agrgar portada

Contenido

[1. CONTEXTUALIZACIÓN DEL NEGOCIO 2](#_Toc41129842)

[2. JUSTIFICACIÓN 3](#_Toc41129843)

[3. OBJETIVO GENERAL 3](#_Toc41129844)

[4. OBJETIVO ESPECIFICOS 3](#_Toc41129845)

[5. MODELO DE CALIDAD 3](#_Toc41129846)

[6. ESTÁNDARES DE CALIDAD 4](#_Toc41129847)

[7. DIFERENCIAS ENTRE MODELO DE CALIDAD Y ESTÁNDARES DE CALIDAD 5](#_Toc41129848)

[8. DIAGRaMA DE CASOS DE USO 5](#_Toc41129849)

[6](#_Toc41129850)

[9. CASOS DE USO DETALLADOS 6](#_Toc41129851)

[10. ESCENARIOS A EVALUAR 10](#_Toc41129852)

[Plan de pruebas 13](#_Toc41129853)

# CONTEXTUALIZACIÓN DEL NEGOCIO

QGIS es un sistema de información geográfica este software es de código abierto y soporta numerosos formatos de datos como: datos de vectores, raster y bases de datos. (*Descubre QGIS*, s. f.)

Una de las funcionalidades principales de QGIS es generar mapas en formato GEOJSON el cual es el que se implementara en el desarrollo de nuestro proyecto de grado.

El formato GEOJSON se caracteriza por tener un objeto features donde se obtienen las diferentes entidades y a su vez geometry y propeties siendo la última que contiene los atributos

Se realizara desarrollo de esta aplicación web utilizando Lealeft el cual se convertirá en formato GEO JSON , este formato de datos está siendo muy popular por muchas tecnologías de servicios GIS, QGIS, ya que es muy ligero, simple y rápido para almacenar en el servidor web.

(Serrano, 2018)

Las pruebas de software se realizaran los estándares de calidad en la norma IEEE 829 y la ISO25010.

# JUSTIFICACIÓN

El proyecto nace con el fin de facilitar la interaccion con los mapas en formato GeoJson que generan los aplicativos de información geográfica, realizándose una investigación respecto a los dos Framework más importantes y utilizados hoy en día por los desarrolladores (Angular vs React) ya que estos lenguaje se ajusta plenamente a lo que se quiere lograr.(*Angular vs React | OpenWebinars*, s. f.)

Agregar contenido

# OBJETIVO GENERAL

* Desarrollar una aplicación que facilite la interacción con mapas en formatos geojson

# OBJETIVO ESPECIFICOS

* Cargar un mapa en formato GeoJson para visualizarlo dentro del sitio web.
* Integrar algunas herramientas de interacción con el mapa en el sitio web
* Evaluar la aplicación atreves de estándares de calidad que garanticen el producto obtenido

# **MODELO DE CALIDAD**

Es un conjunto de actividades que permite planificar y verificar la calidad de un producto obteniendo características principales, también son procedimientos de los estándares para el diseño y análisis de la programación y pruebas de software organizar” para así lograr la mayor confiabilidad ante el producto se tienen en cuenta la calidad interna es muy importante porque es la revisión del código y el análisis del comportamiento del producto.

A continuación se presentan algunos modelos de calidad

* **El modelo Goal Question Metric (GQM)** propuesto por Rombach(1994) este modelo proporciona resultados en cuanto a los objetivos del producto, es diseñado mediante preguntas que son potencialmente medibles. Permite evaluar la calidad de cada proyecto mediante las métricas y analizando los datos **realizar una lista** delos objetivos principales del desarrollo del proyecto y así trazar un camino de solución mediante el desarrollo de preguntas (Scalone, 2006, P129)
* **El modelo funcionabilidad, usabilidad confiabilidad, performance, facilidad de soporte (FURPS**)

Propuesto por Heweltt (1994) permite realizar diferente pruebas de confiabilidad, funcionalidad este modelo analiza las variables durante el ciclo de vida del producto, esta compuestos por fases y permite obtener la optimización en el uso de recursos (Scalone, 2006, P137)

# ESTÁNDARES DE CALIDAD

Los estándares de calidad son los criterios específicos a evaluar en el modelo, es decir principalmente puntos esenciales de la calidad de software. Suministran un conjunto de actividades para que todos los procesos se realicen de la mejor manera “Estos estándares verifican con más fuerza la confiabilidad del producto. Tienen como objetivo principal que las empresas desarrollen de forma ágil y comprensible pasando por etapas de pruebas. Estos desempeñan una mejor gestión de actividades para tener un seguimiento de procesos de software.

Los estándares permiten trabajar por fases de madurez al producto para así poder trabajar paso a paso y minuciosamente al momento de realizar pruebas de software.

(Scalone, 2006, P27)

A continuación se mostraran los tipos de estándares de calidad

* **ISO/9001 (2004)** estándar que promueve los procesos, la adopción de un enfoque de mejora y eficacia.

