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

Jaguel Lozano Argote, Jhanluy Bolívar Guzmán & John Jairo Ramírez

Santiago de Cali, 2019

Institución universitaria Antonio José Camacho

Facultad de ingeniería

Tecnología en sistemas de información

**Resumen**

El presente proyecto consiste en el desarrollo de un módulo para la conexión y transferencia de datos desde una estación de monitoreo pertenecientes a la empresa Sanambiente.

Específicamente para establecer los protocolos de comunicación y la configuración necesaria para dicha conexión, recepción de los datos y almacenaje de los mismos. Con la finalidad de lograr una mejora en eficacia y agilizar los procesos, garantizando una mejor y protección de los datos durante dicho dichos procesos. El software estuvo enmarcado en el tipo de investigación deductiva. 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 actualmente.

Este módulo de la aplicación web, ha sido creado conforme las necesidades específicas de la empresa, será en adelante parte del software de gestión de datos ambientales en dicha entidad, por lo que este documento también se referirá a la transición definitiva del modelo de conexión y transferencia utilizada actualmente de los datos. Para la elaboración de la aplicación web y el cumplimiento de los objetivos planteados se utilizaron diversas herramientas tecnológicas como AngularCli; el framework para el desarrollado de aplicaciones web en TypeScript, el sistema de gestión de base de datos de postgresql.

**Abstract**

The present project consists of the development of a module for the connection and transfer of data from a monitoring station belonging to the company Sanambiente.

Specifically to establish the communication protocols and the necessary configuration for said connection, reception of the data and storage thereof. In order to achieve an improvement in efficiency and streamline processes, ensuring better and data protection during said processes. The software was framed in the type of deductive research. A series of data collection techniques were used, specifically through consultations and interviews with the client about procedures that are performed in each of the processes currently.

This module of the web application, has been created according to the specific needs of the company, will be part of the environmental data management software in that entity, so this document will also refer to the definitive transition of the connection model and currently used transfer of the data. For the elaboration of the web application and the fulfillment of the proposed objectives, various technological tools such as Angular were used; the framework for the development of web applications in Typescript, the database management system of Postgresql.

**Tabla de Contenidos**

[Introducción 1](#_Toc410628920)

[Problema de investigación](#_Toc410628920) 2

Planteamiento del problema2

Formulación del problema 2

Sistematización del problema 2

[Objetivos](#_Toc410628920) 3

Objetivo general3

Objetivos específicos 3

Justificación4

[Marco referencial](#_Toc410628920) 5

Antecedentes5

Marco teórico8

Marco conceptual [1](#_Toc410628922)8

Marco legal [21](#_Toc410628922)

[Desarrollo de contenido (iconix) 22](#_Toc410628925)

[Fase 1 análisis de requerimientos 2](#_Toc410628926)2

[Requerimientos 2](#_Toc410628927)2

[Casos de uso 2](#_Toc410628928)3

[Prototipos de interfaz 2](#_Toc410628928)

[Fase 2 diseño preliminar 2](#_Toc410628926)

[Diagrama de clases 2](#_Toc410628927)

[Fase 3 revisión crítica del diseño 2](#_Toc410628926)

[Diagrama de secuencia 2](#_Toc410628927)

[Fase 4 implementación 2](#_Toc410628926)

[Conclusiones 5](#_Toc410628929)

Bibliografía [6](#_Toc410628930)

