

UNIVERSIDAD POLITECNICA SALESIANA

SEDE QUITO

CARRERA:

COMPUTACIÓN

Trabajo de titulación previo a la obtención del título de:

Ingeniero en Ciencias de la Computación

TEMA:

**ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN SISTEMA
WEB PARA LA GESTIÓN DE DATOS METEOROLÓGICOS MEDIANTE LA
UTILIZACIÓN DE UN API.**

AUTOR:

KEN STEVEN CONSTANTE JAUREGUI

TUTOR:

DANIEL GIOVANNY DÍAZ ORTIZ

Quito, noviembre del 2021

CESIÓN DE DERECHOS DE AUTOR

Yo, Ken Steven Constante Jauregui con documento de identificación N° 1725724437, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado:

ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN DE DATOS METEOROLÓGICOS MEDIANTE LA UTILIZACIÓN DE UN API. Mismo que ha sido desarrollado para optar el título de: **INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en Ley de la Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
Ken Steven Constante Jauregui

1725724437

Quito, 02 de noviembre del 2021

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Artículo académico, con el tema: ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN DE DATOS METEOROLÓGICOS MEDIANTE LA UTILIZACIÓN DE UN API, realizado por Ken Steven Constante Jauregui obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación

Quito, 02 de noviembre del 2021



Daniel Giovanni Díaz Ortiz

CI: 171697550-1

Dedicatoria

Esta tesis está dedicada a mi madre y a mi hermano quien me dieron toda la guía y pautas para poder ser la persona en la que me he convertido. Por ser los principales soportes durante mi carrera, por todo su apoyo y enseñanzas para seguir a delante y poder concluir mis estudios a cabalidad.

A mi madre quien siempre estuvo dándome aliento para poder seguir a delante a pesar de todas las adversidades, por ser la persona con la que siempre puedo contar sin importar que sea, dándome todo su ayuda y haciendo lo inalcanzable por mí.

A mi hermano quien siempre fue un modelo a seguir por darme los consejos los cuales siempre me encaminaron a cumplir mis metas, fijar mis objetivos, por brindarme su apoyo cuando lo necesitaba por ser como un padre para mí.

Agradecimiento

Quiero expresar mi agradecimiento a:

A mi madre Leonor Jauregui la cual me enseñó a nunca rendirme me demostró como una persona lucha por sus seres queridos, como salir siempre a delante, agradezco por poder contar con ella siempre y que me brinde ese amor de madre el cual es el sentimiento más puro. Agradezco por el apoyo incondicional que siempre me ha demostrado. Gracias Mami, por ser una mujer valiente, decidida, invencible ante la vida y luchadora.

A mi hermano Giovanni Herrera por darme esa guía y ser como mi padre y tomar este papel en cada ocasión que se necesitaba, por siempre estar en los momentos difíciles y complejos demostrando su apoyo incondicional, por darme su guía cada vez que la necesite, por ser un ejemplo para seguir ganarse mi admiración desde que tengo uso de razón.

A mi hermana Cynthia Constante la cual siempre confió en mí y siempre fue un ejemplo de superación personal logrando cumplir sus sueños y metas de a poco sin importar las adversidades que se le presentaron en el camino, por la confianza que me brindo y esperando que cada día siga superándome.

A Vanessa Meneses en la cual encontré un apoyo incondicional brindándome su ayuda sin importar el momento ni las circunstancias sin esperar nada a cambio. Por ser la persona la cual confía en mí y no tiene ninguna duda de que siempre poder cumplir cualquier meta que me proponga. Por todo el cariño y amor que me brinda día a día,

A mi padre German Constante el cual me brindo su apoyo económico hasta que las circunstancias se lo permitieron, por ser la persona que estuvo presente a lo largo de todas mis etapas educativas, brindando su cariño para lograr que me convirtiera en la persona que soy.

A mi tutor Daniel Diaz el cual bajo su tutela pude aprender muchísimo, por su guía y consejos útiles que me brindó, ya que sin esta este trabajo no hubiera podido ser posible. Por su presencia infaltable en mis horas de trabajo confusas.

A la Universidad Politécnica Salesiana la cual me abrió sus puertas y permitirme estudiar esta prestigiosa carrera, a mis docentes por brindarme sus conocimientos para poder culminar la misma.

