

**MANUAL DE CALIDAD**

**PROGRAM-ART**

NUMERO DE REVISIÓN 1

CÓDIGO

ISO 9001/2008, ISO 9000/2000 E ISO 25000

|  |  |  |
| --- | --- | --- |
| Elaborado por: | Revisado por: | Aprobado por: |
| Arlinson Esneider Rojas | Program-Art | Yonathan Andrés Pérez |
| Fecha: 30/08 /2015 | Fecha: 31/08 /2015 | Fecha: / / |

Contenido

[1. GENERALIDADES. 6](#_Toc428776219)

[1.1 Objeto 6](#_Toc428776220)

[1.2 Presentación de la empresa. 6](#_Toc428776221)

[1.2.1 ¿Quiénes somos? 6](#_Toc428776222)

[1.2.2 Misión 7](#_Toc428776223)

[1.2.3 Visión 7](#_Toc428776224)

[1.2.4 Objetivos Program – Art 7](#_Toc428776225)

[1.2.5 Proceso de desarrollo del software 8](#_Toc428776226)

[1.3 Objetivo 9](#_Toc428776227)

[1.3.1 Políticas de calidad del equipo de trabajo 10](#_Toc428776228)

[1.4 Alcance 14](#_Toc428776229)

[1.5 Exclusiones 14](#_Toc428776230)

[1.6 Definiciones 14](#_Toc428776231)

[1.6.1 Términos ISO 9000/2000 15](#_Toc428776232)

[1.6.2 Términos Program-Art 16](#_Toc428776233)

[1.6.3 Acrónimos 19](#_Toc428776234)

[2. RESPONSABILIDAD Y AUTORIDAD. 20](#_Toc428776235)

[2.1 Estructura departamental 20](#_Toc428776236)

[2.1.1 El Departamento de Desarrollo: 20](#_Toc428776237)

[2.1.2 El Departamento de Producción: 20](#_Toc428776238)

[2.1.3 El Departamento de Software: 21](#_Toc428776239)

[2.1.4 El Departamento gestión: 21](#_Toc428776240)

[2.2 Responsabilidad y autoridad 22](#_Toc428776241)

[2.2.1 Director de calidad 23](#_Toc428776242)

[2.2.2 Responsable de calidad del diseño 25](#_Toc428776243)

[2.2.3 Responsable de calidad de producción 26](#_Toc428776244)

[2.2.4 Responsable de calidad de servicios 28](#_Toc428776245)

[2.2.5 Responsabilidades de los Directores de Departamento 29](#_Toc428776246)

[2.3 Matriz de responsabilidades 30](#_Toc428776247)

[3. SISTEMA DE GESTIÓN DE CALIDAD 32](#_Toc428776248)

[3.1 Requisitos Generales 32](#_Toc428776249)

[3.1.1 Objetivo Del Plan De Calidad 32](#_Toc428776250)

[3.1.2 Estructura Documental Del Sistema De Calidad 33](#_Toc428776251)

[3.2 Requisitos De La Documentación 34](#_Toc428776252)

[3.2.1 Generalidades 34](#_Toc428776253)

[3.2.2 Manual De Calidad 34](#_Toc428776254)

[3.2.3 Control De Los Documentos 35](#_Toc428776255)

[3.2.4 Control De Los Registros 35](#_Toc428776256)

[4. GESTIÓN DE RECURSOS 36](#_Toc428776257)

[4.1 Consideraciones financieras 36](#_Toc428776258)

[4.2 Recursos humanos 37](#_Toc428776259)

[4.3 Recursos materiales 37](#_Toc428776260)

[5. REALIZACIÓN DEL PRODUCTO 38](#_Toc428776261)

[5.1 Planeación de la Realización del Producto 38](#_Toc428776262)

[5.1.1 Modelo en espiral 39](#_Toc428776263)

[5.2 Procesos Relacionados con el Cliente 42](#_Toc428776264)

[5.3 Diseño y Desarrollo 42](#_Toc428776265)

[5.4 Compras 44](#_Toc428776266)

[5.4.1 Proceso de compras 44](#_Toc428776267)

[5.4.2 beneficio para el sistema 44](#_Toc428776268)

[5.5 Producción y Prestación del Servicio 45](#_Toc428776269)

[5.6 Control de los Dispositivos de Monitoreo y Medición 45](#_Toc428776270)

[5.6.1 Fase de Monitoreo y medición 46](#_Toc428776271)

[6. ADMINISTRACIÓN DE LA CONFIGURACIÓN 48](#_Toc428776272)

[6.1 Introducción 48](#_Toc428776273)

[6.2 Definición de versiones 48](#_Toc428776274)

[6.3 Autoevaluación 50](#_Toc428776275)

[6.4 Mejora continua 51](#_Toc428776276)

[6.5 Métricas de calidad de software 52](#_Toc428776277)

[6.5.1 Introducción 52](#_Toc428776278)

[6.5.2 Objetivo 52](#_Toc428776279)

[6.5.3 Conceptos 52](#_Toc428776280)

[6.5.4 Factores y métricas de la calidad de software 54](#_Toc428776281)

[6.5.5 Análisis comparativo (técnico) 55](#_Toc428776282)

[6.5.6 Conclusión 55](#_Toc428776283)

[6.5.7 Responsables de la Evaluación 55](#_Toc428776284)

[7 ANEXOS 56](#_Toc428776285)

[Anexo #1 56](#_Toc428776286)

[Anexo #2 56](#_Toc428776287)

[WEB GRAFÍA 63](#_Toc428776288)

# 1. GENERALIDADES.

## 1.1 Objeto

El manual de calidad del sistema **Program-Art,** es un documento que tiene por objeto la administración del mismo, en él se describen las disposiciones generales para la calidad del software logrado a través de un análisis basado en la norma ISO 9001: 2008, mejorando así la descripción y calificación del software, manteniendo actualizado el Sistema de Calidad basado en la norma ISO 9001, a través de la indicación de su estructura y componentes.

## 1.2 Presentación de la empresa.

### 1.2.1 ¿Quiénes somos?

Program-Art, nace por la necesidad de sistematizar un proceso manual que se lleva a cabo actualmente en la institución SENA, en el que se realiza el control de asistencia de los aprendices, pero este proceso tiene deficiencias en las que puede presentar suplantación de identidad, además se pierde mucho tiempo realizando este proceso.

Program-art consiste en un software que permite realizar el control de asistencia de los aprendices por medio del código de barras del carnet estudiantil, este proyecto cuenta con un excelente equipo de trabajo de dos analista y diseñadores dos desarrolladores y un integrante en el control de base de datos, el cual tiene como objetivo principal el poder brindar un software que sea de agrado y utilidad para los usuarios.

### 1.2.2 Misión

Mejorar notablemente el procesos de asistencia que se está llevando a cabo en la institución SENA, haciéndolo rápido, eficaz y al alcance de todos los usuarios involucrados en este proceso. Disminuyendo así el gasto de recursos físicos y presencia de suplantación o posibles perdida de documentos.

### 1.2.3 Visión

Extensión del sistema en un periodo de tres (3) años a todas las instituciones SENA, tanto nacional como internacionalmente. Poder abarcar otros campos en el control de asistencia como lo son los institutos, instituciones y universidades que lo deseen implementar.

### 1.2.4 Objetivos Program – Art

#### General.

Desarrollar un aplicativo web, que permita a los instructores realizar un control de asistencia sobre los aprendices en el Sena.

