

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 3**



*Trabajo de Diploma para optar por el Título de Ingeniero en  
Ciencias Informáticas*

**Tema:** Sistema Integrado de Gestión Estadística (SIGE).  
Rol de planificador.

**Autor:** Dinella Aguilera González.  
**Tutor:** Ing. Janet Rodríguez Febles.

**Ciudad de la Habana  
Junio 2007**

*Declaración de autoría.*

Yo, Dinella Aguilera González, declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 16 días del mes de junio del año 2007.

---

Dinella Aguilera González  
Autor

---

Ing. Janet Rodríguez Febles  
Tutor

*"El plan no elimina el riesgo, sino que intenta eliminarlo o reducirlo"*

*Peter Ferdinand Drucker.*

*Dedicatoria:*

A mi madre, porque sin su apoyo nunca hubiera logrado llegar, por ser perfecta en todo momento, por estar siempre ahí y ser especial.

A mi padre, por haberme enseñado su amor por la literatura y la investigación científica, por hacer de cada discusión una clase mágica.

A mi tía Martha, por enseñarme que la superación diaria es la clave del éxito.

Los quiero mucho, y les debo la mayor parte de lo que hoy soy.

## *Agradecimientos:*

A mi madre, por ayudarme a conseguir nuestro sueño, por ser la mayor fuente de mi inspiración.

A mi padre, por recordarme siempre la importancia de terminar mis estudios universitarios.

A Fidel Castro y su larga visión, sin su ideal esta Universidad no hubiera sido posible.

A Yoxsnel, cuyo amor me incentivó a ser mejor persona cada día.

A Pascual, por haberme mostrado que aunque la ley de gravedad nos atrae hacia abajo, siempre es posible levantarse.

A Janet, por su esfuerzo final.

A los amigos que me ayudaron a sobrellevar estos largos 5 años.

A aquellos profesores que me mostraron su profunda amistad.

A todos aquellos que aportaron un granito de arena en la realización de esta tesis, a los que me animaron a seguir cuando mi intención era parar. A todos los que grité y aun así siguieron a mi lado, cuando el stress me tentaba a abandonar todo por lo que había luchado. A todos esos duendes pequeños un millón de gracias. Mi gratitud siempre estará con Uds.

## RESUMEN

¿Qué es la planificación? ¿Qué importancia tiene y qué tareas debe realizar el planificador de un proyecto de software? Cuándo se debe realizar la estimación del mismo y por qué es recomendable hacerlo por más de un método, son algunas de las cosas que se tratan en el siguiente trabajo.

La ejecución de un proyecto de software de cualquier envergadura lleva consigo una serie de cálculos que permiten conocer de antemano la factibilidad de su ejecución, el costo total de recursos que se deben usar, así como el tiempo de duración del mismo, lo que permite gestionar toda una serie de elementos necesarios para la culminación exitosa del proyecto.

¿Qué importancia tiene el uso de una herramienta que permita controlar y dar seguimiento a las tareas planificadas y la definición de algunos mecanismos para gestionar, controlar y seguir la información que debe ser planificada?

¿Qué problemas presenta la planificación de proyectos en la UCI?

Este trabajo propone la planificación del proyecto: Sistema Integrado de Gestión Estadística siguiendo los pasos correctos para una planificación exitosa.

## **ABSTRACT**

What is planning? What is the importance of the planning and what tasks must realize the person on charge of planning the software project? When he must estimate the duration of the project and why is advisable estimate with 2 or more methods? These are some of the things that this work is about.

The execution of any software project has with it a lot of calculus that allows you to know de feasibility of the execution, the total cost of resources and the time of execution; all this allows managing a lot of elements necessary for the successful ending of the project.

What is the importance of the use of a tool that allows following and controlling the tasks that are planned and the definition of some mechanism to manage, control and follow the information.

What problems present the software project planning on the University?

This work suggests the planning of the project: Integrated System of Statistic Manage follows the steps of a successfully planning.

# ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	8
1.1 Introducción.....	8
1.2 Tendencias actuales de la planificación.....	8
1.3 Cuba como Industria del Software.....	10
1.4 Sistemas estadísticos usados a nivel mundial.....	11
1.5 Métodos de planificación.....	14
1.6 Métodos de estimación.....	15
1.7 Herramientas para la planificación de proyectos de software existentes.....	19
1.8 Metodología utilizada.....	27
1.9 Gestión de riesgos.....	28
1.10 Conclusiones del Capítulo 1:.....	35
CAPÍTULO 2 DESARROLLO DE LA PLANIFICACIÓN.....	36
2.1 Introducción.....	36
2.2 Estimación del Proyecto.....	36
2.3 Recursos usados en el proyecto SIGE.....	50
2.4 Organización de la planificación:.....	55
2.5 ¿Qué mecanismos de captura de información se usaron en el proyecto ONE?.....	56
2.6 Mecanismos de seguimiento de la información:.....	60
2.7 Mecanismos de control de la información:.....	60
2.8 Plan de fases e iteraciones.....	62



2.9 Gestión de Riesgos .....	65
2.10 Conclusiones del Capítulo. ....	72
ANÁLISIS DE RESULTADOS .....	73
3.1 Introducción.....	73
3.2 Análisis de Resultados .....	73
3.3 Estimación en el proyecto SIGE .....	76
3.4 Mecanismos de captura, seguimiento y control de la información. ....	77
3.5 Organización de la planificación.....	79
3.6 Planificación .....	80
3.7 Gestión de riesgos.....	81
3.8 Conclusiones del capítulo.....	82
CONCLUSIONES GENERALES.....	83
RECOMENDACIONES.....	84
BIBLIOGRAFIA CONSULTADA .....	85
BIBLIOGRAFIA REFERENCIADA .....	86
GLOSARIO DE TÉRMINOS.....	87

## INTRODUCCIÓN

Cuando César se dispuso a conquistar las Galias, emprendió la guerra contra varios pueblos, derrotando a los helvecios (en la actual Suiza), a la Confederación Belga, entre otros. El 26 de agosto del 55 a.C. intentó la invasión de Britania , y en el 52 a.C. derrotó a una unión de las tribus galas dirigida por Vercingétorix en la batalla de Alesia.

Según muchos historiadores, fue la gesta militar más grandiosa desde los tiempos de Alejandro Magno.

*¿Que tiene que ver Gayo Julio César con la planificación de un proyecto de Software?*

La Respuesta la tienen todas sus victorias, César fue un general intrépido con un don para la estrategia y la planificación. Cada una de sus batallas tenía un alto nivel de detalle en cuanto a cada cosa que se debía hacer. Era meticuloso a la hora de organizar lo mas mínimo, formulaba las estrategias, analizaba diversos factores, estimaba la cantidad de hombres y hacia una combinación de estas características para determinar las tácticas y pasos a seguir, con él no podían haber contratiempos, pues eso significaba la derrota en la batalla. Nada podía fallar y todo debía suceder en el momento indicado. Lo que conduce a pensar que César hubiera sido en estos tiempos, un gran planificador de software, y que sin dudas, su prestigio hubiera crecido. Se puede decir que la planificación de sus batallas significaba un alto por ciento en sus victorias. La planificación de un proyecto de software define en gran medida, la culminación triunfante del mismo.

Roger S. Pressman define planificación como “una actividad que distribuye el esfuerzo estimado a lo largo de la duración prevista del proyecto, asignando el esfuerzo a las tareas específicas de la Ingeniería de Software” [1]

Otros autores definen planificación como “la etapa del proyecto de software en que se comienza a definir y estudiar las metas y los objetivos del mismo, así como los métodos que se utilizan para obtener los factores que dan como resultado la factibilidad de desarrollo del proyecto”. [2]

La Planificación consiste en realizar una serie de actividades que permitan: *DECIDIR* y *HACER*.

*DECIDIR* consiste en lo que generalmente se llama planificar

*HACER* es lo que se denomina ejecutar o implementar

Pero hay que considerarlas como inseparables, en el sentido de que no existe ninguna decisión hasta que no se ejecuta.

*DECIDIR + HACER = Plan.*

Se puede decir entonces que la Planificación define el marco de referencia dentro del cual se desarrolla una organización. [2]

La Planificación de un proyecto es elemental a la hora de definir ciertos marcos como el costo, la cantidad de recursos (Personas, Componentes de Software reutilizables y herramientas de Hardware y Software), así como el tiempo de entrega al cliente.

En resumen: Planificación es el fenómeno de analizar por anticipado el problema a desarrollar, definir los pasos y estrategias a seguir para llevarlo a cabo, calcular la factibilidad de lo que se va a planificar y trazar metas para poder realizarlo en un plazo determinado.

En la época en que el software se hacía de forma artesanal no se hacía planificación del mismo porque no existía una metodología específica para llevar a cabo la ingeniería de software así como una planificación adecuada que se acercara lo más posible a la realidad del tiempo de duración del desarrollo del producto. Sucedió entonces que muchos programadores se enfrentaban a problemáticas que al principio parecían sencillas, y por ende se sentían capaces de culminar el producto en tiempos relativamente cortos, hasta que comenzaban a aparecer pequeños obstáculos que pronto se convertían en su peor pesadilla. Estas historias no tenían finales felices como en los cuentos de hadas, generalmente terminaban con grandes insatisfacciones, fechas de entrega no cumplidas e inconformidades por todos lados.

¿De qué forma evitar el problema de poder entregar a tiempo el proyecto de software?



La solución es muy sencilla y depende de la adecuada planificación que se le haga al proyecto de software al cual se está involucrado.

La planificación consiste en realizar una serie de actividades que permitan ver con claridad QUÉ se va a hacer, CÓMO se va hacer, CUÁNTO se va a tardar y a costar y A QUIÉNES se necesita para que lo realicen con la calidad que se necesita.

Precisamente esto es lo que se va a realizar para lograr que el proyecto SIGE sea entregado en la fecha acordada y con la calidad que se exige.

El proyecto SIGE es un proyecto de la facultad 3 que pertenece a la Línea de Proyectos Nacionales. Surge por la necesidad de actualizar el software utilizado hasta el año 2006 en la Oficina Nacional de Estadística (ONE) llamado MicroSet, que producto al desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC) y a las novedades introducidas en otros Sistemas de Gestión Estadística utilizados en otros países, comenzaba a ser obsoleto. Tenía problemas al manejar códigos de más de 5 dígitos, así como la creación de tablas más complicadas, etc.

El cliente de este proyecto es la Oficina Nacional de Estadísticas (ONE) que es el órgano rector de las estadísticas en Cuba y como tal es el encargado de organizar y dirigir este sistema, estableciendo para ello las normas requeridas para su organización y operación, apoyándose en los resultados de las coordinaciones y de las consultas que a tales efectos debe realizar con los órganos, organismos e instituciones estatales y otras instituciones.

### Situación Problemática.

En la Universidad de Ciencias Informáticas (UCI) no se cuenta con la documentación necesaria que permita guiar el proceso de planificación en un orden lógico. Generalmente se incurre en atrasos dentro del proyecto y en la mayoría de los casos concluyen con insatisfacciones con el cliente e incumplimientos con el tiempo de entrega acordado, además de molestias y conflictos que pueden surgir dentro de los equipos de trabajo.

¿Cómo lograr una adecuada planificación del proyecto SIGE que logre la entrega del sistema en el tiempo establecido y que además pueda generar una documentación de las actividades y responsabilidades que desarrolla el planificador del proyecto?

El Objeto de estudio.

La planificación del proyecto SIGE.

Campo de acción

Rol de Planificación en el proyecto SIGE.

Objetivo general:

Lograr una adecuada planificación del proyecto SIGE y generar documentación para la planificación de futuros proyectos con características similares.

Objetivos específicos:

- Estimar de cada módulo del proyecto SIGE.
- Estimar del tiempo de vida total del producto.
- Definir las tareas que debe realizar el planificador.
- Definir mecanismos que permitan la captura, el seguimiento y el control de la información.
- Establecer un plan de contingencia para los riesgos críticos más importantes.

Hipótesis:

Si se logra una adecuada planificación de proyecto, el software podrá ser entregado en tiempo y generará una documentación que podrá ser utilizada para planificar proyectos similares.

### Marco metodológico

#### Unidad de estudio:

Proyecto SIGE

#### Estrategias de investigación

##### Estrategia exploratoria:

Este tipo de investigación cuenta con 2 fuentes principales: Las fichas bibliográficas y la obtención de datos prácticos. La primera consiste en la revisión de la literatura existente y la segunda su propio nombre lo indica. Para la realización de este trabajo es necesario consultar diversas fuentes bibliográficas, sobre todo en el estudio de las tendencias actuales de la planificación. La obtención de datos prácticos es vital para las planificaciones temporales y su refinamiento.

#### Métodos científicos de investigación

##### Métodos teóricos:

Estos métodos permiten mediante el análisis, hacer una división mental del sistema, en múltiples componentes para facilitar su estudio. Su aplicación es muy conveniente pues SIGE utiliza la metodología Rational Unified Procces (RUP) que propone que los proyectos que son muy complejos o grandes se dividan en pequeños módulos para que sea más fácil su desarrollo.

Métodos empíricos:

Estos métodos permiten la observación, la medición y la experimentación. El trabajo del planificador depende en gran medida, de los fenómenos que observa, la medición de estos y de los resultados experimentados, que se convierten posteriormente en datos históricos, listos para ser usados en nuevas estimaciones.

Método histórico – lógico:

Este método permite el estudio del fenómeno desde el punto de vista histórico, siguiendo una secuencia lógica. El estudio de la planificación en este trabajo se hará con una orden que comienza en el surgimiento del término y culmina analizando las tendencias que tiene hoy día.

Método de la observación:Observación externa – Incluida:

En este método el observador forma parte del proceso observado y participa en él durante el tiempo de observación. El planificador forma parte del equipo de trabajo, igual que él planifica el trabajo de otros, también debe planificar el suyo. Sus datos históricos, también son importantes.

Métodos particulares:

Son métodos específicos basados en las características del propio objeto de investigación.

*1. La entrevista:*

El uso de la entrevista hizo posible esclarecer diferentes aspectos relacionados con las responsabilidades del planificador dentro de los proyectos productivos, precisando que el término planificación de un proyecto de software, no es precisamente familiar, a esto se le sumó la necesidad de conocer

experiencias anteriores y datos históricos de otras personas que han participado en experiencias similares.

*2. La encuesta:*

La encuesta posibilita conocer la opinión de personas relacionadas con el proceso del software, muy importante si se quieren hacer comparaciones o medir el avance de la planificación del proyecto SIGE con respecto a otros proyectos productivos.

*3. Métodos de medición:*

Estos métodos como su nombre lo indica permiten hacer “mediciones” de diversas índoles, en este caso es importante para medir el proceso de desarrollo del Software. Obtener resultados numéricos con los que se pueda calcular el tiempo de vida del proyecto, la cantidad de personas necesarias, etc.



1

## *Capítulo* *Fundamentación teórica.*

### **1.1 Introducción**

Este capítulo abordará las tendencias actuales del término planificación y su importancia en empresas de software tanto nacional como internacionalmente. Se fundamentará la elección de la metodología RUP para llevar a cabo la planificación. Se hará un estudio de diversos software estadísticos, teniendo en cuenta sus principales características, así como los métodos de planificación y estimación que se adecuen más a las particularidades del proyecto SIGE. Un estudio comparativo de las herramientas de planificación usadas frecuentemente en el mundo, proporcionará la herramienta ideal para planificar este proyecto. Además, se mostrará la necesidad de estudiar y combatir detalladamente los riesgos, apoyando de esta forma el proceso de calidad del software en cuestión.

### **1.2 Tendencias actuales de la planificación**

Igor Ansoff identifica la aparición del término Planificación con vistas a lograr una estrategia de desarrollo en la década de 1960 y la asocia a los cambios en los impulsos y capacidades estratégicas. Para otros autores, la Planificación como proceso para ayudar al control de las actividades emerge formalmente en los años setenta, como resultados natural de la evolución del concepto de Planificación.

Actualmente, las grandes compañías internacionales de software utilizan la planificación de sus proyectos como la vía para medir la factibilidad de los mismos. Así, un proyecto que demore una gran

cantidad de años y además tenga un costo de recursos de hardware, software y humanos elevado, una vez que el equipo de planificación determina que su desarrollo no es económicamente posible para la empresa, simplemente es desechado. Solo un desequilibrado se enfrenta a una empresa de gran envergadura con la seguridad de que solo va a obtener pérdidas.

Muchas compañías se jactan de poder lanzar sus productos en un tiempo récord y comienzan a hacer campañas publicitarias, con comerciales en la televisión y carteleras, cuando apenas el producto comienza su desarrollo. ¿Tienen finales felices estas historias? No siempre, La compañía Lotus promocionó el producto Lotus 123 (Un programa de hojas de cálculo) que no pudo terminar en el tiempo para el que dijo estaría listo, para cuando lograron terminarlo, Microsoft tenía una ventaja de 16 meses desarrollando Excel, pero además su producto era obsoleto en todo el sentido de la palabra.

He ahí la importancia de planificar el trabajo y ser realista a la hora de programar las fechas de entrega, a pesar de la presión que puede ejercer el cliente, es necesario enfocar el tema de la fecha límite de tal modo que siempre quede algo de holgura a los desarrolladores.

Si embargo hay cierta tendencia por parte de los equipos de trabajo a no realizar los planes de trabajo ¿Por qué?

1. La mayoría piensa que no vale la pena, que es algo inútil y además una gran pérdida de tiempo que al final no conduce a nada.
2. Creen que el plan inevitablemente es erróneo, por tanto, si es erróneo, ¿Para qué hacerlo?

La planificación esta regida por 5 principios fundamentales:

**Principio unitario:** Se debe realizar un plan para cada tarea (esto incluye tiempo de duración, encargado de realizarla, dependencias, etc.) y el conjunto de estos planes debe estar organizado e integrado de forma tal que se cree un plan general para todas las actividades.

**Principio de precisión:** Los planes deben ser lo más precisos posible en cada una de sus tareas. Es necesario especificar los más mínimos detalles en aras de ganar en organización.

**Principio de flexibilidad:** El plan creado debe tener cierta holgura, para que puedan ser considerados cualquier cambio que pueda surgir, algún atraso imprevisto y por ende muy riesgoso o cualquier otra situación.

**Principio de compromiso:** El plan debe ajustarse a una fecha final. Este tiempo debe coincidir con el tiempo ajustado o acordado con el cliente.

**Principio de la factibilidad:** Quizás uno de los más importantes pues la medición de la perspectiva del proyecto es vital para su ejecución. Nadie quiere pertenecer al proyecto incumplidor o al que arrojó gastos enormes y cero ganancias. [3]

### 1.3 Cuba como Industria del Software

En Cuba, hace muy poco tiempo se comenzó a elevar la cultura informática con el propósito de crear una industria de software similar a la de otros países como India, Puerto Rico, etc. Como sueño naciente que aún es, no posee todavía la madurez en los procesos de planificación vinculados a la producción. Esto es una meta a lograr, con la certera intención de incrementar ganancias, disminuyendo los costos y cumpliendo con los planes establecidos.

