**PLAN DE CALIDAD**

**REINEL YOHANNY LOPEZ VALENCIA**

**SENA**

**CENTRO DE DISEÑO Y MANUFACTURA DEL CUERO**

**TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE INFORMACIÓN**

**ITAGÜÍ**

**CONTENIDO**

[INTRODUCCION 3](#_Toc434686035)

[OBJETIVOS 4](#_Toc434686036)

[OBJETIVO GENERAL 4](#_Toc434686037)

[OBJETIVOS ESPECIFICOS 4](#_Toc434686038)

[METAS DE CALIDAD 5](#_Toc434686039)

[ASIGNACION DE RESPONSABILIDADES 6](#_Toc434686040)

[PLAN DEL PRODUCTO. 7](#_Toc434686041)

[MÈTRICAS A APLICAR 8](#_Toc434686042)

[ MÉTRICAS DE FUNCIONALIDAD: 8](#_Toc434686043)

[ MÉTRICAS DE FIABILIDAD: 8](#_Toc434686044)

[ MÉTRICAS DE USABILIDAD: 9](#_Toc434686045)

[ MÉTRICAS DE EFICIENCIA: 9](#_Toc434686046)

[ MÉTRICAS DE MANTENIBILIDAD: 9](#_Toc434686047)

[ MÉTRICAS DE TRANSPORTABILIDAD: 10](#_Toc434686048)

[HERRAMIENTAS A UTILIZAR 10](#_Toc434686049)

[PLAN DE CAPACITACIÓN 11](#_Toc434686050)

[FACTORES DE CALIDAD 12](#_Toc434686051)

[ OPERACIÓN DEL PRODUCTO: 12](#_Toc434686052)

[ REVISIÓN DEL PRODUCTO: 14](#_Toc434686053)

[CONCLUSIONES 17](#_Toc434686054)

[BIBLIOGRAFIA 18](#_Toc434686055)

# INTRODUCCION

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la metodología del trabajo, en aras de lograr una mayor confiabilidad, mantenimiento y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad; para garantizarle al cliente un software optimo y funcional según sus necesidades, se debe planear las pruebas y diagnósticos a realizar y de esta manera garantizar un software estable para el cliente, el cliente es quien determina si se encuentra satisfacción al interactuar con el software, al observar si es funcional según sus necesidades y además si el software le permite realizar las funciones para la cual fue creado. Para garantizar lo anterior y no poner en tela de juicio el desempeño del desarrollo se debe someter el software a los diferentes criterios de calidad establecidos a nivel internacional.

Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad.

# OBJETIVOS

## OBJETIVO GENERAL

Diseñar e implementar técnicas para garantizar la calidad en el desarrollo del software, y de esta manera garantizarlo al cliente.

## OBJETIVOS ESPECIFICOS

* Descubrir errores en la función, la lógica o la implementación del software.
* verificar que el software bajo revisión alcanza sus requerimientos.
* Garantizar el uso de estándares predefinidos por los diferentes tipos de normas.
* Conseguir que el software se desarrolle de forma uniforme y que cada uno de los integrantes del equipo de trabajo cumplan con los diferentes lineamientos según el resultado esperado.
* Diseñar e implementar planes de acción y planes de contingencia todo esto para garantizar la continuidad y la calidad del desarrollo del software.

# METAS DE CALIDAD

El objetivo fundamental es garantizar que los procesos definidos que estén siendo respetados en el desarrollo, así como poder detectar deficiencias en la forma de trabajar establecida. Las metas de proceso son:

* Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y cada uno de las herramientas ofimáticas utilizadas.
* Identificar y documentar el software y cada uno de los procesos para que se tenga claridad en cuanto su desarrollo.
* Proporcionar información a las personas que están usando los procesos.
* Asegurar de que las no conformidades son tratadas.
* Realizar retroalimentación de cada uno de los procesos establecidos, para realizar mejoras y establecer planes que puedan aumentar la calidad del software.
* Garantizar que cada uno de los requisitos se cumpla y se lleve a cabo su desarrollo.
* Garantizar que el software sea seguro y cumpla con los estándares establecidos por las normas.
* Garantizar que el cliente quede a satisfacción con el software.

