**DESARROLLO DE UN MARCO DE TRABAJO SIGUIENDO EL ESTÁNDAR ISO/IEC 29110 PARA EL CASO DE ESTUDIO DE INTEGRACIÓN DE PROCESOS DEL PROYECTO SANAMBIENTE.**

Aarón Levi Grajales Gómez, aarongrajales@hotmail.com

Cristhian Fernando Balanta Pazú, cristianpazu2@gmail.com

Semillero ITMedia - Grupo de investigación GRINTIC

Trabajo de grado presentado para optar al título de Ingeniero de sistemas

Asesor(a): Beatriz Eugenia Marín, Ingeniera de sistemas



INSTITUCIÓN UNIVERSITARIA ANTONIO JOSÉ CAMACHO

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

SANTIAGO DE CALI - COLOMBIA

2020

**Notas de aceptación:**

**Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Institución Universitaria Antonio José Camacho para optar al título de Ingeniero de Sistemas.**

**Jurado**

**Jurado**

**Santiago de Cali, 1 de agosto de 2020**

**Resumen**

Los productos de software deben garantizar el cumplimiento de criterios mínimos de calidad aplicando métodos y buenas prácticas que se han propuesto a partir de las experiencias de autores y organizaciones que han trabajado en el tema por mucho tiempo.

Estas experiencias brindan lineamientos que se pueden ajustar a diversos contextos. En este proyecto se presenta la adecuación de las prácticas propuestas por la norma ISO/IEC 29110 para mejorar la calidad de los productos y el rendimiento de los procesos implicados, integrando la metodología Iconix e IEEE 829 para guiar tanto el proceso de implementación del software como la ejecución de las pruebas de calidad sobre este, en el marco del desarrollo del módulo de transferencia y comunicación de datos de las estaciones ambientales de la empresa Sanambiente de Cali.

Por último, cabe mencionar que el presente trabajo fue desarrollado en conjunto con dos proyectos de grado de estudiantes de Tecnología e Ingeniería de sistemas pertenecientes a la Institución Universitaria Antonio José Camacho, los cuales tenían a cargo los procesos de implementación de software y ejecución de pruebas de calidad respectivamente.

**Palabras clave:** ISO/IEC 29110, Gerencia de proyecto, Metodología Iconix.

**Abstract**

Software products must guarantee compliance with minimum quality criteria by applying methods and good practices that have been proposed based on the experiences of authors and organizations that have worked on the subject for a long time.

These experiences provide guidelines that can be adjusted to various contexts. This project presents the adaptation of the practices proposed by the ISO/IEC 29110 standard to improve the quality of the products and the performance of the processes involved, integrating the Iconix and IEEE 829 methodology to guide both the software implementation process and the execution of the quality tests on it, within the framework of the development of the data transfer and communication module of the environmental stations of the company Sanambiente de Cali.

Finally, it is worth mentioning that this work was developed in conjunction with two-degree projects of Technology and Systems Engineering students from the Antonio José Camacho University Institution, who were in charge of the software implementation and quality testing processes, respectively.

**Keywords:** ISO/IEC 29110, Project Management, Iconix Methodology.

**Tabla de contenido**

[Introducción 1](#_Toc47024545)

[1. Planteamiento del problema 5](#_Toc47024546)

[2. Problema de investigación 8](#_Toc47024547)

[2.1 Formulación del problema 8](#_Toc47024548)

[2.2 Sistematización del problema 8](#_Toc47024549)

[3. Objetivos 9](#_Toc47024550)

[3.1 Objetivo general 9](#_Toc47024551)

[3.2 Objetivos específicos 9](#_Toc47024552)

[4. Justificación 10](#_Toc47024553)

[5. Marco de referencia 12](#_Toc47024554)

[5.1 Antecedentes 12](#_Toc47024555)

[5.2 Marco conceptual 14](#_Toc47024556)

[5.2.1 Proceso de software 14](#_Toc47024557)

[5.2.2 Metodología de desarrollo de software 15](#_Toc47024558)

[5.2.3 Lenguaje Unificado de Modelado (Unified Modeling Language (UML)) 15](#_Toc47024559)

[5.2.4 Calidad de software 16](#_Toc47024560)

[5.2.5 Modelo de calidad de software 16](#_Toc47024561)

[5.3 Marco teórico 16](#_Toc47024562)

[5.3.1 Revisión de área 17](#_Toc47024563)

[5.3.1.1 PMBOK 19](#_Toc47024564)

[5.3.2 Comparación entre estándares para la gestión de proyectos 19](#_Toc47024565)

[5.3.3 Estándar ISO/IEC 29110. 20](#_Toc47024566)

[5.3.4 Comparación entre metodologías de desarrollo de software 27](#_Toc47024567)

[5.3.5 Metodología Iconix. 29](#_Toc47024568)

[5.3.6 Reconocimiento de las tecnologías utilizadas por los equipos de trabajo 34](#_Toc47024569)

[5.3.6.1 Angular Framework 34](#_Toc47024570)

[5.3.6.2 JavaScript 34](#_Toc47024571)

[5.3.6.3 TypeScript 34](#_Toc47024572)

[5.3.6.4 PostgreSQL 35](#_Toc47024573)

[5.3.6.5 NodeJS 35](#_Toc47024574)

[5.3.6.6 GitHub 35](#_Toc47024575)

[5.3.6.7 GitHub Desktop 35](#_Toc47024576)

[5.3.6.8 Visual Studio Code 35](#_Toc47024577)

[5.3.6.9 Jest 36](#_Toc47024578)

[5.3.6.10 Jmeter 36](#_Toc47024579)

[5.4 Marco legal 36](#_Toc47024580)

[6. Resultados 38](#_Toc47024581)

[6.1 Fase 1: Planificación del proyecto 42](#_Toc47024582)

[6.1.1 Crear el plan del proyecto. 42](#_Toc47024583)

[6.1.2 Crear repositorio del proyecto 79](#_Toc47024584)

[6.1.3 Registrar los resultados de la verificación 81](#_Toc47024585)

[6.2 Fase 2: Ejecución del plan del proyecto 82](#_Toc47024586)

[6.2.1 Diligenciar un acta de reunión 83](#_Toc47024587)

[6.2.2 Diligenciar el registro de estado de progreso 84](#_Toc47024588)

[6.2.3 Diligenciar una solicitud de cambio 85](#_Toc47024589)

[6.3 Fase 3: Evaluación y control del proyecto 87](#_Toc47024590)

[6.3.1 Diligenciar el registro de corrección 87](#_Toc47024591)

[6.4 Fase 4: Cierre del proyecto 88](#_Toc47024592)

[6.4.1 Diligenciar un registro de aceptación. 88](#_Toc47024593)

[7. Conclusiones 91](#_Toc47024594)

[8. Referencias 99](#_Toc47024595)

**Lista de tablas**

[Tabla 1. Diagnóstico y pronóstico del proyecto. 7](#_Toc47024472)

[Tabla 2. Diagramas de UML por categoría. 16](#_Toc47024473)

[Tabla 3. Estándares para la gestión de proyectos. 17](#_Toc47024474)

[Tabla 4. Comparación entre estándares para la gestión de proyectos. 19](#_Toc47024475)

[Tabla 5. Público objetivo de la ISO/IEC 29110. 21](#_Toc47024476)

[Tabla 6. Productos o artefactos según la ISO/IEC 29110. 26](#_Toc47024477)

[Tabla 7. Comparación entre metodologías de desarrollo de software. 28](#_Toc47024478)

[Tabla 8. Entregables del equipo de desarrollo. 43](#_Toc47024479)

[Tabla 9. Entregables del equipo de calidad. 44](#_Toc47024480)

[Tabla 10. Duración estimada de las tareas 1.1 - 3.2 (CU04 y CU08) del equipo de desarrollo. 46](#_Toc47024481)

[Tabla 11. Duración estimada de las tareas 1.1 - 3.2 (CU16 y CU10) del equipo de desarrollo. 46](#_Toc47024482)

[Tabla 12. Duración estimada de las tareas 1.1 - 3.2 (CU06 y CU07) del equipo de desarrollo. 47](#_Toc47024483)

[Tabla 13. Duración estimada de las tareas 1.1 - 3.2 (CU05) del equipo de desarrollo. 48](#_Toc47024484)

[Tabla 14. Duración estimada de las tareas 1.1 - 3.2 (CU01) del equipo de desarrollo. 48](#_Toc47024485)

[Tabla 15. Duración estimada de las tareas 1.1 - 3.2 (CU09 y CU11) del equipo de desarrollo. 49](#_Toc47024486)

[Tabla 16. Duración estimada de las tareas 3.3 y 3.4 (CU04 y CU08) del equipo de desarrollo. 50](#_Toc47024487)

[Tabla 17. Duración estimada de las tareas 3.3 y 3.4 (CU16 y CU10) del equipo de desarrollo. 50](#_Toc47024488)

[Tabla 18. Duración estimada de las tareas 3.3 y 3.4 (CU06 y CU07) del equipo de desarrollo. 50](#_Toc47024489)

[Tabla 19. Duración estimada de las tareas 3.3 y 3.4 (CU05) del equipo de desarrollo. 51](#_Toc47024490)

[Tabla 20. Duración estimada de las tareas 3.3 y 3.4 (CU01) del equipo de desarrollo. 51](#_Toc47024491)

[Tabla 21. Duración estimada de las tareas 3.3 y 3.4 (CU09 y CU11) del equipo de desarrollo. 52](#_Toc47024492)

[Tabla 22. Duración estimada de las tareas 1.1 - 4.1 (CU04 y CU08) del equipo de calidad. 52](#_Toc47024493)

[Tabla 23. Duración estimada de las tareas 1.1 - 4.1 (CU16 y CU10) del equipo de calidad. 53](#_Toc47024494)

[Tabla 24. Duración estimada de las tareas 1.1 - 4.1 (CU06 y CU07) del equipo de calidad. 54](#_Toc47024495)

[Tabla 25. Duración estimada de las tareas 1.1 - 4.1 (CU05) del equipo de calidad. 55](#_Toc47024496)

[Tabla 26. Duración estimada de las tareas 1.1 - 4.1 (CU01) del equipo de calidad. 57](#_Toc47024497)

[Tabla 27. Duración estimada de las tareas 1.1 - 4.1 (CU09 y CU11) del equipo de calidad. 58](#_Toc47024498)

[Tabla 28. Calendario de las tareas 1.1 - 3.2 (CU04 y CU08) del equipo de desarrollo. 60](#_Toc47024499)

[Tabla 29. Calendario de las tareas 1.1 - 3.2 (CU16 y CU10) del equipo de desarrollo. 61](#_Toc47024500)

[Tabla 30. Calendario de las tareas 1.1 - 3.2 (CU06 y CU07) del equipo de desarrollo. 61](#_Toc47024501)

[Tabla 31. Calendario de las tareas 1.1 - 3.2 (CU05) del equipo de desarrollo. 62](#_Toc47024502)

[Tabla 32. Calendario de las tareas 1.1 - 3.2 (CU01) del equipo de desarrollo. 63](#_Toc47024503)

[Tabla 33. Calendario de las tareas 1.1 - 3.2 (CU09 y CU11) del equipo de desarrollo. 63](#_Toc47024504)

[Tabla 34. Calendario de las tareas 3.3 y 3.4 (CU04 y CU08) del equipo de desarrollo. 64](#_Toc47024505)

[Tabla 35. Calendario de las tareas 3.3 y 3.4 (CU16 y CU10) del equipo de desarrollo. 64](#_Toc47024506)

[Tabla 36. Calendario de las tareas 3.3 y 3.4 (CU06 y CU07) del equipo de desarrollo. 65](#_Toc47024507)

[Tabla 37. Calendario de las tareas 3.3 y 3.4 (CU05) del equipo de desarrollo. 65](#_Toc47024508)

[Tabla 38. Calendario de las tareas 3.3 y 3.4 (CU01) del equipo de desarrollo. 66](#_Toc47024509)

[Tabla 39. Calendario de las tareas 3.3 y 3.4 (CU09 y CU11) del equipo de desarrollo. 66](#_Toc47024510)

[Tabla 40. Calendario de las tareas 1.1 - 2.4 (CU04 y CU08) del equipo de calidad. 67](#_Toc47024511)

[Tabla 41. Calendario de las tareas 1.1 - 2.4 (CU16 y CU10) del equipo de calidad. 67](#_Toc47024512)

[Tabla 42. Calendario de las tareas 1.1 - 2.4 (CU06 y CU07) del equipo de calidad. 68](#_Toc47024513)

[Tabla 43. Calendario de las tareas 1.1 - 2.4 (CU05) del equipo de calidad. 69](#_Toc47024514)

[Tabla 44. Calendario de las tareas 1.1 - 2.4 (CU01) del equipo de calidad. 70](#_Toc47024515)

[Tabla 45. Calendario de las tareas 1.1 - 2.4 (CU09 y CU11) del equipo de calidad. 71](#_Toc47024516)

[Tabla 46. Calendario de las tareas 2.5 y 2.6 (CU04 y CU08) del equipo de calidad. 72](#_Toc47024517)

[Tabla 47. Calendario de las tareas 2.5 y 2.6 (CU16 y CU10) del equipo de calidad. 72](#_Toc47024518)

[Tabla 48. Calendario de las tareas 2.5 y 2.6 (CU06 y CU07) del equipo de calidad. 73](#_Toc47024519)

[Tabla 49. Calendario de las tareas 2.5 y 2.6 (CU05) del equipo de calidad. 73](#_Toc47024520)

[Tabla 50. Calendario de las tareas 2.5 y 2.6 (CU01) del equipo de calidad. 73](#_Toc47024521)

[Tabla 51. Calendario de las tareas 2.5 y 2.6 (CU09 y CU11) del equipo de calidad. 74](#_Toc47024522)

[Tabla 52. Calendario de las tareas 3.1 - 4.1 (CU04 y CU08) del equipo de calidad. 74](#_Toc47024523)

[Tabla 53. Calendario de las tareas 3.1 - 4.1 (CU16 y CU10) del equipo de calidad. 75](#_Toc47024524)

[Tabla 54. Calendario de las tareas 3.1 - 4.1 (CU06 y CU07) del equipo de calidad. 75](#_Toc47024525)

[Tabla 55. Calendario de las tareas 3.1 - 4.1 (CU05) del equipo de calidad. 76](#_Toc47024526)

[Tabla 56. Calendario de las tareas 3.1 - 4.1 (CU01) del equipo de calidad. 77](#_Toc47024527)

[Tabla 57. Calendario de las tareas 3.1 - 4.1 (CU09 y CU11) del equipo de calidad. 77](#_Toc47024528)

[Tabla 58. Esfuerzo estimado. 78](#_Toc47024529)

[Tabla 59. Costo estimado. 78](#_Toc47024530)

**Lista de figuras**

[Figura 1. Diagrama Causa y efecto. 7](#_Toc47024531)

[Figura 2. Procesos del perfil básico. 23](#_Toc47024532)

[Figura 3. Diagrama del proceso de gestión del proyecto. 24](#_Toc47024533)

[Figura 4. Diagrama del proceso de implementación de software. 25](#_Toc47024534)

[Figura 5. Procesos de ICONIX. 30](#_Toc47024535)

[Figura 6. Ruta de procesos y herramientas del proyecto. 40](#_Toc47024536)

[Figura 7. Captura de imagen de referencia de la aplicación de escritorio de GitHub: GitHub Desktop. 80](#_Toc47024537)

[Figura 8. Repositorio del proyecto, visto desde la aplicación web de GitHub. 81](#_Toc47024538)

[Figura 9. Ejemplo de una plantilla de resultados de la verificación del plan de proyecto. 82](#_Toc47024539)

[Figura 10. Ejemplo de un acta de reunión lograda con el equipo de trabajo. 83](#_Toc47024540)

[Figura 11. Ejemplo de un registro de estado de progreso sobre los casos de uso 04 y 08 pertenecientes al equipo de desarrollo. 84](#_Toc47024541)

[Figura 12. Ejemplo de una solicitud de cambio iniciada. 86](#_Toc47024542)

[Figura 13. Ejemplo de un registro de corrección en estado pendiente. 88](#_Toc47024543)

[Figura 14. Ejemplo de un registro de aceptación diligenciado y firmado. 90](#_Toc47024544)

**Índice de acrónimos**

**ISO**: International Organization for Standardization.

**IEC:** International Electrotechnical Commission**.**

**IEEE:** Institute of Electrical and Electronics Engineers.

**PMI:** Project Management Institute.

**PMBOK:** Project Management Body of Knowledge.

**APM:** Association for Project Management.

**APMBOK:** Association for Project Management Body of Knowledge.

**CMMI-DEV:** Capability Maturity Model Integration for Development.

**XP:** eXtreme Programming.

# Introducción

Las empresas se están esforzando por procurar desarrollos de calidad a través de la utilización de métodos de gestión que han demostrado su efectividad a nivel mundial. Estándares como el PMBOK e ISO 21500 agrupan buenas prácticas y procesos necesarios para planear, ejecutar, evaluar y dar cierre a un proyecto de manera adecuada.

Para las pymes la utilización de estos estándares y métodos exigen una inversión importante, ya que requieren personal altamente capacitado que pueda articularlos con la cultura organizacional (Legaria Islas, 2018). Por esta razón, la ISO en conjunto con la IEC crearon la norma ISO/IEC 29110, que busca mejorar la calidad de los productos y/o servicios y el rendimiento de los procesos de las pequeñas empresas.

La ISO/IEC 29110 está conformada por 5 partes, de acuerdo con un público objetivo. La parte 1 presenta la visión general de las técnicas proporcionadas por el estándar. La parte 2 introduce los conceptos para los perfiles de ingeniería de software y los que se manejan a lo largo de la ISO. La parte 3 define las guías para el asesoramiento de procesos y cumplimiento de los requisitos para satisfacer un perfil. La parte 4 especifica todos los perfiles contenidos en un grupo de perfil. Por último, la parte 5 provee una guía para los procesos de gestión e implementación de software (ISO/IEC, 2011, pp. vi; vii).

De acuerdo con lo propuesto en la parte 4 del estándar, el perfil genérico es la categoría general y está destinada a las empresas que no desarrollan productos de software críticos. Dentro de este perfil se encuentran varios subgrupos: el perfil de entrada está orientado a VSEs que tienen un tiempo de fundación de tres años máximo o un número de empleados menor o igual a seis; el perfil básico está dirigido a VSEs que elaboran una sola aplicación con un solo equipo; el perfil intermedio está enfocado a VSEs que realizan más de un proyecto con más de un equipo; y finalmente el perfil avanzado, orientado a VSEs que desean mantenerse competitivas en el mercado (Marín Ospina, 2020, p. 3).

El estándar proporciona un paquete de despliegue con unos elementos que pueden servir como apoyo para las VSEs en la implementación de un perfil definido. Entre los elementos se encuentran: descripción técnica, definiciones clave, descripción detallada de procesos, actividades, roles y productos, plantillas y una lista de herramientas (ISO/IEC, 2011, p. 49).