En la mayoría de las empresas relacionadas al desarrollo de software, se acude a la combinación de varias técnicas de estimación con la intención de lograr una planificación lo más real posible.

En la UCI no existe un nivel profundo de conocimiento en cuanto a la planificación, este término en la facultad 3 se comenzó a utilizar luego de la división de los proyectos de software por roles, en el tercer curso académico (2004 -2005) con la llegada del proyecto Registros y Notarías, y posteriormente se aplicó a todos los proyectos productivos de la facultad.

¿Qué resultados se obtuvieron con esto?

**Experiencia:** Toda la facultad comenzó a ganar práctica en el manejo de roles y sobre todo en el grado de complejidad que posee cada uno de ellos y que importancia juegan dentro del proyecto.

**Organización:** Los proyectos comenzaron a ser funcionales, se saltó a una etapa donde la Planificación, la Calidad, la Gestión de la Configuración y Cambios, entre otros, se entrelazan para lograr un proyecto de software capaz de marchar con eficiencia en el tiempo planificado y por tanto, sin incumplimientos.

En la Figura 1, se representa la forma organizativa de los proyectos productivos en la facultad, vinculada a la asignación de responsabilidades.

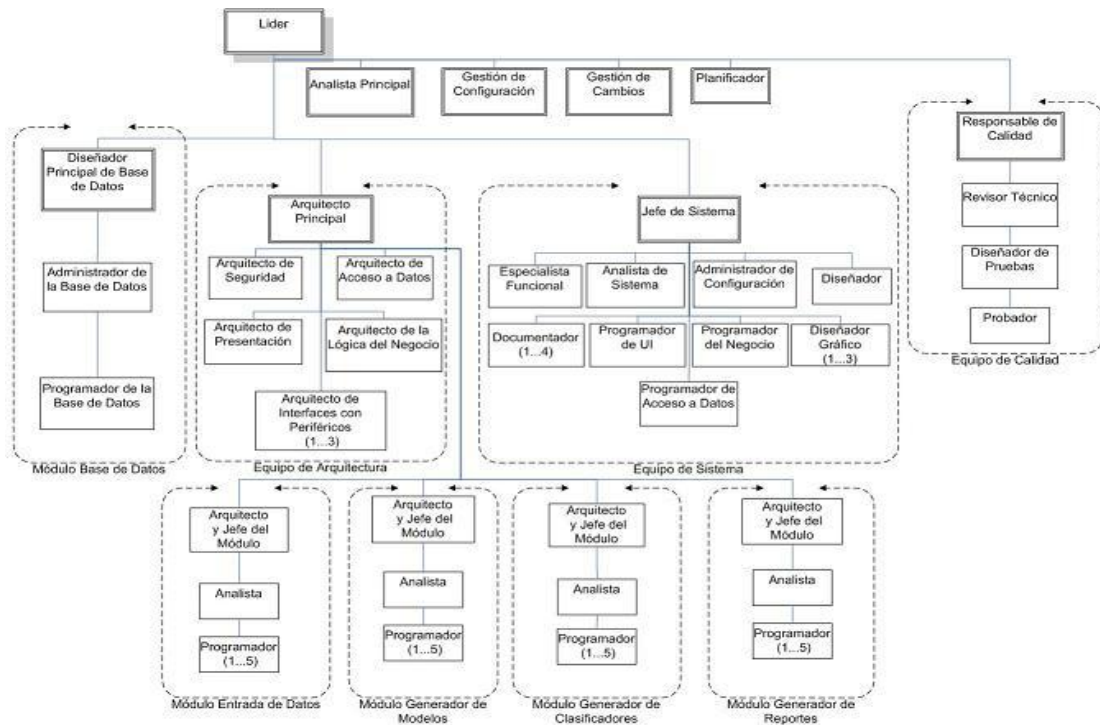


Figura 1. Organización de un proyecto dividido en roles.

### 1.4 Sistemas estadísticos usados a nivel mundial

La adquisición de software estadístico puede ser un problema si no se cuenta con suficientes recursos económicos, pues son excesivamente caros, pudiendo ser el software libre una alternativa. A nivel internacional existen varios sistemas que permiten análisis estadístico o la gestión de datos con un alto nivel de prestigio entre los que se puede mencionar:

### **1.4.1 EpilInfo**

Es un programa gratuito muy extendido tanto para la gestión y análisis de datos provenientes de cuestionarios o encuestas, como para el mantenimiento de sistemas de información de vigilancia epidemiológica. Es quizá el sistema estadístico más popular entre los investigadores de la Salud. No posee métodos multivariados de uso habitual en la investigación mediante encuestas. Esto representa una limitación importante para los usuarios o analistas avanzados, quienes pueden requerir este tipo de técnicas. [4]

### **1.4.2 ViSta**

Es un programa creado por el profesor Forrest W. Young en la Universidad de Carolina del Norte. Posee versiones en diferentes idiomas y para diferentes plataformas (Macintosh, Windows y Unix). Se trata de un programa diseñado originalmente como herramienta para la investigación y docencia, aunque actualmente posee características que lo definen como un Sistema Estadístico de propósitos múltiples. Proporciona acceso al código fuente y herramientas de programación para que los usuarios avanzados puedan expandir las capacidades del programa. [4]

### **1.4.3 WinIdams**

Es un programa creado por la *United Nations Educational, Scientific and Cultural Organization* (UNESCO) en cooperación con varios expertos de diferentes países. WinIDAMS es la versión de "*Internationally Developed Data Analysis and Management Software Package (IDAMS)*" para Windows. El programa está disponible en inglés, francés y español, y corre en Windows 95, 98, NT, 2000 y XP. Este sistema incluye una serie de métodos de gestión y técnicas de análisis que se ejecutan de forma interactiva, mediante una interfaz gráfica de usuario (IGU). Además, incluye una gran variedad de técnicas estadísticas avanzadas que se ejecutan por medio de subprogramas sin interfaz gráfica. [4]

#### **1.4.4 OpenStat (OS3)**

OS3 es un sistema desarrollado totalmente por Bill Miller en la Universidad Estatal de Iowa, Estados Unidos. El sistema funciona en Windows (95/98/NT) y Linux, y solo está disponible en idioma inglés. Su origen era para el estudio de las Ciencias del Comportamiento y con fines educativos, motivo por el cual resulta altamente interactivo y simple de utilizar. El programa cubre una extensa familia de técnicas estadísticas, diferenciándose de otros programas de esta categoría por sus recursos de análisis para cuestionarios, escalas, etc. [4]

#### **1.4.5 Sistema estadístico más usado en Cuba**

Hasta el año 2006 el software estadístico usado en Cuba era el MicroSet, diseñado y programado por los licenciados Tristán Gude Hernández y Luis Alberto Vázquez Piquera. MicroSet surge con el objetivo fundamental de lograr un sistema general e integral para el procesamiento de datos y las ediciones de tablas, y con ello facilitar y mejorar el procesamiento de la información de las estadísticas económicas de la Oficina Nacional de Estadísticas.

La primera versión nació en 1984, y fue sustituida en septiembre de 1987 por el MICROSET II, resultado de un nuevo diseño del sistema y que obtuvo amplia difusión debido a su eficiencia y sencillez, teniendo en cuenta la introducción, en 1986, de las microcomputadoras en toda la red de Centros de Cálculo del Comité Estatal de Estadísticas.

La versión 2.0 del MICROSET II fue programada completamente en el lenguaje de programación Borland Pascal versión 7.0 y constituye un sistema multiusuario. Sin embargo quedó obsoleta debido al camino apresurado de las TIC y a los cambios en términos informáticos que están ocurriendo en Cuba en estos momentos. La necesidad de un sistema que permitiera el manejo de códigos de más de 5 dígitos, así como el manejo de tablas más complicadas, etc., hacen que surja en el calor de la Batalla de Ideas el proyecto SIGE, capaz de mejorar en un 100 % las funcionalidades de MICROSET así como de satisfacer otras necesidades de los clientes.

## **1.5 Métodos de planificación**

### **1.5.1 Diagrama de Gantt**

Una de las formas más extendidas y quizá útiles en la planificación de software es el diagrama de Gantt porque es uno de los diagramas que permite de una forma muy simple, darle seguimiento a un proyecto e incluso gestionar y controlar los atrasos. Es una representación gráfica de 2 ejes, que simulan una gran letra L, en el eje horizontal se representa el tiempo y en el eje vertical se representan las tareas a realizar. Al estar las tareas tan fácilmente distribuidas es muy simple observar su comportamiento en un tiempo definido, resalta el camino crítico, o sea, aquellas tareas que deciden si el proyecto se podría completar dentro del periodo de tiempo establecido o no. [5]

### **1.5.2 Diagrama de Pert**

“*Program Evaluation and Review Technique*” (PERT), es un modelo para la administración y gestión de proyectos que data del año 1958. Increíblemente este diagrama fue inventado por el Departamento de Defensa de los Estados Unidos, que en ese entonces se enfrascaba en un proyecto para el lanzamiento de un misil desde un submarino. Ha trascendido desde entonces por la comodidad que provee para analizar el tiempo de duración de un proyecto. Utiliza las redes PERT, que son diagramas de líneas de tiempo interconectadas. Además, está diseñado para proyectos grandes y complejos.

### **1.5.3 Critical Path Method (CPM)**

El método de la ruta crítica fue inventado por la Corporación DuPont casi al mismo tiempo en que se ideaba el Diagrama de Pert. El camino crítico de un proyecto, no es más que la secuencia de actividades con las que se puede llegar a la meta final y dan lugar a un tiempo máximo de duración. Gracias a esta técnica se puede determinar el camino más corto.

Como precondiciones tiene que es necesario conocer la duración de las actividades. La ruta crítica determina la duración del proyecto, cualquier retraso en un elemento de la ruta crítica afecta

directamente la fecha planificada para finalizar el proyecto (no hay holgura en la ruta crítica).

#### **1.5.4 Precedence Diagram Method (PDM)**

El Método del Diagrama de Precedencias es método de planificación de proyecto enfocada en las precedencias de las actividades. No muestra su duración, solo la dependencia entre actividades. Muy importante para usarlo a la hora de representar restricciones de actividades y enfocar las actividades críticas a realizar para poder hacer otras.

#### **1.5.5 Arrow Diagramming Method (ADM)**

El Método del Diagrama de Flechas es una técnica de planificación de proyectos donde las flechas indican las dependencias entre los nodos que representan las actividades.

Entre sus inconvenientes está que no es tan explicativo como otros diagramas y no se le puede asignar cantidades de tiempo a las tareas.

### **1.6 Métodos de estimación**

Una de las principales actividades que debe realizar el planificador es la Estimación del Proyecto, prestándole especial atención a la estimación de los recursos humanos y materiales que se van a necesitar para llevar a cabo de forma exitosa el proyecto. Esta actividad requiere cierto grado de preparación y de sentido común. Es necesario ser capaces de identificar posibles estimaciones irracionales y poder enmendarlas a tiempo. Se exige experiencia para este tipo de valoraciones, sin embargo es conocido que la experiencia la da la práctica, así que poco a poco es posible llegar a ser un buen planificador capaz de detectar a tiempo posibles errores, de estimar coherentemente la cantidad de personas que son realmente necesarias para llevar a cabo la empresa y que materiales verdaderamente se necesitan para terminarla.



### 1.6.1 COCOMO II y Puntos de Función.

Barry Bohem introduce una jerarquía de modelos de estimación de Software con el nombre de COCOMO (*Constructive Cost Model*). La jerarquía de modelos de Bohem esta constituida por los siguientes:

**Modelo I.** El Modelo COCOMO básico calcula el esfuerzo y el costo del desarrollo de Software en función del tamaño del programa, expresado en las líneas estimadas.

**Modelo II.** El Modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de conductores de costos que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

**Modelo III.** El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costos en cada caso (análisis, diseño, etc.) del proceso de ingeniería de Software. [6]

COCOMO II es la aplicación de ecuaciones matemáticas sobre los puntos de función sin ajustar o la cantidad de líneas de código estimados para un proyecto. Está orientado a la magnitud del producto final, estas ecuaciones son ponderadas por una serie de factores de costo que influyen en el cálculo del esfuerzo que se necesita para llevar a cabo un proyecto de software.

Este método de estimación esta basado en 3 submodelos:

- *Modelo de Composición de Aplicaciones:* Indicado para proyectos construidos con herramientas modernas de construcción de interfaces gráficos para usuario.
- *Modelo Temprano:* es un modelo de alto nivel y ayuda a definir las estrategias a seguir.
- *Modelo Post Arquitectura:* es un modelo detallado donde es posible establecer costos bien definidos.

Los más usados son el Temprano y luego el Post arquitectura, la diferencia entre ambos modelos puede ser realmente abrumadora, lo que la provoca es el número de factores de esfuerzo y el área en la que cada uno de ellos influye.

El diseño temprano estudia las arquitecturas alternativas del sistema y los conceptos de operación. Como su nombre lo indica, en este punto del desarrollo del proyecto aún no se tiene suficiente información del mismo para realizar una estimación mucho más detallada. Inicialmente lo más importante es extraer:

Entradas Externas: EI

Salidas Externas: EO

Ficheros lógicos internos: ILF

Ficheros de interfaz externas: ELF

Consultas (peticiones) externas: EQ

El modelo propone la utilización de los puntos de función como medida de tamaño y un conjunto de 7 factores de esfuerzo.

En cambio el diseño post arquitectura contempla el desarrollo y el mantenimiento del software. Esta estimación es mucho mas fiable y segura, si ya el proyecto tiene una arquitectura definida que ha sido validada y establecida como base para el desarrollo.

El modelo utiliza las Líneas de código fuente y los Puntos de Función como medidores del tamaño, modificadores para indicar el grado de reutilización y descarte del software, un conjunto de 17 estimadores de costo, y un conjunto de 5 factores que afectan de manera exponencial el esfuerzo del proyecto.

Para la estimación inicial del proyecto SIGE se propone el Modelo temprano. ¿Por qué? Aún no se tienen elementos suficientes para desarrollar el diseño post arquitectura pues el proyecto se encuentra iniciando la segunda iteración de la Fase de Elaboración. En el libro *El Proceso Unificado de desarrollo de software* se plantea que este modelo es recomendable cuando se ha terminado la Fase de Elaboración.

Los factores de escala son:

1. **PREC**: variable de precedencia u orden secuencial del desarrollo.
2. **FLEX**: variable de flexibilidad del desarrollo.
3. **RSEL**: indica la fortaleza de la arquitectura y métodos de estimación y reducción de riesgos.
4. **TEAM**: esta variable refleja la cohesión y madurez del equipo de trabajo.
5. **PMAT**: relaciona el proceso de madurez del software.

Los multiplicadores de escala son:

1. **PERS**: Capacidad del personal. Está dado por la suma o la combinación porcentual de los multiplicadores **ACAP, PCAP y PCON**.
2. **RCPX**: Complejidad del producto. Está dado por la combinación de los multiplicadores **RELY, DATA, CPLX y DOCU**.
3. **RUSE**: Reusabilidad. Está dado por el mismo multiplicador RUSE del modelo Post arquitectura.
4. **PDIF**: Dificultad de la plataforma. Está dado por la combinación de los multiplicadores **TIME, STOR y PVOL**.
5. **PREX**: Experiencia del personal. Está dado por la combinación de los multiplicadores **AEXP, PEXP y LTEX**.
6. **SCED**: Calendario. Está dado por el mismo multiplicador SCED del modelo Post arquitectura.
7. **FCIL**: Facilidades. Está dado por la combinación de los multiplicadores **TOOL y SITE**. [6]

### ***Pasos para la estimación por COCOMO II y Puntos de Función***

Obtener los puntos de función. (UFP).

Identificación de las características.

Clasificación.

Ponderación aplicando pesos.

Estimar la cantidad de instrucciones fuente. (SLOC).

Utilizar tabla de lenguajes.

Aplicar las fórmulas de Bohem.

Obtener esfuerzo (PM) y tiempo (TDEV).

Planificar las actividades del proyecto.

### **1.6.2 Puntos de Casos de Usos**

La técnica de estimación por casos de usos permite documentar los requerimientos del software, en cambio la estimación por Puntos de Función permite estimar el tamaño del software, a partir de sus requerimientos.

Cuando comienza el ciclo de vida de un software, los trabajadores del proyecto identifican los actores y casos de uso del sistema, que son documentados en planillas con el fin de mostrar al cliente la forma en que se ideó el software y además, para que los equipos de desarrollo tengan una guía de lo que se va a programar. Si se aplica la técnica de Análisis de Puntos de Casos de Uso al inicio del desarrollo del software se obtiene una estimación quizás alejada de la realidad debido a que en ese momento se tiene muy poca información y es un momento además en que ocurren diversos cambios en el análisis debido a que los clientes modifican, agregan o eliminan requerimientos, por lo que esta información no es estática, al contrario es muy variable, sin embargo permitirá obtener una representación de aquellos elementos necesarios y del esfuerzo aproximado que se va a realizar para llevar a cabo el proyecto de software.

A medida que se puntualicen los casos de uso esta información podrá ser aun más detallada y es posible estimar nuevamente para obtener una estimación mucho más precisa que la que se hizo inicialmente.

La estimación por puntos de casos de uso es un método de estimación del tiempo de desarrollo de un proyecto, mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de los mismos. Es muy importante conocer que la cantidad de transacciones se determina a partir de la descripción textual de casos de uso. Entre más detallada esté la descripción textual, más transacciones se pueden encontrar y la estimación será más exacta. [7]

### **1.7 Herramientas para la planificación de proyectos de software existentes**

David West en su artículo: "*Planning a Project with the Rational Unified Process*" plantea que construir y planificar un software no es como construir un Puente, donde todos pueden observar el avance del proyecto. Si se quiere conocer el avance de un proyecto de software, es necesario crear ciertos mecanismos de planificación que le permitan tanto al líder del proyecto, al planificador u otras personas

interesadas, estar actualizados adecuadamente con el desarrollo del mismo. Para esto es muy recomendable auxiliarse de herramientas diversas de planificación. [8]

Existen muchas herramientas que contribuyen a agilizar el trabajo del planificador, además de otras ventajas que las hacen compañeras inseparables del mismo. Una herramienta que permita analizar los riesgos por atrasos, además del seguimiento de las tareas (dígase la dependencia de unas tareas con otras y el nivel de realización de cada una), es el instrumento ideal para realizar el trabajo de la planificación del software.

