Contenido

[**PAPS** 2](#_Toc469314265)

[CAPITULO 1 Métricas en el Software. 3](#_Toc469314266)

[1.1. Datos históricos usados en las Métricas. 3](#_Toc469314268)

[1.1.1. Personas Perdidas\_1.0 3](#_Toc469314269)

[1.1.2. nFace 3](#_Toc469314270)

[1.1.3. Buscador de Personas 2.0 4](#_Toc469314271)

[1.2. Propósito del plan del proyecto 4](#_Toc469314272)

[1.3. Métricas Orientadas al Tamaño. 5](#_Toc469314273)

[1.4. Métricas Orientadas a la Función. 6](#_Toc469314274)

[CAPITULO 2 Estimación del Software. 12](#_Toc469314275)

[2.1. Características del proyecto. 12](#_Toc469314277)

[2.1.1. Tamaño. 12](#_Toc469314278)

[2.1.2. Complejidad. 12](#_Toc469314279)

[2.1.3. Estructuración del cliente. 13](#_Toc469314280)

[2.2. Ámbito del Proyecto. 13](#_Toc469314281)

[2.2.1. Objetivos. 13](#_Toc469314282)

[2.2.2. Requerimientos Principales. 14](#_Toc469314283)

[2.2.3. Rendimiento. 14](#_Toc469314284)

[2.2.4. Fiabilidad. 15](#_Toc469314285)

[2.2.5. Integridad y seguridad. 15](#_Toc469314286)

[2.2.6. Eficiencia. 15](#_Toc469314287)

[2.2.7. Usabilidad. 15](#_Toc469314288)

[2.2.8. Mantenibilidad. 15](#_Toc469314289)

[2.2.9. Portabilidad. 15](#_Toc469314290)

[2.2.10. Restricciones. 15](#_Toc469314291)

[2.3. Técnicas de Estimación. 16](#_Toc469314292)

[2.3.1. Estimación Orientada al tamaño. 16](#_Toc469314293)

[2.3.2. COCOMO II (Modelo Constructivo de Costo). 16](#_Toc469314294)

[2.3.3. Ecuación del Software. 17](#_Toc469314295)

[CAPITULO 3 Planificación del tiempo. 18](#_Toc469314296)

[3.1. Diagrama de Gantt. 21](#_Toc469314298)

[3.2. Diagrama de Pert. 22](#_Toc469314299)

[CAPITULO 4 Recursos de Proyecto. 23](#_Toc469314300)

[4.1. Tabla de recursos. 23](#_Toc469314302)

[CAPITULO 5 Análisis de Riesgo. 24](#_Toc469314303)

[5.1. Tabla de Riesgo y plan de aversión para cada riesgo. 24](#_Toc469314305)

[CAPITULO 6 Organización Interna. 27](#_Toc469314306)

[6.1. Estructura del equipo. 27](#_Toc469314308)

[CAPITULO 7 Mecanismos de Seguimiento y Control. 28](#_Toc469314309)

[7.1. Descripción. 28](#_Toc469314311)

[7.2. Seguimiento. 28](#_Toc469314312)

[7.3. Control. 28](#_Toc469314313)

[7.4. Reporte de Tareas. 28](#_Toc469314314)

[REFERENCIAS 33](#_Toc469314315)

[ANEXOS 33](#_Toc469314316)

# 

# **PAPS**

# CAPITULO 1 Métricas en el Software.



## Datos históricos usados en las Métricas.

Para realizar la métrica del proyecto, realizamos la aplicación de Métricas orientadas al tamaño y a la función a tres software similares al proyecto que realizaremos, para así poder realizar estadísticas que nos servirán como punto de referencia para tomar decisiones a futuro.

Los tres proyectos son:

### Personas Perdidas\_1.0

**Descripción:**

Brinda información de personas desaparecidas y te brinda consejos por si un familiar tuyo se extravía.

**Características:**

Todos los días se pierden miles de personas alrededor del mundo, y también muy cerca tuyo, ayúdanos a encontrarlas.

Mediante esta plataforma, podes aportar datos acerca de las personas perdidas registradas, y también podes cargar familiares o amigos tuyos que se hayan extraviado.

Esta aplicación te brinda consejos que te son de suma utilidad por si un familiar tuyo se extravía.

Mientas más gente tenga esta aplicación, mucho más fácil y rápido será encontrar a las personas que se pierden en la Argentina.

### nFace

**Descripción:**.

Aplicación detecta automáticamente puntos descriptores de la forma de tu rostro.

**Características:**

Cuán atractiv@ realmente eres? Nuestra Aplicación detecta automáticamente puntos descriptores de la forma de tu rostro y aplica un algoritmo para evaluar cuan atractivo se percibe.

Palabras claves: escaneador de rostros, evaluador de belleza, reconocimiento de caras, índice de belleza

Nota: Para una evaluacion precisa, es necesario que el algoritmo detecte adecuadamente las caracteristicas de su cara, ello se logra con fotos con altos contrastes entre el fondo y su cara, ademas de un area reducida donde buscar.

Aunque la app funciona muy bien, los resultados son solo con fines de entretenimiento.

### Buscador de Personas 2.0

**Descripción:**

Es un buscador de personas gratis

**Características:**

Con solo introducir el nombre y apellidos de una persona, este nos devolverá todos sus datos públicos en internet, como por ejemplo: cuenta de Facebook, cuenta de Twitter, datos de linkedin, menciones en el BOE, direcciones y números de teléfono en guias como qdq, fotos, videos y muchos más datos.

## Propósito del plan del proyecto

Para realizar la métrica del proyecto “DESAROLLO DE SOFTWARE QUE PERMITA PUBLICAR Y BUSCAR PERSONAS DESAPARECIDAS UTILIZANDO LA TECNOLOGÍA DE RECONOCIMIENTO FACIAL”, realizamos la aplicación de Métricas orientadas al tamaño y a la función a tres software similares al proyecto que realizaremos, para así poder realizar estadísticas que nos servirán como punto de referencia para tomar decisiones a futuro.

