Aplicación de un modelo de calidad al módulo de comunicación y traspaso de información del sistema de gestión de datos ambientales de la empresa SAN AMBIENTE

****

Jessica Daniela Romero Dorado

Fabian Alexis Mina Posu

Institución Universitaria Antonio Jose Camacho

Facultad de Ingeniería

Ingeniería en Sistemas

Santiago de Cali 2020

Aplicación de un modelo de calidad al módulo de comunicación y traspaso de información del sistema de gestión de datos ambientales de la empresa SAN AMBIENTE



Jessica Daniela Romero Dorado

Fabian Alexis Mina Posu

ING. Beatriz Eugenia Marín

Institución Universitaria Antonio Jose Camacho

Facultad de Ingeniería

Ingeniería en Sistemas

Santiago de Cali 2020

Nota de Aceptación









Presidente del Jurado



Jurado



Jurado

Ciudad y Fecha (día, mes, año) (Fecha de entrega)

Resumen

Existen diferentes modelos de calidad de software que permiten un proceso de mejora continua con su implementación. En el siguiente trabajo se exponen características de los modelos investigados a través de una revisión de información, se realiza una clasificación de estos de acuerdo con el nivel de procesos; cada modelo cuenta con unas características y una estructura la cual nos permite elegir qué modelo de calidad se puede implementar de acuerdo con el proyecto.

La temática que se encuentra en la implementación de un desarrollo de software se da acorde a la proporción de productos y servicios de calidad estandarizados por las normas que se pueden dar a conocer y se practican en la ingeniería de software que se van implementando acorde al desarrollo estipulado en este caso.

La aplicación del modelo de calidad para evaluación del software desarrollado está basada en la Norma ISO 2501n y ISO 2502n aplicando fases de metodología descriptas también en base a la IEEE829 que nos comparte la estructura necesaria y pasos aplicables a la implementación de las pruebas a realizar para el desarrollo del proyecto.

Abstract

There are different software quality models that allow a continuous improvement process with its implementation. In the following work characteristics of the investigated models are exposed through an information review, a classification of these is carried out according to the level of processes; Each model has some characteristics and a structure which allows us to choose which quality model can be implemented according to the project.

The theme found in the implementation of a software development is given according to the proportion of quality products and services standardized by the standards that can be made known and practiced in software engineering that are being implemented according to the development stipulated in this case.

The application of the quality model for the evaluation of the software developed is based on the ISO 2501n and ISO 2502n Standard, applying the methodology phases described also based on the IEEE829, which shares the necessary structure and steps applicable to the implementation of the tests to be carried out to the development of the project.

Agradecimientos

De parte de Jessica Daniela Romero Dorado y Fabian Alexis Mina Posu queremos agradecer, primero que todo agradecer a Dios por permitirnos llegar a esta gran etapa de nuestras vidas y habernos brindado todo el conocimiento y habilidades para superar todas las adversidades que se presentaron en el camino, Gracias a nuestras familias quienes nos apoyaron día a día y quienes nos dieron esa motivación y sacrificios durante nuestra etapa de desarrollo profesional, gracias a nuestra directora de tesis la Ing. Beatriz Eugenia Marín quien nos apoyó y brindo todo su tiempo y conocimiento para lograr tan anhelada meta, agradecemos a la Universidad Antonio José Camacho quien nos formó, sencillo no ha sido el proceso pero gracias a las ganas y dedicación hemos logrado importantes objetivos como culminar con éxito nuestra tesis de grado y obtener nuestra titulación profesional.

Contenido

[Introducción 10](#_Toc32056043)

[1) Planteamiento del problema 11](#_Toc32056044)

[1.1) Pregunta de formulación del problema 12](#_Toc32056045)

[1.1.1) Preguntas de sistematización 12](#_Toc32056046)

[2) Objetivos 13](#_Toc32056047)

[2.1) Objetivo general 13](#_Toc32056048)

[2.2) Objetivos específicos 13](#_Toc32056049)

[3) Justificación 13](#_Toc32056050)

[3.1) Definición del alcance 15](#_Toc32056051)

[4) Marco referencial 15](#_Toc32056052)

[4.1) Antecedentes 15](#_Toc32056053)

[4.2) Marco conceptual 17](#_Toc32056054)

[4.3) Marco teórico 21](#_Toc32056055)

[4.3.1) Comparación Modelos de Calidad 22](#_Toc32056056)

[4.3.2) Ciclos de vida para pruebas de software. 27](#_Toc32056057)

[4.3.3) Tipos de herramientas para la elaboración de las pruebas 31](#_Toc32056058)

[4.4. Marco legal 32](#_Toc32056059)

[5) Metodología 33](#_Toc32056060)

[5.1) Fases para seguir 33](#_Toc32056061)

[5.1.1) Programación de pruebas 33](#_Toc32056062)

[5.1.2) Ejecución de pruebas 33](#_Toc32056063)

[5.1.3) Termino de las pruebas 34](#_Toc32056064)

[6) Desarrollo 34](#_Toc32056065)

[6.1) Preparación de pruebas 34](#_Toc32056066)

[6.1.1) Plan de pruebas 34](#_Toc32056067)

[6.1.1.1) Definición del alcance 34](#_Toc32056068)

[6.1.1.2) Definición de los recursos 35](#_Toc32056069)

[6.1.1.3) Criterios de calidad 38](#_Toc32056070)

[6.1.1.4) Definición de ítems a probar 39](#_Toc32056071)

[6.1.1.5) Tipos de pruebas escogidas para validar los criterios de calidad. 42](#_Toc32056072)

[6.1.1.6) Riesgos 42](#_Toc32056073)

[6.1.2) Especificaciones del diseño de pruebas 43](#_Toc32056074)

[6.1.2.1) Lista de requerimientos 43](#_Toc32056075)

[6.1.2.2) Estrategias 44](#_Toc32056076)

[6.1.2.3) Diagramas de casos de uso 45](#_Toc32056077)

[6.1.2.4) Casos de uso 49](#_Toc32056078)

[6.1.3) Especificaciones de casos de pruebas 101](#_Toc32056079)

[6.1.3.1) Pruebas de usabilidad 101](#_Toc32056080)

[6.1.3.2) Pruebas funcionales: 102](#_Toc32056081)

[6.1.3.3) Pruebas eficiencia 103](#_Toc32056082)

[6.1.3.4) Pruebas seguridad 103](#_Toc32056083)

[6.1.4) Procedimiento de pruebas 104](#_Toc32056084)

[6.1.5) Reporte de transmisión de pruebas 105](#_Toc32056085)

[6.2) Ejecución de pruebas 107](#_Toc32056086)

[6.2.1) Log de pruebas 107](#_Toc32056087)

[6.2.1)1. Pruebas de usabilidad 107](#_Toc32056088)

[6.2.1)2. Pruebas de funcionalidad 110](#_Toc32056089)

[6.2.1)3. Pruebas unitarias 111](#_Toc32056090)

[6.2.1)4. Pruebas de carga 116](#_Toc32056091)

[6.2.2) Reporte de incidentes de pruebas 119](#_Toc32056092)

[6.3) Termino de pruebas 120](#_Toc32056093)

[6.3.1) Reporte de pruebas 120](#_Toc32056094)

[7) Conclusiones 122](#_Toc32056095)

[8) Bibliografía 123](#_Toc32056096)

Tabla de figuras

[Figura 1 ISO 25010 CALIDAD DEL PRODUCTO SOFTWARE de ISO/IEC 25010 24](#_Toc31900095)

[Figura 2 Fases estándar IEEE 829 30](#_Toc31900096)

[Figura 3 Diagrama general casos de uso 45](#_Toc31900097)

[Figura 4 Diagrama casos de uso validar usuario 46](#_Toc31900098)

[Figura 5 Diagrama casos de uso cuadro de tiempo 46](#_Toc31900099)

[Figura 6 Diagrama casos de uso gestionar rango 47](#_Toc31900100)

[Figura 7 Diagrama casos de uso gestionar estación 47](#_Toc31900101)

[Figura 8 Diagrama General casos de uso 48](#_Toc31900102)

[Figura 9 Test de usuario 98](#_Toc31900103)

[Figura 10 Formato registro de pruebas 99](#_Toc31900104)

[Figura 11 Ejemplo caso de uso 11 100](#_Toc31900105)

[Figura 12 Reporte transmisión de pruebas 101](#_Toc31900106)

[Figura 13 Diagrama herramienta usadas 102](#_Toc31900107)

[Figura 14 Logs de prueba 103](#_Toc31900108)

[Figura 15 pruebas Usabilidad 105](file:///C:\Users\Fabian\Downloads\Documento%20Final%20V1.3%20(2).docx#_Toc31900109)

[Figura 16 pruebas funcionales 106](#_Toc31900110)

[Figura 17 Resultado prueba unitaria componente región 108](#_Toc31900111)

[Figura 18 Resultado prueba unitaria componente ciudad 108](#_Toc31900112)

[Figura 19 Resultado prueba unitaria componente categoría 109](#_Toc31900113)

Índice de Tablas

[Tabla 1Comparación de modelos y estándares de calidad 25](#_Toc32422044)

[Tabla 2 Fases aplicación de pruebas 30](#_Toc32422045)

[Tabla 3 Definición proceso de pruebas 31](#_Toc32422046)

[Tabla 4 Herramienta elaboración de pruebas 34](#_Toc32422047)

[Tabla 5 Listado Requerimientos 44](#_Toc32422048)

[Tabla 6 Casos de uso Traspaso y comunicación 55](#_Toc32422049)

[Tabla 7 Casos de uso Procesamiento 56](#_Toc32422050)

[Tabla 8 Caso de uso 1 57](#_Toc32422051)

[Tabla 9 Caso de uso 2 59](#_Toc32422052)

[Tabla 10 Caso de uso 3 61](#_Toc32422053)

[Tabla 11 Caso de uso 4 63](#_Toc32422054)

[Tabla 12 Caso de uso 5 66](#_Toc32422055)

[Tabla 13 Caso de uso 6 68](#_Toc32422056)

[Tabla 14 Caso de uso 7 70](#_Toc32422057)

[Tabla 15 Caso de uso 8 72](#_Toc32422058)

[Tabla 16 Caso de uso 9 75](#_Toc32422059)

[Tabla 17 Caso de uso 10 78](#_Toc32422060)

[Tabla 18 Caso de uso 11 80](#_Toc32422061)

[Tabla 19 Caso de uso 12 82](#_Toc32422062)

[Tabla 20 Caso de uso 13 84](#_Toc32422063)

[Tabla 21 Caso de uso 14 85](#_Toc32422064)

[Tabla 22 Caso de uso 15 87](#_Toc32422065)

[Tabla 23 Caso de uso 16 89](#_Toc32422066)

[Tabla 24 Caso de uso 17 92](#_Toc32422067)

[Tabla 25 Caso de uso 18 94](#_Toc32422068)

[Tabla 26 Caso de uso 19 95](#_Toc32422069)

[Tabla 27 Caso de uso 20 97](#_Toc32422070)

[Tabla 28 Caso de uso 21 99](#_Toc32422071)

[Tabla 29 Caso de uso 22 102](#_Toc32422072)

[Tabla 30 Caso de uso 23 104](#_Toc32422073)

[Tabla 31 Caso de uso 24 106](#_Toc32422074)

[Tabla 32 Datos de prueba Eficiencia y Unitarias 126](#_Toc32422075)

[Tabla 33 Datos prueba de carga 128](#_Toc32422076)

[Tabla 34 Reporte de incidencias 149](#_Toc32422077)

[Tabla 35 Reporte de pruebas 152](#_Toc32422078)

Acronimos

Capítulo……………………………………………………………………………………….. cap.

Edición………………………………………………………………………………….............ed.

Edición revisada…………………………………………………………………….…….. ed. rev.

Editor (es)……………………………………………………………………………….Ed. (Eds.)

Traductor (es)………………………………………………………………….……Trad. (Trads.)

Sin fecha………………………………………………………………………………………...s.f.

Página (páginas)…………………………………………………………………………....p. (pp.)

Volumen (Volúmenes)……………………………………………………………….. Vol. (Vols.)

Número…………………………………………………………………………………….…núm

Parte…………………………………………………………………………………………….Pte

Suplemento……………………………………………………………………………………Supl

The Institute of Electrical and Electronics…………………………………………………...IEEE

International Organization for Standardization………………………………………………...ISO

International Electrotechnical Commission……………………………………………………IEC

# Introducción

En este proyecto se aplicó un modelo de calidad al aplicativo que será desarrollado para la empresa SAN AMBIENTE, el cual permite gestionar todos los datos ambientales que la empresa recoge en determinadas regiones, el modelo de calidad se aplicó antes y durante la etapa de desarrollo del software para así asegurar que todos los aspectos requeridos por el cliente se cumplan y se entregue al final un producto útil. El modelo de calidad incluye desde el levantamiento de requerimientos, metodología de trabajo, diseño de casos de prueba y resultados de casos de prueba aplicados, siguiendo los estándares de las normas ISO 25010 y la norma IEEE 829 para la ejecución de las pruebas a las cuales fue sometido el software antes de ser entregado al cliente.

Para ello se refinaron y se elaboraron herramientas como casos de uso, diagramas de casos de uso para tener una documentación más completa para la elaboración de los 8 documentos especificados por la norma IEEE 829 y así tener una mejor visión del desarrollo del aplicativo; mediante pruebas de usabilidad, pruebas funcionales y pruebas de carga como resultado se obtiene una documentación más detallada del aplicativo y una perspectiva para futuras mejoras a aplicativo.

Se estableció que para el componente validación de usuario no se ejecutaría el proceso de pruebas ya que este modulo aun no se encontraba en proceso de desarrollo, aun así se dejaron las pruebas documentadas a las cuales será sometido el componente, se deja claro que estas pruebas pueden tener modificaciones futuras.

Antes de ejecutar tanto proyecto como el desarrollo, se realizó una investigación y comparación de los diferentes modelos de calidad existentes para así al final seleccionar el modelo de calidad que mejor se ajustara en el desarrollo del aplicativo, los modelos y estándares de calidad comparados fueron para medir la calidad del producto fueron la norma ISO/IEC 25010, modelo de calidad de McCall y el modelo de calidad de FURPS, de esta comparación se seleccionó el modelo de calidad propuesto por la norma ISO 25010 ya que es una norma que se centra en la usabilidad y permite medir la calidad del producto en 8 criterios los cuales se ajustan en el desarrollo del proyecto, para la ejecución de la pruebas de software se realizó la comparación entre la norma IEEE-829 y la norma ISO/IEC29119, de las cuales se seleccionó la norma ISO-829 que define la documentación detallada que se debe llevar a cabo en el proceso de pruebas de software.

# Planteamiento del problema

La empresa SAN AMBIENTE ofrece soluciones integrales en productos y servicios que aporten al mejoramiento de las condiciones ambientales y medición de parámetros ambientales, orientados a satisfacer las necesidades y requerimientos de los clientes. Busca el mejoramiento continuo de sus procesos y la productividad de la organización a través del compromiso y la participación de los empleados, socios y proveedores, lo que les ha permitido ser reconocidos en el mercado por la seriedad, profesionalismo en la prestación del servicio.

El problema se basa que al no implementar un modelo de calidad en el desarrollo del aplicativo se desencadenaría una serie de inconformidades que afectarían el avance de este mismo, en las que se incluyen retrasos en los tiempo de entrega del producto, inconformidad del cliente, no habría rendimiento en las actividades, no se ejecutarían las pruebas del software antes de ser entregado al cliente y esto llevaría a que en caso de un error de código se tendría que hacer un replanteamiento y tardaría mucho más tiempo.

Se desarrolló un software para la empresa SAN AMBIENTE el cual permitirá gestionar y procesar los datos ambientales que se recolectan en los lugares donde la empresa se encuentre ubicada.

## Pregunta de formulación del problema

- ¿Cómo asegurar la calidad de todos los módulos en el proceso de desarrollo del aplicativo de la empresa San Ambiente?

### Preguntas de sistematización

- ¿Cuáles estrategias de calidad se pueden implementar en el ciclo de vida del aplicativo de comunicación y traspaso de la empresa San Ambiente?

- ¿Cómo se pueden validar los elementos de calidad alcanzados en el producto de San Ambiente al igual que la satisfacción del cliente?

# Objetivos

## Objetivo general

Aplicar un modelo de calidad al módulo de comunicación de información del sistema de gestión de datos ambientales de la empresa SAN AMBIENTE de Cali.

## Objetivos específicos

- Definir los elementos necesarios para asegurar la calidad en el proceso de desarrollo del producto.

- Desarrollar un plan de evaluación del producto de software.

- Generar el plan de evaluación sobre el producto y presentar sus resultados.

# Justificación

El presente trabajo se enfocó en modelar y aplicar un modelo de calidad y su respectivo plan de pruebas para la entrega de la aplicación gestión de datos ambientales para la empresa SAN AMBIENTE; el modelo de calidad aplicado en este proyecto sirvió para direccionar al equipo de desarrollo a cumplir con todos los objetivos y requerimientos establecidos, al igual que un plan de pruebas el analizo todos los mecanismos, acciones y herramientas para detectar errores o fallas en el aplicativo que será entregado. Para realizar estos controles se implementan estándares que deben cumplirse para entregar al final un producto de calidad.

Este trabajo servirá como una guía para la realización de proyectos relacionados con la aplicación del modelo de calidad y plan de pruebas a diferentes; se espera que con este proyecto se realice la entrega del aplicativo a la empresa SAN AMBIENTE cumplimento con todos los requisitos demandados por el cliente y lograr la satisfacción de este mismo.

Según Manuel Prego (Revista digital CW,2004), director general de ESI, considera que, conceptualmente, el auge de los procesos de calidad se debe a que el software está cada vez más “inmerso en los productos y servicios que producimos.

En las empresas de desarrollo de software, la tendencia que tienen es realizar el proceso de las pruebas de software en la última etapa para consolidar la calidad de su producto.

Cada actividad en el desarrollo de software debe ir distribuido a una actividad de pruebas de calidad, para esto se implementan los modelos de calidad los cuales sirven de guía para que el equipo de desarrollo tenga más claro el proceso que se debe seguir durante el desarrollo del software.

## Definición del alcance

Verificar el cumplimiento de las normas y estándares de calidad parametrizados por los requerimientos solicitados en el desarrollo, con el fin de que sea un aplicativo que cumpla su funcionalidad, se manejara la validación concreta de los estándares de calidad con planes de pruebas aplicados a este desarrollo obteniendo pruebas exactas de tolerancias a fallos y de más criterios manejados en las normas de calidad implementadas.

# Marco referencial

## Antecedentes

El objetivo de esta sección es mostrar trabajos relacionados con la normativa de gestión de calidad ISO/IEC 2501n que habla sobre la división de modelo de calidad y actualmente se dividen de la siguiente forma la ISO/IEC 25010 system and software quality models, y ISO/IEC 25012 Data Quality model; lo cual en las investigaciones anteriores se tiene presente la calidad de uso y un modelo en general para la calidad, para todos los datos que se almacenan de una forma estructurada.

En (Callejas-Cuervo, Alarcón-Aldana, & Álvarez-Carreño, 2017) Los modelos de calidad del software, un estado del arte para la universidad pedagógica y tecnológica de Colombia, en el cual se pretende explicar los términos propios de calidad de software y los diferentes modelos de calidad aplicados para la revisión de calidad, realizando la clasificación de modelos acorde a (Proceso, producto y uso) existen varios modelos de calidad los cuales realizan procesos de mejora continua con su implementación, debido que para que un sistema informático complete un ciclo de vida debe tener características eficientes para así garantizar como conclusión se evidencio que en base a modelos anteriores han sido de beneficio para los más actuales, las empresas desarrolladoras deben someter su software a certificación que este bajo una norma o estándar de calidad.

En Estévez (2013) la investigación: llamada modelo de calidad para evaluar el software desarrollado en el centro de investigación aplicada y desarrollo en tecnologías de información CIADTI, donde su principal enfoque se hizo en los modelos de calidad del software para los productos desarrollados en la universidad de pamplona ya que un software sometido a un proceso de calidad es más apetecible para el mercado a la hora de realizar ventas, como resultado les fue posible establecer que no había una herramienta o proceso que permitiera evaluar la calidad del software, dado esto utilizaron varios modelos que se diferenciaban sobre los parámetros que iban a ser evaluados y tomaron en cuenta para la realización el diagnostico el diseño y aplicación.

En Scalone (2006) el estudio comparativo de los modelos y estándares de calidad del software, la forma de escoger el modelo o estándar de calidad adecuado que permita hacer determinación de los mejores resultados evitando que haya una mala inversión en recursos, tiempo y costos, se enfocó en las empresas que hacen desarrollo y que están interesadas en someterse a un proceso de certificación de calidad y así aumentar la competencia que ayudara a acceder a nuevos mercados. esto se encuentra en un contexto económico y empresarial que influencia a la hora de determinar un modelo o estándar de calidad adecuado como conclusión se encontró que el software maneja un papel fundamental para el desarrollo de las empresas esto quiere decir que es fundamental aplicar un modelo de calidad para evaluar un software.

En Ramos, Quintana (2014) el tema de modelo de mejora de procesos para la calidad del software basado en CMMI para una entidad financiera, donde se encuentra la inadecuada gestión de los requisitos para el desarrollo de los proyectos generando cuellos de botella, se encuentra una sobrecarga de trabajo que es inadecuado y no permite la entrega de un proceso con eficiencia, se hicieron mejoras donde los requisitos de gestión en un 82.55% lo cual permitió asegurar la alineación entre ellos, se desarrolló un proceso de monitoreo y control de los proyectos, se hizo mejora en 90%, lo cual permitirá ofrecer y dar soporte a la entrega de productos con calidad, se cumplió con los objetivos principales del desarrollo propuesto de CMMI nivel 2 mejorando todos los procesos en un 65.47% en el ciclo de vida del desarrollo del software.

## Marco conceptual

Este proyecto se realizó tomando como referencia el proyecto de grado, “Implementación de pruebas de calidad en la aplicación currículo de la Institución Universitaria Antonio José Camacho” (Aguado,2019), al cual se aplicó un modelo de calidad y un modelo de pruebas de calidad durante la etapa de desarrollo para prevenir errores a largo plazo. Lo anterior se basa en pruebas unitarias, pruebas en la base de datos, análisis de errores en el código y en la documentación de las fases del desarrollo.

* Adecuación Funcional

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Completitud funcional.

Corrección funcional.

Pertinencia funcional.

* Eficiencia de desempeño

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Comportamiento temporal.

Utilización de recursos.

Capacidad.

* Compatibilidad

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Coexistencia.

Interoperabilidad.

* Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Capacidad para reconocer su adecuación.

Capacidad de aprendizaje.

Capacidad para ser usado

Protección contra errores de usuario.

Estética de la interfaz de usuario.

Accesibilidad.

* Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

* Madurez.

Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.

Disponibilidad.

Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.

* Tolerancia a fallos.

Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.

* Capacidad de recuperación.

Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

## Marco teórico

El presente trabajo se realizó con el fin de aplicar un modelo de calidad para el módulo de comunicación y traspaso de información del sistema de gestión de datos ambientales de la empresa SAN AMBIENTE teniendo en cuenta que en su etapa de desarrollo es de gran magnitud definir las metodologías que se apicaran, para que sea una herramienta útil que pueda validar los procesos que se han ido desarrollando acorde a los aspectos funcionales;

Se aplico un modelo de calidad que contribuya con las especificaciones que se hayan solicitado por parte de la empresa SAN AMBIENTE ya que el manejo de la plataforma es de gran ayuda para llevar un control sobre los documentos de forma ordenada.

