

# Plan de Calidad

## Integrantes

Luis Felipe Ayala Urquiza  
[luisfayala@javeriana.edu.co](mailto:luisfayala@javeriana.edu.co)

Janet Chen He  
[j-chen@javeriana.edu.co](mailto:j-chen@javeriana.edu.co)

Harold Duvan Pinilla Salinas  
[hdpinilla@javeriana.edu.co](mailto:hdpinilla@javeriana.edu.co)

## Asignatura

Calidad de Software

## Profesor

Ing. Fabrizio Bolaño Lopez



PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE SISTEMAS  
BOGOTÁ D.C. 2022

## Contenido

1. Objetivo y ámbito de aplicación .....	3
2. Definiciones y acrónimos .....	3
3. Documentos de referencia.....	3
4. Visión general del plan SQA .....	4
a. Organización e independencia.....	4
b. Riesgo del producto de software .....	4
c. Herramientas .....	5
d. Normas, prácticas y convenciones .....	5
e. Esfuerzo, recursos y calendario .....	5
5. Actividades, resultados y tareas .....	7
a. Garantía del producto.....	7
b. Evaluar la conformidad de los planes .....	7
c. Evaluar la conformidad del producto .....	7
d. Evaluar la aceptabilidad de los planes .....	7
Referencias .....	7

## 1. Objetivo y ámbito de aplicación

Este documento ofrece pautas y herramientas aplicables al proyecto del curso Covenant, para el desarrollo basado en el aseguramiento de la calidad de software. Puesto que se evalúa constantemente el buen desarrollo, la correcta implementación, requerimientos y los estándares necesarios para el desarrollo.

Es decir, el objetivo es entender que es un software de calidad y como esta es aplicable a todas las fases de trabajo. Puesto que es importante, en todo momento identificar posibles errores, técnicas optimas de metodologías, medición y generación de informes.

Permitiendo así, garantizar al cliente una alta capacidad de ejecutar el producto con bajos costos y el buen manejo de los recursos.

Y finalmente con esto se podría definir el ámbito o alcance de proyecto, que sería realizar un sistema de concertación de propuestas sectoriales con aseguramiento de la calidad de software.

## 2. Definiciones y acrónimos

- **Calidad de software:** Es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.
- **EDT:** Estructura detalla de trabajo.
- **Riesgo de producto:** Estado en donde el producto de software puede causar daños a los desarrolladores y/o a los usuarios del sistema.
- **SQA:** Software Quality Assurance – Aseguramiento de la calidad del software
- **Convención:** Norma o práctica aceptada socialmente por un acuerdo general o por la costumbre

## 3. Documentos de referencia

Para este proyecto se presenta varios documentos para el correcto desarrollo, algunos de ellos serían.

- **Estándar IEEE 730-2014:** Es el estándar utilizado para el desarrollo de esta guía que consta de un estándar para planes de aseguramiento de calidad del software.
- **Estándar IEEE 42020-201:** Estándar internacional para software, sistemas y empresas en el proceso de arquitectura.
- **26511-2018 - Norma Internacional ISO/IEC/IEEE:** Requisitos para administradores de información de usuarios de sistemas, software y servicios

Otros documentos de referencias no formales, se presenta en los al final de este documento, en la sección de referencias.

## 4. Visión general del plan SQA

### a. Organización e independencia

En esta sección se especifica la organización que aplica el aseguramiento de la calidad, en este caso, se presenta la siguiente ilustración, que representa el equipo de trabajo encargado tanto de la parte de asegurar el plan SQA, como el desarrollo.



Ilustración 1. Equipo de trabajo

Debido a la falta de personal externo al proyecto, se concluye que el grado de independencia es prácticamente nula, puesto que no se cumplen ninguna de los tres parámetros.

Sin embargo, se puede plantear posibles escenarios para cada uno. Por ejemplo, en el caso de independencia técnica, se puede solicitar a otro grupo de compañeros del curso para formar evaluaciones de las actividades propuestas en este documento y así detectar posibles errores que se hay pasado por alto.

Luego, para la independencia general, se puede presentar en un análisis menos profundo y realizada por una organización desligada a los que desarrollaron el software y la gestión del proyecto.

Finalmente, la independencia financiera, permitiendo una visión más amplia de parte económica y recursos y pueda afectar directa o indirectamente el proyecto.

