

Primer Informe

Ingeniería del Software

NOMBRE DEL PROYECTO: SISTEMA WEB PARA EVALUACIÓN Y AUTOEVALUACIÓN
DOCENTE

URL: [Login - SEADOC \(brayanguerreroxd.github.io\)](http://Login - SEADOC (brayanguerreroxd.github.io))

Claves: _____

Usuario y claves administrador: **Código:** 115 **Documento:** 2 **Contraseña:** admin1234.

Usuario y claves otros roles: _____

Integrantes:

Código: 1151770 **Nombre:** Natalia Ortiz Armesto

Código: 1151983 **Nombre:** Brayan Stewart Guerrero Ordoñez

Código: 0581055 **Nombre:** Jesús Uriel Martínez Alcántara

Fecha de máxima DOMINGO 16 DE ABRIL ANTES DE LAS 2:00PM de subir a la
carpeta compartida al equipo de trabajo

OBSERVACIÓN: EL INFORME tiene un valor del 50% del Primer Parcial + 50% del examen escrito

Índice

ADMINISTRACIÓN DE PROYECTO DE SOFTWARE	7
1. PERSONAL	7
1.1. Introducción	7
1.2. Planteamiento del problema a resolver	7
1.3. Justificación	9
1.4. Objetivos	10
1.4.1. Objetivo General	10
1.4.2. Objetivos Específicos	10
1.5. Alcance y delimitación	10
1.5.1. Alcance del software a desarrollar	10
1.5.2. Limitaciones del software a desarrollar	11
1.6. Administración del personal del proyecto	12
1.6.1. Participantes, líderes del proyecto, equipo de software	12
Tabla 1. Descripción directora del proyecto SWEAD	12
Tabla 2. Descripción gerente del proyecto SWEAD	13
Tabla 3. Descripción coordinadora del proyecto SWEAD	13
Tabla 4. Descripción equipo de desarrollo del proyecto SWEAD	14
1.6.2. PSP alineado al proyecto de software en desarrollo	15
Figura 1. Metodología PSP	16
1.6.3. Proceso para equipos de software (TSP) alineado al proyecto de software en desarrollo	17
Figura 2. Ilustración proceso de software del equipo (TSP)	17
1.7. Opciones para la organización del personal.	18
1.7.1. Administración de la comunicación	18
1.7.2. Opciones para la estructura de las responsabilidades	19

2. PROCESOS	20
GESTIÓN Y COMPRENSIÓN DEL NEGOCIO	20
2.1. Modelado general de procesos	20
Figura 3. Ilustración modelado general de los procesos del proyecto SWEAD	20
2.2. Modelo o ciclo de vida del software aplicado al proyecto y descripción de sus fases	21
Figura 4. Modelo de ciclo de vida en V. Elaboración propia	24
2.3. Tecnología del proceso	24
2.3.1. Descripción de los procesos tecnológicos que se llevaron a cabo	24
2.4 Fusión de producto y proceso	26
2.5 Descomposición del proceso	27
Figura 5. Descomposición de los procesos del problema a desarrollar en el proyecto SWEAD	27
CALENDARIZACIÓN DEL PROYECTO	28
Tabla 1. Calendario de actividades del proyecto SWEAD	31
2.6. Estimación del proyecto	32
Tabla 2. Tipos de componentes de los puntos de fusión	32
Tabla 3. Estimación Basado en Puntos de Fusión del Software SWEAD	34
Tabla 4. Puntos de Fusión sin Ajustar	35
Tabla 5. Tabla de Factores de Ajuste de Complejidad Técnica.	36
Tabla 6. Métrica de cada etapa del proyecto SWEAD	37
2.7. Recursos humanos, tecnológicos, recursos de software reutilizables, recursos del entorno	37
2.7.1. Recursos Humanos	37
Tabla 7. Recursos humanos del proyecto SWEAD	38
2.7.2. Recursos tecnológicos	38
Tabla 8. Recursos tecnológicos del proyecto SWEAD	39
2.7.3. Recursos de software reutilizables	39

Tabla 9. Recursos de software reutilizables del proyecto SWEAD.	40
2.7.4. Recursos del entorno	40
Tabla 10. Recursos del entorno del proyecto SWEAD.	42
2.8. Técnicas de descomposición aplicadas al proyecto en desarrollo	42
2.8.1. Tamaño del software	42
2.8.2. Basada en el problema	42
Tabla 11. Métrica de usabilidad del proyecto SWEAD	43
Tabla 12. Métrica de madurez del proyecto SWEAD	45
Tabla 13. Métrica de adecuación del proyecto SWEAD	46
2.8.3. Basada en LDC	47
Tabla 14. Estimación por líneas de código del proyecto SWEAD.	48
2.8.4. Basada en PF	49
Tabla 15. Tipos de componentes de los puntos de fusión	49
Tabla 16. Estimación Basado en Puntos de Función del Software SWEAD.	52
Tabla 17. Puntos de Fusión sin Ajustar.	52
Tabla 18. Tabla de Factores de Ajuste de Complejidad Técnica.	54
Tabla 19. Puntos de función ajustado (PFA)	55
Tabla 20. Horas de lenguaje de programación 4ta generación	55
Tabla 21. PFA	55
Tabla 22. Métrica de cada etapa del proyecto SWEAD	56
2.8.5. Basada en casos de uso	56
Tabla 23: Factores de Peso de los Actores sin Ajustar.	58
Tabla 24: Factor de Peso en los Casos de Uso sin Ajustar	58
Tabla 25: Peso de los Factores de Complejidad Técnica	60
Tabla 26: Peso de los Factores Ambientales	62
Tabla 27. Distribución Real del Esfuerzo	63
3. PRODUCTO	63

3.1. Ámbito del software	63
3.2. Descomposición del problema del software a desarrollar	63
Figura 6. Descomposición de los procesos del problema a desarrollar en el proyecto SWEAD	64
ADMINISTRACIÓN DEL RIESGO DEL PROYECTO EN DESARROLLO	66
4. RIESGOS	66
4.1. Identificación de los riesgos del proyecto en desarrollo	66
Tabla 28. Riesgos Identificados	69
4.2 Estrategias reactivas del riesgo frente a estrategias proactivas del riesgo	69
4.3. Riesgos del software del proyecto en desarrollo	70
Tabla 29. Riesgos del software	70
4.4. Proyección del riesgo	70
Tabla 30. Criterios de probabilidad	71
Tabla 31. Análisis del impacto	72
Tabla 32. Matriz de probabilidad de impacto	72
Tabla 33. Calificación del riesgo	73
Tabla 34. Analisis y clasificacion del riesgo	75
4.5. Mitigación, monitoreo y manejo de riesgos.	76
Tabla 35. Matriz de riesgos y controles	80
4.6. El Plan de Riesgos	81
Tabla 36. Plan de riesgos	82
ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE	83
5. ASEGURAMIENTO DE CALIDAD	83
Figura 7. Modelo de aseguramiento de la calidad según Bohem para el proyecto SWEAD	83
5.1. Establecer procedimientos organizacionales y estándares para la calidad	84
6. PLANEACIÓN DE CALIDAD	84



6.1. Seleccionar procedimientos aplicables y estándares para un proyecto en particular y modificar estos como sean requeridos	85
Actividades del grupo	85
Tabla 37. Descripción de las actividades del grupo del proyecto SWEAD	85
7. CONTROL DE CALIDAD	86
7.1. Garantizar que procedimientos y estándares son seguidos por el equipo de desarrollo de software	87
Índice de Figuras	87
Índice de Tablas	88
Anexos	89

ADMINISTRACIÓN DE PROYECTO DE SOFTWARE

1. PERSONAL

1.1. Introducción

Todo proyecto software exitoso debe pasar por una serie de etapas, estrategias, herramientas y organización por parte del equipo que desarrolla el proyecto, la buena administración de proyectos de software requiere de la capacidad y experiencia de las personas que participan en él, este proceso conlleva a la implementación de estrategias que incrementan la productividad, calidad y eficacia del mismo. Con el presente informe se da a conocer una de las etapas de la administración del sistema web para la evaluación y autoevaluación docente (SWEAD) con el fin de obtener un producto final con los altos estándares de calidad y organización que permitan al cliente satisfacer sus necesidades.

1.2. Planteamiento del problema a resolver

La docencia en la universidad ha permitido que muchos jóvenes colombianos logren sus objetivos de ser profesionales y así ofrecer soluciones a problemas que se presentan hoy en día, por ende se evidencia la importancia de dicha profesión en la sociedad colombiana. Si bien el desarrollo de competencias por parte de los estudiantes depende de múltiples factores individuales y de contexto, los docentes mantienen un papel importante en los procesos de enseñanza, ya sea desde la dirección de los establecimientos educativos o desde



la práctica pedagógica en las aulas. En pocas palabras, el docente es esencial en cualquier modelo de calidad de la educación, por lo que su evaluación es una acción estratégica para el mejor desempeño y calidad del mismo.

Según el Ministerio de Educación Nacional de Colombia, la evaluación es un medio que nos permite conocer los aciertos y las equivocaciones, verificar si los procesos para alcanzar las metas son adecuados y si el logro de los resultados es conveniente o inconveniente con respecto a los propósitos. Esto nos permite crear alternativas de mejoramiento que comprometan a todos los actores del sector educativo para avanzar más rápidamente.

La evaluación de docentes permite identificar el crecimiento personal y profesional de los mismos. De igual modo, constituye una herramienta a partir de la cual los docentes, los directivos y los establecimientos educativos tengan la posibilidad de trazar estrategias que conduzcan al desarrollo efectivo de las competencias que requieren los maestros y maestras para que los jóvenes estudiantes accedan a una educación de mayor calidad.

En este contexto, el programa de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander carece de una estrategia de evaluación, seguimiento y retroalimentación de los docentes, ya que es importante que entre ellos mismos identifiquen qué competencias necesitan reforzar. Dado esto, se requiere implementar un sistema que permita que entre ellos mismos se evalúen, para así de esta manera asegurar un mejor desarrollo de

estrategias basadas en un fortalecimiento continuo de dichas competencias y así ofrecer una mejor calidad educativa a los estudiantes de dicho programa.

1.3. Justificación

El programa de ingeniería de sistemas de la Universidad Francisco de Paula Santander, no cuenta con un sistema que le permita mantener una evaluación, seguimiento y retroalimentación de los docentes. Para el programa de ingeniería de sistemas, es muy importante tener un seguimiento y evaluación de sus docentes ya que así podrán mantener un constante fortalecimiento de las competencias pedagógicas y laborales de los mismos en el programa.

La necesidad de implementar y garantizar una óptima evaluación docente, permitirá que los mismos conozcan qué competencias deben fortalecer y así poder ofrecer una mejor calidad de educación a los estudiantes del programa para así posteriormente obtener profesionales graduados con alta calidad.

Se espera que con el aplicativo web, el programa de ingeniería de sistemas, siga en el camino de la acreditación y reiterando la importancia con sus docentes.

1.4. Objetivos

1.4.1. Objetivo General

Brindar al programa de Ingeniería de Sistemas de la universidad Francisco de Paula Santander una solución tecnológica que permita la evaluación y autoevaluación de sus docentes con el fin de mejorar la calidad de sus competencias profesionales.

1.4.2. Objetivos Específicos

- Identificar los nuevos requerimientos y nuevas necesidades de usuarios.
- Analizar las tecnologías, herramientas y metodologías de software para la implementación de la más adecuada, siempre con el uso de las mejores prácticas.
- Diseñar e implementar un sistema web que permita la evaluación y autoevaluación de los docentes del programa de ingeniería de sistemas de la universidad Francisco de Paula Santander.

