de Puntos Objeto justificó un 73% de la varianza (R2) en meses-persona ajustados por la reutilización, comparada con el 76% para los Puntos de Función (Longstreet, 2004).

A. Procedimiento de obtención de Puntos Objeto

Definiciones de los términos utilizados en los Puntos de Objetos:

- ✓ **NOP:** Nuevos Puntos Objeto. (Cantidad de Puntos Objeto ajustados por la reutilización).
- ✓ **Servidor:** Número de tablas de datos del servidor (mainframe o equivalente) usadas junto con la pantalla o el informe.
- ✓ **Cliente:** Número de tablas de datos del cliente (estación de trabajo personal) usadas junto con la pantalla o el informe.
- ✓ **Reutilización:** Porcentaje de pantallas, informes y módulos 3GL reutilizados a partir de aplicaciones anteriores prorrateadas por grado de reutilización.

Las reglas de recuento para "objetos" de esa naturaleza cuando se usan en lenguajes como C++, se mostrará en la sección del Modelo Arquitectura Tardía.

Los pasos para estimar con Puntos Objetos son los siguientes:

1. **Hacer el recuento de objetos:** Estimar el número de pantallas, informes y componentes de las que consta la aplicación. Suponer las definiciones estándar de estos objetos en el entorno ICASE correspondiente.

2. **Clasificar cada instancia de objeto** dentro de 3 niveles de complejidad (simple, media, difícil) según valor de las dimensiones de la característica como se muestra en la Tabla 4.

Tabla 4 Complejidad asociada a instancias de objetos (Longstreet, 2004).

Para pantallas	Para Informes	Nº y fuente de tablas de datos		
		Total < 4 (<2 serv <3 clientes)	Total < 8 (2/3 serv 3/5 clientes)	Total > 8 (>3 servi >5 clientes)
<3	0 o 1	Simple	Simple	Medio
3 a 7	2 o 3	Simple	Medio	Difícil
>8	4+	Medio	Difícil	Difícil

3. **Pesar el número de cada celda,** reflejando el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad (Tabla 5):

Tabla 5 Tipo Objeto / Pesos Asociados (Longstreet, 2004).

Tipo de Objeto	Pesos asociados a los niveles de complejidad		
	Simple	Medio	Difícil
Pantalla	1	2	3
Informe	2	5	8
Componente 3 GL			10

4. **Determinar Puntos Objeto:** Suma todas las instancias de objeto pesadas para conseguir un número. El recuento de Puntos Objeto.

5. **Estimar el porcentaje de reutilización** que se espera lograr en este proyecto. Calcular los nuevos Puntos Objeto a desarrollar:

$$NOP = (Object Points) \times (100 - \%Reuse) / 100$$

6. **Determinar un ratio de productividad** PROD = NOP/meses-persona a partir de la Tabla 6:

Tabla 6 Ratio de Productividad (Longstreet, 2004).

Experiencia y capacidad de los desarrolladores	Madurez y capacidad en ICASE	PROD
Muy Baja	Muy Baja	4
Baja	Baja	7
Nominal	Nominal	13
Alta	Alta	25
Muy Alta	Muy Alta	50

Calcular el valor Meses-persona estimado según la ecuación:

$$PM = NOP / PROD$$

2.4. Modelos de Diseño Temprano y Arquitectura Tardía

El Diseño Temprano incluye la exploración de arquitecturas de software/sistema alternativas y conceptos de operación. En esta fase no se sabe lo suficiente como para dar soporte a la estimación de grano fino.

La correspondiente capacidad de COCOMO II incluye el uso de Puntos de Función y un conjunto de siete drivers de costes (por ejemplo, dos drivers de coste para capacidad del personal y experiencia del personal en lugar de los seis drivers de coste del Modelo Arquitectura tardía que cubren varios aspectos de capacidad del personal, continuidad y experiencia) (Yenduri, S.; Munagala, S.; Perkins, L.A., 2007).

El modelo de Diseño Temprano usa Puntos de Función No Ajustados como métrica de medida. Este modelo se utiliza en las primeras etapas de un proyecto software, cuando se conoce muy poco sobre el tamaño del producto que se va a desarrollar, la naturaleza de la plataforma objetivo, la naturaleza del personal involucrado en el proyecto o especificaciones detalladas del proceso que se va a usar. Este modelo puede aplicarse a cada uno de los sectores de desarrollo de Generador de Aplicaciones, Integración de sistemas o Infraestructura.

El Modelo Arquitectura Tardía incluye el actual desarrollo y mantenimiento de un producto software. Esta fase avanza rentablemente si se desarrolla una arquitectura de ciclo de vida software válida con respecto a la misión del sistema, al concepto de operación y al riesgo, y establecido como marca de trabajo del producto. El modelo correspondiente de COCOMO II tiene aproximadamente la misma granularidad que los anteriores modelos COCOMO y Ada COCOMO. Utiliza instrucciones fuente y/o Puntos de Función para medir, con modificadores para reutilización y objetos software;

un conjunto de 17 multiplicadores de costos; y un conjunto de 5 factores que determinan el exponente de escala del proyecto. Estos factores sustituyen los modos de desarrollo (Orgánico, Semilíbre y Rígido) del modelo original COCOMO y refina los 4 factores de exponente-escala en Ada COCOMO (Boehm B., 2000).

Los **dos modelos de Diseño Temprano y de Arquitectura tardía** se basan en la misma filosofía a la hora de proporcionar una estimación. Como se ha indicado, su principal diferencia se produce en la cantidad y detalle de la información que se utiliza para obtener la estimación en cada uno de ellos. La fórmula básica para obtener una estimación de esfuerzo con ambos modelos es:

$$PM_{\text{nominal}} = A * (\text{Size})^B$$

Esta ecuación calcula el esfuerzo nominal para un proyecto de un tamaño dado expresado en PM. Las entradas son la medida del desarrollo del software, una constante, A, y un factor de escala, B. La medida SIZE está en unidades de líneas de código fuente KSLOC (thousands of lines of code). Esto se deriva de la medida de módulos software que constituirán el programa de aplicación, puede estimarse también a partir de Puntos de Función sin ajustar convirtiendo a SLOC (Líneas de código fuente) y luego dividiendo por 1000.

El factor de escala (exponencial B), explica el ahorro o gasto relativo de escala encontrado en proyectos software de distintos tamaños. La constante A, se usa para cortar los efectos multiplicativos de esfuerzo en proyectos de tamaño incremental.

2.4.1. Componentes de los Modelos

1. Constante A

Como se ha visto anteriormente la constante A se usa para capturar los efectos multiplicativos de esfuerzo en proyectos de tamaño incremental. Provisionalmente se le ha estimado un valor de 2,45.

2. Variable SIZE

$$\text{Dónde: } SIZE = SIZE * (1 + BRAK) / 100$$

COCOMO II utiliza un porcentaje de Rotura (BRAK) para ajustar el tamaño eficaz del producto. La rotura refleja la volatilidad de los requisitos en un proyecto. Es el porcentaje de código desperdiciado debido a la volatilidad de los requisitos. El factor BRAK no se usa en el Modelo de Composición de Aplicaciones donde se espera un cierto grado de iteración en el producto y se incluye en la calibración de datos.

El tamaño de una aplicación se mide en unidades de líneas de código fuente (KSLOC). Al igual que en la versión inicial del COCOMO, este valor se deriva de la medida de módulos software que constituirán el programa de aplicación, sin embargo, en la nueva versión COCOMO II puede estimarse también a partir de Puntos de Función sin ajustar convirtiendo a SLOC y luego dividiendo por 1000.

Si se opta por utilizar directamente el valor del número de líneas de código, la meta es medir la cantidad de trabajo intelectual que se emplea en el desarrollo del programa, pero las dificultades aparecen al intentar definir medidas consistentes en diferentes lenguajes.

Si se opta por utilizar los Puntos de Función sin Ajustar para determinar el tamaño del proyecto, éstos deben convertirse en líneas de código fuente en el lenguaje de implementación (ensamblador, lenguajes de alto nivel, lenguajes de cuarta generación, etc.) para evaluar la relativamente concisa implementación por Puntos de Función (ver la Tabla 7 de Conversión de Puntos de Función a Líneas de código). COCOMO II realiza esto tanto en el Modelo de Diseño Temprano como en el de Arquitectura Tardía usando tablas como la tabla mencionada que traducen Puntos de Función sin ajustar al equivalente SLOC.

Tabla 7 Puntos de Función sin ajustar al equivalente SLOC (Banerjee, 2001)

<i>Lenguaje</i>	<i>SLOC/UFP</i>	<i>Lenguaje</i>	<i>SLOC/UFP</i>
Ada	71	AI Shell	49
APL	32	Assembly	320
Assembly (Macro)	213	ANSI/Quick/Turbo Basic	64
Basic - Compiled	91	Basic Interpreted	128
C	128	C++	29
Visual Basic	32	ANSI Cobol 85	91
Fortran77	105	Fortli	64
Jovial	105	Lisp	64
Modula 2	80	Pascal	91
Prolog	64	Report Generator	80
Spreadsheet	6		

Sólo quedaría por hacer la conversión a KSLOC mediante la operación SLOC/1000.

3. Variable B

$$B = 0,91 + 0,01 \sum SF_j$$

Los modelos de estimación a menudo tienen un factor exponencial para considerar los gastos y ahorros relativos de escala encontrados en proyectos software de distinto tamaño. El exponente B se usa para capturar estos efectos. Su valor obtenido en la ecuación anterior se interpreta según la siguiente Tabla 9:

- ✓ Si $B < 1,0$ el *proyecto presenta ahorros de escala*. Si el tamaño del producto se dobla, el esfuerzo del proyecto es menor que el doble. La productividad del proyecto aumenta a medida que aumenta el tamaño del producto. Pueden lograrse algunos ahorros de escala del proyecto con herramientas de proyecto específicas (por ejemplo, simulaciones) pero normalmente es difícil lograrlo. Para proyectos pequeños, fijar costes de salida como herramientas a medida y normas de montaje, e informes administrativos, suele ser una fuente de ahorro de escala.
- ✓ Si $B = 1,0$ los *ahorros y gastos de escala están equilibrados*. Este modelo lineal se usa a menudo para la estimación de coste de proyectos pequeños. Se usa para el modelo COCOMO II en Composición de Aplicaciones.
- ✓ Si $B > 1,0$ el *proyecto presenta gastos de escala*. Esto se debe normalmente a dos factores principales: El crecimiento del gasto en comunicaciones y el gasto en crecimiento de la integración de un gran sistema. Los proyectos más grandes tendrán más personal y por lo tanto más vías de comunicación interpersonales produciendo gasto. Integrar un producto pequeño como parte de uno más grande requiere no sólo el esfuerzo de desarrollar el producto pequeño sino también el gasto adicional en esfuerzo para diseñar, mantener, integrar y probar sus interfaces con el resto del producto.

El exponente B se obtiene mediante los denominados drivers de escala. La selección de drivers de escala se basa en la razón de que ellos son un recurso significante de variación exponencial en un esfuerzo o variación de la productividad del proyecto. Cada driver de escala tiene un rango de niveles de valores desde Muy Bajo hasta Extra Alto (**Tabla 8 Factores de escala para el Modelo de COCOMO II de Diseño Temprano**). Cada nivel de valores tiene un peso SE cuyo valor específico se llama factor de escala. Un factor de escala de un proyecto, SF_j (Tabla 8) se calcula sumando todos los factores y se usa para determinar el exponente B de escala.

Tabla 8 Conceptos de los Factores de escala para el Modelo de Cocomo II de Diseño Temprano pertenecientes a la versión USC-COCOMOII.2000 (B. Boehm, 2000)

Factores de Escala (FE _i)	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PREC	Completamente sin precedentes	Prácticamente sin precedentes	Casi sin precedentes	Algo familiar	Muy familiar	Completamente familiar
FLEX	Riguroso	Relajación ocasional	Algo de relajación	Conformidad general	Algo de conformidad	Metas generales
RESL (*)	Poco (20%)	Algo (40%)	A menudo (60%)	Generalmente (75%)	En su mayor parte (90%)	Por completo (100%)
TIEAM	Interacciones muy difíciles	Algo de dificultad en interacciones	Interacciones básicamente cooperativas	Bastante cooperativo	Altamente cooperativo	Completas interacciones
PMAT	Peso medio de respuestas "Sí" para el cuestionario de Madurez CMM					

(*) % significativo: de módulos de interfaz especificados, % de riesgos eliminados

Tabla 9 Valores de los Factores de escala para el Modelo de Cocomo II de Diseño Temprano pertenecientes a la versión USC-COCOMOII.2000 (B. Boehm, 2000)

Factores de Escala (W _i)	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PREC	6,20	4,96	3,72	2,48	1,24	0,00
FLEX	5,07	4,05	3,04	2,03	1,01	0,00
RESL*	7,07	5,65	4,24	2,83	1,41	0,00
TEAM	5,48	4,38	3,29	2,19	1,10	0,00
PMAT	7,80	6,24	4,68	3,12	1,56	0,00

4. Ajuste por multiplicadores de costos

Los **multiplicadores de costos** se usan para capturar características del desarrollo del software que afectan al esfuerzo para completar el proyecto. Los drivers de coste tienen un nivel de medida que expresa el impacto del driver en el esfuerzo de

desarrollo. Estos valores pueden ir desde Extra Bajo hasta Extra Alto. Para el propósito del análisis cuantitativo, cada nivel de medida de cada driver de coste tiene un propósito asociado. El peso se llama multiplicador de esfuerzo (ME). La medida asignada es 1,0 y el nivel de medida asociado con ese peso se llama nominal. Si un nivel de medida produce más esfuerzo de desarrollo de software, entonces su correspondiente ME está por encima de 1,0. Recíprocamente si el nivel de medida reduce el esfuerzo entonces el correspondiente ME es menor que 1,0.

Los ME se usan para ajustar el esfuerzo Meses-Persona nominal. Hay 7 multiplicadores de esfuerzo para el Modelo de Diseño Temprano y 17 para el modelo de Arquitectura Tardía.

$$MM = A x (Size)^B x \sum ME_i$$

Los **multiplicadores de costos del Diseño Temprano** se obtienen combinando los mismos multiplicadores del modelo Arquitectura Tardía de la Tabla 10 (Multiplicadores de Esfuerzo del Diseño Temprano y Post-Arquitectura). Siempre que una evaluación de un driver de coste está entre niveles de ratio hay que redondear al valor más próximo al nominal. Por ejemplo, si un valor de un driver de coste está entre Muy Bajo y Bajo, entonces seleccionar Bajo.

Tabla 10 Drivers de coste (B. Boehm, 2000)

<i>Drivers de coste combinados, del Diseño temprano tardía</i>	<i>Drivers de coste homólogos, del Post-Arquitectura</i>
RCPX DOCU RUSE PDIF PERS PREX FCIL SCED	RELY, DATA, CPLX, RUSE TIME, STOR, PVOL ACAP, PCAP, PCON AEXP, PEXP, LTEX TOOL, SITE SCED

2.4.2. Modelo de Mantenimiento de las Aplicaciones

COCOMO II utiliza el Modelo de reutilización cuando la cantidad de código fuente base añadido o modificado es menor o igual que el 20% del nuevo código que se está desarrollando. Código base es el código fuente que ya existe y que se cambia para

usarlo en un proyecto actual. Para el mantenimiento de proyectos que incluyen más del 20% de cambio en el código base existente (relativo al nuevo código que se está desarrollando) COCOMO II utiliza el Modelo de mantenimiento, donde el tamaño ($Size_M$) se obtiene de forma relativamente distinta, con lo que la fórmula de estimación del esfuerzo de mantenimiento resultante es la siguiente:

$$MM_{Mantenimiento} = A * (Size_M)^B * \sum ME_i$$

Un tamaño de mantenimiento inicial se puede obtener de dos formas:

1. Si el tamaño de código base se conoce y el porcentaje de cambio también, entonces:

$$(Size)_M = [(BaseCodeSize) * MCF] * MAF$$

El Factor de Cambio MCF, porcentaje de código base modificado, es similar a la tasa de cambio anual de COCOMO 81, sólo que pueden usarse periodos de mantenimiento superior a un año.

$$MCF = (SizeAdded + SizeModified) / BaseCodeSize$$

$$MAF = 1 + [SU / 100 * UNFM]$$

2. Si se conoce la fracción de código añadido o modificado del código base durante el periodo de mantenimiento se utiliza la ecuación siguiente donde:

$$(Size)_M = (SizeAdded + SizeModified) * MAF$$

$$MAF = 1 + [SU / 100 * UNFM]$$

El tamaño puede referirse a miles de líneas de código fuente (KSLOC), Puntos de Función o Puntos Objeto. Cuando se usan Puntos de Función o Puntos Objeto es mejor estimar MCF en términos de fracción de la aplicación global que está siendo modificada, en lugar de la fracción de entradas, salidas, Pantallas, informes,... afectados por los cambios. La experiencia ha demostrado que contar los ítems afectados puede llevar a estimaciones importantes por encima, así como cambios relativamente pequeños pueden afectar a un relativo gran número de ítems.

El cálculo del tamaño de mantenimiento inicial (descrito anteriormente) se ajusta con el Factor de Ajuste del Mantenimiento (MAF). COCOMO 81 utiliza diferentes multiplicadores para los efectos de la Fiabilidad Requerida (RELY) y para las Prácticas de Programación Modernas (MODP) en el mantenimiento, en contra del esfuerzo de desarrollo. COCOMO II, en cambio, utilizó los factores de Comprensión del Software (SU) y de Desconocimiento del Programador (UNFM) de su Modelo de Reutilización para modelar los efectos del software comprensiblemente suficiente o escasamente estructurado por el esfuerzo de mantenimiento (Fenton, 2014).

2.5. Discusiones y Comentarios

El COCOMO II es un modelo bastante simple. No hay entrenamiento necesario para realizar la medición. Una vez que un buen procedimiento está configurado, el conteo de SLOC es trivial. Luego, sólo hay que determinar la calificación de los factores de coste y la fórmula se puede utilizar para obtener el esfuerzo estimado. Sin embargo, la dificultad está en determinar los valores correctos y veraces para estos factores de coste.

Actualmente en la programación orientada a objetos se utilizan técnicas de empaquetamiento y encapsulación, esto hace diferencia a la hora de utilizar líneas de código fuente para estimar ya que muchas veces no se tiene todo a mano para llegar a un control total de líneas de código, de ahí surgen la idea de aportar otra técnica de estimación donde no se incluyan líneas de código ya que es un factor de mucha influencia para estimar con COCOMO.

De ahí se ve la principal desventaja del modelo COCOMO II, es el uso de líneas de código como una medida del tamaño (Yenduri, S.; Munagala, S.; Perkins, L.A., 2007). Aunque el modelo sólo incluye líneas de código fuente, y da una descripción sobre lo que debe incluir, todavía hay algunas de las paradojas presentes, por ejemplo, como resultado del uso de diferentes lenguajes de programación o los diferentes estilos de programación. En la aplicación de este modelo, se debe ser consciente de estas paradojas y tenerlas en cuenta al interpretar los resultados.

Aunque el modelo puede dar un marco de referencia para definir la productividad de un proyecto, la fuerza principal del modelo COCOMO II es la extensa lista de características del proyecto (los factores de escala y los multiplicadores de esfuerzo), de ahí el interés de utilizar estos factores y multiplicadores y reemplazar las líneas de código ya que son las líneas de código los que todavía no se adecúan a lo utilizado hoy en día para el desarrollo basado en metodologías ágiles.

Capítulo 3

3.1. Metodologías Ágiles

Hace casi dos décadas que se comenzó a buscar una alternativa a las metodologías formales o tradicionales que estaban sobrecargadas de técnicas y herramientas y que se consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo.

Las metodologías ágiles (Mendes Calo, 2010) conllevan una filosofía de desarrollo de software liviana, debido a que hacen uso de modelos ágiles. Se considera que un modelo es ágil o liviano cuando se emplea para su construcción una herramienta o técnica sencilla, que apunta a desarrollar un modelo aceptablemente bueno y suficiente en lugar de un modelo perfecto y complejo (Ahmed E. Hassan, 2003). Existen actualmente una serie de metodologías que responden a las características de las metodologías ágiles y cada vez están teniendo más adeptos. Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito El Manifiesto Ágil (The Agile Manifesto, 2001) y coinciden con sus postulados y principios, cada metodología ágil tiene características propias y hace hincapié en algunos aspectos más específicos.

Actualmente se pretende desarrollar software en el menor tiempo posible y con el menor costo. La filosofía de la metodología ágil es aquella que da mayor valor al individuo, a la colaboración activa del cliente en todas las etapas del proceso de desarrollo y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Esta pretensión constituye un cambio esencial de paradigma e implica un choque cultural.

3.2. El Manifiesto Ágil

La Alianza Ágil (The Agile Manifesto, 2001) elaboró un conjunto de doce principios comunes a las metodologías ágiles de desarrollo que se enuncian a continuación:

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se dobligan al cambio como ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la principal medida del progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo que no se hace, es esencial.
11. La simplicidad como arte de maximizar la cantidad de trabajo que no se hace, es esencial.
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

3.3. Metodologías Ágiles de Desarrollo más Exitosas

3.3.1. Scrum

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. En Scrum (Cockburn, 2007) un proyecto se ejecuta en

bloques temporales (iteraciones-sprints) de un mes natural (pueden ser de dos o tres semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea susceptible de ser entregado con el mínimo esfuerzo cuando el cliente lo solicite.

Sprint es el ritmo de los ciclos de Scrum. Está delimitado por la reunión de planificación del sprint y la reunión retrospectiva. Una vez que se fija la duración del sprint es inamovible. La mayoría de los equipos eligen dos, tres o cuatro semanas de duración. Diariamente durante el sprint, el equipo realiza una reunión de seguimiento muy breve. Al final del sprint se entrega el producto al cliente en el que se incluye un incremento de la funcionalidad que tenía al inicio del sprint.

El proceso parte de la lista de requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente ha priorizado los requisitos balanceando el valor que le aportan respecto a su coste y han sido divididos en iteraciones y entregas.

A. Actividades de la metodología Scrum

Las actividades que se plantean en la metodología Scrum son las siguientes:

1. Planificación de la iteración

La planificación de Sprint es una reunión crítica, probablemente la más importante de Scrum. Una planificación de Sprint mal ejecutada puede arruinar por completo todo el Sprint.

Una planificación de Sprint produce, concretamente:

- ✓ Una meta de Sprint.
- ✓ Una lista de miembros (y su nivel de dedicación, si no es del 100%).
- ✓ Una Pila de Sprint (lista de historias incluidas en el Sprint).
- ✓ Una fecha concreta para la Demo del Sprint.
- ✓ Un lugar y momento definidos para el Scrum Diario.

2. Estimando usando cálculos de velocidad

- ✓ Decidir la velocidad estimada.
- ✓ Calcular cuántas historias se pueden añadir sin sobrepasar la velocidad estimada La velocidad es una medida de “cantidad de trabajo realizado”, donde cada elemento se evalúa en función de su estimación inicial.

3. Ejecución de la iteración (sprint)

4. Terminación anormal de la iteración

Sólo en situaciones muy excepcionales el cliente o el equipo pueden solicitar una terminación anormal de la iteración. Esto puede suceder si, por ejemplo, el contexto del proyecto ha cambiado enormemente y no es posible esperar al final de la iteración para aplicar cambios, o si el equipo encuentra que es imposible cumplir con el compromiso adquirido. En ese caso, se dará por finalizada la iteración y se dará inicio a otra mediante una reunión de planificación de la iteración.

5. Reunión diaria de sincronización del equipo

El objetivo de esta reunión es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad. Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para al finalizar la reunión poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso conjunto que el equipo adquirió para la iteración.

6. Retrospectiva (Sprint Retrospective)

El equipo analiza cómo ha sido su manera de trabajar durante la iteración, qué cosas han funcionado bien, cuáles hay que mejorar, qué cosas quiere probar hacer en la siguiente iteración, qué se ha aprendido y cuáles son los problemas que podrían impedirle progresar adecuadamente, con el objetivo de mejorar de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados que el propio equipo no pueda resolver por sí mismo. Se realiza en un tiempo máximo 3 horas.

7. Replanificación del proyecto

Durante el transcurso de una iteración, el cliente va trabajando en la lista de requisitos priorizada del producto o proyecto, añadiendo requisitos, modificándolos, eliminándolos, re-priorizándolos, cambiando el contenido de iteraciones y definiendo un calendario de entregas que se ajuste mejor a sus nuevas necesidades.

8. Incorporar nuevos recursos

Cancelar el proyecto con los requisitos completados hasta el momento plenamente operativos, si el beneficio pendiente de obtener es menor que el coste de desarrollo.

El plan de proyecto se actualiza con la velocidad de desarrollo del equipo, se evitan sorpresas de última hora.

B. Roles

Cuando se aplica la metodología Scrum se determinan las siguientes responsabilidades:

No hay un jefe de proyecto. Las responsabilidades del tradicional jefe de proyecto se distribuyen a los siguientes roles de un equipo Scrum:

- ✓ El cliente o Product Owner.
- ✓ Scrum master o facilitador.
- ✓ Resto del equipo.

C. Herramientas

Entre las herramientas que son necesarias en la aplicación de Scrum se encuentran:

1. Lista de requisitos priorizada (Product Backlog)

La lista de requisitos priorizada representa las expectativas del cliente respecto a los objetivos y entregas del producto o proyecto. El cliente es el responsable de crear y gestionar la lista (con la ayuda del Facilitador (Scrum Master) y del equipo, quien proporciona el coste estimado de completar cada requisito). Al reflejar las expectativas del cliente, esta lista permite involucrarle en la dirección de los resultados del producto o proyecto.

2. Lista de tareas de la iteración (Sprint Backlog)

Esta lista permite ver las tareas donde el equipo está teniendo problemas y no avanza, con lo que le permite tomar decisiones al respecto.

