

Gestión de ingreso

Plan de calidad

Versión: 0100

Versión 1

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito de HyperAccess.

Análisis y Desarrollo de sistemas de información

Brayan Styd Báez Mejía

Valery Teheran Bernett

Tatiana Bermúdez Pinilla

Yulied Alejandra Casas Guerrero

Zaira Julieth Lozano Villarraga



Notas de autor:

Brayan B, Yulied C. Valery T. Zaira L. Tatiana B. Ficha 1964696, Sena, Sede Complejo

Sur, Jornada diurna, Análisis desarrollo de sistemas de información, la correspondencia con este proyecto se dirige a Brayan B, Yulied C. Valery T. Zaira L. Sena, Complejo Sur.

[VTheran@gmail.misena.edu.co](mailto:VTheran@gmail.misena.edu.co) , [zjlozano46@misena.edu.co](mailto:zjlozano46@misena.edu.co) ,

[yacasas05@misena.edu.co](mailto:yacasas05@misena.edu.co) , [bsbaez023@misena.edu.co](mailto:bsbaez023@misena.edu.co) , [tbermuez07@misena.edu.co](mailto:tbermuez07@misena.edu.co) .

Bogotá D.C – Colombia

Tabla de contenido

[1. Introducción 5](#_Toc65601416)

[2. Objetivos 5](#_Toc65601417)

[3. Alcance 5](#_Toc65601418)

[4. CONDICIONES GENERALES 6](#_Toc65601419)

[Glosario 8](#_Toc65601420)

# Introducción

Para lograr que un producto de software sea de calidad adecuada es necesario realizar determinadas actividades de manera sistemática, esto implica planificar desde el comienzo del proyecto la calidad y realizar un plan para alcanzar los objetivos trazados en cuanto a la misma. A este plan se le conoce comúnmente como Plan de la calidad.

En el presente trabajo se brindan los elementos para planificar la calidad en un proyecto desoftware, identificando qué es necesario documentar en función de la forma de trabajo que se haya decidido utilizar por el equipo. Esta forma de trabajo puede diferir en función del riesgo delproyecto, su tamaño, el número de personas involucradas, así como también otros aspectos queserán analizados más adelante.

Dado que el plan de la calidad es un instrumento de apoyo de un equipo de proyecto para la

definición de su forma de trabajo, éste debe adecuarse a las características de cada situación.

# Objetivos

Definir la organización, actividades y responsabilidades asociadas al proceso de SQA durante todo el ciclo de vida del proyecto. Además, entregar guías para la ejecución de las actividades de SQA, definir los estándares, los procedimientos y las convenciones que serán utilizados durante estas actividades y establecer las herramientas, técnicas y metodologías que soportarán las prácticas de SQA.

## Objetivos de SQA

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

* Mejorar la calidad del software monitoreando apropiadamente tanto los productos de software como el proceso de desarrollo que los genera.
* Asegurar el cumplimiento de los estándares y procedimientos establecidos para el software y el proceso de software establecidos.
* Asegurar que cualquier desviación en el producto, el proceso, o los estándares son elevados a la gerencia para poder resolverlas.

# Documentos Relacionados

## Definición de proceso SQA

El SQA se define como un conjunto de actividades planificadas y sistemáticas, cuyo primer objetivo es evaluar la calidad y la adherencia de los productos de software a los estándares, procesos y procedimientos.

