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

Presentado por:

John Jairo Ramírez

Jhanluy Bolívar Guzmán

Jaguel Lozano Argote

Institución universitaria Antonio José Camacho

Facultad de ingeniería

Tecnología en sistemas de información

Santiago de Cali

2019

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

John Jairo Ramírez

Jhanluy bolívar guzmán

Jaguel lozano Argote

Asesor

Beatriz Marín

Institución universitaria Antonio José Camacho

Facultad de ingeniería

Tecnología en sistemas de información

Santiago de Cali

2019

**Tabla de contenido**

[Introducción ……………………………………………………………………………. 5](#_30j0zll)

[1. Problema de investigación …………………………………………………………... 6](#_1fob9te)

[1.1 Planteamiento del problema](#_3znysh7) ……………………………………………………... [6](#_3znysh7)

[1.2 Formulación del problema ………………………………………………………. 6](#_2et92p0)

[1.3 Sistematización del problema …………………………………………………… 6](#_tyjcwt)

[2. Objetivos …………………………………………………………………………….. 7](#_3dy6vkm)

[2.1 Objetivo general …………………………………………………………………. 7](#_1t3h5sf)

[2.2 Objetivos específicos ……………………………………………………………. 7](#_4d34og8)

[3. Justificación](#_2s8eyo1) …………………………………………………………………………. [8](#_2s8eyo1)

[4. Marco referencial](#_17dp8vu) ……………………………………………………………………. 9

[4.1 Marco histórico](#_3rdcrjn) ………………………………………………………………….. 9

[4.2 Marco teórico](#_lnxbz9) …………………………………………………………………... 12

[4.3 Marco conceptual](#_35nkun2) ………………………………………………………………..

[4.4 Marco legal](#_1ksv4uv) ………………………………………………………………………

[5. Desarrollo de contenido](#_2bn6wsx) (Metodología ICONIX) …………………………………...

[5.1 Fase 1 (Análisis de requerimientos) …………………………………………….](#_qsh70q)

[5.1.1 Requerimientos ……………………………………………………………...](#_3as4poj)

[5.1.2 Casos de uso …………………………………………………………………](#_1pxezwc)

[5.1.3 Prototipos de Interfaz ………………………………………………………..](#_1pxezwc)

[5.2 Fase 2 (Análisis y Diseño Preliminar) ……………………………………………](#_2p2csry)

[5.2.1 Diagrama de clases …………………………………………………………..](#_147n2zr)

[5.3 Fase 3 (Diseño) …………………………………………………………………...](#_2p2csry)

[[5.3.1 Diagramas de Secuencia ………………………](#_qsh70q)](#_147n2zr)[…………………………….](#_qsh70q)

[5.4 Fase 4 (Implementación y pruebas)](#_qsh70q)

[6. Resumen](#_32hioqz) ……………………………………………………………………………..

7. Abstracción …………………………………………………………………………

[8. Conclusiones](#_1hmsyys) …………………………………………………………………………

[9. Bibliografía](#_41mghml)

**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.

1. **Planteamiento** 
   1. **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 desarrollara por módulos, el modulo 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.

* 1. **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?

* 1. **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?

1. **Objetivos**
   1. **Objetivo general**

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

* 1. **Objetivos específicos**
* Facilitar 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.
* Desarrollar interfaces amigables que permitan la fácil interacción de los datos para el control y seguimiento de la información

1. **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.

1. **Marco referencial**
   1. **Marco histórico o 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.

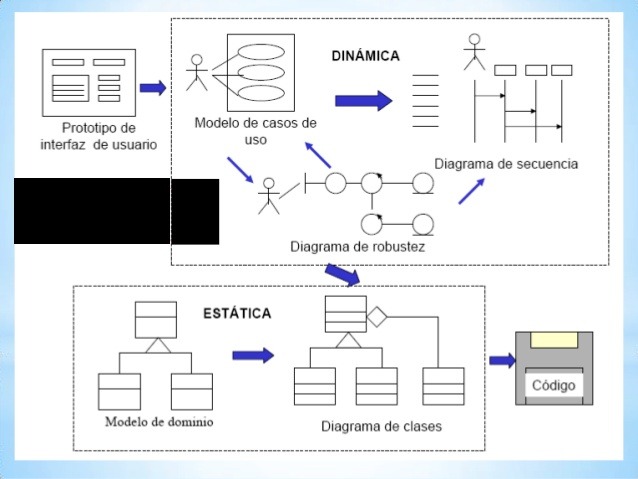
* 1. [**Marco teórico**](#_lnxbz9)