La lista contiene las tareas, el esfuerzo pendiente para finalizarlas y la auto-asignación que han hecho los miembros del equipo.

El progreso de la iteración y su velocidad con respecto a tareas u horas pendientes se muestra mediante un gráfico de trabajo pendiente (gráfica burndown).

3. Gráficos de trabajo pendiente (Burndown)

Un gráfico de trabajo pendiente a lo largo del tiempo muestra la velocidad a la que se está completando los requisitos. Permite extrapolar si el Equipo podrá completar el trabajo en el tiempo estimado.

Es una gráfica (Ilustración 3) que en un simple vistazo muestra la evolución del equipo respecto a los requisitos del usuario y muestra cuando se espera terminar:

- ✓ Cuanto trabajo ha sido hecho.
- ✓ Cuanto trabajo queda por hacer.
- ✓ Velocidad del equipo.
- ✓ Fecha fin esperado.
- ✓ En el eje Y cantidad de trabajo pendiente de terminar.
- ✓ En el eje X tiempo (iteraciones).

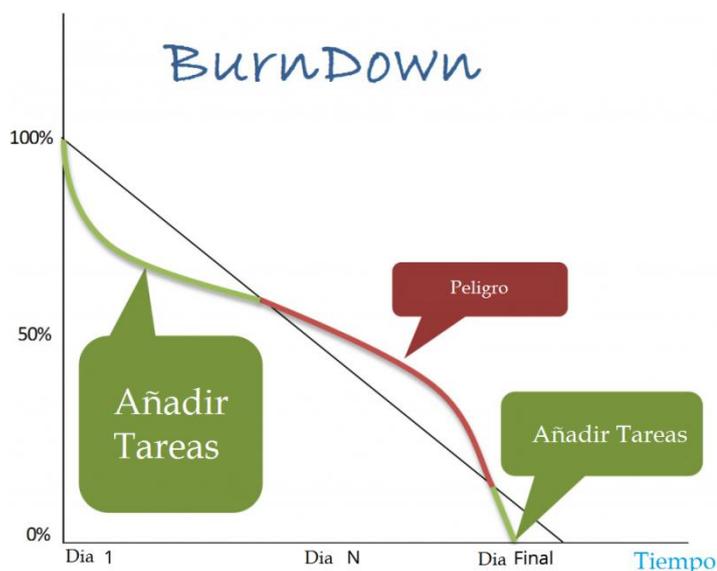


Ilustración 3 Gráfica burndown (Cockburn, 2007).

Este tipo de gráfico permite realizar diversas simulaciones: ver cómo se aplazan las fechas de entrega si se le añaden requisitos, ver cómo se avanzan si se le quitan requisitos o se añade otro equipo, etc.

3.3.2. XP (Extreme Programming)

XP (Wells, 2013) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se

define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

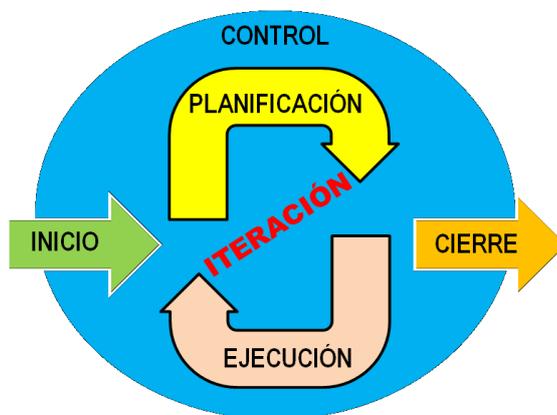


Ilustración 4 Ciclos XP (Wells, 2013).

A. Fases

Las fases de XP se muestran en la Ilustración 4.

1. Fase de exploración

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuario”. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que están basadas en datos de muy alto nivel), y podrían variar cuando se analicen en más detalle en cada iteración.

2. Fase de planificación

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas que se detallará en la sección “Reglas y Practicas”.

3. Fase de iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al

principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo.

Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

4. Fase de puesta en producción

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa.

En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste.

B. Reglas de XP

1. Planificación

La metodología XP plantea la planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando "historias de usuarios", las que sustituyen a los tradicionales "casos de uso". Una vez obtenidas las "historias de usuarios", los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

2. Historias de usuarios

Las "historias de usuarios" sustituyen a los documentos de especificación funcional, y a los "casos de uso". Estas "historias" son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo.

Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia.

3. Plan de entregas (Release Plan)

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.) de desarrollo realizadas por los desarrolladores.

4. Plan de iteraciones

Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores.

5. Reuniones diarias de seguimiento

El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar. Para no quitar tiempo innecesario del equipo, se sugiere realizar estas reuniones en círculo y de pie.

C. Diseño

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

1. Simplicidad

Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

Características fundamentales del código: Testeable, legible, comprensible y explicable.

2. Metáforas

Una “metáfora” es algo que todos entienden, sin necesidad de mayores explicaciones.

La metodología XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, y guiar la estructura y arquitectura del mismo. Por ejemplo, puede ser una guía para la nomenclatura de los métodos y las clases utilizadas en el diseño del código. Tener nombres claros, que no requieran de mayores explicaciones, redundará en un ahorro de tiempo.

3. Solución “spike”

Una solución “spike”, es una solución muy simple para plantear posibles soluciones, de manera, que solamente se aborda el problema en concreto y se aísla de otro tipo de preocupaciones.

4. Refactorización

La recodificación consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible.

D. Implementación

1. Cliente disponible

Las historias de usuario son escritas por el cliente con la ayuda de los desarrolladores, además de establecer la prioridad de las mismas. Su presencia asegura que los desarrollos cubren toda la funcionalidad descrita.

El cliente es imprescindible a la hora de realizar pruebas funcionales, en caso de error él será el encargado de decidir si el código puede pasar a producción o no.

2. Estándares de codificación

Todos los programadores deben escribir y documentar el código en la misma manera. El código debe seguir los estándares de codificación. Las normas de codificación ayudan a mantener el código legible y fácil de mantener y refactorizar.

3. Implementación dirigida por las pruebas unitarias

En las metodologías tradicionales, la fase de pruebas, incluyendo la definición de las pruebas, se realiza al final del proyecto, o al final del desarrollo de cada módulo. La metodología XP propone un modelo inverso, en el que, lo primero que se escriben los test que el sistema debe pasar.

Las pruebas a las que se refiere esta práctica, son las pruebas unitarias, realizados por los desarrolladores. La definición de estos test al comienzo, condiciona y dirige el desarrollo.

4. Programación en parejas

XP promueve que todo el código sea escrito en parejas trabajando en el mismo ordenador. La programación en parejas incrementa la calidad del código sin impactar en la fecha de entrega.

En contra de lo que parece, dos personas que trabajan en un mismo equipo añadirán la misma funcionalidad que dos personas trabajando por separado, excepto que el código será de mucha mayor calidad.

5. Integración secuencial

Todos los desarrolladores necesitan trabajar siempre con la “última versión”.

Realizar cambios o mejoras sobre versiones antiguas causan graves problemas, y retrasan al proyecto. Es por eso que XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores.

Idealmente, todos los días deben existir nuevas versiones publicadas. Para evitar errores, solo una pareja de desarrolladores puede integrar su código a la vez.

6. Propiedad colectiva del código

La propiedad colectiva anima a todos los miembros del equipo a aportar nuevas ideas. Cualquier desarrollador puede cambiar líneas de código para añadir funcionalidad, solucionar errores, mejorar el diseño o refactorizar el código.

7. Ritmo constante

La metodología XP indica que debe llevarse un ritmo sostenido de trabajo.

Anteriormente, ésta práctica se denominaba “Semana de 40 horas”. Sin embargo, lo importante no es si se trabajan, 35, 40 o 42 horas por semana. El concepto que se desea establecer con esta práctica es el de planificar el trabajo de manera de mantener un ritmo constante y razonable, sin sobrecargar al equipo.

E. Pruebas

1. Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por

otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Test-driven programming”). Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo.

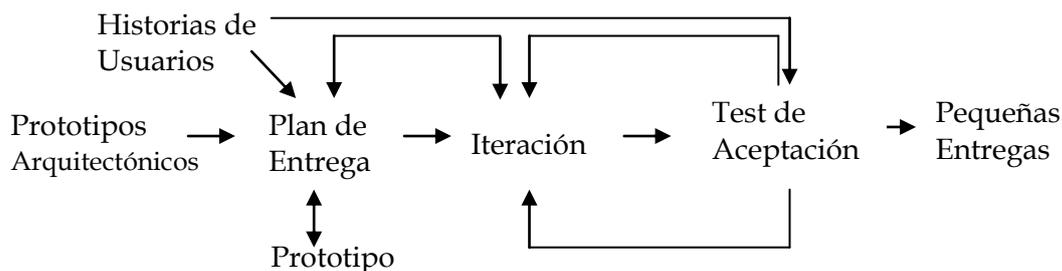


Ilustración 5 Proyecto XP (Wells, 2013).

E. Valores en XP

Simplicidad: Hacer exactamente lo que se ha pedido, no más.

Comunicación: La comunicación entre los componentes del equipo XP es fundamental.

Dado que la documentación es escasa, el diálogo frontal, cara a cara, entre desarrolladores, gerentes y el cliente es el medio básico de comunicación. Una buena comunicación tiene que estar presente durante todo el proyecto.

Retroalimentación: Siempre tener en cuenta la valoración del cliente una vez que se hace una entrega e intentar mejorar haciendo cambios en el proceso si es necesario.

Coraje: Se trata que el equipo asuma la responsabilidad de su trabajo, tanto si es un éxito como un fracaso, además de ser emprendedor a la hora de implementar cambios en la aplicación (refactorizaciones...).

G. Roles

1. Cliente

El cliente es el responsable de conducir el proyecto. Define el proyecto y sus objetivos.

Cuanto más preciso es su trabajo y cuanto mayor sea su involucración, mayores serán las oportunidades de éxito.

Toma decisiones de negocio y debe entender los cambios del mismo a largo del tiempo: identificando si una historia tiene el mismo valor ahora que cuando se identificó, si se puede retrasar o simplificar, además de tener en consideración las implicaciones técnicas detectadas por los desarrolladores: es mejor cambiar el orden de la planificación de las historias de usuarios por sugerencia de los desarrolladores.

2. Programador

Una vez que se han comprendido las historias de usuario, el XP adjudica a los programadores la responsabilidad de tomar decisiones técnicas. Los desarrolladores estiman el tiempo que les va a tomar cada historia. Transforma las historias de usuario a código.

3. Encargado de pruebas (Tester)

El encargado de pruebas ayuda al cliente a definir y escribir las pruebas de aceptación de las historias de usuario. Este rol en un equipo XP también es responsable realizar test periódicamente y informar de los resultados al equipo. A medida que el volumen de pruebas aumenta, el encargado de pruebas necesitará una herramienta para crear y mantener la batería de pruebas, ejecutarlas y obtener los resultados más rápidamente.

4. Encargado de seguimiento (Tracker)

Hace el seguimiento de acuerdo a la planificación. La métrica más importante para XP es la velocidad del equipo, que se define como el tiempo ideal estimado para las tareas frente al tiempo real dedicado. Esta métrica ayuda a determinar si el proyecto está dentro del tiempo de la iteración.

5. Entrenador (Coach)

No es un rol cubierto en todos los equipos de XP. Su papel es guiar y orientar al equipo, especialmente cuando un equipo comienza a trabajar siguiendo la metodología XP. Esto se debe a que no es fácil aplicar XP de forma consistente. Aunque son prácticas de sentido común, se necesita un tiempo para interiorizarlas. También hay situaciones especiales en las que se requiere la sabiduría de un especialista en XP para aplicar sus normas frente a un obstáculo en el proyecto.

6. Gestor (Big Boss)

Es el gerente del proyecto, debe tener una idea general del proyecto y estar familiarizado con su estado. El cliente puede asumir este papel.

3.3.3. Kanban

Su objetivo es gestionar de manera general cómo se van completando tareas, pero en los últimos años se ha utilizado en la gestión de proyectos de desarrollo software.

Las principales reglas de Kanban (Mendes Calo, 2010) son las siguientes:

- ✓ Visualizar el trabajo y las fases del ciclo de producción o flujo de trabajo.
- ✓ Determinar el límite del “trabajo en curso” (WIP - Work In Progress).
- ✓ Medir el tiempo en completar una tarea (Lead time).

A. Visualizar el trabajo y las fases del ciclo de producción o flujo de trabajo

Kanban se basa en el desarrollo incremental, dividiendo el trabajo en partes. Una de las principales aportaciones es que utiliza técnicas visuales para ver la situación de cada tarea, y que se representa en pizarras llenas de post-it como se muestra en la Ilustración 6.



Ilustración 6 Muro Kanban (Mendes Calo, 2010).

El objetivo de esta visualización es que quede claro el trabajo a realizar, en qué está trabajando cada persona, que todo el mundo tenga algo que hacer y el tener clara la prioridad de las tareas.

Las fases del ciclo de producción o flujo de trabajo se deben decidir según el caso, no hay nada acotado. En la figura se han puesto un conjunto de fases de ejemplo.

B. Determinar el límite del trabajo en curso (Work In Progress)

En Kanban se debe definir cuantas tareas como máximo puede realizarse en cada fase del ciclo de trabajo (ejemplo, como máximo 4 tareas en desarrollo, como máximo 1 en pruebas, etc.), a ese número de tareas se le llama límite del “work in progress”. A esto se añade otra idea tan razonable como que para empezar con una nueva tarea alguna otra tarea previa debe haber finalizado.

C. Medir el tiempo en completar una tarea (Lead time)

El tiempo que se tarda en terminar cada tarea se debe medir, a ese tiempo se le llama “lead time”. El “lead time” cuenta desde que se hace una petición hasta que se hace la entrega.

Aunque la métrica más conocida del Kanban es el “lead time”, normalmente se suele utilizar también otra métrica importante: el “cycle time”. El “cycle time” mide desde que el trabajo sobre una tarea comienza hasta que termina. Si con el “lead time” se mide lo que ven los clientes, lo que esperan, y con el “cycle time” se mide más el rendimiento del proceso.

Puede haber más métricas, pero las anteriores son las realmente importantes y necesarias para el control y mejora continua.

D. Roles

La metodología Kanban no prescribe roles. Tener un papel asignado y las tareas asociadas a dicho papel crean una identidad en el individuo. Por lo tanto, pedir que adopten un nuevo papel o un nuevo puesto de trabajo puede ser entendido como un ataque a su identidad. Habría una resistencia al cambio. Kanban trata de evitar esa resistencia emocional, entiende que la ausencia de papeles es una ventaja para el equipo.

3.3.4. Scrumban

Scrumban (Mendes Calo, 2010) es una metodología derivada de los métodos de desarrollo Scrum y Kanban. Es un modelo de desarrollo especialmente adecuado para proyectos de mantenimiento o proyectos en los que las historias de usuarios (requisitos del software) varíen con frecuencia o en los cuales surjan errores de programación inesperados durante todo el ciclo de desarrollo del producto. Para estos casos, los

sprints (periodos de duración constante en los cuales se lleva a cabo un trabajo en sí) de la metodología Scrum no son factibles, dado que los errores/impedimentos que surgirán a lo largo de las tareas son difíciles de determinar y por lo tanto, no es posible estimar el tiempo que conlleva cada historia. Por ello, resulta más beneficioso adoptar flujo de trabajo continuo propio del modelo Kanban.

A. De Scrum

- ✓ Roles: Cliente, equipo (con los diferentes perfiles que se necesiten).
- ✓ Reuniones: reunión diaria.
- ✓ Herramientas: pizarra.

B. De Kanban

- ✓ Flujo visual.
- ✓ Hacer lo que sea necesario, cuando sea necesario y solo la cantidad necesaria.
- ✓ Limitar la cantidad de trabajo (WIP).
- ✓ Optimización del proceso.

En la Tabla 11 se observa el método SCRUMBAN (Scrum vs Scrumban).

Tabla 11 Scrum vs Scrumban (Mendes Calo, 2010).

Normas	Scrum	Scrumban
Pizarra / Herramientas	Pizarra Backlogs Gráfica burn-down	Pizarra
Reuniones	Reunión diaria Planificación Retrospectiva	Reunión diaria
Iteraciones	Sí, Sprints	No, flujo continuo
Estimaciones	Sí	No
Esquipo	Multidisciplinar	Puede ser especializado
Roles	Product Owner Scrum Master Equipo	Equipo + otros
WIP (Work In Progress)	Controlado por el contenido del Sprint	Controlado por el estado de la tarea.
Cambios	Se pasan al siguiente Sprint	Se añaden al tablero en la columna "TO DO".
Impedimentos	Solución inmediata	Se evitan.

3.3.5. Otras Metodologías Ágiles

Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito el manifiesto ágil y coinciden con los principios enunciados anteriormente, cada metodología tiene características propias y hace hincapié en algunos aspectos más específicos. A continuación se resumen otras metodologías ágiles. La mayoría de ellas ya estaban siendo utilizadas con éxito en proyectos reales pero les faltaba una mayor difusión y reconocimiento.

- ✓ **Crystal Methodologies** (Crystal Methodologies, 2013). Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).
- ✓ **Dynamic Systems Development Method (DSDM)** (Agile, 2001). Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.
- ✓ **Adaptive Software Development (ASD)** (Schwaber K., 2001). Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

- ✓ **Feature-Driven Development (FDD)** (Metodología FDD, 2012). Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.

3.4. Comentarios y Discusiones

En este capítulo se muestra el funcionamiento de las metodologías ágiles más utilizadas actualmente. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo. Esto ha llevado hacia un interés creciente en las metodologías ágiles.

El contexto ideal para el desarrollo ágil es aquel donde el equipo de desarrollo tiene una estrecha colaboración con el cliente, en el cual, si existen cambios en los requisitos del producto, el cliente entiende y acepta probables modificaciones del alcance inicialmente acordado. En este contexto ideal, las ventajas de estimar se diluyen puesto que si bien existen entregas frecuentes, ellas pueden modificarse sobre la marcha en cuanto a su alcance, es decir, el éxito está asegurado puesto que, acercándose a una entrega el equipo puede acordar con el cliente quitar alcance sin que esto signifique necesariamente un fracaso en cuanto a sus expectativas.

Una cuestión que parece estar consensuada en el contexto ágil es que la estimación se hace mediante "el juicio del experto", es decir, la decide el desarrollador, de forma individual o consensuada con todo el equipo de desarrollo.

Si bien la literatura asociada a las metodologías ágiles no es bastante extensa, como es de esperar, no existen recetas generales respecto a estimación de costos. En el siguiente capítulo se demostrará cómo es posible lograr una estimación de costos utilizando metodologías ágiles de desarrollo.

Capítulo 4

Propuesta Metodológica

En este trabajo se presenta un nuevo enfoque para estimar costo de desarrollos de software utilizando metodologías ágiles. El objetivo principal de este trabajo es llegar a determinar en etapas tempranas del desarrollo de software la estimación de costos en lo que se refiere exclusivamente al esfuerzo humano en el desarrollo de software incorporando pautas de calidad en la estimación.

4.1. Introducción

En la actualidad se vive en un mundo de licitaciones y contrataciones de terceros para desarrollos de software y esto aun diciendo que se utiliza un enfoque ágil es imposible no predecir el costo final del producto.

En metodologías ágiles trabajan en equipo entre el principal interesado en el producto, "El Cliente", y las personas que se encargarán de desarrollarlo e implementarlo, "Los Desarrolladores", por ello no suele ser necesario conocer el costo del producto como un todo, debido a que el cliente conoce los detalles de la metodología.

Sin embargo, la necesidad de ambas partes de conocer lo que costará el producto es de suma importancia, de ahí el énfasis de esta tesis en solucionar esta brecha y resolver el problema.

Entre las medidas de estimación tradicionales existen los Puntos de Función y COCOMO II. Abordando la estimación por líneas de código (SLOC), se ve ciertas dificultades entre las cuales se podrían nombrar las siguientes:

- ✓ El número de SLOC dependerá en gran parte de la habilidad y destreza del programador que lo esté realizando.
- ✓ Existe una relación directa entre el nivel del lenguaje (lenguaje de alto nivel o lenguaje ensamblador) y la producción de trabajo.
- ✓ El número real de SLOC no se sabrá hasta que el desarrollo del software esté terminado.

- ✓ No hay un método aceptado para contar líneas de código, aunque si existen diversas aplicaciones que contribuyen a la realización de esta labor, entre las cuales se podrían nombrar: CodeCount, CCCC, K-LOC Calculator, EZMetrix y Code Analyzer.

Con relación a Puntos de Función resulta en lo mismo al estimar con COCOMO II, este cuenta con una conversión de puntos de función a SLOC y volvemos a los mismos inconvenientes citados anteriormente.

En el capítulo 2 se puede observar que en COCOMO II, se tiene importantes parámetros bien ajustados, estos actúan de gran impacto en el resultado de las mediciones, también se habla de sus factores de escala y multiplicadores de esfuerzo. Dichos parámetros dan énfasis para lograr una buena calidad a la hora de estimar. Este es uno de los objetivos al cual se desea llegar en nuestro trabajo, por este motivo se ha elegido COCOMO II como enfoque para obtener una nueva metodología de estimación.

Se propone así una estimación orientada a metodologías ágiles considerando calidad en la estimación con los factores de escala y los multiplicadores de esfuerzo de COCOMO II y eliminando el conteo de líneas de código (LOC).

4.2. Planificación

A continuación se describe el marco de trabajo que compone una serie de etapas, que definen pasos preestablecidos y procedimientos que se siguieron para componer la nueva metodología de estimación de desarrollo de software.

Las definiciones se basaron en seis etapas que aseguran el cumplimiento del objetivo:

- ✓ Análisis de parámetros.
- ✓ SCRUM, como gestión del proyecto.
- ✓ COCOMO II, como pauta de calidad.
- ✓ Definición de la propuesta.
- ✓ Formulas y cálculos resultantes.
- ✓ Evaluación de la propuesta.

4.2.1. Etapa 1 - Análisis de Parámetros

Antes de comenzar a comentar los parámetros seleccionados para la propuesta, se procede a una breve comparación de los parámetros utilizados en metodologías tradicionales para estimar con COCOMO II y aquellos parámetros que se puede disponer utilizando SCRUM como gestión del proyecto de desarrollo de software.

COCOMO II usa Puntos de Función como método de estimación para desarrollos de software orientados a objetos, en la Tabla 12 se observa la comparación entre parámetros de COCOMO II y parámetros manejados en SCRUM.

Tabla 12 Parámetros de COCOMO & SCRUM (fuente propia).

COCOMO II	SCRUM
Casos de Usos	Historias de Usuarios
Número de Entradas de Usuarios	
Número de Salidas de Usuarios	
Número de Peticiones de Usuarios	
Número de Archivo de Usuarios	MER - Diagrama de Modelo Entidad - Relación
Número de Interfaces de Usuarios	Prototipos de Interfaz
Factor de Ponderación (Simple Medio Complejo)	Complejidad de Cada Historia
Puntos de Función a Líneas de Códigos	Suma de Complejidades de Historias
Factores de Costos	Porcentaje de dedicación
Multiplicadores de Esfuerzos	
Esfuerzo medido en Meses Personas	Sprint Iteraciones

Actualmente no se cuenta con ajustes ni calibraciones para convertir puntos de historias estimadas con base a complejidades a LOC, aunque tampoco ese es el objetivo ya que líneas de código en desarrollo de software orientado a objetos se quedaron obsoletas debido a que esto depende del estilo y conocimiento del programador.

4.2.2. Etapa 2 - SCRUM, como Gestión del Proyecto.

Se ha elegido SCRUM como gestión de proyecto ágil por ser un modelo de referencia a nivel internacional. SCRUM se define como un conjunto de prácticas, roles y parámetros útiles como punto de partida para la definición de la nueva metodología de estimación de costo de desarrollo de software.

A continuación se cita los artefactos y herramientas utilizados para la nueva propuesta.

1. Pilas de Productos.
2. Historias de Usuario.
3. Complejidad de cada Historia.
4. Prioridad por cada Historias.
5. Sprint.

A continuación se detallan cada una de ellas:

1. Pila de producto

La pila de producto es el corazón del Scrum. Es donde inicia la descripción de lo preparado para desarrollar. Es básicamente una lista de requisitos o tareas en donde se describen lo que el cliente desea y qué se debe hacer, a esto lo llamamos *Historias de Usuarios*, o simplemente elementos de la Pila. En la Tabla 13 se observa qué compone una Pila de Producto, se considera como ejemplo a utilizar en la nueva propuesta.

Tabla 13 Pila de Producto (fuente propia).

Pila de Producto				
Nombre	Prioridad	Complejidad	Descripción	Notas
Préstamo de Libro	300	4	Cómo cliente quiero que los socios puedan pedir prestado un libro.	No permitir en caso que ya tengan tres libros prestados
Total		4		

2. Historias de usuarios

Cada registro de la pila de producto son las historias de usuarios. Comúnmente se dice que las metodologías ágiles usan historias de usuarios y las tradicionales el requisito funcional o casos de usos. En realidad hay muchos aspectos que los diferencian pero principalmente las historias de usuarios no dicen "el cómo" se debe hacer la tarea, sólo dice "el qué" se debe hacer.

En la Ilustración 7 se observa la estructura de una historia de usuario que está formada por los siguientes datos:

- ✓ ID de la Historia, identificación única para cada Historia.
- ✓ Nombre breve o Título.
- ✓ Descripción de la funcionalidad en forma de diálogo o monólogo del usuario.
- ✓ Pruebas de aceptación o criterio de validación y verificación que usado para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita.
- ✓ Complejidad Estimada por el equipo de desarrollo.
- ✓ Prioridad estimada para la Historia.
- ✓ Dependencias si posee o resulte necesaria.

