



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE SISTEMAS Y COMPUTACIÓN**

Sistema de información geográfica para la gestión de alertas tempranas fitosanitarias en el cantón Francisco de Orellana.

Trabajo de Titulación para optar al título de ingeniero en Sistemas y Computación

Autores:

**Aguaiza Tutin, Leonidas Gustavo
Vacacela Gaibor, Alexander Sebastián**

Tutor:

Ing. Gonzalo Allauca Peñafiel MgS.

Riobamba, Ecuador. 2022

DERECHOS DE AUTORÍA

Nosotros, Leonidas Gustavo Aguaiza Tutin y Alexander Sebastián Vacacela Gaibor, con cédula de ciudadanía 0504015710 y 0604537845, autores del trabajo de investigación titulado: Sistema de información geográfica para la gestión de alertas tempranas fitosanitarias en el cantón Francisco de Orellana, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, a la fecha de su presentación.



Leonidas Gustavo Aguaiza Tutin

C.I: 0504015710



Alexander Sebastián Vacacela Gaibor

C.I: 0604537845

DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL

Quienes suscribimos, catedráticos designados Tutor y Miembros del Tribunal de Grado para la evaluación del trabajo de investigación Sistema de información geográfica para la gestión de alertas tempranas fitosanitarias en el cantón Francisco de Orellana por Leonidas Gustavo Aguaiza Tutin y Alexander Sebastián Vacacela Gaibor, con cédula de identidad número 0504015710 y 0604537845, certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha asesorado durante el desarrollo, revisado y evaluado el trabajo de investigación escrito y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 28 de noviembre de 2022.

Ximena Quintana, PhD.
PRESIDENTE DEL TRIBUNAL DE GRADO



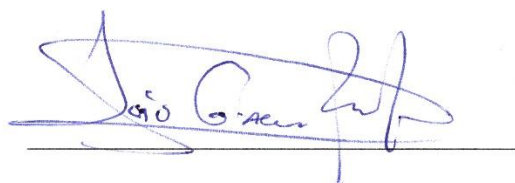
Pamela Buñay, Msc.
MIEMBRO DEL TRIBUNAL DE GRADO



Hugo Paz, Dr.
MIEMBRO DEL TRIBUNAL DE GRADO



Gonzalo Allauca, Mgs.
TUTOR



CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación Sistema de información geográfica para la gestión de alertas tempranas fitosanitarias en el cantón Francisco de Orellana por Leonidas Gustavo Aguaiza Tutin y Alexander Sebastián Vacacela Gaibor, con cédula de identidad número 0504015710 y 0604537845, bajo la tutoría de Mgs. Luis Gonzalo Allauca Peñafiel; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 28 de noviembre de 2022.

Presidente del Tribunal de Grado
PhD. Ximena Alexandra Quintana López



Miembro del Tribunal de Grado
Msc. Pamela Alexandra Buñay Guisñan



Miembro del Tribunal de Grado
Dr. Hugo Humberto Paz León





CERTIFICACIÓN

Que, AGUAIZA TUTIN LEONIDAS GUSTAVO con CC: **0504015710** y VACACELA GAIBOR ALEXANDER SEBASTIAN con CC: **0604537845**, estudiantes de la Carrera **SISTEMAS Y COMPUTACIÓN, NO VIGENTE**, Facultad de **INGENIERIA**; han trabajado bajo mi tutoría el trabajo de investigación titulado " **SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA GESTIÓN DE ALERTAS TEMPRANAS FITOSANITARIAS EN EL CANTÓN FRANCISCO DE ORELLANA**", cumple con el **6 %**, de acuerdo al reporte del sistema Anti plagio **URKUND**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 15 de noviembre de 2022



Firmado electrónicamente por:
**LUIS GONZALO
ALLAUCA
PENAFIEL**

Ing. Luis Gonzalo Allauca Peñafiel Mgs.
TUTOR TRABAJO DE INVESTIGACIÓN

DEDICATORIA

El presente trabajo de investigación, le dedico a toda mi familia. principalmente a mis padres que me dieron la vida y han estado conmigo en todo momento. Gracias por enseñarme a afrontar las dificultades sin perder nunca la cabeza ni morir en el intento. Me han enseñado a ser la persona que soy hoy, mis principios, mis valores, mi perseverancia y mi empeño. Todo esto con una enorme dosis de amor y sin pedir nada a cambio. También quiero dedicar A todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que me abrieron las puertas y compartieron sus conocimientos.

Leonidas Gustavo Aguaiza Tutin.

Esta nota de agradecimiento está dirigida a todos los que fueron un apoyo considerable e imprescindible en el transcurso de mi carrera y para el desarrollo posterior de esta tesis. En primer lugar, a mis padres, que fueron el apoyo moral y espiritual necesario para continuar cuando las dificultades se presentaron. Agradezco al equipo de profesores que fueron guías y soporte intelectual en este proceso de investigación, sin ellos esto no hubiese podido ser posible. Gracias por su esfuerzo, paciencia, y consideración.

Alexander Sebastián Vacacela Gaibor

AGRADECIMIENTO

Agradezco a nuestros padres, amigos y compañeros por el apoyo que siempre recibimos por parte de ellos. A la escuela de Sistemas y Computación de la UNACH, en donde adquirimos nuevas experiencias y nos formamos profesionalmente. A nuestros docentes de la carrera que compartieron varios de sus conocimientos y fueron un gran complemento en nuestra formación personal. A los docentes Ing. Luis Gonzalo Allauca Peñafiel MgS, PhD. Hugo Humberto Paz Leon y e MsC. Pamela Alexandra Buñay Guisñan, que fueron una gran guía en la culminación del presente proyecto de investigación. Finalmente agradezco a todas aquellas personas que siempre nos apoyaron para lograr que este sueño se haga realidad.

ÍNDICE GENERAL

DERECHOS DE AUTORIA.....	II
DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DEL TRIBUNAL.....	III
CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL	IV
CERTIFICADO ANTIPLAGIO	V
ÍNDICE DE TABLAS.....	XI
ÍNDICE DE FIGURAS	XII
RESUMEN.....	XIII
ABSTRACT	XIV
CAPÍTULO I. INTRODUCCIÓN.....	15
1. ANTECEDENTES	15
2. PLANTEAMIENTO DEL PROBLEMA.....	15
3. JUSTIFICACIÓN.....	16
4. OBJETIVOS.....	16
4.1. Objetivo General.....	16
4.2. Objetivos específicos	16
CAPÍTULO II. MARCO TEÓRICO.....	17
1. SISTEMAS DE INFORMACIÓN GEOGRÁFICOS	17
1.1. Historia.....	17
1.2. Tipos de Sistemas de información geográfica	17
1.2.1. Visualizadores de datos.....	17
1.2.2. Programas SIG de escritorio	18
1.2.3. Programas SIG profesionales.....	18
1.2.4. SIG para la web (Webmapping).....	18
1.2.5. SIG específicos.....	18
1.3. Componentes de los Sistemas de información geográfica.....	18
1.3.1. Base de datos geográfica.....	18
1.3.2. Sistema gestor de base de datos	18
1.3.3. Sistema de análisis geográfico	18
1.3.4. Sistema de presentación de información	18
1.4. Webmapping.....	19
1.4.1. Evolución de los webmapping	19
1.5. Herramientas de desarrollo	19
1.5.1. Cuadro comparativo librerías Webmapping	20
2. METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	21
2.1. Metodologías ágiles	21
2.1.1. Metodología Scrum.....	21
2.1.2. Metodología Crystal.....	22

2.1.3.	Metodología FDD	24
2.1.4.	Metodología ASD	24
2.1.5.	Metodología XP	25
3.	ESTACIONES METEOROLÓGICAS	27
3.1.	Datos entregados	27
3.2.	Alertas fitosanitarias	27
4.	ISO 25000.....	28
4.1.	Fiabilidad ISO/IEC 25000	29
4.2.	Parámetros de aceptación.....	29
CAPÍTULO III. METODOLOGÍA.....		31
1.	TIPO Y DISEÑO DE LA INVESTIGACIÓN.....	31
1.1.	Según el tipo de inferencia.....	31
2.	POBLACIÓN DE ESTUDIO Y TAMAÑO DE MUESTRA.....	31
3.	HIPÓTESIS	31
3.1.	Variables	31
3.1.1.	Variable dependiente.....	31
3.1.2.	Variable independiente.....	31
3.2.	Operacionalización de variables	32
4.	MÉTODOS DE ANÁLISIS	32
4.1.	Método estadístico	32
4.1.1.	Análisis exploratorio de datos.....	32
4.2.	Técnicas de recolección de datos.....	33
4.2.1.	Observación.....	33
4.2.2.	Entrevistas	33
5.	PROCEDIMIENTO	33
6.	PROCESAMIENTO Y ANÁLISIS	33
6.1.	Análisis de metodologías de desarrollo	33
6.2.	Desarrollo del sistema para la gestión de alertas tempranas.....	34
6.3.	Evaluación de la fiabilidad del sistema.....	41
6.4.	Resultados de la evaluación.....	41
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN.....		42
7.	RESULTADOS	42
7.1.	Resultado del análisis de metodología.....	42
7.1.1.	Esquema básico	42
7.1.2.	Metodologías adaptadas al desarrollo de sistemas de información geográfica 43	
7.2.	Resultado del desarrollo del sistema de información geográfica.....	45
7.2.1.	Fase de planificación	45
7.2.2.	Fase de diseño	45
7.2.3.	Fase de codificación	46
7.2.4.	Fase de pruebas	48

7.3.	Resultados de evaluación de fiabilidad.....	49
7.3.1.	Sub característica: Madurez	50
7.3.1.1.	Métrica: Eliminación de fallos.....	50
7.3.1.2.	Métrica: Cobertura de pruebas.....	51
7.3.1.3.	Métrica: Tiempo medio entre fallos.....	54
7.3.2.	Sub característica. Disponibilidad.....	56
7.3.2.1.	Métrica: Tiempo de servicio	56
7.3.2.2.	Métrica: tiempo medio de inactividad.	57
7.3.3.	Sub característica. Tolerancia a fallos.....	59
7.3.3.1.	Métrica: Prevención de fallos	59
7.3.3.2.	Métrica: redundancia.	61
7.3.4.	Sub característica. Capacidad de recuperación	62
7.3.4.1.	Métrica. tiempo medio de recuperación	62
7.3.5.	Resultados finales de evaluación de fiabilidad al sistema web.....	63
7.3.6.	Resultados totales.....	63
7.4.	Comprobación de hipótesis.....	64
8.	DISCUSIÓN.....	66
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES		67
CONCLUSIONES		67
RECOMENDACIONES		67
BIBLIOGRAFÍA.....		68
ANEXOS.....		70

ÍNDICE DE TABLAS

Tabla 1: Librerías <i>Webmapping</i>	20
Tabla 2: Operacionalización de variables.....	32
Tabla 3: Tabla de requerimientos funcionales y no funcionales	34
Tabla 4: Estructura de clases.	38
Tabla 5: Metodologías ágiles.....	43
Tabla 6: Resultados eliminación de fallos.	50
Tabla 7: Formula de eliminación de fallos.	51
Tabla 8: Resultados obtenidos de la eliminación de fallos.....	51
Tabla 9: Formula de cobertura de pruebas.	54
Tabla 10: Resultados de cobertura de pruebas.	54
Tabla 11: Resultados de Tiempo medio entre fallos.	56
Tabla 12: Formula de disponibilidad.....	56
Tabla 13: Resultados de disponibilidad.....	56
Tabla 14: Ponderación de valores de tiempo medio de inactividad.	58
Tabla 15: Formula de tiempo medio de inactividad.	58
Tabla 16: Resultados de tiempo medio de inactividad.....	59
Tabla 17: Prevención de fallos, datos.....	59
Tabla 18: Formula de prevención de fallos.	60
Tabla 19: Resultados de prevención de fallos.	60
Tabla 20: Redundancia, datos.....	61
Tabla 21: Formula de redundancia.	61
Tabla 22: Resultados de redundancia.	61
Tabla 23: Formula de tiempo medio de recuperación.	62
Tabla 24: Resultados de tiempo medio de recuperación.	62
Tabla 25: Resultados de evaluación sub característica de Madurez.....	63
Tabla 26: Resultados de evaluación sub característica de Disponibilidad.	63
Tabla 27: Resultados de evaluación sub característica de Tolerancia a fallos.	63
Tabla 28: Resultados de evaluación sub característica de Capacidad de recuperación.....	63
Tabla 29: Resultados globales de la evaluación del criterio de fiabilidad ISO/IEC.....	63
Tabla 30: Resultados de metodologías analizadas.	64
Tabla 31: Pruebas de normalidad	65
Tabla 32: Pruebas de Hipótesis para una proporción	66

ÍNDICE DE FIGURAS

Figura 1: Historia de los Sistemas de información geográfica.....	17
Figura 2: Miembros de equipos de metodología Cristal.	23
Figura 3: Fases de FDD.....	24
Figura 4: Fases de metodología ASD.....	25
Figura 5: Fases de la metodología XP.....	26
Figura 6: División de norma ISO/IEC 25000.....	28
Figura 7: Métricas de calidad de ISO/IEC 25000.	28
Figura 8: Parámetros de la característica Madurez.....	29
Figura 9: Parámetros de la característica Disponibilidad.....	30
Figura 10: Parámetros de la característica Tolerancia a fallos.....	30
Figura 11: Parámetros de la característica Capacidad de recuperación.....	30
Figura 12: Diagrama de caso de uso Usuario.....	36
Figura 13: Diagrama de Despliegue de componentes.	36
Figura 14: Diagrama Entidad relación.....	37
Figura 15: Diagrama de colaboración.....	37
Figura 16: Estructura de archivos de módulos.	39
Figura 17: Módulos del sistema Web de alertas fitosanitarias.....	40
Figura 18: Prueba unitaria.....	40
Figura 19: Prueba de componentes.....	41
Figura 20: Documento SRS.....	45
Figura 21: Diseño de Interfaz.	46
Figura 22: Estándar de codificación.....	47
Figura 23: Página principal de Alertas tempranas fitosanitarias.	47
Figura 24: <i>Dashboard</i> de página web de alertas fitosanitarias.....	48
Figura 25: Tabla de alertas fitosanitarias.....	48
Figura 26: Resultado de pruebas unitarias.....	49
Figura 27: Resultado de pruebas de componentes.....	49
Figura 28: Cobertura de pruebas, primer caso de prueba.	52
Figura 29: Cobertura de pruebas, resultados de primer caso de prueba.	52
Figura 30: Cobertura de pruebas, segundo caso de prueba.	53
Figura 31: Cobertura de pruebas, resultados de segundo caso de prueba.	53
Figura 32: Tiempo medio entre fallos, parámetros.....	55
Figura 33: Tiempo medio entre fallos, resultados de evaluación.....	55
Figura 34: Tiempo medio de inactividad, primer caso de prueba.	57
Figura 35: Tiempo medio de inactividad, segundo caso de prueba.....	58
Figura 36: Grafica de radar. Cumplimiento funcional de metodologías.....	65

RESUMEN

En esta investigación se implementó un Sistema de Información Geográfica para la gestión de alertas tempranas fitosanitarias en el cantón Francisco de Orellana, con la finalidad de facilitar los múltiples desafíos existentes en el cultivo principalmente del cacao. Problemas como la falta de sistemas que integren los datos climatológicos de las estaciones meteorológicas, y las conviertan en alertas fitosanitarias, que permitan notificar de manera oportuna a los productores, el desarrollo de plagas en los cultivos, de esta forma se logren tomar acciones preventivas y correctivas mejorando la productividad en el sector. Este sistema se ejecutó basándose en los requerimientos establecidos en el ERS (“Especificación de requisitos de software”). Para cumplir con los objetivos planteados, se desarrolló bajo la metodología XP, misma que fue seleccionada, tras investigar varias metodologías para determinar la que se adapte a la naturaleza del proyecto como es un equipo reducido, requerimientos cambiantes y se adapte al desarrollo de un SIG. Obteniendo como resultado el sistema web con los siguientes módulos: Modulo de Administración, Estaciones, Webmapping, Dashboard y Alertas. Dicho sistema fue desarrollado con la librería de Open Layers para el módulo SIG, el backend con PHP, frontend con HTML y Bootstrap, y la base de datos en PostgreSQL.

Mediante la norma ISO/IEC 25000, se ejecutó una evaluación de la fiabilidad del sistema web, misma que proporciona los subcriterios: madurez, disponibilidad, tolerancia a fallos y capacidad de recuperación, cada uno con sus correspondientes métricas. Para llevar a cabo esta evaluación se hizo uso de la herramienta de simulación “APACHE JMETER”, mediante él envió de peticiones http, al servidor donde se encuentra implementado. Con el propósito de evaluar el rendimiento del sistema web, bajo condiciones y tiempo establecidos. Obteniendo como resultado de la evaluación un 98.3 % de calidad bajo el criterio de fiabilidad.

Palabras claves: Alertas fitosanitarias, datos meteorológicos, evaluación de fiabilidad, SIG, Webmapping.

ABSTRACT

The current research applied a Geographic Information System to manage early phytosanitary warnings in Francisco de Orellana town, to facilitate the multiple challenges in the cultivation, mainly cocoa. Problems such as the lack of systems that integrate the climatological data of the meteorological stations and convert them into phytosanitary alerts, which allow producers to be notified promptly, the development of pests in crops since preventive actions can be taken and corrective measures improving productivity by the cacao farmers. The researchers executed a system based on the requirements established in the SRS ("Software Requirements Specification"). Due to the necessity to accomplish the research's objectives, it was developed under the XP methodology. The researchers chose it after investigating various methodologies to determine the one that suits the nature of the project, such as a reduced team and changing requirements, and it adapts to the development of a GIS. As a result, the web system has the following modules: Administration Module, Stations, Web mapping, Dashboard, and Alerts. Open Layers bookshop worked with the system mentioned above for the GIS module, the backend with PHP, the BackEnd with HTML and Bootstrap, and the database in PostgreSQL. Through the ISO/IEC 25000 standard, an evaluation of the reliability of the web system was carried out, which provides the sub criteria: maturity, availability, fault tolerance, and recovery capacity, each with its corresponding metrics. It was necessary to use the "APACHE JMETER" simulation tool by sending HTTP requests to the server where it is implemented. To evaluate the performance of the web system under established conditions and time. Obtaining because of the evaluation a 98.3% of quality under the criterion of reliability.

Keywords: Phytosanitary alerts; Meteorological data; Reliability evaluation; GIS; Web mapping.



Firmado electrónicamente por:
**JESSICA MARIA
GUARANGA LEMA**

Reviewed by:

Mgs. Jessica María Guaranga Lema

ENGLISH PROFESSOR

C.C. 0606012607

CAPÍTULO I. INTRODUCCIÓN

1. Antecedentes

En los últimos tiempos la importancia de implementar las TICS en los procesos tradicionales es sumamente importante, permitiendo gestionar los mismos de la manera más eficaz y eficiente. En los últimos años se han llevado a cabo una serie de automatizaciones dentro de las diferentes áreas como: educación, industria, deporte, agricultura, etc.

En el cantón Francisco de Orellana de la provincia de Orellana del Ecuador, la actividad agrícola ocupa uno de los principales lugares en las actividades comerciales, destinando alrededor de cincuenta mil hectáreas de suelo para cultivos permanentes y transitorios, principalmente de cacao destinado a la exportación aportando unas once mil toneladas métricas de producción; esta realidad conlleva a la necesidad de una eficiente gestión de alertas fitosanitarias para el control de daños en los cultivos, dicha gestión se ha venido desarrollando de manera manual, descentralizada e ineficiente. Resultado de lo cual en el año 2020 se contabilizaron cerca de quince mil hectáreas de cultivos perdidos por factores climáticos, asociados las tres estaciones meteorológicas que se encuentran en todo el cantón evidenciándose la necesidad de mostrar la información que estos generan de manera centralizada, estructurada y relacionarla de tal forma que permita ayudar a los cerca de veinte y siete mil beneficiados del cultivo en el cantón, de igual manera permita a los encargados del monitoreo mejorar su labor diaria en el control fitosanitario. (INEC, 2020)

Para llevar a cabo el desarrollo del proyecto se plantea la investigación de las diferentes metodologías existentes, esto, debido a que en la actualidad es una base fundamental de para la ejecución de un proyecto. Este proceso de investigación permitirá obtener una retroalimentación, y lograr un conocimiento concreto mismo que facilitará la comprensión de manera precisa lo que se requiere en las fases de: análisis, diseño, desarrollo e implementación del sistema de información geográfica. (Santander Universidades, 2021). Así también tener énfasis en la parte relevante de la investigación. Evaluar la fiabilidad del sistema web, de acuerdo con la norma ISO/IEC 25000, con la finalidad de determinar si el Webmapping implementado, incide en la fiabilidad de las alertas tempranas fitosanitarias agrícolas en el cantón Francisco de Orellana.