El desarrollo de aplicaciones web ha permitido hacer de manera más eficiente la automatización de procesos que antes requerían mucha demanda de tiempo lo cual retrasaba la ejecución de tareas, por lo tanto, se requiere el análisis y evaluación de los aplicativos que se vayan a entregar o a modificar debido a que un software con calidad tiene más rentabilidad y confiabilidad por parte del cliente que desea adquirirlo ya que este debe cumplir con todos los requerimientos funcionales y de rendimiento solicitados.

La aplicación de las pruebas de calidad se trabajó con el fin de mostrar la forma efectiva de medir la calidad del aplicativo en cada etapa de su desarrollo para así brindar a sus clientes servicios y productos de alto nivel demandado.

Para este proyecto se tuvo en cuenta que la Norma más Completa para aplicar el control de calidad es la ISO 25000 conocida como SQuaRE (System and Software Quality Requirements and Evaluation), el cual es una familia de normas que tiene por objetivo evaluar la calidad del producto software.

### Comparación Modelos de Calidad

Se realizo la comparación entre tres estándares y modelos de calidad, existe una amplia gama de modelos de calidad por lo cual es muy difícil seleccionar uno de ellos, sería favorable determinar cuáles modelos o estándares poseen una visión más integradora de la calidad del producto de software. Para este proyecto se realizó la comparación entre la norma ISO/IEC 25010, el modelo de McCall y el modelo de FRUPS.

En la siguiente tabla se compararon los tres modelos y estándares en cuanto a las características y criterios de calidad que cumple cada estándar.

Después de realizar la comparación se decidido trabajar con los criterios de calidad de la norma ISO/IEC 25010.

Tabla 1Comparación de modelos y estándares de calidad

|  |  |  |  |
| --- | --- | --- | --- |
| Características de Calidad/Modelos o Estándares de Calidad de Software | ISO/IEC 25010 | McCall | FRUPS |
| 1. Funcionalidad Adecuación funcional | X |  | X |
| 2. Usabilidad o Facilidad de uso | X | X | X |
| 3. Integridad o Seguridad | X | X |  |
| 4. Corrección, Precisión o Exactitud |  | X |  |
| 5. Confiabilidad o fiabilidad | X | X | X |
| 6. Eficiencia o Rendimiento | X | X | X |
| 7. Facilidad de mantenimiento | X | X |  |
| 8. Facilidad de prueba |  | X |  |
| 9. Flexibilidad, mutabilidad, Facilidad de modificación, Facilidad de cambio |  | X |  |
| 10. Portabilidad o Facilidad de trasportación | X | X |  |
| 11. Compatibilidad | X |  |  |
| 12. Soporte o Facilidad de soporte |  |  | X |
| Total | 8 | 9 | 5 |

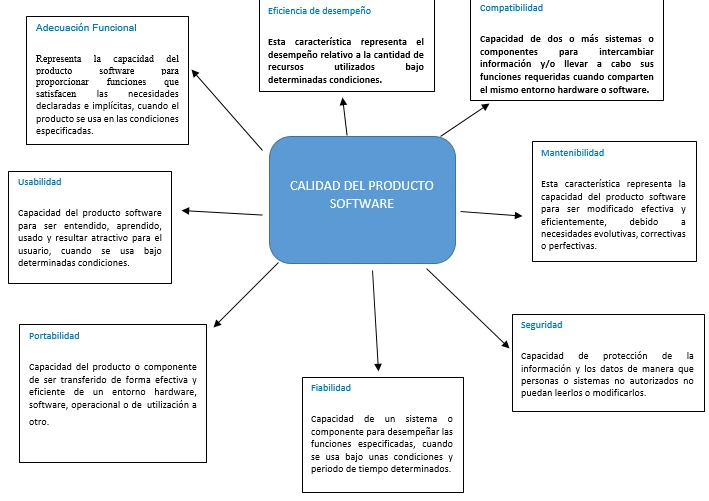
Autor: Daniela Romero-Fabian Mina

NORMA ISO/IEC 25010

En este modelo de calidad se determinaron las características de calidad que se evaluaron a un producto de software determinado. Se tuvieron en cuenta aspectos como la adecuación funcional, eficiencia de desempeño, la compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad, criterios que se evalúan conforme a los requerimientos solicitados por los usuarios los cuales tienen un valor referente a evaluar para así garantizar la

calidad de las características y subcaracterísticas del producto.

Figura 1 ISO 25010 CALIDAD DEL PRODUCTO SOFTWARE de ISO/IEC 25010



Nota. Recuperado de <https://modelos-de-calidad-para-evaluar-red.fandom.com/es/wiki/Modelo_De_Calidad_ISO/IEC_25010_-_ISO/IEC_25001>

La ISO/IEC 25010 en marca la calidad en los productos software, garantizando también la articulación de los procesos para obtener los productos, es una referencia óptima para garantizar la calidad del producto

Adecuación Funcional: Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades cuando se usa en condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

* Completitud funcional. Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
* Corrección funcional. Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
* Pertinencia funcional. Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

Eficiencia: Asegura que la aplicación logre cumplir los objetivos, pero en un tiempo óptimo y sin retrasos.

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

* Comportamiento temporal. Los tiempos de respuesta y procesamiento.
* Utilización de recursos. Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
* Capacidad. Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

Usabilidad: Es la medida de la calidad de la experiencia que tiene un usuario cuando interactúa con un producto o sistema

Capacidad del producto software para ser entendido, aprendido, usado por el usuario bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

* Capacidad para reconocer su adecuación. Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
* Capacidad de aprendizaje. Capacidad del producto que permite al usuario aprender su aplicación.
* Capacidad para ser usado. Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
* Protección contra errores de usuario. Capacidad del sistema para proteger a los usuarios de hacer errores.
* Estética de la interfaz de usuario. Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
* Accesibilidad. Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Seguridad: Es la medida de la confidencialidad que tienen los archivos del aplicativo

Capacidad de protección de la información y los datos para que personas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

* Confidencialidad. Protección contra el acceso de datos e información no autorizados.
* Integridad. Componente para prevenir accesos o modificaciones no autorizados.
* No repudio. Demostrar las acciones o eventos que han tenido lugar, y no puedan ser repudiados posteriormente.
* Responsabilidad. Rastrear de forma inequívoca las acciones de una entidad.
* Autenticidad. Demostrar la identidad de un sujeto o un recurso

### Ciclos de vida para pruebas de software.

Se realizo la comparación entre dos normas para la aplicación y ejecución de las pruebas de software, las normas que con las que se realizó la comparación fueron la norma IEEE-829 que define 3 fases y 8 documentos para la realización de pruebas de software la norma ISO/IEC 29119 la cual define 5 fases.

En la siguiente tabla se relaciona cada una de las fases que cumple las normar IEEE-829 y la norma ISO/IEC para la aplicación y ejecución de pruebas de software.

Tabla 2 Fases aplicación de pruebas

|  |  |  |  |
| --- | --- | --- | --- |
| METODOLOGIA IEEE-829 | PRODUCTO | METODOLOGIA ISO / IEC29119 | PRODUCTO |
| Fase1: Preparación de pruebas | * Plan de pruebas * Especificación del diseño de pruebas * Especificación de casos de prueba * Procedimiento de pruebas * Reporte de transmisión de ítems de pruebas | Fase1: Definiciones y Vocabulario | Aplica Fases, pero no relaciona un producto |
| Fase 2: Proceso de pruebas |
| Fase3: Documentación de prueba |
| Fase2: Ejecución de las pruebas | * Log de pruebas * Reporte de incidentes de pruebas |
| Fase3: Termino de las pruebas | * Reporte de las pruebas | Fase4: Técnicas de Prueba |
|  | | Fase5: Pruebas dirigidas por palabras clave |

Autor: Fabian Mina – Daniela Romero

En la siguiente tabla se define cada uno de los documentos con los cuales la norma IEEE-829 específica para la aplicación y ejecución de pruebas de software.

Tabla 3 Definición proceso de pruebas

|  |
| --- |
| Plan de pruebas: Es un documento eje sobre el cual se desarrollan las pruebas, se define el alcance, el enfoque de recursos y también identifica claramente los ítems y características a probar (herramientas y que se va a utilizar para el plan de pruebas) |
| Especificación del diseño de pruebas: Se determina qué necesita ser probado y cómo sería una prueba exitosa. |
| Especificación de casos de prueba: Valores exactos de entrada y otros que se requieran.  Valores exactos de salida y cambios del sistema esperados. (resultados que debería de ingresar y dar)  Se especifican pasos para ejecutar las pruebas y identificar las tareas de pruebas a desarrollar. |
| Procedimientos de pruebas: Describe cómo el tester ejecutará físicamente la prueba y los pasos necesarios. |
| Reporte de transmisión de ítems de pruebas: Describe los ítems para prueba, dónde encontrarlos y da la aprobación para su liberación.  Es importante porque garantiza de que los ítems están listos para ser probados. |
| Log de pruebas: Registra los detalles sobre qué casos de pruebas se han ejecutado, en qué orden y sus resultados.  Si hay inconformidades, se levanta o actualiza un reporte de incidentes |
| Reporte de incidentes de prueba: Descripción de los detalles encontrados cuando la  prueba no pasó. |
| Reporte de pruebas: Resume la información importante sobre las pruebas, incluyendo:  ● evaluación de qué tan bien se realizaron las pruebas,  ● número de incidentes reportados,  ● evaluación sobre la calidad del sistema  El documento es importante porque sirve para decidir si la calidad del  sistema es suficiente para continuar. |

Autor: Fabian Mina – Daniela Romero

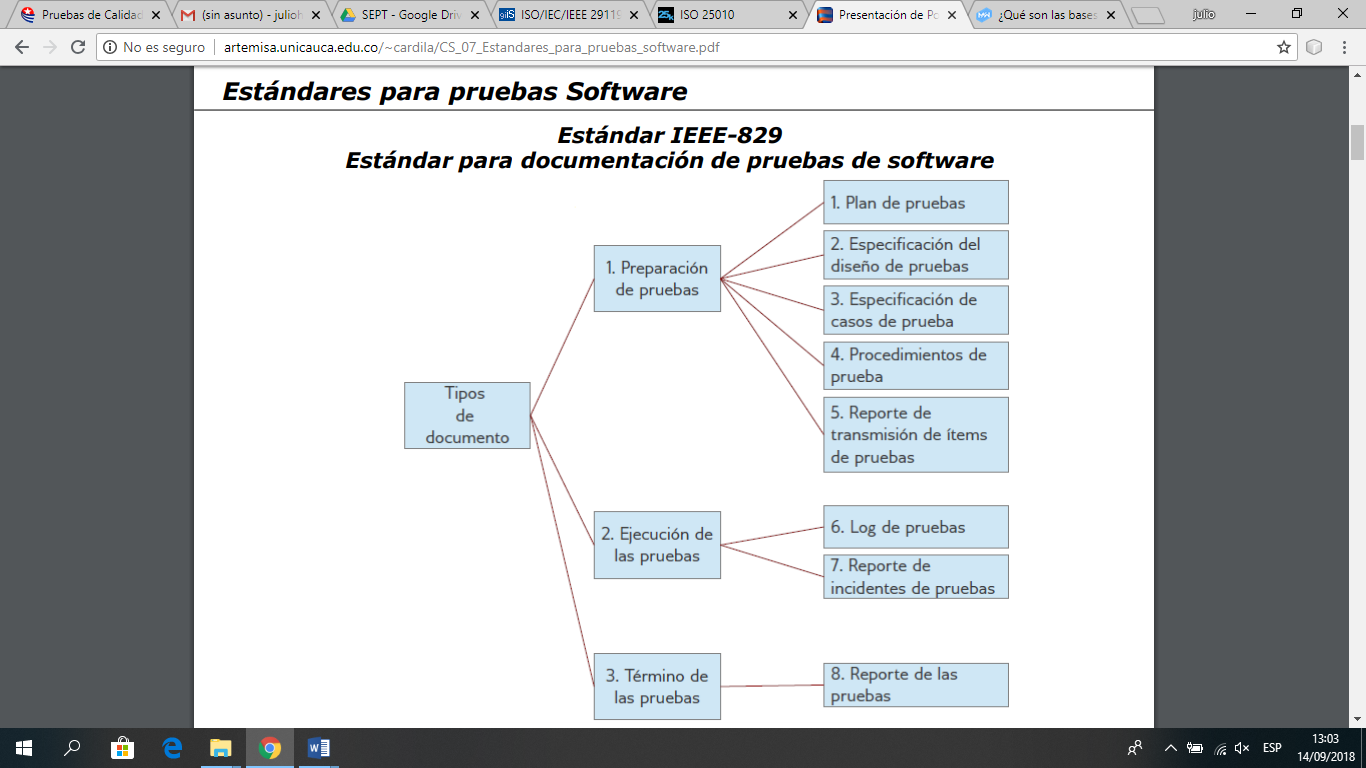
NORMA IEEE 829

El desarrollo de el plan de pruebas está basado por la norma IEEE 829-1983, la cual describe un estándar para la documentación de las pruebas de software que se muestran los tipos de documentos que pueden producirse durante un proceso de prueba.

El objetivo de la norma es proporcionar un conjunto estandarizado de pasos a seguir para la documentación de pruebas de software. Existen 8 tipos de pasos que pueden usarse en 3 etapas distintas de las pruebas de software.

El modelo de calidad del producto definido por la IEEE 829 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente Imagen:

Figura 2 Fases estándar IEEE 829



Nota. Recuperado de <https://docplayer.es/5992899-Aseguramiento-de-la-calidad-y-pruebas-de-software-5-pruebas-del-software-estandar-ieee-829-standard-for-software-test-documentation.html>

### Tipos de herramientas para la elaboración de las pruebas

En todo proceso de aplicación y ejecución de pruebas de software es necesario utilizar una serie de herramientas para llevar a cabo dicho proceso, se realizan pruebas a nivel de base de datos, interfaz gráfica y pruebas de código.

En la siguiente tabla se relacionan las herramientas que fueron necesarias utilizar para la aplicación y ejecución de las pruebas de software

Tabla 4 Herramienta elaboración de pruebas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabla de tipos de herramientas para elaboración de pruebas | | | | |
| Tipo de prueba | Herramienta | Descripción | Adecuada para este proyecto | Facilidad de uso |
| Pruebas de código | Visual Studio | Identificar e interactuar con objetos gráficos | Si | Media |
| Visual Studio | Interfaz web | Si | Media |
| Jest | Pruebas unitarias | Si | Media |
| Jmeter | Prueba de stress | Si | Media |
| Análisis de errores de código | Visual Studio Code | Recorrido de código y análisis de errores | Si | Media |

Autor: Fabian Mina – Daniela Romero

Las pruebas realizadas en este proyecto se realizaron con las siguientes herramientas integrándolas para una correcta ejecución.

* PostgreSQL

Es un gestor de base de datos relacional orientado a objetos y libre, que va dirigido a una comunidad de desarrolladores, no tiene gestor de defectos haciendo muy difícil conocer el estado de sus defectos.

• Angular Material

Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

• Node.js

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript

* Visual Studio Code

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

* Jest

Es una plataforma de pruebas universal, la cual se adapta a cualquier marco o biblioteca de JavaScript, tiene como objetivo trabajar sin configuración en la mayoría de proyectos de JavaScript; jest se puede ejecutar en proyectos usando Angular, Babel, Node, Typescript, Read, Vue entre otros.

* JMeter

Es un proyecto de apache la cual puede ser utilizada como herramienta para pruebas de carga para analizar el rendimiento de una variedad de servicios, es utilizada para pruebas unitarias en conexión de base de datos, conexiones HTML y conexiones TCP genéricas para llevar simulaciones sobre cualquier recurso de software.

## Marco legal

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

# Metodología

En esta parte se observará la metodología a emplear para cumplir con los objetivos postulados anteriormente.

## Fases para seguir

### Programación de pruebas

* Plan de pruebas
* Especificación del diseño de pruebas
* Especificación de casos de prueba
* Procedimiento de pruebas
* Reporte de transmisión de ítems de pruebas

### Ejecución de pruebas

* Log de pruebas
* Reporte de incidentes de pruebas

### Termino de las pruebas

* Reporte de las pruebas
* Cierre

# Desarrollo

## Preparación de pruebas

### Plan de pruebas

De los siguientes elementos se compone el plan de pruebas:

• Definición del alcance

• Definición de los recursos

• Criterios de calidad

• Definición de ítems a probar

• Tipos de pruebas escogidas para validar los criterios

• Riesgos

#### Definición del alcance

Verificar el cumplimiento de las normas y estándares de calidad parametrizados por los requerimientos solicitados en el desarrollo, con el fin de que sea un aplicativo que cumpla su funcionalidad, se manejara la validación concreta de los estándares de calidad con planes de pruebas aplicados a este desarrollo obteniendo pruebas exactas de tolerancias a fallos y de más criterios manejados en las normas de calidad implementadas.

El documento de desarrollo de pruebas se elabora con el nombre de aplicación de modelo calidad para el módulo de comunicación de traspaso de la empresa SAN AMBIENTE, y en él se especifica paso a paso las etapas para que lograr hacer una prueba exacta definiendo el alcance, enfoque y recursos utilizados en la etapa de prueba, ítems a probar, herramientas que se usaran, alcance de cada prueba y los riesgos.

Como parte de ejecución de las pruebas se realizó primeramente el levantamiento de requerimientos de acuerdo a las necesidades especificadas por la empresa SAN AMBIENTE, priorizando cada requerimiento y se realizó el modelamiento de los casos de uso detallados para guiar al equipo de desarrollo en la ejecución del aplicativo.

#### Definición de los recursos

Las herramientas que se usaron definitivamente en el desarrollo de este proyecto fueron escogidas por que se ajustan tanto al desarrollo como a la correcta ejecución del proyecto, y son herramientas las cuales pueden ser usadas por cualquier usuario ya que son de fácil acceso y no son complejos a la hora de manejar e interpretar los resultados obtenidos.

Herramientas usadas para la funcionalidad del proyecto

* Visual Studio Code

Visual Studio fue necesario en este proceso de pruebas, ya que con esta herramienta se desplegó toda la lógica del proyecto con esta herramienta se pudo visualizar el código y las rutas de los componentes desarrollados del lado tanto del cliente como del servidor, en visual studio code se pudo estudiar y explorar el código fuente de todo el aplicativo.

* PostgreSQL 11

Es un gestor de base de datos relacional orientado a objetos y libre, que va dirigido a una comunidad de desarrolladores, se trata de un sistema de código abierto el cual permite la manipulación de acuerdo a las reglas del algebra lineal, los datos son almacenados en tablas de columnas y renglones, con el uso de llaves estas tablas se pueden relacionar unas a otras.

Con el gestor PostgreSQL se realizaron consultas directamente a la base de datos, se realizaron pruebas de carga simulando múltiples conexiones de usuarios en un tiempo determinado, verificando el comportamiento y rendimiento del sistema a estas consultas generadas.

* Angular

El aplicativo fue desarrollado en typescript, por lo que fue necesario instalar librerías del framework Angular para así poner en funcionamiento el aplicativo y todas sus interfaces; lo primero que se hizo una vez teniendo el código fue instalar las Librerías mediante el comando npm isntall; con este comando se instaló angular, typescript, por el lado del cliente se instalaron las librerías para Bootstrap, JQuery y poppers, una vez instaladas las librerías con el comando ng serve se ejecuta el aplicativo.

Este un framework para aplicaciones web fue necesario para hacer pruebas en la interfaz gráfica y todas las vistas del aplicativo que se desarrollo

* Jest

La herramienta Jets fue utilizada para escribir cada una de las pruebas unitarias, a los métodos que fueron sometidos al respectivo proceso de pruebas. Para configurar jest fue necesario instalar todas las librerías mediante el comando npm install jest, typescript crea archivos que permitieron escribir las pruebas unitarias estos archivos terminan con la extensión .ts

* JMeter

La herramienta Jmeter se utilizó junto con la herramienta PostgreSQL para realizar la carga simulada de usuarios a la base de datos, se realizó mediante la configuración del puerto 4200, el servidor o ip local que fue localhost, protocolo http y la ruta de cada una de las vistas del aplicativo.

#### Criterios de calidad

Para el plan de pruebas de este proyecto se tomaron los siguientes criterios de calidad según la norma ISO 25010

Adecuación Funcional: Se propusieron las pruebas unitarias, para evaluar fragmentos de código para validar que se cumpla la funcionalidad.

Para la adecuación funcional se realizarán las siguientes pruebas:

* Pruebas Unitarias
* Pruebas de casos de uso

Eficiencia: Se plantearon las pruebas de eficiencia con el fin de evaluar que el aplicativo cumpla con un tiempo de respuesta al momento de ser desplegado, de igual forma se propusieron las pruebas de carga a la base de datos simulando múltiples conexiones de usuarios al mismo tiempo.

Para la eficiencia dentro del software se realizarán las siguientes pruebas:

* Pruebas de carga o consulta masiva
* Tiempos de respuesta del aplicativo

Para desarrollar las pruebas de eficiencia se propuso el siguiente proceso, en el que se tuvo en cuenta los tiempos de respuesta de la interfaz de cada uno de los componentes, para los componentes

Ciudad, Región, Categoría, Rango, Estaciones de tiempo se tomó el tiempo de ingreso a la interfaz de crear, consultar, actualizar, verificando que el tiempo no excediera los 5 segundos en cargar el contenido y mostrar los mensajes configurados en el sistema.

Usabilidad: En este ítem se planteó evaluar el comportamiento que tiene un usuario cuando interactúa con un producto o sistema; para la elaboración de pruebas de usabilidad se realizaron Test para usuarios donde se mostrarán los resultados de la revisión y el refinamiento del trabajo realizado, dentro del test de prueba se evalúa si el usuario al momento de interactuar con el aplicativo logra encontrar cada una de las opciones y funcionalidades con las que cuenta el sistemas, como diligenciar el formulario, guardar la información, buscar la opción crear nuevo, consultar la información; este test sea aplico para cada uno de los módulos desarrollados Ciudad, Región, Categoría, Rango, Estaciones de tiempo.

Seguridad: Las pruebas de seguridad se dejaron planteadas ya que el módulo de validación del usuario se encontraba en etapa de desarrollo cuando se ejecutó este proyecto; se plantearon pruebas de protección de la información y los datos para que personas no autorizados no puedan acceder, leer o modificar información.

Para la seguridad dentro del software se plantearon pruebas para validar la autenticación de los usuarios al sistema se realizarán las siguientes pruebas:

* Pruebas de casos de uso
* Pruebas Unitarias.

#### Definición de ítems a probar

Antes de iniciar el proceso de desarrollo y aplicación de pruebas fue importante definir y levantar los requerimientos funcionales los cuales la empresa SAN AMBIENTE requiere para tener un producto útil al final del desarrollo; los requerimientos que se definieron y fueron organizados por orden de prioridad se relacionan en la siguiente tabla.