Los tres proyectos son:

P1: Personas Perdidas\_1.0

P2: nFace

P3: Buscador de Personas 2.0

## Métricas Orientadas al Tamaño.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto | KLDC | Costo ($us)  (000) | Tiempo (mes) | Esfuerzo  (personas-mes) | Gente | Pag. Doc. | Errores | Defectos |
| Personas Perdidas\_1.0 | 7.5 | 12 | 10.4 | 4 | 3 | 100 | 18 | 4 |
| nFace | 18.7 | 55 | 24.2 | 11 | 5 | 130 | 40 | 7 |
| Buscador de Personas 2.0 | 15.6 | 36 | 19.6 | 9 | 4 | 116 | 26 | 5 |

Tabla 1. Medición de los datos históricos

Calidad = **** Productividad = 

Costo= **** Errores= (Calidad \* KLDC)-D

**Personas Perdidas\_1.0:**

**Calidad =** (18 + 4) / 7.5 **= 2.93**

**Productividad = ((**7.5) / 12) \* 1000) **= 625**

**nFace:**

**Calidad =** (40 + 7) /18.7 **= 2.5**

**Productividad = ((**18.7) / 55**) \*1000) = 340**

**Buscador de Personas 2.0:**

**Calidad =** (26 + 5) / 15.6  **= 1.9**

**Productividad = ((**15.6) / 36) **\*1000) = 433**

|  |  |  |
| --- | --- | --- |
| Proyecto | Calidad | Productividad |
| Personas Perdidas\_1.0 | 2.9 = 3 | 625 |
| nFace | 2.5 = 2 | 340 |
| Buscador de Personas 2.0 | 1.9 = 2 | 433 |

Tabla 2. Análisis de la calidad y productividad de los datos históricos

## Métricas Orientadas a la Función.

De los datos de métricas orientadas a la Función, se obtiene los valores esperados de la cuenta, utilizando las siguientes preguntas para los diferentes parámetros de medición de cada proyecto.

* **Personas Perdidas\_1.0:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parámetros de medición** | **Factores de Peso** | | | | **Total** |
| **Cuentas** | **Simple** | **Medio** | **Compleja** |
| # De Entradas de usuario | 1 | 3 | 4 | 6 | 3 |
| # de salidas de usuario | 2 | 4 | 5 | 7 | 14 |
| # de peticiones | 3 | 3 | 4 | 6 | 9 |
| # de archivos | 17 | 7 | 10 | 15 | 255 |
| # de interfaces externas | 2 | 5 | 7 | 10 | 14 |
| **295** |

Medición del Proyecto 1 según el factor de peso

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **No Influyente** | **Incidental** | **Moderada** | **Medio** | **Significativo** | **Esencial** |
| **Factor** | **0** | **1** | **2** | **3** | **4** | **5** |
| Copia de seguridad y recuperación | X |  |  |  |  |  |
| Comunicaciones de datos |  |  | X |  |  |  |
| Proceso distribuido | X |  |  |  |  |  |
| Rendimiento critico | X |  |  |  |  |  |
| Entorno operativo existente |  | X |  |  |  |  |
| Entrada de datos en línea (on line) | X |  |  |  |  |  |
| Transacciones de entrada en múltiples pantallas | X |  |  |  |  |  |
| Archivos maestros actualizados en línea (on line) | X |  |  |  |  |  |
| Complejidad de valores del dominio de información | X |  |  |  |  |  |
| Complejidad del procesamiento interno |  |  |  |  | X |  |
| Código diseñado para la reutilización |  |  | X |  |  |  |
| Conversión / instalación en diseño para la reutilización | X |  |  |  |  |  |
| Instalaciones múltiples | X |  |  |  |  |  |
| Aplicación diseñada para el cambio |  |  |  |  |  | x |
| **Valor total** | **14** | | | | |

Medición del Proyecto 1 según las métricas orientadas a la función

Entonces calculando el PF tenemos:

**PF = CTAOT \* [0.65+0.01 \*]**

**PF = 295 \* [0.65+0.01\*14]**

**PF = 233 (puntos de función)**

* **nFace**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parámetros de medición** | **Factores de Peso** | | | | **Total** |
| **Cuentas** | **Simple** | **Medio** | **Compleja** |
| # De Entradas de usuario | 1 | 3 | 4 | 6 | 3 |
| # de salidas de usuario | 3 | 4 | 5 | 7 | 21 |
| # de peticiones | 3 | 3 | 4 | 6 | 9 |
| # de archivos | 20 | 7 | 10 | 15 | 300 |
| # de interfaces externas | 3 | 5 | 7 | 10 | 21 |
| **354** |

Medición del Proyecto 2 según el factor de peso

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **No Influyente** | **Incidental** | **Moderada** | **Medio** | **Significativo** | **Esencial** |
| **Factor** | **0** | **1** | **2** | **3** | **4** | **5** |
| Copia de seguridad y recuperación | X |  |  |  |  |  |
| Comunicaciones de datos |  |  | x |  |  |  |
| Proceso distribuido | X |  |  |  |  |  |
| Rendimiento critico | X |  |  |  |  |  |
| Entorno operativo existente |  | X |  |  |  |  |
| Entrada de datos en línea (on line) | X |  |  |  |  |  |
| Transacciones de entrada en múltiples pantallas | X |  |  |  |  |  |
| Archivos maestros actualizados en línea (on line) | X |  |  |  |  |  |
| Complejidad de valores del dominio de información | X |  |  |  |  |  |
| Complejidad del procesamiento interno |  |  |  |  | X |  |
| Código diseñado para la reutilización |  |  | x |  |  |  |
| Conversión / instalación en diseño para la reutilización | X |  |  |  |  |  |
| Instalaciones múltiples | X |  |  |  |  |  |
| Aplicación diseñada para el cambio |  |  |  |  |  | x |
| **Valor total** | **14** | | | | |

Medición del Proyecto 2 según las métricas orientadas a la función

Entonces calculando el PF tenemos:

**PF = CTATOTAL \* [0.65+0.01 \*]**

**PF = 354 \* [0.65+0.01\*14]**

**PF = 280** (puntos de función)

* **Buscador de Personas 2.0**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parámetros de medición** | **Factores de Peso** | | | | **Total** |
| **Cuentas** | **Simple** | **Medio** | **Compleja** |
| # De Entradas de usuario | 1 | 3 | 4 | 6 | 3 |
| # de salidas de usuario | 2 | 4 | 5 | 7 | 21 |
| # de peticiones | 3 | 3 | 4 | 6 | 9 |
| # de archivos | 19 | 7 | 10 | 15 | 285 |
| # de interfaces externas | 2 | 5 | 7 | 10 | 14 |
| **332** |

Medición del Proyecto 2 según el factor de peso

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **No Influyente** | **Incidental** | **Moderada** | **Medio** | **Significativo** | **Esencial** |
| **Factor** | **0** | **1** | **2** | **3** | **4** | **5** |
| Copia de seguridad y recuperación | X |  |  |  |  |  |
| Comunicaciones de datos |  |  | x |  |  |  |
| Proceso distribuido | X |  |  |  |  |  |
| Rendimiento critico | X |  |  |  |  |  |
| Entorno operativo existente |  | X |  |  |  |  |
| Entrada de datos en línea (on line) | X |  |  |  |  |  |
| Transacciones de entrada en múltiples pantallas | X |  |  |  |  |  |
| Archivos maestros actualizados en línea (on line) | X |  |  |  |  |  |
| Complejidad de valores del dominio de información | X |  |  |  |  |  |
| Complejidad del procesamiento interno |  |  |  |  | X |  |
| Código diseñado para la reutilización |  |  | x |  |  |  |
| Conversión / instalación en diseño para la reutilización | X |  |  |  |  |  |
| Instalaciones múltiples | X |  |  |  |  |  |
| Aplicación diseñada para el cambio |  |  |  |  |  | x |
| **Valor total** | **14** | | | | |

Medición del Proyecto 3 según las métricas orientadas a la función

Entonces calculando el PF tenemos:

**PF = CTAOT \* [0.65+0.1 \*]**

**PF = 332 \* [0.65+0.1\*14]**

**PF = 262** (puntos de función)

# CAPITULO 2 Estimación del Software.



## Características del proyecto.

### Tamaño.

El tamaño del software es una medida de la información procesada y proporcionada por el sistema, que se puede expresar en: puntos de función (PF), líneas de código (LDC), número y tipos de requerimientos. Podemos considerar como un proyecto mediano debido que la cantidad de funciones a realizar no es masivo.

### Complejidad.

La complejidad se puede definir en relación con las interacciones entre componentes y subsistemas del sistema, y por otra, con la variedad de cada uno de los subsistemas. Entendemos por variedad, el número de estados posibles que puede alcanzar un sistema o un componente.

El factor técnico de complejidad toma en cuenta la medida de varias técnicas y otros rendimientos implicados en el desarrollo y en el implemento de la información procesada requerida.

Los factores técnicos de complejidad se determinan, estimando el grado de influencia de algunos componentes “características generales de aplicación”. El grado de influencia en la escala recorre de cero (no presente o no influenciada) hasta 5 (influencia fuerte) según la métrica orientada a la función. Para determinar la complejidad del proyecto se debe tomar en cuenta lo siguiente:

La complejidad es relativa a la experiencia en proyectos anteriores.

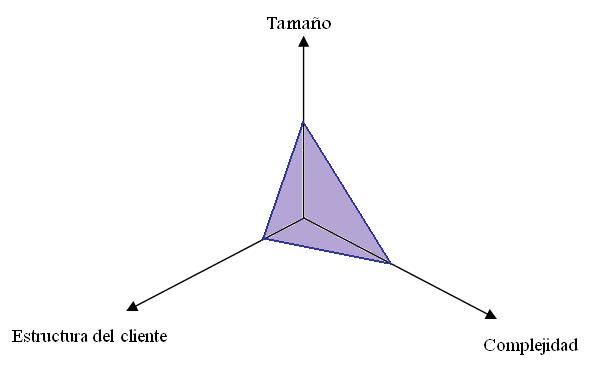
Existen medidas sobre la complejidad de proyectos basadas en el diseño y código (métricas).En la fase de estimación no son aplicables porque no hay ni diseño ni código, por eso hay que utilizar medidas más subjetivas como por ejemplo: los PF.

### Estructuración del cliente.

Para determinar la estructuración del cliente se deben tomar en cuenta los siguientes puntos:

* El grado de seguridad que tiene el cliente sobre sus requerimientos.
* El grado de incertidumbre de los requisitos que se han definido.
* La facilidad con la que se pueden dividir las funciones.
* La naturaleza jerárquica de la información a procesar.

Cuando el grado de incertidumbre estructural resulta ser bajo se puede descomponer mejor el producto y realizar una mejor estimación.

****

## Ámbito del Proyecto.

### Objetivos.

#### Objetivo General.

Desarrollar un software que permita publicar y buscar personas desaparecidas utilizando la tecnología de reconocimiento facial.

#### Objetivo Específico.

Se plantean los siguientes objetivos específicos:

* Registrar datos básicos de algún desaparecido, otorgando una o más fotos más su última ubicación.
* Mirar lista de desaparecidos que estén cerca de su zona.
* Recibir una notificación en caso de que alguna persona lo haya visto o encontrado.
* Definir las actividades del proyecto con el Proceso Unificado de Desarrollo de Software (P.U.D.S.), como metodología de desarrollo del software.
* Realizar la captura de los requerimientos funcionales de la aplicación.
* Analizar los requerimientos para definir los requisitos del software.
* Realizar especificaciones de los requisitos del software mediante modelos propuestos por la herramienta UML, a utilizarse para el modelado de este proyecto.
* Diseñar el software a partir del resultado del análisis del proyecto de manera que los datos se encuentren estructurados de una forma consistente para la óptima utilización de la información.
* Realizar el modelo respectivo de la Base de Datos en tiempo real.
* Implementar el Sistema Móvil y Web utilizando patrón de desarrollo.
* Diseñar una interfaz visual amigable para el usuario, de modo que sea comprensible y fácil de manejar, eludiendo las posibles complicaciones en la utilización del mismo.
* Realizar las pruebas necesarias que nos permitan garantizar el óptimo funcionamiento del software.

