**GUÍA DE APRENDIZAJE 3 CLASIFICACIÓN DE METRICAS**

**JENNIFER ANDREA ROMANI JAMAICA**

**JUAN DAVID SABOYÁ JIMENEZ**

**SERVICIO NACIONAL DE APRENDIZAJE (SENA)**

**CENTRO DE DISEÑO Y METROLOGÍA**

**ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN**

**FICHA 683019**

**BOGOTÁ D.C.**

**2015**

**TABLA DE CONTENIDO**

[3.1 Actividades de reflexión inicial 3](#_Toc429478224)

[3.1.1 Elabore Factores de Calidad de McCall A 10 programas que usualmente utiliza 4](#_Toc429478225)

[3.2.1 Realizar la métrica bang a 10 programas 4](#_Toc429478226)

[3.3.1. Realizar la métrica del modelo diseño a 10 programas 4](#_Toc429478227)

[3.3.2. Realizar la métrica del código fuente a 10 programas 4](#_Toc429478228)

[3.3.2.1 Definición métrica del código fuente 4](#_Toc429478229)

[3.3.2.2 Cálculo de los 10 programas 7](#_Toc429478230)

[3.3.3. Realizar las métricas para pruebas a 10 programas 12](#_Toc429478231)

[3.3.3.1 Definición métrica para pruebas 12](#_Toc429478232)

[3.3.3.2 Cálculo de los 10 programas 13](#_Toc429478233)

[3.3.4. Realizar la métrica de mantenimiento para prueba a 10 programas 15](#_Toc429478234)

[3.3.4.1 Definicion métrica de mantenimiento 15](#_Toc429478235)

[3.3.4.2 Cálculo de los 10 programas 16](#_Toc429478236)

[REFERENCIAS 18](#_Toc429478237)

# 3.1 Actividades de reflexión inicial

Identificar conocimientos previos, como estrategia para garantizar la aplicación de los Modelos de calidad en el desarrollo de software: implantación, objetivos organizacionales, técnicas de entrenamiento, definición de planes de mejoramiento (acciones correctivas, preventivas y de mejoramiento), conocimientos previos de procesadores de texto, flujo de información, redacción y ortografía, didáctica, software para presentaciones, Internet, navegadores y otros sistemas y herramientas tecnológicas necesarias.

**Programas a los que se les realizarán las métricas:**

1. Skype
2. Facebook
3. Gmail
4. WhatsApp
5. Twitter
6. Instagram
7. Sena Sofía
8. Google
9. YouTube
10. Hotmail

# 3.1.1 Elabore Factores de Calidad de McCall A 10 programas que usualmente utiliza

# 3.2.1 Realizar la métrica bang a 10 programas

# 3.3.1. Realizar la métrica del modelo diseño a 10 programas

# 3.3.2. Realizar la métrica del código fuente a 10 programas

## 3.3.2.1 Definición métrica del código fuente

La teoría de la ciencia del software propuesta por Halstead es probablemente la medida de complejidad mejor conocida y minuciosamente estudiada. La ciencia del software propuso la primera ley analítica y cuantitativa para el software de computadora.

Halstead utiliza medidas primitivas para desarrollar expresiones par la longitud global del programa; volumen mínimo potencial para un algoritmo; el volumen real (número de bits requeridos para especificar un programa); el nivel del programa (una medida de la complejidad del software); nivel del lenguaje (una constante para un lenguaje dado); y otras características tales como el esfuerzo de desarrollo, tiempo de desarrollo e incluso el número esperado de fallos en el software.

**Halstead propone las siguientes métricas:**

· Longitud N se puede estimar como: N = n1 log2 n1 + n2 log2 n2.

· Volumen de programa se define como: V = N n1 log2 (n1 + n2).

Tomando en cuenta que V variará con el lenguaje de programación y representa el volumen de información (en bits) necesarios para especificar un programa.

**Medidas:**

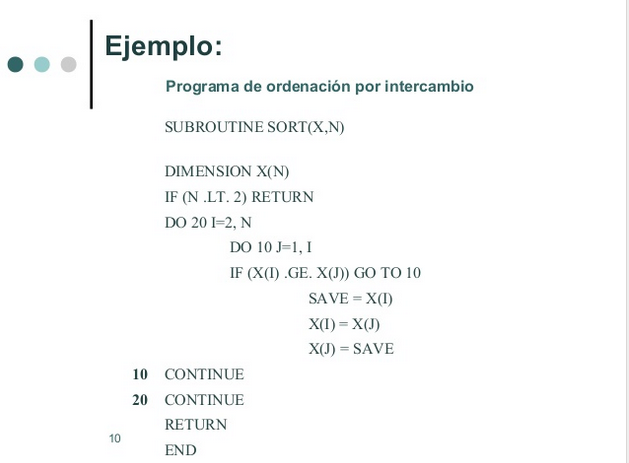
**n1:** número de operadores diferentes que aparecen en el programa.

