Sistema de Información Integrado Colciencias – Módulo de Seguimiento

Measurement Plan

Versión 2.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20/Sept/2014 | 1.0 | Plan de medición para el módulo Seguimiento realizado | Nelson Andres Castillo  Gustavo Adolfo López |
| 05/Oct/2014 | 2.0 | Corrección de fórmulas en varias métricas | Nelson Andres Castillo  Gustavo Adolfo López |
| 06/Oct/2014 | 2.1 | Corrección de una métrica | Nelson Andres Castillo Saza |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Overview 4

2. Management Goals and Subgoals 4

3. Metrics 5

3.1 Porcentaje de casos de uso terminados 6

3.2 Porcentaje de casos de uso corregidos 7

3.3 Reducción de tiempo con respecto a la manera antigua 7

3.4 Tiempo promedio en procesamiento de transacción 7

3.5 Consumo promedio de memoria RAM. Consumo de disco duro 8

3.6 Cantidad de usuarios que puede atender concurrentemente 8

3.7 Numero de módulos con los que hay comunicación 9

3.8 Funciones evidentes 9

3.9 Grado de satisfacción con la navegación del sitio 10

3.10 Cantidad de errores cometidos 10

3.11 Grado de satisfacción del usuario con la interfaz grafica 10

3.12 Cantidad de operaciones empezadas y finalizadas a la semana 11

3.13 Porcentaje de disponibilidad diario 11

3.14 Cantidad de casos de uso ligados entre si 12

3.15 Tiempo medio de recuperación 12

3.16 Numero de funciones visibles por usuario 13

3.17 Razón de ataques detectados y denegados 13

3.18 Porcentaje de pruebas de autorización por roles 13

3.19 Control de accesos 14

3.20 Porcentaje de módulos intercambiables 14

3.21 Porcentaje de módulos reusables 15

3.22 Registro de cambios 15

3.23 Porcentaje de bugs detectados y corregidos 15

3.24 Tipos de dispositivos en los que se puede visualizar 16

4. [Primitive Metrics](#_Primitive_Metrics) 16

4.1 [Cantidad de bugs encontrados](#_Cantidad_de_bugs) 17

4.2 [Cantidad de líneas de código](#_Cantidad_de_líneas) 17

4.3 [Elementos de Datos de Entrada](#_Elementos_de_Datos) 18

4.4 [Cantidad de pruebas ejecutadas](#_Cantidad_de_pruebas) 18

4.5 [Utilización de memoria](#_Utilización_de_memoria) 18

4.6 [Tiempo de finalización de consultas](#_Tiempo_de_finalización) 19

5. [Annexes](#_Annexes) 19

# Introduction

El plan de medición define las metas, las métricas asociadas, y las métricas primitivas que deben recopilarse en el proyecto para monitorear su progreso en términos que puedan ser medibles cuantitativamente.

## Purpose

El plan de medición de software especifica que métricas deben ser consideradas y computadas durante el monitoreo del progreso del proyecto, todo esto relacionado con metas claras y especificas a ser alcanzadas dentro de periodo de tiempo.

## Scope

El alcance para este Plan de Medición corresponde al Módulo de Seguimiento del proyecto SII Comunidad Colciencias para evaluar el progreso en el desarrollo de los casos de uso asignados (SEG10 – SEG16) y comprobar que se esté cumpliendo con los lineamientos establecidos en cuanto a estimaciones.

## Overview

El presente Plan de Medición muestra la técnica escogida para el uso de métricas (ISO 25010) y explica cada una de las métricas definidas para medir la calidad de la aplicación.

# Management Goals and Subgoals

Para todo proyecto de software es muy importante tener los lineamientos claros en cuanto a la definición de las metas y objetivos que deben ser alcanzados así como la consecución de los mismos. Esto último necesita de un proceso de validación y verificación para poder tener la certeza de que las metas propuestas están siendo alcanzadas, allí es donde vemos la importancia de las métricas.

Las métricas en la ingeniería de software permiten asegurarse del avance del proyecto en términos que puedan ser cuantificados. Este no es un proceso fácil, pero vale la pena realizarse ya que “lo que no se puede medir, no se puede controlar”; existen algunos enfoques que nos permiten la medición y derivación de métricas como por ejemplo el SQM (Software-Quality Metric).