Esta ISO tiene un sistema de procesos individuales así como la combinación de las iteraciones, este sistema tiene como objetivo, la importancia, compresión y cumplimiento de los requisitos con eficacia del proceso. (Scalone, 2006, P155)

* **ISO/12207 (1995)** este estándar hace referencia al ciclo de vida del software a la implementación de actividades y tareas en los procesos se permiten agrupar esta serie de procesos principales para así evaluar el comportamiento del software y obtener resultados. (Scalone, 2006, P157)

# DIFERENCIAS ENTRE MODELO DE CALIDAD Y ESTÁNDARES DE CALIDAD

* **MODELO**: Es una serie de buenas prácticas que se utilizaran para evaluar la estructura del producto con métricas establecidas, teniendo el modelo escogido y así siguiendo los lineamientos obtener el producto eficaz y de buen rendimiento.
* **ESTÁNDAR:** Es una serie de procesos del ciclo de vida del producto teniendo en cuenta el estándar escogido, se analizan muy detenidamente, el comportamiento y madurez del producto final.

# 

# DIAGRaMA DE CASOS DE USO

# C:\Users\CLAYDER E CUERO Q\Downloads\Diagrama en blanco - Página 1 .jpeg

# CASOS DE USO DETALLADOS

**CASOS DE USO 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Cargar mapa | | | | | CU1 |
| Actores | Administrador ,Creador, Visualizador | | | | | |
| Referencias | Requerimiento RF 1 | | |  | | |
| Precondición | El usuario debe haber iniciado sesión en el sistema | | | | | |
| Postcondición | El sistema debe permitir cargar mapas en formato Geo-Json .en leaflef | | | | | |
| Autor | Clayder Cuero | Fecha |  | | Versión |  |

|  |
| --- |
| Propósito |
| Se debe permitir cargar mapas GeoJson |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| acción del usuario | | | acción del sistema | |
| 1 | El usuario debe dar clic en el botón cargar mapa |  | **2** | El sistema debe permitir seleccionar el archivo GeoJson |
| 3 | El usuario selecciona un archivo |  | **4** | El sistema debe cargar el mapa Geojson |

|  |  |
| --- | --- |
| Cursos Alternos | |
|  | En caso de no obtener todos los permisos, el sistema solo podrá tener habilitada la opción de cargar mapa y visualizar mapa |
| 4 | El sistema debe permitir solo cargar archivos geo.Json , en caso que los datos no sean válidos mostrar un mensaje en pantalla especificando el error que se presento |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Consultar Mapa | | | | | CU2 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF | | |  | | |
| Precondición | El usuario debe haber cargado al menos 1 mapa en formato Geojson | | | | | |
| Postcondición | El sistema debe permitir consultar mapas en formato Geo-Json | | | | | |
| Autor | Clayder Cuero | Fecha |  | | Versión |  |

|  |
| --- |
| Propósito |
| Se debe permitir consultar mapas GeoJson |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| acción del usuario | | | acción del sistema | |
| 1 | El usuario debe dar clic en el botón consultar |  | **2**  **3** | El sistema debe permitir escribir el nombre del formato GeoJson  El sistema debe mostrar la lista de archivos encontrados |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso De Uso | Editar Mapa | | | | | CU3 |
| Actores | Administrador | | | | | |
| Referencias | Requerimiento RF 3 | | |  | | |
| Precondición | El usuario debe haber cargado al menos 1 mapa en formato Geojson | | | | | |
| Postcondición | El sistema debe permitir consultar mapas en formato Geo-Json | | | | | |
| Autor | Clayder Cuero | Fecha |  | | Versión |  |

|  |
| --- |
| Propósito |
| Se debe permitir editar mapas GeoJson |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| acción del usuario | | | acción del sistema | |
| 1 | El usuario debe dar clic en el botón editar |  | **2**  **3** | El sistema debe permitir seleccionar un archivo      El sistema debe permitir editar los campos de el archivo seleccionado |
|  |  |  |  |  |

# ESCENARIOS A EVALUAR

|  |  |
| --- | --- |
| **Escenario #1** | Descripción    El sistema no debe dar acceso a un usuario no registrado |
| **Afecta:**  Usuario final | Seguridad |
| **Origen del estimulo** | Intentar acceder a la información |
| **Estimulo** | Usuario no autorizado |
| **Artefacto** | usuario |
| **Entorno** | sistema |
| **Respuesta** | El sistema mostrara un mensaje indicando que se debe comunicar con el administrador |
| **Tiempo de respuesta** | inmediatamente |

|  |  |
| --- | --- |
| **Escenario #2** | Descripción  El sistema debe permitir usuario debe ingresar |
| **Afecta:**  **Usuario final** | usabilidad |
| **Origen del estimulo** | Sistema |
| **Estimulo** | Ingresar al sistema |
| **Artefacto** | Capa de presentación |
| **Entorno** | Condiciones normales |
| **Respuesta** | Cargar el ingreso al sistema |
| **Tiempo de respuesta** | inmediatamente |