### Requerimientos Principales.

El software tendrá la siguiente funcionalidad:

* Registrar Usuario
* Registrar Extraviado
* Inicio de Sesión
* Registrar Comentarios
* Registrar Fotos
* Requisitos Reconocimiento Facial

### Rendimiento.

Al ser un software de interés social que está dirigido al público en general el rendimiento debe ser eficiente puesto que es necesario un tiempo de respuesta ágil por la naturaleza de la aplicación influyendo en esto las características del hardware y del software con que se cuenta

### Fiabilidad.

Al manejar el Software es importante que esté libre de errores, el software tiene que ser lo más confiable posible para que las personas puedan publicar y buscar lo más exactamente posible.

### Integridad y seguridad.

La aplicación es capaz de proteger sus componentes contra los procesos o elementos que no tengan derecho de acceso a los mismos

### Eficiencia.

Implementado con mecanismos de seguridad y es capaz de hacer un buen uso de los recursos que manipula

### Usabilidad.

El usuario podrá interactuar fácilmente con la interfaz que tiene la aplicación web. Seguimos estándares de diseño y de interfaz, es un software de fácil manejo

### Mantenibilidad.

Ya que es software fue diseñado con el modelo vista controlador, tendrá la posibilidad de ser sometido a actualizaciones o mejoras sin necesidad ni perder tiempo, dinero o esfuerzo en ello

### Portabilidad.

El software contara con la facilidad de ser migrado entre diferentes versiones de sistema operativo, es 100% portable, realizado en plataforma web.

### Restricciones.

#### Restricciones Técnicas.

* Se aplicará en el desarrollo del proyecto, el Proceso Unificado de Software (PUDS), utilizando el Lenguaje Unificado de Modelado (UML).
* El Sistema Operativo que podrá utilizarse será: Cualquiera solo precisa de un navegador en él.
* El lenguaje de programación a utilizar será HTML, PHP, JAVASCRIPT sobre el entorno de desarrollo Web.
* Para la elaboración del diseño de diagramas y otros, del desarrollo se dará uso de una herramienta CASE como es el Enterprise Architect.
* Los usuarios finales del producto imponen la creación de una interfaz AMIGABLE y agradable a la vista.

#### Restricciones legales.

* No existen restriccioneslegales.

#### Restricciones recursos.

* El tiempo dado para el desarrollo parcial del proyecto es de 3 meses, lo que implica mayor esfuerzo por parte de los desarrolladores.
* Se cuenta con 3 personas para la elaboración del proyecto

## Técnicas de Estimación.

Para realizar las estimaciones correspondientes al proyecto, se han recurrido a 2 tipos de estimaciones distintas, algunas de las cuales toman en cuenta diversos factores como líneas de código del programa, grado de complejidad del proyecto, esfuerzo, objetos a emplearse y otros más.

* Estimación orientada al Tamaño (**KLDC**).
* Estimación mediante el modelo empírico **COCOMO II**.

Las técnicas de estimaciones que se usará para determinar costo del proyecto, tiempo de construcción del software y la cantidad de personas que intervendrán son:

### Estimación Orientada al tamaño.

* El propósito de este método es poder estimar cuantas KLDC podría tener un proyecto de software que recién vamos a desarrollar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proyecto** | **KLDC** | | | |
| **Optimista** | **Más Probable** | **Pesimista** | **Esperado** |
| MoodAgend |  |  | 7,05 | 2,8 |
| Songza | 1,09 |  |  |
| Stereo |  | 2,17 |  |

**Tabla 10. Estimación Orienta al Tamaño (KLDC)**

1,09 + (4\*2,17) +7,05

Optimista + 4 más probable + pesimista

VE = =

6

6

VE = 2,8

Se estima que se escribirán **2,8 KLDC.**

* Seguiremos el mismo procedimiento para calcular el tiempo que podría requerir el software a desarrollar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proyecto** | **TIEMPO** | | | |
| **Optimista** | **Más Probable** | **Pesimista** | **Esperado** |
| MoodAgend |  |  | 7 | 2,8 |
| Songza |  | 5 |  |
| StereoMood | 3 |  |  |

**Tabla 11. Estimación Orienta al Tamaño (TIEMPO)**

3+ (4\*5) +7

Optimista + 4 más probable + pesimista

VE = =

6

6

VE = **5**

Se estima que se empleara **5** meses para finalizar el proyecto.

Seguiremos el mismo procedimiento para calcular el personal que podría requerir el software a desarrollar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proyecto** | **PERSONAS** | | | |
| **Optimista** | **Más Probable** | **Pesimista** | **Esperado** |
| MoodAgend |  |  | 7 | 2,8 |
| Songza |  | 5 |  |
| StereoMood | 2 |  |  |

**Tabla 12. Estimación Orienta al Tamaño (PERSONAS)**

2+ (4\*5) + 7

Optimista + 4 más probable + pesimista

VE = =

6

6

VE = 4,8 = **5**

Se estima que el equipo de desarrollo estará formado por **5** personas.

### COCOMO II (Modelo Constructivo de Costo).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tipo de Objeto** | **Cta** | **Factor de peso** | | | **Total** |
| **Simple** | **Medio** | **Avanzado** |
| Pantalla | **6** | 1 | **2** | *3* | 12 |
| Informes | **5** | 2 | ***5*** | 8 | 25 |
| Componente 3GL | **2** | - | - | ***10*** | 20 |
| **P.O.** | **57** |

**Tabla 13. Estimación COCOMO II**

**PO =** Puntos Objetos

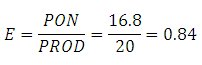
**PON =** Número de puntos objetos

**PON=** PO \* [(100 - % reutilización) / 100] = 57\* [(100 -30%)/100]

= **39,9 (Objeto a desarrollar)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Proporciones de productividad** | **Muy baja** | **Baja** | **Normal** | **Alta** | **Muy alta** |
| Capacidad y madurez del desarrollador |  | **X** |  |  |  |
| Capacidad y madurez del entorno |  | **X** |  |  |  |
| **PROD** | 4 | **7** | 13 | 25 | 50 |
| **14**  **PROD**. |

**Tabla 14. Estimación COCOMO II**

**Esfuerzo**

39,9

***E = = 3 (*personas/mes*)***

14

### Ecuación del Software.

**E = (LDC \* B 0.28 / P)3 \*** (1 / t4) .

Dónde:

**E =** esfuerzo en hombres-año.

**t =** duración del proyecto en años. T=3 meses = 0.25 (años)

**B =** factor especial de destrezas. B=**0.16**

Para programas pequeños B vale 0.16, para programas intermedios vale 0.28, para programas mayores vale 0.39.