**Lista de tablas**

[Tabla 1. Requerimientos.](#_Toc410629016) 22

[Tabla 2. Caso de uso No.1.](#_Toc410629016) 22

[Tabla 3. Caso de uso No.2.](#_Toc410629016) 22

[Tabla 4. Caso de uso No.3.](#_Toc410629016) 22

[Tabla 5. Caso de uso No.4.](#_Toc410629016) 22

[Tabla 6. Caso de uso No.5.](#_Toc410629016) 22

[Tabla 7. Caso de uso No.6.](#_Toc410629016) 22

[Tabla 8. Caso de uso No.7.](#_Toc410629016) 22

[Tabla 9. Caso de uso No.8.](#_Toc410629016) 22

[Tabla 10. Caso de uso No.9.](#_Toc410629016) 22

[Tabla 11. Caso de uso No.10.](#_Toc410629016) 22

**Lista de figuras**

[Figura 1. Proceso de ICONIX.](#_Toc410629185) 9

Figura 2. Ubicación del modelado de dominio en el proceso de ICONIX 11

Figura 3. Ejemplo de diagrama de caso de uso [.](#_Toc410629185) 12

Figura 4. Objetos del diagrama de robustez [.](#_Toc410629185) 13

Figura 5. Ejemplo de un diagrama de robustez [.](#_Toc410629185) 14

Figura 6. Elementos de un diagrama de secuencia [. 1](#_Toc410629185)6

Figura 7. Prototipo conectar estacion.[. 1](#_Toc410629185)6111

Figura 8. Prototipo mantenimiento. [. 15](#_Toc410629185)

Figura 9. Prototipo crear region crud [. 15](#_Toc410629185)

Figura 10. Prototipo crear estacion crud [. 15](#_Toc410629185)

Figura 11. Prototipo crear organizacion crud [. 15](#_Toc410629185)

Figura 12. Prototipo crear categoria crud [. 15](#_Toc410629185)

Figura 13. Prototipo crear ciudad crud [. 15](#_Toc410629185)

Figura 14. Prototipo crear rango crud [. 15](#_Toc410629185)

Figura 15. Prototipo crear alerta crud[. 15](#_Toc410629185)

Figura 16. Prototipo crear cuadro de tiempo crud[. 15](#_Toc410629185)

# 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 Sanambiente, ya que actualmente no existe un módulo que permita un manejo adecuado, efectivo y unificado de los datos correspondientes a las estaciones y los contaminantes.**

**Se desarrolla una base de datos; mediante el sistema de gestión de bases de datos relacional postgresql y la interfaz usando angular y bootstrap.**

**El objeto principal del módulo de comunicación y transferencia de datos es brindar las herramientas necesarias para lograr optimizar y controlar de mejor manera los procesos de conexión y transferencia de los datos obtenidos por las diferentes estaciones de medición ubicadas en diferentes zonas geográficas del país.**

# Problema de investigación

## Planteamiento del problema

La empresa Sanambiente ofrecer soluciones integrales en productos y servicios que aporten 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 actuales de forma rápida, eficaz, verídica, accesible y fácil, actualmente la empresa cuenta con un software de escritorio que entrega a cada cliente para que administre su sistema el cual no satisface completamente las necesidades de la empresa.

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

El software se desarrollará por módulos, el módulo que desarrollaremos en el actual documento es la conexión por ModBus TCP/IP y FTP con el dataloger que recibe los datos de los sensores que tenga casa estación para ser procesados.

## Formulación del problema

¿De qué manera se puede transferir la información desde el dataloger a una base de datos facilitando el control y seguimiento de los datos mejorando el modelo actual?

## Sistematización del problema

* ¿Cómo realizar la conexión con el dataloger?
* ¿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

## Objetivo general

Desarrollar un módulo que permita conectar y transferir información con los diferentes dataloger de la empresa SANAMBIENTE.

## Objetivos específicos

* Establecer la conexión con el dataloger 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 conexión y transferencia de datos es necesario para el personal operativo de Sanambiente; porque les brinda la facilidad necesaria para realizar de manera fácil y rápida procesos como la conexión con una determinada estación y la recepción de datos sobre contaminantes, llevar un control sobre cada uno de los dispositivos de medición con los cuales pueden establecer una conexión mediante un determinado protocolo de comunicación, su respectiva ubicación y tener una descripción acerca de la compañía a la cual pertenece o para la cual está realizando la toma de datos la estación de monitoreo. Así mismo es favorable para el personal ya que la interfaz permite mejorar y facilitar el establecer la conexión y la recepción de los datos, lo cual originara que estos datos, sean obtenidos y almacenados de forma más rápida y veraz.

