

**HUACHO- PERU**

**FACULTAD: Ingeniera Industrial, Sistemas e Informática**

**ESCUELA: Ingeniería Informática**

**DOCENTE: Nicho Viru, Wigberto Marin**

**CURSO: Calidad de Software**

**ALUMNOS:**

1. **Luis Mendoza, Samuel**
2. **Porlles Pardo, Julio**
3. **Montoro Correa, Jesús Alberto**
4. **Meléndez Panana, Cesar Yair**
5. **Príncipe Henostroza, Jhordy Anderson**
6. **Lopez Castro, Angello Jones**
7. **Guerrero Sánchez, Weiner Brayan**
8. **Sandon Mateo, Amilkar**
9. **Solis Ocaña, Luis**
10. **Valenzuela Paucar, Junior**
11. **Rivera La Rosa, Jaime Jacob**
12. **Yanac Jiménez, Daniel Nehemias**
13. **León Crispín, Mao Jovaldo**
14. **Marin Nuñez, Jose Fernando**
15. **Silva Dolores, Cristhian Jean Piere**
16. **Calcina Pacori, Julio Rolando**

**EMPRESA “PAPA NOEL”**

|  |  |  |
| --- | --- | --- |
|  | **Sub Carácterística** | **Descripción** |
| **Funcionalidad** | Aplicabilidad | La capacidad del producto software para proveer un conjunto apropiado de  funciones para las tareas y objetivos especificados por el usuario. |
| Precisión | La capacidad del producto software para proveer los resultados o efectos acordados  con un grado necesario de precisión. |
| Conformidad de la funcionalidad | La capacidad del producto software de adherirse a los estándares, convenciones o  regulaciones legales y prescripciones similares referente a la funcionalidad. |
| Recuperabilidad | La capacidad del producto software para reestablecer un nivel especificado de  funcionamiento y recuperar los datos afectados directamente en el caso de una falla. |
| **Usabilidad** | Entendibilidad | La capacidad del producto software para permitir al usuario entender si el software  es aplicable, y cómo puede ser utilizado para las tareas y las condiciones particulares de la aplicación. |
| Facilidad de aprendizaje | La capacidad del producto software para permitir al usuario aprender su aplicación. |
| Operabilidad | La capacidad del producto software para permitir al usuario operarlo y controlarlo. |
| Atractividad | La capacidad del producto software de ser atractivo al usuario. |
| Conformidad de usabilidad | La capacidad del producto software para adherirse a las normas, convenciones,  guías de estilo o regulaciones relacionadas a su usabilidad. |
| Utilización de recursos | La capacidad del producto software para utilizar apropiadas cantidades y tipos de  recursos cuando éste funciona bajo las condiciones establecidas. |
| Conformidad de eficiencia | La capacidad del producto software para adherirse a normas o convenciones  relacionados a la eficiencia. |
| **Mantenibilidad** | Analizabilidad | La capacidad del producto software para ser diagnosticado por deficiencias o causas  de fallas en el software o la identificación de las partes a ser modificadas. |
| Modificabilidad | La capacidad del software para permitir que una determinada modificación sea  implementada. |
| Capacidad de ser probado | La capacidad del software para permitir que las modificaciones puedan ser  validadas. |
| Reusabilidad | La capacidad del producto software para ser utilizado en lugar de otro producto  software, para el mismo propósito y en el mismo entorno. |

**Métricas para un sistema Web**

Las métricas de software se pueden dividir en dos categorías: métricas de control y métricas de predicción. Las métricas de control son usadas para controlar el proceso de software. Ejemplos de este tipo de métricas son el tiempo y esfuerzo invertido. La estimación y medida de estas métricas pueden ser usados para mejorar el proceso de desarrollo del software, con lo que se pueden lograr productos finales de mejor calidad. **Por otra parte, las métricas de predicción son medidas de atributos de un producto,** que sirven para predecir cosas sobre el producto. Por ejemplo, se puede predecir la facilidad de mantenimiento de un componente de software midiendo su complejidad ciclomática. (métrica del software en ingeniería del software que proporciona una medición cuantitativa de la complejidad lógica de un programa)

Con el fin de poder estimar con mayor precisión el tipo y cantidad de mantenimiento que requiere un sistema Web, hemos desarrollado algunas métricas de predicción basadas en los tipos de archivos que componen el sistema completo. Estas métricas también son útiles para determinar el esfuerzo requerido para desarrollar, a futuro, sistemas con características similares. Las dos métricas se refieren a la funcionalidad del sistema Web, y al diseño gráfico del mismo.

1. **Funcionalidad en el sistema web**

**Scripts de servidor (server-side scripts)**

Normalmente, un servidor Web está asociado a diversas aplicaciones que pueden ejecutar o interpretar scripts del lado del servidor. Estos scripts generan las páginas que finalmente son enviadas al cliente, y pueden ser de tres tipos:

**1. Interpretados:** para los cuales es necesario invocar un parser que permita compilar y ejecutar una aplicación de este tipo. Ejemplos de esta clase de aplicación son las tecnologías PHP (HyperText Processor), Microsoft Active Server Pages (ASP) y Cold Fusion.