**P =** parámetro de productividad, para un software de tiempo real, P vale **2000**, para comunicaciones y sistemas vale 10,000, para software científico vale 12,000 y para aplicaciones comerciales de sistemas vale 28,000.

t min = 8.14 \* (LDC / P) **0,43** para t min >6 meses

**P=**2000

**KLDC =** 2.8 => 2800 LDC

t min = 8.14 \* (2800/2000) **0,43** = 9.4 **= 9 meses**

Se espera un tiempo mínimo de 9 meses para la realización del proyecto.

**E= 180 B(t)3** para E > 20 persona-meses

**B=** 0.16

t = t min /12 = 9/12 = 0.75

**E = 180\*0.16\*(0.75) 3**

**E = 12 (personas/ mes)**

# CAPITULO 3 Planificación del tiempo.



Todo proyecto requiere una planificación del tiempo a emplear en las diversas actividades que se van a llevar a cabo para el cumplimiento del mismo, a través de 2 diagramas se pretende mostrar la distribución de tiempos planificada, primeramente el diagrama de Gantt, a través del cual se podrá apreciar el tiempo que se le va a otorgar para la realización de cada actividad y las actividades que son requisitos para realizar otras actividades.

Mientras que en el diagrama PERT se podrá apreciar las relaciones de cada actividad una con otra y así mismo se podrá ver la ruta critica el proyecto, es decir aquellas actividades que un retraso en las mismas ocasionaría un retraso en todas las demás actividades y por lo tanto un retraso en el proyecto en sí.

En planificación temporal se consideran 2 etapas:

**IDENTIFICAR ACTIVIDADES**

Para desarrollar el software aplicaremos como estrategias las Fases del Proceso Unificado:

**1ra: Fase de Inicio**

1. Requisitos del Software
   1. Captura de Requisitos como Casos de Uso
      1. Encontrar Actores y Casos de Uso
      2. Priorizar Casos de Uso
      3. Detallar un Caso de Uso
      4. Prototipar la Interfaz de Usuario
      5. Estructurar el Modelo de Casos de Uso

**2da: Fase de Elaboración**

1. Análisis del Software
   1. Realizar el Análisis de la Arquitectura
   2. Analizar Casos de Uso
   3. Analizar Clases
   4. Analizar Paquetes