1.5. Alcance y delimitación

1.5.1. Alcance del software a desarrollar

El sistema web permitirá la evaluación y autoevaluación entre los docentes del programa de ingeniería de sistemas, en el que los docentes subirán las actividades a realizar durante el semestre para que, de acuerdo al cumplimiento de las mismas, otros docentes puedan

evaluar su desempeño y viceversa, con respecto a unos criterios de evaluación anteriormente establecidos por el administrador.

Por parte del administrador del sistema, es decir, el director del programa, podrá gestionar las categorías, sub categorías y criterios de las evaluaciones que vaya a implementar en el sistema web. A su vez podrá consultar información de los docentes y generar informes de las evaluaciones de los mismos.

El sistema web contará con 2 módulos:

- Módulo Administrador (Director del programa)
- Módulo de Docentes

1.5.2. Limitaciones del software a desarrollar

Este proyecto será llevado a cabo en el programa de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander de la ciudad de San José de Cúcuta . La población directa que se ve beneficiada con este proyecto son los docentes y el director de programa. Se enfoca en la evaluación y autoevaluación de los docentes mediante módulo de administrador y módulo de docente, en donde se busca mantener un fortalecimiento continuo de las competencias pedagógicas de los docentes del programa. Se cuenta con una limitación de tiempo de un semestre académico (2023-I) para llevar a cabo el presente proyecto.

1.6. Administración del personal del proyecto

En el presente punto, se da a conocer el personal que participa en el proyecto, se presenta los roles de cada uno de ellos, algunas de sus funciones y responsabilidades. Se trabaja en el mejoramiento de habilidades y destrezas que ayudan al desarrollo del proyecto. También se establecen normas de trabajo con el fin de lograr cada uno de los objetivos propuestos.

1.6.1. Participantes, líderes del proyecto, equipo de software

Director de Línea	
Funciones:	Directora del programa de ingeniería de sistemas
Nombre	Judith del Pilar Rodríguez Tenjo
Cargo	Docente y Directora del programa
Teléfono	
Mail	ingsistemas@ufps.edu.co

Tabla 1. Descripción directora del proyecto SWEAD

Gerente del Proyecto	
Funciones:	Gerente, Desarrollador Full Stack
Nombre	Brayan Stewart Guerrero Ordoñez
Cargo	Gerente del Proyecto, Desarrollador Full Stack
Teléfono	3025972778
Mail	brayanstewartguor@ufps.edu.co

Tabla 2. Descripción gerente del proyecto SWEAD

Coordinadora del Proyecto	
Funciones:	Coordinadora, Desarrolladora Full Stack
Nombre	Natalia Ortiz Armesto
Cargo	Desarrolladora Full Stack
Teléfono	3117636650
Mail	nataliaoa@ufps.edu.co

Tabla 3. Descripción coordinadora del proyecto SWEAD

Equipo de Desarrollo	
Nombre	Brayan Stewart Guerrero Ordoñez
Cargo	Gerente - Desarrollador Full Stack
Teléfono	3025972778
Mail	brayanstewartguor@ufps.edu.co
Nombre	Jesus Uriel Martínez Alcántara
Cargo	Desarrollador Full Stack
Teléfono	+52 5615963752
Mail	jmartineza1908@alumno.ipn.mx
Nombre	Natalia Ortiz Armesto
Cargo	Desarrolladora Full Stack
Teléfono	3117636650
Mail	nataliaoa@ufps.edu.co

Tabla 4. Descripción equipo de desarrollo del proyecto SWEAD



1.6.2. PSP alineado al proyecto de software en desarrollo

PSP (Personal Software Process), es el proceso para que cada integrante del equipo y desarrollador pueda controlar y mejorar su trabajo. A continuación, se establecen los siguientes objetivos para llevar a cabo el proyecto:

- Ser productivos y medir la calidad de nuestras tareas.
- Trabajar en la disciplina para conseguir un mejoramiento continuo
- Mejorar la calidad del proceso de desarrollo empleando las buenas prácticas de la programación
- Medir, estimar, planificar y controlar el proceso de desarrollo de software.

La metodología del PSP emplea los scripts con el fin de especificar los requerimientos, el proceso o tareas y los resultados esperados.

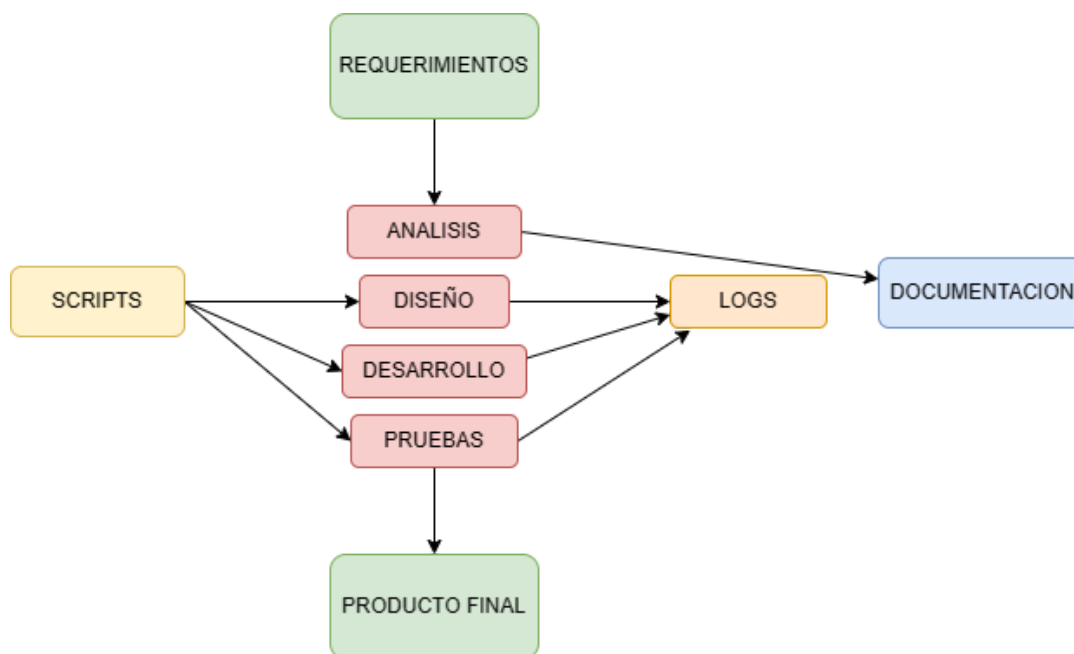


Figura 1. Metodología PSP

En nuestro proyecto aplicaremos este modelo, el cual ayudará a que todos los integrantes mejoren en cuanto a su productividad, trabajando en base a las tareas asignadas y midiendo la calidad y el tiempo empleado en ellas.

1.6.3. Proceso para equipos de software (TSP) alineado al proyecto de software en desarrollo

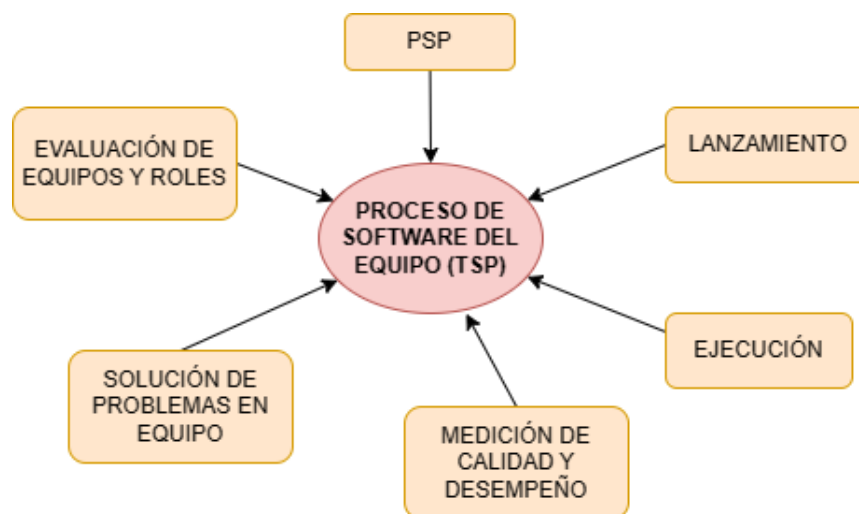


Figura 2. Ilustración proceso de software del equipo (TSP)

Con este modelo se espera el mejoramiento del producto de software, puesto que habrá supervisión en cuanto a la evaluación del desempeño individual y de equipo, así mismo se priorizará la solución de problemas en el ambiente de trabajo puesto que estos pueden llevar a un retraso de tareas y disminución de productividad.

1.7. Opciones para la organización del personal.

Para este paso, se tiene en cuenta diferentes procesos, empezando por el reclutamiento del personal, donde se busca que el equipo de trabajo cuente con las capacidades, habilidades y aptitudes necesarias para llevar a cabo el proyecto.

Así mismo se realiza la asignación de roles y responsabilidades, donde se estudian las capacidades de cada integrante y de acuerdo a ellas, se estudia a qué perfil se ajusta para posteriormente destinar una serie de tareas a cumplir.

1.7.1. Administración de la comunicación

Para la administración de la comunicación dentro del equipo, se toma como referencia la guía de administración de proyectos PMBOK. Aquí se definen una serie de características a la hora de llevar a cabo una comunicación efectiva: escuchar de manera activa y eficaz, formular preguntas, sondear ideas y situaciones para garantizar una mejor comprensión, educar para aumentar el conocimiento del equipo a fin de que sea más eficaz, investigar para identificar o confirmar información, identificar y gestionar expectativas, persuadir a una persona u organización para llevar a cabo una acción, negociar a fin de lograr acuerdos entre partes, que resulten mutuamente aceptables, resolver conflictos para prevenir impactos negativos, resumir, recapitular e identificar las próximas etapas.



1.7.2. Opciones para la estructura de las responsabilidades

Se deben definir las responsabilidades a llevar por cada integrante del equipo y el tiempo estimado para su debido cumplimiento. La calidad de las tareas a realizar debe ser medida y evaluada con el fin de mantener el mínimo margen de errores. Así mismo, se debe contar con un plan de riesgos que respalde cualquier imprevisto que resulte ser una amenaza para el proyecto

2. PROCESOS

GESTIÓN Y COMPRENSIÓN DEL NEGOCIO

2.1. Modelado general de procesos

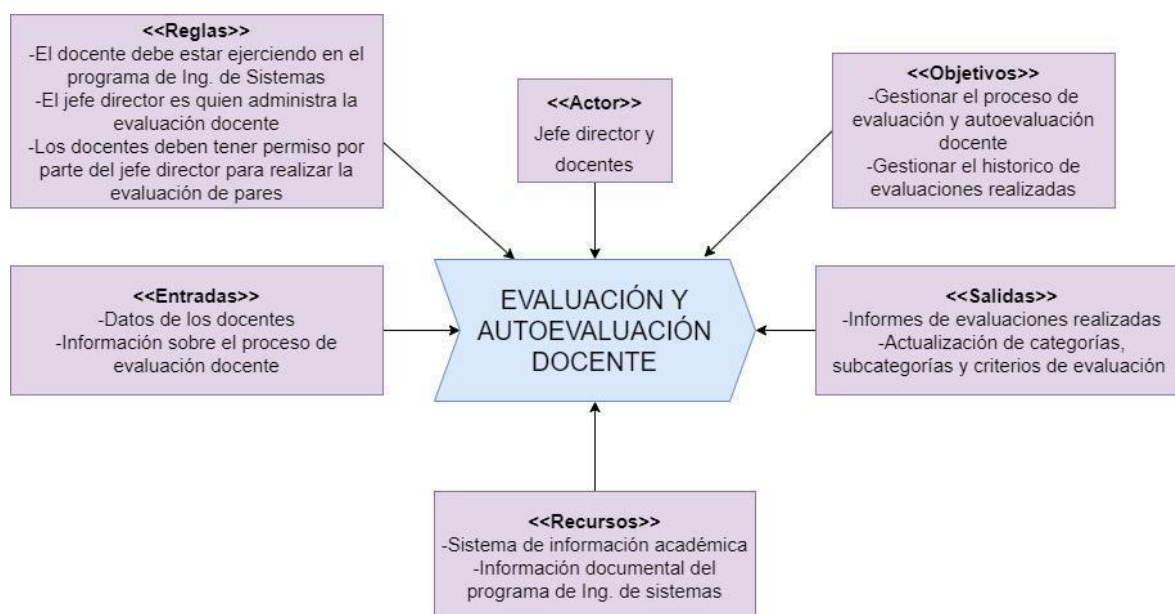


Figura 3. Ilustración modelado general de los procesos del proyecto SWEAD

En el proceso de evaluación docente, los actores son el jefe director, que es quien gestiona la evaluación, y los docentes que son los evaluados y evaluadores de sus pares



académicos. Tiene como objetivo gestionar los procesos de evaluación docente y el historial de evaluaciones realizadas para fomentar la mejora continua de la docencia.