**2. Compilados:** para los cuales un binario es invocado directamente para generar la página final. Ejemplos de este tipo son las tecnologías Microsoft ISAPI y Netscape NSAPI.

**3. Híbridos:** que son un punto medio entre los dos tipos anteriores. Un ejemplo de esta categoría lo constituyen las Java Server Pages (JSP), de Sun MicroSystems. Una página JSP es un script que, la primera vez que es solicitado, es compilado y almacenado en el lado del servidor. Las siguientes peticiones de la página generada por este script son respondidas con la invocación del binario que resultó de la primera compilación.

1. **Funcionalidad según tipos de archivos**

A pesar de la forma en que pueden estar mezclados los distintos tipos de funcionalidad mencionados anteriormente, es posible determinar con exactitud la forma de un sistema Web, estudiando los tipos de archivos que lo componen. La tabla 1 presenta una clasificación de los tipos de archivos que normalmente se encuentran presentes en un sistema Web.

Esta última métrica pretende medir (con un valor entre 0 y 1) la funcionalidad presente en un sistema Web y provee mayor precisión para poder hablar de sitios o aplicaciones Web. Si aplicamos la métrica a un sistema Web, según el valor del índice podemos decir lo siguiente:

**Tabla 1.**

Tipos de archivos que componen un sistema Web

|  |  |  |
| --- | --- | --- |
| **Tipo de Archivo** | **Extensiones de archivo** | **Descripción** |
| Estructurales | Html, ihtml | Archivos que contienen la descripción lógica de las interfaces de los objetos que componen un sistema web. |
| Funcionales (servidor) | Php, jsp, asp, pl, cgi | Archivos que implementan funcionalidad de un Sistema Web, del lado del servidor. |
| Funcionales (cliente) | Js, css, vs Archivos | Archivos de funcionalidad externa, que pueden ser visualizados a través de una funcionalidad añadida al browser, como applets, plug-in’s, etc. |
| Funcionales (incrustados) | Class, swf, dir | Archivos de funcionalidad externa, que puden ser visualizados a través de una funcionalidad añadida al browser , como applets, plug-in’s, etc |
| Imágenes | Gif, jpg, bmp | Archivos binarios de imágenes en varios formatos. A pesar de que existen muchos más formatos de imágenes, sólo estos son actualmente desplegables a través de browser. |
| Documentos | Pdf, ps, doc, xls, ppt, rtf | Documentos varios que pueden ser desplegados o bajados (download) |
| Otros |  | Archivos que no encajan en los tipos anteriores |

1. **Índice de funcionalidad**

De los archivos incluidos en la clasificación de la tabla 1, para medir funcionalidad nos interesan principalmente los archivos estructurales y funcionales. Si, dado un conjunto de archivos, consideramos sus pesos en kilobytes, podemos definir la siguiente métrica, llamada Índice de Funcionalidad (IF)

donde TIPO puede corresponder a uno de los tres tipos de archivos funcionales en un sistema: de servidor, de cliente o incrustada. Dado esto, se define un Índice de Funcionalidad Neto (IFN) como una combinación lineal de los valores anteriores:

Con

* Si IFN < 0,2, hablamos de un sitio Web no funcional.
* Si 0, 2 ≤ IFN < 0,5, hablamos de un sitio Web funcional.
* Si 0, 5 ≤ IFN, hablamos de software en Web o aplicación Web.

1. **Métricas Orientadas a la Función**

Las métricas del software orientadas a la función utilizan una medida de la funcionalidad entregada por la aplicación como un valor de normalización. Ya que la «funcionalidad>>no se puede medir directamente, se debe derivar indirectamente mediante otras medidas directas.

Las métricas orientadas a la función fueron propuestas por primera vez por **Albretch**, quien sugirió una medida llamada punto defunción.

Los puntos de función se derivan con una relación empírica según las medidas contables (directas) del dominio de información del software y las evaluaciones de la complejidad del software.

|  |  |
| --- | --- |
| **Número de entradas de usuario** | Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada. |
| **Número de salidas de usuario** | Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. |
| **Número de peticiones de usuario.** | Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. |
| **Número de archivos.** | Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente). |
| **Número de interfaces externas** | Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema. |

Una vez que se han recopilado los datos anteriores, a la cuenta se asocia un valor de complejidad. Para calcular puntos de función (PF), se utiliza la (Figura 2) relación siguiente:

**Pasos para hallar las medidas de función con el método de Puntos de Función de Albrecht**

1. Llenar la columna de CUENTA de la siguiente tabla de valores, donde se determinan 5 características del ámbito de la información.

Tabla 3

Tabla de valores del dominio de la información.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parámetro | Cuenta | Factor de Ponderación | | | | | Subtotal |
| Simple | Medio | | Complejo | |
| Entradas de usuario | 3 | 4 | | 6 | | 6 |  |
| Salidas de usuario | 4 | 5 | | 7 | | 7 |  |
| Peticiones de usuario | 3 | 4 | | 6 | | 6 |  |
| Archivos | 7 | 10 | | 15 | | 15 |  |
| Interfaces externas | 5 | 7 | | 10 | | 10 |  |
| **CUENTA TOTAL** | | | | | |  |  |

1. Multiplicar el valor de CUENTA por el Factor de Ponderación indicado, dependiendo de su complejidad para obtener Subtotal. Se asocia un valor de complejidad a cada medida, tomando en cuenta la heurística de la siguiente tabla.