Tabla 5 Listado Requerimientos

|  |  |
| --- | --- |
| # | Transferencia y comunicación |
| RQ1 | El sistema debe permitir el enlace y transferencia de datos de los dataloger mediante Internet. |
| RQ2 | El sistema debe permitir crear, modificar, borrar, visualizar comandos para ser utilizados por los protocolos de comunicación X y Y. |
| RQ3 | 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. |
| RQ4 | El sistema debe permitir establecer los parámetros necesarios para garantizar la comunicación vía FTP como cliente o servidor. |
| RQ5 | El sistema debe permitir crear plantillas para leer los archivos planos y cargar los datos al sistema. |
| RQ6 | El sistema debe permitir usar las plantillas de textos planos de FTP para otras estaciones. |
| RQ7 | 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. |
| RQ8 | El sistema debe permitir establecer los parámetros necesarios para garantizar la comunicación vía Modbus como cliente o servidor. |
| RQ9 | El sistema debe permitir configurar los comandos de módbus para acceder a los datalogers y obtener la información de estos. |
| RQ10 | 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. |
| RQ11 | El sistema debe permitir usar las plantillas comandos de Modbus para otras estaciones. |
| RQ12 | El sistema debe permitir subir textos planos de manera manual cuando una estación no tenga acceso a Internet |

Nota: Autor: Fabian Mina – Daniela Romero

• Interfaz gráfica: Se tendrá en cuenta cada una de las vistas del aplicativo, las vistas a las cuales se les realizaron pruebas fueron las siguientes:

|  |
| --- |
| Crear región |
| Crear ciudad |
| Crear mantenimiento |
| Crear Alerta |
| Crear Organización |
| Crear Categoría |
| Validar Usuario |
| Crear estación |
| Conectar con estación |
| Crear Rango |
| Crear base de tiempo |

Pruebas de usabilidad: Se prueba el ítem mediante encuestas realizadas a diferentes usuarios ajenos al software desarrollado. Se Realizaron pruebas a los siguientes módulos:

|  |
| --- |
| Crear región |
| Crear ciudad |
| Crear mantenimiento |
| Crear Alerta |
| Crear Organización |
| Crear Categoría |
| Crear estación |
| Crear rango |
| Login\* |

Dentro de cada prueba realizada se evaluaron las siguientes tareas:

|  |
| --- |
| Selecciona la opción crear |
| Completar el formulario de creación |
| Guardar datos ingresados |
| Consultar Ítem ingresado |

Pruebas Eficiencia: Se prueba el comportamiento del aplicativo y los tiempos de respuesta a las peticiones realizadas por los usuarios.

• lógica del aplicativo:

Pruebas Unitarias: Se prueban métodos de clases, por ejemplo: métodos get/set, se realizarán pruebas en las líneas de código validando que el método codificado reciba y entregue valores correctos e incorrectos. a continuación, se listan los métodos a los cuales se les realiza pruebas

|  |
| --- |
| Regions\_Component |
| Cities\_Component |
| Maintenances\_Component |
| Alerts\_Component |
| Organizations\_Component |
| Categories\_Component |
| Stations\_Component |
| Ranks\_Component |
| Times\_Component |

• Base de datos:

Pruebas de consultas masivas: Se realizaron pruebas de carga donde se simula la conexión de múltiples usuarios consultando la misma base de datos. Se realiza las pruebas conectando la herramienta Jmeter con el puerto 4200 que es donde se encuentra alojado el aplicativo, el servidor o la ip en este caso se utilizó el servidor localhost y la ruta de cada uno de los componentes ejemplo. /webpack-dev-server/category; se configura haciendo la consulta a los siguientes componentes

|  |
| --- |
| Regions\_Body.Component |
| Cities \_Body. Component |
| Alerts \_Body. Component |
| Organizations \_Body. Component |
| Categories \_Body. Component |
| Stations \_Body. Component |
| Ranks \_Body. Component |
| Times \_Body. Component |

Pruebas Eficiencia: Se prueba el comportamiento del aplicativo y los tiempos de respuesta a las peticiones realizadas por los usuarios.

Login (Seguridad): Para el módulo de seguridad se dejaron las pruebas planteadas ya que no se tuvo cobertura hasta este módulo, se espera que en futuras mejoras se abarque el módulo de validación del usuario, se propuso que este módulo se probará la validación de los datos, autenticación de los usuarios y encriptación de contraseñas en base de datos, se evitará que los datos personales como contraseñas no se ingresen en texto plano.

#### Tipos de pruebas escogidas para validar los criterios de calidad.

Con las siguientes pruebas se quiere satisfacer los criterios de calidad que se plantearan

• Pruebas Unitarias

Son pruebas las cuales se utilizan para verificar que un método o una clase funcionan correctamente, estas pruebas suelen realizarse durante la etapa de desarrollo de aplicaciones de software.

• Pruebas de usabilidad

Son pruebas las cuales evalúan el comportamiento del usuario frente al aplicativo desarrollado, con estas pruebas de usabilidad lo que se busca es buscar optimizar la experiencia de los usuarios frente al software, los resultados de estas pruebas de usabilidad pueden llevar a que se realicen modificaciones en cuanto a la interfaz, diseño o redacción del aplicativo.

• Pruebas eficiencia

Son pruebas las cuales determinan si el sistema desarrollado cumple o satisface con los requisitos de rendimiento, evalúa el sistema frente a condiciones de volúmenes altos de carga y concurrencia y permite detectar defectos funcionales que se producen bajo condiciones de estrés.

• Pruebas de seguridad

Son pruebas con las cuales se mide el nivel de vulnerabilidad, e instrucciones al sistema desarrollado, con estas pruebas se busca establecer que tan seguro es el sistema frente a ataques o accesos no autorizados al sistema.

#### Riesgos

Los riesgos que se tuvieron en este plan de pruebas son que al ejecutar las pruebas con las herramientas planteadas anteriormente no se esperen los resultados que se tienen planeados, ya que el volumen de la aplicación puede ser un poco grande para las herramientas, y que puede que no sean óptimas para ejecutarse con este tipo de proyectos o que al ser ejecutadas tenga errores en el código y dificulte el proceso de pruebas; otro de los riesgos importante que se tuvo en cuenta al momento de desarrollar este proyecto fue que no se lograra abarcar con totalidad los componentes del aplicativo a desarrollar.

### Especificaciones del diseño de pruebas

Para definir lo que se necesita ser probado se levantaron los requerimientos y se plantearon los casos de uso que se listan a continuación

|  |  |
| --- | --- |
| Caso de Uso | Nombre |
| 1 | Conectar con estación FTP |
| 2 | Conectar con estación Modbus |
| 3 | Crear Validación |
| 4 | Crear región (Tabla parámetro) |
| 5 | Crear estación (Tabla parámetro) |
| 6 | Crear organización (Tabla parámetro) |
| 7 | Crear categoría (Tabla parámetro) |
| 8 | Crear Ciudad (Tabla parámetro) |
| 9 | Crear rango (Tabla parámetro) |
| 10 | Crear alerta (Tabla parámetro) |
| 11 | Crear base de tiempo (Tabla parámetro) |
| 12 | Validar Usuario (login) |
| 13 | Modificar datos descargados |
| 14 | Dar de baja datos procesados |
| 15 | Conectar Automáticamente con estación |
| 16 | Crear Mantenimiento |

Pruebas Unitarias, pruebas funcionales

|  |  |  |
| --- | --- | --- |
| **Funcionalidad** | **Elementos a Probar** | **Resultados Esperados** |
|
|
| crear región | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de crear región, el tiempo de repuesta sea inferior y como máximo a 5 segundo, para ingresar y cargar su contenido. |
| crear región | **Campos nombre, Descripción** | Se espera que el aplicativo, al momento de ingresar datos correctos se activen los mensajes de validación, mostrando mensaje en pantalla, el aplicativo deberá seguir el curso normal. |
| crear región | **Campos nombre, Descripción** | Se espera que el aplicativo, al momento de ingresar datos incorrectos se activen los mensajes de validación, mostrando mensaje en pantalla, al ingresar datos correctos el aplicativo deberá seguir el curso normal. |
| crear región | **Botón Guardar** | Se espera que el aplicativo almacene la información diligenciada correctamente en base de datos, se muestren los mensajes de confirmación en pantalla. |
| Crear región | **Botón Editar** | Se espera que al momento de seleccionar la opción editar, el aplicativo deberá enviar al usuario a la pantalla de editar y habilitar los campos nombre y descripción |
| crear región | **Método crear** | al ejecutar el test de la prueba unitaria deberá ser positivo, realizando sincronización interna con el servidor y cargando el método |
| crear región | **Método consultar** | al ejecutar el test de la prueba unitaria deberá ser positivo, realizando sincronización interna con el servidor y cargando el método consultar |
| crear región | **Método actualizar** | Al ejecutar el test de la prueba unitaria se deberá actualizar la región creada con la información actual |
| crear ciudad | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de crear Ciudad, el tiempo de repuesta sea inferior y como máximo a 5 segundo, para ingresar y cargar su contenido. |
| crear ciudad | **Campos nombre, Descripción** | Se espera que el aplicativo, al momento de ingresar datos correctos se activen los mensajes de validación, mostrando mensaje en pantalla, el aplicativo deberá seguir el curso normal. |
| crear ciudad | **Campos nombre, Descripción** | Se espera que el aplicativo, al momento de ingresar datos incorrectos se activen los mensajes de validación, mostrando mensaje en pantalla, al ingresar datos correctos el aplicativo deberá seguir el curso normal. |
| crear ciudad | **Botón Guardar** | Se espera que el aplicativo almacene la información diligenciada correctamente en base de datos, se muestren los mensajes de conformación en pantalla. |
| Crear ciudad (Tabla Parámetro) | **Opción Región** | Verificar que el aplicativo permita seleccionar las regiones creadas anteriormente. |
| crear ciudad | **Método agregar** | al ejecutar el test de la prueba unitaria deberá realizar el ciclo con el servidor y dar como resultado positivo |
| crear ciudad | **Método consultar** | El ejecutar el test de la prueba unitaria, deberá mostrar en pantalla, la consulta realizada para la Ciudad creada anteriormente |
| crear ciudad | **Método actualizar** | Al ejecutar el test de la prueba unitaria se deberá actualizar la Ciudad creada con la información actual |
| crear Mantenimiento | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de crear Mantenimiento, el tiempo de repuesta sea inferior y como mínimo a 5 segundo, para ingresar y cargar su contenido. |
| crear Mantenimiento | **Campos nombre, fecha, estación, observación** | Se espera que el aplicativo, al momento de ingresar datos correctos se activen los mensajes de validación, mostrando mensaje en pantalla, el aplicativo deberá seguir el curso normal. |
| crear Mantenimiento | **Campos nombre, fecha, estación, observación** | Se espera que el aplicativo, al momento de ingresar datos incorrectos se activen los mensajes de validación, mostrando mensaje en pantalla, al ingresar datos correctos el aplicativo deberá seguir el curso normal. |
| crear Mantenimiento | **Botón Guardar** | Se espera que el aplicativo almacene la información diligenciada correctamente en base de datos, se muestren los mensajes de conformación en pantalla. |
| crear Mantenimiento | **Método agregar** | al ejecutar el test de la prueba unitaria deberá imprimir mensaje en pantalla "Guardado con éxito" |
| crear Mantenimiento | **Método consultar** | El ejecutar el test de la prueba unitaria, deberá mostrar en pantalla, la consulta realizada para la Mantenimiento creada anteriormente |
| crear Mantenimiento | **Método actualizar** | Al ejecutar el test de la prueba unitaria se deberá actualizar la Mantenimiento creada con la información actual |
| crear Mantenimiento | **Método consultar Id** | Al ejecutar el test de la prueba unitaria se deberá consultar la información de la Mantenimiento creada con ese Id |
| crear Alerta | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de crear Alerta, el tiempo de repuesta sea inferior y como mínimo a 5 segundo, para ingresar y cargar su contenido. |
| crear Alerta | **Campos nombre, estación, Observación** | Se espera que el aplicativo, al momento de ingresar datos correctos se activen los mensajes de validación, mostrando mensaje en pantalla, el aplicativo deberá seguir el curso normal. |
| crear Alerta | **Campos nombre, estación, observación** | Se espera que el aplicativo, al momento de ingresar datos incorrectos se activen los mensajes de validación, mostrando mensaje en pantalla, al ingresar datos correctos el aplicativo deberá seguir el curso normal. |
| crear Alerta | **Botón Guardar** | Se espera que el aplicativo almacene la información diligenciada correctamente en base de datos, se muestren los mensajes de conformación en pantalla. |
| crear Alerta | **Método agregar** | al ejecutar el test de la prueba unitaria deberá imprimir mensaje en pantalla "Guardado con éxito" |
| crear Alerta | **Método consultar** | El ejecutar el test de la prueba unitaria, deberá mostrar en pantalla, la consulta realizada para la Alerta creada anteriormente |
| crear Alerta | **Método actualizar** | Al ejecutar el test de la prueba unitaria se deberá actualizar la Alerta creada con la información actual |
| crear Alerta | **Método consultar Id** | Al ejecutar el test de la prueba unitaria se deberá consultar la información de la Alerta creada con ese Id |
| crear Organización | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de crear Organización, el tiempo de repuesta sea inferior y como mínimo a 5 segundo, para ingresar y cargar su contenido. |
| crear Organización | **Campos nombre, observación** | Se espera que el aplicativo, al momento de ingresar datos correctos se activen los mensajes de validación, mostrando mensaje en pantalla, el aplicativo deberá seguir el curso normal. |
| crear Organización | **Campos nombre, observación** | Se espera que el aplicativo, al momento de ingresar datos incorrectos se activen los mensajes de validación, mostrando mensaje en pantalla, al ingresar datos correctos el aplicativo deberá seguir el curso normal. |
| crear Organización | **Botón Guardar** | Se espera que el aplicativo almacene la información diligenciada correctamente en base de datos, se muestren los mensajes de conformación en pantalla. |
| crear Organización | **Método agregar** | al ejecutar el test de la prueba unitaria deberá imprimir mensaje en pantalla "Guardado con éxito" |
| crear Organización | **Método consultar** | El ejecutar el test de la prueba unitaria, deberá mostrar en pantalla, la consulta realizada para la Organización creada anteriormente |
| crear Organización | **Método actualizar** | Al ejecutar el test de la prueba unitaria se deberá actualizar la Organización creada con la información actual |
| crear Organización | **Método consultar Id** | Al ejecutar el test de la prueba unitaria se deberá consultar la información de la Organización creada con ese Id |
| crear Categoría | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de crear Categoría, el tiempo de repuesta sea inferior y como mínimo a 5 segundo, para ingresar y cargar su contenido. |
| crear Categoría | **Campos nombre, observación** | Se espera que el aplicativo, al momento de ingresar datos correctos se activen los mensajes de validación, mostrando mensaje en pantalla, el aplicativo deberá seguir el curso normal. |
| crear Categoría | **Campos nombre, observación** | Se espera que el aplicativo, al momento de ingresar datos incorrectos se activen los mensajes de validación, mostrando mensaje en pantalla, al ingresar datos correctos el aplicativo deberá seguir el curso normal. |
| crear Categoría | **Botón Guardar** | Se espera que el aplicativo almacene la información diligenciada correctamente en base de datos, se muestren los mensajes de conformación en pantalla. |
| Crear Categoría | **Método agregar** | al ejecutar el test de la prueba unitaria deberá imprimir mensaje en pantalla "Guardado con éxito" |
| crear Categoría | **Método consultar** | El ejecutar el test de la prueba unitaria, deberá mostrar en pantalla, la consulta realizada para la Categoría creada anteriormente |
| crear Categoría | **Método actualizar** | Al ejecutar el test de la prueba unitaria se deberá actualizar la Categoría creada con la información actual |
| crear Categoría | **Método consultar Id** | Al ejecutar el test de la prueba unitaria se deberá consultar la información de la Categoría creada con ese Id |
| Login (Validación Usuario) | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de login, el tiempo de repuesta sea inferior y como mínimo a 5 segundo, para ingresar y cargar su contenido. |
| Login (Validación Usuario) | **Campos email, password** | Se espera que el aplicativo muestre mensaje de validación y no permita ingresar a la aplicación. |
| Login (Validación Usuario) | **Campos email, password** | Se espera que el aplicativo, al momento de ingresar datos incorrectos se activen los mensajes de validación, mostrando mensaje en pantalla, al ingresar datos correctos el aplicativo deberá seguir el curso normal. |
| Login (Validación Usuario) | **Botón Ingresar** | Se espera que el sistema valide los formatos de email, si son correctos se ingresara al aplicativo, si son incorrectos deberá mostrar mensaje de validación en pantalla. |
| Login (Validación Usuario) | **Encriptación de password** | Se espera que el sistema permita el ingreso al aplicativo con credenciales correctas |
| Login (Validación Usuario) | **Encriptación de password** | Se espera que el aplicativo internamente dentro del código realice la encriptación del texto obtenido en el campo password, se realizara por el método SHA1 o el método de encryptacion MD5 |
| crear Estación | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de crear Estación, el tiempo de repuesta sea inferior y como mínimo a 5 segundo, para ingresar y cargar su contenido. |
| crear Estación | **Campos nombre, nombre corto estación, serial, observación, localización, latitud, longitud, elevación** | Se espera que el aplicativo, al momento de ingresar datos correctos se activen los mensajes de validación, mostrando mensaje en pantalla, el aplicativo deberá seguir el curso normal. |
|  |  |  |
| crear Estación | **Campos nombre, nombre corto estación, serial, observación, localización, latitud, longitud, elevación** | Se espera que el aplicativo, al momento de ingresar datos incorrectos se activen los mensajes de validación, mostrando mensaje en pantalla, al ingresar datos correctos el aplicativo deberá seguir el curso normal. |
| crear Estación | **Botón Guardar** | Se espera que el aplicativo almacene la información diligenciada correctamente en base de datos, se muestren los mensajes de conformación en pantalla. |
| crear Estación | **Método agregar** | al ejecutar el test de la prueba unitaria deberá imprimir mensaje en pantalla "Guardado con éxito" |
|  |  |  |
| crear Estación | **Método consultar** | El ejecutar el test de la prueba unitaria, deberá mostrar en pantalla, la consulta realizada para la Estación creada anteriormente |
| Conectar con estación | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de conectar con estación, el tiempo de repuesta sea inferior y como mínimo a 5 segundo, para ingresar y cargar su contenido. |
| Conectar con estación | **Estación creada para conexión** | Se espera que el aplicativo cargue la información de la estación consultada. |
| Conectar con estación | **Botón Conectar** | Se espera que el aplicativo realice la confirmación de los campos obligatorios, en caso de presentarse error se mostrara mensaje en pantalla "Error de Conexión" |
| Conectar con estación | **Conexión con estación** | Verificar que se realice la conexión satisfactoriamente |
| Conectar con estación | **Protocolo de comunicación** | Verificar que el aplicativo se conecte satisfactoriamente, mediante protocolo configurado (FTP,modbus) |
| crear Alerta | **Tiempo de respuesta** | Se espera que el aplicativo al ingresar en la pantalla de crear Alerta, el tiempo de repuesta sea inferior y como mínimo a 5 segundo, para ingresar y cargar su contenido. |
| crear Alerta | **Campos nombre, estación, observación** | Se espera que el aplicativo, al momento de ingresar datos correctos se activen los mensajes de validación, mostrando mensaje en pantalla, el aplicativo deberá seguir el curso normal. |
| crear Alerta | **Campos nombre, estación, observación** | Se espera que el aplicativo, al momento de ingresar datos incorrectos se activen los mensajes de validación, mostrando mensaje en pantalla, al ingresar datos correctos el aplicativo deberá seguir el curso normal. |
| crear Alerta | **Botón Guardar** | Se espera que el aplicativo almacene la información diligenciada correctamente en base de datos, se muestren los mensajes de conformación en pantalla. |
| crear Alerta | **Método agregar** | al ejecutar el test de la prueba unitaria deberá imprimir mensaje en pantalla "Guardado con éxito" |
| crear Alerta | **Método consultar** | El ejecutar el test de la prueba unitaria, deberá mostrar en pantalla, la consulta realizada para la Alerta creada anteriormente |
| crear Alerta | **Método actualizar** | Al ejecutar el test de la prueba unitaria se deberá actualizar la Alerta creada con la información actual |
| crear Alerta | **Método consultar Id** | Al ejecutar el test de la prueba unitaria se deberá consultar la información de la Alerta creada con ese Id |

Tabla 6 Resultados esperados pruebas unitarias y funcionales

Pruebas de Carga

|  |  |
| --- | --- |
| **Elementos a Probar** | **Resultados Esperados** |
|
|
| **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad máxima de 100 usuarios conectándose simultáneamente al aplicativo |

Tabla 7 Resultados esperados pruebas de carga

#### Estrategias

Para las estrategias de este proyecto se utilizaron y refinaron diagramas de casos de uso y casos de uso detallados.

A partir de los diagramas de casos de uso y casos de uso detallados, se presenta a continuación:

#### Diagramas de casos de uso

Se refinaron los casos de uso y diagramas de caso de uso para orientar al equipo de desarrollo a tener una visión clara del software que fue desarrollado, es importante durante la etapa de desarrollo, que todos los miembros del equipo tengan un mismo enfoque sobre lo que se va a desarrollar para así disminuir los márgenes de error que se puedan presentar.

En el diagrama de casos de uso, se muestra el actor y las actividades que debe realizar en la aplicación desarrollada.

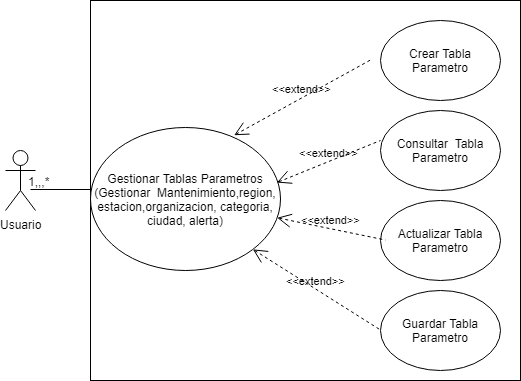


Figura 3 Diagrama general casos de uso

Autor: Fabian Mina -Daniela Romero

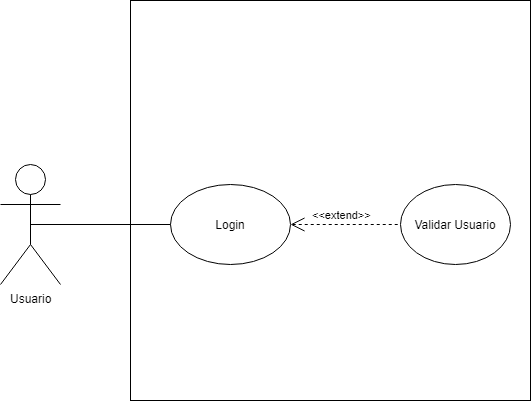


Figura 4 Diagrama casos de uso validar usuario

Autor: Fabian Mina -Daniela Romero

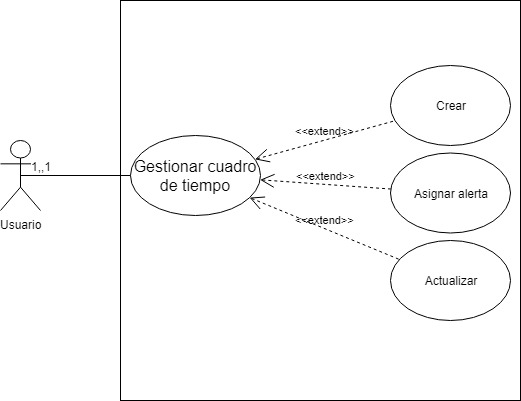


Figura 5 Diagrama casos de uso cuadro de tiempo

Autor: Fabian Mina -Daniela Romero

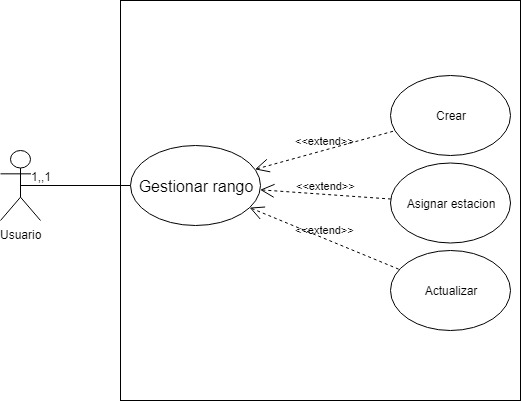


Figura 6 Diagrama casos de uso gestionar rango

Autor: Fabian Mina -Daniela Romero

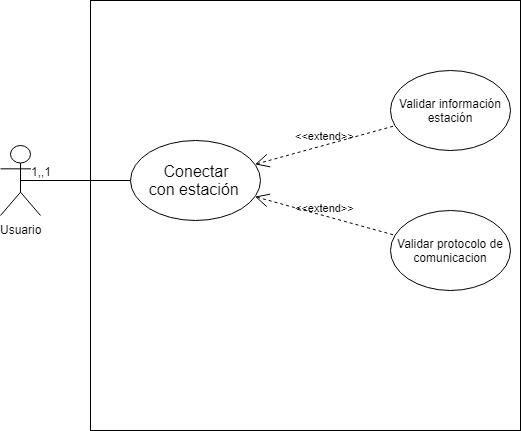


Figura 7 Diagrama casos de uso gestionar estación

Autor: Fabian Mina -Daniela Romero



Figura 8 Diagrama General casos de uso

Autor: Fabian Mina -Daniela Romero

#### Casos de uso

En los casos de uso, se muestra detalladamente las actividades o acciones que realiza actor y la respuesta que tiene el sistema para llevar a cabo algún proceso. Estos casos de uso orientan la funcionalidad que se va a desarrollar, se validó con el equipo de desarrollo el detalle de cada caso de uso.