2. Planteamiento del problema

Actualmente en el cantón Francisco de Orellana, existen varios desafíos en la agricultura entre los cuales se destaca la limitada aplicación de tecnología, el cambio climático y la falta de respuesta rápida ante las enfermedades de los cultivos. En el proceso de alertas fitosanitarias en las plantaciones del sector se debe contar con información precisa, eficaz y oportuna que permita un adecuado análisis situacional y la correspondiente toma de decisiones, para una buena gestión, evaluación y prevención de las plagas en las plantas. El comportamiento de este tipo de enfermedades en los cultivos es dinámico, uno de los factores que actúa directamente con esta situación es el cambio climático, proliferando este tipo de plagas, por lo mismo es importante el análisis de los datos recolectados de las estaciones

meteorológicas que se encuentran en el sector, de tal manera se logre gestionar los recursos y actividades en la toma de decisiones oportunas para evitar problemas en los cultivos. Con este precedente se plantea el desarrollo de un sistema de información geográfica, delimitado en la creación de un Web Mapping, que permitirá analizar, visualizar y comunicar las alertas fitosanitarias según la localización de los centros de control meteorológicos establecidos dentro del cantón Francisco de Orellana y de esta manera ayudar a tomar normas preventivas de control para los cultivos del sector.

Este sistema de información geográfica será presentado mediante un aplicativo web de fácil acceso, donde se encontrarán niveles de usuarios especializados a los cuales se les comunicará de manera oportuna los cambios en las variables que son tomadas en cuenta para la gestión de las alertas mediante un aviso que le llegara al respectivo correo electrónico.

3. Justificación

En los últimos tiempos la importancia de implementar las TICS en los procesos tradicionales es sumamente importante, permitiendo gestionar los procesos tradicionales de la manera más eficaz y eficiente. En los últimos años se han llevado a cabo una serie de automatizaciones dentro de las diferentes áreas como: educación, industria, deporte, agricultura, etc. Ante lo mencionado se ha visto importante la implementación de un sistema de información geográfica para la gestión de alertas tempranas fitosanitarias en el cantón francisco de orellana. Esto debido a que la producción del cacao en el sector se ha visto afectado por la falta de gestión y eficacia en la toma de acciones en los cultivos, trayendo consigo perdidas innumerables en el cultivo de este producto. Por lo que la implementación de este sistema permitirá a los productores visualizar, analizar y recibir notificaciones de cambios en las alertas fitosanitarias de acuerdo con los datos emitidos por las estaciones meteorológicas ubicadas en el sector. Posibilitando la toma de decisiones de manera puntual y eficiente para efectivizar acciones preventivas y correctivas en los cultivos.

4. Objetivos

4.1. Objetivo General

Implementar un Sistema de Información Geográfica para la gestión de alertas tempranas fitosanitarias en el cantón Francisco de Orellana.

4.2. Objetivos específicos

- Analizar metodologías de desarrollo para la implementación de sistemas de información geográfica en el ámbito de la gestión de alertas tempranas.
- Desarrollar el sistema de información geográfica aplicando la mejor metodología analizada.
- Evaluar la fiabilidad del sistema de información geográfica para la gestión de alertas tempranas fitosanitarias en el cantón Francisco de Orellana.

CAPÍTULO II. MARCO TEÓRICO

1. Sistemas de información geográficos

El término, es netamente vinculado con “planificación territorial”, análisis de información “geoespacial”, y se aplica a la solución de problemas de cartografía, ambientales socioeconómicos, agrícolas, etc. El SIG permite recolectar, gestionar y analizar datos espaciales, para presentar mediante de mapas y escenas en 3D. El objetivo del análisis de los datos es revelar un conocimiento primordial oculto en los datos, con el propósito de ayudar a los usuarios a tomar decisiones inteligentes y oportunas. (Lara, Simeón, & Navarro, 2006) Una definición formal propuesta se refiere a “sistema empleado para describir y categorizar la Tierra y otras geografías con el objetivo de mostrar y analizar la información a la que se hace referencia espacialmente. Este trabajo se realiza fundamentalmente con los mapas”. (Esri, 2022)

1.1. Historia

Existen varios autores quienes relatan el origen de los SIG y la aparición de las herramientas de la cartografía. (Ruiz, 2006) Asegura que el comienzo de los SIG se dio tras el cambio del formato de cartografía convencional, a la digital; por lo cual se puede atribuir que marca una estrecha relación con la evolución de los sistemas informáticos tanto en el hardware y software.

	TECNOLOGÍAS	USUARIOS	APLICACIONES
1950	-> Primeros ordenadores electrónicos	-> Ejército	-> Militares -> Atlas of the British Flora
1960	-> Mesas de digitalización -> Miniordenadores 16 bits -> Plotters -> Uso estructuras raster en SIG	-> LCG -> CGIS -> Universidades EEUU	-> Investigación y educación -> Planeamiento urbano -> Gestión y análisis de recursos naturales
1970	-> Miniordenadores 32 bits -> Uso estructuras vectoriales en SIG	-> ING	-> Investigación y educación -> Censos -> Atlas Nacional de España
1980	-> Ordenadores personales PC -> SIG para PC -> Extensión uso escáners	-> Universidades españolas -> Institut Cartogràfic de Catalunya -> Centro de Gestión Catastral y -> Cooperación Tributaria	-> Investigación y educación -> Sistema de Intormación territorial de Cataluña -> Sistema de Información Geográfica Nacional -> Systeme de Información Territorial -200 -> SIG Catastral
1990	-> Integración SIG/GPS/Teledetección -> Sistemas multimedia -> Ordenadores proceso paralelo -> Integración raster/vectorial	-> Ayuntamientos -> Administraciones Autonómicas	-> Investigación y educación -> Aplicaciones globales -> Digital Chart of the world

Figura 1: Historia de los Sistemas de información geográfica.

Fuente: Tomado de (Ruiz, 2006)

1.2. Tipos de Sistemas de información geográfica

1.2.1. Visualizadores de datos

Este tipo de SIG permite únicamente la visualización de los datos, rara vez permiten realizar operaciones básicas como, calcular distancias o superficies. Dentro de este grupo de software existen: *ArcExplorer*, *GeoMedia*, *SIGNA*, *SIGPAG*.

1.2.2. Programas SIG de escritorio

Este grupo de SIG permite el análisis avanzado de datos, por otro lado, tiene una capacidad limitada, en lo referente a modificación y reestructuración de la información, esto sucede cuando los datos provienen de diferentes fuentes. En este grupo de sistemas se encuentran: *ArcView*, *MapInfo*, *Idrisi*, Carta digital.

1.2.3. Programas SIG profesionales

Cuenta con una gestión eficiente de la información sin importar de la fuente que estos provengan, permite: acceder, modificar, estructurar y explotar la información de acuerdo con los fines necesarios. También cuenta con la facultad de formar mapas de alta calidad adaptados a las necesidades de los especialistas. En este grupo de software están: *ArcInfo*, *GeoMedia* profesional, *gvSIG*, versiones superiores a 4.0 de carta digital.

1.2.4. SIG para la web (Webmapping)

Estos son sistemas que están basados en *SIG Viewer*, orientados en plataformas web. Estos son mapas dinámicos que permiten la carga, análisis, visualización de la información referente a una temática específica. Dentro de este tipo se encuentran: *ArcWeb*, *Webmap*, *Geomedia*, etc.

1.2.5. SIG específicos

Estos sistemas están desarrollados para cubrir una necesidad en específico, estos están generalmente en dispositivos y con plataformas limitadas al procesamiento de la información. Estos son orientados a sistemas de navegación de aeronaves y barcos. (Ceupe Magazine, 2018)

1.3. Componentes de los Sistemas de información geográfica

1.3.1. Base de datos geográfica

Es un almacén donde están contenidos todos los datos necesarios para formar un SIG. Generalmente está organizado con un conjunto de tablas que mantienen una relación entre sí, es importante que una de estas tablas contenga la información trigonométrica del mapa, para que al momento de presentar los datos puedan graficarse.

1.3.2. Sistema gestor de base de datos

Este es un software el cual nos permite almacenar, modificar y extraer la información de la base de datos.

1.3.3. Sistema de análisis geográfico

El componente nos permite crear relación entre los datos espaciales y a partir de ellos obtener nueva información, los datos relacionados nos permiten graficar, generar estadística, etc.

1.3.4. Sistema de presentación de información

Permite graficar todos los datos del SIG, a través de una pantalla final al usuario, también permite enviar a cualquier otro dispositivo de salida como: plotter, impresora, scanner, etc.

Los datos resultantes pueden exportarse en forma de mapas, imágenes, tablas, gráficos. (Ceupe Magazine, 2018)

1.4. Webmapping

Se trata de un sistema interactivo de información geográfica, en forma de página web. Con el avance tecnológico ahora los datos pueden ser compartidos, visualizados y editados desde el navegador web. El punto importante es la accesibilidad, puede ser alcanzado por cualquier persona en cualquier dispositivo que tenga un navegador web y conexión a internet. (Dorman, 2021)

1.4.1. Evolución de los webmapping

En las últimas décadas gracias al desarrollo tecnológico se logra expandir los sistemas de información geográfico al mundo digital mediante internet; la década de los noventa fue la protagonista del comienzo del despliegue de los visores cartográficos los cuales comenzaron poco después de la aparición de la *World Wide Web* con la introducción del visualizador de Mapas *Xerox PARC* en 1993.

Desde el año 2000, la OGC ha desarrollado un conjunto de estándares para la cartografía Web, siendo el primero *Web Map Service*. Durante el año 2005 fue un punto de inflexión para la creación de nuevas aplicaciones y tipos de software para el desarrollo e implementación del SIG en la web desde el lanzamiento de las *Application Programming Interfaces* de Mapas. (Brovelli, 2012)

En la actualidad existen varias herramientas orientadas al desarrollo de aplicaciones web que permiten manipular los datos geográficos mediante interfaces cada vez más complejas y completas.

1.5. Herramientas de desarrollo

El campo de Web Mapping, como las demás tecnologías, sufren evoluciones constantes lo cual desemboca en un sinnúmero de herramientas adaptadas a las necesidades. Entre ellas tenemos a los componentes básicos del desarrollo web, *HTML* y *CSS*, lenguaje de programación *JavaScript* que funciona como base para librerías enfocadas al desarrollo como lo son *Leaflet*, *OpenLayers*, *Turf* y demás; los datos geográficos son manipulados en extensiones de bases de datos como lo son *PostGis*, *MySQL Spatial Data Types* y otras más. Estas son las tecnologías básicas consideradas en un sistema de este tipo, junto con los componentes de software necesarios para desarrollar una página web como lo son los editores de texto, un navegador web moderno, herramientas web de desarrollo y un servidor web.

1.5.1. Cuadro comparativo librerías *Webmapping*

Tabla 1: Librerías *Webmapping*.

Librería	Descripción	Licencia	Aspecto positivo
<i>OpenLayers</i>	Generar mapas web tanto para versión de escritorio como para versión móvil debido a su integración con <i>CSS3</i> .	<i>Open source</i>	La librería es accesible a través de su <i>API OpenLayers</i> , excelentemente documentada.
<i>Leaflet</i>	Librería <i>Open Source</i> centrada totalmente en el web mapping interactivo y en facilitar las soluciones de cartografía web tanto para navegadores de escritorio como para dispositivos móviles.	<i>Open source</i>	Su gran implementación y uso preferido por gran cantidad de desarrolladores se debe a su desempeño y rapidez, flexibilidad y adaptabilidad.
<i>Google Maps API</i>	<i>Google</i> ofrece la posibilidad de acceder a sus servicios de mapas web mediante su <i>Google Maps Javascript API</i> . Mediante esta <i>API</i> se pueden crear mapas interactivos basados en los mapas base de <i>Google</i> y añadir contenido como capas e imágenes.	Privada	Funcionalidades complementarias con arquitectura robusta y estable.
<i>Mapbox GL JS</i>	No es tan sólo una librería de <i>JavaScript</i> enfocada al <i>web mapping</i> . Se define como una plataforma de localización en tiempo real dirigida al desarrollo de aplicaciones basadas en mapas y la visualización de información espacial.	Gratuita y pagada	Incorpora tecnologías de <i>Leaflet</i> , <i>Google Maps</i> , <i>OpenStreetMaps</i> y otros proveedores.
<i>ESRI</i>	Cuenta con multitud de posibilidades para el desarrollo de soluciones de visualización de mapas en aplicaciones web y	Privada	<i>ArcGIS API for JavaScript</i> está bien documentada y extensa, incluyendo guías y tutoriales para ejercitarla y

	móviles, gracias a los <i>SDKs</i> que <i>ESRI</i> pone a disposición de los desarrolladores.		aprender los conceptos básicos.
Carto	Plataforma orientada a la visualización de datos espaciales mediante cartografía en línea, a la analítica de datos y al desarrollo de aplicaciones con componente geográfico.	Privado	Compleja y poderosa plataforma que combina servicios de análisis, alojamiento y visualización de datos capaz de ofrecer resultados muy potentes.

Fuente: Tomado del artículo (Farkas, 2007)

2. Metodologías de desarrollo de software

Es un proceso ordenado, para el control de producción de software o sistemas informáticos, permite mantener una estructura ordenada durante todo el proceso de desarrollo. Estos procesos metodológicos han ido mejorando con el pasar de los tiempos, con el propósito de facilitar el desarrollo. (Maida & Paziencia, 2015)

En la última década, con el acelerado incremento de las necesidades de automatizar los procesos: empresariales, industriales, manufactura, mediante la implementación de software. Por lo que se ha vuelto más imprescindible el desarrollo ágil. Ante esta situación las metodologías de desarrollo de software han ido revolucionando para adaptarse a las nuevas formas de desarrollar sistemas informáticos con nuevas características como: rapidez, sencillez, adaptabilidad a cambios, menor documentación, etc. Esto permite que los generadores de software mantengan una ventaja en la producción y desarrollo, cumpliendo con las necesidades de los usuarios. (Rivas, Corona, & Gutiérrez, 2015)

2.1. Metodologías ágiles

El termino ágil en el desarrollo de software ofrece una modificación en los valores y principios naturales de los programadores, los cuales permiten a los equipos de software un desarrollo rápido y capaz de responder a los cambios espontáneos originados a lo largo del proyecto. Regido por valores que permiten evaluar las iteraciones sobre el proceso y las herramientas, desarrollar software funcional en lugar de conseguir una documentación detallada, el compromiso con la satisfacción del cliente en contra de la negociación del contrato y que permita responder a los cambios sin dar prioridad a seguimiento de una planificación estática.

2.1.1. Metodología Scrum

Esta metodología, nos permite trabajar los proyectos medianos y grandes, simples y complejos. Las ventajas de usar esta metodología es que es flexible y dinámico ante los cambios que se presenten en los requerimientos del proyecto. También permite a los

desarrolladores que se puedan hacer entregas parciales a los clientes, otra de las ventajas importantes de esta metodología es que existe una relación colaborativa para el trabajo en equipo. (Salazar, 2016)

Esta metodología cuenta con 5 fases y 19 procesos las cuales se las detalla a continuación:
Primera fase iniciación. - Dentro de esta sección, se genera y se establece la visión del proyecto con la finalidad de que la misma sirva de enfoque y dirección de este. También se establecen los principales actores del proyecto como son: “*Scrum master, Product Owner, interesados y equipo de proyecto*”. Así también se determina la lista de prioridades del proyecto la cual servirá para el plan de lanzamiento y el tamaño de cada sprint. (Salazar, 2016)

Segunda fase planificación y estimación. - Dentro de esta fase se enmarcan los *sprints*, las historias de usuario, se realizan las estimaciones del esfuerzo y del tiempo que tomara en cumplir cada una de las mismas. Esto se refiere a realizar una lista de tareas con su correspondiente tiempo de entrega, también se establece los sprint backlog, la misma que contiene las tareas las cuales deben completarse en cada sprint. (Salazar, 2016)

Tercera fase implementación. - Aquí se desarrolla las tareas, correspondientes a cada backlog definido en la fase anterior, con la finalidad de concluir con cada uno de los *sprints*. Para llevar a cabo estas tareas con éxito se toma en cuenta el seguimiento ordenado y estructurado de cada una de las tareas y actividades que contiene el sprint. En esta fase también se mantienen las llamadas (*Daily standup meeting*) las cuales son reuniones de tiempos cortos, donde el equipo de trabajo presenta las actividades desarrolladas y los inconvenientes o dudas que existieron o existen en las mismas. Aquí también se actualizan las listas de prioridades, definidas en las fases anteriores. (Salazar, 2016)

Cuarta fase revisión y retrospectiva. - Para proyectos grandes y complejos los cuales tienen varios equipos de scrums, aquí se convoca a reuniones las mismas que sirvan como espacio para recolectar los avances del proyecto, así también discutir los impedimentos o dudas existentes en el mismo. Después de la recepción de avances se presenta al cliente el correspondiente sprint que demuestra el avance hasta la fecha. De la misma forma los equipos correspondientes se reúnen en tiempos cortos para discutir las lecciones aprendidas durante el sprint. (Salazar, 2016)

Quinta fase lanzamiento. - Esta fase es la más importante, y la más esperada. Aquí los equipos desarrolladores entregan el proyecto a los clientes, mediante un acuerdo formal. De la misma manera se ejecutan las tareas de retrospectiva las cuales permiten identificar mejoras del proyecto. (Salazar, 2016)

2.1.2. Metodología Crystal

Esta metodología, está centrada en un marco de trabajo ágil, enfocada en las interacciones de las personas. Los métodos de desarrollo de esta metodología están codificados en colores

de acuerdo con el riesgo significativo de las personas. Se lo creo basándose en 2 fundamentos principales. (Garzas, 2012)

- “Encuentre su propia forma y métodos para optimizar el flujo de trabajo”
- “Hacer uso de métodos únicos para hacer que el proyecto sea único y dinámico”

Las ventajas principales de esta metodología son:

- Entregas frecuentes
- Mejoras flexivas
- Seguridad personal
- Fácil acceso a usuarios expertos
- Herramientas técnicas

La codificación de colores esta generada de la manera que se observa en la figura 2.



Figura 2: Miembros de equipos de metodología Cristal.

Fuente: Tomado de: (Garzas, 2012)

Primera fase puesta en escena. - Dentro de esta fase se pone en marca la planificación del incremento, esta se la realiza cada cierto tiempo dependiendo de la naturaleza y el tamaño del proyecto. Se basa en seleccionar un requerimiento del proyecto y planificar el desarrollo de este. (Martinez, 2014)

Segunda fase revisiones. - Dentro de cada incremento existen tareas e iteraciones de “construcción, demostración y resumen”. A lo cual esta fase pone énfasis a revisar estas iteraciones acordes a los objetivos y requerimientos. (Martinez, 2014)

Tercera fase monitoreo. – En esta fase se toma en cuenta las revisiones técnicas de cada uno de los incrementos, aquí se desarrollan los hitos, y la estabilidad de las fases. (Martinez, 2014)

Cuarta fase paralelismo y flujo. - Aquí se determina si el incremento o interacción cumple lo suficiente del requerimiento, puede pasar a la siguiente etapa, en caso contrario el equipo debe trabajar de manera más eficiente. (Martinez, 2014)

Quinta fase técnica de puesta a punto de la metodología. - En esta fase se ejecutan entrevistas y talleres con la finalidad de modificar o fijar los procesos de desarrollo del proyecto. (Martinez, 2014)

Sexta fase puntos de vista del usuario. - Aquí se reúnen con los usuarios finales para recolectar las diferentes opiniones de estos con la finalidad de realizar correcciones al proyecto. (Martinez, 2014)

2.1.3. Metodología FDD

De acuerdo con (Tecnologías Informacion, 2020), Esta metodología está basada principalmente en funciones. Se orienta generalmente en el diseño y desarrollo refinado. De acuerdo con la metodología esta divide al proyecto en pequeñas partes con la finalidad de realizar un trabajo más flexible. Realizando así lanzamientos con mayor frecuencia. (Rivera, 2020) menciona que FDD cuenta con las siguientes fases de desarrollo, las cuales se exponen en la figura 3.



Figura 3: Fases de FDD.

Fuente: Tomado de (Rivera, 2020)

2.1.4. Metodología ASD

(Zamora, 2014), menciona en su investigación, ASD, es una metodología, la cual se basa en el principio de adaptación de manera continua. Esto se refiere a adaptarse a los cambios del tiempo y no tener resistencia a los mínimos. En esta metodología no existe un ciclo de vida estático, al contrario, ofrece un ciclo de vida iterativo.