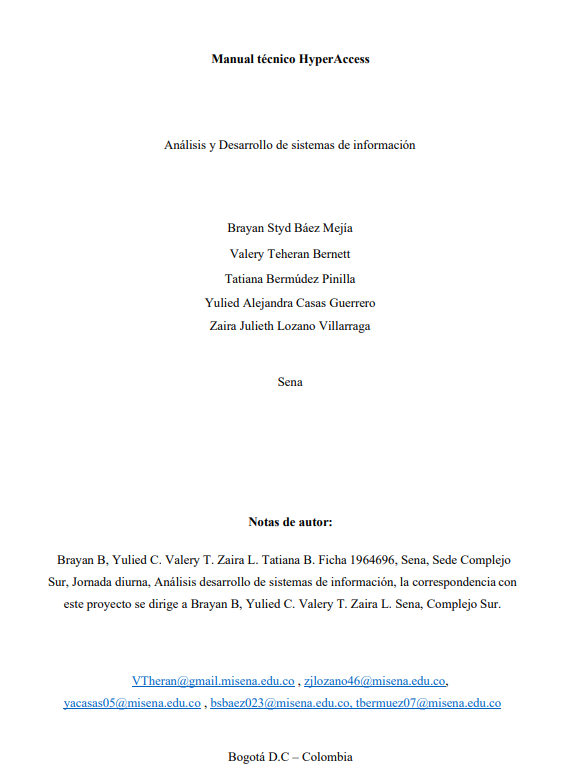
Además, el SQA permite proveer de una visibilidad adecuada del proceso utilizado y los productos construidos mediante acciones planificadas y sistemáticas. Estas acciones permiten asegurar la calidad de dichos procesos y productos.

El desarrollo de un plan de SQA servirá para que un producto de software cumpla con los requerimientos y necesidades que tiene el cliente. El resultado final se encontrará certificado y asegurando la calidad de éste.

Además de lo anterior, va a permitir que se puedan planificar las actividades de aseguramiento de la calidad, revisar y auditar objetivamente los productos y las actividades. Proporciona además los resultados de estas revisiones o auditorias informando a la dirección del proyecto y aumentar la calidad de los entregables durante todo el proceso de desarrollo.

La realización de un plan de aseguramiento de la calidad evitará que la empresa se exponga a situaciones de riesgo.

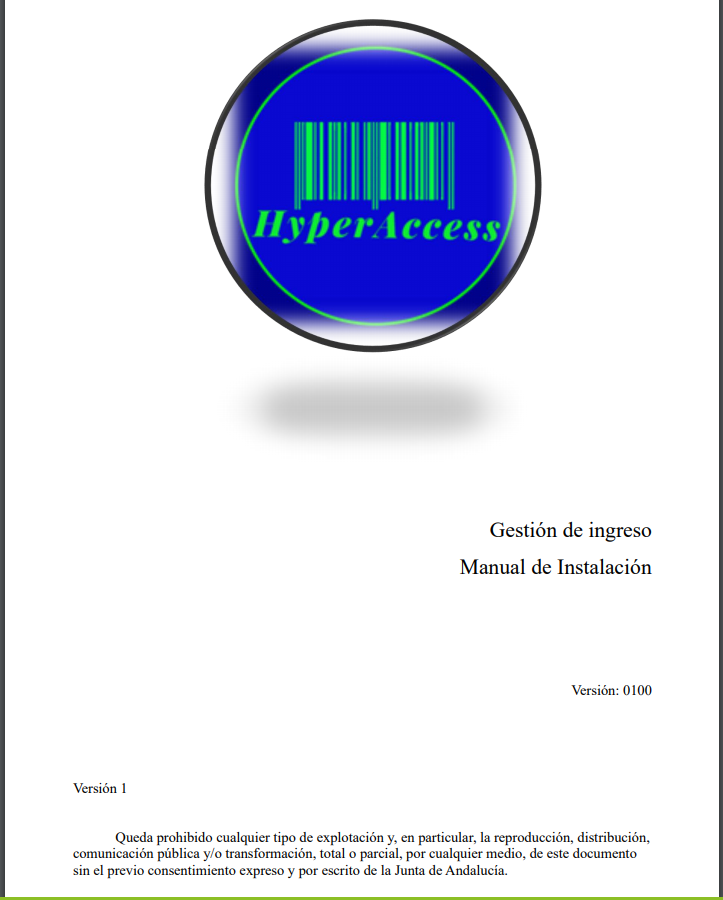
## Manual técnico



[Manual Tecnico.pdf](file:///C:\Users\l\Desktop\ADSI\HyperAccess\Manual%20Tecnico.pdf)

## Cronograma del proyecto

## Manual de instalación.



[Manual de instalación.pdf](file:///C:\Users\l\Desktop\ADSI\HyperAccess\Manual%20de%20instalación.pdf)