Para garantía de los efectos de calidad se estima un cumplimento del 85%, teniendo como principal objetivo alcanzar un 100%. También se espera logar que la seguridad del software cumpla con el 90%, que el cliente tenga una satisfacción del 95% alcanzando y desarrollando cada uno de los requisitos establecidos en el momento de plantear el proyecto.

# ASIGNACION DE RESPONSABILIDADES

|  |  |  |
| --- | --- | --- |
| NOMBRE | CARGO | FUNCIONES |
| YOHANNY LOPEZ VALENCIA | Programador | * Realizar en el lenguaje de programación estipulado por el cliente, las diferentes tareas de programar y armar el software, según los requisitos elaborados con anterioridad * Recolectar la información necesaria en el proceso de programación las diferentes eventualidades y modificaciones realizadas. * Documentar el proceso realizado * Crear código limpio y funcional * Aplicar el modelo de calidad escogido. |
| Documentador | * Recolectar la información de todas las fases del desarrollo del software. * Realizar formatos para recolectar la información de cada proceso. * diseñar cronogramas; en los que se deben cumplir cada una de las tareas programadas * Aplicar el modelo de calidad escogido. * Revisar los diferentes documentos del proyecto. |
| Diseñador | * Diseñar los diferentes prototipos del sistema. * Documentar el proceso realizado. * Aplicar el modelo de calidad escogido. * Crear un diseño único, funcional e iterativo para el cliente. |
| Analista de Bases de Datos | * Realizar el proceso de levantamiento de requisitos. * Diseñar y evaluar la base de datos que el software va utilizar. * Hacer pruebas y aplicar diferentes métodos para determinar el correcto flujo de la información. * Documentar el proceso realizado. * Aplicar el modelo de calidad escogido. |

# PLAN DEL PRODUCTO.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ACTIVIDAD | HORAS | RESPONSABLES | RESULTADO | FECHA INICIO | FECHA FINAL | | RECURSOS |
| Identificar del sistema | 5 horas | Yohanny Lopez Valencia | Información del sistema a realizar. Después de haber escuchado y revisado las necesidades del cliente | 10/10/2016 | | 10/10/2016 | Microsoft Word |
| levantar de requisitos | 30 horas | los diferentes documentos donde se especifican las funciones del software (DAS y DRS) | 09-10-2016 | | 03-11-2015 | Microsoft Word |
| Analizar los procesos | 10 horas | Procesos a realizar (se definen tareas y actividades) | 06-11-2016 | | 17-11-2016 | Microsoft Project |
| Reestructurar procesos | 10horas | Durante todo el proceso se deben ajustar tareas según lo que valla surgiendo. | 02-10-2016 | | 05-05-2017 | Microsoft Project |
| Planificar | 8 horas | Cronograma de actividades | 21-11-2016 | | 24-11-2016 | Microsoft Project |
| Diseñar la BD | 58 horas | Visualización de la futura BD | 27-11-2016 | | 31-11-2016 | Microsoft Visio |
| Ejecutar de la BD | 50 horas | Base de datos funcional | 03-12-2016 | | 21-12-2016 | MYSQL |
| Diseñar el software | 100 horas | Prototipos funcionales | 01-11-2016 | | 25-01-2017 | Adobe photoshop |
| Codificar/Programar | 290 horas | Software funcional en un 70%  Código y funcionalidad del software | 26-02-2017 | | 04-04-2017 | Sublime text en el lenguaje php |
| Implantar el software | 150 horas | Software listo en un 90% | 04-04-2017 | | 20-04-2017 | Navegador Web |
| Explotar | indefinido | indefinido | 20-04-2017 | | 30-04-2017 | Navegador web |
| Mejorar | indefinido | indefinido | 20-04-2017 | | 30-04-2017 | Todos |
| Ampliar | indefinido | indefinido | 20-04-2017 | | 30-04-2017 | todos |
| Corregir de errores | 150horas | Software funcional en un 100% | 20-04-2017 | | 30-04-2017 | todos |