Índice de contenido

1	INTRODUCCIÓN.....	9
1.1	Antecedentes.....	9
1.2	PROBLEMA DE ESTUDIO.....	12
1.3	Justificación del problema.....	16
1.4	Objetivos.....	17
1.4.1	Objetivo General:.....	17
1.4.2	Objetivos específicos:.....	17
1.5	Marco Metodológico.....	18
1.5.1	Metodologías Ágiles.....	18
2	CAPITULO I: MARCO TEÓRICO.....	23
2.1	Meteorología.....	23
2.1.1	Métricas Meteorológicas.....	24
2.2	Datos Meteorológicos.....	27
2.2.1	Formatos de Recolección Datos.....	28
2.3	API.....	30
2.3.1	Ventajas y Desventajas.....	30
2.4	API Meteorológica.....	31
2.4.1	API Weatherstack.....	32
2.5	Open Source software.....	34
2.5.1	Ventajas y Desventajas.....	35
2.5.2	Requisitos del software open source.....	35
2.5.3	Licencias de Open Source.....	36
2.6	Herramientas de desarrollo.....	38
2.6.1	Tecnologías de BackEnd.....	38
2.6.2	Tecnologías FrontEnd.....	38
2.7	Sistema Web.....	38
2.7.1	Características.....	38
2.7.2	Ventajas y Desventajas.....	39
2.8	Software.....	40
2.8.1	Características.....	40
2.8.2	Ventajas y Desventajas.....	40
2.9	MangoDB Atlas.....	41

2.9.1	Características	41
2.9.2	Ventajas y Desventajas.....	42
2.10	Node JS.....	42
2.10.1	Características	42
2.10.2	Ventajas y Desventajas.....	43
2.10.3	Módulos de Node.js.....	44
2.11	Metodología SCRUM.....	44
2.11.1	Características	44
2.11.2	Ventajas y Desventajas.....	45
2.11.3	Tres Pilares de Scrum.....	46
2.11.4	Etapas de Scrum	47
3	CAPITULO 2: ANÁLISIS Y DISEÑO	49
3.1	Requerimientos Funcionales y No Funcionales	49
3.2	Historias de Usuario	53
3.3	Diagramas de Caso de Uso.....	59
3.4	Diagrama de Secuencias	62
3.5	Modelo Conceptual de la BBDD.....	66
4	CAPITULO 3: CONSTRUCCIÓN Y PRUEBAS.....	67
4.1	Arquitectura de la Propuesta.....	67
4.1.1	Modelo Cliente-Servidor.....	67
4.1.2	Módulo usuario	68
4.1.3	Modulo web.....	69
4.1.4	Diagrama de bloques general	69
4.2	Diseño de interfaces.....	70
4.2.1	Diseño de interfaces del usuario web	70
4.2.2	Diseño interfaz principal	71
4.2.3	Diseño interfaz consulta	72
4.2.4	Diseño interfaz subir información.....	75
4.3	Diagrama de navegación	76
4.4	Arquitectura del sistema	77
4.4.1	Heroku.....	77
4.4.2	Base de datos No SQL: MongoDB Atlas Data base	78
4.5	Diagrama de Despliegue.....	78
4.5.1	Como se comunica la aplicación entre capas	78
4.6	Algoritmos.....	80

4.7	Pruebas.....	85
4.7.1	Plan de pruebas.....	86
4.7.2	Pingdom Website Speed Test.....	86
4.7.3	Apache JMeter.....	88
4.7.4	Pruebas no funcionales.....	89
4.7.5	Prueba de Caja Negra.....	93
4.7.6	Prueba de Caja y Stress.....	96
4.8	Análisis de Resultados.....	106
4.8.1	Resultados pruebas no funcionales.....	108
4.8.2	Resultados pruebas de caja negra.....	110
4.8.4	Resultados de pruebas de stress JMeter.....	111
	CONCLUSIONES.....	114
	RECOMENDACIONES.....	115
	Bibliografía.....	117
	ANEXOS.....	119
	Manual técnico.....	119

