**Desarrollo de los módulos de transferencia y comunicación de datos ambientales para la empresa de SANAMBIENTE de Cali**

Jagüel Lozano Argote, Jhanluy Bolívar Guzmán & Jhon Jairo Ramírez

Santiago de Cali, 2019

Institución Universitaria Antonio José Camacho

Facultad de Ingeniería

Tecnología en Sistemas de Información

**Resumen**

El presente proyecto consistió en el desarrollo de un módulo para realizar la conexión y la transferencia de datos ambientales desde una estación física de monitoreo perteneciente a la empresa Sanambiente.

Este trabajo se concentra en establecer la conexión a través de los protocolos de comunicación y la configuración necesaria para ello, la recepción de los datos y almacenaje de los mismos, con la finalidad de lograr una mejora en los procesos y una protección de los datos durante dichos procesos. El software está enmarcado en el tipo de investigación aplicada. Se emplearon una serie de técnicas de recolección de datos, específicamente a través de consultas y entrevistas con el cliente; acerca de procedimientos que se realizan en cada uno de los procesos para generar prototipos y casos de uso, los cuales sirvieron para desarrollar el aplicativo web. Este proyecto se realizó en paralelo con dos proyectos de ingeniería de sistemas de la UNIAJC (Institución Universitaria Antonio José Camacho) que se encargaron de los procesos de gestión y calidad.

Esta aplicación web fue elaborada buscando dar cumplimiento a los objetivos planteados, para ello se utilizaron diversas herramientas tecnológicas como AngularCli que es una herramienta de NodeJS para el desarrollado de aplicaciones web, y se utilizó el sistema de gestión de base de datos de PostgreSql. Además, se implementó la metodología Iconix para el desarrollo del proyecto y patrones de diseño tales como Inyección de Dependencias y Repository.

**Abstract**

The present project consists of the development of a module to carry out the connection and the transfer of environmental data from a physical monitoring station belonging to the Sanambiente Company.

This work focuses on establishing the connection through the communication protocols and the necessary configuration for it, the reception of the data and storage of the same, in order to achieve an improvement in the processes and a protection of the data during said processes. The software is framed in the type of applied research. A series of data collection techniques were employed, specifically through consultations and interviews with the client; about procedures that are performed in each of the processes to generate prototypes and use cases, which served to develop the web application. This project was carried out in parallel with two systems engineering projects of the UNIAJC (Antonio José Camacho University Institution) that will be in charge of the management and quality processes.

This web application was developed seeking to fulfill the stated objectives, for this, various technological tools are used, such as AngularCli, which is a NodeJS tool for the development of web applications, and the PostgreSql database management system is used. In addition, the Iconix methodology will be implemented for project development and story design patterns such as Dependency Injection and Repository.

**Tabla de Contenidos**

[Introducción 1](#_Toc410628920)

1) [Problema de investigación](#_Toc410628920) 3

1.1) Planteamiento del problema3

1.2) Formulación del problema 3

1.3) Sistematización del problema 4

2) [Objetivos](#_Toc410628920) 4

2.1) Objetivo general4

2.2) Objetivos específicos 4

3) Justificación5

4) [Marco referencial](#_Toc410628920) 6

4.1) Antecedentes6

4.2) Marco conceptual8

4.3) Marco teórico [1](#_Toc410628922)4

4.4) Marco legal [1](#_Toc410628922)9

5) [Desarrollo de contenido (Iconix) 2](#_Toc410628925)0

5.1) [Fase 1 Análisis de requerimientos 2](#_Toc410628926)0

5.1.1) [Requerimientos 2](#_Toc410628927)0

5.1.2) [Casos de uso 2](#_Toc410628928)1

5.1.3) [Prototipos de interfaz 4](#_Toc410628927)0

5.2) [Fase 2 Analisis y diseño preliminar 52](#_Toc410628926)

5.2.1) [Diagrama de clases 52](#_Toc410628927)

5.3) [Fase 3 Diseño detallado 67](#_Toc410628926)

5.3.1) [Diagramas de secuencia 67](#_Toc410628927)

5.4) [Fase 4 Implementación 80](#_Toc410628926)

5.4.1) [Pruebas unitarias 80](#_Toc410628927)

5.4.2) [Pruebas de carga 8](#_Toc410628927)1

6) [Resultados](#_Toc410628929) 85

7) [Conclusiones](#_Toc410628929) 87

8) Bibliografía88

**Lista de Tablas**

[Tabla 1. Requerimientos.](#_Toc410629016) 20

Tabla 2. Caso de uso No.1 conectar con estación FTP 21

[Tabla 3. Caso de uso No.2 conectar con estación ModBus](#_Toc410629016) 22

[Tabla 4. Caso de uso No.4 crear región](#_Toc410629016) 23

[Tabla 5. Caso de uso No.5 crear estación](#_Toc410629016) 24

[Tabla 6. Caso de uso No.6 crear organización](#_Toc410629016) 26

[Tabla 7. Caso de uso No.7 crear categoría](#_Toc410629016) 28

[Tabla 8. Caso de uso No.8 crear ciudad](#_Toc410629016) 29

[Tabla 9. Caso de uso No.9 crear rango](#_Toc410629016) 30

[Tabla 10. Caso de uso No.10 crear alerta](#_Toc410629016) 32

[Tabla 11. Caso de uso No.11 crear cuadro de tiempo](#_Toc410629016) 33

[Tabla 12. Caso de uso No.15 conectar automáticamente con estación](#_Toc410629016) 35

[Tabla 13. Caso de uso No.16 crear mantenimiento](#_Toc410629016) 36

[Tabla 14. Caso de uso No.25 crear tipos de mantenimiento](#_Toc410629016) 37

[Tabla 15. Caso de uso No.26 crear periodicidad de mantenimiento](#_Toc410629016) 38

[Tabla 16. Caso de uso No.27 crear partes de mantenimiento](#_Toc410629016) 39

**Lista de Figuras**

[Figura 1.Proceso de Iconix](#_Toc410629185) 19

Figura 2. Prototipo conectar estación con FTP, ModBus y automáticamente 40

Figura 3. Prototipo crear región41

Figura 4. Prototipo crear estación42

Figura 5. Prototipo crear organización[.](#_Toc410629185) 43

Figura 6. Prototipo crear ciudad[.](#_Toc410629185) 44

Figura 7. Prototipo crear rango [45](#_Toc410629185)

Figura 8. Prototipo crear alerta[. 46](#_Toc410629185)

Figura 9. Prototipo crear base de tiempo [47](#_Toc410629185)

Figura 10. Prototipo crear mantenimiento  [48](#_Toc410629185)

Figura 11.Prototipo crear tipo de mantenimiento [49](#_Toc410629185)

Figura 12. Prototipo crear periodicidad de mantenimiento[. 50](#_Toc410629185)

Figura 13. Prototipo crear partes de mantenimiento  [51](#_Toc410629185)

Figura 14. Diagrama de clases conectar con FTP, ModBus y automáticamente  [52](#_Toc410629185)

Figura 15. Diagrama de clases tabla parámetro región  [5](#_Toc410629185)3

Figura 16. Diagrama de clases tabla parámetro estación [5](#_Toc410629185)4

Figura 17. Diagrama de clases tabla parámetro organización [. 5](#_Toc410629185)5

Figura 18. Diagrama de clases tabla parámetro categoría [. 5](#_Toc410629185)6

Figura 19. Diagrama de clases tabla parámetro ciudad [. 5](#_Toc410629185)7

Figura 20. Diagrama de clases tabla parámetro rango [5](#_Toc410629185)8

Figura 21. Diagrama de clases tabla parámetro alerta59

Figura 22. Diagrama de clases tabla parámetro base de tiempo[. 60](#_Toc410629185)

Figura 23. Diagrama de clases mantenimiento  [6](#_Toc410629185)1

Figura 24. Diagrama de clases de partes de mantenimiento  [6](#_Toc410629185)2

Figura 25. Diagrama de clases de periodicidad de mantenimiento [. 6](#_Toc410629185)3

Figura 26. Diagrama de clases de tipo de mantenimiento[. 6](#_Toc410629185)4

Figura 27. Diagrama de secuencias conectar con FTP y ModBus y automáticamente  [6](#_Toc410629185)5

Figura 28. Diagrama de secuencias crear región.[. 6](#_Toc410629185)6

Figura 29. Diagrama de secuencias actualizar región.[. 6](#_Toc410629185)6

Figura 30. Diagrama de secuencias crear estación[. 6](#_Toc410629185)7

Figura 31. Diagrama de secuencias actualizar estación [. 6](#_Toc410629185)7

Figura 32. Diagrama de secuencias crear organización [.](#_Toc410629185) 68

Figura 33. Diagrama de secuencias actualizar organización[.](#_Toc410629185) 68

Figura 34. Diagrama de secuencias crear categoria[.](#_Toc410629185) 69

Figura 35. Diagrama de secuencias actualizar categoria [.](#_Toc410629185) 69

Figura 36. Diagrama de secuencias crear ciudad[.](#_Toc410629185) 70

Figura 37. Diagrama de secuencias actualizar ciudad [.](#_Toc410629185) 70

Figura 38. Diagrama de secuencias crear rango[.](#_Toc410629185) 71

Figura 39. Diagrama de secuencias actualizar rango[.](#_Toc410629185) 71

Figura 40. Diagrama de secuencias crear alerta[.](#_Toc410629185) 72

Figura 41. Diagrama de secuencias actualizar alerta[.](#_Toc410629185) 72

Figura 42. Diagrama de secuencias crear base de tiempo [.](#_Toc410629185) 73

Figura 43. Diagrama de secuencias actualizar base de tiempo[.](#_Toc410629185) 73

Figura 44. Diagrama de secuencias crear mantenimiento[.](#_Toc410629185) 74

Figura 45. Diagrama de secuencias crear tipos de mantenimiento[.](#_Toc410629185) 75

Figura 46. Diagrama de secuencias actualizar tipos de mantenimiento[.](#_Toc410629185) 75

Figura 47. Diagrama de secuencias crear periodicidad de mantenimiento[.](#_Toc410629185) 76

Figura 48. Diagrama de secuencias actualizar periodicidad de mantenimiento [.](#_Toc410629185) 76

Figura 49. Diagrama de secuencias crear partes de mantenimiento[.](#_Toc410629185) 77

Figura 50. Diagrama de secuencias actualizar partes de mantenimiento[.](#_Toc410629185) 77

