

Experiencia de aplicación de la técnica de Punto Función en estimaciones para software de Tiempo Real

Lic. P. Pesado¹, Ing. A. De Giusti², Lic. M. Boracchia³, Lic. A. Vicenzi⁴

*Laboratorio de Investigación y Desarrollo en Informática⁵
Departamento de Informática - Facultad de Ciencias Exactas
Universidad Nacional de La Plata*

Resumen

Se analizan las experiencias de aplicación de las métricas basadas en punto función para la evaluación de software de Tiempo Real.

En este trabajo se parte de la extensión de la técnica de punto función para Tiempo Real propuesta por el Software Engineering Laboratory in Applied Metrics (SELAM) de la Universidad de Quebec (Canadá) y se analiza sistemáticamente su aplicación a dos clases de sistemas tiempo real: un controlador de comunicaciones distribuido y un adquisidor de datos con control de funciones centralizado.

Dado que ambos sistemas han sido desarrollados en el LIDI y se cuenta con la posibilidad de evaluarlos con las métricas clásicas, se analizan comparativamente los resultados de la técnica de punto función extendida y se discute la posibilidad de obtener una pauta de complejidad del software desarrollado.

Por último se realiza un análisis conceptual de la técnica de punto función en general y en particular de las extensiones utilizadas para Tiempo Real, marcando las líneas de investigación futuras, con el objetivo de establecer una línea base para las estimaciones de desarrollo de Sistemas de Tiempo Real de cierta complejidad.

Palabras clave: Métricas. Punto Función. Ingeniería de Software. Sistemas de Tiempo Real.

¹ Prof. Titular con Ded. Excl. LIDI. Dpto. de Informática, Facultad de Ciencias Exactas, UNLP.

Profesional CIC Pcia. BS. AS. E-mail ppesado@lidi.info.unlp.edu.ar

² Inv. Principal CONICET. Profesor Tit. Ded. Excl., Dpto. de Informática, Facultad de Cs. Exactas, UNLP.

E-mail degiusti@lidi.info.unlp.edu.ar

³JTP Ded. SExcl. LIDI. Dpto. de Informática, Facultad de Ciencias Exactas, UNLP.

E-mail mboracchia@lidi.info.unlp.edu.ar

⁴Ayte. Diplomado Ded. SExcl. LIDI. Dpto. de Informática, Facultad de Ciencias Exactas, UNLP.

E-mail avicenzi@lidi.info.unlp.edu.ar

⁵ Calle 50 y 115 Primer Piso, (1900) La Plata, Argentina, Teléfono 54-21-227707

WEB: lidi.info.unlp.edu.ar

Experiencia de aplicación de la técnica de Punto Función en estimaciones para software de Tiempo Real

Introducción

Uno de los aspectos fundamentales de la gestión de un proyecto es la posibilidad de contar con estimaciones fiables que permitan hacer una planificación efectiva del proceso de desarrollo.

En ese contexto aparece la necesidad de contar con métricas del software tempranas. Dentro de este conjunto aparecen las métricas orientadas a la función que utilizan una medida de la funcionalidad entregada por la aplicación como valor de normalización [PRE98]. Estas métricas fueron propuestas por primera vez por Albretch [ALB79] [ALB84]. Las medidas funcionales sirven para medir el desempeño de los productos desde la perspectiva del usuario y para analizar la productividad, es necesario que sean independientes del desarrollo técnico y de las decisiones de implementación, para que permitan comparar la productividad de diferentes técnicas y tecnologías [GRA92] [KAN95] [PUT92].

La técnica de Punto Función ha evolucionado a lo largo del tiempo, y se han concebido extensiones que han permitido su aplicación a un amplio espectro de sistemas.

En particular nos interesa la aplicación a Sistemas Distribuidos de Tiempo Real (SDTR). El tratamiento de datos en tiempo real ha sido uno de los ejes del desarrollo de la tecnología de hardware y software en los últimos 25 años. Aún antes del microprocesador, el control de procesos y máquinas industriales con datos y señales físicamente distribuidas [COU95][NIE90][UMA93] constituía una de las áreas de investigación tecnológica más dinámicas.

Estas aplicaciones son dependientes del tiempo y procesan un porcentaje importante de información orientada al control de eventos y dispositivos externos al sistema de software.