Para ello, se utilizan unas entradas que son los datos de los docentes y la información sobre el proceso de evaluación docente (categorías, subcategorías y criterios de evaluación, propósito de la evaluación docente, entre otros) las cuales generan unas salidas que se resumen en los informes de las evaluaciones y la actualización de la forma de evaluar, si se considera necesario.

Así mismo, se utilizan unos recursos que, en este caso, son el sistema de información académica y la información documental proporcionada por parte del programa de ingeniería de sistemas.

2.2. Modelo o ciclo de vida del software aplicado al proyecto y descripción de sus fases

Modelo Ciclo de vida en V

Uno de los grandes problemas del modelo en cascada es que solo se pasa a la siguiente fase si se completa la anterior y no se puede volver atrás si hay errores en etapas posteriores. Así, el modelo en V da más opciones de evaluación del software en cada etapa.

En cada fase se crea la planificación de las pruebas y los casos de pruebas para verificar y validar el producto en función de los requisitos de la misma. De esta manera, verificación y validación van en paralelo.

Una de las principales ventajas que ofrece este modelo es que no hace falta que los requisitos estén totalmente definidos al inicio del desarrollo, sino que se pueden ir refinando en cada una de las planificaciones y pruebas.(Sommerville, 2005: 475-480)

Las siguientes son las actividades de que se usan en cada uno de los procesos:

- **Planificación:** Antes de empezar un proyecto de desarrollo de un sistema de información, es necesario hacer ciertas tareas que influirá decisivamente en el éxito del mismo.
- **Análisis:** hay que averiguar qué es exactamente lo que tiene que hacer el software.
- **Diseño:** El diseño es una etapa compleja y su proceso debe realizarse de manera iterativa.
- **Implementación:** Esta elección dependerá tanto de las decisiones de diseño tomadas como del entorno en el que el software deba funcionar.
- **Pruebas:** Una vez se ha generado el código, comienzan las pruebas de software del sistema que se ha desarrollado, de acuerdo con Pressman el proceso de pruebas se

centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado y los procesos funcionales, es decir, la realización de las pruebas para la detección de errores.

- **Uso y mantenimiento:** Esta es una de las fases más importantes del ciclo de vida de desarrollo del software. Puesto que el software ni se rompe ni se desgasta con el uso, su mantenimiento incluye tres puntos diferenciados:
 - Eliminar los defectos detectados durante su vida útil (mantenimiento correctivo).
 - Adaptarlo a nuevas necesidades (mantenimiento adaptativo).
 - Añadirle nuevas funcionalidades (mantenimiento perfectivo).

(Sommerville, 2005:62-163)

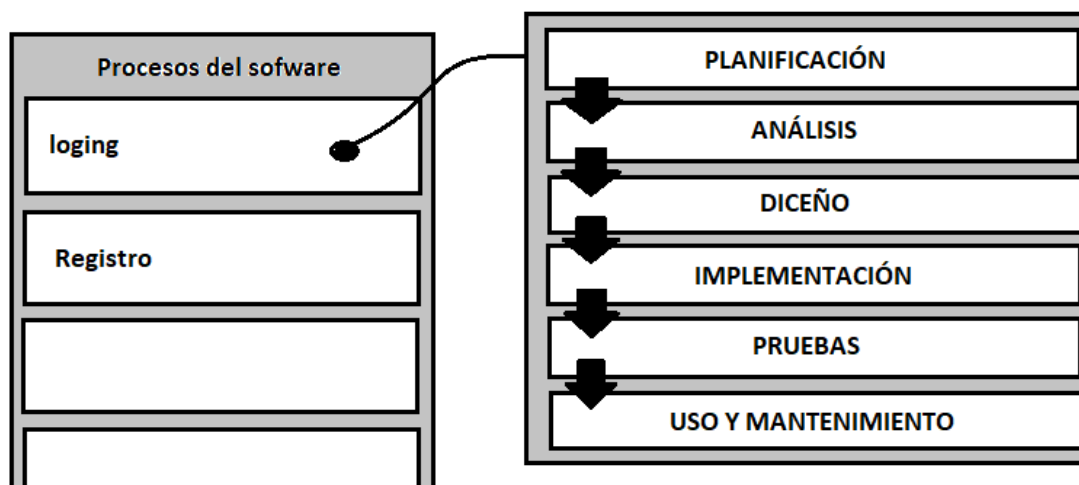


Figura 4. Modelo de ciclo de vida en V. Elaboración propia

2.3. Tecnología del proceso

2.3.1. Descripción de los procesos tecnológicos que se llevaron a cabo

Tecnologías utilizadas en el frontend:

- **HTML:** Es el lenguaje de marcado utilizado para crear páginas web. HTML se utiliza para estructurar y dar formato a la información que se muestra en una página web, y se utiliza en conjunto con otros lenguajes como CSS y JavaScript para crear páginas web dinámicas y atractivas.

- **CSS:** Es un lenguaje de diseño utilizado para dar estilo a las páginas web. CSS se utiliza para controlar el aspecto visual de una página web, incluyendo el diseño, el color, la tipografía y otros aspectos de la presentación.
- **JavaScript:** Es un lenguaje de programación utilizado para crear páginas web interactivas y dinámicas. JavaScript se utiliza para agregar interactividad a una página web, como animaciones, efectos visuales y validaciones de formularios.
- **Bootstrap:** Es un framework de diseño web que facilita la creación de páginas web responsivas y móviles. Bootstrap proporciona una serie de componentes de diseño predefinidos, como botones, formularios y carruseles, que se pueden utilizar para crear páginas web de forma rápida y sencilla.

Tecnologías utilizadas en el backend:

- **Java:** Es un lenguaje de programación orientado a objetos utilizado en el desarrollo de aplicaciones empresariales. Java se utiliza para desarrollar aplicaciones web, aplicaciones de escritorio y aplicaciones móviles.
- **Spring Boot:** Es un framework de Java utilizado para crear aplicaciones web y servicios web. Spring Boot proporciona una serie de características y herramientas que hacen que sea fácil crear aplicaciones web escalables y mantenibles.



- **MySQL:** Es un sistema de gestión de bases de datos relacionales utilizado para almacenar y administrar datos. MySQL es un software de código abierto que se utiliza para crear bases de datos para aplicaciones web y empresariales.

2.4 Fusión de producto y proceso

El producto software desarrollado va dirigido principalmente al programa de Ingeniería de sistemas de la Universidad Francisco de Paula Santander. El proceso de realización y ejecución del proyecto constó de diversas etapas como lo fueron planificación, análisis, diseño, desarrollo o implementación, pruebas y despliegue. El producto final o producto destino es el sistema web para la evaluación y autoevaluación docente que cumplirá con los diseños preestablecidos y cumplirá las pautas o funcionalidades que surjan de las necesidades del cliente permitiendo así adaptabilidad a los cambios a corto y mediano plazo.

2.5 Descomposición del proceso

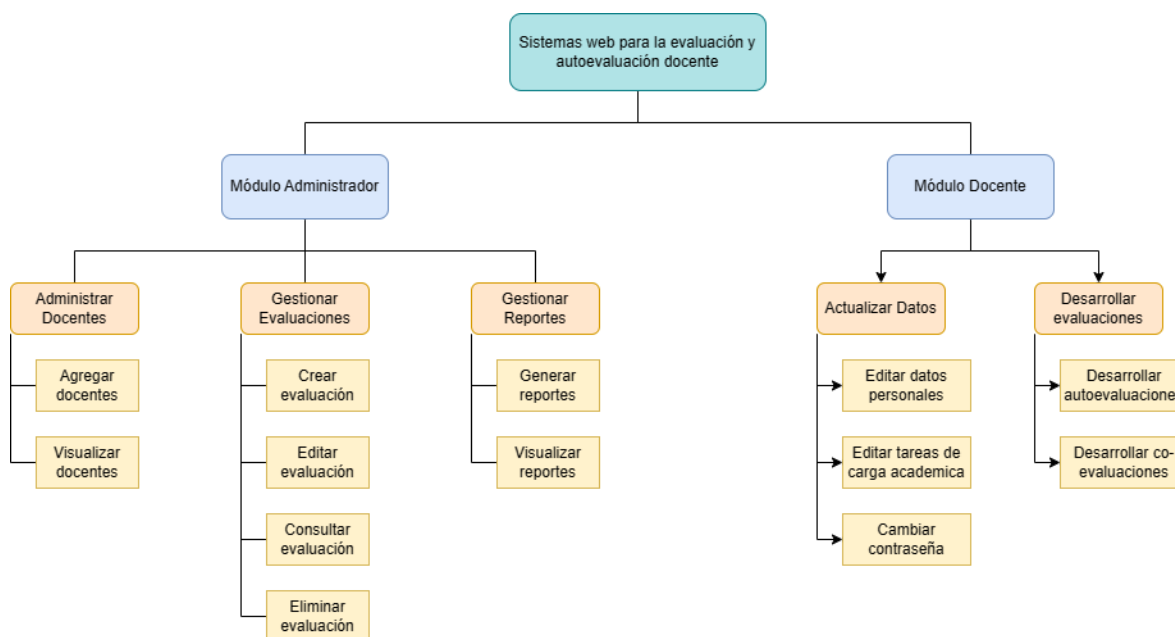


Figura 5. Descomposición de los procesos del problema a desarrollar en el proyecto SWEAD

La anterior figura representa el desglose de las funcionalidades que se realizan en cada uno de los módulos. En ese caso, para el módulo de administrador, se encarga de realizar la gestión de los docentes mediante dos funcionalidades: agregar a docentes con información parcial de estos, ya que mediante una lista de correos de los mismos, se envían las solicitudes de registro a los usuarios docente; aparte, el administrador puede visualizar la información ya diligenciada de cada docente. Similarmente, el administrador puede

gestionar las evaluaciones mediante cuatro funcionalidades, las cuales son crear, editar, consultar y eliminar las evaluaciones dentro del sistema. Además, el administrador puede realizar la gestión de los reportes. Esto le permitirá generar reportes de las diferentes evaluaciones y visualizar los resultados de cada docente.

Por otro lado, en el módulo de docentes, para dar solución al problema planteado, se presentan dos funcionalidades, el poder realizar la actualización de sus datos personales y de las tareas correspondientes a su carga académica integral. También pueden desarrollar las evaluaciones que el administrador diseñe.