Quizá uno de los mayores problemas a la hora de presupuestar un proyecto, es saber cuántas horas va a llevar, cuánto personal especializado vas a necesitar y qué recursos materiales vas a usar (muy importante si tienes que compartirlos entre varios proyectos como a veces sucede en la UCI). Cuando se planifica se está pisando en un terreno de arenas movedizas, pues se está estimando la cantidad de recursos y el tiempo necesario para entregar un producto en la fecha pactada, o sea **NO SE ASEGURA** que en realidad sea así. Se puede listar incluso una serie de riesgos y en la realidad presentarse situaciones que no se habían imaginado. Este tipo de cosas son hartito conocidas. Esta es una de las razones que hace importante el uso de herramientas de planificación, para ayudar a disminuir esta serie de factores que pueden incurrir en fallas de planificación e inevitablemente en atrasos.

Por conveniencias a la hora de escoger la herramienta con la que se va a llevar a cabo la planificación del Proyecto SIGE se ha querido clasificar las mismas en Herramientas del tipo: Software Propietario y Software No Propietario.

¿Por qué?

El bloqueo impuesto a Cuba por los Estados Unidos le ha obligado a prescindir de diversos software que aunque ventajosos en su uso, se le hace difícil adquirirlos por cuestiones de costos, licencia, legalidad, etc. En cambio el uso de software libre, facilita el proceso legal de su uso, además de las ventajas de poseer y modificar su código.

## **1.7.1 Software Propietario. Herramientas para la planificación:**

### **1.7.1.1 B-kin Project Monitor**

B-kin Project Monitor es una herramienta básica para obtener visibilidad sobre el avance del proyecto y su impacto sobre costos y recursos. Genera automáticamente informes sobre los recursos, tanto humanos como materiales asignados. Con él se puede dar seguimiento a los proyectos. Este software puede utilizarse de forma online (ofrece a través de la Web una visión permanentemente actualizada de la situación de los procesos de negocio más dinámicos.)

Una característica interesante de este software es que exporta datos a Microsoft Project y Microsoft Excel para adaptar informes a las necesidades del cliente, o para la integración con otras fuentes de datos. Garantiza la seguridad y confidencialidad de los clientes online. Desafía uno de los retos más importantes de los líderes de proyectos de software y es la organización de equipos, donde las personas realizan diversas tareas, con distintos sistemas informáticos y a veces en ubicaciones geográficas diversas. Es por ello que se dice que el software online corrige las limitaciones técnicas de las herramientas de gestión de proyectos convencionales. Resuelve el problema de la aplicación corriendo en la estación de trabajo del planificador, evita el problema del mantenimiento, actualizaciones manuales etc. No consume espacio en disco al ser una aplicación que corre en la Web, suponiendo que el planificador no desee instalarla en su ordenador. Puede ser consultada desde cualquier estación de trabajo y en cualquier lugar del mundo.

### **1.7.1.2 Project KickStart 4**

Project KickStart 4 es una herramienta que posee nuevas características con respecto a sus versiones anteriores, entre ellas una interfaz gráfica mucho más fácil de usar incluso para aquellos usuarios que se inician en el mundo de la planificación de software y que aún no tienen gran experiencia en el uso de estas herramientas. Entre sus inconvenientes se encuentra que es caro (\$ 129.95), para usarse debe usar

una computadora con tecnología Pentium, Sistema Operativo (SO) Windows 95, 98, Me, 2000, NT y XP y 32 MB RAM. O sea, al remitirse solo a la familia Windows como SO lo deja inutilizable para usuarios que usen Linux, Macintosh u otros. Sus necesidades tecnológicas lo hacen imposible de usar para usuarios del tercer mundo u otros lugares subdesarrollados del mundo que no usen tecnología Pentium. Además teniendo en cuenta su precio, y el no conocimiento de su código fuente, que hace imposible realizarle modificaciones para familiarizarlo con el entorno de trabajo en que se desea emplear, es muy posible que muchos usuarios lo desechen a pesar de algunas de sus ventajosas características.

### **1.7.1.3 Visual Studio 2005 Team System: Software Project Management (VSTS)**

VSTS cuenta con capacidades para líderes y planificadores de proyectos, permite integrar la planificación al proceso de software, balancea las cargas y organiza los recursos con las herramientas ya conocidas (Excel y Project, de Office). Estimula una comunicación eficiente entre los miembros de los equipos de desarrollo, proporcionando para cada proyecto un Portal *SharePoint Services* que permite consolidar en un repositorio central toda esa información relacionada al proyecto, como la documentación, planes, especificaciones y reportes que comunican avances en áreas específicas del desarrollo tales como pruebas, errores, riesgos, estimaciones, etc. Posee la capacidad de crear y administrar requerimientos, tareas y errores así como controlar las versiones y administrar los cambios del proyecto. Independientemente del rol que se juegue dentro de un equipo de desarrollo, colabora con el desarrollo del proyecto en un ambiente de alta colaboración. Entre las mayores inconveniencias que tiene VSTS es que es un software cuya interfaz puede resultar algo complicada para personal inexperto, es excesivamente caro, su proveedor es Microsoft, lo que lo hace imposible de ser comprado para usuarios cubanos, por las legislaciones impuestas por el bloqueo del Gobierno norteamericano para la compra de productos estadounidenses por personal cubano. Además, es un software que puede ralentizar la estación de trabajo, si no se cuenta con la memoria necesaria que exige su instalación. Es inútil para usuarios que son partidarios de usar Linux como Sistema Operativo. [9]

#### **1.7.1.4 Microsoft Project 2003**

Microsoft Project estándar 2003 es una herramienta para la administración de proyectos, que se puede usar muy fácilmente para planificar, programar y controlar el proyecto. Con esta herramienta, que facilita la administración del proyecto, se puede tener un control en diferentes niveles de acuerdo a las necesidades del cliente.

Permite generar varios tipos de diagramas, entre ellos Gantt y Pert, los cuales son muy fáciles de realizar con esta herramienta, igualmente mejora la capacidad de organización del el trabajo y se obtiene una comunicación eficaz entre los equipos de desarrollo del software. Esta herramienta permite que cada uno de los trabajadores cheque su estado de desarrollo y conozca el día que el equipo de calidad o el líder de proyecto van a revisar su trabajo.

Posibilita que el líder del proyecto ofrezca una panorámica actual al cliente sobre la situación de cada uno de los equipos de trabajo, y del software en general. En cuanto a seguridad, ofrece el uso de contraseñas si el proyecto lo necesita, para evitar que personal ajeno realice cambios en el mismo.

Quizás una de sus principales ventajas sea que en la UCI, el Microsoft Project esta contenido dentro del Programa de la Asignatura Ingeniería de Software, lo que lo convierte en un software popular y con el cual esta familiarizada la comunidad Universitaria.

Entre sus grandes inconvenientes está que es un software que produce la compañía norteamericana Microsoft, por ende, es un software con el cual no se le es permitido trabajar a los cubanos. Su costo asciende los 100 dólares y es incompatible con varios sistemas operativos, entre ellos el Linux. Puede ser muy engorroso para proyectos muy grandes (de más de 100 tareas) pues resulta casi imposible mostrar en pantalla el desarrollo completo de los mismos.

#### **1.7.2. Software libre. Herramientas para la planificación:**

El uso de software libre es muy importante para cubrir las licencias de uso en Cuba, además de que al ser código abierto es posible adaptar sus funcionalidades al entorno de trabajo en que se va a planificar el



proyecto de software. Sobre todo, su uso apoya la política de migración al software libre planteada por la UCI, que ya está siendo implementada en otras facultades.

### **1.7.2.1 GanttProject**

GanttProject es una herramienta opensource que tiene como principal objetivo colaborar con la administración y planificación de proyectos de cualquier envergadura, (a pesar de que se conoce que trabajar con un Diagrama de Gantt en proyectos grandes (más de 30 tareas) resulta bastante complejo a la hora de presentarlo en una pantalla).

Es muy útil en la planificación de proyectos de software, aunque utilizable para programar cualquier otro tipo de proyecto (empresariales, domésticos, etc.)

Al igual que otras herramientas esta permite una amplia gama de variedades para mejorar la planificación en si. Usa el formato XML para las copias de seguridad y permite exportar los proyectos como imágenes PNG o JPG, en documentos PDF o páginas HTML. Este programa está traducido a 20 idiomas entre los cuales se incluye el español y dispone de un instalador independiente del sistema operativo con lanzadores para Linux, MacOSX y Windows [10]

Una de sus principales ventajas es que es código abierto y por tanto es posible adaptar sus funcionalidades al propio entorno en que se trabaja e implementar características ausentes; con el único requisito de registrar esos cambios como una nueva versión y compartirlos con la comunidad de GanttProject, que cada día gana en adeptos. Es además gratuito para cualquier propósito (esta característica es muy importante teniendo en cuenta lo encarecidas que son las herramientas de gestión de proyectos comerciales).

### **1.7.2.2 OpenWorkbench**

Open Workbench es una herramienta de planificación de proyectos, al estilo de Microsoft Project. Se ejecuta en entornos Windows y es completamente libre y gratuito. Soporta un amplio abanico de funciones tales como la asignación de tareas hacia delante y hacia atrás, análisis de hitos, y tiempo estimado para la terminación de la solución. Posee una habilidad única de programar tareas de acuerdo a restricciones de recursos, y esta es una variedad nueva con respecto a otros software de planificación de

proyectos. Esta característica ofrece un plan mucho más real y utiliza menos tiempo, pues calcula la duración de la tarea en esfuerzo por disponibilidad de recursos.

Entre sus principales inconvenientes esta que solo funciona con Windows, por tanto es imposible de usar para usuarios amantes de Linux u otros Sistemas Operativos. Sin embargo es una herramienta que al ser OpenSource viabiliza el problema de licencias en Cuba que es un tema tan complicado. Adaptable al entorno de trabajo con tan solo modificar cualquiera de sus componentes.

### 1.7.2.3 Gforge

*GForge* es un portal y la vez una ramificación de la versión 2.61 del software de *SourceForge*. El enfoque que toma actualmente no es solo dirigido a la comunidad del Software Libre, también se consideran un objetivo las empresas de software y otras entidades que necesiten un entorno colaborativo para sus proyectos. Fue iniciado por Tim Perdue, autor de gran parte del código original de *SourceForge*. En realidad no es una herramienta de planificación, sino de Gestión de Proyectos, sin embargo es posible adaptarla a la planificación, pues es posible especificar las fechas de inicio y finalización de las tareas, el número de horas para cada elemento, asignar el encargado de cumplimentar la tarea de la forma persona-asignación o equipo-asignación. Gestiona dependencias entre tareas y las mismas se encuentran organizadas en subproyectos. Una tarea tiene como información asociada; porcentaje de completitud, prioridad, descripción breve y detallada, fecha de inicio y de finalización, asignación/es, dependencias y una estimación en horas, del tiempo que se empleará en llevar a cabo la misma.

### 1.7.2.4 Trac

Trac es una herramienta *opensource* con una interfaz web muy simple que integra herramientas básicas para comunicación, gestión y seguimiento de proyectos. Posee además una wiki que es un sitio web colaborativo que puede ser editado por varios usuarios. Los usuarios de una wiki pueden crear, editar, borrar o modificar el contenido de una página web, de una forma interactiva, fácil y rápida; dichas facilidades hacen de una wiki una herramienta efectiva para el trabajo en equipo. Su interfaz de subversion permite la gestión de versiones y una de las características más interesantes, que resulta verdaderamente atrayente para la planificación, es que cuenta con opciones para seguimiento de hitos, eventos y evolución del desarrollo del proyecto de software y un sistema de tickets para gestión de bugs,

tareas y otras incidencias. Su uso se ha comenzado a difundir en la Universidad y sería realmente destacado que fuera incluido a todos los proyectos en desarrollo y muy recomendable además favorecer su estudio en el estudiantado.

La siguiente tabla muestra una comparación de las herramientas estudiadas teniendo en cuenta algunas de sus principales características.

Leyenda: (1). Seguimiento y Control de tareas

(2) Precio

(3) Multiplataforma

(4) Integración con otras herramientas

(5) Seguridad

(6) Dificultad de uso.

Herramienta	(1)	(2)	(3)	(4)	(5)	(6)
B-kin Project Monitor	si	15 USD mensuales	no	Alta	Media	Media
Project KickStart 4	no	129.95 USD	no	Alta		Baja
VSTS	si	+ de 100 USD	Si	Alta	Media	Alta
Microsoft Project 2003	si	+ de 100 USD	Sí	Alta	Media	Baja.
GanttProject	no	gratis	si	Media	Media	Baja
OpenWorkbench	no	gratis	no	Baja	Media	Baja
Trac	si	gratis	si	Alta	Alta	Alta
Gforge	si	gratis	no	Media	Media	Alta

**Tabla 1. Comparación entre diferentes herramientas para la planificación.**

**Herramienta escogida para la planificación del proyecto SIGE:**

Microsoft Project 2003 es la herramienta que se enseña como parte del programa de la asignatura Ingeniería de software y por tanto es una con la cuales esta más familiarizado el estudiantado y como se mencionaba antes es muy importante que los equipos de trabajo conozcan la herramienta de planificación. Es además muy fácil de usar y posee un entorno grafico sencillo y fiable. A pesar de ciertos inconvenientes entre los que se encuentra que MSProject requiere Microsoft Outlook u otro programa de correo electrónico compatible con MAPI para mantener a los equipos de desarrollo informados y de que es software propietario con todos los inconvenientes antes mencionados que esto puede acarrear.

*¿Por qué MS Project 2003 y no una herramienta de software libre?*

Evidentemente hubiese sido muy conveniente utilizar software libre en la planificación del proyecto SIGE, colaborando así, con la política de migración a tecnologías opensource propuesta por la UCI. Evitando un problema que costaría millones de dólares por uso ilícito de software, sin licencia de uso. Esto sin contar la política agresiva norteamericana con nuestro país, en el uso de las tecnologías, que nos impiden hacer uso de muchas de ellas. En cambio esta política de migración debe comenzar desde los cimientos, incluyéndose en el programa de estudio de la carrera y proporcionando a los estudiantes un marco para su utilización. El aprendizaje del manejo correcto de una nueva herramienta para la planificación, control y seguimiento de las actividades del proyecto hubiese costado mucho tiempo, con el cual no se contaba inicialmente. Es propósito del proyecto cambiar en próximos meses al uso de una herramienta de software libre no solo para llevar la planificación, sino para otras actividades del proyecto.

**1.8 Metodología utilizada.**

El desarrollo de cada proyecto posee diversas características, según las mismas se determina que metodología para el desarrollo de software se adecua más a las condiciones de este. Existe cierta tendencia en la facultad relacionada a la utilización de Rational Unified Process (RUP), porque además de ser muy conocida, es muy flexible a la hora de idear el proyecto, sobre todo porque es iterativa e incremental, lo que permite a su vez realizar una planificación detallada de cada uno de los artefactos a

generar, nutriéndose de la iteración anterior en la planificación de la próxima, teniendo presente el momento donde se sitúa dentro del ciclo de desarrollo. RUP le presta una gran importancia al flujo de trabajo Administración de Proyectos, el cual se desarrolla desde la fase de Inicio hasta la Fase de transición, dentro de este flujo se encuentra enmarcada la Planificación. Además, esta metodología recomienda la división de proyectos complejos en módulos o mini proyectos para que sea más fácil su trabajo y la planificación. SIGE esta dividido en 4 módulos:

Módulo de Entrada a Datos.

Módulo Generador de Modelos.

Módulo Registros y Clasificadores.

Módulo Generador de Reportes.

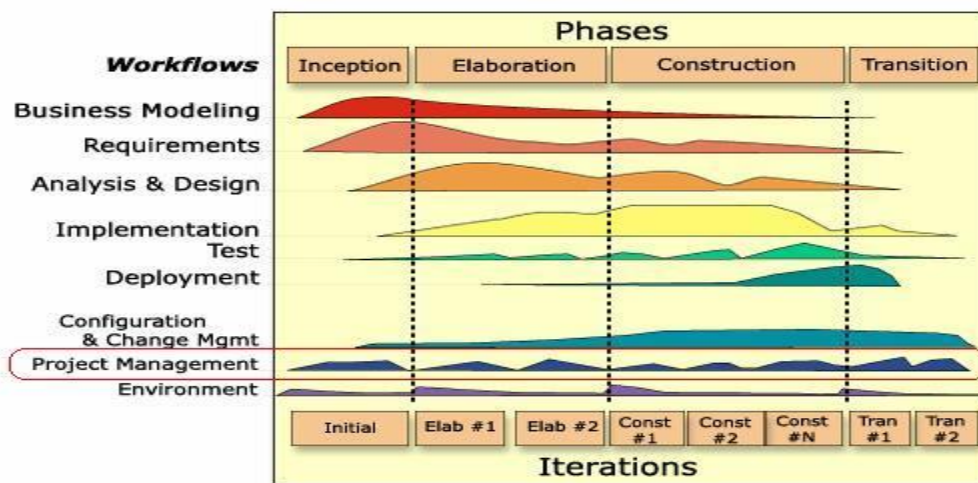


Figura 2. Fases y Flujos de trabajo propuestos por RUP. [1]

## 1.9 Gestión de riesgos

La planificación de un proyecto implica un riesgo, a pesar de un buen enfoque de estimación y datos históricos con los que trabajar, la estimación puede o no ser certera, esto solo se sabrá cuando se concluya el proceso de desarrollo del sistema, Toni Gilb planteó que : “Si usted no ataca los riesgos activamente, ellos le atacarán activamente a usted.” Además de la estimación, es muy importante crear un plan de mitigación de riesgos, siempre existe algún por ciento de posibilidades de que la planificación

de riesgos no haya abordado todos los peligros posibles, en cuyo caso, es el planificador el encargado de minimizar estas consecuencias a un punto que apenas sean perceptibles.

*Riesgo* es "la contingencia o proximidad de un daño" y contingencia significa la posibilidad de que algo suceda o no.

El principal objetivo de la Gestión de Riesgos es identificar los posibles peligros que puedan afectar el correcto desarrollo del software. Para ello debe trazar una estrategia, cuidando de en caso de que el riesgo se produzca el plan de riesgos elaborado lo contemple y sus efectos puedan ser mitigados de tal forma que los efectos sobre la organización sean mínimos.

*Fases de la Gestión del Riesgos:*

**Fase I:** Identificación y clasificación de los factores que implican riesgo y evaluación previa de la importancia de los mismos.

**Fase II:** Análisis del Riesgo: análisis y evaluación de dichos factores de manera que la incertidumbre sea cuantificada en lo posible.

**Fase III:** Definición de estrategias (evitar el riesgo, retenerlo y asumirlo, transferirlo) para evitar el impacto del riesgo e incrementar su control.