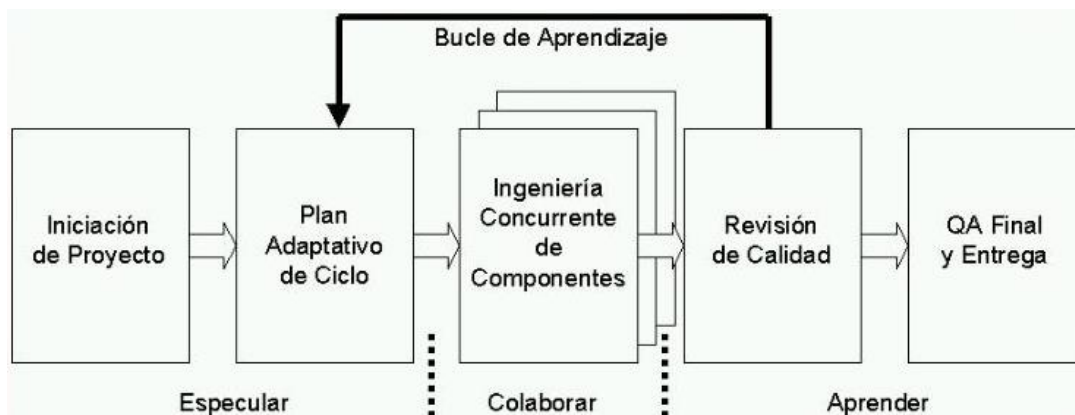


Figura 4: Fases de metodología ASD.

Fuente: Tomado de (Zamora, 2014)

De acuerdo con la figura 4 se puede afirmar que la metodología cuenta con 3 principales fases: Especular, Colaborar y Aprender. Dentro de los cuales tienen sus diferentes procesos.

Fase especulación. – Dentro de esta fase existe el proceso de iniciación del proyecto en el cual se detalla la planificación del proyecto. Así como, requerimientos, objetivos. También se desarrolla el plan adaptativo de cada uno de los ciclos.

Fase colaboración. – En esta fase se toma en cuenta y se trabaja en la ingeniería de los componentes y el desarrollo de cada uno de los ciclos planificados.

Fase aprender. – En la fase 3 se revisa la calidad del ciclo terminado el cual debe tener la calidad de acuerdo con los requerimientos y objetivos planteados en la fase inicial, de la misma manera una vez ya ejecutado la revisión se procede a la entrega del ciclo terminado. (Zamora, 2014)

2.1.5. Metodología XP

(Bryan, Harry, & Jefferson, 2018) creada por Kent Beck en 1996 diseñada para ser utilizada por equipos de desarrollo pequeños y medianos que se enfrenten a proyectos con requerimientos imprecisos o cambiantes; se convirtió en una de las metodologías preferidas, centrada en potenciar relaciones interpersonales como la clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, el aprendizaje de los desarrolladores y proporcionar un ambiente de trabajo ideal. De igual manera está centrado en la retroalimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y la capacidad de respuesta ante los cambios.

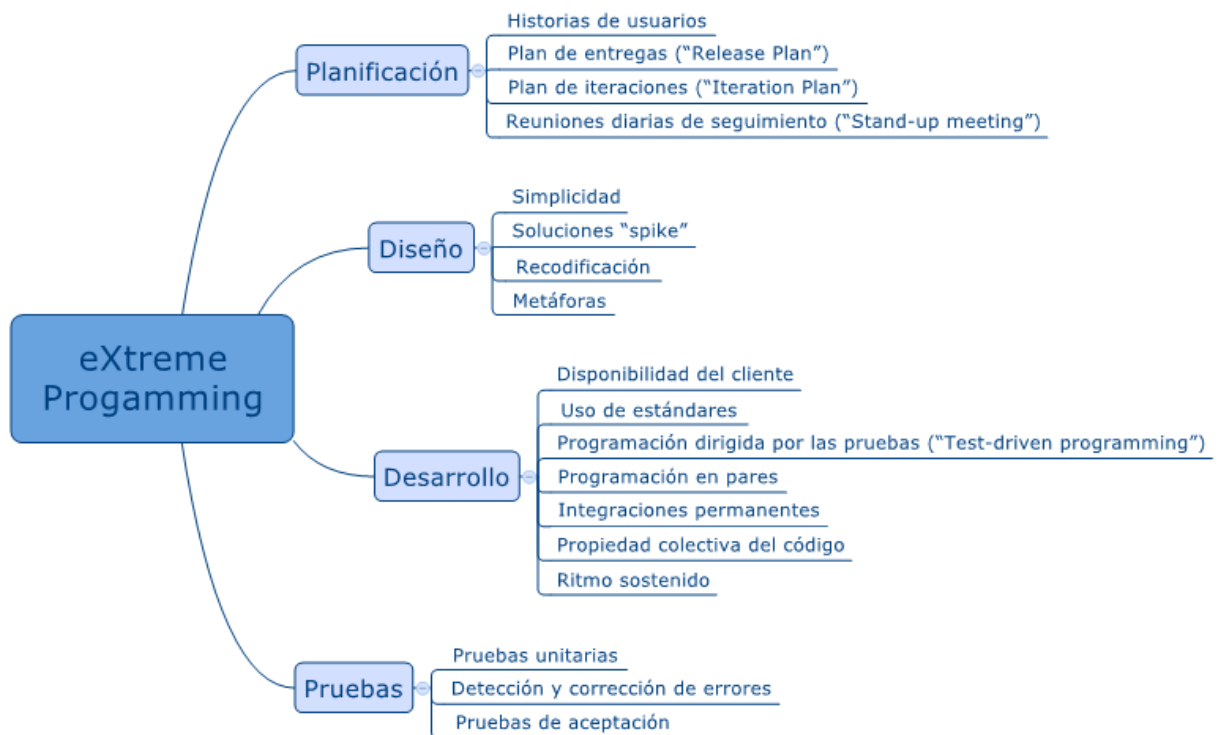


Figura 5: Fases de la metodología XP.

Fuente: Tomado del sitio web Proagilist. (Juan Luis, 2016)

El ciclo de vida de un proyecto XP tiene la característica de ser dinámico, es capaz de separar el proceso en varias fases las cuales, agrupadas de tal manera, permiten realizar las actividades necesarias en planificación, diseño, desarrollo y pruebas.

Fase de planificación. – el inicio del proyecto permite a los interesados establecer un diálogo continuo, logrando recopilar historias de usuario, mismas que serán evaluadas por los desarrolladores obteniendo el tiempo de desarrollo, riesgo, complejidad y costo para lograr fundamentar un cronograma de entregas acorde a la capacidad de trabajo de los encargados, así como las necesidades del cliente.

Fase de diseño. – con el fin de enfatizar los diseños simples, claros y precisos, la fase mantiene diferentes conceptos como el de simplicidad el cual propone implementar diseños simples que no opaquen la funcionalidad sugiriendo no adelantar procesos que no correspondan a la iteración actual, el uso de soluciones “*spike*” permite mitigar problemas técnicos por medio de pequeños programas de pruebas los cuales son desechados después de cumplir su objetivo, la recodificación sustenta el cambio del código para volverlo eficiente cada vez que se vea necesario ampliar o modificar la funcionalidad, por último el uso de metáforas ayuda a comprender el propósito del proyecto de manera sencilla siendo una guía de la arquitectura y estructura del código.

Fase de desarrollo. – los aspectos tomados en cuenta durante la fase tienen como objetivo relacionar al cliente y a los desarrolladores para un correcto desempeño de las actividades planificadas logrando reducir la documentación detallada de las especificaciones del producto final, de igual manera fomenta la creación de estándares de codificación con el fin de incluir a todo el equipo en el desarrollo continuo donde cada individuo puede aportar con ideas para la solución de errores; la forma en la cual se controlan los errores es denominada programación en pares, consiste en asignar dos personas a un trabajo conjunto para minimizar errores y maximizar la calidad, compensando el uso de los recursos humanos así como el tiempo.

Fase de pruebas. – durante esta fase se comprueba la calidad del software por medio de la detección y corrección de errores, esto se lleva a cabo a través de pruebas unitarias las cuales son aplicadas a todos los módulos antes de ser publicados, dando paso a la respectiva aprobación por parte del cliente al considerar los resultados de la ejecución de la fase actual y posterior finalización del proyecto en caso de ser aceptado. (Joskowicz, 2008)

3. Estaciones meteorológicas

Infraestructura que permite realizar labores de medición y registro de parámetros meteorológicos, con el fin de conocer el comportamiento de la atmósfera, así como del medio ambiente. La recopilación de los datos en dichas estaciones permite predecir eventos climatológicos en un área específica, aportando de igual manera en las investigaciones científicas. (Novoa & Byron, 2018)

Entre sus principales componentes se encuentran la estructura de protección contra ambientes hostiles, registro de estadística de variables climáticas, unidad de transmisión / recepción de datos a larga distancia, sistema de energía, unidad de almacenamiento continuo de información y los diferentes sensores específicos para cada variable climática. (Picon & Vazquez, 2022)

3.1. Datos entregados

Los diferentes tipos de estaciones meteorológicas mantienen una recolección de parámetros relacionados a la dirección y velocidad del viento, la temperatura del ambiente, el nivel de humedad, nivel de lluvia, la radiación ultravioleta, nivel acústico, características de las nubes y un sin número de variables relacionadas con las condiciones climáticas. (Novoa & Byron, 2018)

3.2. Alertas fitosanitarias

Son eventualidades que permiten proporcionar información sobre los riesgos en los cultivos, estas condiciones pueden ser interpretadas por enfermedades que son producidas en las plantas por distintos factores como: condiciones climáticas, sequía, insectos, etc. la información que proporciona es de vital importancia para: prevenir, destruir, y combatir las enfermedades en las plantas de manera parcial o completa, ayudando a los granjeros a evitar pérdidas de manera oportuna. (Pilaloo David, 2021)

4. ISO 25000

De acuerdo con (ISO/IEC, 2022), son normas las cuales mantienen el objetivo de evaluar la calidad del software. Esta norma mantiene una composición de patrones referenciales y definición de términos. De la misma manera se encuentra conformada por 6 secciones las cuales permiten evaluar los diferentes estándares: “ISO/IEC 2500n, ISO/IEC 2501n, ISO/IEC 2502n, ISO/IEC 2503n, ISO/IEC 2504n, y ISO/IEC 25050-25099”.



Figura 6: División de norma ISO/IEC 25000.

Fuente: Tomado de (ISO/IEC, 2022)

Dentro de esta norma tenemos diferentes características de evaluación dependiendo de los requerimientos establecidos por el cliente, en la figura 7 se puede observar las métricas que se pueden evaluar con esta norma.

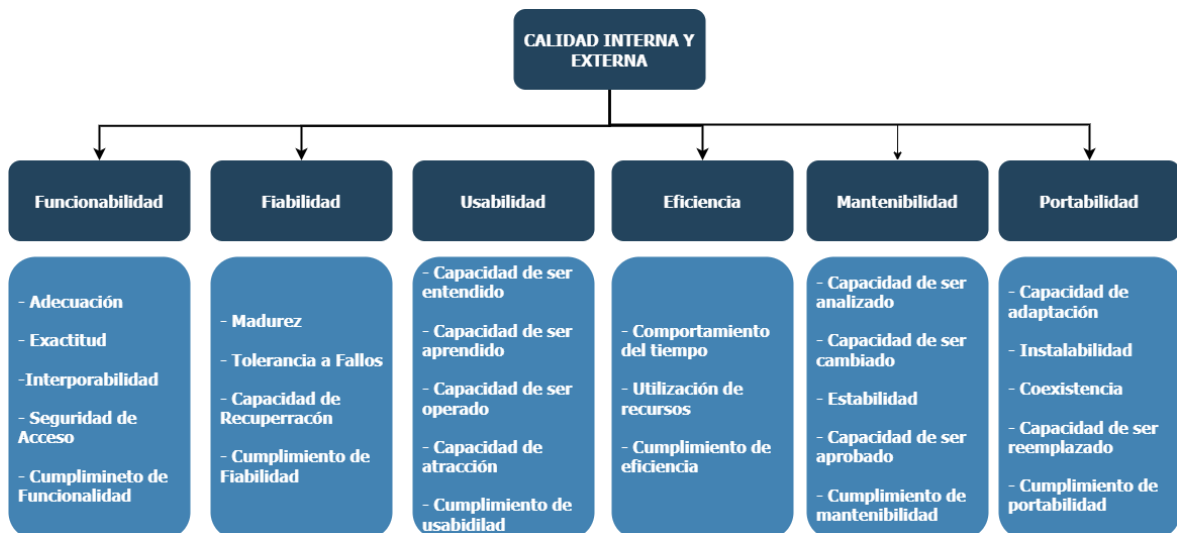


Figura 7: Métricas de calidad de ISO/IEC 25000.

Fuente: Tomado de (ISO/IEC, 2022)

4.1. Fiabilidad ISO/IEC 25000

La fiabilidad de un software se refiere a la capacidad en el que este pueda desempeñar las funciones asignadas, bajo ciertas condiciones y tiempos establecidos. Esta característica cuenta con varias Sub características para lograr determinar dicha capacidad. (ISO/IEC, 2022)

- **Madurez:** Esta sub característica mide la capacidad que tiene el sistema para ejecutar funciones en condiciones normales.
- **Disponibilidad:** Esta mide el tiempo en el que el sistema esté disponible de manera operativa para el acceso de los usuarios.
- **Tolerancia a fallos:** Esta se encarga de evaluar la capacidad del sistema, para desempeñar las funciones bajo la presencia de fallos de hardware o software.
- **Capacidad de recuperación:** Dicha sub característica evalúa el tiempo en el que el sistema se recupera ante una eventual falla, interrupción o error.

4.2. Parámetros de aceptación

Las especificaciones de las características, sub características, métricas, parámetros de aceptación, formulas y el tipo de medida, correspondiente a la característica de fiabilidad son determinadas de manera distribuida y específica para cada una.

Característica	Métrica	Fase de ciclo de vida	Propósito métrico	Método de aplicación	Formulas/ variables	Parámetro aceptación	de	Tipo de medida
Madurez	Eliminación de errores	Interna /externa	Cuantos errores detectados han sido corregidos	Contar el número de fallas corregidas en las fases de desarrollo hasta pruebas y el número de fallas detectadas en las pruebas	$X=A/B$ A= Número de fallas corregidas en la fase de diseño /pruebas B= Numero de fallas detectadas en las pruebas Donde: $B>0$	$0 \leq X \leq 1$ Cuanto más cerca del 1 este es mejor		X, A, B= contable
	Cobertura de pruebas	Interna /externa	Cuantos casos de prueba requeridos han sido ejecutados durante la etapa de pruebas	Contar el número de casos de prueba realizadas en un escenario de operación	$X=A/B$ A= número de casos de prueba realizados en escenarios de operación durante la prueba B= Número de casos de prueba realizados para cumplir los requerimientos Donde: $B>0$	$0 \leq X \leq 1$ Cuanto más cerca del 1 este es mejor		X, A, B= contable
	Tiempo medio entre fallos	Externa	Cuál es la frecuencia que el sistema en la operación	Tomar el tiempo de operación y contar el número total de fallas detectadas actualmente	$X=A/T$ A= número total de fallas detectadas actualmente T= tiempo de operación Donde: $T>0$	$X=A/T$ El valor más cercano a 0/t es el mejor		T= tiempo

Figura 8: Parámetros de la característica Madurez

Fuente: Tomado de (ISO/IEC, 2022)

Característica	Métrica	Fase de ciclo de vida	Propósito métrico	Método de aplicación	Formulas/ variables	Parámetro aceptación	de	Tipo de medida
Disponibilidad	Tiempo de servicio	Externa	Cuál es el tiempo de servicio que el sistema proporciona	Tomar el tiempo de servicio del sistema que se proporciona actualmente y tomar el tiempo de servicio del sistema regulado en el cronograma operacional.	$X = A/B$ A=Tiempo del servicio del sistema que se proporciona actualmente B= Tiempo de servicio del sistema regulado en el cronograma operacional Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más cerca del 1 este es mejor		X, A, B= Tiempo
	Tiempo de inactividad	Externa	Cuál es el tiempo promedio en el que el sistema esta inactivo después de que ocurra un fallo	Tomar el tiempo total de inactividad y contar el número de fallos observados.	$X = T/A$ A-Número de fallos observados. T-Tiempo total de inactividad.	$X = A/T$ El valor más cercano a 0/t es el mejor		X= contable/tiempo A= contable T= tiempo

Figura 9: Parámetros de la característica Disponibilidad

Fuente: Tomado de (ISO/IEC, 2022)

Característica	Métrica	Fase de ciclo de vida	Propósito métrico	Método de aplicación	Formulas/ variables	Parámetro aceptación	de	Tipo de medida
Tolerancia a fallos	Prevención de fallas	Externa	Cuántas fallas iniciales estuvieron bajo control para evitar fallas críticas	Contar el número de ocurrencia de fallas serias y críticas evitadas contra los casos de pruebas de fallas iniciales y el número de casos de pruebas de fallas iniciales ejecutados durante las pruebas.	$A = A/B$ A= Numero de Ocurrencia de fallas evitadas contra los casos de pruebas de fallas iniciales. B= Número de casos de pruebas de fallas iniciales Ejecutadas durante las pruebas. Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más cerca del 1 este es mejor		X, A, B= contable
	Redundancia	Interna/ Externa	Cuántos tipos de componentes / sistemas son instalados de forma redundante para evitar un fallo en el sistema		$X = A/B$ A= Número componentes/sistemas instalados de forma redundante. B-Número total de componentes sistemas instalados. Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más cerca del 1 este es mejor		X, A, B= contable

Figura 10: Parámetros de la característica Tolerancia a fallos

Fuente: Tomado de (ISO/IEC, 2022)

Característica	Métrica	Fase de ciclo de vida	Propósito métrico	Método de aplicación	Formulas/ variables	Parámetro aceptación	de	Tipo de medida
Capacidad de recuperación	Tiempo de recuperación	Interna / externa	Cuál es el tiempo promedio que toma el sistema en recuperarse completamente después de un fallo?	Tomar el tiempo que le tomo al sistema en recuperarse y contar el número de caos en los cuales se ha observado que el sistema entro en recuperación.	$X = A/T$ A=Número de casos en los cuales se ha observado que el sistema entro en recuperación. T =Tiempo que le tomo al sistema en recuperarse. Donde: $T > 0$	$X = A/T$ El más Cercano a 0/t es el mejor. Donde el peor caso es $\geq 10/t$		X, A= Contable T= tiempo

Figura 11: Parámetros de la característica Capacidad de recuperación

Fuente: Tomado de (ISO/IEC, 2022)

CAPÍTULO III. METODOLOGÍA.

1. Tipo y diseño de la investigación.

Esta investigación se llevó a cabo basado en el método mixto el cual, en el apartado cuantitativo, se basa en la norma ISO/IEC 25000. Con la cual se evaluó la fiabilidad del sistema de información geográfica para alertas tempranas fitosanitarias agrícolas. Se tomaron en cuenta las distintas métricas de esta norma con la finalidad de determinar la fiabilidad de la gestión de las alertas. Por el apartado cualitativo se analizó los resultados obtenidos y se emite una descripción consolidada la cual corresponde a la efectividad de la evaluación de fiabilidad de la gestión de las alertas.

1.1. Según el tipo de inferencia

Investigación inductiva. Se aplicó este tipo de investigación debido a la inferencia de métricas de calidad del software correspondientes a la familia ISO/IEC 25000, en específico el aspecto de fiabilidad (nivel de madurez, tolerancia a fallos, recuperación, disponibilidad), y a partir de esto se generaron los resultados, de acuerdo con el planteamiento de la hipótesis, esperando la posibilidad de una conclusión universal del proyecto.

Investigación deductiva. Fue aplicado mediante la implementación del sistema de información geográfico, por medio específico del *web mapping*, se solucionó el problema de gestión de las alertas fitosanitarias a través de la aplicación de procesos automáticos que permitan identificar los riesgos de manera temprana y de esta manera concluir con la hipótesis planteada.

2. Población de estudio y tamaño de muestra

Se considera el tipo de investigación planteada, se define una población infinita. Mediante la aplicación de simulación con la herramienta “*Apache JMeter*”, se obtuvieron los datos, para evaluar la fiabilidad de acuerdo con las métricas establecidas por el modelo de calidad ISO/IEC 25000.

3. Hipótesis

La implementación del sistema de información geográfica incide en la fiabilidad de las alertas tempranas fitosanitarias en el cantón Francisco de Orellana.

3.1. Variables

3.1.1. Variable dependiente

Fiabilidad de la gestión de alertas tempranas fitosanitarias.

3.1.2. Variable independiente

Sistema de información geográfica.

3.2. Operacionalización de variables

Tabla 2: Operacionalización de variables.

Variables	Definición	Dimensiones	Indicadores
Variable independiente			
Sistema de información geográfica.	Es la capacidad que actualmente tiene un SIG para efectuar el análisis que ofrece la posibilidad de identificar las relaciones espaciales de los fenómenos que se estudian.	Funcionalidad	Número de metodologías de desarrollo de software GIS analizadas. Porcentaje de cumplimiento de los requisitos funcionales.
Variable dependiente			
Fiabilidad de la gestión de alertas tempranas fitosanitarias.	Capacidad de ayudar a la toma de decisiones mediante la prevención y curación de las enfermedades de las plantas o relacionado con ello.	Fiabilidad	Nivel de Madurez Métricas: <ul style="list-style-type: none"> • Eliminación de fallos • Cobertura de pruebas • Tiempo medio entre fallos Nivel de Disponibilidad Métricas: <ul style="list-style-type: none"> • Tiempo de servicio • Inactividad Nivel de Tolerancias a fallos Métrica: <ul style="list-style-type: none"> • Prevención de fallas • Redundancia Nivel de Capacidad de recuperación. Métricas: <ul style="list-style-type: none"> • Tiempo de recuperación

4. Métodos de análisis

4.1. Método estadístico

4.1.1. Análisis exploratorio de datos.

Después de haber ejecutado la evaluación de fiabilidad por medio de las métricas de calidad de la ISO/IEC 25000 este método nos permitió realizar un análisis de los datos recolectados, mediante gráficos estadísticos, para posteriormente emitir una interpretación detallada de los valores hallados.