CALENDARIZACIÓN DEL PROYECTO

Id	Nombre de tarea	Predecesoras
1	SISTEMA WEB PARA LA EVALUACIÓN Y AUTOEVALUACIÓN DE DOCENTES	
2	ITERACIÓN 1 - MÓDULO ADMINISTRADOR	
3	Análisis	
4	Reunión con la clienta	
5	Identificar los requerimientos del módulo administrador	
6	Descripción de los requerimientos del módulo administrador	5

7	Revisión de los requerimientos del módulo administrador	6
8	Diseño	
9	Realizar los diagramas de secuencia para el módulo administrador	7FC+1 día
10	Realización y análisis del mockup del módulo administrador	9
11	Aceptación del mockup por parte del cliente	10
12	Desarrollo	
13	Realizar la base de datos físico del módulo administrador	9
14	Desarrollo del frontend de módulo administrador	13
15	Desarrollo del backend de módulo administrador	13
16	Comprobar que el módulo administrativo esté funcionando correctamente con base a los requerimientos	14;15
17	Comprobar persistencia en los datos, rendimiento, seguridad y compatibilidad del módulo administrador	14;15
18	Pruebas y mejoras	
19	Elaborar y ejecutar pruebas unitarias para el módulo administrador	17
20	Corregir los errores en las pruebas unitarias del módulo administrador	19
21	Elaborar y ejecutar pruebas de aceptación del cliente para el módulo administrador	17

22	Corregir e integrar los ajustes establecidos por el cliente en las pruebas de aceptación.	21
23	Aceptación del módulo administrador por el cliente	20;22
24	ITERACIÓN 2 - MÓDULO DE DOCENTES	
25	Análisis	
26	Reunión con la clienta	4FF
27	Identificar los requerimientos del módulo de docentes	5FF
28	Descripción de los requerimientos del módulo de docentes	27
29	Revisión de los requerimientos del módulo de docentes	28
30	Diseño	
31	Realizar los diagramas de secuencia para el módulo de docentes	29FC+1 día
32	Realización y análisis del mockup del módulo de docentes	31
33	Aceptación del mockup por parte del cliente	32
34	Desarrollo	
35	Realizar la base de datos físico del módulo de docentes	31
36	Desarrollo del frontend del módulo de docentes	35
37	Desarrollo del backend del módulo de docentes	35
38	Comprobar que el módulo de docentes esté funcionando	36;37

	correctamente con base a los requerimientos	
39	Comprobar persistencia en los datos, rendimiento, seguridad y compatibilidad del módulo de docentes	36;37
40	Pruebas y mejoras	
41	Elaborar y ejecutar pruebas unitarias para el módulo de docentes	39
42	Corregir los errores en las pruebas unitarias del módulo de docentes	41
43	Elaborar y ejecutar pruebas de aceptación del cliente para el módulo de docentes	39
44	Corregir e integrar los ajustes establecidos por el cliente en las pruebas de aceptación.	43
45	Aceptación del módulo de docentes por el cliente	42;44

Tabla 1. Calendario de actividades del proyecto SWEAD

En el siguiente enlace se podrá visualizar el calendario de manera más detallada:

[Cronograma de actividades SWEAD](#)

2.6. Estimación del proyecto

A continuación, se puede apreciar la medición y estimación del proyecto basado en puntos de fusión:

Tipo de componente	Complejidad baja	Complejidad media	Complejidad alta
Entrada externa (EI) (pantallas entradas para ingresos de registro, actualización o eliminación)	3	4	6
Salida Externa (EO) (informes, gráficos, listados, reportes, entre otros)	4	5	7
Consulta externa (EQ) (búsqueda, recuperar y mostrar datos)	3	4	6
Archivo lógico interno (ILF) (Tablas)	7	10	15
Archivos Interfaz Externa (IEF) (otros sistemas)	5	7	10

Tabla 2. Tipos de componentes de los puntos de fusión

Componente (Requerimiento o HU)			Tipo de componente	Nivel de complejidad	# Transacciones	EI	EQ	EO
RF1	Iniciar sesión		EQ	Bajo	1	0	1	0
RF2	El administrador puede gestionar el proceso de inscripción					0	0	0
	El administrador puede gestionar el proceso de inscripción							
	El administrador puede gestionar el proceso de inscripción							
	El administrador puede gestionar el proceso de inscripción							
	RF2.1	Abrir inscripción	EI	Bajo	2	1	0	0
	RF2.2	Consultar docentes inscritos	EQ	Bajo	1	0	1	0
	RF2.3	Cerrar inscripción	EI	Bajo	2	1	0	0
RF3	El administrador puede gestionar permisos					0	0	0
	RF3.1	Habilitar permiso	EI	Bajo	2	1	0	0
	RF3.3	Eliminar permiso	EI	Bajo	2	1	0	0
RF4	El administrador puede consultar un docente		EQ	Bajo	1	0	1	0
RF5	El administrador puede generar informes		EO	Bajo	2	0	0	1
RF6	El administrador podrá gestionar las evaluaciones					0	0	0
	RF6.1	Crear evaluación	EI	Bajo	2	1	0	0
	RF6.2	Consultar evaluación	EQ	Bajo	1	0	1	0
	RF6.3	Actualizar evaluación	EI	Bajo	2	1	0	0
RF7	El administrador podrá gestionar categorías y subcategorías de evaluación					0	0	0

	RF6.1	Crear categoría/subcategoría	EI	Bajo	2	1	0	0
	RF6.2	Consultar categoría/subcategoría	EQ	Bajo	1	0	1	0
	RF6.3	Actualizar categoría/subcategoría	EI	Bajo	2	1	0	0
RF8	El administrador podrá gestionar los criterios de evaluación					0	0	0
	RF6.1	Crear criterio	EI	Bajo	2	1	0	0
	RF6.2	Consultar criterio	EQ	Bajo	1	0	1	0
	RF6.3	Actualizar criterio	EI	Bajo	2	1	0	0
RF9	El administrador podrá gestionar las fechas hábiles para realizar la evaluación docente					0	0	0
	RF6.1	Crear período de fechas	EI	Bajo	2	1	0	0
	RF6.3	Actualizar periodo de fechas	EI	Bajo	2	1	0	0
RF10	Los docentes podrán evaluar a otro docente		EI	Bajo	2	1	0	0
RF11	El docente podrá autoevaluarse		EI	Bajo	2	1	0	0
RF12	El docente podrá gestionar las actividades de carga académica integral							
	RF6.1	Crear actividad	EI	Bajo	2	1	0	0
	RF6.2	Consultar actividad	EQ	Bajo	1	0	1	0
	RF6.3	Actualizar actividad	EI	Bajo	2	1	0	0
					Total	16	7	1

Tabla 3. Estimación Basado en Puntos de Fusión del Software SWEAD



PUNTOS DE FUNCIÓN SIN AJUSTAR						
Componente			COMPLEJIDAD			
			BAJO	MEDIO	ALTO	total
EI			48	0	0	48
EO			4	0	0	4
EQ			21	0	0	21
ILF			0	0	0	0
IEF			0	0	0	0
TOTAL			73			
		PFSA	73			

Tabla 4. Puntos de Fusión sin Ajustar

TABLA DE FACTORES DE AJUSTE DE COMPLEJIDAD TÉCNICA	
Factor	Puntaje (nivel de Influencia)
1. Comunicación de datos	3
2. Procesamiento distribuido	0
3. Perfomance (desempeño)	4
4. Configuración del equipamiento	3
5. Volumen de transacciones	4

6. Entrada de datos on-line	4
7. Interfase con el usuario	5
8. Actualización on-line	0
9. Procesamiento complejo	2
10. Reusabilidad	4
11. Facilidad de implementación	3
12. Facilidad de operación	3
13. Puesto múltiples	0
14. Facilidad de cambio	5
Total	40

Tabla 5. Tabla de Factores de Ajuste de Complejidad Técnica.


Métrica			
Fase	%	Thoras	Tdías
Análisis	20	77	10
Diseño	25	96	12
Construcción	40	153	19
Pruebas y mejoras	15	57	7
Total	48		

Tabla 6. Métrica de cada etapa del proyecto SWEAD

2.7. Recursos humanos, tecnológicos, recursos de software reutilizables, recursos del entorno

2.7.1. Recursos Humanos

En la siguiente tabla se muestran todos los integrantes del equipo de desarrollo valorados como recurso humano en el proyecto.

RECURSOS HUMANOS DEL SOFTWARE SWEAD			
NOMBRE	CARGO	ESTUDIO	IMAGEN
Brayan Stewart Guerrero Ordoñez	Gerente del Proyecto, Desarrollador Fullstack	Programa de Sistemas de la Universidad Francisco de Paula Santander	

Natalia Ortiz Armesto	Coordinadora del Proyecto, Desarrollador Fullstack	Programa de Sistemas de la Universidad Francisco de Paula Santander	
Jesús Uriel Martínez Alcántara	Desarrollador Fullstack	Programa de Sistemas de la Universidad Francisco de Paula Santander	

Tabla 7. Recursos humanos del proyecto SWEAD

2.7.2. Recursos tecnológicos

En la siguiente tabla se listan los recursos tecnológicos en el proyecto SWEAD:

RECURSOS TECNOLÓGICOS	
NOMBRE	CANTIDAD
Computador portatil	3
Monitor	3
Router	3
Licencia Figma	1
Licencia Microsoft Project	1
Visual Studio Code	3
IntelliJ IDEA	3

Tabla 8. Recursos tecnológicos del proyecto SWEAD

2.7.3. Recursos de software reutilizables

En la siguiente tabla se listan los recursos de software reutilizables en el proyecto SWEAD:

Recursos software reutilizables	
Componentes ya desarrollados	Componentes del software que ya han sido validados, se pueden reutilizar e implementar en esta siguiente fase del proyecto.
Componentes ya experimentados	Especificaciones, diseños, código o datos de prueba existentes que ya han sido desarrollados en proyectos anteriores.
Componentes nuevos	Componentes que el equipo de software requiere construir específicamente para el proyecto.

Tabla 9. Recursos de software reutilizables del proyecto SWEAD.

2.7.4. Recursos del entorno

En la siguiente tabla se listan los recursos del entorno en el proyecto SWEAD:

CARACTERÍSTICAS DEL ENTORNO	
Requisitos del servidor	
Hardware	<p>Sistema operativo: 64 bits</p> <p>Procesador: 2.5 GHz o más rápido de 32 bits(x86)</p> <p>Memoria: 4 GB de RAM (o más)</p> <p>Espacio en disco: 8 GB</p>

software	Windows
Requisitos del sistema (cliente)	
Hardware	Sistema operativo: 32 bits (mínimo) Procesador: 1 GHz o más rápido de 32 bits(x86) Memoria: 1 GB de RAM (o más) Espacio en disco: 1 GB
Software	Linux, Windows, Mac OS.
Otros	Conexión a internet
Condiciones técnicas mínimas	
Plataforma	Navegador Web: Chrome, Mozilla Firefox, Microsoft Edge, etc.
Sistema operativo	Windows, Linux, Mac OS.
Base de datos	MySQL

Dispositivos	Escritorio, mediante navegador web Móviles, mediante navegador web
--------------	---

Tabla 10. Recursos del entorno del proyecto SWEAD.

2.8. Técnicas de descomposición aplicadas al proyecto en desarrollo

2.8.1. Tamaño del software

El proyecto SWEAD se enfocará en la estimación basada en el tamaño del software, en la que mediante las métricas basadas en puntos de fusión (PF), que es una manera indirecta de determinar qué tan grande es el software que se está desarrollando, adicional a eso también se hará la medición mediante la técnica basada en líneas de código (LDC) que se determina mediante el tamaño de las líneas de código.