**Fase IV:** Realizar de manera continuada las tres fases anteriores. Por lo que debe incluir los procedimientos básicos a seguir para la Gestión del Riesgo durante el resto del Proyecto. [2]

Según Charette [11] los riesgos pueden ser:

**Predecibles:** Aquí se encuentran todos aquellos riesgos que pueden ser obtenidos de la experiencia en proyectos anteriores.

**Impredecibles:** Estos son realmente una caja de sorpresas, pues pueden ocurrir pero son verdaderamente difíciles de encontrar por adelantado.

Otras clasificaciones son:

**Riesgos específicos:** Aquí se concentran los riesgos propios del producto que se está desarrollando y que nada tienen que ver con otras organizaciones y/o proyectos.

**Riesgos administrativos o de entorno:** (Ej. Tramitación de permisos, licencias, etc.).

**Riesgos técnicos:** (Ej.: Uso de nuevas tecnologías (y por ende desconocidas) o de tecnologías obsoletas.)

**Riesgos financieros:** (Ej.: Escasos fondos para respaldar el proyecto de software.)

**Riesgos sobre la disponibilidad de los recursos:** (Ej.: Escasos recursos para el desarrollo del software o recursos críticos compartidos entre muchas personas)

**Riesgos sobre la planificación:** (Ej.: No se puede construir un producto de tal envergadura en el tiempo asignado.)

Los riesgos dentro de un proyecto son muy variables, la misma función de enlistarlos y además darle un seguimiento de acuerdo a la prioridad que se les haya asignado es vital, de esta forma un riesgo que era inminente en la fase de inicio del proyecto ya en la fase de prueba quizás sea desechado.

### **Planificación de riesgos:**

Según Catherine Aguilar Ramos [12] el objetivo de la planificación de riesgos es describir como serán abordados y planeadas las actividades de riesgos del proyecto. Evidentemente la planificación de riesgos tiene como objetivo principal el identificarlos, clasificarlos, priorizarlos, proponer una solución para cada uno de ellos y mantener un chequeo estricto. Es muy importante tener en cuenta la adaptación de la regla de Pareto 80/20 a la Gestión de riesgos del software, que especifica que el 80 % del riesgo total de un proyecto se debe solamente al 20 % de los riesgos identificados.

La siguiente figura muestra el orden lógico a seguir para lograr un óptimo análisis cualitativo de los riesgos.



**Figura 3. Análisis cualitativo de los riesgos [12]**

El Líder del proyecto y el planificador son las 2 personas encargadas de gestionar nuevas personas para trabajar en la Gestión de riesgos y además gestionar todas las fases que lleva las mismas. El planificador debe sobre todo evaluar detalladamente el impacto en cuánto a tiempo de atraso y costo que tendrá un riesgo determinado, es importante que si son identificados muchos riesgos críticos la Gestión de riesgos prácticamente pasa a ser un mini proyecto.

Los recursos humanos necesarios para una gestión de riesgos adecuada son:

**Líder del proyecto:** Es el encargado de gestionar o autorizar los recursos necesarios para mitigar los riesgos producidos. Revisa la prioridad de los riesgos y se preocupa porque se garantice una correcta Gestión de riesgos.

**Planificador:** Es el Encargado de planificar la Gestión de Riesgos y de hacer el Plan de mitigación de riesgos.

**Jefes de módulo:** Son los encargados de hacer una estimación precisa de acuerdo a los riesgos, sus probabilidades de ocurrencia y el impacto que tendrán.

**Personal del proyecto:** Se debe seleccionar dentro del proyecto cierta cantidad de personas que tengan como tarea coordinar las actividades de hallar y planificar los riesgos conjuntamente con el planificador. Mientras más grande es el proyecto, mayor es la necesidad de conformar este equipo cuya principal tarea



se enmarca en mantener actualizada la lista de riesgos e informar constantemente sobre ella y las actividades de contingencia.

***Pasos a seguir para una buena Gestión de riesgos:***

**1. Identificación de la lista de riesgos:** Es un buen paso buscar inicialmente en la literatura y luego remitirse a buscar más detalladamente los riesgos específicos de nuestro proyecto, o de nuestro entorno.

**2. Evaluar los riesgos de acuerdo a la matriz de probabilidad e impacto:** Consiste en que el equipo de Gestión de riesgo se reúna con las demás personas involucradas y lleguen a un consenso de acuerdo a la probabilidad de ocurrencia que tiene cada riesgo y la probabilidad de impacto además.

<b>Escala de probabilidad y ocurrencia de riesgos</b>		
Escala	Definición	Descripción
0.1	Muy improbable	Seria una sorpresa que ocurriera
0.3	Poco probable	Mas probabilidad que no ocurra a que si
0.5	Probable	50 % de probabilidades
0.7	Altamente probable	Grandes probabilidades de que ocurra
0.9	Casi cierto	Seria una sorpresa si no ocurriera

**Tabla 2. Definición de la escala de probabilidad y ocurrencias [12]**

Estas escalas de probabilidad se deben corresponder con una escala de ocurrencia, así un riesgo que tuviera una probabilidad de ocurrencia de 0.9 y una probabilidad de impacto alta seria una verdadera catástrofe.

Definición de escala de impacto negativo en los objetivos del proyecto					
Objetivos del proyecto	Escala				
	Muy poco (0.05)	Poco (0.10)	Moderado (0.20)	Alto (0.40)	Muy alto (0.80)
Costo	Incremento insignificante	Incremento en costo < 5%	Incrementos en costos entre 5-10%	Incremento en costos entre 10-20%	Incremento en costos > 20%
Tiempo	insignificante	Retraso < 5%	Retraso global entre 5-10%	Retraso global 10-20%	Retraso global > 20%
Alcance	Reducción escasamente apreciable	Áreas menores de alcance afectadas	Áreas mayores de alcance afectadas	Reducción de alcance inaceptable a la Empresa	Fin del Proyecto. Ítem es inutilizable
Calidad	Degradación escasamente apreciable	Solo aplicaciones muy exigentes se afectan	Reducción de calidad que requiere aprobación	Reducción de calidad inaceptable por la Empresa	Fin del Proyecto. Ítem es efectivamente inutilizable

**Tabla 3. Escala de impacto negativo de los riesgos previstos [12]**

Con la probabilidad y el impacto en la mano es posible calcular la exposición al riesgo real. Esta magnitud de exposición es permisible utilizarla para la clasificación de escala o lo que es lo mismo darle una prioridad a los riesgos. Retomando el ejemplo anterior de probabilidad de ocurrencia 0.9 y alta probabilidad de impacto, este riesgo sin dudas algunas sería Prioridad 1

Probabilidad	Amenazas				
0.90	0.05	0.09	0.18	0.36	0.72
0.70	0.04	0.07	0.14	0.28	0.56
0.50	0.03	0.05	0.10	0.20	0.40
0.30	0.02	0.03	0.06	0.12	0.24
0.10	0.01	0.01	0.02	0.04	0.08
Impacto	0.05	0.10	0.20	0.40	0.80

**Figura 4. Matriz de probabilidad e impacto. [12]**

En esta matriz el área azul oscura representa aquellos riesgos que son muy altos, el área azul mas clara representa los riesgos moderados y el área morada representa los riesgos menos importantes.

Evidentemente a los riesgos que se encuentran en la parte azul oscura es necesario prestarle más atención porque son los que presentan más probabilidad de impacto en los objetivos del proyecto. Sería muy conveniente proponer un plan de mitigación ofensiva con respecto a ellos.

### 3. Clasificar los riesgos según su impacto:

**Prioridad 1** si son riesgos muy altos

**Prioridad 2** si son riesgos moderados

**Prioridad 3** si son menos importantes

### 4. Análisis cuantitativo de los riesgos

En esta etapa muchos proyectos escogen simular el riesgo para obtener resultados, representar en un árbol de decisiones las posibles vías a tomar etc.

### 5. Planificación de respuesta al riesgo.

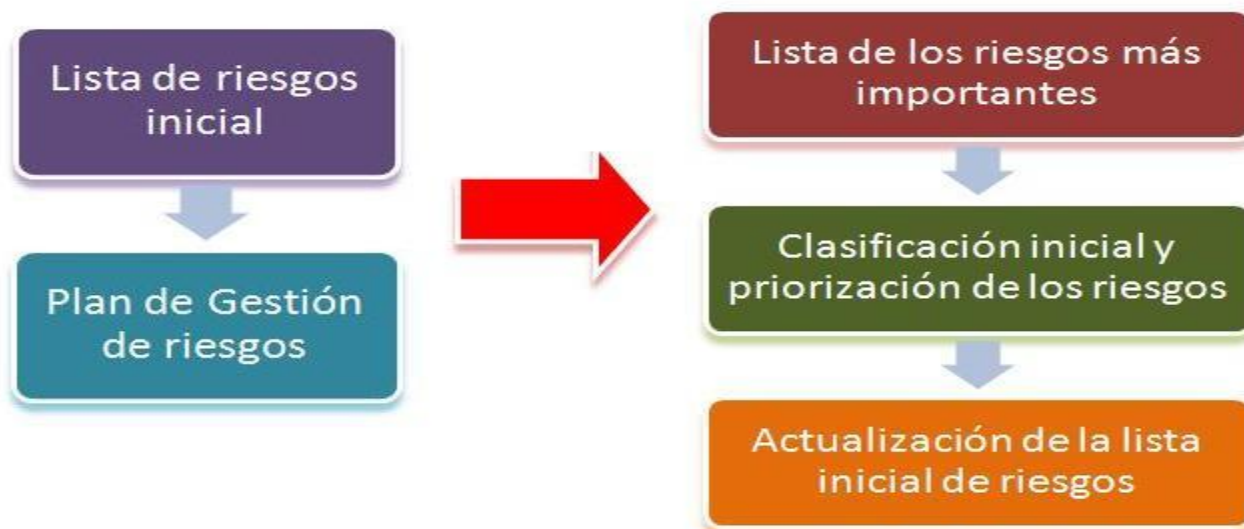


Figura 5. Planificación de Respuesta al riesgo. [12]

## **6. Seguimiento y Control del riesgo.**

Mantener constantemente informado al resto del equipo del proyecto de las amenazas que pueden posibilitar el riesgo, como el mismo ha evolucionado y que nivel de control se tiene sobre el mismo.

### **1.10 Conclusiones del Capítulo 1:**

El uso de una herramienta específica para planificar un proyecto de software es esencial para el buen funcionamiento de la planificación. Es importante a la hora de seleccionar alguna, estudiar el grado de acoplamiento que tiene a las características del proyecto en cuestión. La familiarización del equipo de trabajo con la misma es elemental, pues cualquier error introducido en la planificación puede repercutir en grandes errores más adelante y por ende en considerables pérdidas de tiempo e incumplimiento con el compromiso de entrega del producto. Es necesario capacitar previamente a los equipos de trabajo con las herramientas de planificación seleccionadas, con el propósito de lograr una unidad de criterio en cuanto a posibles variantes llevadas a cabo dentro del proyecto.

La Estimación del proyecto es vital realizarla para tener una idea cercana a los recursos que necesita el proyecto, además del tiempo que necesita para desarrollarse. Esta estimación es el único lazo del cual se puede agarrar el equipo de desarrollo para convencer al cliente, en caso que este desee el producto terminado en un plazo de tiempo irreal.

La Gestión de Riesgos se inicia desde que se inicia el proyecto, un proyecto de software que no le presta importancia es un proyecto suicida. A pesar de que muchos autores recomiendan prestarle alta importancia a los 10 primeros riesgos de Prioridad 1, la práctica ha demostrado que todos los riesgos son importantes, que aquellos que se creyeron insignificantes pueden convertirse en un verdadero dolor de cabeza. Es aconsejable tratar a todos los riesgos, independientemente del grado de prioridad que posean y trazar un plan de contingencia desde la formación del equipo de desarrollo.



## *Capítulo*

# *Desarrollo de la planificación.*

### **2.1 Introducción**

En este capítulo se realizará la estimación de los módulos del proyecto SIGE por 2 métodos de estimación, lo que permite comparar los resultados de ambos y determinar cuál de ellos se ajustó más al desarrollo del proyecto y a partir de ahí poder establecer la cantidad de recursos materiales, humanos y de entorno necesarios para el mismo; así como una valoración del tiempo total de vida del proyecto. Se hará una definición de los mecanismos establecidos para la captura, seguimiento y control de la información. Un breve bosquejo sobre el Plan de Fases, así como el desarrollo de una adecuada Gestión de Riesgos, estableciendo un breve plan de contingencia para los riesgos críticos.

### **2.2 Estimación del Proyecto**

El uso de las técnicas específicas para la estimación es vital para el buen funcionamiento de la planificación. En el caso del proyecto SIGE se usó Análisis de Puntos de Casos de Uso y COCOMO II y Análisis de Puntos de Función.

¿Por qué se escogieron estas técnicas de estimación y no una sola u otras?

Es muy conveniente usar más de una técnica de estimación con el propósito de poder comparar los resultados de las mismas y decidir entonces un número lo más real posible de recursos necesarios. Estas técnicas se ajustaban a las características del proyecto, además de ser COCOMO II uno de los métodos más recomendados y completos para la estimación. La estimación y planificación se hizo de forma separada para cada módulo del proyecto.

### 2.2.1 Estimación del Módulo Generador de Modelos (SIGE MGM) por la técnica de Puntos de Casos de Usos.

Identificación los Puntos de casos de uso Desajustados (UUCP)

$$UUCP = UAW + UUCW$$

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Para calcular UAW

<b>Tipo</b>	<b>Descripción</b>	<b>Peso</b>	<b>Cantidad * peso</b>
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	2*3
Total			6

**Tabla 4. Cálculo del peso de los actores sin ajustar.**

Para calcular UUCW

<b>Tipo</b>	<b>Descripción</b>	<b>Peso</b>	<b>Cantidad * peso</b>
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	4*5
Medio	El Caso de Uso contiene de 4 a 7	10	0*10

	transacciones.		
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	4*15
Total			80

**Tabla 5. Cálculo del Factor de peso de los Casos de Usos sin ajustar.**

$$UUCP = 6 + 80$$

$$UUCP = 86$$

Ajuste de los Puntos de casos de uso

$$UCP = UUCP * TCF * EF$$

UCP: Puntos de Casos de Uso ajustados

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Cálculo del TCF

$$TCF = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor}_i)$$

Significado de los valores

0: No presente o sin influencia,

1: Influencia incidental o presencia incidental

2: Influencia moderada o presencia moderada

3: Influencia media o presencia media

4: Influencia significativa o presencia significativa

5: Fuerte influencia o fuerte presencia

Factor	Descripción	Peso (P)	Valor (V)	Comentario	$\Sigma(P_i * V_i)$
T1	Sistema distribuído	2	3	El sistema será instalado en todas las provincias pero inicialmente no habrá	6

				gran interacción entre ellas mediante el sistema	
T2	Objetivos de performance o tiempo de respuesta.	1	5	Tiempo de respuesta alto.	5
T3	Eficiencia del usuario final	1	0	Ninguna restricción de eficiencia	0
T4	Procesamiento interno complejo	1	0	No hay cálculos complejos.	0
T5	El código debe ser reutilizable	1	5	El código es reutilizable para que sea posible usarlo en próximas versiones	0
T6	Facilidad de instalación	0.5	5	Muy fácil de instalar.	2.5
T7	Facilidad de uso	0.5	3	Normal	1.5
T8	Portabilidad	2	0	El sistema solo corre sobre Plataforma Windows	0
T9	Facilidad de cambio	1	3	Se requiere un costo moderado de mantenimiento	3
T10	Concurrencia	1	0	No hay concurrencia.	0
T11	Incluye objetivos especiales de seguridad	1	3	Seguridad media.	3
T12	Provee acceso directo a terceras partes	1	1	Solo usuarios autorizados pueden acceder a la aplicación.	1
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	5	Los usuarios del sistema deben entrenarse para usar el software. .	5
Total					27
TCF	$0.6+0.01 * \sum (P_i * V_i)$				0.87

**Tabla 6. Total del peso por el Valor de los factores de complejidad técnica del SIGE MGM**



## Cálculo de EF

Factor	Descripción	Peso	Valor	Comentario	$\Sigma(P_i * V_i)$
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	El grupo está muy familiarizado con el modelo.	7.5
E2	Experiencia en la aplicación	0.5	3	La mayoría del grupo ha trabajado desde el inicio del proyecto, aunque aún faltan detalles por pulir.	1.5
E3	Experiencia en orientación a objetos	1	3	No todos los integrantes del módulo conocen de tecnología orientada a objetos	3
E4	Capacidad del analista líder	0.5	5	Alta capacidad del analista	2.5
E5	Motivación	1	5	El equipo de trabajo está altamente motivado	5
E6	Estabilidad de los requerimientos	2	4	Se mantienen bastante estables.	8
E7	Personal part-time	-1	4	El equipo de trabajo es part -time	-4
E8	Dificultad del lenguaje de programación	-1	3	Se usó lenguaje C++.	-3
Total					20.5
EF	$1.4 - 0.03 * \Sigma (P_i * V_i)$				0.79

**Tabla 7. Total del peso por el valor de los Factores de ambiente del SIGE MGM**

Luego

$$UCP = UUCP * TCF * EF$$

$$UCP = 86 * 0.87 * 0.79$$

$$UCP = 59.11$$

Cálculo esfuerzo del Flujo de Trabajo de Implementación

$$E = UCP * CF$$

**E:** esfuerzo estimado en horas-hombre **CF:** factor de conversión

**CF** = Total<sub>EF</sub> = Cantidad EF < 3 (entre E1 - E6) + Cantidad EF > 3 (entre E7- E8)

Si el total es 2 o menos, se utiliza el **factor de conversión 20 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 20 horas-hombre.

Si el total es 3 o 4, se utiliza el **factor de conversión 28 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 28 horas-hombre.

Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que **el riesgo de fracaso del mismo es demasiado alto**.

**Total**<sub>EF</sub> = 2 + 0

**Total**<sub>EF</sub> = 2

**CF** = 20 horas-hombre

Luego **E** = 59.11 \* 20 horas-hombre

**E** = 1182 horas-hombre

Cálculo del esfuerzo de todo el proyecto

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	296 horas-hombre
Diseño	20%	591 horas-hombre
Implementación	40%	1182 horas-hombre
Prueba	15%	443 horas-hombre
Sobrecarga	15%	443 horas-hombre
<b>Total</b>	100%	2955 horas-hombre

**Tabla 8. Esfuerzo distribuido por fases del SIGE MGM.**

**Esfuerzo total (E<sub>T</sub>).** = 2955 horas- hombre

Si se quiere llevar el Esfuerzo a mes-hombre:

Un día tiene 24 horas, diariamente se trabajan en el proyecto solo 4 horas por persona, por dificultades de hardware (no hay suficientes máquinas y el personal está distribuido en 2 computadoras (PC) aproximadamente), además de otras responsabilidades que deben cumplir las personas asociadas al proyecto. Un mes tiene como promedio 30 días, de ellos solo 22 son laborables (se exceptúan los fines de semana), la cantidad de horas que puede trabajar una persona en un mes son 88 horas.