# Destinatarios

|  |  |
| --- | --- |
| Lector | Sector o Rol |
| Sonia Isabel Riveros Gómez | Cliente/ Rectora del colegio |
|  | Administrador del sistema |

# Administración – Planeación

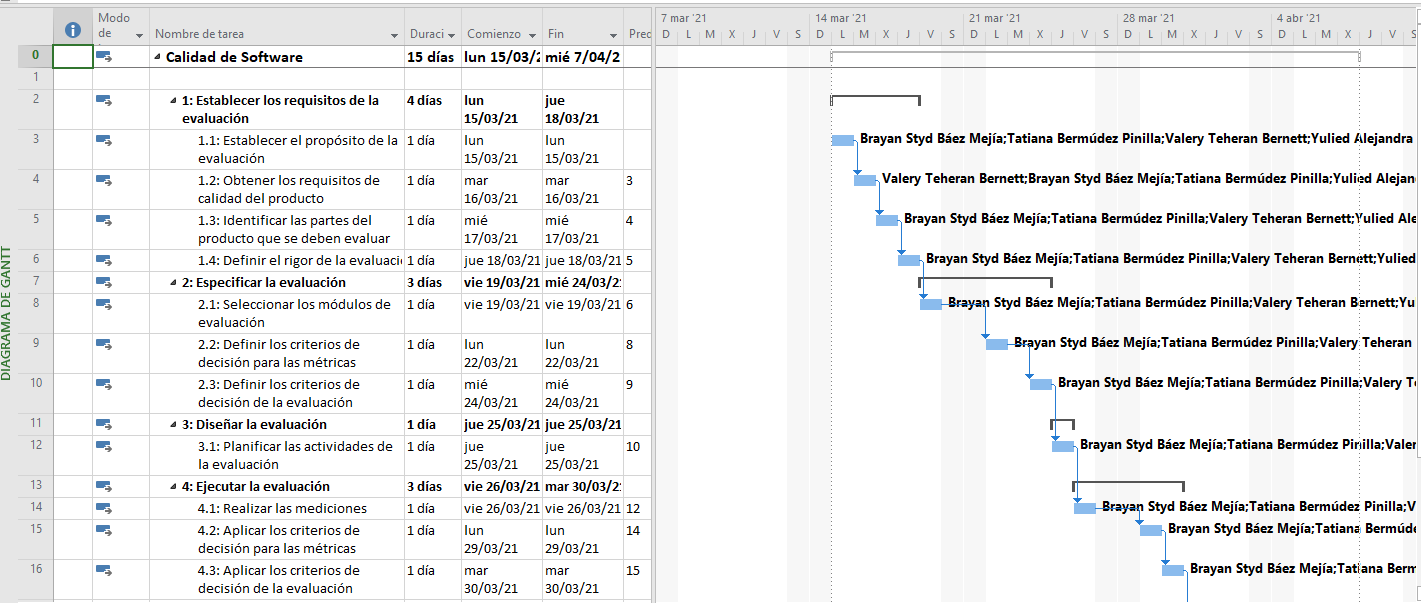
## Organización

|  |  |
| --- | --- |
| Lector | Sector o Rol |
| Brayan Styd Báez Mejía | Líder o moderador |
| Valery Teheran Bernett | Autora |
| Tatiana Bermúdez Pinilla | Revisora |
| Yulied Alejandra Casas Guerrero | Revisora |
| Zaira Julieth Lozano Villarraga | Secretaria o escribano |

## Responsabilidades

* **Líder o moderado**r. Es la persona responsable de la revisión. En cooperación con el autor establece el tipo de revisión y la composición del equipo de revisión. Revisa la planificación, prepara las listas de comprobación, distribuye los documentos a revisar antes de las reuniones y gestiona la reunión de revisión.
* **Autor.** Es la persona que ha escrito el documento a revisar.
* **Revisores.** Son las personas que revisan los documentos.
* **Secretario o escribano.** Registra y documenta los problemas, defectos y recomendaciones.