El Software-Quality Metric usa como modelos de referencia Gilb, McCall, Bohem, FURPS, entre otros. Para este proyecto aplicaremos el modelo ISO/IEC 25010 que define 8 atributos de calidad para evaluar el software.



Con base en los elementos presentados por este modelo, se establecen las siguientes metas para el proyecto:

* [M1] Abarcar la totalidad de las funcionalidades requeridas por el cliente.
* [M2] Soportar la implementación web del módulo con una correcta visualización en los navegadores web de computadores convencionales y sobre diferentes dispositivos móviles como tabletas y celulares.
* [M3] Realizar pruebas de funcionalidad y rendimiento que comprueben la realización de los casos de uso SEG10, SEG13, SEG14, SEG15 y SEG16 de tal manera que se obtengan los comportamientos esperados.
* [M4] Definir un alcance claro en las operaciones y los accesos sobre la información para los usuarios de acuerdo a sus privilegios en el sistema.
* [M5] Auditar todas las operaciones que involucren manipulación de datos para así garantizar un mejor seguimiento a los procesos del sistema.
* [M6] Lograr un grado de satisfacción de usuario por encima del 80% a nivel de usabilidad.
* [M7] Cumplir con la entrega del proyecto dentro de los tiempos estimados inicialmente.
* [M8] Garantizar que el sistema sea fácil de mantener a través del tiempo por parte de los desarrolladores.
* [M9] Mejorar la confiabilidad en el sistema por parte de los usuarios finales.
* [M10] Asegurar la optimización del tiempo empleado en cada operación por parte del usuario final.
* [M11] Garantizar el correcto funcionamiento de la aplicación con los recursos disponibles y definidos inicialmente.
* [M12] Aumentar el grado en que el uso de la aplicación es de fácil aprendizaje y ocurra de manera intuitiva.

# Metrics

Basados en las metas anteriormente descritas y en el modelo ISO/IEC 25010 se ha definido el siguiente listado de métricas que apoya fuertemente el progreso de cada meta y finalmente el grado de consecución de cada una de ellas en las diferentes etapas del proyecto. Permitiendo así realizar un mejor seguimiento y control de la calidad del producto software y tomar las medidas correctivas/preventivas que sean necesarias en el momento que se requieran.

|  |  |  |
| --- | --- | --- |
| Características | Sub-características | Métricas Asociadas |
| Funcionalidad | Completitud Funcional | 1. Porcentaje de casos de uso terminados |
| Corrección Funcional | 1. Porcentaje de casos de uso corregidos |
| Pertinencia Funcional | 1. Reducción de tiempo con respecto a la manera antigua |
| Rendimiento | Comportamiento del Tiempo | 1. Tiempo promedio en procesamiento de transacción |
| Uso de Recursos | 1. Consumo promedio de memoria RAM. |
| Capacidad | 1. Cantidad de usuarios que puede atender concurrentemente |
| Compatibilidad | Co-existencia | No aplica |
| Interoperabilidad | 1. Numero de módulos con los que hay comunicación |
| Usabilidad | Inteligibilidad | 1. Funciones evidentes |
| Aprendizaje | No aplica |
| Operabilidad | 1. Grado de satisfacción con la navegación del sitio |
| Protección a Errores del Usuario | 1. Cantidad de errores cometidos |
| Estética | 1. Grado de satisfacción del usuario con la interfaz grafica |
| Accesibilidad | No Aplica |
| Fiabilidad | Madurez | 1. Cantidad de operaciones empezadas y finalizadas a la semana |
| Disponibilidad | 1. Porcentaje de disponibilidad diario |
| Tolerancia a Fallos | 1. Cantidad de casos de uso ligados entre si |
| Recuperabilidad | 1. Tiempo medio de recuperación |
| Seguridad | Confidencialidad | 1. Numero de funciones visibles por usuario |
| Integridad | 1. Razón de ataques detectados y denegados |
| No Repudio | No aplica |
| Autenticidad | 1. Porcentaje de pruebas de autorización por roles |
| Responsabilidad | 1. Control de accesos |
| Mantenibilidad | Modularidad | 1. Porcentaje de módulos intercambiables |
| Reusabilidad | 1. Porcentaje de módulos reusables |
| Analizabilidad | No aplica |
| Modificabilidad | 1. Registro de cambios |
| Testeabilidad | 1. Porcentaje de bugs detectados y corregidos |
| Portabilidad | Adaptabilidad | 1. Tipos de dispositivos en los que se puede visualizar |
| Instalabilidad | No aplica |
| Reemplazabilidad | No aplica |