Un SDTR debe interactuar con el mundo real, en puntos físicamente distantes, en períodos de tiempo que vienen determinados por el contexto o las restricciones de la especificación (en muchos casos a partir de una activación asincrónica). La evolución tecnológica en el tratamiento de señales (locales o remotas) y en los sistemas de comunicaciones ha impulsado enormemente esta área temática en los últimos años, sobre todo en los aspectos de planificación y desarrollo de software para Sistemas Distribuidos de Tiempo Real. [HAT88][LAP93][SHU92].

En este trabajo se ha investigado la aplicación de técnicas de punto función extendidas [ABR97] [SEL97] al análisis de un sistema distribuido de entrenamiento militar [ROM98] que por sus características (datos distribuidos, comunicaciones y toma de decisiones en tiempo real, criptografiado de información, etc.) tiene una complejidad media y puede ser un caso de interés para comparar las estimaciones con el desarrollo real del sistema.

Las conclusiones son preliminares y marcan la utilidad de las extensiones a las métricas basadas en punto función, para analizar proyectos de SDTR.

Puntos de función

A fines de 1970, A. J. Albretch sugirió que la unidad de salida de los proyectos software debería ser válida para cualquier lenguaje de programación y debería representar tópicos de interés para los usuarios del software. Lo que él proponía era medir la *funcionalidad* del software.

Albretch consideró que los aspectos del software externos podían ser enumerados mediante cinco ítems:

- *Número de entradas externas*: se cuenta cada entrada de usuario que proporciona al software diferentes datos orientados a la aplicación.
- *Número de salidas externas*: se cuenta cada salida que proporciona al usuario información orientada a la aplicación.
- *Número de consultas externas*: una petición está definida como una entrada interactiva que resulta en la generación de algún tipo de respuesta por parte del sistema en forma de salida interactiva.
- *Número de archivos internos*: se cuenta cada archivo lógico.
- *Número de interfaces externas*: se cuentan todas las interfaces legibles por la máquina (por ejemplo archivos en cinta o disco) que son utilizados como comunicación con otro sistema.

Cálculo tradicional de Puntos de Función

Los puntos de función (*FP*, function points), se obtienen utilizando una relación empírica basada en medidas cuantitativas del dominio de información del software y valoraciones subjetivas de la complejidad del software[ABR96][KEM93].

Por ejemplo en [ART85] los puntos de función se calculan siguiendo la tabla que se muestra a continuación:

Parámetro de medida	Cuenta	Factor de peso			Subtotal
		Simple	Medio	Complejo	
Nro. de entradas de usuario	n1	3	4	6	s1
Nro. de salidas de usuario	n2	4	5	7	s2
Nro. de peticiones de usuario	n3	3	4	6	s3
Nro. de archivos lógicos internos	n4	7	10	15	s4
Nro. de interfaces externas	n5	5	7	10	s5

Tabla para cálculo de puntos de función.

Cuando se han recolectado los datos anteriores se asocia un valor de complejidad a cada cuenta. En general este valor es subjetivo, si bien pueden utilizarse criterios para su determinación automática.

El cálculo de los puntos de función se realiza mediante la siguiente fórmula:

$$FP = \left(\sum_{i=1}^5 n_i * w_i \right) * (K_1 + K_2 * \sum F_i) = \left(\sum_{i=1}^5 s_i \right) * (K_1 + K_2 * \sum F_i)$$

donde los w_i son los factores de peso; k_1 y k_2 son coeficientes empíricos con un valor por defecto de 0.65 y 0.01 y donde los valores F_i corresponden al ajuste de complejidad que algunos autores [ART85] ubican en el rango de 0 a 5. En el Anexo 1 se muestra una tabla donde para $N_{max}=14$ se muestran las preguntas asociadas con cada uno de los 14 posibles aspectos a estudiar.

Los puntos de función a su vez, proveen un medio de predecir la cantidad de líneas de código que deben ser escritas para un determinado sistema. A cada lenguaje se le puede asociar un *nivel de lenguaje*, el cual es el número promedio de sentencias requeridas para implementar un punto de función en dicho lenguaje.

El nivel de lenguajes es de interés considerable por las siguientes razones:

- Ya en la fase de especificación de requerimientos o diseño se puede predecir la cantidad de sentencias de código fuente que serán requeridas.
- La posibilidad de calcular puntos de función de sistemas existentes (para los cuales puede que se tenga disponible su tamaño en cantidad de sentencias) sin el trabajo laborioso de computar puntos de función.
- La posibilidad de convertir la medida de tamaño de una aplicación (en líneas de código fuente) escrita en un lenguaje al tamaño equivalente si la aplicación fuera escrita en algún otro lenguaje.
- La posibilidad de medir la productividad de proyectos escritos en múltiples lenguajes.