En el presente trabajo se implementó la parte 5 del estándar, dada las necesidades y características que Sanambiente proponía en su proyecto de software. Debido al número de actividades, productos y personas que debían intervenir en el desarrollo de este último, se hizo necesario el incluir estrategias de gestión para garantizar su éxito, y esta parte de la norma las proporcionaba.

El perfil seguido en este trabajo dentro del grupo de perfiles genéricos fue el básico, puesto que era aplicable al desarrollo de una sola aplicación con un solo equipo, y en este caso era así. El proyecto propuesto tenía como fin la elaboración de una sola aplicación que no poseía riesgo especial y estaba a cargo de un solo equipo de trabajo.

Para el proyecto en mención, no se empleó el paquete de despliegue proporcionado por el estándar, dado que no se vio necesario el seguir este conjunto de elementos como medio para el control y seguimiento de los recursos que iban a interceder en el proyecto de la empresa Sanambiente. La utilización de este paquete es opcional, pues la ISO/IEC 29110 lo pone a disposición de las VSEs como alternativa para facilitar la ejecución de los procesos brindados por esta; por lo tanto, no tendría impacto negativo el no implementarlo. En su lugar, se decidió seguir únicamente los pasos descritos en los procesos contenidos en la norma.

Como se mencionó anteriormente, la parte 5 de este estándar proporcionaba una guía para dos procesos: gestión de proyectos e implementación de software; para el desarrollo de este trabajo, se decidió seguir únicamente la guía del proceso de gestión y adecuarlos al contexto del proyecto de la empresa Sanambiente.

La guía con la que contaba el estándar para el proceso de implementación de software no fue empleada, dado que la documentación que exigía para ser aplicada era demasiada y no se ajustaba al contexto de este proyecto para ser adoptada. En su lugar, se decidió seguir las fases de la metodología Iconix e IEEE 829, puesto que proporcionaban los elementos necesarios para guiar adecuadamente el desarrollo del software y la ejecución de las pruebas de calidad de acuerdo con las características y necesidades propias de la empresa.

Teniendo en cuenta lo anteriormente mencionado, el proyecto describe el cómo implementar un marco de trabajo siguiendo estándares y metodologías para facilitar el control y seguimiento de todos los recursos que intervienen en el desarrollo de un producto de software. Este marco incluye las principales etapas para gestión de un proyecto de desarrollo de software, desde la planificación del proyecto hasta el cierre de este, siguiendo como base los pasos y artefactos proporcionados por el estándar ISO/IEC 29110.

Cabe anotar que antes de ejecutar el desarrollo del presente proyecto, se realizaron investigaciones y comparaciones entre algunos estándares para la gestión de proyectos, con el fin de seleccionar aquel estándar que mejor se pudiera seguir y aplicar en el proyecto propuesto por la empresa Sanambiente. Los estándares para la gestión de proyectos que se compararon fueron la ISO/IEC 29110 y el PMBOK, de esta comparación se eligió el primer estándar mencionado, dado que brindaba las fases y artefactos necesarios para tener un fácil control y seguimiento de los recursos que intervinieron en el proyecto.

Adicional a esto, se hizo una comparación entre algunas metodologías para el desarrollo de software, con el fin de elegir aquella que pudiera guiar de una manera apropiada a los equipos durante todo el proceso de implementación del software. Las metodologías para el desarrollo de software que se compararon fueron Scrum, Iconix y XP, de esta comparación se eligió la segunda metodología mencionada, dado que brindaba sencillas reglas y prácticas, flexibilidad para los cambios y facilitaba su aplicación en equipos pequeños de desarrolladores (Bolívar Guzmán et al., 2020, p. 2).

# Planteamiento del problema

Sanambiente S.A.S es una empresa especializada en ofrecer soluciones integrales en productos y/o servicios que aportan al cuidado y mejoramiento de las condiciones ambientales en un medio ambiente específico.

La estructura organizacional de la empresa se encuentra conformada brevemente de la siguiente manera: en el puesto más alto se encuentra la junta directiva y el gerente general, luego se encuentran las gerencias de ventas, estrategia, técnica y administrativa (ésta última posee las oficinas de tesorería, logística y contabilidad), seguidas cada una por su respectiva jefatura (ejemplo: la gerencia técnica posee jefe técnico, jefe de tecnología, coordinador de servicios integrales y coordinador de proyectos), y adicionalmente por un líder jurídico que poseen.

Los datos que maneja la empresa son de tipo ambiental, todos estos dependiendo del producto y/o servicio y la información específica que tiene como labor transferir de los datalogger de las estaciones concretas para brindar al cliente, entre los que se encuentran: datos de muestreo continuo de agua por diferentes metodologías, como son: muestreo continuo en intervalos de tiempo, muestreo continuo o simple y muestreo por eventos (caudal, calidad de agua, lluvia) ; medidas en forma instantánea de la humedad del suelo que se especifique, conductividad eléctrica/salinidad y temperatura; muestreo de aguas superficiales, agua potable, aguas residuales y no residuales con toma de muestra puntual, compuesta e integrada en cuerpo lótico (ecosistema de aguas fluyentes) y medición en sitio de caudal, pH, temperatura, conductividad, oxígeno disuelto; mediciones continuas de partículas PM10 o PM2.5 o PM10-2.5; medición del ruido ambiental (ruido que existe en un determinado lugar y en un momento dado); medición de la calidad del aire, particularmente, la medición de CO (monóxido de carbono) y O3 (ozono), pero éstas se llevan a cabo en un área delimitada, dependiendo en donde se encuentre ubicado el producto y/o servicio encargado de realizar esta labor; medición y transmisión de información meteorológica, nivel o hidrometeorológica gracias a la implementación de comunicación satelital GOES.

La empresa Sanambiente maneja actualmente un sistema para realizar la transferencia de datos ambientales de los datalogger de las estaciones gracias a los protocolos de comunicación que utilizan, FTP y Modbus.

Para realizar la transferencia de datos por el protocolo Modbus, el sistema manda una ID, el puerto, la IP y el mensaje solicitando los datos ambientales, de igual forma la estación donde se encuentra el datalogger responde con un mensaje al sistema, con los datos ambientales requeridos, luego, lo que hace el sistema es capturarlos y almacenarlos en la base de datos. Para realizar la transferencia de datos por el protocolo FTP, es un poco diferente, ya que inicialmente se debe configurar un pequeño servidor FTP que generalmente se configura con el Internet Information Services propio de Windows, en el cual se encontrarán almacenados en una carpeta los datos ambientales transferidos del datalogger de la estación específica, luego, con otra aplicación llamada FTP Import/Export se leen estos datos de la carpeta, y la forma en que esta aplicación realiza la lectura es de acuerdo a una plantilla ya creada que está estructurada por canales (ej. SO2, NH3), luego de leerlos la aplicación procede a realizar el almacenamientos de los datos en la base de datos.

El inconveniente que posee la empresa Sanambiente está relacionado con la desarticulación de aplicativos, que requiere intervención manual de los desarrolladores para que la comunicación y transferencia de los datos ambientales se lleve a cabo sin problemas.

Para dar solución a este inconveniente se generará un aplicativo que integre los diferentes procesos de la gestión de datos. Para ello se han propuesto dos proyectos de grado que abordan diferentes perspectivas: calidad y desarrollo, pero se requiere un marco de trabajo que permita integrarlas correctamente.

Tabla 1. Diagnóstico y pronóstico del proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| **Síntoma** | **Causa** | **Efecto** | **Consecuencia** |
| Los equipos de calidad y desarrollo abordan el proyecto por separado. | No hay integración de los equipos. | Esto conlleva que el proyecto pierda la orientación. | Productos pequeños sin conexión. |
| Falta de comunicación de los equipos: calidad y desarrollo. | Falta de coordinación con los integrantes de los equipos. | Que cada equipo realice el proyecto por su cuenta. | Productos incompatibles que dificultan su integración necesaria para su correcto funcionamiento. |
| **Diagnóstico** | | **Pronóstico** | |

*Nota:* Elaboración propia.

Figura 1. Diagrama Causa y efecto.



*Nota:* Elaboración propia.

# Problema de investigación

## Formulación del problema

¿Cómo integrar los equipos del proyecto de gestión de datos de Sanambiente para que cumplan con criterios de calidad y robustez hacia el futuro?

## Sistematización del problema

* ¿Cuáles elementos de la ISO/IEC 29110 se pueden ajustar a las características del proyecto Sanambiente?
* ¿Cuáles son las herramientas propuestas por la ISO/IEC 29110 que permitan la integración de los diferentes equipos de trabajo del proyecto Sanambiente?
* ¿Cómo asegurar que el proyecto Sanambiente cumple adecuadamente un marco de calidad?
* ¿Cómo asegurar que el marco propuesto cumple los estándares seleccionados para enmarcar el proyecto?

# Objetivos

## Objetivo general

* Establecer un marco de trabajo siguiendo los lineamientos dados por la ISO/IEC 29110 utilizando como caso de estudio la integración de procesos del proyecto de la empresa Sanambiente.

## Objetivos específicos

1. Identificar los procesos y medidas propuestas por la ISO/IEC 29110 que se adecúen a las características del gestor de datos ambientales de la empresa Sanambiente.
2. Establecer la ruta de procesos y herramientas que permitan la integración de los equipos del proyecto de la empresa Sanambiente de acuerdo a la ISO/IEC 29110.
3. Aplicar el marco de trabajo identificado en la ISO/IEC 29110 al proyecto de la empresa Sanambiente.
4. Evaluar los resultados de la aplicación del marco de trabajo de acuerdo a la ISO/IEC 29110.

# Justificación

En el presente proyecto se realizó el aporte de un marco de trabajo para la integración de los equipos (calidad y desarrollo), el cual se llevó a cabo tomando como base la norma ISO/IEC 29110, la norma está conformada por 5 partes. La 1 parte presente una visión general de las técnicas por el estándar, la parte 2 se enfoca en los perfiles de ingeniería de softwares y los que se manejan a lo largo de la ISO/IEC 29110. La parte 3 se ha definido como guías para los proceso y cumplimiento de los requisitos para los perfiles. La parte 4 define todos los perfiles que están contenidos en un grupo de perfil. Por último, la parte 5 brinda una guía para la gestión y la implementación del software.

En el presente trabajo se implementó la parte 5 del estándar. Dado el contexto del proyecto propuesto por la empresa Sanambiente y los recursos que debían participar para su desarrollo, se hizo necesario el incluir estrategias de gestión para garantizar su éxito, y esta parte de la norma las proporcionaba.

La norma nos ofrece perfiles y paquetes de despliegue, en los perfiles existe el genérico donde se divide en subgrupos que van desde el nivel más bajo hasta el más alto como el perfil de entrada, perfil básico, perfil intermedio, perfil avanzado. En este trabajo se seleccionó el perfil básico por tiempo y viabilidad del proyecto propuesto por la empresa Sanambiente, además este permite ejecutar buenas prácticas y gestión para el desarrollo de una aplicación.

La ISO/IEC 29110 ofrece paquete de despliegue con elementos que pueden servir como apoyo para las VSEs en la implementación de un perfil definido, pero no se empleó dado que no se vio necesario al seguir este conjunto de elementos como medio para el control y seguimiento de los recursos que iban a intervenir en el proyecto de la empresa San ambiente.

Hay que destacar que el proyecto enfocó en los procesos de gestión para poder direccionar a los equipos, con sus objetico y requerimientos que requería el proyecto; adicional a esto, la investigación otorgó unos conocimientos para ser empleados en el futuro, en un área de trabajo que requiera la aplicación de un marco con unos lineamientos a seguir para realizar la integración efectiva de todos los equipos implicados en un proyecto. Además, la integración de los equipos conllevó a tener una mejor comunicación en el proceso de la realización del aplicativo obteniendo avances significativos gracias a la participación de todos en conjunto, y dando como resultados el desarrollo de software, y un marco de trabajo para ser base de implementación en proyectos futuros.

El proyecto realizado está vinculado particularmente al programa académico de Ingeniería de sistemas, y se pusieron en práctica algunas de las temáticas aprendidas a lo largo del este.

# Marco de referencia

## Antecedentes

Para el desarrollo del presente trabajo, se hizo necesario el analizar diferentes proyectos que tuvieran el mismo o similar objeto de estudio, con el fin de poder ser utilizados como guía. La información que será mostrada en este marco es de aquellos proyectos que tienen relación con los temas que el documento en mención aborda: ISO/IEC 29110 y Metodología ICONIX. El tipo de revisión que se hizo a estos proyectos está enfocado en la parte donde se tratan los temas mencionados anteriormente, qué uso hicieron de estos temas en sus proyectos, cómo lo ejecutaron y qué resultados les dieron. Para la realización de este marco se hizo el análisis y estudio de cinco documentos (tesis concretamente); pero entre ellos, fueron cuatro los que tenían mejor dominio de los temas y mostraban claramente sus aplicaciones al detalle.

Según (Madruñero Padilla, 2018) cuyo proyecto denominado “Implementación del estándar ISO/IEC 29110 en el proceso de desarrollo de software de la dirección de desarrollo tecnológico e informático de la universidad técnica del norte” este trabajo se manejó la ISO/IEC 29110 donde se llevó a cabo la gestión de procesos de desarrollo de software en la universidad técnica del norte, de la ciudad Ibarra-Ecuador, donde se buscaba la formalidad en todas las etapas del proceso de desarrollo mejorando la gestión del desarrollo permitiendo incorporar hitos de control en el proceso y su gestión para lograr una toma de decisiones oportuna en el proyecto.

Como resultado que se obtuvo en esa investigación, fue la implementación del estándar ISO/IEC 29110 ya que pudo establecer una relación con la metodología SCRUM, esto logrando aportar a la institución una metodología para la elaboración del desarrollo de software.

En la tesis (Laporte et al., 2016) bajo el título general de “ THE IMPLEMENTATION OF ISO/IEC 29110 SOFTWARE ENGINEERING STANDARDS AND GUIDES IN VERY SMALL ENTITIES” se demuestra que el ISO/IEC 29110 tiene perfiles de ciclo de vida para empresas muy pequeñas (de desarrollo de software) que deben seguir para que puedan elaborar software con altos estándares de calidad, así mismo, este estándar permite que las empresas pequeñas puedan hacer gestión a sus proyectos que permite tener un seguimiento a los proceso que se realizan, dichos procesos ayudan al mejoramiento y el rendimiento de la empresa.

El resultado que se puede obtener por este proyecto es que las empresas de desarrollo de software pueden hacer software que cumplan con los estándares de calidad debido a ya que la ISO/IEC 29110 ayuda a las pequeñas empresas en desarrollar software de calidad, como resultado de la implementación de esta ISO a cualquier empresa pequeña contribuye a ser competitiva en el mercado nacional o internacional.

Según la tesis de (Legaria Islas, 2018) con el título de “IMPLEMENTACION DE PROCESO ORGANIZACIONAL DE GESTION DE PROYECTOS EN DEVELOPIT” donde nos habla de cómo la ISO/IEC 29110 cumple con los proceso de la gestión de proyecto, DevelopIT como empresa necesita una norma que se adecue a la compañía para eso se utiliza la ISO/IEC 29110 ya que tiene menos procesos respecto a la gestión de proyectos, estos procesos que tiene la ISO/IEC 29110 se adaptara para los proyectos de infraestructura tecnológica que implementa DevelopIT esto permita a la organización pueda ejecutar proyectos considerados complejos y no complejos.

Se puede concluir que, para implementar un proyecto de gestión de proyectos en una pequeña empresa es necesario que esta sea adaptada a las necesidades de la organización, En ese sentido la norma ISO/IEC 29110 cumple con las características para ser adaptada a pequeñas organizaciones. También es posible que otras organizaciones con características similares a DevelopIT puedan hacer uso de este proceso de gestión de proyectos, ya que, al basarse en una norma, y al diseñarse con la intención de adaptarse a las características de pequeñas organizaciones, debiera ser por lo tanto útil y usable por organizaciones con similares características, es decir, pequeñas organizaciones, que no tienen personal especializado en gestión de proyectos, sus recursos son escasos y que los roles de los integrantes son multitarea.

De acuerdo (PORRAS, 2019) en el estudio “METODOLOGÍA ÁGIL ICONIX EN LA CALIDAD DEL PRODUCTO SOFTWARE, LIMA, 2017” tiene como objetivo de que la metodología ágil ICONIX brinda proceso para el desarrollo de software mediante la aplicación de técnicas probadas y aceptadas de ingeniería de software para empresas micro, pequeñas y medianas esto permitirá que este tipo de empresas tenga un mejoramiento de desarrollo de software, además puede tener productos aceptables y sostenible.

El resultado que se obtuvo de este proyecto fue la implementación de la metodología ICONIX que ayudo mejorar la calidad del desarrollo del software que se estaba buscando, gracias a esta metodología que brinda pasos a seguir donde se enfoca en cómo se debe desarrollar un sistema, Este proyecto ayudara a la empresa que brinda el servicio de desarrollo de software nuevas formas de hacer aplicaciones con agilidad que tiene beneficios como: disminuir el tiempo y los costos.

## Marco conceptual

En esta sección se hará una breve descripción de aquellos conceptos claves que fueron utilizados durante el desarrollo de este proyecto, con el fin de mejorar la comprensión de todo el contenido que en este documento se presenta al lector.

### Proceso de software

Es un marco de trabajo de las tareas que se necesitan para construir software de alta calidad. Un proceso de software establece la perspectiva que se debe elegir cuando el software es manipulado por la ingeniería (Pressman, 2002, p. 13). Es el conjunto de actividades que se relacionan y enfocan en la creación de un producto de software. Este software elaborado puede ser un producto nuevo, la modificación o una software ya existente (Ramos & Mendoza, 2014, p. 13).

### **Metodología de desarrollo de software**

Es aquella que posee la función de realizar la organización, planificación y el control del proceso de desarrollo de software (Hugues et al., 2010).

Una metodología de desarrollo de software especifica cómo hay que dividir el proyecto por etapas o fases, las tareas que se deben llevar a cabo en cada una, las salidas que se deben producir y cuando producir, las herramientas que hay que emplear, las restricciones apropiadas que se deben aplicar, y la manera en que se debe administrar y llevar el control pertinente sobre un proyecto (Cataldi, 2000, p. 40).

### **Lenguaje Unificado de Modelado (Unified Modeling Language (UML))**

El Lenguaje Unificado de Modelado, es un lenguaje estándar de modelado para productos de software, que proporciona la visualización, definición, elaboración y documentación de aquellos artefactos de sistemas en los que un producto de software posee una labor o función significativa. En esencia, UML permite a los desarrolladores ver los logros de su trabajo representados en esquemas o diagramas definidos (Jacobson et al., 2000, p. 407). Este lenguaje fue creado por un grupo de personas con amplio conocimiento en la ingeniería de software: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995 (González Toste et al., 2017, p. 7).