El módulo de conexión y transferencia es unos de los componentes que brinda un gran impacto social y ambiental porque permite obtener y salvaguardar los datos sobre los contaminantes y sus niveles, pueden tomarse decisiones que permitan optimizar los procesos de prevención, se obtiene un mejoramiento cultural e intelectual acerca de los contaminantes y sus orígenes. Lo cual aumentara la reducción de los mismos, permitiendo un mayor grado de cooperación, conocimiento y sanidad en las vidas de las personas y de sus familias.

# Marco referencial

## Antecedentes

Para el presente proyecto se analizaron diferentes proyectos que tuvieron el mismo o similar objeto de estudio que el presente proyecto tiene, estos fueron tomados como base para llevar a cabo el desarrollo de este proyecto, estos proyectos brindan una guía para saber cómo proceder y de qué forma iniciar la elaboración del proyecto, los pasos a seguir, cuáles pueden ser de utilidad para el presente proyecto y poderlos adecuar de acuerdo a las necesidades que éste tiene determinadas. La información que será mostrada en este marco es de aquellos proyectos que tienen relación con los temas que el presente proyecto aborda: ISO/IEC 29110, DevOps y Metodología ICONIX. El tipo de revisión que se hizo a estos proyectos es meramente enfocado en la parte donde tocan 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, cuatro fueron los que más tenían mejor dominio del tema y mostraban de forma clara la aplicación de estos temas 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 en la cual con llevo aportar a la institución una metodología para la elaboración del desarrollo de software.

En la tesis (Laporte, O´connor, & García, 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 ya que la ISO/IEC 29110 ayuda a las pequeñas empresas poder 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, 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 teórico

Existen diferentes metodologías que facilitan la ejecución del ciclo de vida de un proyecto de software. Cada una es útil en un contexto en particular, están las ágiles que reducen la documentación y se enfocan 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 (Silva & Garcia, 2018).

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 casos de uso y los prototipos para un mejor entendimiento con el usuario quien acompaña el proyecto eventualmente.

ICONIX.

Es una metodología de desarrollo de software de tamaño medio, cuyo análisis y capacidad de diseño se basa en UML. Fue elaborada por Doug Rosenberg y Kendall Scott. Se dice que ICONIX se encuentra entre la complejidad de RUP y la simplicidad de XP, sin eliminar las tareas de análisis y de diseño que XP no contempla. (Valdez, García Ruíz, Jiménez López. 2014).

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).

#### 

Figura 1. Proceso de ICONIX.

El proceso ICONIX se originó varios años antes de UML y el proceso unificado como una síntesis y destilación de 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.

Se realizó una síntesis de estas tres escuelas diferentes de pensamiento orientado a objetos (OO) porque las fortalezas y debilidades de estas metodologías se complementaban entre sí.

### Se destacan tres características fundamentales de ICONIX:

* Iterativo e incremental: múltiples iteraciones ocurren entre el desarrollo del modelado del dominio y la identificación de los casos de uso. El modelo estático se refina incrementalmente durante las iteraciones sucesivas a través del modelo dinámico (Fiestas Jacinto, 2015, p. 40).
* Trazabilidad: Cada paso está referenciado por algún requerimiento. Nunca hay un punto en que el proceso le permita desviarse lejos de las necesidades del usuario (Fiestas Jacinto, 2015, p. 40). Según (Amavizca Valdez et al., 2014) trazabilidad es la capacidad de seguir una relación entre los diferentes artefactos producidos.
* Aerodinámica del UML: la metodología ofrece un uso “aerodinámico” del UML, como lo son los diagramas de caso de uso, diagramas de secuencia y de colaboración (Amavizca Valdez et al., 2014, p. 3).