Figura 51. Resultados de las pruebas unitarias[.](#_Toc410629185) 79

Figura 52. Resultados de las pruebas de carga [.](#_Toc410629185) 81

Figura 53. Resultados pantallazo de inicio .[.](#_Toc410629185) 82

Figura 54. Resultados pantallazo listar estaciones. [.](#_Toc410629185) 82

Figura 55. Resultados pantallazo información para conectar 83

Figura 56. Resultados pantallazo conectar y transferir [.](#_Toc410629185) 84

# Introducción

**En el presente documento hace referencia al módulo de comunicación y transferencia de datos de la aplicación web la cual fue desarrollado debido a la necesidad de mejorar el proceso de conexión con las estaciones de monitoreo y la administración durante la transferencia de los datos obtenidos por los dispositivos de medición de contaminantes ambientales de la empresa Sanambiente, ya que actualmente no existe un módulo que permita un manejo adecuado y unificado de los datos correspondientes a las estaciones y los contaminantes.**

**El objetivo principal del módulo de comunicación y transferencia de datos es brindar las herramientas necesarias para lograr controlar los procesos de conexión y comunicación de datos obtenidos por las diferentes estaciones de medición ubicadas en todo el país ya que el sistema actual no permite adecuaciones y trae elementos adicionales que la empresa no requiere.**

**Se creó una base de datos mediante el sistema de gestión PostgreSql y el desarrollo del frontend y backend se utilizó TypeScripty Bootstrap todo con la ayuda del framework Angular siguiendo los patrones de diseño inyección de dependencias de angular (Dependency Injection o DI) y el patrón de diseño Repositorio (Repository).**

**Con la finalidad de tener un control de tiempo y desarrollar las actividades de forma precisa, se realizó una comparación entre algunas metodologías para el desarrollo de software que pudieran asegurar los plazos de tiempo y funcionalidad para el desarrollo de un aplicativo web que logre conectar y transferir información de las estaciones de la empresa Sanambiente. Se considero utilizar la metodología Iconix que es una metodología** Semi-Ágil**, debido a su disposición para equipos pequeños de desarrolladores, su sencillez de reglas y prácticas, su flexibilidad para los cambios y su tendencia de colaboración es precisa para este proyecto. Este módulo se formó en base a dos proyectos de ingeniería de sistemas de la UNIAJC que se encargaron de los procesos de ingeniería de requerimientos los cuales fueron vitales para identificar las necesidades del cliente y calidad-gestión quienes en simultaneo ayudaban al proceso de desarrollo aportando correcciones y pruebas del software, todo es tomado de** Grajales Gómez, & Balanta Pazu (2019) y Romero Dorado & Mina Posu, (2020).

# Problema de Investigación

## Planteamiento del Problema

La empresa Sanambiente ofrece soluciones integrales en productos y servicios que aportan al cuidado y mejoramiento de las condiciones ambientales en los países donde tienen presencia, actualmente cuenta con clientes potenciales a nivel nacional que a diario necesita información de las mediciones ambientales de forma rápida, verídica, accesible y fácil, en este momento la empresa cuenta con un software de escritorio que entrega a cada cliente el cual no satisface completamente las necesidades de la empresa y de sus clientes. La aplicación que usan actualmente no fue hecha a la medida y trae gran cantidad de componentes que dificultan su manejo.

Así se identifica el siguiente problema de gestión: dificultad para acceder a la información requerida de forma oportuna desde cualquier parte del país.

El aplicativo se desarrolla por módulos, el módulo que se desarrolló en el actual documento es para la conexión a través de protocolos de comunicación como ModBusTCP/IP y Ftp, la transferencia de datos desde el datalogger que recibe la información de los sensores de las estaciones y el almacenamiento de dichos datos.

## Formulación del Problema

¿De qué manera se puede transferir la información que se encuentra en el datalogger al cliente, facilitando el control y seguimiento de los datos?

## Sistematización del Problema

* ¿Cómo realizar la conexión con el datalogger?
* ¿Cómo transferir los datos para almacenarlos en una base de datos?
* ¿Cómo facilitar la interacción del usuario para el control y seguimiento de la información?

# Objetivos

## 2.1) Objetivo General

Desarrollar un módulo que permita conectar y transferir información de los diferentes datalogger de la empresa SANAMBIENTE con sede en Cali.

## 2.2) Objetivos Específicos

* Establecer la conexión con el datalogger siguiendo los protocolos de comunicación ModBus TCP/IP y FTP.
* Transferir información de las estaciones de acuerdo con las plantillas definidas por el usuario para almacenar en una base de datos la información.
* Desarrollar interfaces que permitan la interacción del usuario con la información de las estaciones.

# Justificación

El módulo de transferencia y comunicación de datos es necesario para el personal operativo de Sanambiente, porque les brinda la facilidad para realizar la conexión con una determinada estación y la recepción de datos sobre contaminantes capturados por los sensores, lo cual permite llevar un registro sobre la información de las variables recolectadas por cada uno de los dispositivos de medición, su respectiva ubicación e información relevante de cada datalogger.

La transferencia de datos es almacenada de forma rápida y veraz gracias a los protocolos de comunicación.

Este proyecto permite obtener y salvaguardar los datos sobre los contaminantes y sus niveles, por lo tanto, podrían ayudar a tomar decisiones en la prevención y cuidado del medio ambiente.

# Marco Referencial

## 4.1) Antecedentes

Para el desarrollo del módulo se analizaron diferentes proyectos que tuvieron el mismo o similar objeto de estudio, estos fueron tomados como base para llevar a el proyecto, brindando una guía para saber cómo proceder y de qué forma iniciar la elaboración, los pasos a seguir, cuáles pueden ser de utilidad para el presente módulo y poderlos adecuar de acuerdo a las necesidades del usuario. La información que será mostrada en este marco es aquella que tienen relación con los temas del presente proyecto: tales como ISO/IEC 29110, Metodología Iconix y Angular Framework.

* **ISO/IEC 29110.**

Acogido por este marco se han desplegado varias investigaciones que tienen como meta demostrar su efectividad en diferentes entornos, un ejemplo de esto es el trabajo realizado por Madruñero (2018) que buscaba dar formalidad a la gestión de los desarrollos en el área de la Dirección de desarrollo de la Universidad Técnica del Norte. Los desarrollos de software en esta unidad se ambientaron con elementos de Scrum, pero sin llegar a ser rigurosos. Las fases de este proyecto comenzaron con la investigación documental del marco, el entendimiento del negocio y finaliza con la propuesta de un método mixto de integración de la norma con la metodología. Una conclusión a resaltar son los obstáculos presentes al tratar de articular esta norma con las metodologías ágiles, el escenario final conecto propiedades, roles y funciones de Scrum con principios de documentación de metodologías tradicionales que siguen las fases de inicio, planificación, ejecución-evaluación y cierre. (Marin, 2020)

* En el ensayo de Laporte, O´connor, & García, (2016) se exhibe el uso de esta norma en 7 casos diferentes que abarcan pequeñas empresas del Perú y Canadá revelando su eficacia para organizar y efectuar proyectos sin obstaculizar la inventiva de los desarrolladores, teniendo en presente hitos como costos y tiempo.

Metodología Iconix.

* En el proyecto de Porras, (2019) se implemento la metodología Iconix y se obtuvo como resultado la mejora en la calidad del desarrollo del software que se estaba buscando, gracias a esta metodología que establece los pasos a seguir y los cuales se enfocan en cómo desarrollar un sistema, Este trabajo ayudara a la empresa que brinda el servicio de desarrollo de software, nuevas formas de crear aplicaciones de manera ágil y así obtener beneficios como: disminuir el tiempo y los costos.
* En la tesis de Silva Ascuntar & García G., (2018) se demuestra que la metodología es Semi-Ágil, interactiva, trazable y que principalmente se implementa en sistemas de gestión con la activa participación de los usuarios finales quienes tienen un rol muy importante ya que ayudan a definir los casos de uso.

El resultado que se puede obtener de este proyecto es que la metodología Iconix tiene ciertas características de la metodología RUP y otras de la metodología XP debido a esto se transforma en una metodología Semi-Ágil. Lo cual llevo el proyecto a un cumplir con los tiempos estipulados y lograron el resultado esperado.

**Angular framework.**

* De acuerdo con la tesis de Valencia, (2018) se demuestra que Angular Framework es muy util para crear una aplicación más rápida y fácil de usar. La arquitectura SPA traslada gran parte de la lógica de presentación y negocios al navegador web, pero el servidor sigue siendo muy importante para el proyecto.