Si  $E_T = 2955 \text{ horas-hombre}$  y por cada 88 horas se tiene 1 mes de trabajo, da como resultado  $E_T = 34 \text{ mes-hombre}$

Esto quiere decir que 1 persona puede realizar el problema analizado en más o menos 34 meses.

En el equipo hay 5 personas y aunque no todas realizan el mismo esfuerzo entonces el módulo analizado debe culminarse aproximadamente en 7 meses.

### 2.2.2 Estimación del SIGE MGM POR COCOMO II Y Puntos de función.

Entradas Externas (Ver Anexo 1)

EI	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Crear modelos	1	Más de 15	Media
Modificar modelos	1	Más de 15	Media
Enviar modelo	1	Más de 15	Media
Actualizar Base Metodológica	1	Más de 15	Media
Eliminar Base Metodológica	1	Más de 15	Media
Crear Nomenclatura	1	De 5 a 15	Baja

Actualizar nomenclatura	1	De 5 a 15	Baja
Crear Cuadres	1	Más de 15	Media
Modificar cuadros	1	Más de 15	Media
Eliminar Cuadres	1	Más de 15	Media
Exportar	0	De 1 a 4	Baja
Importar	0	De 1 a 4	Baja
Imprimir	0	De 1 a 4	Baja

**Tabla 9. Entradas Externas del Módulo SIGE MGM**

*Salidas Externas (Ver Anexo 1)*

Salidas Externas	C. de Ficheros	C. Elem de datos	Clasificación.
Modelos	1	Más de 19	Media
Base Metodológica	1	De 6 a 19	Baja
Nomenclatura	1	De 6 a 19	Baja
Cuadres	1	Más de 19	Media

**Tabla 10. Salidas Externas del Módulo SIGE MGM.**

*Ficheros lógicos internos (Ver Anexo 1)*

ILF	Cantidad de record	Cantidad de elementos de datos	Clasificación
TB_ Modelo	1	Más de 50	Media
TB_ Base M	1	Más de 50	Media
TB_ Nomenclatura	1	De 1 a 19	Baja
TB_ Cuadres	1	Mas de 50	Media

**Tabla 11. Ficheros lógicos internos del Módulo SIGE MGM.**

Aporte:

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas	5*3	8*4	0	47
Salidas Externas	2*4	2*5	0	18
Archivos Lógicos Internos	1*7	3*10	0	37
			Total	102

**Tabla 12. Puntos de función desajustados del módulo SIGE MGM.**

Puntos de Función sin ajustar (UFP) = 47 + 18 + 37 = 102

Estimar la cantidad de instrucciones fuentes

*SLOC (Source Lines of Code)*

SLOC = UFP \* ratio

= 102 \* 53

= 5406

SIZE = SLOC / 1000

= 5.406

Aplicación de la fórmula de Boehm:

PM nominal =  $A \times (\text{Size})^B$

PM ajustado = PM Nominal \*  $\prod_{i=1}^n \Pi^i$  (Emi)

**A:** Se toma el valor por defecto del modelo  $A = 2.94$ .

**B:** Es una constante denominada Factor escalar, la cual tiene un impacto exponencial en el esfuerzo y su valor está dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto.

**Si  $B < 1.0$ .** El proyecto presenta ahorros de escala.

**Si  $B = 1.0$ .** Los ahorros y gastos de escala están equilibrados.

Si  $B > 1.0$ . El proyecto presenta gastos de escala.

**C y D** son coeficientes definidos en la organización.  $C= 3.67$ ,  $D= 0.28$

$$B = 0.91 + 0.01 * \Sigma (W_i)$$

Donde el valor de las **W<sub>i</sub>** se muestra en la siguiente tabla:

Factores de escala (SF) (Ver Anexo 2)

Variable	Descripción	Ponderación	Valor
PREC	El sistema es novedoso pero el equipo de desarrollo lo comprende.	Nominal	3.72
FLEX	Algo de relajación en cuanto a la flexibilidad del desarrollo	Nominal	3.04
RESL	La arquitectura aún no es sólida. El plan identifica pocos riesgos críticos.	Bajo	5.65
TEAM	La interacción del equipo es principalmente cooperativa	Muy Alto	2.19
PMAT	La madurez del proceso software es baja nivel 1 superior	Bajo	6.24
		Total	20.84

**Tabla 13. Valor de los factores de escala del Módulo SIGE MGM.**

$$B = 0.91 + 0.01 * 20.84$$

$$B = 0.91 + 0.2084$$

$$B = 1.1184$$

Finalmente, el esfuerzo nominal resulta:

$$\text{PM nominal} = A \times (\text{Size})^B$$

$$= 2.94 \times (5.406)^{1.118}$$

$$= 2.94 \times 6.60$$

$$= 19.40$$

$$\text{PM ajustado} = \text{PM Nominal} * \prod_{i=1}^n (E_{mi})$$

$$= 19.40 \times 1.296$$

$$= 25.15 \text{ meses-hombre}$$

Multiplicadores de escala (EM): (Ver Anexo 2)

Multiplicador	Descripción	Ponderación	Valor
PERS	El Personal tiene una alta capacidad en el análisis pero no en la programación. La capacidad de equipo aun necesita fortalecerse.	Bajo	1.26
RCPX	Las exigencias de confiabilidad, documentación y volumen de datos son altas, y la complejidad del producto es alta.	Alto	1.30
RUSE	Se pretende que todo sea reutilizable	Bajo	0.95

PDIF	La plataforma es estable, el tiempo de respuesta del sistema debe ser rápido.	Bajo	0.87
PREX	Los programadores son bastante inexpertos en la aplicación, el lenguaje y las herramientas utilizadas.	Alto	0.87
SCED	Se requiere terminar el proyecto en el tiempo estimado.	Nominal	1
FCIL	Se tienen herramientas CASE simples.	Bajo	1.10
		Total	1.2956

**Tabla 14. Valor de los multiplicadores de escala.**

**PM ajustado** = 25.15 meses-hombre, expresando el mismo valor en horas-hombre, y teniendo en cuenta que un mes de trabajo en el proyecto SIGE es aproximadamente 88 horas, el esfuerzo resulta:  $25.15 \times 88 = 2214$  horas-hombre



## Tiempo de vida del módulo SIGE MGM

Cálculo de:	Justificación:
Tiempo de desarrollo (TDEV)	$TDEV = C * (PM)^F \text{ (meses)}$ <p>donde:</p> $F = D + 0.2 * 0.01 * \sum SF$ $F = 0.28 + 0.2 * 0.01 * 20.84$ $F = 0.33$ $TDEV = C * (PM)^F$ $= 3,67 * (25.15)^{0,33}$ $= 3.37 * 2.90$ $= 9.77 \text{ meses} = 10 \text{ meses}$

Tabla 15. Estimación del tiempo de vida del Módulo SIGE MGM.

## Cantidad de hombres

Cálculo de:	Justificación:
Cantidad de hombres (CH)	$CH = \frac{PM}{TDEV}$ $CH = \frac{25.15}{10} = 2.51 \text{ hombres} \sim 3 \text{ hombres.}$ <p>Los valores obtenidos indican que el proyecto necesitaría 3 hombres para su realización en aproximadamente 10 meses, pero en la práctica se cuenta con 5 hombres, razón por la cual se debe ajustar el tiempo de desarrollo para esta nueva entrada.</p> $= \frac{25.15}{5} = 5.03 \text{ aproximadamente 6 meses}$

Tabla 16. Estimación de la cantidad de hombres del módulo SIGE MGM.

La siguiente tabla muestra los resultados de la estimación por Puntos Casos de Usos de los módulos restantes del proyecto SIGE. (Ver Anexo 2)

Módulo	UAW	UUCW	UUCP	TCF	EF	UCP	CF	E total H/hombre.	Meses	Pers onas
<b>MED</b>	15	70	85	0.98	0.89	74.13	4	5186	15	4
<b>MGR</b>	9	50	59	0.98	0.62	36	4	2520	8	4
<b>MRC</b>	3	55	58	0.92	0.63	33.62	2	1680	5	4

**Tabla 17. Resultados de la estimación por Puntos de Casos de Uso de los módulos restantes del proyecto SIGE**

La siguiente tabla muestra los resultados de la Estimación por COCOMO II y Puntos de Función para los módulos restantes del proyecto SIGE. (Ver Anexo 2)

Módulo	UFP	SIZE	PM Nominal	PM Ajustado	E total Horas - hombre	TDEV	CH
<b>MED</b>	111	5.883	21.34	27.74	2441	7	4
<b>MGR</b>	83	4.399	15.38	19.93	1760	5	4
<b>MRC</b>	96	5.088	18.14	23.58	2076	6	4

**Tabla 18. Resultados de la Estimación por COCOMO II y Puntos de Función para los módulos restantes del proyecto SIGE.**

Las siguientes tablas muestran una comparación entre los resultados de los métodos utilizados para cada módulo.

Módulo	Cocomo II y PF (horas/hombre)	Puntos de CU (horas/hombre)
MGM	2214	2955
MED	2441	5186

MRC	2076	1680
MGR	1760	2520

**Tabla 19. Comparación de los resultados entre ambos métodos (en horas/hombre)**

Módulo	Recursos humanos	Cocomo II y PF (en meses)	Puntos de CU (en meses)
MGM	5	6	7
MED	4	7	15
MRC	4	6	5
MGR	4	5	8

**Tabla 20. Comparación entre ambos métodos (en meses)**

### 2.3 Recursos usados en el proyecto SIGE

Los recursos materiales usados en un proyecto son muy importantes, pues son los que viabilizan las tareas a realizar. Sin ellos, sería imposible realizar el software. Es lo mismo que la disyuntiva de hacer pan sin tener fuego donde cocerlo. El principal objetivo de este estudio es conocer las características de todos los medios destinados al proyecto SIGE.

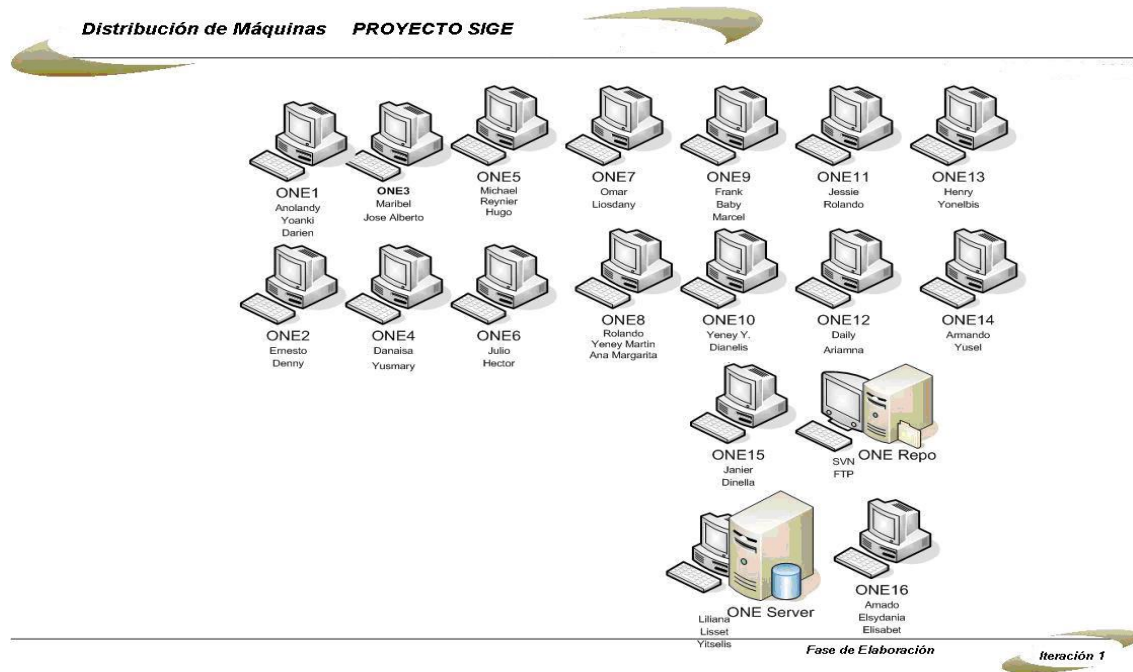
A continuación se muestra una tabla que especifica los recursos materiales usados en el proyecto SIGE, el estado en que se encuentran, así como otras particularidades.

Recurso	Descripción	Estado	Disponibilidad	Tiempo que será aplicado
3 Computadoras	Distribuidas en el MED y compartidas con otros módulos.	Bueno	Tiempo completo	Hasta la culminación del sistema.
3 Computadoras	Distribuidas en el MGM y compartidas con otros módulos.	Bueno	Tiempo completo	Hasta la culminación

				del sistema.
2 Computadoras	Distribuidas en el MGR	Bueno	Tiempo completo	Hasta la culminación del sistema.
2 computadoras	Distribuidas en MRC, una de ellas compartida con otros módulos.	Bueno	Tiempo completo	Hasta la culminación del sistema.
3 computadoras.	Distribuidas en el equipo de arquitectura.	Bueno	Tiempo completo	Hasta la culminación del sistema.
3 computadoras	Distribuidas en el equipo de Base de Datos y compartidas con otros equipos.	Bueno	Tiempo completo	Hasta la culminación del sistema.
6 computadoras	Distribuidas en el equipo de calidad y compartidas con otros equipos	Bueno	Tiempo completo	Hasta la culminación del sistema.
1 computadora	Destinada a la planificación y se comparte con el equipo de calidad.	Bueno	Tiempo completo	Hasta la culminación del sistema.
1 Computadora Servidor de Base de datos	Destinada para ser el servidor de Base de Datos del proyecto, es compartida además con otros equipos.	Bueno	Tiempo completo	Hasta la culminación del sistema.
1 computadora Repositorio del Proyecto	Destinada para ser el repositorio del proyecto donde se guarda todo lo referente a el.	Bueno	Tiempo completo	Hasta la culminación del sistema.

**Tabla 21. Recursos materiales del proyecto SIGE.**

La figura muestra la distribución de las máquinas en el laboratorio donde se desarrolla el proyecto.



**Figura 6. Distribución y organización de recursos del proyecto SIGE**

Los recursos humanos son los hilos que componen el suéter, sin ellos no habría software porque: ¿Quién lo construiría? Es menester prestar especial atención a los trabajadores del proyecto, pues sobre ellos recae la parte más fuerte: El desarrollo del software. Con este análisis se pretende conocer la cantidad de personas implicadas en el proyecto, así como su responsabilidad y el grado de calificación que poseen para ejercer su trabajo. La siguiente tabla muestra la cantidad de personas del proyecto SIGE, organizadas por el equipo de desarrollo al que pertenece y especificando el rol que desempeña y la experiencia que posee.

Nombre del equipo	Cantidad	Rol desempeñado	Experiencia
Equipo de dirección	5	Líder	Alta

		Planificador Gestor de configuración y Cambios Jefe de Calidad	Media Media Media
Equipo de Arquitectura	5	Arquitecto principal Arquitecto de Seguridad Arquitecto de Acceso a Datos Arquitecto de lógica del negocio	Alta Media Media Media
Equipo de Base de Datos (BD)	3	Diseñador de la Base de Datos Administrador de la BD Programador de la BD	Media Media Media
Equipo de Calidad	6	Responsable de Calidad Revisor técnico Diseñador de pruebas 2 Probadores	Media Media Media Media
MGM	5	Jefe de Módulo Arquitecto Analista 2 programadores	Alta Alta Alta Media
MED	4	Jefe de Módulo y Arquitecto Analista 2 programadores	Alta Alta Media
MRC	4	Jefe de Módulo y Arquitecto Analista 2 programadores	Media Alta Alta
MR	4	Jefe de Módulo y Arquitecto Analista 2 programadores	Media Media Media

**Tabla 22. Recursos humanos del proyecto SIGE.**

Los recursos de entorno son el engranaje que une a los recursos materiales con los recursos humanos y posibilita la creación del software. Ellos son el soporte sobre el cual se trabaja para lograr el resultado final. Se proyecta tener una idea lo mas clara posible de los que se requieren para realizar la aplicación de forma exitosa. En la siguiente tabla se muestra los software más usados en el proyecto, así como una breve descripción y la experiencia que poseen las personas que trabajan con ellos.

Nombre del Software	Descripción	Experiencia del equipo que lo utiliza.
Project 2003	Este software se utiliza para llevar a cabo la planificación del proyecto.	Alta.
Racional Rose Enterprise Edition.	Este software se utiliza para hacer el diseño del software.	Alta
Subversion	Este software se utiliza para el control de versiones.	Alta
Tortoise	Este software es el cliente que permite conectarse al repositorio del proyecto.	Alta.
SQL Server 2005	Este software se utiliza para programar la Base de datos	Alta
Visual Studio 2005	Este software se utiliza para programar los módulos.	Alta
Erwin Studio 6.0	Esta herramienta CASE se utiliza para diseñar la BD	Alta
Tier Develop	Generación de Capas de Acceso a Datos.	Alta.

Tabla 23. Software más usados en el proyecto SIGE.

## 2.4 Organización de la planificación:

Conocer el ámbito del proyecto que se planifica es vital para conocer con qué se está tratando, saber lo que se piensa hacer, y cómo quiere el cliente que se haga y en qué tiempo, y sobre todo el grado de complejidad que tiene el proyecto y por ende, la complejidad de las tareas que se van a desarrollar.

Inicialmente la planificación debe responder a una estrategia básica: La reunión inicial con el cliente. Conocer como sueña él la aplicación. Partiendo de este punto, se debe entrar a analizar entonces las acciones a seguir y sobre todo si es posible realizar lo acordado en ese plazo. Al final de cada iteración se debe hacer una reunión con el cliente, explicándole el avance del proyecto hasta el momento. Se hace una propuesta inicial sobre los nuevos artefactos en los cuales se va a enfocar el equipo de desarrollo y la fase y la iteración que se va a comenzar (o a continuar). Se le explican cuáles son los hitos fundamentales y los hitos secundarios trazados para llegar a ellos.