En el Anexo 2 se da una tabla tentativa con las líneas de código por punto función para diferentes niveles de lenguaje, así como los datos de productividad esperada por programador para cada nivel de lenguaje [JON91].

A partir de la estimación de los puntos función de un proyecto de software, y utilizando las tablas mencionadas anteriormente, se puede tener una estimación del costo y esfuerzo de desarrollo requeridos para el proyecto.

Naturalmente, resulta muy difícil adaptar a toda clase de sistemas una métrica de carácter general como la de punto función. Por esto aparecen extensiones y modificaciones para ajustar el comportamiento del estimador al tipo de sistema a desarrollar [JON96].

Extensiones para Tiempo Real a las métricas de Punto Función

Tomando como base los trabajos del Laboratorio de Investigación de Administración de la Ingeniería del Software en la Universidad de Quebec en Montreal [GAL95][IFP94] se ha adaptado la técnica de Punto Función para que sea aplicable a los sistemas de tiempo real.

El aspecto de control del software de tiempo real se considera con nuevos tipos de funciones:

- Número de grupos de datos de control internos: datos lógicamente relacionados que son actualizados por la aplicación evaluada.
- Número de grupos de datos de control externos: datos lógicamente relacionados que son usados (pero no actualizados) por la aplicación evaluada.
- Número de entradas de control externas: un dato de control entra desde afuera del limite de la aplicación, es un subproceso único.
- Número de salidas de control externas: un dato de control sale del limite de la aplicación, es un subproceso único.
- Número de lecturas de control internas: un dato de control es leído, es un subproceso único.
- Número de escrituras de control internas: un dato de control es escrito, es un subproceso único.

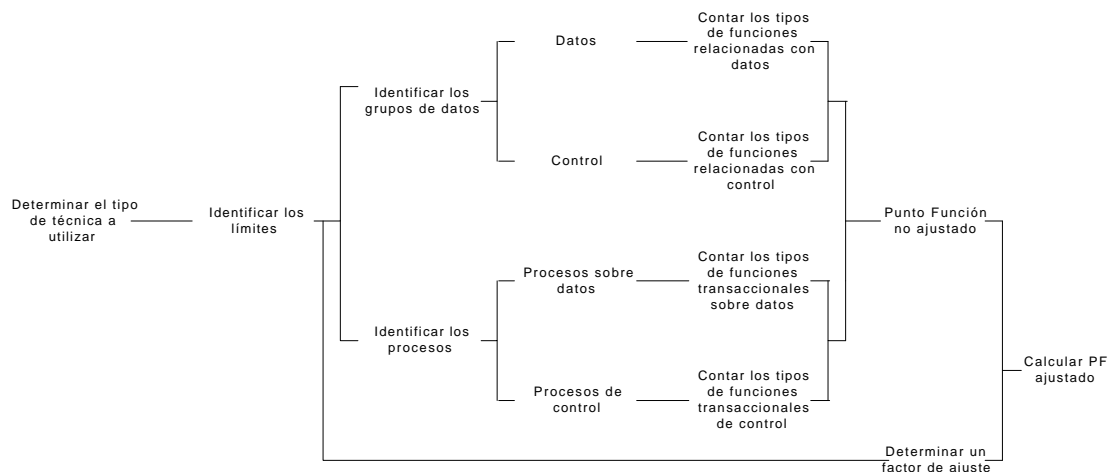
El conteo de Puntos Función con la extensión puede definirse como la suma de los Punto Función standard y los Punto Función de Control.

Existen dos particularidades en los sistemas de tiempo real que deben tenerse en cuenta en el momento de hacer el conteo:

- Hay dos clases de estructuras de datos de control: grupos lógicos de ocurrencia múltiple (pueden tener más de una instancia del mismo tipo de registro) y grupo lógico de ocurrencia simple (tienen sólo una instancia del mismo tipo de registro). Estas últimas son las que priman en los sistemas de tiempo real y son difíciles de agrupar como archivos internos o interfaces externas.

