

# Marco de Trabajo para la definición de Métricas de Complejidad

Carlos Montenegro A.  
carlos.montenegro@epn.edu.ec

## Resumen

Importantes esfuerzos de investigación en la temática de la Complejidad, se dirigen a la identificación y valoración de sus características mediante análisis específicos aplicados a los ámbitos del conocimiento del caso de estudio. Por esta razón, aunque se han definido decenas de métricas de Complejidad, no existen criterios generales que faciliten la definición, análisis, comparación y generalización de los resultados.

En este artículo se especifica un Marco de Trabajo para la definición de métricas de complejidad, en ámbitos diferentes a la complejidad algorítmica (computacional), sobre la base de tres conceptos: la identificación del sistema mediante su *Contenido*, una calificación de la relación de ese contenido con las características de la complejidad organizadas en *Principios*; y, el cálculo de la *Métrica de Complejidad*, usando un modelo de ponderación y acumulación. Este Marco de Trabajo se sustenta en un enfoque formal que garantiza su consistencia y posibilita la comparación y generalización.

**PALABRAS CLAVE:** Sistema, Complejidad, Marco de Trabajo, Modelo Formal, Métrica

## INTRODUCCION

Los estudios acerca de la Complejidad tienen enfoques diferentes y su definición, características y medidas son variadas. En múltiples estudios, se han definido y/o compilado decenas de características y métricas ((Couture, 2007a), (Couture, 2007b), (Couture, 2007d), (Adami, 2002), (Maldonado, C. y Gómez, N, 2010), (Seel, 1999), (Edmonds, 1997), (Bar-Yam, 2000). En (Couture, 2007c) se encuentra una extensa bibliografía de los trabajos sobre complejidad, los mismos que se refieren a ámbitos de física, química, biología, ciencias sociales, informática, entre otros. Estudios interdisciplinarios se pueden encontrar en (Stepanic, 2010).

Desde una perspectiva filosófica, según Morin:

“A primera vista la complejidad es un tejido (*complexus*: lo que está tejido en conjunto) de constituyentes heterogéneos inseparablemente asociados: presenta la paradoja de lo uno y lo múltiple. Al mirar con más atención, la complejidad es, efectivamente, el tejido de eventos, acciones, interacciones, retroacciones, determinaciones, azares, que constituyen nuestro mundo fenoménico. Así es que la complejidad se presenta con los rasgos inquietantes de lo enredado, de lo inextricable, del desorden, la ambigüedad, la incertidumbre...” (Morin, 1998).

Morin considera además que complejidad está íntimamente ligada a los sistemas y a la organización.

Desde una perspectiva técnica, Warren Weaver<sup>1</sup> plantea que la complejidad de un sistema particular constituye el grado de dificultad para predecir las propiedades del mismo, si se tienen a disposición las propiedades de los componentes. Considera que la complejidad se presenta de dos formas: complejidad desorganizada y complejidad organizada. (Weaver, 1948)

La complejidad desorganizada resulta de un sistema con una gran cantidad de partes. Las interacciones del sistema pueden ser vistas como altamente aleatorias y las propiedades pueden comprenderse mediante métodos probabilísticos y estadísticos.

En la complejidad organizada se presentan interacciones no aleatorias o correlacionadas entre las partes. Estas relaciones permiten distinguir una estructura de sistema que puede interactuar con otros sistemas. El sistema presenta propiedades diferentes a las partes y su grado de organización “emerge” sin una guía. Esta emergencia constituye apenas una característica de los sistemas complejos.

Weaver y Morin, desde distintas orillas, coinciden en la descripción de algunas propiedades generales de la complejidad. Bajo la perspectiva actual del desarrollo de las ciencias, esta complejidad se manifiesta mediante estructuras y comportamientos que para su explicación requieren enfoques multidisciplinarios y transdisciplinarios (Revisar por ejemplo (Maldonado et. al, 2010) y (Stepanic, 2010)). Así, en el ámbito de tecnologías de la información, se reconoce a Internet (Park, 2005), a los Sistemas de Sistemas (Firesmith, 2010) y aplicaciones varias (Magee y Weack, 2008), como sistemas de alta complejidad.

Aquí se parte del criterio que es indispensable un enfoque formal que sustente la consistencia de las métricas de complejidad. Para ello se define un Marco de Trabajo que contiene Definiciones, Axiomas y Propiedades. Este Marco posibilita el análisis de modelos y métricas, la comparación la generalización de esos modelos y de los resultados de su aplicación. Por el enfoque del trabajo, se excluye a la complejidad algorítmica (computacional)<sup>2</sup>.

## **1. DESARROLLO FORMAL DE LA COMPLEJIDAD EN LOS SISTEMAS**

Acerca de la complejidad, desde la perspectiva de modelos formales, se han realizado aportaciones en algunos campos; particularmente, los modelos matemáticos de la Teoría de Sistemas y los trabajos en Ingeniería de Software. En este último campo, se han desarrollado axiomas, propiedades y definiciones de métricas; algunos aportes se encuentran en (Fenton, 1994), (Morasca, 1996) y (Morasca, 2008). Estos trabajos son algunos antecedentes que muestran la factibilidad para la formalización de una perspectiva general de la complejidad, como se plantea adelante<sup>3</sup>.