4.2. Técnicas de recolección de datos

Para recabar la información se tomaron en cuenta distintas técnicas como encuestas, entrevistas, observación. Con la ejecución de la evaluación de la métrica de fiabilidad con la norma ISO/IEC 2500 se aplicaron los distintos parámetros establecidos en esta métrica.

4.2.1. Observación

Mediante esta técnica se logró determinar las falencias e ineficiencias que existían en el proceso de gestión de alertas fitosanitarias agrícolas, por lo cual no garantiza acciones inmediatas en el proceso del cultivo del cacao. Las pérdidas económicas para los propietarios de las plantaciones del sector. Con este precedente se procedió a realizar la investigación para la implementación del sistema de alertas tempranas fitosanitarias agrícolas con la ayuda de un Sistema de información geográfico. Y mediante la métrica de fiabilidad de la ISO/IEC 2500 evaluar que tan fiable es la gestión de las alertas.

4.2.2. Entrevistas

Con esta técnica se logró establecer los requerimientos tanto funcionales como no funciones para el desarrollo del proyecto, así también los parámetros de éxito que determinan una gestión eficiente y fiable de las alertas fitosanitarias agrícolas. Con esta información se planteó la metodología de desarrollo y planificación para proceder con la investigación

5. PROCEDIMIENTO

A continuación, se mencionan los pasos a seguir para realizar el desarrollo de la investigación:

- Revisar el sustento teórico sobre el tema de investigación.
- Analizar las metodologías de desarrollo enfocadas a los Sistemas de Información Geográfica.
- Planificar el desarrollo del Web Mapping de alertas tempranas fitosanitarias.
- Desarrollar el SIG Web Mapping para el manejo de alertas tempranas fitosanitarias.
- Implantar el sistema de información geográfica para el manejo de alertas tempranas fitosanitarias.
- Evaluar la fiabilidad del sistema implantado tomando en cuenta la familia de normas ISO/IEC 25000.
- Interpretar, discutir y analizar los resultados obtenidos.
- Comprobar la hipótesis planteada.
- Realizar las conclusiones y recomendaciones.

6. Procesamiento y análisis

6.1. Análisis de metodologías de desarrollo

Para seleccionar la metodología de desarrollo de software se realizó una comparación de metodologías tradicionales y ágiles más representativas, para lo cual se tomó en cuenta el los requerimientos y la naturaleza del proyecto, analizando a profundidad las metodologías ágiles, con el fin de adecuar dicha metodología a los sistemas de información geográfica,

esto se realizó por medio de un desglose de las características fundamentales de los SIG y un respectivo encaje con las soluciones que presenta cada metodología de manera individual.

6.2. Desarrollo del sistema para la gestión de alertas tempranas

La metodología XP fue el eje fundamental para el desarrollo del sistema, en el cual, se adecuo el trabajo durante las diferentes etapas correspondientes a planificación, diseño, desarrollo y pruebas.

La primera etapa se identificaron los requerimientos funcionales y no funcionales como indica en la Tabla 3. Los cuales permitieron encaminar el software al uso de herramientas libres para lograr solventar todas las necesidades tanto en el desarrollo *frontend*, *backend* y configuración del servidor donde será alojado, permitiendo establecer el lineamiento de desarrollo, así como las fechas de entrega y la construcción de la planificación por parte de los desarrolladores.

Tabla 3: Tabla de requerimientos funcionales y no funcionales

Requerimientos funcionales	Requerimientos no funcionales
El sistema permitirá mostrar las estaciones mediante un mapa interactivo.	La aplicación debe ser alojada en un servidor Linux.
El sistema permitirá mostrar la información relevante de cada estación presentada en el mapa.	La aplicación web debe funcionar en los navegadores web modernos.
El sistema permitirá mostrar la información de los tipos de alertas y su significado.	La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.
El sistema permitirá el registro de usuarios para recibir alertas a su correo.	El sistema debe contar con manuales de usuario estructurados adecuadamente.
El sistema ejecutara el modelo (Model H5) mismo que genera las alertas fitosanitarias y enviara las notificaciones a cada uno de los correos de los usuarios registrados en la BD, así como insertara en una BD para mostrar el historial de alertas en el sistema web.	El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
El sistema podrá emitir los siguientes gráficos estadísticos de los datos de cada estación, de acuerdo con los datos meteorológicos emitidos por cada estación, como: humedad, temperatura, humedad	El sistema debe poseer interfaces gráficas bien formadas.

relativa. A través de gráficos de barras y gráficos de líneas.

El sistema deberá contener manual de usuario y acerca de.

El sistema contendrá la distribución de interfaz de acuerdo a los estándares establecidos por el cliente.

Los datos obtenidos de los usuarios serán revisados durante el desarrollo y controlará el acceso según las necesidades.

La planificación del sistema no afectará ningún aspecto funcional del software.

La base de datos cambiará durante el desarrollo y será acoplada al funcionamiento final del sistema.

La funcionalidad del sistema cambiara durante el desarrollo sin afectar el aspecto modular.

La interfaz será definida y modificada durante el desarrollo según la estandarización.

El sistema permitirá la integración de un Modelo matemático entregado a lo largo del desarrollo.

El sistema permitirá acoplar las necesidades en su completitud mediante el trabajo de un equipo de dos personas.

El sistema debe ser desarrollado utilizando herramientas y software libres.

El sistema debe asegurar que los datos estén protegidos del acceso no autorizado

El sistema debe contar mínimo con un 85 % de fiabilidad evaluado bajo la norma ISO/IEC 25000

En la fase correspondiente al diseño se realizaron bosquejos de la interfaz y la respectiva distribución de los componentes gráficos para su consecuente aprobación por parte del cliente, de la misma manera se escogieron las herramientas destinadas al desarrollo de la interfaz como la librería de *Bootstrap* y *AdminLTE*.

De la misma manera se ejecutó el diagramado del software con la finalidad de entender el funcionamiento del sistema web y como estará desplegado los componentes de este. Como podemos observar en la figura 12 se muestra el caso de uso del Usuario el mismo que tiene opciones de registro, y visualización de los diferentes módulos del sistema web.

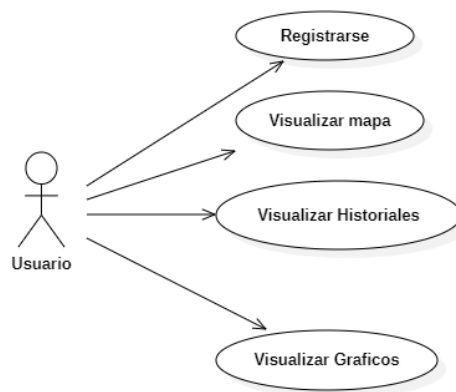


Figura 12: Diagrama de caso de uso Usuario

En la figura 13 Podemos observar el diagrama con los componentes necesarios para el funcionamiento del sistema web en la cual interactúan con la arquitectura como: modelo, vista, controlador.

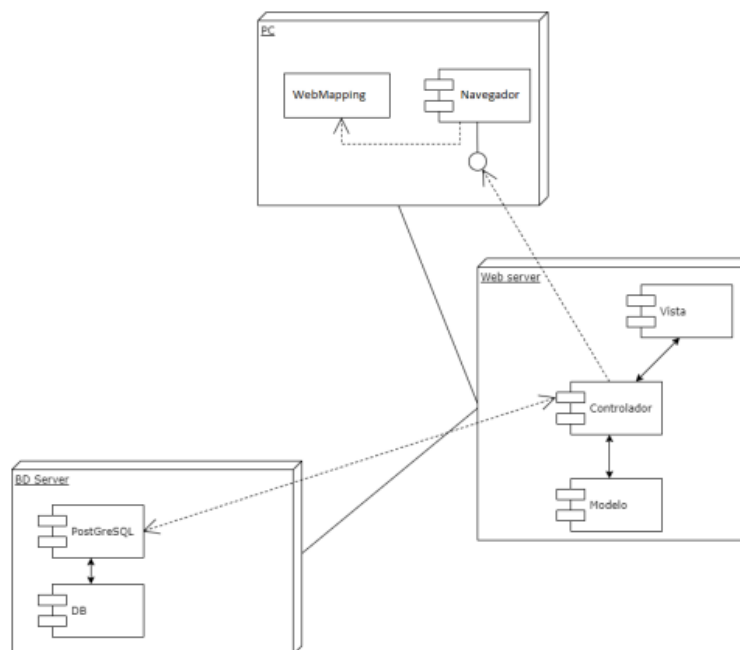


Figura 13: Diagrama de Despliegue de componentes.

La figura 14 muestra el diagrama entidad relacion, la cual muestra cada una de las entidades y relaciones entre las tablas que forman la base de datos, la cual consume el sistema para mostrar los datos de las estaciones metereologicas.

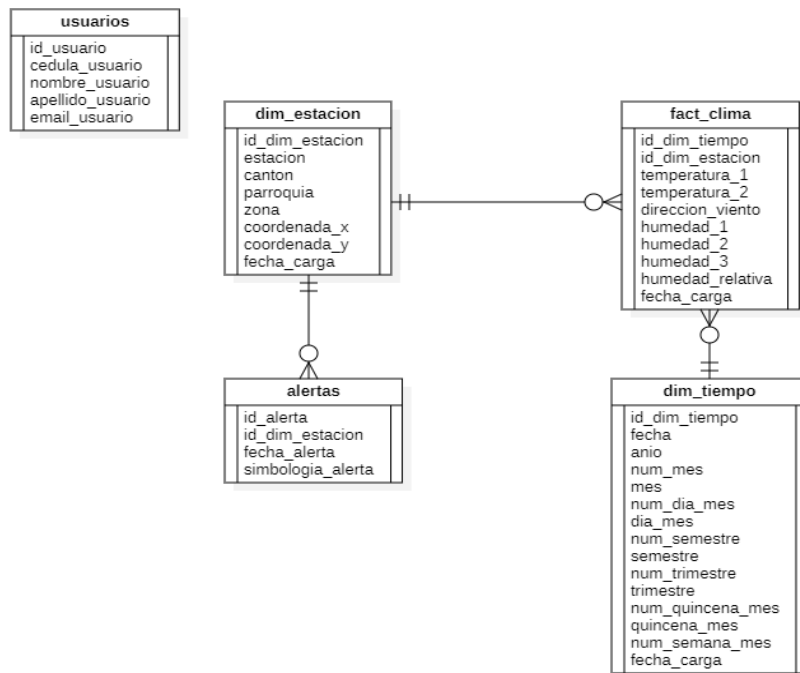


Figura 14: Diagrama Entidad relación.

En la figura 15 podemos observar el diagrama de colaboración en el cual muestra la secuencia que tendrá el usuario para realizar las operaciones existentes dentro del sistema.

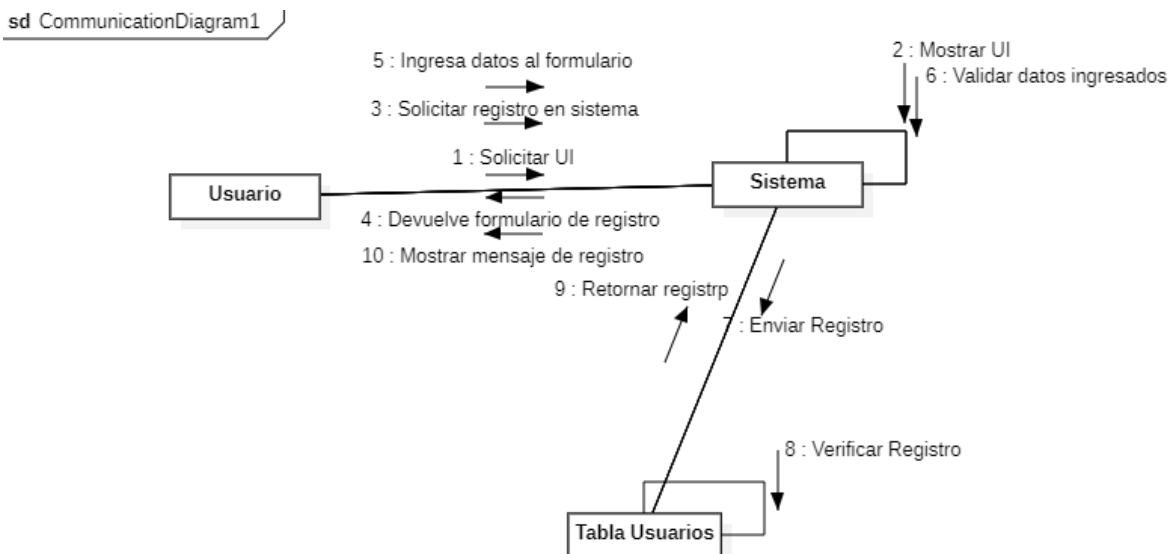


Figura 15: Diagrama de colaboración

Posteriormente en la fase de desarrollo se construyó el software de manera continua con el acompañamiento del cliente el cual especificaba cambios según las necesidades generadas, de esta manera se utilizaron las herramientas de software libre según las especificaciones en la fase de planificación, utilizando el lenguaje de programación *PHP* como base para el desarrollo web, PostgreSQL como gestor de bases de datos, *JavaScript* para la manipulación de las funcionalidades de la librería *OpenLayers* que permite visualizar los mapas necesarios

en la aplicación y de igual manera se utilizó el lenguaje de programación *Python* con el objetivo de utilizar la librería *TensorFlow* en el desarrollo de las alertas enviadas al correo. Todas las herramientas se instalaron en un sistema operativo *Linux Server* versión 20.04 mismo que permitió alojar el sistema conectando todas las tecnologías necesarias para su correcto funcionamiento.

Dentro de la codificación del sistema, se plantearon estándares y normas para la codificación de tal forma que se pueda obtener un código ordenado mediante: funciones, clases, reglas, etc. De tal forma se observa en la tabla 4 la cual muestra las clases de cada módulo debe contener el nombre del módulo seguido por `_class`. Dentro de esta clase debe contener cada una de las funciones: Mostrar, Agregar, Modificar.

Tabla 4: Estructura de clases.

Clase user_class
<pre> class user_class{ function Crear_Usuario(){ include('../DAO/DAO.php'); if(isset(\$_POST["operation"])) { if(\$_POST["operation"] == "Add") { try { \$sentencia = \$connection_string->prepare("SELECT public.sp_proceso_usuarios(1, :cedula_usuario, :nombre_usuario, :apellido_usuario, :email_usuario, 1) as respuesta"); \$result = \$sentencia->execute(array('cedula_usuario' => \$_POST["cedula"], 'nombre_usuario' => \$_POST["nombre"], 'apellido_usuario' => \$_POST["apellido"], 'email_usuario' => \$_POST["email"])); \$rest= \$sentencia->fetchAll(); if(!empty(\$rest)) { if(\$rest[0]["respuesta"]=="") { echo 'REGISTRO INSERTADO CORRECTAMENTE'; } else { echo \$rest[0]["respuesta"]; } } } catch(Exception \$e) { echo 'HA OCURRIDO UN ERROR AL REGISTRARSE INTENTE OTRA VEZ'; } } } } </pre>

```

}

function Mostrar_Usuario(){
    try
    {
        $query = "";
        $output = array();
        $query .= "SELECT * FROM usuarios ";
        $query1 = "SELECT * FROM usuarios ";
        $sentencia = $connection_string->prepare($query1);
        $sentencia->execute();
        $result = $sentencia->fetchAll();
        $data = array();
        $filtered_rows = $sentencia->rowCount();
        foreach($result as $row)
        {
            $sub_array = array();
            $sub_array[] = $row["id_usuario"];
            $sub_array[] = $row["cedula_usuario"];
            $sub_array[] = $row["nombre_usuario"];
            $sub_array[] = $row["apellido_usuario"];
            $sub_array[] = $row["email_usuario"];
            $sub_array[] = $row["nivel_usuario"];
            $sub_array[] = '<div align="center"> <button type="button" name="update" id="'. $row["id_usuario"].'" class="btn btn-efecto
            btn-info update"> <i type="button" class="fa fa-edit" > </i> </button>'. " <button type="button" name="delete"
            id="'. $row["id_usuario"].'" class="btn btn-efecto btn-danger delete"> <i type="button" class="fa fa-trash btn-
            xs"></i></button></div> ';
            $data[] = $sub_array;
        }
        $output = array(
            "draw" => intval($_POST["draw"]),
            "recordsTotal" => $filtered_rows,
            "recordsFiltered" => $sentencia->rowCount(),
            "data" => $data
        );
        echo json_encode($output);
    }
    catch(Exception $e)
    {
    }
}

```

Así mismo en cada módulo estará estructurado de la manera que se muestra en la figura 16, La carpeta principal del módulo el Formulario con el nombre del módulo y tendrá una carpeta con los recursos que den funcionamiento a dicho modulo como son. Css, javascript y archivos PHP.

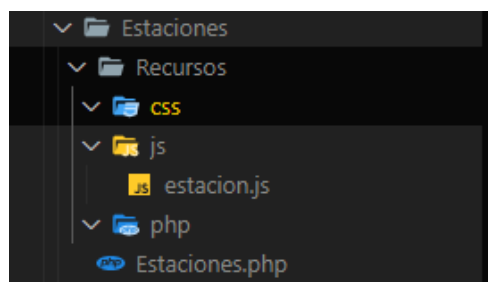


Figura 16: Estructura de archivos de módulos.

Finalmente, se ejecutaron las pruebas respectivas a los módulos del sistema, los cuales indican en la figura 17, con el objetivo de validar el funcionamiento integro de los componentes individuales del sistema y así dar paso a la aprobación por parte del cliente.

MODULO	FORMULARIO	CARACTERÍSTICA PARA PROBAR
Administración	Login	<ul style="list-style-type: none"> • Ingresar credenciales de ingreso • Validar campos • Mensaje de confirmación
Alertas	Alertas	<ul style="list-style-type: none"> • Notificar alertas al correo • Notificar alertas en el sistema web
Geo	Geo	<ul style="list-style-type: none"> • Graficar los marcadores de las estaciones meteorológicas en el mapa
Gráficos	Dashboard	<ul style="list-style-type: none"> • Mostrar gráficos estadísticos
Pop Up	Popup	<ul style="list-style-type: none"> • Cargar información del marcador para visualizar gráficos

Figura 17: Módulos del sistema Web de alertas fitosanitarias

Se llevaron a cabo pruebas unitarias con la finalidad de verificar el funcionamiento de cada función individual dentro del sistema web, como se muestra en la figura 18 la estructura de las pruebas que evalúan las distintas características.

Nº	Encargado	Tipo de pruebas	Modulo	Formulario	Función																
1	Leonidas Aguaiza	Unitarias	Administración	Login	Registrar Usuario																
CONDICIONES																					
		Entradas	Valor																		
		Cedula:	1234567890																		
		Nombres:	Luis Fernando																		
		Apellidos:	Lopez Chavez																		
		E-mail:	maycoleonidas1997@gmail.com																		
		Nivel:	Usuario																		
		Usuario:	Mafe5																		
		Contraseña:	Admin123\$																		
Resultados			/ evidencia																		
Esperado: Iniciar sesión				Obtenido: Error																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Log</th> <th style="width: 50%;">Reference</th> </tr> </thead> <tbody> <tr> <td>11. type on id=email with value tecnoshop365@gmail.com</td> <td>OK</td> </tr> <tr> <td>12. click on id=usuario</td> <td>OK</td> </tr> <tr> <td>13. type on id=usuario with value maria</td> <td>OK</td> </tr> <tr> <td>14. click on id=contrasena</td> <td>OK</td> </tr> <tr> <td>15. type on id=contrasena with value Admin123\$</td> <td>OK</td> </tr> <tr> <td>16. click on id=action</td> <td>OK</td> </tr> <tr> <td>17. assertAlert on REGISTRO INSERTADO CORRECTAMENTE</td> <td>Failed: Alert message doesn't match actual message</td> </tr> </tbody> </table>						Log	Reference	11. type on id=email with value tecnoshop365@gmail.com	OK	12. click on id=usuario	OK	13. type on id=usuario with value maria	OK	14. click on id=contrasena	OK	15. type on id=contrasena with value Admin123\$	OK	16. click on id=action	OK	17. assertAlert on REGISTRO INSERTADO CORRECTAMENTE	Failed: Alert message doesn't match actual message
Log	Reference																				
11. type on id=email with value tecnoshop365@gmail.com	OK																				
12. click on id=usuario	OK																				
13. type on id=usuario with value maria	OK																				
14. click on id=contrasena	OK																				
15. type on id=contrasena with value Admin123\$	OK																				
16. click on id=action	OK																				
17. assertAlert on REGISTRO INSERTADO CORRECTAMENTE	Failed: Alert message doesn't match actual message																				
Descripcion del error: El error se debe a que no esta llamando(no referenciado) a la funcion insertar al dar clic en el boton ingresar																					

Figura 18: Prueba unitaria

De la misma manera se ejecutaron pruebas de componentes para validar el funcionamiento de los módulos y la integración de estos en general, como se muestra en la figura 19 la cual permite verificar la estructura de cómo se debe realizar cada prueba.