El cliente debe dar su aprobación, si creyese que es mucho tiempo entonces se le debe convencer mediante cifras históricas, que no es posible hacerlo en menos tiempo del propuesto y con los recursos asignados. (En cada una de estas reuniones es muy importante recalcar la influencia negativa que tiene el cambio de requisitos en fases avanzadas). Una vez que el cliente apruebe la planificación propuesta entonces los equipos de desarrollo se planifican de forma individual. Cada jefe de módulo debe conocer de antemano los hitos de esta nueva iteración, por tanto programa a su equipo para cumplirlos, para ello especifica los recursos asignados para el cumplimiento de cada tarea y el tiempo otorgado para cada una de ellas. Siempre se debe dejar una holgura del 10 por ciento en estos planes. La tarea del planificador en este caso es chequear la organización, revisar que no haya recursos abrumados de trabajo mientras que otros demasiado holgados, lograr este balance es fundamental. Los equipos de desarrollo entregarán al planificador una propuesta de la organización de su trabajo en un documento *word* y en el formato designado previamente por el proyecto. El planificador analizará la dependencia de las tareas, la cantidad de recursos materiales disponibles, así como los recursos humanos, y el tiempo que necesita cada actividad. Esta planificación será llevada entonces al Microsoft Project, indicando en el documento la fase, la iteración, el nombre del módulo al que pertenece, quiénes son las personas que deben llevar a cabo esta tarea y el tiempo previsto para la realización de cada uno de los artefactos. Esta información será luego publicada en el repositorio central del proyecto para que sea consultada y actualizada a medida que pase el tiempo y su avance pueda ser medido.



¿Por qué organizar la planificación y no simplemente planificar?

La respuesta es sencilla, la planificación necesita nutrirse de varias fuentes, nada se logra planificando algo que al final no se va a cumplir, es como decirle al Sol, que recorra el día en 8 horas cuando se sabe que son 12 horas las que necesita para esto.

La retroalimentación cliente – equipo de desarrollo – planificador es vital para el buen funcionamiento organizativo de la planificación, de qué sirve que se le especifique al cliente lo que se va a desarrollar, si al final la planificación se convertirá en una fecha arbitraria impuesta por alguien que generalmente, no conoce el esfuerzo que se debe realizar en cada una de las fases y por qué se prioriza determinado artefacto. Su principal preocupación es que el proyecto esté listo en el menor tiempo posible y con toda la calidad. Sin embargo, poco tiempo y alta calidad es sin dudas, una función inversa. Mientras menos tiempo se le otorgue a la realización de un artefacto, más tiempo se necesitará para refinarlo, lo más seguro es que haya que rehacerla un par de veces y la calidad se verá enormemente afectada en este proceso. En cambio una tarea planificada, ajustada al personal que la va a desarrollar, tiene evidentemente mayores posibilidades de salir airosa.

## 2.5 ¿Qué mecanismos de captura de información se usaron en el proyecto ONE?

La recogida de información puede ser de 2 formas: Digital o Personal

¿Que ventajas o desventajas tiene cada una de ellas?

	Forma de recogida	Seguridad	Eficiencia	Comodidad
<b>Digital</b>	La información es enviada por el responsable al planificador, es una vía más rápida y económica. Generalmente es segura, implica solo a 2 personas: el que la envía y la recibe.	La información (aunque poco probable) puede perderse si por error es enviada a otro destinatario u ocurren errores en los servidores de correos.	Es una vía rápida, confiable y muy económica.	Es mas fácil, pues la información necesaria esta a un solo clic de las partes involucradas, además de que no necesita grandes recursos para su entrega.

<b>Personal</b>	La información es recogida por el planificador directamente del responsable. Puede ser una pérdida de tiempo si ambos trabajadores se encuentran en estaciones de trabajo alejadas.	La información es entregada directamente, muy difícilmente puede haber pérdidas de información.	Es una vía lenta, confiable y puede incurrir en gastos de recursos materiales para elaborar las planillas de recogida de información.	Es muy incómodo si las estaciones de trabajo del responsable y el planificador se encuentran lejos, además de que requiere ciertos gastos de recursos materiales.
-----------------	---	---	---	---

**Tabla 24. Comparación entre las vías de recogida de información digital y personal**

En el proyecto SIGE la vía más conveniente de captura de información es la forma digital, pues es más cómoda para ser recogida y no requiere grandes gastos de recursos materiales, ni pérdidas de tiempo. Además de que toda esta información es planificada de forma digital y guardada en una carpeta en el repositorio del proyecto para acceso y consulta de otros integrantes del equipo de dirección, así como los clientes u otras personas interesadas y autorizadas.

¿Que información debe ser recogida?

- Tareas prioritarias a desarrollar en un lapso de tiempo predeterminado. .
- Tiempo de duración para de cada tarea.
- Responsables del cumplimiento de las tareas.
- Fecha de chequeo de la realización de la tarea.

A continuación se muestra un ejemplo de la plantilla para la captura de información definida por SIGE.

**Plan comprendido entre el:** fecha de inicio en el formato dd/mm/aa y **el** fecha de culminación (Este plan debe hacerse quincenal).

**Confeccionado por:** Nombre de la persona que lo elabora

- 1: Fase
- 2: Numero de iteración
- 3: Flujo de Trabajo
- 4: Actividades
- 5: Trabajadores
- 6: Artefactos
- 7: Fecha de Revisión
- 8: Fecha de entrega

1	2	3	4	5	6	7	8
Nombre de la Fase	Aquí va el numero de iteración en la que se encuentra el proyecto	Aquí va el flujo de trabajo en el que se encuentra el proyecto	Aquí va la actividad que se va a realizar. Ejemplo: Realización de casos de usos.	Aquí va el Responsable de cumplir esa tarea. Se puede agregar además el Responsable de controlar la tarea.	Aquí van los artefactos que van a ser generados durante la realización de esa actividad.	Aquí va la fecha en que esa tarea va a ser revisada. Pueden ser despachos diarios	Aquí va la fecha en que la actividad con todos los artefactos que generó se va a entregar

**Tabla 25. Modelo para la captura de la información.**

Esta plantilla debe ser llenada por el Jefe de módulo, si el proyecto está dividido en módulos, si no, se debe designar una persona encargada para llenarla y entregarla. La información (tareas a realizar), se las debe dar el Líder del proyecto a cada módulo o persona encargada de llevar a cabo las mismas. Estas tareas deben ser planificadas previamente por los módulos o equipos designados para llevar a cabo las tareas. Posteriormente el planificador analizará cada una de las propuestas y finalmente ajustará el tiempo designado para cada equipo de trabajo otorgándole a cada tarea, un lapso de tiempo necesario, un responsable de llevarla a cabo, una fecha de revisión, así como la cantidad de recursos que tiene disponibles para cumplimentarla. Esta forma de planificación es muy importante, pues permite la retroalimentación de los equipos de software. La planificación en sí se convierte en un acuerdo realizado

por el planificador y los equipos de trabajo. Ojo a la hora de planificar, es primordial dejar siempre una holgura del 10 % dentro del plan.

## **2.6 Mecanismos de seguimiento de la información:**

Es muy necesario verificar la realización de la planificación, o sea, rastrear el cumplimiento del tiempo asignado para cada tarea. Es recomendable establecer un modo por módulo que permita al Jefe de módulo chequear todo lo que se ha realizado de forma diaria. ¿Por qué en tan poco tiempo? Mientras menos tiempo se tarde en conocer qué no está funcionando o qué ha adquirido cierto grado de atraso con respecto a lo que se había planificado, más fácil y eficiente es tomar medidas en caso de un posible atraso, incumplimiento o riesgo de no poder realizar la tarea o al menos no poder realizarla en el tiempo definido.

¿Cómo lograr que se realice este chequeo?

Cada Jefe de módulo debe definir un horario al final del día y hacer un resumen de las tareas que se han realizado, el grado de evolución de las mismas y el tiempo requerido para cada una de ellas. Esta información será publicada en el repositorio del proyecto en una carpeta habilitada por el personal de planificación. De este modo el Líder del proyecto podrá consultar esta información diariamente y el equipo de planificación podrá medir el avance. Que el proyecto sea chequeado de forma diaria, permite tener un gran radio de acción para mitigar algún riesgo por atraso en caso de que el equipo de trabajo no esté cumpliendo con las fechas asignadas.

¿Por qué? Este es el tiempo de recuperación en caso de que un riesgo se volviera realidad, es el tiempo de holgura para volver a coger el ritmo y salir ilesos de cualquier contingencia que se presente. Esta planificación se realizará con la herramienta escogida para la misma y luego será publicada en el repositorio central del proyecto para que pueda ser consultada por el líder de proyecto, los integrantes de los equipos de desarrollo, el equipo de calidad etc., para que puedan ser establecidas las fechas de revisión de los artefactos generados.

## **2.7 Mecanismos de control de la información:**

Semanalmente el equipo de dirección de proyecto debe reunirse y verificar el avance de los equipos de desarrollo. El equipo de calidad y prueba debe estar presente. Para este encuentro el planificador

debe concebir un plan de reuniones en el que se especifique el día del contacto así como los artefactos que serán chequeados. Este plan debe ser publicado en el repositorio del proyecto o enviado a todos los interesados por correo. Al terminar este control el planificador debe documentar el estado de los equipos de desarrollo. A las tareas se les debe de clasificar de Correctas (OK) si están realizadas en el plazo previsto. Atrasadas (AT) si están atrasadas, pero aún se pueden terminar en la fecha prevista, Alerta (AL) si ni siquiera se han comenzado. Este documento es vital para poder conocer los equipos que han incumplido las fechas de entrega, el por qué ha sucedido esto y que riesgo corren los mismos.

En caso de que existiera algún peligro, este documento es vital para analizar las estrategias de mitigación de riesgos.

Además de esto cada equipo de desarrollo debe emitir un parte establecido internamente por el proyecto, en aras de controlar y medir el avance interno. En este parte, desarrollado en formato Excel y modificado del parte general de la facultad 3 para el control de los proyectos y polos productivos, se miden la cantidad de hitos propuestos semanalmente, y cuántos de ellos han sido cumplidos. Además de la cantidad de Puntos de control que fueron establecidos para controlar su realización, prestándole una especial atención a los elementos de configuración en sus diferentes fases: desarrollo, revisión, mejoras de revisión y terminados. Lo más importante de la utilización del mismo es que contribuyó a medir el avance del proyecto desde el punto de vista de los hitos alcanzados. Este debía ser entregado semanalmente, el equipo que no lo entregara automáticamente incurría en un incumplimiento que afectaba su evaluación dentro del proyecto. La fomentación de actividades extracurriculares era además muy importante pues promovía el trabajo en equipo, que era una de las principales políticas del proyecto.

La siguiente figura muestra las principales tareas que tenía en cuenta este mecanismo para controlar la información.

Presentación de trabajos	Taller, seminario científico, conferencia científica, etc.	Unidad de medida
		Nombre de la actividad
Participación con resultados concretos y calidad en los proyectos productivos.	Hitos del período semanal	Numero de hitos
	Puntos de control para lograr los hitos semanales	Numero de hitos
	Cantidad de puntos de control cumplidos.	Numero de hitos
	Cantidad de hitos cumplidos.	Numero de hitos
	Cantidad de elementos de configuración en desarrollo.	Total de elementos
	Cantidad de elementos de configuración en revisión	Total de elementos
	Cantidad de elementos de configuración en mejoras de revisión.	Total de elementos
	Cantidad de elementos de configuración terminados.	Total de elementos
	Razón de casos de uso terminados.	Cantidad del total
	Razón de clases terminadas.	Cantidad del total
	Cursos optativos impartidos en el proyecto.	Nombre del curso
Actividades extracurriculares	Actividades deportivas y las actividades extracurriculares.	Nombre de la actividad

**Figura 7. Parte para controlar el avance interno del proyecto.**

## 2.8 Plan de fases e iteraciones.

Para realizar este Plan de Fases es vital tener en cuenta que tiempo se le va a asignar a cada fase y cuáles son los principales hitos que tendrá cada una de ellas, así como los hitos secundarios, que son aquellos que posibilitan el cumplimiento de los hitos principales. Es menester profundizar en la dependencia hitos secundarios-hitos principales, pues priorizando el desarrollo de los primeros es que se alcanzan al fin los hitos que marcan el final de la iteración.

Es muy importante llevar a consideración la cantidad de iteraciones que debe tener cada fase.

Para evaluar cada iteración es muy acertado proponer criterios de evaluación, si la primera iteración de inicio logró que el cliente y los desarrolladores se expresaran en un mismo lenguaje entonces esta fase tuvo un final satisfactorio, si en la Primera fase de Elaboración se elaboró la arquitectura candidata, y en la segunda iteración, luego de profundos refinamientos se logró establecer la Línea Base de la arquitectura, entonces esta Fase también tuvo resultados satisfactorios. Estos criterios de

evaluación pueden ser tomados como otra forma de medir el avance del proyecto, pues mediante ellos se puede medir qué requisitos han sido desarrollados de forma eficiente.

En la figura 2 se puede observar como RUP sugiere la realización de más de una iteración por cada fase, excepto quizás en la fase inicial.

Fase	Número de Iteraciones	Duración
Fase de Inicio	2	3 mes
Fase de Elaboración	2	5 meses
Fase de Construcción	2	6 meses
Fase de Transición	1	1 mes

**Tabla 26. Plan de fases del proyecto SIGE**

Cada fase se caracteriza por una cierta cantidad de hitos que deben ser cumplidos al finalizar la misma. La siguiente tabla muestra los principales hitos determinados para cada fase del proyecto SIGE. Según las iteraciones planificadas para cada una de estas fases, estos hitos podrán ser refinados para completar un producto final de alta calidad.

La siguiente tabla muestra el plan de fases del proyecto SIGE:

Descripción	Hito
Fase de Inicio	<p>En la Primera Iteración de esta Fase se realizan las primeras reuniones con el cliente en aras de entender el negocio, se hace el levantamiento de requisitos, se comienzan a idear las funcionalidades críticas del sistema. Se Elaboran los Casos de Usos Críticos y a partir de ellos se desarrolla una primera Estimación del proyecto. Se traza una arquitectura provisional. Se Elabora el Plan de proyecto.</p> <p>En la Segunda iteración se hará un refinamiento de los requisitos levantados y se identifican la mayoría de los casos de uso, Se refina el Plan de Proyecto. Se llega a un consenso sobre las funcionalidades del sistema y se elabora un documento oficial que es firmado por ambas partes. Se elabora un prototipo no funcional, Se identifican los riesgos críticos. Se comienza a elaborar el Plan de Riesgos.</p>



Fase de Elaboración	<p>En la Primera Iteración de esta fase se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y/o críticas del sistema). Se realizan las 5 vistas arquitectónicas. Se hace además un modelo preliminar del Modelo de Análisis/Diseño Tiene como resultado la línea base de la arquitectura. Se estima nuevamente, esta vez con datos más precisos.</p> <p>En la Segunda iteración de esta fase, todos los casos de uso que serán implementados en la primera iteración de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis y/o Diseño). Se hace una reestimación si es necesario de la planificación para asegurar el cumplimiento de los objetivos. La segunda iteración se enmarca en el refinamiento de la primera iteración así como a la elaboración de artefactos.</p>
Fase de Construcción	<p>En la primera iteración de esta fase se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis/Diseño. Esta fase exige tantas iteraciones como sean necesarias. Se comienza la elaboración de material de apoyo al usuario. Se articula por fin el sistema.</p> <p>En la segunda iteración se hace un refinamiento, se agregan funcionalidades y se reelabora la documentación.</p>
Fase de Transición	<p>Se hace entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto. La versión Beta del producto esta lista para su entrega. Se hacen pruebas al software y se documentan los defectos y deficiencias.</p>

**Tabla 27. Descripción de los hitos fundamentales por cada fase**

Es importante recalcar que la planificación de las fases se refina a medida que se pasa por las iteraciones de cada una de ellas. Podemos enmarcar esto entonces en la planificación de las iteraciones. De esta forma, el proyecto no se planifica entero desde principio a fin, sino poco a poco, tomando siempre como punto de partida la iteración anterior. Prácticamente es por gusto planificar la fase de Construcción si antes no se ha planificado la Fase de elaboración, en cambio si se puede ir adelantando algo, que daría una guía de cómo enfrentar la nueva Fase. Recuérdese siempre que una

iteración comienza cuando la otra esta concluyendo, en este lapso de tiempo es necesario tener prevista la planificación de la próxima iteración. Los equipos de desarrollo pueden colaborar mucho para realizar esta planificación.

## **2.9 Gestión de Riesgos**

Estos son los riesgos identificados en el proyecto SIGE, clasificados de acuerdo a su naturaleza y expresada su probabilidad de ocurrencia y probabilidad de impacto.

### **2.9.1 Lista Inicial de riesgos detectados**

#### Planificación

1. La planificación de los recursos y del producto ha sido impuesta por el cliente o un directivo superior. No se ha hecho un estudio de las mismas.
2. La planificación se ha basado en la utilización de personas específicas de un equipo, pero estas personas no están disponibles.
3. Planificación ambiciosa del calendario de trabajo
4. No se puede construir un producto de tal envergadura en el tiempo asignado.
5. El producto es más grande que el estimado.
6. El esfuerzo es mayor que el estimado.
7. La reestimación debida a un retraso en la planificación es demasiado optimista o ignora la historia del proyecto.
8. La presión excesiva en la planificación reduce la productividad.
9. Las áreas desconocidas del producto llevan más tiempo del esperado en el diseño y en la implementación.
10. Dependencia de las tareas.

Riesgos de Administrativos o de Entorno:

11. El lugar (laboratorio) donde se trabaja en el proyecto es ruidoso, no esta climatizado y sus condiciones son malas.
12. La curva de aprendizaje para la nueva herramienta de desarrollo es más larga de lo esperado
13. El ciclo de revisión/decisión de la directiva es más lento de lo esperado.
14. La dirección toma decisiones que reducen la motivación del equipo de desarrollo.
15. La dirección pone más énfasis en las heroicidades que en informarse exactamente del estado, lo que reduce su habilidad para detectar y corregir problemas.
16. Límites legales

Personal:

17. Los miembros del equipo no se implican en el proyecto, y por lo tanto no alcanzan el nivel de rendimiento deseado.
18. Alguien de la plantilla abandona el proyecto antes de su finalización.
19. La incorporación de nuevo personal de desarrollo al proyecto ya avanzado, reducen la eficiencia de los miembros del equipo existentes.
20. Los miembros del equipo no trabajan bien juntos.
21. Inexperiencia del personal, que produce carga laboral mal distribuida.
22. Mala comunicación entre los equipos de desarrollo.

Específicos:

23. Se añaden requisitos extra.
24. Diseño inadecuado del producto.
25. Se escatiman los procesos de calidad
26. Falta de control de configuración.
27. Ante los retrasos se añade más personal.
28. No se documentan las revisiones técnicas
29. No hay ayuda en línea y escasa documentación.

30. El proyecto no se hace sobre la base de un repositorio
31. Construir un producto o sistema excelente que no quiere nadie en realidad
32. Escasa calidad de la documentación entregada al cliente
33. No se usan métodos específicos para el diseño de pruebas.
34. No se lleva cuaderno del Ingeniero.
35. Mala comunicación con el cliente.