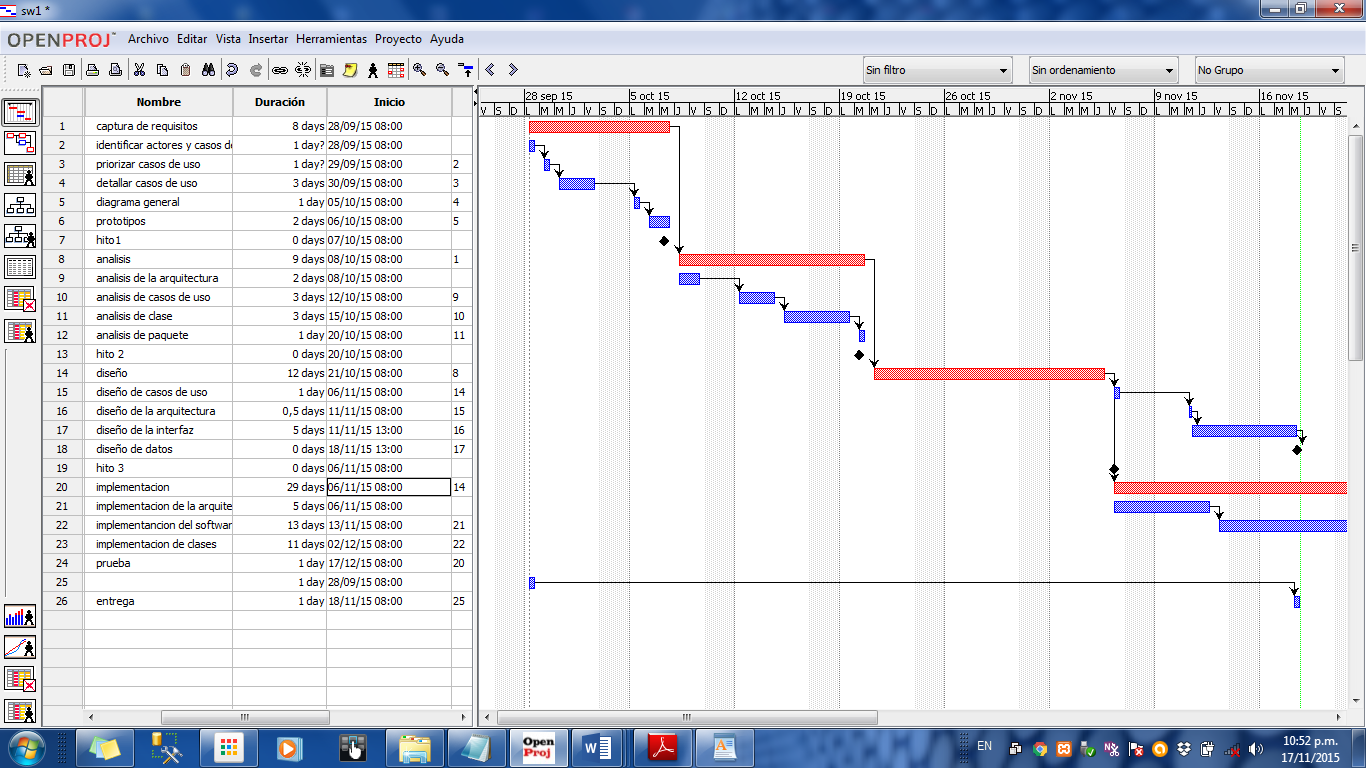
**3ra: Fase de Construcción**

1. Diseño del Software
   1. Realizar el Diseño de la Arquitectura
   2. Diseñar Casos de Uso
   3. Diseñar Clases

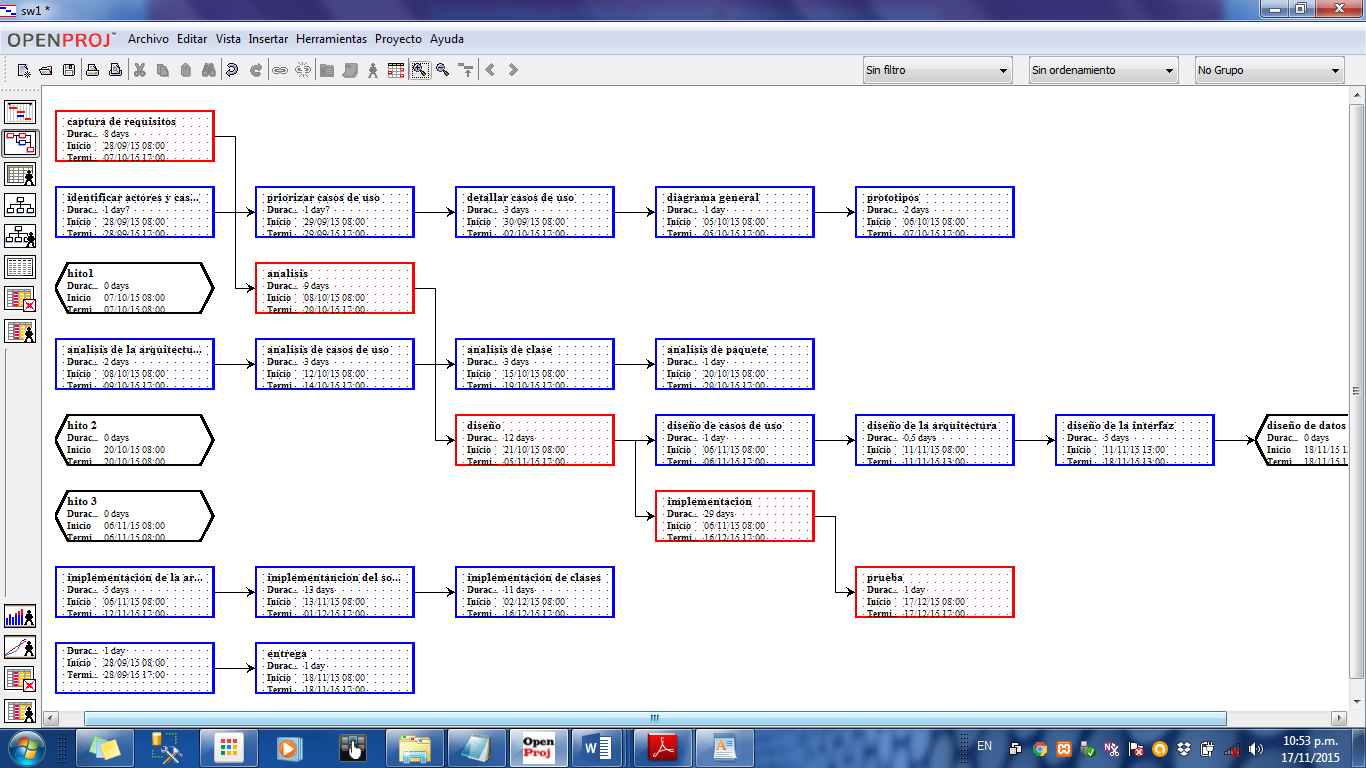
**4ta: Fase de Implementación**

1. Implementación del Software
   1. Realizar la Implementación de la Arquitectura
   2. Integrar el Sistema
   3. Implementar Clases
   4. Realizar pruebas de Unidad

## Diagrama de Gantt.



## Diagrama de Pert.



# CAPITULO 4 Recursos de Proyecto.



## Tabla de recursos.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Recursos | Fechas | | Cantidad | Precio $ | % | Precio $ | Precio $ |
|  | Desde | Hasta |  | Unitario | Depreciación | Unit. Neto | Total |
| Hardware |  |  |  |  |  |  |  |
| Computador Portatil (A1) | 06/09/2016 | 04/10/2016 | 3 | 740 | 25 | 185 | 555 |
| Disposivo Movil (A2) | 10/09/2016 | 04/10/2016 | 1 | 300 | 50 | 150 | 150 |
| Impresora | 06/09/2016 | 04/10/2016 | 1 | 50 | 25 | 12.5 | 12.50 |
| Modem | 06/09/2016 | 04/10/2016 | 1 | 25 | 25 | 6.25 | 6.25 |
| Software |  |  |  |  |  |  |  |
| S.O.(Incluido con computadora portátil) | 06/09/2016 | 04/10/2016 | 3 | 0 | 33 | 0,00 | 0,00 |
| Architect | 06/07/2016 | 04/08/2016 | 3 | 135 | 30 | 40.5 | 40,50 |
| Process | 10/09/2016 | 04/10/2016 | 2 | 0 | 0 | 0,00 | 0,00 |
| Gente |  |  |  |  |  |  |  |
| Secretaria | 06/07/2016 | 07/08/2016 | 1 | 300 | 0 | 300 | 900 |
| Analista | 06/07/2016 | 26/08/2016 | 1 | 600 | 0 | 600 | 1200 |
| Diseñador | 27/07/2016 | 09/08/2016 | 1 | 500 | 0 | 500 | 500 |
| Programadores | 10/07/2016 | 04/08/2016 | 2 | 400 | 0 | 400 | 800 |
| Seguridad | 06/07/2016 | 07/07/2016 | 1 | 300 | 0 | 300 | 900 |
|  |  |  |  |  |  |  |  |
| Infraestructura |  |  |  |  |  |  |  |
| Alquiler oficina | 06/07/2016 | 07/10/2016 |  | 300 | 0 | 300 | 900 |
| Energia Electrica | 06/07/2016 | 07/07/2016 |  | 35 | 0 | 35 | 105 |
| Servicios Agua potable | 06/07/2016 | 07/07/2016 |  | 15 | 0 | 15 | 45 |
| Internet | 06/07/2016 | 07/07/2016 |  | 65 | 0 | 65 | 195 |
|  |  |  |  |  |  |  |  |
| Logistica |  |  |  |  |  |  |  |
| Mat. Escritorio | 06/07/2016 | 07/08/2016 |  | 25 | 0 | 25 | 75 |
| Mat. De Limpieza | 06/07/2016 | 07/0/2016 |  | 15 | 0 | 15 | 45 |
| Refrigerio | 06/07/2016 | 07/07/2016 |  | 42 | 0 | 42 | 126 |
|  |  |  |  |  |  |  | 6429,25 |