---

<sup>1</sup> Weaver junto a Shannon, son los creadores de la teoría de la información

<sup>2</sup> La complejidad algorítmica, también conocida como estructural, estudia la complejidad inherente a la resolución de un problema computable. En este trabajo se trata de cualificar y cuantificar la presencia de la complejidad con un enfoque sistémico. Existe una diferencia sutil con el término computacional que involucra el programa que implementa el algoritmo.

<sup>3</sup> Una definición formal, para el caso, contiene axiomas, propiedades y operaciones que se originan en la teoría de conjuntos.

Como es evidente, la complejidad constituye una propiedad de los sistemas y es a partir de la definición formal de Sistema que a continuación se presentan las definiciones, axiomas y propiedades.

**Definición de Sistema.** Un sistema  $S$  es un par  $\langle E, R \rangle$ , donde  $E$  representa el conjunto de elementos de  $S$  y  $R$  es una relación binaria en  $E$  ( $R \subseteq E \times E$ ) y representa las relaciones entre los elementos de  $S$ . Los elementos pueden agruparse en componentes que pueden compartir elementos.

**Definición de Componente.** Dado un Sistema  $S = \langle E, R \rangle$ , un componente  $c = \langle E_c, R_c \rangle$  es un sistema tal que  $E_c \subseteq E \wedge R_c \subseteq R$ .

Para generalidad y simplicidad, un Componente es un subsistema.

**Definición de Relación.**  $R$  especifica la relación física o conceptual entre elementos de  $E$ . Tiene la forma  $\langle (e_1, e_2), t \rangle$ , donde  $t$  es la definición de la relación y  $e_1, e_2$  son los elementos relacionados.

Dado un componente  $c$  de un Sistema  $S = \langle E, R \rangle$ , el conjunto de relaciones que involucran a elementos de  $c$  y a elementos fuera de  $c$ , se denomina **Conjunto de Relaciones Externas de  $c$** , denotado  $R_{\text{ext}}(c)$ , con  $R_{\text{ext}}(c) \subseteq R$ .

Las relaciones cumplen con las siguientes operaciones y características:

- **Inclusión.**  $R_1 \subseteq R_2 \Leftrightarrow \forall \langle (a, b), t_1 \rangle \in R_1, \exists \langle (a, b), t_2 \rangle \in R_2 \wedge t_1 = t_2$ , es decir,  $R_2$  contiene al menos todas las ocurrencias de las relaciones en  $R_1$ .
- **Unión.**  $R_3 = R_1 \cup R_2 \Leftrightarrow \forall \langle (a, b), t_3 \rangle \in R_3, \exists \langle (a, b), t_1 \rangle \in R_1, \langle (a, b), t_2 \rangle \in R_2, t_3=t_1 \vee t_3=t_2$ , es decir,  $R_3$  abarca todas las ocurrencias de  $R_1$  y  $R_2$ .
- **Intersección.**  $R_3 = R_1 \cap R_2 \Leftrightarrow \forall \langle (a, b), t_3 \rangle \in R_3, \exists \langle (a, b), t_1 \rangle \in R_1, \langle (a, b), t_2 \rangle \in R_2, t_3= t_1 \wedge t_3 = t_2$ , es decir,  $R_3$  contiene todas las ocurrencias de relaciones comunes a  $R_1$  y  $R_2$ .

A su vez, los componentes cumplen con las siguientes operaciones y características:

- **Inclusión.** El Componente  $c_1$  está incluido en el Componente  $c_2$  (notación:  $c_1 \subseteq c_2$ ), si  $E_1 \subseteq E_2 \wedge R_1 \subseteq R_2$ .
- **Unión.** La Unión de Componentes  $c_1$  y  $c_2$  (notación:  $c_1 \cup c_2$ ) es un componente  $\langle E_1 \cup E_2, R_1 \cup R_2 \rangle$ .
- **Intersección.** La intersección de Componentes  $c_1$  y  $c_2$  (notación:  $c_1 \cap c_2$ ) es el componente  $\langle E_1 \cap E_2, R_1 \cap R_2 \rangle$ .
- **Componente Vacío.** El Componente  $\langle \emptyset, \emptyset \rangle$  (denotado por  $\emptyset$ ) es el Componente Vacío.
- **Componentes Disjuntos.** Los Componentes  $c_1$  y  $c_2$  son disjuntos si  $c_1 \cap c_2 = \emptyset$
- **Componentes Desconectados.** Dos componentes disjuntos  $c_1$  y  $c_2$  de un sistema se denominan Desconectados si  $R_{\text{ext}}(c_1) \cap R_{\text{ext}}(c_2) = \emptyset$