Tecnológicos:

36. Las herramientas de desarrollo no trabajan como se esperaba; el personal de desarrollo necesita tiempo para resolverlo o adaptarse a nuevas herramientas.
37. Las herramientas de desarrollo no se han elegido en función de sus características técnicas, y no proporcionan las prestaciones previstas.
38. Ataques por virus.
39. Ataques por hackers o crackers.
40. Afectaciones del fluido eléctrico.

Naturales:

41. Olas de calor
42. Epidemias
43. Erupciones volcánicas
44. Terremotos
45. Huracanes
46. Incendios
47. Tormentas
48. Descargas eléctricas

La siguiente tabla muestra la clasificación de los riesgos de acuerdo a la Matriz de Probabilidad e impacto (MPI)

ID. de riesgo	Probabilidad	Impacto	Efectos	Prioridad según la MPI
1	0.1	0.80	Marginal	2
2	0.5	0.20	Marginal	2
3	0.1	0.20	Despreciable	3
4	0.3	0.20	Marginal	2
5	0.7	0.40	Critico	1
6	0.1	0.20	Despreciable	3
7	0.5	0.20	Marginal	2
8	0.3	0.80	Critico	1
9	0.5	0.40	Critico	1
10	0.9	0.40	Critico	1
11	0.5	0.40	Critico	1
12	0.5	0.40	Critico	1
13	0.7	0.20	Marginal	2
14	0.7	0.40	Critico	1
15	0.3	0.80	Critico	1
16	0.1	0.80	Marginal	2
17	0.5	0.40	Critico	1
18	0.7	0.10	Marginal	2
19	0.3	0.40	Marginal	2
20	0.5	0.80	Critico	1
21	0.9	0.40	Critico	1
22	0.5	0.40	Critico	1
23	0.7	0.80	Marginal	2
24	0.5	0.80	Catastrófico	1
25	0.7	0.80	Critico	1

26	0.5	0.80	Critico	1
27	0.3	0.80	Catastrófico	1
28	0.5	0.40	Catastrófico	1
29	0.5	0.20	Marginal	2
30	0.5	0.80	Catastrófico	1
31	0.1	0.80	Marginal	2
32	0.5	0.40	Catastrófico	1
33	0.3	0.40	Marginal	2
34	0.5	0.80	Critico	1
35	0.5	0.80	Catastrófico	1
36	0.3	0.20	Marginal	2
37	0.1	0.20	Despreciable	3
38	0.7	0.80	Catastrófico	1
39	0.1	0.80	Marginal	2
40	0.5	0.40	Crítico	1
41	0.1	0.05	Despreciable	3
42	0.3	0.10	Despreciable	3
43	0.1	0.10	Despreciable	3
44	0.1	0.10	Despreciable	3
45	0.5	0.20	Marginal	2
46	0.3	0.40	Marginal	2
47	0.7	0.10	Marginal	2
48	0.7	0.10	Marginal	2

**Tabla 28. Prioridad de los riesgos encontrados en el proyecto SIGE.**

Fueron encontrados 23 Riesgos con Prioridad 1 categorizados en Críticos y Catastróficos, 18 riesgos marginales y 7 riesgos despreciables.

### 2.9.2 Planificación de Respuesta al riesgo (Contingencia del Riesgo)

**ID de Riesgo:** 5 (El producto es más grande que el estimado)

**Indicadores:** El equipo de desarrollo sobrepasa la fecha de fin establecida y aún falta mucho por hacer.

**Plan de contingencia:** Estimar por varios métodos, estudiar datos históricos, estudiar el software.

**Estrategia a seguir:** Evitar

**ID de Riesgo:** 8 (La presión excesiva en la planificación reduce la productividad.)

**Indicadores:** El equipo de desarrollo se siente agobiado por el trabajo y el escaso tiempo para cumplirlo

**Plan de contingencia:** Planificar con una holgura del 10%. Hacer una planificación retroalimentada.

**Estrategia a seguir:** Evitar

**ID de Riesgo:** 9 (Las áreas desconocidas del producto llevan más tiempo del esperado en el diseño y en la implementación.)

**Indicadores:** El equipo de desarrollo se demora más de lo previsto en esas fases.

**Plan de contingencia:** Ajustar la planificación a las nuevas necesidades.

**Estrategia a seguir:** Mitigar

**ID de Riesgo:** 10 (Dependencia de las tareas.)

**Indicadores:** Algunas tareas se retrasan en espera del cumplimiento de otras.

**Plan de contingencia:** Darle prioridad a las tareas críticas y darle más control además

**Estrategia a seguir:** Evitar

**ID de Riesgo:** 11 (El lugar (laboratorio) donde se trabaja en el proyecto es ruidoso, no está climatizado y sus condiciones son malas.)

**Indicadores:** El equipo de desarrollo no puede trabajar cómodamente, se distrae y no cumple con los horarios de trabajo establecidos.

**Plan de contingencia:** Buscar recursos para arreglar las malas condiciones

**Estrategia a seguir:** Aceptar

**ID de Riesgo:** 12 (La curva de aprendizaje para la nueva herramienta de desarrollo es más larga de lo esperado)

**Indicadores:** El equipo no puede trabajar porque no domina las herramientas de trabajo.

**Plan de contingencia:** Trabajar con herramientas que sean conocidas por el equipo de trabajo.

**Estrategia a seguir:** Evitar

**ID de Riesgo:** 14. (La dirección toma decisiones que reducen la motivación del equipo de desarrollo.)

**Indicadores:** Los equipos no están de acuerdo con nuevas reglas y no están trabajando con el mismo entusiasmo

**Plan de contingencia:** Proponer nuevos métodos, nunca agregarlos arbitrariamente.

**Estrategia a seguir:** Evitar

**ID de Riesgo:** 15 (La dirección pone más énfasis en las heroicidades que en informarse exactamente del estado, lo que reduce su habilidad para detectar y corregir problemas.)

**Indicadores:** Inconformidades en el equipo de desarrollo. Problemas no detectados

**Plan de contingencia:** Control estricto y de forma periódica

**Estrategia a seguir:** Evitar

**ID de Riesgo:** 17 (Los miembros del equipo no se implican en el proyecto, y por lo tanto no alcanzan el nivel de rendimiento deseado.)

**Indicadores:** Los equipos de desarrollo andan disociados y no producen lo planificado

**Plan de contingencia:** Hacer actividades motivadoras.

**Estrategia a seguir:** Mitigar

**ID de Riesgo:** 20 (. Los miembros del equipo no trabajan bien juntos.)

**Indicadores:** Problemas internos en los equipos.

**Plan de contingencia:** Actividades para fomentar la unión de los equipos de desarrollo

**Estrategia a seguir:** Mitigar



**2.10 Conclusiones del Capítulo.**

Luego de la estimación de los módulos del proyecto SIGE por 2 métodos diferentes de estimación, se determinó que es un proyecto factible. No hay dependencia entre los módulos, por lo que se escogió el mayor tiempo de vida de los mismos para establecerlo como base para la duración total del sistema. A partir de estos cálculos se estimó la cantidad de recursos materiales y humanos necesarios. Se definieron los mecanismos para la captura, seguimiento y control de la información. Se hizo alusión al Plan de Gestión de riesgos, dando una amplia medida de los principales riesgos que podrían afectar al proyecto, su prioridad y las formas de enfrentarlos.



## *Capítulo* *Análisis de Resultados*

### **3.1 Introducción**

En este capítulo se pretende hacer un análisis de los resultados obtenidos, luego de aplicar los diferentes procedimientos propuestos vinculados a la Gestión de la Planificación del Proyecto SIGE. Se enfatiza en el cumplimiento de las tareas, el éxito de los mecanismos establecidos para la captura, seguimiento y control de la información. La factibilidad en la utilización de la herramienta Microsoft Project 2003 y el grado de acercamiento de la estimación efectuada contra el desarrollo real del proyecto. Además, se muestra la influencia de los riesgos presentados, durante el desarrollo del producto.

### **3.2 Análisis de Resultados**

Con el fin de conocer el estado de la planificación en otros proyectos productivos de la facultad, se realizó una encuesta a los planificadores, obteniendo los resultados que se detallan más adelante.

Este es el cuestionario sobre el cual se obtuvieron las informaciones:

1. Nombre del proyecto:
2. ¿Se realizó desde el inicio la estimación del proyecto para determinar la cantidad de recursos necesarios?

En caso positivo

¿Se adecuó a las necesidades del proyecto?

En caso negativo

- ¿Ya las personas estaban determinadas al igual que los recursos materiales?
3. ¿El mecanismo para la recogida de información resultó satisfactorio?
  4. ¿El proyecto considera la planificación como algo que lo ayuda a trabajar con eficiencia?
  5. ¿Se ajustan a la planificación efectuada?
  6. ¿En el proyecto se realiza una buena gestión de riesgos?

La realización de la estimación desde la fase de Inicio, para determinar la cantidad de recursos necesarios para el desarrollo del proyecto, solo uno de los encuestados la hizo, en el tiempo mencionado anteriormente, obteniendo resultados muy alejados de la realidad. (Ver Figura 8). Evidentemente esta primera evaluación muy pocas veces se corresponde al tiempo real de duración, pero al menos ofrece una ligera idea de la cantidad de tiempo y personas que va a requerir el sistema. En los demás proyectos las personas y los recursos materiales fueron asignados sin haberse efectuado estimación alguna.

El mecanismo de recogida de información resultó ser muy positivo en todos los proyectos que lo utilizaron. Algo que resultó curioso es que en el 100 % de los proyectos, no consideran la planificación como algo que los ayuda a trabajar eficientemente. Craso error entre los desarrolladores de software, que piensan que simplemente pueden hacer y deshacer sin regirse por una planificación, estos son los que luego se enfrascan en proyectos grandes, siguen sin planificar su trabajo y el resultado final es un proyecto fracasado. Este pensamiento constituye el principal motivo de que muy pocos proyectos se ajusten a la planificación hecha.

En el 75 % de los proyectos encuestados se realizó una buena Gestión de Riesgos. ¡Bravo por ellos! La realización de una buena Gestión de riesgos es fundamental para el éxito del proyecto, a pesar de que otras cosas atenten contra él. Estos proyectos tienen grandes posibilidades de poder enfrentar cualquier conflicto que se les venga encima, si se han detenido a analizar cada uno de los posibles riesgos, que probabilidad de impacto tienen sobre el proyecto en cuestión y si ocurriesen que tamaño tendría este desastre.



**Figura 8. Por ciento de proyectos encuestados que realizaron la estimación desde la Fase de Inicio.**

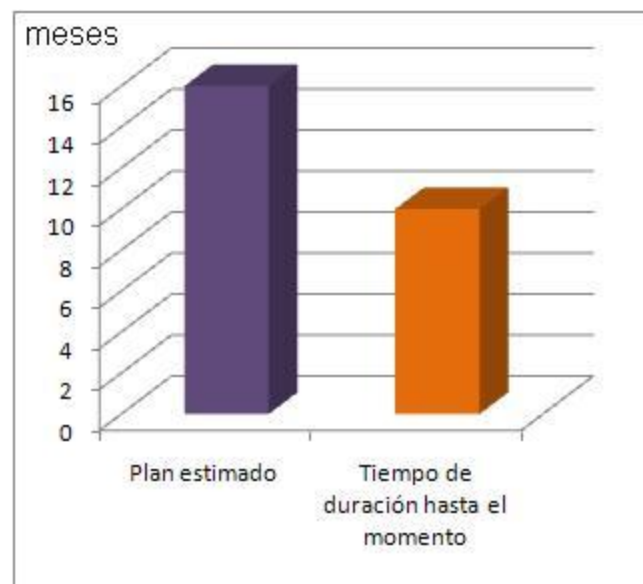
El proceso de estimación inicial en la facultad 3 es deficiente pues no se realiza en la fecha que se debe realizar (en la Fase Inicial del proyecto), simplemente se agregan estudiantes y profesores a los proyectos sin tener en cuenta la cantidad real de recursos humanos que se necesitan, ni los recursos materiales, que pueden ser computadoras, impresoras u otros materiales necesarios para la gestión y desarrollo del proyecto en cuestión. Esto trae como consecuencias que a veces haya estudiantes en exceso en los proyectos y que estén ociosos dentro de los mismos y contando con que el número de computadoras generalmente es limitado, a veces las mismas estén sobre utilizadas. Puede ocurrir que los recursos materiales y humanos sean deficientes y por tanto existe un gran riesgo de que la fecha de entrega pactada no sea cumplida, se debe recordar que los recursos humanos son personas, que necesitan alimentarse, descansar etc. para rendir eficientemente en el trabajo, no se le puede pedir al olmo peras.

### 3.3 Estimación en el proyecto SIGE

Antes de evaluar el resultado de la estimación es muy importante tener en cuenta que la misma trabaja con valores aproximados, e inicialmente ofrece un marco de apoyo para la posible duración del proyecto. A partir de la mayor estimación para los módulos del proyecto SIGE, se estimó la duración total del proyecto para 15 meses por el método de estimación de Puntos de Casos de Uso, añadiéndole a esto 1.5 meses que representa el tiempo de holgura (40 días aproximadamente), porque no había ninguna dependencia entre los módulos de desarrollo.

El comportamiento entre la estimación realizada y la duración real del proyecto hasta se ha mantenido exitoso.

En la figura se muestra el comportamiento del tiempo estimado y el transcurrido.



**Figura 9. Comportamiento de la duración del proyecto.**

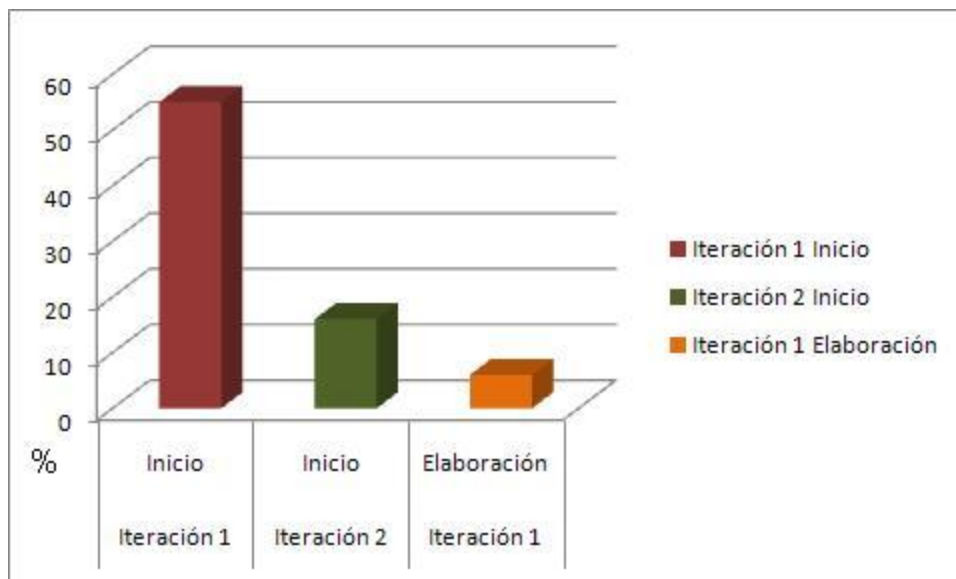
Esta diferencia tiene su base en diversos factores: Uno de ellos es que los estudiantes no se dedican tiempo completo a la producción, sino que deben cumplir otras responsabilidades.

La valoración por 2 métodos diferentes, permitió hacer una comparación entre los resultados de ambos, pudiéndose apreciar que a pesar de que diferían, en la mayoría de los casos, mantenían una diferencia muy escasa. El método que según sus cálculos se acercó a la duración del proyecto fue Puntos de Casos

de Usos, seguramente porque tiene en cuentas aspectos importantes e influyentes en el desarrollo del software, como lo son los factores de ambiente y la complejidad técnica, etc.

A partir de esta estimación realizada, se proyectó la cantidad de recursos humanos y materiales que debía tener el proyecto. El uso de las computadoras por el personal al cual correspondían tuvo un ascenso significativo desde la primera iteración, hasta la presente. De 18 computadoras asignadas al proyecto, en la primera iteración casi 10 computadoras permanecían cierto tiempo sin ser utilizadas. En la segunda esta cifra disminuyó a solo 3 computadoras y el salto de avance cuando ocasionalmente hubo 1 computadora sin ser usada en la Fase de Elaboración.

La siguiente figura muestra el gran salto que se dio en el aprovechamiento de los recursos materiales.



**Figura 10. Aumento significativo del uso de las PC en el proyecto SIGE**

### 3.4 Mecanismos de captura, seguimiento y control de la información.

El mecanismo de captura de información que se definió, cumplió cabalmente sus objetivos porque logró una máxima eficiencia al planificar las tareas. La retroalimentación planificador – equipo de trabajo fue muy importante porque cumplió con que el equipo de trabajo no se sintiera abrumado con las tareas que debía realizar, sino que comprendiera que podía cumplirlas a tiempo, pues el mismo estudió y estableció

el tiempo que le tomaría culminarlas con la calidad esperada. La vía digital establecida para recolectar la información fue eficiente porque siempre fue recogida a tiempo, de una forma cómoda, sin ningún gasto de recursos materiales. Fue guardada en el repositorio del proyecto, lo que posibilitó su protección en cuanto a pérdidas o extravíos, además de que cualquier interesado podía consultarla y expresar su opinión en cuanto a la totalidad de tiempo otorgado a cada tarea por parte de los equipos de desarrollo.

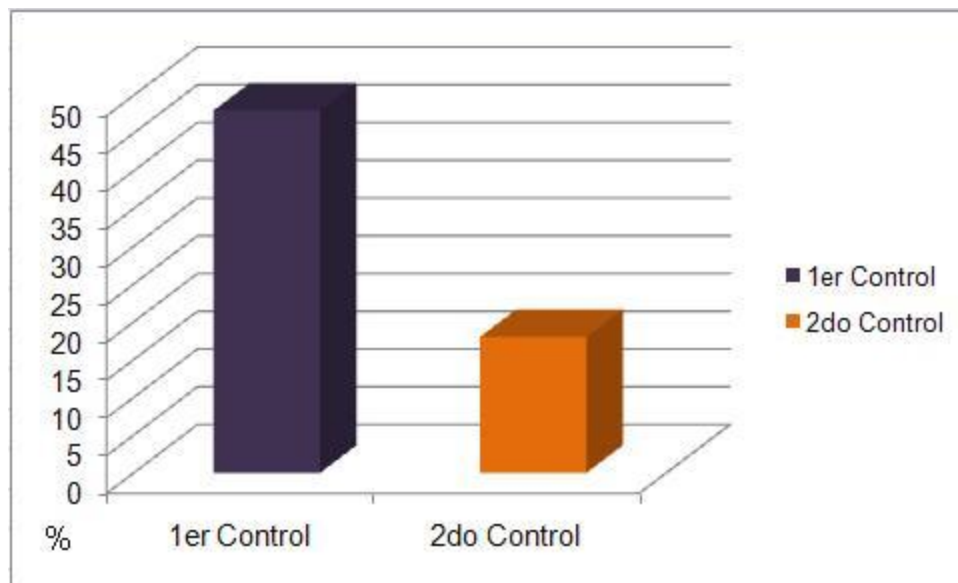
El seguimiento de la información diaria no tuvo buenos resultados, porque los jefes de módulo aún no tienen la conciencia necesaria para hacerlo. Los demás integrantes del equipo tampoco les brindaron mucha ayuda.