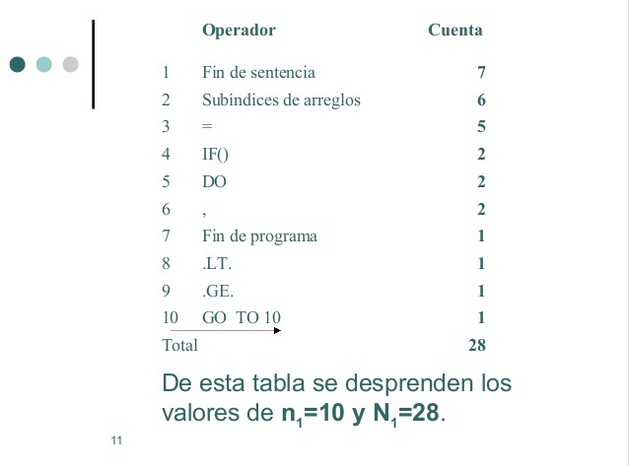
**n2:** número de operandos diferentes que aparecen en el programa.

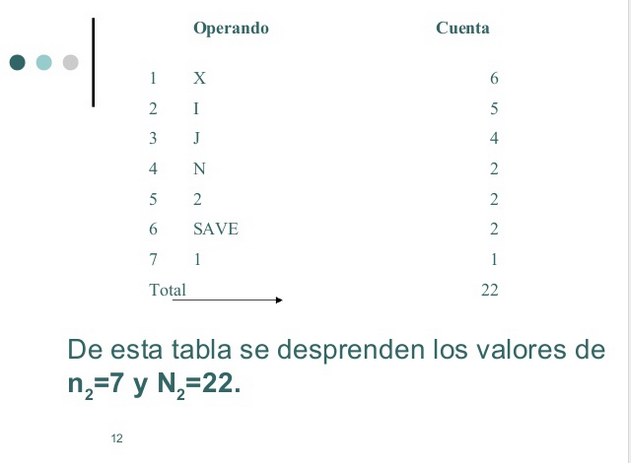
**N1:** número total de veces que aparece el operador.