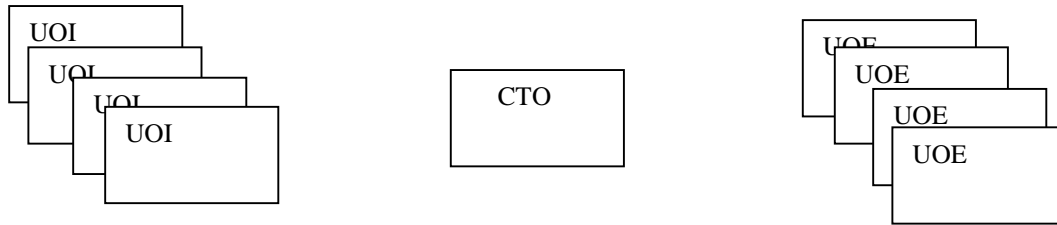
- La cantidad de subprocesos varía mucho en los sistemas de tiempo real.

Analizado el SDTR, en el momento del conteo se sigue un análisis como el de la Figura:



Aplicación: Complejidad de un Sistema Distribuido de Tiempo Real

El Sistema de Mensajería de Entrenamiento Militar [ROM98] cuya planificación se estudió con la técnica de punto función extendida, puede resumirse estructuralmente en las siguientes características:



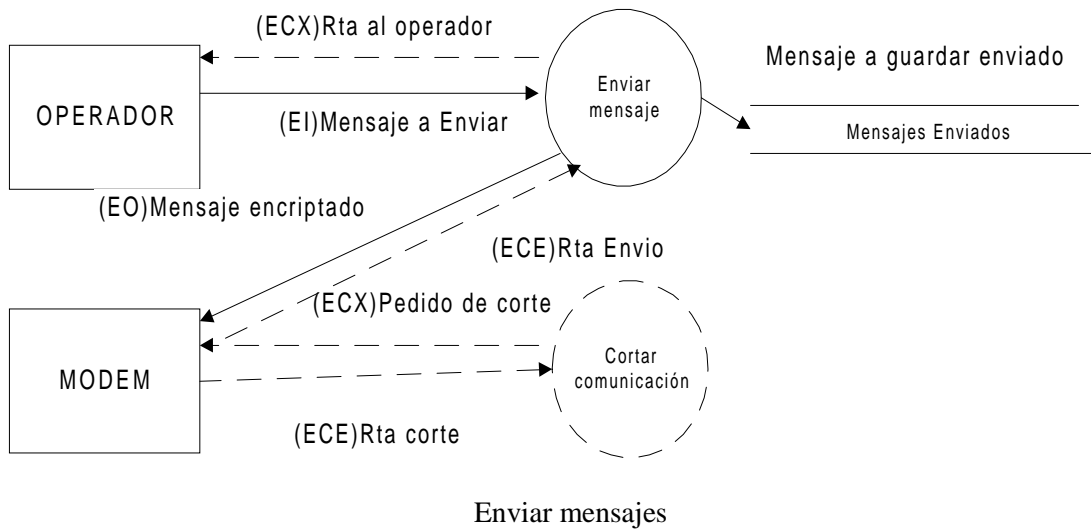
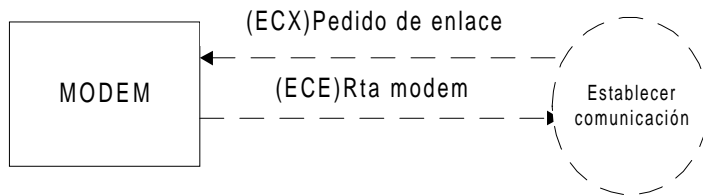
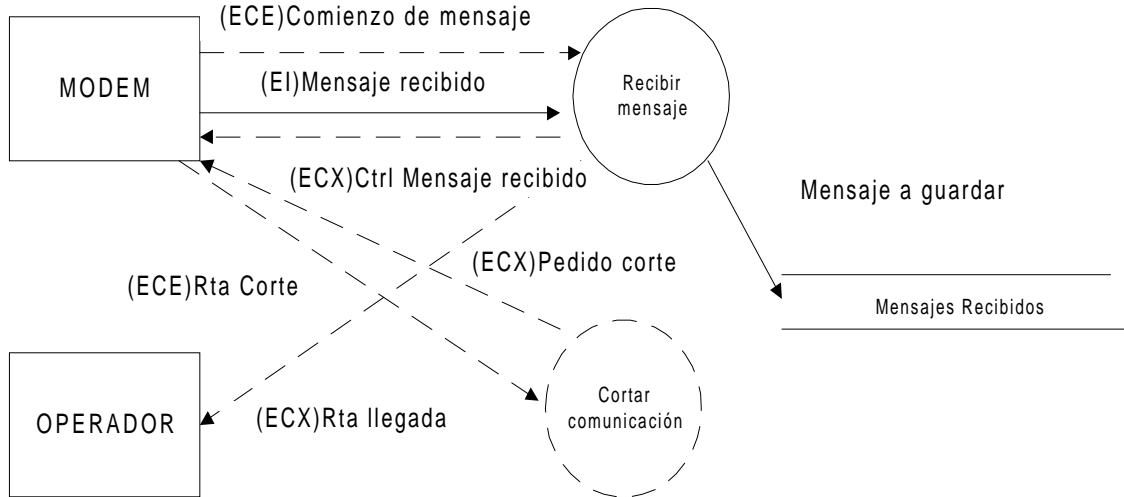
- Existen múltiples unidades operativas UOI distribuidas en un área de operaciones. Todas están enlazadas por un subsistema de comunicaciones que puede ser manual, cableado, telefónico o por radio.
- En el área de operaciones hay una central de tráfico de operaciones (CTO) que recibe las comunicaciones desde las UOI locales y también desde unidades operativas externas UOE, que incluyen puestos de comando de mayor jerarquía.
- Existe un subsistema de control de comunicaciones interno, que a partir de la CTO coordina los mensajes que circulan entre las unidades operativas propias, incluyendo órdenes militares.
- Dentro de este subsistema de control de comunicaciones hay funciones de seguridad, que incluyen el encriptado y desencriptado de mensajes internos y externos, funciones que deben resolverse en tiempo real, en forma independiente de la tecnología de comunicaciones y con la máxima seguridad.