Como UML posee su enfoque en la descripción de métodos y procesos, se facilita la definición de las interacciones que deben llevarse a cabo por parte de los programadores y los usuarios con el software que se está desarrollando, a través de diferentes tipos de diagramas. Estos se encuentran categorizados de la siguiente manera (ver tabla 2) (Matla Cruz, 2014, pp. 24-26):

Tabla 2. Diagramas de UML por categoría.

|  |  |  |
| --- | --- | --- |
| **De estructura** | **De comportamiento** | **De interacción** |
| Diagrama de clases | Diagrama de estado | Diagrama de secuencia |
| Diagrama de objetos | Casos de uso | Diagrama de colaboración |
| Diagrama de componentes |  |  |
| Diagrama de paquetes |  |  |

*Nota:* Elaboración propia con base en Matla Cruz, E. O. (2014). *Desarrollo de software guiado por la norma ISO/IEC 29110 y Scrum: SIDEP v.2.0*. Universidad Nacional Autónoma de México, Ciudad de México.

### **Calidad de software**

El concepto de calidad planteado en la norma (UNE, 1995), expresa que “la calidad es el conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explicitas o implícitas”. En el campo de la ingeniería de software, de acuerdo a la (IEEE, 1991), la calidad de software es el nivel con el que un sistema o proceso acata correctamente con los requerimientos definidos y satisface con certeza las necesidades que el cliente expresó.

### **Modelo de calidad de software**

Según (Scalone, 2006), los modelos de calidad son documentaciones que: abarcan en gran medida las mejores prácticas para dar soporte a los procesos que conlleva un producto de software, ofrecen apoyo en temas de gestión en los que cada organización hay que tener presentes y, permiten medir los avances del producto en materia de calidad.

## Marco teórico

En esta sección se presentarán y se darán a conocer algunas de las investigaciones que se realizaron en las áreas de gestión y desarrollo de proyectos de software, como lo son: tendencias actuales del área de gestión, comparaciones entre algunos estándares y metodologías de estas dos áreas, y finalmente definición, así como explicación de aquel estándar y metodología seleccionada para ser aplicada en el presente proyecto, gracias al análisis efectuado en las comparaciones mencionadas.

### Revisión de área

El objetivo de esta sección es dar a conocer las principales tendencias en modelos y estándares que en este momento están siendo utilizadas para la gestión de los proyectos de software en las organizaciones.

La “crisis del software” del año 1969 perdura hoy en día, llegando al punto en que aún están presentes los problemas asociados al fracaso de los proyectos. El Standish Group y Dove confirman que el fracaso de un proyecto es por la carencia de gestión en este (Calvo Manzano & García, 2008, p. 8).

La gestión de proyectos ha tenido un alto crecimiento en las últimas dos décadas, debido a la necesidad de muchas organizaciones de mejorar sus procesos, y tanto las herramientas como técnicas se están estableciendo cada vez más como aplicaciones estándares para la gestión de los proyectos, no sólo de software, sino también de las otras áreas (Roberts & Wallace, 2002, p. 43).

Es tanto el crecimiento, que las organizaciones líderes en modelos y estándares para la gestión, han tenido que ir creando o mejorando estos para su aplicación en los proyectos de alta complejidad de la actualidad. A continuación, se muestran algunos de los principales modelos y estándares que, hoy en día, son los más utilizados por las organizaciones para la gestión de sus proyectos (ver tabla 3).

Tabla 3. Estándares para la gestión de proyectos.

|  |  |  |  |
| --- | --- | --- | --- |
| **Modelo/Estándar** | **Organización** | **Primera publicación** | **Última publicación** |
| PMBOK | PMI | 1996 | 6ta. Edición, 09/2017 |
| APMBOK | APM | 1992 | 7ta. Edición, 05/2019 |
| ISO 10006 | ISO | 12/1997 | 3ra. Edición, 11/2017 |
| ISO 12207 | ISO | 1995 | 3ra. Edición, 11/2017 |
| ISO 21500 | ISO | 09/2012 | 1ra. Edición, 09/2012 |
| PRINCE2 | OGC | 1996 | 5ta. Edición, 06/2009 |
| CMMI-DEV | CMMI Institute | 2002 | Versión 2.0, 2018 |

*Nota:* Elaboración propia con información recopilada de: Estrada Reyes, J. N. (2015). Análisis de la gestión de proyectos a nivel mundial. *Palermo Business Review*, *12*, 38**,** ISO. (s. f.). *International Organization for Standardization* **y** OGC. (2009). *Éxito en la gestión de proyectos con PRINCE2*. The Stationery Office.

El uso de estos modelos y estándares cada vez está cobrando más fuerza en las organizaciones, dado que al aplicarlos en sus proyectos, están proporcionando consigo beneficios de mejora continua y satisfacción al cliente (Lugo García & Marín Sánchez, 2016).

La gestión de proyectos como toda disciplina está en constante evolución, debido a que el mercado de software así lo está demandando. Los proyectos requieren más tiempo por su complejidad, por sus grandes volúmenes de actividades y roles; y los modelos o estándares de gestión deben estar a la altura brindando las herramientas y elementos adecuados para encargarse de ellos de la mejor manera.

Cabe mencionar que esta disciplina no está completamente estandarizada, dicho de otro modo, no dispone de una única manera para ser aplicada, debido a la existencia de diferentes puntos de vista (PMI, ISO, CMMI Institute, etc.) (Vásquez González, 2007, p. 12). No obstante, el auge que ha tenido la gestión de proyectos a lo largo del tiempo es alto, y ha logrado que se internacionalice y posicione hoy en día como uno de los líderes del camino hacia el desarrollo de prácticas profesionales (Roberts & Wallace, 2002, p. 44).

Para terminar, sería propicio conocer un poco acerca de la guía más conocida y usada por las organizaciones para la gestión de sus proyectos: PMBOK. Por lo tanto, a continuación, se dará una breve explicación de esta.

#### 5.3.1.1 PMBOK

Es una guía de apoyo que proporciona los pasos necesarios para realizar proyectos de forma exitosa y explica a detalle aquellas variables que deben ser tenidas en cuenta para la ejecución de un proyecto (Estrada Reyes, 2015, p. 80).

PMBOK define formalmente un total de 5 grupos de procesos los cuales son: Inicio, Planeación; Ejecución; Monitoreo y Control; y Cierre.

Estos grupos son la base de partida para comprender la gestión de proyectos, cada uno de estos posee los pasos necesarios para que sean aplicados correctamente, de manera ordenada y con la probabilidad de que aumente el éxito (Estrada Reyes, 2015, p. 81).

### Comparación entre estándares para la gestión de proyectos

Se llevó a cabo la comparación entre dos estándares para la gestión de proyectos, con el fin de determinar cuál de estos era el ideal para ser aplicado en este caso.

Existe un amplio número de estándares que se pueden aplicar para la gestión de un proyecto, por lo cual, es muy difícil seleccionar uno de ellos; no obstante, sería propicio determinar cuáles de estos estándares pueden asegurar que el proyecto dé resultados con alta probabilidad de éxito.

Con el fin de seleccionar qué estándar era el adecuado para seguir y aplicar en este proyecto, se hizo la comparación entre la guía PMBOK y el estándar ISO/IEC 29110. En la tabla 4, se compararon estos dos estándares en cuanto a las características que cada uno posee.

Tabla 4. Comparación entre estándares para la gestión de proyectos.

|  |  |
| --- | --- |
| **ISO/IEC 29110** | **PMBOK** |
| Guía para los procesos de Gestión de proyectos e Implementación de software. | Guía para el proceso de Gestión de proyectos. |
| Las fases del proceso de gestión de proyectos son: Planificación del proyecto; ejecución del Plan del proyecto; evaluación y control del proyecto; y cierre del proyecto. | Las fases del proceso de gestión de proyectos son: Inicio, Planificación; Ejecución; Monitorización y control; y Cierre. |
| Estándar dirigido a empresas pequeñas. | Estándar dirigido a empresas de cualquier tamaño interesadas en mejorar la gestión de sus proyectos. |
| No define áreas de conocimientos para las fases. | Defina áreas de conocimientos y esta las asocia a las fases. |
| Es impulsada por los requisitos del cliente. | Es impulsada por los requisitos del cliente. |
| Está dirigida a empresas de desarrollo de software. | Está dirigida a empresas de cualquier área. |
| No es muy amplia la documentación que proporciona, y la que se encuentra, es destinada para el área de desarrollo de software. | Como apoyo, tiene una amplia documentación general de proyectos de distintas áreas. |
| Posibilidad de certificación ISO/IEC 29110. | Tiene dos niveles de certificación: PMP (Project Management Professional) y CAMP (Certificate Associate in Project Mangement). |
| Estándar creado por la ISO. | Estándar creado por el PMI. |

*Nota:* Elaboración propia con información recopilada de: Assaff, R. (s. f.). *PMBOK - El cuerpo de conocimientos de la gestión de proyectos*. 73-82**,** Fernández Parra, K., Garrido Saroza, A., Ramírez Martínez, Y., & Perdomo Bello, I. (s. f.). PMBOK y PRINCE 2 similitudes y diferencias. *Revista Científica*, *23*, 117-130. https://doi.org/10.14483/udistrital.jour.RC.2015.23.a9 **y** Puerta Marín, H. A., & Espinosa Bedoya, A. (2015). UNA ADAPTACIÓN DE PMBOK AL CICLO DE VIDA DE DESARROLLO DE PROYECTOS SOFTWARE EN PEQUEÑAS EMPRESAS. *VI Congreso Iberoamericano de Ingeniería de Proyectos - CIIP*.

Luego de haber realizado la comparación entre los dos estándares, se decidió aplicar la ISO/IEC 29110 en este caso, debido a que su enfoque son los proyectos de desarrollo de software; brinda todas las herramientas necesarias para llevar a cabo el seguimiento y control adecuado de todos los recursos que intervienen en estos; y está dirigida a las empresas pequeñas, como lo es Sanambiente.

### **Estándar ISO/IEC 29110.**

Es una norma creada por la ISO bajo el título *Ingeniería de Software – Perfiles del ciclo de vida para entidades muy pequeñas* (Very Small Enterprises (VSEs)) y se basa en subconjuntos de elementos normativos apropiados, conocidos como perfiles VSE. El propósito de los perfiles VSE es definir un subconjunto de normas internacionales para el contexto de VSE (ISO/IEC, 2011, p. vi).

La norma se desarrolló para mejorar la calidad del software ayudando en el rendimiento de los productos en pymes, donde puedan contar con los mismos niveles de competitividad en mercado de las grandes industrias. La ISO/IEC 29110 se encuentra divida en 5 partes, de acuerdo a un público objetivo (ver tabla 5):

Tabla 5. Público objetivo de la ISO/IEC 29110.

|  |  |  |
| --- | --- | --- |
| **ISO/IEC 29110** | **Título** | **Público objetivo** |
| Parte 1 | Panorama General | VSEs, Evaluadores, productores de estándares, vendedores de herramientas y vendedores de metodología |
| Parte 2 | Marco y Taxonomía | Productores de normas, proveedores de herramientas de metodología. No está diseñado para VSEs. |
| Parte 3 | Guía de evaluación | Asesores y VSEs |
| Parte 4 | Especificaciones del perfil | Productores de normas, proveedores de herramientas de metodología. No está diseñado para VSEs. |
| Parte 5 | Guía de gestión e ingeniería | VSEs |

*Nota:* ISO/IEC. (2011). *Software engineering - Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile*, p.vi.

En la parte 1 se definen los términos generales que van a ser usados a lo largo de todos los documentos que contiene la ISO/IEC 29110. Además, menciona las características y requisitos que una VSE debe tener para la aplicación de este estándar, y por último describe brevemente los perfiles y guías de este.

En la parte 2 se introducen los conceptos para el perfil estandarizado de ingeniería de software para las VSEs, y además define los términos comunes que se utilizarán en el estándar. También, se establece y explica la lógica definida para los perfiles. Por último, se especifican los elementos de todos los perfiles y se realiza la introducción de la taxonomía de estos.

En la parte 3 se definen las directrices de evaluación y los requisitos de cumplimiento para satisfacer un perfil. También contiene información útil para los desarrolladores de métodos y herramientas de evaluación. Esta parte es especialmente dirigida a las personas que están relacionadas directamente con el proceso de evaluación.

En la parte 4 se especifican todos los perfiles pertenecientes al grupo de perfiles genéricos. Este grupo es aplicable a las VSEs que no desarrollan productos de software críticos. Por último, esta parte está dirigida a los autores/proveedores de guías, herramientas y otros materiales de apoyo.

En la parte 5 se proporciona una guía para los procesos de gestión e ingeniería de software del perfil básico perteneciente al grupo descrito en la parte 4 (Ramos & Mendoza, 2014, pp. 22; 23).

El perfil genérico es la categoría general y está destinada a las empresas que no desarrollan productos de software críticos. Dentro de este perfil se encuentran varios subgrupos, en el nivel más bajo se encuentra el perfil de entrada y está dirigido a VSEs con no más de tres años de fundación o con un número de empleados menor o igual a seis. El perfil que sigue es el básico, dirigido a VSEs que desarrollan una sola aplicación con un solo equipo. El perfil intermedio está destinado a VSEs que elaboran más de un proyecto con más de un equipo. Y finalmente, en el nivel más alto se encuentra el perfil avanzado, orientado a VSEs que tienen la ambición de mantenerse competitivas en el mercado (Marín Ospina, 2020, p. 3).

La ISO/IEC 29110 proporciona un paquete de despliegue como herramienta de ayuda para las VSEs en la implementación de un perfil definido. Entre los elementos brindados, se encuentran: descripción técnica del documento, descripción detallada de los procesos, actividades, roles y productos, plantillas, ejemplos, y una lista de herramientas. Es importante resaltar que el paquete de despliegue no busca cambiar la metodología que una VSEs utiliza para la gestión de sus proyectos (ISO/IEC, 2011, p. 49).

El estándar define en la parte 5 los procesos de gestión de proyectos y desarrollo de software (los cuales están interrelacionados), y las prácticas que integran están basadas en la selección de la ISO/IEC 12207:2008, bajo el título *Sistemas e ingeniería de software – Procesos del ciclo de vida del software* e ISO/IEC 15289:2006, bajo el título *Sistemas e ingeniería de software – Contenido de los sistemas y productos de información del proceso del ciclo de vida del software (Documentación),* presentando los productos y documentos requeridos en cada etapa (ver figura 2) (ISO/IEC, 2011, p. 3).

Figura 2. Procesos del perfil básico.

Declaración de trabajo

**Proceso:** Gestión del proyecto

**Proceso:** Implementación del

Software

Configuración de Software

*Nota:* ISO/IEC. (2011). *Software engineering - Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile*, p.4.

A continuación, se realizará una breve descripción de cada uno de los procesos y las actividades que conllevan.

* **Proceso de gestión del proyecto (PM):**

Tiene como propósito administrar de manera jerárquica las tareas del proyecto de implementación de software, permitiendo cumplir con los objetivos del proyecto en la calidad, tiempo y costos esperados. Para este proceso se tienen las siguientes actividades: planificación del proyecto, ejecución del proyecto, evaluación y control del proyecto y por último cierre del proyecto.

El siguiente diagrama muestra de manera clara el flujo entre las actividades del proceso de gestión del proyecto, incluyendo sus productos más importantes y su relación (ver figura 3).

Figura 3. Diagrama del proceso de gestión del proyecto.



*Nota:* ISO/IEC. (2011). *Software engineering - Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile*, p.9.

1. **Proceso de implementación de software (SI):**

Tiene como propósito la realización sistemática de las actividades de análisis, construcción, integración y pruebas de software nuevos o modificados según los requisitos (ISO/IEC, 2011, p. 16). Para este proceso se tienen las siguientes actividades: inicio de la implementación del software, análisis de los requerimientos de software, diseño arquitectónico y detallado de programas informáticos, construcción de software, integración de software y pruebas, y por último la entrega de productos.

El siguiente diagrama muestra de manera clara el flujo entre las actividades del proceso de implementación de software, incluyendo sus productos más importantes y su relación (ver figura 4).

Figura 4. Diagrama del proceso de implementación de software.



*Nota:* ISO/IEC. (2011). *Software engineering - Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile*.

* **Productos o artefactos:**

La ISO/IEC 29110 define unos productos o artefactos obligatorios para su implementación. En la siguiente tabla se listan junto con sus descripciones y el proceso de donde provienen (ver tabla 6).

Tabla 6. Productos o artefactos según la ISO/IEC 29110.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Descripción** | **Proceso de donde proviene** |
| 1. | Acceptance Record (Registro de aceptación) | Documenta la aceptación del cliente de los entregables del proyecto. | Gestión del proyecto. |
| 2. | Change Request (Solicitud de cambio) | Cuando se requiere una modificación en partes de algún producto, se diligencia una de estas para solicitarla. | Implementación de software.  Cliente.  Gestión del proyecto. |
| 3. | Correction Register  (Registro de corrección) | Identifica las actividades establecidas para corregir una desviación o problema relacionado con la realización de un producto. | Gestión del proyecto. |
| 4. | Meeting Record (Acta de reunión) | Registro de los acuerdos con el equipo de trabajo del proyecto y con el cliente. | Gestión del proyecto. |
| 5. | Progress Status Record (Registro de estado de progreso) | Registro del avance en el proyecto con respecto al Plan del proyecto. | Gestión del proyecto. |
| 6. | Project Plan (Plan del Proyecto) | Presenta la planeación de todas las actividades del proyecto que se van a realizar con sus fechas. Esto asegura la finalización exitosa del proyecto. | Gestión del proyecto. |
| 7. | Project Repository (Repositorio del proyecto) | Almacenamiento de los productos y sus respectivas entregas del proyecto. | Gestión del proyecto. |
| 8. | Project Repository Backup (Copia de seguridad del repositorio del proyecto) | Repositorio de respaldo del repositorio principal, como medida de contingencia para poder realizar la recuperación de la información. | Gestión del proyecto. |
| **#** | **Nombre** | **Descripción** | **Proceso de donde proviene** |
| 9. | Software Configuration (Configuración del software) | Conjunto de productos de software identificados. Estos son:   * Especificación de requerimientos * Diseño de software * Registro de trazabilidad * Componentes de software * Software * Casos de prueba y procedimientos de prueba * Reporte de pruebas * Manual de operación * Manual de usuario * Manual de mantenimiento | Implementación de software. |
| 10. | Verification Results (Resultados de la verificación) | Documenta la ejecución de la verificación del plan del proyecto. | Gestión del proyecto.  Implementación de software. |
| 11. | Validation Results (Resultados de la validación) | Documento la ejecución de la validación. | Implementación de software. |

*Nota:* ISO/IEC. (2011). *Software engineering - Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile*, p.33-40.

Y finalmente, la ISO/IEC 29110 define unos roles para el desarrollo del proyecto. Estos son: analista, cliente, diseñador, programador, gestor del proyecto, líder técnico y equipo de trabajo.

### Comparación entre metodologías de desarrollo de software

Se llevó a cabo la comparación entre metodologías de desarrollo de software, con el fin de determinar cuál de estas era la ideal para ser aplicada en este caso.