2.8.2. Basada en el problema

Para el proyecto SWEAD se proponen las siguientes métricas entendiendo que cada concepto del proyecto es adecuado y responder a la necesidad del correcto funcionamiento de toda la aplicación.

Métrica de Usabilidad: La usabilidad es la intersección entre la efectividad, la eficiencia y la satisfacción en el contexto del uso. En otras palabras, consiste en lograr finalizar las tareas y en la satisfacción de los usuarios mientras las realizan.

Métricas de Usabilidad	
Nombre:	Seguridad
Propósito:	Medir impacto de la seguridad y accesos de usuarios no autorizados.
Método de aplicación:	Contar con la modularidad y arquitectura del aplicativo.
Medición	X = medio
Interpretación	Calificación de la métrica de manera cuantitativa: La estructura y la arquitectura (Excelente, sobresaliente, medio, regular y bajo).
Tipo de escala:	Escala absoluta o total.
Fuente de medición:	Especificación de requerimientos, diseño, informe de la revisión.
ISO/IEC 12207 SLCP:	Verificación, y revisión conjunta.
Audiencia:	Desarrolladores y requeridores.

Tabla 11. Métrica de usabilidad del proyecto SWEAD

Métricas de Madurez: La simpleza al momento de su implementación se basa en Reglas, Conceptos, Atributos y Niveles de Descomposición. Estos elementos son fáciles de encontrar y contabilizar dentro del ámbito de desarrollo del sistema.

Métricas de Madurez	
Nombre:	Funciones evidentes y suficientes.
Propósito:	Qué proporción de las funciones del sistema son evidentes al usuario.
Método de aplicación:	Contar todas las funciones que son evidentes al usuario y comparar con su totalidad de funciones.
Medición	$X = \frac{A}{B}$ <p>A = Cantidad de casos de prueba en el plan B = Cantidad de casos de prueba requeridos.</p>
Interpretación	Entre más X sea positivo, mejor será la eficiencia.
Tipo de escala:	Escala absoluta o total.
Fuente de medición:	Plan de pruebas y especificación de los requisitos, informe de la revisión.
ISO/IEC 12207 SLCP:	Aseguramiento de la calidad, resolución de problemas y la validación de estas.
Audiencia:	Desarrolladores y equipos encargados del mantenimiento.

Tabla 12. Métrica de madurez del proyecto SWEAD

Métricas de Adecuidad

Métricas de Adecuidad	
Nombre:	Cumplimiento de la implementación de todos los requerimientos funcionales.
Propósito:	Saber si se implementan todos los requerimientos funcionales planteados inicialmente, si no, qué tanto avanzó.
Método de aplicación:	Saber cuántos requisitos funcionales se plantearon en el análisis y cuantos se ejecutaron.
Medición	$X = 1 - \frac{A}{B} * 100\%$ <p>B = Cantidad de funciones descritas en la especificación de requisitos. A = Cantidad de funciones faltantes.</p>
Interpretación	De acuerdo a los valores anteriores, entre más cerca de 1, se llega a la totalidad de la funcionalidad del proyecto.
Tipo de escala:	Tipo de escala absoluta, para contar los elementos y solamente es posible una representación: El recuento real de elementos, siendo en nuestro caso los requerimientos funcionales.
Fuente de medición:	Se revisa el análisis y diseño, la especificación de requisitos y el código fuente.

ISO/IEC 12207 SLCP:	Verificación, validación y revisión conjunta.
Audiencia:	Desarrolladores, clientes, equipos de testeo.

Tabla 13. Métrica de adecuación del proyecto SWEAD

2.8.3. Basada en LDC

En la tabla siguiente se muestran las estimaciones de los costos de desarrollo de cada caso de uso en número de líneas de código.

#	Caso de uso	Optimista	Más Probable	Pesimista	Esperado	\$ Línea	Línea/Día	Costo	Días
1	Iniciar Sesión Usuario	80	150	200	146,67	\$2.927,73	36,67	\$429.400	4
2	Gestionar proceso de inscripción de docentes	240	260	300	263,33	\$3.261,27	32,92	\$858.800	8
3	Gestionar permisos para la inscripción de docentes	300	350	450	358,33	\$2.396,65	44,79	\$858.800	8
4	Gestionar evaluación docente	350	400	500	408,33	\$3.154,78	34,03	\$1.288.200	12
5	Gestionar categorías y subcategorías de la evaluación docente	250	300	400	308,33	\$4.177,95	25,69	\$1.288.200	12

6	Gestionar criterios de evaluación docente	200	250	350	258,33	\$4.986,58	21,53	\$1.288.200	12
7	Generar informes de los docentes evaluados	200	250	300	250,00	\$4.294,00	25,00	\$1.073.500	10
8	Realizar proceso de evaluación de docentes y autoevaluación	250	300	400	308,33	\$3.481,62	30,83	\$1.073.500	10
9	Gestionar actividades de carga académica integral de los docentes	300	350	450	358,33	\$2.396,65	44,79	\$858.800	8
Total		2.170	2.610	3.350	2.660,00	\$28.680,57	251,46	\$8.158.600	84

Tabla 14. Estimación por líneas de código del proyecto SWEAD.

2.8.4. Basada en PF

Estimación Basado en Puntos de Función del Software

Tipo de componente	Complejidad baja	Complejidad media	Complejidad alta
Entrada externa (EI) (pantallas entradas para ingresos de registro, actualización o eliminación)	3	4	6
Salida Externa (EO) (informes, gráficos, listados, reportes, entre otros)	4	5	7
Consulta externa (EQ) (búsqueda, recuperar y mostrar datos)	3	4	6
Archivo lógico interno (ILF) (Tablas)	7	10	15
Archivos Interfaz Externa (IEF) (otros sistemas)	5	7	10

Tabla 15. Tipos de componentes de los puntos de fusión



Componente (Requerimiento o HU)			Tipo de componente	Nivel de complejidad	# Transacciones	EI	EQ	EO
RF1	Iniciar sesión		EQ	Bajo	1	0	1	0
RF2	El administrador puede gestionar el proceso de inscripción							
	El administrador puede gestionar el proceso de inscripción							
	El administrador puede gestionar el proceso de inscripción							
	El administrador puede gestionar el proceso de inscripción					0	0	0
	RF2.1	Abrir inscripción	EI	Bajo	2	1	0	0
	RF2.2	Consultar docentes inscritos	EQ	Bajo	1	0	1	0
	RF2.3	Cerrar inscripción	EI	Bajo	2	1	0	0
RF3	El administrador puede gestionar permisos					0	0	0
	RF3.1	Habilitar permiso	EI	Bajo	2	1	0	0
	RF3.3	Eliminar permiso	EI	Bajo	2	1	0	0
RF4	El administrador puede consultar un docente		EQ	Bajo	1	0	1	0
RF5	El administrador puede generar informes		EO	Bajo	2	0	0	1
RF6	El administrador podrá gestionar las evaluaciones					0	0	0
	RF6.1	Crear evaluación	EI	Bajo	2	1	0	0
	RF6.2	Consultar evaluación	EQ	Bajo	1	0	1	0
	RF6.3	Actualizar evaluación	EI	Bajo	2	1	0	0



RF7	El administrador podrá gestionar categorías y subcategorías de evaluación					0	0	0
	RF6.1	Crear categoría/subcategoría	EI	Bajo	2	1	0	0
	RF6.2	Consultar categoría/subcategoría	EQ	Bajo	1	0	1	0
	RF6.3	Actualizar categoría/subcategoría	EI	Bajo	2	1	0	0
RF8	El administrador podrá gestionar los criterios de evaluación					0	0	0
	RF6.1	Crear criterio	EI	Bajo	2	1	0	0
	RF6.2	Consultar criterio	EQ	Bajo	1	0	1	0
	RF6.3	Actualizar criterio	EI	Bajo	2	1	0	0
RF9	El administrador podrá gestionar las fechas hábiles para realizar la evaluación docente					0	0	0
	RF6.1	Crear período de fechas	EI	Bajo	2	1	0	0
	RF6.3	Actualizar periodo de fechas	EI	Bajo	2	1	0	0
RF10	Los docentes podrán evaluar a otro docente		EI	Bajo	2	1	0	0
RF11	El docente podrá autoevaluarse		EI	Bajo	2	1	0	0
RF12	El docente podrá gestionar las actividades de carga académica integral							
	RF6.1	Crear actividad	EI	Bajo	2	1	0	0
	RF6.2	Consultar actividad	EQ	Bajo	1	0	1	0
	RF6.3	Actualizar actividad	EI	Bajo	2	1	0	0
					Total	16	7	1

Tabla 16. Estimación Basado en Puntos de Función del Software SWEAD.

PUNTOS DE FUNCIÓN SIN AJUSTAR					
Componente		COMPLEJIDAD			total
		BAJO	MEDIO	ALTO	
EI		48	0	0	48
EO		4	0	0	4
EQ		21	0	0	21
ILF		0	0	0	0
IEF		0	0	0	0
TOTAL		73			
		PFSA	73		

Tabla 17. Puntos de Fusión sin Ajustar.

TABLA DE FACTORES DE AJUSTE DE COMPLEJIDAD TÉCNICA

Factor	Puntaje (nivel de Influencia)
1. Comunicación de datos	3
2. Procesamiento distribuido	0
3. Performance (desempeño)	4
4. Configuración del equipamiento	3
5. Volumen de transacciones	4
6. Entrada de datos <i>on-line</i>	4
7. Interfase con el usuario	5

8. Actualización <i>on-line</i>	0
9. Procesamiento complejo	2
10. Reusabilidad	4
11. Facilidad de implementación	3
12. Facilidad de operación	3
13. Puesto múltiples	0
14. Facilidad de cambio	5
Total	40

Tabla 18. Tabla de Factores de Ajuste de Complejidad Técnica.

Puntos de Función Ajustado (PFA)	76,65
---	-------

Tabla 19. Puntos de función ajustado (PFA)

Horas lenguaje de Programación 4 Generación	6
--	---

Tabla 20. Horas de lenguaje de programación 4ta generación

PFA	459,9
	57,4875

Tabla 21. PFA

1 mes	2,3953125
Integrantes	3
H/H	153,3

Métrica			
Fase	%	Thoras	Tdías
Análisis	20	77	10
Diseño	25	96	12
Construcción	40	153	19
Pruebas y mejoras	15	57	7
Total	48		

Tabla 22. Métrica de cada etapa del proyecto SWEAD

2.8.5. Basada en casos de uso

Lista de casos de uso:

#	Casos de Uso	Número Transacciones
1	Iniciar Sesión Usuario	1
2	Gestionar proceso de inscripción de docentes	5
3	Gestionar permisos para la inscripción de docentes	4
4	Gestionar evaluación docente	5
5	Gestionar categorías y subcategorías de la evaluación docente	5
6	Gestionar criterios de evaluación docente	5

7	Generar informes de los docentes evaluados	2
8	Realizar proceso de evaluación de docentes y autoevaluación	8
9	Gestionar actividades de carga académica integral de los docentes	5

Tipos de Caso de Uso	Descripción		# Casos
Simple	1	3	2
Intermedio	4	7	6
Complejo	8	8	1

Tabla 1: Factores de Peso de los Actores sin Ajustar.

Tipo de Actor	Descripción	Factor de Peso	Número de Actores	Resultado
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación(API, Application Programming Interface)	1	0	0
Intermedio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0

Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	2	<u>6</u>
Factores de peso de los actores sin ajustar (UAW)				6

Tabla 23: Factores de Peso de los Actores sin Ajustar.