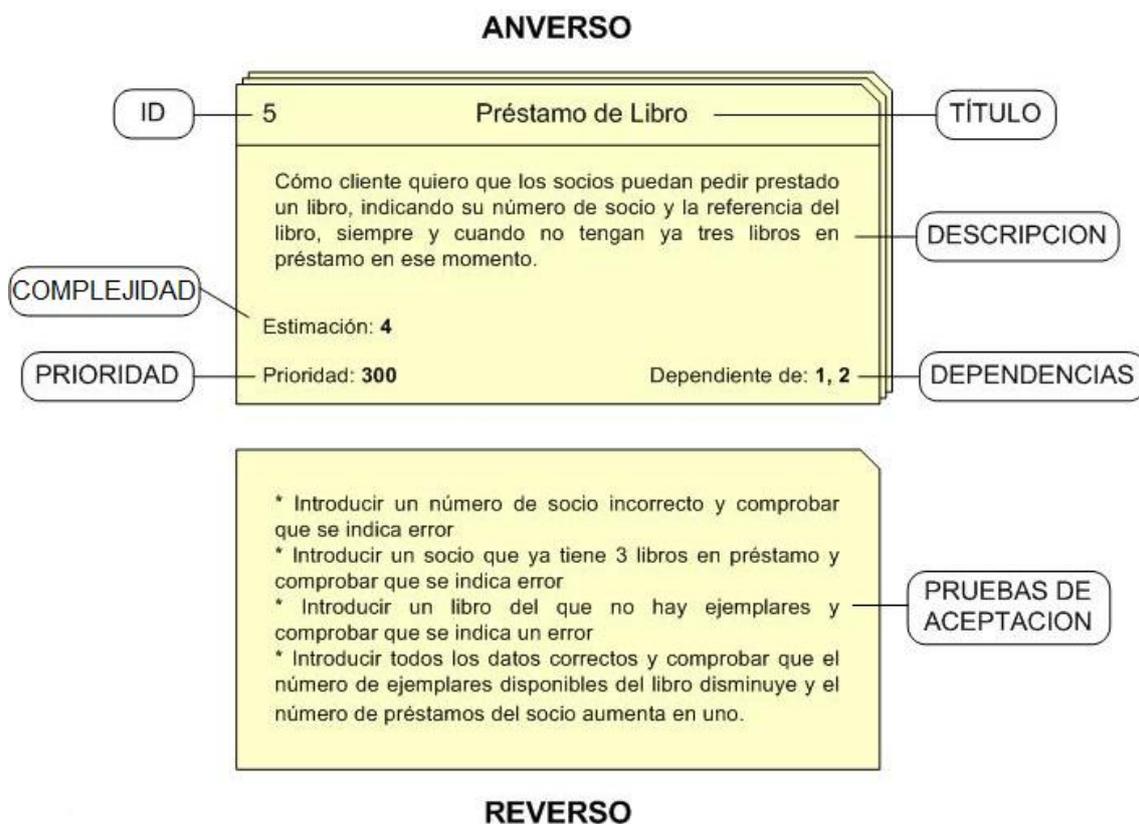


Ilustración 7 Historia de Usuarios (Schwaber K., 2001)

3. Complejidad

La complejidad se refiere al esfuerzo que demanda la realización de una historia de usuario. Determinar la complejidad es una labor del equipo: todos los

miembros del equipo deben involucrarse para estimar cada historia. Estas estimaciones pueden realizarse a través de técnicas como Pocker Planning (Ilustración 8).

Planning Poker es una técnica de estimación para proyectos ágiles, propuesta que se inicio con Grenning en el año 2002 y concluyo Mike Cohn. Esta técnica permite hacer una estimación inicial del proyecto de manera consensuada con base en los requisitos.

Definir la complejidad es una tarea difícil en etapas tempranas del desarrollo por eso se lleva a cabo a través de una reunión utilizando la técnica de “Juicio de Expertos” donde todos llegan a un acuerdo con respecto al valor a determinar para cada historia, a continuación algunas consideraciones a tener en cuenta.

- ✓ A la hora de planificar, no se sabe exactamente quién implementará qué partes de cada historia, pero en consenso el equipo va estimando el valor promedio de complejidad.
- ✓ Las historias de Usuarios involucran a bastantes personas y de diferentes áreas de experiencia (diseño de interfaz de usuario, programación, pruebas, etc.).

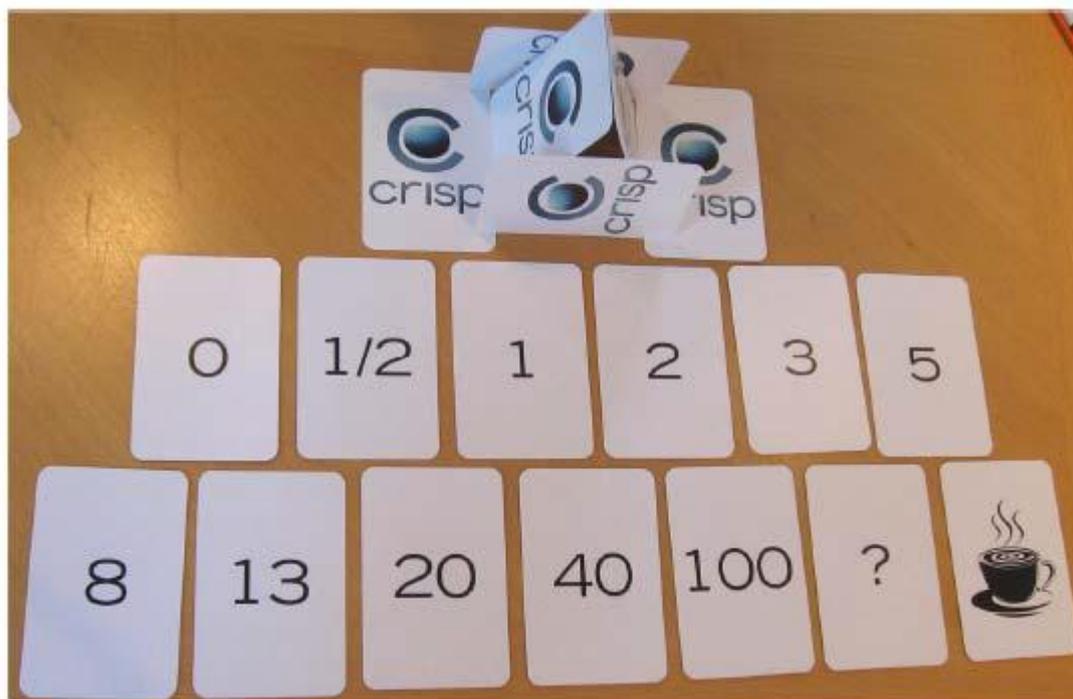


Ilustración 8 Planning poker (Cockburn, 2007)

Para estimar la complejidad con la técnica de Planning Poker, se juega con una baraja de cartas numeradas siguiendo la serie de Fibonacci como indica la Ilustración 8.

El valor que se le asigna a cada número se puede definir como: días de trabajo, días ideales o simplemente puntos. Es el líder de grupo quien debe indicar esto dependiendo del proyecto antes de comenzar el proceso de estimación.

Durante la reunión de estimación, se le da a cada persona un mazo de cartas. La reunión se planifica de la siguiente manera:

- ✓ Se define un cliente. Esta persona es la voz del cliente dentro de la organización. Es la que debe conocer que es lo que quiere el cliente para cada funcionalidad o tarea que se va a estimar.
- ✓ Se define un moderador, el cual no va a estimar. Puede ser el perfil del Líder de Equipo o Líder de Proyecto.
- ✓ El moderador escoge una funcionalidad de la lista. El cliente da una pequeña introducción al tema. Es la oportunidad que tiene el equipo para hacer preguntas y clarificar puntos oscuros y/o riesgos de la tarea a realizar.
- ✓ Cada persona escoge una carta de su mazo (sin mostrarla).
- ✓ Todos al mismo tiempo muestran sus cartas.
- ✓ El moderador entrevista a la gente que haya realizado las estimaciones más altas y bajas (con respecto al promedio) para averiguar de dónde surge la diferencia. También se invita a estas personas a discutir entre ellas para llegar a un acuerdo.

Es importante no extender las reuniones demasiado tiempo. Terminar las discusiones por consenso o por tiempo, siempre se puede dejar una funcionalidad sin consenso y volver luego de seguir con las demás (tal vez analizando otra funcionalidad sirvió para entender mejor la primera). Llegado el caso, también se puede dejar alguna funcionalidad con un rango amplio (poco conocimiento de lo que hay que hacer, estimación con mucho error), siempre y cuando sea un porcentaje muy bajo respecto al total de funcionalidades estimadas. En el caso hipotético que no se pueda estimar correctamente muchas funcionalidades, hay que definir mejor las tareas. Algunas ideas son: el cliente debe proveer más información, dividir las tareas en dos o más partes para disminuir la complejidad, posponer la reunión de estimación y asignar un tiempo a investigación sobre el tema, etc.

A menor número, menor esfuerzo demanda una historia de usuario. Y cuanto más elevado es el valor, mayor esfuerzo.

4. Prioridad

Los ítems de la pila de producto deben tener un orden de prioridad, cuya base se apoye en responder a las siguientes preguntas:

¿Cuáles son las funcionalidades con mayor prioridad desde la perspectiva del negocio? o ¿cuáles son las funcionalidades que tendrán mayor impacto en la reducción de riesgos?.

Además se deben considerar las siguientes características:

- ✓ Beneficios de implementar una funcionalidad.
- ✓ Pérdida o costo que demande posponer la implementación de una funcionalidad.
- ✓ Riesgos de implementarla.
- ✓ Coherencia con los intereses del negocio.
- ✓ Valor diferencial con respecto a productos de la competencia.

5. Sprint

El Sprint es el ritmo de los ciclos de Scrum. Está delimitado por la reunión de planificación del sprint y la reunión retrospectiva. Una característica importante es que una vez que se fija la duración del sprint, esta es inamovible. La mayoría de los equipos eligen entre dos a cuatro semanas de duración. Diariamente, el equipo realiza una reunión de seguimiento muy breve. Al final del sprint se entrega el producto al cliente en el que se incluye un incremento de la funcionalidad que tenía al inicio del sprint.

Para comenzar a organizar el Sprint se calcula la velocidad del equipo, esto es el número de puntos ideales de trabajo que es capaz de asumir el equipo en el Sprint. Es necesario tener en cuentas los siguientes parámetros:

- ✓ Duración del Sprint.

Se define la duración del Sprint, esto significa que debes determinar cuánto va durar en días el sprint. Una de las incógnitas que surgen es cuál es la mejor duración del Sprint. Los Sprint cortos que duran entre 8 a 15 días permiten ser “ágil”, es decir, cambiar de dirección frecuentemente. Sprint corto interactúa con el cliente en un lapso menos de tiempo, cuanto más corto, más entregas y más frecuencias.

Con sprint largo, el equipo tiene más tiempo para conseguir impulso, tienen más espacio para recuperarse de los problemas que surjan y cumplir la meta del Sprint

- ✓ Número de desarrolladores del equipo y su unidad de medida diaria en puntos de historias llamadas *velocidad inicial* del equipo.

Esto es llegar a un promedio equivalente de complejidades de historias que cada desarrollador puede atender diariamente y luego calcular el valor promedio de trabajo que es capaz de asumir el equipo.

En Tabla 14 se puede ver el ejemplo, se estima por cada desarrollador el valor de puntos de historias que pueda resolver en un día de trabajo. Como se observa en la Tabla 14, el desarrollador 3 es el que menos puntos de historia pueda resolver, esto puede darse en el caso de un programador junior.

Tabla 14 Velocidad del Equipo (fuente propia).

	Puntos de Historias	Días de trabajo en el Sprint
Desarrollador 1	4	1 día
Desarrollador 2	6	1 día
Desarrollador 3	3	1 día
Desarrollador 4	8	1 día

La sumatoria de los puntos de historias indica lo que el equipo es capaz de asumir diariamente en el Sprint.

Ejemplo:

$(4+6+3+8) = 21$ Puntos de Historias al día se puede realizar por un Equipo de 4 desarrolladores.

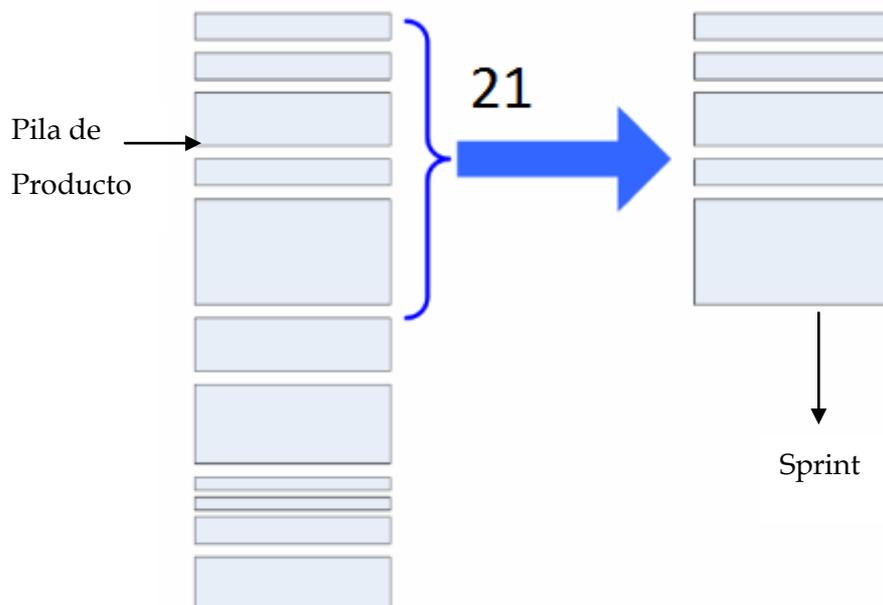


Ilustración 9 Pila de Producto & Sprint 1 (fuente propia).

En la Ilustración 9 se observa la definición del Sprint con respecto a la cantidad de puntos de historias que el equipo pueda cumplir. Cada rectángulo representa una historia, todas ordenadas por prioridad (importancia). La historia más importante está al principio de la lista. El tamaño de cada rectángulo representa el tamaño de la historia (es decir la complejidad estimada en cada historia). La altura del corchete representa la *velocidad inicial* del equipo, es decir, cuántos puntos de historia se cree que el equipo pueda completar durante el Sprint.

La Pila de Sprint de la derecha es una instancia de las historias la Pila de Producto. Representa las historias a las que el equipo se compromete durante este Sprint. *El equipo* decide cuántas historias incluirá en el Sprint y cuántos días con lleva la duración de un Sprint.

4.2.3. Etapa 3 - COCOMO II, como Pauta de Calidad.

El COCOMO II usa cinco factores de escala (SF) que servirán para ajustar las economías y deseconomías de escala que se presentan en proyectos de software de diferentes tamaños. A continuación se muestra la fórmula de la ecuación del cálculo los factores de escala.

$$E = B + 0.01 \cdot \sum_{j=1}^5 SF_j$$

En esta fórmula B es una constante, el valor para este parámetro fue obtenido en la calibración de 161 proyectos en las base de datos de COCOMO II y es inicialmente igual a 0.91 (Boehm B., 2000).

Cuando el valor E es inferior a 1, se tendrá una economía de escala. Esto significa que cuando el tamaño del proyecto se duplica, el efecto sobre el esfuerzo será menor que el doble. Sin embargo, cuando E es mayor que 1 el proyecto muestra una diseconomía de escala lo que doblaría el tamaño y podría provocar un aumento de más del doble en el esfuerzo. La sumatoria $\sum_{j=1}^5 SF_j$ consiste en 5 factores de escala: Precedencia, Flexibilidad de Desarrollo, Resolución de Riesgos y Arquitectura, Cohesión del Equipo y Maduración del Proceso. Los factores de escala son definidos de acuerdo al nivel de cada proyecto, cada uno tiene un rango de niveles de calificación muy baja a muy alta. Cada nivel de calificación tiene un peso. Este peso se determina utilizando inicialmente la base de datos de COCOMO II de 161 proyectos.

Además, COCOMO II cuenta también con 17 multiplicadores para el diseño Post-Arquitectura.

Los multiplicadores de esfuerzo en COCOMO, son características del proyecto que tienen un efecto lineal en el esfuerzo de un proyecto. El modelo Post-Arquitectura define 17 multiplicadores de esfuerzo, similar a los factores de escala, los multiplicadores de esfuerzo tienen un rango de niveles definido, cada uno con un peso inicial definido con la base de datos de COCOMO II de 161 proyectos. Cada multiplicador de esfuerzo con excepción de "Planificación del Desarrollo", podría ser considerado como un módulo individual. Se podría dividir los multiplicadores de esfuerzo en varias clases:

Factores del Producto (fiabilidad del software, tamaño de la bases de datos, complejidad del producto, reusabilidad en el desarrollo y documentación), Factores de la Plataforma (volatibilidad de la plataforma, limitación en el tiempo de ejecución, limitación de almacenamiento principal), Factores del personal (capacidad de análisis, capacidad del programador, continuidad del personal, experiencia en la aplicaciones, experiencia en la plataforma, experiencia en el lenguaje de programación y en las herramientas).

Estos multiplicadores no sólo sirven como parámetros a la hora de estimar el costo sino también como referencia para ir conociendo específicamente en qué área del

desarrollo hay más necesidades, esto surge en medio de las mediciones de cada multiplicador.

En esta propuesta se decide utilizar esos factores de escala y los multiplicadores de esfuerzo del COCOMO II y resolver la incertidumbre de llegar a lo más exacto en la estimación.

4.2.4. Etapa 4 - Definición de la Propuesta

Se presenta esta nueva metodología de estimación de costos definida bajo el enfoque de metodologías ágiles, que sirve para estimar proyectos a través de métodos sencillos y aplicables a equipos de desarrollo ágil. La misma ofrece estándares, recomendaciones y artefactos que requiere una metodología de estimación de costos.

Se define esta arquitectura (Ilustración 10), con su principal objetivo, “la estimación de costo” al proceso de desarrollo de software y, en segundo lugar la planificación de los tiempos de las diferentes actividades necesarias en el proyecto.

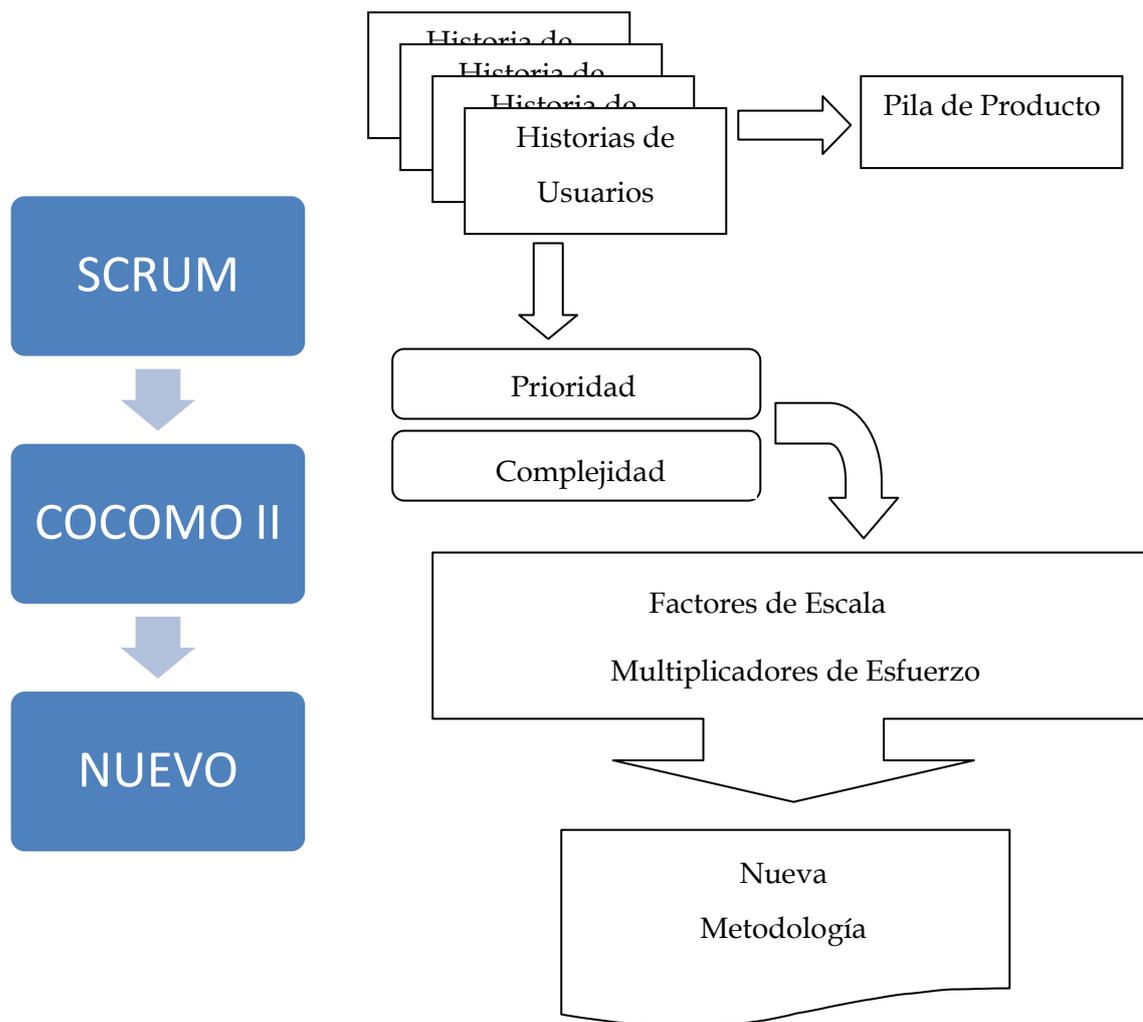


Ilustración 10 Nueva Arquitectura (fuente propia).

Una explicación práctica de la idea inicial, se presenta en la Ilustración 10, donde se puede visualizar que comienza con SCRUM. Se debe comprender que no sólo se utilizan los artefactos del SCRUM, también se debe cumplir las características que ofrece esta metodología para lograr el éxito en la estimación.

El COCOMO II será utilizado como una medida de productividad por sus principales fortalezas de controladores de costos (factores de escala y multiplicadores de esfuerzo). Estos controladores de costo capturan características del desarrollo de software que afectan al esfuerzo que se necesita para completar el proyecto. Por lo tanto, estos factores ofrecen una información verdaderamente útil. Al observar las calificaciones de los controladores de costo para cada proyecto se puede indicar qué características del proyecto tuvieron o no una influencia en el esfuerzo estimado.

También se debe considerar que existen otros costos que intervienen en el producto, por ejemplo costos de hardware, pero el aumento del costo de software hace necesario enfocarse principalmente en la construcción de software. Por ello, cuando aquí se habla de estimación de costos se hace referencia exclusivamente al esfuerzo humano que ha sido necesaria en el proyecto de desarrollo de software.

4.2.5. Etapa 5 - Formulas y Cálculos Resultantes

Para cuantificar el desarrollo de software, se necesita fijar medidas de magnitudes para obtener los resultados precisos, a partir de las cuales se evalúen los resultados, la nueva metodología propone fórmulas que se detallan a continuación para lograr la estimación de costos.

Esta metodología de medición puede dividirse en diez pasos:

1. Determinar el conteo de puntos de historias.
2. Determinar la duración del sprint.
3. Determinar la velocidad del equipo.
4. Determinar el número de personas en el equipo.
5. Determinar el factor de escala del proyecto.
6. Determinar los multiplicadores de esfuerzo del proyecto.
7. Determinar los puntos de historias ideales para el sprint.
8. Determinar los puntos de historias ajustados para el sprint.
9. Determinar la cantidad de iteraciones para el proyecto.
10. Determinar el costo final del desarrollo de software.

A continuación se detallan cómo se obtiene cada uno.

1. Total de Puntos de Historias (**TStoryPoint**)

Es el resultado de la suma de todas las complejidades de las historias incluidas en la pila de producto. Para ello primero se cargan las historias de usuario a la pila de producto y se estima su complejidad como se describe en la Etapa 2, luego se suman dichas complejidades.

2. Duración de Sprint (**DSprint**)

Al iniciar el proyecto el equipo de desarrollo determina el número de días ideal para la duración del Sprint, esto se realiza a través de una reunión con los participantes involucrados en el desarrollo del software.

3. Velocidad Inicial del Equipo (**V**)

Se determina por cada persona del equipo el número de puntos de historias que cada uno puede asumir diariamente, la suma de este valor es la velocidad inicial que el equipo puede asumir en un día de sprint. Esto se realiza con una reunión entre los participante del desarrollo de software.

4. N° de Personas en el Equipo (**N**)

Determina el número de personas involucradas en el proyecto de desarrollo de software.

5. Puntos de Historias del Proyecto (**StoryPoint**)

Determina la cantidad de puntos de historias asumidas para el sprint, este resulta de multiplicar la velocidad inicial del equipo (**V**) por la duración indicada para el Sprint (**DSprint**). A continuación se puede observar la formula resultante:

$$\mathbf{StoryPoint} = \mathbf{V} * \mathbf{DSprint}$$

6. Exponente de Factor de Escala (**E**)

Resulta de sumar la constante **B** que es igual a 0.091 y la suma de los cinco factores de escala multiplicando por 0.01 como indica la fórmula siguiente:

$$E = B + 0.01 \cdot \sum_{j=1}^5 SF_j$$

7. Multiplicadores de Esfuerzo (**EM**)

Resulta de multiplicar los diecisiete multiplicadores definidos en la Etapa 3 como indica la fórmula a continuación:

$$\prod_{i=1}^n EM_i$$

8. Puntos de Historias Ajustadas (**StoryPointAjust**)

Se determinan los puntos de historias ajustado para el sprint, resulta de elevar el resultado de los puntos de historias (**StoryPoint**) al exponente (**E**), factor de escala por los multiplicadores de esfuerzo (**ME**) como indica la formula a continuación:

$$\mathbf{StoryPointAjust} = \mathbf{StoryPoint}^E \prod_{i=1}^n EM_i$$

9. Cantidad de Iteraciones para el proyecto (**I**)

Se determina dividiendo el total de puntos de historias del proyecto y la cantidad de puntos de historias ajustada (**StoryPointAjust**). Este valor indica la cantidad de Sprint que se necesita para terminar el proyecto.

$$I = \left\{ \frac{T\mathbf{Story Point}}{\mathbf{StoryPointAjust}} \right.$$

10. Costo promedio diario del equipo (**CostDay**)

Determina el valor promedio que percibe el equipo diariamente.