**N2:** número total de veces que aparecen el operando.

**Ejemplo:**

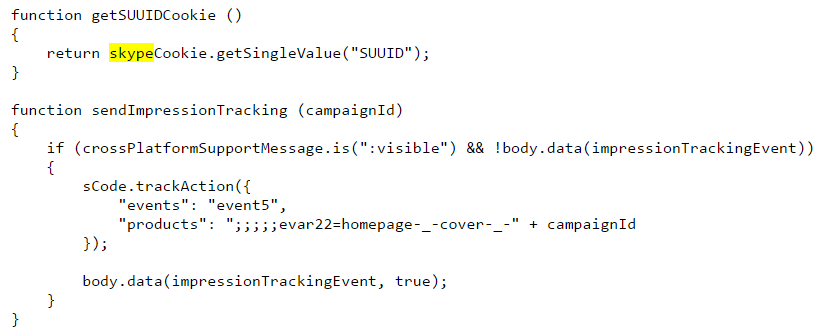
****

****

****

## 3.3.2.2 Cálculo de los 10 programas

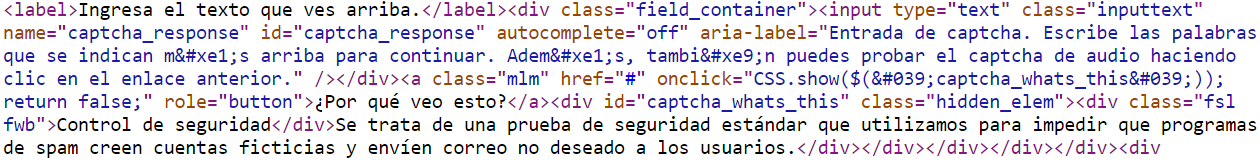
1. Skype



**N = n1 log2 n1 + n2 log2 n2 => N=25 log2 25+9 log2 9 =4.64 + 3.16 = 7.8**

**V = N n1 log2 (n1 + n2) => 10\*25 log2 (25+9)= 5.12 (34) = 174**

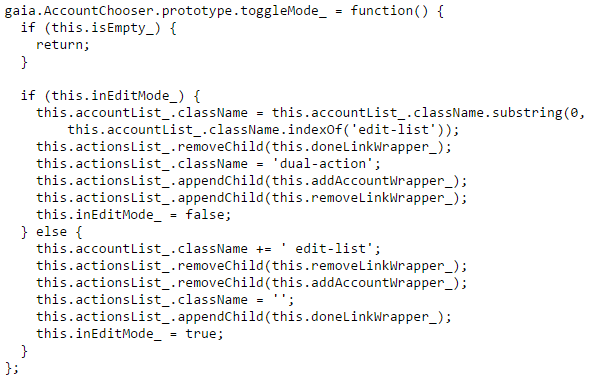
1. Facebook



**N = n1 log2 n1 + n2 log2 n2 => N=30 log2 30+15 log2 15 =4.90 + 3.90 = 8.8**

**V = N n1 log2 (n1 + n2) => 12\*30 log2 (30+15)= 8.4 (45) = 378**

1. Gmail



**N = n1 log2 n1 + n2 log2 n2 => N=32 log2 32+25 log2 25 =5 + 4.64 = 9.64**

**V = N n1 log2 (n1 + n2) => 10\*32 log2 (32+25)= 8.32 (57) = 474.24**

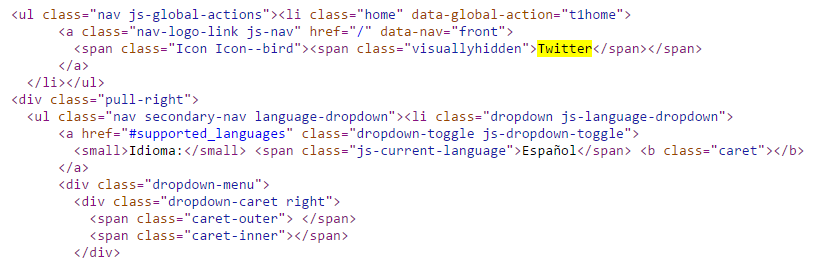
1. WhatsApp



**N = n1 log2 n1 + n2 log2 n2 => N=30 log2 30+15 log2 15 =4.90 + 3.90 = 8.8**

**V = N n1 log2 (n1 + n2) => 12\*30 log2 (30+15)= 8.4 (45) = 378**

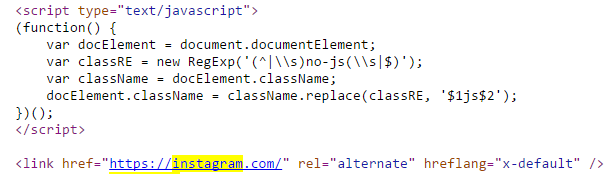
1. Twitter



**N = n1 log2 n1 + n2 log2 n2 => N=30 log2 30+15 log2 15 =4.90 + 3.90 = 8.8**

**V = N n1 log2 (n1 + n2) => 12\*30 log2 (30+15)= 8.4 (45) = 378**

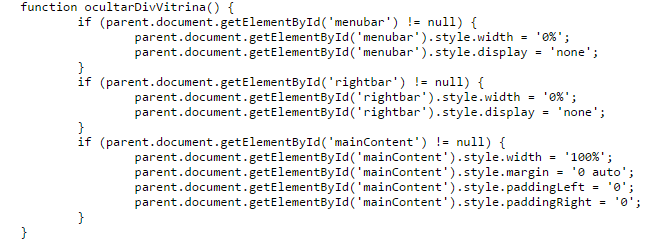
1. Instagram



**N = n1 log2 n1 + n2 log2 n2 => N=12 log2 12+7 log2 7=3.58 + 2.80 = 6.38**

**V = N n1 log2 (n1 + n2) => 5\*12 log2 (5+7)= 5.9 (12) = 70.8**

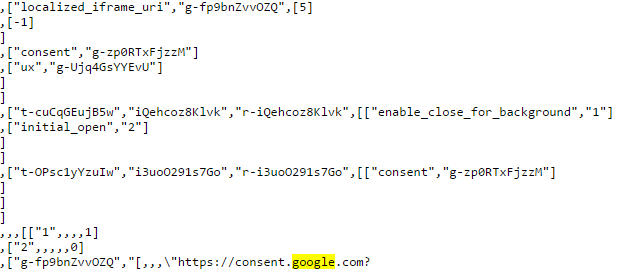
1. Sena Sofía



**N = n1 log2 n1 + n2 log2 n2 => N=25 log2 25+12 log2 12=4.64 + 3.58 = 8.22**

**V = N n1 log2 (n1 + n2) => 7\*25 log2 (25+12)= 7.45 (37) = 275.65**

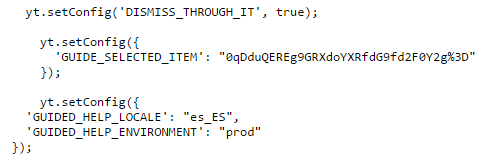
1. Google



**N = n1 log2 n1 + n2 log2 n2 => N=25 log2 25+12 log2 12=4.64 + 3.58 = 8.22**

**V = N n1 log2 (n1 + n2) => 7\*25 log2 (25+12)= 7.45 (37) = 275.65**

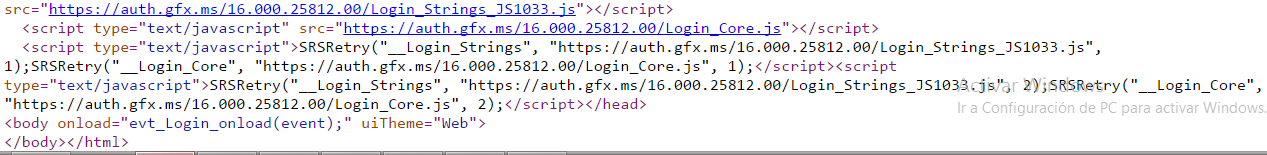
1. YouTube



**N = n1 log2 n1 + n2 log2 n2 => N=12 log2 12+7 log2 7=3.58 + 2.80 = 6.38**

**V = N n1 log2 (n1 + n2) => 5\*12 log2 (5+7)= 5.9 (12) = 70.8**

1. Hotmail



**N = n1 log2 n1 + n2 log2 n2 => N=7 log2 7+5 log2 5=2.80 + 2.32 = 5.12**

**V = N n1 log2 (n1 + n2) => 5\*7 log2 (7+5)= 35 (12) = 420**

# 3.3.3. Realizar las métricas para pruebas a 10 programas

## 3.3.3.1 Definición métrica para pruebas

Las métricas para pruebas se concentran en el proceso de prueba, no en las características técnicas de las pruebas mismas. En general, los responsables de las pruebas deben fiarse en las métricas de análisis, diseño y código para que sirvan de guía en el diseño y ejecución de los casos de prueba.