Tabla 2: Factor de Peso en los Casos de Uso sin Ajustar

Tipo de Casos de Uso	Descripción	Factor de peso	Número de Casos de Uso	Resultado
Simple	1 a 3 Transacciones	5	2	10
Intermedio	4 a 7 Transacciones	10	6	60
Complejo	más de 7 Transacciones	15	1	<u>15</u>
Factores de peso en los casos de uso sin ajustar (UUCW)				85
Puntos de Casos de Uso sin Ajustar (UUCP)				91

Tabla 24: Factor de Peso en los Casos de Uso sin Ajustar

Tabla 3: Peso de los Factores de Complejidad Técnica.

Número de Factor	Descripción	Escala de Asignación	Peso	Valor	Factor	Comentario
T1	Sistema Distribuido	0=no importante 5=esencial	2	1	2	El sistema es Web, por lo que posee cierto nivel de distribución
T2	Objetivos de Desempeño o Tiempo de Respuesta	0=no importante 5=esencial	1	2	2	El tiempo de respuesta es adecuado
T3	Eficiencia Usuario Final (online)	0=no importante 5=esencial	1	3	3	Los roles necesitan estar relacionados con el sistema para su mejor funcionamiento.
T4	Procesamiento Interno Complejo	0=no importante 5=esencial	1	4	4	El sistema proporciona datos lógicos que necesitan un nivel medio de conocimiento
T5	Código Debe Ser Reusable	0=no importante 5=esencial	1	4	4	Será orientado a objetos y podrá ser usado por sistemas similares.
T6	Facilidad de Instalación	0=no importante 5=esencial	0,5	0	0	Por ser un sistema Web la complejidad de instalación es mínima.
T7	Facilidad de Uso	0=no importante 5=esencial	0,5	5	2,5	El sistema debe ser fácil de usar
T8	Portabilidad	0=no importante 5=esencial	2	3	6	El sistema se encuentra diseñado para que sea usado en otro contexto

						similar
T9	Facilidad de Cambio	0=no importante 5=esencial	1	2	2	El sistema se encuentra estructurado para que los cambios realizados afecten lo menos posible las funcionalidades del sistema.
T10	Concurrencia	0=no importante 5=esencial	1	3	3	La concurrencia es tratada con suma importancia.
T11	Incluye Características Especiales de Seguridad	0=no importante 5=esencial	1	5	5	El sistema sólo permite que un usuario realice las funcionalidades correspondientes a su rol dentro del sitio.
T12	Provee Acceso Directo a Terceros	0=no importante 5=esencial	1	5	5	La aplicación es accesible a cualquier usuario.
T13	Se Requieren Ayudas Especiales de Entrenamiento de Usuarios	0=no importante 5=esencial	1	4	4	Se hace necesario el entrenamiento de los usuarios finales, pero se debe incluir un manual de usuario para garantizar la correcta usabilidad de dicho sistema.
Total Factores Técnicos					42,5	
Factor de Complejidad Técnica (TCF)		.06 + (.01*Factor Técnico)			1,025	

Tabla 25: Peso de los Factores de Complejidad Técnica

Tabla 4: Peso de los Factores Ambientales.

Número del Factor	Descripción	Escala de 0 a 5	Peso	Valor	Factor	Comentario	Experiencia/Estabilidad
F1	Familiaridad con un Proceso Definido	0 = sin experiencia, 3=media, 5=experto	1,5	3	4,5	Se está familiarizado con el modelo del proyecto, pero la experiencia en el modelado es media.	0
F2	Experiencia en el Dominio de Aplicación	0 = sin experiencia, 3=media, 5=experto	0,5	4	2	Se necesita de un equipo capacitado y de conocimientos suficientes	1
F3	Experiencia en Orientación a Objetos	0 = sin experiencia, 3=media, 5=experto	1	4	4	Se considera cierto grado de experiencia en la programación orientada a objetos (POO)	0
F4	Capacidad de Liderazgo de Analistas	0 = sin experiencia, 3=media, 5=experto	0,5	3	1,5	Los analistas que integran el equipo de trabajo poseen capacidad media.	1
F5	Motivación	0=sin, 3=media, 5=alta	1	5	5	Alta	0
F6	Requerimientos Estables	0=extremadamente inestable, 5=no cambian	2	4	8	El sistema brinda las funcionalidades esenciales que dan cumplimiento a los objetivos que iniciaron su realización.	0

F7	Miembros a Tiempo Parcial	0=tiempo parcial, 5=tiempo completo	1	2	2	Se trabajará a tiempo parcial.	0
F8	Dificultad del Lenguaje de Programación	0=fácil, 3=medio,5=difícil	1	2	<u>2</u>	se considera una dificultad media su empleo.	<u>0</u>
Total Factores Ambientales					29		2
Factor Ambiente (EF)		1.4 + (-0.03*Factor Ambiental)			0,53		

Tabla 26: Peso de los Factores Ambientales

Puntos de Casos de Uso Ajustados (UCP)				49,43575
Horas Persona por Punto de Caso de Uso				20
Horas Persona Estimación Inicial				988,72

Tabla 5. Distribución Real del Esfuerzo.

Actividad	Porcentaje	Valor
Análisis	25,00%	617,95
Diseño	20,00%	494,36
Programación	40,00%	988,72
Pruebas y Mejoras	15,00%	370,77
Total	100,00%	2.471,79

Tabla 27. Distribución Real del Esfuerzo

Cálculo del Esfuerzo Total (ETotal):	2.471,79	horas/hombre
Cálculo del Tiempo de Desarrollo (TDesarrollo):	1.235,89	horas
Cálculo del Tiempo de Desarrollo (TDesarrollo):	154,49	días
Cálculo del Costo:	2.038.730,33	pesos

3. PRODUCTO

3.1. Ámbito del software

El producto software desarrollado va dirigido principalmente a la directora y a los docentes del programa académico Ingeniería de Sistemas de la Universidad Francisco de Paula Santander.

3.2. Descomposición del problema del software a desarrollar

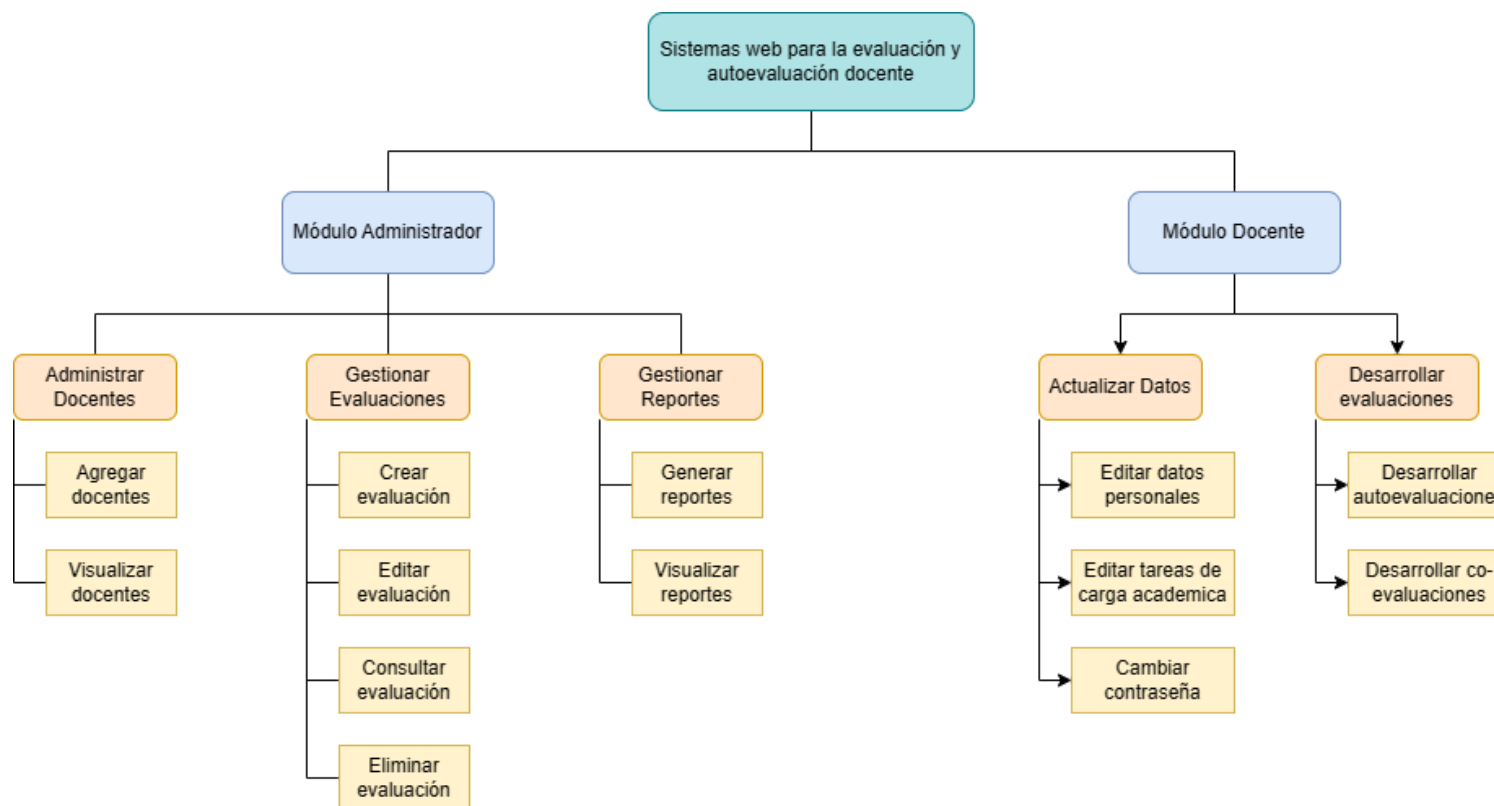


Figura 6. Descomposición de los procesos del problema a desarrollar en el proyecto SWEAD

La anterior figura representa el desglose de las funcionalidades que se realizan en cada uno de los módulos. En ese caso, para el módulo de administrador, se encarga de realizar la gestión de los docentes mediante dos funcionalidades: agregar a docentes con información parcial de estos, ya que mediante una lista de correos de los mismos, se envían las solicitudes de registro a los usuarios docente; aparte, el administrador puede visualizar la información ya diligenciada de cada docente. Similarmente, el administrador puede gestionar las evaluaciones mediante cuatro funcionalidades, las cuales son crear, editar, consultar y eliminar las evaluaciones dentro del sistema. Además, el administrador puede realizar la gestión de los reportes. Esto le permitirá generar reportes de las diferentes evaluaciones y visualizar los resultados de cada docente.

Por otro lado, en el módulo de docentes, para dar solución al problema planteado, se presentan dos funcionalidades, el poder realizar la actualización de sus datos personales y de las tareas correspondientes a su carga académica integral. También pueden desarrollar las evaluaciones que el administrador diseñe.

