ENTREGABLE INFORME

COLORADO ORTIZ DIEGO ARMANDO

MÉNDEZ MORA CRISTHIAN CAMILO

PÁEZ CASTRO HAROLD CAMILO

DEMETRIO MAURICIO ESTUPIÑAN FINO

FICHA: 1803170 G2-G3

SERVICIO NACIONAL DE APRENDIZAJE SENA  
CENTRO DE ELECTRICIDAD, ELECTRÓNICA Y TELECOMUNICACIONES

ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

BOGOTÁ D.C

**Plan de calidad de software**

Proyecto: Fit Gym System

Versión: 1.0

Historial de Revisiones

| **VERSIÓN** | **FECHA** | **AUTOR** | **DESCRIPCIÓN** |
| --- | --- | --- | --- |
| 1.0 | 18 – 06 - 2020 | Diego Colorado  Cristhian Méndez  Harold Páez | En este documento se realizó el plan de calidad de software el cual especifica riesgos, pruebas y métricas de dicho plan. |
| 1.1 | 24-06-2020 | Diego Colorado  Cristhian Méndez  Harold Páez | El informe se evidencia con tres errores en los siguientes puntos:   * Documentación relacionada * Estándares * Auditorias |

**TABLA DE CONTENIDO**

1. Introducción…………………………………………………………………. 5
2. Objetivo………………………………………………………………………5
   1. Objetivo de SQA…………………………………………………………5
3. Documentos relacionados……………………………………………………6
4. Destinarios…………………………………………………………………...7
5. Administración – Planeación………………………………………………...7
   1. Organización………………………………………………………..........7
   2. Responsabilidades………………………………………………………..7
   3. Cronograma del proyecto……………………………………………….. 8
   4. Riesgos del proyecto……………………………………………………. 8
6. Estándares, Practicas, Convenciones y Mediciones…………………………8
   1. Estándares……………………………………………………….……….9
7. Métricas de Calidad………………………………………………………….10
8. Pruebas del Software………………………………………………………...11
9. Costos Asociados a la Calidad………………………………………………12
10. Reportes De Problemas Y Acciones Correctivas……………………………13
11. Auditorias de Calidad………………………………………………………..13
12. Solicitudes de cambio…………………………………………………..........13
13. Apéndices……………………………………………………………………13
14. Glosario……………………………………………………………………...13

# Introducción:

Este informe tiene como objetivo mostrar el proceso del aplicativo donde se plantea el control y manejo que se le dará al software con el fin de mejorar el control y comunicación entre los administradores y usuarios.

# Objetivo:

En este documento se elabora el plan administrativo para la calidad del software.

Incluir métodos de calidad en los sistemas para así evaluar métricas e indicadores en el software.

## 2.1 Objetivos de SQA

Los principales objetivos del Aseguramiento de la Calidad del Software son los siguientes:

* Cumplir las actividades y procedimientos establecidos para el desarrollo del software.
* Evaluar la calidad del software y su proceso de desarrollo mediante procedimientos.
* Notificar los problemas relacionados con el software para luego solucionarlos.

# Documentos Relacionados

|  |  |
| --- | --- |
| **Documento** | **Autor** |
| Requerimientos | Diego Colorado |
| Diagrama casos de usos, Diagrama de caso extendido, Diagrama BPMN, Diagrama de Gantt, Diagrama de clases, Diagrama de distribución | Harold Páez, Diego Colorado, Camilo Méndez |
| Modelo entidad de relación | Diego Colorado |
| Diccionario de datos | Camilo Méndez |
| Desarrollo de prototipo | Camilo Méndez |
| Prototipo de interfaz "Mockups" | Harold Páez |
| Inventario del sistemas de información | Diego Colorado |
| Normalización de diagrama de entidad | Camilo Méndez |
| Implementación de sentencia DML, DDL | Camilo Méndez, Diego Colorado, Harold Páez |
| Sistema de integración | Camilo Méndez |
| prototipo funcional usando CCS | Diego Colorado |
| Informe de costos, Informe de migración de datos | Harold Páez, Camilo Méndez, Diego Colorado |
| Manual Técnico, Manual de instalación | Camilo Méndez, Diego Colorado, Harold Páez |
| Pruebas TDD y BDD | Camilo Méndez, Diego Colorado, Harold Páez |
| plan de instalación, plan de respaldo | Camilo Méndez, Diego Colorado, Harold Páez |
| Cuadro comparativo, Realización de contrato, Documentación de pruebas | Camilo Méndez, Diego Colorado, Harold Páez |
| Identificar las características de los procesos de desarrollo de software RAE 31 | Camilo Méndez, Diego Colorado, Harold Páez |
| Identificar los puntos críticos de control en los proceso de desarrollo de software RAE 32 | Camilo Méndez, Diego Colorado, Harold Páez |
| Aplicar los estándares de calidad involucrados en los procesos de desarrollo software. RAE 33 | Camilo Méndez, Diego Colorado, Harold Páez |
| Elaborar instrumentos e instructivos, requeridos por el aseguramiento de calidad. RAE 34 | Camilo Méndez, Diego Colorado, Harold Páez |
| COBIT 5 | José Ángel Peña Ibarra |