#### Específicos.

* Analizar las dificultades que se presentan a la hora de llevar el control de asistencia en los aprendices al momento de iniciar la formación y de las tecnologías viables para diseñar la solución.
* Diseñar una aplicación web que se acople a las necesidades del usuario y brinde una solución eficaz de acuerdo a las dificultades analizadas anteriormente y en función de las tecnologías existentes en el mercado escogidas para dicha solución.
* Desarrollar la aplicación web de acuerdo con el diseño elaborado.
* Implementar la aplicación web en el Sena.
* Evaluar la satisfacción de los instructores, aprendices y directivos del Sena, en cuanto al funcionamiento del aplicativo web implementado.

### 1.2.5 Proceso de desarrollo del software

Program-Art se desarrolla bajo la opinión y calificación del usuario en todo momento, además cuenta con la opinión y aporte de excelentes ingenieros y especializados en el área en el cual se está llevando a cabo la fase de desarrollo del sistema.

Durante todas las fases por las que ha avanzado el sistema se ha evaluado la mejor posibilidad de desarrollo, tomando en cuenta el leguaje, modelo y plataforma en la cual se desempeñaría de una mejor manera, para así poder brindar un servicio el cual se describa con toda seguridad como fiable, robusto, con estándares internacionales y lo más importante de excelente calidad.

## 1.3 Objetivo

El manual de calidad **Program-Art**, describe las disposiciones generales para asegurar la eficacia del servicio brindado por el equipo de trabajo, con el fin de evitar la inconformidad, en caso de aparición se tomaran las medidas necesarias para evitar repetición alguna y con la buena gestión de los procesos mejorar el sistema así como la satisfacción de los usuarios.

Para ello, **Program-Art**, adopta como referencia la norma ISO 9001/2008 y a otras políticas del equipo de trabajo.

Solo se podrán alcanzar los objetivos de calidad si el sistema de control lo facilita, para esto se necesita llevar a cabo una planificación, la cual se logra por medio de revisiones frecuentes al sistema y también al equipo de trabajo. Para verificar que se estén brindando los recursos necesarios en el desarrollo de las tareas y que se esté haciendo de forma satisfactoria. Estas revisiones están basadas en, la necesidad de cambios que surgen a medida que avanza el sistema, aparición de algún problema o se necesite llevar a cabo cambios planificados. Por lo mismo se mantiene en continua implementación el sistema de control.

### 1.3.1 Políticas de calidad del equipo de trabajo

Entre las políticas del equipo de trabajo podemos encontrar la elaboración de un producto que cumpla con las siguientes normas:

1.3.1.1 Funcionalidad

* Adecuación: Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.
* Exactitud: Capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.
* Interoperabilidad: Capacidad del producto software para interactuar con uno o más sistemas especificados.
* Seguridad de acceso: Capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados
* Cumplimiento funcional: Capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad.

1.3.1.2 Fiabilidad

* Madurez: Capacidad del producto software para evitar fallar como resultado de fallos en el software.
* Tolerancia a fallos: Capacidad del software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados.
* Capacidad de recuperación: Capacidad del producto software para reestablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo.
* Cumplimiento de la fiabilidad: Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con la fiabilidad.

1.3.1.3 Usabilidad

* Capacidad para ser entendido: Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.
* Capacidad para ser aprendido: Capacidad del producto software que permite al usuario aprender sobre su aplicación.
* Capacidad para ser operado: Capacidad del producto software que permite al usuario operarlo y controlarlo.
* Capacidad de atracción: Capacidad del producto software para ser atractivo al usuario.
* Cumplimiento de la usabilidad: Capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

1.3.1.4 Eficiencia

* Comportamiento temporal: Capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas.
* Utilización de recursos: Capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.
* Cumplimiento de la eficiencia: Capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.

1.3.1.5 Mantenimiento

* Capacidad para ser analizado: Es la capacidad del producto software para serle diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas.
* Capacidad para ser cambiado: Capacidad del producto software que permite que una determinada modificación sea implementada.
* Estabilidad: Capacidad del producto software para evitar efectos inesperados debidos a modificaciones del software.
* Capacidad para ser probado: Capacidad del producto software que permite que el software modificado sea validado.
* Cumplimiento de la mantenibilidad: Capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.

#### 1.3.1.6 Portabilidad

* Adaptabilidad: Capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado.
* Instalabilidad: Capacidad del producto software para ser instalado en un entorno especificado.
* Coexistencia: Capacidad del producto software para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes.
* Capacidad para reemplazar: Capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno.
* Cumplimiento de la portabilidad: Capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad.

## 1.4 Alcance

El alcance del presente manual están limitados al éxito o fracaso que presente el sistema Program-Art una vez se encuentre en funcionalidad en el ámbito al que compete.

## 1.5 Exclusiones

El manual de calidad Program-Art, está basado en la estructura de las normas ISO 9001/2008, ISO 9000/2000 e ISO 25000. Tomando como principales, los siguientes puntos: Términos y definiciones (\*), Sistema de gestión de la calidad (\*), Requisitos de la documentación (\*), Responsabilidad de la dirección (\*), Gestión de los recursos (\*), Realización del producto (\*), Autoevaluación (\*), Mejora continua (\*), Principales normas y estándares de la Calidad del Producto Software (\*), entre otros.

## 1.6 Definiciones

### 1.6.1 Términos ISO 9000/2000

A través del texto de este Manual se emplean términos cuyo significado es específico para el Sistema de Gestión de Calidad.

**Cliente:** organización o persona que recibe un producto (ISO 9000:2000).

**Producto:** resultado de un proceso (ISO 9000:2000).

NOTA 1: Existen cuatro categorías genéricas de productos:

* Servicios (por ejemplo, transporte);
* Software (por ejemplo, programas de computadora, diccionario);
* Hardware (por ejemplo, parte mecánica de un motor);
* Materiales procesados (por ejemplo, lubricante).

La mayoría de los productos contienen elementos que pertenecen a diferentes categorías genéricas de producto. La denominación del producto en cada caso como servicio, software, hardware o material procesado depende del elemento dominante.

NOTA 2: Cuando en el Manual de Calidad se utiliza el término “producto”, éste puede significar también “servicio”.

**Dependencia:** Los diferentes entes pertenecientes a la empresa.

**Proveedor:** organización o persona que proporciona un producto (ISO 9000:2000).

### 1.6.2 Términos Program-Art

A continuación se pasa a definir conceptos presentes en el manual.

**Funcionalidad:** capacidad del software para proporcionar funciones que satisfacen las necesidades fijadas y solicitadas, cuando el software se usa bajo condiciones especificadas.

**Fiabilidad:** capacidad del software para mantener su nivel de funcionamiento, cuando se usa en condiciones específicas.

**Usabilidad:** capacidad del software para ser entendido, aprendido, usado y aceptado por el usuario, bajo condiciones especificadas. Esto supone poner al usuario en el centro del proceso de creación del producto software, así como permitir que se pueda operar con él de manera natural y con sencillez.

**Eficiencia:** capacidad del software para proporcionar el funcionamiento requerido, relativo a la cantidad de recursos usados, bajo condiciones establecidas.

**Mantenimiento:** capacidad del software para ser modificado. Dichos cambios o modificaciones pueden incluir correcciones, mejoras, ampliaciones o adaptaciones al entorno, tanto en los requisitos como en las especificaciones funcionales del software.

**Inter-operatividad:** capacidad del software para la comunicación y utilización de la información recibida entre dos productos que se están ejecutando en diferentes equipos.

**Seguridad:** capacidad del software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que se permite el acceso a las personas o sistemas autorizados.

**Tolerancia a fallos:** capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.

**Capacidad de recuperación:** capacidad del software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.

**Fácil comprensión:** capacidad del software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.

**Fácil aprendizaje:** capacidad del producto software para permitir al usuario aprender su funcionamiento.

**Operatividad:** capacidad del software para permitir que el usuario pueda operar con él, así como que pueda controlarlo fácilmente.