ADMINISTRACIÓN DEL RIESGO DEL PROYECTO EN DESARROLLO

4. RIESGOS

4.1. Identificación de los riesgos del proyecto en desarrollo

Código	Categoría	Riesgo	Causas	Efectos
01	Elaboración de la Planificación	Realización de mala estimación	-El gestor del proyecto no tiene experiencia -Expectativas sin definir ni gestionar	El producto resulta mucho más grande de lo estimado
02	Elaboración de la Planificación	La planificación se hace dentro de un ámbito optimista	Falta de conocimiento en las posibles entradas y salidas del software.	El producto contiene errores que no se tuvieron en cuenta cuando se hicieron las pruebas del software.
03	Elaboración de la Planificación	Retraso de tareas	-Falta de comunicación entre el equipo -Falta de planificación teniendo en cuenta posibles contratiempos	Se retrasan otras tareas dependientes



04	Organización y Gestión	Estructura inadecuada del equipo	Poco conocimiento en el tiempo y esfuerzo que conllevan las actividades asociadas a cada rol.	Reducción de productividad
05	Ambiente/Infraestructura de Desarrollo	No se cuenta con un espacio adecuado	No existen ambientes propicios y tranquilos para un óptimo desarrollo del proyecto	Disminución de avances en el proyecto debido al estrés.
06	Ambiente/Infraestructura de Desarrollo	No se cuenta con herramientas de desarrollo óptimas	Las herramientas no funcionan como se esperaban y se requiere una adaptación a las herramientas disponibles	Resultados menores a los esperados
07	Requisitos	Nacen nuevos requisitos en un momento inesperado	Falta de comunicación entre el ing. de requerimientos y los usuarios finales	Requerimientos que no son cubiertos por parte del equipo de desarrollo
08	Usuarios Finales	Insatisfacción con respecto al software por parte de los usuarios finales	Falta de comunicación y falta de mockups donde el usuario pueda visualizar el futuro	Rediseño del software

			desarrollo del software	
09	Producto	Módulos con posibilidad de errores	Planificación ineficiente en la fase de diseño del software	Se requiere de un esfuerzo y tiempo adicional para el tratamiento de errores
10	Producto	Funciones erróneas del software	Errores en la toma de requerimientos y planificación del diseño	Atraso en el desarrollo debido a la necesidad de rediseñar las funciones y volver a implementarlas
11	Personal	No hay un óptimo trabajo en equipo	Falta de comunicación asertiva entre los miembros del equipo	Errores en el desarrollo del software y atraso de la productividad
12	Personal	El personal trabaja lento	Falta de capacitación y experiencia del personal	Atraso en las actividades del proyecto
13	Diseño e Implementación	Diseño demasiado complejo	-Alta complejidad en los procesos del negocio -Requerimientos complejos	El diseño no alcanza a cubrir los requerimientos del proyecto

Tabla 28. Riesgos Identificados

4.2 Estrategias reactivas del riesgo frente a estrategias proactivas del riesgo

Las estrategias proactivas se han diseñado antes de que el riesgo suceda, mientras que las reactivas son las que se realizan después de que el riesgo haya sucedido.

Estrategias Proactivas

- Reclutamiento de un personal adecuado con experiencia en proyectos similares
- Confirmación de requerimientos por parte del cliente final
- Socialización de los mockups y diseño del software con el cliente
- Testeos de los módulos que se van desarrollando

Estrategias Reactivas

- Remodelamiento de la base de datos
- Rediseño de la interfaz de usuario
- Ajuste de requerimientos

4.3. Riesgos del software del proyecto en desarrollo

Código	Categoría	Riesgo	Causas	Efectos
01	Producto	Diseño de interfaz de usuario inadecuada	Mala comunicación entre el diseño de UI/UX y el cliente	Rediseño de la interfaz de usuario y atraso en resultados
02	Producto	Módulos con errores	Falta de testeos en los módulos desarrollados	Mayor esfuerzo en la corrección de errores
03	Producto	Requerimientos que no fueron satisfechos	Falta de confirmación de los requisitos por parte del cliente	Ajuste de requerimientos y atraso del proyecto

Tabla 29. Riesgos del software

4.4. Proyección del riesgo

Criterios de probabilidad:

Criterio de Probabilidad	
Casi certeza	5

Probable	4
Moderada	3
Improbable	2
Raro	1

Tabla 30. Criterios de probabilidad

Análisis de Impacto:

Impacto	Escala	Definición de Categorías
Catastrófico	5	Si ocurre un evento causaría fallas en el proyecto e inhabilita el alcance de los requerimientos mínimos
Mayor	4	Si ocurre un evento causa incrementos severos en el costo y tiempo y los requerimientos secundarios no serán alcanzados
Moderado	3	Si ocurre un evento causa incrementos moderados en el costo y tiempo, pero se los requerimientos importantes aún se pueden lograr

Menor	2	Si ocurre un evento causa incrementos bajos en el costo y tiempo, pero se los requerimientos importantes se pueden lograr
Insignificante	1	Si ocurre un evento no tendrá efectos en el proyecto

Tabla 31. Análisis del impacto

Matriz de probabilidad de impacto:

Impacto	Muy Bajo	Bajo	Moderado	Alto	Muy Alto
Probabilidad	1	2	3	4	5
5	5	10	15	20	25
4	4	8	12	16	20
3	3	6	9	12	15
2	2	4	6	8	10
1	1	2	3	4	5

Tabla 32. Matriz de probabilidad de impacto

Calificación del Riesgo:

Calificación de Riesgo		Rango de PXI
	Riesgo Bajo	1-4
	Riesgo Moderado	5-11
	Riesgo Alto	12-25

Tabla 33. Calificación del riesgo

Análisis y Clasificación del Riesgo:

Código	Categoría	Riesgo	Probabilidad	Impacto	Total
01	Elaboración de la Planificación	Realización de mala estimación	2	2	4
02	Elaboración de la Planificación	La planificación se hace dentro de un ámbito optimista	3	2	6



03	Elaboración de la Planificación	Retraso de tareas	3	3	9
04	Organización y Gestión	Estructura inadecuada del equipo	2	2	4
05	Ambiente/Infraestructura de Desarrollo	No se cuenta con un espacio adecuado	1	2	2
06	Ambiente/Infraestructura de Desarrollo	No se cuenta con herramientas de desarrollo óptimas	2	3	6
07	Requisitos	Nacen nuevos requisitos en un momento inesperado	3	3	9
08	Usuarios Finales	Insatisfacción con respecto al software por parte de los usuarios finales	2	2	4
09	Producto	Módulos con	3	4	12



		posibilidad de errores			
10	Producto	Funciones erróneas del software	3	4	12
11	Personal	No hay un óptimo trabajo en equipo	2	2	4
12	Personal	El personal trabaja lento	3	2	6
13	Diseño e Implementación	Diseño demasiado complejo	2	2	4

Tabla 34. Analisis y clasificacion del riesgo

4.5. Mitigación, monitoreo y manejo de riesgos.

Matriz de Riesgos y Controles											
#	Categoría	Riesgo	Estado	Descripción de la situación a la fecha	Causas	Efectos	Probabilidad	Impacto	Riesgos	Controles Existentes	Responsable
01	Elaboración de la Planificación	Realización de mala estimación	Seguimiento	Aún se realizan procesos de planificación	-El gestor del proyecto no tiene experiencia -Expectativas sin definir ni gestionar	El producto resulta mucho más grande de lo estimado	Improbable	Menor	Bajo	Investigación en proyectos similares para tener una guía de estimación	
02	Elaboración de la Planificación	La planificación se hace dentro de un ámbito optimista	Retirado	Se tuvieron en cuenta todos los posibles escenarios.	Falta de conocimiento en las posibles entradas y salidas del software.	El producto contiene errores que no se tuvieron en cuenta cuando se hicieron las pruebas del software.	Moderada	Menor	Moderado	Socialización con el cliente y el equipo de desarrollo para conocer los procesos de negocio y todos sus escenarios	

03	Elaboración de la Planificación	Retraso de tareas	Seguimiento	Constantemente se hacen reuniones de seguimiento de tareas	-Falta de comunicación entre el equipo -Falta de planificación teniendo en cuenta posibles contratiempos	Se retrasan otras tareas dependientes	Moderada	Moderado	Moderado	Socialización con el equipo de desarrollo sobre los avances de actividades	
04	Organización y Gestión	Estructura inadecuada del equipo	Retirada	Se realizaron reuniones con el fin de asignar roles de acuerdo a las habilidades de cada integrante	Poco conocimiento en el tiempo y esfuerzo que conllevan las actividades asociadas a cada rol.	Reducción de productividad	Improbable	Menor	Bajo	Reuniones constantes con el equipo de trabajo para conocer posibles dificultades	
05	Ambiente/Infraestructura de Desarrollo	No se cuenta con un espacio adecuado	Seguimiento	Aún se realizan reuniones para conocer el estado de cada integrante del equipo	No existen ambientes propicios y tranquilos para un óptimo desarrollo del proyecto	Disminución de avances en el proyecto debido al estrés.	Raro	Menor	Bajo	Reuniones con el equipo para conocer las causas de atraso en el proyecto	
06	Ambiente/Infraestructura de	No se cuenta con herramientas	Retirado	Se realizaron reuniones con el equipo para	Las herramientas no funcionan	Resultados menores a los	Improbable	Moderado	Moderado	Reuniones de seguimiento	

	Desarrollo	de desarrollo óptimas		conocer con qué herramientas contaba cada uno	como se esperaban y se requiere una adaptación a las herramientas disponibles	esperados				con el equipo de trabajo	
07	Requisitos	Nacen nuevos requisitos en un momento inesperado	Activo	El proyecto puede contar con nuevas necesidades	Falta de comunicación entre el ing. de requerimientos y los usuarios finales	Requerimientos que no son cubiertos por parte del equipo de desarrollo	Moderado	Moderado	Moderado	Reuniones entre el ing de requerimientos y el cliente	
08	Usuarios Finales	Insatisfacción con respecto al software por parte de los usuarios finales	Seguimiento	El contexto del negocio es complejo y pueden haber mala interpretación en la comunicación con el cliente	Falta de comunicación y falta de mockups donde el usuario pueda visualizar el futuro desarrollo del software	Rediseño del software	Improbable	Menor	Bajo	Comunicación constante con el cliente para mostrar avances	
09	Producto	Módulos con posibilidad de errores	Activo	Todos los módulos desarrollados	Planificación ineficiente en la fase de diseño	Se requiere de un esfuerzo y	Moderada	Mayor	Alto	Testeos continuos de los módulos	

				se encuentran en tests continuos	del software	tiempo adicional para el tratamiento de errores				desarrollados	
10	Producto	Funciones erróneas del software	Seguimiento	Todas las funciones permanecen bajo seguimiento y testeo continuo	Errores en la toma de requerimientos y planificación del diseño	Atraso en el desarrollo debido a la necesidad de rediseñar las funciones y volver a implementar las	Moderada	Mayor	Alto	Testeos continuos de las funciones del software	
11	Personal	No hay un óptimo trabajo en equipo	Seguimiento	Se realizan reuniones de seguimiento de trabajo	Falta de comunicación asertiva entre los miembros del equipo	Errores en el desarrollo del software y atraso de la productividad	Improbable	Menor	Bajo	Reuniones de seguimiento para mantener la unión del equipo de trabajo	
12	Personal	El personal trabaja lento	Seguimiento	Se asignan fechas para cada tarea y se realizan	Falta de capacitación y experiencia del personal	Atraso en las actividades del proyecto	Moderada	Menor	Moderado	Asignación de fechas límites para cada	

				reuniones para conocer los avances						actividad y seguimiento de las mismas	
13	Diseño e Implementa ción	Diseño demasiado complejo	Activo	Los procesos de negocio requieren que todas las modificaciones sean tenidas en cuenta desde la fase de diseño	-Alta complejidad en los procesos del negocio -Requerimiento s complejos	El diseño no alcanza a cubrir los requerimient os del proyecto	Improb able	Menor	Bajo	-Investigació n profunda sobre los procesos del negocio. -Comunicaci ón constante con el cliente	