# Destinatarios

El presente documento está destinado a las siguientes personas:

|  |  |
| --- | --- |
| **Lector** | **Sector o Rol** |
| Demetrio Mauricio Estupiñan | Instructor |

# Administración - Planeación

Fit Gym System es un aplicativo web, el cual inició en el 2019 con un objetivo y un fin; como objetivo es el manejo del Gym, ayudar al control de usuarios y su conocimiento, colaborando con los entrenadores y gerentes para una eficaz eficiencia. Desarrollado por un grupo de estudiantes del SENA.

Este proyecto cuenta con cinco fases, análisis, diseño, desarrollo, pruebas. El fin es cumplir con los requerimientos que el usuario o cliente dio para así cumplir con un estándar de calidad.

Para cumplimiento contaremos con un plan de administración de riegos para así prevenir fallos en el proceso y ejecución del software.

## 5.1 Organización

Fit Gym System es una organización creada por tres estudiantes del SENA en el año 2019.

La organización del equipo de trabajo SQA se conforma por el Scrum Master y el equipo de desarrolladores. Cuando una falla se presente ellos serán los responsables en detectarlos y solucionarlos.

## 5.2 Responsabilidades

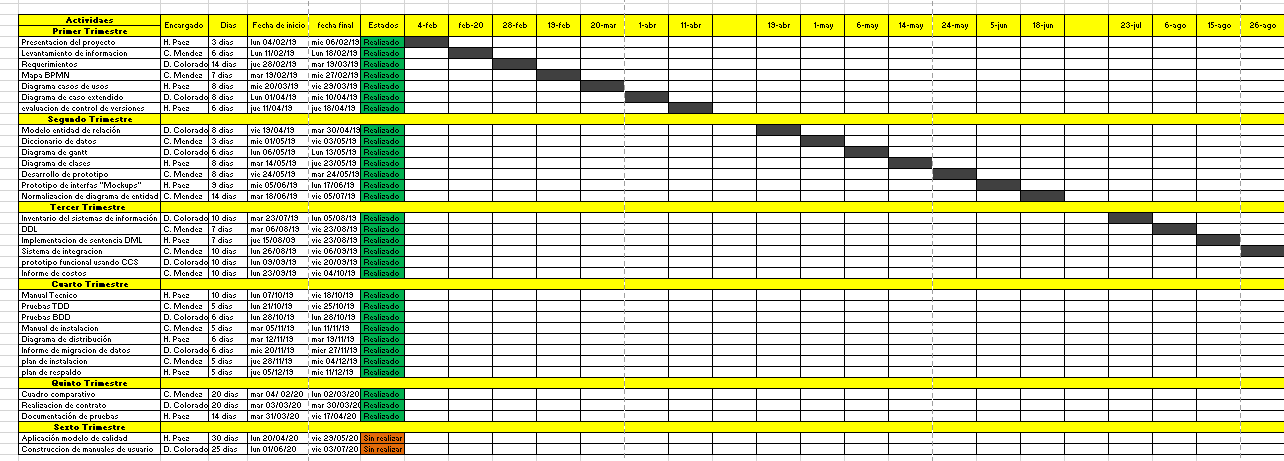
* Las responsabilidades de SQA como desarrolladores de aplicativos FIT GYM SYSTEM son las siguientes.
* Realizar mockups o prototipos para tener una idea de la interfaz del aplicativo y visualizar errores que se pueden encontrar más adelantes.
* Hacer un testing para asegurar que el software funcione de la manera más correcta.
* Hacer evaluaciones a los usuarios y administradores para que sepan cómo manejar el aplicativo.