# MÈTRICAS A APLICAR

Las métricas de software se utilizan para propósitos estratégicos y son utilizadas en el proyecto para minimizar la planificación de desarrollo haciendo los ajustes necesarios que eviten retrasos y reduzcan problemas y riesgos potenciales, son utilizadas también para evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad. Para establecer objetivos de mejora durante el proceso de desarrollo de software, se debe comprender el estado actual del desarrollo del software. Si no se mide, no hay una forma real de determinar si se está mejorando y si no se está mejorando, se está perdido. A continuación listaremos las diferentes métricas y daremos una breve definición de cada una:

* **Métricas de Funcionalidad:**

**Nombre:** Completitud de implementación funcional

**Propósito:** Qué tan completa está la implementación funcional.

**Método de aplicación:** Contar las funciones faltantes detectadas en la evaluación y comparar con el número de funciones descritas en la especificación de requisitos.

**Medición, fórmula:** X = 1 - A/B

A = número de funciones faltantes

B = número de funciones descritas en la especificación de requisitos

**Interpretación:** 0 <= X <= 1

Entre más cercano a 1, más completa.

* **Métricas de Fiabilidad:**

**Nombre:** Suficiencia de las pruebas

**Propósito:** Cuántas de los casos de prueba necesarios están cubiertas por el plan de pruebas.

**Método de aplicación:** Contar las pruebas planeadas y comparar con el número de pruebas requeridas para obtener una cobertura adecuada.

**Medición, fórmula:** X = A/B

A = número de casos de prueba en el plan

B = número de casos de prueba requeridos

**Interpretación:** 0 <= X

Entre X sea mayor, mejor la suficiencia.

* **Métricas de Usabilidad:**

**Nombre:** Funciones evidentes

**Propósito:** Qué proporción de las funciones del sistemas son evidentes al usuario.

**Método de aplicación:** Contar las funciones evidentes al usuario y comparar con el número total de funciones.

Medición, fórmula: X = A/B

A = número de funciones (o tipos de funciones) evidentes al usuario

B = total de funciones (o tipos de funciones)

**Interpretación:** 0 <= X <= 1

Entre más cercano a 1, mejor.

* **Métricas de Eficiencia**:

**Nombre:** Tiempo de respuesta

**Propósito:** Cuál es el tiempo estimado para completar una tarea.

**Método de aplicación:** Evaluar la eficiencia de las llamadas al SO y a la aplicación.

Estimar el tiempo de respuesta basado en ello. Puede medirse:

Todo o partes de las especificaciones de diseño.

Probar la ruta completa de una transacción.

Probar módulos o partes completas del producto.

Producto completo durante la fase de pruebas.

**Medición, fórmula:** X = tiempo (calculado o simulado)

**Interpretación:** Entre más corto, mejor.

* **Métricas de Mantenibilidad:**

**Nombre:** Registrabilidad de cambios

**Propósito:** ¿Se registran adecuadamente los cambios a la especificación y a los módulos con comentarios en el código?

**Método de aplicación:** Registrar la proporción de información sobre cambios a los módulos

**Medición, fórmula:** X = A/B

A = número de cambios a funciones o módulos que tienen comentarios confirmados

B = total de funciones o módulos modificados

**Interpretación:** 0 <= X <= 1

Entre más cercano a 1, más registrable.

0 indica un control de cambios deficiente o pocos cambios y alta estabilidad.

* **Métricas de Transportabilidad:**

**Nombre:** Conformidad de transportabilidad.

**Propósito:** Qué tan conforme es la transportabilidad del producto con regulaciones, estándares y convenciones aplicables.

**Método de aplicación:** Contar los artículos encontrados que requieren conformidad y comparar con el número de artículos en la especificación que requieren conformidad.

**Medición, fórmula:** X = A/B

A = número de artículos implementados de conformidad

B = total de artículos que requieren conformidad

**Interpretación:** 0 <= X <= 1

Entre más cercano a 1, más completa.