Caso de prueba DAO					
Modulo	DAO				
Tarea	Conexión				
Tester	Alexander Vacacela	Fecha	10/02/2022		
# Prueba	Nombre	Condición	Pasos de ejecución	Respuesta esperada	Resultado
cp1	Comrpobar conexión	Acceso a la base de datos	1. Abrir el archivo en el navegador	No se muestra ningún error	Se establece correctamente la conexión
cp2	Comrpobar errores	Acceso al archivo sin conexión a la base de datos	1. Abrir el archivo en el navegador	Se visualiza un error en la consola	Se envia el mensaje "Ocurrió un error con la base de datos:" junto al error suscitado

Figura 19: Prueba de componentes

6.3. Evaluación de la fiabilidad del sistema

Una vez concluido el sistema con su correspondiente aprobación por parte del cliente, se procede a evaluar el software mediante el uso de la familia de normas ISO/IEC 25000, específicamente el apartado de fiabilidad, gracias a la herramienta de simulación “*Apache JMeter*” la cual se ejecutó una vez el sistema se encuentre configurado en el servidor donde residirá; como resultado de la evaluación se obtuvieron los respectivos datos pertenecientes a las métricas de cada una de las características de madurez, disponibilidad, tolerancia a fallos y la capacidad de recuperación que permiten evaluar la fiabilidad según la norma.

En el caso de la métrica eliminación de fallos de la sub característica de madurez, la métrica de prevención de fallos y la métrica de redundancia de la sub característica de tolerancia a fallos se toman en cuenta las cualidades necesarias para lograr una ponderación de dichas métricas, puntualizando el desempeño durante el desarrollo del sistema para lograr una correcta evaluación.

6.4. Resultados de la evaluación

Al final de la evaluación se obtuvieron los datos finales de las métricas que permiten describir el estado del sistema, dando lugar a un análisis profundo de los objetivos propuestos en donde se adecuaron los respectivos datos a manera de lograr procesarlos para posteriormente comprender los resultados de manera individual y agrupada.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

7. Resultados

7.1. Resultado del análisis de metodología

Para el desarrollo del *Webmapping* fue necesario el uso de una metodología de desarrollo de software “Xtreme Programming” (XP), La misma que se logró dar un enfoque a los sistemas de información geográfica, para lo cual se plantea el uso de un esquema que reúna las características básicas que deban ser satisfechas para completar de la mejor manera el software y posterior realizar la comparativa de varias metodologías que tengan la capacidad de cumplir las necesidades del cliente.

7.1.1. Esquema básico

Como primer punto se analizó la naturaleza de los requerimientos esenciales de un SIG dando como resultado los siguientes aspectos.

Proceso conceptual de diseño. - Consiste en modelar la realidad del problema existente mediante estructuras y funciones que permitan reflejar los eventos requeridos por el cliente. Teniendo los siguientes pasos para el flujo de las actividades:

- Seminario sobre SIG.
- Evaluación de las necesidades del usuario.
- Análisis de requerimientos.
- Diseño conceptual.
- Plan de implementación.

Desarrollo de la base de datos. - Toma los aspectos considerados en el proceso conceptual para realizar la implementación primaria de los requerimientos principales. Dentro de esta etapa se consideran las siguientes actividades:

- Diseño físico y diccionario de datos.
- Estudio piloto.
- Conversión de datos.
- Generación de productos.
- Manual de procedimientos.

Desarrollo de la aplicación. - Conlleva la creación de los procedimientos específicos que solventen las necesidades operativas de los usuarios. Las actividades que se sugieren en esta etapa son las siguientes:

- Requerimientos y especificaciones.
- Diseño prototipo y pruebas.
- Instalación y entrenamiento.

Instalación del sistema y entrenamiento. - Por último, se considera la implantación del sistema teniendo en cuenta las siguientes características:

- Especificaciones de adquisición de métodos, equipos y sistemas.

- Instalación de Hardware y software.
- Entrenamiento.
- Soporte continuo al usuario.

7.1.2. Metodologías adaptadas al desarrollo de sistemas de información geográfica

A través de un cuadro comparativo se resumen las metodologías capaces de satisfacer las necesidades mencionadas proporcionando una breve descripción de cada una, así como la ponderación por medio de ventajas y desventajas generales.

Tabla 5: Metodologías ágiles.

Metodología	Descripción	Fases para el desarrollo	Ventajas	Desventajas
XP	<i>Extreme Programming</i> está centrada en potenciar relaciones interpersonales, promoviendo el trabajo en equipo, tomando en cuenta el aprendizaje de los desarrolladores y ofreciendo un buen ambiente de trabajo.	<ul style="list-style-type: none"> • Exploración. • Planificación de la entrega. • Iteraciones. • Producción. • Mantenimiento. • Muerte del proyecto. 	<ul style="list-style-type: none"> • Crecimiento lento del costo del cambio. • Relación estrecha con el cliente. 	<ul style="list-style-type: none"> • Arquitectura definida de manera evolutiva. • El tiempo puede aumentar indefinidamente.
<i>Scrum</i>	Caracterizado por el desarrollo mediante iteraciones “ <i>Sprint</i> ” con duración de 30 días y reuniones diarias para la integración del proyecto.	Planeación del Sprint. Reuniones de equipo Scrum. Refinamiento del Backlog. Revisión del Sprint. Retrospectiva del Sprint.	Gestión de riesgos y expectativas del cliente. Flexibilidad y adaptación.	Requerimiento de equipos de desarrollo con vasta experiencia. Funciona con equipos pequeños.
<i>Crystal methodologies</i>	Centrada en el número de integrantes del equipo y la	<ul style="list-style-type: none"> • Puesta en escena. • Revisiones. • Monitoreo. 	<ul style="list-style-type: none"> • Adaptativa al tamaño de los miembros del equipo. 	<ul style="list-style-type: none"> • Delimita el alcance del proyecto.

	reducción de artefactos producidos.	<ul style="list-style-type: none"> • Paralelismo y flujo. • Estrategia de diversidad holística. • Técnica de puesta a punto de la metodología. • Puntos de vista de usuario. 	<ul style="list-style-type: none"> • Planificación transparente. 	<ul style="list-style-type: none"> • No es útil para proyectos grandes.
<i>FDD</i>	<i>Feature Driven Development</i> se centra en las fases de diseño e implementación con iteraciones cortas.	<ul style="list-style-type: none"> • Desarrollar un modelo global. • Construir una lista de funcionalidades. • Planificar por funcionalidad. • Construir por funcionalidad. 	<ul style="list-style-type: none"> • Enfocado en las necesidades. • Simplicidad en el diseño. 	<ul style="list-style-type: none"> • Difícil durante desarrollo de interfaz de usuario. • Visión general compleja.
<i>ASD</i>	<i>Adaptative Software Development</i> se caracteriza por ser iterativo, orientado a los componentes software sobre las tareas y tolerante a los cambios.	<ul style="list-style-type: none"> • Especulación. • Colaboración. • Aprendizaje. 	<ul style="list-style-type: none"> • Promulga la colaboración e interacción de las personas. • Aprende de los errores iniciando un ciclo nuevo de desarrollo. 	<ul style="list-style-type: none"> • Los errores no detectados afectan la calidad y costo final.

Fuente: Tomado del artículo (Letelier, 2006)

Debido a la naturaleza de los requerimientos del sistema y de igual manera considerando la adaptación de un esquema básico; basado en (Guevara, 1992), quien presenta una metodología para el diseño e implementación de un SIG, el cual se centra en colocar al usuario como eje principal alrededor del diseño, considerando a la base de datos como un reflejo del modelo conceptual y operativo de las necesidades del aplicativo a desarrollar, de igual manera considerando la necesidad del cliente por construir un software de calidad con intermitencia en los requerimientos y la limitación en el tamaño del equipo de desarrollo se concluye apropiado utilizar la metodología *Extreme Programming* como guía en la construcción del sistema de alertas tempranas fitosanitarias.

7.2. Resultado del desarrollo del sistema de información geográfica

A partir de la investigación realizada, se desarrolló en base a la metodología seleccionada, (XP). Bajo lo cual se basó en cada una de las fases correspondientes. A continuación, se presenta los resultados obtenidos en cada una de las tareas ejecutadas dentro de la planificación de la metodología.

7.2.1. Fase de planificación

En la primera fase se llevaron a cabo, reuniones, entrevistas con los encargados del proyecto con el objetivo de dar resolución a los requerimientos funcionales y no funcionales los mismos que fueron punto de partida para la ejecución del proyecto, dentro de esta etapa se desarrolló un documento de requerimientos.

Documento de especificaciones de requisitos de software

“Sistema de información geográfica para la gestión de alertas tempranas fitosanitarias en el cantón francisco de orellana”

Leonidas Gustavo Aguaiza Tutin
Alexander Sebastián Vacacela Gaibor

Documento de Especificaciones de Requisitos de Software

Versión: 1.0

Fecha: 2/12/2021

Figura 20: Documento SRS.

7.2.2. Fase de diseño

Dentro de esta fase se tomó en cuenta el desarrollo de la interfaz del proyecto misma que debe ser llamativa, manejable y comprensible para el cliente, para lo cual se trabajó bajo el framework de Bootstrap con la finalidad de obtener un diseño limpio y estructurado. Bajo este concepto se obtuvo como resultado el diseño que se muestra a continuación, el cual se adapta a las necesidades de los requerimientos.



Figura 21: Diseño de Interfaz.

7.2.3. Fase de codificación

En esta fase se realizaron reuniones periódicas con las partes del proyecto con la finalidad de ir codificando y cumpliendo con los requerimientos planteados, debido a que el involucrar al cliente es importante con el propósito de ir corrigiendo fallos o equivocaciones en el momento del desarrollo, de la misma forma se tomaron en cuenta estándares de codificación con el fin de tener un código ordenado y estructurado mediante funciones y clases. Estas normas se utilizaron en todos los módulos del proyecto.

```

1 <?php
2
3 class user_class{
4
5     function Crear_usuario(){
6
7         include('../dao/dao.php');
8
9         if(isset($_POST["operation"]))
10        {
11            if($_POST["operation"] == "Add")
12            {
13                try
14                {
15                    $sentencia = $connection_string->prepare("SELECT public.sp_proceso_usuarios(1, :cedula_usuario, :nombre
16                    $result = $sentencia->execute()
17                    array(
18                        ':cedula_usuario' => $_POST["cedula"],
19                        ':nombre_usuario' => $_POST["nombre"],
20                        ':apellido_usuario' => $_POST["apellido"],
21                        ':email_usuario' => $_POST["email"]
22                    )
23                );
24                $rest= $sentencia->fetchAll();
25
26                if(!empty($rest))
27                {
28                    if($rest[0]["respuesta"]=="")
29                    {
30                        echo 'REGISTRO INSERTADO CORRECTAMENTE';
31                    }
32                    else
33                }
34            }
35        }
36    }
37 }

```

Figura 22: Estándar de codificación.

De la misma forma se aplicó la programación en pares misma que el proyecto fue desarrollado por módulos, cada módulo culminado se revisó de manera grupal con la finalidad de detectar errores y corregirlos a tiempo. Resultado del cual se obtuvo el sistema web el cual se muestra en la figura 23.

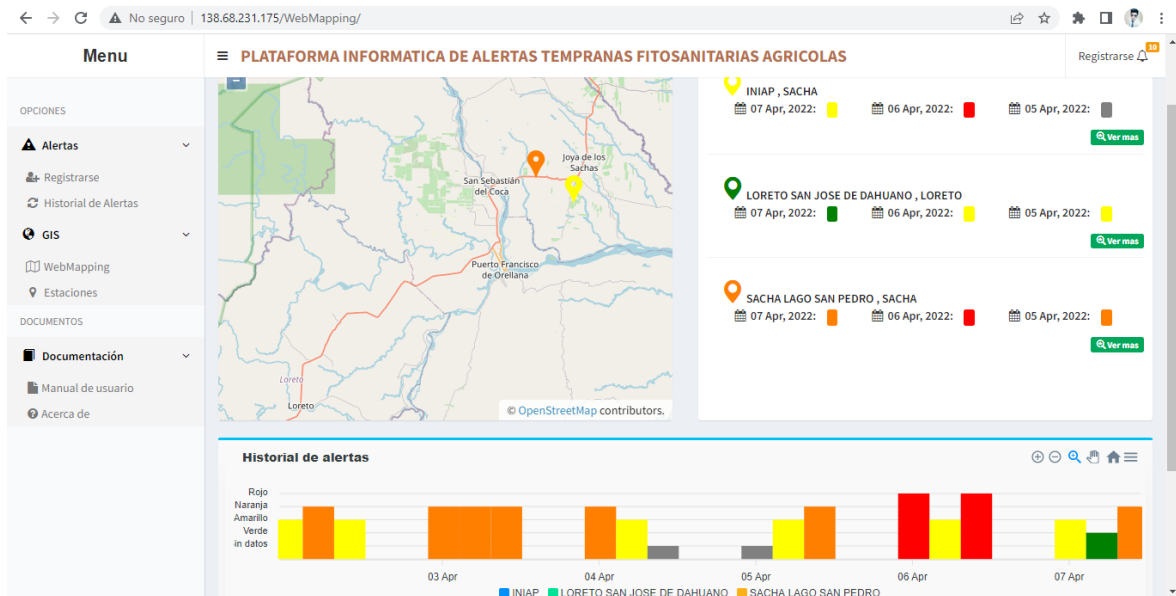


Figura 23: Página principal de Alertas tempranas fitosanitarias.

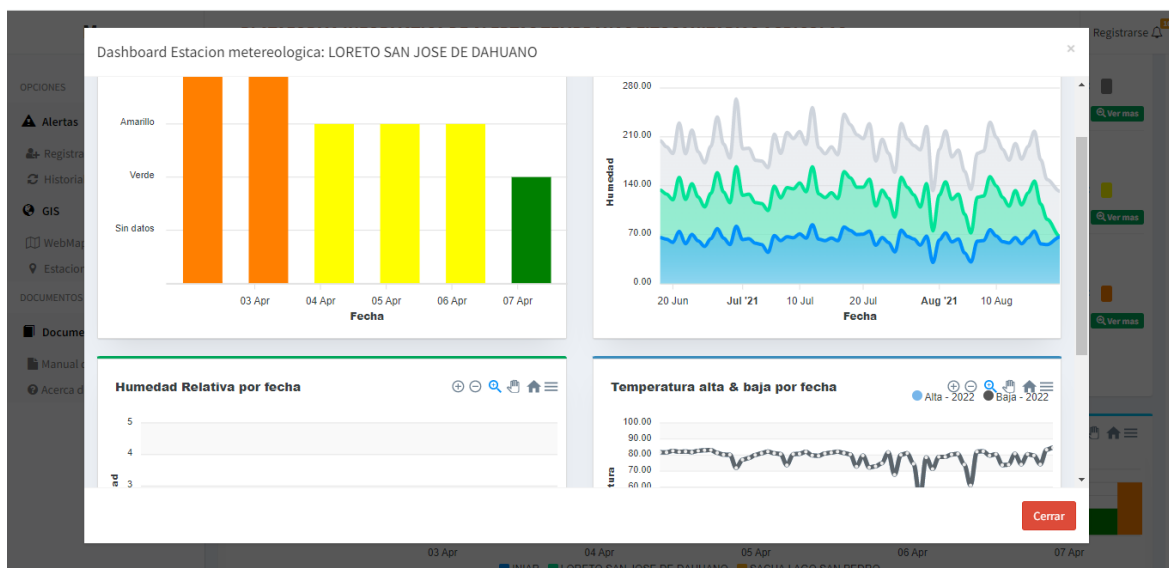


Figura 24: Dashboard de página web de alertas fitosanitarias.

Menu

PLATAFORMA INFORMATICA DE ALERTAS TEMPRANAS FITOSANITARIAS AGRICOLAS

Show 10 entries

Nº	Canton	Estacion	Fecha Alerta	Simbologia
0	LORETO	LORETO SAN JOSE DE DAHUANO	07 Apr, 2022 13:00:00	Verde
1	SACHA	SACHA LAGO SAN PEDRO	07 Apr, 2022 13:00:00	Amarillo
2	SACHA	INIAP	07 Apr, 2022 13:00:00	Amarillo
3	LORETO	LORETO SAN JOSE DE DAHUANO	06 Apr, 2022 13:00:00	Amarillo
4	SACHA	SACHA LAGO SAN PEDRO	06 Apr, 2022 13:00:00	Naranja
5	SACHA	INIAP	06 Apr, 2022 13:00:00	Naranja
6	LORETO	LORETO SAN JOSE DE DAHUANO	05 Apr, 2022 13:00:00	Amarillo
7	SACHA	INIAP	05 Apr, 2022 13:00:00	Sin datos

Historial de alertas

Figura 25: Tabla de alertas fitosanitarias.

7.2.4. Fase de pruebas

Dentro de la fase de pruebas se tomó en cuenta las pruebas ejecutadas al finalizar los módulos y agrupar en un solo sistema, para lo cual se ejecutaron pruebas unitarias, pruebas de componentes. Con la finalidad de detectar errores y corregirlos antes de la entrega del producto.

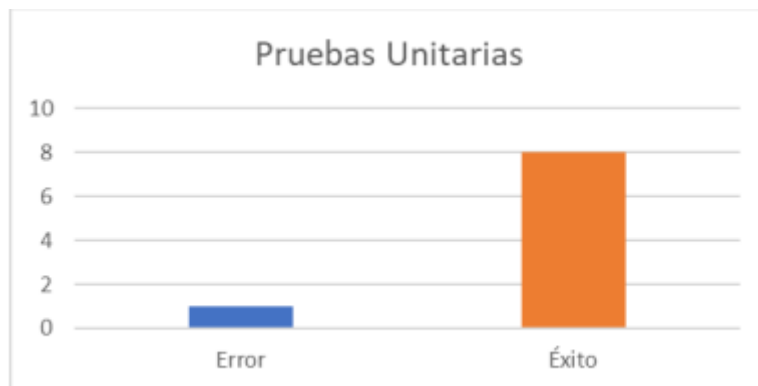


Figura 26: Resultado de pruebas unitarias.



Figura 27: Resultado de pruebas de componentes.

7.3. Resultados de evaluación de fiabilidad

En el presente bloque se redacta de manera detallada los resultados obtenidos a través de los criterios de fiabilidad establecidos por la norma ISO/IEC 25000, contrastando las sub características madurez, disponibilidad, tolerancia a fallos y capacidad de recuperación. La evaluación del sistema de información geográfica para alertas tempranas fitosanitarias agrícolas en el cantón francisco de orellana espera alcanzar una calificación general mayor o igual al 80% como lo especifica el cliente.

7.3.1. Sub característica: Madurez

7.3.1.1. Métrica: Eliminación de fallos

Para evaluar los resultados de la métrica eliminación de fallos se tomaron en cuenta los datos y resultados obtenidos en la documentación del desarrollo del sistema. Dentro del cual se obtuvieron los siguientes resultados. 7 fallos detectadas y 6 fallos corregidas.

Tabla 6: Resultados eliminación de fallos.

Fallas detectadas en las pruebas	Descripción	Fallas corregidas en la fase de diseño/codificación / pruebas
Error al registrar usuario en la DB.	El presente error corresponde a que no se estaba guardando el registro en la base de datos, al momento de dar clic en el botón registrar.	Solución. Direccionar de manera correcta a la clase la cual contiene el código de insertar en la BD.
Error al cargar historial de alertas.	El error aparece cuando se va a la opción historial de alertas.	Solución. Especificar los nombres de los componentes y variables declarados de manera correcta.
No se muestran los marcadores en el mapa.	El error aparece al momento de cargar los marcadores correspondientes de las estaciones.	Solución. Especificar el sistema de coordenadas correspondiente para el tipo de <i>shape</i> utilizado.
La ventana <i>Popup</i> con el nombre de la estación no permite dar el evento clic en el marcador.	El error se presenta al momento de sobreponer el puntero del mouse en los marcadores y la ventana <i>popup</i> , no permite dar el evento clic.	Solución. Hacer que el <i>popup</i> sea <i>responsive</i> y que aparezca 10 pixeles más por encima del marcador.
El botón registrar se oculta inclusive sin registrar correctamente en la DB	El error se presenta al momento de dar en el botón registrar y si hay algún valor duplicado el botón desaparece, lo cual debería desaparecer únicamente cuando el registro sea exitoso.	Solución. Agregar un control que se oculte el botón únicamente cuando el mensaje de inserción sea “El usuario se ha registrado exitosamente”

Los controles de navegación de la información de alertas se sobrepone al texto.	Esto se presenta cuando se carga la página y los controles horizontales se presentan encima del texto.	Solución. No se resolvió debido a que se trabajó con Bootstrap y los tamaños de los controles ya están establecidos dentro de la librería por lo que no se lograron modificar.
El mapa se carga en un solo tamaño.	El error se presenta al momento de expandir la pantalla. El mapa no se expande y queda un recorte en la media pantalla.	Solución. Agregar la clase responsive al contenedor del mapa.