**Software atractivo:** capacidad del producto software desarrollado para atraer al cliente, acomodándose a aquello que busca o desea.

**Uso de recursos:** capacidad del software para utilizar gran cantidad y tipos de recursos apropiados, cuando el software realiza su función bajo determinadas condiciones.

**Capacidad para cambios:** capacidad del software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación.

**Facilidad para pruebas:** capacidad que presenta el producto software para permitir validar las partes modificadas.

**Facilidad de instalación:** capacidad del software que le permite ser instalado de manera simple por cualquier usuario.

**Facilidad de reemplazo:** capacidad del producto para poder ser retirado o desinstalado de forma sencilla y sin efectos colaterales adversos.

**Versión:** Modificación crítica en un sistema ya liberado.

**Liberación:** Lanzar el sistema.

**Variante del software:** Modificación (pequeña) al sistema después de ser liberado.

**EC aprobado:** EC autorizado por la administración del proyecto.

**Artefacto:** producto final o intermedio (como documentos, código fuente, etc.)

**Archivo Maestro:** archivo específico designado para el proyecto XX.

### 1.6.3 Acrónimos

EC: Elemento de Configuración, un elemento controlado por el sistema de configuración.

AC: Administración de la Configuración, el proceso de conservar las versiones relevantes del proyecto.

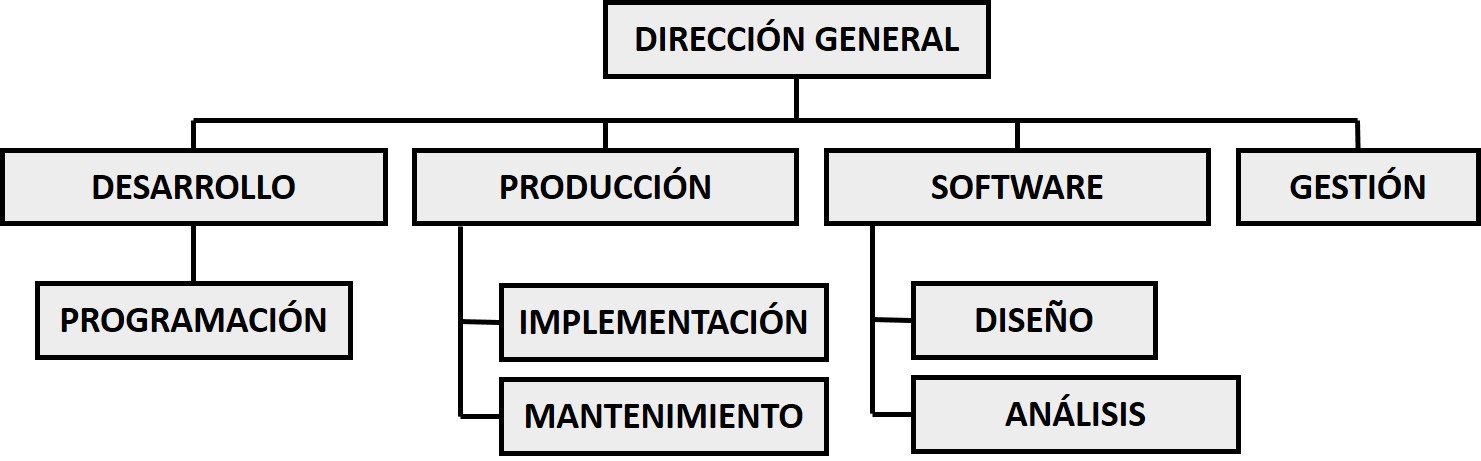
PACS: Plan de Administración de la Configuración del Software.

# 2. RESPONSABILIDAD Y AUTORIDAD.

## 

## 2.1 Estructura departamental

El equipo de trabajo está formado por departamentos, cuya relación es la que sigue:



2.1.1 El Departamento de Desarrollo: se encargará de convertir a código lo que se realiza en el departamento de software.

2.1.2 El Departamento de Producción: se encarga de la entrega del producto inicial un producto que necesita manutención, en este departamento se llevan a cabo dos tareas muy importantes. Como lo son la implementación y el mantenimiento, el objeto de la primera es identificar posibles mejoras que se requieran para aumentar la satisfacción de los usuarios y el objeto de la segunda es efectuar estas mejoras, modificando también la parte de desarrollo y software.

2.1.3 El Departamento de Software: se encargará de convertir en un sistema informático adecuado, la idea inicial del producto a realizar, encargándose de toda su funcionalidad de acuerdo a los requisitos de los usuarios.

2.1.4 El Departamento gestión: se encarga de definir los aspectos económicos implicados en el desarrollo del sistema, la reutilización de información, la compra de materiales, utilización de recursos, etc.

Se guardarán registros de todos los avances que ha tenido el sistema al pasar por cada una de las etapas en caso de que se necesite reutilizar de algún tipo de recurso. También se necesitaran las condiciones en las que el sistema ha presentado un estado crítico para poder evaluarlo y compararlo, mejorando así la versión actual sin que presente los mismos errores de la versión pasada.

También se encargara de los detalles del personal implicado de una u otra forma en el desarrollo del sistema para para mejorar la distribución y utilización.

Todos los documentos relacionados en este manual deberán ser especificados de una forma clara, indicando la fecha y ordenándolos de una forma cronológica.

## 2.2 Responsabilidad y autoridad

Uno de los rasgos inherentes de la calidad es que compete a todos los niveles de la empresa y, por lo tanto, todos y cada uno de sus integrantes son responsables de la misma en mayor o menor medida.

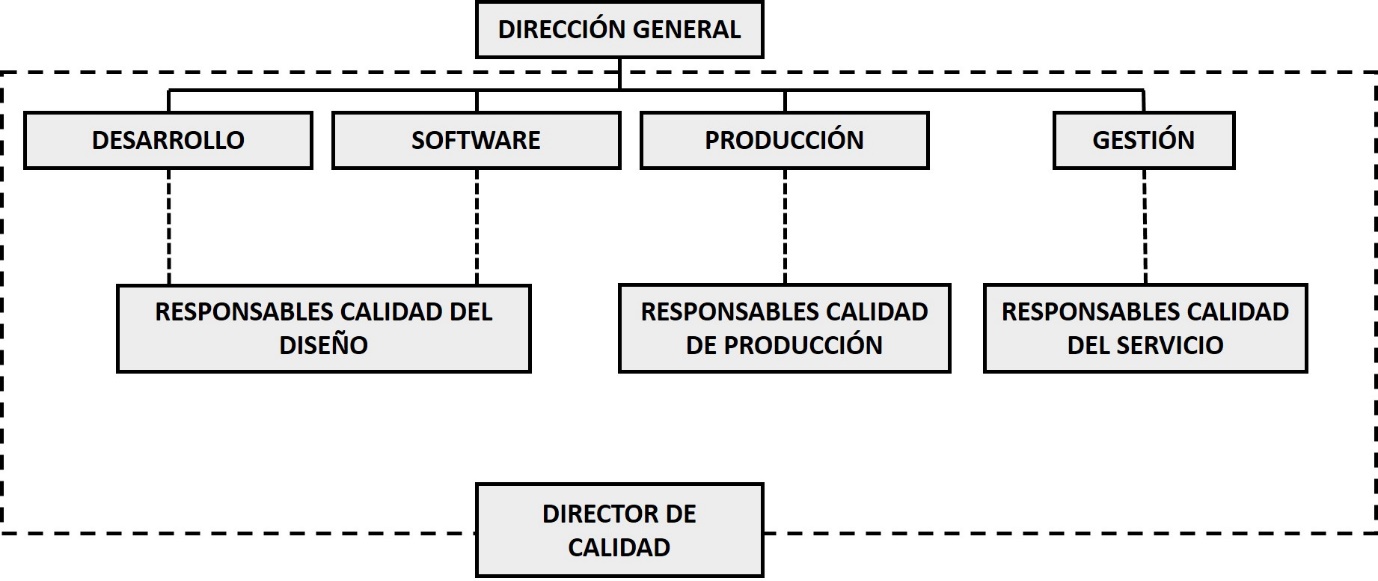
En relación con la calidad, el Director General tiene la responsabilidad final sobre:

* La calidad del producto cuando está ya en el mercado.
* Asegurar que la política de la empresa en relación con la calidad y seguridad es bien entendida por todo el personal.
* Comprobar que la política de calidad está totalmente implementada, realizando controles efectivos del sistema.
* Asegurar que todos los recursos necesarios están disponibles para la implementación del sistema de seguridad.