Como se dijo anteriormente, ICONIX basa su análisis y diseño en UML, por lo tanto, utiliza un conjunto de diagramas que auxilian a la metodología a proveer la mejor respuesta. Los principales diagramas que maneja ICONIX son:

## Modelado de dominio:

Es la tarea de crear un glosario de proyecto o un diccionario de términos utilizados en el proyecto. Su propósito es asegurarse de que todos en el proyecto entiendan el espacio problemático en términos inequívocos. El modelado de dominio para un proyecto define el alcance y forma la base sobre la cual construir los casos de uso (Rosenberg et al., 2005, p. 7).

#### 

Figura 2. Ubicación del modelado de dominio en el proceso de ICONIX[[1]](#footnote-1).

## Diagramas de caso de uso:

Es el centro conceptual del desarrollo, porque guía todo el proceso de ICONIX (Bona, 2002, p. 68).

Un caso de uso es una secuencia de acciones que un actor (generalmente una persona, pero quizás una entidad externa, como otro sistema) realiza dentro de un sistema para lograr un objetivo particular. Un caso de uso completo e inequívoco, describe un aspecto del sistema sin suponer ningún diseño o implementación específica. El resultado del modelado de los casos de uso debería ser, que toda la funcionalidad requerida del sistema, se describa en los casos de uso. Si no se adhiere a este principio básico, se corre el riesgo de que los ingenieros construyan un sistema excelente que no es el que desea el cliente (Rosenberg et al., 2005, p. 42).

#### 

*Figura 3.* Ejemplo de diagrama de caso de uso[[2]](#footnote-2).

## Análisis de robustez:

Es una forma de analizar el texto del caso de uso e identificar un primer conjunto de posibles objetos de primera estimación para cada caso de uso. Estos objetos se clasifican en tres estereotipos (Rosenberg et al., 2005, p. 49):

* Objetos de límite (boundary objects): los actores los usan para comunicarse con el sistema (a menudo referidos como objetos de interfaz o frontera).
* Objetos de entidad (entity objects): son normalmente objetos del modelado de dominio, generalmente asignados en tablas de una base de datos.
* Objetos de control (control objects): funcionan como integradores entre los objetos de límite y los objetos de identidad. Generalmente, se convierten en métodos de objetos de entidad o de objetos de límite.

#### 

Figura 4. Objetos del diagrama de robustez[[3]](#footnote-3).

#### 

Figura 5. Ejemplo de un diagrama de robustez[[4]](#footnote-4).

Las reglas básicas que se deben aplicar al diagrama de robustez son (Rosenberg & Scott, 2001, pp. 61-62):

* Los actores sólo se pueden comunicar con objetos de límite.
* Los objetos de límite sólo se pueden comunicar con actores y objetos de control.
* Los objetos de entidad sólo se pueden comunicar con los objetos de control.
* Los objetos de control sólo se pueden comunicar con los objetos de límite y de c Diagrama de secuencia:

## Diagrama de secuencia:

El diagrama de secuencia muestra la colaboración dinámica entre los distintos objetos del sistema (Bona, 2002, p. 74). Es el núcleo del modelo dinámico y muestra todos los cursos alternos que pueden tomar los casos de uso (Carbajal & Martín, 2013, p. 25).

En (Rosenberg & Scott, 2001, pp. 83-84), se destacan cuatro tipos de elementos en un diagrama de secuencia, que son:

* Texto del caso de uso (flujo de acción). Copiar este elemento al margen izquierdo del diagrama de secuencia.
* Objetos de los diagramas de robustez. Están representado en cajas rectangulares con el nombre del objeto y opcionalmente con el nombre de la clase a la que pertenece (objeto: clase).
* Mensajes. Representados como flechas entre objetos.
* Los métodos (operaciones) se muestran como rectángulos que se encuentran encima de las líneas de puntos que pertenecen a los objetos a los que se asignan los métodos.