En el gráfico, S es el Sistema; C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub> son los componentes, E = {e<sub>1</sub>, e<sub>2</sub>, e<sub>3</sub>, e<sub>4</sub>, e<sub>5</sub>, e<sub>6</sub>, e<sub>7</sub>} son los elementos y R = {(e<sub>1</sub>, e<sub>2</sub>), 1), ((e<sub>2</sub>, e<sub>3</sub>), 2), ((e<sub>3</sub>, e<sub>4</sub>), 3), ((e<sub>4</sub>, e<sub>6</sub>), 4)} constituye un conjunto de relaciones definidas por dígitos. Entonces:

$$S = \langle \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}, \{((e_1, e_2), 1), ((e_2, e_3), 2), ((e_3, e_4), 3), ((e_4, e_6), 4)\} \rangle$$

$$C_1 = \langle \{e_1, e_2, e_3\}, \{((e_1, e_2), 1), ((e_2, e_3), 2)\} \rangle$$

$$C_2 = \langle \{e_3, e_4\}, \{((e_3, e_4), 3)\} \rangle$$

$$C_3 = \langle \{e_5, e_6, e_7\}, \emptyset \rangle$$

$$C_1 \cup C_2 = \langle \{e_1, e_2, e_3, e_4\}, \{((e_1, e_2), 1), ((e_2, e_3), 2), ((e_3, e_4), 3)\} \rangle$$

$$C_1 \cap C_2 = \langle \{e_3\}, \emptyset \rangle$$

$$R_{\text{ext}}(C_1) = \{(e_3, e_4), 3\}$$

$$R_{\text{ext}}(C_2) = \{((e_2, e_3), 2), ((e_4, e_6), 4)\}$$

$$R_{\text{ext}}(C_3) = \{(e_4, e_6), 4\}$$

$$R_{\text{ext}}(C_1) \cap R_{\text{ext}}(C_2) = \emptyset$$

$$R_{\text{ext}}(C_2) \cap R_{\text{ext}}(C_3) = \{(e_4, e_6), 4\}$$

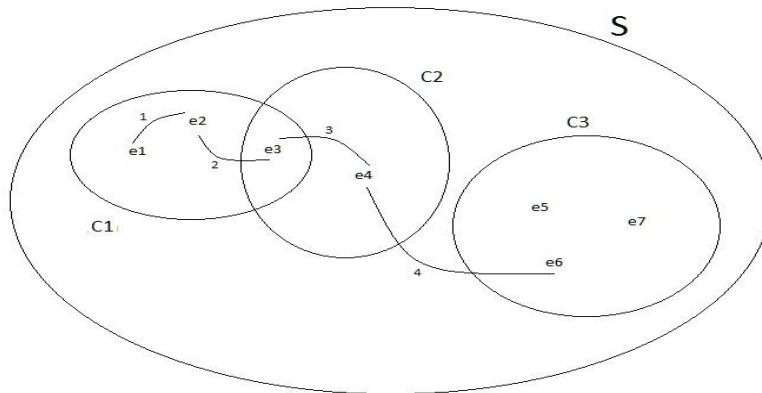


Figura 1. Sistema, Componentes, Elementos y Relaciones

**Axioma 1.** La complejidad de un sistema S no es menor que la suma de las complejidades de dos componentes cualesquiera c<sub>1</sub>, c<sub>2</sub> cuya unión se incluye en S, es decir:

$$S \supseteq c_1 \cup c_2 \Rightarrow \text{Complejidad}(S) \geq \text{Complejidad}(c_1) + \text{Complejidad}(c_2)$$

Los dos componentes c<sub>1</sub> y c<sub>2</sub> pueden tener elementos en común. Por lo tanto c<sub>1</sub> y c<sub>2</sub> o sus elementos pueden estar enlazados por relaciones. Tiene sentido el signo '≥', puesto que las relaciones de c<sub>1</sub> y c<sub>2</sub> no toman en cuenta las relaciones externas que pueden involucrar a los dos componentes, que ahora están incluidas en S, así como las emergencias<sup>4</sup>.

<sup>4</sup> La emergencia hace referencia a aquellas propiedades o procesos de un sistema, no reducibles a las propiedades o procesos de sus partes constituyentes. El concepto de emergencia se relaciona estrechamente con los conceptos de autoorganización y supervivencia; y, se define en oposición a los conceptos de reduccionismo y dualismo.

Los valores de Complejidad(S) son posibles de calcular mediante *Funciones de Ponderación y Acumulación* (Ver adelante).

**Axioma 2. Aditividad de Componentes Desconectados.** La complejidad de un sistema S compuesto de dos componentes desconectados  $c_1, c_2$ , es igual a la suma de las complejidades de los dos componentes

$$S = c_1 \cup c_2 \wedge c_1 \cap c_2 = \emptyset \wedge E_{\text{ext}}(c_1) \cap E_{\text{ext}}(c_2) = \emptyset \Rightarrow$$

$$\text{Complejidad}(S) = \text{Complejidad}(c_1) + \text{Complejidad}(c_2)$$

A continuación se describen propiedades que pueden derivarse de los axiomas anteriores.