Existe un amplio número de metodologías que se pueden aplicar para el desarrollo de software, por lo cual, es muy difícil seleccionar una de ellas; no obstante, sería propicio determinar cuáles de estas metodologías pueden contribuir al mejoramiento de la productividad en el desarrollo y la calidad de un producto de software.

Con el fin de seleccionar qué metodología era la adecuada para aplicar en este proyecto, se hizo la comparación entre Scrum, XP e Iconix. En la tabla 7, se compararon estas metodologías en cuanto a las características que cada una posee.

Tabla 7. Comparación entre metodologías de desarrollo de software.

|  |  |  |
| --- | --- | --- |
| **Scrum** | **XP** | **Iconix** |
| * Scrum es una metodología ágil que brinda una gestión de proyecto y donde su enfoque es iterativo e incremental. * Scrum emplea en su proceso tres elementos: * Planificación del blacklog. * Seguimiento de Sprint * Revisión del Sprint. * El ciclo de vida de este marco de trabajo está compuesto de cuatro fases: planeación, puesta en escena, desarrollo y entrega. | * XP también es una metodología ligera que se basa en la simplicidad. * Esta metodología está fundamentada en pruebas y error. * Los requisitos pueden cambiar a lo largo del desarrollo. * Los grupos de trabajo por lo general pueden estar conformados de 2 -12 personas, pero lo más principal es de 2 personas. | * Iconix es una pesada-ligera donde su proceso es simplificado a comparación con otras metodologías y además es iterativo e incremental. * Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo. * Iconix contiene las siguientes bases: * Revisión del diseño. * Análisis y diseño preliminar. * Implementación y pruebas. |

*Nota:* Elaboración propia con información recopilada de: Alva Salcedo, A. B., & Reyes Laynes, J. B. (2019). *Desarrollo e implementación de un sistema de ventasa basado en la metodologia scrum y xp el proceso de ventas de servicio de la empresa emsoir*. Universidad Autónoma del Perú, Amavizca Valdez, L. O., Garcia Ruiz, A. C., Jiménez López, E., Duarte Guerrero, G. L., & Vázquez Brindis, J. C. (2014). *Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software: Implementación y publicación de un sitio WEB para una empresa SPIN - OFF en el Sur de Sonora, México.* Universidad Tecnológica del Sur de Sonora. Lozano Argote, J., Bolivar Guzman, J., & Ramírez, J. J. (2019). *Desarrollo de los módulos de transferencia y comunicación de datos ambientales para la empresa de SANAMBIENTE de Cali*. Instituto Universitario Antonio Jose Camacho, Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (2013, septiembre 20). *Revisión de metodologías ágiles para el desarollo de software*. 11.

Luego de realizar la comparación, se seleccionó la metodología Iconix, dado que proporcionaba mediante sus fases, los elementos necesarios para la implementación del software de acuerdo con las características y necesidades propias de la empresa Sanambiente. La ISO/IEC 29110 contaba con una guía para la Implementación de software, pero los elementos que proporcionaba eran de difícil comprensión para ser aplicados y seguidos en este proyecto, por ende, se tomó la decisión de no emplearse.

### **Metodología Iconix.**

Metodología de desarrollo de software de proporción media, en la que el análisis y capacidad de su diseño se basa en UML (Unified Modeling Language) (Mnkandla & Dwolatzky, 2004, p. 1). Fue elaborada por Doug Rosenberg y Kendall Scott. Se dice que ICONIX se encuentra entre la complejidad de RUP (Rational Unified Processes) y la simplicidad de XP (eXtreme Programming), sin pasar por alto las tareas de análisis y de diseño que éste último no contempla (Amavizca Valdez et al., 2014, p. 3).

La esencia de ICONIX está en que un 80% de los casos pueden ser resueltos tan solo con un uso del 20% del UML, lo cual simplifica en gran medida el proceso, al dejar solo aquella documentación necesaria (ICONIX Brand Group, 2016, p. 1).

El proceso de ICONIX (ver figura 5) da facilidad en el paso de los casos de uso a la codificación de forma fiable, en el menor tiempo posible. Razón por la cual, la principal preocupación de ICONIX es el análisis y diseño por modelado de la producción de software. (Rosenberg et al., 2005, p. 41).

Figura 5. Procesos de ICONIX.



*Nota:* Rosenberg, D., Stephens, M., & Collins-Cope, M. (2005). *Agile Development with ICONIX Process*. United States of America: Apress, p.45.

* **Orígenes** (Rosenberg et al., 2005, p. 40):

El proceso de ICONIX se originó años antes de UML y el proceso unificado, como una agrupación que cuenta con las mejores técnicas de las metodologías originales que formaron UML: la Técnica de modelado de objetos (Object Modeling Technique (OMT)) de Jim Rumbaugh, el método Objectory de Ivar Jacobson y el método Booch de Grady Booch; debido a que las fortalezas y debilidades que estas metodologías poseían se complementaban entre sí.

* **Principales diagramas que maneja Iconix**

Como se mencionaba con anterioridad, ICONIX basa su análisis y diseño en UML, por lo tanto, hace uso de un conjunto de diagramas como apoyo para proveer un resultado óptimo. Estos diagramas son:

* + **Modelado de dominio:** es un glosario de proyecto o un diccionario de términos que se utilizan en el proyecto. Su principal función es asegurar que todos en el proyecto entiendan el problema que están enfrentando sin que haya equivocación (Rosenberg et al., 2005, p. 7).
  + **Diagrama de caso de uso:** describe la secuencia de acciones que un actor realiza dentro del sistema (Rosenberg et al., 2005, p. 42). Guía todo el proceso de la metodología ICONIX. Según (Bona, 2002, p. 68) es el centro conceptual del desarrollo.
  + **Diagrama de robustez:** es el encargado de identificar el primer conjunto de posibles objetos de primera estimación de cada caso de uso, todo logrado gracias al análisis del texto de estos. El diagrama se divide en tres estereotipos:
    - **Objetos de límite:** usados por los actores para comunicarse con el sistema.
    - **Objetos de entidad:** son objetos del modelado de dominio.
    - **Objetos de control:** son encargados de la integración entre los dos primeros objetos.
  + **Diagrama de secuencia:** diagrama que muestra todos los cursos posibles que los casos de uso pueden elegir y además muestra tanto los métodos pertenecientes a cada objeto como los mensajes que se producen al realizar una acción determinada al sistema.
* **Características de Iconix:**

Se destacan tres características fundamentales de ICONIX:

* + **Iterativo e incremental:** cuando se lleva a cabo el desarrollo del modelado de dominio y se precisan los casos de uso, se producen múltiples iteraciones. En el ciclo incremental se desarrolla el producto por partes para que se puedan integrar funcionalmente sin complicaciones y en cada iteración se procesa a su revisión y mejora (Fiestas Jacinto, 2015, p. 40) (Silva Ascuntar & García G., 2018, p. 34).
  + **Trazabilidad:** Cada paso culminado está ligado a un requerimiento. Su enfoque principal son las necesidades del usuario, no hay un momento en que durante el proceso se desvíe lejos de este (Fiestas Jacinto, 2015, p. 40).
  + **Aerodinámica del UML:** ofrece un uso flexible del UML, debido a que no exige la utilización de todos los diagramas que tiene ligados (Silva Ascuntar & García G., 2018, p. 34).

En (Rosenberg & Scott, 2001, p. 23) se presentan las fases principales de la metodología ICONIX. Estas fases cuentan con unos hitos específicos asociados, como se va a mostrar a continuación.

* **Análisis de requerimientos:**

Las actividades que se llevan a cabo en esta fase son (Rosenberg & Scott, 2001, p. 23) (Fiestas Jacinto, 2015, p. 46):

* Identificar en el “mundo real”, los objetos y todas las relaciones de agregación y generalización que tienen entre sí. En esta actividad se hace uso del **modelado de dominio**.
* Como sugerencia, crear un **prototipado** de la interfaz del sistema o unos diagramas de navegación con los que el cliente esté más familiarizado del sistema que propusieron.
* Realizar el análisis de los requerimientos e identificar los casos de uso y sus actores involucrados. En esta actividad se hace uso de los **diagramas de caso de uso**.
* Se deben organizar los casos de uso en grupos y para realizarlo se hace uso del **diagrama de paquete**.
* Realizar la asociación de los requerimientos funcionales que estén relacionados que se hallaron en los casos de uso y objetos de dominio.

**Hito:** Revisión de los requisitos.

* **Análisis y diseño preliminar:**

Las actividades que se llevan a cabo en esta fase son (Rosenberg & Scott, 2001, p. 24):

* Se deben escribir los **casos de uso**, como el flujo principal de las acciones, con la capacidad de contener el flujo alternativo y el flujo de excepción.
* Se realiza el **diagrama de robustez**. Ya que para cada caso de uso se debe realizar la localización de un conjunto de objetos y llevar a cabo la actualización del modelado de dominio.
* Culminar en su totalidad la modificación del **diagrama de clases.**

**Hito:** Revisión del diseño preliminar.

* **Diseño detallado:**

Las actividades que se llevan a cabo en esta fase son (Rosenberg & Scott, 2001, p. 24) (Bona, 2002, p. 64):

* Detallar los cursos posibles de cada caso de uso a través del **diagrama de secuencia,** e identificar los mensajes entre los diferentes objetos.
* Culminar en su totalidad el modelo estático, ultimando detalles del proyecto en el **diagrama de clase**.
* Comprobar junto con el equipo si el proyecto satisface los requerimientos definidos al inicio.

**Hito:** Revisión crítica del diseño.

* **Implementación:**

Las actividades que se llevan a cabo en esta fase son (Rosenberg & Scott, 2001, p. 25):

* Se escribe escribe/genera el **código** del software.
* Se realizan las pruebas unitarias y de integración al software creado.
* Se llevan a cabo las pruebas de aceptación del usuario del proyecto finalizado.

**Hito:** Entrega.

### Reconocimiento de las tecnologías utilizadas por los equipos de trabajo

En esta sección se hará una breve descripción de las principales tecnologías que fueron utilizadas por parte de los equipos de trabajo (gestión, desarrollo y calidad) para llevar a cabo sus actividades durante el desarrollo del macro proyecto propuesto por la empresa Sanambiente.

#### 5.3.6.1 Angular Framework

El objetivo principal de este framework es facilitar la creación de aplicaciones cliente en HTML y JavaScript o en el lenguaje TypeScript que se agrupa en JavaScript. Una aplicación consta de plantillas de componentes HTML, clases de componentes que se encargan de administrar estas, y además de la lógica que se encuentra en los servicios y componentes (Bolívar Guzmán et al., 2020, p. 16).

#### 5.3.6.2 JavaScript

Es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos y además se caracteriza por ser imperativo, dinámico y tipado (Bolívar Guzmán et al., 2020, p. 16).

#### 5.3.6.3 TypeScript

Lenguaje de programación de código abierto, orientado a objetos, ideal para ser usado en proyectos a gran escala (Bolívar Guzmán et al., 2020, p. 17).

#### 5.3.6.4 PostgreSQL

Es un gesto de base de datos relacional libre, de código abierto. Dirigido a la comunidad de desarrolladores. Está bajo licencia Berkeley Software Distribution (BSD) (Mina Posu & Romero Dorado, 2020, p. 33) (Bolívar Guzmán et al., 2020, p. 16).

#### 5.3.6.5 NodeJS

Es un entorno multiplataforma, que se encuentra en tiempo de ejecución, igualmente de código abierto. Útil para la creación de aplicaciones web altamente escalables (Bolívar Guzmán et al., 2020, p. 18).

#### 5.3.6.6 GitHub

Plataforma de desarrollo colaborativo de aplicativos, ideal para el alojamiento de proyectos mediante el sistema de control de versiones que esta proporciona Git.

#### 5.3.6.7 GitHub Desktop

Aplicación de escritorio, con interfaz gráfica que permite el fácil mantenimiento de versiones de un proyecto. Básicamente fue diseñado para brindar facilidad al realizar control de versiones sobre los proyectos mediante una interfaz gráfica amigable. Su principal diferencia con el software Git, es que este último se usa mediante líneas de comando por consola.

#### 5.3.6.8 Visual Studio Code

Editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Entre sus características incluye control integrado de Git, depuración, finalización inteligente de código y resaltado de errores por sintaxis (Mina Posu & Romero Dorado, 2020, p. 34).

#### 5.3.6.9 Jest

Plataforma de pruebas universal, la cual se adapta a cualquier biblioteca de JavaScript. Esta plataforma se puede ejecutar sobre proyectos que estén basado en Angular, Node, TypeScript, Vue, entre otros (Mina Posu & Romero Dorado, 2020, p. 34).

#### 5.3.6.10 Jmeter

Es un proyecto de Apache, que puede ser utilizado como herramienta para pruebas de carga, para analizar el rendimiento de una variedad de servicios. Además, es utilizada para realizar pruebas unitarias sobre conexiones de bases de datos y conexiones TCP genéricas (Mina Posu & Romero Dorado, 2020, p. 35).

## Marco legal

Todas las actividades propuestas y desarrolladas dentro del proyecto quedan inmersas dentro del acuerdo de cooperación interinstitucional realizado entre la empresa Sanambiente y la Institución Universitaria Antonio José Camacho, firmado en el mes de abril de 2019.

Conforme al (Estatuto de Investigaciones y Producción Intelectual, 2018, p. 8), un proyecto de investigación se puede definir como *“la unidad básica de trabajo que comprende un conjunto de actividades organizadas coherentemente en una propuesta de investigación…”* y para poder llevar a cabo estas actividades y lograr unos resultados concretos, se debe hacer uso de recursos tanto humanos como financieros, bibliográficos y de infraestructura.

Además, las normas internas de la Institución Universitaria Antonio José Camacho reglamentan sobre la Propiedad Intelectual, los derechos sobre las diversas formas de creación de productos derivados del talento y actividad investigativa en la Institución y *“…reconoce, respeta y se subordina a lo establecido en la Constitución Política de Colombia, así como en la ley 23 de 1982, sobre derechos de Autor, Decreto 1360 de 1989 sobre la inscripción del soporte lógico (software) en el Registro Nacional de derecho de autor…”*. Además, cuando un proyecto de investigación es aprobado por parte de la Institución, se debe suscribir el Acta de Propiedad Intelectual de este.

# Resultados

A partir del estudio realizado en el marco teórico sobre los lineamientos proporcionados por la ISO/IEC 29110, se decidió ejecutar la parte 5 del estándar en el presente trabajo, en vista de lo planteado por la empresa Sanambiente en su proyecto de software. Debido a los recursos que intervendrían en el desarrollo de este último, se hizo necesario el incluir estrategias de gestión para garantizar su éxito, y esta parte de la norma las proporcionaba. Por esta razón, fue que se eligió seguir este estándar y parte en concreto.

El perfil que se siguió dentro del grupo de perfiles genéricos fue el básico, debido al tiempo, número de actividades, productos e integrantes que el proyecto exigía. Además, se decidió no emplear el paquete de despliegue que brindaba el estándar, dado que no se vio necesario el seguir este conjunto de elementos como medio para el control y seguimiento de los recursos que iban a interceder en el desarrollo del proyecto propuesto la empresa Sanambiente. Se decidió seguir en su lugar únicamente los pasos descritos en los procesos contenidos en la norma.

Luego del análisis realizado al estándar, se identificaron dos procesos en la parte 5 de la ISO/IEC 29110: gestión de proyectos e implementación de software. Se determinó que para el desarrollo de este caso se ejecutaría únicamente el proceso de gestión de proyectos definido en esta parte del estándar, junto con sus fases y artefactos correspondientes; pero no el proceso de implementación de software, puesto que la documentación que exigía para ser aplicado era demasiada y no se ajustaba al contexto de este caso para ser adoptada. En su lugar, gracias al análisis de las comparaciones efectuadas en el marco teórico, se decidió seguir las fases de la metodología Iconix, dado que proporcionaba mediante estas, los elementos necesarios para la implementación del software de acuerdo a las características y necesidades propias de la empresa Sanambiente. Además, se incorporaron las fases de la IEEE 829 para guiar la ejecución de las pruebas de calidad que se debían realizar en el proceso de implementación del software, puesto que dieron muy buen resultado en proyectos anteriores (Aguado Álvarez, 2019) (Mina Posu & Romero Dorado, 2020).

Del proceso de gestión proporcionado en la parte 5 del estándar ISO/IEC 29110, las fases y artefactos propuestos que se adecuaron y aplicaron para la elaboración de este marco de trabajo fueron:

**Fase 1: Planificación del proyecto**

**Artefacto(s):**

* Plan del proyecto.
* Repositorio del proyecto
* Resultados de la verificación del proyecto.

**Fase 2: Ejecución del plan del proyecto**

**Artefacto(s):**

* Acta de reunión.
* Registro de estado de progreso.
* Solicitud de cambio.

**Fase 3: Evaluación y control del proyecto**

**Artefacto(s):**

* Registro de corrección.

**Fase 4: Cierre del proyecto**

**Artefacto(s):**

* Registro de aceptación.

Cada uno de los artefactos se elaboró de acuerdo con el contexto del proyecto de la empresa Sanambiente, eliminando algunos ítems que por defecto el estándar tenía definidos para estos y adicionando otros de los que carecían. Además, se optó por no incluir el artefacto de Copia de seguridad del repositorio que se elabora en la Fase 2, ya que no se vio necesario tener un repositorio adicional como respaldo del repositorio principal; sin embargo, se utilizaron los equipos personales como respaldos adicionales, a modo de contingencia ante cualquier pérdida de información.

La ruta de procesos y herramientas establecidas para este proyecto fueron (ver figura 6):

Figura 6. Ruta de procesos y herramientas del proyecto.



*Nota:* Marín Ospina, B. E. (2020). *Adecuación de la norma ISO/IEC 29110 e IEEE 829 para la gestión de proyectos de desarrollo con metodología Iconix*. 15.

La metodología Iconix toma como insumo el plan de proyecto donde se describieron alcance, tareas, duración estimada y se definió el cronograma a seguir. A partir de los requerimientos del cliente se construyó un prototipo base por parte del equipo de desarrollo y a partir de este producto se ejecutaron las siguientes fases:

* En la fase de análisis de requisitos, se desarrollaron las clases conceptuales del problema y el modelo de casos de uso, y paralelamente se iban diligenciando solicitudes de cambio y corrección si lo ameritaba. Adicional se actualizaba el repositorio y el registro de estado de progreso del proyecto.
* En el análisis y diseño preliminar se refinó el modelado del dominio, actualizando el repositorio, solicitudes de cambio y corrección y el registro de estado de progreso del proyecto.
* En la etapa de diseño detallado se realizaron los diagramas de clases a partir de los diagramas de secuencias. También se diseñaron los documentos de plan de pruebas, especificación del diseño de pruebas, especificación de los casos de pruebas, los procedimientos de pruebas y los reportes de transmisión propuestos por la IEEE 829. Además, se sigue actualizando el repositorio y el registro de estado de progreso del proyecto.
* Los atributos que se querían probar, se identificaron a partir de la norma ISO/IEC 25010 en acuerdo con el usuario y se listan a continuación: adecuación funcional, eficiencia, usabilidad y seguridad. Adicional, se sigue actualizando el registro de estado de progreso, el repositorio del proyecto y se diligencian las solicitudes de cambio y corrección que se ameritan.
* En la etapa de implementación se desarrollaron los productos de software acordados en el plan del proyecto, generando logs de pruebas, reportes de incidentes y reporte general de pruebas respectivos, y finalmente, actualizando el registro de estado progreso y el repositorio del proyecto (Marín Ospina, 2020, pp. 11; 12).