#### 

Figura 6. Elementos de un diagrama de secuencia[[5]](#footnote-5).

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

## Análisis de requerimientos:

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

* Identificar en el “mundo real” los objetos y todas las relaciones de agregación de generalización entre ellos. Utilizar el diagrama de clases de alto nivel definido como modelado de dominio.
* Presentar, si es posible, un prototipado rápido de la interfaz del sistema, o diagramas de navegación, de manera que el cliente pueda comprender mejor el sistema propuesto.
* Identificar los casos de uso del sistema, mostrando los actores involucrados. Utilizar diagramas de caso de uso.
* Organizar los casos de uso en grupos. Capturar esta organización en un diagrama de paquete.
* Asociar los requerimientos funcionales en los casos de uso y objetos de dominio en este escenario.

## Análisis y diseño preliminar:

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

* Escribir los casos de uso, como flujo principal de acciones, pudiendo contener el flujo alternativo y el flujo de excepción.
* Presentar el análisis de robustez. Siendo que, para cada caso de uso se debe identificar un conjunto de objetos y actualizar el diagrama de clases del modelado de dominio.
* Finalizar la actualización del diagrama de clases.

## Revisión del diseño preliminar.

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

* Especificar comportamiento a través del diagrama de secuencia. Para cada caso de uso, identificar los mensajes entre los diferentes objetos. Si es necesario, utilizar diagrama de colaboración para representar las transacciones claves entre los objetos.
* Terminar el modelo estático, añadiendo detalles al proyecto en el diagrama de clase.
* Verificar con el equipo si el proyecto satisface todos los requerimientos identificados.

## Revisión crítica del diseño.

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

* Si es necesario, generar diagramas de implementación y componentes para apoya la fase de implementación.
* Escribir/generar código.
* Realizar pruebas unitarias y de integración.
* Realizar pruebas de aceptación del usuario.

(Cristian Fernando Balanta pazu, 2019)

## Marco conceptual

* **AngularJS:** AngularJS es un framework modelo vista control de JavaScript que se utiliza para el desarrollo páginas Web Front End que permite crear aplicaciones.
* **Framework:** Es un marco de trabajo o conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, en el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.
* **JavaScript:** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.
* **Front End:** En un compilador el front-end traslada el lenguaje del código fuente a una representación intermedia que a su vez funciona con el back-end para producir en la salida el código.
* **PostgreSQL:** PostgreSQL es un avanzado sistema de bases de datos relacionales basado en Open Source. Esto quiere decir que el código fuente del programa esta disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus

necesidades. PostgreSQL está bajo licencia BSD.

* **Bootstrap:** Boostrap es el framework de CSS, HTML y JavaScript más popular, que nos permite desarrollar webs que se ajustan a cualquier resolución y dispositivo. («Bootstrap 4. ¿Vale la pena actualizar? « Blog SEOCOM», s. f.)
* **Componente en Angular:** Un componente en Angular es una combinación de un archivo html con un ts y algunas veces scss para crear un elemento con características propias tanto de comportamiento como de apariencia que se puede mostrar en un navegador. Lo puedes pensar como en una pieza de lego la cual vas juntando con otras piezas de lego para formar algo interesante para interactuar con un usuario. (Rojas, s. f.)
* **GitHub:** Es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.
* **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. («Git», 2019)
* TypeScript: Es un lenguaje tipado de javascript pensado en desarrollar aplicaciones robustas, aplicando clases, módulos y optimizando el flujo de desarrollo.
* Patrón de diseño Repositoy: 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 CRUD) con la intención de que la implementación de dichas operaciones se oculte de las capas superiores. (Flower, 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.