**Propiedad 1. Valor Nulo.** Sea  $c_1 = \langle E_1, \emptyset \rangle$  un componente sin relaciones. Entonces  $\text{Complejidad}(c_1) = 0$ .

*Prueba.* Dado un Sistema S, se tiene  $S = S \cup \emptyset$ . Aplicando el Axioma 2, tomando en cuenta que S y  $\emptyset$  son obviamente desconectados, se tiene  $\text{Complejidad}(S) = \text{Complejidad}(S \cup \emptyset) = \text{Complejidad}(S) + \text{Complejidad}(\emptyset)$ , luego  $\text{Complejidad}(\emptyset) = 0$ .

Por el Axioma 1, dado que  $c_1 \supseteq \emptyset \cup \emptyset$ , se tiene  $\text{Complejidad}(c_1) \geq 0$ . Por otro lado,  $c_1 \supseteq c_1 \cup c_1$ , por lo tanto  $\text{Complejidad}(c_1) \geq 2 \cdot \text{Complejidad}(c_1)$ , por el Axioma 1. Así,  $0 \geq \text{Complejidad}(c_1)$  y  $\text{Complejidad}(c_1) = 0$ .

**Propiedad 2. Valor Menor.** La complejidad de un sistema  $S_2 = \langle E_2, R_2 \rangle$  no es menor que la complejidad del sistema  $S_1 = \langle E_1, \emptyset \rangle$  que no contiene relaciones, es decir,  $\text{Complejidad}(S_2) \geq \text{Complejidad}(S_1)$

*Prueba.* Sobre la base del Axioma 1,  $S_2 \supseteq \emptyset \cup \emptyset$ , entonces  $\text{Complejidad}(S_2) \geq 2 \cdot \text{Complejidad}(\emptyset) = 0 = \text{Complejidad}(S_1)$ .

**Propiedad 3. Monotónica.** Al añadir relaciones a un sistema, varía la complejidad.

*Prueba.* Suponiendo una nueva relación  $\langle (e_1, e_2), t \rangle$  que se añade entre dos elementos  $e_1, e_2$  de un sistema S, para obtener un nuevo sistema  $S' = S \cup c \supset S$ , donde  $c = \langle \{e_1, e_2\}, \{((e_1, e_2), t)\} \rangle$ , es decir, es el componente cuyos elementos son  $e_1$  y  $e_2$  con una relación única t. Como  $S' \supseteq S \cup c$ , por el Axioma 1  $\text{Complejidad}(S') \geq \text{Complejidad}(S) + \text{Complejidad}(c)$ , y la Propiedad 2 muestra que  $\text{Complejidad}(c) \geq 0$ , se tiene  $\text{Complejidad}(S') \geq \text{Complejidad}(S)$

## 2. MARCO DE TRABAJO Y METRICA DE COMPLEJIDAD

El Marco de Trabajo y la Métrica se estructuran a partir de las siguientes definiciones:

**Definición 1.** Contenido es el conjunto CS constituido por relaciones que gobiernan al Sistema S. CS permite conocer y comprender a S. CS está conformado por Áreas Clave ( $AK_1, AK_2, \dots, AK_p$ ) únicas, cada una estructurada, en forma de un grafo tipo árbol, por Áreas Clave únicas de

menor jerarquía (Subareas); y, así sucesivamente, hasta las Hojas que constituyen los detalles del Área Clave

$$CS = \langle AK_1, AK_2, \dots, AK_p \rangle$$

Ejemplos:

$S = \langle ((\text{complejidad, emergencia}), CS) \rangle$ , y

$CS = \langle AK_1 \rangle$

$AK_1 = ((\text{complejidad, emergencia}), \text{“la emergencia genera mayor complejidad”})$

$S = \langle ((\text{costo, calidad, cronograma}), CS) \rangle$ , y

$CS = \langle AK_1, AK_2 \rangle$

$AK_1 = ((\text{costo, calidad}), \text{“el costo aumenta con la calidad”})$

$AK_2 = ((\text{costo, cronograma}), \text{“el costo aumenta con el cronograma”})$

En los dos ejemplos se tienen  $AK_i$  simples. Sin embargo, CS puede constituirse por un conjunto de  $AK_i$  altamente complejas, por ejemplo los “Cuerpos de Conocimiento”<sup>5</sup> de distintas disciplinas, como: Ingeniería de Software (SWEBOK), Desarrollo de Nuevos Productos (NPDBOK), Gestión de Proyectos (PMBOK), Análisis de Negocios (BABOK), Bases de Datos (DBBOK), Gestión de la Práctica Médica (MPMBOK), Gestión Universal de Servicios (USMBOK), etc. Como ejemplo, SWEBOK está organizado en diez áreas clave, cada una con sus subareas, sus tópicos y los detalles que constituyen las hojas del árbol. Ver detalles en (IEEE, 2004).

A continuación un ejemplo del SWEBOK, de acuerdo a la estructura citada, referida al Área Clave de “Requerimientos de Software” en una de sus Subareas y Tópicos, con su Detalle.