Traspaso y comunicación

|  |  |
| --- | --- |
| **#CU** | **Nombre** |
| 1 | Conectar con estación FTP |
| 2 | Conectar con estación Modbus |
| 3 | Crear Validación |
| 4 | Crear región (Tabla parámetro) |
| 5 | Crear estación (Tabla parámetro) |
| 6 | Crear organización (Tabla parámetro) |
| 7 | Crear categoría (Tabla parámetro) |
| 8 | Crear Ciudad (Tabla parámetro) |
| 9 | Crear rango (Tabla parámetro) |
| 10 | Crear alerta (Tabla parámetro) |
| 11 | Crear base de tiempo (Tabla parámetro) |
| 12 | Validar Usuario (login) |
| 13 | Modificar datos descargados |
| 14 | Dar de baja datos procesados |
| 15 | Conectar Automáticamente con estación |
| 16 | Crear Mantenimiento |

Tabla 8 Casos de uso Traspaso y comunicación

Autor Daniela Romero -Fabian Mina

Procesamiento

|  |  |
| --- | --- |
| **#CU** | **Nombre** |
| 17 | Criterios de validación |
| 18 | Crear plantillas |
| 19 | Ver Plantillas |
| 20 | Crear usuario |
| 21 | Crear roles |
| 22 | Crear permisos |
| 23 | Crear estados |
| 24 | Asignar estados |

Tabla 9 Casos de uso Procesamiento

Autor Daniela Romero -Fabian Mina

**CASOS DE USO 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Conectar con estación FTP | | | | | CU1 |
| Actores | Administrador, Operador | | | | | |
| Referencias | Requerimiento RF 1 | | |  | | |
| Precondición | El sistema deberá estar conectado a la red de internet, la estación ya debe estar creada con anterioridad | | | | | |
| Postcondición | El sistema debe mostrar los datos básicos de la estación a la que se ha conectado | | | | | |
| Autor | Fabián Mina | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Conectarse mediante protocolo FTP a una estación |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
| 1 | El usuario selecciona el nombre de la estación. | **2** | El sistema muestra la información almacenada para ese nombre de estación.  -Id  -Serial  -Abreviación nombre corto de la -estación  -organización  -categoría  -región.  -Localización  -ciudad  -latitud y longitud  -elevación  -Base de tiempo |
| 3 | El usuario confirma la información botón conectar | **4** | El sistema se conecta a la estación seleccionada. |
|  |  | **5** | El sistema se conecta a la estación seleccionada y muestra mensaje en pantalla “Conexión Exitosa” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de que los datos no sean válidos o se presente algún error al momento de llenar el formulario, el sistema debe mostrar mensaje en pantalla, especificando el error que se presentó |
| 2 | Si no existe protocolo de comunicación con esta estación, se mostrará mensaje en pantalla “No existe protocolo de comunicación”. |

Tabla 10 Caso de uso 1

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Conectar con estación Modbus | | | | | CU2 |
| Actores | Administrador, Operador | | | | | |
| Referencias | Requerimiento RF 1 | | |  | | |
| Precondición | El sistema deberá estar conectado a la red de internet, la estación ya debe estar creada con anterioridad | | | | | |
| Postcondición | El sistema debe mostrar los datos básicos de la estación a la que se ha conectado | | | | | |
| Autor | Fabián Mina | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Conectarse mediante protocolo Modbus a una estación |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
| 1 | El usuario selecciona el nombre de la estación. | 2 | El sistema muestra la información almacenada para ese nombre de estación.  -Id  -Serial  -Abreviación nombre corto de la -estación  -organización  -categoría  -región.  -Localización  -ciudad  -latitud y longitud  -elevación  -Base de tiempo |
| 3 | El usuario confirma la información botón conectar | 4 | El sistema se conecta a la estación seleccionada. |
|  |  | 5 | El sistema se conecta a la estación seleccionada y muestra mensaje en pantalla “Conexión Exitosa” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de que los datos no sean válidos o se presente algún error al momento de llenar el formulario, el sistema debe mostrar mensaje en pantalla, especificando el error que se presentó |
| 2 | Si no existe protocolo de comunicación con esta estación, se mostrará mensaje en pantalla “No existe protocolo de comunicación”. |

Tabla 11 Caso de uso 2

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear Validación | | | | | | CU3 |
| Actores | Administrador, Operador | | | | | | |
| 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 | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | 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 | |
| 1 | El sistema intenta escribir la hora y los minutos: El campo está inhabilitado para escritura y solo permite la selección. |
| 2 | El usuario digita un valor alfanumérico: El campo solo permite valores numéricos por lo tanto no registra ningún valor diferente |
| 3 | En caso de que el usuario no calibre los datos, el sistema lo hará automáticamente. |
| 4 | 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 12 Caso de uso 3

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear región (Tabla parámetro) | | | | | | CU4 |
| Actores | Administrador, Operador | | | | | | |
| Referencias | Requerimiento RF 3 | | | |  | | |
| Precondición | El sistema deberá estar abierto en el menú de parametrización | | | | | | |
| Postcondición | La región queda creada en el sistema. | | | | | | |
| Autor | Fabián Mina | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Gestionar las regiones |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Creación de la región | | | |
| 1 | El usuario oprime el botón nueva región | 2 | El sistema carga el Id automático que corresponde a la nueva región y habilita los campos de la región. |
| 3 | El usuario digita el nombre de la región con una longitud máxima de 50 caracteres. |  |  |
| 4 | El usuario digita una descripción de la región con una longitud máxima de 50 caracteres. |  |  |
| 5 | El usuario oprime el botón guardar | 6 | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación de transacción: “Región creada correctamente” |
|  |  | 7 | El sistema muestra en el listado la nueva región creada. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 2 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 3 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de la región | | | |
| 1 | El usuario digita el nombre de la región o el Id de la región. | 2 | El sistema carga los datos de la región. |
| 3 | El usuario oprime el botón editar. | 4 | El usuario habilita los campos nombre y descripción para actualización |
| 4 | El usuario actualiza el campo nombre y descripción y oprime botón guardar | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación “Guardado exitosamente”. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje y el cursor se ubica en el campo. |

Tabla 13 Caso de uso 4

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear estación (Tabla parámetro) | | | | | | CU5 |
| Actores | Administrador, Operador | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | El sistema deberá estar abierto en el menú de parametrización, los datos GMT deberán estar cargados en base de datos. | | | | | | |
| Postcondición | La estación queda creada en el sistema. | | | | | | |
| Autor | Fabián Mina | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Gestionar las estaciones |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Creación de la estación | | | |
| 1 | El usuario oprime el botón nueva estación. | 2 | El sistema carga el Id automático que corresponderá a la nueva estación y habilita los campos de la estación. |
| 3 | El usuario digita el nombre de la estación. |  |  |
| 4 | El usuario digita serial de la estación |  |  |
| 5 | El usuario digita un nombre corto para la estación |  |  |
| 6 | El usuario selecciona GMT de la estación (Zona horaria) | 7 | El sistema despliega la lista de GMT |
| 10 | El usuario selecciona categoría | 11 | El sistema carga la lista de categorías almacenadas |
| 12 | El usuario selecciona región | 13 | El sistema carga la lista de regiones almacenadas |
| 14 | El usuario selecciona protocolo de comunicación (TCP o modbus) |  |  |
| 15 | El usuario selecciona el campo localización | 16 | El sistema carga la lista de localizaciones almacenadas |
| 17 | El usuario Selecciona el campo ciudad | 18 | El sistema carga la lista de ciudades almacenadas |
| 19 | El usuario diligencia los campos latitud y longitud.  El usuario digita estos campos en grados decimales |  |  |
| 20 | El usuario diligencia el campo elevación  El usuario digita estos campos en grados decimales |  |  |
| 21 | El usuario ingresa un número entero entre 0 y 120 para la base de tiempo.  (base de tiempo entre 0 y 2 horas) | 22 | El sistema valida que una estación no tenga asignada la misma base de tiempo. |
| 23 | El usuario digita una descripción de la estación. |  |  |
| 24 | El usuario oprime el botón guardar | 25 | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación.  “Estación creada correctamente” |
|  |  | 26 | El sistema muestra en el listado la nueva estación creada. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 2 | El usuario no selecciona ningún valor de la lista. El sistema por defecto debe colocar el valor “Sin definir” y se debe tomar como un valor vacío al intentar guardar. |
| 3 | El usuario digita valores alfanuméricos Se muestra el mensaje “Los valores deben ser decimales” |
| 4 | La estación ya existe con la misma base de tiempo muestra mensaje “Ya existe esta base de tiempo en la estación” |
| 5 | Hay campos vacíos, el sistema muestra el mensaje “Hay campos vacíos”, selecciona el campo en rojo y el cursor se ubica en el campo. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de la estación | | | |
| 1 | El usuario digita el nombre de la estación o el Id de la región. | 2 | El sistema carga los datos de la estación. |
| 3 | El usuario oprime el botón editar. | 4 | El sistema habilita los campos nombre, serial, nombre corto, GMT, descripción, región.  Localización  Ciudad, latitud y longitud, elevación para actualización |
| 5 | El usuario actualiza los campos nombre, serial, nombre corto, GMT, descripción, región.  Localización  Ciudad, latitud y longitud y oprime botón guardar | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación.  “Guardado con Satisfactoriamente” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 2 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 3 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

Tabla 14 Caso de uso 5

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear organización (Tabla parámetro) | | | | | | CU6 |
| Actores | Administrador | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | El sistema deberá estar abierto en el menú de parametrización | | | | | | |
| Postcondición | La organización quedara creada en el sistema. | | | | | | |
| Autor | Fabián Mina | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Gestionar las organizaciones del sistema |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Creación de la organización. | | | |
| 1 | El usuario oprime el botón nueva organización | 2 | El sistema carga el Id automático que corresponde a la nueva organización y habilita los campos de la organización. |
| 3 | El usuario digita el nombre de la organización. Longitud máxima de 50 caracteres. |  |  |
| 4 | El usuario digita el Email. Longitud máxima de 50 caracteres. |  |  |
| 5 | El usuario digita número de contacto. Longitud máxima de 50 caracteres. |  |  |
| 6 | El usuario digita una descripción para la organización. Longitud máxima de 50 caracteres. |  |  |
| 7 | El usuario oprime el botón guardar | 8 | El sistema valida que todos los campos estén llenos y guarda. Muestra mensaje de confirmación.  “Organización creada correctamente” |
|  |  | 9 | El sistema muestra en el listado la nueva organización creada. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 2 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de la organización | | | |
| 1 | El usuario digita el nombre de la organización o el Id de la organización. | 2 | El sistema carga los datos de la organización. |
| 3 | El usuario oprime el botón editar. | 4 | El sistema habilita los campos nombre de la organización, email, numero contacto y descripción para actualización. |
| 5 | El usuario actualiza el campo nombre de la organización, email, numero de contacto, descripción y oprime botón guardar | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación: “Organización actualizada correctamente” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 2 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 3 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

Tabla 15 Caso de uso 6

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear categoría (Tabla parámetro) | | | | | | CU7 |
| Actores | Administrador | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | El sistema deberá estar abierto en el menú de parametrización | | | | | | |
| Postcondición | La categoría quedara creada en el sistema. | | | | | | |
| Autor | Fabián Mina | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Gestionar las categorías del sistema |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Creación de la categoría. | | | |
| 1 | El usuario oprime el botón nueva categoría | 2 | El sistema carga el Id automático que corresponde a la nueva categoría y habilita los campos de la categoría. |
| 3 | El usuario digita el nombre de la categoría. |  |  |
| 4 | El usuario digita una descripción para la categoría. |  |  |
| 5 | El usuario oprime el botón guardar | 6 | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación de transacción  “Categoría creada correctamente” |
|  |  | 7 | El sistema muestra en el listado la nueva categoría creada. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 2 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de la categoría | | | |
| 1 | El usuario digita el nombre de la categoría o el Id de la organización. | 2 | El sistema carga los datos de la categoría. |
| 3 | El usuario oprime el botón editar. | 4 | El sistema habilita los campos nombre de la categoría y descripción para actualización. |
| 5 | El usuario actualiza el campo nombre de la categoría, descripción y oprime botón guardar | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 2 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 3 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

Tabla 16 Caso de uso 7

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear Ciudad (Tabla parámetro) | | | | | | CU8 |
| Actores | Administrador | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | El sistema deberá estar abierto en el menú de parametrización, por cada región, las ciudades deberán corresponder. Se mostrará combo box para seleccionar región a la que pertenece la ciudad. | | | | | | |
| Postcondición | La ciudad quedara creada en el sistema. | | | | | | |
| Autor | Daniela Romero | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Gestionar las ciudades del sistema |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Creación de la Ciudad. | | | |
| 1 | El usuario oprime el botón nueva Ciudad | 2 | El sistema carga el Id automático que corresponde a la nueva Ciudad y habilita los campos de la Ciudad. |
| 3 | El usuario digita el nombre de la Ciudad. |  |  |
| 4 | El usuario selecciona la región a la que pertenece la ciudad | 5 | El sistema despliega la lista de regiones almacenadas en base de datos. |
| 6 | El usuario digita una descripción para la Ciudad. |  |  |
| 7 | El usuario oprime el botón guardar | 8 | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación. |
|  |  | 9 | El sistema muestra en el listado la nueva ciudad creada. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 2 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de la Ciudad | | | |
| 1 | El usuario digita el nombre de la ciudad o el Id de la ciudad. | 2 | El sistema carga los datos de la ciudad. |
| 3 | El usuario oprime el botón editar. | 4 | El sistema habilita los campos nombre de la ciudad, región y descripción para actualización. |
| 5 | El usuario actualiza el campo nombre de la Ciudad, región, descripción y oprime botón guardar | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. “Ciudad actualizada correctamente” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 2 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 3 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

Tabla 17 Caso de uso 8

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 9

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear rango (Tabla parámetro) | | | | | | CU9 |
| Actores | Administrador | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | El sistema deberá estar abierto en el menú de parametrización, por cada estación, los rangos deberán corresponder. Se mostrará combo box para seleccionar estación a la que pertenece el rango. | | | | | | |
| Postcondición | Los rangos quedaran creados en el sistema. | | | | | | |
| Autor | Daniela Romero | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Gestionar los rangos del sistema, los cuales definen el porcentaje de variación al gestionar la información. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Creación de rango. | | | |
| 1 | El usuario oprime el botón nuevo Rango | 2 | El sistema carga el Id automático que corresponde al nuevo rango y habilita los campos. Rango mínimo, rango máximo, rango, descripción. |
| 3 | El usuario digita un nombre para el rango. Longitud máxima de 50 caracteres. |  |  |
| 4 | El usuario digita valor mínimo del nuevo rango.  (Valor numérico) valor mínimo 0 |  |  |
| 5 | El usuario digita valor máximo del nuevo rango.  (Valor numérico) |  |  |
| 6 | El usuario selecciona la estación a la que pertenece el rango | 7 | El sistema despliega la lista de estaciones almacenadas en base de datos. |
| 8 | El usuario digita una descripción para el rango.  Longitud máxima de 50 caracteres. |  |  |
| 9 | El usuario oprime el botón guardar | 10 | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación de transacción “Guardado exitosamente” |
|  |  | 11 | El rango guardado se muestra en la cuadrícula donde están listados todos los rangos |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 2 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 3 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización del rango | | | |
| 1 | El usuario digita el nombre del rango o el Id. | 2 | El sistema carga los datos del rango. |
| 3 | El usuario oprime el botón editar. | 4 | El sistema habilita los campos nombre del rango, rango, región y descripción para actualización. |
| 5 | El usuario actualiza el campo nombre del rango, rango, región, descripción y oprime botón guardar  Longitud máxima de 50 caracteres. | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. “Rango actualizado correctamente” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 2 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 3 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

Tabla 18 Caso de uso 9

Autor: Fabian Mina – Daniela Romero

CASOS 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 | Daniela Romero | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | | |

|  |
| --- |
| Propósito |
| Gestionar las alertas del sistema |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Creación alerta. | | | |
| 1 | El usuario oprime el botón nuevo alerta. | 2 | El sistema carga el Id automático que corresponde a la nueva alerta y habilita los campos. |
| 3 | El usuario digita un nombre para la alerta. Longitud máxima de 50 caracteres. |  |  |
| 4 | El usuario digita una descripción para la alerta creada. Longitud máxima de 50 caracteres. |  |  |
| 5 | El usuario selecciona tipo de alerta.  (Fax - Email – SMS ) |  |  |
| 6 | El usuario digita Email |  |  |
| 7 | El usuario digita contraseña |  |  |
| 8 | El usuario digita puerto SMTP |  |  |
| 9 | El usuario digita servidor SMTP |  |  |
| 10 | El usuario selecciona tipo de seguridad (SSL, TLS, SSL/TLS) |  |  |
| 11 |  |  |  |
| 12 | El usuario oprime el botón guardar | 13 | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación de transacción.  “Alerta guardada con éxito” |
|  |  | 14 | Se muestra en la cuadrícula donde están listados todas las alertas creadas. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de alerta | | | |
| 1 | El usuario digita el nombre o el Id de la alerta | 2 | El sistema carga los datos. |
| 3 | El usuario oprime el botón editar. | 4 | El sistema habilita los campos nombre de alerta, tipo y descripción para actualización. |
| 5 | El usuario actualiza el campo nombre de alerta, estación, tipo y oprime botón guardar | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 2 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |
| 3 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

Tabla 19 Caso de uso 10

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 11

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear base de tiempo (Tabla parámetro) | | | | | | CU11 |
| Actores | Administrador | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | El sistema deberá estar abierto en el menú de parametrización. | | | | | | |
| Postcondición | Los tiempos quedaran creados en el sistema | | | | | | |
| Autor | Daniela Romero | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Gestionar los tiempos para validar 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 1440 minutos. |
| 7 | El usuario digita una descripción para los tiempos registrados  Longitud máxima de 50 caracteres. |  |  |
| 8 | El usuario selecciona la alerta deseada, donde recibirá notificación | 9 | El sistema despliega las alertas creadas y almacenadas en base de datos.  Fax  E-mail  Mensaje  Ventana Emergente |
| 10 | El usuario oprime el botón guardar | 11 | El sistema valida la información guardad, muestra mensaje en pantalla guardado con éxito  . |
|  |  | 12 | Se muestra en la cuadrícula donde están listados todos los tiempos creadas. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de presentarse un error el sistema mostrara mensaje en pantalla indicando el error. |
| 2 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 1 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de tiempos | | | |
| 1 | El usuario digita el nombre o el Id | 2 | El sistema carga los datos. |
| 3 | El usuario oprime el botón editar. | 4 | El sistema habilita los campos nombre y rango de tiempos. |
| 5 | El usuario actualiza el campo nombre, rango de tiempos y oprime botón guardar.  Longitud máxima de 50 caracteres. | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 1 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 1 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

Tabla 20 Caso de uso 11

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 12

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Validar Usuario (login) | | | | | | CU12 |
| Actores | Administrador Operador Visualizador editor final editor preeeliminar | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | Los usuarios para realizar login deberán estar almacenados en base de datos. | | | | | | |
| Postcondición | Los tiempos quedaran creados en el sistema | | | | | | |
| Autor | Daniela Romero | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Iniciar sesión |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Iniciar sesión. | | | |
| 1 | El usuario digita usuario o correo electrónico.  Longitud máxima de 50 caracteres. | 2 | El sistema valida formato correo electrónico (ejemplo@ejemplo.com) |
| 3 | El usuario digita contraseña Longitud máxima de 20 caracteres mínimo 7 caracteres, debe contener mínimo 1 mayúscula y 1 carácter numérico. |  |  |
| 4 | El usuario oprime la opción Ingresar | 5 | El sistema valida que la información ingresada sea correcta. |
|  |  | 6 | El sistema ingresa al aplicativo |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 2 | El usuario ingresa contraseña diferente a las reglas (1mayuscula 1numero, min 7 dígitos. |
| 3 | En caso de que el usuario no se encuentre registrado, se deberá mostrar mensaje en pantalla “El usuario no existe” |
| 4 | Se digita contraseña o usuario no valido, el sistema muestra mensaje “Credenciales no validas” |
| 5 | Hay campos en blanco. El sistema no ingresa al aplicativo y muestra mensaje de validación. “Debe llenar los campos obligatorios” |
| 6 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

Tabla 21 Caso de uso 12

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 13

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Modificar datos descargados | | | | | | CU13 |
| Actores | Administrador | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | -Los datos se encuentren habilitados para ser modificados.  -El editor preliminar y editor final solo podrán modificar los datos.  - si el editor final bloquea los datos, el editor preliminar no podrá modificar los datos. | | | | | | |
| Postcondición | Los tiempos quedaran creados en el sistema | | | | | | |
| Autor | Daniela Romero | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Modificar datos procesados |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Editor Preliminar. | | | |
| 1 | El editor preliminar ingresa a la plantilla creada. |  |  |
| 2 | El editor preliminar selecciona la estación para modificar datos. | 3 | El sistema carga los datos de la estación seleccionada. |
| 4 | El editor preliminar selecciona los datos de la estación que serán modificados. | 5 |  |
| 5 | El editor preliminar modifica los datos procesados. |  |  |
| 7 | El editor final guarda los datos modificados | 8 | El sistema muestra mensaje en pantalla “Guardado con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de presentarse un error el sistema mostrara mensaje en pantalla indicando el error. |
| 2 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 3 | El editor final bloquea los datos, no podrán ser modificados |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Editor Final | | | |
| 1 | El editor final ingresa a la plantilla creada. |  | . |
| 2 | El editor final selecciona la estación para modificar datos. | 3 | El sistema carga los datos de la estación seleccionada. |
| 5 | El editor final selecciona los datos de la estación que serán modificados. | 6 | El sistema muestra los datos modificados por el usuario preliminar |
| 6 | El editor final modifica los datos procesados. |  |  |
| 7 | El editor final guarda los datos procesados. | 8 | El sistema muestra mensaje en pantalla, “Guardado con éxito” |
| 9 | El editor final bloquea los datos procesados | 10 | El sistema inhabilita los datos para ser modificados por el usuario preliminar.  Los datos no podrán ser modificados por ningún usuario incluyendo el usuario final. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de error el sistema mostrara mensaje en pantalla “Error al guardar los datos” |
| 2 | El usuario intente modificar los datos, el sistema muestra mensaje en pantalla “los datos han sido bloqueados" |