Para evaluar el resultado se lo hizo mediante al formula que elimina los fallos que se muestra en la tabla 7.

Tabla 7: Formula de eliminación de fallos.

Formula	Variable A	Variable B	Variable X
$X=A/B$ A= Número de fallas corregidas en la fase de diseño /pruebas B= Numero de fallas detectadas en las pruebas Donde: $B>0$	6	7	0.86

Finalmente, ya aplicado la fórmula para eliminar fallos detectados, en el valor esperado especifica que mientras más se acerque a 1 es un resultado excelente.

Tabla 8: Resultados obtenidos de la eliminación de fallos.

Métrica	Valor no deseado	Valor esperado	Valor obtenido	Porcentaje de calidad externa obtenido	Porcentaje de calidad externa faltante.
Eliminación de fallos	0	1	0,86	86%	14%

De acuerdo con el valor final obtenido en la métrica, se obtuvo una calificación excelente, demostrando una eficaz corrección de fallos durante el desarrollo del sistema, aportando de esta manera a un aumento en la calidad del resultado final.

7.3.1.2. Métrica: Cobertura de pruebas

Para la evaluación de esta métrica se ejecutaron 2 conjuntos de pruebas utilizando el software de simulación “*Apache JMeter*”, este conjunto de simulaciones se ejecutó con él envío de

peticiones directamente al servidor, en el primer caso de prueba se enviaron 200 peticiones por cada segundo en un total de 5 repeticiones. Como se puede observar en la figura 28. Para el segundo caso de prueba se envió 200 peticiones en 500 segundos en un total de 5 repeticiones. Como se puede observar en la figura 30.

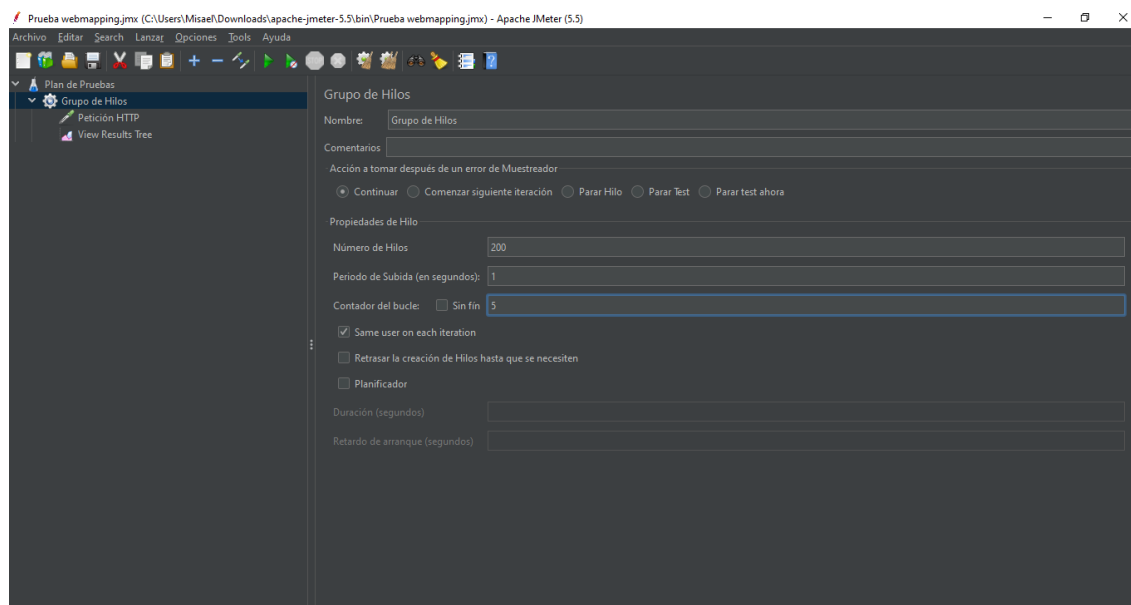


Figura 28: Cobertura de pruebas, primer caso de prueba.

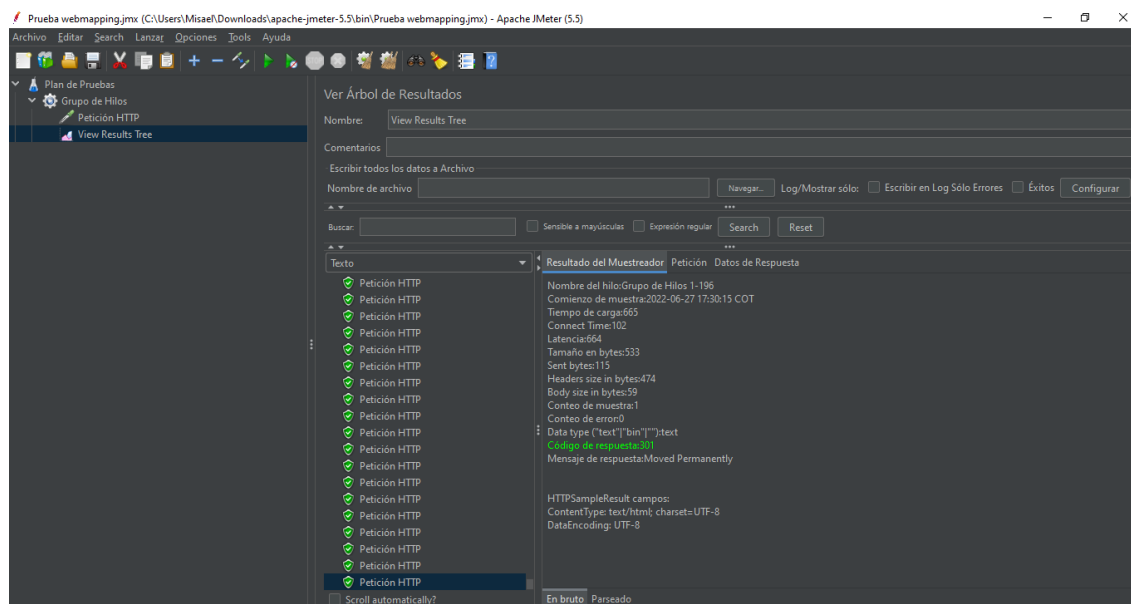


Figura 29: Cobertura de pruebas, resultados de primer caso de prueba.

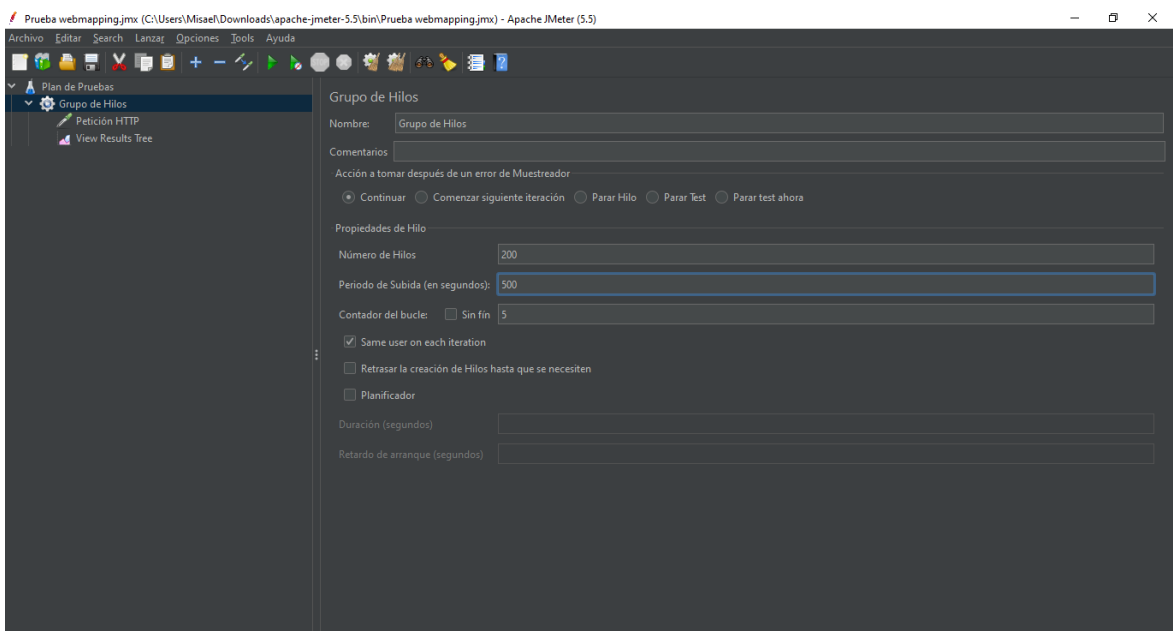


Figura 30: Cobertura de pruebas, segundo caso de prueba.

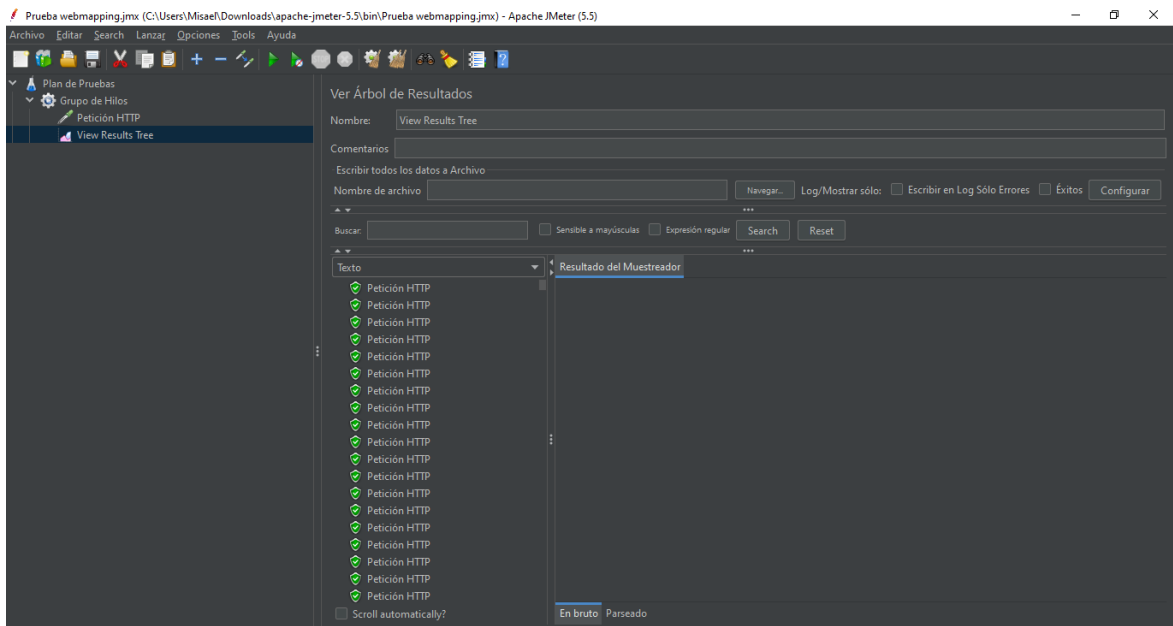


Figura 31: Cobertura de pruebas, resultados de segundo caso de prueba.

Para valorar los resultados obtenidos se lo realizo con la fórmula de cobertura de pruebas como se puede verificar en la tabla 9.

Tabla 9: Formula de cobertura de pruebas.

Formula	Variable A	Variable B	Variable X
$X=A/B$ A= número de casos de prueba realizados en escenarios de operación durante la prueba B= Número de casos de prueba realizados para cumplir los requerimientos Donde: $B>0$	1	1	1

Finalmente, ya aplicado la correspondiente fórmula, los resultados se presentan en la tabla 10. En la cual indica que el valor mientras más se acerque a 1 es mejor y será 0 únicamente en el peor de los casos.

Tabla 10: Resultados de cobertura de pruebas.

Métrica	Valor no deseado	Valor esperado	Valor obtenido	Porcentaje de calidad externa obtenido	Porcentaje de calidad externa faltante.
Cobertura de pruebas	0	1	1	100%	0%

De acuerdo con el valor final obtenido en la métrica, se obtuvo una calificación excelente, demostrando la capacidad de respuesta del servidor ante las peticiones de múltiples clientes de manera simultánea.

7.3.1.3. Métrica: Tiempo medio entre fallos

Para ejecutar la evaluación del tiempo medio entre fallos se efectuó una simulación de estrés en la cual se parametrizo con el número de peticiones de 15000 por segundo en 10 veces como se ve en la figura 32, los resultados encontrados fueron: a partir de la petición 30185 empieza a generar errores debido a que ya no se reciben datos de respuesta, como se visualiza en la figura 33. Esta evaluación se efectuó con la finalidad de medir que capacidad tiene el servidor.

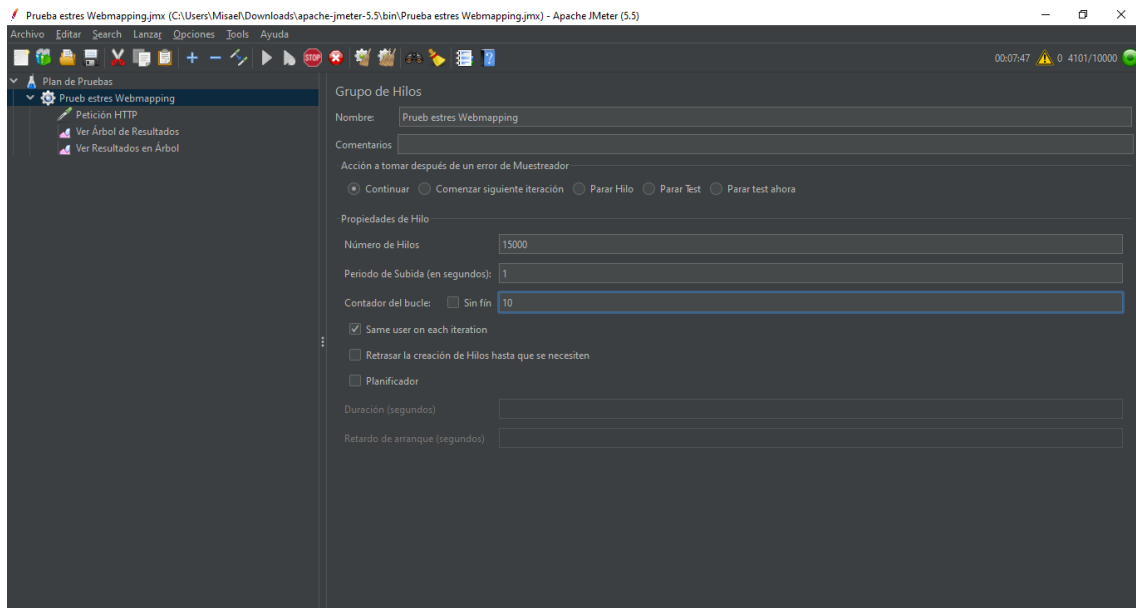


Figura 32: Tiempo medio entre fallos, parámetros.

Figure 33 shows the 'Tree View of Results' (Ver Resultados en Árbol) table. The table lists 20 samples (30181 to 30200) with columns for sample number, time, thread name, label, time to receive, status, bytes, sent bytes, latency, and connect time. Samples 30181-30184 show successful requests, while samples 30185-30200 show failures with a status of 'Error'.

Muestra #	Tiempo de co...	Nombre del hilo	Etiqueta	Tiempo de Mue...	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
30181	20:59:20.470	Prueb estres ...	Petición HTTP	4754	✓	392	110	4754	3651
30182	20:59:23.624	Prueb estres ...	Petición HTTP	1600	✓	392	110	1600	514
30183	20:59:07.491	Prueb estres ...	Petición HTTP	17729	✓	392	110	17729	15353
30184	20:59:07.488	Prueb estres ...	Petición HTTP	17733	✓	392	110	17733	15355
30185	20:59:04.188	Prueb estres ...	Petición HTTP	21034	✗	2328	0	0	21034
30186	20:59:03.959	Prueb estres ...	Petición HTTP	21069	✗	2328	0	0	21069
30187	20:59:04.189	Prueb estres ...	Petición HTTP	21033	✗	2328	0	0	21033
30188	20:59:04.188	Prueb estres ...	Petición HTTP	21034	✗	2328	0	0	21034
30189	20:59:04.188	Prueb estres ...	Petición HTTP	21034	✗	2328	0	0	21034
30190	20:59:04.188	Prueb estres ...	Petición HTTP	21034	✗	2328	0	0	21034
30191	20:59:04.188	Prueb estres ...	Petición HTTP	21034	✗	2328	0	0	21034
30192	20:59:04.188	Prueb estres ...	Petición HTTP	21034	✗	2328	0	0	21034
30193	20:59:04.189	Prueb estres ...	Petición HTTP	21033	✗	2328	0	0	21033
30194	20:59:04.189	Prueb estres ...	Petición HTTP	21033	✗	2328	0	0	21033
30195	20:59:04.192	Prueb estres ...	Petición HTTP	21030	✗	2328	0	0	21030
30196	20:59:04.189	Prueb estres ...	Petición HTTP	21033	✗	2328	0	0	21033
30197	20:59:04.189	Prueb estres ...	Petición HTTP	21033	✗	2328	0	0	21033
30198	20:59:04.190	Prueb estres ...	Petición HTTP	21032	✗	2328	0	0	21032
30199	20:59:04.188	Prueb estres ...	Petición HTTP	21034	✗	2328	0	0	21034
30200	20:59:04.191	Prueb estres ...	Petición HTTP	21031	✗	2328	0	0	21031

Figura 33: Tiempo medio entre fallos, resultados de evaluación.

Tomando en cuenta el caso de prueba y las peticiones que fueron enviadas al servidor para el sistema web “Sistema de información geográfica para alertas tempranas fitosanitarias en el cantón Francisco de Orellana”. Tomando en cuenta las peticiones reales ya implementado el sistema, no llegan a las 30185, teniendo en cuenta que la cantidad de usuarios a usar el sistema no supera los 1000 usuarios se lograría cumplir el objetivo de dicha métrica de manera satisfactoria.

Tabla 11: Resultados de Tiempo medio entre fallos.

Métrica	Valor no deseado	Valor esperado	Valor obtenido	Porcentaje de calidad externa obtenido	Porcentaje de calidad externa faltante.
Tiempo medio entre fallas	0	1000	30185	100%	0%

7.3.2. Sub característica. Disponibilidad

7.3.2.1. Métrica: Tiempo de servicio

Para obtener el resultado de la métrica del tiempo de servicio que proporciona el sistema en este caso está activo y disponible las 24 horas del día. Esto con la finalidad que el usuario pueda acceder a los datos en cualquier momento. El tiempo de servicio operacional es de 24 horas.

Para la evaluación de los resultados se aplicó la fórmula para determinar el tiempo de servicio como se muestra en la tabla 12.

Tabla 12: Formula de disponibilidad.

Formula	Variable A	Variable B	Variable X
$X = A/B$ A=Tiempo del servicio del sistema que se proporciona actualmente B= Tiempo de servicio del sistema regulado en el cronograma operacional Dónde: $B > 0$	24	24	1

Habiendo aplicado la fórmula de medida del tiempo de servicio, los resultados se pueden visualizar en la tabla 13 la misma que indica que mientras sea un valor más acercado a 1 es mejor y el más cercano a 0 es en el peor de los casos.

Tabla 13: Resultados de disponibilidad.

Métrica	Valor no deseado	Valor esperado	Valor obtenido	Porcentaje de calidad externa obtenido	Porcentaje de calidad externa faltante.
Tiempo de servicio	0	1	1	100%	0%

De acuerdo con el valor final obtenido en la métrica, se obtuvo una calificación excelente, demostrando la disponibilidad del servicio ofrecido por el sistema, sintetizando que el sistema se encuentra disponible para el público las 24 horas, siempre y cuando no existan eventualidades que dañen el servidor.

7.3.2.2. Métrica: tiempo medio de inactividad.

Para la evaluación del tiempo medio de inactividad se lo realizo a través de los datos obtenidos de la evaluación de los casos de prueba ejecutados inicialmente en los cuales se simularon mediante la herramienta “*Apache JMeter*” en la cual se enviaron 200 peticiones por segundo en 5 repeticiones y el siguiente caso de prueba se enviaron 200 peticiones en 500 segundos por 5 repeticiones.

Una vez ejecutado el primer caso de prueba se obtuvieron que todas las peticiones fueron efectuadas con satisfacción del 100% el tiempo que se tardó en ejecutar las 200 peticiones fue de 3 segundos como se puede ven en la figura 34. De la misma manera el siguiente caso de prueba fue ejecutado con 200 peticiones cada 500 segundos por 5 repeticiones lo cual fueron ejecutadas satisfactoriamente sin encontrar errores. Esta prueba tardo un tiempo de 8 minutos con 18 segundos como se muestra en la figura 35.

