## Especificación del diseño de pruebas

Este plan de pruebas es elaborado con la finalidad de elegir los componentes que se van a probar, de esta forma se busca validar que los requerimientos funcionales y no funcionales del aplicativo desarrollado para la empresa San Ambiente se efectúen. Igualmente, el documento de pruebas permite continuar con el proceso de evolución del proyecto, de esta manera se espera llegar satisfactoriamente a una correcta ejecución en el proceso de pruebas.

Al ejecutar el plan de pruebas, es posible recopilar información sobre las posibles no conformidades o incidencias, que se presenten en el aplicativo, de esa forma se informa de manera oportuna correcciones que se deban emplear y ajustar al programa, buscando así asegurar el correcto funcionamiento y cumplimiento de los requerimientos.

Con el plan de pruebas se busca ejecutar pruebas unitarias con la herramienta Visual Studio, realizar pruebas de análisis y depuración de código para así ver que errores hay dentro de él, al momento de ser escrito y así realizar las correcciones necesarias, se realizaran pruebas de carga masiva o de Strees con aplicación JMeter se le harán las consultas a la base de datos que está alojada en el gestor de base de datos PostgreSQL.

De esta misma forma se realizarán pruebas de usabilidad, las cuales serán ejecutadas de forma manual y tendrán su propio formato de aceptación.

**Lista De Requerimientos**

Se listan a continuación los requerimientos por prioridades del aplicativo para la empresa San Ambiente con el fin de validar las funciones que se van a probar

|  |  |
| --- | --- |
| **ID** | **Descripción** |
|  | **Comunicación y traspaso de información** |
| RQ1 | El sistema debe permitir el enlace y transferencia de datos de los datalogger 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 pueden 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 crear plantillas para leer los archivos planos y cargar los datos al sistema. |
| RQ7 | El sistema debe permitir acceder al FTP de acuerdo con la base de tiempo de las estaciones, leer los archivos de texto y almacenar la información en la base de datos. |
| **ID** | **Descripción** |
|  | **Procesamiento** |
| RQ14 | El sistema debe permitir recibir o capturar los datos de las estaciones y almacenarlos en la base de datos teniendo en cuenta que los datos pasan por dos formas de almacenar, la primera son los datos crudos, como llegan de las estaciones, la segunda son los datos editados, los cuales ya pasaron por el procesamiento de datos. |
| RQ15 | El sistema debe permitir crear reglas o criterios de validación de la siguiente manera: rangos máximos, mínimos, valores de pico, valores constantes, valores negativos, % de variación, pérdida de datos parciales o totales. |
| RQ16 | Permitir correcciones, estados, fórmulas lineales, fórmulas extendidas, corrección de alertas en los datos históricos por parámetro ambiental de forma manual por un rango de tiempo. |
| RQ17 | El sistema debe permitir que el usuario pueda crear plantillas de procesamiento de información para los parámetros ambientales, con lo anterior se podrá ej. SO2 en valor alto 200 ppb, valor bajo 30 ppb, valor constante, valor pico de 5%, etc. Que pueden ser aplicados a X estaciones del mismo cliente. (O de otros clientes) Sin necesidad de volverlos a crear. |
| RQ18 | El sistema debe permitir bloquear por un usuario de permisos altos como (Editor final) intervalos de tiempo donde los datos hayan sido validados tanto automática como manualmente, con el fin que no puedan ser modificados alterando la información oficial. |

## Especificación de casos de prueba

Componentes a evaluar

Para la evaluación de componentes se elaboró una tabla dividiendo los módulos que serán sometidos a pruebas, especificando que métodos de la aplicación serán evaluados y que tipo de pruebas se realizaron

Tabla 6: Tabla evaluación de componentes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MÓDULO | PRUEBAS | DESCRIPCIÓN | Datos  Entrada | Resultados esperados |
| Interfaz gráfica | Pruebas de usabilidad | Mostrar la facilidad con la que el usuario pueda ingresar al aplicativo. | Se elaboraron pruebas manuales. Se selecciona un grupo de estudiantes. | Facilidad de uso, vista de fácil acceso para el usuario. |
| Pruebas caja Negra | Verificar la funcionalidad del aplicativo.  Casos de uso a evaluar: Se evalúan todos los casos de uso armados. | Todas las vistas del aplicativo.   * login * Crear * Seleccionar * Modificar * Consultar * Asignar roles * Asignar estados | Validar que toda la funcionalidad de estas opciones funcionen correctamente. |
| Lógico | Pruebas unitarias | Escribir las pruebas unitarias  Con datos de entrada y salida, serán datos verdaderos y falsos, se validará el comportamiento del aplicativo ante estos escenarios. | Metodos a probar: | Los resultados son satisfactorios |
| Controlador | Pruebas de Breakpoint | Para las pruebas de análisis y depuración del código se realizó con la herramienta debug, de la aplicación visual Studio code. | Módulos:  Traspaso y comunicación  Procesamiento | Los métodos evaluados imprimen las variables correctas durante la ejecución de estas pruebas. |
| Base de datos | Pruebas de Consultas masivas | Realizar consultas masivas y verificar el punto de stress de la base de datos con la aplicación JMeter conectada con el gestor de base de datos PostgreSQL consultando |  | Verificar en que punto es critico la consulta de usuarios a la base de datos, ingresando de forma masiva. |