### b. Riesgo del producto de software

- Riesgos físicos del producto:

Para la problemática tratada en el proyecto del curso, no existen riesgos físicos ya que no se creará ningún sistema embebido en algún hardware o firmware que pueda poner en riesgo la integridad de las personas o que se pueda ver afectado por algún factor o estímulo físico.

- Riesgo de seguridad del producto:

Al igual que sucede con los riesgos físicos del producto, los riesgos de seguridad en este proyecto no aplican ya que no se requiere construir software que funcione en diferentes dispositivos que puedan afectar la seguridad de las personas o del mismo dispositivo como tal.

- Riesgos financieros del producto:

En este proyecto, los riesgos financieros que se pudieran presentar lo harían, principalmente, en la inversión que haría el Departamento Nacional de Planeación para la construcción de este y en su implementación y despliegue ya que puede que no funcione de manera eficiente o que se generen inconsistencias en el tratamiento de datos a la hora de registrar distintos usuarios y/o departamentos. Estos datos erróneos podrían ocasionar inconsistencias en los registros de la aplicación que haría que se tuvieran que corregir y volver a implementar, lo que haría que se incrementara cada vez más el costo de operación.

### *c. Herramientas*

Se identifica para este proyecto herramientas útiles para el correcto desarrollo y aseguramiento de la calidad de software, las cuales son:

- **Arquitectura de la solución:** Permite identificar en el alto nivel la problemática y la solución.
- **Diagrama de contexto:** Detalla en niveles la propuesta presentada en la arquitectura de la solución.
- **EDT:** Permite clasificar las tareas necesarias para desarrollar en fase.
- **Diagrama de Gantt:** Permite presentar el cronograma del proyecto y así mismo estimar el tiempo de trabajo.
- **Organigrama:** Permite visualizar el equipo de trabajo asociado al proyecto
- **Tabla de adquisiciones:** Presenta el plan a seguir para la adquisición de recursos necesarios para la correcta implementación del proyecto.
- **Tabla de comunicaciones:** Se establece los canales y los roles de cada integrante para estipular una buena comunicación a lo largo del trabajo
- **Tabla de riesgos:** Permite visualizar posibles riesgos y clasificarlos según la gravedad, para poder tomar acción al respecto según prioridad.

### *d. Normas, prácticas y convenciones*

En esta sección se presenta estimaciones necesarias para completar las actividades o tareas y así dar resultado con base al SQA.

Se presenta que para esta primera entrega. Existen dos hitos, que es la entrega de este documento hasta el ítem 5 d. y la entrega de la propuesta técnica hasta parte del ítem 3, con ayuda de la guía del curso entregado por el docente.

Y se menciona que estas estimaciones se hacen en base también en trabajos presentados con anterioridad a lo largo de la carrera de ingeniería de sistemas como en las entregas recientes de talleres y trabajos en clase en el curso de Calidad de Software.

### *e. Esfuerzo, recursos y calendario*

Existen dos fases de análisis, durante la planificación y durante la ejecución. Para esta primera entrega solo se analizará la primera parte.

- Durante la fase de planificación

Para el desarrollo del proyecto como técnica de estimación y programación se hará uso de una estructura detalla de trabajo (EDT), que consiste en subdividir el proyecto de manera organizada y jerárquicamente en base a algún criterio para lograr completar con éxito lo propuesto.

Teniendo lo anterior, es importante mencionar también que existe tres modelos para esta estructura, la primera es descendente, que consta de un enfoque desde la parte más general hasta lo más específico, usado principalmente si se tiene experiencia en el proyecto.

Luego, el segundo modelo, es lo contrario el ascendente, por lo que va desde lo más bajo hasta lo general, es decir, es como si se recompilara la información necesaria para luego unirlo y generar lo que es el proyecto en general.

Finalmente, el enfoque yoyó que combina las dos técnicas anteriores, en donde se presenta las divisiones del proyecto y tareas asociadas a ello, haciendo lo que es un trabajo al par.