A continuación, se procede a detallar la aplicación de cada una de las fases y artefactos del marco de trabajo en la ejecución del proyecto de la empresa Sanambiente.

## Fase 1: Planificación del proyecto

En esta fase se realizó la apropiada documentación en detalle de los elementos necesarios de planificación para llevar a cabo la gestión del proyecto, incluyendo la adecuación del proceso de implementación de software con las fases de la metodología Iconix e IEEE 829 para el desarrollo del programa y la ejecución de pruebas de calidad sobre este. Se produjeron los siguientes artefactos:

### **Crear el plan del proyecto.**

Plan que fue socializado y aprobado por el cliente (empresa Sanambiente) para ser ejecutado sobre el proyecto. A continuación, se presenta este:

**Plan del proyecto de la empresa Sanambiente**

**Descripción del producto:**

**Propósito:**

Se elaborará un sistema que va a permitir transferir y comunicar los datos ambientales de la empresa Sanambiente. Este sistema involucra el desarrollo de una arquitectura robusta, bajo diseño centrado en el usuario y riguroso seguimiento a estándares de calidad internacionales.

**Requisitos generales del cliente**

**Comunicación y traspaso de información.**

1. El sistema debe permitir el enlace y transferencia de datos de los datalogger mediante Internet.
2. El sistema debe permitir crear, modificar, borrar, visualizar comandos para ser utilizados por el protocolo de comunicación FTP.

**Alcance**

El alcance del proyecto será el desarrollo de los requerimientos 1 y 2 del módulo de comunicación y transferencia de información. Dependiendo del tiempo que quede después de haberse solucionado estos requerimientos, se podría avanzar con la solución de otros.

**Objetivos para el plan del proyecto siguiendo la ISO/IEC 29110**

**General**

* Desarrollar una herramienta que permita gestionar los datos ambientales de la empresa San Ambiente de Cali.

**Específicos**

* Establecer un marco de trabajo siguiendo los lineamientos dados por la ISO 29110 utilizando como caso de estudio la integración de procesos del proyecto de la empresa Sanambiente.
* Desarrollar un módulo que permita conectar y transferir información de los diferentes datalogger de la empresa Sanambiente con sede en Cali.
* Aplicar un modelo de calidad al módulo de comunicación de información del sistema de gestión de datos ambientales de la empresa Sanambiente de Cali.

**Entregables**

Los entregables que se darán al cliente por parte del equipo de desarrollo están basados en los establecidos por la metodología ICONIX.

Tabla 8. Entregables del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fase 1:**  **Análisis de requerimientos** | **Fase 2:**  **Análisis y diseño preliminar** | **Fase 3:**  **Diseño detallado** | **Fase 4:**  **Implementación** |
| Realizar el prototipado (diseño de la interfaz). | Diagrama de robustez. | Diagrama de secuencia. | Escribir/generar el código (Desarrollo de los módulos de comunicación y transferencia de datos). |
| Diagramas de caso de uso. | Diagrama de clases. | Diagrama de clases refinado. | Refinamiento de acuerdo con las pruebas unitarias. |
| --------------------- | --------------------- | Planes de prueba | Realizar pruebas de aceptación del usuario. |

*Nota:* Elaboración propia.

Los entregables que se darán al cliente por parte del equipo de calidad están basados en los establecidos por la IEEE-829.

Tabla 9. Entregables del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fase 1: Planeación** | **Fase 2:**  **Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)** | **Fase 3:**  **Ejecución de pruebas** | **Fase 4: Finalización de pruebas** |
| Diagrama de casos de uso | Modelado de dominio. | Logs de pruebas. | Reporte final de pruebas. |
| Validación de diagrama de casos de uso | Plan de pruebas. | Realizar pruebas unitarias y de funcional. | -------------- |
| Tipos de pruebas. | Especificación del diseño de pruebas. | Reporte de incidencias. | ------------- |
| ------------- | Especificación de casos de prueba. | ----------------- | ------------- |
| ------------- | Procedimiento de pruebas. | ------------- | ------------- |
| ------------- | Reporte de trasmisión de pruebas. | ------------- | -------------- |
|  | | | |

*Nota:* Elaboración propia.

**Tareas y su duración estimada**

**Nota para los entregables del equipo de desarrollo:** En la fase 1, los diagramas de casos de uso pasan a ser responsabilidad del equipo de calidad; por lo tanto, en esta fase sólo queda la actividad de Realizar el prototipado.

En la fase 3, el refinamiento del diagrama de clases y los planes de prueba se realizarán de forma implícita en la fase 4, por lo cual no se mencionará como tarea independiente. Adicionalmente el diagrama de secuencia de la fase 3 se pasará a realizar en la fase de 2 en conjunto con el diagrama de clases; por lo tanto, la fase 3 se elimina por completo.

En la fase 4, se adiciona como entregable el diccionario de funciones o módulos.

Finalmente, tras los cambios realizados, las fases quedan de la siguiente manera:

**Fase 1:** Análisis de requerimientos.

* Realizar el prototipado (diseño de la interfaz).
* **Fase 2:** Análisis y diseño preliminar.
* Diagrama de clases.
* Diagrama de secuencia.
* **Fase 3:** Implementación.
* Escribir/generar código.
* Diccionario de funciones o módulos.
* Refinamiento de acuerdo con las pruebas unitarias.
* Realizar pruebas de aceptación del usuario.

Las tareas y su duración estimada serán definidas y estipuladas por casos de uso, que se obtuvieron del análisis a los requisitos del proyecto por parte del equipo de calidad.

Se realizaron agrupaciones de dos o tres casos de uso para ser entregados en la misma fecha debido a su poca complejidad, obteniendo así un ahorro de tiempo en días que se puede aplicar para otros casos de uso que sí requieren mayor tiempo para ser desarrollados y posteriormente ser entregados.

**Equipo de desarrollo (Requerimientos de comunicación y transferencia de información 1 y 2)**

**Casos de uso #04 y #08**

**Nombre CU04: Crear región (tabla parámetro)**

**Nombre CU08: Crear ciudad (tabla parámetro)**

Tabla 10. Duración estimada de las tareas 1.1 - 3.2 (CU04 y CU08) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Plantilla Bootstrap. 2. Realizar el prototipado (diseño de la interfaz).  * Documentar. | 6 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | 3 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | 7 |

*Nota:* Elaboración propia.

**Casos de uso #16 y #10**

**Nombre CU16: Crear mantenimiento (tabla parámetro)**

**Nombre CU10: Crear alerta (tabla parámetro)**

Tabla 11. Duración estimada de las tareas 1.1 - 3.2 (CU16 y CU10) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | 6 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | 3 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | **7** |

*Nota:* Elaboración propia.

**Casos de uso #06, #07**

**Nombre CU06: Crear organización (tabla parámetro)**

**Nombre CU07: Crear categoría (tabla parámetro)**

Tabla 12. Duración estimada de las tareas 1.1 - 3.2 (CU06 y CU07) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | 6 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | 3 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | 7 |

*Nota:* Elaboración propia.

**Caso de uso #05**

**Nombre CU05: Crear estación (tabla parámetro)**

Tabla 13. Duración estimada de las tareas 1.1 - 3.2 (CU05) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | 7 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | 6 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | 18 |

*Nota:* Elaboración propia.

**Caso de uso #01**

**Nombre CU01: Conectar con estación FTP**

Tabla 14. Duración estimada de las tareas 1.1 - 3.2 (CU01) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | 7 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | 6 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | 18 |

*Nota:* Elaboración propia.

**Casos de uso #09 y #11**

**Nombre CU09: Crear rango (tabla parámetro)**

**Nombre CU11: Crear base de tiempo (tabla parámetro)**

Tabla 15. Duración estimada de las tareas 1.1 - 3.2 (CU09 y CU11) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | 6 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | 3 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | 7 |

*Nota:* Elaboración propia.

**Fase 3: Actividades 3.3 y 3.4**

**Casos de uso #04 y #08**

**Nombre CU04: Crear región (tabla parámetro)**

**Nombre CU08: Crear ciudad (tabla parámetro)**

Tabla 16. Duración estimada de las tareas 3.3 y 3.4 (CU04 y CU08) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | 6 |

*Nota:* Elaboración propia.

**Casos de uso #16 y #10**

**Nombre CU16: Crear mantenimiento (tabla parámetro)**

**Nombre CU10: Crear alerta (tabla parámetro)**

Tabla 17. Duración estimada de las tareas 3.3 y 3.4 (CU16 y CU10) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | 6 |

*Nota:* Elaboración propia.

**Casos de uso #06 y #07**

**Nombre CU06: Crear organización (tabla parámetro)**

**Nombre CU07: Crear categoría (tabla parámetro)**

Tabla 18. Duración estimada de las tareas 3.3 y 3.4 (CU06 y CU07) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | 6 |

*Nota:* Elaboración propia.

**Caso de uso #05**

**Nombre CU05: Crear estación (tabla parámetro)**

Tabla 19. Duración estimada de las tareas 3.3 y 3.4 (CU05) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | 9 |

*Nota:* Elaboración propia.

**Caso de uso #01**

**Nombre CU01: Conectar con estación FTP**

Tabla 20. Duración estimada de las tareas 3.3 y 3.4 (CU01) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | 9 |

*Nota:* Elaboración propia.

**Casos de uso #09 y #11**

**Nombre CU09: Crear rango (tabla parámetro)**

**Nombre CU11: Crear base de tiempo (tabla parámetro)**

Tabla 21. Duración estimada de las tareas 3.3 y 3.4 (CU09 y CU11) del equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración (días)** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | 6 |

*Nota:* Elaboración propia.

**Equipo de calidad (Requerimientos de comunicación y transferencia de información 1 y 2)**

**Casos de uso #04 y #08**

**Nombre CU04: Crear región (tabla parámetro)**

**Nombre CU08: Crear ciudad (tabla parámetro)**

Tabla 22. Duración estimada de las tareas 1.1 - 4.1 (CU04 y CU08) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Duración(días)** | |
| **1** | **Fase 1: Planeación.**  **Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.   1. Diagrama de cosos de uso 2. Validación de diagramas de casos de uso. 3. Tipos de pruebas. | 9 | |
| **2** | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25110(adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio 2. Plan de pruebas. 3. Especificación del diseño de pruebas. 4. Especificación de casos de prueba 5. Procedimiento de pruebas 6. Reporte de trasmisión de pruebas. | 17 | | |
| **3** | **Fase 3: Ejecución de pruebas**  **Obj:** Se busca con los datos de la fase 2 ejecutar las pruebas.  Actividades:   1. Logs de prueba 2. Realizar Pruebas 3. Reporte de incidencias | | 7 | |
| **4** | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  Actividades   1. Reporte final de pruebas | | 2 | |

*Nota:* Elaboración propia.

**Casos de uso #16 y #10**

**Nombre CU16: Crear mantenimiento (tabla parámetro)**

**Nombre CU10: Crear alerta (tabla parámetro)**

Tabla 23. Duración estimada de las tareas 1.1 - 4.1 (CU16 y CU10) del equipo de calidad.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración(días)** |
| **1** | **Fase 1: Planeación Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de cosos de uso 2. Validación de diagramas de casos de uso. 3. Tipos de pruebas | 9 |
| **2** | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010(adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad) Obj:** Se busca elaborar las pruebas a la cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio 2. Plan de pruebas. 3. Especificación del diseño de prueba. 4. Especificación de casos de prueba. 5. Procedimiento de pruebas. 6. Reporte de trasmisión de pruebas. | 19 |
| **3** | **Fase 3: Ejecución de pruebas Obj:** Se busca con los datos de la fase 2 ejecutar las pruebas.  **Actividades:**   1. Logs de prueba 2. Realizar Pruebas 3. Reporte de incidencias | 3 |
| **4** | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas | 1 |

*Nota:* Elaboración propia.

**Casos de uso #06 y #07**

**Nombre CU06: Crear organización (tabla parámetro)**

**Nombre CU07: Crear categoría (tabla parámetro)**

Tabla 24. Duración estimada de las tareas 1.1 - 4.1 (CU06 y CU07) del equipo de calidad.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración(días)** |
| **1** | **Fase 1: Planeación Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de casos de uso. 2. Validación de diagramas de casos de uso. 3. Tipos de pruebas | 9 |
| **2** | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010(adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio 2. Plan de pruebas 3. Especiación del diseño de pruebas. 4. Especificación de casos de prueba. 5. Procedimiento de pruebas 6. Reporte de trasmisión de pruebas. | 19 |
| **3** | **Fase 3: Ejecución de pruebas Obj:** Se busca con los datos de fase 2 ejecutar las pruebas.  **Actividades:**   1. Logs de prueba. 2. Realizar Pruebas. 3. Reporte de incidencias. | 3 |
| **4** | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | 1 |

*Nota*: Elaboración propia.

**Caso de uso #05**

**Nombre CU05: Crear estación (tabla parámetro)**

Tabla 25. Duración estimada de las tareas 1.1 - 4.1 (CU05) del equipo de calidad.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración(días)** |
| **1** | **Fase 1: Planeación Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de casos de uso. 2. Validación de diagramas de casos de uso. 3. Tipos de pruebas | 9 |
| **2** | **Fase 2: Preparación de pruebas según las características de la ISO/IEC25010(adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a la cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio 2. Plan de pruebas. 3. Especificación del diseño de prueba. 4. Especificación de casos de prueba. 5. Procedimiento de pruebas. 6. Reporte de trasmisión de pruebas. | 19 |
| **3** | **Fase 3: Ejecución de pruebas Obj:** Se busca con los datos de fase 2 ejecutar las pruebas.  Actividades:   1. Logs de prueba. 2. Realizar Pruebas. 3. Reporte de incidencias. | 3 |
| **4** | **Fase 4: Finalización de pruebas Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | 1 |

*Nota***:** Elaboración propia.

**Caso de uso #01**

**Nombre CU01: Conectar con estación FTP**

Tabla 26. Duración estimada de las tareas 1.1 - 4.1 (CU01) del equipo de calidad.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración(días)** |
| **1** | **Fase 1: Planeación Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de casos de uso. 2. Validación de diagramas de casos de uso. 3. Tipos de pruebas | 9 |
| **2** | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010(adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a la cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio 2. Plan de pruebas. 3. Especificación del diseño de prueba. 4. Especificación de casos de prueba. 5. Procedimiento de pruebas. 6. Reporte de trasmisión de pruebas. | 19 |
| **3** | **Fase 3: Ejecución de pruebas Obj:** Se busca con los datos de fase 2 ejecutar las pruebas.  Actividades:   1. Logs de prueba. 2. Realizar Pruebas. 3. Reporte de incidencias. | 3 |
| **4** | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | 1 |

*Nota*: Elaboración propia.

**Casos de uso #09 y #11**

**Nombre CU09: Crear rango (tabla parámetro)**

**Nombre CU11: Crear base de tiempo (tabla parámetro)**

Tabla 27. Duración estimada de las tareas 1.1 - 4.1 (CU09 y CU11) del equipo de calidad.

|  |  |  |
| --- | --- | --- |
| **#** | **Descripción** | **Duración(días)** |
| **1** | **Fase 1: Planeación Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de casos de uso. 2. Validación de diagramas de casos de uso. 3. Tipos de pruebas | **9** |
| **2** | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010(adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a la cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio 2. Plan de pruebas. 3. Especificación del diseño de prueba. 4. Especificación de casos de prueba. 5. Procedimiento de pruebas. 6. Reporte de trasmisión de pruebas. | **19** |
| **3** | **Fase 3: Ejecución de pruebas**  **Obj:** Se busca con los datos de fase 2 ejecutar las pruebas.  Actividades:   1. Logs de prueba. 2. Realizar Pruebas. 3. Reporte de incidencias. | **5** |
| **4** | **Fase 4: Finalización de pruebas Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | **4** |

*Nota*: Elaboración propia.

**Recursos**

**-Recursos humanos:**

* Equipo de gerencia del proyecto.
* Equipo de desarrollo.
* Equipo de calidad.

**-Estándares:**

* ISO/IEC 29110.
* ISO/IEC 25010.

**-Equipos:**

* Los equipos básicos de cómputo que serán utilizados para llevar a cabo todo el desarrollo del proyecto poseen las siguientes configuraciones:
* Sistema operativo: Windows 10
* Procesador: i5 7200U
* RAM: 8GB
* Disco duro: 1TB

Cabe mencionar que estos equipos de cómputo son portátiles.

**-Herramientas:**

* Todas las herramientas que se utilizarán para el desarrollo del proyecto serán de uso libre, ejemplo: Visual Studio Code.

**Composición del equipo de trabajo:**

El equipo de trabajo se encuentra conformado de la siguiente manera:

* Equipo de gerencia del proyecto: conformado por 2 personas.
* Equipo de desarrollo: conformado por 3 personas.
* Equipo de calidad: conformado por 2 personas.

**Calendario de las tareas del proyecto**

Las fechas de las tareas de cada equipo de trabajo serán definidas para los casos o grupo de casos de uso que fueron establecidos en **Tareas y su duración estimada.**

**Equipo de desarrollo (Requerimientos de comunicación y transferencia de información 1 y 2)**

**Casos de uso #04 y #08**

**Nombre CU04: Crear región (tabla parámetro)**

**Nombre CU08: Crear ciudad (tabla parámetro)**

Tabla 28. Calendario de las tareas 1.1 - 3.2 (CU04 y CU08) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Plantilla de bootstrap. 2. Realizar el prototipado (diseño de la interfaz).  * Documentar. | **Actividad 1.1:**  16/06/2019  **Actividad 1.2:**  16/06/2019 | **Actividad 1.1:**  22/06/2019  **Actividad 1.2: 6**  22/06/2019 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | **Actividad 2.1:**  22/06/2019  **Actividad 2.2:**  22/06/2019 | **Actividad 2.1:**  25/06/2019  **Actividad 2.2: 3**  25/06/2019 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | **Actividad 3.1:**  25/06/2019  **Actividad 3.2:**  29/06/2019 | **Actividad 3.1: 4**  29/06/2019  **Actividad 3.2: 3**  02/07/2019 |

*Nota:* Elaboración propia.