## Cronograma del proyecto



## Riesgos del proyecto

[Matriz Mapa-de-Riesgos\_V1.xlsx](Matriz%20Mapa-de-Riesgos_V1.xlsx)

# Estándares, Practicas, Convenciones y Mediciones

**ISO/IEC 25000**

El objetivo general de la creación del estándar ISO/IEC 25000 SQuaRE es organizar, enriquecer y unificar las series que cubren dos procesos principales: especificación de requisitos de calidad del software y evaluación de la calidad del software, soportada por el proceso de medición de calidad del software.

ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

La Norma ISO 25000, proporciona una guía para el uso de las series de estándares internacionales llamados requisitos y Evaluación de Calidad de Productos Software (SQuaRE). La norma establece criterios para la especificación de requisitos de calidad de productos software, sus métricas y su evaluación, e incluye un modelo de calidad para unificar las definiciones de calidad de los clientes con los atributos en el proceso de desarrollo.

Las características de calidad y sus mediciones asociadas pueden ser útiles no solamente para evaluar el producto software sino también para definir los requerimientos de calidad.

## Estándares

### IEEE

[IEEE PROYECTO HYPER ACCESS.pdf](IEEE%20PROYECTO%20HYPER%20ACCESS.pdf)

# Métricas de calidad

Métricas de calidad: Proporcionan una indicación de cómo se ajusta el software a los

requisitos implícitos y explícitos del cliente. Es decir, cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.

## [Auditoria de calidad CMMI.pdf](Auditoria%20de%20calidad%20CMMI.pdf)

## <METRICAS_DE_CALIDAD.xlsx>

# Pruebas del Software

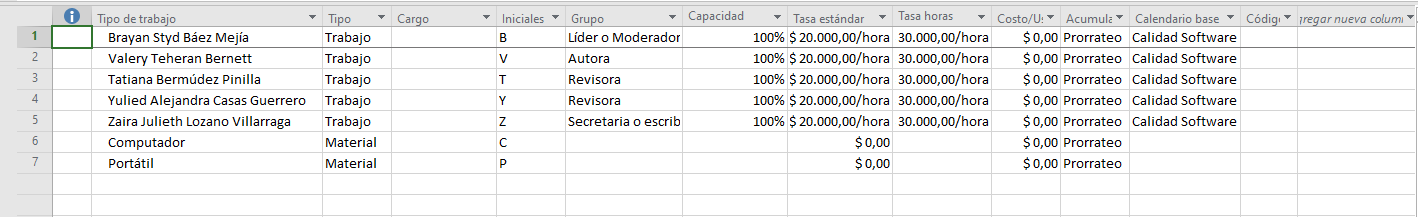
El plan de pruebas es un conjunto de casos de pruebas que se encarga de probar una funcionalidad completa de un producto o software en concreto. Es una guía para la realización de las pruebas, con la ejecución de los casos de prueba y en base al resultado de los mismos, sabremos si el software cumple con las necesidades establecidas en el proyecto y cubre los estándares de calidad definidos.



Debe de realizarse a partir de los requisitos aportados en el proyecto o en las historias de usuario, depende de la metodología que se utilice. Además, se deben de realizar a partir de las especificaciones, del diseño funcional, del prototipado y de los casos de uso, entre otra documentación que se pueda obtener dentro del proyecto.

[Plan de pruebas HyperAccess.pdf](Plan%20de%20pruebas%20HyperAccess.pdf)

# Costos Asociados a la Calidad



# 10. Reportes De Problemas Y Acciones Correctivas

SQA debe documentar, o hacer referencia a, un procedimiento para manejar desviaciones de lo real respecto a lo esperado, que típicamente indica que:

* Las desviaciones relacionadas con Plan de Desarrollo de Software, y con los estándares y procedimientos designados para el proyecto, son documentados y resueltos con los líderes involucrados o el gerente del proyecto, cuando sea posible.
* Las desviaciones relacionadas con Plan de Desarrollo de Software, y con los
* estándares y procedimientos designados para el proyecto, y que no se resuelvan con
* los líderes o el gerente del proyecto, son documentados y presentados al gerente
* superior designado para recibir problemas de no cumplimiento.
* Los puntos de no cumplimiento presentados al gerente superior son revisados periódicamente hasta que sean resueltos.
* La documentación de puntos de no cumplimiento es administrada y controlada.