La migración del sistema convencional a un sistema automático, basado en comunicaciones externas a través de una central telefónica convencional controlada por una microcomputadora y comunicaciones internas con tecnología de radio-packet se realizaría a través de un convenio entre la UNLP y el Batallón de Comunicaciones de City Bell, contemplando:

- Integración del hardware de comunicaciones preexistente para permitir la conexión y transmisión de mensajes.
- Análisis y diseño de un sistema de tiempo real con restricciones de tiempo muy rígidas. Por otra parte existen restricciones específicas impuestas por la tecnología de la central utilizada y los modems de comunicaciones, que obligan al desarrollo de los drivers específicos para la aplicación.
- Encriptado y desencriptado de datos en tiempo real con manejo de claves jerárquicas, automatizando la codificación y decodificación de mensajes.
- Capacidad de almacenamiento en la central de tráfico de toda la documentación correspondiente al mensaje recibido o transmitido. Los operadores autorizados pueden recuperar/consultar el tráfico de mensajes.
- Modelización y prueba off-line.

Análisis del Software y Conteo de los Puntos de Función

A partir del diagrama de flujo de control del Kernel del Sistema en estudio, se puede obtener la tabla de puntos función que sigue [VIC97].



	Puntos
Entrada externa (EI)	2
Salida externa (EO)	1
Entrada de control externa(ECE)	5
Salida de control externa (ECX)	6
Archivos internos	2
Total de Puntos	16

Cálculo estimado de las líneas de código a desarrollar

El sistema a desarrollar requería recursos de lenguaje de 3ra. Y 4ta. Generación ya que debe hacerse control directo de hardware y también desarrollar una interfaz gráfica en un lenguaje visual.

A partir de la estimación de puntos función, nos ubicamos en la tabla del Anexo 2, suponiendo un 20 % de código de 3ra. Generación y un 80% de código de 4ta. Generación, lo que daba una estimación de 500 líneas fuente para la parte del sistema correspondiente sólo a las funciones de envío y recepción de mensajes.

Resultados Obtenidos y Líneas de Trabajo actuales

El sistema ha sido desarrollado, está actualmente operativo y es posible su generalización para diversas unidades militares.

En su desarrollo se utilizaron los lenguajes Visual Basic, C y tiene unas 470 líneas de código fuente para las funciones estudiadas.

El resultado más significativo ha sido la concordancia notoria entre la estimación realizada a partir del análisis con punto función extendidos y los resultados concretos del desarrollo, lo que pareciera significativo a la luz de la limitada experiencia de los autores en la estimación de SDTR con este tipo de técnicas.

La línea de trabajo actual está centrada en la investigación de la aplicación sistemática de las técnicas de punto función en proyectos de SDTR de mayor envergadura, especialmente en sistemas de control industrial de tiempo real duro. Por otra parte, se trata de estudiar el empleo de estas técnicas en la estimación de los desarrollos de interfaces hombre/máquina basadas en eventos.