## 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

## Fase 1

## Análisis de requerimientos:

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 | El sistema 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 |

## Casos de uso:

Tabla 2. Caso de uso No.1.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Conectar con estación | | | | | | CU1 |
| Actores | | | Administrador | | | | | | |
| Referencias | | | Requerimiento RF 1 | | | |  | | |
| Precondición | | | El sistema deberá estar conectado a la red de internet | | | | | | |
| Postcondición | | | El sistema debe permitir el enlace y transferencia de datos de los dataloger mediante Internet. | | | | | | |
| Autor | | | Fabián Mina | Fecha | |  | | Versión |  |
| Aprobación | | | Aprobado/No aprobado | | | | | | |
| Propósito | | | | | | | | | |
| Conectarse mediante protocolo FTP o 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 | | | |
| 3 | El usuario confirma la información botón conectar  Serial -Abreviación nombre corto de la estación -organización -categoría  -Región -Localización -ciudad -latitud y longitud -elevación. | | | | **4** | El sistema valida los campos obligatorios  Id -Serial -organización -categoría -Región -Ciudad | | | |
|  |  | | | | **5** | El sistema omite campos vacíos (No son relevantes y no afectan operación)  -Abreviación nombre corto de la estación. -Localización -ciudad -latitud y longitud -elevación | | | |
|  |  | | | | **6** | El sistema se conecta a la estación. | | | |
| Cursos Alternos | | | | | | | | | |
| 5 | | Si entre los campos vacíos se encuentran: organización, categoría, región, ciudad; se mostrará mensaje en pantalla que estos campos son obligatorios y no se procede a realizar la conexión con la estación. | | | | | | | |
| 6 | | 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 | | | | | | | |
| 6 | | 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 | | | Crear Mantenimiento | | | | | | | | CU2 |
| Actores | | | Administrador | | | | | | | | |
| Referencias | | | Requerimiento RF 1 | | | | | |  | | |
| Precondición | | | El sistema deberá tener almacenada información del RF 1 para realizar calibración de datos. | | | | | | | | |
| Postcondición | | | En la pantalla de selección quedan los datos con los cuales se realiza el ajuste calibración. | | | | | | | | |
| Autor | | | Fabián Mina | | Fecha | |  | | | Versión |  |
| Aprobación | | | | Aprobado/No aprobado | | | | | | | |
| Propósito | | | | | | | | | | | |
| Gestionar estación | | | | | | | | | | | |
| Curso Normal | | | | | | | | | | | |
| 1 | El usuario Selecciona la hora, minutos  (GMT) | | | | | **2** | | El sistema valida el GMT por estación y corrige la hora | | | |
| 3 | El usuario Digita el campo (Factor) X | | | | |  | |  | | | |
| 4 | El usuario Digita el valor en campo (Factor) +b | | | | | **5** | | El sistema valida de manera automática los campos con la formula Y= mx+b | | | |
| 6 | El usuario presiona botón calibrar. Para realizar calibración de los datos manualmente | | | | | **7** | | El sistema realiza validación de los datos digitados | | | |
| 8 | El usuario guardar la información diligenciada | | | | | **9** | | El sistema guarda la información, Muestra mensaje en pantalla. “Información Guardada con éxito” | | | |
| Cursos Alternos | | | | | | | | | | | |
| 2 | | El sistema intenta escribir la hora y los minutos: El campo está inhabilitado para escritura y solo permite la selección. | | | | | | | | | |
| 3 | | El usuario digita un valor alfanumérico: El campo solo permite valores numéricos por lo tanto no registra ningún valor diferente | | | | | | | | | |
| 6 | | En caso de que el usuario no calibre los datos, el sistema lo hará automáticamente. | | | | | | | | | |
| 9 | | En caso de que los datos no sean válidos o se presente algún problema al momento de llenar el formulario, el sistema debe mostrar mensaje en pantalla, especificando el error presentado. | | | | | | | | | |