11. Estimación de costo del desarrollo de software (**CostDev**)

Determina el valor del desarrollo de software, que resulta de multiplicar el valor promedio diario del equipo (**CostDay**), el número de días definidos para el sprint (**DSprint**), el número de iteraciones resultante (**I**) y la cantidad de personas involucradas al proyecto (**N**). A continuación se muestra la fórmula resultante:

$$\mathbf{Cost} = ((\mathbf{CostDay} * \mathbf{DSprint}) * \mathbf{I}) * \mathbf{N}$$

Cabe mencionar que como indica la metodología ágil, esta estimación es realizada por el propio equipo de desarrollo de software, para ello tener en cuenta que

es muy importante que el equipo indique su capacidad indicando los puntos de historias que puede resolver diariamente de forma verdadera.

Otra opción sería que a la hora de estimar no se cuente con este dato, el número de personas en el equipo. Una manera de estimar sería indicar una velocidad inicial deseada y luego calcular a partir de ahí el esfuerzo necesario dependiendo de la capacidad de cada desarrollador de completar lo estimado.

Se puede obtener una tabla en donde se indique el nivel del desarrollador (sénior, pleno o junior) y la complejidad satisfactoria que cada uno pueda resolver diariamente. A partir de ahí se puede obtener el esfuerzo de acuerdo a los puntos de historias que sean necesarios por día para concluir lo estimado con base a necesidades.

4.2.1. Etapa 6 - Evaluación de la Propuesta

Para evaluar la propuesta se procede a definir ciertas características que deben contemplar una metodología de estimación de costos clasificada de manera esencial a medir el desarrollo de software.

La calidad de las medidas debería facilitar el desarrollo de modelos que sean capaces de predecir el comportamiento de determinados parámetros que afectan al desarrollo de software.

Una metodología de medida ideal debería ser:

- ✓ Objetiva.
- ✓ Sencilla, definible con precisión para que pueda ser evaluada.
- ✓ Fácilmente obtenible (a un coste razonable).
- ✓ Válida, la métrica debería medir exactamente lo que se quiere medir y no otra cosa.
- ✓ Robusta. Debería de ser relativamente insensible a cambios poco significativos en el proceso o en el producto.

Además, para una mejor utilización de estas medidas, a la hora de realizar estudios analíticos o análisis estadísticos deberían de tener unos valores que se ajusten a una cierta escala de medida.

A modo de analizar los cumplimientos de las características citadas anteriormente se define criterios para poder evaluar la metodología propuesta. Estos criterios son: (muy satisfactorio, satisfactorio, normal, no contempla).

En la Tabla 15 se puede visualizar los valores que contempla la nueva metodología propuesta.

Tabla 15 Características evaluadas (fuente propia).

Características	Valor
Objetiva	Muy Satisfactorio
Sencilla	Satisfactorio
Fácilmente obtenible	Normal
Válida	Muy Satisfactoria
Robusta	Normal

Además de las características antes mencionadas, la metodología debe reunir los siguientes requisitos para considerarse un buen método de estimación.

Un método de estimación tendrá éxito si:

- ✓ La estimación inicial está dentro del 30% de desviación del coste final real.
- ✓ El método permite el refinamiento de la estimación durante su tiempo de desarrollo.
- ✓ El método es fácil de usar por el estimador. Esto permite una rápida re-estimación cuando sea necesario; por ejemplo, para evaluar distintas alternativas.
- ✓ Las reglas de la estimación son entendidas por todas las personas a las que afectan los resultados de la misma. Los directivos se sienten más seguros cuando el proceso de estimación es fácilmente comprensible.
- ✓ El método es soportado por herramientas y está documentado. La disponibilidad de herramientas aumenta la eficacia de cualquier método. Esto es debido, principalmente, a que los resultados pueden ser obtenidos más rápidamente y de una forma estándar.

Para demostrar el cumplimiento de estos requisitos, en el siguiente capítulo se desarrolla un caso práctico real como toma de ejemplo, con respecto al último requisito, este trabajo aporta unas plantillas útiles como herramienta para la estimación de la nueva propuesta planteada.

4.3. Discusiones y Comentarios

Los multiplicadores de esfuerzo no sólo constituyen una parte importante en el cálculo del esfuerzo, sino que además la puntuación que se le otorgue a cada multiplicador ayudará en la calidad de la estimación y con esto cumple otro objetivo importante de la tesis.

La identificación del contexto es muy importante, tener en cuenta que esta estimación se emplea donde el equipo trabaje con metodologías ágiles, donde las características como ubicación, reuniones y definición de tareas se cumplen y el desarrollo de software siga un curso normal por el apoyo con base a esta metodología.

Los parámetros elegidos para la nueva propuesta fueron en base a investigación de los estándares más utilizados en metodologías ágiles y COCOMO II.

Se logró definir un esquema simple y beneficioso para la estimación de desarrollo de software basado en metodologías ágiles, tema que aún no está siendo difundido en la actualidad y esta tesis lo ofrece al mercado actual.

Capítulo 5

Pruebas Experimentales

5.1. Introducción

En este capítulo se utilizó un caso real, ocurrido en la Fundación Parque Tecnológico de Itaipú a la cual se lo denomina “parque” dentro del contexto, el caso del parque es muy importante mencionar ya que maneja el área de desarrollo de software, una de las más renombradas del país, caracterizada bajo ese concepto por seleccionar bajo concursos los mejores profesionales en cada área respectiva.

Esta fundación tiene como principal inconveniente el cambio constante de directivos en la empresa y esto conlleva a cambios de requisitos constantes en la administración del parque.

El equipo de desarrollo inició en Julio del 2010 al levantamiento de requisitos para proporcionar un sistema que informatices toda la gestión del parque. Comenzó con un equipo de tres profesionales, y luego conforme se iban necesitando se contrató a más profesionales, con un límite de quince personas en el equipo.

Dicha empresa hoy día cuenta con el software de gestión pero el desarrollo de dicho sistema nunca termina y se vuelve una actividad de constantes cambios. De aquí la necesidad del parque, saber si realmente es como indica el departamento de desarrollo, el tiempo que se ha demorado y de contraparte de ser justo, el monto total que se ha gastado por el sistema a lo largo de estos años.

Como sucede con todas las empresas, existe una necesidad de estimar el valor del producto, de ahí surge la propuesta de una metodología de estimación de costos en el desarrollo del software utilizando metodologías ágiles.

Viendo la primera necesidad, la de saber cuánto es el valor del software de gestión actual, se procedió a estimar el producto con puntos de función y líneas de código que ofrece COCOMO II.

Luego se deberá estimar cuánto costará una reingeniería del software con la utilización de la nueva propuesta metodológica utilizando metodologías ágiles de desarrollo.

5.2. Casa Práctico 1

El primer caso práctico es evaluar el software existente con la metodología normal de COCOMO II, comparar los resultados obtenidos a base de puntos de función y a base de Líneas de Código, esto sirve como aporte a la empresa para conocer el valor del software SGPTI (Sistema de Gestión Parque Tecnológico de Itaipú), como se le denomina al producto, según el modelo COCOMO II.

Para conocer acerca de la realidad de cómo se gestionó el desarrollo del software SGPTI, antes de comenzar con la estimación se elaboró un cuestionario con el fin de tener en claro las técnicas que fueron utilizadas por el parque y así completar los factores de escala y multiplicadores de esfuerzo antes de la estimación de software.

En la entrevista se respondieron a las preguntas encontradas en el anexo 1.

Las respuestas encontradas no fueron nada favorables para la valoración del producto. El proceso de la estimación se encuentra desde el anexo 2 hasta 9.

5.3. Caso Práctico 2

El segundo caso práctico ha sido probar la nueva metodología propuesta en esta tesis, estimando la reingeniería de un módulo del software del parque utilizando la nueva metodología y así comprobar si contempla los requisitos citados en la Etapa 6 de la propuesta.

El proceso de la estimación se encuentra desde el anexo 10 hasta 15.

5.4. Resultados Obtenidos

5.4.1. Caso Práctico 1

El tamaño del software SGPTI realizado por el modelo USC COCOMO II.2000.4 ha sido de 252.598 LOC y la productividad resultante en meses persona ha sido 463.5 PM que básicamente constituye un promedio de 3 años y medio de proyecto con una cantidad promedio de diez personas trabajando en el equipo. Se puede deducir que el proyecto del primer caso práctico es PRODUCTIVO bajo los cálculos estimados de COCOMO II; a continuación se listan los posibles motivos por los cuales se generaron dichos resultados:

- ✓ El nivel de organización del Jefe de Proyecto, este debía atender solicitudes de mantenimiento de sistemas, por lo cual se veía obligado a cambiar las prioridades

(baja, urgente, inmediato) de las tareas de los desarrolladores. Esto ocasionaba trabajos pendientes y pérdida de consistencia de la información recabada.

- ✓ Falta de coordinación al momento de tomar los datos; es decir, que el desarrollador registra la información después de un tiempo prolongado de haber terminado una o más tareas, por lo que se presume que los tiempos no son muy precisos.

5.4.2. Caso Práctico 2

En el segundo caso práctico se observa que el proyecto tendrá un costo de 47.072.032 Gs. para un equipo de seis personas para culminar con dos iteraciones de Sprint que duran 20 días cada una. Se debe considerar que hubo un análisis previo de lo que desea el cliente y luego de la formulación de la plantilla 3, se procedió al cálculo de la estimación.

Se pueden manipular los resultados de acuerdo a los factores de escala y multiplicadores de esfuerzo como se observa en la Tabla 16.

Tabla 16 Tabla de valores de factores y multiplicadores (fuente propia).

		Costo	Iteraciones
TEAM	ALTA	37.454.374	1.56
	NOM	40.428.713	1.68
	BAJA	47.072.032	1.82
MARURITY	ALTA	41.741.521	1.73
	NOM	37.454.374	1.56
	BAJA	33.607.548	1.40
PLEX	ALTA	31.269.248	1.42
	NOM	37.454.374	1.56
	BAJA	40.098.213	1.67

Como se observa en la Tabla 16 el factor de escala TEAM, la que se refiere a la cohesión del Equipo, modificando este valor para NOMINAL el costo varía como se visualiza en la Ilustración 11.

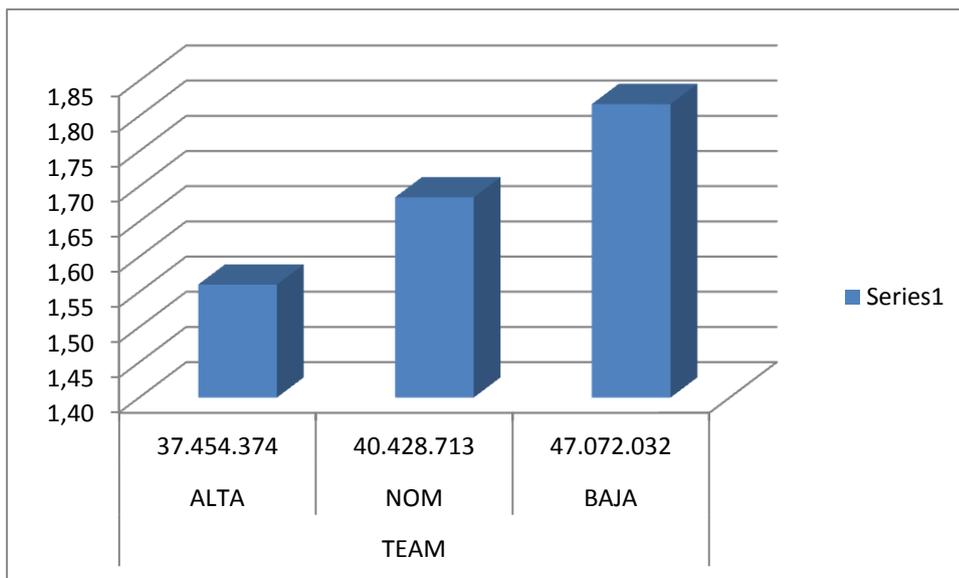


Ilustración 11 Factor TEAM (fuente propia).

Otro factor importante es el MATURITY, podemos ver que si cambiamos este valor a ALTA el costo sube a 41.741.521 Gs (Ilustración 12).

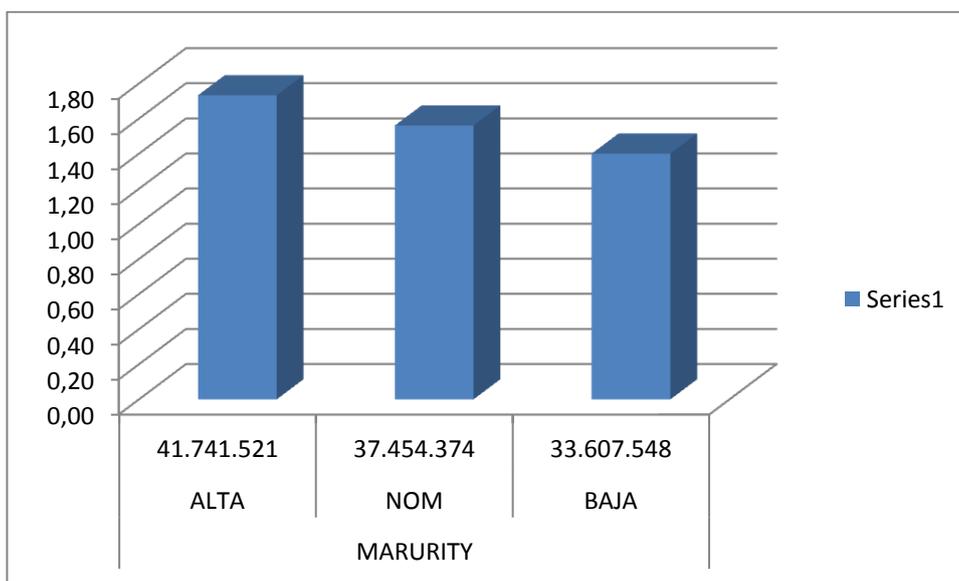


Ilustración 12 Factor MATURITY (fuente propia).

Entre los multiplicadores de esfuerzo se analiza el PLEX, que se refiere a experiencia en la plataforma, si modificamos este valor a BAJA podemos observar un aumento en el costo a 40.098.213 Gs (Ilustración 13).

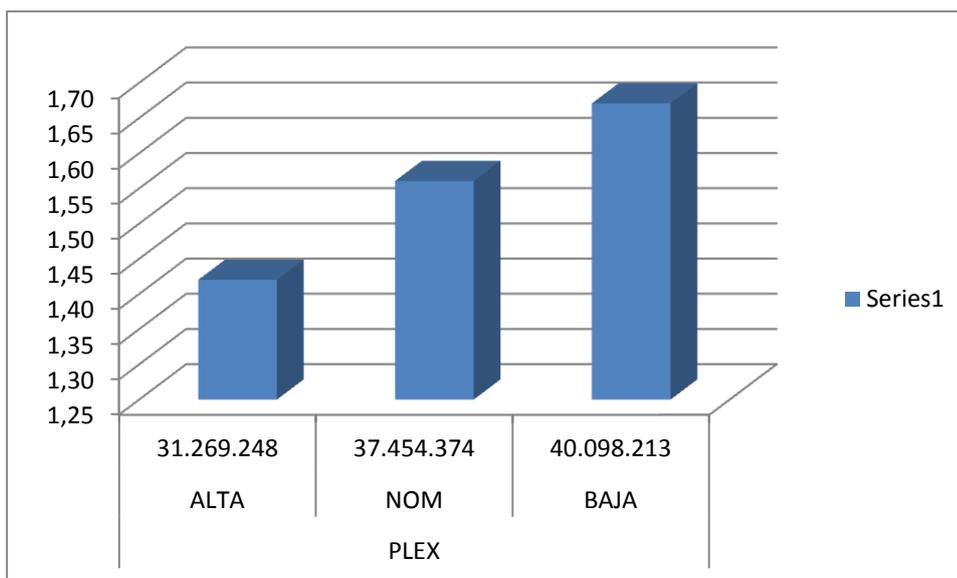


Ilustración 13 Multiplicador de Esfuerzo PLEX (fuente propia).

Como se observa en los análisis todos estos factores y multiplicadores cumplen un papel fundamental a la hora de estimar el costo.

5.5. Lecciones Aprendidas

- ✓ Cuanto más definidos estén los procesos del sistema a implementarse, la nueva metodología obtendrá valores más reales.
- ✓ Si se considera el documento de especificación de requisitos como base para la creación del modelo conceptual, puede dar motivo a diversas interpretaciones entre los planificadores encargados del desarrollo de la estimación.
- ✓ Cuanto más motivado se mantenga al equipo de desarrollo los resultados son positivos.
- ✓ Trabajar en equipo en la estimación es un requisito fundamental, dadas sus diferentes especializaciones, experiencias y puntos de vista la precisión de la estimación aumenta.
- ✓ El equipo trabaja de manera eficiente y con más calidad cuando ellos mismos se comprometen a entregar los resultados en los momentos determinados. Cada persona estima el esfuerzo y la optimización de su tarea.

- ✓ El equipo trabaja más motivado y satisfechos cuando puede usar su creatividad para resolver problemas y cuando pueden decidir su trabajo en la organización.

5.6. Ventajas Obtenidas con la Nueva Metodología

Acerca del proyecto de desarrollo de software bien planificado:

- ✓ El cliente maneja en etapas tempranas sus objetivos y los costos que estos implican gracias a la lista priorizadas que fue definida.
- ✓ Valoriza y lo califica a cada personal por los resultados obtenidos en la estimación por los resultados en cada iteración estimada.
- ✓ El cliente conoce la labor del personal y puede proveer información acerca de lo que el equipo de desarrollo realice.
- ✓ Un proyecto bien estimado es un proyecto bien planificado y esto implica un orden y entendimiento entre las partes.
- ✓ Gracias a las iteraciones planificadas, sus costos es posible mitigar riesgos de manera anticipada ahuyentando errores técnicos.

Acerca de la gestión regular de las expectativas del cliente:

- ✓ El cliente comprueba de manera regular si van cumpliendo sus expectativas, ya desde el inicio del proyecto conoce el resultado y caso haya impedimentos el lapso de solución y comunicación al cliente es corta.
- ✓ El cliente conoce lo que va recibiendo y verifica si el plan se va cumpliendo, conoce sus costos y con esto se ahorra esfuerzo y tiempo al evitar hipótesis.

Acerca de los resultados anticipados:

- ✓ El cliente puede empezar a utilizar los resultados más importantes del proyecto antes de que esté finalizado por completo.
- ✓ Siguiendo la ley de Pareto (el 20% del esfuerzo proporciona el 80% del valor), el cliente puede empezar antes a recuperar su inversión con las automatización proveídos por el sistema informático.

Capítulo 6

Conclusiones y Recomendaciones y Trabajos Futuros

6.1. Conclusiones

Por los resultados obtenidos de la estimación con la nueva propuesta metodológica a proyectos reales de software, se puede confirmar la hipótesis planteada, esto es que, la utilización de metodologías ágiles en el desarrollo de software puede llegar a ser cada vez más preciso en cuanto a estimación de tiempo y costo. Aunque no puede generalizarse, esto está cada vez más estable de acuerdo a las prácticas actuales en el desarrollo de software.

La investigación de campo realizada en la Fundación Parque Tecnológico de Itaipú ayudó a definir la situación real y actual sobre cómo manejan la estimación en las empresas, se comprobó que no usan técnicas, ni metodología aplicable para el tipo de software a desarrollar, por lo general se basan en la experiencia del profesional del equipo de trabajo. Por esta razón la falencia en la estimación temprana produce desfases en la entrega del producto.

La especificación de requerimientos no es considerada como etapa dentro del desarrollo de de software, y se utiliza erradamente el modelo conceptual de datos como parte de la etapa de diseño.

Esta metodología propuesta apoya en la medición temprana y logra definir el esfuerzo, tiempo y costo necesarios para cumplir con las entregas del producto al finalizar cada ciclo del Sprint.

La metodología puede objetarse por estar relacionada a una metodología ágil particular de desarrollo y en consecuencia no podrán estandarizarse, sin embargo, casi todas utilizan casi los mismos parámetros y se adecuan a la realidad actual para obtener una estimación real.

Se requiere prototipar las historias de usuarios para conseguir una mejor definición y una mejor ejecución del plan.

Para realizar la estimación de complejidad de historias debe existir un conceso dentro de la organización para llegar a lo más exacto posible para cada historia de usuario, y así lograr que la medición se acerque lo máximo a la realidad.

La correcta redivisión de los factores de escala y los multiplicadores de esfuerzo varían notablemente en el resultado final, se requiere buena disposición y trabajo en equipo para consensuar cada valor correspondiente, y por consiguiente proveer calidad a la estimación de costos.

Con esta metodología se puede estimar desarrollos de software en menor tiempo, definiendo la arquitectura y trabajando en paralelo con los requisitos que se van sumando al proyecto.

Gracias a esta metodología el equipo podrá demostrar el trabajo diario y entregar el producto con calidad y una precisión de tiempo exitosa, a su vez la empresa en cuestión valorará el pago correcto al equipo por los resultados más reales en la estimación.

Se demostró la problemática que se vive en la actualidad en equipos de desarrollos de software por no contar con una metodología de estimación de costo.

6.2. Recomendaciones

Es necesario hacer énfasis en el levantamiento de los procesos definidos para cada área de la organización para evitar que al cambio de autoridades varíen los requisitos planteados.

Se recomienda mantener el equipo unido en ubicaciones estratégicas para que todos en conjuntos puedan ayudarse y comprometerse a cumplir lo estimado sin impedimentos graves.

Una buena estimación dependerá de la precisión de los parámetros utilizados, para ello es bueno mantener un esquema para realizar la evaluación.

Actualmente en Paraguay, es muy común la licitación para la implementación de software, esta metodología ayudará de forma rápida y segura para obtener el valor correcto del desarrollo de software.

Se tiene previsto publicar los resultados obtenidos y difundir la nueva propuesta metodológica para la estimación del costo del desarrollo de software en congresos y revistas nacionales e internacionales.

6.3. Trabajos Futuros

A los efectos de validar, de forma empírica, la nueva metodología que se presenta en este trabajo, es necesario obtener más confirmaciones experimentales de la efectividad de la propuesta realizada, tanto cualitativamente como cuantitativamente. Es decir, cotejar mediante un mayor trabajo de campo para corroborar así la eficacia y la eficiencia de la nueva metodología propuesta y enriquecerla con los ajustes necesarios si fuera conveniente.

Otra línea de investigación interesante, sería el análisis y la especificación formal de un modelado y prototipado para la posterior implementación de la nueva propuesta metodológica.

Bibliografía

- Agarwal, R. y. (2001). *Estimating software projects*. Software engineers Notes.
- Agile, W. i. (2001). *DSDM Consortium*. Retrieved from <http://www.dsdm.org/content/what-agile>
- Agile?, W. i. (2001). *DSDM Consortium*. Retrieved from <http://www.dsdm.org/content/what-agile>
- Ahmed E. Hassan, R. C. (2003). *The Chaos of Software Development*. In IEEE Computer Society.
- Alish., T. H. (2007). *Accuracy of Estimation Models with Discrete*. Proceedings of Software Measurement European Forum.
- Banerjee, G. (2001). *Use Case Point - An Estimation Approach*.
- Boehm B., C. A. (2000). *Software Cost Estimation with COCOMO II*. Prentice-Hall.
- Capers, J. (2004). *Conflict and Litigation Between Software Clients and Developers*. Burlington. MA: Software Productivity Research.
- Clemmons, R. K. (2006). *Project estimation with Use Case Points*. EEUU: Crosstalk: The Journal of Defense Software Engineering.
- Cockburn, A. (2007). *Agile Software Development The cooperative Game Second Edition*. Boston. USA.: Addison Wesley.
- Crystal Methodologies*. (2013). Retrieved from <http://crystalmethodologies.org/>
- Dekkers, C. (2003). *Measuring the logical or functional size of software projects and software application*. ISO Bulletin.
- Fenton, N. E. (2014). *Software Metrics: A Practical and Rigorous Approach" (3rd Edition)* . For publication de Taylor and Francis.
- Jones, C. (2008). *Applied Software Measurement* . McGraw Hill.
- Jones, C. (2007). *Estimating software cost*. Hill Edition.
- Kamaruddin, N. A. (2012). *A. Chaos issues on communication in Agile Global Software Development*. In IEEE Computer Society.
- Kidd, L. a. (1994). *Object Oriented Metrics*. Englewood: Prentice Hall.
- Leungh, Z. (2001). *Software cost estimation in Handbook of software engineering and knowledge engineering*. NJ: World Scientific Pub. Co, River Edge.
- Longstreet, C. I. (2004). *Function Points Analysis Training Course*. Retrieved from <http://www.softwaremetrics.com/Function%20Point%20Training%20Booklet%20New.pdf>

- Mendes Calo, K. E. (2010). *Evaluación de Metodologías Ágiles para Desarrollo de Software*. XII Workshop de Investigadores en Ciencias de la Computación.
- Metodología FDD. (2012). *Una exclusiva base de datos de ensayos para estudiantes* .
- Milicic, D. (2004). *Applying COCOMO II, A case study*. Master Thesis Software Engineering.
- Pressman R., R. (2005). *Ingeniería del Software. Un enfoque práctico. 6ta Edición*. México: McGrawHill Interamericana.
- Schwaber K., B. M. (2001). *Agile Software Development with SCRUM*. Prentice Hall.
- Standish Group. (n.d.). *Standish Group*. Retrieved from <http://blog.standishgroup.com/BigBangBoom.pdf>
- The Agile Manifesto. (2001). *The Agile Manifesto*. Retrieved from <http://agilemanifesto.org/authors.html>
- Wells, D. (2013). *Extreme Programming*. Retrieved from www.extremeprogramming.org, www.xprogramming.com,
- Yenduri, S.; Munagala, S.; Perkins, L.A. (2007). *Estimation Practices Efficiencies: A Case Study*. Information Technology.