Tabla 22 Caso de uso 13

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 14

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Dar de baja datos procesados | | | | | | CU14 |
| Actores | Usuario Preliminar | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | Los datos deberán estar bloqueados | | | | | | |
| Postcondición | Los datos no serán eliminados, quedarán inhabilitados | | | | | | |
| Autor | Fabian Mina | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Inhabilitar datos |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
| 1 | El editor final ingresa a la plantilla creada. |  |  |
| 2 | El editor final selecciona la estación para modificar datos. | 3 | El sistema carga los datos de la estación seleccionada. |
| 4 | El editor preliminar selecciona los datos de la estación que serán bloqueados | 5 |  |
| 5 | El editor final bloquea los datos. | 6 | El sistema muestra mensaje de advertencia. “Desea bloquear los datos, estos no se podrán volver a modificar” |
| 7 | El editor final confirma el bloqueo de datos | 8 | El sistema guarda la fecha y usuario de realización de bloqueo de datos. |
|  |  | 9 | El sistema muestra mensaje en pantalla “Bloqueado con éxito” |
|  |  | 10 | Los datos bloqueados se guardan en base de datos |

Tabla 23 Caso de uso 14

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 15

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Conectar Automáticamente con estación | | | | | CU15 |
| Actores | Administrador, Operador | | | | | |
| Referencias | Requerimiento RF 1 | | |  | | |
| Precondición | -El sistema deberá estar conectado a la red de internet  -la estación ya debe estar creada con anterioridad,  -La estación deberá tener configurado el protocolo de comunicación con el cual se hará la conexión. | | | | | |
| Postcondición | El sistema debe mostrar los datos básicos de la estación a la que se ha conectado | | | | | |
| Autor | Fabián Mina | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Conectarse Automáticamente a una mediante protocolo FTP o Modbus |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
| 1 | El usuario selecciona el nombre de la estación. | 2 | El sistema muestra la información almacenada para ese nombre de estación.  -Id  -Serial  -Abreviación nombre corto de la -estación  -organización  -categoría  -región.  -Localización  -ciudad  -latitud y longitud  -elevación  -Base de tiempo |
|  |  | 3 | El sistema verifica el protocolo de comunicación configurado a la estación |
|  |  | 4 | El sistema se conecta a la estación seleccionada. |
|  |  | 5 | El sistema se conecta a la estación seleccionada y muestra mensaje en pantalla “Conexión Exitosa con al estación” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 2 | En caso de que los datos no sean válidos o se presente algún error al momento de llenar el formulario, el sistema debe mostrar mensaje en pantalla, especificando el error que se presentó |
| 3 | Si no existe protocolo de comunicación con esta estación, se mostrará mensaje en pantalla “No existe protocolo de comunicación”. |

Tabla 24 Caso de uso 15

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 16

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear Mantenimiento | | | | | | CU16 |
| Actores | Administrador, Operador | | | | | | |
| Referencias | Requerimiento RF 1 y 2 | | | |  | | |
| Precondición | El registro de mantenimiento no podrá ser modificado | | | | | | |
| Postcondición | El registro de mantenimiento creado quedara almacenado en base de datos con la respectiva hora y usuario de creación. | | | | | | |
| Autor | Daniela Romero | | Fecha | 3/08/19 | | Versión | V 6.0 |
| Aprobación | | Aprobado | | | | | |

|  |
| --- |
| Propósito |
| Gestionar el registro de actividad |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 1  Crear mantenimiento. | | | |
| 1 | El usuario oprime el botón nuevo registro | 2 | El sistema carga el Id automático que corresponde al nuevo registro. |
| 3 | El usuario selecciona el nombre de la estación, en el campo registrar estación. |  |  |
| 4 | El usuario digita el nombre del funcionario, en el campo registrar funcionario. |  |  |
| 5 | El usuario digita una descripción o novedad, en el campo novedad |  |  |
| 6 | El usuario oprime el botón guardar | 7 | El sistema valida que todos los campos estén llenos y guarda.  Muestra mensaje de confirmación “Registro guardado con éxito”. |
|  |  | 8 | El sistema carga la hora y fecha de creación automáticamente |
|  |  | 9 | El nuevo registro se almacena en base de datos, y se muestra listado de registros en grilla. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje, selecciona el campo en rojo y el cursor se ubica en el campo. |
| 2 | El usuario intenta escribir más de 50 caracteres: el campo no permite escribir más de 50 caracteres. |
| 3 | Los campos de texto permiten valores alfanuméricos y caracteres especiales |

Tabla 25 Caso de uso 16

Autor: Fabian Mina – Daniela Romero

Casos de uso Procesamiento

CASOS DE USO 17

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Criterios de validación | | | | | CU17 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF 14 | | |  | | |
| Precondición | El usuario deberá estar en el módulo procesamiento – Gestión de plantillas de procesamiento. | | | | | |
| Postcondición | Los criterios deben quedar configurados por cada elemento seleccionado. | | | | | |
| Autor | Fabian Mina | Fecha | 3/08/19 | | Versión | V 6.0 |

|  |
| --- |
| Propósito |
| Creación de criterios |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal Escenario 1  Procesar datos | | | |
| 1 | El usuario ingresa a la pantalla de transferencias |  |  |
| 2 | El usuario selecciona la variable que desea consultar | 3 | El sistema carga la lista de variables almacenadas |
| 4 | El usuario selecciona la fecha y hora que desea consultar |  |  |
| 5 | El usuario ingresa la cantidad de datos a consultar |  |  |
| 6 | El usuario oprime consultar | 7 | El sistema valida la cantidad de transferencias realizadas. |
|  |  | 8 | El sistema carga la cantidad de transferencias realizadas. |
| 9 | El usuario ingresa el porcentaje con el cual se realizará la operación. |  |  |
| 10 | El usuario oprime la opción procesar | 11 | El sistema realiza la operación interna |
|  |  | 12 | El sistema muestra mensaje en pantalla al realizar este proceso “Procesando información” |
|  |  | 13 | El sistema carga la información en pantalla. |
| 14 | El usuario guarda esta información | 15 | El sistema almacena la información cargada en base de datos. |
|  |  | 16 | El sistema muestra mensaje en pantalla “Guardado con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de presentarse error al momento de realizar operación, se deberá mostrar mensaje en pantalla, la pantalla se recargará para ingresar los datos nuevamente. |
| 2 | En caso de que los datos no sean válidos o se presente algún problema se deberá mostrar en pantalla un mensaje de validación. “Error al guardar información”. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal Escenario 2  Datos Inválidos transferencias realizadas | | | |
|  |  | 1 | El sistema realiza validación de datos |
|  |  | 2 | El sistema muestra mensaje en pantalla con los datos inválidos |
| 3 | El usuario presiona la opción ver | 4 | El sistema carga y muestra la información con los datos inválidos encontrados al momento de procesar la información.  (Topes altos, Topes bajos, Problemas de comunicación, recepción de datos). |
| 5 | El usuario presiona la opción guardar | 6 | El sistema guarda la información, muestra mensaje en pantalla guardado con éxito. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de presentarse error al momento de realizar operación, se deberá mostrar mensaje de validación en pantalla. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal Escenario 3  Corregir datos Inválidos transferencias realizadas | | | |
| 1 | El usuario presiona la opción corregir | 2 | El sistema habilita los campos con los datos inválidos |
| 3 | El usuario modifica los campos inválidos | 2 | El sistema muestra mensaje en pantalla con los datos inválidos |
| 3 | El usuario presiona la opción guardar | 4 | El sistema guarda la información digitada  . |
|  |  | 5 | El sistema guarda la información, muestra mensaje en pantalla guardado con éxito. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de presentarse error al momento de realizar operación, se deberá mostrar mensaje de validación en pantalla. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal Escenario 4  Invalidar datos Inválidos transferencias realizadas | | | |
| 1 | El usuario presiona la opción invalidar | 2 | El sistema muestra mensaje en pantalla “¿desea invalidar los datos?” |
| 3 | El usuario selecciona la opción si | 4 | El sistema muestra mensaje en pantalla “Los datos fueron inhabilitados” |
|  |  | 5 | El sistema recarga la pagina para realizar procesamiento de datos nuevamente |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de presentarse error al momento de realizar operación, se deberá mostrar mensaje de validación en pantalla. |

Tabla 26 Caso de uso 17

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 18

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear plantillas | | | | | CU18 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF 17 | | |  | | |
| Precondición | El sistema selecciona la estación, para la cual se creara la plantilla. | | | | | |
| Postcondición | La plantilla creada deberá visualizarse en el listado de plantillas de configuración. | | | | | |
| Autor | Daniela Romero | Fecha | 3/08/19 | | Versión | V 6.0 |

|  |
| --- |
| Propósito |
| Creación de plantilla de procesamiento |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
| 1 | El usuario selecciona variable | 2 | El sistema despliega la lista de elementos almacenados en base de datos. |
| 3 | El usuario selecciona rango máximo para elemento. |  |  |
| 4 | El usuario selecciona rango mínimo para elemento. |  |  |
| 5 | El usuario selecciona valor pico |  |  |
| 6 | El usuario selecciona valor constante para el elemento. |  |  |
| 7 | El usuario guarda la plantilla | 8 | El sistema muestra mensaje de validación. “Plantilla creada con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | SI hay un campo sin diligenciar, el sistema mostrara mensaje de validación y seleccionara el campo en rojo“ |

Tabla 27 Caso de uso 18

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 19

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Ver Plantillas | | | | | CU19 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF 17 | | |  | | |
| Precondición | El usuario deberá estar en el módulo procesamiento – Gestión de plantillas de procesamiento. | | | | | |
| Postcondición | Los datos de reglas y criterios deben quedar configurados por cada elemento seleccionado. | | | | | |
| Autor | Daniela Romero | Fecha | 3/08/19 | | Versión | V 6.0 |

|  |
| --- |
| Propósito |
| Creación de reglas o criterios |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
| 1 | El usuario selecciona el elemento a configurar | 2 |  |
| 3 | El usuario despliega plantillas creadas | 4 | El sistema desplegara todas las plantillas creadas en un campo select |
| 5 | El usuario modifica la plantilla seleccionada y guarda los datos. | 6 | El sistema muestra los datos de la plantilla a modificar |
|  | El usuario inhabilita plantilla |  | El sistema muestra mensaje que la plantilla seleccionada fue inhabilitada. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de que los datos no sean válidos o se presente algún problema al realizar el crud se deberá mostrar en pantalla un mensaje de alerta. |

Tabla 28 Caso de uso 19

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 20

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear usuario | | | | | CU20 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF 17 | | |  | | |
| Precondición | El usuario deberá estar en el módulo procesamiento – Gestión de usuarios de procesamiento,  La organización ya debe estar creada. | | | | | |
| Postcondición | Los usuarios quedaran atados a los roles configurados y asignados | | | | | |
| Autor | Daniela Romero | Fecha | 3/08/19 | | Versión | V 6.0 |

|  |
| --- |
| Propósito |
| Creación de reglas o criterios |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
| 1 | El usuario oprime botón nuevo o agregar | 2 | El sistema carga el Id automático que corresponde al nuevo usuario |
| 3 | El usuario digita documento de identidad |  |  |
| 4 | El usuario digita nombres del usuario |  |  |
| 5 | El usuario Digita apellidos |  |  |
| 6 | El usuario selecciona cargo |  |  |
| 7 | El usuario selecciona la organización |  |  |
| 8 | El usuario selecciona rol para el usuario. | 9 | El sistema despliega la lista de roles almacenados en base de datos. |
| 10 | El usuario guarda la información. | 11 | El sistema muestra mensaje en pantalla “Usuario Guardado con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de que los datos no sean válidos o se presente algún problema se deberá mostrar mensaje de validación en pantalla. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de usuario | | | |
| 1 | El usuario digita el documento de usuario o el Id. | 2 | El sistema carga los datos del usuario. |
| 3 | El usuario oprime el botón editar. | 4 | El usuario habilita los campos nombres, apellidos, documento, cargo, rol para actualización |
| 5 | El usuario actualiza el nombre, apellidos, documento, cargo, rol y oprime botón guardar. | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje y el cursor se ubica en el campo. |

Tabla 29 Caso de uso 20

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 21

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear Roles | | | | | CU21 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF | | |  | | |
| Precondición | El administrador, operador y visualizador no afectaran los datos descargados. | | | | | |
| Postcondición | Los usuarios tendrán asignados diferentes permisos por usuario | | | | | |
| Autor | Fabian Mina | Fecha | 3/08/19 | | Versión | V 6.0 |

|  |
| --- |
| Propósito |
| Creación roles del sistema |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
| 1 | El usuario oprime botón nuevo o agregar | 2 | El sistema carga el Id automático que corresponde al nuevo rol |
| 3 | El usuario digita nombre del rol.  (Operador, Visualizador, editor preliminar, editor final) |  |  |
| 4 | El usuario selecciona permisos a cada rol |  |  |
| 5 | El usuario Digita descripción para cada rol |  |  |
| 6 | El usuario guarda la información | 7 | El sistema muestra mensaje en pantalla “Usuario Guardado con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de que los datos no sean válidos o se presente algún problema se deberá mostrar mensaje de validación en pantalla. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de rol | | | |
| 1 | El usuario digita el Id del rol guardado. | 2 | El sistema carga los datos almacenados. |
| 3 | El usuario oprime el botón editar. | 4 | El usuario habilita los campos nombres, permisos y descripción para actualización |
| 5 | El usuario actualiza el nombre, permisos, descripción y oprime botón guardar. | 6 | El sistema valida que todos los campos estén llenos, guarda y muestra mensaje de confirmación. “Guardado con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje y el cursor se ubica en el campo. |

Tabla 30 Caso de uso 21

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 22

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear permisos | | | | | CU22 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF | | |  | | |
| Precondición | El usuario deberá estar en el módulo procesamiento – Gestión de datos de procesamiento. | | | | | |
| Postcondición | El usuario debe contar con permisos administrador | | | | | |
| Autor | Daniela Romero | Fecha | 3/08/19 | | Versión | V 6.0 |

|  |
| --- |
| Propósito |
| Creación de permisos para cada rol. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal Escenario 1  Visualizador | | | |
| 1 | El usuario selecciona la opción crear permisos. | 2 | El sistema carga el id automático que corresponde al nuevo permiso. |
| 3 | El usuario selecciona el rol visualizador |  |  |
| 4 | El usuario asigna los permisos para visualizador. | 5 | Visualizador: solo podrá ver datos de las estaciones |
| 4 | El usuario digita descripción para el estado. |  |  |
| 5 | El usuario guarda la información | 6 | El sistema muestra mensaje en pantalla “Usuario Guardado con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de que los datos no sean válidos o se presente algún problema se deberá mostrar mensaje de validación en pantalla. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal Escenario 2  Operador | | | |
| 1 | El usuario selecciona la opción crear permisos. | 2 | El sistema carga el id automático que corresponde al nuevo permiso. |
| 3 | El usuario selecciona el rol operador |  |  |
| 4 | El usuario asigna los permisos para operador. | 5 | Operador: Puede modificar datos de las estaciones (nombre, serial, organización, categoría, región,  localización, cuidad, latitud y longitud, elevación, parametros) |
| 4 | El usuario digita descripción para el estado. |  |  |
| 5 | El usuario guarda la información | 6 | El sistema muestra mensaje en pantalla “Usuario Guardado con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 6 | En caso de que los datos no sean válidos o se presente algún problema se deberá mostrar mensaje de validación en pantalla. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal Escenario 3  Editor Preliminar | | | |
| 1 | El usuario selecciona la opción crear permisos. | 2 | El sistema carga el id automático que corresponde al nuevo permiso. |
| 3 | El usuario selecciona el rol Editor Preliminar |  |  |
| 4 | El usuario asigna los permisos para Editor Preliminar. | 5 | Editor Preliminar: Valida y realiza corrección de datos. |
| 4 | El usuario digita descripción para el estado. |  |  |
| 5 | El usuario guarda la información | 6 | El sistema muestra mensaje en pantalla “Usuario Guardado con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de que los datos no sean válidos o se presente algún problema se deberá mostrar mensaje de validación en pantalla. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal Escenario 4  Editor Final | | | |
| 1 | El usuario selecciona la opción crear permisos. | 2 | El sistema carga el id automático que corresponde al nuevo permiso. |
| 3 | El usuario selecciona el rol Editor Final |  |  |
| 4 | El usuario asigna los permisos para editor Final. | 5 | Editor Final: Realiza validación, corrección de datos, edita ediciones realizadas por editor preliminar y cierra datos |
| 4 | El usuario digita descripción para el estado. |  |  |
| 5 | El usuario guarda la información | 6 | El sistema muestra mensaje en pantalla “Usuario Guardado con éxito” |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 5  Actualización de estado | | | |
| 1 | El usuario digita el Id del estado guardado. | 2 | El sistema carga los datos almacenados. |
| 3 | El usuario oprime el botón editar. | 4 | El usuario habilita los campos nombre, descripción para actualización |
| 5 | El usuario actualiza el nombre, descripción y oprime botón guardar. | 6 | El sistema valida que todos los campos estén llenos, guarda la información |
|  |  | 7 | El sistema muestra mensaje de confirmación |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje y el cursor se ubica en el campo. |

Tabla 31 Caso de uso 22

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 23

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Crear estados | | | | | CU23 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF | | |  | | |
| Precondición | El usuario deberá estar en el módulo procesamiento – Gestión de datos de procesamiento. | | | | | |
| Postcondición | El usuario debe contar con permisos administrador | | | | | |
| Autor | Fabian Mina | Fecha | 3/08/19 | | Versión | V 6.0 |

|  |
| --- |
| Propósito |
| Creación de estados bandera |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
| 1 | El usuario oprime botón nuevo o agregar | 2 | El sistema carga el Id automático que corresponde al nuevo estado |
| 3 | El usuario digita nombre del estado.  (Dato valido)  (Dato invalido)  (tope alto)  (tope bajo)  (Problema de comunicación)  (Recepción de datos)  (Valor negativo) |  |  |
| 4 | El usuario digita descripción para el estado. |  |  |
| 5 | El usuario guarda la información | 6 | El sistema muestra mensaje en pantalla “Usuario Guardado con éxito” |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | En caso de que los datos no sean válidos o se presente algún problema se deberá mostrar mensaje de validación en pantalla. |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal escenario 2  Actualización de estado | | | |
| 1 | El usuario digita el Id del estado guardado. | 2 | El sistema carga los datos almacenados. |
| 3 | El usuario oprime el botón editar. | 4 | El usuario habilita los campos nombre, descripción para actualización |
| 5 | El usuario actualiza el nombre, descripción y oprime botón guardar. | 6 | El sistema valida que todos los campos estén llenos, guarda la información |
|  |  | 7 | El sistema muestra mensaje de confirmación |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | Hay campos vacíos, el sistema muestra el mensaje y el cursor se ubica en el campo. |

Tabla 32 Caso de uso 23

Autor: Fabian Mina – Daniela Romero

CASOS DE USO 24

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Asignar estados | | | | | CU24 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF | | |  | | |
| Precondición | El usuario deberá estar en el módulo procesamiento – Gestión de datos de procesamiento. | | | | | |
| Postcondición | El usuario debe contar con permisos administrador | | | | | |
| Autor | Daniela Romero | Fecha | 3/08/19 | | Versión | V 6.0 |

|  |
| --- |
| Propósito |
| Asignación de estados bandera |

|  |  |  |  |
| --- | --- | --- | --- |
| Curso Normal | | | |
|  |  | 1 | El sistema valida la información procesada. |
|  |  | 2 | El sistema valida valores con topes altos. Asigna estado (Tope alto) |
|  |  | 3 | El sistema valida valores con topes bajos. Asigna estado (Tope bajo) |
|  |  | 4 | El sistema valida valores con topes bajos. Asigna estado (Tope bajo) |
|  |  | 5 | El sistema valida valores con problemas de comunicación. Asigna estado (problemas de comunicación) |
|  |  | 6 | El sistema valida valores con Recepción de datos. Asigna estado (Recepción de datos) |
|  |  | 7 | El sistema valida valores negativos. Asigna estado (valores negativos) |

|  |  |
| --- | --- |
| Cursos Alternos | |
| 1 | El sistema pregunta al usuario que desea hacer:” Corregir” “Invalidar” “No hacer nada” |
|  | El usuario selecciona Corregir: Envía a CU 1 Curso Normal Escenario 3  El usuario selecciona Invalidar: Envía CU1 Curso normal Escenario 4 |
| 2 | El sistema pregunta al usuario que desea hacer:” Corregir” “Invalidar” “No hacer nada” |
|  | El usuario selecciona Corregir: Envía a CU 1 Curso Normal Escenario 3  El usuario selecciona Invalidar: Envía CU1 Curso normal Escenario 4 |
| 3 | El sistema pregunta al usuario que desea hacer:” Corregir” “Invalidar” “No hacer nada” |
|  | El usuario selecciona Corregir: Envía a CU 1 Curso Normal Escenario 3  El usuario selecciona Invalidar: Envía CU1 Curso normal Escenario 4 |
| 4 | El sistema pregunta al usuario que desea hacer:” Corregir” “Invalidar” “No hacer nada” |
|  | El usuario selecciona Corregir: Envía a CU 1 Curso Normal Escenario 3  El usuario selecciona Invalidar: Envía CU1 Curso normal Escenario 4 |
| 5 | El sistema pregunta al usuario que desea hacer:” Corregir” “Invalidar” “No hacer nada” |
|  | El usuario selecciona Corregir: Envía a CU 1 Curso Normal Escenario 3  El usuario selecciona Invalidar: Envía CU1 Curso normal Escenario 4 |
| 6 | El sistema pregunta al usuario que desea hacer:” Corregir” “Invalidar” “No hacer nada” |
|  | El usuario selecciona Corregir: Envía a CU 1 Curso Normal Escenario 3  El usuario selecciona Invalidar: Envía CU1 Curso normal Escenario 4 |
| 7 | El sistema pregunta al usuario que desea hacer:” Corregir” “Invalidar” “No hacer nada” |
|  | El usuario selecciona Corregir: Envía a CU 1 Curso Normal Escenario 3  El usuario selecciona Invalidar: Envía CU1 Curso normal Escenario 4 |