|  |  |
| --- | --- |
| **Escenario #3** | Descripción    El sistema debe permitir que se cargue archivos en formato.Json |
| **Afecta:**  **Usuario final** | adaptabilidad |
| **Origen del estimulo** | Cargar archivos |
| **Estimulo** | Sistema Cargar formatos |
| **Artefacto** |  |
| **Entorno** | Condiciones normales |
| **Respuesta** | Archivo cargado correctamente |
| **Tiempo de respuesta** | inmediatamente |

|  |  |
| --- | --- |
| **Escenario #4** | Descripción    El sistema debe permitir que al ingresar al sistema un usuario no administrador se deshabiliten el botón editar, eliminar y consultar |
| **Afecta:**  **Usuario final** | Seguridad |
| **Origen del estimulo** | Sistema |
| **Estimulo** | Sistema |
| **Artefacto** |  |
| **Entorno** | Condiciones normales |
| **Respuesta** | Rol creador |
| **Tiempo de respuesta** | inmediatamente |

## 

## PLAN DE PRUEBAS

* + - Definición del alcance
    - Definición de los recursos
    - Criterios de calidad
    - Definición de ítems a probar
    - Tipos de pruebas escogidas para validar los criterios
    - Riesgos

**ALCANCE**

En este aplicativo se requiere validar que se cumplan los requerimientos desarrollados con la finalidad de que el aplicativo sea totalmente funcional y útil, cumpliendo con los estándares de calidad, se realizara una validación concreta del plan de pruebas, pruebas de stress y pruebas unitarias.

En este proyecto se especifica los pasos detallados para lograr el alcance en las pruebas y los recursos utilizados en cada etapa de prueba, ítems a probar, herramientas que se utilizaran en cada prueba y los riesgos.

**DEFINICIÓN DE LOS RECURSOS**

* Se nombraran las herramienta para el uso de pruebas unitarias

|  |  |  |
| --- | --- | --- |
| HERRAMIENTAS DE PRUEBA | TIPO DE PRUEBA | DESCRIPCION |
| Visual Studio | Pruebas de código | Identificar e interactuar con objetos gráficos |
| Visual Studio | Pruebas de código | Interfaz web |
| Visual Studio | Pruebas de código | Pruebas unitarias |
| Jmeter | Pruebas de Stress | Prueba de stress |
| Visual Studio Debugging | Análisis de errores de código | Recorrido de código y análisis de errores |

* **PostgreSQL**

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

* **Angular**

Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles. (Devs, 2019)

* **Visual Studio Code**

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

* **Leaflet**:es una biblioteca de JavaScript de codigo abierto líder para mapas interactivos aptos para dispositivos móviles (*Leaflet — an open-source JavaScript library for interactive maps*, s. f.)

#### **CRITERIOS DE CALIDAD**

Para el plan de pruebas de este proyecto se tomarán algunos criterios de calidad según la norma ISO 25010 y los estadares de calidad de la norma IEEE 829

* **Adecuación Funcional:** Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades cuando se usa en condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
* Completitud funcional. Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
* Corrección funcional. Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
* Pertinencia funcional. Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

Para la adecuación funcional se realizarán las siguientes pruebas:

* Pruebas Unitarias
* Casos de uso

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

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

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

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

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

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

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

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

Para la elaboración de pruebas de usabilidad se realizaron formatos para usuarios donde se mostrarán los resultados de la revisión y el refinamiento del trabajo realizado

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

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

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

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

* Casos de uso
* Pruebas Unitarias

#### **DEFINICION DE ITEMS A PROBAR**

Los ítems para probar se dividen en varios módulos:

* + **Interfaz gráfica:** Se tendrá en cuenta cada una de las vistas del aplicativo.
* **Pruebas de usabilidad:** Se prueba el ítem mediante encuestas realizadas a diferentes usuarios ajenos al software desarrollado.
* **Pruebas Eficiencia:** Se prueba el comportamiento del aplicativo y los tiempos de respuesta a las peticiones realizadas por los usuarios.
  + **lógica del aplicativo:**
* **Pruebas Unitarias:** Se prueban métodos de clases, por ejemplo: métodos get/set, se realizarán pruebas en las líneas de código validando que el método codificado reciba y entregue valores correctos e incorrectos.
  + **base de datos:**
* **Pruebas de consultas masivas**: Se realizan pruebas de carga donde se simula la conexión de múltiples usuarios consultando la misma base de datos.
* **Pruebas Eficiencia:** Se prueba el comportamiento del aplicativo y los tiempos de respuesta a las peticiones realizadas por los usuarios.
* **Login (Seguridad):** Se probará la validación de los datos, autenticación de los usuarios y encriptación de contraseñas en base de datos, se evitará que los datos personales como contraseñas no se ingresen en texto plano.