**HERRAMIENTAS A UTILIZAR**

* **Microsoft Word:** Esta herramienta nos permitirá llevar la documentación digital de todos los procesos, de cada una de las pruebas de calidad realizadas y de los diferentes documentos necesarios para el desarrollo.
* **Microsoft Project:** Esta herramienta es indispensable para realizar la planeación estratégica de las actividades y tareas.
* **Microsoft Visio:** Con esta útil herramienta podremos realizar cada uno de los diagramas necesarios para los diferentes documentos que apoyan la labor de análisis y de desarrollo.
* **Microsoft Excel:** gracias a este programa podremos realizar cada una de las operaciones matemáticas y de medición establecidos para el software, de igual manera nos permitirá visualizar mediante graficas el desempeño y la calidad del software.
* **WORKBENCH:** En este programas se realiza la programación y ejecución de la base de datos utilizada para el software.
* **Adobe photoshop:** Con este programa realizaremos los prototipos y bocetos funcionales del software.
* **Sublime text:** Con esa herramienta programaremos el código utilizando el lenguaje php.

# PLAN DE CAPACITACIÓN

El plan de capacitación del proyecto se realiza mediante el desarrollo de algunos pasos los cuales se diseñara e implementara diferentes materiales, con el apoyo bibliográfico necesario para que el grupo de trabajo tenga un enfoque dinámico y que además cada uno de los integrantes pueda desenvolverse con facilidad cuando se habla del proyecto y de las diferentes métricas de calidad que se aplicaran. Se guiara con la información del sistema aclarando dudas y utilizando estrategias para que el trabajo sea lo más dinámico posible. Para obtener buenos resultados es aplicando procesos de mejora continua, siempre buscando tener claros todos los conceptos, habilidades, conocimientos y estructuras técnicas del proyecto; teniendo como principal objetivo el alcance de alta calidad del software.

Este plan de calidad tiene algunos pasos por eso a continuación se muestran:

* Diseñar
* Analizar
* Validar
* Aplicar
* Evaluar

# FACTORES DE CALIDAD

* **Operación del producto:**

|  |  |
| --- | --- |
| **Facilidad de uso** | - Facilidad de operación: Atributos del software que determinan la facilidad de operación del software. En el diseño se realizaran menús intuitivos y de fácil manejo para cualquier persona.  - Facilidad de comunicación: Atributos del software que proporcionan entradas y salidas fácilmente asimilables.  - Facilidad de aprendizaje: Atributos del software que facilitan la familiarización inicial del usuario con el software y la transición del modo actual de operación, utilizando manuales y guías de multimedia más cercanas e intuitivas y no teóricas.  - Formación: El grado en que el software ayuda para permitir que nuevos usuarios apliquen el sistema. |
| **Integridad** | - Control de accesos. Atributos del software que proporcionan control de acceso al software y los datos que maneja; un correcto y adecuado sistema d seguridad.  - Facilidad de auditoría: Atributos del software que facilitan la auditoría de los accesos al software. Se deben diseñas los diferentes formatos que permitan una ágil y adecuada documentación de cada proceso.  - Seguridad: La disponibilidad de mecanismos que controlen o protejan los programas o los datos. En gran medida al momento de diseñar la base de datos asignar y controlar correctamente los roles del cada usuario. |
| **Corrección** | - Completitud: Atributos del software que proporcionan la implementación completa de todas las funciones requeridas. Teniendo como base y objetivo principal satisfacer al cliente y alcanzar cada uno de los requisitos expuestos en el DRS.  - Consistencia: Atributos del software que proporcionan uniformidad en las técnicas y notaciones de diseño e implementación. Se debe tener en cuenta que el código debe estar correctamente identado, comentado y documentado todo con el fin de evitar redundancias y reprocesos.  - Trazabilidad o rastreabilidad: Atributos del software que proporcionan una traza desde los requisitos a la implementación con respecto a un entorno operativo concreto. |
| **Fiabilidad** | - Precisión: Atributos del software que proporcionan el grado de precisión requerido en los cálculos y los resultados. En la medida que cada algoritmo matemático sea optimo y contenga las formulas y cálculos correctos, en nuestro caso para los módulos de facturación.  - Consistencia.  - Tolerancia a fallos: Atributos del software que posibilitan la continuidad del funcionamiento bajo condiciones no usuales. Diseñar planes de contingencia al momento de armar el código para que el software siga en funcionamiento.  - Modularidad: Atributos del software que proporcionan una estructura de módulos altamente independientes. En nuestro caso los 10 módulos en los que está compuesto el software cada uno independiente.  - Simplicidad: Atributos del software que posibilitan la implementación de funciones de la forma más comprensible posible.  - Exactitud: La precisión de los cálculos y del control. |
| **Eficiencia** | - Eficiencia en ejecución: Atributos del software que minimizan el tiempo de procesamiento. Tanto en los tiempos de respuesta como en las funciones que sean lo más simples posibles. Para el proyecto stile soft se implementaran menús intuitivos y respuestas rápidas desde la base de datos.  - Eficiencia en almacenamiento: Atributos del software que minimizan el espacio de almacenamiento necesario. |