A continuación se pasan a detallar las características de los principales responsables de impulsar la calidad y garantizar que todo el mundo trabaje orientado a ella. Cuatro son los encargados de esta tarea:

* El director de calidad.
* El responsable de calidad del diseño.
* El responsable de calidad de producción.
* El responsable de calidad de servicios.

Cada uno ellos están relacionado con un área concreta de la empresa, como se puede observar en el siguiente organigrama:



### 2.2.1 Director de calidad

El director de calidad debe ser una persona de gran nivel profesional, con bastante experiencia en la realización de procesos de evaluación de calidad, con amplios conocimientos de las normas existentes y una visión de alto nivel clara, del trabajo que realiza la empresa.

Sus responsabilidades son:

* Identificar y establecer sistemas y procedimientos adecuados para:
* La adquisición y mantenimiento de instalaciones y equipos.
* La contratación de profesionales capacitados.
* El mejor desarrollo posible de la actividad de la empresa.
* Velar por el cumplimiento de los principios de buenas prácticas y procedimientos de trabajo, incluyendo el establecimiento de programas de control y garantía de calidad y seguridad.
* Fijar políticas institucionales.
* Desarrollo de presupuestos para la implantación y mantenimiento de las políticas de calidad dentro de la empresa, así como controlar y evaluar su cumplimiento.
* Mantener, desarrollar, controlar e informar sobre los elementos de calidad de la empresa, tal como especifica la norma ISO 9001, así como cualquier otra directiva de otras empresas o entidades relacionadas con la calidad.
* Tomar decisiones en el lanzamiento del producto software. Esto incluye el aseguramiento de que se toma la decisión adecuada en relación con los productos no conformes con las normas de calidad de la empresa.
* Asegurar que la empresa tiene un adecuado conocimiento del desarrollo del producto en relación con la seguridad, calidad y legislación vigente.
* La aprobación y supervisión de todos los materiales suministrados por los proveedores.
* Coordinar la investigación sobre los productos no conformes con la calidad y supervisar la eficacia de las acciones correctivas.
* Control de la documentación de calidad, incluyendo este manual.

### 2.2.2 Responsable de calidad del diseño

Debe tener formación profesional y experiencia práctica abundante en el proceso de desarrollo de productos software.

Sus responsabilidades son:

* Asegurar que su departamento emplea adecuadamente las herramientas, técnicas y métodos analíticos adecuados, así como los procedimientos previamente aprobados y verificados relativos al aseguramiento de la calidad.
* Establecer los procedimientos adecuados para asegurar el control de la calidad de las tareas, y establecer las acciones correctivas pertinentes.
* Trasladar los requisitos de los clientes a especificaciones de diseño, y comprobar que los productos desarrollados satisfacen tanto las especificaciones de diseño como los requisitos de los clientes.
* Velar por el cumplimiento de los principios de buenas prácticas y procedimientos de trabajo, incluyendo el establecimiento de programas de control y garantía de calidad y seguridad.
* Motivar al personal en la aplicación de los principios y procedimientos de calidad, y en el cumplimiento de las acciones del programa de control de calidad.
* Detectar las necesidades de capacitación y formación del personal y coordinar planes con los responsables a tal efecto.
* Mantener relaciones de coordinación y colaboración con otras empresas o entidades relacionadas con la calidad.

### 2.2.3 Responsable de calidad de producción

Debe tener formación profesional y experiencia práctica abundante en el proceso de desarrollo de productos software.

Sus responsabilidades son:

* Asegurar que su departamento emplea adecuadamente las herramientas, técnicas y métodos analíticos adecuados, así como los procedimientos previamente aprobados y verificados relativos al aseguramiento de la calidad.
* Establecer los procedimientos adecuados para asegurar el control de la calidad de las tareas, y establecer las acciones correctivas pertinentes.
* Velar por el cumplimiento de los principios de buenas prácticas y procedimientos de trabajo, incluyendo el establecimiento de programas de control y garantía de calidad y seguridad.
* Motivar al personal en la aplicación de los principios y procedimientos de calidad, y el cumplimiento de las acciones del programa de control de calidad.
* Detectar las necesidades de capacitación y formación del personal y coordinar planes con los responsables a tal efecto.
* Mantener relaciones de coordinación y colaboración con otras empresas o entidades relacionadas con la calidad.
* Desarrollo de presupuestos para la implantación y mantenimiento de las políticas de calidad dentro de la empresa, así como controlar y evaluar su cumplimiento.

### 2.2.4 Responsable de calidad de servicios

Debe tener formación profesional y experiencia práctica abundante en la atención al cliente, soporte y gestión de incidencias.

Sus responsabilidades son:

* Asegurar que sus departamentos emplean adecuadamente las herramientas, técnicas y métodos analíticos adecuados, así como los procedimientos previamente aprobados y verificados relativos al aseguramiento de la calidad.
* Establecer los procedimientos adecuados para asegurar el control de la calidad de las tareas, y establecer las acciones correctivas pertinentes.
* Velar por el cumplimiento de los principios de buenas prácticas y procedimientos de trabajo, incluyendo el establecimiento de programas de control y garantía de calidad y seguridad.
* Motivar al personal en la aplicación de los principios y procedimientos de calidad, y el cumplimiento de las acciones del programa de control de calidad.
* Detectar las necesidades de capacitación y formación del personal y coordinar planes con los responsables a tal efecto.
* Mantener relaciones de coordinación y colaboración con otras empresas o entidades relacionadas con la calidad.
* Desarrollo de presupuestos para la implantación y mantenimiento de las políticas de calidad dentro de la empresa, así como controlar y evaluar su cumplimiento.
* Asegurar que las órdenes del cliente se cumplen tan rápido como sea posible y en su totalidad.
* Fomentar la venta de los productos, realizando la oportuna publicidad, promoción y distribución del producto.
* Realizar una adecuada revisión del tipo de cliente más representativo, para establecer unos requisitos adecuados y un alto nivel de satisfacción.
* Comprobar que se establecen canales apropiados de comunicación entre la empresa y el cliente, asegurando así la satisfacción del mismo.

### 2.2.5 Responsabilidades de los Directores de Departamento

Los directores de cada departamento son responsables de:

* Comprobar que todas las actividades de su departamento están adecuadamente documentadas y que siguen los procedimientos de control establecidos por la empresa, que estos documentos están orientados a la calidad y que son revisados y mantenidos de forma actualizada.
* Asegurar que todos los procedimientos documentados, planificados y programados son realizados tal y como ha especificado la empresa en este manual de calidad y que todos estos documentos están guardados y custodiados de forma segura.
* Establecer cursos de formación y asegurar que todos los empleados son adecuadamente entrenados para las tareas que han de llevar a cabo, así como mantener los registros de todos los cursos de formación interna realizados en la empresa.
* Comprobar que los procesos de su departamento son evaluados periódicamente en relación con la calidad y con la satisfacción de los clientes, y realizar las acciones correctivas oportunas para evitar productos software no conformes con el sistema de calidad establecido por la empresa.