Anexo

A. Cuestionario de entrevista.

¿Qué metodología(s) de gestión de desarrollo de software se utilizó para el proyecto SGPTI? (Espiral, Cascada, Iterativo, etc.).

Cascada

¿Qué metodología(s) de desarrollo de software se utilizó en el proyecto de software SGPTI dentro de la empresa? (Estructuradas, Orientadas a Objetos).

Orientada a Objetos

En base a dichas metodologías, ¿en qué fases dividió el desarrollo de sus proyectos (Planificación, Análisis, Diseño, etc.)?

¿Qué tipo de técnicas de modelado del sistema y su entorno utilizó (Diagrama de clase, diagrama de secuencia, Modelo Entidad-Relación, Diagrama de Flujo de Datos, etc.)?

¿Utilizó alguna de estas técnicas de modelado para obtener o mejorar la descripción de los requisitos?

Se utilizaron los Casos de Usos

En cuanto a las herramientas CASE, ¿utilizó alguna para facilitar el trabajo? ¿Qué herramientas (Visio, Rational Rose, Dome, MetaEdit, ArgoUML (open source), etc.)?

SQL Power Architect, RedMine, DotProject

¿Qué estándares utilizó la empresa para asegurar la calidad en el desarrollo de los proyectos de software? ¿Tiene alguna certificación de calidad en los proyectos software (CMMI, SPICE, ISO/IEC, ISO 9001:2000, etc.)?

No posee certificaciones, pero el Caso de Uso corresponde a estándares de CMMI.

¿Se cumplió con los plazos previstos en la estimación inicial del proyecto de software?

No.

¿Qué tipo de métricas utilizó para el desarrollo de los proyectos software (De Calidad, Productividad, Orientadas a la Persona, Orientadas al Tamaño, Orientadas a la Función)?

Ninguno

¿Qué tipo(s) de técnicas de estimación realizó para Planificar el desarrollo del proyecto de software (Ej. Datos Históricos, Técnicas Delphi, Líneas de Código y Puntos de Función, Puntos Objeto)?

A ojo.

¿Cómo se determinó el tiempo, costo y RRHH estimado necesario para el desarrollo del proyecto de software en la empresa?

En base a necesidades.

¿A la hora de estimar un proyecto de software, tomó como base alguno de los modelos conceptuales descritos anteriormente (Modelo Entidad-Relación, Diagrama de Clase, Diagrama de Secuencia, etc.)?

Sí. Con base en experiencias de módulos terminados.

¿En base a qué parámetros se distribuyó el tiempo para cada una de las fases?
¿Tiene alguna proporción? (Análisis, diseño, implementación, pruebas).

En base a experiencia del gerente.

¿Dedicó más tiempo del desarrollo del software a la funcionalidad o a la arquitectura?

Se dedicó más tiempo en el desarrollo del software.

B. Caso Práctico 1

1. Estimación por Puntos de Función del Módulo de Bienes Patrimoniales.

SGPTI - GESTIÓN DEL PARQUE TECNOLÓGICO DE ITAIPU

Identificación de los datos del Proyecto a estimar

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu

Propósito de la Estimación

Obtener el número de puntos de función de la aplicación.

Contenidos a estimar

Esta estimación abarca el Modulo de Bienes

- . Mantenimiento de Tipo de Bienes
- . Mantenimiento de Ubicación
- . Mantenimiento de Bien Patrimonial
- . Registrar Inventario Físico
- . Registrar Bien Patrimonial
- . Registrar Término de Responsabilidad
- . Registrar Asignación de Bienes
- . Registrar Recepción de Bienes

LISTADOS

Tipo de Bienes
Ubicación
Bien Patrimonial

INFORMES

Inventario Físico
Inventario Físico Registrados
Inventario Físico No Registrados
Devolución de Bien Patrimonial
Término de Responsabilidad
Detalles de Recepción de Bien Patrimonial
Asignación de Bien Patrimonial
Recepción de Bien Patrimonial

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
ARCHIVOS						
1	TipoBien	ALI	7	1	Baja	7
2	Ubicación	ALI	5	1	Baja	7
3	BienPatrimonial	ALI	29	12	Alta	15
4	BienPatrimonialEquipo	ALI	3	1	Baja	7
5	BienPatrimonialAgrupador	ALI	5	1	Baja	7
6	BienPatrimonialDetalle	ALI	8	1	Baja	7
7	InventarioFisico	ALI	15	4	Baja	7
8	InventarioFisicoDetalle	ALI	9	5	Baja	7
9	DevolucionBienPatrimonial	ALI	17	4	Baja	7
10	DevolucionBienPatrimonialDetalle	ALI	12	2	Baja	7
11	TerminoResponsabilidad	ALI	20	6	Alta	15
12	TerminoResponsabilidadDetalle	ALI	15	5	Baja	7
13	AsignacionBien	ALI	29	12	Alta	15
14	RecepcionBien	ALI	25	5	Media	10
15	RecepcionBienDetalle	ALI	15	6	Baja	10
TIPO DE BIENES						
16	TipoBien - Incluir	EE	9	1	Baja	3
17	TipoBien - Editar	EE	9	1	Baja	3
18	TipoBien - Eliminar	EE	9	1	Baja	3
19	TipoBien - Consulta	CE	9	1	Baja	3
UBICACION						
20	Ubicacion - Incluir	EE	7	1	Baja	3
21	Ubicacion - Editar	EE	7	1	Baja	3
22	Ubicacion - Eliminar	EE	7	1	Baja	3
23	Ubicacion - Consulta	CE	7	1	Baja	3
BIEN PATRIMONIAL						
24	BienPatrimonial - Incluir	EE	31	12	Alta	6
25	BienPatrimonial - Editar	EE	31	12	Alta	6
26	BienPatrimonial - Eliminar	EE	31	12	Alta	6
27	BienPatrimonial - Consulta	EE				
		CE				
INVENTARIO FISICO						
28	InventarioFisico - Incluir	EE	17	4	Alta	6
29	InventarioFisico - Editar	EE	17	4	Alta	6
30	InventarioFisico - Anular	EE	17	4	Alta	6
31	InventarioFisico - Consulta	CE	17	4	Alta	6
32	InventarioFisico - Ver	CE	17	4	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
DEVOLUCION BIEN PATRIMONIAL						
33	DevolucionBienPatrimonial - Incluir	EE	19	4	Alta	6
34	DevolucionBienPatrimonial - Procesar	EE	19	4	Alta	6
35	DevolucionBienPatrimonial - Anular	EE	19	4	Alta	6
36	DevolucionBienPatrimonial - Consulta	CE	19	4	Alta	6
37	DevolucionBienPatrimonial - Ver	CE	19	4	Alta	6
TERMINOS DE RESPONSABILIDAD						
38	TerminoResponsabilidad - Incluir	EE	22	6	Alta	6
39	TerminoResponsabilidad - Anular	EE	22	6	Alta	6
40	TerminoResponsabilidad - Consulta	CE	22	6	Alta	6
41	TerminoResponsabilidad - Ver	CE	22	6	Alta	6
ASIGNACION DE BIENES						
42	AsignacionBien - Incluir	EE	31	12	Alta	6
43	AsignacionBien - Anular	EE	31	12	Alta	6
44	AsignacionBien - Ver	CE	31	12	Alta	6
45	AsignacionBien - Consulta	CE	31	12	Alta	6
RECEPCION DE BIENES						
46	RecepcionBien - Incluir	EE	27	5	Alta	6
47	RecepcionBien - Anular	EE	27	5	Alta	6
48	RecepcionBien - Editar	EE	27	5	Alta	6
49	RecepcionBien - Consulta	CE	27	5	Alta	6
LISTADOS						
50	Ubicacion	CE	6	1	Baja	3
51	TipoBien	CE	6	2	Medio	4
52	BienPatrimonial	CE	14	8	Alta	6
53	DevolucionBienPatrimonial	CE	6	2	Medio	4
54	TerminoResponsabilidad	CE	6	1	Baja	3
INFORMES						
55	BienPatrimonial	CE	13	8	Alta	6
56	InventarioFisico	CE	7	1	Baixa	3
57	InventarioFisicoRegistrado	CE	11	6	Alta	6
58	InventarioFisicoNoRegistrado	CE	9	4	Alta	6
59	DevolucionBienPatrimonial	CE	10	5	Alta	6
60	TerminoResponsabilidad	CE	12	8	Alta	6
61	RecepcionBien	CE	11	6	Alta	6
62	RecepcionBienDetalle	CE	9	2	Média	4
63	AsignacionBien	CE	7	4	Alta	6

Puntos de Función no Ajustados						
Tipo de Función	Complejidad Funcional			Total por Complejidades	Total por Tipo de Función	
EE	6	Baja	x 3	18	114	30,16%
	0	Media	x 4	0		
	16	Alta	x 6	96		
SE	0	Baja	x 4	0	0	0,00%
	0	Media	x 5	0		
	0	Alta	x 7	0		
CE	5	Baja	x 3	15	129	34,13%
	3	Media	x 4	12		
	17	Alta	x 6	102		
ALI	10	Baja	x 7	70	135	35,71%
	2	Media	x 10	20		
	3	Alta	x 15	45		
AIE	0	Baja	x 5	0	0	0,00%
	0	Media	x 7	0		
	0	Alta	x 10	0		
Total de Puntos de Función no Ajustados					378	

Ilustración 14 PF - Variable de Conteo del Módulo de Bienes (fuente propia).

2. Estimación por Puntos de Función del Módulo de Compras.

SGPTI - GESTIÓN DEL PARQUE TECNOLÓGICO DE ITAIPU

Identificación de los datos del Proyecto a estimar

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu

Propósito de la Estimación

Obtener el número de puntos de función de la aplicación producto.

Contenidos a estimar

Esta estimación abarcará el Módulo de Compra

- . Mantenimiento de Admin Proveedores
- . Mantenimiento de Admin Items
- . Mantenimiento de Tipo Items
- . Mantenimiento de Ciudad
- . Mantenimiento de País
- . Mantenimiento de Tipo Documento
- . Mantenimiento de Barrio
- . Mantenimiento de Moneda
- . Mantenimiento de Sede
- . Mantenimiento de Persona
- . Mantenimiento de Tipo Adquisición
- . Mantenimiento de Modalidad Compra
- . Mantenimiento de Forma Pago
- . Mantenimiento Contactos Proveedor
- . Mantenimiento Marcas Proveedor
- . Registrar Pedido de Cotización
- . Registrar Orden de Compra
- . Registrar Orden de Servicio
- . Registrar Recepción

LISTADOS

Ciudad

País

Tipo Documento

Barrio

Admin Items

Moneda

Contenidos a estimar

Persona
Forma Pago
Proveedor
Empresa
Tipo Adquisicion
Empresa Empleado
Contactos de Proveedor
Marca

INFORMES

Psm
Psm Detalle
Pedido Cotizacion
Pedido Cotizacion Proveedor Detalle
Pedido Cotizacion Detalle
Recepcion
Recepcion Detalle
Orden de Servicio
Orden de Servicio Detalle
Centro de Costo
Orden Compra
Orden Compra Detalle
Pedido Cotizacion Detalle
Recepcion
Recepcion Detalle
Orden de Servicio
Orden de Servicio Detalle
Centro de Costo
Orden Compra
Orden Compra Detalle

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
1	ARCHIVOS					
2	Ciudad	ALI	8	3	Baja	7
3	Pais	ALI	8	1	Baja	7
4	Tipo Documento	ALI	7	1	Baja	7
5	Barrio	ALI	4	1	Baja	7
6	Admin Items	ALI	5	1	Baja	7
7	Tipo Item	ALI	5	1	Baja	7
8	Moneda	ALI	5	1	Baja	7
9	Sede	ALI	7	1	Baja	7
10	Persona	ALI	15	3	Baja	7
11	Forma Pago	ALI	7	1	Baja	7
12	Proveedor	ALI	11	1	Baja	7
13	Psm	ALI	21	8	Alta	15
14	Psm Detalle	ALI	11	3	Baja	7
15	Pedido Cotizacion	ALI	15	4	Baja	7
16	Pedido Cotizacion Proveedor Detalle	ALI	12	2	Baja	7
17	Pedido Cotizacion Detalle	ALI	16	3	Baja	7
18	Recepcion	ALI	24	4	Media	10
19	Recepcion Detalle	ALI	18	4	Baja	7
20	Orden de Servicio	ALI	34	11	Alta	15
21	Orden de Servicio Detalle	ALI	16	2	Baja	7
22	Centro de Costo	ALI	11	2	Baja	7
23	Empresa	ALI	17	1	Baja	7
24	Tipo Adquisicion	ALI	4	1	Baja	7
25	Orden Compra	ALI	36	11	Alta	15
26	Orden Compra Detalle	ALI	20	2	Media	10
27	Empresa Empleado	ALI	7	7	Media	10
28	Modalidad Compra	ALI	34	11	Alta	15
29	Contactos de Proveedor	ALI	5	2	Baja	7
30	Proveedor Marca	ALI	2	2	Baja	7
31	Marca	ALI	3	1	Baja	7
	ADMIN PROVEEDORES					
32	admin. Proveedores - Incluir	EE	13	1	Baja	3
33	admin. Proveedores - Editar	EE	13	1	Baja	3
34	admin. Proveedores - Eliminar	EE	13	1	Baja	3
35	admin. Proveedores - Consulta	CE	13	1	Baja	3
	CONTACTOS DE PROVEEDORES					
36	Contactos de proveedores - Incluir	EE	7	2	Medio	4
37	Contactos de proveedores - Eliminar	EE	7	2	Medio	4
38	Contactos de proveedores - Consulta	EE	7	2	Medio	4

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
ORDEN DE SERVICIO						
74	Orden de Servicio - Generar	EE	52	13	Alta	6
75	Orden de Servicio - Consulta	CE	52	13	Alta	6
76	Orden de Servicio - Autorizar	EE	52	13	Alta	6
77	Orden de Servicio - Facturas	EE	52	13	Alta	6
78	Orden de Servicio - Reabrir	EE	52	13	Alta	6
79	Orden de Servicio - Editar	EE	52	13	Alta	6
80	Orden de Servicio - Eliminar	EE	52	13	Alta	6
81	Orden de Servicio - Listar	EE	52	13	Alta	6
RECEPCION						
82	Recepcion - Generar	EE	44	8	Alta	6
83	Recepcion - Reabrir	CE	44	8	Alta	6
84	Recepcion - Editar	EE	44	8	Alta	6
85	Recepcion - Eliminar	EE	44	8	Alta	6
86	Recepcion - Listar	EE	44	8	Alta	6
87	Recepcion - Consulta	CE	44	8	Alta	6
CIUDAD						
88	Ciudad - Incluir	EE	10	3	Alta	6
89	Ciudad - Editar	EE	10	3	Alta	6
90	Ciudad - Eliminar	EE	10	3	Alta	6
91	Ciudad - Consulta	CE	10	3	Média	4
PAIS						
92	Pais - Incluir	EE	10	1	Baixa	3
93	Pais - Editar	EE	10	1	Baixa	3
94	Pais - Eliminar	EE	10	1	Baixa	3
95	Pais - Consulta	CE	10	1	Baixa	3
TIPO DOCUMENTO						
96	Tipo Documento - Incluir	EE	9	1	Baixa	3
97	Tipo Documento - Editar	EE	9	1	Baixa	3
98	Tipo Documento - Eliminar	EE	9	1	Baixa	3
99	Tipo Documento - Consulta	CE	9	1	Baixa	3
BARRIO						
100	Barrio - Incluir	EE	6	1	Baixa	3
101	Barrio - Editar	EE	6	1	Baixa	3
102	Barrio - Eliminar	EE	6	1	Baixa	3
103	Barrio - Consulta	CE	6	1	Baixa	3

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
MONEDA						
104	Moneda - Incluir	EE	7	1	Baixa	3
105	Moneda - Editar	EE	7	1	Baixa	3
106	Moneda - Eliminar	EE	7	1	Baixa	3
107	Moneda - Consulta	CE	7	1	Baixa	3
SEDE						
108	Sede -Incluir	EE	9	1	Baixa	3
109	Sede - Editar	EE	9	1	Baixa	3
110	Sede - Eliminar	EE	9	1	Baixa	3
111	Sede - Consulta	CE	9	1	Baixa	3
PERSONA						
112	Persona - Incluir	EE	17	3	Alta	6
113	Persona - Editar	EE	17	3	Alta	6
114	Persona - Eliminar	EE	17	3	Alta	6
115	Persona - Consulta	CE	17	3	Média	4
Tipo Adquisicion						
116	Tipo de Adquisicion - Incluir	EE	6	1	Baixa	3
117	Tipo de Adquisicion - Editar	EE	6	1	Baixa	3
118	Tipo de Adquisicion - Eliminar	EE	6	1	Baixa	3
119	Tipo de Adquisicion - Consulta	CE	6	1	Baixa	3
MODALIDAD COMPRA						
120	Modalidad Compra - Incluir	EE	36	11	Alta	6
121	Modalidad Compra - Editar	EE	36	11	Alta	6
122	Modalidad Compra - Eliminar	EE	36	11	Alta	6
123	Modalidad Compra - Consulta	CE	36	11	Alta	6
FORMA PAGO						
124	Forma de Pago - Incluir	EE	9	1	Baixa	3
125	Forma de Pago - Editar	EE	9	1	Baixa	3
126	Forma de Pago - Eliminar	EE	9	1	Baixa	3
127	Forma de Pago - Consulta	CE	9	1	Baixa	3
LISTADOS						
128	Ciudad	CE	7	2	Média	4
129	Pais	CE	6	1	Baixa	3
130	Tipo Documento	CE	5	1	Baixa	3
131	Barrio	CE	4	1	Baixa	3
132	Admin Items	CE	7	1	Baixa	3
133	Moneda	CE	7	1	Baixa	3
134	Persona	CE	5	1	Baixa	3
135	Forma Pago	CE	5	1	Baixa	3
136	Tipo Adquisicion	CE	4	1	Baixa	3

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu							
#	Proceso elemental o Grupos de datos	Tipo	Calculos				
			TD	AR/TR	Complex.	PF	
137	Proveedor	CE	5	1	Baixa	3	
138	Empresa	CE	7	1	Baixa	3	
139	Empresa Empleado	CE	9	7	Alta	6	
140	Contactos de Proveedor	CE	5	1	Baixa	3	
141	Marca	CE	2	1	Baixa	3	
INFORMES							
142	Psm	CE	10	5	Alta	6	
143	Psm Detalle	CE	11	3	Média	4	
144	Pedido Cotizacion	CE	8	5	Alta	6	
145	Pedido Cotizacion Proveedor Detalle	SE	6	2	Média	5	
146	Pedido Cotizacion Detalle	CE	11	2	Média	4	
147	Recepcion	CE	9	4	Alta	6	
148	Recepcion Detalle	CE	9	3	Média	4	
149	Orden de Servicio	CE	20	13	Alta	6	
150	Orden de Servicio Detalle	CE	14	1	Baixa	3	
151	Centro de Costo	CE	9	1	Baixa	3	
152	Orden Compra	CE	20	11	Alta	6	
153	Orden Compra Detalle	CE	14	1	Baixa	3	

Puntos de Función no Ajustados						
Tipo de Función	Complejidad Funcional			Total por Complejidades	Total por Tipo de Función	
EE	27	Baja	x 3	81	363	45,32%
	6	Media	x 4	24		
	43	Alta	x 6	258		
SE	0	Baja	x 4	0	5	0,62%
	1	Media	x 5	5		
	0	Alta	x 7	0		
CE	24	Baja	x 3	72	182	22,72%
	8	Media	x 4	32		
	13	Alta	x 6	78		
ALI	23	Baja	x 7	161	251	31,34%
	3	Media	x 10	30		
	4	Alta	x 15	60		
AIE	0	Baja	x 5	0	0	0,00%
	0	Media	x 7	0		
	0	Alta	x 10	0		
Total de Puntos de Función no Ajustados					801	

Ilustración 15 PF - Variable de Conteo del Módulo de Compras (fuente propia).

3. Estimación por Puntos de Función del Módulo de Contabilidad.

SGPTI - GESTIÓN DEL PARQUE TECNOLÓGICO DE ITAIPU

Identificación de los datos del Proyecto a estimar

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu

Propósito de la Estimación

Obtener el número de puntos de función de la aplicación producto.

Contenidos a estimar

Esta estimación abarcara el Modulo de Contabilidad

- . Mantenimiento de Asiento Manual
 - . Mantenimiento de Plan Cuenta
 - . Mantenimiento de Comprobante IVA
 - . Mantenimiento de Corizacion Moneda
 - . Mantenimiento de Composicion Cuenta
 - . Mantenimiento de Convenio
 - . Mantenimiento de Centro Costo
 - . Mantenimiento de Centro Costo Convenio
 - . Mantenimiento de Tipo Asiento
 - . Mantenimiento de Monedas
 - . Mantenimiento de Tipo Comprobante
 - . Mantenimiento de Cuenta Contable para Cuenta Bancaria
 - . Mantenimiento de Cuenta Contable para Centro Costo
- Registrar Cobro Cheque
Registrar Cerrar Periodo
Registrar Cierre Anual

LISTADOS

- Libro Diario
- Libro Mayor
- Balance General
- Libro Compra
- Libro Venta
- Libro Mayor Rubricado
- Clasificador de Conceptos de Viaticos
- SET Balance General
- SET Estado de Resultados
- SET Estado de Flujo de Efectivo
- Consulta de Asientos Contables

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
ARCHIVOS						
1	ContAsiento	ALI	18	2	Baja	7
2	ContCentroCostoCtaCtble	ALI	8	2	Baja	7
3	ContClasificacion	ALI	9	3	Baja	7
4	ContComposicionCuenta	ALI	12	2	Baja	7
5	ContComprobanteIva	ALI	25	4	Media	10
6	ContConfiguracionAsientoAutomata	ALI	14	6	Media	10
7	ContCotizacion	ALI	10	2	Baja	7
8	ContCuentaBancariaCtaContable	ALI	8	2	Baja	7
9	ContAsientoDetalle	ALI	23	5	Media	10
10	ContClasificacionItemDetalle	ALI	8	2	Baja	7
11	ContClasificacionViaticoDetalle	ALI	8	2	Baja	7
12	ContComposicionCuentaDetalle	ALI	7	2	Baja	7
13	ContHistoricoCierre	ALI	9	1	Baja	7
14	ContHistoricoReaperturas	ALI	6	1	Baja	7
15	ContMoneda	ALI	9	1	Baja	7
16	ContPlanCuenta	ALI	17	4	Baja	7
17	ContSecuencia	ALI	5	1	Baja	7
18	ContTipoAsiento	ALI	7	1	Baja	7
19	ContTipoComprobante	ALI	7	1	Baja	7
ASIENTO MANUAL						
20	ContAsiento - Incluir	EE	20	2	Alta	6
21	ContAsiento - Editar	EE	20	2	Alta	6
22	ContAsiento - Ver	EE	20	2	Alta	6
23	ContAsiento - GenerarAsiento	EE	20	2	Alta	6
24	ContAsiento - Consulta	CE	20	2	Alta	6
PLAN DE CUENTAS						
25	ContPlanCuenta - Incluir	EE	19	4	Alta	6
26	ContPlanCuenta - Ver	EE	19	4	Alta	6
27	ContPlanCuenta - Editar	EE	19	4	Alta	6
28	ContPlanCuenta - Eliminar	EE	19	4	Alta	6
29	ContPlanCuenta - Consulta	CE	19	4	Alta	6
COMPROBANTE IVA						
30	ContComprobanteIVA - Incluir	EE	27	4	Alta	6
31	ContComprobanteIVA - Ver	EE	27	4	Alta	6
32	ContComprobanteIVA - Editar	EE	27	4	Alta	6
33	ContComprobanteIVA - Editar	EE	27	4	Alta	6
34	ContComprobanteIVA - Eliminar	EE	27	4	Alta	6
35	ContComprobanteIVA - Consulta	CE	27	4	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
COTIZACION DE MONEDAS						
36	CotMoneda - Incluir	EE	12	2	Medio	4
37	CotMoneda - Editar	EE	12	2	Medio	4
38	CotMoneda - Ver	EE	12	2	Medio	4
39	CotMoneda - Consulta	CE	12	2	Medio	4
COMPOSICION CUENTA						
40	ContComposicionCuenta - Incluir	EE	14	2	Medio	4
41	ContComposicionCuenta - Editar	EE	14	2	Medio	4
42	ContComposicionCuenta - Ver	EE	14	2	Medio	4
43	ContComposicionCuenta - Elimina	EE	14	2	Medio	4
44	ContComposicionCuenta - Consul	CE	14	2	Medio	4
COBRO CHEQUES						
45	TesrChequeEmitido - ConsultaCor	CE	29	6	Alta	6
CERRAR PERIODO						
46	ContHistoricoCierre - ProcesoCerr	EE	11	1	Baja	3
PROCESAR CIERRE ANUAL						
47	ContAsientoDetalle - ProcesoCierr	EE	34	6	Alta	6
CONVENIO						
48	Convenio - Incluir	EE	15	2	Medio	4
49	Convenio - Editar	EE	15	2	Medio	4
50	Convenio - Ver	EE	15	2	Medio	4
51	Convenio - Eliminar	EE	15	2	Média	4
52	Convenio - Consulta	CE	15	2	Média	4
CENTRO COSTO						
53	CentroCosto - Incluir	EE	13	2	Média	4
54	CentroCosto - Editar	EE	13	2	Média	4
55	CentroCosto - Ver	EE	13	2	Média	4
56	CentroCosto - Eliminar	EE	13	2	Média	4
57	CentroCosto - Consultar	EE	13	2	Média	4
CENTRO COSTO DE CONVENIO						
58	CostoConvenio - Incluir	EE	10	2	Média	4
59	CostoConvenio - Editar	EE	10	2	Média	4
60	CostoConvenio - Ver	EE	10	2	Média	4
61	CostoConvenio - Eliminar	EE	10	2	Média	4
62	CostoConvenio - Consulta	CE	10	2	Média	4