**Metodología ICONIX**

Es una metodología de desarrollo de software de tamaño medio, cuyo análisis y capacidad de diseño se basa en UML (Unified Modeling Language (UML))(Mnkandla & Dwolatzky, 2004, p. 1). Fue elaborada por Doug Rosenberg y Kendall Scott. Se dice que ICONIX se encuentra entre la complejidad de RUP (Rational Unified Processes) y la simplicidad y pragmatismo de XP (eXtreme Programming), sin eliminar las tareas de análisis y de diseño que XP no contempla (Amavizca Valdez, García Ruíz, Jiménez López, Duarte Guerrero, & Vásquez Brindis, 2014, p. 3).

El fundamento de ICONIX es el hecho de que un 80% de los casos pueden ser resueltos tan solo con un uso del 20% del UML, con lo cual se simplifica en gran medida el proceso sin perder documentación al dejar solo aquello que es necesario. Esto implica un uso dinámico del UML de tal forma que siempre se pueden utilizar otros diagramas aparte de los ya estipulados si se cree conveniente (ICONIX Brand Group, 2016, p. 1).

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, p. 41).



*Figura 10. Proceso de ICONIX .*

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

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.

* Características de ICONIX:

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).

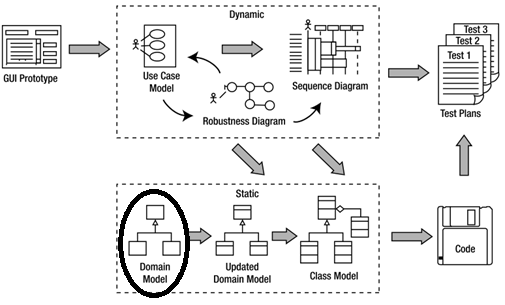


Figura 11. Ubicación del modelado de dominio en el proceso de ICONIX[[1]](#footnote-1).

* 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).

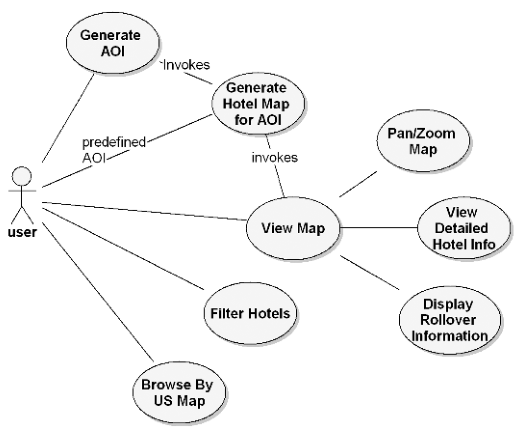


Figura 12. Ejemplo de diagrama de caso de uso[[2]](#footnote-2).

* 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.

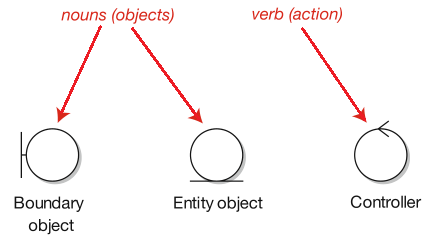


Figura 13. Objetos del diagrama de robustez[[3]](#footnote-3).

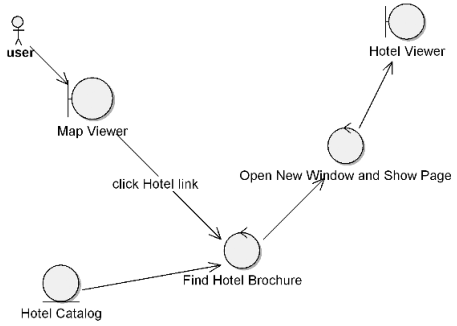


Figura 14. Ejemplo de un diagrama de robustez[[4]](#footnote-4).