<b>Area Clave</b>	Requerimientos de Software
<b>Subarea Clave</b>	Fundamentos de Requerimientos de Software
<b>Topico</b>	Requerimientos del Sistema y Requerimientos de Software
<b>Detalle</b>	In this topic, system means ‘an interacting combination of elements to accomplish a defined objective. These include hardware, software, firmware, people, information, techniques, facilities, services, and other support elements’ as defined by the International Council on Systems Engineering (INCOSE00). System requirements are the requirements for the system as a whole. In a system containing software components, software requirements are derived from system requirements. The literature on requirements sometimes calls system requirements ‘user requirements’. The Guide defines ‘user requirements’ in a restricted way as the requirements of the system’s customers or end-users. System requirements, by contrast, encompass user requirements, requirements of other stakeholders (such as regulatory authorities), and requirements without an identifiable human source.

Tabla 1. Ejemplo del contenido del Cuerpo de Conocimiento de la Ingeniería de Software

<sup>5</sup> Los “Cuerpos de Conocimiento” son estructuras del conocimiento, definidas por grupos de expertos en el tema, generalmente tipo todo-parte. Es una especie de taxonomía generalizada.

**Definición 2.** Un *Principio de Complejidad* ( $P_i$ ) constituye un conjunto de una o más características de la complejidad, nombradas en forma única. El *conjunto de Principios PR*, constituye la definición de la Complejidad

$$PR = \langle P_1, P_2, \dots, P_n \rangle$$

Normalmente en los trabajos sobre complejidad se define a esta última mediante la especificación y descripción de sus características ((Seel, 1999), (Cilliers, 2008)). De acuerdo al enfoque de este trabajo, cada característica constituye un “Principio”. Cabe anotar que la lista de características varía según el enfoque y caso de estudio.

Una posibilidad para especificar PR es el enfoque de Morin, que consolida las características de complejidad en “Principios”: Sistémico u organizativo, Holográfico, de Retroalimentación, de Recursión, de Autonomía-Dependencia, Dialógico y de Reintroducción del Cognoscente en el Conocimiento (Morin, 2003). Esta posibilidad está pendiente de explorar y desarrollar.

En general, un Principio contendrá un conjunto de características. Por ejemplo un Principio de Retroalimentación puede componerse de: retroalimentaciones positivas, retroalimentaciones negativas, homeóstasis, histéresis generalizada, entre otras.

Para lo que viene a continuación, se adopta la notación de “Principio” en su definición general.

**Definición 3.** Una *función de ponderación sistémica* es una función matemática lineal o no-lineal, discreta o real; y, que genera valores que determinan la importancia del ítem a ser ponderado.

**Definición 4.** Una *función de acumulación sistémica* es una función matemática lineal o no lineal, discreta o real; y, representa acumulación, en sentido positivo o negativo, es decir, permite incorporar complejidades parciales a la total.

**Definición 5.** Las *Complejidades Parciales* ( $CP_i$ ), se definen en función de los principios ( $PR$ ) y del Contenido del Sistema ( $CS$ ). Así:

$$CP_i(S) = F_{2i} (F_{3i} (CS), CS, PR)$$

Donde  $F_{3i}$  es una función de ponderación sistémica del contenido CS para la i-esima complejidad parcial. La función  $F_{2i}$  es una función de acumulación sistémica de la influencia (presencia) de  $PR$  en  $CS$  para la i-esima complejidad parcial

**Definición 6.** La *Complejidad Total de un Sistema S*, *Complejidad(S)*, o *Métrica de Complejidad*, es el resultado de la acumulación de las Complejidades Parciales del Sistema S,  $CP_i(S)$ , mediante una función de acumulación sistémica general  $F_1$ :

$$Complejidad(S) = F_1 (CP_1(S), CP_2(S), \dots, CP_i(S), \dots, CP_r(S))$$

Complejidad(S) cumple con los axiomas y propiedades descritos con anterioridad.

El modelo de la Métrica de Complejidad, en total concordancia con el carácter inacabado del paradigma de complejidad, deja abierto la posibilidad de explorar el tipo de funciones  $F_1$ ,  $F_2$ ,  $F_3$  que se pueden usar, de acuerdo al caso de estudio particular.

### 3. EL MARCO DE TRABAJO Y LOS PUNTOS DE FUNCION (PF)

Un ejemplo interesante en el que es posible aplicar el Marco de Trabajo planteado, en la línea de Ingeniería de Software, es el Cálculo de Puntos de Función (PF), que utiliza un modelo que incluye operaciones y ponderaciones con valores constantes. El valor de PF en última instancia es una medida de la complejidad sobre la base de ciertos contenidos del sistema de software. A continuación se resume el modelo en su versión básica.

Para calcular los puntos de función (PF) de una aplicación de software se utiliza la siguiente relación

$$PF = CUENTA\_TOTAL * [0.65 + 0.01 * SUM(fi)]$$

Donde CUENTA\_TOTAL es la suma de las entradas obtenidas de la siguiente tabla.