## Porcentaje de casos de uso terminados

|  |  |
| --- | --- |
| Name | Porcentaje de casos de uso terminados |
| Definition | Esta métrica define que tan completa esta la implementación funcional estableciendo una relación entre casos de uso finalizados y casos de uso sin terminar. Permite medir la completitud funcional como atributo de Funcionalidad. |
| Goals | [M1] [M7] |
| Analysis Procedure | El valor de esta métrica estará comprendido entre 0% y 100% asumiendo que todos los casos de uso ya están listos para ser implementados en términos de software.  La fórmula de medición está dada por:  Donde:  A = Casos de uso finalizados  B = Casos de uso totales |
| Responsibilities | Desarrolladores |

## Porcentaje de casos de uso corregidos

|  |  |
| --- | --- |
| Name | Porcentaje de casos de uso corregidos |
| Definition | Esta métrica permite apreciar el porcentaje de casos de uso que han sido corregidos en su implementación con respecto a los que se identificaron que no funcionan de la forma esperada. |
| Goals | [M1], [M3], [M7], [M9] |
| Analysis Procedure | El valor de esta métrica está comprendido en un rango entre 0% y 100% y su medición está dada por:  Donde:  A = Casos de uso corregidos  B = Casos de uso que no funcionan de la forma esperada |
| Responsibilities | Desarrolladores |

## Reducción de tiempo con respecto a la manera antigua

|  |  |
| --- | --- |
| Name | Reducción de tiempo con respecto a la manera antigua |
| Definition | Esta métrica se encarga de medir la pertinencia funciona midiendo la variación de tiempo existente entre el tiempo empleado para realizar un proceso de la antigua forma y el tiempo empleado para realizar el mismo proceso a través de la plataforma online. |
| Goals | [M1], [M10] |
| Analysis Procedure | El valor de esta métrica está dada en unidades de tiempo (horas, minutos, segundos) y su cálculo se obtiene de la siguiente forma:  Donde:  A = Tiempo empleado en la forma actual  B = Tiempo empleado a través de la plataforma  Entre más grande sea el valor de X significa que mayor ha sido la reducción de tiempo.  Si X < 0 entonces significa que la plataforma es poco optima y ocasiona retrasos. |
| Responsibilities | Desarrolladores |

## Tiempo promedio en procesamiento de transacción

|  |  |
| --- | --- |
| Name | Tiempo promedio en procesamiento de transacción |
| Definition | Esta métrica sirve para calcular el tiempo medio que tarda una transacción en ser realizada lo cual permite observar el rendimiento de la aplicación con respecto al tiempo empleado en las operaciones. |
| Goals | [M3], [M10] |
| Analysis Procedure | Esta métrica tendrá un valor en unidades de tiempo (segundos y milisegundos) y se puede calcular de la siguiente forma:  Donde:  Entre menor sea el valor de X mejor es el resultado para la métrica.  Se asumen que los datos necesarios para la transacción pertenecientes a otros casos de uso ya se encuentran listos desde antes del momento de la transacción. |
| Responsibilities | Desarrolladores |

## Consumo promedio de memoria RAM. Consumo de disco duro

|  |  |
| --- | --- |
| Name | Consumo promedio de memoria RAM |
| Definition | Esta métrica permite observar la cantidad media de memoria RAM requerida por la aplicación en modo de operación normal para así observar el comportamiento en el aprovechamiento de los recursos. |
| Goals | [M8], [M11] |
| Analysis Procedure | El valor para esta métrica estará dado en Megabytes (MB) que indican el consumo en términos de uso de la memoria RAM.  Para conocer este valor se deberá hacer uso del monitor de actividad del sistema operativo del servidor (Ubuntu server) y llevar un registro de sus resultados.  Entre más cerca se encuentre el nivel de de la significa que hay más riesgo de sobre cargar la memoria principal. Entre más lejos estén estos valores, habrá más estabilidad y mejor rendimiento tendrá la aplicación.  Se asume que el entorno de producción en el que se encuentra la aplicación cuenta con los requerimientos mínimos impuestos por las tecnologías sobre las que se está trabajando. |
| Responsibilities | Desarrolladores |