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 control.
* 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.

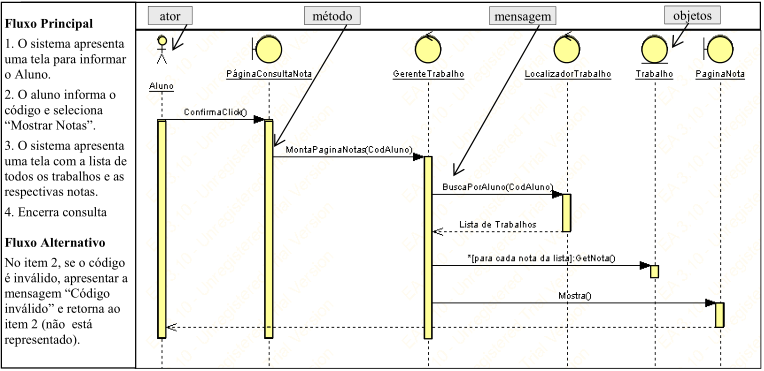


Figura 15. Elementos de un diagrama de secuencia[[5]](#footnote-5).

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.

Hito**:** Revisión de los requisitos.

* 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.

Hito: Revisión del diseño preliminar.

* Diseño detallado:

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.

Hito: Revisión crítica del diseño.

* Implementación:

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)

* 1. **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 esta 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)

* 1. [**Marco legal**](#_1ksv4uv)

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)).

1. **Desarrollo Metodología ICONIX**
   1. **Fase 1 (Análisis de requerimientos)**

* Requerimientos:

1. El sistema debe permitir el enlace y transferencia de datos de los datalogger mediante Internet.
2. El sistema debe permitir crear, modificar, borrar, visualizar comandos para ser utilizados por los protocolos de comunicación X y Y.
3. El sistema debe permitir elegir que tipo de protocolo de comunicación usará una estación, pero en una organización pueden 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 dataloggers y obtener la información de estos.
10. El sistema debe permitir acceder a las estaciones mediante Modbus de acuerdo con 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.

* Modelo de Casos de Uso:

**Caso de Uso #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** |  | Fecha |  | | Versión |  |

|  |
| --- |
| **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. |  |  |
| **4** | El usuario digita una descripción de la región. |  |  |
| **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 |

|  |  |
| --- | --- |
| **Cursos Alternos** | |
| **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. |

|  |  |
| --- | --- |
| **Cursos Alternos** | |
| **6** | Hay campos vacíos, el sistema muestra el mensaje y el cursor se ubica en el campo. |

**Caso de Uso #7**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caso De Uso** | **Crear Ciudad (Tabla parámetro)** | | | | | CU**7** |
| **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 region a la que pertenece la ciudad. | | | | | |
| **Postcondición** | La ciudad quedara creada en el sistema. | | | | | |
| **Autor** |  | Fecha |  | | Versión |  |

|  |
| --- |
| **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. |
| **4** | El usuario digita una descripción para la Ciudad. |  |  |
| **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. |

|  |  |
| --- | --- |
| **Cursos Alternos** | |
| **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 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. |

|  |  |
| --- | --- |
| **Cursos Alternos** | |
| **6** | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |

**Caso de Uso #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** |  | Fecha |  | | Versión |  |

|  |
| --- |
| **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. |  |  |
| **5** | El usuario selecciona la estación a la que pertenece la alerta | **6** | El sistema despliega la lista de estaciones almacenadas en base de datos. |
| **7** | El usuario digita una descripción para la alerta creada. |  |  |
| **8** | El usuario oprime el botón guardar | **9** | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación de transacción. |

|  |  |
| --- | --- |
| **Cursos Alternos** | |
| **9** | 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, estación y descripción para actualización. |
| **5** | El usuario actualiza el campo nombre de alerta, estación, 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. |

* 1. **Fase 2**
  2. **Fase 3**
  3. **Fase 4**

**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 Angular; el framework para el desarrollado de aplicaciones web en typescript, el sistema de gestión de base de datos de postgresql.

**Abstrac**

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.

[**9. 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/

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)