**Casos de uso #16 y #10**

**Nombre CU16: Crear mantenimiento (tabla parámetro)**

**Nombre CU10: Crear alerta (tabla parámetro)**

Tabla 29. Calendario de las tareas 1.1 - 3.2 (CU16 y CU10) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | **Actividad 1.1:**  02/07/2019 | **Actividad 1.1: 6**  08/07/2019 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | **Actividad 2.1:**  08/07/2019  **Actividad 2.2:**  08/07/2019 | **Actividad 2.1:**  11/07/2019  **Actividad 2.2: 3**  11/07/2019 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | **Actividad 3.1:**  11/07/2019  **Actividad 3.2:**  15/07/2019 | **Actividad 3.1: 4**  15/07/2019  **Actividad 3.2: 3**  18/07/2019 |

*Nota:* Elaboración propia.

**Casos de uso #06 y #07**

**Nombre CU06: Crear organización (tabla parámetro)**

**Nombre CU07: Crear categoría (tabla parámetro)**

Tabla 30. Calendario de las tareas 1.1 - 3.2 (CU06 y CU07) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | **Actividad 1.1:**  18/07/2019 | **Actividad 1.1: 6**  24/07/2019 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | **Actividad 2.1:**  24/07/2019  **Actividad 2.2:**  24/07/2019 | **Actividad 2.1:**  27/07/2019  **Actividad 2.2: 3**  27/07/2019 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | **Actividad 3.1:**  27/07/2019  **Actividad 3.2:**  31/07/2019 | **Actividad 3.1: 4**  31/07/2019  **Actividad 3.2: 3**  03/08/2019 |

*Nota:* Elaboración propia.

**Caso de uso #05**

**Nombre CU05: Crear estación (tabla parámetro)**

Tabla 31. Calendario de las tareas 1.1 - 3.2 (CU05) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | **Actividad 1.1:**  03/08/2019 | **Actividad 1.1: 7**  10/08/2019 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | **Actividad 2.1:**  10/08/2019  **Actividad 2.2:**  13/08/2019 | **Actividad 2.1: 3**  13/08/2019  **Actividad 2.2: 3**  16/08/2019 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | **Actividad 3.1:**  16/08/2019  **Actividad 3.2:**  31/08/2019 | **Actividad 3.1: 15**  31/08/2019  **Actividad 3.2: 3**  03/09/2019 |

*Nota:* Elaboración propia.

**Caso de uso #01**

**Nombre CU01: Conectar con estación FTP**

Tabla 32. Calendario de las tareas 1.1 - 3.2 (CU01) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | **Actividad 1.1:**  03/09/2019 | **Actividad 1.1: 7**  10/09/2019 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | **Actividad 2.1:**  10/09/2019  **Actividad 2.2:**  13/09/2019 | **Actividad 2.1: 3**  13/09/2019  **Actividad 2.2: 3**  16/09/2019 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | **Actividad 3.1:**  16/09/2019  **Actividad 3.2:**  01/10/2019 | **Actividad 3.1: 15**  01/10/2019  **Actividad 3.2: 3**  04/10/2019 |

*Nota:* Elaboración propia.

**Casos de uso #09 y #11**

**Nombre CU09: Crear rango (tabla parámetro)**

**Nombre CU11: Crear base de tiempo (tabla parámetro)**

Tabla 33. Calendario de las tareas 1.1 - 3.2 (CU09 y CU11) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Análisis y requerimientos**  **Obj:** Revisión de los requerimientos.  **Actividades:**   1. Realizar el prototipado (diseño de la interfaz).  * Documentar. | **Actividad 1.1:**  04/10/2019 | **Actividad 1.1: 6**  10/10/2019 |
| 2 | **Fase 2: Análisis y diseño preliminar**  **Obj:** Revisión del diseño preliminar.  **Actividades:**   1. Diagrama de clases. 2. Diagrama de secuencia.  * Documentar. | **Actividad 2.1:**  10/10/2019  **Actividad 2.2:**  10/10/2019 | **Actividad 2.1:**  13/10/2019  **Actividad 2.2: 3**  13/10/2019 |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Escribir/generar código. 2. Diccionario de funciones o módulos.  * Documentar. | **Actividad 3.1:**  13/10/2019  **Actividad 3.2:**  17/10/2019 | **Actividad 3.1: 4**  17/10/2019  **Actividad 3.2: 3**  20/10/2019 |

*Nota:* Elaboración propia.

**Fase 3: actividades 3.3 y 3.4 (de los mismos casos de uso anteriores)**

**Casos de uso #04 y #08**

**Nombre CU04: Crear región (tabla parámetro)**

**Nombre CU08: Crear ciudad (tabla parámetro)**

Tabla 34. Calendario de las tareas 3.3 y 3.4 (CU04 y CU08) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | **Actividad 3.3:**  20/10/2019  **Actividad 3.4:**  23/10/2019 | **Actividad 3.3: 3**  23/10/2019  **Actividad 3.4: 3**  26/10/2019 |

*Nota:* Elaboración propia.

**Casos de uso #16 y #10**

**Nombre CU16: Crear mantenimiento (tabla parámetro)**

**Nombre CU10: Crear alerta (tabla parámetro)**

Tabla 35. Calendario de las tareas 3.3 y 3.4 (CU16 y CU10) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | **Actividad 3.3:**  26/10/2019  **Actividad 3.4:**  29/10/2019 | **Actividad 3.3: 3**  29/10/2019  **Actividad 3.4: 3**  01/11/2019 |

*Nota:* Elaboración propia.

**Casos de uso #06 y #07**

**Nombre CU06: Crear organización (tabla parámetro)**

**Nombre CU07: Crear categoría (tabla parámetro)**

Tabla 36. Calendario de las tareas 3.3 y 3.4 (CU06 y CU07) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | **Actividad 3.3:**  01/11/2019  **Actividad 3.4:**  04/11/2019 | **Actividad 3.3: 3**  04/11/2019  **Actividad 3.4: 3**  07/11/2019 |

*Nota:* Elaboración propia.

**Caso de uso #05**

**Nombre CU05: Crear estación (tabla parámetro)**

Tabla 37. Calendario de las tareas 3.3 y 3.4 (CU05) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | **Actividad 3.3:**  07/11/2019  **Actividad 3.4:**  12/11/2019 | **Actividad 3.3: 5**  12/11/2019  **Actividad 3.4: 4**  16/11/2019 |

*Nota:* Elaboración propia.

**Caso de uso #01**

**Nombre CU01: Conectar con estación FTP**

Tabla 38. Calendario de las tareas 3.3 y 3.4 (CU01) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | **Actividad 3.3:**  16/11/2019  **Actividad 3.4:**  21/11/2019 | **Actividad 3.3: 5**  21/11/2019  **Actividad 3.4: 4**  25/11/2019 |

*Nota:* Elaboración propia.

**Casos de uso #09 y #11**

**Nombre CU09: Crear rango (tabla parámetro)**

**Nombre CU11: Crear base de tiempo (tabla parámetro)**

Tabla 39. Calendario de las tareas 3.3 y 3.4 (CU09 y CU11) del equipo de desarrollo.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Implementación**  **Obj:** Codificación y creación del software.  **Actividades:**   1. Refinamiento de acuerdo con las pruebas unitarias. 2. Realizar pruebas de aceptación del usuario.  * Documentar. | **Actividad 3.3:**  25/11/2019  **Actividad 3.4:**  28/11/2019 | **Actividad 3.3: 3**  28/11/2019  **Actividad 3.4: 3**  01/12/2019 |

*Nota:* Elaboración propia.

**Equipo de calidad (Requerimientos de comunicación y transferencia de información 1 y 2)**

**Casos de uso #04 y #08**

**Nombre CU04: Crear región (tabla parámetro)**

**Nombre CU08: Crear ciudad (tabla parámetro)**

Tabla 40. Calendario de las tareas 1.1 - 2.4 (CU04 y CU08) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Planeación**  **Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de caso de uso. 2. Validación de diagrama de casos de uso. 3. Tipos de pruebas. | **Actividad 1.1**  18/06/2019  **Actividad 2:**  20/06/2019  **Actividad 1.3:**  22/06/2019 | **Actividad 1.1:2**  20/06/2019  **Actividad 1.2:2**  22/06/2019  **Actividad 1.3:2**  24/06/2019 |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio. 2. Plan de pruebas. 3. Especificación del diseño de pruebas. 4. Especificación de casos de prueba. | **Actividad 2.1:**  24/06/2019  **Actividad 2.2:**  26/06/2019  **Actividad 2.3:**  28/06/2019  **Actividad 2.4:** 30/06/2019 | **Actividad 2.1:2**  26/06/2019  **Actividad 2.2:2** 28/06/2019  **Actividad 2.3:2**  30/06/2019  **Actividad 2.4:2**  02/07/2019 |

*Nota*: Elaboración propia.

**Casos de uso #16 y #10**

**Nombre CU16: Crear mantenimiento (tabla parámetro)**

**Nombre CU10: Crear alerta (tabla parámetro)**

Tabla 41. Calendario de las tareas 1.1 - 2.4 (CU16 y CU10) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Planeación**  **Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de caso de uso. 2. Validación de diagrama de casos de uso. 3. Tipos de pruebas. | **Actividad 1.2:**  02/07/2019  **Actividad 1.2:**  04/07/2019  **Actividad 1.3:**  06/07/2019 | **Actividad 1.2:2**  04/07/2019  **Actividad 1.2:2**  06/07/2019  **Actividad 1.3:2**  08/07/2019 |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio. 2. Plan de pruebas. 3. Especificación del diseño de pruebas. 4. Especificación de casos de prueba. | **Actividad 2.1:**  08/07/2019  **Actividad 2.2:**  10/07/2019  **Actividad 2.3:**  12/07/2019  **Actividad 2.4:** 14/07/2019 | **Actividad 2.1:2**  10/07/2019  **Actividad 2.2:2** 12/08/2019  **Actividad 2.3:2**  14/07/2019  **Actividad 2.4:2**  16/07/2019 |

*Nota*: Elaboración propia

**Casos de uso #06 y #07**

**Nombre CU06: Crear organización (tabla parámetro)**

**Nombre CU07: Crear categoría (tabla parámetro)**

Tabla 42. Calendario de las tareas 1.1 - 2.4 (CU06 y CU07) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Planeación**  **Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de caso de uso. 2. Validación de diagrama de casos de uso. 3. Tipos de pruebas. | **Actividad 1.1:**  16/07/2019  **Actividad 1.2:**  18/07/2019  **Actividad 1.3:**  20/07/2019 | **Actividad 1.1:2**  18/07/2019  **Actividad 1.2:2**  20/07/2019  **Actividad 1.3:2**  22/07/2019 |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio. 2. Plan de pruebas. 3. Especificación del diseño de pruebas. 4. Especificación de casos de prueba. | **Actividad 2.1:**  22/07/2019  **Actividad 2.2:**  24/07/2019  **Actividad 2.3:**  26/07/2019  **Actividad 2.4:** 28/07/2019 | **Actividad 2.1:2**  24/07/2019  **Actividad 2.2:2** 26/07/2019  **Actividad 2.3:2**  28/07/2019  **Actividad 2.4:2**  30/07/2019 |

*Nota*: Elaboración propia.

**Caso de uso #05**

**Nombre CU05: Crear estación (tabla parámetro)**

Tabla 43. Calendario de las tareas 1.1 - 2.4 (CU05) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Planeación**  **Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de caso de uso. 2. Validación de diagrama de casos de uso. 3. Tipos de pruebas. | **Actividad 1.1:**  30/07/2019  **Actividad 1.2:**  02/08/2019  **Actividad 1.3:**  04/08/2019 | **Actividad 1.1:2**  02/08/2019  **Actividad 1.2:2**  04/08/2019  **Actividad 1.3:2**  06/08/2019 |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio 2. Plan de pruebas. 3. Especificación del diseño de pruebas. 4. Especificación de casos de prueba. | **Actividad 2.1:**  06/08/2019  **Actividad 2.2:**  10/08/2019  **Actividad 2.3:**  12/08/2019  **Actividad 2.4:** 14/08/2019 | **Actividad 2.1:2**  08/08/2019  **Actividad 2.2:2** 12/08/2019  **Actividad 2.3:2**  14/08/2019  **Actividad 2.4:2**  16/08/2019 |

Nota: Elaboración propia.

**Caso de uso #01**

**Nombre CU01: Conectar con estación FTP**

Tabla 44. Calendario de las tareas 1.1 - 2.4 (CU01) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Planeación**  **Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de caso de uso. 2. Validación de diagrama de casos de uso. 3. Tipos de pruebas. | **Actividad 1.1:**  16/08/2019  **Actividad 1.2:**  18/08/2019  **Actividad 1.3:**  20/08/2019 | **Actividad 1.1:1**  18/08/2019  **Actividad 1.2:1**  20/08/2019  **Actividad 1.3:1**  22/08/2019 |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio. 2. Plan de pruebas. 3. Especificación del diseño de pruebas. 4. Especificación de casos de prueba. | **Actividad 2.1:**  22/08/2019  **Actividad 2.2:**  24/08/2019  **Actividad 2.3:**  26/08/2019  **Actividad 2.4:** 28/08/2019 | **Actividad 2.1:1**  24/08/2019  **Actividad 2.2:1** 26/08/2019  **Actividad 2.3:1**  28/08/2019  **Actividad 2.4:1**  30/08/2019 |

*Nota*: Elaboración propia.

**Casos de uso #09 y #11**

**Nombre CU09: Crear rango (tabla parámetro)**

**Nombre CU11: Crear base de tiempo (tabla parámetro)**

Tabla 45. Calendario de las tareas 1.1 - 2.4 (CU09 y CU11) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 1 | **Fase 1: Planeación**  **Obj:** Se define el alcance del proyecto a desarrollar al igual que los roles del equipo de trabajo.  **Actividades:**   1. Diagrama de caso de uso. 2. Validación de diagrama de casos de uso. 3. Tipos de pruebas. | **Actividad 1.1:**  30/08/2019  **Actividad 1.2:**  01/09/2019  **Actividad 1.3:**  04/09/2019 | **Actividad 1.1:2**  01/09/2019  **Actividad 1.2:3**  04/09/2019  **Actividad 1.3:2**  07/09/2019 |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Modelado de dominio. 2. Plan de pruebas. 3. Especificación del diseño de pruebas. 4. Especificación de casos de prueba. | **Actividad 2.1:**  07/09/2019  **Actividad 2.2:**  10/09/2019  **Actividad 2.3:**  16/09/2019  **Actividad 2.4:** 19/09/2019 | **Actividad 2.1:3**  10/09/2019  **Actividad 2.2:3** 16/09/2019  **Actividad 2.3:3**  19/09/2019  **Actividad 2.4:3**  25/09/2019 |

*Nota*: Elaboración propia.

**Fase 2: actividades 2.5 y 2.6 (de los mismos casos de uso anteriores)**

**Casos de uso #04 y #08**

**Nombre CU04: Crear región (tabla parámetro)**

**Nombre CU08: Crear ciudad (tabla parámetro)**

Tabla 46. Calendario de las tareas 2.5 y 2.6 (CU04 y CU08) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Procedimiento de pruebas. 2. Reporte de trasmisión de pruebas. | **Actividad 2.5:**  25/09/2019  **Actividad 2.6:**  27/09/2019 | **Actividad 2.5:**  27/09/2019  **Actividad 2.6:**  29/09/2019 |

Nota: Elaboración propia.

**Casos de uso #16 y #10**

**Nombre CU16: Crear mantenimiento (tabla parámetro)**

**Nombre CU10: Crear alerta (tabla parámetro)**

Tabla 47. Calendario de las tareas 2.5 y 2.6 (CU16 y CU10) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Procedimiento de pruebas. 2. Reporte de trasmisión de pruebas. | **Actividad 2.5:**  29/09/2019  **Actividad 2.6:**  01/10/2019 | **Actividad 2.5:**  01/10/2019  **Actividad 2.6:**  04/10/2019 |

*Nota*: Elaboración propia.

**Casos de uso #06 y #07**

**Nombre CU06: Crear organización (tabla parámetro)**

**Nombre CU07: Crear categoría (tabla parámetro)**

Tabla 48. Calendario de las tareas 2.5 y 2.6 (CU06 y CU07) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Procedimiento de pruebas. 2. Reporte de trasmisión de pruebas. | **Actividad 2.5:**  04/10/2019  **Actividad 2.6:**  07/10/2019 | **Actividad 2.5:**  07/10/2019  **Actividad 2.6:**  10/10/2019 |

*Nota*: Elaboración propia**.**

**Caso de uso #05**

**Nombre CU05: Crear estación (tabla parámetro)**

Tabla 49. Calendario de las tareas 2.5 y 2.6 (CU05) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Procedimiento de pruebas. 2. Reporte de trasmisión de pruebas. | **Actividad 2.5:**  10/10/2019  **Actividad 2.6:**  13/10/2019 | **Actividad 2.5:**  13/10/2019  **Actividad 2.6:**  16/10/2019 |

*Nota*: Elaboración propia.

**Caso de uso #01**

**Nombre CU01: Conectar con estación FTP**

Tabla 50. Calendario de las tareas 2.5 y 2.6 (CU01) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Procedimiento de pruebas. 2. Reporte de trasmisión de pruebas. | **Actividad 2.5:**  16/10/2019  **Actividad 2.6:**  19/10/2019 | **Actividad 2.5:1**  19/10/2019  **Actividad 2.6:1**  22/10/2019 |

*Nota*: Elaboración propia.

**Casos de uso #09 y #11**

**Nombre CU09: Crear rango (tabla parámetro)**

**Nombre CU11: Crear base de tiempo (tabla parámetro)**

Tabla 51. Calendario de las tareas 2.5 y 2.6 (CU09 y CU11) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 2 | **Fase 2: Preparación de pruebas según las características de la ISO/IEC 25010 (adecuación funcional, eficiencia de desempeño, usabilidad, seguridad y mantenibilidad)**  **Obj:** Se busca elaborar las pruebas a las cuales se va a someter el software.  **Actividades:**   1. Procedimiento de pruebas. 2. Reporte de trasmisión de pruebas. | **Actividad 2.5:**  22/10/2019  **Actividad 2.6:**  25/10/2019 | **Actividad 2.5:1**  25/10/2019  **Actividad 2.6:1**  28/10/2019 |

*Nota*: Elaboración propia.

**Fase 3 y 4: completas (de los mismos casos de uso anteriores)**

**Casos de uso #04 y #08**

**Nombre CU04: Crear región (tabla parámetro)**

**Nombre CU08: Crear ciudad (tabla parámetro)**

Tabla 52. Calendario de las tareas 3.1 - 4.1 (CU04 y CU08) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Ejecución de pruebas**  **Obj**: Se busca con los datos de la fase 2 ejecutar las pruebas.  **Actividades:**   1. Logs de pruebas. 2. Realizar pruebas unitarias. 3. Reporte de incidencias. | **Actividad 3.1:**  28/10/2019  **Actividad 3.2:**  01/11/2019  **Actividad 3.3:**  03/11/2019 | **Actividad 3.1: 2**  01/11/2019  **Actividad 3.2:2**  03/11/2019  **Actividad 3.3:2**  05/11/2019 |
| 4 | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | **Actividad 4.1:**  05/11/2019 | **Actividad 4.1: 2**  07/11/2019 |

*Nota*: Elaboración propia.