# CAPITULO 5 Análisis de Riesgo.



## Tabla de Riesgo y plan de aversión para cada riesgo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgo** | **% Prob*.*** | **Impacto** | **Plan de aversión** | |
| **Reducir probabilidad** | **Reducir Impacto** |
| R1: Fallas en el Hardware o software. | 40 | Significativo | - Hacer el respectivo mantenimiento preventivo a los equipos periódicamente.  - Guardar la información con la que se está trabajando en otros dispositivos. | - Obtener equipos y/o Accesorios Informáticos con garantía.  - Hacer los trabajos en equipos nuevos y las instalaciones se encuentren en buenas condiciones. |
| R2: Perdida de la información por software malintencionado (virus). | 30 | Critico | - Realizar periódicamente copias de las aplicaciones en desarrollo.  -Realizar las actualizaciones correspondientes a los antivirus de los equipos. | -Almacenar en otros dispositivos como CD, Discos Duros de reserva y que sean de buena calidad. |
| R3: Desconocimiento de la estrategia de desarrollo. | 20 | Moderado | - Emplear metodologías usadas con anterioridad.  - Realizar la capacitación a los desarrolladores en las nuevas tecnologías. | -Trabajar de manera organizada y minuciosamente, empleando toda la documentación posible acerca de la estrategia a utilizar. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgo** | **% Prob.** | **Impacto** | **Plan de aversión** | |
| **Reducir Probabilidad** | **Reducir Impacto** |
| R4: Herramienta de desarrollo desconocida. | 20 | Significativo | - Trabajar con herramientas empleadas previamente.  - Capacitar periódicamente al personal. | -Buscar información y ayuda en el manejo de la misma. |
| R5: Mala estimación de tiempo debido a no contemplar actividades ajenas al proyecto. | 70 | Critico | - En el momento de realizar la planificación de tiempos y actividades, realizarla siguiendo un calendario en el cual contemple posibles eventualidades y tiempos reales de trabajo de los integrantes del equipo. | -Contemplar dentro de la planificación de tiempo un tiempo de demora u holgura asumiendo cualquier tipo de eventualidad. |
| R6: Mala estimación de costo. | 60 | Critico | -Elaborar un plan de estimación bien detallado y con datos lo más aproximadamente posibles al proyecto. | - Realizar un informe detallado del costo real del software.  - Consultar un Tutor. |
| R7: Mala estimación de esfuerzo. | 55 | Moderado | - Elaborar varios métodos de estimación del software. | -Aumentar el personal asignado al desarrollo del software. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgo** | **% Prob.** | **Impacto** | **Plan de aversión** | |
| **Reducir Probabilidad** | **Reducir Impacto** |
| R8: Problemas de comunicación entre los desarrolladores. | 80 | Significativo | - Realizar actividades para incentivar la buena comunicación entre desarrolladores. | -Solucionar los problemas de comunicación. |
|  |  |  | - Especificar las tareas que debe realizar cada desarrollador detallando las fechas de presentación. | - Explicar a las personas que hayan malinterpretado las tareas que deberían realizar para que lo corrijan. |
|  |  |  | - Que el inmediato superior al desarrollador este siempre dispuesto a despejar las dudas del desarrollador en cada etapa del desarrollo del proyecto. | - Incentivar al equipo de desarrollo. |
| R9: Cambio de Requisitos. | 65 | Critico | - Realizar una captura de requerimientos minuciosa sobre las funciones que tiene que realizar el software e investigar detalladamente como lo hacían antes. | - Informar al cliente de que los cambios que solicita afectan en el tiempo, costo y esfuerzo del proyecto. |
|  |  |  |  | - Realizar los cambios en el software y tratar de mantener la planificación. |
|  |  |  |  | - Informar al equipo de desarrollo de los cambios en la planificación del proyecto. |

# CAPITULO 6 Organización Interna.



## Estructura del equipo.

La estructura de equipo que se utilizará en el desarrollo del proyecto, será Descentralizada Democrática (DD), planteada por Mantei, la cual especifica lo siguiente:

Este equipo de ingeniería del software no tiene un jefe permanente. Más bien, “se nombran coordinadores de tareas a corto plazo y se sustituyen por otros para diferentes tareas”.

Las decisiones sobre problemas y los enfoques se hacen por consenso del grupo.

La comunicación entre los miembros del equipo es horizontal. En consecuencia, el organigrama será el siguiente:

**DESARROLLADOR**

**DESARROLLADOR**

**DESARROLLADOR**

**DESARROLLADOR**

# CAPITULO 7 Mecanismos de Seguimiento y Control.



## Descripción.

En toda organización de software confronta los problemas implícitos que conlleva el desarrollo de un software; tales como no entregar el software en el tiempo establecido, que el software se desarrolla con el presupuesto y recursos definidos.