## 4.2) Marco Conceptual

* **AngularJS.** Es un framework modelo vista control de JavaScript que se utiliza para el desarrollo páginas web frontend que permite crear aplicaciones SPA (Single-Page Applications). (Azaustre, 2013)
* **Framework.** Es un marco de trabajo o conjunto estandarizado de conceptos, prácticas y criterios para orientar un tipo de problemática particular que sirve como referencia, en el desarrollo de aplicaciones, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que sirven de base para la organización y desarrollo de aplicativos. (Haro & Camps Riba , 2008)
* **Angular framework.** El objetivo principal de Angular es facilitar la creación de aplicaciones cliente en HTML y JavaScript o en el lenguaje TypeScript que se agrupa en JavaScript. El framework consta de varias bibliotecas, el programador escribe una aplicación Angular con templates de componentes en HTML con el framework Angular, se desarrolla clases de componentes para administrar estas plantillas, se agrega lógica de aplicaciones a los servicios y componentes en todos los módulos. (Angular., 2018)
* **JavaScript.** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se precisa como orientado a objetos, basado en prototipos, imperativo, dinámico y tipado. (Flanagan, JavaScript: The Definitive Guide, 6th Edition, 2011)
* **PostgreSQL.** Es un avanzado sistema de bases de datos relacionales basado en Open Source (código abierto). Esto quiere decir que el código fuente del programa está disponible a cualquier usuario libre de cargos directos, permitiendo a cualquiera persona colaborar con el desarrollo del proyecto o editar el sistema para ajustarlo a sus necesidades. PostgreSQL está bajo licencia Berkeley Software Distribution (BSD). (The PostgreSQL Global Development Group, 2019)
* Frontend. Es el módulo de un aplicativo web que interactúa con los usuarios, por lo cual es la parte que el cliente o usuario final visualiza.
* Backend. Es el módulo que se conecta con la base de datos y el servidor que utiliza un aplicativo web, este hace parte del servidor.
* **Bootstrap.** Es el framework de CSS, HTML y Java Script más popular, que nos permite desarrollar aplicaciones webs que se ajustan a cualquier resolución y dispositivo del mundo. (team of developers Bootstrap , 2011)
* **Css.** (siglas en inglés de Cascading Style Sheets) y en español «Hojas de estilo encascada» es un lenguaje de diseño gráfico para precisar y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado actualmente para establecer el diseño visual de los aplicativos webs, e interfaces de usuario escritas en Html. (Developer.mozilla.org, 2019).
* **Html.** Es un estándar que sirve para páginas web en sus diferentes versiones, define una estructura básica para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. (Berners-Lee)
* **Componente en Angular.** Un componente en Angular es una combinación de un archivo Html con un TypeScript y algunas veces Scss para crear un elemento con características propias tanto de comportamiento como de apariencia que se puede visualizar en un navegador web. (Rojas, 2019)
* **GitHub.** Es una plataforma de desarrollo colaborativo de aplicativos para alojar proyectos utilizando el sistema de control de versiones Git. (github)
* **Git.** Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. (github)
* **TypeScript.** Es un lenguaje de programación de código abierto con herramientas de programación orientada a objetos, muy propicio si se tienen proyectos a gran escala. (Caceres)
* **NodeJS.** Es un entorno en tiempo de ejecución multiplataforma, de código abierto, la arquitectura es orientada a eventos y basado en el motor V8 de Google. Es muy útil en la creación de software web altamente escalables. (Finley, 2011)
* Protocolo ModBus TCP/IP. Es un protocolo de comunicaciones empleado en los niveles 1, 2 y 7 del Modelo OSI (Open System Interconnection), basado en la arquitectura cliente/servidor (TCP/IP), es un protocolo de comunicaciones estándar. Las principales razones de su uso son:
  + Se diseñó teniendo en cuenta su uso para softwares industriales.
  + Es gratuito y público.
  + Maneja bloques de datos sin suponer restricciones.
  + Requiere poco desarrollo y es fácil de implementar. (Drury, 2009)
* **Protocolo Ftp.** El Protocolo de transferencia de archivos Ftp (File Transfer Protocol) es un protocolo para la transferencia de archivos entre sistemas conectados a una red Transmission Control Protocol (TCP), basado en la arquitectura cliente-servidor. (Bolo, 2011)
* **Datalogger.** Es un aparato electrónico que registra o almacena los datos en tiempo real y en relación a todos los sensores internos que se encuentren conectados. (JM Industrial, 2016)
* **Metodología Rup.** Significa Proceso Unificado Racional (Rational UnifiedProcess) es un proceso de desarrollo de aplicación es implementado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado (UML), forma la metodología estándar más utilizada para el diseño, análisis, implementación y documentación de programas orientados a objetos. (Taft, 2002)
* **Metodología Xp.** (Programación Extrema) Es una metodología de desarrollo que corresponde a las conocidas como metodologías ágiles cuyo objetivo es el desarrollo y gestión de proyectos con flexibilidad, eficacia y control. (IZQUIERDO, 2014)
* **Arquitectura Spa.** Es una aplicación de página única o es un aplicativo web que cabe en una sola página con el propósito de dar una experiencia más fluida a los usuarios finales como un software de escritorio. (Flanagan, JavaScript - The Definitive Guide, 2006)
* **Templates Angular.** Es lo que nos permite definir la vista de un componente. (Oriol, Introducción a Angular 2 (parte I) – Modulo, Componente, Template y Metadatos, 2016)
* **Componentes Angular.** Es quien controla un espacio de la pantalla que podríamos llamar vista. (Oriol, Introducción a Angular 2 (parte I) – Modulo, Componente, Template y Metadatos, 2016)
* **Servicio Angular.** Son fundamentales en AngularJS, siguen siendo imprescindibles en Angular 2 en adelante, todo valor, función o característica que la aplicación necesite, desde constantes a la lógica de negocio, se encapsula dentro de un servicio. (Oriol, Introducción a Angular 2 (parte III) – Servicios, 2016)
* **Patrón de diseño Repositorio.** Es un patrón de diseño de tipo estructural que opera como mediador entre el dominio y la capa de acceso a datos, permitiendo encapsular la lógica requerida para acceder a las fuentes de datos. Abstrae las operaciones sobre los modelos-capa-persistencia (operaciones Create, Read, Update **y** Delete) con la intención de que la implementación de dichas operaciones se oculte de las capas superiores. (Flower, Rice, Foemmel, Hieatt, Mee, & Stafford , 2002)
* **Inyección de dependencia en angular.** La inyección de dependencia que de forma abreviada es (DI) es un patrón de diseño de aplicación muy importante dentro del desarrollo en angular. Angular tiene su propio marco DI, que generalmente se usa en el diseño de aplicaciones angulares para aumentar su eficiencia y modularidad.

Las dependencias son servicios que una clase necesita para realizar una función. DI es un patrón de codificación en el que una clase solicita dependencias de fuentes externas en vez de crearlas por sí mismas. En Angular, el marco DI genera dependencias declaradas a una clase cuando se instancia esa clase. Este patrón de diseño proporciona que las aplicaciones sean más eficaces, flexibles y robustas, además de comprobables y mantenibles. (Cea, 2018)

## 4.3) Marco teórico

Existen diferentes metodologías que proporcionan la ejecución del ciclo de vida de un proyecto para un aplicativo web. Cada una es útil en un contexto en particular, están las ágiles que reducen la documentación y se orientan en las habilidades individuales del equipo, las robustas que parten de la jerarquización y se respaldan con documentación detallada y las semiágiles que combinan características de las dos anteriores expuestas.

Para este proyecto por políticas del equipo de ingeniería se ha optado por la metodología semiágil Iconix que facilita la trazabilidad con una documentación suficiente y utiliza como pilar los prototipos y los casos de uso para un mejor entendimiento con el usuario final quien acompaña eventualmente el proyecto.

**4.3.1) Iconix**

Es una metodología de desarrollo de software de tamaño medio, cuenta con una serie de pasos que se debe seguir para determinar rotundamente las actividades a desarrollar en cada período de un proyecto de desarrollo.

El proceso de Iconix describe cómo pasar de los casos de uso a la codificación de forma fiable, en el menor tiempo posible. Por tal motivo, la principal preocupación de Iconix es el análisis y diseño de los aspectos de modelado de la producción de software. (Rosenberg, Stephens, & Collins-Cope, 2005)

4.3.1.1) Características.

* Incremental e iterativo. Durante el desarrollo del modelo del dominio y la definición de los casos de uso se producen varias iteraciones. El ciclo de vida incremental consiste en desarrollar por etapas el producto de manera que puedas integrarlas funcionalmente, y en cada ciclo de iteración se revisa y mejora el producto.
* **Dinámica del UML.** Brinda un uso dinámico del UML ya que utiliza algunos diagramas UML, sin exigir el uso de todos.
* **Trazabilidad.** Cada paso realizado está definido por un requisito puntual, se define la trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos de software producidos. (Silva Ascuntar & García G., 2018)

**4.3.2) Fases de Iconix.**

**4.3.2.1) Revisión de los requisitos / análisis de requisitos.**Es la primera fase la cual conciste en identifcar el mundo real, los objetos y todas las relaciones de agregación y generalización entre ellos. Se deben estudiar todos los requisitos que formaran parte del sistema y con estos construir el diagrama de clases, que representa las agrupaciones funcionales que estructuraran el sistema en el desarrollo del aplicativo web.

***Para esta fase se utilizan 3 herramientas.***

* *Modelo de dominio.* Este modelo se refiere a identificar objetos y cosas del mundo real que intervienen con el sistema. (Estatico)
* *Prototipo de interfaz de usuario****.***Implica la creación de modelos operativos del trabajo de un sistema, en el que analistas y clientes deben estar de acuerdo. (Dinámico/ los usuarios se hacen participantes activos en el desarrollo)
* *Modelo de casos de uso.*Describe el comportamiento que un usuario realiza dentro del software, participan actores, casos de uso y el sistema. (Silva Ascuntar & García G., 2018)

**4.3.2.2) Revisión del diseño preliminar / análisis y diseño preliminar.** En esta fase a partir de cada caso de uso se lograrán unas fichas de casos de uso, (esto no pertenece a UML), está formada por un nombre, una descripción, una precondición que debe cumplir antes de iniciarse, una pos condición que debe cumplir al terminar si termina correctamente.

***Para esta fase se utiliza 2 herramientas.***

* *Diagrama de robustez.* Es un híbrido entre un diagrama de actividades y un diagrama de clases. Es un instrumento que permite capturar el que hacer y él como hacerlo, facilita el reconocimiento de objetos y hace más fácil la lectura del sistema. También ayuda a identificar los objetos que participan en cada caso de uso.

*El diagrama de robustez se divide.*

* + *Objetos fronterizos.* Usado por los actores para comunicarse directamente con el sistema.
  + *Objetos entidad.* Son objetos del modelo del dominio.
  + *Objetos de Control****.*** Es la unión entre la interfaz y los objetos de entidad.(Silva Ascuntar & García G., 2018)
* *Diagrama de clases.* Describe la estructura de un sistema mostrando todas sus clases, atributos y las relaciones que existen entre ellos.

4.3.3) Revisión crítica del diseño. En esta fase se examinan todos los elementos que forman parte del sistema.

*Diagramas de secuencia.*Muestra los métodos que llevaran las clases del sistema ymuestra todos los cursos alternos de los casos de uso.

4.3.4) Implementacion. En esta fase a partir de un buen diseño se generan pruebas suficientes antes de la implementacion.

***Se debe tener en cuenta factores.***

* *Reusabilidad.*Es la facilidad de hacer uso de los componentes en diferentes software.
* *Extensibilidad.*Consiste en poder editar con facilidad el software.
* *Confiabilidad.*Realización de software descartandoerrores.
* *Realizar pruebas.* Test de unidades, de casos, datos y resultados. Test de integración con los usuarios para verificar la aceptación de los resultados. (Silva Ascuntar & García G., 2018)

#### 

Figura1. Ubicación del modelado de dominio en el proceso de ICONIX[[1]](#footnote-1).

## 4.4) 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.

Para la realización e implementación de este proyecto se tomará de base el estándar ISO/IEC 29110, bajo el título general Ingeniería de Software – Perfiles del ciclo de vida para entidades muy pequeñas (Very Small Enterprises (VSEs)).

# Desarrollo Metodología ICONIX

De lo expuesto en el marco teórico se asumió los siguientes elementos de Iconix para el desarrollo del módulo, utilizando como proceso de gestión la norma ISO 29110 a cargo de (Grajales Gomez & Balanta Pazu, 2019).

## 5.1) Fase 1 Análisis de requerimientos

## 5.1.1) Requerimientos. Se levantaron los requerimientos con el equipo de calidad, Romero Dorado & Mina Posu, (2020) a través de reuniones y entrevistas con el cliente, a continuación, se muestran los resultados los cuales se fueron actualizando a petición del cliente.