El esfuerzo de las pruebas también se puede estimar utilizando métricas obtenidas de las medidas de Halstead. 

Usando la definición del volumen de un programa, V, y nivel de programa, NP, el esfuerzo de la ciencia del software puede calcularse como:

**NP = 1/[(n1/2) x (N2/n2)]     (Ec. 1)**

**e = V/NP                              (Ec. 2)**

El porcentaje del esfuerzo global de pruebas a asignar a un módulo k se puede estimar utilizando la siguiente relación:

**Porcentaje de esfuerzo de pruebas (k) = e(k)/åe(i)         (Ec. 3)**

 Donde e(k) se calcula para el módulo k utilizando las ecuaciones (Ec. 1) y (Ec. 2), la suma en el denominador de la ecuación (Ec. 3) es la suma del esfuerzo de la ciencia del software a lo largo de todos los módulos del sistema.

A medida que se van haciendo las pruebas, tres medidas diferentes proporcionan una indicación de la compleción de las pruebas:

|  |  |
| --- | --- |
| **Medida de amplitud de las pruebas.** | Proporciona una indicación de cuantos requisitos se han probado del número total de ellos. Indica la compleción del plan de pruebas. |
| **Profundidad de las pruebas** | Medida del porcentaje de los caminos básicos independientes probados con relación al número total de estos caminos en el programa. Se puede calcular una estimación razonablemente exacta del número de caminos básicos sumando la complejidad ciclomática de todos los módulos del programa. |
| **Perfiles de fallos** | Se emplean para dar prioridad y categorizar los errores. La prioridad indica la severidad del problema. Las categorías de los fallos proporcionan una descripción de un error, de manera que se puedan llevar a cabo análisis estadístico de errores. |

## 3.3.3.2 Cálculo de los 10 programas

**Nota:** para estas pruebas se utilizarán los mismos fragmentos de código utilizados en las métricas del código fuente [(Ctrl+clic para ver)](#_3.3.2.2_Cálculo_de)

1. Skype

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(25/2) x (5/9) = 1/12.5 x 0.55 = 0.08 x 0.55 = 0.044**