## Cantidad de usuarios que puede atender concurrentemente

|  |  |
| --- | --- |
| Name | Cantidad de usuarios que puede atender concurrentemente |
| Definition | Con esta métrica es posible determinar la cantidad de usuarios que pueden hacer uso de la aplicación de manera concurrente sin comprometer el rendimiento de la misma. |
| Goals | [M3], [M8], [M10], [M11] |
| Analysis Procedure | El valor de esta métrica está comprendido por un número entero que indica la cantidad de usuarios concurrentes a ser atendidos sin que la aplicación colapse.  X = Tope de usuarios/Cantidad esperada  Para calcular este valor hay 2 posibles formas, la primera es mediante el uso de pruebas de estrés hasta llegar al punto de quiebre para la aplicación y la segunda es tomando el tiempo promedio que tarda una transacción y mirando cuantas transacciones pueden ser resueltas en 1 segundo para luego multiplicar este valor por la cantidad límite de instancias que se definieron para el servidor. |
| Responsibilities | Desarrolladores |

## Numero de módulos con los que hay comunicación

|  |  |
| --- | --- |
| Name | Numero de módulos con los que hay comunicación |
| Definition | Esta métrica permite medir la cantidad de módulos con los cuales la aplicación se relaciona y de las cuales toman y envía datos. Esto permite medir la compatibilidad del sistema con otros sistemas. |
| Goals | [M8], [M5] |
| Analysis Procedure | La medida de esta métrica está dada en un valor numérico que determinar la cantidad de módulos con los que el sistema se debe comunicar.    Entre mayor sea el valor de X mayor es el grado interoperabilidad del sistema.  Se asume que existen interfaces definidas para el intercambio de información entre los módulos. |
| Responsibilities | Desarrolladores |

## Funciones evidentes

|  |  |
| --- | --- |
| Name | Funciones evidentes |
| Definition | Esta métrica permite saber cuáles son las funcionalidades que un tipo de usuario percibe dentro del sistema con fin de comprobar el grado de entendimiento que ofrece la aplicación. |
| Goals | [M6], [M10], [M12] |
| Analysis Procedure | El valor de esta métrica está dado por un número entero que indica cuantas funcionalidades fue capaz de hallar el usuario final.  X = Cantidad de funciones evidentes/Funcionalidades Totales  Entre más próximo se encuentre el valor de entonces mayor inteligibilidad presenta el software. |
| Responsibilities | Desarrolladores |

## Grado de satisfacción con la navegación del sitio

|  |  |
| --- | --- |
| Name | Grado de satisfacción con la navegación del sitio |
| Definition | Métrica que permite medir la operabilidad presente en la aplicación para los usuarios mediante el nivel de satisfacción que indican con respecto a la navegación del sitio. |
| Goals | [M6], [M9], [M12] |
| Analysis Procedure | Para el valor de esta métrica el usuario final después de haber interactuado con la aplicación deberá de seleccionar un valor de satisfacción con la navegación del sitio. Para esto se tiene definida una escala que va de 1 a 5.  Se asume que las vistas y las plantillas han sido renderizadas y que se tienen menús de navegación. |
| Responsibilities | Desarrolladores |

## Cantidad de errores cometidos

|  |  |
| --- | --- |
| Name | Cantidad de errores cometidos |
| Definition | Métrica que permite tomar la cantidad de errores a nivel de navegación y usabilidad cometidos por el usuario final durante el uso de la aplicación. |
| Goals | [M6], [M12] |
| Analysis Procedure | El valor para esta medida está dada por un número entero que representa la cantidad de errores cometidos por el usuario.  Entre menor sea el valor de X mayor el nivel en que la aplicación protege a los usuarios de cometer errores. |
| Responsibilities | Desarrolladores |

## Grado de satisfacción del usuario con la interfaz grafica

|  |  |
| --- | --- |
| Name | Grado de satisfacción del usuario con la interfaz grafica |
| Definition | Esta métrica sirve para evaluar el nivel de satisfacción de los usuarios con respecto a la estética de la interfaz gráfica de usuario (GUI). |
| Goals | [M2], [M6] |
| Analysis Procedure | Para el valor de esta métrica el usuario final después de haber interactuado con la aplicación deberá de seleccionar un valor de satisfacción para indicar su agrado hacia la estética de la interfaz gráfica. Para esto se tiene definida una escala que va de 1 a 5. |
| Responsibilities | Desarrolladores |