## Cronograma del proyecto

Se anexa cronograma con fechas de inicio y fin.

[Cronograma\_proyecto.xlsx](file:///C:\Users\Camilo\AppData\Roaming\Microsoft\Word\Cronograma_proyecto.xlsx)

Fecha Inicio: 04/02/2019 Fecha Fin: 03/07/20



*Imagen 1: El cronograma* *realizado se pueden observar las fechas en que se realizó cada documento, desde su inicio hasta su fin. También se puede observar el responsable de cada documento.*

## Riesgos del proyecto

Los riegos que puede tener este proyecto a futuro puede ser:

* Caída del hosting.
* Difícil proceso de instalación del software.
* Robo de información.
* Código basura y redundancia en el código que lo hace incomprensible.
* Inadecuada estructura miento del código.

# Estándares, Practicas, Convenciones y Mediciones

Para el aplicativo de FIT GYM SYSTEM se empleó ISO/IEC 25000 porque se centra en la especificación de requisitos y evaluación de calidad de software para aplicar estándares y metodologías para la medición en el desarrollo del producto ya que el ISO/IEC 25000 se divide en 5 áreas donde son de gran utilidad cuando se evalúe la calidad del aplicativo.

## 6.1 Estándares

En el proceso del desarrollo de la calidad del software se implementarán estándares a los resultados obtenidos durante el análisis de la calidad. Estos serán:

* **ISO/IEC 2500n División de gestión de calidad:**
  + esta división define todos los modelos comunes, términos y referencias a los que se alude en las demás divisiones de SQuaRE
* **ISO/IEC 2501n – División de modelo de calidad:**
  + presenta un modelo de calidad detallado, incluyendo características para la calidad interna, externa y en uso.
* **ISO/IEC 2502n – División de medición de calidad:**
  + Estas normas incluyen un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad (interna, externa y en uso) y guías prácticas para su aplicación.
* **ISO/IEC 2503n – División de requisitos de calidad:**
  + Las normas que forman este apartado ayudan a especificar requisitos de calidad que pueden ser utilizados en el proceso de elicitación de requisitos de calidad del producto software a desarrollar o como entrada del proceso de evaluación.
* **ISO/IEC 2504n – Divisiones de evaluación de calidad:**
  + Este apartado incluye normas que proporcionan requisitos recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software.

# Métricas de Calidad

# Calidad de uso: Calidad del software que el usuario final refleja, la forma como el usuario final logra realizar los procesos con satisfacción, eficiencia y exactitud

# Confiabilidad: Es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas.

# Funcionalidad: Es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas.

# Eficiencia: La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas.

# Portabilidad: La capacidad que tiene el software para ser trasladado de un entorno a otro.

# Usabilidad: Es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva.

# Capacidad de Mantenimiento: Es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software

Cada calidad de métrica tendrá un plan de administración de riesgos el cual sería este:

**Identificar los puntos críticos del proyecto:**

Se harán pruebas donde el software se ejecutará en diferentes navegadores y dispositivos.

**Cuantitativo**

Se analizarán los resultados de las pruebas y las fallas detectas irán a un informe donde se explique el punto crítico para que grupo de desarrolladores definían de manera numérica si es necesario mitigar el riesgo.

**Cuantitativo**

Se calificará los puntos críticos mediante calificación numérica.

1 a 2 Inaceptable

3 a 4 Mínimo aceptable

4 a 5 Aceptable

Si el punto crítico es un fallo que puede afectar la conformidad de portabilidad, adaptabilidad y facilidad de instalación. Se colocará como calificación Inaceptable. Donde pasará a un plan de mitigación de riesgos.

**Control de respuesta**

El grupo encargado de donde se percató del punto crítico se le informará para que tomen acciones de mitigaciones.