## 10.2. Técnicas de detección de defectos. Las revisiones y el análisis estático

Las técnicas de detección de defectos incluyen las técnicas relacionadas con lo que llamamos pruebas estáticas, en las que encontramos las revisiones y el análisis estático.

### 10.2.1 Revisiones

El objetivo de las revisiones es encontrar defectos examinando entregables no ejecutables.

Tanto si consideramos las revisiones como actividades del aseguramiento de la calidad como si son parte de las pruebas, las revisiones se diferencian de las pruebas "clásicas" en que se realizan durante todo el ciclo de vida de desarrollo y su objeto de estudio es cualquier tipo de entregable no ejecutable: requisitos y especificaciones, documentos de diseño (funcional, técnico...), código fuente, documentos de prueba, manuales de usuario, etc.

**10.2.1.1. Tipos de revisiones.**

Podemos diferenciar varios tipos de revisiones formales:

* **Tutorial.** El autor del documento a revisar guía a los participantes a través del documento a fin de que lo entiendan. Este tipo de revisión es especialmente útil para documentos de alto nivel, como especificaciones de requisitos o documentos de arquitectura, cuando los revisores no tienen conocimientos de los documentos de desarrollo.
* **Revisión técnica.** Su objetivo es la revisión del contenido técnico de un documento por parte de expertos (arquitectos, diseñadores, usuarios clave...). A veces se llaman "revisión de igual a igual".
* **Inspección**. En este caso, el documento a revisar es preparado y evaluado antes de la reunión por todos los revisores, que comparan el documento con otros documentos referenciados y usan listas de comprobación. A diferencia de los anteriores tipos de revisiones, hay un moderador que no es el autor, y se realiza un seguimiento posterior de las anotaciones y cambios en el documento.
* **Revisión de gestión.** Su objetivo es monitorizar el progreso y los riesgos de un proyecto, sistema, actividad o proceso y tomar decisiones de mejora o actuación. Los gestores que están implicados en el elemento a revisar son a menudo quienes realizan estas revisiones, pero en algunos casos se contratan consultores externos para hacer evaluaciones de procesos. El resultado de la revisión de gestión es el establecimiento de acciones y recomendaciones.

**10.2.1.2. Listas de comprobación**

Las listas de comprobación son herramientas de soporte muy prácticas para las revisiones. Establecen un guion o lista de los elementos que tienen que ser revisados o ser comprobados. Estas listas de comprobación se construyen basándose en la experiencia y estándares.

Una lista de comprobación incluye habitualmente, para cada criterio o regla a comprobar, dos columnas de valoración, una para indicar si se cumple (Sí, No, No aplicable) y otra para describir la causa de la valoración. Para mejorar su lectura, los criterios se suelen clasificar en grupos.



### Análisis estático

El análisis estático es una técnica para evaluar código y modelos (en UML, por ejemplo) realizada con el apoyo de herramientas.

Podemos detectar, por ejemplo, si se referencia una variable que no tiene valor definido, qué variables no se utilizan nunca, si existe código muerto que nunca se ejecutará, el incumplimiento de estándares de programación, vulnerabilidades de seguridad, etc.

Existen tres técnicas principales de análisis estático:

* + - 1. **Análisis de la estructura del código**

Con la estructura del código podemos analizar su flujo de control y su flujo de datos.

Con el flujo de control se evalúa la secuencia en la que las instrucciones serán ejecutadas (bucles, condiciones, etc.) para poder detectar, entre otros, código muerto y funciones que nunca son llamadas.

El flujo de datos permite analizar cómo un elemento de datos –una variable– es accedido y modificado en el código. Podemos detectar el uso de una variable sin definir (sin valor), variables que nunca son usadas o variables que se usan después de haber sido destruidas.

* + - 1. **Cumplimiento de estándares de programación**

Mediante una herramienta de análisis estático de código se puede verificar el seguimiento de cualquiera de las convenciones definidas en un estándar de codificación o programación.