**e = V/NP = 174/0.044 = 3954.54**

1. Facebook

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(30/2) x (5/15) = 1/15 x 0.33 = 0.06 x 0.33 = 0.01**

**e = V/NP = 474.24/0.01 = 37800**

1. Gmail

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(32/2) x (5/25) = 1/16 x 0.2 = 0.06 x 0.2 = 0.12**

**e = V/NP = 378/0.12= 47424**

1. WhatsApp

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(30/2) x (5/15) = 1/15 x 0.33 = 0.06 x 0.33 = 0.01**

**e = V/NP = 474.24/0.01 = 37800**

1. Twitter

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(30/2) x (5/15) = 1/15 x 0.33 = 0.06 x 0.33 = 0.01**

**e = V/NP = 474.24/0.01 = 37800**

1. Instagram

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(5/2) x (5/7) = 1/2.5 x 0.71 = 0.4 x 0.71 = 0.28**

**e = V/NP = 70.8/0.28 = 252.85**

1. Sena Sofía

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(25/2) x (5/12) = 1/12.5 x 0.41 = 0.08 x 0.41 = 0.03**

**e = V/NP = 275.65/0.03 = 9188.3**

1. Google

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(25/2) x (5/12) = 1/12.5 x 0.41 = 0.08 x 0.41 = 0.03**

**e = V/NP = 275.65/0.03 = 9188.3**

1. YouTube

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(5/2) x (5/7) = 1/2.5 x 0.71 = 0.4 x 0.71 = 0.28**

**e = V/NP = 70.8/0.28 = 252.85**

1. Hotmail

**NP = 1/[(n1/2) x (N2/n2)] => NP = 1/(7/2) x (5/5) = 1/3.5 x 1 = 0.28 x 1 = 0.28**

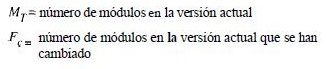
**e = V/NP = 420/0.28 = 1500**

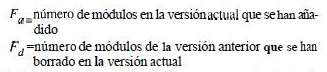
# 3.3.4. Realizar la métrica de mantenimiento para prueba a 10 programas

## 3.3.4.1 Definición métrica de mantenimiento

Todas las métricas del software pueden usarse para el desarrollo de nuevo software y para el mantenimiento del existente. Sin embargo, se han propuesto métricas diseñadas explícitamente para actividades de mantenimiento.

El estándar EEE 982.1-1988 sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto software (basada en los cambios que ocurren con cada versión del producto). Se determina la siguiente información:





El índice de madurez del software se calcula de la siguiente manera:

http://1.bp.blogspot.com/-Ha8zpuqmyZU/UQLA6ZPLX9I/AAAAAAAAALY/BZlnXnuM-w8/s1600/5.jpg

A medida que el IMS se aproxima a 1, o el producto se empieza a estabilizar. EL IMS puede emplearse también como métrica para la planificación de las actividades de mantenimiento del software. El tiempo medio para producir una versión de un producto software puede correlacionarse con el IMS desarrollándose modelos empíricos para el mantenimiento.

## 3.3.4.2 Cálculo de los 10 programas

1. Skype

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 20 – (2+3+1)/20 = 20-6/20 =14/20= 0.7**

1. Facebook

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 28 – (5+2+2)/28 = 28-10/28 =18/28= 0.64**

1. Gmail

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 19 – (2+1+1)/19 = 19-4/19 =15/19= 0.78**

1. WhatsApp

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 19 – (2+1+1)/19 = 19-4/19 =15/19= 0.78**

1. Twitter

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 10 – (5+3+2)/10 = 10-10/10 =0/10= 0**

1. Instagram

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 25 – (4+1+1)/25 = 25-6/25 =19/25= 0.76**

1. Sena Sofía

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 45 – (8+2+2)/45 = 45-12/45 =33/45= 0.73**

1. Google

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 32 – (4+1+1)/32 = 32-6/32 =26/32= 0.81**

1. YouTube

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 24 – (8+4+2)/24 = 24-14/24 =10/24= 0.41**

1. Hotmail

**IMS= {Mt - (Fa + Fc + Fd)} / Mt => 20 – (5+3+1)/20 = 20-9/20 =11/20= 0.55**

# REFERENCIAS

Métricas del código fuente

<https://msdn.microsoft.com/es-es/library/bb385914.aspx>

<http://es.slideshare.net/KevinCastillo15/metricas-de-codigo-fuente-y-metricas-de-prueba>

Métricas de mantenimiento

<http://ing-software3.blogspot.com.co/2013/01/metricas-del-mantenimiento.html>