Las soluciones podrían ser:

* Agregar líneas de código,
* Generación de un nuevo código o restauración del código.

**Responsables**

Scrum Master y grupo de desarrolladores.

**Monitoreo y seguimiento**

El representante debe interactuar con el grupo desarrolladores para que seleccionen el error y sea mitigado.

# Pruebas del Software

**Pruebas de stress:**

Verificar que el sistema funcione correctamente teniendo en cuenta la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremas como:

* Envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. el objetivo de saturar el programa hasta un punto donde aparezcan errores peligrosos para el aplicativo

**Pruebas unitarias**

Las pruebas unitarias es una forma de comprobar el correcto funcionamiento de una unidad de código.  esto sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado

* Automatizable: No debería requerirse una intervención manual. Esto es especialmente útil para la integración continua.
* Completas: Deben cubrir la mayor cantidad de código.
* Repetibles o Reutilizables: No se deben crear pruebas que sólo puedan ser ejecutadas una sola vez.
* Independientes: La ejecución de una prueba no debe afectar a la ejecución de otra.

**Pruebas de caja blanca:**

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

* Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
* Ejerciten todas as decisiones lógicas en las vertientes verdadera y falsa.
* Ejecuten todos los bucles en sus límites operacionales.
* Ejerciten las estructuras internas de datos para asegurar su validez.

**Pruebas de caja negra:**

Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándose.

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca. Muchos autores consideran que estas pruebas permiten encontrar:

* Funciones incorrectas o ausentes.
* Errores de interfaz.
* Errores en estructuras de datos o en accesos a las Bases de Datos externas.
* Errores de rendimiento.
* Errores de inicialización y terminación.

# Costos Asociados a la Calidad

**Costos de fallas externas:**

es cuando el cliente encuentra un error se le dará una garantía o devoluciones

**Costos de evaluación:**

Si el punto crítico es un fallo que puede afectar la conformidad de portabilidad, adaptabilidad y facilidad de instalación. Se colocará como calificación Inaceptable.  Dónde pasará a un plan de mitigación de riesgos.

**Costos de prevención**

Son los costos de todas las actividades diseñados para prevenir fallas de calidad en los productos como:

* Planificación de inspección
* Auditoria de calidad

# Reportes De Problemas Y Acciones Correctivas

Para generar un reporte de problemas se tuvo que hacer unas pruebas y si su resultado fue inaceptable se deberá documentar dicho error en el siguiente anexo. [Reporte\_problemas.docx](file:///C:\Users\Camilo\AppData\Roaming\Microsoft\Word\Reporte_problemas.docx)

# Auditorias de Calidad

**Realización de auditoría de calidad**

**Auditoria interna**

Se realizan las pruebas de caja blanca y negra, cada integrante del proyecto quedará responsabilizado a realizar dichas pruebas en base a la documentación.

**Auditoria externa**

Se utilizan programas como JMeter programa externo al software también como una prueba de stress.

Pruebas a realizar:

* [Caja Blanca\_Software.docx](Caja%20Blanca_Software.docx) - [Caja Negra\_Software.docx](Caja%20Negra_Software.docx)

# Solicitudes de cambio

Si un cliente o usuario desea realizar un cambio con aprobación administrativa se debe diligenciar el siguiente documente anexado. [Solicitud\_de\_cambio.docx](file:///C:\Users\Camilo\AppData\Roaming\Microsoft\Word\Solicitud_de_cambio.docx)

# Apéndices

Métricas en la Ingeniería de Software.pdf (Estévez, 2016)

Evaluación de calidad (Risti, 2018 - 2019)

COBIT 5 (Peña, 2012)

# Glosario

**SQA:** Aseguramiento de la Calidad del Software que implica revisar y auditar los productos y actividades de software para verificar que se cumplen los procedimientos y los estándares, además de proveer a las gerencias apropiadas (incluyendo a la de proyectos) con los resultados de estas revisiones. Por lo tanto, SQA envuelve al PROCESO de desarrollo de software completo: monitoreando y mejorando el proceso; asegurándose que cualquier estándar y procedimientos adoptados sean seguidos; y, asegurándose que los problemas sean encontrados y tratados. Tomado de: (Findingtc, s.f.)