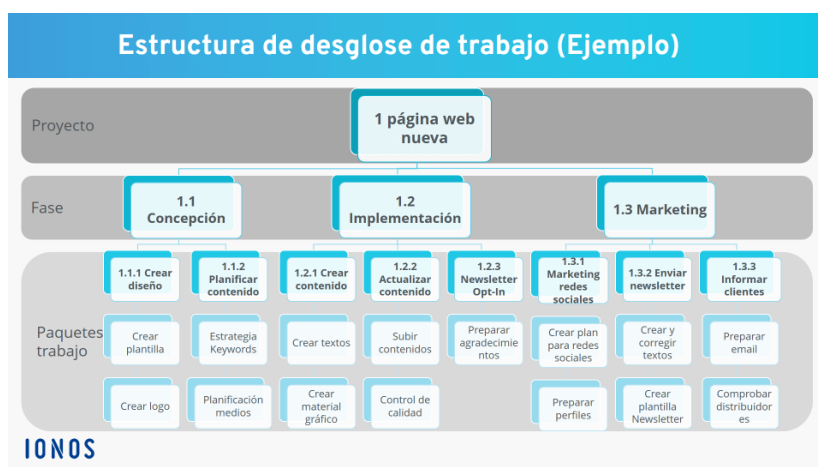


Ilustración 2. Ejemplo de una EDT

Después, a partir de cualquiera de los enfoques de la estructura EDT, se puede realizar un cronograma del proyecto denominado, diagrama de Gantt, que permite distribuir las tareas específicas y las importantes, en fechas y estimar un control del trabajo a lo largo del tiempo estipulado para su realización.

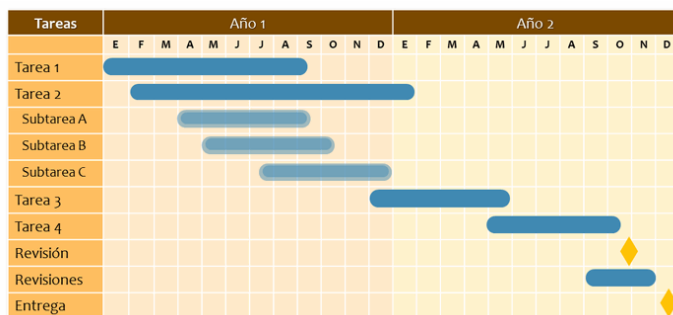


Ilustración 3. Estructura de un diagrama de Gantt

## 5. Actividades, resultados y tareas

### *a. Garantía del producto*

Algo importante a la hora de realizar un producto, en todas las fases, es poder garantizar que el producto cumple con los requisitos o aspectos importantes del SQA que establezca una confianza en la calidad.

Para ello se establece la revisión y entrega del trabajo al docente del curso, evaluando de esta manera además del correcto funcionamiento, los lineamientos con lo establecido a lo largo de este documento.

### *b. Evaluar la conformidad de los planes*

Se evalúa todos los planes requeridos que hayan sido preparados y se espera que sean consistentes a lo solicitado.

En el caso de Covenant sería, analizar si la propuesta técnica es acorde y presenta herramientas útiles para mejorar el desarrollo del proyecto.

Es decir, estructuras como un plan de gestión de proyecto, el monitoreado e informe de seguimiento, identificación de recursos y posibles fallas, etc.

### *c. Evaluar la conformidad del producto*

Se establece la evaluación en que el producto software y la documentación relacionada se ajuste a los requisitos establecidos. Para ello, se establece el análisis de la sección del plan de desarrollo del documento de la propuesta técnica.

Evaluando de este modo las limitaciones que pueda tener la tecnología a utilizar, es decir, la capacidad de cómputo o el manejo por lo todos los integrantes, etc. Otros aspectos pueden ser, los criterios de verificación, si se desarrolló una buena estrategia de validación, etc.

### *d. Evaluar la aceptabilidad de los planes*

Finalmente, en esta sección se establece el nivel de confianza de los productos, es decir, como se identifica que serán aceptado un plan. Para ello se realizará preguntas como, si se ha desarrollado una estrategia adecuada de en la instalación del programa. Si se ha evaluado en un contexto adecuado, etc.

## Referencias

1. Descubre los beneficios del Aseguramiento de calidad de software (SQA) - Globalbit. (s. f.). Globalbit. Tomado de: <https://www.globalbit.co/2020/04/25/descubre-los-beneficios-del-aseguramiento-de-calidad-de-software-sqa/>
2. Cómo se elabora una estructura de desglose de trabajo. (s. f.). IONOS Startupguide. Tomado de: <https://www.ionos.es/startupguide/productividad/estructura-de-desglose-de-trabajo/>
3. Software quality concepts and practice - Galin, Daniel. Capítulo 7. SQA plan and project plan