El mecanismo establecido para controlar la información resultó ser bastante satisfactorio. El parte establecido para el control fue muy eficiente porque permitió conocer el estado de los hitos planteados, establecer el por ciento de hitos cumplidos, conocer y contabilizar todos los artefactos que estaban en revisión y los que una vez revisados estaban en mejoras de revisión. Gracias a este control fue posible contabilizar el atraso de un 48 % en la primera revisión de la Fase de Inicio, que puso en marcha un plan para poder culminar en el tiempo establecido. El segundo detectó un leve atraso de un 18 %. (Ver figura 11) El por ciento de atraso por revisión fue calculado teniendo en cuenta la fórmula:

$$\frac{\text{Cantidad de artefactos}}{\text{Total de artefactos a entregar}} * 100$$

La emulación interna entre los módulos del proyecto posibilitó el desarrollo de actividades extracurriculares que fueron reflejadas en el parte lo que constituyó una estrategia interna para fomentar la unión entre el personal del proyecto.

Las reuniones previstas no se realizaron en el tiempo propuesto, pues a pesar de que había mucho trabajo, los equipos las consideraban como algo tedioso que les restaba tiempo de faena, para ellos con tan solo un día de diferencia no era posible medir el avance del proyecto. La cotidianidad ofreció otros resultados, el avance continuo en cuanto a código, nuevos modelos, etc. dieron al traste con estas concepciones y se concibieron reuniones al menos 2 veces a la semana.



**Figura 11. Atrasos detectados por los controles realizados.**

### 3.5 Organización de la planificación

La planificación de un proyecto de software que toma la opinión de los 2 principales implicados: cliente – equipo de desarrollo y logra llegar a un acuerdo es sin dudas una planificación exitosa. Esta forma de organizar la planificación logró un contacto cercano con el cliente y el quipo de desarrollo, esta forma de planificar fue cumplida sin variaciones en cada una de las iteraciones del proyecto. Fue una forma de tocar de cerca la visión del cliente y adaptarla a las necesidades del proyecto y de las personas que lo desarrollaban. Se convirtió en una vía para mantener al cliente informado de las transformaciones que ocurrían dentro del proyecto, de las actividades que se desarrollaban para lograr los objetivos propuestos y sobre todo para confrontar la opinión del equipo de desarrollo y el cliente en cuanto al tiempo de duración de cada tarea. Aunque en cuanto a esto último, nunca existieron grandes divergencias.



### 3.6 Planificación

La planificación comienza cuando se concibe el proyecto y solo termina cuando se entrega el producto y se recibe la aceptación del cliente. Se realiza una planificación inicial y se continúa con una más detallada, que se obtiene a partir de los datos históricos que se van acumulando en el proyecto y a medida que se van conociendo las características personales de los individuos.

El proyecto ONE se dividió inicialmente en 4 módulos o mini proyectos, siguiendo las reglas de RUP que plantean que un proyecto de software grande es conveniente dividirlo en proyectos más pequeños y por ende más fáciles de ser manejados. Cada uno de ellos poseía objetivos específicos a desarrollar a pesar de unirse todos para formar el Sistema de Gestión Estadística.

¿Se ajustaron estos módulos a la planificación hecha?

Lamentablemente la respuesta inicial es no, aunque con el transcurso del tiempo el ajuste a la planificación hecha fue esencial para el buen funcionamiento del proyecto SIGE

¿Qué consecuencias trajo esto?

La planificación del proyecto SIGE en sus inicios no fue satisfactoria, pues el planificador no contaba con la experiencia suficiente para ajustarla a las necesidades del proyecto y los trabajadores del mismo, no existían datos históricos, el personal era nuevo y no se conocía entre si. El trabajo en equipo apenas comenzaba a fluir, harían falta tiempo de entrenamiento para adaptarse a todo el trabajo que se venía encima. Aprender a ajustarse a la planificación hecha y sobre todo tomarla en cuenta como algo necesario para el buen funcionamiento y la marcha sana del trabajo a realizar.

A toda esta novatada se le agregó la creencia general por parte de los equipos de desarrollo de que la planificación en si constituía solo una pérdida de tiempo, que era posible cumplir con las metas propuestas solamente trabajando cuando ellos entendían. Esto provocó un atraso muy serio en la primera reunión con el cliente donde se pretendía analizar los resultados hasta la fecha.

Algo imperdonable fue que no se enfatizó en que el personal del proyecto llevara un Cuaderno del Ingeniero en aras de registrar los datos históricos del rendimiento personal.

Se logró el cumplimiento de las iteraciones propuestas, aunque fue necesario readaptar la planificación temporal un par de veces y sobre todo dar más tiempo al refinamiento de artefactos y requisitos.

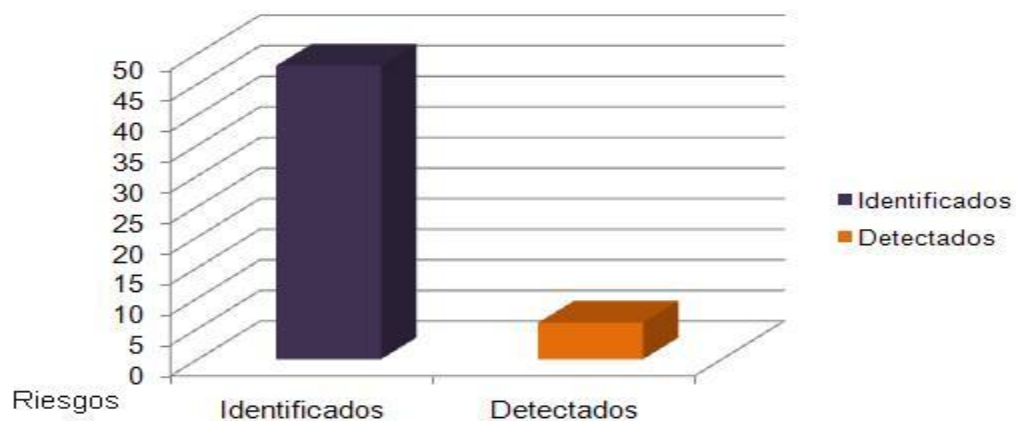
### 3.7 Gestión de riesgos.

De 48 riesgos identificados, solo 6 tuvieron incidencia en el proyecto, lo que representa un 12.5 % del total, sin embargo, sus consecuencias no fueron significativas, gracias a una adecuada gestión de riesgos, así como a un seguimiento y valoración acertada.

Los riesgos a los que se enfrentó el proyecto fueron:

1. La reestimación debida a un retraso en la planificación es demasiado optimista o ignora la historia del proyecto. (Detectado o en la primera iteración de la Fase de Inicio)
2. Las áreas desconocidas del producto llevan más tiempo del esperado en el diseño y en la implementación.( Detectado en la Iteración 1 de la Fase de Elaboración)
3. El lugar (laboratorio) donde se trabaja en el proyecto es ruidoso, no esta climatizado y sus condiciones son malas.( Detectado en la iteración 2 de la Fase de Inicio)
4. Se añaden requisitos extra. (Detectado en la iteración 1 de la Fase de Elaboración)
5. Escasa calidad de la documentación entregada al cliente. (Detectado en la iteración 1 de la Fase de Inicio)
6. No se lleva cuaderno del Ingeniero (Detectado desde la iteración 2 de la Fase de Inicio )

A continuación se muestra una figura con que permite comparar la cantidad de riesgos planificados contra los riesgos que incidieron de algún modo en el proyecto.



**Figura 12. Comportamiento de los riesgos en el proyecto**

Estos resultados reflejan que el proyecto logró una buena gestión de riesgos, pues aquellos que lo golpearon no representan ni siquiera  $\frac{3}{4}$  del total. Si embargo, es necesario asegurar aún más la calidad del proceso de detección y seguimiento de riesgos. Mientras más énfasis se haga en esto, muchos dolores de cabeza pueden ser evitados.

### **3.8 Conclusiones del capítulo**

Fue posible hacer un análisis de los resultados obtenidos, sobre todo llegar a la conclusión de cuál fue el método de estimación que se ajustó más a las características del proyecto y que factores posibilitaron esto. Cómo fue el comportamiento de la estimación y de la duración real del proyecto. La influencia tuvieron los mecanismos de captura, seguimiento y control de la información dentro del proyecto. Las tendencias dentro de los equipos de desarrollo en cuanto a la planificación establecida y sobre todo el ajuste a la misma. Como evolucionó el proceso de planificación dentro de SIGE.

Se determinó Puntos de Casos de Usos, como el método de estimación que más se ajustó a las características reales del proyecto SIGE, de acuerdo a las cifras obtenidas en el transcurso de la planificación.

## CONCLUSIONES GENERALES

Una buena planificación es posible. Solo se necesita el enfoque en diversos aspectos que pueden atenuar la eficiencia de la misma. El principal problema que tiene la planificación hoy en la facultad 3 es que no hay una cultura de planificación. Los estudiantes e incluso algunos profesores vinculados a proyectos desconocen la importancia de la misma, y por tanto le restan importancia frente a otras cuestiones. Es necesario crear la conciencia de aprender a planificar y a planificarnos, porque esto no solo sirve para que sea utilizado en los proyectos de software a los que inevitablemente se estará asociado algún día, sino para la vida diaria y agitada que se lleva en la UCI. Aprender a planificar el estudio, el proyecto e incluso la vida personal es un paso de avance para otorgar a las tareas cotidianas la prioridad y el tiempo que realmente requiere, pudiendo utilizar este en otras tareas que lo necesiten. Se logró una estimación bastante exitosa. Se ganó bastante en cuanto a la conciencia de los integrantes del proyecto, Se definieron mecanismos de captura, seguimiento y control de la información que funcionaron de forma triunfante.

Tras este estudio realizado fue posible concluir cuales son las principales tareas de la planificación.

## RECOMENDACIONES

Los proyectos de la facultad se rigen por el TSP (Team software Process) y el PSP (Personal Software Process), sin embargo en ninguno se lleva Diario del Proyecto o Cuaderno del ingeniero.

Es recomendable llevar un Diario de proyecto, que permita registrar el tiempo que se dedica a cada actividad para poder refinar continuamente la planificación. El cuaderno del ingeniero en cambio posibilita registrar el tiempo dedicado a cada artefacto pero de forma personal, no en equipo y esto constituye un dato histórico vital para la planificación.

El uso de un Cuaderno del Ingeniero permite registrar las actividades que se hacen de forma diaria, el tiempo que se le dedica a cada una, así como la frecuencia con la que son realizadas. ¿Qué resultados se obtienen con esto? Una planificación que se especialice en el tiempo de cada persona, que tenga un nivel de detalle tan alto, que adecue la planificación a la eficiencia y a las necesidades de la persona a quien se planifica.

Se recomienda la realización de talleres y encuentros donde se sensibilice el tema de la planificación en el estudiantado, donde se convenza de la importancia y la necesidad que tiene la misma, donde se debata sobre nuevas formas de estimación, y la importancia de cada una de ellas, y por qué algunas se adecuan más al proceso de desarrollo en la UCI que otras. ¡Entre todos, es posible planificarnos!

Se recomienda además el fomento y la inserción de herramientas de software libre de planificación en el programa de estudio de la carrera, en aras de lograr la familiarización del estudiantado con las mismas y apoyar de esa forma, la política de migración a tecnologías de código abierto propuesta por la UCI.

**BIBLIOGRAFIA CONSULTADA**

1. AGUILAR RAMOS, C. *Aplicación de conceptos de Gestión de Proyectos y Gestión de Riesgos en el desarrollo de productos nuevos en el campo de Tecnología de Información.*: Recinto Universitario de Mayagüez. Universidad de Puerto Rico 2005. p.
2. [23,12, 2006]. Disponible en: <http://www.projectkickstart.com/>
3. AMADOR, J. P. *Administración de Empresas.*
4. BOEHM, B. *Software Cost Estimation with COCOMO II.* New Jersey, Prentice Hall, 2000. p.
5. BOHEM, B. T., RICHARD. *¿Cómo es la estrella de tu proyecto?: Navegapolis.net*, 2005.
6. CHARETTE, R. *Software Engineering Risk Analysis and Management.* McGraw-Hill, 1989. p.
7. COMPARINI, A.: *Comentarios en Paraisoft.*
8. CONCEPCION, P. *Planificación de proyectos de software.: Monografias.com.*
9. GARCÍA, D. *Planificación*, Instituto Universitario Pedagógico Monseñor. Rafael Arias Blanco, 2006.
10. GONZÁLEZ, H. *Visual Studio 2005 Team System, herramientas integradas para un efectivo desarrollo.: MTJ. net Online*
11. HALL, J. *Imendio Planer and Taskjuggler. Tux Magazine*, 2006. 29-31.
12. HINOJOSA, M. A. *Diagrama de Gantt.*
13. JACOBSON, I., BOOCH, GRADY, RUMBAUGH, JAMES. *El Proceso Unificado de desarrollo de software.* . La Habana, Félix Varela, 2004. p.
14. LEDESMA, R. *Sistemas estadísticos de propósitos múltiples: Una revisión de programas gratuitos.*
15. . en., Universidad Nacional Mar de Plata, 2004. 6: 105-117.p.
16. PRESSMAN, R. *Ingeniería de Software. Un enfoque práctico.* 5. La Habana, Félix Varela, 2005. p.
17. RODRÍGUEZ; LOURDES, E. S. *INTRODUCCIÓN A LA DIRECCIÓN INTEGRADA DE PROYECTOS (DIP) - Project Management*, p.
18. UCI. *Conferencias de Ingeniería y Gestión de software.* Conferencia 5. *Planificación y Estimación de Proyectos*, 2005-2006. p.
19. UCI. *Ingeniería y Gestión de Proyectos* Conferencia 5. *Planificación y Estimación de Proyectos*, 2004-2005. p.
20. UCI. *Técnicas de Estimación*, 2005-2006. p.
21. WEST, D. *Planning a Project with the Rational Unified Process*, 2000.

---

## BIBLIOGRAFIA REFERENCIADA

- [1] PRESSMAN, R. *Ingeniería de Software. Un enfoque práctico*. 5. La Habana, Félix Varela, 2005. p.
- [2] RODRÍGUEZ; LOURDES, E. S. INTRODUCCIÓN A LA DIRECCIÓN INTEGRADA DE PROYECTOS (DIP) - Project Management, p.
- [3] GARCÍA, D. *Planificación*, Instituto Universitario Pedagógico Monseñor. Rafael Arias Blanco, 2006.
- [4] LEDESMA, R. Sistemas estadísticos de propósitos múltiples: Una revisión de programas gratuitos. . en., Universidad Nacional Mar de Plata, 2004. 6: 105-117.p.
- [5] HINOJOSA, M. A. *Diagrama de Gantt*.
- [6] BOEHM, B. *Software Cost Estimation with COCOMO II*. New Jersey, Prentice Hall, 2000. p.
- [7] UCI. *Técnicas de Estimación*, 2005-2006. p.
- [8] WEST, D. Planning a Project with the Rational Unified Process., 2000.
- [9] GONZÁLEZ, H. *Visual Studio 2005 Team System, herramientas integradas para un efectivo desarrollo.:* MTJ. net Online.
- [10] COMPARINI, A.: *Comentarios en Paraisoft*.
- [11] CHARETTE, R. *Software Engineering Risk Analysis and Management*. McGraw-Hill, 1989. p.
- [12] AGUILAR RAMOS, C. *Aplicación de conceptos de Gestión de Proyectos y Gestión de Riesgos en el desarrollo de productos nuevos en el campo de Tecnología de Información.:* Recinto Universitario de Mayagüez. Universidad de Puerto Rico 2005. p.

**GLOSARIO DE TÉRMINOS.**

1. Helvecios: tribu gala muy ruda y belicosa.
2. Vercingétorix: Jefe galo que se rebeló contra los romanos.
3. *Portal SharePoint Services: Portal que permite la colaboración entre equipos.*
4. Opensource: Código abierto.
5. XML: *eXtensible Markup Language* (lenguaje de marcas extensible)
6. PNG: *Portable Network Graphics*, es un formato gráfico basado en un algoritmo de compresión sin pérdida para bitmaps(imagen rasterizada).
7. JPG: o JPEG *Joint Photographic Experts Group*, es un algoritmo de compresión con pérdida.
8. PDF: Portable Document Format, Formato de almacenamiento de documentos.
9. HTML: Hyper text Mark Language, lenguaje diseñado para estructurar textos y presentarlos en forma de hipertexto.
10. Linux: Sistema Operativo. paradigmas más prominentes del software libre y del desarrollo del código abierto.
11. MacOSX: es el actual sistema operativo de la familia de ordenadores Macintosh.
12. MAPI: Interfaz de programación que permite que una aplicación envíe y reciba correo a través del sistema de mensajería Microsoft Mail
13. tickets: En el trac representa las tareas
14. bug: Error.
15. Ponderadas: de ponderación, examinar, evaluar.
16. API: *Application Programming Interface*
17. RELY Fiabilidad
18. DATA Tamaño Base de datos
19. CPLX Complejidad
20. RUSE Reutilización requerida
21. DOCU Documentación
22. TIME Restricción tiempo de ejecución
23. STOR Restricción de almacenamiento principal
24. PVOL Volatilidad plataforma



- 25. ACAP Capacidad del analista
- 26. PCAP Capacidad programador
- 27. AEXP Experiencia aplicaciones
- 28. PEXP Experiencia plataforma
- 29. LTEX Experiencia lenguaje y herramientas
- 30. PCON Continuidad del personal
- 31. TOOL Uso de herramientas de software
- 32. SITE Desarrollo Multi-lugar
- 33. SCED Planificación requerida.
- 34. PREC Precedencia
- 35. FLEX Flexibilidad de desarrollo
- 36. RESL Resolución de Arquitectura / Riesgos
- 37. TEAM Cohesión del equipo
- 38. PMAT Madurez del proceso
- 39. VLO Muy bajo
- 40. LO Bajo
- 41. NOM Nominal
- 42. HI Alto
- 43. VHI Muy Alto
- 44. XHI Extra alto
- 45. POO: Programación orientada a objetos.
- 46. OTE Oficina Territorial de estadísticas.