Parámetros de medición	Cuenta	Factor de ponderación			=	Cuenta
		Simple	Medio	Complejo		
Número de entradas de usuario	<input type="text"/>	x 3	4	6	=	<input type="text"/>
Número de salidas de usuario	<input type="text"/>	x 4	5	7	=	<input type="text"/>
Número de peticiones de usuario	<input type="text"/>	x 3	4	6	=	<input type="text"/>
Número de archivos	<input type="text"/>	x 7	10	15	=	<input type="text"/>
Número de interfaces externas	<input type="text"/>	x 5	7	10	=	<input type="text"/>
Cuenta total	→					<input type="text"/>

Tabla 2. Factores de Ponderación de los Parámetros

$f_i$ , donde  $i$  toma valores de 1 hasta 14, son los valores de ajuste de complejidad basados en las respuestas a las preguntas señaladas de la siguiente lista (Cada factor se evalúa en escala 0 a 5).

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?
2. ¿Se requiere comunicación de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento?
5. ¿Se ejecutará 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 u operaciones?
8. ¿Se actualizan 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?

La relación entre los conceptos de la Métrica de Complejidad (MC) y el modelo de PF se muestra a continuación:

MAPEO MC vs PF		
MC	PF	
CONTENIDO	Parametros de Medicion de la Complejidad (Areas Clave)	<ol style="list-style-type: none"> <li>1. Salidas de Usuario</li> <li>2. Entradas de Usuario</li> <li>2. Peticiones de Usuario</li> <li>3. Archivos</li> <li>4. Interfases externas</li> </ol>
PRINCIPIOS	Factores de Complejidad	<ol style="list-style-type: none"> <li>1. Fiabilidad de copias de seguridad y de recuperación</li> <li>2. Existencia de comunicación de datos</li> <li>3. Existencia de procesamiento distribuido</li> <li>4. Criticidad del rendimiento</li> <li>5. Grado de utilizacion del entorno operativo</li> <li>6. Existencia de entrada de datos interactiva</li> <li>7. Existencia de entrada de datos interactiva con transacciones sobre múltiples pantallas u operaciones?</li> <li>8. Actualizacion interactiva de archivos maestros</li> <li>9. Complejidad de las entradas, las salidas, los archivos y las peticiones</li> <li>10. Complejidad del procesamiento interno</li> <li>11. Existencia de diseño de código reutilizable</li> <li>12. Presencia de la conversión y la instalación en el diseno</li> <li>13. Grado de diseño para soportar múltiples instalaciones en diferentes organizaciones</li> <li>14. Grado del diseño para facilitar los cambios y para ser fácilmente utilizada por el usuario</li> </ol>
FUNCION DE PONDERACION	Tabla del Factor de Ponderacion de los Parametros de la Complejidad	
	Evaluacion de 0 a 5 de los Factores de Complejidad fi (Ajuste), según lista de preguntas	
FUNCION DE ACUMULACION	$CUENTA\_TOTAL * [0.65 + 0.01 * SUM(fi)]$	
COMPLEJIDADES PARCIALES	En este caso se tiene el calculo de un solo valor: PF	
COMPLEJIDAD TOTAL	$PF = CUENTA\_TOTAL * [0.65 + 0.01 * SUM(fi)]$	

Tabla 3. Mapeo MC vs PF

Los “Parámetros de Medición de la Complejidad” constituyen las Áreas Clave de un Sistema de Software, las mismas que se cualifican y cuantifican con los Puntos de Función. Los “Factores de la Complejidad” son los Principios que en este caso interrogan sobre ciertas características de diseño y funcionales. La función de ponderación tiene dos componentes: una tabla con valores discretos no lineales predefinidos aplicable a los parámetros; y, un rango discreto lineal que permite la calificación de los factores de complejidad. La función de acumulación es lineal. Finalmente, la fórmula de cálculo de los Puntos de Función (PF) es no lineal.

Con relación a los axiomas y propiedades, en este caso su aplicación significa lo siguiente:

- Los PF de un Sistema de software de varios componentes (subsistemas) conectados, es *mayor o igual* a la suma de los PF de un subconjunto de ellos o de todos ellos. Referencia: Axioma 1.
- Los PF de un Sistema de software de varios componentes (subsistemas) desconectados (o aparentemente conectados) es *igual* a la suma de los PF de cada uno de ellos. Referencia: Axioma 2.
- Si se conecta un componente (subsistema) a un Sistema, su PF *aumenta*. Referencia: Propiedad 3.

Como fruto de la aplicación de los axiomas y propiedades de la complejidad, las conclusiones anteriores son directas y no requieren prueba.