Para que estos problemas no se den en el momento menos esperado y no se produzca un impacto sobre el desarrollo del software se debe realizar un seguimiento y control del proyecto.

## Seguimiento.

Realizar RTF (reuniones técnicas formales) con el personal de desarrollo de software para tratar sobre el estado del proyecto, en las cuales cada miembro:

* Dé un informe de los progresos de las tareas definidas y de los hitos, y también de los problemas.
* Dé a conocer sus valoraciones subjetivas acerca de los problemas se pueden presentar más adelante en el desarrollo del software.

## Control.

Para llevar un control del desarrollo del software, una vez determinado un problema, se deben disponer recursos para el área problemática; se puede volver a organizar el personal o rediseñar la agenda del proyecto.

## Reporte de Tareas.

Es importante plasmar cada observación de las actividades en información física que puede ser almacena como constancia, para ello utilizaremos reportes elaborados para motivos diferentes:

* Reporte de Actividad.
* Reporte de Problemas.
* Reporte de Prueba.
* Reporte de Actividad concluida.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **REPORTE DE ACTIVIDAD** | | | | | | **Pág.**  …...... | |
| **Proyecto:**  **Responsable:** | | | | **Fecha de Elaboración:** | | | |
| ***Nombre del Sistema o Unidad:***  **Fase:** | | | | | | | |
| 1. Nombre del Empleado: 2. Unidad o Proceso Revisado:   ( ) Terminado ( ) No terminado   1. Descripción del Avance (Procesos que Faltan):        1. Tiempo estimado para concluir lo faltante: | | | | | | | |
|  | |  | | | | |  |
| **COMENTARIOS**: | | | | |
|  | | | | |
|  | | | | | | | |
| **REPORTE DE PROBLEMA** | | | | | **Pág.**  …...... | |
| **Proyecto:**  **Responsable:** | | | **Fecha de Elaboración:** | | | |
| ***Nombre del Sistema o Unidad:*** | | | | | | |
| 1. Identificación del Problema:      1. Descripción:          1. Evento ejecutado cuando se presentó el Problema:        1. Posibles orígenes del Problema: | | | | | | |
|  |  | | | | |  |
| **COMENTARIOS**: | | | | |
|  | | | | |
| **FIRMA DEL RESPONSABLE** | | | | | | |
| **REPORTE DE PRUEBA REALIZADA** | | | | | **Pág.**  …...... | |
| **Proyecto:**  **Responsable:** | | | **Fecha de Elaboración:** | | | |
| ***Nombre del Sistema o Unidad:*** | | | | | | |
| **TIPO DE PRUEBA** ( ) De unidad ( ) De integración ( ) De sistema ( ) De aceptación  **ELEMENTO A PROBAR:**   |  |  |  |  | | --- | --- | --- | --- | | **Dato(s)**  **Introducidos** | **Resultado**  **Esperado** | **Resultado**  **Obtenido** | **Valido (V)**  **Invalido (I)** | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | | | | | | |
|  |  | | | | |  |
| **COMENTARIOS**: | | | | |
|  | | | | |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **FIRMA DEL RESPONSABLE** | | | | | | |
| **REPORTE CONCLUSIÓN DE TAREA** | | | | | **Pág.**  …...... | |
| **Proyecto:**  **Responsable:** | | | **Fecha de Elaboración:** | | | |
| ***Nombre del Sistema o Unidad:*** | | | | | | |
| **DESCRIPCIÓN DE LA TAREA:**  **Fecha Estimada:** **Fecha de Conclusión:**  **DATOS DE ERRORES:**   |  |  |  |  | | --- | --- | --- | --- | | **Errores encontrados** | **Número** | **Coste Unitario** | **Totales** | | **Llevando a cabo Revisiones** | | | | | **Durante el Diseño** |  |  |  | | **Antes de la Prueba** |  |  |  | | **Durante la Prueba** |  |  |  | | **Sin Revisiones** | | | | | **Antes de la Prueba** |  |  |  | | **Durante la Prueba** |  |  |  | |  |  |  |  | | | | | | | |
|  |  | | | | |  |
| **COMENTARIOS**: | | | | |
|  | | | | |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **FIRMA DEL RESPONSABLE** | | | | | | |

# REFERENCIAS

* + Jacobson, Booch, Rumbaugh -“PUDS” El proceso Unificado de Desarrollo de Software
  + Jacobson, Booch, Rumbaugh -“UML” Lenguaje Unificado Modelado
  + Roger S. Pressman- Ingeniería del Software Séptima Edición.
  + Internet
* [www.google.com](http://www.google.com)
* [www.ACM.org](http://www.ACM.org)
* [www.taringa.net](http://www.taringa.net)
* www.wikipedia.com

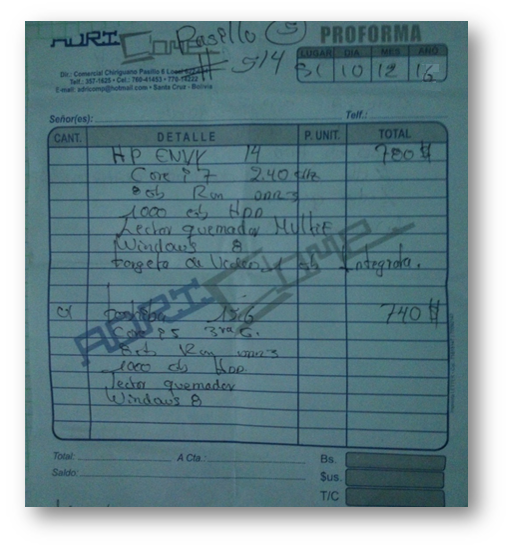
# ANEXOS

**Recursos.**

**Cada Computador Portátil**

Deberá contar con las siguientes características:

* DISCO DURO 1000 GB SATA 3.5”.
* Memoria 8 GB DDR3 1333.
* Procesador Intel Core I7 6700 6ta Sexta Generacion.
* Lector quemador.



Cada Impresora:

* EPSON L355

Software que se precisan son:

* Sistema Operativo: Windows 10 Pro
* Android Studio