## Cantidad de operaciones empezadas y finalizadas a la semana

|  |  |
| --- | --- |
| Name | Cantidad de operaciones empezadas y finalizadas a la semana |
| Definition | Métrica que define la cantidad de transacciones empezadas y terminadas satisfactoriamente en promedio por semana de manera que permite evidenciar la madurez de la aplicación y aumente la fiabilidad en ella. |
| Goals | [M9] |
| Analysis Procedure | Para el valor de esta métrica se tiene como medida un numero entero que representa la cantidad de transacciones completas en una semana.  Entre mayor sea el número de operación mayor grado de madurez se refleja en el sistema. |
| Responsibilities | Desarrolladores |

## Porcentaje de disponibilidad diario

|  |  |
| --- | --- |
| Name | Porcentaje de disponibilidad diario |
| Definition | Esta métrica representa la cantidad de caídas que sufre el sistema en un dia y nos permite medir la disponibilidad tenida por el mismo en ese tiempo determinado. |
| Goals | [M9], [M11] |
| Analysis Procedure | El valor de esta métrica está determinado por la siguiente formula:  Donde:  A = Cantidad de horas en las que se toma la medición (24 horas)  Entre más cercano este el valor de X al 100% significa que la disponibilidad del software es alta.  Se asume que el sistema ha sido lanzado a entorno de producción 24 horas/7 días |
| Responsibilities | Desarrolladores |

## Cantidad de casos de uso ligados entre si

|  |  |
| --- | --- |
| Name | Cantidad de casos de uso ligados entre si |
| Definition | Métrica que permite medir la cantidad de casos de uso ligados entre sí para así poder tener una referencia de los puntos críticos de la aplicación en los cuales un problema en un caso de uso afectaría a los demás así como también los casos de uso que pueden fallar sin comprometer a los otros. |
| Goals | [M8], [M9] |
| Analysis Procedure | El valor de esta métrica será un número entero que represente la cantidad de casos de usos ligados entre sí.  X = 1 – (Cantidad de casos de uso ligados/Cantidad total de casos de uso)  Se asume que los casos de uso han sido implementados ya en el software. |
| Responsibilities | Desarrolladores |

## Tiempo medio de recuperación

|  |  |
| --- | --- |
| Name | Tiempo medio de recuperación |
| Definition | Esta métrica se usa para medir el tiempo de recuperación empleado por el sistema al momento de recuperar los datos cuando han sido afectados para poder reestablecer el servicio en caso de interrupciones o fallos. |
| Goals | [M9], [M11] |
| Analysis Procedure | El valor de esta métrica está dada en unidades de tiempo (horas, minutos, segundos y milisegundos) y su cálculo se obtiene de la siguiente forma:  Donde:  A = Tiempo en que el sistema tuvo una pérdida de datos o interrupción  B = Tiempo en el que el servicio fue reestablecido  Entre menor sea el valor de X menor es el tiempo empleado por el sistema para recuperarse. |
| Responsibilities | Desarrrolladores |

## Numero de funciones visibles por usuario

|  |  |
| --- | --- |
| Name | Numero de funciones visibles por usuario |
| Definition | Mediante el uso de esta métrica se verifica a que tanto contenido tiene acceso determinado tipo de usuario para garantizar que únicamente pueda ver la información que le corresponde. |
| Goals | [M4], [M9] |
| Analysis Procedure | Para la medida de esta métrica se tiene que el valor estará dado en un número entero que va a representar la cantidad de funcionalidades y datos que el usuario puede ver y que están más allá de las que le corresponden.  X = Cantidad de funciones adicionales/Cantidad de funciones disponibles  La medida deseada es X=0 ya que esto significa que únicamente accede a la información que le corresponde. Por cada unidad en que X>0 se tendrá un acceso extra a información que el usuario no debe poder acceder.  Si X<0 esto significa que hay funcionalidades que el usuario debería poder acceder pero por algún motivo no puede. |
| Responsibilities | Desarrolladores |

## Razón de ataques detectados y denegados

|  |  |
| --- | --- |
| Name | Razón de ataques detectados y denegados |
| Definition | Métrica que permite evaluar aspectos de seguridad relacionados con ataques a la aplicación o directamente a la información. |
| Goals | [M4], [M5] |
| Analysis Procedure | El valor de esta métrica corresponde a un valor entre 0% y 100% dado entre la razón de los ataques detectados y los ataques detenidos.  Donde:  Entre más alto sea el valor de X más estará protegida la aplicación contra los tipos de ataques detectados. |
| Responsibilities | Desarrolladores |