Al modelo básico de PF se añaden detalles debidamente validados. Por ejemplo, en referencia a la Tabla 2, en el ítem del *Número de Entradas de Usuario*, el *Factor de Ponderación* correspondiente se calcula con la siguiente Tabla, usando información adicional de los archivos referenciados y los elementos de datos involucrados (Longstreet):

Factor de Ponderacion de las Entradas de Usuario			
Archivos Referenciados	Elementos de Datos		
	1 a 4	5 a 15	Mayor a 15
Menor a 2	Bajo (3)	Bajo (3)	Promedio (4)
Igual a 2	Bajo (3)	Promedio (4)	Alto (6)
Mayor a 2	Promedio (4)	Alto (6)	Alto (6)

Tabla 4. Ponderación de las Entradas de Usuario

Igualmente, el factor de complejidad referente a la existencia de la comunicación de datos se califica de acuerdo a la siguiente tabla.

Calificacion del Factor de Complejidad: Comunicacion de Datos	
Calificación	Descripción del grado de influencia
1	Aplicacion Batch o de PC
2	Aplicacion Batch con entradas o impresion remotas
3	Aplicacion incluye entrada de datos en linea o teleprocesamiento con acceso a un proceso batch o sistema de consulta
4	La aplicacion es mas que un front end y utiliza un solo protocolo de comunicacion
5	La aplicacion es mas que un front end y utiliza mas de un protocolo de comunicacion

Tabla 5. Calificación de la Comunicación de Datos

Como es evidente, los detalles completan y afinan el modelo, no lo cuestionan. En el ejemplo anterior, detallan el *Contenido* (Áreas Claves) y los *Principios* (Factores de Complejidad).

#### 4. DISCUSION

Como se ha evidenciado, se define un Marco de Trabajo que parte de un enfoque formal de la complejidad y establece pautas generales para la definición de métricas.

El enfoque formal es importante pues predefine y prueba las características generales de las métricas, preservando adicionalmente su consistencia. Esto evita pruebas empíricas mediante “casos de prueba”, cuya consistencia y completitud normalmente son difíciles de conseguir.<sup>6</sup>

Las métricas se refieren a los *Principios*, razón por la cual pueden ser métricas específicas si se refieren a elementos de un Principio (por ejemplo la complejidad funcional), generales si se refieren a uno o más principios (por ejemplo, al Principio de Retroalimentación), o sistémicos si se refieren a un conjunto “universal”<sup>7</sup> de Principios.

El modelo general de la métrica, en total concordancia con un paradigma complejo, deja abierto la posibilidad de explorar en el tipo de funciones que se pueden usar para los cálculos. Cabe resaltar que se plantea que las funciones de ponderación y acumulación pueden generar valores positivos o negativos cuando el ítem aporta positivamente o negativamente a la complejidad. Si bien es cierto que la presencia de un nuevo componente en un sistema aumenta la complejidad de su estructura, este ítem puede aportar negativamente a la complejidad de su comportamiento en el momento que su presencia “ordena el desorden”.<sup>8</sup>

La Métrica de Complejidad del Sistema, abarca en general dos componentes: la complejidad de la estructura del Sistema y la complejidad de su comportamiento. Sin embargo, el cálculo expreso de la complejidad estructural incluye la consideración de la topología detallada del Sistema; y, por otro lado, el cálculo expreso de la complejidad de comportamiento incluye la consideración de las características algorítmicas. En este trabajo se adopta un enfoque global a partir del *Contenido* y los *Principios*, los mismos que permiten calcular una Complejidad sistémica, que implícitamente contiene la complejidad estructural y la de comportamiento.

Finalmente, el uso del Marco, en concordancia con su contenido, parte de consideraciones conceptuales cualitativas puesto que se requiere contar con dos ítems: el *Contenido* y los *Principios*, sobre cuya base se aplican las funciones cuantitativas de cálculo. Este hecho posibilita que la métrica pueda ser mejorada o afinada mediante a la acumulación de detalles (por

---

<sup>6</sup> Las pruebas son procesos complejos. Considérese el caso de los Sistemas de Software, cuya prueba exhaustiva generalmente es imposible y cuya consistencia suele ser dependiente de la metodología de desarrollo que se usa. El fin de las pruebas, en este ejemplo, se transforma en un proceso probabilístico, empírico o intuitivo. En sistemas físicos las pruebas son imposibles o altamente costosas: un puente por ejemplo.

<sup>7</sup> El carácter “universal” de los Principios, contradictoriamente es relativo pues la complejidad de los sistemas debe redefinirse según el enfoque del cognoscente (que establece los límites artificiales o reales).

<sup>8</sup> Según Morin existe una relación íntima entre orden y desorden. Esto significa que existen complejidades del orden y del desorden y que su variación positiva o negativa depende de la relación entre estos conceptos. Por ejemplo, en sistemas con lenguaje gráfico, el desorden es sinónimo de complejidad y el orden la disminuye.

ejemplo incluir detalles en el Contenido, características en un Principio o cambiar una función), así como, en términos epistemológicos, posibilita un cambio cualitativo o “ruptura” (por ejemplo incluir o eliminar Principios o redefinir el Contenido).