Tabla 4. Caso de uso No.3.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | | Crear región (Tabla parámetro) | | | | | | | | CU3 |
| Actores | | | | Administrador | | | | | | | | |
| Referencias | | | | Requerimiento RF 1 y 2 | | | | | |  | | |
| 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/No 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** | | El sistema muestra en el listado la nueva región creada. | | | |
| Cursos Alternos | | | | | | | | | | | | |
| 3 y 4 | | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 6 | | | 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 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 | | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje y el cursor se ubica en el campo. | | | | | | | | | | |

Tabla 5. Caso de uso No.4.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | | | Crear estación (Tabla parámetro) | | | | | | | | | | CU4 |
| Actores | | | | | Administrador | | | | | | | | | | |
| 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/No 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 | | | |
| 8 | | | El usuario selecciona el campo nombre de organización | | | | | | | **9** | | El sistema carga la lista de organizaciones almacenadas | | | |
| 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 | | | | | | |  | |  | | | |
| 20 | | | El usuario diligencia el campo elevación. | | | | | | |  | |  | | | |
| 21 | | | El usuario ingresa un número entero entre 0 y 120 para la base de tiempo | | | | | | | **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 | | | | | | | | | | | | | | | |
| 3,4,5,19,20,23 | | | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | | | |
| 6, 8, 10, 12,14, 15, 17 | | | | 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. | | | | | | | | | | | |
| 19 y 20 | | | | El usuario digita valores alfanuméricos Se muestra el mensaje “Los valores deben ser decimales” | | | | | | | | | | | |
| 22 | | | | La estación ya existe con la misma base de tiempo muestra mensaje “Ya existe esta base de tiempo en la estación” | | | | | | | | | | | |
| 27 | | | | Hay campos vacíos, el sistema muestra el mensaje “Hay campos vacíos”, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | | | |
| 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. | | | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | | | | | |

*Tabla 6.* Caso de uso No.5.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear organización (Tabla parámetro) | | | | | | | | CU5 |
| 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/No 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 una descripción para la organización. 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. “Organización creada correctamente” | | | |
|  |  | | | | | **7** | | El sistema muestra en el listado la nueva organización creada. | | | |
| Cursos Alternos | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 3,4 | | 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 y descripción para actualización. | | | |
| 5 | El usuario actualiza el campo nombre de la organización, 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 | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 1,5 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |

*Tabla 7.* Caso de uso No.6*.*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear categoría (Tabla parámetro) | | | | | | | | CU6 |
| 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/No 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 | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 3,4 | | 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 | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 5 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |

*Tabla 8.* Caso de uso No.7.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear Ciudad (Tabla parámetro) | | | | | | | | CU7 |
| 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/No 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 | | | | | | | | | | | |
| 7 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 3,6 | | 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 | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 1,5 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |

*Tabla 9.* Caso de uso No.8.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear rango (Tabla parámetro) | | | | | | | | CU8 |
| 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/No 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 | | | | | | | | | | | |
| 3,7 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |
| 10 | | 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 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 | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 5 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |

*Tabla 10.* Caso de uso No.9.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear alerta (Tabla parámetro) | | | | | | | | CU9 |
| 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/No 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 oprime el botón guardar | | | | **7** | | El sistema valida que todos los campos estén llenos y guarda. Muestra mensaje de confirmación de transacción. “Alerta guardada con éxito” | | | | |
|  |  | | | | **8** | | Se muestra en la cuadrícula donde están listados todas las alertas creadas. | | | | |
| Cursos Alternos | | | | | | | | | | | |
| 7 | | 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 | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |

*Tabla 11.* Caso de uso No.10.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | | | Crear cuadro de tiempo (Tabla parámetro) | | | | | | | | CU10 |
| 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/No 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 | | | | | | | | | | | |
| 11 | | En caso de presentarse un error el sistema mostrara mensaje en pantalla indicando el error. | | | | | | | | | |
| 4 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 3,7 | | 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 | | | | | | | | | | | |
| 6 | | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. | | | | | | | | | |
| 5 | | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. | | | | | | | | | |

## Prototipos de interfaz:

Caso de uso Numero 1.