## 2.3 Matriz de responsabilidades

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dir. calidad | Dir. General. | Dir. Gestión. | Dir. Producción. |
|  |  |  |  |  |
| Sistema de gestión de calidad. |  |  |  |  |
| Documentación del sistema de calidad | 2 | 1 | NA | NA |
| Control de los documentos. | 2 | 1 | NA | NA |
| Control de los registros. | 1 | NA | NA | 2 |
| Responsabilidad de la dirección |  |  |  |  |
| Política de calidad | 2 | 1 | 1 | 1 |
| Planificación del sistema de calidad | 2 | 2 | 2 | 1 |
| Revisión por la dirección | 1 | 2 | NA | NA |
| Gestión de los recursos |  |  |  |  |
| Provisión de los recursos | NA | 1 | 2 | NA |
| Recursos humanos | NA | 1 | 2 | NA |
| Infraestructura y ambiente de trabajo | 1 | 2 | 1 | NA |
| Realización del producto |  |  |  |  |
| Planificación de la realización del producto | 1 | 2 | 1 | 2 |
| Procesos relacionados con los clientes |  |  |  |  |
| Requisitos relacionados con el producto | 2 | 1 | NA | 2 |
| Comunicación con los clientes | 2 | 2 | NA | 2 |
| Diseño y desarrollo | 1 | 1 | NA | 2 |
| Producción y prestación del servicio |  |  |  |  |
| Control y validación de la producción y la prestación del servicio | 1 | 2 | NA | 2 |
| Identificación y trazabilidad | 1 | 1 | 1 | 2 |
| Propiedad del cliente (P.R.) | NA | 2 | 2 | 1 |
| Medición y seguimiento |  |  |  |  |
| Satisfacción del cliente | 2 | 2 | NA | 2 |
| Auditoria interna | NA | 2 | NA | NA |
| Seguimiento y medición de los procesos | 1 | 1 | 2 | NA |
| Seguimiento y medición del producto | 1 | 2 | NA | 2 |
| Control del producto no conforme (P. R.) | 1 | 2 | NA | 2 |
| Análisis de datos | NA | NA | NA | 2 |
| Mejora continua | 2 | 1 | 1 | 2 |
| Acciones correctivas | 2 | NA | 1 | 2 |
| Acciones preventivas | 2 | 2 | 1 | NA |

# 3. SISTEMA DE GESTIÓN DE CALIDAD

## 3.1 Requisitos Generales

En el equipo de trabajo **Program-Art**, las actividades llevadas a cabo se hacen de forma planeada y sistemática para que se cumplan las políticas, los objetivos de calidad y algunos requisitos mencionados a continuación.

* El constate uso y actualización del control que se tiene en la calidad.
* El constante cambio de las técnicas de producción, inspección y servicio.
* La implementación de las verificaciones adecuadas en las etapas de realización del producto.

### 3.1.1 Objetivo Del Plan De Calidad

Verificar que el personal encargado del control de calidad y operación, realice con claridad, las inspecciones y pruebas que se plantearon y acordaron con los usuarios. Para así poder garantizar la efectividad del producto fabricado.

### 3.1.2 Estructura Documental Del Sistema De Calidad

La estructura del sistema de aseguramiento de la calidad de Program-art se ilustra a continuación.



## 

## 3.2 Requisitos De La Documentación

### 

### 3.2.1 Generalidades

En la documentación del sistema de calidad de Program-Art se incluye:

Documentación de una política de calidad, objetivos de calidad, manual de calidad, documentación de los procedimientos utilizados de la norma ISO 9001:2008, procedimientos e instructivos necesarios por el equipo de trabajo para asegurar la operación y control de los procesos, los registros requeridos por la norma y por el equipo de trabajo Program-Art.

### 

### 3.2.2 Manual De Calidad

El presente manual de calidad incluye:

El alcance del sistema de gestión de calidad, incluyendo los detalles y la justificación de cualquier exclusión como esta descrito en la sección de objeto y campo de aplicación de este manual, las referencias de los procedimientos documentados establecidos para el sistema de gestión de calidad.

### 3.2.3 Control De Los Documentos

En el equipo de trabajo existe un procedimiento para el control de los documentos, este se realiza para asegurarse de que se revisen y se aprueben cuando se necesita, y una vez que se identifican los cambios ser aprobados nuevamente, se incluye el estado de la revisión actual de los documentos y las revisiones anteriores, los documentos que se encuentran legibles, entendibles y reutilizables.

Existe un procedimiento que define los controles necesarios para la identificación, el almacenamiento, la protección, la recuperación, el tiempo de repetición y la disposición de los registros.

### 3.2.4 Control De Los Registros

Los registros son establecidos y mantenidos para proporcionar evidencias de la conformidad con los requisitos, así como de la operación que se lleva a cabo para aumentar esa satisfacción.

Existe un procedimiento documentado del control de registros de calidad.

# 4. GESTIÓN DE RECURSOS

## 

## 4.1 Consideraciones financieras

Para cada actividad de desarrollo del software, la programación inicial de una tarea debe estar basada en el esfuerzo requerido y en los recursos disponibles. Si se requiere, la programación puede reducirse de dos maneras:

* Modificando la manera en que las tareas son ejecutadas, para reducir el esfuerzo
* Modificando los recursos disponibles para tener un mejor producto.

El equipo de trabajo pero primordialmente el Director del Departamento de Software y el de Producción tienen como reto principal proponer programaciones que tengan en cuenta:

* Actividades secuenciales que pueden ser superpuestas.
* Prioridad de una actividad sobre otra.
* Limitación en los recursos:
* Equipos.
* Personas.
* Actividades duplicadas, que pueden concentrarse eficazmente.
* Actividades innecesarias que se pueden eliminar sin afectar a la calidad final del producto.
* Requisitos que no se necesitan de inmediato, o que ha sido exagerada su necesidad.

Los planes de desarrollo del producto del equipo de trabajo deberían basarse en las mejores estimaciones y en la experiencia.

El coste y las programaciones están inversamente relacionados. Reducir la duración añadiendo recursos, incrementa el coste; reducir el coste aplazando algunas actividades, incrementa la duración.

## 

## 4.2 Recursos humanos

Para poder obtener un producto de excelente calidad se necesita disponer de un equipo de trabajo con los conocimientos y habilidades adecuadas, aunque no muchos toman en cuenta la cantidad de personal que se requiere es igual mente importante.

## 

## 4.3 Recursos materiales

A la hora de hacer el debido registro de los recursos materiales se debe tomar en cuenta desde lo más mínimo hasta lo más importante como lo son: edificios, equipos, impresoras, fotocopiadoras, electricidad, información, transporte, dispositivos para la comunicación, etc.

Se debe planificar con mucho cuidado el coste de cada uno de los productos ya que estos son muy elevados. En los riesgos de valoración del proceso, se pueden incluir los recursos materiales como uno de los aspectos más importantes.

# 5. REALIZACIÓN DEL PRODUCTO

## 

## 5.1 Planeación de la Realización del Producto

En este punto del manual se planificaran y desarrollaran las necesidades que se tienen en los procesos a la hora de realizar el producto.

En el plan de desarrollo del software se determinan los requisitos específicos para el producto, incluyendo referencias o definiciones de:

* Procedimientos y herramientas estándar aplicables.
* Procedimientos o herramientas nuevos que se emplean para ejecutar el plan de desarrollo, entre los que se incluye cualquier programa de formación para adquirir las destrezas necesarias.