Tabla 33 Caso de uso 24

Autor: Fabian Mina – Daniela Romero

### Especificaciones de casos de pruebas

Criterios con los que se con los que se realizó la evaluación

Datos pruebas eficiencia y pruebas unitarias

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Prueba** | **Funcionalidad** | **Descripción Prueba** | **Datos de prueba** |
|
|
| Funcional | crear región | El usuario ingresará a la pantalla crear región, el sistema deberá ingresar y cargar todo el contenido en un tiempo máximo de 5 segundo. | Se dará click en el botón nueva región, se ingresará a la pantalla de crear región y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser máximo 5 segundos. |
| Funcional | crear región | El usuario ingresará a la pantalla crear región, el sistema cargará Id de la nueva región a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo **int**  Se ingresa nombre campo región: Caribe tipo **String.**  Se ingresan valores [a-z] tipo String min 50 caracteres.  Se ingresa nombre campo descripción: Esto es una prueba tipo **String.**  Se ingresan valores [a-z] tipo **String** min 50 caracteres. |
| Funcional | crear región | El usuario ingresará a la pantalla crear región, el sistema cargará Id de la nueva región a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo **int**  Se ingresan valores aleatorios tipo **int** mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación)   Se ingresa nombre campo descripción: Esto es una prueba tipo **String.**  Se ingresan valores [a-z] tipo **String** mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación) |
| Funcional | crear región | Se diligencias los campos nombre y descripción, el usuario oprime el botón guardar, el sistema deberá guardar la información en base de datos y mostrar mensaje de confirmación en pantalla. | Se diligencia le campo nombre: **Caribe**  Se diligencia campo Descripción: **Esto es una prueba.**  Se oprime botón guardar (El aplicativo deberá mostrar mensaje de confirmación Guardado con éxito) |
| Funcional | Crear región | Se selecciona la opcion editar | Se selecciona la opcion editar, se modifica el campo nombre por **Andina**  y descripcion por **Prueba dos** |
| Unitaria | crear región | Se probara el método crear región | **createRegion(id\_region,nombre\_region, observacion\_region)**, se escribira prueba unitaria donde: Se valide la sincronización del método crear región |
| Unitaria | crear región | Se probara el método consultar | **viewRegions()(id\_region,nombre\_region, observacion\_region)**se escribira prueba unitaria donde: Se valide la sincronización del método consulta de las regiones |
| Unitaria | crear región | Se probaran el método actualizar | **updateRegion(id\_region,nombre\_region, observacion\_region**)se escribira prueba unitaria donde: Se valide la sincronización del método actualizar regiones |
| Funcional | crear ciudad | El usuario ingresará a la pantalla crear Ciudad, el sistema deberá ingresar y cargar todo el contenido en un tiempo máximo de 5 segundo. | Se dará click en el botón nueva Ciudad, se ingresará a la pantalla de crear Ciudad y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser máximo 5 segundos. |
| Funcional | crear ciudad | El usuario ingresará a la pantalla crear Ciudad, el sistema cargará Id de la nueva Ciudad a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int  Se ingresa nombre campo Ciudad: Cali tipo String.   Se ingresan valores [a-z] tipo String min 50 caracteres.  Se ingresa nombre campo descripción:Valle del cauca tipo String.  Se ingresan valores [a-z] tipo String min 50 caracteres. |
| Funcional | crear ciudad | El usuario ingresará a la pantalla crear Ciudad, el sistema cargará Id de la nueva Ciudad a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int  Se ingresan valores aleatorios tipo int [0-9] mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación)   Se ingresa nombre campo descripción: Esto es una prueba tipo String.  Se ingresan valores [a-z] tipo String mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación) |
| Funcional | crear ciudad | Se diligencias los campos nombre y descripción, el usuario oprime el botón guardar, el sistema deberá guardar la información en base de datos y mostrar mensaje de confirmación en pantalla. | Se diligencia le campo nombre: **Cali** Se diligencia campo Descripción: **Valle del cauca**  Se oprime botón guardar (El aplicativo deberá mostrar mensaje de confirmación Guardado con éxito) |
| Funcional | Crear ciudad (Tabla Parametro) | Se requiere seleccionar región creada anteriormente | Se selecciona la opcion Region, se escoge la opcione al cargarse Andina |
| Unitaria | crear ciudad | Se probaran los metodos agregar Ciudad | **createCity**(id\_Ciudad,nombre\_Ciudad, observacion\_Ciudad), se escribira prueba unitaria donde: Se validara la sincronización y funcionamiento del método crear ciudad |
| Unitaria | crear ciudad | Se probaran los metodos consultar | **viewCities(id\_Ciudad,nombre\_Ciudad, observacion\_Ciudad**)se escribira prueba unitaria donde: Se validara la sincronización y funcionamiento del método crear ciudad |
| Unitaria | crear ciudad | Se probaran los metodos actualizar | **updateCity(id\_Ciudad,nombre\_Ciudad, observacion\_Ciudad)**se escribira prueba unitaria donde: id\_Ciudad:1 nombre\_Ciudad:Jamundi observacion\_Ciudad: Ubicada al sur del valle |
| Eficiencia | crear Mantenimiento | El usuario ingresará a la pantalla crear Mantenimiento, el sistema deberá ingresar y cargar todo el contenido en un tiempo mínimo de 5 segundo. | Se dará click en el botón nuevo Mantenimiento, se ingresará a la pantalla de crear Mantenimiento y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser mínimo 5 segundos. |
| Eficiencia | crear Mantenimiento | El usuario ingresará a la pantalla crear Mantenimiento, el sistema cargará Id de la nueva Mantenimiento a crear automáticamente, el usuario diligencia los campos nombre,fecha,estacion,observacion, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int  Se ingresa nombre campo Mantenimiento:calibrar dato tipo String.   Se ingresan valores [a-z] tipo String min 50 caracteres.   Se ingresa formato de fecha [dd-mm-aaaa] tipo date   Se ingresa nombre campo observacion: Esto es una prueba tipo String.  Se ingresan valores [a-z] tipo String min 50 caracteres. |
| Eficiencia | crear Mantenimiento | El usuario ingresará a la pantalla crear Mantenimiento, el sistema cargará Id de la nueva Mantenimiento a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int Se ingresa nombre campo Mantenimiento: Valores numéricos [0-9] (El sistema no debe permitir el ingreso)  Se ingresan valores aleatorios tipo int mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación)   Se ingresa nombre campo observacion: Esto es una prueba tipo String.  Se ingresan valores [a-z] tipo String mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación) |
| Eficiencia | crear Mantenimiento | Se diligencias los campos nombre y descripción, el usuario oprime el botón guardar, el sistema deberá guardar la información en base de datos y mostrar mensaje de confirmación en pantalla. | Se diligencia le campo nombre: **Mantenimiento estación** fecha: 03-09-2019 Se diligencia campo observación: **Esto es una prueba.**  Se oprime botón guardar (El aplicativo deberá mostrar mensaje de confirmación Guardado con éxito) |
| Unitaria | crear Mantenimiento | Se probaran los metodos agregar Mantenimiento | **agregarMantenimiento(id\_mtto, nombre\_tec\_ mtto, fecha\_ mtto, estación\_ mtto, observacion\_ mtto)**, se escribira prueba unitaria donde: id\_mtto:1 nombre\_mtto:Mantenimiento estación fecha\_mtto:03-09-2019 observacion\_mtto: Prueba observación pruebas |
| Unitaria | crear Mantenimiento | Se probaran los metodos consultar | **consultarMantenimiento(id\_mtto, nombre\_tec\_ mtto, fecha\_ mtto, estación\_ mtto, observacion\_ mtto)**se escribira prueba unitaria donde: id\_Mantenimiento:1 nombre\_Mantenimiento:Atlantico observacion\_Mantenimiento: Ubicacda al norte |
| Unitaria | crear Mantenimiento | Se probaran los metodos actualizar | **actualizarMantenimiento(id\_mtto, nombre\_tec\_ mtto, fecha\_ mtto, estación\_ mtto, observacion\_ mtto)**se escribira prueba unitaria donde: id\_mtto:1 nombre\_mtto:Mantenimiento estación fecha\_mtto:03-09-2019 observacion\_mtto: Prueba observación pruebas |
| Unitaria | crear Mantenimiento | Se probaran los metodos consultar Mantenimiento ID | **consultarRegioId(nombre\_Mantenimiento)**se escribira prueba unitaria donde: id\_mtto:1 |
| Eficiencia | crear Alerta | El usuario ingresará a la pantalla crear Alerta, el sistema deberá ingresar y cargar todo el contenido en un tiempo mínimo de 5 segundo. | Se dará click en el botón nueva Alerta, se ingresará a la pantalla de crear Alerta y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser mínimo 5 segundos. |
| Eficiencia | crear Alerta | El usuario ingresará a la pantalla crear Alerta, el sistema cargará Id de la nueva Alerta a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int  Se ingresa nombre campo Alerta: Alerta estación tipo **String.**   Se ingresan valores [a-z] tipo String min 50 caracteres.   Se ingresa nombre campo estación: Decepaz tipo **String.**  Se ingresa nombre campo observacion: Esto es una prueba tipo **String.**  Se ingresan valores [a-z] tipo String min 50 caracteres. |
| Eficiencia | crear Alerta | El usuario ingresará a la pantalla crear Alerta, el sistema cargará Id de la nueva Alerta a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int Se ingresa nombre campo Alerta: Valores numéricos [0-9] (El sistema no debe permitir el ingreso)  Se ingresan valores aleatorios tipo int mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación)   Se ingresa nombre campo descripción: Esto es una prueba tipo String.  Se ingresan valores [a-z] tipo String mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación) |
| Eficiencia | crear Alerta | Se diligencias los campos nombre y descripción, el usuario oprime el botón guardar, el sistema deberá guardar la información en base de datos y mostrar mensaje de confirmación en pantalla. | Se diligencia le campo nombre: Alertas estación  Se ingresa nombre campo estación: calimio tipo **String.**  Se diligencia campo observacionn: **Esto es una prueba.**  Se oprime botón guardar (El aplicativo deberá mostrar mensaje de confirmación Guardado con éxito) |
| Unitaria | crear Alerta | Se probaran los metodos agregar Alerta | agregarAlerta(id\_Alerta,nombre\_Alerta, observacion\_Alerta), se escribira prueba unitaria donde: id\_Alerta:1 nombre\_Alerta:Atlantico observacion\_Alerta: Ubicacda al norte |
| Unitaria | crear Alerta | Se probaran los metodos consultar | **consultarAlerta(id\_Alerta,nombre\_Alerta, observacion\_Alerta)se escribira prueba unitaria donde: id\_Alerta:1 nombre\_Alerta:Atlantico observacion\_Alerta: Ubicacda al norte** |
| Unitaria | crear Alerta | Se probaran los metodos actualizar | actualizarAlerta(id\_Alerta,nombre\_Alerta, observacion\_Alerta)se escribira prueba unitaria donde: id\_Alerta:1 nombre\_Alerta:Orinoquia observacion\_Alerta: Ubicacda al Occidente |
| Unitaria | crear Alerta | Se probaran los metodos consultar Alerta ID | **consultarAlertaId(nombre\_alerta)**se escribira prueba unitaria donde: nombre\_alerta(String) : calimio |
| Eficiencia | crear Organización | El usuario ingresará a la pantalla crear Organización, el sistema deberá ingresar y cargar todo el contenido en un tiempo mínimo de 5 segundo. | Se dará click en el botón nueva Organización, se ingresará a la pantalla de crear Organización y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser mínimo 5 segundos. |
| Eficiencia | crear Organización | El usuario ingresará a la pantalla crear Organización, el sistema cargará Id de la nueva Organización a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int  Se ingresa nombre campo Organización: Organización estación tipo String.   Se ingresan valores [a-z] tipo String min 50 caracteres.   Se ingresa nombre campo estación: Decepaz tipo String.  Se ingresa nombre campo observacion: Esto es una prueba tipo String.  Se ingresan valores [a-z] tipo String min 50 caracteres. |
| Eficiencia | crear Organización | El usuario ingresará a la pantalla crear Organización, el sistema cargará Id de la nueva Organización a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int Se ingresa nombre campo Organización: Valores numéricos [0-9] (El sistema no debe permitir el ingreso)  Se ingresan valores aleatorios tipo int mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación)   Se ingresa nombre campo descripción: Esto es una prueba tipo String.  Se ingresan valores [a-z] tipo String mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación) |
| Eficiencia | crear Organización | Se diligencias los campos nombre y descripción, el usuario oprime el botón guardar, el sistema deberá guardar la información en base de datos y mostrar mensaje de confirmación en pantalla. | Se diligencia le campo nombre: Organizacions estación  Se ingresa nombre campo estación: calimio tipo String.  Se diligencia campo observacionn: Esto es una prueba.  Se oprime botón guardar (El aplicativo deberá mostrar mensaje de confirmación Guardado con éxito) |
| Unitaria | crear Organización | Se probaran los metodos agregar Organización | **agregarOrganizacion(id\_Organizacion,nombre\_Organizacion, observacion\_Organizacion)**, se escribira prueba unitaria donde: id\_Organizacion:1 nombre\_Organizacion:San ambiente observacion\_Organizacion: Empresa de gestion del medio ambiente |
| Unitaria | crear Organización | Se probaran los metodos consultar | **consultarOrganizacion(id\_Organizacion,nombre\_Organizacion, observacion\_Organizacion)**se escribira prueba unitaria donde: id\_Organizacion:1 nombre\_Organizacion:Atlantico observacion\_Organizacion: Ubicacda al norte |
| Unitaria | crear Organización | Se probaran los metodos actualizar | **actualizarOrganizacion(id\_Organizacion,nombre\_Organizacion, observacion\_Organizacion)**se escribira prueba unitaria donde: id\_Organizacion:1 nombre\_Organizacion:San ambiente observacion\_Organizacion:Empresa de gestion del medio ambiente |
| Unitaria | crear Organización | Se probaran los metodos consultar Organización ID | **consultarOrganizacionId(nombre\_Organizacion)**se escribira prueba unitaria donde: nombre\_Organizacion(String) : San ambiente |
| Eficiencia | crear Categoría | El usuario ingresará a la pantalla crear Categoría, el sistema deberá ingresar y cargar todo el contenido en un tiempo mínimo de 5 segundo. | Se dará click en el botón nueva Categoría, se ingresará a la pantalla de crear Categoría y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser mínimo 5 segundos. |
| Eficiencia | crear Categoría | El usuario ingresará a la pantalla crear Categoría, el sistema cargará Id de la nueva Categoría a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int  Se ingresa nombre campo Categoria:Residuos tipo **String.**   Se ingresan valores [a-z] tipo String min 50 caracteres.    Se ingresa campo observacion: Esto es una prueba tipo **String.**  Se ingresan valores [a-z] tipo String min 50 caracteres. |
| Eficiencia | crear Categoría | El usuario ingresará a la pantalla crear Categoría, el sistema cargará Id de la nueva Categoría a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int Se ingresa nombre campo Categoría: Valores numéricos [0-9] (El sistema no debe permitir el ingreso)  Se ingresan valores aleatorios tipo int mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación)   Se ingresa nombre campo descripción: Esto es una prueba tipo String.  Se ingresan valores [a-z] tipo String mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación) |
| Eficiencia | crear Categoría | Se diligencias los campos nombre y descripción, el usuario oprime el botón guardar, el sistema deberá guardar la información en base de datos y mostrar mensaje de confirmación en pantalla. | Se diligencia le campo nombre: Residuos   Se diligencia campo observacionn: Esto es una prueba.  Se oprime botón guardar (El aplicativo deberá mostrar mensaje de confirmación Guardado con éxito) |
| Unitaria | Crear Categoría | Se probaran los metodos agregar Categoría | agregarCategoria(id\_Categoria,nombre\_Categoria, observacion\_Categoria), se escribira prueba unitaria donde: id\_Categoria:1 nombre\_Categoria:Residuos observacion\_Categoria: Empresa de gestion del medio ambiente |
| Unitaria | crear Categoría | Se probaran los metodos consultar | **consultarCategoria(id\_Categoria,nombre\_Categoria, observacion\_Categoria)**se escribira prueba unitaria donde: id\_Categoria:1 nombre\_Categoria:Residuos observacion\_Categoria: Empresa de gestion del medio ambiente |
| Unitaria | crear Categoría | Se probaran los metodos actualizar | actualizarCategoria(id\_Categoria,nombre\_Categoria, observacion\_Categoria)se escribira prueba unitaria donde: id\_Categoria:1 nombre\_Categoria:San ambiente observacion\_Categoria:Empresa de gestion del medio ambiente |
| Unitaria | crear Categoría | Se probaran los metodos consultar Categoría ID | **consultarCategoriaId(nombre\_Categoria)**se escribira prueba unitaria donde: nombre\_Categoria(String) : Residuos |
| Eficiencia | Login (Validación Usuario) | El usuario ingresará a la pantalla login, el sistema deberá ingresar y cargar todo el contenido en un tiempo mínimo de 5 segundo. | se ingresara al aplicativo, el sistema deberá ubicarnos en la pantalla principal del login y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser mínimo 5 segundos. |
| Eficiencia | Login (Validación Usuario) | El usuario no ingresa valores en los campos email y password | Se deja campo email null    se deja campo password null |
| Eficiencia | Login (Validación Usuario) | El usuario ingresará a la pantalla crear Categoría, el sistema cargará Id de la nueva Categoría a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Se ingresa email: Valores tipo texto (string) [a-z] longitud maxima de 50 carateres (El sistema no debe permitir el ingreso)   Se ingresa password: tipo string mayor a 20 caracteres. (El sistema debe mostrar mensaje de validación) y no permite el ingreso |
| Eficiencia | Login (Validación Usuario) | Se diligencias los campos nombre y descripción, el usuario oprime el botón guardar, el sistema deberá guardar la información en base de datos y mostrar mensaje de confirmación en pantalla. | Se diligencia le campo  email:  email(String)   se diligecia campo password: password(String). El sistema no deberá mostrar los caracteres en el campo password |
| Seguridad | Login (Validación Usuario) | Se probaran los metodos agregar Categoría | Se ingresa campo email: famina@famina.com  se diligencia campo email: Fabian1234.  El sistema deberá permitir el ingreso al aplicativo. |
| Seguridad Unitaria | Login (Validación Usuario) | Se probaran los metodos consultar | Se ingresa campo email: famina@famina.com  se diligencia campo email: Fabian1234.  El sistema deberá con el campo password realizar la encryptacion del texto |
| Eficiencia | crear Estación | El usuario ingresará a la pantalla crear Estación, el sistema deberá ingresar y cargar todo el contenido en un tiempo mínimo de 5 segundo. | Se dará click en el botón nuevo Estación, se ingresará a la pantalla de crear Estación y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser mínimo 5 segundos. |
| Eficiencia | crear Estación | El usuario ingresará a la pantalla crear Estación, el sistema cargará Id de la nueva Estación a crear automáticamente, el usuario diligencia los campos nombre, nombre corto estacion,serial,observación, localización, latitud,longitud, elevación, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int  Se ingresa nombre campo Estacion:Chapinero   Se ingresan valores [a-z] tipo String min 50 caracteres.    Se ingresa nombre campo nombre corto estación: Chap Se ingresan valores [a-z] tipo String min 50 caracteres.  Se ingresa nombre campo nombre corto: Chap Se ingresan valores [a-z] tipo String min 50 caracteres.   Se ingresa campo serial: chapinero Se ingresan valores [a-z] tipo String min 50 caracteres.   Se ingresa campo ,observación: Esto es una prueba. Se ingresan valores [a-z] tipo String min 50 caracteres.  Se ingresa campo ,observación: Esto es una prueba. Se ingresan valores [a-z] tipo String min 50 caracteres.  Se ingresa campo , localización: Ubicada al centro de la ciudad. Se ingresan valores [a-z] tipo String min 50 caracteres.  Se ingresa campo , latitud: Ubicada al centro de la ciudad. Se ingresan valores [a-z] tipo String min 50 caracteres.  Se ingresa campo , longitud: Ubicada al centro de la ciudad. Se ingresan valores [a-z] tipo String min 50 caracteres.  Se ingresa campo , elevacion: Ubicada al centro de la ciudad. Se ingresan valores [a-z] tipo String min 50 caracteres. |
|  |  |  |
| Eficiencia | crear Estación | El usuario ingresará a la pantalla crear Estación, el sistema cargará Id de la nueva Estación a crear automáticamente, el usuario diligencia los campos nombre, nombre corto estacion,serial,observación, localización, latitud,longitud, elevación, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int Se ingresa nombre campo Estación: Valores numéricos [0-9] (El sistema no debe permitir el ingreso)  Se ingresan valores aleatorios tipo int mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación)   Se ingresa nombre campo observacion: Esto es una prueba tipo String.  Se ingresan valores [a-z] tipo String mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación) |
| Eficiencia | crear Estación | Se diligencias los campos nombre, nombre corto estacion,serial,observación, localización, latitud,longitud, elevación, el usuario oprime el botón guardar, el sistema deberá guardar la información en base de datos y mostrar mensaje de confirmación en pantalla. | Se diligencia le campo nombre: Estación estación  nombre, nombre corto estacion,serial,observación, localización, latitud,longitud, elevació. |
| Unitaria | crear Estación | Se probaran los metodos agregar Estación | agregarEstacion(id\_estacion, nombre\_estacion, nombre\_corto\_estacion, serial\_estacion, organización\_estacion, categoría\_estacion, base\_tiempo\_estacion, observación\_estacion, región\_estacion, ciudad\_estacion, localización\_estacion, latitud\_estacion, longitud\_estacion, elevación\_estacion, gmt\_estacion, protocolo\_estacion), se escribira prueba unitaria donde: id\_estacion:**1** nombre\_estacion:**Estacion** Prueba nombre\_corto\_estacion:**Est.Prue** |
|  |  |  |
| Unitaria | crear Estación | Se probaran los metodos consultar | consultarEstacion(id\_estacion, nombre\_estacion, nombre\_corto\_estacion, serial\_estacion, ciudad\_estacion, protocolo\_estacion), se escribira prueba unitaria donde:  id\_estacion:1  se deberá cargar la información asociada a este id |
| Eficiencia | Conectar con estación | El usuario ingresará a la pantalla conectar con estación, el sistema deberá ingresar y cargar todo el contenido en un tiempo mínimo de 5 segundo. | se ingresará a la pantalla de conectar con estación y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser mínimo 5 segundos. |
| Eficiencia | Conectar con estación | Se requiere desplegar información guardada por estación | En la pantalla conectar estación, se ingrasa nombre de la estación: Chapinero |
| Eficiencia | Conectar con estación | Se requiere confirmar conexión con estación | Se oprime el boton conectar. El sistema deberá confirmar los campos obligatorios: -id -Serial  -organización -categoría -Región. -Ciudad |
| Eficiencia | Conectar con estación | Se requiere conectar con estación | Al presionar el boton confirmar, y despues de haberse realizado la confirmación de los campos el aplicativo se conectara con la estación Chapinero |
| Seguridad | Conectar con estación | Se requiere conexión mediante protocolo FTP | al presionar el boton confirmar, el aplicativo deberá realizar validación del protocolo de conexión: Se conectara mediante protocolo FTP |
| Eficiencia | crear Alerta | El usuario ingresará a la pantalla crear Alerta, el sistema deberá ingresar y cargar todo el contenido en un tiempo mínimo de 5 segundo. | Se dará click en el botón nueva Alerta, se ingresará a la pantalla de crear Alerta y se tomará el tiempo en que el aplicativo da respuesta. Este tiempo debe ser mínimo 5 segundos. |
| Eficiencia | crear Alerta | El usuario ingresará a la pantalla crear Alerta, el sistema cargará Id de la nueva Alerta a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int  Se ingresa nombre campo Alerta: Alerta estación tipo **String.**   Se ingresan valores [a-z] tipo String min 50 caracteres.   Se ingresa nombre campo estación: Decepaz tipo **String.**  Se ingresa nombre campo observacion: Esto es una prueba tipo **String.**  Se ingresan valores [a-z] tipo String min 50 caracteres. |
| Eficiencia | crear Alerta | El usuario ingresará a la pantalla crear Alerta, el sistema cargará Id de la nueva Alerta a crear automáticamente, el usuario diligencia los campos nombre, descripción, los campos solo recibirán máximo 50 caracteres. | Id carga automáticamente tipo int Se ingresa nombre campo Alerta: Valores numéricos [0-9] (El sistema no debe permitir el ingreso)  Se ingresan valores aleatorios tipo int mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación)   Se ingresa nombre campo descripción: Esto es una prueba tipo String.  Se ingresan valores [a-z] tipo String mayor a 50 caracteres. (El sistema debe mostrar mensaje de validación) |
| Eficiencia | crear Alerta | Se diligencias los campos nombre y descripción, el usuario oprime el botón guardar, el sistema deberá guardar la información en base de datos y mostrar mensaje de confirmación en pantalla. | Se diligencia le campo nombre: Alertas estación  Se ingresa nombre campo estación: calimio tipo **String.**  Se diligencia campo observacionn: **Esto es una prueba.**  Se oprime botón guardar (El aplicativo deberá mostrar mensaje de confirmación Guardado con éxito) |
| Unitaria | crear Alerta | Se probaran los metodos agregar Alerta | agregarAlerta(id\_Alerta,nombre\_Alerta, observacion\_Alerta), se escribira prueba unitaria donde: id\_Alerta:1 nombre\_Alerta:Atlantico observacion\_Alerta: Ubicacda al norte |
| Unitaria | crear Alerta | Se probaran los metodos consultar | **consultarAlerta(id\_Alerta,nombre\_Alerta, observacion\_Alerta)se escribira prueba unitaria donde: id\_Alerta:1 nombre\_Alerta:Atlantico observacion\_Alerta: Ubicacda al norte** |
| Unitaria | crear Alerta | Se probaran los metodos actualizar | actualizarAlerta(id\_Alerta,nombre\_Alerta, observacion\_Alerta)se escribira prueba unitaria donde: id\_Alerta:1 nombre\_Alerta:Orinoquia observacion\_Alerta: Ubicacda al Occidente |
| Unitaria | crear Alerta | Se probaran los metodos consultar Alerta ID | **consultarAlertaId(nombre\_alerta)**se escribira prueba unitaria donde: nombre\_alerta(String) : calimio |