Tabla 1. Requerimientos.

|  |  |
| --- | --- |
| # | Transferencia y comunicación |
| 1 | El sistema debe permitir el enlace y transferencia de datos de los dataloger mediante Internet. |
| 2 | El sistema debe permitir crear, modificar, borrar, visualizar comandos para ser utilizados por los protocolos de comunicación X y Y. |
| 3 | El sistema debe permitir elegir qué tipo de protocolo de comunicación usará una estación, pero en una organización puede haber múltiples estaciones con diferentes protocolos y a su vez, que usen diferentes comandos. |
| 4 | El sistema debe permitir establecer los parámetros necesarios para garantizar la comunicación vía FTP como cliente o servidor. |
| 5 | Debe permitir crear plantillas para leer los archivos planos y cargar los datos al sistema. |
| 6 | El sistema debe permitir usar las plantillas de textos planos de FTP para otras estaciones. |
| 7 | El sistema debe permitir acceder al FTP de acuerdo a la base de tiempo de las estaciones, leer los archivos de texto y almacenar la información en la base de datos. |
| 8 | El sistema debe permitir establecer los parámetros necesarios para garantizar la comunicación vía Modbus como cliente o servidor. |
| 9 | El sistema debe permitir configurar los comandos de módbus para acceder a los datalogers y obtener la información de estos. |
| 10 | El sistema debe permitir acceder a las estaciones mediante Modbus de acuerdo a la base de tiempo creado y almacenar la información en la base de datos. |
| 11 | El sistema debe permitir usar las plantillas comandos de Modbus para otras estaciones. |
| 12 | El sistema debe permitir subir textos planos de manera manual cuando una estación no tenga acceso a Internet |

## 5.1.2) Casos de uso:

Tabla 2. Caso de uso No.1.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Conectar con estación FTP | | | | | | | CU1 |
| Actores | | | Administrador, Operador | | | | | | | |
| Referencias | | | Requerimiento RF 1 | | | | |  | | |
| Precondición | | | El sistema deberá estar conectado a la red de internet, la estación ya debe estar creada con anterioridad | | | | | | | |
| Postcondición | | | El sistema debe mostrar los datos básicos de la estación a la que se ha conectado | | | | | | | |
| Autor | | | Fabián Mina | Fecha | |  | | | Versión |  |
| Aprobación | | | Aprobado | | | | | | | |
| Propósito | | | | | | | | | | |
| Conectarse mediante protocolo FTP a una estación | | | | | | | | | | |
| Curso Normal | | | | | | | | | | |
| 1 | El usuario selecciona el nombre de la estación. | | | | **2** | | El sistema muestra la información almacenada para ese nombre de estación.  -Id  -Serial  -Abreviación nombre corto de la -estación  -organización  -categoría  -región.  -Localización  -ciudad  -latitud y longitud  -elevación  -Base de tiempo | | | |
| 3 | El usuario confirma la información botón conectar | | | | **4** | | El sistema se conecta a la estación seleccionada. | | | |
|  |  | | | | **5** | | El sistema se conecta a la estación seleccionada y muestra mensaje en pantalla “Conexión Exitosa” | | | |
| Cursos Alternos | | | | | | | | | | |
| 1 | | En caso de que los datos no sean válidos o se presente algún error al momento de llenar el formulario, el sistema debe mostrar mensaje en pantalla, especificando el error que se presentó | | | | | | | | |
| 2 | | Si no existe protocolo de comunicación con esta estación, se mostrará mensaje en pantalla “No existe protocolo de comunicación”. | | | | | | | | |

Tabla 3. Caso de uso No.2.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Conectar con estación Modbus | | | | | | | CU2 |
| Actores | | | Administrador, Operador | | | | | | | |
| Referencias | | | Requerimiento RF 1 | | | | |  | | |
| Precondición | | | El sistema deberá estar conectado a la red de internet, la estación ya debe estar creada con anterioridad | | | | | | | |
| Postcondición | | | El sistema debe mostrar los datos básicos de la estación a la que se ha conectado | | | | | | | |
| Autor | | | Fabián Mina | Fecha | |  | | | Versión |  |
| Aprobación | | | Aprobado | | | | | | | |
| Propósito | | | | | | | | | | |
| Conectarse mediante protocolo Modbus a una estación | | | | | | | | | | |
| Curso Normal | | | | | | | | | | |
| 1 | El usuario selecciona el nombre de la estación. | | | | **2** | | El sistema muestra la información almacenada para ese nombre de estación.  -Id  -Serial  -Abreviación nombre corto de la -estación  -organización  -categoría  -región.  -Localización  -ciudad  -latitud y longitud  -elevación  -Base de tiempo | | | |
| 3 | El usuario confirma la información botón conectar | | | | **4** | | El sistema se conecta a la estación seleccionada. | | | |
|  |  | | | | **5** | | El sistema se conecta a la estación seleccionada y muestra mensaje en pantalla “Conexión Exitosa” | | | |
| Cursos Alternos | | | | | | | | | | |
| 1 | | En caso de que los datos no sean válidos o se presente algún error al momento de llenar el formulario, el sistema debe mostrar mensaje en pantalla, especificando el error que se presentó | | | | | | | | |
| 2 | | Si no existe protocolo de comunicación con esta estación, se mostrará mensaje en pantalla “No existe protocolo de comunicación”. | | | | | | | | |
| 3 | | Los campos latitud, longitud y elevación permiten valores numéricos (enteros, decimales, positivos y negativos) | | | | | | | | |

Tabla 4. Caso de uso No.4.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | | Crear región (Tabla parámetro) | | | | | | | CU4 |
| Actores | | | | Administrador, Operador | | | | | | | |
| Referencias | | | | Requerimiento RF 3 | | | | |  | | |
| Precondición | | | | El sistema deberá estar abierto en el menú de parametrización | | | | | | | |
| Postcondición | | | | La región queda creada en el sistema. | | | | | | | |
| Autor | | | | Fabián Mina | Fecha | |  | | | Versión |  |
| Aprobación | | | | Aprobado | | | | | | | |
| Propósito  Gestionar las regiones | | | | | | | | | | | |
| Curso Normal escenario 1.  Creación de la región | | | | | | | | | | | |
| 1 | El usuario oprime el botón nueva región | | | | | **2** | | El sistema carga el Id automático que corresponde a la nueva región y habilita los campos de la región. | | | |
| 3 | El usuario digita el nombre de la región con una longitud máxima de 50 caracteres. | | | | |  | |  | | | |
| 4 | El usuario digita una descripción de la región con una longitud máxima de 50 caracteres. | | | | |  | |  | | | |
| 5 | El usuario oprime el botón guardar | | | | | **6** | | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación de transacción: “Región creada correctamente” | | | |
|  |  | | | | | **7** | | Muestra en el listado la nueva región creada. | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | |
| 2 | | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | |
| 3 | | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | | |
| 4 | | | Los campos latitud, longitud y elevación permiten valores numéricos (enteros, decimales, positivos y negativos) | | | | | | | | |
| Curso Normal escenario 2.  Actualización de la región | | | | | | | | | | | |
| 1 | El usuario digita el nombre de la región o el Id de la región. | | | | | **2** | | El sistema carga los datos de la región. | | | |
| 3 | El usuario oprime el botón editar. | | | | | **4** | | El usuario habilita los campos nombre y descripción para actualización | | | |
| 4 | El usuario actualiza el campo nombre y descripción y oprime botón guardar | | | | | **6** | | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación “Guardado exitosamente”. | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje y el cursor se ubica en el campo. | | | | | | | | | |

Tabla 5. Caso de uso No.5.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear estación (Tabla parámetro) | | | | | | | CU5 |
| Actores | | | Administrador, Operador | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | |  | | |
| Precondición | | | El sistema deberá estar abierto en el menú de parametrización, los datos GMT deberán estar cargados en base de datos. | | | | | | | |
| Postcondición | | | La estación queda creada en el sistema. | | | | | | | |
| Autor | | | Fabián Mina | | Fecha | |  | | Versión |  |
| Aprobación | | | | Aprobado | | | | | | |
| Propósito Gestionar las estaciones | | | | | | | | | | |
| Curso Normal escenario 1  Creación de la estación | | | | | | | | | | |
| 1 | El usuario oprime el botón nueva estación. | | | | **2** | El sistema carga el Id automático que corresponderá a la nueva estación y habilita los campos de la estación. | | | | |
| 3 | El usuario digita el nombre de la estación. | | | |  |  | | | | |
| 4 | El usuario digita serial de la estación | | | |  |  | | | | |
| 5 | El usuario digita un nombre corto para la estación | | | |  |  | | | | |
| 6 | El usuario selecciona GMT de la estación (Zona horaria) | | | | **7** | El sistema despliega la lista de GMT | | | | |
| 10 | El usuario selecciona categoría | | | | **11** | El sistema carga la lista de categorías almacenadas | | | | |
| 12 | El usuario selecciona región | | | | **13** | El sistema carga la lista de regiones almacenadas | | | | |
| 14 | El usuario selecciona protocolo de comunicación (TCP o modbus) | | | |  |  | | | | |
| 15 | El usuario selecciona el campo localización | | | | **16** | El sistema carga la lista de localizaciones almacenadas | | | | |
| 17 | El usuario Selecciona el campo ciudad | | | | **18** | El sistema carga la lista de ciudades almacenadas | | | | |
| 19 | El usuario diligencia los campos latitud y longitud.  El usuario digita estos campos en grados decimales | | | |  |  | | | | |
| 20 | El usuario diligencia el campo elevación  El usuario digita estos campos en grados decimales. | | | |  |  | | | | |
| 21 | El usuario ingresa un número entero entre 0 y 120 para la base de tiempo.  (base de tiempo entre 0 y 2 horas) | | | | **22** | El sistema valida que una estación no tenga asignada la misma base de tiempo. | | | | |
| 23 | El usuario digita una descripción de la estación. | | | |  |  | | | | |
| 24 | El usuario oprime el botón guardar | | | | **25** | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación.  “Estación creada correctamente” | | | | |
|  |  | | | | **26** | El sistema muestra en el listado la nueva estación creada. | | | | |
| Cursos Alternos | | | | | | | | | | |
| 1 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 2 | El usuario no selecciona ningún valor de la lista. El sistema por defecto debe colocar el valor “Sin definir” y se debe tomar como un valor vacío al intentar guardar. | | | | | | | | | |
| 3 | El usuario digita valores alfanuméricos Se muestra el mensaje “Los valores deben ser decimales” | | | | | | | | | |
| 4 | La estación ya existe con la misma base de tiempo muestra mensaje “Ya existe esta base de tiempo en la estación” | | | | | | | | | |
| 5 | Hay campos vacíos, el sistema muestra el mensaje “Hay campos vacíos”, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 6 | Los campos latitud, longitud y elevación permiten valores numéricos (enteros, decimales, positivos y negativos) | | | | | | | | | |
| Curso Normal escenario 2  Actualización de la estación | | | | | | | | | | |
| 1 | El usuario digita el nombre de la estación o el Id de la región. | | | | **2** | El sistema carga los datos de la estación. | | | | |
| 3 | El usuario oprime el botón editar. | | | | **4** | El sistema habilita los campos nombre, serial, nombre corto, GMT, descripción, región.Localización, Ciudad, latitud y longitud, elevación para actualización | | | | |
| 5 | El usuario actualiza los campos nombre, serial, nombre corto, GMT, descripción, región.  Localización  Ciudad, latitud y longitud y oprime botón guardar | | | | **6** | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación.  “Guardado con Satisfactoriamente” | | | | |
| Cursos Alternos | | | | | | | | | | |
| 1 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | |
| 2 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | | |
| 4 | | Los campos latitud, longitud y elevación permiten valores numéricos (enteros, decimales, positivos y negativos) | | | | | | | | |