ANEXO 1

Factores de Ajuste de la Complejidad

Evaluar cada uno de los factores siguientes en una escala de 0 a 5:

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

1. Requiere el sistema copias de seguridad y de recuperación fiables ?
2. Se requieren comunicaciones de datos ?
3. Existen funciones de procesamiento distribuido ?
4. Es crítico el rendimiento ?
5. Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado ?
6. Requiere el sistema entrada de datos interactiva ?
7. Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas operaciones ?
8. Se actualizarán los archivos maestros de forma interactiva ?
9. Son complejas las entradas, las salidas, los archivos o las peticiones ?
10. Es complejo el procesamiento interno ?
11. Se ha diseñado el código para ser reutilizable ?
12. Están incluidas en el diseño la conversión y la instalación ?
13. Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones ?
14. Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

ANEXO 2

Puntos Función. Niveles de lenguaje y Productividad.

Como un ejemplo de niveles de lenguajes, mostramos aquí una tabla para las distintas generaciones de lenguajes.

Lenguaje	Nivel	Líneas de código por punto función
1era. generación	1.00	320
2da. generación	3.00	107
3era. generación	4.00	80
4ta. generación	16.00	20
5ta. generación	70.00	4

Niveles de lenguajes

La siguiente tabla relaciona niveles de lenguaje con la productividad

Nivel de lenguaje	Productividad promedio por persona-mes
1-3	5 a 10 PF
4-8	10 a 20 PF
9-15	16 a 23 PF
16-23	15 a 30 PF
24-55	30 a 50 PF
más de 55	40 a 100 PF

Bibliografía

[ABR96] Abran A., Robillard P.N. *Function point Analysis. An Empirical Study of its Measurement Processes*. IEEE Transactions on Software Engineering. Vol 22 no. 12 pp 895-909 Dec 1996.

[ABR97] Abran A., Maya M., Desharnais J. M., *Adapting Function Points to real-time software*, American Programmer, 1997

[ALB79] Albrecht A. J. *Measuring Application Development Productivity*, Proc IBM Application Development Symposium, 1984.

[ALB84] Albrecht A. J. *AD/M Productivity Measurement and Estimate Validation*, IBM Corporate Information Systems, IBM Corp. Purchase N.Y. 1984.

[ART85] Arthur L. J. , *Measuring Programmer Productivity and Software Quality*, Wiley, 1985.

[COU95] Coulouris G, Dollimore J, *Distributed System Concepts and Design*, Ad. Wesley, 1995.

[GAL95] Galea S. *The Boeing Company: 3D Function Point Extensions*, V2.0 Release 1.0, Boeing Information and Support Service, Research and Technology Engineering. June 1995.

[GRA92] Grady R. B. *Practical software metrics for project management and process improvement*. Prentice Hall, New Jersey. 1992

[HAT88] Hatley D., Pirbhai I. *Strategies for Real-Time System Specification*, Dorset House, 1988.

[IFP94] IFPUG, *Function Point Counting Practices Manual* , Release 4.0 International Function point Users Group- IFPUG- Westerville, 1994.

[JON91] Jones C. *Applied Software Measurement- Assuring Productivity and Quality*, McGraw Hill 1991.

[JON96] Jones C. *Applied Software Measurement*, McGraw Hill 1996.

[KAN95] Kan S. *Metrics and Models in Software Quality Engineering*. Addison Wesley, 1995.

[KEM93] Kemerer C.F. *Reliability of function point measurement: a field experiment*. *Communications of the ACM*, Vol 36 1993

[LAP93] Laplante P. *Real Time System Design and Analysis: An Engineer's Handbook*. The Institute of Electrical and Electronics Engineers inc. New York 1993.

[NIE90] Nielsen *Ada in Distributed Real-Time System*. 1990

[PRE98] Pressman R. *Ingeniería de Software*, McGraw Hill, 1998.

[PUT92] Putnam L. *Measures for excellence*, Yourdon Press, 1992.

[ROM98] Romero F., Sosa R., De Giusti A., *Sistemas distribuidos de Tiempo Real. Experiencia en el diseño y puesta a punto de un sistema de Mensajería para entrenamiento*

militar. Informe Técnico LIDI, 01-016-98,1998.

[SEL97] SELAM, *Full Function Points: Counting practices manual*, Tech. Report 1997-04,1997

[SHU92] Shumate K., *Software specification and design for real-time systems*, Wiley, 1992.

[VIC97] Vicenzi A. L. ,Pesado P. Boracchia M., *Técnicas de conteo de Punto Función aplicadas a Tiempo Real*, Informe Técnico LIDI, 01-023-97,1997.