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
TIPO ASIENTO						
63	ContTipoAsiento - Incluir	EE	9	1	Baixa	3
64	ContTipoAsiento - Editar	EE	9	1	Baixa	3
65	ContTipoAsiento - Eliminar	EE	9	1	Baixa	3
66	ContTipoAsiento - Ver	EE	9	1	Baixa	3
67	ContTipoAsiento - Consulta	CE	9	1	Baixa	3
Moneda						
68	ContMoneda - Incluir	EE	11	1	Baixa	3
69	ContMoneda - Editar	EE	11	1	Baixa	3
70	ContMoneda - Ver	EE	11	1	Baixa	3
71	ContMoneda - Eliminar	EE	11	1	Baixa	3
72	ContMoneda - Consulta	CE	11	1	Baixa	3
TIPO COMPROBANTE						
73	ContTipoComprobante - Incluir	EE	9	1	Baixa	3
74	ContTipoComprobante - Editar	EE	9	1	Baixa	3
75	ContTipoComprobante - Ver	EE	9	1	Baixa	3
76	ContTipoComprobante - Eliminar	EE	9	1	Baixa	3
77	ContTipoComprobante - Consulta	CE	9	1	Baixa	3
CTA CTBLE PARA CTA BRIA						
66	ContCuentaBancariaCtaContable -	EE	10	2	Média	4
67	ContCuentaBancariaCtaContable -	EE	10	2	Média	4
68	ContCuentaBancariaCtaContable -	EE	10	2	Média	4
69	ContCuentaBancariaCtaContable -	CE	10	2	Média	4
CTA CTBLE PARA CENTRO DE COSTO						
70	ContCentroCostoCtaCtble - Incluir	EE	10	2	Média	4
71	ContCentroCostoCtaCtble - Agreg	EE	10	2	Média	4
72	ContCentroCostoCtaCtble - Editar	EE	10	2	Média	4
73	ContCentroCostoCtaCtble - Ver	EE	10	2	Média	4
74	ContCentroCostoCtaCtble - Consu	CE	10	2	Média	4
INFORMES						
75	Libro Diario	CE	21	3	Alta	6
76	Libro Mayor	CE	28	3	Alta	6
77	Balance General	CE	10	3	Média	4
78	Libro Compra	CE	25	3	Alta	6
79	Libro Venta	CE	33	4	Alta	6
80	Libro Mayor Rubricado	CE	32	3	Alta	6
81	Clasificador de Conceptos de Viati	CE	8	2	Média	4
82	SET Balance General	CE	33	5	Alta	6
83	SET Estado de Resultados	CE	33	5	Alta	6
84	SET Estado de Flujo de Efectivo	CE	19	9	Alta	6
85	Consulta de Asientos Contables	CE	16	3	Média	4

Puntos de Función no Ajustados						
Tipo de Función	Complejidad Funcional			Total por Complejidades	Total por Tipo de Función	
EE	13	Baja	x 3	39	231	47,14%
	27	Media	x 4	108		
	14	Alta	x 6	84		
SE	0	Baja	x 4	0	0	0,00%
	0	Media	x 5	0		
	0	Alta	x 7	0		
CE	3	Baja	x 3	9	117	23,88%
	9	Media	x 4	36		
	12	Alta	x 6	72		
ALI	16	Baja	x 7	112	142	28,98%
	3	Media	x 10	30		
	0	Alta	x 15	0		
AIE	0	Baja	x 5	0	0	0,00%
	0	Media	x 7	0		
	0	Alta	x 10	0		
Total de Puntos de Función no Ajustados					490	

Ilustración 16 PF - Variable de Conteo del Módulo de Contabilidad (fuente propia).

4. Estimación por Puntos de Función del Módulo de Gestión de Documentos.

SGPTI - GESTIÓN DEL PARQUE TECNOLÓGICO DE ITAIPU

Identificación de los datos del Proyecto a estimar

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu

Propósito de la Estimación

Obtener el número de puntos de función de la aplicación producto.

Contenidos a estimar

Esta estimación abarcará el Módulo de Gestión de Documentos

- . Mantenimiento de Persona Física
- . Mantenimiento de Persona Jurídica
- . Mantenimiento de Tipo Documento
- . Mantenimiento de Tipo Expediente
- . Mantenimiento de Tipo Expediente Dependencia
- . Mantenimiento de Dependencia Detalle
- . Registrar Movimiento Expediente

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
ARCHIVOS						
1	GdocDependenciaDetalle	ALI	8	2	Baja	7
2	GdocDependenciaPersonaDetalle	ALI	8	2	Baja	7
3	GdocDependenciaTipoExp	ALI	8	2	Baja	7
4	GdocExp	ALI	24	9	Alta	15
5	GdocExpDetalle	ALI	10	2	Baja	7
6	GdocExpMovimiento	ALI	18	7	Media	10
7	GdocFisicoExterno	ALI	6	1	Baja	7
8	GdocJuridicoExterno	ALI	6	1	Baja	7
9	GdocTipoDoc	ALI	7	1	Baja	7
10	GdocTipoExp	ALI	7	1	Baja	7
11	GdocTipoMovimiento	ALI	7	1	Baja	7
PERSONA FISICA						
12	GdocFisicoExterno - Incluir	EE	8	1	Baja	3
13	GdocFisicoExterno - Editar	EE	8	1	Baja	3
14	GdocFisicoExterno - Ver	EE	8	1	Baja	3
15	GdocFisicoExterno - Eliminar	EE	8	1	Baja	3
16	GdocFisicoExterno - Consulta	CE	8	1	Baja	3
PERSONA JURIDICA						
17	GdocJuridicoExterno - Incluir	EE	8	1	Baja	3
18	GdocJuridicoExterno - Editar	EE	8	1	Baja	3
19	GdocJuridicoExterno - Ver	EE	8	1	Baja	3
20	GdocJuridicoExterno - Eliminar	EE	8	1	Baja	3
21	GdocJuridicoExterno - Consulta	CE	8	1	Baja	3
PERSONA JURIDICA						
22	GdocJuridicoExterno - Incluir	EE	8	1	Baja	3
23	GdocJuridicoExterno - Editar	EE	8	1	Baja	3
24	GdocJuridicoExterno - Ver	EE	8	1	Baja	3
25	GdocJuridicoExterno - Eliminar	EE	8	1	Baja	3
26	GdocJuridicoExterno - Consulta	CE	8	1	Baja	3
TIPO DOCUMENTO						
27	GdocTipoDoc - Incluir	EE	9	1	Baja	3
28	GdocTipoDoc - Editar	EE	9	1	Baja	3
29	GdocTipoDoc - Ver	EE	9	1	Baja	3
30	GdocTipoDoc - Eliminar	EE	9	1	Baja	3
31	GdocTipoDoc - Consulta	CE	9	1	Baja	3

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
TIPO EXPEDIENTE						
32	GdocTipoExp - Incluir	EE	9	1	Baja	3
33	GdocTipoExp - Editar	EE	9	1	Baja	3
34	GdocTipoExp - Ver	EE	9	1	Baja	3
35	GdocTipoExp - Eliminar	EE	9	1	Baja	3
36	GdocTipoExp - Consulta	CE	9	1	Baja	3
TIPO EXPEDIENTE DEPENDENCIA						
37	GdocDependenciaTipoExp - Incluir	EE	10	2	Media	4
38	GdocDependenciaTipoExp - Editar	EE	10	2	Media	4
39	GdocDependenciaTipoExp - Ver	EE	10	2	Media	4
40	GdocDependenciaTipoExp - Eliminar	EE	10	2	Media	4
41	GdocDependenciaTipoExp - Consulta	CE	10	2	Media	4
DEPENDENCIA DETALLE						
42	GdocDependenciaDetalle - Incluir	EE	10	2	Media	4
43	GdocDependenciaDetalle - Editar	EE	10	2	Media	4
44	GdocDependenciaDetalle - Ver	EE	10	2	Media	4
45	GdocDependenciaDetalle - Eliminar	EE	10	2	Media	4
46	GdocDependenciaDetalle - Consulta	CE	10	2	Media	4
PERSONA DEPENDENCIA DETALLE						
47	GdocDependenciaPersonaDetalle - Incluir	EE	10	2	Media	4
48	GdocDependenciaPersonaDetalle - Editar	EE	10	2	Media	4
49	GdocDependenciaPersonaDetalle - Ver	EE	10	2	Media	4
50	GdocDependenciaPersonaDetalle - Eliminar	EE	10	2	Media	4
51	GdocDependenciaPersonaDetalle - Consulta	CE	10	2	Media	4
ORIGINAL ENTRADA						
52	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
53	GdocExpMovimiento - Editar	EE	20	7	Alta	6
54	GdocExpMovimiento - Ver	EE	20	7	Alta	6
55	GdocExpMovimiento - Anular	EE	20	7	Alta	6
56	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
57	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
58	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
59	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
60	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
61	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
62	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
ORIGINAL ACTIVO						
63	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
64	GdocExpMovimiento - Editar	EE	20	7	Alta	6
65	GdocExpMovimiento - Ver	EE	20	7	Alta	6
66	GdocExpMovimiento - Anular	EE	20	7	Alta	6
67	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
68	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
69	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
70	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
71	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
72	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
73	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6
ORIGINAL SALIDA						
74	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
75	GdocExpMovimiento - Editar	EE	20	7	Alta	6
76	GdocExpMovimiento - Ver	EE	20	7	Alta	6
77	GdocExpMovimiento - Anular	EE	20	7	Alta	6
78	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
79	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
80	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
81	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
82	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
83	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
84	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6
ORIGINAL ENVIO						
85	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
86	GdocExpMovimiento - Editar	EE	20	7	Alta	6
87	GdocExpMovimiento - Ver	EE	20	7	Alta	6
88	GdocExpMovimiento - Anular	EE	20	7	Alta	6
89	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
90	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
91	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
92	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
93	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
94	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
95	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
COPIA DE ENTRADA						
96	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
97	GdocExpMovimiento - Editar	EE	20	7	Alta	6
98	GdocExpMovimiento - Ver	EE	20	7	Alta	6
99	GdocExpMovimiento - Anular	EE	20	7	Alta	6
100	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
101	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
102	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
103	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
104	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
105	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
106	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6
COPIA DE ACTIVO						
107	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
108	GdocExpMovimiento - Editar	EE	20	7	Alta	6
109	GdocExpMovimiento - Ver	EE	20	7	Alta	6
110	GdocExpMovimiento - Anular	EE	20	7	Alta	6
111	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
112	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
113	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
114	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
115	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
116	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
117	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6
COPIA DE SALIDA						
118	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
119	GdocExpMovimiento - Editar	EE	20	7	Alta	6
120	GdocExpMovimiento - Ver	EE	20	7	Alta	6
121	GdocExpMovimiento - Anular	EE	20	7	Alta	6
122	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
123	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
124	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
125	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
126	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
127	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
128	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
COPIA DE ENVIO						
129	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
130	GdocExpMovimiento - Editar	EE	20	7	Alta	6
131	GdocExpMovimiento - Ver	EE	20	7	Alta	6
132	GdocExpMovimiento - Anular	EE	20	7	Alta	6
133	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
134	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
135	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
136	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
137	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
138	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
139	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6
USUARIO DE ENTRADA						
140	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
141	GdocExpMovimiento - Editar	EE	20	7	Alta	6
142	GdocExpMovimiento - Ver	EE	20	7	Alta	6
143	GdocExpMovimiento - Anular	EE	20	7	Alta	6
144	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
145	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
146	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
147	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
148	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
149	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
150	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6
USUARIO ACTIVO						
151	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
152	GdocExpMovimiento - Editar	EE	20	7	Alta	6
153	GdocExpMovimiento - Ver	EE	20	7	Alta	6
154	GdocExpMovimiento - Anular	EE	20	7	Alta	6
155	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
156	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
157	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
158	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
159	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
160	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
161	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupos de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
USUARIO SALIDA						
162	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
163	GdocExpMovimiento - Editar	EE	20	7	Alta	6
164	GdocExpMovimiento - Ver	EE	20	7	Alta	6
165	GdocExpMovimiento - Anular	EE	20	7	Alta	6
166	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
167	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
168	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
169	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
170	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
171	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
172	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6
USUARIO ENVIADOS						
173	GdocExpMovimiento - Incluir	EE	20	7	Alta	6
174	GdocExpMovimiento - Editar	EE	20	7	Alta	6
175	GdocExpMovimiento - Ver	EE	20	7	Alta	6
176	GdocExpMovimiento - Anular	EE	20	7	Alta	6
177	GdocExpMovimiento - Consulta	CE	20	7	Alta	6
178	GdocExpMovimiento - AcusarArch	EE	20	7	Alta	6
179	GdocExpMovimiento - AcusarReci	EE	20	7	Alta	6
180	GdocExpMovimiento - Rechazar	EE	20	7	Alta	6
181	GdocExpMovimiento - EnviarCopia	EE	20	7	Alta	6
182	GdocExpMovimiento - CancelarEn	EE	20	7	Alta	6
183	GdocExpMovimiento - ReabrirExp	EE	20	7	Alta	6

Puntos de Función no Ajustados						
Tipo de Función	Complejidad Funcional			Total por Complejidades	Total por Tipo de Función	
EE	20	Baja	x 3	60	828	81,58%
	12	Media	x 4	48		
	120	Alta	x 6	720		
SE	0	Baja	x 4	0	0	0,00%
	0	Media	x 5	0		
	0	Alta	x 7	0		
CE	5	Baja	x 3	15	99	9,75%
	3	Media	x 4	12		
	12	Alta	x 6	72		
ALI	9	Baja	x 7	63	88	8,67%
	1	Media	x 10	10		
	1	Alta	x 15	15		
AIE	0	Baja	x 5	0	0	0,00%
	0	Media	x 7	0		
	0	Alta	x 10	0		
Total de Puntos de Función no Ajustados					1015	

Ilustración 17 PF - Variable de Conteo del Módulo de Documentos (fuente propia).

5. Estimación por Puntos de Función del Módulo de Presupuesto.

SGPTI - GESTIÓN DEL PARQUE TECNOLÓGICO DE ITAIPU

Identificación de los datos del Proyecto a estimar

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu

Propósito de la Estimación

Obtener el número de puntos de función de la aplicación producto.

Contenidos a estimar

Esta estimación abarcara el Modulo de Presupuesto

- . Mantenimiento de Rubro
- . Mantenimiento de Unidad de Medida
- . Mantenimiento de Tipo Presupuesto
- . Mantenimiento de Presupuesto
- . Mantenimiento de Modificacion Presupuestaria
- . Registrar Compromiso, Obligacion y Ejecucion

LISTADOS

Presupuesto
ModificacionPresupuestaria
Rubros
TipoPresupuesto
Unidad de Medida

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
ARCHIVOS						
1	PresupModificacionDetalle	ALI	12	3	Baja	7
2	PresupPresupuestoDetalle	ALI	31	7	Alta	15
3	PresupModificacion	ALI	11	1	Baja	7
4	PresupPresupuesto	ALI	19	3	Baja	7
5	PresupProyeccionModificacion	ALI	8	3	Baja	7
6	PresupProyeccionPresupuesto	ALI	14	2	Baja	7
7	PresupRubro	ALI	10	2	Baja	7
8	PresupTipoPresupuesto	ALI	7	1	Baja	7
9	PresupUnidadMedida	ALI	8	1	Baja	7
10	PresupPresupuestaria	ALI	22	5	Media	10
11						
RUBRO						
12	PresupRubro - Incluir	EE	12	2	Media	4
13	PresupRubro - Editar	EE	12	2	Media	4
14	PresupRubro - Ver	EE	12	2	Media	4
15	PresupRubro - Eliminar	EE	12	2	Media	4
16	PresupRubro - Consulta	CE	12	2	Media	4
UNIDAD DE MEDIDA						
17	PresupUnidadMedida - Incluir	EE	10	1	Baja	3
18	PresupUnidadMedida - Editar	EE	10	1	Baja	3
19	PresupUnidadMedida - Ver	EE	10	1	Baja	3
20	PresupUnidadMedida - Eliminar	EE	10	1	Baja	3
21	PresupUnidadMedida - Consulta	CE	10	1	Baja	3
TIPO PRESUPUESTO						
22	PresupTipoPresupuesto - Incluir	EE	9	1	Baja	3
23	PresupTipoPresupuesto - Editar	EE	9	1	Baja	3
24	PresupTipoPresupuesto - Ver	EE	9	1	Baja	3
25	PresupTipoPresupuesto - Eliminar	EE	9	1	Baja	3
26	PresupTipoPresupuesto - Consulta	CE	9	1	Baja	3
PRESUPUESTO						
27	PresupPresupuesto - Incluir	EE	21	3	Alta	6
28	PresupPresupuesto - Editar	EE	21	3	Alta	6
29	PresupPresupuesto - Ver	EE	21	3	Alta	6
30	PresupPresupuesto - Eliminar	EE	21	3	Alta	6
31	PresupPresupuesto - Consulta	CE	21	3	Alta	6
32	PresupPresupuesto - AgregarDet	EE	21	3	Alta	6
33	PresupPresupuesto - EditarDet	EE	21	3	Alta	6
34	PresupPresupuesto - EliminarDet	EE	21	3	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
MODIFICACION PRESUPUESTARIA						
35	PresupModificacion - Incluir	EE	13	1	Baja	3
36	PresupModificacion - Editar	EE	13	1	Baja	3
37	PresupModificacion - Ver	EE	13	1	Baja	3
38	PresupModificacion - Eliminar	EE	13	1	Baja	3
39	PresupModificacion - Consulta	CE	13	1	Baja	3
COMPROMISO Y OBLIGACION						
41	PresupPresupuestaria - Ver	EE	24	5	Alta	6
42	PresupPresupuestaria - Consulta	EE	24	5	Alta	6
43	PresupPresupuestaria - Comprom	EE	24	5	Alta	6
44	PresupPresupuestaria - Obligar	EE	24	5	Alta	6
45	PresupPresupuestaria - Ejecutar	CE	24	5	Alta	6
LISTADOS						
46	ListadoPresupuesto	EE	17	5	Alta	6
47	ListadoModificacionPresupuestaria	EE	14	2	Media	4
48	ListadoRubros	EE	11	2	Media	4
49	TipoPresupuesto	EE	7	1	Baja	3
50	ListadoUnidad de Medida	EE	8	1	Baja	3

Puntos de Función no Ajustados						
Tipo de Función	Complejidad Funcional			Total por Complejidades	Total por Tipo de Función	
EE	14	Baja	x 3	42	138	56,56%
	6	Media	x 4	24		
	12	Alta	x 6	72		
SE	0	Baja	x 4	0	0	0,00%
	0	Media	x 5	0		
	0	Alta	x 7	0		
CE	3	Baja	x 3	9	25	10,25%
	1	Media	x 4	4		
	2	Alta	x 6	12		
ALI	8	Baja	x 7	56	81	33,20%
	1	Media	x 10	10		
	1	Alta	x 15	15		
AIE	0	Baja	x 5	0	0	0,00%
	0	Media	x 7	0		
	0	Alta	x 10	0		
Total de Puntos de Función no Ajustados					244	

Ilustración 18 PF - Variable de Conteo del Módulo de Presupuesto (fuente propia).

6. Estimación por Puntos de Función del Módulo de RRHH.

SGPTI - GESTIÓN DEL PARQUE TECNOLÓGICO DE ITAIPU

Identificación de los datos del Proyecto a estimar

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu

Propósito de la Estimación

Obtener el número de puntos de función de la aplicación producto.

Contenidos a estimar

Esta estimación abarcara el Modulo de RRHH

- . Mantenimiento de Grupo Sanguineo
- . Mantenimiento de Tipo Documento
- . Mantenimiento de Tipo Contrato
- . Mantenimiento de Tipo Parentezco
- . Mantenimiento de Departamento
- . Mantenimiento de Ciudad
- . Mantenimiento de Tipo Capacitacion
- . Mantenimiento de Dependencia
- . Mantenimiento de Titulo Grado
- . Mantenimiento de Tipo Contacto
- . Mantenimiento de Cargo Funcion
- . Mantenimiento de Tipo Movimiento
- . Mantenimiento de Area
- . Mantenimiento de Funcion
- . Mantenimiento de Concepto Liquidacion
- . Mantenimiento de Concepto Transaccion
- . Mantenimiento de Tipo Anticipo
- . Mantenimiento de Parametro Global
- . Mantenimiento de Tipo Marcacion
- . Mantenimiento de Tipo Solicitud
- . Mantenimiento de Horario Turno
- . Mantenimiento de Jornada Especial
- . Mantenimiento de Esquema Marcacion
- . Mantenimiento de Legajo Personal
- . Mantenimiento de Contrato Personal

Contenidos a estimar

- . Mantenimiento de Transaccion Salario
- . Mantenimiento de Pago Aguinaldo
- . Mantenimiento de Horario Persona
- . Mantenimiento de Admin Solicitudes
- . Mantenimiento de Firma Documento
- . Registrar Salario
- . Registrar Mes Salario
- . Registrar Asistencia Personal
- . Registrar Proceso Marcacion

LISTADOS

GrupoSanguineo
TipoDocumento
TipoContacto
TipoParentezco
Departamento
Ciudad
TipoCapacitacion
TipoGrado
TipoContrato
TipoCargo
TipoMovimiento
Area
Tipo de Funcion
LiquidacionSalario
ConceptoTransaccion
TipoAnticipo
Parametro Global
TipoMarcacion
TipoSolicitud
HorarioTurno
JornadaEspecial
EsquemaMarcacion
Dependencia

RELATORIOS

Imprimir Asistencia Personal
Imprimir Asistencia Importada
Imprimir Horas Extras Aprobadas
Imprimir Horas de Descuento
Imprimir Cumpleaños
Imprimir Reporte Pago IPS
Imprimir Reporte de Sueldos y Jornales
Imprimir Reporte Estadísticas MJT
Imprimir Reporte Ministerio de Justicia y Trabajo
Imprimir Reporte Ministerio de Justicia y Trabajo Detalle
Imprimir Reporte de Sueldos y Jornales Detalle

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
ARCHIVOS						
1	Persona	ALI	13	4	Baja	7
2	PersonaFisica	ALI	15	1	Baja	7
3	PersonaFisicaDocumento	ALI	5	1	Baja	7
4	PersonaContacto	ALI	5	2	Baja	7
5	PersonaJuridica	ALI	7	1	Baja	7
6	PersonaCapacitacion	ALI	23	3	Media	10
7	PersonaDireccion	ALI	14	5	Baja	7
8	PersonaFormacionAcademica	ALI	22	4	Media	10
9	PersonaParentezco	ALI	29	6	Alta	15
10	PersonaCuentaBancaria	ALI	9	2	Baja	7
11	PersonaDisciplinaAcademica	ALI	7	1	Baja	7
12	Contrataciones	ALI	16	4	Baja	7
13	ContratacionesConvenio	ALI	8	2	Baja	7
14	ContratacionesConvenioFisica	ALI	17	6	Media	10
15	Convenio	ALI	13	3	Baja	7
16	GrupoSanguineo	ALI	7	1	Baja	7
17	TipoDocumento	ALI	7	1	Baja	7
18	TipoContrato	ALI	7	1	Baja	7
19	TipoParentezco	ALI	7	1	Baja	7
20	Departamento	ALI	8	1	Baja	7
21	Ciudad	ALI	8	2	Baja	7
22	TipoCapacitacion	ALI	7	1	Baja	7
23	Dependencia	ALI	10	2	Baja	7
24	TituloGrado	ALI	9	2	Baja	7
25	TipoContacto	ALI	7	1	Baja	7
26	CargoFuncion	ALI	8	2	Baja	7
27	TipoMovimiento	ALI	7	1	Baja	7
28	Area	ALI	8	1	Baja	7
29	Funcion	ALI	7	1	Baja	7
30	LiquidacionSalario	ALI	13	2	Baja	7
31	LiquidacionSalarioDetalle	ALI	20	6	Alta	15
32	LiquidacionSalarioAnticipo	ALI	14	4	Baja	7
33	LiquidacionSalarioTransaccion	ALI	27	9	Alta	15
34	Banco	ALI	1	1	Baja	7
35	ConceptoLiquidacion	ALI	12	2	Baja	7
36	ConceptoTransaccion	ALI	10	1	Baja	7
37	TipoAnticipo	ALI	9	1	Baja	7
38	ParametroGlobal	ALI	11	1	Baja	7
39	MigracionMarcacion	ALI	11	1	Baja	7
40	RegistroMarcacionTMP	ALI	14	2	Baja	7