Para la realización del producto nos basamos en el modelo de desarrollo **Espiral.**

### 5.1.1 Modelo en espiral

Este es un modelo de proceso de software evolutivo, el cual enlaza la naturaleza iterativa de la construcción de prototipos, pero conservando aquellas propiedades del modelo en cascada.

El modelo en espiral fue desarrollado por Boehm, quien lo describe así:

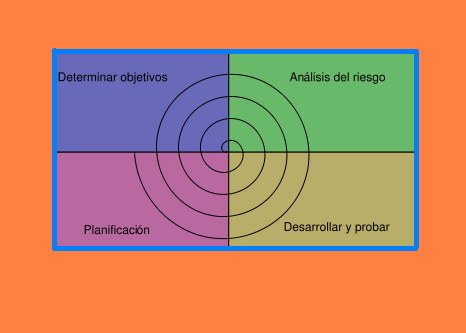
El modelo de desarrollo en espiral es un generador de modelo de proceso guiado por el riesgo que se emplea para conducir sistemas intensivos de ingeniería de software concurrente y a la vez con muchos usuarios.

#### 5.1.1.1 Características

* Un enfoque cíclico para el crecimiento incremental del grado de definición e implementación de un sistema, mientras que disminuye su grado de riesgo.
* Un conjunto de puntos de fijación para asegurar el compromiso del usuario con soluciones de sistema que sean factibles y mutuamente satisfactorias.

El modelo espiral no es una alternativa del modelo cascada, ellos son completamente compatibles.

#### 5.1.1.2 Funcionamiento del modelo Espiral.

****

#### 5.1.1.3 En cada vuelta tomamos en cuenta

* **Los Objetivos:** Que necesidad debe envolver el programa.
* **Alternativas:** Los varios métodos de alcanzar los objetivos de manera exitosa, a través de diferentes puntos como son:
* Características: experiencia del personal, exigencias a efectuar.
* Formas de gestión del programa.
* Riesgo tomado con cada alternativa.
* **Desarrollar y Verificar:** Programar y probar el programa.
* **Se planificaran** los siguientes pasos y se volverá a empezar la espiral. La espiral tiene una forma de caracola y se dice que mantiene dos dimensiones la radial y la angular:
* Angular=Avance del proyecto Software, dentro de un ciclo.
* Radial=Aumento del coste del proyecto, ya que con cada nueva iteración se pasa más tiempo desarrollando.

Este sistema es muy utilizado en proyectos largos como pueden ser la creación de un Sistema Operativo. Y que necesitan constantes cambios.

Al ser un modelo de Ciclo de Vida orientado al riesgo se dice que uno de los aspectos fundamentales de su éxito radica en que el equipo que lo aplique sea capaz de detectar y catalogar correctamente dicho riesgo.

## 5.2 Procesos Relacionados con el Cliente

Como prioridad se tienen que tener los requisitos que influyen directamente en la realización del producto:

* Los requisitos especificados por los usuario
* El conocimiento anterior adquirido en la investigación de productos similares.
* Los estatutos, códigos y regulaciones vigentes que se deben aplicar
* El usuario.

En el equipo de trabajo hay una persona responsable de proporcionar y satisfacer el conjunto de requisitos relacionados directamente con el producto.

Para poder lograr la mayor satisfacción del cliente se debe determinar cómo lo más importante la comunicación con el mismo.

## 5.3 Diseño y Desarrollo

En esta fase se lleva a cabo: el proceso que trasforma los requisitos funcionales y no funcionales, en características específicas o en especificación de un producto, proceso o sistema.

Las interfaces se deben establecer para así garantizar la comunicación del departamento de software y el departamento de producción con los demás departamentos del equipo de trabajo, para así conocer a la perfección los responsables de cada tarea que se lleva a cabo.

Las etapas de las que consta la fase de diseño son: Análisis de requisitos, Diseño, Implementación y Validación, Pruebas e integración. El equipo de trabajo asegurara que no se omiten las actividades de revisión, verificación y validación de cada una de estas etapas.

Las revisiones en esta fase tienen un doble objetivo:

* Evaluar la capacidad que se tiene a la hora de satisfacer los requisitos.
* Identificar problemas y proponer acciones para resolverlos.

La revisión más específicamente se supone para la validación de la calidad del producto, a través de esta podemos confirmar que los requisitos han sido satisfechos. Se puede llevar a cabo a través de revisiones técnicas, inspecciones, pruebas de unidad, integración y prueba de la integración.

Por último, la validación se centra en aquellos requisitos relacionados directamente con el sistema o el uso que se quiere hacer del producto. Esto se realiza a atreves de prueba de defectos, pruebas de esfuerzo, pruebas de interfaces.

## 5.4 Compras

### 5.4.1 Proceso de compras

El equipo de trabajo Program-Art, llevo a cabo la compra de un lector de código de barras por medio de una petición del departamento de gestión, este departamento se encargó de que el producto cumpliera con las especificaciones y requisitos necesarios para la funcionalidad dentro del sistema.

### 5.4.2 beneficio para el sistema

El sistema Program-Art, anteriormente mencionado basa su funcionalidad en el control de asistencia de los aprendices de la ficha 683019, por medio de un lector de código de barras. Al realizar la compra de este dispositivo se resuelve uno de los problemas principales, disminuyendo así el trabajo de los instructores en el proceso de registro de asistencia y mejorando la calidad del sistema en cuanto a efectividad e implementación de tecnologías en la sistematización de procesos cotidianos.

## 5.5 Producción y Prestación del Servicio

En las actividades que se llevan a cabo en esta fase se define como principales responsables los departamentos de producción y de gestión.

* Documentación y duplicación del software (realización de copias para después ser distribuidas).
* Creación, validación y distribución de correcciones y actualizaciones.
* Servicios profesionales: personalización, integración con otros productos.
* Registro de asesorías (Program-Art, lleva un registro de las consultas y/o opiniones del usuario respecto al sistema para asegurar su mejora continua).
* Propiedad del cliente (una ves de concluya la menor cantidad de errores en el sistema se puede determinar como un producto completo lo que equivaldría a que el usuario tendrá total control de propiedad en el producto).

## 5.6 Control de los Dispositivos de Monitoreo y Medición

Para verificar la capacidad de las herramientas de prueba para detectar defectos, se realizan revisiones sistemáticas de los planes de prueba, casos de prueba y programas de prueba, y se actualizan los conjuntos de prueba cuando se detectan nuevos tipos de defectos en el producto.

Se requiere de un procedimiento documentado que defina los controles y responsabilidades implementados por el equipo de trabajo, para asegurar que el producto que no sea conforme a los requisitos, sea identificado y controlado para evitar un uso o distribución no intencionado.

Si se encuentra un producto que no es conforme a los requisitos, la empresa puede:

* Corregir el problema: el producto corregido será verificado para comprobar que, después de la corrección, se ajusta a los requisitos.
* Usarlo a pesar de la disconformidad, bajo una concesión aprobada por el Director General.
* Tomar las medidas oportunas para no permitir que se intente usar.

### 5.6.1 Fase de Monitoreo y medición

Esta fase requiere que el equipo de trabajo establezca procesos para asegurar que la supervisión y medida del producto pueden llevarse a cabo de forma consistente con los requisitos establecidos, que son aquellos que se necesitan para satisfacer:

* Las necesidades internas relacionadas con el proceso y el producto.
* Los requisitos establecidos y reguladores del equipo de trabajo.
* Los requisitos de la norma ISO 9001.

Los pasos necesarios para realizar con éxito esta fase son:

* Verificar a intervalos específicos o después de usar el programa. Se realiza comparando los resultados con medidas estándares.
* Ajustar y reajustar si es necesario.
* Evitar ajustes que invalidarían los resultados de la medida.
* Evitar daños y deterioros al manejar, mantener y almacenar el producto de software desarrollado.