Tabla 34 Datos de prueba Eficiencia y Unitarias

Autor: Fabian Mina – Daniela Romero

Datos Pruebas de Carga

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tipo de Prueba** | **Funcionalidad** | **Descripción Prueba** | **Datos de prueba** | **Elementos a Probar** | **Resultados Esperados** |
|
|
| Pruebas de carga | crear región | Se requiere someter el aplicativo con la ayuda del plugin Jmeter, a verificar cuantas consultas de usuarios soporta la aplicación. | Consultar la cantidad de usuarios que pueden consultar las regiones creadas | **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad maxima de 100 usuarios conectandose simultaneamente al aplicativo |
| Pruebas de carga | crear ciudad | Se requiere someter el aplicativo con la ayuda del plugin Jmeter, a verificar cuantas consultas de usuarios soporta la aplicación. | Consultar la cantidad de usuarios que pueden consultar las ciudades creadas | **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad maxima de 100 usuarios conectandose simultaneamente al aplicativo |
| Pruebas de carga | crear Mantenimiento | Se requiere someter el aplicativo con la ayuda del plugin Jmeter, a verificar cuantas consultas de usuarios soporta la aplicación. | Consultar la cantidad de usuarios que pueden consultar los mantenimientos creados | **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad maxima de 100 usuarios conectandose simultaneamente al aplicativo |
| Pruebas de carga | crear Alerta | Se requiere someter el aplicativo con la ayuda del plugin Jmeter, a verificar cuantas consultas de usuarios soporta la aplicación. | Consultar la cantidad de usuarios que pueden consultar las alertas creadas | **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad maxima de 100 usuarios conectandose simultaneamente al aplicativo |
| Pruebas de carga | crear Organización | Se requiere someter el aplicativo con la ayuda del plugin Jmeter, a verificar cuantas consultas de usuarios soporta la aplicación. | Consultar la cantidad de usuarios que pueden consultar las organizaciones creadas | **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad maxima de 100 usuarios conectandose simultaneamente al aplicativo |
| Pruebas de carga | crear Categoría | Se requiere someter el aplicativo con la ayuda del plugin Jmeter, a verificar cuantas consultas de usuarios soporta la aplicación. | Consultar la cantidad de usuarios que pueden consultar las categorias creadas | **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad maxima de 100 usuarios conectandose simultaneamente al aplicativo |
| Pruebas de carga | Validar Usuario | Se requiere someter el aplicativo con la ayuda del plugin Jmeter, a verificar cuantas consultas de usuarios soporta la aplicación. | Validar la cantidad de usuarios maximos que pueden conectarse simultaneamente al aplicativo | **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad maxima de 100 usuarios conectandose simultaneamente al aplicativo |
| Pruebas de carga | Conectar con estación | Se requiere someter el aplicativo con la ayuda del plugin Jmeter, a verificar cuantas consultas de usuarios soporta la aplicación. | Validar cuantos usuarios soporta el aplicativo al intentar conectarse a una estación. | **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad maxima de 100 usuarios conectandose simultaneamente al aplicativo |
| Pruebas de carga | Crear estación | Se requiere someter el aplicativo con la ayuda del plugin Jmeter, a verificar cuantas consultas de usuarios soporta la aplicación. | Consultar la cantidad de usuarios que pueden consultar las estaciones creadas | **Usuarios soportados** | Se espera que el aplicativo soporte una cantidad maxima de 100 usuarios conectandose simultaneamente al aplicativo |

Tabla 35 Datos prueba de carga

Autor: Fabian Mina – Daniela Romero

#### Pruebas de usabilidad

Para las pruebas de usabilidad realizamos un test de usuario que se basa en la observación y análisis de cómo un grupo de usuarios reales manipulan el software y cuál es su comportamiento en cada actividad que realice, esta prueba es recomendable usarla para descubrir errores de diseño motivados por el no cumplimiento en el desarrollo de principios generales de usabilidad

Para el test de usuario se evaluaron las siguientes actividades.

* Seleccionar estación para conexión
* Crear Mantenimiento
* Crear región
* Crear estación
* Crear organización
* Crear Ciudad
* Crear rango
* Crear alerta
* Crear cuadro de tiempo
* Login\*

Para el test de usuario se entrevistaron a 10 personas entre los 22 y 40 años de edad las cuales no conocían el funcionamiento del aplicativo; consisten en realizar una encuesta donde se le pidió al entrevistado realizar tareas como seleccionar la opción para crear una región, seleccionar la opción para completar el formulario de creación de la región, buscar y seleccionar la opción de guardar y obtener como resultado que tan amigable es el software para las partes interesadas; Se observa y evalúa que comportamientos tiene el usuario al interactuar con el aplicativo.

Se definieron para el test de usuario algunas tareas donde debemos seleccionar alguna de las opciones dependiendo del comportamiento del usuario

* El usuario Cumplió con éxito rápidamente.
* El usuario tardo bastante en encontrar el botón, pero lo hizo con autonomía.
* El usuario solicitó su ayuda para encontrarlo.
* La aplicación no responde.
* El usuario desiste.
* El usuario cumplió con rapidez.
* El usuario tardó bastante en completar el formulario, pero lo hizo con autonomía.
* El usuario solicitó su ayuda porque no entendió que significaba algún campo.
* El usuario solicitó su ayuda porque el tamaño del texto era muy pequeño o estaba mal escrito.
* El usuario solicitó su ayuda porque no tenía claro como escribir la región.
* El usuario solicito ayuda porque no sabía que poner en las observaciones.
* El usuario dejo en blanco el campo observaciones.
* El sistema informo al usuario que guardo con éxito
* El usuario salió seguro de haber guardado

**Ejemplo de Test de usuario prueba exitosa**

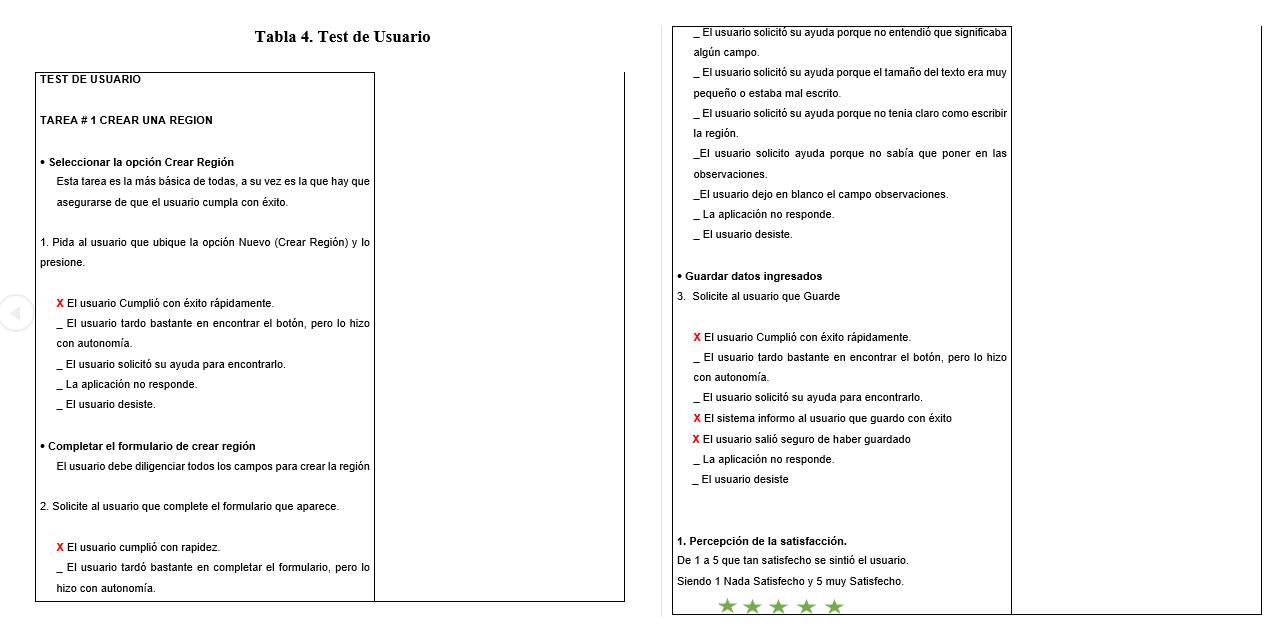


Figura 9 Test de usuario

Autor: Zapata2017

#### Pruebas funcionales:

Incluyen lo que son las pruebas unitarias y pruebas de integración, con las cuales se evaluara fragmentos de código de cada componente para validar que se cumpla la funcionalidad exitosamente, para que así se entregue al usuario final un producto útil y sin errores.

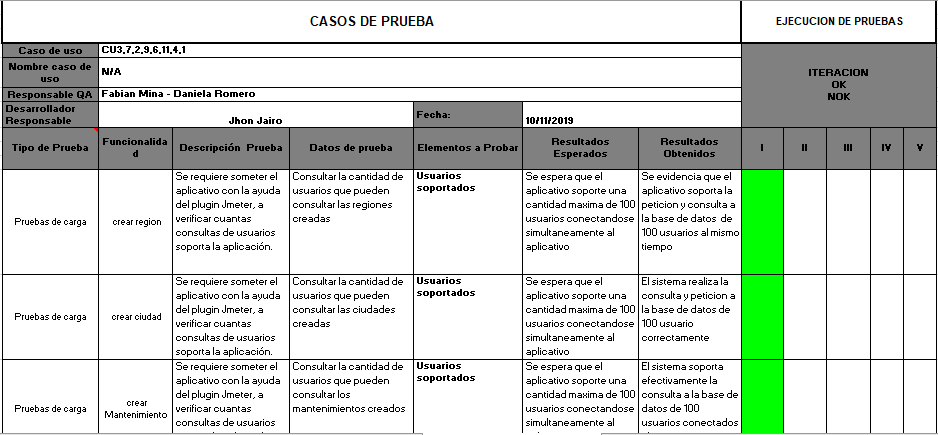
**Formato registro de pruebas de carga**

Figura 10 Formato registro de pruebas

Autor: Fabian Mina – Daniela Romero

#### Pruebas eficiencia

Se probo el comportamiento del aplicativo y los tiempos de respuesta a las peticiones realizadas por los usuarios, para ello se realizaron los casos de prueba (Figura 10) donde se establece los tiempos y los resultados esperados; para llevar un control a la hora de realizar pruebas se diseña un formato donde se debe ir verificando si los tiempos de espera en el software cumplieron con lo establecido.

#### Pruebas seguridad

Las pruebas de seguridad se dejaron planteadas en el registro de casos de prueba, ya que este módulo aún se encuentra en etapa de desarrollo; Se probará la, autenticación de los usuarios y encriptación de contraseñas en base de datos, para esta prueba utilizamos el CU11

Validación de usuario

A continuación, se muestra cómo se realizó el test basándonos en el caso de prueba de CU 11

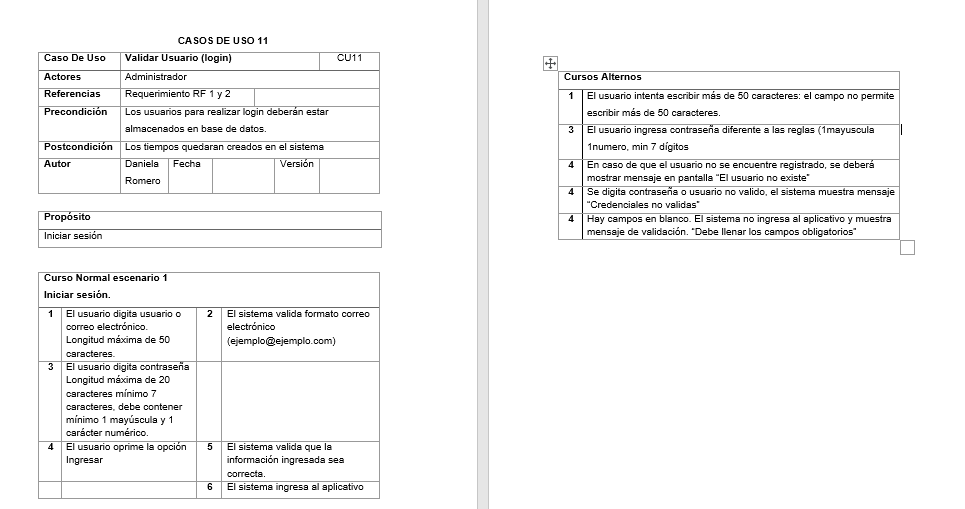


Figura 11 Ejemplo caso de uso 11

Autor: Fabian Mina – Daniela Romero

### Procedimiento de pruebas

Herramientas que se usaran para la elaboración de este proyecto.

**Framework Jest:** Se validaron fragmentos de código de cada componente, probando los métodos completos, crear, consultar, actualizar, ver; la respuesta deberá ser positiva (pass) para cada test de prueba; con el framework jest lo que se hizo fue escribir la prueba unitaria que permitió validar si el método que se desarrolló, cumplió correctamente la funcionalidad requerida, esto se hizo con cada uno de los métodos.

**Carga masiva de datos:** Se realiza con la herramienta Jmeter V.5.2, se verificará la conexión de la herramienta con el gestor de base de datos PostgreSQL, se realizó una carga de 100 usuario en un término de 15 segundos, esta prueba se realizó configurando el puerto del servidor que en su defecto fue 4200, se configuro el servidor o ip en este caso el aplicativo se corrió localmente lo que quiere decir que se utilizó el servidor localhost y se configura la ruta en la que se encuentra alojada el componente.

### Reporte de transmisión de pruebas

Se Describió cada uno de los ítems para pruebas, donde encontrarlos y se aprueba para su liberación. Es muy importante ya que garantiza al encargado de pruebas tester QA de que los ítems están listos para ser probados, este reporte permitió llevar un control o bitácora de cada uno de los componentes que pasaron por el proceso de pruebas.

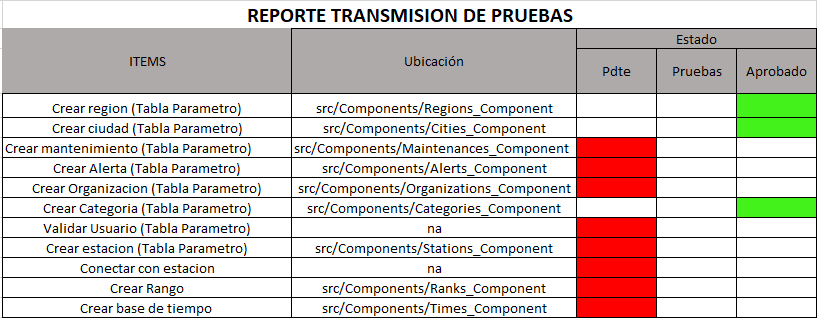


Figura 12 Reporte transmisión de pruebas

Autor: Fabian Mina – Daniela Romero

Las pruebas realizadas en este proyecto se realizaron con las siguientes herramientas integrándolas para una correcta ejecución. Ver figura 12

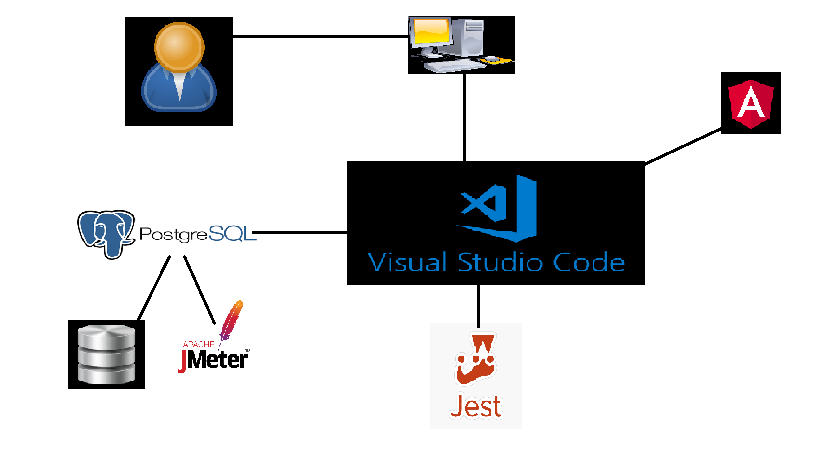


Figura 13 Diagrama herramienta usadas

Nota. Autor: Fabian Mina – Daniela Romero

Con el diagrama anterior se muestra la estructura e integración de las herramientas para la ejecución de pruebas del aplicativo.

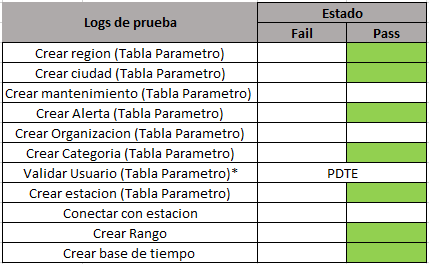
## Ejecución de pruebas

### Log de pruebas

En este documento se registraron los detalles sobre qué casos de pruebas se han ejecutado, en que orden y sus resultados (Pass/fail). Si se presentaron inconsistencias, se levantó o se actualizo un reporte de incidencias.

A continuación, se muestra registro llamado log de pruebas donde se evidencia el estado de cada componente si pasa el test de prueba se registra como Pass, si no paso el test de pruebas se registra como Fail. Ver figura 14

Figura 14 Logs de prueba



Autor: Fabian Mina – Daniela Romero

#### Pruebas de usabilidad

Se realizaron las pruebas de usabilidad con un test de usuario que se basa en la observación y análisis de cómo un grupo de usuarios reales manipulan el software y cuál es su comportamiento en cada actividad que realice; consiste en una evaluación a 10 personas entre los 22 y 40 años de edad la cual se observa y evalúa que comportamientos tiene el usuario al interactuar con el aplicativo.

|  |  |
| --- | --- |
| ***TEST DE USUARIO***  ***NOMBRE: AARON GRAJALES***  ***TAREA CREAR UNA CIUDAD***   * ***Seleccionar la opción Crear Ciudad***   *Esta tarea es la más básica de todas, a su vez es la que hay que asegurarse de que el usuario cumpla con éxito.*   1. *Pida al usuario que ubique la opción Nuevo (Crear Ciudad) y lo presione.*   ***X*** *El usuario Cumplió con éxito rápidamente.*  *\_El usuario tardo bastante en encontrar el botón, pero lo hizo con autonomía.*  *\_ El usuario solicitó su ayuda para encontrarlo.*  *\_ La aplicación no responde.*  *\_ El usuario desiste.*   * ***Completar el formulario de crear Ciudad***   *El usuario debe diligenciar todos los campos para crear la ciudad*   1. *Solicite al usuario que complete el formulario que aparece.*   ***X*** *El usuario cumplió con rapidez.*  *\_ El usuario tardó bastante en completar el formulario, pero lo hizo con autonomía.*  *\_ El usuario solicitó su ayuda porque no entendió que significaba algún campo.*  *\_ El usuario solicitó su ayuda porque el tamaño del texto era muy pequeño o estaba mal escrito.*  *\_ El usuario solicitó su ayuda porque no tenía claro como escribir el nombre de la ciudad*  *\_El usuario solicito ayuda porque no sabía que poner en las observaciones.*  *\_El usuario dejo en blanco el campo observaciones.*  *\_ La aplicación no responde.*  *\_ El usuario desiste.*   * ***Tarea guardar datos ingresados***  1. *Solicite al usuario que Guarde*   ***X*** *El usuario Cumplió con éxito rápidamente.*  *\_ El usuario tardo bastante en encontrar el botón, pero lo hizo con autonomía.*  *\_ El usuario solicitó su ayuda para encontrarlo.*  *\_ El sistema informo al usuario que guardo con éxito*  *\_El usuario salió seguro de haber guardado*  *\_ La aplicación no responde.*  *\_ El usuario desiste*  ***1.Percepción de la satisfacción.***  *De 1 a 5 que tan satisfecho se sintió el usuario.*  *Siendo 1 Nada Satisfecho y 5 muy Satisfecho.*  ***Observaciones.*** *(Si nota algo especialmente particular proceda a escribirlo por favor. Recuerde: Palabras o frases dichas por el usuario o preguntas que este le haga a usted).*  *El usuario fue muy eficiente a la hora de ubicar los campos y de ordenarle cualquier funcionalidad* |  |

Figura 15 pruebas Usabilidad

Autor: Fabian Mina – Daniela Romero

#### Pruebas de funcionalidad

Se realizaron estas pruebas funcionales de forma manual, probando la funcionalidad de cada acción del aplicativo, entre las cuales se encuentra el tiempo de inicio a la pantalla principal, mensajes mostrados en pantalla, guardar información, mensajes de confirmación, actualizar información, visualizar la información almacenada en base de datos.