**Casos de uso #16 y #10**

**Nombre CU16: Crear mantenimiento (tabla parámetro)**

**Nombre CU10: Crear alerta (tabla parámetro)**

Tabla 53. Calendario de las tareas 3.1 - 4.1 (CU16 y CU10) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Ejecución de pruebas**  **Obj**: Se busca con los datos de la fase 2 ejecutar las pruebas.  **Actividades:**   1. Logs de pruebas. 2. Realizar pruebas unitarias. 3. Reporte de incidencias. | **Actividad 3.1:**  07/11/2019  **Actividad 3.2:**  08/11/2019  **Actividad 3.3:**  09/11/2019 | **Actividad 3.1: 1**  08/11/2019  **Actividad 3.2:1**  9/11/2019  **Actividad 3.3:1**  10/11/2019 |
| 4 | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | **Actividad 4.1:**  10/11/2019 | **Actividad 4.1: 2**  11/11/2019 |

*Nota*: Elaboración propia.

**Casos de uso #06 y #07**

**Nombre CU06: Crear organización (tabla parámetro)**

**Nombre CU07: Crear categoría (tabla parámetro)**

Tabla 54. Calendario de las tareas 3.1 - 4.1 (CU06 y CU07) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Ejecución de pruebas**  **Obj**: Se busca con los datos de la fase 2 ejecutar las pruebas.  **Actividades:**   1. Logs de pruebas. 2. Realizar pruebas unitarias. 3. Reporte de incidencias. | **Actividad 3.1:**  11/11/2019  **Actividad 3.2:**  13/11/2019  **Actividad 3.3:**  15/11/2019 | **Actividad 3.1: 2**  13/11/2019  **Actividad 3.2:2**  14/11/2019  **Actividad 3.3:2**  16/11/2019 |
| 4 | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | **Actividad 4.1:**  16/11/2019 | **Actividad 4.1: 4**  17/11/2019 |

*Nota*: Elaboración propia.

**Caso de uso #05**

**Nombre CU05: Crear estación (tabla parámetro)**

Tabla 55. Calendario de las tareas 3.1 - 4.1 (CU05) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Ejecución de pruebas**  **Obj**: Se busca con los datos de la fase 2 ejecutar las pruebas.  **Actividades:**   1. Logs de pruebas. 2. Realizar pruebas unitarias. 3. Reporte de incidencias. | **Actividad 3.1:**  17/11/2019  **Actividad 3.2:**  18/11/2019  **Actividad 3.3:**  19/11/2019 | **Actividad 3.1: 2**  18/11/2019  **Actividad 3.2:2**  19/11/2019  **Actividad 3.3:2**  20/11/2019 |
| 4 | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | **Actividad 4.1:**  20/11/2019 | **Actividad 4.1: 3**  21/11/2019 |

*Nota*: Elaboración propia.

**Caso de uso #01**

**Nombre CU01: Conectar con estación FTP**

Tabla 56. Calendario de las tareas 3.1 - 4.1 (CU01) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Ejecución de pruebas**  **Obj**: Se busca con los datos de la fase 2 ejecutar las pruebas.  **Actividades:**   1. Logs de pruebas. 2. Realizar pruebas unitarias. 3. Reporte de incidencias. | **Actividad 3.1:**  21/11/2019  **Actividad 3.2:**  22/11/2019  **Actividad 3.3:**  24/11/2019 | **Actividad 3.1: 2**  22/11/2019  **Actividad 3.2:2**  23/11/2019  **Actividad 3.3:2**  25/11/2019 |
| 4 | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | **Actividad 4.1:**  25/11/2019 | **Actividad 4.1: 4**  26/11/2019 |

*Nota*: Elaboración propia.

**Casos de uso #09 y #11**

**Nombre CU09: Crear rango (tabla parámetro)**

**Nombre CU11: Crear base de tiempo (tabla parámetro)**

Tabla 57. Calendario de las tareas 3.1 - 4.1 (CU09 y CU11) del equipo de calidad.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Descripción** | **Fecha inicio** | **Fecha fin** |
| 3 | **Fase 3: Ejecución de pruebas**  **Obj**: Se busca con los datos de la fase 2 ejecutar las pruebas.  **Actividades:**   1. Logs de pruebas. 2. Realizar pruebas unitarias. 3. Reporte de incidencias. | **Actividad 3.1:**  26/11/2019  **Actividad 3.2:**  27/11/2019  **Actividad 3.3:**  29/11/2019 | **Actividad 3.1:**  27/11/2019  **Actividad 3.2:**  29/11/2019  **Actividad 3.3:2**  01/12/2019 |
| 4 | **Fase 4: Finalización de pruebas**  **Obj:** Documentar los resultados  **Actividades**   1. Reporte final de pruebas. | **Actividad 4.1:**  01/12/2019 | **Actividad 4.1: 4**  05/12/2019 |

*Nota*: Elaboración propia.

**Esfuerzo y costo estimado**

**- Esfuerzo estimado**

Para este proyecto se requiere un esfuerzo del 100%, que está repartido entre los distintos equipos de trabajo que lo conforman, de la siguiente manera:

Tabla 58. Esfuerzo estimado.

|  |  |
| --- | --- |
| **Equipos de trabajo** | **Esfuerzo (%)** |
| Equipo de gerencia del proyecto | 25 |
| Equipo de desarrollo | 30 |
| Equipo de calidad | 45 |

*Nota:* Elaboración propia.

**- Costo estimado**

Tabla 59. Costo estimado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Equipos** | **Dedicación (horas/semana)** | **Semanas (junio - diciembre)** | **Valor hora ($)** | **Total** |
| Equipo de gerencia del proyecto | 12 | 28 | 15000 | 5040000 |
| Equipo de desarrollo | 16 | 28 | 15000 | 6720000 |
| Equipo de calidad | 20 | 28 | 15000 | 8400000 |
| **Total** |  |  |  | **20160000** |

*Nota:* Elaboración propia.

**Estrategia de control de versiones**

Para la creación del repositorio del proyecto se hará uso de la plataforma de desarrollo colaborativo GitHub y se utilizará para el control de versiones del proyecto la aplicación de escritorio de GitHub: GitHub Desktop, Interfaz Gráfica de Usuario (GUI) diseñada para facilitar el uso de Git. Adicional se almacenarán semanalmente copias de respaldo de las versiones del proyecto en el servicio de la nube de Google Drive. De esta manera si se desea volver a un estado anterior del proyecto, se tiene tanto la herramienta y su sistema de control de versiones como las copias de seguridad almacenadas en Google Drive para hacerlo.

El mecanismo de almacenamiento que se utilizará para el proyecto será el gestor de base de datos relacional PostgreSQL.

**Instrucciones de entrega**

* Se pacta la entrega completa de documentación y productos por caso o grupo de casos de uso (definidos anteriormente) a cargo de cada equipo, en el momento en que todas las actividades con las que cuentan estos casos de uso, estén culminadas en su totalidad.
* Se hará la entrega del repositorio del proyecto a los nuevos miembros del equipo de trabajo que continuarán con el desarrollo de los demás requisitos planteados por el cliente.
* Se diligenciará un registro de aceptación por equipo que listará toda la documentación y productos que se le van a proporcionar al cliente en el momento en que el proyecto esté finalizado en su totalidad.

**Nota:** El proyecto será aceptado siempre y cuando se encuentre la firma del cliente en estos registros de aceptación, o en un caso particular, las firmas de los gerentes del proyecto.

### **Crear repositorio del proyecto**

Para la creación del repositorio del proyecto se hará uso de la plataforma de desarrollo colaborativo GitHub y se utilizará para el control de versiones del proyecto, la aplicación de escritorio de GitHub: GitHub Desktop. A continuación, se muestra una captura de imagen de referencia de la aplicación de escritorio de GitHub (ver figura 7):

Figura 7. Captura de imagen de referencia de la aplicación de escritorio de GitHub: GitHub Desktop.



*Nota:* Elaboración propia.

Cada equipo de trabajo tenía bajo su responsabilidad, la entrega de unas tareas definidas. Para la entrega ordenada de estas tareas por parte de los equipos de trabajo, se hizo la creación de ramas (branches) específicas para cada uno con el objetivo de que, al momento de subirlas, se lograra tener un seguimiento y control de estas. A continuación, se muestra cómo es visto el repositorio desde la aplicación web de GitHub (ver figura 8):

Figura 8. Repositorio del proyecto, visto desde la aplicación web de GitHub.



*Nota:* Elaboración propia.

El uso del repositorio como estrategia para la entrega ordenada de las actividades fue de crucial importancia, puesto que facilitó el seguimiento continuo del trabajo realizado por cada equipo, a través de la revisión de fecha, hora y descripción que cada cambio proporcionaba en el momento en que era producido.

### **Registrar los resultados de la verificación**

Antes de que se ejecutara el Plan del proyecto (Fase 2), se diligenciaba este artefacto, con el fin de verificar que los elementos del Plan eran viables o no completamente, y se llevaban a cabo (si las había) las correcciones correspondientes hasta que el documento fuera aprobado por todo el equipo, y de esta manera fuera ejecutado sin ningún problema o contratiempo. A continuación, se muestra un ejemplo de una verificación realizada al Plan del proyecto (ver figura 9):

Figura 9. Ejemplo de una plantilla de resultados de la verificación del plan de proyecto.

****

*Nota:* Elaboración propia con base en lo propuesto por la ISO/IEC 29110.

Este formato no tuvo como tal un impacto alto en el desarrollo del proyecto, pero se utilizó como una opción de registro de los cambios necesarios que se debían realizar al plan del proyecto antes de ejecutarse. Cabe mencionar, que los cambios a realizar sobre el documento mencionado se acordaban junto con todo el equipo de trabajo.

## Fase 2: Ejecución del plan del proyecto

En esta fase se llevó a cabo la ejecución del plan sobre el proyecto de la empresa Sanambiente, y el uso del Repositorio del proyecto definido en la fase 1.

### **Diligenciar un acta de reunión**

Plantilla que fue diligenciada los días sábados entre el mes de agosto e inicios del mes de diciembre en cada reunión lograda con el equipo de trabajo de forma presencial o virtual. A continuación, un ejemplo (ver figura 10):

Figura 10. Ejemplo de un acta de reunión lograda con el equipo de trabajo.



*Nota:* Elaboración propia con base en lo propuesto por la ISO/IEC 29110.

El acta de reunión se consideró crucial, ya que por medio del diligenciamiento de esta se certificaba la realización de las reuniones, lo que se logró, lo que quedó pendiente y lo que se acordaba en ellas al finalizarlas.

### **Diligenciar el registro de estado de progreso**

Para la realización de esta plantilla se tomó como base el calendario de tareas definido en el Plan del proyecto. A continuación, un ejemplo de un registro de estado que se realizó sobre los casos de uso 03 y 07 pertenecientes al equipo de desarrollo (ver figura 11):

Figura 11. Ejemplo de un registro de estado de progreso sobre los casos de uso 04 y 08 pertenecientes al equipo de desarrollo.



*Nota:* Elaboración propia con base en lo propuesto por la ISO/IEC 29110.

Este artefacto fue fundamental, ya que permitió tener control y seguimiento sobre los equipos de trabajo en cuanto a las entregas de actividades que tenían asignadas. Verificar aquellas actividades pendientes por entregar, confirmar aquellas que fueron completadas y notificar sobre aquellas que estaban próximas a entregar dentro de pocos días. Este artefacto se puede decir que fue uno de los principales pilares tanto para guiar como para realizar seguimiento a los equipos de trabajo en la realización de sus actividades.

### **Diligenciar una solicitud de cambio**

Plantilla que fue diligenciada por cada equipo de trabajo (incluido el equipo de Gestión) cuando era necesario.

Cuando una solicitud era iniciada, el equipo de gestión era informado para realizar su seguimiento; y si eran notificados de su solución, se procedía a corroborar su veracidad y llevar a cabo el cambio respectivo del estado de iniciada o evaluada al estado de resuelta en la plantilla y se notificaba a los equipos implicados acerca de esta actualización. A continuación, se muestra un ejemplo de una solicitud de cambio en estado iniciada (ver figura 12):

Figura 12. Ejemplo de una solicitud de cambio iniciada.



*Nota:* Elaboración propia con base en lo propuesto por la ISO/IEC 29110.

Las solicitudes de cambio fueron fundamentales, dado que a través de ellas se notificaba a los equipos implicados sobre los cambios que necesitaban que ejecutaran en algunas de sus actividades como consecuencia de que dependían de ellas para poder avanzar en las suyas. Estas solicitudes tuvieron seguimiento y control por parte del equipo de gestión para constatar que fueran resueltas en el plazo acordado por los equipos que las tenían a cargo.

## Fase 3: Evaluación y control del proyecto

En esta fase se llevó a cabo la evaluación de desempeño y progreso del proyecto, con respecto a lo documentado y pactado en el Plan del proyecto, dando uso para la comparación el Registro de estado de progreso. Adicional, se siguieron realizando solicitudes de cambio para abordar desviaciones importantes que surgieron y se realizó seguimiento de estas hasta que fueron resueltas. Se produjeron los siguientes artefactos:

### **Diligenciar el registro de corrección**

Plantilla que fue encargada para ser diligenciada por cada equipo de trabajo.

Cuando una corrección era iniciada, el equipo de gestión era informado para realizar su seguimiento; y si eran notificados de su realización, se procedía a corroborar su veracidad y llevar a cabo el cambio respectivo del estado de pendiente al estado de realizada en la plantilla y se notificaba a los equipos implicados acerca de esta actualización. A continuación, se muestra un ejemplo de un registro de corrección en estado pendiente (ver figura 13):

Figura 13. Ejemplo de un registro de corrección en estado pendiente.



*Nota:* Elaboración propia con base en lo propuesto por la ISO/IEC 29110.

Este formato fue uno de los más importantes y usados durante el desarrollo del proyecto, ya que permitió evidenciar mediante estos, los problemas que se les presentaba a cada equipo de trabajo durante la ejecución de sus actividades a entregar. Un ejemplo:

* Atrasos en las entregas de las actividades por falta de compromiso a la hora de cumplir con las fechas estipuladas en el Plan del proyecto y el Registro de estado de progreso.

## Fase 4: Cierre del proyecto

En esta fase se llevó a cabo la entrega de documentación y productos al cliente, tal y como se detalló en las Instrucciones de entrega del Plan del proyecto. Se produjeron los siguientes artefactos:

### **Diligenciar un registro de aceptación.**

Esta plantilla se diligenció por cada equipo (calidad y desarrollo), cuando se hizo la entrega completa al cliente de la documentación y productos a cargo de cada uno.

El registro de aceptación fue el artefacto que se utilizó como medio escrito para registrar el cierre del proyecto. Se diligenció un formato de estos por cada equipo para evidenciar los elementos o productos que cada uno entregaba al cliente, con el propósito de dar finalización y cierre a su parte del proyecto.

**Nota:** El propósito era que el proyecto fuera aceptado por parte del cliente, pero dada las circunstancias actuales que se presentan a nivel nacional causadas por la emergencia sanitaria, la aceptación se dará en este caso por el equipo de gerencia.

A continuación, se muestra un ejemplo de un registro de aceptación diligenciado y firmado por los gerentes (ver figura 14):

Figura 14. Ejemplo de un registro de aceptación diligenciado y firmado.



*Nota:* Elaboración propia con base en lo propuesto por la ISO/IEC 29110.

# Conclusiones

Luego de haber realizado el análisis de los lineamientos proporcionados por la ISO/IEC 29110, se concluye que su aplicación desde de la parte 5 del estándar en el presente trabajo, logró cumplir con las necesidades y características que Sanambiente planteaba en su proyecto de software. Dado los recursos que intervinieron en el desarrollo de este último, el incluir estrategias de gestión por medio de esta norma fue lo adecuado para garantizar su éxito.

Seguir el perfil básico dentro del grupo de perfiles genéricos en este trabajo fue conveniente, puesto que la documentación que proporcionaba el estándar para este tipo de perfiles, se acomodaba a este caso adecuadamente dado el tiempo, número de actividades, productos e integrantes exigidos para su desarrollo. Además, el no emplear el paquete de despliegue que brindaba el estándar fue acertado, dado que seguir este conjunto de elementos como medio para el control y seguimiento de los recursos que iban a interceder en el proyecto de la empresa Sanambiente no iba a ser la mejor opción como guía para su desarrollo. En su lugar, el seguir únicamente los pasos descritos en los procesos contenidos en el estándar permitieron la efectiva ejecución y cierre del proyecto con éxito.

De acuerdo a lo propuesto en el primer objetivo específico y mediante el estudio realizado, se identificaron dos procesos en la parte 5 de la ISO/IEC 29110: proceso de gestión de proyecto y proceso de implementación de software. En este trabajo se ejecutó únicamente el proceso de gestión de proyectos, junto con sus fases y artefactos correspondientes; pero no el proceso de implementación de software, puesto que la documentación que exigía para ser aplicado era demasiada y no se ajustaba al contexto de este caso para ser adoptada. En su lugar, se efectuó la implementación de las fases de la metodología Iconix dando buenos resultados, puesto que los elementos que brindaban permitieron un adecuado desarrollo del software en conformidad con las características y necesidades propias que la empresa Sanambiente planteó en su proyecto. Además, la incorporación de las fases de la IEEE 829 para guiar la ejecución de las pruebas de calidad que se debían realizar en el proceso de implementación del software, lograron encaminar procedimientos bien estructurados en la calidad final del producto realizado (Marín Ospina, 2020, p. 12).

Con respecto a lo propuesto en el segundo objetivo específico, la ruta de procesos y herramientas establecida para este proyecto, se muestra en la figura 6 ubicada en el capítulo de Resultados. En esta figura, se presentan los procesos y actividades que fueron ejecutados durante el desarrollo del proyecto por parte de todos los equipos de trabajo.

El seguimiento de esta ruta en el proyecto, permitió realizar un trabajo efectivo en conjunto con los equipos, gracias a que lograba integrar todos los procesos y actividades que intervenían en el desarrollo. Cabe mencionar que, al inicio del proyecto seguir esta ruta tomó tiempo, puesto que los equipos carecían de orden en la realización y entrega de las actividades a su cargo en una fecha estipulada; pero, con el transcurrir del tiempo, los equipos lograron asumir la responsabilidad de comprender y seguir la ruta, logrando así mejorar la productividad y entrega de las actividades propuestas.

De acuerdo a lo propuesto en el tercer objetivo específico, se encontró que la vinculación del estándar ISO/IEC 29110 con la metodología Iconix tuvo un poco de complejidad, dado que esta metodología exigía cumplir con unas fases y actividades puntuales con numerosa documentación de por medio; por lo tanto, se tuvo que realizar al inicio una adecuación de estas exigencias para que se ajustaran y pudieran cumplir perfectamente con lo requerido en este proyecto.

Además, se identificaron dificultades de los equipos al momento de diligenciar algunos de los formatos o artefactos proporcionados por el equipo de gestión, puesto que presentaban confusiones y dudas con respecto a cuál se debía diligenciar y en qué momento; a pesar de que el equipo de gestión en distintas ocasiones les aclaraba las dudas sobre cómo debían emplearse.