*Tabla 6.* Caso de uso No.6.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear organización (Tabla parámetro) | | | | | | | | CU6 |
| Actores | | | Administrador | | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | | |  | | |
| Precondición | | | El sistema deberá estar abierto en el menú de parametrización | | | | | | | | |
| Postcondición | | | La organización quedara creada en el sistema. | | | | | | | | |
| Autor | | | Fabián Mina | | Fecha |  | | | | Versión |  |
| Aprobación | | | | Aprobado | | | | | | | |
| Propósito Gestionar las organizaciones del sistema | | | | | | | | | | | |
| Curso Normal escenario 1  Creación de la organización. | | | | | | | | | | | |
| 1 | El usuario oprime el botón nueva organización | | | | | | **2** | El sistema carga el Id automático que corresponde a la nueva organización y habilita los campos de la organización. | | | |
| 3 | El usuario digita el nombre de la organización. Longitud máxima de 50 caracteres. | | | | | |  |  | | | |
| 4 | El usuario digita el Email. Longitud máxima de 50 caracteres. | | | | | |  |  | | | |
| 5 | El usuario digita número de contacto. Longitud máxima de 50 caracteres. | | | | | |  |  | | | |
| 6 | El usuario digita una descripción para la organización. Longitud máxima de 50 caracteres. | | | | | |  |  | | | |
| 7 | El usuario oprime el botón guardar | | | | | | **8** | El sistema valida que todos los campos estén llenos y guarda. Muestra mensaje de confirmación.  “Organización creada correctamente” | | | |
|  |  | | | | | | **9** | El sistema muestra en el listado la nueva organización creada. | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | | |
| 2 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | | |
| Curso Normal escenario 2  Actualización de la organización | | | | | | | | | | | |
| 1 | El usuario digita el nombre de la organización o el Id de la organización. | | | | | | **2** | El sistema carga los datos de la organización. | | | |
| 3 | El usuario oprime el botón editar. | | | | | | **4** | El sistema habilita los campos nombre de la organización, email, numero contacto y descripción para actualización. | | | |
| 5 | El usuario actualiza el campo nombre de la organización, email, numero de contacto, descripción y oprime botón guardar | | | | | | **6** | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación: “Organización actualizada correctamente” | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales. | | | | | | | | | |

*Tabla 7.* Caso de uso No.7*.*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear categoría (Tabla parámetro) | | | | | | | | CU7 |
| Actores | | | Administrador | | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | | |  | | |
| Precondición | | | El sistema deberá estar abierto en el menú de parametrización | | | | | | | | |
| Postcondición | | | La categoría quedara creada en el sistema. | | | | | | | | |
| Autor | | | Fabián Mina | | Fecha | |  | | | Versión |  |
| Aprobación | | | | Aprobado | | | | | | | |
| Propósito Gestionar las categorías del sistema | | | | | | | | | | | |
| Curso Normal escenario 1  Creación de la categoría. | | | | | | | | | | | |
| 1 | El usuario oprime el botón nueva categoría | | | | | **2** | | El sistema carga el Id automático que corresponde a la nueva categoría y habilita los campos de la categoría. | | | |
| 3 | El usuario digita el nombre de la categoría. | | | | |  | |  | | | |
| 4 | El usuario digita una descripción para la categoría. | | | | |  | |  | | | |
| 5 | El usuario oprime el botón guardar | | | | | **6** | | El sistema valida que todos los campos estén llenos y guarda. Muestra mensaje de confirmación de transacción  “Categoría creada correctamente” | | | |
|  |  | | | | | **7** | | El sistema muestra en el listado la nueva categoría creada. | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| Curso Normal escenario 2  Actualización de la categoría | | | | | | | | | | | |
| 1 | El usuario digita el nombre de la categoría o el Id de la organización. | | | | | **2** | | El sistema carga los datos de la categoría. | | | |
| 3 | El usuario oprime el botón editar. | | | | | **4** | | El sistema habilita los campos nombre de la categoría y descripción para actualización. | | | |
| 5 | El usuario actualiza el campo nombre de la categoría, descripción y oprime botón guardar | | | | | **6** | | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | | | |

*Tabla 8.* Caso de uso No.8.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear Ciudad (Tabla parámetro) | | | | | | | | CU8 |
| Actores | | | Administrador | | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | | |  | | |
| Precondición | | | El sistema deberá estar abierto en el menú de parametrización, por cada región, las ciudades deberán corresponder. Se mostrará combo box para seleccionar región a la que pertenece la ciudad. | | | | | | | | |
| Postcondición | | | La ciudad quedara creada en el sistema. | | | | | | | | |
| Autor | | | Daniela Romero | | Fecha | | |  | | Versión |  |
| Aprobación | | | | Aprobado | | | | | | | |
| Propósito  Gestionar las ciudades del sistema | | | | | | | | | | | |
| Curso Normal escenario 1  Creación de la Ciudad. | | | | | | | | | | | |
| 1 | El usuario oprime el botón nueva Ciudad | | | | | **2** | El sistema carga el Id automático que corresponde a la nueva Ciudad y habilita los campos de la Ciudad. | | | | |
| 3 | El usuario digita el nombre de la Ciudad. | | | | |  |  | | | | |
| 4 | El usuario selecciona la región a la que pertenece la ciudad | | | | | **5** | El sistema despliega la lista de regiones almacenadas en base de datos. | | | | |
| 6 | El usuario digita una descripción para la Ciudad. | | | | |  |  | | | | |
| 7 | El usuario oprime el botón guardar | | | | | **8** | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación. | | | | |
|  |  | | | | | **9** | El sistema muestra en el listado la nueva ciudad creada. | | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| Curso Normal escenario 2  Actualización de la Ciudad. | | | | | | | | | | | |
| 1 | El usuario digita el nombre de la ciudad o el Id de la ciudad. | | | | | **2** | El sistema carga los datos de la ciudad. | | | | |
| 3 | El usuario oprime el botón editar. | | | | | **4** | El sistema habilita los campos nombre de la ciudad, región y descripción para actualización. | | | | |
| 5 | El usuario actualiza el campo nombre de la Ciudad, región, descripción y oprime botón guardar | | | | | **6** | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. “Ciudad actualizada correctamente” | | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | | | |

*Tabla 9.* Caso de uso No.9.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear rango (Tabla parámetro) | | | | | | | | CU9 |
| Actores | | | Administrador | | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | | |  | | |
| Precondición | | | El sistema deberá estar abierto en el menú de parametrización, por cada estación, los rangos deberán corresponder. Se mostrará combo box para seleccionar estación a la que pertenece el rango. | | | | | | | | |
| Postcondición | | | Los rangos quedaran creados en el sistema. | | | | | | | | |
| Autor | | | Daniela Romero | | Fecha | |  | | | Versión |  |
| Aprobación | | | | Aprobado | | | | | | | |
| Propósito | | | | | | | | | | | |
| Gestionar los rangos del sistema, los cuales definen el porcentaje de variación al gestionar la información. | | | | | | | | | | | |
| Curso Normal escenario 1  Creación de rango. | | | | | | | | | | | |
| 1 | El usuario oprime el botón nuevo Rango | | | | | **2** | | El sistema carga el Id automático que corresponde al nuevo rango y habilita los campos. Rango mínimo, rango máximo, rango, descripción. | | | |
| 3 | El usuario digita un nombre para el rango. Longitud máxima de 50 caracteres. | | | | |  | |  | | | |
| 4 | El usuario digita valor mínimo del nuevo rango. (Valor numérico) valor mínimo 0 | | | | |  | |  | | | |
| 5 | El usuario digita valor máximo del nuevo rango. (Valor numérico) | | | | |  | |  | | | |
| 6 | El usuario selecciona la estación a la que pertenece el rango | | | | | **7** | | El sistema despliega la lista de estaciones almacenadas en base de datos. | | | |
| 8 | El usuario digita una descripción para el rango. Longitud máxima de 50 caracteres. | | | | |  | |  | | | |
| 9 | El usuario oprime el botón guardar | | | | | **10** | | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación de transacción “Guardado exitosamente” | | | |
|  |  | | | | | **11** | | El rango guardado se muestra en la cuadrícula donde están listados todos los rangos | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 2 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | | | |
| Curso Normal escenario 2  Actualización del rango | | | | | | | | | | | |
| 1 | El usuario digita el nombre del rango o el Id. | | | | | **2** | | El sistema carga los datos del rango. | | | |
| 3 | El usuario oprime el botón editar. | | | | | **4** | | El sistema habilita los campos nombre del rango, rango, región y descripción para actualización. | | | |
| 5 | El usuario actualiza el campo nombre del rango, rango, región, descripción y oprime botón guardar  Longitud máxima de 50 caracteres. | | | | | **6** | | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. “Rango actualizado correctamente” | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | | | |

*Tabla 10.* Caso de uso No.10.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear alerta (Tabla parámetro) | | | | | | | | CU10 |
| Actores | | | Administrador | | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | | |  | | |
| Precondición | | | El sistema deberá estar abierto en el menú de parametrización, las alertas se crearán igual para todas las regiones. | | | | | | | | |
| Postcondición | | | Las alertas quedaran creadas en el sistema, enviaran notificación automáticamente. | | | | | | | | |
| Autor | | | Daniela Romero | | | Fecha | |  | | Versión |  |
| Aprobación | | | | Aprobado | | | | | | | |
| Propósito | | | | | | | | | | | |
| Gestionar las alertas del sistema | | | | | | | | | | | |
| Curso Normal escenario 1  Creación alerta. | | | | | | | | | | | |
| 1 | El usuario oprime el botón nuevo alerta. | | | | **2** | | El sistema carga el Id automático que corresponde a la nueva alerta y habilita los campos. | | | | |
| 3 | El usuario digita un nombre para la alerta. Longitud máxima de 50 caracteres. | | | |  | |  | | | | |
| 4 | El usuario digita una descripción para la alerta creada. Longitud máxima de 50 caracteres. | | | |  | |  | | | | |
| 5 | El usuario selecciona tipo de alerta. (Fax - Email – SMS) | | | |  | |  | | | | |
| 6 | El usuario digita Email | | | |  | |  | | | | |
| 7 | El usuario digita contraseña | | | |  | |  | | | | |
| 8 | El usuario digita puerto SMTP | | | |  | |  | | | | |
| 9 | El usuario digita servidor SMTP | | | |  | |  | | | | |
| 10 | El usuario selecciona tipo de seguridad (SSL, TLS, SSL/TLS) | | | |  | |  | | | | |
| 11 | El usuario oprime el botón guardar | | | | **12** | | El sistema valida que todos los campos estén llenos y guarda. Muestra mensaje de confirmación de transacción. “Alerta guardada con éxito” | | | | |
|  |  | | | | **13** | | Se muestra en la cuadrícula donde están listados todas las alertas creadas. | | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| Curso Normal escenario 2  Actualización de alerta | | | | | | | | | | | |
| 1 | El usuario digita el nombre o el Id de la alerta | | | | **2** | | El sistema carga los datos. | | | | |
| 3 | El usuario oprime el botón editar. | | | | **4** | | El sistema habilita los campos nombre de alerta, tipo y descripción para actualización. | | | | |
| 5 | El usuario actualiza el campo nombre de alerta, estación, tipo y oprime botón guardar | | | | **6** | | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. | | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 2 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales. | | | | | | | | | |

*Tabla 11.* Caso de uso No.11.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear base de tiempo (Tabla parámetro) | | | | | | | | CU11 |
| Actores | | | Administrador | | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | | |  | | |
| Precondición | | | El sistema deberá estar abierto en el menú de parametrización. | | | | | | | | |
| Postcondición | | | Los tiempos quedaran creados en el sistema | | | | | | | | |
| Autor | | | Daniela Romero | | Fecha | |  | | | Versión |  |
| Aprobación | | | | Aprobado | | | | | | | |
| Propósito | | | | | | | | | | | |
| Gestionar los tiempos para calibrar los datos del sistema, en el cuadro de tiempo selecciona el rango de tiempo para el cual se van a consultar los registros gestionados en el sistema. | | | | | | | | | | | |
| Curso Normal escenario 1  Creación tiempos. | | | | | | | | | | | |
| 1 | El usuario oprime el botón nuevo tiempo | | | | | **2** | | El sistema carga el Id automático que corresponde a los tiempos y habilita los campos. | | | |
| 3 | El usuario digita un nombre para los tiempos.  Longitud máxima de 50 caracteres. | | | | |  | |  | | | |
| 4 | El usuario digita una escala de tiempos a partir de 1 minuto.  La escala de tiempo ira de 1 y finaliza en 240 (ej.1 2 5 6 10 15 30 60 120 125 130 150 170 190 200 210 215 220 225 230 240). | | | | | **5** | | El sistema valida que el tiempo se configure en formato minutos. | | | |
|  |  | | | | | **6** | | El sistema validad que el rango de tiempo no supere los 240 minutos. | | | |
| 7 | El usuario digita una descripción para los tiempos registrados  Longitud máxima de 50 caracteres. | | | | |  | |  | | | |
| 8 | El usuario selecciona la alerta deseada, donde recibirá notificación | | | | | **9** | | El sistema despliega las alertas creadas y almacenadas en base de datos.  Fax  E-mail  Mensaje  Ventana Emergente | | | |
| 10 | El usuario oprime el botón guardar | | | | | **11** | | El sistema valida la información guardad, muestra mensaje en pantalla guardado con éxito. | | | |
|  |  | | | | | **12** | | Se muestra en la cuadrícula donde están listados todos los tiempos creadas. | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | En caso de presentarse un error el sistema mostrara mensaje en pantalla indicando el error. | | | | | | | | | |
| 2 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 3 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| Curso Normal escenario 2  Actualización de tiempos | | | | | | | | | | | |
| 1 | El usuario digita el nombre o el Id | | | | | **2** | | El sistema carga los datos. | | | |
| 3 | El usuario oprime el botón editar. | | | | | **4** | | El sistema habilita los campos nombre y rango de tiempos. | | | |
| 5 | El usuario actualiza el campo nombre, rango de tiempos y oprime botón guardar.  Longitud máxima de 50 caracteres. | | | | | **6** | | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales. | | | | | | | | | |

## 

*Tabla 12.* Caso de uso No.15.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Conectar Automáticamente con estación | | | | | | | CU15 |
| Actores | | | Administrador, Operador | | | | | | | |
| Referencias | | | Requerimiento RF 1 | | | | |  | | |
| Precondición | | | -El sistema deberá estar conectado a la red de internet  -la estación ya debe estar creada con anterioridad,  -La estación deberá tener configurado el protocolo de comunicación con el cual se hará la conexión. | | | | | | | |
| Postcondición | | | El sistema debe mostrar los datos básicos de la estación a la que se ha conectado | | | | | | | |
| Autor | | | Fabián Mina | Fecha | |  | | | Versión |  |
| Aprobación | | | Aprobado | | | | | | | |
| Propósito | | | | | | | | | | |
| Conectarse Automáticamente a una mediante protocolo FTP o Modbus | | | | | | | | | | |
| Curso Normal | | | | | | | | | | |
| 1 | El usuario selecciona el nombre de la estación. | | | | **2** | | El sistema muestra la información almacenada para ese nombre de estación: -Id -Serial -Abreviación nombre corto de la -estación -organización  -categoría -región -Localización -ciudad -latitud y longitud -elevación -Base de tiempo | | | |
|  |  | | | | **3** | | El sistema verifica el protocolo de comunicación configurado a la estación | | | |
|  |  | | | | **4** | | El sistema se conecta a la estación seleccionada. | | | |
|  |  | | | | **5** | | El sistema se conecta a la estación seleccionada y muestra mensaje en pantalla “Conexión Exitosa con la estación” | | | |
| Cursos Alternos | | | | | | | | | | |
| 2 | | En caso de que los datos no sean válidos o se presente algún error al momento de llenar el formulario, el sistema debe mostrar mensaje en pantalla, especificando el error que se presentó | | | | | | | | |
| 3 | | Si no existe protocolo de comunicación con esta estación, se mostrará mensaje en pantalla “No existe protocolo de comunicación”. | | | | | | | | |

*Tabla 13.* Caso de uso No.16.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear Mantenimiento | | | | | | | | CU16 |
| Actores | | | Administrador, Operador | | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | | |  | | |
| Precondición | | | El registro de mantenimiento no podrá ser modificado | | | | | | | | |
| Postcondición | | | El registro de mantenimiento creado quedara almacenado en base de datos con la respectiva hora y usuario de creación. | | | | | | | | |
| Autor | | | Daniela Romero | | Fecha | |  | | | Versión |  |
| Aprobación | | | | Aprobado | | | | | | | |
| Propósito | | | | | | | | | | | |
| Gestionar el registro de actividad | | | | | | | | | | | |
| Curso Normal escenario 1  Crear mantenimiento. | | | | | | | | | | | |
| 1 | El usuario oprime el botón nuevo registro | | | | | **2** | | El sistema carga el Id automático que corresponde al nuevo registro. | | | |
| 3 | El usuario selecciona el nombre de la estación, en el campo registrar estación. | | | | |  | |  | | | |
| 4 | El usuario digita el nombre del funcionario, en el campo registrar funcionario. | | | | |  | |  | | | |
| 5 | El usuario selecciona partes de estación | | | | |  | |  | | | |
| 6 | El usuario selecciona tipo de mantenimiento | | | | | **7** | | El sistema despliega lista de tipo de mantenimiento almacenadas | | | |
| 8 | El usuario ingresa fecha inicial y fecha final | | | | |  | |  | | | |
| 9 | El usuario ingresa periodicidad | | | | | **10** | | El sistema despliega lista de periodicidad almacenada | | | |
| 11 | El usuario digita una descripción o novedad, en el campo novedad | | | | |  | |  | | | |
| 12 | El usuario oprime el botón guardar | | | | | **13** | | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación “Registro guardado con éxito”. | | | |
|  |  | | | | | **14** | | El sistema carga la hora y fecha de creación automáticamente | | | |
|  |  | | | | | **15** | | El nuevo registro se almacena en base de datos, y se muestra listado de registros en grilla. | | | |
| Cursos Alternos | | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | | | |

*Tabla 14.* Caso de uso No.25.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear tipo Mantenimiento (Tabla parámetro) | | | | | | CU25 |
| Actores | | | Administrador, Operador | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | | | El registro de mantenimiento no podrá ser modificado | | | | | | |
| Postcondición | | | El registro de tipo de mantenimiento creado quedara almacenado en base de datos con la respectiva hora y usuario de creación. | | | | | | |
| Autor | | | Daniela Romero | Fecha | | 6/02/20 | | Versión | V 6.0 |
| Aprobación | | | Aprobado | | | | | | |
| Propósito | | | | | | | | | |
| Gestionar el registro de actividad | | | | | | | | | |
| Curso Normal escenario 1  Crear mantenimiento. | | | | | | | | | |
| 1 | El usuario oprime el botón nuevo registro | | | **2** | El sistema carga el Id automático que corresponde al nuevo tipo de mantenimiento. | | | | |
| 3 | El usuario digita el nombre del nuevo tipo de mantenimiento | | |  |  | | | | |
| 4 | El usuario digita un detalle del tipo de mantenimiento en el campo de observación. | | |  |  | | | | |
| 5 | El usuario oprime el botón guardar | | | **6** | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación “Registro guardado con éxito”. | | | | |
|  |  | | | **7** | El nuevo registro se almacena en base de datos, y se muestra listado de registros en grilla. | | | | |
| Cursos Alternos | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | |

*Tabla 15.* Caso de uso No.26.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear periodicidad (Tabla parámetro) | | | | | | | CU26 |
| Actores | | | Administrador, Operador | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | |  | | |
| Precondición | | | El registro de mantenimiento no podrá ser modificado | | | | | | | |
| Postcondición | | | El registro de tipo de periodicidad creado quedara almacenado en base de datos con la respectiva hora y usuario de creación. | | | | | | | |
| Autor | | | Fabian Mina | Fecha | | | 6/02/20 | | Versión | V 6.0 |
| Aprobación | | | Aprobado | | | | | | | |
| Propósito | | | | | | | | | | |
| Gestionar el registro de actividad | | | | | | | | | | |
| Curso Normal escenario 1  Crear periodicidad. | | | | | | | | | | |
| 1 | El usuario oprime el botón nuevo registro | | | | **2** | El sistema carga el Id automático que corresponde al nuevo tipo de periodicidad. | | | | |
| 3 | El usuario digita el nombre del nuevo tipo de periodicidad | | | |  |  | | | | |
| 4 | El usuario digita un detalle del tipo de periodicidad en el campo de observación. | | | |  |  | | | | |
| 5 | El usuario oprime el botón guardar | | | | **6** | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación “Registro guardado con éxito”. | | | | |
|  |  | | | | **8** | El nuevo registro se almacena en base de datos, y se muestra listado de registros en grilla. | | | | |
| Cursos Alternos | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | | |

*Tabla 16.* Caso de uso No.27

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Partes de mantenimiento | | | | | | | CU27 |
| Actores | | | Administrador, Operador | | | | | | | |
| Referencias | | | Requerimiento RF 1 y 2 | | | | |  | | |
| Precondición | | | El registro de mantenimiento no podrá ser modificado | | | | | | | |
| Postcondición | | | El registro de tipo de periodicidad creado quedara almacenado en base de datos con la respectiva hora y usuario de creación. | | | | | | | |
| Autor | | | Fabian Mina | Fecha | | 6/02/20 | | | Versión | V 6.0 |
| Aprobación | | | Aprobado | | | | | | | |
| Propósito | | | | | | | | | | |
| Gestionar el registro de actividad | | | | | | | | | | |
| Curso Normal escenario 1  Crear periodicidad. | | | | | | | | | | |
| 1 | El usuario oprime el botón nueva parte | | | | **2** | | El sistema carga el Id automático que corresponde a la nueva parte a crear. | | | |
| 3 | El usuario digita el nombre de la nueva parte | | | |  | |  | | | |
| 4 | El usuario digita un código de la parte a crear. | | | |  | |  | | | |
| 5 | El usuario digita una descripción de la parte a crear. | | | |  | |  | | | |
| 6 | El usuario oprime el botón guardar. | | | | **7** | | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación “Registro guardado con éxito”. | | | |
|  |  | | | | **8** | | El nuevo registro se almacena en base de datos, y se muestra listado de registros en grilla. | | | |
| Cursos Alternos | | | | | | | | | | |
| 1 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | |
| 2 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | |
| 3 | | Los campos de texto permiten valores alfanuméricos y caracteres especiales | | | | | | | | |

## 5.1.3) Prototipos de interfaz.

Los prototipos para los casos de uso número 1, 2 y 15 tienen la misma estructura por ende se toma como ejemplo a continuación el caso de uso conectar por FTP.

#### 

*Figura 2.* Prototipo conectar estación FTP, ModBus y Automático.

Caso de uso Numero 4 (tabla parámetro)

#### 

*Figura 3.* Prototipo crear región crud.

Caso de uso Numero 5 (tabla parámetro)

#### 

*Figura 4.* Prototipo crear estación crud.

Caso de uso Numero 6 (tabla parámetro)

#### 

*Figura 5.* Prototipo crear organización crud.

Caso de uso Numero 8 (tabla parámetro)

#### 

*Figura 6.* Prototipo crear ciudad crud.

Caso de uso Numero 9 (tabla parámetro)

#### 

*Figura 7.* Prototipo crear rango crud.

Caso de uso Numero 10 (tabla parámetro)

#### 

*Figura 8.* Prototipo crear alerta crud.

Caso de uso Numero 11 (tabla parámetro)

#### 

*Figura 9.* Prototipo crear base de tiempo crud.

Caso de uso Numero 16

#### 

*Figura 10.* Prototipo crear mantenimiento crud.

## 

Caso de uso Numero 25

#### 

*Figura 11.* Prototipo crear tipo de mantenimiento crud.

Caso de uso Numero 26

#### 

*Figura 12.* Prototipo crear periodicidad de mantenimiento crud.

Caso de uso Numero 2.

#### 

*Figura 13.* Prototipo crear partes de mantenimiento crud.

## 5.2) Fase 2 Análisis y diseño preliminar

## 5.2.1) Diagrama de clases

Los diagramas de clases para los casos de uso número 1, 2 y 15 tienen la misma estructura por ende se toma como ejemplo a continuación el caso de uso conectar por FTP.

#### 

*Figura 14.* Diagrama de clases conectar con estación con FTP, ModBus y Automático.

Caso de uso #4

#### 

*Figura 15.* Diagrama de clases tabla parámetro Región.

Caso de uso #5

#### 

*Figura 16.* Diagrama de clases tabla parámetro Estación.

Caso de uso #6

#### 

*Figura 17.* Diagrama de clases tabla parámetro Organización.

Caso de uso #7

#### 

*Figura 18.* Diagrama de clases tabla parámetro Categoría.

Caso de uso #8

#### 

*Figura 19.* Diagrama de clases tabla parámetro Ciudad.

Caso de uso #9

#### 

*Figura 20.* Diagrama de clases tabla parámetro Rango.

Caso de uso #10

#### 

*Figura 21.* Diagrama de clases tabla parámetro Alerta.

Caso de uso #11

#### 

*Figura 22.* Diagrama de clases tabla parámetro Base de tiempo.

Caso de uso #16

#### 

*Figura 23.* Diagrama de clases Mantenimiento.

Caso de uso #25

#### 

*Figura 24.* Diagrama de clases de partes de Mantenimiento.

Caso de uso #26

#### 

*Figura 25.* Diagrama de clases de periodicidad de Mantenimiento.

Caso de uso #27

#### 

*Figura 26.* Diagrama de clases de tipo de Mantenimiento.

## 5.3) Fase 3 Diseño detallado

## 5.3.1) Diagramas de secuencia

Los diagramas de secuencia

para los casos de uso número 1, 2 y 15 tienen la misma estructura por ende se toma como ejemplo a continuación el caso de uso conectar por FTP.

#### 

*Figura 27.*Diagrama de secuencias conectar con estación FTP, ModBus y Automático.

Caso de uso #4

#### 

*Figura 28.* Diagrama de secuencias crear región.

#### 

*Figura 29.* Diagrama de secuencias actualizar región.

Caso de uso #5

#### 

*Figura 30.* Diagrama de secuencias crear estación.

#### 

*Figura 31.* Diagrama de secuencias actualizar estación.

Caso de uso #6

#### 

*Figura 32.* Diagrama de secuencias crear organización.

#### 

*Figura 33.* Diagrama de secuencias actualizar organización.

Caso de uso #7

#### 

*Figura 34.* Diagrama de secuencias crear categoría.

#### 

*Figura 35.* Diagrama de secuencias actualizar categoría.

Caso de uso #8

#### 

*Figura 36.* Diagrama de secuencias crear ciudad.

#### 

*Figura 37.* Diagrama de secuencias actualizar ciudad.

Caso de uso #9

#### 

*Figura 38.* Diagrama de secuencias crear rango.

#### 

*Figura 39.* Diagrama de secuencias actualizar rango.

Caso de uso #10

#### 

*Figura 40.* Diagrama de secuencias crear alerta.

#### 

*Figura 41.* Diagrama de secuencias actualizar alerta.

Caso de uso #11

#### 

*Figura 42.* Diagrama de secuencias crear cuadro de tiempo.

#### 

*Figura 43.* Diagrama de secuencias actualizar cuadro de tiempo.

Caso de uso #16

#### 

*Figura 44.* Diagrama de secuencias crear mantenimiento.

Caso de uso #25

#### 

*Figura 45.* Diagrama de secuencias crear tipos de mantenimientos.

#### 

*Figura 46.* Diagrama de secuencias actualizar tipos de mantenimientos.

Caso de uso #26

#### 

*Figura 47.* Diagrama de secuencias crear periodicidad de mantenimiento.

#### 

*Figura 48.* Diagrama de secuencias actualizar periodicidad de mantenimiento.

Caso de uso #27

#### 

*Figura 49.* Diagrama de clases de partes de mantenimiento.

#### 

*Figura 50.* Diagrama de clases de partes de mantenimiento.

## 5.4) Fase 4 Implementación

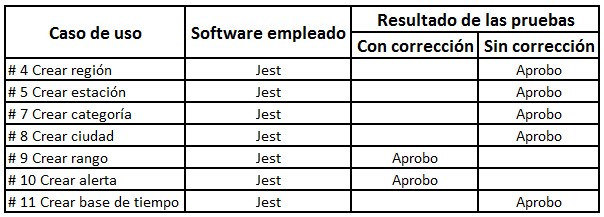
El proyecto a cargo de (Romero Dorado & Mina Posu, 2020) quienes conforman el equipo de calidad, se enfocó en ejecutar las pruebas con las herramientas necesarias para garantizar un adecuado testeo a todos los componentes y revisar que cada método cumpla con su función.

## 5.4.1) Pruebas unitarias. En el módulo de calidad se ejecutaron pruebas unitarias con la herramienta Jest, que es una plataforma de prueba universal, con la capacidad de adaptarse a cualquier biblioteca o marco de Java Script. Se realizó un test completo de cada método de guardar, consultar y modificar.

A continuación, se define el resumen de los resultados.

* ***Test suites.*** Define a cuantos componentes se les aplico el test de prueba
* ***Test.*** Define a cuantos métodos de les aplico el test de prueba unitaria
* ***Pass.*** Define que el módulo paso la prueba satisfactoriamente
* ***Passed.*** define si el método al cual se le aplico el test paso la prueba satisfactoriamente
* ***Fail.*** Define si el método al cual se le aplico el test no pasa la prueba
* ***Snapshots.*** Definen las capturas o estados especificado de nuestro aplicativo
* ***Time.*** Define el tiempo el cual el test realiza el recorrido por todo el método y sus componentes.

En la siguiente figura se relaciona el resultado de las pruebas unitarias aplicadas a los componentes de Región, Ciudad, Categoría, Base de tiempo, Rango, Alerta, Estación.



*Figura 51.* Resultados de las pruebas unitarias

## 5.4.2) Pruebas de carga. Se realizaron las pruebas de carga con la herramienta Jmeter, la cual es un software hecho en Java, que permite hacer pruebas de carga y rendimiento de diferentes tipos de servicios, para este proyecto se simularon 500 usuarios ingresando simultáneamente en un ciclo de 15 segundos.