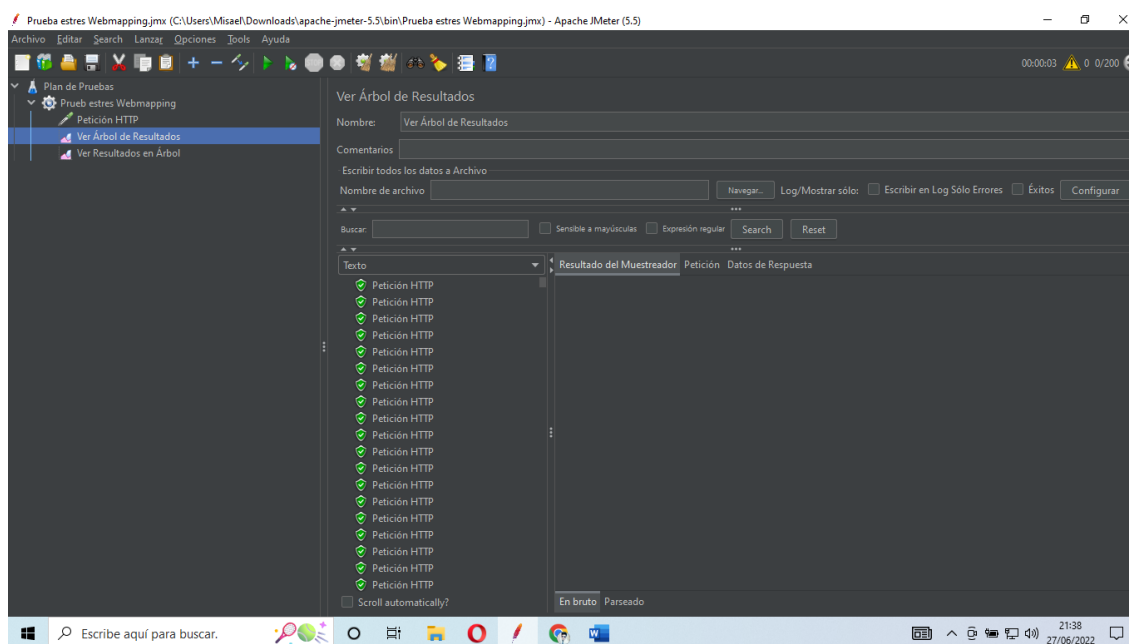


Figura 34: Tiempo medio de inactividad, primer caso de prueba.

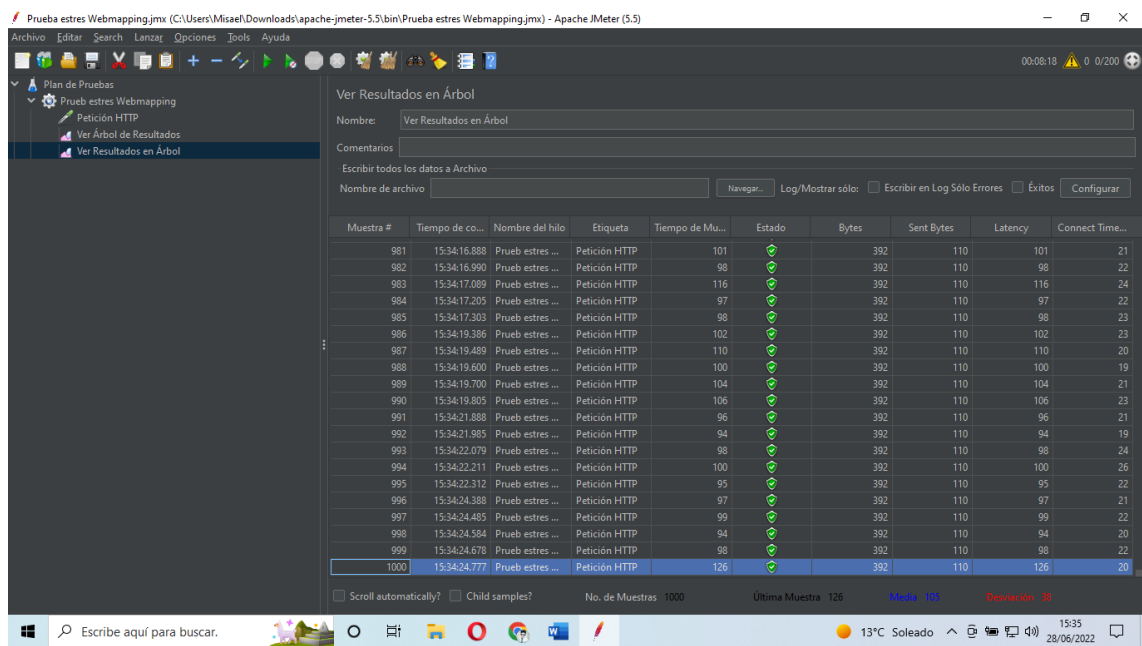


Figura 35: Tiempo medio de inactividad, segundo caso de prueba.

Estos resultados se pueden observar en la tabla 14.

Tabla 14: Ponderación de valores de tiempo medio de inactividad.

Test	Peticiones de caso de prueba	Repeticiones del caso de prueba	Periodo por petición	Peticiones exitosas	Tiempo en ejecutar la prueba	Rendimiento	% error
1	200	5	1 s.	1000	3s	333.3/seg.	0%
2	200	5	500 s.	1000	8 minutos 18 segundos	2/seg.	0%

Para efectuar los resultados se hizo uso de la fórmula de la medida del tiempo medio de inactividad, en donde A es el % de error correspondientemente para presentar los valores en la tabla 15.

Tabla 15: Formula de tiempo medio de inactividad.

Formula	Variable A	Variable B	Variable X
$X = T/A$ A-Número de fallos observados. T-Tiempo total de inactividad.	0	0	0

Una vez aplicado la formula y obtenido los resultados en la tabla 16 indican los parámetros para el cumplimiento de esta métrica lo cual el valor más cercano a $0/t$, es mejor. Y será el 100 por ciento de calidad externa, caso contrario si se aleja de los valores referidos existirá un porcentaje faltante de calidad.

Tabla 16: Resultados de tiempo medio de inactividad.

Métrica	Valor no deseado	Valor esperado	Valor obtenido	Porcentaje de calidad externa obtenido	Porcentaje de calidad externa faltante.
Tiempo medio de inactividad	$1/t$	$0/t$	0	100%	0%

De acuerdo con el valor final obtenido en la métrica, se obtuvo una calificación excelente, demostrando la efectividad del sistema ante las necesidades del cliente dando como resultado un correcto desempeño ante los fallos por las altas tasas de peticiones recibidas en la simulación.

7.3.3. Sub característica. Tolerancia a fallos

7.3.3.1. Métrica: Prevención de fallos

Para valorar la métrica prevención de fallas se tomó en cuenta las advertencias y errores que se encontraron, en la interfaz de cada uno de los casos de prueba: presentación de la interfaz en los navegadores las cuales se listan en la tabla 17.

Tabla 17: Prevención de fallos, datos.

Caso de prueba	Lista de errores encontrados	Descripción del error	Solución del error
Presentación de la interfaz en los navegadores.	Error “Failed to load resource: the server responded with a status of 404 (Not Found)”	Este error se presenta al momento de presentar la interfaz de los cuadros de mando en el navegador.	La solución al error fue reemplazar la librería para graficar los cuadros de mando por la más actual, debido a que la que se estaba usando estaba por finalizar el soporte por lo cual generaba el error.

Presentación de la interfaz en los navegadores.	Error “Uncaught (in promise) TypeError: Cannot read properties of null (reading 'length')”	El error se presenta al momento de presentar la interfaz de los cuadros de mando, al momento que se cargan datos nulos de la BD.	La solución a este problema fue restringir el envío de valores nulos, se planteó un control de al momento de graficar el cuadro de mando se envíen series que no contengan valores nulos.
---	--	--	---

Para el cálculo de los resultados se lo realizo mediante la fórmula de la prevención de fallos. La cual se muestra en la tabla 18.

Tabla 18: Formula de prevención de fallos.

Formula	Variable A	Variable B	Variable X
A=A/B A= Numero de Ocurrencia de fallas evitadas contra los casos de pruebas de fallas iniciales. B= Número de casos de pruebas de fallas iniciales Ejecutadas durante las pruebas. Dónde: B>0	2	2	1

Los datos que muestra la tabla 19 son los resultados, los cuales indican el valor esperado el cual debe ser aproximado a 1 para obtener un buen porcentaje de calidad caso contrario será 0.

Tabla 19: Resultados de prevención de fallos.

Métrica	Valor no deseado	Valor esperado	Valor obtenido	Porcentaje de calidad externa obtenido	Porcentaje de calidad externa faltante.
Prevención de fallos	0	1	1	100%	0%

De acuerdo con el valor final obtenido en la métrica, se obtuvo una calificación excelente, demostrando el correcto desempeño de las pruebas realizadas al sistema para evitar posibles fallos en caso de que se detenga algún proceso esencial para el funcionamiento del sistema.

7.3.3.2. Métrica: redundancia.

Para poder evaluar la métrica de redundancia se tomaron en cuenta los componentes vinculados directamente con el sistema, que ofrecen servicio de redundancia como lo son SaaS, que permiten mantener datos en la nube para que posteriormente se puedan respaldar en posibles fallos o errores en el sistema web.

Tabla 20: Redundancia, datos.

Componente vinculado con el sistema web	Acciones que realiza ante fallos o errores	Redundancia
GitHub: (SaaS)	Es un sistema el cual está encargado de realizar copias de seguridad del código fuente del sistema, con la finalidad de que ante posteriores errores se puedan restaurar los valores anteriores en el código fuente	Redundancia pasiva

Para la valoración final se aplicó la fórmula correspondiente de la redundancia, con la finalidad de obtener resultados

Tabla 21: Fórmula de redundancia.

Fórmula	Variable A	Variable B	Variable X
$X=A/B$ A= Número componentes/sistemas instalados de forma redundante.	1	1	1

Finalmente, en la tabla 22 se muestra los resultados deseados y obtenidos, así como los porcentajes de cumplimiento de calidad de la métrica, lo cual indica que mientras el valor sea más cercano a 1 es mejor.

Tabla 22: Resultados de redundancia.

Métrica	Valor no deseado	Valor esperado	Valor obtenido	Porcentaje de calidad externa obtenido	Porcentaje de calidad externa faltante.
Redundancia	0	1	1	100%	0%

De acuerdo con el valor final obtenido en la métrica, se obtuvo una calificación excelente, la métrica permite generar respaldos activos y pasivos según las necesidades para lo cual se tomó en cuenta un respaldo pasivo en caso de pérdida de los dispositivos físicos donde se encuentra almacenado y en funcionamiento el sistema.

7.3.4. Sub característica. Capacidad de recuperación

7.3.4.1. Métrica. tiempo medio de recuperación

Para determinar el porcentaje de tiempo medio de recuperación, se tomaron en cuenta la evaluación de la métrica redundancia, y una prueba de eficacia la misma que se realizó en pasos anteriores con el simulador “*Apache JMeter*” con la finalidad de obtener los resultados de la proporción de las tareas completadas correctamente mediante peticiones al sistema web, en el parámetro redundancia y eficacia se tiene un porcentaje del 100% de calidad, de la misma manera se tomaron en cuenta los valores de la prueba de capacidad de recuperación con el fin de tener en cuenta el tiempo de recuperación frente a posibles fallos. (ZÁRATE, 2016)

Para lograr dar un valor cuantitativo a este resultado se utilizó la formula del cálculo tiempo medio de recuperación como se muestra en la tabla 23.

Tabla 23: Formula de tiempo medio de recuperación.

Formula	Variable A	Variable B	Variable X
$X=A/T$ A=Número de casos en los cuales se ha observado que el sistema entro en recuperación. T =Tiempo que le tomo al sistema en recuperarse. Donde: $T>0$	0	0	1

Finalmente, en la tabla 24 se muestra el valor deseado, el valor obtenido y el porcentaje de calidad, siendo que al obtener un valor cercano a 0/t será una mejor calidad y viceversa será una menor calidad.

Tabla 24: Resultados de tiempo medio de recuperación.

Métrica	Valor no deseado	Valor esperado	Valor obtenido	Porcentaje de calidad externa obtenido	Porcentaje de calidad externa faltante.
Tiempo medio de recuperación	$\geq 10/t$	0	0	100%	0%

De acuerdo con el valor final obtenido en la métrica, se obtuvo una calificación excelente, demostrando el correcto desempeño del sistema el cual demuestra su calidad por medio de las métricas conglomeradas utilizadas para el cálculo del tiempo de recuperación,

concluyendo que el sistema posee la capacidad de recuperación en caso de alguna eventualidad negativa que afecte a los componentes externos en cual reside.

7.3.5. Resultados finales de evaluación de fiabilidad al sistema web

Tabla 25: Resultados de evaluación sub característica de Madurez.

Subcriterio	Métrica	Resultado no deseado	Resultado esperado	Resultado adquirido	Porcentaje de calidad obtenido	Porcentaje de calidad faltante
Madurez	Eliminación de fallos	0	1	0.86	86%	14%
	Cobertura de pruebas	0	1	1	100%	0%
Promedio alcanzado				1.86	93%	7%

Tabla 26: Resultados de evaluación sub característica de Disponibilidad.

Subcriterio	Métrica	Resultado no deseado	Resultado esperado	Resultado adquirido	Porcentaje de calidad obtenido	Porcentaje de calidad faltante
Disponibilidad	Tiempo de servicio	0	1	1	100%	0%
	Tiempo medio de inactividad	$\geq 10/t$	0/t	1	100%	0%
Promedio alcanzado				2	100%	0%

Tabla 27: Resultados de evaluación sub característica de Tolerancia a fallos.

Subcriterio	Métrica	Resultado no deseado	Resultado esperado	Resultado adquirido	Porcentaje de calidad obtenido	Porcentaje de calidad faltante
Tolerancia a fallos	Prevención de fallos	0	1	1	100%	0%
	Redundancia	0	1	1	100%	0%
Promedio alcanzado				2	100%	0%

Tabla 28: Resultados de evaluación sub característica de Capacidad de recuperación.

Subcriterio	Métrica	Resultado no deseado	Resultado esperado	Resultado adquirido	Porcentaje de calidad obtenido	Porcentaje de calidad faltante
Capacidad de recuperación	Tiempo de medio de recuperación	$\geq 10/t$	0/t	1	100%	0%
Promedio alcanzado				1	100%	0%

7.3.6. Resultados totales

Tabla 29: Resultados globales de la evaluación del criterio de fiabilidad ISO/IEC.

Criterio	Subcriterio	Resultado obtenido	Promedio resultado obtenido	Promedio de porcentaje de calidad	Grado de satisfacción	Nivel de aceptación
Fiabilidad	Madurez	9.3/10	1.83	18.3%	Muy satisfactorio	Nivel planificado
	Disponibilidad	10/10	4.5	45%	Muy satisfactorio	Nivel planificado
	Tolerancia fallos	a 10/10	2	20%	Muy satisfactorio	Nivel planificado
	Capacidad de recuperación	de 10/10	1.5	15%	Muy satisfactorio	Nivel planificado
Promedio alcanzado		9.83	9.83	98.3%	Muy satisfactorio	Nivel planificado

Como resultado de la evaluación del criterio de fiabilidad aplicando la norma ISO/IEC 25000, en concordancia con la tabla 29 donde se muestran los resultados detallados con un promedio de 98,3% de calidad en el sistema web de alertas tempranas fitosanitarias, un puntaje de 9.83; estos resultados son calculados a partir de las fórmulas y métricas mencionadas en las tablas 25, 26, 27 y 28 de métricas de calidad interna y externa de la característica de fiabilidad ISO/IEC y valorados en la escala con los parámetros tomados de la norma ISO/IEC 25040, en la cual muestra que el resultado obtenido de la evaluación es muy satisfactorio, así mismo el nivel de aceptación está en un nivel planificado de desarrollo del sistema detallado por parte del cliente.

7.4. Comprobación de hipótesis

Con efecto de visibilizar las competencias más robustas en las metodologías estudiadas (XP, Scrum, Crystal, Methodologies, FDD, ASD) se contabilizó 15 requisitos de funcionalidad para evaluar el cumplimiento de la gestión fitosanitaria, el resumen se presenta en la tabla siguiente:

Tabla 30: Resultados de metodologías analizadas.

Metodologías	Cantidad de requisitos cumplidos	Porcentaje de cumplimiento funcional	Gestión
XP	14	93,00	1
Scrum	10	67,00	0
Crystal			
methodologies	8	53,00	0
FDD	9	60,00	0
ASD	11	73,00	0

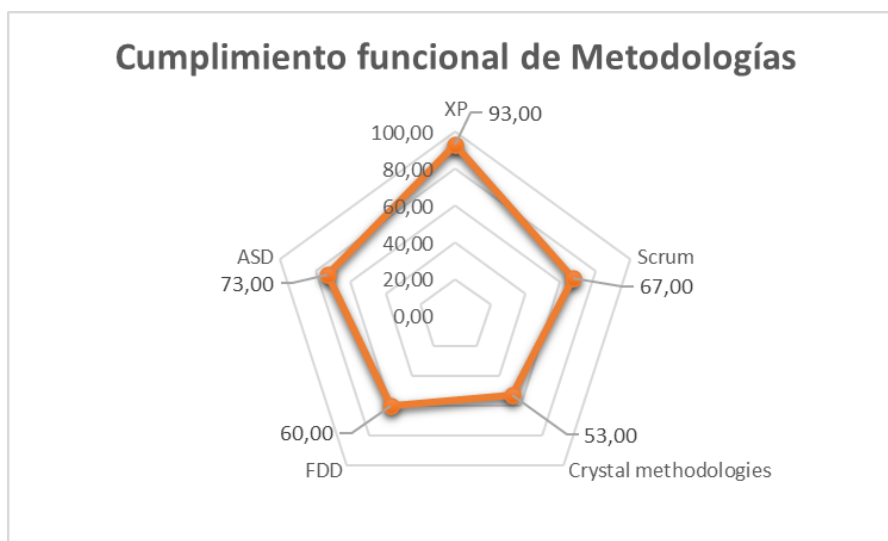


Figura 36: Grafica de radar. Cumplimiento funcional de metodologías

El gráfico radial evidenció el porcentaje de cumplimiento más alto de las metodologías en función de los requisitos analizados, es así como la metodología XP ocupa el primer lugar con un 93,33% de cumplimiento, seguido por la metodología ASD con un 73%, el último lugar lo ocupa la metodología Crystal Methodologies con un 53%; sin embargo, al tomar como referencia los valores de los clientes de acuerdo a los SRS se decidió utilizar la metodología XP que cumple con el requisito de superar el 85% de fiabilidad, para desarrollar un estudio de la fiabilidad del sistema SE tomó como referencia el nivel de madurez, nivel de disponibilidad y capacidad de recuperación.

Para la selección de la técnica estadística se desarrollaron pruebas de normalidad de Shapiro Wilk al 5% de significancia para las variables nivel de madurez, nivel de disponibilidad y capacidad de recuperación bajo las siguientes hipótesis:

H_0 : Las variables X, Y, Z se ajustan a una distribución normal

H_1 : Las variables X, Y, Z no se ajustan a una distribución normal

X: nivel de madurez, Y: nivel de disponibilidad y Z: capacidad de recuperación

Tabla 31: Pruebas de normalidad

Fiabilidad	Shapiro-Wilk		
	Estadístico	gl	Sig.
Nivel de madurez	,705	30	0,200
Nivel de disponibilidad	,965	30	0,419
Capacidad de recuperación	,750	30	0,312

Los valores de probabilidad evidenciaron que las variables se ajustaron a una distribución normal lo que permite ejecutar pruebas paramétricas de proporción para una muestra al 5%

de significancia con el fin de validar que los componentes de la fiabilidad superan el 85% de cumplimiento bajo las siguientes hipótesis

$$H_0: p_{fiabilidad} \leq 0.85$$

$$H_1: p_{fiabilidad} > 0.85$$

Tabla 32: Pruebas de Hipótesis para una proporción

Fiabilidad	Número de ensayos	Estadístico	Sig.
Nivel de madurez	30	2,7	0,011
Nivel de disponibilidad	30	2,3	0,048
Capacidad de recuperación	30	2,0	0,008

Los valores de probabilidad destacaron que los componentes de fiabilidad superaron el 85% garantizando el funcionamiento adecuado de las aplicaciones ejecutadas con la metodología XP.

8. DISCUSIÓN

Por medio del presente estudio se logró sintetizar información con respecto al desarrollo de una aplicación basada en los SIG, específicamente el tipo Web Mapping; el uso de las metodologías ágiles permiten al desarrollador un trabajo continuo a pesar de los inconvenientes, demostrando el acoplamiento de los requerimientos y a la naturaleza del producto. A través del análisis de los sistemas de información geográfica, así como la respectiva recopilación bibliográfica de las metodologías de desarrollo permitieron a los investigadores elegir *Extreme Programming* por medio de sus cuatro fases finalizando por completo con un resultado positivo, gracias a la aprobación del encargado de verificar el estado final del software.