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
DEPARTAMENTO						
79	Departamento - Incluir	EE	10	1	Baixa	3
80	Departamento - Editar	EE	10	1	Baixa	3
81	Departamento - Eliminar	EE	10	1	Baixa	3
82	Departamento - Ver	EE	10	1	Baixa	3
83	Departamento - Consulta	CE	10	1	Baixa	3
CIUDAD						
84	Ciudad - Incluir	EE	10	2	Média	4
85	Ciudad - Editar	EE	10	2	Média	4
86	Ciudad - Eliminar	EE	10	2	Média	4
87	Ciudad - Ver	EE	10	2	Média	4
88	Ciudad - Consulta	CE	10	2	Média	4
TIPO CAPACITACION						
89	TipoCapacitacion - Incluir	EE	9	1	Baixa	3
90	TipoCapacitacion - Editar	EE	9	1	Baixa	3
91	TipoCapacitacion - Eliminar	EE	9	1	Baixa	3
92	TipoCapacitacion - Ver	EE	9	1	Baixa	3
93	TipoCapacitacion - Consulta	CE	9	1	Baixa	3
DEPENDENCIA						
94	Dependencia - Incluir	EE	12	2	Média	4
95	Dependencia - Editar	EE	12	2	Média	4
96	Dependencia - Eliminar	EE	12	2	Média	4
97	Dependencia - Ver	EE	12	2	Média	4
98	Dependencia - Consulta	CE	12	2	Média	4
TITULO GRADO						
99	TipoGrado - Incluir	EE	11	2	Média	4
100	TipoGrado - Editar	EE	11	2	Média	4
101	TipoGrado - Eliminar	EE	11	2	Média	4
102	TipoGrado - Ver	EE	11	2	Média	4
103	TipoGrado - Consulta	CE	11	2	Média	4
TIPO CONTACTO						
104	TipoContrato - Incluir	EE	9	1	Baixa	3
105	TipoContrato - Editar	EE	9	1	Baixa	3
106	TipoContrato - Eliminar	EE	9	1	Baixa	3
107	TipoContrato - Ver	EE	9	1	Baixa	3
108	TipoContrato - Consulta	CE	9	1	Baixa	3

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
CARGO FUNCION						
109	TipoCargo - Incluir	EE	10	2	Média	4
110	TipoCargo - Editar	EE	10	2	Média	4
111	TipoCargo - Eliminar	EE	10	2	Média	4
112	TipoCargo - Ver	EE	10	2	Média	4
113	TipoCargo - Consulta	CE	10	2	Média	4
TIPO MOVIMIENTO						
114	TipoMovimiento - Incluir	EE	9	1	Baixa	3
115	TipoMovimiento - Editar	EE	9	1	Baixa	3
116	TipoMovimiento - Eliminar	EE	9	1	Baixa	3
117	TipoMovimiento - Ver	EE	9	1	Baixa	3
118	TipoMovimiento - Consulta	CE	9	1	Baixa	3
AREA						
119	Area - Incluir	EE	10	1	Baixa	3
120	Area - Editar	EE	10	1	Baixa	3
121	Area - Eliminar	EE	10	1	Baixa	3
122	Area - Ver	EE	10	1	Baixa	3
123	Area - Consulta	CE	10	1	Baixa	3
FUNCION						
124	Funcion - Incluir	EE	9	1	Baixa	3
125	Funcion - Editar	EE	9	1	Baixa	3
126	Funcion - Eliminar	EE	9	1	Baixa	3
127	Funcion - Ver	EE	9	1	Baixa	3
128	Funcion - Consulta	CE	9	1	Baixa	3
CONCEPTO LIQUIDACION						
129	ConceptoLiquidacion - Incluir	EE	14	2	Média	4
130	ConceptoLiquidacion - Editar	EE	14	2	Média	4
131	ConceptoLiquidacion - Eliminar	EE	14	2	Média	4
132	ConceptoLiquidacion - Ver	EE	14	2	Média	4
133	ConceptoLiquidacion - Consulta	CE	14	2	Média	4
CONCEPTO TRANSACCION						
134	ConceptoTransaccion - Incluir	EE	12	1	Baixa	3
135	ConceptoTransaccion - Editar	EE	12	1	Baixa	3
136	ConceptoTransaccion - Eliminar	EE	12	1	Baixa	3
137	ConceptoTransaccion - Ver	EE	12	1	Baixa	3
138	ConceptoTransaccion - Consulta	CE	12	1	Baixa	3
TIPO ANTICIPO						
139	TipoAnticipo - Incluir	EE	11	1	Baixa	3
140	TipoAnticipo - Editar	EE	11	1	Baixa	3
141	TipoAnticipo - Eliminar	EE	11	1	Baixa	3
142	TipoAnticipo - Ver	EE	11	1	Baixa	3
143	TipoAnticipo - Consulta	CE	11	1	Baixa	3

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
PARAMETRO GLOBAL						
144	ParametroGlobal - Incluir	EE	13	1	Baixa	3
145	ParametroGlobal - Editar	EE	13	1	Baixa	3
146	ParametroGlobal - Eliminar	EE	13	1	Baixa	3
147	ParametroGlobal - Ver	EE	13	1	Baixa	3
148	ParametroGlobal - Consulta	CE	13	1	Baixa	3
TIPO MARCACION						
149	TipoMarcacion - Incluir	EE	11	1	Baixa	3
150	TipoMarcacion - Editar	EE	11	1	Baixa	3
151	TipoMarcacion - Eliminar	EE	11	1	Baixa	3
152	TipoMarcacion - Ver	EE	11	1	Baixa	3
153	TipoMarcacion - Consulta	CE	11	1	Baixa	3
TIPO SOLICITUD						
154	TipoSolicitud - Incluir	EE	14	1	Baixa	3
155	TipoSolicitud - Editar	EE	14	1	Baixa	3
156	TipoSolicitud - Eliminar	EE	14	1	Baixa	3
157	TipoSolicitud - Ver	EE	14	1	Baixa	3
158	TipoSolicitud - Consulta	CE	14	1	Baixa	3
LEGAJO PERSONAL						
159	PersonaFisica - Incluir	EE	17	1	Média	4
160	PersonaFisica - Editar	EE	17	1	Média	4
161	PersonaFisica - Anular	EE	17	1	Média	4
162	PersonaFisica - Ver	EE	17	1	Média	4
163	PersonaFisica - Consulta	CE	17	1	Baixa	3
CONTRATO PERSONAL						
164	Contrataciones - Incluir	EE	18	4	Alta	6
165	Contrataciones - Editar	EE	18	4	Alta	6
166	Contrataciones - Anular	EE	18	4	Alta	6
167	Contrataciones - Ver	EE	18	4	Alta	6
168	Contrataciones - Consulta	CE	18	4	Alta	6
GENERACION DE REGISTRO SALARIO						
169	Generacion de Registro Salario - FEE		71	21	Alta	6
170	Generacion de Registro Salario - CCE		71	21	Alta	6
TRANSACCION SALARIO						
171	Transaccion - Incluir	EE	29	11	Alta	6
172	Transaccion - Editar	EE	29	11	Alta	6
173	Transaccion - Anular	EE	29	11	Alta	6
174	Transaccion - Ver	EE	29	11	Alta	6
175	Transaccion - Consulta	CE	29	11	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
CIERRE MES SALARIO						
176	CierreMesSalario - Proceso	EE	35	16	Alta	6
177	CierreMesSalario - Consulta	CE	35	16	Alta	6
PAGO AGUINALDO						
178	LiquidacionSalario - Incluir	EE	15	2	Média	4
179	LiquidacionSalario - Editar	EE	15	2	Média	4
180	LiquidacionSalario - Anular	EE	15	2	Média	4
181	LiquidacionSalario - Ver	EE	15	2	Média	4
182	LiquidacionSalario - Consulta	CE	15	2	Média	4
183	LiquidacionSalarioDetalle - Incluir	EE	22	6	Alta	6
184	LiquidacionSalarioDetalle - Editar	EE	22	6	Alta	6
185	LiquidacionSalarioDetalle - Anular	EE	22	6	Alta	6
ASISTENCIA PERSONAL						
186	AsistenciaPersonal - Proceso	EE	45	3	Alta	6
187	AsistenciaPersonal - Consulta	CE	45	3	Alta	6
ADMIN HORARIO PERSONA						
188	HorarioPersona - Incluir	EE	11	2	Média	4
189	HorarioPersona - Editar	EE	11	2	Média	4
190	HorarioPersona - Anular	EE	11	2	Média	4
191	HorarioPersona - Ver	EE	11	2	Média	4
192	HorarioPersona - Consulta	CE	11	2	Média	4
PROCESO DE MARCACION						
193	MigracionMarcacion - Proceso	EE	27	3	Alta	6
194	MigracionMarcacion - Consulta	CE	27	3	Alta	6
SOLICITUD PERSONA						
195	SolicitudPersonaDetalle - Incluir	EE	9	2	Média	4
196	SolicitudPersonaDetalle - Editar	EE	9	2	Média	4
197	SolicitudPersonaDetalle - Anular	EE	9	2	Média	4
198	SolicitudPersona - Ver	EE	20	3	Alta	6
199	SolicitudPersona - Consulta	CE	20	3	Alta	6
200	SolicitudPersona - Aprobar	CE	20	3	Alta	6
201	SolicitudPersona - Confirmar	CE	20	3	Alta	6
FIRMA DE DOCUMENTO						
202	DocFirma - Incluir	EE	12	4	Alta	6
203	DocFirma - Editar	EE	12	4	Alta	6
204	DocFirma - Anular	EE	12	4	Alta	6
205	DocFirma - Ver	EE	12	4	Alta	6
206	DocFirma - Consulta	CE	12	4	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
HORARIO TURNO						
207	HorarioTurno - Incluir	EE	13	2	Média	4
208	HorarioTurno - Editar	EE	13	2	Média	4
209	HorarioTurno - Eliminar	EE	13	2	Média	4
210	HorarioTurno - Ver	EE	13	2	Média	4
211	HorarioTurno - Consulta	CE	13	2	Média	4
JORNADA ESPECIAL						
212	JornadaEspecial - Incluir	EE	15	1	Baixa	3
213	JornadaEspecial - Editar	EE	15	1	Baixa	3
214	JornadaEspecial - Eliminar	EE	15	1	Baixa	3
215	JornadaEspecial - Ver	EE	15	1	Baixa	3
216	JornadaEspecial - Consulta	CE	15	1	Baixa	3
ESQUEMA MARCACION						
217	EsquemaMarcacion - Incluir	EE	9	1	Baixa	3
218	EsquemaMarcacion - Editar	EE	9	1	Baixa	3
219	EsquemaMarcacion - Eliminar	EE	9	1	Baixa	3
220	EsquemaMarcacion - Ver	EE	9	1	Baixa	3
221	EsquemaMarcacion - Consulta	CE	9	1	Baixa	3
LISTADOS						
222	GrupoSanguineo	CE	4	1	Baixa	3
223	TipoDocumento	CE	5	1	Baixa	3
224	TipoContacto	CE	5	1	Baixa	3
225	TipoParentezco	CE	5	1	Baixa	3
226	Departamento	CE	6	1	Baixa	3
227	Ciudad	CE	7	2	Média	4
228	TipoCapacitacion	CE	5	1	Baixa	3
229	TipoGrado	CE	5	1	Baixa	3
230	TipoContrato	CE	5	1	Baixa	3
231	TipoCargo	CE	6	1	Baixa	3
232	TipoMovimiento	CE	5	1	Baixa	3
233	Area	CE	6	1	Baixa	3
234	Tipo de Funcion	CE	5	1	Baixa	3
235	LiquidacionSalario	CE	4	1	Baixa	3
236	ConceptoTransaccion	CE	5	1	Baixa	3
237	TipoAnticipo	CE	6	1	Baixa	3
238	Parametro Global	CE	6	1	Baixa	3
239	TipoMarcacion	CE	7	1	Baixa	3
240	TipoSolicitud	CE	5	1	Baixa	3
241	HorarioTurno	CE	8	2	Média	4
242	JornadaEspecial	CE	8	1	Baixa	3
243	EsquemaMarcacion	CE	5	1	Baixa	3

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
244	Dependencia	CE	6	1	Baixa	3
INFORMES						
245	Imprimir Asistencia Personal	CE	26	8	Alta	6
246	Imprimir Asistencia Importada	CE	7	1	Baixa	3
247	Imprimir Horas Extras Aprobadas	CE	13	2	Média	4
248	Imprimir Horas de Descuento	CE	13	2	Média	4
249	Imprimir Cumpleaños	CE	7	1	Baixa	3
250	Imprimir Reporte Pago IPS	CE	8	2	Média	4
251	Imprimir Reporte de Sueldos y Jo	CE	45	15	Alta	6
252	Imprimir Reporte Estadísticas MJT	CE	59	11	Alta	6
253	Imprimir Reporte Ministerio de Ju	CE	16	2	Média	4
254	Imprimir Reporte Ministerio de Ju	CE	44	5	Alta	6
255	Imprimir Reporte de Sueldos y Jo	CE	59	15	Alta	6

Puntos de Función no Ajustados						
Tipo de Función	Complejidad Funcional			Total por Complejidades	Total por Tipo de Función	
EE	68	Baja	x 3	204	480	40,03%
	39	Media	x 4	156		
	20	Alta	x 6	120		
SE	0	Baja	x 4	0	0	0,00%
	0	Media	x 5	0		
	0	Alta	x 7	0		
CE	41	Baja	x 3	123	269	22,44%
	14	Media	x 4	56		
	15	Alta	x 6	90		
ALI	50	Baja	x 7	350	450	37,53%
	4	Media	x 10	40		
	4	Alta	x 15	60		
AIE	0	Baja	x 5	0	0	0,00%
	0	Media	x 7	0		
	0	Alta	x 10	0		
Total de Puntos de Función no Ajustados					1199	

Ilustración 19 PF - Variable de Conteo del Módulo de RRHH (fuente propia).

7. Estimación por Puntos de Función del Módulo de Tesorería.

SGPTI - GESTIÓN DEL PARQUE TECNOLÓGICO DE ITAIPU

Identificación de los datos del Proyecto a estimar

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu

Propósito de la Estimación

Obtener el número de puntos de función de la aplicación producto.

Contenidos a estimar

Esta estimación abarcará el Módulo de Tesorería

- . Registrar de Transferencia Bancaria
- . Registrar de Confirmar Transferencia
- . Registrar de Saldos
- . Mantenimiento de Banco
- . Mantenimiento de Sucursal
- . Mantenimiento de Cuenta Bancaria
- . Mantenimiento de Chequera
- . Mantenimiento de Tipo Cuenta Bancaria
- . Registrar de Orden Pago General
- . Mantenimiento de Concepto Viatico
- . Mantenimiento de Nivel Concepto Viatico
- . Mantenimiento de Motivo Viatico
- . Mantenimiento de Medio de Traslado
- . Registrar Ingreso
- . Mantenimiento de Tipo Concepto Venta
- . Mantenimiento de Concepto Venta
- . Mantenimiento de Tipo Cliente
- . Mantenimiento de Cliente
- . Mantenimiento de Cheque
- . Registrar de Cheque para Pago
- . Registrar de Recibo Cheque
- . Mantenimiento de Margen de Cotizacion
- . Registrar de Facturas
- . Registrar de Generacion de Orden Pago Factura
- . Registrar Orden Pago Cheque Salario

Contenidos a estimar

- . Registrar Orden Pago Salario
- . Registrar Aprobacion Orden Pago Salario
- . Registrar Pago Salario Debito
- . Registrar Cobro Salario Banco
- . Registrar Solicitud Viatico
- . Registrar Autorizacion de Viatico
- . Registrar Viatico Tesoreria
- . Registrar Anulacion de Viatico Tesoreria
- . Registrar Generacion de Orden Pago Viatico
- . Registrar Orden de Pago Viatico
- . Registrar Medio de Traslado
- . Registrar Orden Pago Caja Chica
- . Registrar Configuracion de Viatico
- . Registrar Confirmar Ingreso
- . Registrar Factura Cliente

LISTADOS

ListadoBancoSucursales
ListadoTransferenciaBancaria
ListadoSumaYSaldos
ListadoBanco
ListadoCuentaBancaria
ListadoChequera
ListadoTipoCuentaBancaria
ListadoOrdenPago
ListadoMargenCotizacion
ListadoFacturas
ListadoFacturasCliente
ListadoOrdenPagoSalario
ListadoViatico
ListadoIngreso

RELATORIOS

Imprimir Cheques
Calcular Margen Cotizacion
Listar Facturas
Listar Cheques a Aprobar
Cheques Emitidos Cobrados y no Cobrados
Imprimir Planilla de Pagos
Imprimir Libro Banco por Convenio
Imprimir Resumen de Libro Banco
Imprimir Listado de Ordenes de Pago

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
ARCHIVOS						
1	TesrBanco	ALI	8	1	Baja	7
2	TesrBancoSucursal	ALI	13	1	Baja	7
3	TesrCajaChica	ALI	13	3	Baja	7
4	TesrCajaChicaMovimiento	ALI	14	3	Baja	7
5	TesrChequeEmitido	ALI	27	6	Alta	15
6	TesrChequera	ALI	17	3	Baja	7
7	TesrClasificadorCuenta	ALI	4	1	Baja	7
8	TesrCliente	ALI	7	1	Baja	7
9	TesrConceptoVenta	ALI	8	1	Baja	7
10	TesrConceptoViatico	ALI	8	1	Baja	7
11	TesrConfViatico	ALI	8	2	Baja	7
12	TesrConvenioCuentaBancaria	ALI	8	2	Baja	7
13	TesrCuentaBancaria	ALI	14	3	Baja	7
14	TesrFacturaCab	ALI	44	11	Alta	15
15	TesrFacturaCliente	ALI	23	3	Media	10
16	TesrFacturaClienteDet	ALI	11	2	Baja	7
17	TesrFacturaConfig	ALI	17	1	Baja	7
18	TesrFacturaDet	ALI	16	5	Baja	7
19	TesrIngreso	ALI	14	3	Baja	7
20	TesrMedioTraslado	ALI	7	1	Baja	7
21	TesrMotivoViatico	ALI	7	1	Baja	7
22	TesrMovimientoContable	ALI	1	1	Baja	7
23	TesrNivelConceptoViatico	ALI	9	2	Baja	7
24	TesrNivelViatico	ALI	7	1	Baja	7
25	TesrOrdenPago	ALI	26	3	Media	10
26	TesrOrdenPagoDet	ALI	16	7	Media	10
27	TesrPagoBanco	ALI	4	3	Baja	7
28	TesrPersonaDependencia	ALI	7	2	Baja	7
29	TesrReciboPago	ALI	10	1	Baja	7
30	TesrSaldo	ALI	9	1	Baja	7
31	TesrTipoCliente	ALI	7	1	Baja	7
32	TesrTipoConceptoVenta	ALI	8	1	Baja	7
33	TesrTipoCuenta	ALI	7	1	Baja	7
34	TesrTipoPago	ALI	8	1	Baja	7
35	TesrTransaccion	ALI	10	1	Baja	7

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I							
#	Proceso elemental o Grupo de datos	Tipo	Calculos				
			TD	AR/TR	Complex.	PF	
36	TesrTransferencia	ALI	20	5	Media	10	
37	TesrViatico	ALI	23	8	Alta	15	
38	TesrViaticoAprob	ALI	8	2	Baja	7	
39	TesrViaticoDet	ALI	14	2	Baja	7	
40	TesrViaticoDocumento	ALI	10	1	Baja	7	
41	RRHH_pagoSalario	ALI	16	5	Baja	7	
TRANSFERENCIA BANCARIA							
45	TesrTransferencia - Incluir	EE	22	5	Alta	6	
46	TesrTransferencia - Editar	EE	22	5	Alta	6	
47	TesrTransferencia - Ver	EE	22	5	Alta	6	
48	TesrTransferencia - Eliminar	EE	22	5	Alta	6	
49	TesrTransferencia - Consulta	CE	22	5	Alta	6	
CONFIRMAR TRANSFERENCIA							
51	TesrTransferencia - Ver	EE	22	5	Alta	6	
52	TesrTransferencia - Confirmar	EE	22	5	Alta	6	
53	TesrTransferencia - Consulta	CE	22	5	Alta	6	
SALDOS							
54	TesrSaldo - Incluir	EE	11	1	Baja	3	
55	TesrSaldo - Editar	EE	11	1	Baja	3	
56	TesrSaldo - Ver	EE	11	1	Baja	3	
57	TesrSaldo - Eliminar	EE	11	1	Baja	3	
58	TesrSaldo - Consulta	CE	11	1	Baja	3	
BANCO							
59	TesrBanco - Incluir	EE	10	1	Baja	3	
60	TesrBanco - Editar	EE	10	1	Baja	3	
61	TesrBanco - Ver	EE	10	1	Baja	3	
62	TesrBanco - Eliminar	EE	10	1	Baja	3	
63	TesrBanco - Consulta	CE	10	1	Baja	3	
SUCURSAL							
64	TesrBancoSucursal - Incluir	EE	15	1	Baja	3	
65	TesrBancoSucursal - Editar	EE	15	1	Baja	3	
66	TesrBancoSucursal - Ver	EE	15	1	Baja	3	
67	TesrBancoSucursal - Eliminar	EE	15	1	Baja	3	
68	TesrBancoSucursal - Consulta	CE	15	1	Baja	3	
CUENTA BANCARIA							
69	TesrCuentaBancaria - Incluir	EE	16	3	Alta	6	
70	TesrCuentaBancaria - Editar	EE	16	3	Alta	6	
71	TesrCuentaBancaria - Ver	EE	16	3	Alta	6	
72	TesrCuentaBancaria - Eliminar	EE	16	3	Alta	6	
73	TesrCuentaBancaria - Consulta	CE	16	3	Media	4	

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
CHEQUERA						
74	TesrChequera - Incluir	EE	19	3	Alta	6
75	TesrChequera - Editar	EE	19	3	Alta	6
76	TesrChequera - Ver	EE	19	3	Alta	6
77	TesrChequera - Eliminar	EE	19	3	Alta	6
78	TesrChequera - Consulta	CE	19	3	Media	4
TIPO CUENTA BANCARIA						
79	TesrTipoCuenta - Incluir	EE	9	1	Baja	3
80	TesrTipoCuenta - Editar	EE	9	1	Baja	3
81	TesrTipoCuenta - Ver	EE	9	1	Baja	3
82	TesrTipoCuenta - Eliminar	EE	9	1	Baja	3
83	TesrTipoCuenta - Consulta	CE	9	1	Baja	3
ORDEN PAGO GENERAL						
84	TesrOrdenPago - Incluir	EE	28	3	Alta	6
85	TesrOrdenPago - Editar	EE	28	3	Alta	6
86	TesrOrdenPago - Ver	EE	28	3	Alta	6
87	TesrOrdenPago - Anular	EE	28	3	Alta	6
88	TesrOrdenPago - Consulta	CE	28	3	Alta	6
CHEQUE						
89	TesrChequeEmitido - Consulta	CE	29	6	Alta	6
CHEQUE PARA PAGOS						
90	TesrChequeEmitido - Consulta	EE	29	6	Alta	6
91	TesrChequeEmitido - Proceso	CE	29	6	Alta	6
RECIBO CHEQUE						
92	TesrChequeEmitido - Consulta	EE	29	6	Alta	6
93	TesrChequeEmitido - Proceso	CE	29	6	Alta	6
MARGEN COTIZACION						
94	MargenCotizacion - Asignar	EE	55	9	Alta	6
95	MargenCotizacion - Ver	EE	55	9	Alta	6
96	MargenCotizacion - Anular	EE	55	9	Alta	6
97	MargenCotizacion - Consulta	CE	55	9	Alta	6
FACTURA						
98	TesrFacturaCab - Proceso	EE	46	11	Alta	6
99	TesrFacturaCab - Consulta	CE	46	11	Alta	6
GENERAR ORDEN PAGO FACTURA						
100	TesrFacturaCab - Proceso	EE	46	11	Alta	6
101	TesrFacturaCab - Consulta	CE	46	11	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
ORDEN PAGO FACTURA						
102	TesrOrdenPago - Incluir	EE	28	3	Alta	6
103	TesrOrdenPago - Editar	EE	28	3	Alta	6
104	TesrOrdenPago - Ver	EE	28	3	Alta	6
105	TesrOrdenPago - Anular	EE	28	3	Alta	6
106	TesrOrdenPago - Consulta	CE	28	3	Alta	6
ORDEN PAGO CHEQUE SALARIO						
107	OrdenPagoChequeSalario - Proceso	EE	61	18	Alta	6
108	TesrOrdenPago - Consulta	CE	28	3	Alta	6
ORDEN PAGO SALARIO						
109	TesrOrdenPago - Incluir	EE	28	3	Alta	6
110	TesrOrdenPago - Editar	EE	28	3	Alta	6
111	TesrOrdenPago - Ver	EE	28	3	Alta	6
112	TesrOrdenPago - Anular	EE	28	3	Alta	6
113	TesrOrdenPago - Consulta	CE	28	3	Alta	6
APROBAR ORDEN PAGO SALARIO						
114	TesrOrdenPago - Ver	EE	28	3	Alta	6
115	TesrOrdenPago - Proceso	EE	28	3	Alta	6
116	TesrOrdenPago - Consulta	CE	28	3	Alta	6
PAGO SALARIO DEBITO						
117	TesrOrdenPago - Incluir	EE	28	3	Alta	6
118	TesrOrdenPago - Editar	EE	28	3	Alta	6
119	TesrOrdenPago - Ver	EE	28	3	Alta	6
120	TesrOrdenPago - Anular	EE	28	3	Alta	6
121	TesrOrdenPago - Consulta	CE	28	3	Alta	6
CHEQUE PAGO SALARIO						
122	TesrChequeEmitido - AsignarCheque	EE	28	3	Alta	6
123	TesrChequeEmitido - LiberarCheque	EE	28	3	Alta	6
124	TesrChequeEmitido - Consulta	CE	28	3	Alta	6
COBRO SALARIO BANCO						
125	TesrPagoBanco - Proceso	EE	38	12	Alta	6
126	TesrOrdenPagoDet - Consulta	CE	18	7	Alta	6
SOLICITUD VIATICO						
127	TesrViatico - Incluir	EE	25	8	Alta	6
128	TesrViatico - Editar	EE	25	8	Alta	6
129	TesrViatico - Ver	EE	25	8	Alta	6
130	TesrViatico - Anular	EE	25	8	Alta	6
131	TesrViatico - Consulta	CE	25	8	Alta	6