* **Revisión del producto:**

|  |  |
| --- | --- |
| **Facilidad de mantenimiento** | - Modularidad.  - Simplicidad.  - Consistencia.  - Concisión: Atributos del software que posibilitan la implementación de una función con la menor cantidad de códigos posible. Diseño de código lo más óptimo posible.  - Auto descripción: Atributos del software que proporcionan explicaciones sobre la implementación de las funciones. |
| **Facilidad de prueba** | - Modularidad.  - Simplicidad.  - Auto descripción.  - Instrumentación: Atributos del software que posibilitan la observación del comportamiento del software durante su ejecución para facilitar las mediciones del uso o la identificación de errores. |
| **Flexibilidad** | - Auto descripción.  - Capacidad de expansión: Atributos del software que posibilitan la expansión del software en cuanto a capacidades funcionales y datos.  - Generalidad: Atributos del software que proporcionan amplitud a las funciones implementadas.  - Modularidad. |
| **Reusabilidad** | - Auto descripción.  - Generalidad.  - Modularidad.  -Independencia entre sistema y software: Atributos del software que determinan su dependencia del entorno operativo.  - Independencia del hardware: Atributos del software que determinan su dependencia del hardware. |
| **Interoperabilidad** | - Modularidad.  - Compatibilidad de comunicaciones: Atributos del software que posibilitan el uso de protocolos de comunicación e interfaces estándar.  - Compatibilidad de datos: Atributos del software que posibilitan el uso representaciones de datos estándar.  - Estandarización en los datos: El uso de estructuras de datos y de tipos estándar a lo largo de todo el programa. |
| **Portabilidad** | - Auto descripción.  - Modularidad.  -Independencia entre sistema y software.  - Independencia del hardware. |

# CONCLUSIONES

* Para las evaluaciones que se quieran obtener es necesario la utilización de medidas técnicas, que evalúan la calidad de manera objetiva y que permitan establecer fallos y puntos a mejorar de esta manera lograr una buena calidad del producto.
* Cada integrante debe tener claro sus funciones y también debe ser consiente de las labores que está realizando, comprometiéndose con la calidad del software en todo momento.
* También son productos los artefactos, documentos, modelos, y componentes que conforman el software. Se miden cosas como el tamaño, la calidad, la totalidad, la volatilidad, y el esfuerzo.
* El modelo de Mccall describe la calidad como un concepto elaborado mediante relaciones jerárquicas entre factores de calidad, en base a criterios.
* Las métricas de calidad son comúnmente utilizadas por las organizaciones como una forma de medir el rendimiento de los procesos internos, específicamente los procesos que son difíciles de medir como el software.

**BIBLIOGRAFIA**

Instituto para la calidad PUCP, 07-nov 2015, lima-Perú: <http://calidad.pucp.edu.pe/wiki-calidad/como-medir-las-metricas-de-calidad#sthash.BMMJlOgr.dpbs>

Métricas de calidad de software, 07- nov- 2015: http://ldc.usb.ve/~abianc/materias/ci4712/metricas.pdf