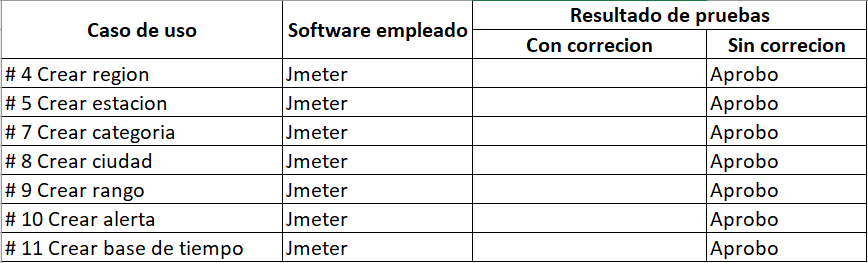
El protocolo HTTP genera códigos de estado de respuesta con los cuales se evalúa si fue exitosa o no exitosa la conexión, se configura la herramienta Jmeter con el puerto de servicio en el que se encuentra alojado el aplicativo puerto 4200, se apunta a la dirección en la cual se encuentra la vista.

A continuación, se definen los códigos de después de HTTP

* ***100 continue:*** Define que todo se encuentra correcto dentro de las peticiones al servidor
* ***200 ok.*** Define que la solicitud al servidor ha sido exitosa
* ***202 accepted.*** Define que la solicitud ha sido recibida
* ***400 badrequest.*** Define que el servidor no ha podido interpretar la solicitud
* ***403 forbidden.*** Define que el usuario no cuenta con los permisos para acceder
* ***404 no found.*** Define que el servidor no ha encontrado el contenido solicitado
* ***500 internal server error.*** Define que el servidor ha generado un error interno

Para la prueba que se realizó como código de respuesta de HTTP se obtuvo como resultado el código 200, para cada una de las consultas hechas en cada uno de los componentes. (Romero Dorado & Mina Posu, 2020)

En la siguiente figura se relaciona el resultado de las pruebas de carga aplicadas a los componentes de Región, Ciudad, Categoría, Base de tiempo, Rango, Alerta, Estación.



*Figura 52.* Resultado pruebas de carga

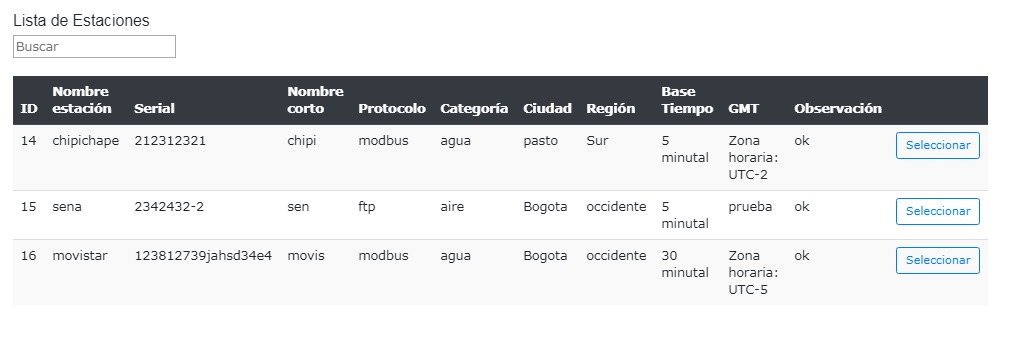
1. **Resultados**

Pantallazo de inicio

****

*Figura 53.* Resultados pantallazo de inicio.

Pantallazo de listar estaciones para hacer la conexión

****

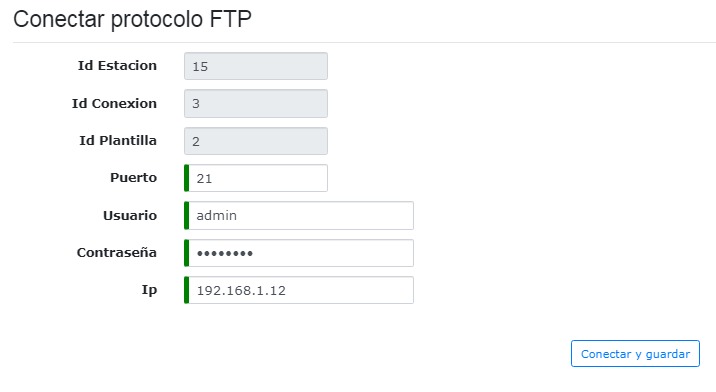
*Figura 54.* Resultados pantallazo listar estaciones

Pantallazo de los datos para hacer la conexión

****

*Figura 55.* Resultados pantallazo información para conectar

Pantallazo para conectar FTP

****

*Figura 56.* Resultados pantallazo conectar y transferir.

**7) Conclusiones**

Para finalizar se evidencia que el punto clave de este proyecto fue crear un módulo que permita establecer la conexión con el datalogger mediante los protocolos Modbus TCP/IP y FTP, posibilitando tener acceso a los datos de forma más eficiente. Debido a la posibilidad de emplear herramientas como las plantillas predefinidas que brinda este software; las cuales ayudan a que la información se almacene de forma ordenada y comprensible dentro de la base de datos, haciendo uso de las interfaces desarrolladas para la interacción del usuario con las estaciones y su respectiva configuración.

1. **Bibliografía**

Referencias

Angular. (2018). *Architecture Overview. Angular.io.* Obtenido de https://angular.io/guide/architecture

Azaustre, C. (09 de 09 de 2013). *carlosazaustre*. Obtenido de https://carlosazaustre.es/empezando-con-angular-js/

Berners-Lee, T. (s.f.). *Tutorial básico del programador web*. Obtenido de https://www.aprenderaprogramar.es/index.php?option=com\_content&view=article&id=435:ique-es-y-para-que-sirve-html-el-lenguaje-mas-importante-para-crear-paginas-webs-html-tags-cu00704b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192

Bolo, F. (02 de 08 de 2011). *La historia del protocolo FTP (y sus desventajas actuales)*. Obtenido de https://managefiletransfer.wordpress.com/2011/08/02/mft-b2bconsulting/

Caceres, M. (s.f.). *¿Qué es TypeScript?* Obtenido de https://devcode.la/blog/que-es-typescript/

Cea, O. (19 de 12 de 2018). *Inyección de Componentes y Directivas en Angular*. Obtenido de https://medium.com/angular-chile/inyecci%C3%B3n-de-componentes-y-directivas-en-angular-6ae75f64be66

Developer.mozilla.org. (8 de 11 de 2019). *Learn to style HTML using CSS*. Obtenido de https://developer.mozilla.org/en-US/docs/Learn/CSS

Drury, B. (2009). *Control Techniques Drives and Controls Handbook (2nd Edition)*. Obtenido de https://app.knovel.com/web/toc.v/cid:kpCTDCHE08/viewerType:toc/root\_slug:control-techniques-drives/url\_slug:control-techniques-drives/?

Finley, K. (01 de 25 de 2011). *Wait, What's Node.js Good for Again?* Obtenido de https://web.archive.org/web/20130621203930/http://readwrite.com/2011/01/25/wait-whats-nodejs-good-for-aga#awesm=~poWk5dk8ztKLtf

Flanagan, D. (2006). *JavaScript - The Definitive Guide.* O'Reilly, Sebastopol, CA.

Flanagan, D. (2011). *JavaScript: The Definitive Guide, 6th Edition.* O'Reilly Media.

Flower, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R., & Stafford , R. (2002). *Patterns of Enterprise Application Architecture, Addison-Wesley Professional .*

github. (s.f.). *Cómo trabajan los desarrolladores*. Obtenido de https://github.com/features

Grajales Gomez, A. L., & Balanta Pazu, C. F. (2019). *Identificación e implementación de un marco de trabajo orientado a procesos DEVOPS siguiendo el estándar ISO/IEC 29110 para el caso de estudio de integración de procesos del proyecto SANAMBIENTE.* Cali, Institucion universitaria Antonio Jose Camacho.

Haro, J. M., & Camps Riba , J. M. (14 de 01 de 2008). *http://openaccess.uoc.edu/*. Obtenido de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/876/1/00765tfc.pdf

IZQUIERDO, J. (04 de 09 de 2014). *¿Qué es el XP Programming?* Obtenido de https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum/

JM Industrial. (15 de 07 de 2016). *Datalogger, ¿qué es?* Obtenido de https://www.jmi.com.mx/literatura/blog/item/11-data-logger-que-es.html

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.

Madruñero, P. (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, Ibarra.

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

Oriol, E. (30 de 06 de 2016). *Introducción a Angular 2 (parte I) – Modulo, Componente, Template y Metadatos*. Obtenido de http://blog.enriqueoriol.com/2016/06/introduccion-a-angular-2-parte-i-componente.html#template

Oriol, E. (14 de 07 de 2016). *Introducción a Angular 2 (parte III) – Servicios*. Obtenido de http://blog.enriqueoriol.com/2016/07/introduccion-angular-2-parte-iii-servicios.html

Porras. (2019). *METODOLOGÍA ÁGIL ICONIX EN LA CALIDAD DEL PRODUCTO SOFTWARE, LIMA, 2017.* Universidad Nacional Federico villareal, LIMA – PERÚ. .

Rojas, C. (29 de 06 de 2019). *¿Como crear un componente en Angular?* Obtenido de https://blog.ng-classroom.com//blog/angular/Angular-Crear-Componentes/

Romero Dorado, J. D., & Mina Posu, F. A. (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 SAN AMBIENTE.* Cali, Institucion universitaria Antonio Jose Camacho.

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

Silva Ascuntar, S., & García G., Y. (2018). *DOCUMENTACIÓN DE LA METOLODOGÍA ICONIX A TRAVÉS DEL DESARROLLO DEL CASO "ORIÉNTATE CALI.* INSTITUCIÓN UNIVERSITARIA ANTONIO JOSE CAMACHO. CALI, VALLE DEL CAUCA.

Taft, D. K. (12 de 2002). *IBM Acquires Rational*. Obtenido de https://www.eweek.com/pc-hardware/ibm-acquires-rational

team of developers Bootstrap . (2011). *getbootstrap*. Obtenido de https://getbootstrap.com/docs/4.4/about/overview/

The PostgreSQL Global Development Group. (14 de 11 de 2019). *postgresql*. Obtenido de https://www.postgresql.org/about/

Valencia, D. G. (2018). *http://repositorio.utn.edu.ec/*. Obtenido de http://repositorio.utn.edu.ec/bitstream/123456789/8264/1/PG%20659%20TESIS.pdf

1. Rosenberg, D., & Scott, K. (2001). *Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example* (First Edition). Addison Wesley, p.27. [↑](#footnote-ref-1)