Tabla 4

Tabla de complejidades.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipos de archivos referenciados | Tipos de datos elementales | | |
| 1-5 | 6-19 | 20+ |
| 0-1 | Bajo | Bajo | Medio |
| 2-3 | Bajo | Medio | Alto |
| 4+ | Medio | Alto | Alto |

1. Obtener CUENTA TOTAL
2. Evaluar los siguientes 14 atributos que impactan en el desarrollo, en escala de 0 a 5.

Tabla 5

Factores de influencia en la dificultad del sistema.

|  |  |
| --- | --- |
| **Factores de Influencia en la dificultad del sistema** | **Ejemplo** |
| 1. Comunicaciones de Datos | Una aplicación para el sector bancario, donde se requieren numerosas transacciones monetarias. |
| 1. Procesamiento distribuido | Un motor de búsqueda en Internet, donde el procesamiento está distribuido en decenas de máquinas. |
| 1. Objetivos de rendimiento | Una aplicación para el control del tráfico aéreo, que debe proporcionar continuamente información precisa. |
| 1. Configuración de uso intensivo | Un sistema para matriculas en una universidad, donde concurren cientos de alumnos al mismo tiempo. |
| 1. Tasas de transacción rápidas | Una aplicación para el sector bancario, donde deben realizarse millones de transacciones durante la noche. |
| 1. Entrada de datos en línea | Un programa en el que los datos de entrada provienen de papeles o formularios impresos. |
| 1. Amigabilidad en el diseño | Un programa de análisis financiero utilizado por el directivo de una empresa capaz de orientarte y asesorarte. |
| 1. Actualización de datos en línea | Una aplicación para reserva de billetes en la que deben bloquearse y modificarse ciertos registros en las BB.DD. para evitar que un mismo asiento sea vendido dos veces.. |
| 1. Procesamiento Completo | Un sistema para diagnóstico médico, el cual realiza costosas operaciones de decisión lógica hasta obtener un resultado. |
| 1. Reusabilidad | Un procesador de textos en el que, por ejemplo, su barra de menús puede utilizarse desde una hoja de cálculo, un generador de informes de una base de datos, etc. |
| 1. Facultad de Instalación | Cualquier aplicación de propósito general, de tal forma que cualquier persona pueda realizar la instalación fácilmente. |
| 1. Facilidad Operacional | Una aplicación para tratamiento de grandes cantidades de información, donde es muy importante la efectividad de los procesos de backup y recuperación de datos. |
| 1. Adaptabilidad | Una aplicación software para una multinacional con oficinas en varios países. |
| 1. Versatilidad | Un sistema que admite diversas situaciones de uso, tanto para facilitar cambios como para ser utilizada por el usuario. |

A cada atributo se le asignará un valor dependiendo del grado de influencia de éstos.

* **Sin Influencia (0).** El sistema no contempla este atributo.
* **Influencia Mínima (1).** La influencia de este atributo es muy poco significativa.
* **Influencia Moderada (2).** El sistema contempla este atributo y su influencia, aunque pequeña, ha de ser considerada.
* **Influencia Apreciable (3).** La importancia de este atributo debe ser tenida en cuenta, aunque no es fundamental.
* **Influencia Significativa (4).** Este atributo tiene una gran importancia para el Sistema.
* **Influencia Muy Fuerte (5).** Este atributo es esencial para el sistema y se lo debe tomar en cuenta a la hora del diseño.

5. Sumar los puntos asignados a cada factor y obtener un TOTAL GI que indica un valor de ajuste de complejidad.

6. Calcular Punto de Función, utilizando la siguiente fórmula:

**GI=** Valor de ajuste de complejidad.

Los valores constantes de la ecuación anterior y los factores de peso aplicados en las encuestas de los ámbitos de información han sido determinados empíricamente.

Una vez calculado el punto de función se usan de forma analógica a las LDC como medida de la productividad, calidad y otros productos del software.

* Productividad = PF / Eficiencia
* Calidad = Errores / PF
* Costo = Dólares / PF
* Documentación = Págs. Doc. / PF

**Exactitud**

La capacidad del producto software para proveer los resultados o efectos acordados.

Propósito: ¿Cuán frecuente los usuarios finales encuentran resultados inexactos?

**Pasos para medir la exactitud**

Registrar el número de cálculos inexactos basado en especificaciones

**Aplicar la Fórmula**

|  |
| --- |
| X=A/T |
| A= Número de cálculos inexactos encontrados por los usuarios  T= Tiempo de operación |

A= Número de cálculos inexactos encontrados por los usuarios

T= Tiempo de operación

NOTA: Mientras 0 <= X , lo más cercano a 0,0 es lo mejor.

**Audiencia y objetivo:** Desarrollador y usuario

**e**