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de I						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
AUTORIZAR VIATICO						
132	TesrViatico - Proceso	EE	33	10	Alta	6
133	TesrViatico - Consulta	CE	25	8	Alta	6
ADMIN VIATICO TESORERIA						
134	TesrViatico - Proceso	EE	25	8	Alta	6
135	TesrViatico - Consulta	CE	25	8	Alta	6
ANULAR VIATICO TESORERIA						
136	TesrViatico - Proceso	EE	51	11	Alta	6
137	TesrViatico - Consulta	CE	25	8	Alta	6
GENERAR ORDEN PAGO VIATICO						
138	TesrViatico - Proceso	EE	33	10	Alta	6
139	TesrViatico - Consulta	CE	25	8	Alta	6
ORDEN PAGO VIATICO						
140	TesrOrdenPago - Ver	EE	28	3	Alta	6
141	TesrOrdenPago - Editar	EE	28	3	Alta	6
142	TesrOrdenPago - Consulta	CE	28	3	Alta	6
CONCEPTO VIATICO						
143	TesrNivelViatico - Incluir	EE	9	1	Baja	3
144	TesrNivelViatico - Ver	EE	9	1	Baja	3
145	TesrNivelViatico - Editar	EE	9	1	Baja	3
146	TesrNivelViatico - Eliminar	EE	9	1	Baja	3
147	TesrNivelViatico - Consulta	CE	9	1	Baja	3
NIVELCONCEPTO VIATICO						
148	TesrConceptoViatico - Incluir	EE	11	2	Media	4
149	TesrConceptoViatico - Ver	EE	11	2	Media	4
150	TesrConceptoViatico - Editar	EE	11	2	Media	4
151	TesrConceptoViatico - Eliminar	EE	11	2	Media	4
152	TesrConceptoViatico - Consulta	CE	11	2	Media	4
MOTIVO DE VIATICO						
153	TesrMotivoViatico - Incluir	EE	9	1	Baja	3
154	TesrMotivoViatico - Ver	EE	9	1	Baja	3
155	TesrMotivoViatico - Editar	EE	9	1	Baja	3
156	TesrMotivoViatico - Eliminar	EE	9	1	Baja	3
157	TesrMotivoViatico - Consulta	CE	9	1	Baja	3
MEDIO DE TRASLADO						
158	TesrMedioTraslado - Incluir	EE	9	1	Baja	3
159	TesrMedioTraslado - Ver	EE	9	1	Baja	3
160	TesrMedioTraslado - Editar	EE	9	1	Baja	3
161	TesrMedioTraslado - Eliminar	EE	9	1	Baja	3
162	TesrMedioTraslado - Consulta	CE	9	1	Baja	3

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
CONFIGURACION VIATICO						
163	TesrConfViatico - Incluir	EE	10	2	Media	4
164	TesrConfViatico - Ver	EE	10	2	Media	4
165	TesrConfViatico - Consulta	CE	10	2	Media	4
ORDEN PAGO CAJA CHICA						
166	TesrCajaChicaMovimiento - Proce	EE	16	3	Alta	6
167	TesrCajaChicaMovimiento - Consu	CE	16	3	Media	4
INGRESO						
168	TesrIngreso - Incluir	EE	16	3	Alta	6
169	TesrIngreso - Ver	EE	16	3	Alta	6
170	TesrIngreso - Editar	EE	16	3	Alta	6
171	TesrIngreso - Eliminar	EE	16	3	Alta	6
172	TesrIngreso - Consulta	CE	16	3	Media	4
CONFIRMAR INGRESO						
173	TesrIngreso - Proceso	EE	16	3	Alta	6
174	TesrIngreso - Consulta	CE	16	3	Media	4
FACTURA CLIENTE						
175	TesrFacturaCliente - Incluir	EE	25	3	Alta	6
176	TesrFacturaCliente - Editar	EE	25	3	Alta	6
177	TesrFacturaCliente - Ver	EE	25	3	Alta	6
178	TesrFacturaCliente - Anular	EE	25	3	Alta	6
179	TesrFacturaCliente - Consulta	EE	25	3	Alta	6
TIPO CONCEPTO VENTA						
180	TesrTipoConceptoVenta - Incluir	EE	10	1	Baja	3
181	TesrTipoConceptoVenta - Editar	EE	10	1	Baja	3
182	TesrTipoConceptoVenta - Ver	EE	10	1	Baja	3
183	TesrTipoConceptoVenta - Elimina	EE	10	1	Baja	3
184	TesrTipoConceptoVenta - Consult	CE	10	1	Baja	3
CONCEPTO VENTA						
185	TesrConceptoVenta - Incluir	EE	10	1	Baja	3
186	TesrConceptoVenta - Editar	EE	10	1	Baja	3
187	TesrConceptoVenta - Ver	EE	10	1	Baja	3
188	TesrConceptoVenta - Eliminar	EE	10	1	Baja	3
189	TesrConceptoVenta - Consulta	CE	10	1	Baja	3

Identificación, Determinación de la Complejidad y Cálculo de la Contribución a los PF no Ajustados

Proyecto : SGPTI - Gestión del Parque Tecnológico de Itaipu						
#	Proceso elemental o Grupo de datos	Tipo	Calculos			
			TD	AR/TR	Complex.	PF
TIPO CLIENTE						
190	TesrTipoCliente - Incluir	EE	9	1	Baja	3
191	TesrTipoCliente - Editar	EE	9	1	Baja	3
192	TesrTipoCliente - Ver	EE	9	1	Baja	3
193	TesrTipoCliente - Eliminar	EE	9	1	Baja	3
194	TesrTipoCliente - Consulta	CE	9	1	Baja	3
CLIENTE						
195	TesrCliente - Incluir	EE	9	1	Baja	3
196	TesrCliente - Editar	EE	9	1	Baja	3
197	TesrCliente - Ver	EE	9	1	Baja	3
198	TesrCliente - Eliminar	EE	9	1	Baja	3
199	TesrCliente - Consulta	CE	9	1	Baja	3
INFORMES						
200	Imprimir Cheques	CE	16	3	Media	4
201	Calcular Margen Cotizacion	CE	6	1	Baja	3
202	Listar Facturas	CE	19	3	Media	4
203	Listar Cheques a Aprobar	CE	9	3	Media	4
204	Cheques Emitidos Cobrados y no	CE	12	3	Media	4
205	Imprimir Planilla de Pagos	CE	16	3	Media	4
206	Imprimir Libro Banco por Conven	CE	12	1	Baja	3
207	Imprimir Resumen de Libro Banco	CE	12	3	Media	4
208	Imprimir Listado de Ordenes de P	CE	24	6	Alta	6
LISTADOS						
209	ListadoBancoSucursales	CE	11	2	Media	4
210	ListadoTransferenciaBancaria	CE	5	2	Baja	3
211	ListadoSumaYSaldos	CE	16	3	Media	4
212	ListadoBanco	CE	7	1	Baja	3
213	ListadoCuentaBancaria	CE	13	4	Alta	6
214	ListadoChequera	CE	11	3	Media	4
215	ListadoTipoCuentaBancaria	CE	5	1	Baja	3
216	ListadoOrdenPago	CE	12	3	Media	4
217	ListadoMargenCotizacion	CE	6	1	Baja	3
218	ListadoFacturas	CE	19	3	Media	4
219	ListadoFacturasCliente	CE	13	4	Alta	6
220	ListadoOrdenPagoSalario	CE	24	6	Alta	6
221	ListadoViatico	CE	44	15	Alta	6
222	ListadoIngreso	CE	13	5	Alta	6

Puntos de Función no Ajustados						
Tipo de Función	Complejidad Funcional			Total por Complejidades	Total por Tipo de Función	
EE	44	Baja	x 3	132	540	46,79%
	6	Media	x 4	24		
	64	Alta	x 6	384		
SE	0	Baja	x 4	0	0	0,00%
	0	Media	x 5	0		
	0	Alta	x 7	0		
CE	17	Baja	x 3	51	291	25,22%
	18	Media	x 4	72		
	28	Alta	x 6	168		
ALI	34	Baja	x 7	238	323	27,99%
	4	Media	x 10	40		
	3	Alta	x 15	45		
AIE	0	Baja	x 5	0	0	0,00%
	0	Media	x 7	0		
	0	Alta	x 10	0		
Total de Puntos de Función no Ajustados					1154	

Ilustración 20 Variable de Conteo del Módulo de Tesorería (fuente propia).

8. Resultados obtenidos con USC-COCOMO II.2000.4 a partir de los puntos de función sin ajustar de cada modulo.

Project Name: SGPTI		Scale Factor: 3.98		Schedule									
Project Notes		Development Model: Post Architecture											
X	Module Name	Module Size	LABOR Rate (\$/month)	EMF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK	
	BIENES	F:20034	1000.00	0.87	JAVA	42.4	36.8	545.0	36762.70	1.8	1.3	0.0	
	COMPRAS	F:42453	1000.00	0.87	JAVA	89.8	77.9	545.0	77901.92	1.8	2.8	0.0	
X	CONTABILIDAD	F:25970	1000.00	0.87	JAVA	54.9	47.7	545.0	47655.36	1.8	1.7	0.0	
	TESORERIA	F:61162	1000.00	0.87	JAVA	129.4	112.2	545.0	112233.22	1.8	4.0	0.0	
	RRHH	F:63547	1000.00	0.87	JAVA	134.5	116.6	545.0	116609.74	1.8	4.2	0.0	
	PRESUPUESTO	F:12932	1000.00	0.87	JAVA	27.4	23.7	545.0	23730.42	1.8	0.8	0.0	
	DOCUMENTO	F:26500	1000.00	0.87	JAVA	56.1	48.6	545.0	48627.91	1.8	1.7	0.0	
Total Lines of Code: 252598						Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Hours/PM: 160.00						Optimistic	370.8	26.2	681.2	370817.02	1.5	14.1	
						Most Likely	463.5	27.9	545.0	463521.27	1.8	16.6	0.0
						Pessimistic	579.4	29.8	436.0	579401.59	2.3	19.4	

Ilustración 21 Resultado COCOMO II.2000.4 (Boehm B., 2000).

4. Plantilla 3 – Pila de producto.

Estimación ágil: Pila de Producto (Product Backlog)

Elaborado por: Lorena Franco

Estimación por: Pocker Planning

[1 - 3 - 5 - 8 - 13 - 20 - 40]

Ident. (ID) de la Historia	Enunciado de la Historia	Alias	Dimensión / Esfuerzo	Prioridad
1	Ciudad	Ciudad	1	1
2	Pais	Pais	1	1
3	Tipo Documento	Tipo Documento	1	1
4	Barrio	Barrio	1	1
5	Admin Items	Admin Items	5	1
6	Tipo Item	Tipo Item	1	1
7	Moneda	Moneda	1	1
8	Sede	Sede	1	1
9	Persona	Persona	3	1
10	Forma Pago	Forma Pago	1	1
11	Proveedor	Proveedor	3	1
12	Psm	Psm	3	1
13	Psm Detalle	Psm Detalle	3	1
14	Pedido Cotizacion	Pedido Cotizacion	3	1
15	Pedido Cotizacion Proveedor Detalle	Pedido Cotizacion Proveedor Detalle	3	1
16	Pedido Cotizacion Detalle	Pedido Cotizacion Detalle	3	1
17	Recepcion	Recepcion	3	1
18	Recepcion Detalle	Recepcion Detalle	3	1
19	Orden de Servicio	Orden de Servicio	3	1
20	Orden de Servicio Detalle	Orden de Servicio Detalle	3	1
21	Centro de Costo	Centro de Costo	3	1
22	Empresa	Empresa	1	1
23	Tipo Adquisicion	Tipo Adquisicion	1	1
24	Orden Compra	Orden Compra	3	1
25	Orden Compra Detalle	Orden Compra Detalle	3	1
26	Empresa Empleado	Empresa Empleado	3	1
27	Modalidad Compra	Modalidad Compra	1	1
28	Contactos de Proveedor	Contactos de Proveedor	1	1
29	Proveedor Marca	Proveedor Marca	1	1
30	Marca	Marca	1	1
31	admin. Proveedores - Incluir	admin. Proveedores - Incluir	3	1
32	admin. Proveedores - Editar	admin. Proveedores - Editar	3	1
33	admin. Proveedores - Eliminar	admin. Proveedores - Eliminar	3	1
34	admin. Proveedores - Consulta	admin. Proveedores - Consulta	3	1
35	Contactos de proveedores - Incluir	Contactos de proveedores - Incluir	5	1

Estimación ágil: Pila de Producto (Product Backlog)

Elaborado por: Lorena Franco

Estimación por: Pocker Planning

[1 - 3 - 5 - 8 - 13 - 20 - 40]

Ident. (ID) de la Historia	Enunciado de la Historia	Alias	Dimensión / Esfuerzo	Prioridad
36	Contactos de proveedores - Eliminar	Contactos de proveedores - Eliminar	3	1
37	Contactos de proveedores - Consulta	Contactos de proveedores - Consulta	3	1
38	Marcas por proveedores - Incluir	Marcas por proveedores - Incluir	5	1
39	Marcas por proveedores - Eliminar	Marcas por proveedores - Eliminar	3	1
40	Marcas por proveedores - Consulta	Marcas por proveedores - Consulta	3	1
41	admin. Items - Incluir	admin. Items - Incluir	13	1
42	admin. Items - Editar	admin. Items - Editar	8	1
43	admin. Items - Eliminar	admin. Items - Eliminar	3	1
44	admin. Items - Consulta	admin. Items - Consulta	3	1
45	Tipo Items - Incluir	Tipo Items - Incluir	3	1
46	Tipo Items - Editar	Tipo Items - Editar	3	1
47	Tipo Items - Eliminar	Tipo Items - Eliminar	3	1
48	Tipo Items - Consulta	Tipo Items - Consulta	3	1
49	Psm - Incluir	Psm - Incluir	40	1
50	Psm - Autorizar	Psm - Autorizar	13	1
51	Psm - Aprobar	Psm - Aprobar	13	1
52	Psm - Reabrir	Psm - Reabrir	40	1
53	Psm - Editar	Psm - Editar	8	1
54	Psm - Eliminar	Psm - Eliminar	5	1
55	Psm - Listar	Psm - Listar	13	1
56	Psm - Consulta	Psm - Consulta	5	1
57	Pedido de Cotizacion - Generar	Pedido de Cotizacion - Generar	13	1
58	Precios de Proveedores - Consulta	Precios de Proveedores - Consulta	8	1
59	Pedido de Cotizacion - Consulta	Pedido de Cotizacion - Consulta	8	1
60	Pedido de Cotizacion - Adjudicar	Pedido de Cotizacion - Adjudicar	13	1
61	Pedido de Cotizacion - Eliminar	Pedido de Cotizacion - Eliminar	5	1
62	Pedido de Cotizacion - Reabrir	Pedido de Cotizacion - Reabrir	40	1
63	Pedido de Cotizacion - Editar	Pedido de Cotizacion - Editar	13	1
64	Pedido de Cotizacion - Listar	Pedido de Cotizacion - Listar	8	1
65	Orden de Compra - Generar	Orden de Compra - Generar	40	1
66	Orden de Compra - Autorizar	Orden de Compra - Autorizar	13	1
67	Orden de Cotizacion - Facturas	Orden de Cotizacion - Facturas	13	1
68	Orden de Compra - Reabrir	Orden de Compra - Reabrir	13	1
69	Orden de Compra - Editar	Orden de Compra - Editar	13	1
70	Orden de Compra - Eliminar	Orden de Compra - Eliminar	8	1

Estimación ágil: Pila de Producto (Product Backlog)

Elaborado por: Lorena Franco

Estimación por: Pocker Planning

[1 - 3 - 5 - 8 - 13 - 20 - 40]

Ident. (ID) de la Historia	Enunciado de la Historia	Alias	Dimensión / Esfuerzo	Prioridad
71	Orden de Compra - Listar	Orden de Compra - Listar	5	1
72	Orden de Compra - Consulta	Orden de Compra - Consulta	13	1
73	Orden de Servicio - Generar	Orden de Servicio - Generar	40	1
74	Orden de Servicio - Consulta	Orden de Servicio - Consulta	13	1
75	Orden de Servicio - Autorizar	Orden de Servicio - Autorizar	8	1
76	Orden de Servicio - Facturas	Orden de Servicio - Facturas	13	1
77	Orden de Servicio - Reabrir	Orden de Servicio - Reabrir	13	1
78	Orden de Servicio - Editar	Orden de Servicio - Editar	8	1
79	Orden de Servicio - Eliminar	Orden de Servicio - Eliminar	8	1
80	Orden de Servicio - Listar	Orden de Servicio - Listar	8	1
81	Recepcion - Generar	Recepcion - Generar	20	1
82	Recepcion - Reabrir	Recepcion - Reabrir	40	1
83	Recepcion - Editar	Recepcion - Editar	13	1
84	Recepcion - Eliminar	Recepcion - Eliminar	13	1
85	Recepcion - Listar	Recepcion - Listar	13	1
86	Recepcion - Consulta	Recepcion - Consulta	13	1
87	Ciudad - Incluir	Ciudad - Incluir	3	1
88	Ciudad - Editar	Ciudad - Editar	3	1
89	Ciudad - Eliminar	Ciudad - Eliminar	3	1
90	Ciudad - Consulta	Ciudad - Consulta	3	1
91	Pais - Incluir	Pais - Incluir	3	1
92	Pais - Editar	Pais - Editar	3	1
93	Pais - Eliminar	Pais - Eliminar	1	1
94	Pais - Consulta	Pais - Consulta	3	1
95	Tipo Documento - Incluir	Tipo Documento - Incluir	3	1
96	Tipo Documento - Editar	Tipo Documento - Editar	3	1
97	Tipo Documento - Eliminar	Tipo Documento - Eliminar	1	1
98	Tipo Documento - Consulta	Tipo Documento - Consulta	3	1
99	Barrio - Incluir	Barrio - Incluir	3	1
100	Barrio - Editar	Barrio - Editar	3	1
101	Barrio - Eliminar	Barrio - Eliminar	1	1
102	Barrio - Consulta	Barrio - Consulta	3	1
103	Moneda - Incluir	Moneda - Incluir	3	1
104	Moneda - Editar	Moneda - Editar	3	1
105	Moneda - Eliminar	Moneda - Eliminar	1	1
106	Moneda - Consulta	Moneda - Consulta	3	1
107	Sede -Incluir	Sede -Incluir	3	1
108	Sede - Editar	Sede - Editar	3	1
109	Sede - Eliminar	Sede - Eliminar	1	1
110	Sede - Consulta	Sede - Consulta	3	1
111	Persona - Incluir	Persona - Incluir	40	1
112	Persona - Editar	Persona - Editar	20	1
113	Persona - Eliminar	Persona - Eliminar	20	1
114	Persona - Consulta	Persona - Consulta	20	1
115	Tipo de Adquisicion - Incluir	Tipo de Adquisicion - Incluir	3	1

Estimación ágil: Pila de Producto (Product Backlog)

Elaborado por: Lorena Franco

Estimación por: Pocker Planning

[1 - 3 - 5 - 8 - 13 - 20 - 40]

Ident. (ID) de la Historia	Enunciado de la Historia	Alias	Dimensión / Esfuerzo	Prioridad
116	Tipo de Adquisicion - Editar	Tipo de Adquisicion - Editar	3	1
117	Tipo de Adquisicion - Eliminar	Tipo de Adquisicion - Eliminar	1	1
118	Tipo de Adquisicion - Consulta	Tipo de Adquisicion - Consulta	3	1
119	Modalidad Compra - Incluir	Modalidad Compra - Incluir	40	1
120	Modalidad Compra - Editar	Modalidad Compra - Editar	20	1
121	Modalidad Compra - Eliminar	Modalidad Compra - Eliminar	20	1
122	Modalidad Compra - Consulta	Modalidad Compra - Consulta	20	1
123	Forma de Pago - Incluir	Forma de Pago - Incluir	3	1
124	Forma de Pago - Editar	Forma de Pago - Editar	3	1
125	Forma de Pago - Eliminar	Forma de Pago - Eliminar	1	1
126	Forma de Pago - Consulta	Forma de Pago - Consulta	3	1
127	Ciudad	Ciudad	8	1
128	Pais	Pais	8	1
129	Tipo Documento	Tipo Documento	8	1
130	Barrio	Barrio	8	1
131	Admin Items	Admin Items	8	1
132	Moneda	Moneda	8	1
133	Persona	Persona	8	1
134	Forma Pago	Forma Pago	8	1
135	Proveedor	Proveedor	8	1
136	Empresa	Empresa	8	1
137	Tipo Adquisicion	Tipo Adquisicion	8	1
138	Empresa Empleado	Empresa Empleado	8	1
139	Contactos de Proveedor	Contactos de Proveedor	8	1
140	Marca	Marca	8	1
141	Psm	Psm	8	1
142	Psm Detalle	Psm Detalle	8	1
143	Pedido Cotizacion	Pedido Cotizacion	8	1
144	Pedido Cotizacion Proveedor Detalle	Pedido Cotizacion Proveedor Detalle	8	1
145	Pedido Cotizacion Detalle	Pedido Cotizacion Detalle	8	1
146	Recepcion	Recepcion	8	1
147	Recepcion Detalle	Recepcion Detalle	8	1
148	Orden de Servicio	Orden de Servicio	8	1
149	Orden de Servicio Detalle	Orden de Servicio Detalle	8	1
150	Centro de Costo	Centro de Costo	8	1
151	Orden Compra	Orden Compra	8	1
152	Orden Compra Detalle	Orden Compra Detalle	8	1

Total de Puntos de Historias	1213
-------------------------------------	-------------

Ilustración 24 Plantilla 3 - Pila de Producto (fuente propia).

5. Plantilla 4 – Factores de Escala

Estimación ágil: Factores de Escala

Elaborado por: Lorena Franco

SF	Descripción	Nivel	Valor		
Maturity	Madurez de Procesos	Alta	3,12		
PREC	La experiencia de proyectos similares	Extremadamente Alta	0		
FLEX	La flexibilidad requerida en el Sistema	Nominal	2,03		
TEAM	La cohesión del equipo	Extremadamente Alta	0		
RESL	Riesgos del Proyecto y la complejidad de la arquitectura	Baja	1,41		
			6,56		
SF	PREC	FLEX	TEAM	RESL	Maturity
Muy Baja	6,20	0,00	5,48	0,00	7,80
Baja	4,96	1,01	4,38	1,41	6,24
Nominal	3,72	2,03	3,29	2,83	4,68
Alta	2,48	3,04	2,19	4,24	3,12
Muy Alta	1,24	4,05	1,10	5,65	1,56
Extremadamente Alta	0,00	5,07	0,00	7,07	0,00

Ilustración 25 Plantilla 5 - Factores de Escala (fuente propia).

6. Plantilla 5 – Multiplicadores de Esfuerzo.

Estimación ágil: Multiplicadores de Esfuerzo

Elaborado por: Lorena Franco

Factores	Valores						Valor
	Muy Baja VL	Baja L	Nominal N	Alto H	Muy Alto VH	Extra Alto EH	
Relacionado al Producto							
Confiabilidad Requerida	0,82	0,92	1,00	1,10	1,26		H
Medida de Volumen de Datos		0,90	1,00	1,14	1,28		VH
Complejidad del Producto	0,73	0,87	1,00	1,17	1,34	1,74	N
Grado de Reusabilidad		0,95	1,00	1,07	1,15	1,24	L
Documentación Requerida	0,81	0,91	1,00	1,11	1,23		N
Relacionadas a Instalaciones							
Restricción de Ejecución			1,00	1,11	1,29	1,63	N
Capacidad de Almacenamiento			1,00	1,05	1,17	1,46	N
Volatilidad de la Plataforma		0,87	1,00	1,15	1,30		L
Relacionado Al Personal							
Capacidad de los Analistas	1,42	1,19	1,00	0,85	0,71		N
Conocimiento del Negocio	1,22	1,10	1,00	0,88	0,81		H
Capacidad de los Programadores	1,34	1,15	1,00	0,88	0,76		N
Continuidad del Personal	1,29	1,12	1,00	0,90	0,81		N
Experiencia en la Plataforma	1,19	1,09	1,00	0,91	0,85		H
Experiencia en Lenguajes y Herramientas	1,20	1,09	1,00	0,91	0,84		VH
Relacionado al Proyecto							
Uso de Herramienta de Software	1,17	1,09	1,00	0,90	0,78		H
Desarrollos en Sitios Múltiples	1,22	1,09	1,00	0,93	0,86	0,80	N
Desarrollo del Cronograma	1,43	1,14	1,00	1,00	1,00		N
Factor Multiplicador:							0,7045

Ilustración 26 Plantilla 5 - Multiplicadores de Esfuerzo (fuente propia).

7. Plantilla 6 – Calculo de Costos.

Estimación ágil: Cálculos

Elaborado por: Lorena Franco

Descripción	Variables	Formula	Valor
1. Determinar el conteo de puntos de historias definidas en la pila de producto.	TStoryPoint		1.213,0000
2. Determinar la duración de días del sprint.	DSprint		20,0000
3. Determinar la velocidad del equipo.	V		52,0000
4. Determinar el número de personas en el equipo.	N		6,0000
5. Determinar el factor de escala del proyecto.	E	$E = B + 0.01 \cdot \sum_{j=1}^5 SF_j$	1,0085
6. Determinar los multiplicadores de esfuerzo del proyecto.	EM	$\prod_{i=1}^n EM_i$	0,7045
7. Determinar los puntos de historias ideales para el sprint.	StoryPoint	StoryPoint = V * DSprint	1.040,0000
8. Determinar los puntos de historias ajustados para el sprint.	StoryPointAjust	StoryPointAjust = StoryPoint^E $\prod_{i=1}^n EM_i$	777,2657
9. Determinar la cantidad de iteraciones para el proyecto.	I	$I = \left\{ \frac{\text{TStory Point}}{\text{StoryPointAjust}} \right\}$	1,5606
10. Costo promedio diario del equipo	CostDay		200.000
11. Determinar el costo final del desarrollo de software.	CostDev	Cost = ((CostDay * DSprint) * I) * N	37.454.374

Ilustración 27 Plantilla 6 - Calculo de Costos (fuente propia).