Tabla 35. Matriz de riesgos y controles

4.6. El Plan de Riesgos

Riesgo	Plan de Supervisión
Realización de mala estimación	Investigación en proyectos similares para tener una guía de estimación
La planificación se hace dentro de un ámbito optimista	Socialización con el cliente y el equipo de desarrollo para conocer los procesos de negocio y todos sus escenarios
Retraso de tareas	Socialización con el equipo de desarrollo sobre los avances de actividades
Estructura inadecuada del equipo	Reuniones constantes con el equipo de trabajo para conocer posibles dificultades
No se cuenta con un espacio adecuado	Reuniones con el equipo para conocer las causas de atraso en el proyecto
No se cuenta con herramientas de desarrollo óptimas	Reuniones de seguimiento con el equipo de trabajo
Nacen nuevos requisitos en un momento inesperado	Reuniones entre el ing de requerimientos y el cliente
Insatisfacción con respecto al software por parte de los usuarios finales	Comunicación constante con el cliente para mostrar avances
Módulos con posibilidad de errores	Testeos continuos de los módulos desarrollados
Funciones erróneas del software	Testeos continuos de las funciones del software

No hay un óptimo trabajo en equipo	Reuniones de seguimiento para mantener la unión del equipo de trabajo
El personal trabaja lento	Asignación de fechas límites para cada actividad y seguimiento de las mismas
Diseño demasiado complejo	-Investigación profunda sobre los procesos del negocio. -Comunicación constante con el cliente

Tabla 36. Plan de riesgos

A continuación se adjuntará el documento del plan de gestión de riesgos donde se explica más a detalle la identificación, clasificación, proyección, monitoreo y estrategias de los riesgos del software SWEAD:

[Plan de riesgos](#)

ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

5. ASEGURAMIENTO DE CALIDAD

Se realiza mediante el diagrama de Boehm

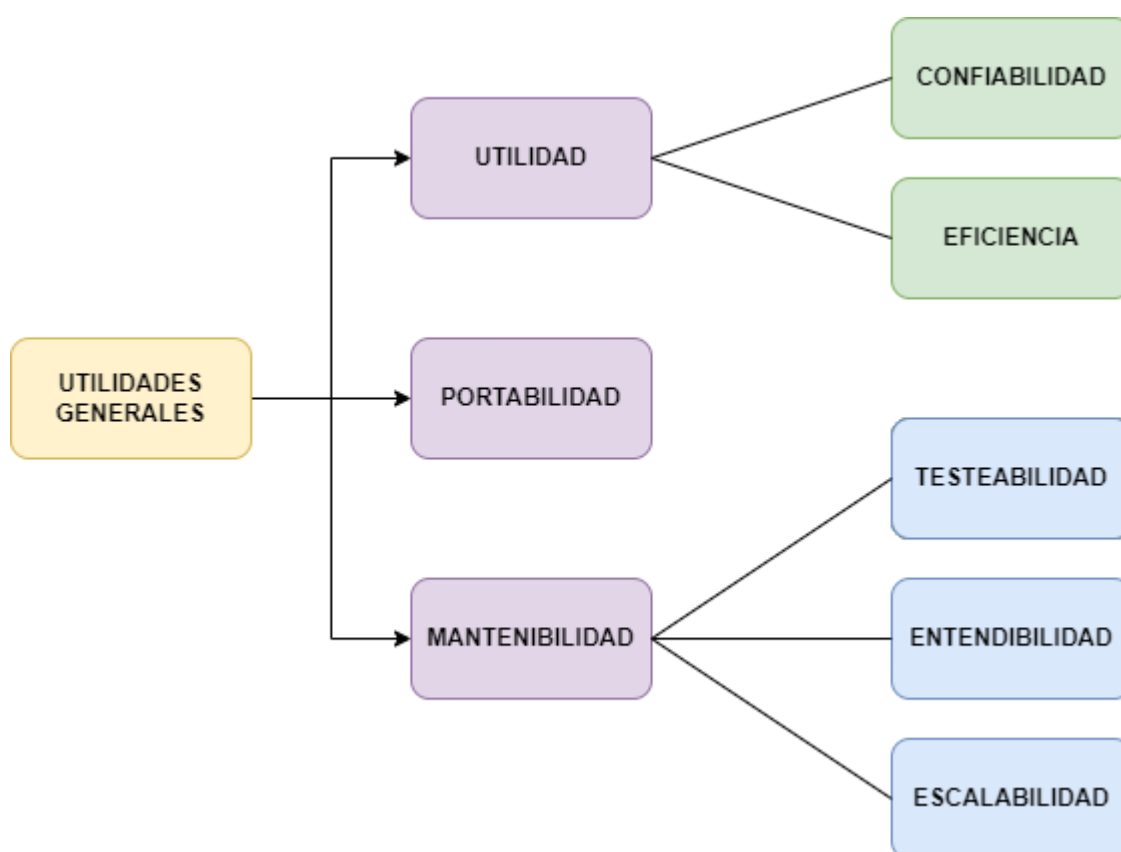


Figura 7. Modelo de aseguramiento de la calidad según Bohem para el proyecto SWEAD

5.1. Establecer procedimientos organizacionales y estándares para la calidad

Estándares

Estos son esenciales para la conformación de la calidad del software, nos permiten evaluar y mejorar los distintos aspectos del software en cualquiera de sus fases de desarrollo, también nos permiten evaluar los posibles cambios y a su vez detectar, reportar y reparar los distintos errores que se puedan presentar.

Campo de acción que cubren

- Ciclo de vida del software
- Documentación
- Código fuente
- Factores de riesgos
- Mejoras y corrección de errores
- Procedimientos y protocolos

6. PLANEACIÓN DE CALIDAD

Para los siguientes puntos, se recomienda revisar el anexo del Plan de Calidad.

[D-ACA002 - Plan de calidad - SWEAD](#)

6.1. Seleccionar procedimientos aplicables y estándares para un proyecto en particular y modificar estos como sean requeridos

Para los procedimientos de la calidad, se plantean diferentes actividades grupales para desarrollar durante la evaluación de calidad.

Actividades del grupo

ID	Descripción
ASC1	Implementar el plan de desarrollo de calidad
ASC2	Realizar la meditación sistemática de calidad
ASC3	Indicar los estándares de comparación con otros sistemas
ASC4	Verificar el seguimiento de los procesos de control de calidad en el software.
ASC5	Asegurar la terminación de los documentos a desarrollar
ASC6	Registrar y reportar las inconsistencias en la documentación a las áreas específicas.
ASC7	Apoyar la recolección y análisis de métricas de software

Tabla 37. Descripción de las actividades del grupo del proyecto SWEAD

7. CONTROL DE CALIDAD

La calidad del software según es una metodología que ayuda a recopilar evidencias del uso de un producto o software, para garantizar esta calidad se utilizan directrices para medir el control de la calidad de este, además de prolongar la vida del software. La correlación entre ocupación de la auditoría, garantía de calidad, control de calidad y prueba de software frecuentemente se complica. La garantía de calidad es el conjunto de tareas que sirven como soporte para proveer una adecuada seguridad donde cada uno de los procesos creados es permanentemente optimizado, tratando que los productos cumplan con los requerimientos y especificaciones del producto o software, además es necesario que cumplan con las condiciones ideales para su uso. El encargado de comparar la calidad del producto con las políticas adoptadas y las medidas establecidas de no conformidad cuando se detecta una inconsistencia se conoce como control de calidad. La auditoría es la encargada de la evaluación y verificación del seguimiento de las normas, procedimientos y planes.

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.

7.1. Garantizar que procedimientos y estándares son seguidos por el equipo de desarrollo de software

Para el control de calidad, se garantizan que procedimientos y estándares son seguidos por el equipo de desarrollo de software.

En el siguiente anexo se realiza la especificación de los requerimientos de calidad que se presentan como criterios de aceptación en cada una de las fases del ciclo de vida del proyecto de software.

[AC-CCP-05 CONTROL DE CALIDAD SWEAD](#)

Índice de Figuras

Figura 1. Metodología PSP	16
Figura 2. Ilustración proceso de software del equipo (TSP)	17
Figura 3. Ilustración modelado general de los procesos del proyecto SWEAD	20
Figura 4. Modelo de ciclo de vida en V. Elaboración propia	24
Figura 5. Descomposición de los procesos del problema a desarrollar en el proyecto SWEAD	27
Figura 6. Descomposición de los procesos del problema a desarrollar en el proyecto	

SWEAD

64

Figura 7. Modelo de aseguramiento de la calidad según Bohem para el proyecto SWEAD83

Índice de Tablas

Tabla 1. Calendario de actividades del proyecto SWEAD	31
Tabla 2. Tipos de componentes de los puntos de fusión	32
Tabla 3. Estimación Basado en Puntos de Fusión del Software SWEAD	34
Tabla 4. Puntos de Fusión sin Ajustar	35
Tabla 5. Tabla de Factores de Ajuste de Complejidad Técnica.	36
Tabla 6. Métrica de cada etapa del proyecto SWEAD	37
Tabla 7. Recursos humanos del proyecto SWEAD	38
Tabla 8. Recursos tecnológicos del proyecto SWEAD	39
Tabla 9. Recursos de software reutilizables del proyecto SWEAD.	40
Tabla 10. Recursos del entorno del proyecto SWEAD.	42
Tabla 11. Métrica de usabilidad del proyecto SWEAD	43
Tabla 12. Métrica de madurez del proyecto SWEAD	45
Tabla 13. Métrica de adecuación del proyecto SWEAD	46
Tabla 14. Estimación por líneas de código del proyecto SWEAD.	48
Tabla 15. Tipos de componentes de los puntos de fusión	49
Tabla 16. Estimación Basado en Puntos de Función del Software SWEAD.	52
Tabla 17. Puntos de Fusión sin Ajustar.	52
Tabla 18. Tabla de Factores de Ajuste de Complejidad Técnica.	54
Tabla 19. Puntos de función ajustado (PFA)	55
Tabla 20. Horas de lenguaje de programación 4ta generación	55
Tabla 21. PFA	55

Tabla 22. Métrica de cada etapa del proyecto SWEAD	56
Tabla 23: Factores de Peso de los Actores sin Ajustar.	58
Tabla 24: Factor de Peso en los Casos de Uso sin Ajustar	58
Tabla 25: Peso de los Factores de Complejidad Técnica	60
Tabla 26: Peso de los Factores Ambientales	62
Tabla 27. Distribución Real del Esfuerzo	63
Tabla 28. Riesgos Identificados	69
Tabla 29. Riesgos del software	70
Tabla 30. Criterios de probabilidad	71
Tabla 31. Análisis del impacto	72
Tabla 32. Matriz de probabilidad de impacto	72
Tabla 33. Calificación del riesgo	73
Tabla 34. Analisis y clasificacion del riesgo	75
Tabla 35. Matriz de riesgos y controles	80
Tabla 36. Plan de riesgos	82
Tabla 37. Descripción de las actividades del grupo del proyecto SWEAD	85

Anexos

TÍTULO	RUTA	DESCRIPCIÓN
Gestión de cambios	Carpeta gestión de cambios del proyecto SWEAD	Documentos que contienen el control de cambios del sistema web para evaluación y autoevaluación docente de los módulos administrador y graduado.



Plan de Gestión de riesgos	<u>Carpeta plan de gestión de riesgos del proyecto SWEAD</u>	Carpeta que contiene un documento del detallado completo del plan de gestión de riesgos del software SWEAD con la identificación, clasificación, proyección, monitoreo y estrategias de los riesgos.
Cronograma	<u>Cronograma del proyecto SWEAD</u>	Project del cronograma de las actividades del software SWEAD.
Estimaciones	<u>Carpeta de estimaciones del proyecto SWEAD</u>	Excel que contiene la estimación de duración del proyecto de desarrollo SWEAD.
Control de la Calidad	<u>Control de calidad del proyecto SWEAD</u>	Documento de google docs que contiene la especificación de los criterios de calidad que se deben cumplir en cada una de las fases del desarrollo del proyecto SWEAD