Por ejemplo, se puede comprobar, entre otros:

* Seguimiento de convenciones de nombres de las clases, métodos, parámetros.
* Seguimiento de buenas prácticas de programación: uso de cadenas, literales, concatenación, seguridad, inicialización de variables fuera de los bucles, etc.
* Seguimiento de formato: número de líneas por método, por clase, etc.
  + - 1. **Cálculo de métricas de código**

Cuando se realiza el análisis estático del código se extrae información de su estructura interna, como, por ejemplo:

* Líneas de código
* Complejidad ciclo matica
* Índice de mantenibilidad
* Número de descendientes

Las diferentes medidas obtenidas tienen que ser evaluadas y comparadas con el tiempo para comprobar si se están realizando mejoras en el código o estamos empeorando su calidad.

# Auditorias de Calidad

En un Libro de Excel, implementar una lista de chequeo del modelo de calidad.

En cada una de las hojas de dicho libro, se implementará cada uno de los roles que intervienen en nuestro proyecto. Por ejemplo, Rol “Administrador”. Dentro de esta hoja, implementamos todos los módulos de nuestro proyecto. Por ejemplo, Página de inicio, registro de estudiantes, listado de docentes etc.

Cada uno de estos módulos, tendrá sus correspondientes ítems de auditoría del proyecto. En decir, que cada una de las hojas del libro, tendrán todos los módulos que pueden intervenir en dicho módulo.

[Auditoria de calidad CMMI.pdf](Auditoria%20de%20calidad%20CMMI.pdf)

# Apéndices

# [Manual de usuario.pdf](Manual%20de%20usuario.pdf)

# [Manual de Operaciones.pdf](Manual%20de%20Operaciones.pdf)

# Glosario

* **Métrica:** Una métrica de calidad es una definición operativa que describe un atributo del producto o del proyecto. Una medición es un valor real.

Una métrica indica la manera en que el proceso de control de calidad medirá el trabajo o el producto.

A su vez, la tolerancia define la variación permisible de las métricas.

* **Calidad:** Son características propias del software, aquellas que tu quieres controlar y asegurar. El software es un producto inmaterial que no se fabrica, tampoco se degrada físicamente, pero sí se desarrolla. El software puede tener errores e incidencias, pero no son similares a las de cualquier equipo de carácter físico.

La calidad del software se encuentra casi a la par de la calidad tradicional, ligeramente detrás, debido a que la calidad tradicional tiene varias décadas de historia, mientras que la calidad de software tiene entre 50 y 30 años de haber surgido.

El software necesita ser actualizado.

* **Mantenibilidad:** Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.
* **Concatenación:** Concatenación es el proceso de anexar una cadena al final de otra cadena. Las cadenas se concatenan con el operador. En el caso de los literales y las constantes de cadena, la concatenación se produce en tiempo de compilación, y no en tiempo de ejecución.
* **Vulnerabilidades:** Una vulnerabilidad en un programa informático o software es simplemente un error, un problema en su código o en su configuración. Es muy probable –por no decir que se produce siempre– que los programas contengan errores, puesto que han sido creados por seres humanos.
* **ISO 25000:** ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

* **Defecto:** A los defectos del software se les conoce comúnmente en el argot de los desarrolladores como Bug (bicho), y corresponde a un error, imperfecto o falla de una aplicación para computador que puede causar un resultado no deseado o incumplimiento de un requerimiento.

# Referencias

ISO/IEC 25000. (2020, 13 de noviembre). Wikipedia, La enciclopedia libre. Fecha de consulta: 19:52, mayo 22, 2021 desde <https://es.wikipedia.org/w/index.php?title=ISO/IEC_25000&oldid=130890505>.

JOSE ESTERKIN. (13 ABRIL 2010). Las métricas de calidad en el proyecto. Métricas de Calidad. En wordpress. De <https://iaap.wordpress.com/2010/04/13/las-metricas-de-calidad-en-el-proyecto/>

Calidad de software. (2021, 22 de marzo). Wikipedia, La enciclopedia libre. Fecha de consulta: 23:15, junio 29, 2021 desde <https://es.wikipedia.org/w/index.php?title=Calidad_de_software&oldid=134169437>.

iso25000. (2021). Mantenibilidad. En iso25000. De <https://iso25000.com/index.php/normas-iso-25000/iso-25010/26-mantenibilidad>