## Porcentaje de pruebas de autorización por roles

|  |  |
| --- | --- |
| Name | Porcentaje de pruebas de autorización por roles |
| Definition | Esta métrica permite medir las pruebas realizadas a la autorización a las funciones de cada uno de los roles. |
| Goals | [M4], [M5] |
| Analysis Procedure | El valor de esta métrica se encuentra entre 0% y 100% usando la cantidad de pruebas de autorización por roles a las funcionalidades a las que están autorizados y se calcula de la siguiente forma:  Donde: |
| Responsibilities | Desarrolladores |

## Control de accesos

|  |  |
| --- | --- |
| Name | Control de accesos |
| Definition | Con esta métrica es posible medir la cantidad de accesos que ha tenido un tipo de usuario a un caso de uso en particular. |
| Goals | [M4], [M5], [M9] |
| Analysis Procedure | El valor de estas métricas es un número entero que representa la cantidad de veces que ha accedido un usuario a realizar operaciones sobre un proceso en específico.  Se asume que en la base de datos se almacena un historial de accesos sobre los elementos y los usuarios que los realizan. |
| Responsibilities | Desarrolladores |

## Porcentaje de módulos intercambiables

|  |  |
| --- | --- |
| Name | Porcentaje de módulos intercambiables |
| Definition | Métrica que permite saber que tan modular es la aplicación conociendo la cantidad de módulos que forman la aplicación y que pueden ser fácilmente reemplazados por otros sin afectar mucho los demás. |
| Goals | [M8] |
| Analysis Procedure | El valor de esta métrica esta dado entre 0% y 100% y puede ser calculado con la siguiente formula:  Donde:  Un alto valor de X permite expresar que la aplicación puede ser mantenida fácilmente a través del tiempo reemplazando algunos de sus módulos sin afectar grandemente el funcionamiento del resto del sistema. |
| Responsibilities | Desarrolladores |

## Porcentaje de módulos reusables

|  |  |
| --- | --- |
| Name | Porcentaje de módulos reusables |
| Definition | Esta métrica permite determinar el porcentaje de módulos que integran el producto de software que pueden ser de alguna utilidad en otros proyectos. |
| Goals | [M8] |
| Analysis Procedure | El valor de esta métrica está dada en un rango entre 0% y 100% y su medición está dada por:  Donde:  A = Cantidad de módulos reusables en otros proyectos.  B = Cantidad de módulos que componen el software. |
| Responsibilities | Desarrolladores |

## Registro de cambios

|  |  |
| --- | --- |
| Name | Registro de cambios |
| Definition | Métrica que permite medir los registros de cambios a los módulos con comentarios en el código. |
| Goals | [M8] |
| Analysis Procedure | El valor de esta medida registra la proporción de información sobre cambios a los módulos mediante la fórmula:  Donde:  A = Numero de cambios a funciones o módulos que tienen comentarios  B = Total de funciones o módulos modificados |
| Responsibilities | Desarrolladores |

## Porcentaje de bugs detectados y corregidos

|  |  |
| --- | --- |
| Name | Porcentaje de bugs detectados y corregidos |
| Definition | Esta métrica permite determinar el porcentaje de bugs (errores, defectos o fallas) corregidos con respecto a los que han sido detectados en la implementación. Esto con el fin de evidenciar la efectividad del proceso de pruebas para la aplicación. |
| Goals | [M3][M9] |
| Analysis Procedure | El valor de esta métrica está dada en un rango entre 0% y 100% y su medición está dada por:  Donde:  A = Cantidad de bugs corregidos  B = Cantidad de bugs que se han encontrado de acuerdo a las pruebas realizadas. |
| Responsibilities | Desarrolladores |

## Tipos de dispositivos en los que se puede visualizar

|  |  |
| --- | --- |
| Name | Tipos de dispositivos en los que se puede visualizar |
| Definition | Esta métrica permite determinar la cantidad de tipos de dispositivos móviles en los que podrá visualizarse el software, contando para ello las distintas resoluciones en las que la adaptación se hará de manera eficiente. |
| Goals | [M2] |
| Analysis Procedure | El valor de esta métrica está dada en un rango entre 0% y 100% y su medición está dada por:  Donde:  A = Cantidad de resoluciones en las que podrá ser visualizada la aplicación con responsive.  B = Cantidad de resoluciones convencionales para pc y dispositivos móviles. |
| Responsibilities | Desarrolladores |