#### 

*Figura 7.* Prototipo conectar estación.

Caso de uso Numero 2

#### 

*Figura 8.* Prototipo mantenimiento.

Caso de uso Numero 3 (tabla parámetro)

#### 

*Figura 9.* Prototipo crear región crud.

Caso de uso Numero 4 (tabla parámetro)

#### 

*Figura 10.* Prototipo crear estación crud.

Caso de uso Numero 5 (tabla parámetro)

#### 

*Figura 11.* Prototipo crear organización crud.

Caso de uso Numero 6 (tabla parámetro)

#### 

*Figura 12.* Prototipo crear categoría crud.

Caso de uso Numero 7 (tabla parámetro)

#### 

*Figura 13.* Prototipo crear ciudad crud.

Caso de uso Numero 8 (tabla parámetro)

#### 

*Figura 14.* Prototipo crear rango crud.

Caso de uso Numero 9 (tabla parámetro)

#### 

*Figura 15.* Prototipo crear alerta crud.

Caso de uso Numero 10 (tabla parámetro)

#### 

*Figura 16.* Prototipo crear cuadro de tiempo crud.

## Fase 2

## Diseño preliminar:

Diagrama de clases del caso de uso Numero 1

[**Bibliografía**](#_41mghml)

Aaron Levi Grajales Gomez, & Cristian Fernando Balanta pazu. (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 (Tesis). Antonio Jose Camacho.

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, Ibarra.

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.

LEGARIA. (2018). IMPLEMENTACIÓN DE PROCESO ORGANIZACIONAL DE GESTIÓN DE PROYECTOS EN DEVELOPIT. SANTIAGO DE CHILE.

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

Mnkandla, E., & Dwolatzky, B. (2004). A Survey of Agile Methodologies [Científico].

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].

ICONIX Brand Group. (2016). Manual introductorio de ICONIX.

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

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, Trujillo.

Bona, C. (2002). AVALIAÇÃO DE PROCESSOS DE SOFTWARE: UM ESTUDO DE CASO EM XP E ICONIX. Universidade Federal de Santa Catarina, Florianópolis.

Carbajal, W., & Martín, D. (2013). Implementación de un sistema informático web para la gestión de compras de la empresa Certicom S.A.C usando la metodología ICONIX y frameworks Spring, Hibernate y Richfaces. Universidad Privada Antenor Orrego, Trujillo.

Bootstrap 4. ¿Vale la pena actualizar? « Blog SEOCOM. (s. f.). Recuperado 29 de junio de 2019, de <https://seocom.agency/es/blog/bootstrap-4-vale-la-pena-actualizar/>

Git. (2019). En Wikipedia, la enciclopedia libre. Recuperado de <https://es.wikipedia.org/w/index.php?title=Git&oldid=116653857>

Rojas, C. (s. f.). ¿ Como crear un componente en Angular ? Recuperado 29 de junio de 2019, de ion-book website: <https://blog.ng-classroom.com//blog/angular/Angular-Crear-Componentes/>

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

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)
2. Bona, C. (2002). *AVALIAÇÃO DE PROCESSOS DE SOFTWARE: UM ESTUDO DE CASO EM XP E ICONIX*. Universidade Federal de Santa Catarina, Florianópolis, p.44. [↑](#footnote-ref-2)
3. Rosenberg, D., & Stephens, M. (2007). *Use Case Driven Object Modeling with UML*. United States of America: Apress, p.103. [↑](#footnote-ref-3)
4. Rosenberg, D., Stephens, M., & Collins-Cope, M. (2005). *Agile Development with ICONIX Process*. United States of America: Apress, p.48. [↑](#footnote-ref-4)
5. Bona, C. (2002). *AVALIAÇÃO DE PROCESSOS DE SOFTWARE: UM ESTUDO DE CASO EM XP E ICONIX*. Universidade Federal de Santa Catarina, Florianópolis, p.75. [↑](#footnote-ref-5)