Otro aspecto que se halló en la aplicación del marco de trabajo fue la creación de los formatos o artefactos de cada fase que el estándar ISO/IEC 29110 proporcionaba para el proceso de gestión de proyectos. Los formatos expuestos por el estándar poseían un exceso de documentación, por lo que se decidió no tomarlos todos, y emplear solamente los necesarios para desarrollar este trabajo. Además de elegir los formatos adecuados, se tomó un tiempo el ejecutarlos, puesto que en primera instancia se tuvieron que adecuar a los parámetros y características que este proyecto exigía, con el objetivo de facilitar su aplicación y seguimiento por los equipos implicados en el desarrollo del proyecto de la empresa Sanambiente.

A pesar de todas las dificultades que se presentaron durante la aplicación del presente marco de trabajo al proyecto propuesto por la empresa Sanambiente, se logró que fuera el adecuado para ser seguido por todos los integrantes que estaban implicados en su desarrollo, alcanzando la finalización y cierre del mismo con satisfacción y cumpliendo con lo requerido por el cliente.

Se podría decir que este proyecto fue muy extenso, puesto que se prolongó más de lo esperado. la implementación de la ISO/IEC 29110 con el proceso de gestión con los equipos fue muy complejo, debido que hubo  diversos factores que afectaron la integración exitosa  como la carencia de conocimientos por parte de los equipos esto llevo a un retraso con las entregas de la actividades establecidas y así afectando el avance del proyecto, por otra parte la falta comunicación donde al principio no se tenía nada claro con el cliente para  así poder efectuar la elaboración del proyecto, ya que casi no había comunicación con el esto afecto a todos los equipos porque cada uno dependía del otro.

El proyecto propuesto por la empresa Sanambiente pasó por diversas etapas con los equipos, una de ellas fue la integración de todos los miembros y las actividades que tenían a cargo. Al inicio del proyecto hubo incertidumbre en cuanto al rumbo del proyecto por parte de algunos equipos, puesto que hubo ingreso de nuevos integrantes a este, pero al corto tiempo desertaron. Entre las responsabilidades que iban a cumplir estas personas estaban: procesamiento y sistema geográfico de los datos ambientales y aplicación de usabilidad en el aplicativo. Luego de ocurrir todo este proceso, finalmente quedaron consolidados todos los equipos: calidad, desarrollo y gestión.

Otra etapa por la que pasó el proyecto durante su desarrollo fue la carencia de comunicación con el cliente. Fue complejo entablar comunicación continua con el cliente, puesto que este no contaba con la disponibilidad suficiente para atender las solicitudes que todo el equipo poseía.

La etapa por la que se pasaron las mayores dificultades fue durante la aplicación del marco de trabajo en el proyecto. Entre las dificultades que se presentaron, estas fueron las principales:

1. **Dificultad de aprendizaje que retrasaron las entregas de las tareas que tenían programadas para una fecha específica.** Los equipos tuvieron demora en la adquisición de conocimiento necesario para llevar acabo sus tareas, por lo cual sufrieron atrasos que afectaron el avance de las otras tareas. Debido al aumento de tiempo en la curva de aprendizaje, fue necesario actualizar las fechas que inicialmente estaban estipuladas tanto en el plan del proyecto como en el registro de estado de progreso.
2. **Demoras en las entregas de las tareas en las fechas acordadas.** Estas demoras están en parte ligadas al primer punto, dado que fue unos de los factores que ocasionaron estos atrasos. Otro factor que afecto la entrega cumplida de las tareas, fue la carencia de compromiso y responsabilidad por parte de los equipos, puesto que no estaban acostumbrados a seguir un plan, un cronograma para llevar a cabo la ejecución de actividades o entregas. Se identificó que gran parte de los atrasos se encontraban en aquellas tareas que tenían un grado de dificultad alto para su realización, por esta razón fue necesario reordenar las actividades junto con sus fechas de modo que quedaran por grado de dificultad, es decir, las actividades con un grado de dificultad bajo se realizaban primero y aquellas con un grado de dificultad alto se realizaban al final. Esta estrategia permitió que las actividades fueran entregadas cumplidamente en las nuevas fechas planteadas.
3. **Dificultad de comunicación entre los equipos de trabajo.** Se encontraba poca comunicación entre los equipos de trabajo con los gestores del proyecto en cuanto a entrega de tareas se refiere. En el momento en que alguno de los equipos subía una determinada cantidad de tareas al repositorio con el fin de cumplir las fechas estipuladas, no le era notificado al equipo de gestión de esta acción, que en primera instancia se les había indicado hacerlo para llevar un control y seguimiento apropiado.

Los equipos de calidad y desarrollo tenían actividades de dependencia mutua, es decir, entre ellos dependía el avance en algunas tareas. Por ejemplo, si uno de los equipos tenía a cargo una actividad, y de esta dependía el otro equipo, este último no podía avanzar en la realización de la suya. Los gestores eran los encargados de notificar al equipo dependiente de una actividad de otro equipo si ésta ya fue entregada; pero, debido a esta carencia de comunicación, se presentaban situaciones donde un equipo subía una tarea de la cual dependía el otro equipo, y estos no les informaban a los gestores para realizar el seguimiento respectivo, ocasionando evidentemente muchos atrasos y demoras en las entregas de los equipos implicados.

Como estrategia para dar solución a esta dificultad, se realizaron reuniones virtuales vía llamada o chat como medios de comunicación con todos los equipos para conocer el estado actual de las actividades que tenían a cargo; manifestar las dificultades que de pronto estaban teniendo al realizarlas (por ej. debido a actividades de dependencia mutua) y de esta manera darles un tiempo adicional para entregarlas; y notificarles a los equipos aquellas que debían entregar en pocos días. Esto permitió que la comunicación entre todos los equipos mejorara, haciendo más fácil el desarrollo, finalización y entrega de sus actividades en un menor tiempo.

1. **Carencia de compromiso en la asistencia a las reuniones pactadas.** Se presentaron varias ocasiones donde los equipos no asistían a las reuniones que se acordaban en conjunto, algunas veces sí se justificaba su inasistencia debido a problemas de fuerza mayor, pero en muchas de ellas no se encontraba la causa de su inasistencia, por lo que se evidenció la falta de compromiso y seriedad de los integrantes en el proyecto. Estas reuniones tenían como fin informar a todos los equipos el avance de sus actividades, el estado de estas (pendientes o completadas), o si tenían dudas o problemas con la realización de algunas; pero, dado que no asistían, se creaba una incertidumbre acerca del estado actual de las actividades que tenían a su cargo, causa de carencia de compromiso y responsabilidad.

Para dar solución a esta dificultad, se establecía comunicación con los equipos de trabajo vía chat al final de cada reunión a la que no asistían, con el fin de conocer el motivo de su ausencia y llegar a un acuerdo con ellos para fijar un día diferente al de la próxima reunión para la revisión de las actividades pendientes por entregar. Además, se les daba una advertencia, informándoles que su permanencia en el proyecto estaba poniéndose en riesgo dadas las inasistencias constantes que estaban incurriendo.

Gracias a la implementación de esta estrategia, se logró mantener sin problema alguno la entrega continua de las actividades por parte de cada equipo de trabajo.

De este trabajo se obtuvieron muchas lecciones aprendidas que permitieron ajustar los lineamientos del estándar ISO/IEC 29110 y la metodología Iconix a las necesidades del equipo de trabajo y el proyecto. Estas lecciones han sido la mayor ganancia para el equipo, permitiendo confrontar las dificultades que se pueden llegar a tener en la industria.

Entre las lecciones aprendidas, cabe mencionar que:

* Se adquirieron habilidades básicas para la adecuación e implementación de estándares y metodologías a proyectos de software que poseen características particulares.
* Se adquirieron conocimientos básicos relacionados con la implementación de marcos de trabajo sobre proyectos de software propuestos por empresas.
* Se obtuvieron habilidades de comunicación para la coordinación y dirección adecuada de equipos de trabajo pertenecientes a un proyecto de software.
* Se asimilaron competencias para la asociación entre estándares y metodologías de desarrollo de software con el objetivo de crear un marco de trabajo capaz de ser implementado en un proyecto de software.

Las mejoras que se pueden aplicar a este proyecto, pueden estar basadas en las dificultades mencionadas anteriormente. Si desde el inicio se ejecuta una buena planificación, se podrá tener una implementación del marco de trabajo sin problemas y contratiempos; esto no se pudo lograr con el presente proyecto, dado que era la primera vez que todo el equipo de trabajo (calidad, desarrollo y gestión) se enfrentaba a un caso de estos que estaba bajo las exigencias de una empresa, y se tuvo que tomar un amplio tiempo para adquirir los conocimientos necesarios para iniciar en forma el desarrollo del proyecto. Este es otro punto, los equipos desde el inicio deben capacitarse al menos con los conocimientos básicos para tener la capacidad de afrontar el proyecto que se les aproxima.

Como recomendación fundamental, tener en cuenta las dificultades presentadas en este trabajo, para que los equipos que vayan a enfrentar a un proyecto similar, sepan cómo actuar y evitarlas.

Respecto a la ISO/IEC 29110, la metodología Iconix e IEEE 829, no se tiene mejora alguna para adicionarles. Tal vez, reducir la documentación que exigen para poder aplicarlas, pero basta sólo con adecuarlas al contexto del proyecto y no habría problemas para implementarlas.

En este proyecto se buscó aplicar técnicas que facilitaran la gestión de los procesos en torno a las características del equipo de trabajo, para garantizar un producto de calidad para la empresa Sanambiente. El resultado final fue un marco de trabajo elaborado bajo los lineamientos utilizados en la industria para el desarrollo de proyectos de software.

Para concluir, el presente trabajo queda a disposición de mejoras por parte de cualquier persona que desee aumentar la eficacia de su implementación en proyectos futuros.

# Referencias

Aguado Álvarez, J. C. (2019). Implementación de pruebas de calidad en la aplicación control currículo v.1 de la Institución Universitaria Antonio José Camacho. Institución Universitaria Antonio José Camacho.

Amavizca Valdez, L. O., García Ruíz, A. C., Jiménez López, E., Duarte Guerrero, G. L., & Vásquez Brindis, J. C. (2014). Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software: Implementación y publicación de un sitio WEB para una empresa SPIN -OFF en el Sur de Sonora, México. [Científico]. http://laccei.org/LACCEI2014-Guayaquil/RefereedPapers/RP246.pdf

Bolívar Guzmán, J., Lozano Argote, J., & Ramírez, J. J. (2020). Desarrollo de los módulos de transferencia y comunicación de datos ambientales para la empresa de Sanambiente de Cali. Institución Universitaria Antonio José Camacho.

Bona, C. (2002). AVALIAÇÃO DE PROCESSOS DE SOFTWARE: UM ESTUDO DE CASO EM XP E ICONIX [Universidade Federal de Santa Catarina]. https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/82842/193091.pdf?sequence=1&isAllowed=y

Calvo Manzano, J., & García, I. (2008). Hacia la gestión cuantitativa en la gestión de proyectos en el ámbito de las pymes. Revista Español de Innovación, Calidad e Ingeniería del Software, 4(2), 7-19.

Cataldi, Z. (2000). Una metodología para el diseño, desarrollo y evaluación de software educativo [Universidad Nacional de la Plata]. http://sedici.unlp.edu.ar/bitstream/handle/10915/4055/Documento\_completo.pdf?sequence=20&isAllowed=y

Estrada Reyes, J. N. (2015). Análisis de la gestión de proyectos a nivel mundial. Palermo Business Review, 12, 38.

Fiestas Jacinto, J. E. (2015). La implementación de un sistema de inteligencia de negocios que permita mejorar la toma de decisiones respecto a las remuneraciones de la empresa pesquera Carlos Eduardo S.R.L.-2014 [Universidad Nacional de Trujillo]. http://dspace.unitru.edu.pe/bitstream/handle/UNITRU/7684/Tesis%20Maestr%C3%ADaX%20-%20Juan%20Eduardo%20Fiestas%20Jacinto.pdf?sequence=1&isAllowed=y

González Toste, D., Valdés González, A. A., Romero Gómez, Y., & García Pérez, Y. (2017). El uso de las TIC como apoyo al proceso de formación máster en «Ciencias de la educación». 2, núm. 4(octubre-diciembre (2014)), 1-11.

Hugues, P., Fisher, P., & Mc Daniel, J. (2010). System development life cycle models and methodologies. Canadian Society for International Health Certificate Course in Health Information Systems.

ICONIX Brand Group. (2016). Manual introductorio de ICONIX. http://ima.udg.edu/%7Esellares/EINF-ES2/Present1011/MetodoPesadesICONIX.pdf

IEEE. (1991). A Compilation of IEEE Standard Computer Glossaries”, IEEE Standard Computer Dictionary (IEEE-STD-610). https://ieeexplore.ieee.org/document/182763

ISO/IEC. (2011). Software engineering—Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile. https://www.iso.org/obp/ui/#iso:std:iso-iec:tr:29110:-5-2-1:ed-1:v1:en

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). El proceso unificado de desarrollo de software (S. Sánchez, M. Á. Sicilia, C. Canal, & F. J. Durán, Trads.). Addison Wesley. https://es.scribd.com/doc/50327385/El-Proceso-Unificado-de-Desarrollo-de-Software-Jacobson-Booch-Rumbaugh

Laporte, C., O´connor, R., & García, L. (2016). THE IMPLEMENTATION OF ISO/IEC 29110 SOFTWARE ENGINEERING STANDARDS AND GUIDES IN VERY SMALL ENTITES [École de technologie supérieure, Montréal, Canada, School of Computing, Dublin City University, Dublin, Ireland, Universidad Peruana de Ciencias Aplicadas, Lima, Peru]. https://link.springer.com/chapter/10.1007/978-3-319-30243-0\_9

Legaria Islas, D. J. (2018). IMPLEMENTACIÓN DE PROCESO ORGANIZACIONAL DE GESTIÓN DE PROYECTOS EN DEVELOPIT. http://repositorio.uchile.cl/handle/2250/159559

Lugo García, J. A., & Marín Sánchez, J. (2016). Control de proyectos de software: Actualidad y retos para la industria cubana. 24(1). http://dx.doi.org/10.4067/S0718-33052016000100010

Madruñero Padilla, E. R. (2018). Implementación del estándar ISO/IEC 29110 en el proceso de desarrollo de software de la dirección de desarrollo tecnológico e informático de la Universidad Técnica del Norte [Universidad Técnica del Norte]. http://repositorio.utn.edu.ec/bitstream/123456789/8165/1/PG%20638%20TESIS.pdf

Marín Ospina, B. E. (2020). Adecuación de la norma ISO/IEC 29110 e IEEE 829 para la gestión de proyectos de desarrollo con metodología Iconix. 15.

Matla Cruz, E. O. (2014). Desarrollo de software guiado por la norma ISO/IEC 29110 y Scrum: SIDEP v.2.0 [Universidad Nacional Autónoma de México]. http://132.248.9.195/ptd2014/enero/0707739/0707739.pdf#page=32&zoom=100,0,152

Mina Posu, F. A., & Romero Dorado, J. D. (2020). Aplicación de un modelo de calidad al módulo de comunicación y traspaso de información del sistema de gestión de datos ambientales de la empresa Sanambiente. Institución Universitaria Antonio José Camacho.

Mnkandla, E., & Dwolatzky, B. (2004). A Survey of Agile Methodologies [Científico]. https://www.ee.co.za/wp-content/uploads/legacy/Trans12p236-247.pdf

PORRAS. (2019). “METODOLOGÍA ÁGIL ICONIX EN LA CALIDAD DEL PRODUCTO SOFTWARE, LIMA, 2017” [Universidad Nacional Federico villareal]. http://repositorio.unfv.edu.pe/handle/UNFV/2956

Pressman, R. S. (2002). Ingeniería del Software. Un enfoque práctico (R. Martín Ojeda, V. Yagüe Galaup, I. Morales Jareño, & S. Sánchez Alonso, Trads.; Quinta Edición). Concepción Fernández Madrid. http://www.javier8a.com/itc/bd1/ld-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF

Puerta Marín, H. A., & Espinosa Bedoya, A. (2015). UNA ADAPTACIÓN DE PMBOK AL CICLO DE VIDA DE DESARROLLO DE PROYECTOS SOFTWARE EN PEQUEÑAS EMPRESAS. VI Congreso Iberoamericano de Ingeniería de Proyectos - CIIP. https://www.researchgate.net/profile/Albeiro\_Espinosa\_Bedoya/publication/284717563\_UNA\_ADAPTACION\_DE\_PMBOK\_AL\_CICLO\_DE\_VIDA\_DE\_DESARROLLO\_DE\_PROYECTOS\_SOFTWARE\_EN\_PEQUENAS\_EMPRESAS/links/5746166608ae9f741b4313cb/UNA-ADAPTACION-DE-PMBOK-AL-CICLO-DE-VIDA-DE-DESARROLLO-DE-PROYECTOS-SOFTWARE-EN-PEQUENAS-EMPRESAS.pdf

Ramos, C., & Mendoza, L. (2014). Implementación del estándar ISO/IEC 29110-4-1 para pequeñas organizaciones de desarrollo de software [Universidad Peruana de Ciencias Aplicadas]. https://repositorioacademico.upc.edu.pe/handle/10757/336665?show=full

Roberts, A., & Wallace, W. (2002). Gestión de Proyectos (1.a ed., p. 54). Edinburg Business School, Heriot-Watt University. https://www.ebsglobal.net/documents/course-tasters/spanish/pdf/pr-bk-taster.pdf

Rosenberg, D., & Scott, K. (2001). Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example (First Edition). Addison Wesley. https://pja.mykhi.org/0sem/MAS/books/Addison.Wesley.Applying.Use.Case.Driven.Object.Modeling.pdf

Rosenberg, D., Stephens, M., & Collins-Cope, M. (2005). Agile Development with ICONIX Process. Apress.

Scalone, F. (2006). Estudio comparativo de los modelos y estándares de calidad del software [Universidad Tecnológica Nacional Facultad Regional Buenos Aires]. http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf

Silva Ascuntar, S., & García G., Y. C. (2018). Documentación de la metodología ICONIX a través del desarrollo del caso «Oriéntate Cali». Institución Universitaria Antonio José Camacho.

UNE. (1995). Gestión de la calidad y aseguramiento de la calidad. Vocabulario. (ISO 8402). https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0013715

Estatuto de Investigaciones y Producción Intelectual, Pub. L. No. Acuerdo No. 14, 11 (2018). http://www.uniajc.edu.co/documentos/secgeneral/ESTATUTO\_GENERAL\_\_022\_2007.pdf

Vásquez González, P. (2007). Metodologías de Gestión de proyectos, alcance, impacto y tendencias [Universidad de Chile]. http://repositorio.uchile.cl/bitstream/handle/2250/108432/ec-vasquez\_g.pdf?sequence=3&isAllowed=y