Con estos antecedentes, el Marco de Trabajo definido puede usarse para las aplicaciones siguientes:

- Comparación con Marcos de Trabajo -explícitos o implícitos-, subyacentes a las métricas de Complejidad.
- Definición de métricas en ámbitos variados del conocimiento, sobre la base del Marco de Trabajo, particularmente a partir de Contenidos complejos (como los Cuerpos de Conocimiento) y de variados Principios de Complejidad que sea posible identificar y medir. Está pendiente la posibilidad de adoptar enfoques estructurados de los Principios, a partir de la Epistemología compleja.
- Comparación y evaluación de modelos de métricas a la luz de las características formales del Marco de Trabajo.
- Análisis de características implícitas y explícitas de los modelos de métricas, sobre la base de los axiomas, propiedades, operaciones y definiciones del Marco de Trabajo.
- Mejoramiento y redefinición de métricas que se ajusten al esquema planteado

## REFERENCIAS

Adami, C. (2002). What is complexity?. *BioEssays*. 24:1085–1094, Wiley Periodicals, Inc.

Bar-Yam, Y. (2000). General features of complex systems. New England Complex Systems Institute. Cambridge, MA, USA. Recuperado de: [www.sci.unich.it/~aroli/dida/iasc/.../EOLSSBriefIntrotoCX.pdf](http://www.sci.unich.it/~aroli/dida/iasc/.../EOLSSBriefIntrotoCX.pdf)

Cilliers, P. (2008). Knowing Complex Systems: The limits of understanding. In *A Vision of Transdisciplinarity; Laying Foundations for a World Knowledge Dialogue*. (eds.) Darbellay, F., Cockell, M., Billotte, J. and Waldvogel, F. Lausanne, France. EPFL & CRC Press.

Couture, M. (2007a). *Complexity and chaos – State-of-the-art; Overview of theoretical concepts*. (Technical Memorandum DRDC Valcartier TM 2006-453). Defence R&D Canada-Valcartier. August.

Couture, M. (2007b). *Complexity and Chaos - State-of-the-Art; Formulations and Measures of Complexity*. (Technical Memorandum DRDC Valcartier TM 2006-451). Defence R&D Canada-Valcartier. September.

Couture, M. (2007c). *Complexity and Chaos - State-of-the-Art; List of Works, Experts, Organizations, Projects, Journals, Conferences and Tools*. (Technical Memorandum DRDC Valcartier TM 2006-450). Defence R&D Canada-Valcartier. September.

Couture, M. (2007d). *Complexity and Chaos - State-of-the-Art; Glossary*. (Technical Memorandum DRDC Valcartier TM 2006-452). Defence R&D Canada-Valcartier. September.

Edmonds, B. (1997). Complexity and scientific modelling. Proc. 20th Int'l Wittgenstein Symposium, Austria. August.

Fenton, N. (1994). Software Measurement: A Necessary Scientific Basis. IEEE Transactions on Software Engineering. Vol. 20, No. 3. March.

Firesmith, D. (2010). Profiling Systems Using the Defining Characteristics of Systems of Systems (SoS). (Technical Note CMU/SEI-2010-TN-001). February

IEEE Computer Society. (2004). Guide to the Software Engineering Body of Knowledge. SWEBOK. California, EEUU.

Longstreet, D. Function Points Analysis Training Course. Recuperado de <http://www.SoftwareMetrics.Com>

Park, K. (2005). The Internet as a complex system. Chapter 1 (pp. 1-89) of *The Internet as a Large-Scale Complex System*, K. Park and W. Willinger (eds.), SFI Studies in the Sciences of Complexity, Oxford University Press. Recuperado de <http://www.cs.purdue.edu/homes/park/interest-synopsis.html>

Maldonado, C. y Gómez, N. (2010). *El mundo de las ciencias de la complejidad. Un estado del arte*. Bogotá, Colombia. Editorial Universidad del Rosario. Mayo

Magee, C. L. y Weck, O. L. (2008). Complex System Classification. Fourteenth Annual International Symposium of the INCOSE. June 20 - 24. Recuperado de: <http://dspace.mit.edu/bitstream/handle/1721.1/6753/INCOSEComplex%20System%20Classification.doc.txt?sequence=6>.

Morasca, S. (1996). Property-based software engineering measurement. IEEE Transactions on Software Engineering. Vol. 22, No. 1. January

Morasca, S. (2008). Refining the Axiomatic Definition of Internal Software Attributes. Recuperado de [portal.acm.org/citation.cfm?id=1414035](http://portal.acm.org/citation.cfm?id=1414035)

Morin, E. (1998). Introducción al pensamiento complejo. Barcelona, España. Editorial Gedisa.

Morin, E. (2003). Educar en la era planetaria. Barcelona, España. Editorial Gedisa.

Seel, R. (1999). Complexity & OD. An Introduction. Recuperado de <http://www.new-paradigm.co.uk/complex-od.htm>

Stepanic, J. (2010). Interdisciplinary description of Complex Systems. Scientific Journal. Vol 8, pp. 1-58. July. Recuperado de: [http://indecs.eu/2009/indecs\\_7\\_2.pdf](http://indecs.eu/2009/indecs_7_2.pdf).

Weaver, W. (1948). Science and Complexity. *American Scientist*, 36: 536.