# 6. ADMINISTRACIÓN DE LA CONFIGURACIÓN

## 

## 6.1 Introducción

En este apartado se manejara la administración del software, como se realizar el almacenamiento de la información de nuestro proyecto, las políticas, estándares y procedimientos que se tendrán en cuenta para dar a luz las diferentes versiones, y liberaciones de nuestros productos.

Para el almacenamiento de la información se utilizó la base de datos SQL-Server versión 2008\_R2.

## 6.2 Definición de versiones

Para la respectiva calificación de los manuales que conforman la documentación del sistema se determinó un estándar fijo, el cual está conformado por cuatro (4) tipo de versiones, las cuales se mencionan a continuación.

* **Versión 0:** Esta es la versión inicial, en la cual se determina los procesos y métodos que se necesitan para desarrollar el manual.
* **Versión 1:** En esta versión el manual se encuentra en desarrollo, lo que quiere decir que no se ha hecho ninguna revisión y se abarca la posibilidad de que no se encuentre completo.
* **Versión 2:** Esta es la versión post final, la cual se determina una vez el manual está terminado pero espera la aprobación de las correcciones finales por instructor correspondiente.
* **Versión 3:** Esta es la versión final, la cual se determina una vez el manual no requiera más cambios y es aprobado por el instructor correspondiente.

El versiona miento se maneja de una manera simple y eficaz indicando el estado del archivo en su versión 0, 1, 2, o 3 para referencia la etapa en la que se encuentra y agregando la cantidad de correcciones al lado derecho.

Por ejemplo: 1.2, esto nos indica que el manual se encuentra en la versión uno (1) y cuenta con dos (2) correcciones.

Cuando el manual supere la cantidad de correcciones necesarias se dará lugar a la siguiente versión.

Por ejemplo: 2.4 esto nos indica que el manual se encuentra en la versión dos (2) y cuenta con cuatro (4) correcciones.

La representación gráfica del versiona miento de los manuales se puede visualizar en el [informe de calidad](INFORME-DE-CALIDADsi.docx) del proyecto.

## 6.3 Autoevaluación

La autoevaluación de una organización es una revisión completa y sistemática de las actividades y resultados de la organización con referencia al sistema de gestión de la calidad o a un modelo de excelencia.

La autoevaluación puede proporcionar una visión global del desempeño de la organización y del grado de madurez del sistema de gestión de la calidad. Asimismo, puede ayudar a identificar las áreas que precisan mejora en la organización y a determinar las prioridades.

Como resultado de la autoevaluación que llevo a cabo Program-Art se tomó la iniciativa de reorganizar las áreas de desarrollo para mejorar la calidad y brindar un aire de nuevas opiniones, las cuales beneficiaran en gran proporción el sistema.

## 6.4 Mejora continua

El objetivo de la mejora continua del sistema de gestión de la calidad es incrementar la probabilidad de aumentar la satisfacción de los clientes y de otras partes interesadas. Las siguientes son acciones destinadas a la mejora:

* análisis y evaluación de la situación existente para identificar áreas para la mejora;
* el establecimiento de los objetivos para la mejora;
* la búsqueda de posibles soluciones para lograr los objetivos;
* la evaluación de dichas soluciones y su selección;
* la implementación de la solución seleccionada;
* la medición, verificación, análisis y evaluación de los resultados de la implementación para determinar que se han alcanzado los objetivos;
* la formalización de los cambios.

Los resultados se revisan, cuando es necesario, para determinar oportunidades adicionales de mejora. De esta manera, la mejora es una actividad continua. La información proveniente de los clientes y otras partes interesadas, las auditorías, y la revisión del sistema de gestión de la calidad pueden, así mismo, utilizarse para identificar oportunidades para la mejora.

## 6.5 Métricas de calidad de software

### 6.5.1 Introducción

A medida que aumenta el uso de la Tecnología de la Información, aumenta también el número de sistemas de computación que son críticos, en los cuales un error puede traer consecuencias graves. La mejora en la calidad del software es un objetivo importante. Una falla en sistemas puede implicar serias consecuencias. Por eso se tiene que aplicar la calidad en cada uno del software a desarrollar; es necesario hacer las mediciones ya que ayudan a aplicar y dar seguimiento a las diferentes formas de evaluar y determinar métricas de calidad para un mejor desempeño en la calidad de software.

### 6.5.2 Objetivo

Analizar y medir el código de la aplicación Program-Art para así producir un software con la mejor calidad posible y de esta forma satisfacer a los usuarios.

### 6.5.3 Conceptos

**Corrección:** Capacidad de los productos software para realizar con exactitud las tareas expresadas en su especificación.

**Confiabilidad: se** refiere a la precisión con la que una aplicación proporciona, sin errores los servicios que se establecieron en las especificaciones originales.

**Eficiencia:** Capacidad del software para hacer buen uso de los recursos que manipula.

**Integridad:** Característica de un sistema de ser capaz de proteger sus diferentes componentes contra los procesos o elementos que no tengan derecho de acceso a los mismos.

**Mantenimiento:** Es el proceso de mejora y optimización del software después de su entrega al usuario final.

**Flexibilidad:** Es el esfuerzo requerido para modificar un programa operativo.

**Portabilidad:** Facilidad con que un sistema software puede ser migrado entre diferentes plataformas hardware o software.

**Reusabilidad:** Capacidad de los productos software para funcionar como bloques básicos de la construcción de diferentes aplicaciones.

**Interoperabilidad:** La interoperabilidad es la condición que permite que sistemas o productos diferentes puedan relacionarse entre sí.

**Usabilidad:** Facilidad de verificación de corrección de un software. Que tan sencillo es la realización de pruebas que garanticen la funcionalidad del sistema.

### 6.5.4 Factores y métricas de la calidad de software

Métrica hace referencia a la medición del software en base a parámetros predeterminados, como puede ser el número de líneas de código de que consta o el volumen de documentación asociada. A veces en vez de hablar de métrica se usa el término “Indicadores” del software. Algunos ingenieros lo usan como sinónimos mientras que otros les atribuyen significados distintos.

Program-Art realizo la respectiva medición del sistema como resultado a esto se tiene la respectiva grafica [anexos.](#_Anexo_#2_1)

### 6.5.5 Análisis comparativo (técnico)

Las métricas fueron identificadas en base a las características del producto, requerimientos, entre otros.

Del análisis se determinan las características técnicas mínimas [anexos.](#_7.3_Anexo_#3)

### 6.5.6 Conclusión

El desarrollo de Software ha traído consigo la necesidad de producir software de Calidad, y para lograrlo se tienen en cuenta numerosos factores entre los que se encuentran las métricas de software, estas métricas son muy importantes ya que esto nos permite evaluar y determinar el desempeño en cada software.