Posterior a la finalización del software, de acuerdo con la norma ISO/IEC 25000, se demostró la calidad del software tomando en cuenta los resultados totales donde se obtuvo un promedio del 98.3% dando un grado muy satisfactorio y un nivel de aceptación acorde al nivel planificado. Este tipo de evaluación difiere con las demás gracias a que se toman aspectos de madurez, disponibilidad, tolerancia a fallos y capacidad de recuperación; permiten describir la capacidad del sistema para desempeñar sus funciones bajo parámetros específicos de tiempo y condiciones normales como anómalas.

En comparación con otros trabajos relacionados, que centran los esfuerzos en buscar la mejor usabilidad, la aplicación de las normas mencionadas permiten establecer la relación que existe en escoger la mejor metodología adaptada a las necesidades específicas del proyecto y la naturaleza de los sistemas de información geográfico; respaldado por la comprobación de la hipótesis donde se evalúa la fiabilidad con todos sus aspectos para llegar a la conclusión que el sistema cumple con la normativa internacional en gestión de servicios de TI.

CAPÍTULO V. CONCLUSIONES y RECOMENDACIONES

CONCLUSIONES

- Se seleccionó la metodología *Extreme Programming* (XP), ya que permite acoplar la estructura básica de componentes a los sistemas de información geográfica dando solución inmediata a las necesidades del cliente para la gestión de alertas tempranas fitosanitarias.
- El desarrollo del web Mapping por medio de la metodología XP permite el uso de librerías como *OpenLayers*, *TensorFlow* y *Gmail API* logrando desarrollar los módulos de administración, alertas, estaciones y gráficos obteniendo como resultado un sistema funcional y fiable para la gestión de las alertas tempranas fitosanitarias.
- El sistema de información geográfica de alertas fitosanitarias de acuerdo con la norma ISO 25000, es fiable en un 98.3%, basado en los resultados del criterio madurez se obtuvo un 93% de calidad, en la disponibilidad un 100%, la tolerancia de fallos un 100%, en la capacidad de recuperación se obtuvo un 100%. El restante 1,7% se tomarán en cuenta para futuras actualizaciones y mejoras en el sistema.
- En base a la investigación se puede afirmar que es factible el desarrollo de sistemas Web Mapping los cuales pueden ser aplicables no solo a la gestión de alertas fitosanitarias sino en otros ámbitos como prevención de riesgos, salud, seguridad y estudios de impacto ambiental, logrando adaptarse a cualquier entorno.

RECOMENDACIONES

- La metodología de desarrollo debe ser adaptada a las necesidades específicas de cada proyecto de software, teniendo en cuenta los antecedentes y las especificaciones del cliente que van generalmente cambiando sobre la marcha, por lo cual, es recomendado socializar los detalles del proyecto general y de manera específica los por menores durante todas las etapas del proceso.
- Se debe seguir los pasos establecidos en la metodología de desarrollo, sin importar la seleccionada, completando todos los componentes o fases que permitan obtener un producto final de calidad.
- En el proceso de desarrollo de software todos los productos deben ser evaluados con el criterio de fiabilidad de la norma ISO/IEC 25000, puesto que esta métrica cuantifica las condiciones y periodos de tiempo, capacidad de recuperación, prevención de fallos en la que el software se desempeña, logrando así determinar los errores y por ende minimizarlos para tener un sistema de mejor calidad.

BIBLIOGRAFÍA

- Brovelli, M. A. (2012). Free and open source desktop and Web GIS solutions. *Applied Geomatics*, 65-66.
- Bryan, M. M., Harry, V. C., & Jefferson, D. C. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales*, 9.
- Ceupe Magazine. (2018, 10 31). *Ceupe Magazine*. Retrieved from <https://www.ceupe.com/blog/los-sistemas-de-informacion-geografica.html>
- Dorman, M. (2021). Introduction to Web Mapping. *Ben-Gurion University of th Negev*, 1-9.
- Esri. (2022). *ArcGis Resources*. Retrieved from <https://resources.arcgis.com/es/help/getting-started/articles/026n0000000t000000.htm>
- Farkas, G. (2007). Applicability of open-source web mapping libraries for building massive Web GIS clients. *SpringerLink*, 273–295.
- Garzas, J. (2012, Septiembre 25). *Javiergarzas.com*. Retrieved from <https://www.javiergarzas.com/2012/09/metodologias-crystal.html>
- Guevara, J. A. (1992). Esquema metodológico para el diseño e implementación de un Sistema de Información Geográfico. In I. F. Diputación de Zaragoza, *V Coloquio de Geografía Cuantitativa* (pp. 21-32). Zaragoza: Dialnet.
- INEC. (2020). *Tabulados de la Encuesta de Superficie y Producción Agropecuaria Continua ESPAC 2020*. Quito: INEC.
- ISO/IEC. (2022). *ISO 25000*. Retrieved from <https://iso25000.com/index.php/normas-iso-25000>
- Joskowicz, J. (2008). *Reglas y prácticas en Extreme Programming*. Vigo: Universidad de Vigo.
- Juan Luis, V. G. (2016, Julio 8). *Proagilist*. Retrieved from <https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-metodologia-xp/>
- Lara, E. L., Simeón, C. P., & Navarro, J. G. (2006). LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA. *I Congreso de Ciencia Regional de Andalucía: Andalucía en el umbral del siglo XXI* (p. 16). Geoenseñanza.
- Letelier, P. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Técnica Administrativa*, 13.
- M, R. (1995). *Sistemas de información geográfica y análisis espacial, en Prácticas de Análisis espacial*. Barcelona: Oikostau.
- Maida, E. G., & Paziencia, J. (2015). *Metodologías de desarrollo de software*.
- Martinez, J. (2014, Octubre 27). *Slideshare*. Retrieved from <https://es.slideshare.net/jasonjosemartinezgarcia1/metodologa-crystal-clear>
- Molina Ríos, J. R. (2017). Metodologías de desarrollo web. *3c Tecnología*, 18.
- Molina, B. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales*, 114-121.
- Novoa, G., & Byron, G. (2018). *DESARROLLO DE UNA ESTACIÓN AGRO-METEOROLÓGICA AUTOMÁTICA REMOTA PARA EL LEVANTAMIENTO DE*

- INFORMACIÓN CLIMÁTICA EN LA CUENCA DEL RÍO PISQUE*. Quito: Universidad Politécnica Salesiana.
- Picon, M., & Vazquez, P. (2022). *Impacto de la precipitación pluvial en la turbidez del agua superficial usada para la potabilización en el cantón Azogues durante los años 2014 – 2020*. Cuenca: UCUENCA.
- Pilaloa David, W. A. (2021). Manejo agroecológico de la Moniliasis en el cultivo de cacao (*Theobroma cacao*) mediante la utilización de biofungicidas y podas fitosanitarias en el cantón La Troncal. *Revista Alfa*, 15.
- Pressman, R. S. (2013). *Ingeniería de Software un enfoque práctico*. Journal of Chemical Information and Modeling.
- Rivas, C. I., Corona, V. P., & Gutiérrez, J. F. (2015). Metodologías actuales de desarrollo de software. *Tecnología e Innovación*, 7.
- Rivera, I. (2020, Enero 17). *Ivan Rivera PMP*. Retrieved from <https://ivanrivera-pmp.blogspot.com/2020/01/en-que-consiste-el-desarrollo-dirigido.html>
- Ruiz, M. (2006). *Sistemas de información geográfica y análisis espacial, en Prácticas de Análisis espacial*. Barcelona: Oikos-Tau.
- Salazar, A. (2016, Octubre 16). *PROZESS GROU*. Retrieved from <http://www.prozessgroup.com/procesos-de-scrum/>
- Sandra Milena, V. R., Juan David, V. M., Ana Alexandra, P. M., Erika Julieta, R. Z., & Sebastián, L. M. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software . *Cintex*, 11.
- Santander Universidades. (2021, 10 26). *Becas Santander*. Retrieved from <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- Tecnologías Informacion. (2020). *Tecnologías Informacion*. Retrieved from <https://www.tecnologias-informacion.com/metodologia-funciones.html>
- Zamora, E. (2014, Marzo). *Files wordpress*. Retrieved from <https://darjelingsilva.files.wordpress.com/2018/05/1-metd-asd.pdf>
- ZÁRATE, R. F. (2016). *Core*. Retrieved from <https://core.ac.uk/download/pdf/323348271.pdf>

ANEXOS

Manual técnico.

Universidad Nacional de Chimborazo

Anexo – Manual técnico

SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA GESTIÓN DE
ALERTAS TEMPRANAS FITOSANITARIAS EN EL CANTÓN
FRANCISCO DE ORELLANA

Fecha: 22-08-2022

Página 1

Figura 37: Anexo manual técnico

Manual de usuario

Universidad Nacional de Chimborazo

Manual de usuario

SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA GESTIÓN DE
ALERTAS TEMPRANAS FITOSANITARIAS EN EL CANTÓN
FRANCISCO DE ORELLANA

Fecha: 11-03-2022

Página 1

Figura 38: Anexo manual de usuario

Manual de pruebas

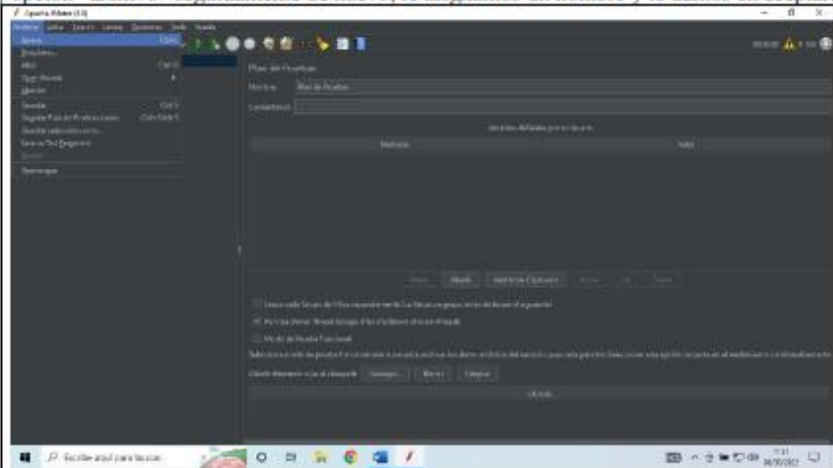
APACHE JMETER

Fecha: 06-08-2022

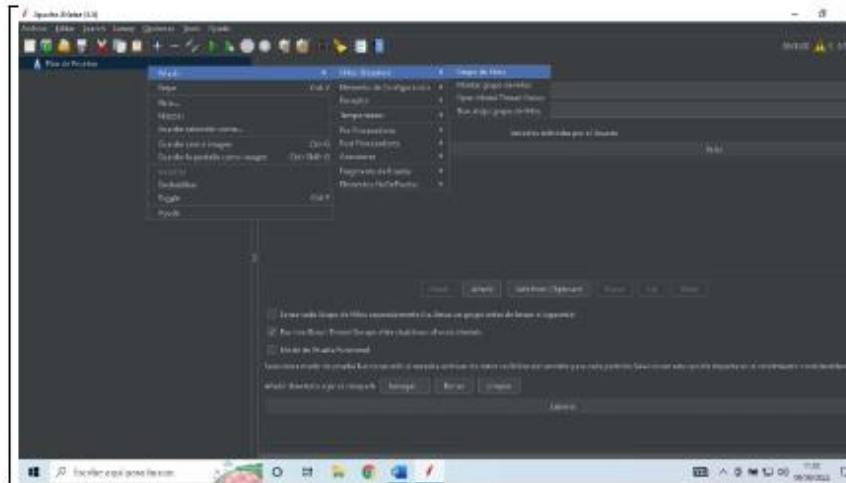
1. ACCIONES PASO A PASO PARA REALIZAR PRUEBAS EN APACHE JMETER

Apache JMeter es un software de código abierto, una aplicación diseñada totalmente en JAVA para medir el rendimiento y comportamiento de servidores mediante pruebas. Originalmente se diseñó para probar aplicaciones Web, pero se ha ampliado desde entonces a otras funciones. En el presente documento encontraras acciones rápidas las cuales te permitirán ejecutar pruebas de estrés y rendimiento.

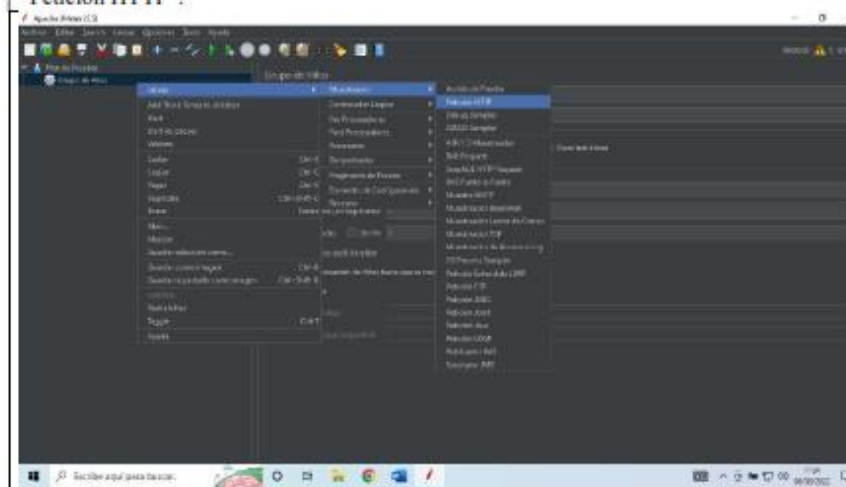
Para empezar con las pruebas debemos ejecutar nuestro simulador "Jmeter", el cual presenta una interfaz de usuario intuitiva y fácil de usar. Primeramente, debemos crear un nuevo plan de pruebas para el cual presionamos en la opción "archivo" seguidamente de nuevo, le asignamos un nombre y le damos en aceptar.



Seguidamente debemos agregar los componentes para las pruebas en este caso vamos a agregar. Un grupo de hilos por lo que nos posicionamos en nuestro plan de pruebas, damos clic alterno y seleccionamos la opción "añadir" seguido de "hilos" y un grupo de hilos.

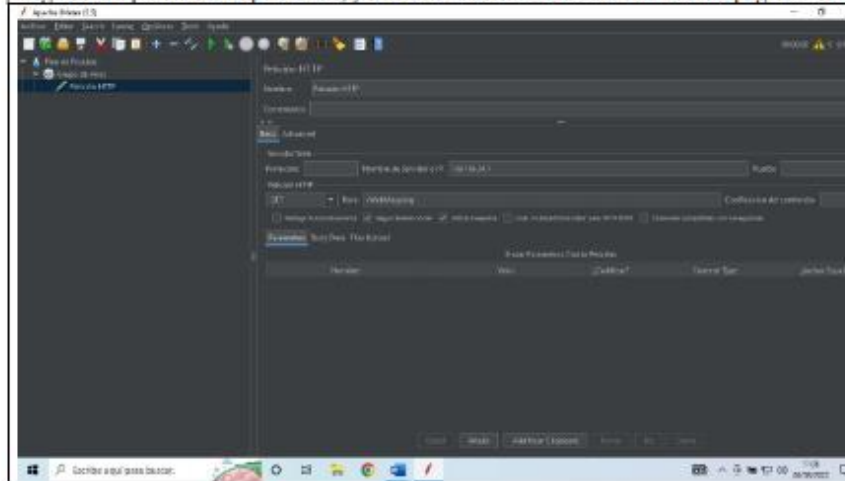


Seguidamente debemos agregar el componente de peticiones HTTP. Para los cual encima del grupo de hilos que añadimos damos clic alterno y luego en: “añadir”, “muestreador”, “Petición HTTP”.

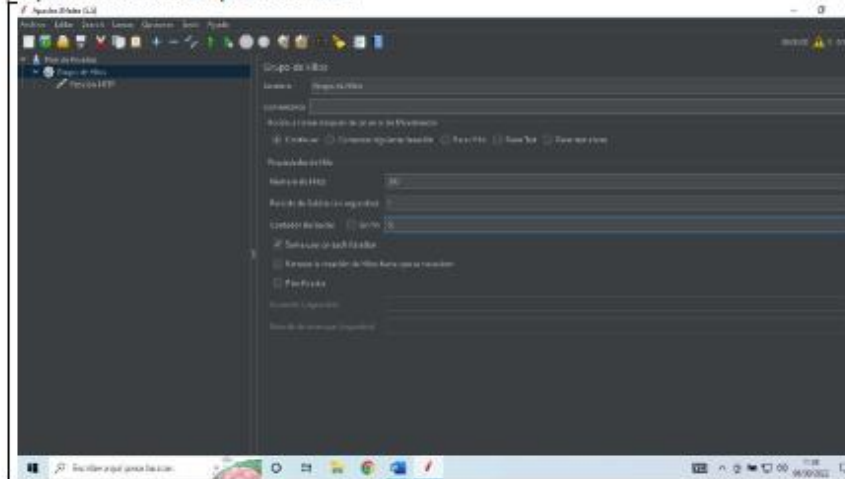


Dentro de este componente debemos configurar el ip del servidor o el dominio para lo cual debemos llenar las opciones como muestra en la siguiente captura. En el servidor o ip

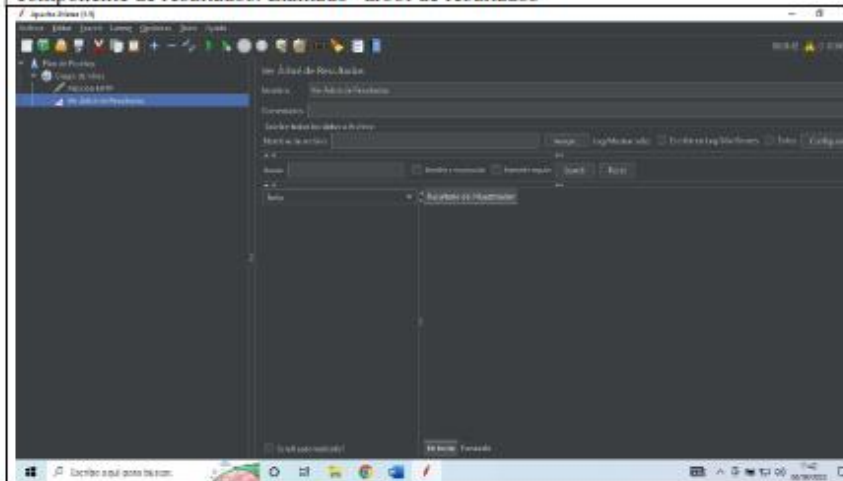
colocamos el ip de nuestro servidor donde se encuentra alojada nuestra pagina web. Seguidamente del puerto correspondiente, y finalmente la ruta en la cual esta nuestra pagina web.



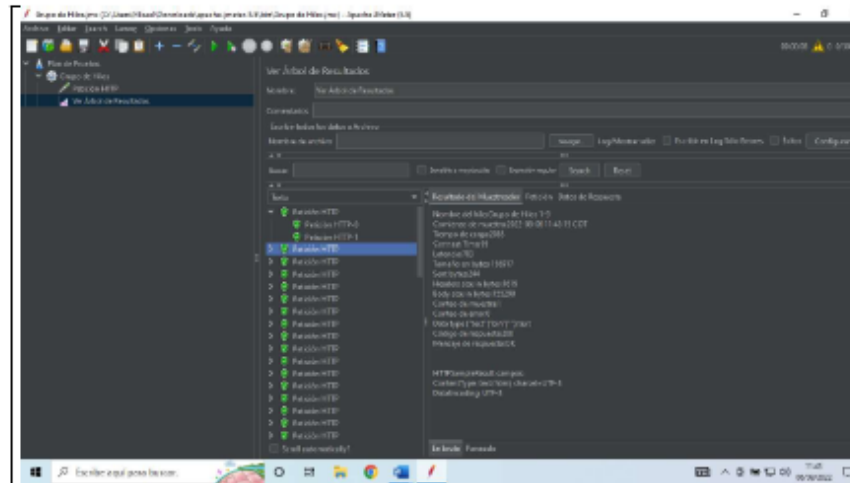
Seguidamente debemos configurar el componente de grupo de hilos, para lo cual damos clic en el grupo de hilos y debemos configurar de acuerdo a las necesidades, el numero de hilos es el número de repeticiones de petición, el periodo de subida es el tiempo de cada cuanto se va realizar una petición al servidor, el contador de bucle es la cantidad de repeticiones de hilos que va realizar.



Una vez configurado debemos agregar un componente en el cual nos muestre los resultados de las pruebas para los añadimos de la misma manera anterior solo que esta vez un componente de resultados. Llamado "árbol de resultados"



Seguidamente ejecutamos las pruebas y podemos ver los resultados en nuestro árbol de resultados.



Dentro del simulador podemos realizar múltiples casos de pruebas como:

- Web HTTP y HTTPS.
- SOAP.
- Base de datos a través de JDBC.
- LDAP.