# 

# Primitive Metrics

Las métricas primitivas son aquellas que se pueden calcular en forma directa definiendo un criterio de conteo para medirlas. Estas métricas establecen el primer nivel de los datos que necesitan ser recolectados para implantar otras métricas

## Cantidad de bugs encontrados

|  |  |
| --- | --- |
| Name | Cantidad de bugs encontrados |
| Definition | Los errores en la implementación y codificación pueden ser contados de acuerdo a los que se encontraron y corrigieron, respectivamente, siendo buena idea tener un contador independiente por cada categoría. |
| Collection procedure | Se debe registrar la fase de desarrollo en la cual los bugs fueron encontrados, al igual que el tipo de error (Entrada/Salida, Gestión de Memoria, Recarga de páginas, etc) con el objeto de garantizar la calidad y retroalimentación a los programadores.  Lo que se medirá es el número de absoluto de errores que quedan por corregir, es decir, la diferencia entre los bugs hallados y el número de correcciones resultantes de una revisión.  Y = X1 – X2  Donde:  X1 = errores encontrados  X2 = errores corregidos |
| Responsibilities | Desarrolladores |

## 

## Cantidad de líneas de código

|  |  |
| --- | --- |
| Name | Cantidad de líneas de código |
| Definition | Se cuenta la cantidad de líneas de código (LOC) empleadas una vez finalice la aplicación para el módulo de Seguimiento. |
| Collection procedure | Es posible distinguir dos mediciones para esta métrica:   1. Líneas de código efectivas   ELOC = NCLOC – CLOC  Donde:  ELOC = líneas de código efectivas NCLOC = líneas de código no comentadas CLOC = líneas de código comentadas   1. Total de líneas de código   LOC = NCLOC + CLOC  Donde:  LOC = total de líneas de código  NCLOC = líneas de código no comentadas CLOC = líneas de código comentadas |
| Responsibilities | Desarrolladores |

## Elementos de Datos de Entrada

|  |  |
| --- | --- |
| Name | Elementos de Datos de Entrada |
| Definition | Esta métrica mide la cantidad de elementos de datos que son introducidos en el sistema para llevar a cabo un registro, actualización o cualquier otra interacción que requiera entrada de datos por parte del usuario. |
| Collection procedure | La medición de la métrica consiste en hacer una sumatoria de la cantidad de formularios y otros elementos de entrada por cada caso de uso definido.  X = ∑Input(i)  Donde:  Input(i) = Cantidad de datos de entrada necesarios para el caso de uso (i) |
| Responsibilities | Desarrolladores |

## Cantidad de pruebas ejecutadas

|  |  |
| --- | --- |
| Name | Cantidad de pruebas ejecutadas |
| Definition | Esta métrica ayuda a proporcionar información objetiva e independiente sobre la calidad el producto a las partes interesadas, siendo una actividad más en el proceso de gestión de calidad. |
| Collection procedure | Como métrica de producto se encarga de contar la totalidad de pruebas realizadas sobre las diferentes funcionalidades establecidas en las secuencias de evento de los casos de uso.  X = Cantidad de pruebas ejecutadas. |
| Responsibilities | Desarrolladores |

## Utilización de memoria

|  |  |
| --- | --- |
| Name | Utilización de memoria |
| Definition | Esta métrica se encarga de medir la cantidad de memoria RAM que está siendo utilizada al ejecutar los procesos requeridos mientras esté corriendo la aplicación. |
| Collection procedure | X = Cantidad de memoria RAM utilizada al ejecutar la aplicación. |
| Responsibilities | Desarrolladores |

## Tiempo de finalización de consultas

|  |  |
| --- | --- |
| Name | Tiempo de finalización de consultas |
| Definition | Esta métrica proporciona la medición del tiempo que tarda en llevarse a cabo cada consulta sobre la base de datos. |
| Collection procedure | El cálculo de tiempo de la consulta está ligado al motor de base de datos utilizado. Para postgresql, el cual se usará para gestionar la persistencia de la aplicación, se utiliza el siguiente comando antes de la consulta:  EXPLAIN ANALYZE |
| Responsibilities | Desarrolladores |

# Annexes

Instrumento en Excel para aplicar las métricas y evaluar cuantitativamente el avance del proyecto final en función de las métricas definidas.