### 6.5.7 Responsables de la Evaluación

Jessica Natalia Pulido León- Desarrolladora de Software

Cindy Jineth Saldaña Gutiérrez – Diseñadora de Software

# 7 ANEXOS

## Anexo #1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Métricas de calidad del software  Factor de cantidad | Corrección | Confiabilidad | Eficiencia | Integridad | Mantenimiento | Flexibilidad | Capacidad de pruebas | Portabilidad | Reusabilidad | Interoperabilidad | Usabilidad |
| Exactitud |  |  | x | x |  |  |  |  |  |  | x |
| Complejidad |  |  | x |  |  |  | x |  |  | x |  |
| Consistencia |  |  | x | x |  |  |  |  |  |  |  |
| Eficiencia de ejecución |  | x | x | x |  |  |  |  |  |  |  |
| Capacidad de expansión | x |  |  |  | x | x |  |  |  |  | x |
| Generalidad | x |  |  |  | x |  |  |  |  |  |  |
| Independencia del hardware |  |  | x |  |  |  | x | x |  |  |  |
| Instrumentación |  |  | x |  |  |  |  |  |  | x |  |
| Modularidad |  |  |  |  | x |  | x |  | x |  |  |
| Operatividad |  |  | x |  |  |  |  |  |  |  | x |
| Seguridad |  | x | x | x |  |  | x |  |  |  |  |
| Auto documentación | x |  |  |  |  |  |  |  |  |  |  |
| Simplicidad |  |  |  |  | x | x | x |  | x | x |  |
| Independencia del sistema |  |  | x |  |  |  |  | x |  |  |  |
| Trazabilidad | x |  |  | x |  |  | x |  |  |  |  |
| Facilidad de información |  | x | x | x |  |  |  |  |  |  | x |

## Anexo #2

|  |  |  |
| --- | --- | --- |
| **CARACTERISTICAS** | **SUBCARACTERISTICAS** | **PUNTAJE** |
| **CALIDAD INTERNA Y EXTERNA** |  | 42.5 |
| ***1.Funcionalidad*** | Provee funciones que satisfacen las necesidades cuando el software se utiliza bajo condiciones especificas | 9 |
| **Adecuación** | Provee un adecuado conjunto de funciones para las tareas y objetivos especificados por el usuario | 2.0 |
| **Exactitud** | Provee los resultados o efectos acordados con un grado necesario de precisión | 2.0 |
| **Interoperabilidad** | Interactúa con uno o más sistemas específicos. | 1.5 |
| **Seguridad** | Protege la información y los datos. | 2.0 |
| **Conformidad de la funcionalidad** | Se adhiere a los estándares, convenciones o regulaciones legales y prescripciones similares referentes a la funcionabilidad. | 1.5 |
| ***2.Fiabilidad*** | Mantiene un nivel especifico de funcionamiento | 8 |
| **Madurez** | Evita fallas como resultado de errores en el software. | 2.0 |
| **Tolerancia a errores** | Mantiene un nivel especificado de funcionamiento en casos de errores del software o de incumplimiento de su interfaz. | 2.0 |
| **Recuperabilidad** | Restablece un nivel de funcionamiento y recupera los datos afectados en caso de una falla. | 2.0 |
| **Conformidad de la fiabilidad** | Se adhiere a las normas, convenciones o regulaciones relativas de la fiabilidad | 2.0 |
| ***3.Usabilidad*** | Es capaz de ser entendido, aprendido, usado y atractivo al usuario. | 8.5 |
| **Entendimiento** | Permite entender al usuario si es software es adecuado y como puede ser utilizado para las tareas y condiciones de la aplicación. | 2.0 |
| **Aprendizaje** | Permite al usuario aprender su aplicación. | 2.0 |
| **Operabilidad** | Permite al usuario operarlo y controlarlo. | 1.5 |
| **Atracción** | Es atractivo para el usuario | 1.5 |
| **Conformidad de uso** | Se adhiere a los estándares, guías de estilo o regulaciones relacionadas a su usabilidad. | 1.5 |
| ***4.Eficiencia*** | Provee un desempeño adecuado, de acuerdo a la cantidad de recursos utilizados y bajo las condiciones planteadas. | 5 |
| **Comportamiento de tiempos** | Provee tiempos adecuados de respuesta y procesamiento. | 2.5 |
| **Conformidad de eficiencia** | Se adhiere a los estándares o convenciones referentes a la eficiencia. | 2.5 |
| ***5.Capacidad de mantenimiento*** | Permite ser modificado ( correcciones, mejoras, adaptaciones, cambios en el entorno y especificaciones de requerimientos, entre otros) | 7.5 |
| **Capacidad de ser analizado** | Se le pueden realizar diagnósticos de deficiencias o causas de fallas. | 1.0 |
| **Cambiabilidad** | Permite que una determinada modificación sea implementada. | 1.5 |
| **Estabilidad** | Evita efectos inesperados debidos a la modificación del software. | 1.5 |
| **Facilidad de prueba** | Permite que las modificaciones sean validas | 1.5 |
| **Conformidad de facilidad de mantenimiento** | Se adhiere a los estándares o convenciones relativas a la facilidad de mantenimiento. | 2.0 |
| ***6.Portabilidad*** | Permite ser trasladado de un entorno a otro (entornos organizacionales, hardware y software). | 4.5 |
| **Adaptabilidad** | Se adapta a diferentes entornos sin aplicar acciones o medios diferentes de los previstos para el propósito del software. | 1.5 |
| **Coexistencia** | Coexiste con otros productos de software independientes dentro de un mismo entorno, compartiendo recursos comunes. | 1.5 |
| **Conformidad de portabilidad** | Se adhiere a los estándares o convenciones relacionadas a la portabilidad. | 1.5 |
| **CALIDAD EN USO** |  | 39 |
| ***1.Eficacia*** | Permite a los usuarios lograr las metas con exactitud e integridad. | 10.0 |
| ***2.Productividad*** | Permite a los usuarios emplear cantidades apropiadas de recursos, en relación a la eficacia | 10.0 |
| ***3.Seguridad*** | Logra niveles adaptables de riesgo de daño a las personas, institución, software, entre otros. | 9.0 |
| ***4.Satisfaccion*** | Satisface a los usuarios en un contexto especificado de uso. | 10.0 |
|  | **TOTAL** | 81.5 |

|  |  |
| --- | --- |
| **CARACTERISTICAS** | **PUNTAJE MAXIMO** |
| **CALIDAD INTERNA Y EXTERNA** | 60.0 |
| ***1.Funcionalidad*** | 10.0 |
| **Adecuación** | 2.0 |
| **Exactitud** | 2.0 |
| **Interoperabilidad** | 2.0 |
| **Seguridad** | 2.0 |
| **Conformidad de la funcionalidad** | 2.0 |
| ***2.Fiabilidad*** | 10.0 |
| **Madurez** | 2.5 |
| **Tolerancia a errores** | 2.5 |
| **Recuperabilidad** | 2.5 |
| **Conformidad de la fiabilidad** | 2.5 |
| ***3.Usabilidad*** | 10.0 |
| **Entendimiento** | 2.5 |
| **Aprendizaje** | 2.5 |
| **Operabilidad** | 2.0 |
| **Atracción** | 1.5 |
| **Conformidad de uso** | 1.5 |
| ***4.Eficiencia*** | 10.0 |
| **Comportamiento de tiempos** | 3.0 |
| **Conformidad de eficiencia** | 4.0 |
| ***5.Capacidad de mantenimiento*** | 3.0 |
| **Capacidad de ser analizado** | 10.0 |
| **Cambiabilidad** | 2.0 |
| **Estabilidad** | 2.0 |
| **Facilidad de prueba** | 2.0 |
| **Conformidad de facilidad de mantenimiento** | 2.5 |
| ***6.Portabilidad*** | 10.0 |
| **Adaptabilidad** | 2.0 |
| **Coexistencia** | 2.0 |
| **Conformidad de portabilidad** | 2.0 |
| **CALIDAD EN USO** | 40.0 |
| ***1.Eficacia*** | 10.0 |
| ***2.Productividad*** | 10.0 |
| ***3.Seguridad*** | 10.0 |
| ***4.Satisfaccion*** | 10.0 |

# WEB GRAFÍA

<http://es.slideshare.net/kellypt1/modelos-de-desarrollo-de-software>

<http://repository.lasalle.edu.co/bitstream/handle/10185/2790/11.10%20N558d.pdf?sequence=1>

<http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=198:calidad-del-software-metricas-y-fiabilidad-de-aplicaciones-1a-parte-dv00103a&catid=45:tendencias-programacion&Itemid=164>