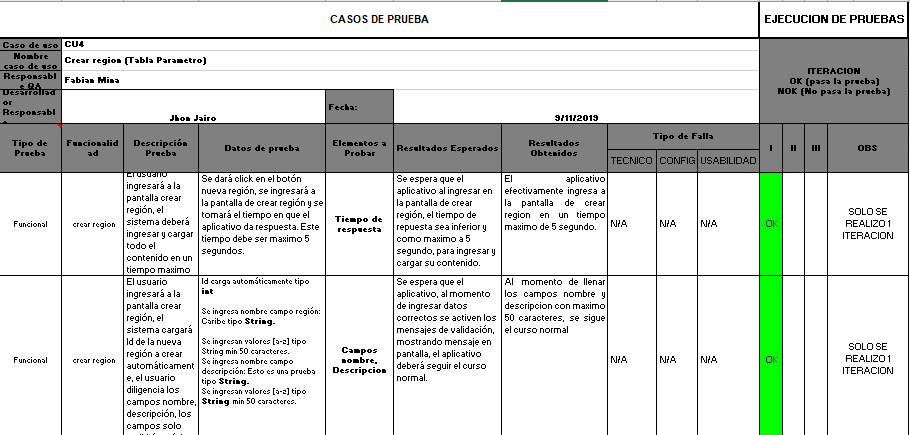


Figura 16 pruebas funcionales

Autor: Fabian Mina – Daniela Romero

#### Pruebas unitarias

Se escribieron pruebas unitarias con la herramienta Jest, realizando un test completo de cada método. Se probaron los métodos guardar, crear, consultar, modificar.

Se probaron los componentes Ciudad, Región, Categoría. Al ejecutar la prueba con la herramienta jest, esta realiza un testeo a todo el componente y revisa que los métodos cumplan cada uno con su función.

A continuación, se define los resultados:

**Test suites:** Define a cuantos componentes se les aplico el test de prueba

**Test:** Define a cuantos métodos de les aplico el test de prueba unitaria

**Pass:** Define que el modulo paso la prueba satisfactoriamente

**Passed:** Define si el método al cual se le aplico el test paso la prueba satisfactoriamente

**Fail:** Define si el método al cual se le aplico el test no pasa la prueba

**Snapshots:** definen las capturas o estados especificado de nuestro aplicativo

**Time:** Define el tiempo el cual el test realiza el recorrido por todo el método y sus componentes.

En las siguientes figuras se evidencia el resultado de las pruebas unitarias aplicadas a los componentes de Región, Ciudad, Categoría, Base de tiempo, Rango, Alerta, Estación, Inicio

Se realizo el test de prueba al componente región y a cada uno de los módulos, el cual obtuvo como resultado Pass, lo cual significa que la prueba paso satisfactoriamente

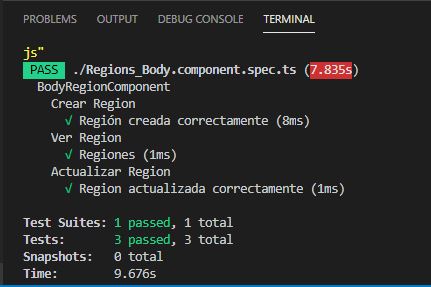


Figura 17 Resultado prueba unitaria componente región

Autor: Fabian Mina – Daniela Romero

Se realizo el test de prueba al componente Ciudad y a cada uno de los módulos, el cual obtuvo como resultado Pass, lo cual significa que la prueba paso satisfactoriamente

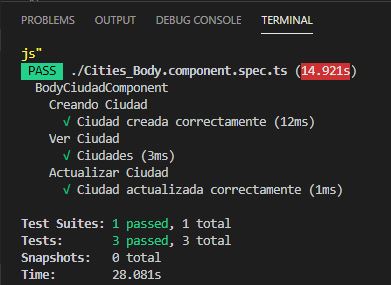


Figura 18 Resultado prueba unitaria componente ciudad

Autor: Fabian Mina – Daniela Romero

Se realizo el test de prueba al componente categoria y a cada uno de los módulos, el cual obtuvo como resultado Pass, lo cual significa que la prueba paso satisfactoriamente

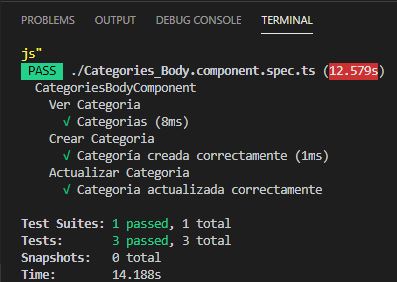


Figura 19 Resultado prueba unitaria componente categoría

Autor: Fabian Mina – Daniela Romero

Se realizo el test de prueba al componente Alerta y a cada uno de los módulos, el cual obtuvo como resultado Pass, lo cual significa que la prueba paso satisfactoriamente

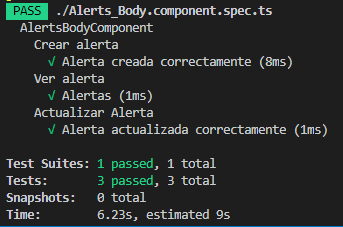


Figura 20 Resultado prueba unitaria componente Alerta

Autor: Fabian Mina – Daniela Romero

Se realizo el test de prueba al componente Estación y a cada uno de los módulos, el cual obtuvo como resultado Pass, lo cual significa que la prueba paso satisfactoriamente

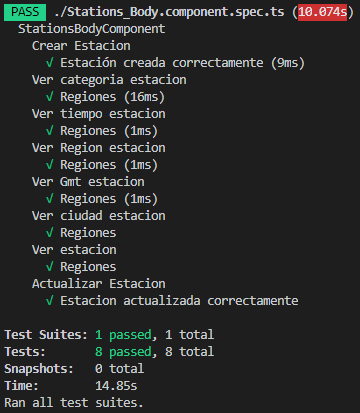


Figura 22 Resultado prueba unitaria componente Estación

Autor: Fabian Mina – Daniela Romero

Se realizo el test de prueba al componente Rango y a cada uno de los módulos, el cual obtuvo como resultado Pass, lo cual significa que la prueba paso satisfactoriamente

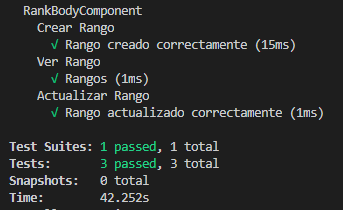


Figura 23 Resultado prueba unitaria componente Rango

Autor: Fabian Mina – Daniela Romero

Se realizo el test de prueba al componente Base de tiempo y a cada uno de los módulos, el cual obtuvo como resultado Pass, lo cual significa que la prueba paso satisfactoriamente

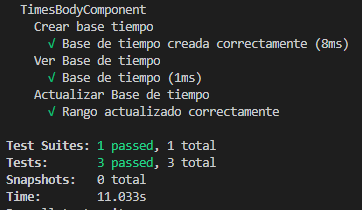


Figura 23 Resultado prueba unitaria componente Base de tiempo

Autor: Fabian Mina – Daniela Romero

Se realizo el test de prueba a la pantalla principal, el cual obtuvo como resultado Pass, lo cual significa que la prueba paso satisfactoriamente

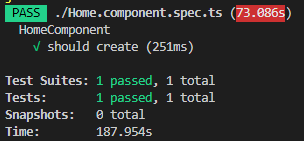


Figura 24 Resultado prueba unitaria componente Inicio

Autor: Fabian Mina – Daniela Romero

#### Pruebas de carga

Se realizaron las pruebas de carga con la herramienta Jmeter, donde se simularon 500 usuarios ingresando simultáneamente en un ciclo de 15 segundos.

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

A continuación, se definen los códigos de después de HTTP

100 continue:Define que todo se encuentra correcto dentro de las peticiones al servidor

200 ok:define que la solicitud al servidor ha sido exitosa

202 Accepted:Define que la solicitud ha sido recibida

400 Bad Request:Define que el servidor no ha podido interpretar la solicitud

403 Forbidden:Define que el usuario no cuenta con los permisos para acceder

404 no found:Define que el servidor no ha encontrado el contenido solicitado

500 internal server error**:** Define que el servidor ha generado un error interno

Para la prueba que se realizó como código de respuesta de HTTP se obtuvo como resultado el código 200, para cada una de las consultas hechas en cada uno de los componentes.

A continuación, se evidencia los resultados de las consultas realizadas a los métodos Ciudad, Región y Categoría

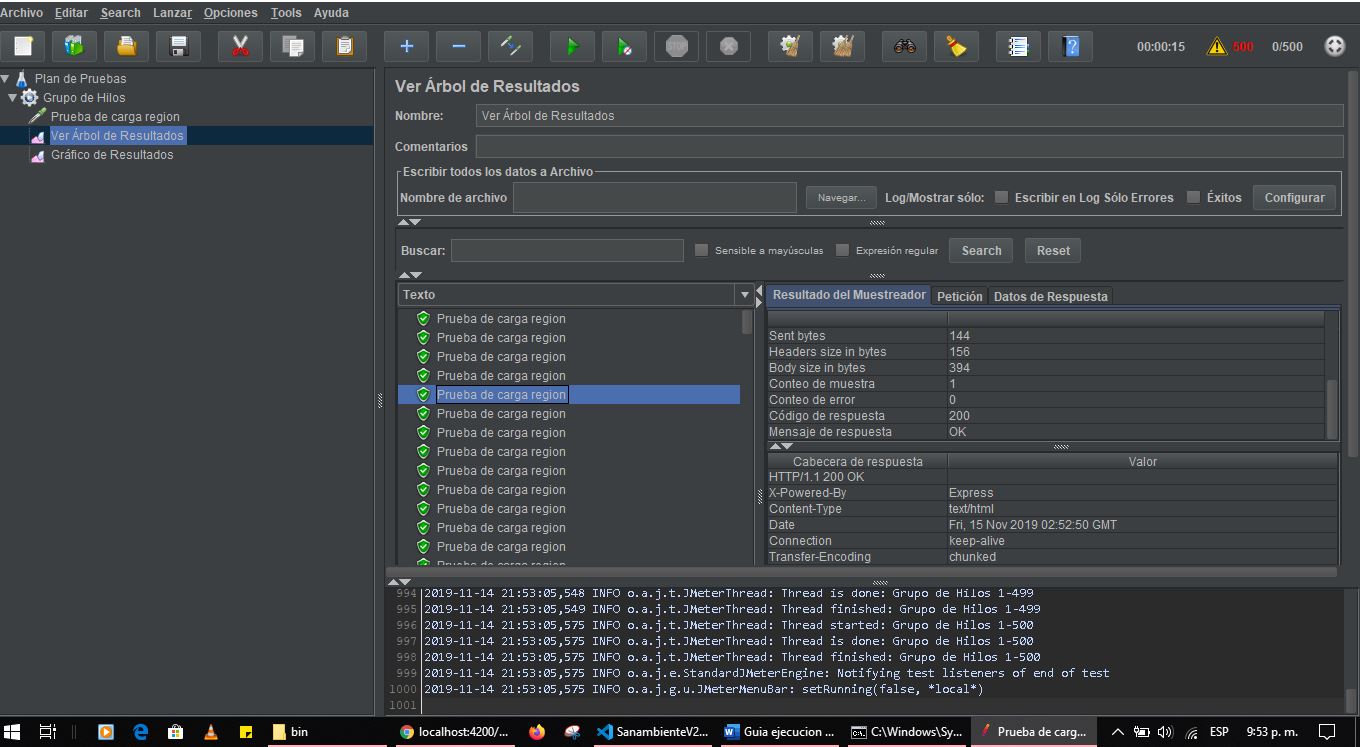


Figura 25Resultado prueba de carga componente región

Autor: Fabian Mina – Daniela Romero

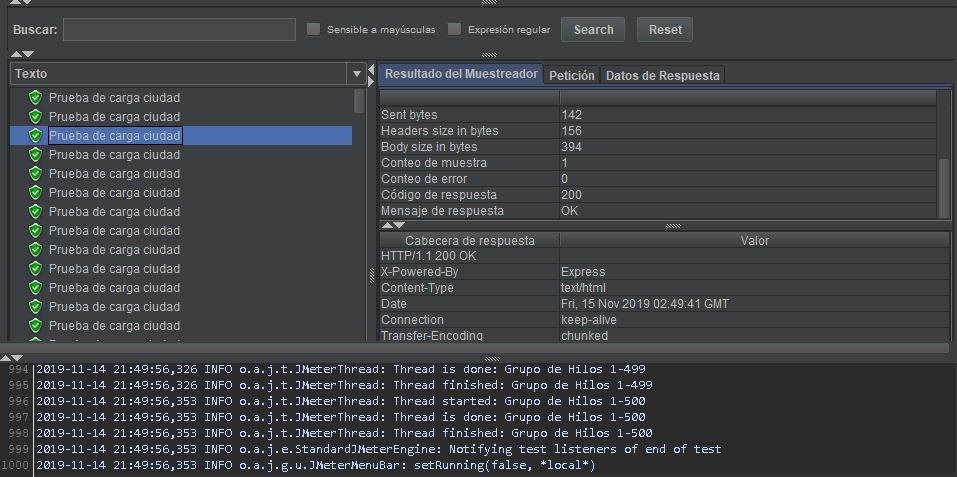


Figura 26 Resultado prueba de carga componente Ciudad

Autor: Fabian Mina – Daniela Romero

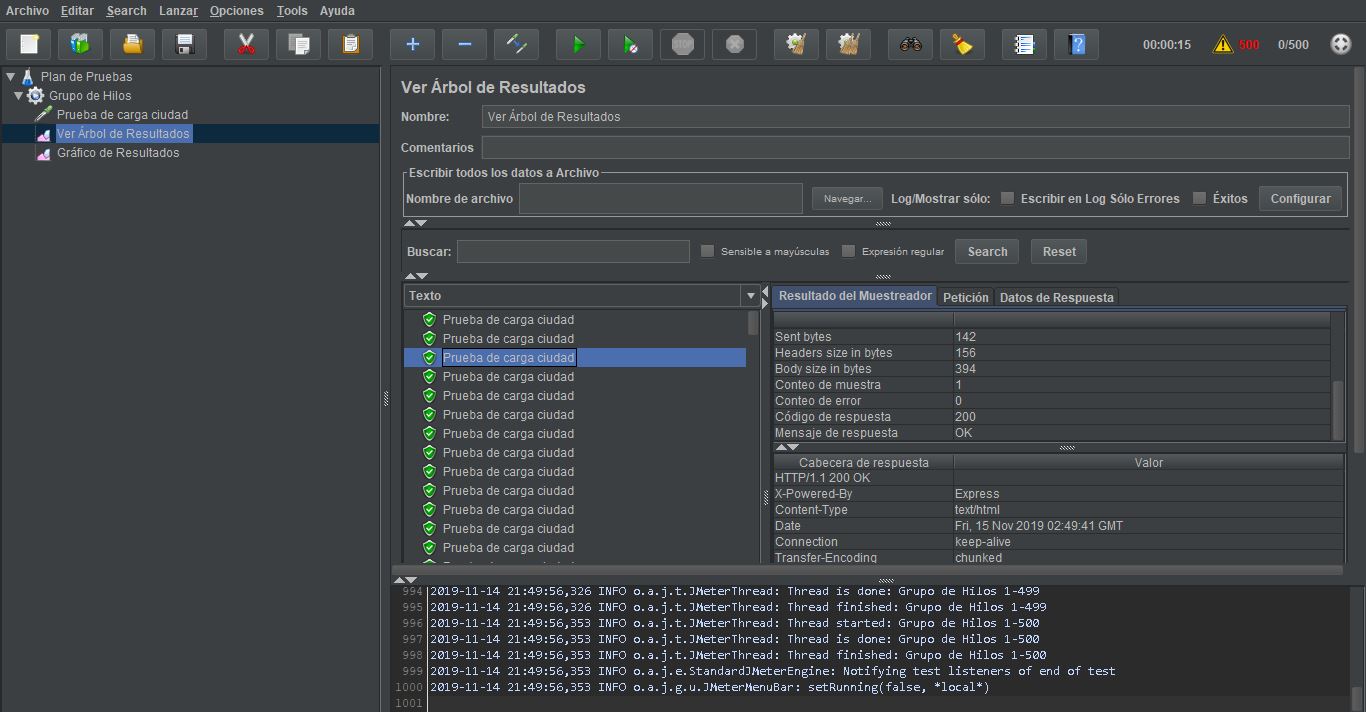


Figura 27 Resultado prueba de carga componente Categoría

Autor: Fabian Mina – Daniela Romero

### Reporte de incidentes de pruebas

Durante la ejecución de pruebas, se registra un reporte de incidencia, donde se evidencian todas las inconformidades o errores presentados al momento de realizarle pruebas a los componentes.

Tabla 36 Reporte de incidencias

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Reporte de incidentes de pruebas** | | | | |
|  |  | | | |
| **CU** | | **ELEMENTO PROBADO** | **ITERACION** | **DETALLE** |
| CU 3  Crear región | | Botón Guardar | I | Cuando se llenan los campos nombre y descripción, y se da en la opción guardar el aplicativo no muestra mensaje de confirmación. |
| CU 7  Crear ciudad | | Botón Guardar | I | Al momento de guardar la información, y presionar el botón guardar, no se muestra mensaje de confirmación |
| CU 6  Crear Organización | | Guardar Información | I | Cuando se llena el formulario con toda la información y dar en el botón guardar, la información no está quedando almacenada en la base de datos |
| CU 10  Crear Alerta | | Guardar Información | I | Cuando se llena el formulario con toda la información y dar en el botón guardar, la información no está quedando almacenada en la base de datos |
| CU 9  Crear Rango | | Guardar Información | I | Cuando se llena el formulario con toda la información y dar en el botón guardar, la información no está quedando almacenada en la base de datos |

Autor: Fabian Mina – Daniela Romero

## Termino de pruebas

Una vez terminado el proceso de prueba, se realiza una evaluación de los resultados de las pruebas, que tan bien salieron, el número de incidentes reportados, evaluación de la calidad del sistema.

### Reporte de pruebas

|  |  |
| --- | --- |
| **REPORTE DE PRUEBAS**  **Evaluación** | |
| **Unitarias**  **Numero de componentes probados** | **7** |
| **Numero de test aplicados** | **7** |
| **Numero de incidencias reportadas** | **0** |
| **Porcentaje de incidencias 0%**  **Tipo de Incidencias n/a**  **Eficiencia**  **Numero de componentes probados 7**  **Numero de test aplicados 7**  **Numero de incidencias 0**  **Porcentaje de incidencias 0%**  **Tipo de incidencia n/a**  **Funcionabilidad**  **Numero de componentes probados 7**  **Numero de test aplicados 35**  **Numero de incidencias 5**  **Porcentaje de incidencias 6%**  **Tipo de incidencia configuración**  **Usabilidad**  **Numero de componentes probados 7**  **Numero de test aplicados 10**  **Numero de incidencias 0**  **Porcentaje de incidencias 0%**  **Tipo de incidencia n/a**  **Observación.**  Al realizar las pruebas a los componentes Región, Ciudad, Categoría, Base de tiempo, Rango, Alerta, Estación, Inicio, se puede evidenciar que el aplicativo en etapa de desarrollo ha cumplido con los requerimientos establecidos desde un principio, se evidencia que el porcentaje de incidencias sobre el número de test aplicados es del 4,4%, lo cual es un valor mínimo de incidencias, el tipo de incidencias presentadas han sido por configuración se espera la integración de los componentes faltantes para así asegurar la calidad de los mismos y lograr entregar un producto útil para el usuario final. | |

Tabla 37 Reporte de pruebas

Autor: Fabian Mina – Daniela Romero

Se realizó mediante gráficos los resultados de la evaluación los cuales se muestran a continuación.

En la evaluación se establecieron la cantidad de componentes que fueron probados, la cantidad de test o pruebas que se realizaron, el numero de incidencias encontradas y el porcentaje de incidencias respecto al numero de componentes probados.

# Conclusiones

Para concluir se demuestra que al aplicar un modelo de pruebas es una buena opción para diagnosticar si el producto es confiable y de buena calidad, con ello al aplicar este modelo de pruebas al aplicativo, se demuestra que va en un proceso de desarrollo viable y se evidencia que aun esta presto a mejoras continuas para futuras versiones, ya que el aplicativo está en etapa de desarrollo y proceso de madurez.

En la ejecución de las pruebas de calidad fue notorio que el aplicativo cumple con los requerimientos y especificaciones que desde un inicio fueron definidos, el aplicativo se encuentra en una etapa de madurez lo que lleva a que se pueda someter a futuras modificaciones y mejoras.

Para obtener un software con calidad se requieren la utilización de metodologías y procedimientos para llevar a cabo una correcta practica que garantice la buena ejecución de procesos que cumple requerimientos específicos y las necesidades o expectativas del cliente.

Para este proyecto fue importante el seguimiento de documentos en base a la IEEE 829 la cual nos ayudó a aplicar las fases de calidad y llevar una buena estructura a la hora de la implementación de cada de una de las pruebas.

Las herramientas escogidas para la elaboración de pruebas con las cuales se ejecutaron y aplicaron las pruebas se adaptaron a las condiciones y a lo que se esperaba, el uso de cada una de estas herramientas fue de fácil manejo y los resultados que arroja estas herramientas son de fácil comprensión e interpretación para quien realiza pruebas de software, por el tipo de lenguaje en que fue escrito el código del aplicativo, se tiene la ventaja de usar múltiples herramientas en las que se pueda desplegar y realizar las pruebas del mismo.

# Bibliografía

Callejas-Cuervo, M., Alarcón-Aldana, A. C., & Álvarez-Carreño, A. M. (2017). Modelos de calidad del software, un estado del arte. *ENTRAMADO*, *13*(1), 236-250. https://doi.org/10.18041/entramado.2017v13n1.25125

Chumacero Botet, I. (2010). Sistema de Calidad Total. Metodología y ejemplo de aplicación a Empresas de Diseño. *Revista de Arquitectura e Ingeniería*, *4*(2). Recuperado de http://www.redalyc.org/resumen.oa?id=193915954005

CS\_07\_Estandares\_para\_pruebas\_software.pdf. (s. f.). Recuperado de http://artemisa.unicauca.edu.co/~cardila/CS\_07\_Estandares\_para\_pruebas\_software.pdf

ESTÁNDARES DE CALIDAD DEL SOFTWARE. (s. f.). Recuperado 7 de octubre de 2018, de http://estandarescalidadsoftware.blogspot.com/

Fernando Arciniega. (2017, mayo 23). Normas y Estándares de calidad para el desarrollo de Software. Recuperado 7 de octubre de 2018, de https://fernandoarciniega.com/normas-y-estandares-de-calidad-para-el-desarrollo-de-software/

IEEE 829 Norma para el software y la documentación de prueba. (s. f.). Recuperado 25 de noviembre de 2018, de https://prezi.com/yh-yob98\_ayd/ieee-829-norma-para-el-software-y-la-documentacion-de-prueba/

Metodologias De Un Modelo De Calidad. (s. f.). Recuperado 25 de noviembre de 2018, de http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/tqm/3\_metodologia/3\_metodologia.htm

NORMAS ISO 25000. (s. f.). Recuperado 7 de octubre de 2018, de https://iso25000.com/index.php/normas-iso-25000

Scalone, L. F. (s. f.). MAESTRIA EN INGENIERIA EN CALIDAD, 461.

Pruebas de software

«Pruebas de software», s. f. <https://www.iti.es/servicios/calidad-de-software/pruebas-de-software/>.