Índice de Figuras

1. Figura	Pilares de Scrum.....	46
2. Figura	<i>Diagrama de caso 1: RF1</i>	59
3. Figura	<i>Diagrama de caso 2: RF2</i>	59
4. Figura	Diagrama de caso 3: RF3.....	60
5. Figura	<i>Diagrama de caso 4: RF4</i>	60
6. Figura	<i>Diagrama de caso 5: RF5</i>	61
7. Figura	<i>Diagrama de caso 6: RF6</i>	61
8. Figura	<i>Diagrama de Secuencia 1</i>	62
9. Figura	<i>Diagrama de Secuencia 2</i>	62
10. Figura	<i>Diagrama de Secuencia 3</i>	63
11. Figura	<i>Diagrama de secuencia 4</i>	64
12. Figura	<i>Diagrama de Secuencia 5</i>	65
13. Figura	<i>Diagrama de Secuencia 6</i>	65
14. Figura	<i>Diagrama de la base de datos.</i>	66
15. Figura	<i>Diagrama modelo cliente-servidor.</i>	68
16. Figura	<i>Descripción de diagrama de bloques</i>	70

17. Figura	<i>Interfaz pantalla principal (consultar clima)</i>	72
18. Figura	<i>Interfaz pantalla principal (Histórico Ecuador)</i>	74
19. Figura	<i>Interfaz pantalla principal (Histórico Ecuador)</i>	75
20. Figura	<i>Interfaz pantalla principal (Histórico Ecuador)</i>	76
21. Figura	<i>Mapa de navegación del aplicativo</i>	77
22. Figura	<i>Diagrama de despliegue entre arquitecturas</i>	78
23. Figura	<i>Diagrama de despliegue entre redes</i>	79
24. Figura	<i>Algoritmo de la aplicación web</i>	80
25. Figura	<i>Bloque de código aplicación web principal</i>	81
26. Figura	<i>Algoritmo de la consulta del clima por ciudad, fecha y hora en la aplicación web</i>	82
27. Figura	<i>Bloque de código búsqueda por clima, hora y fecha</i>	83
28. Figura	<i>Algoritmo de la consulta del clima para descargar los datos de la página web</i> 83	
29. Figura	<i>Bloque de código descargar datos del clima (HTML)</i>	84
30. Figura	<i>Algoritmo de la consulta del clima para subir datos en la página web</i>	84
31. Figura	<i>Bloque de código subir información a la aplicación web</i>	85
32. Figura	<i>Prueba de rendimiento</i>	86
33. Figura	<i>Rendimiento de la página</i>	87
34. Figura	<i>Peticiones de la página web</i>	87
35. Figura	<i>Parámetros con 10 usuarios en JMeter</i>	97
36. Figura	<i>Resultados en tabla con 10 usuarios en JMeter</i>	97
37. Figura	<i>Gráfica con 10 usuarios en JMeter</i>	98
38. Figura	<i>Gráfica de datos del resultado con 10 usuarios JMeter</i>	98
39. Figura	<i>Parámetros con 100 usuarios en JMeter</i>	99
40. Figura	<i>Resultados en tabla con 100 usuarios en JMeter</i>	100
41. Figura	<i>Gráfica con 100 usuarios en JMeter</i>	100
42. Figura	<i>Gráfica de datos del resultado con 100 usuarios JMeter</i>	101
43. Figura	<i>Parámetros con 1000 usuarios en JMeter</i>	102
44. Figura	<i>Resultados en tabla con 1000 usuarios en JMeter</i>	102
45. Figura	<i>Gráfica con 1000 usuarios en JMeter</i>	103
46. Figura	<i>Gráfica de datos del resultado con 1000 usuarios en JMeter</i>	103
47. Figura	<i>Parámetros con 10000 peticiones en JMeter</i>	104
48. Figura	<i>Resultados en tabla con 10000 peticiones en JMeter</i>	105
49. Figura	<i>Gráfica con 10000 peticiones en JMeter</i>	105
50. Figura	<i>Gráfica de datos del resultado con 10000 peticiones en JMeter</i>	106

Índice de Tablas

1.Tabla	<i>Cuadro comparativo de metodologías ágiles:</i>	18
2.Tabla	<i>Cuadro comparativo de metodologías ágiles:</i>	21
3.Tabla	<i>Análisis comparativo</i>	49
4.Tabla	<i>Requerimientos Funcionales</i>	50
5.Tabla	<i>Requerimientos NO Funcionales</i>	51
6.Tabla	<i>Historia de Usuario 1</i>	53
7.Tabla	<i>Historia de Usuario 2</i>	54
8.Tabla	<i>Historia de Usuario 3</i>	55
9.Tabla	<i>Historia de Usuario 4</i>	56
10.Tabla	<i>Historia de Usuario 5</i>	57
11.Tabla	<i>Historia de Usuario 6</i>	58
12.Tabla	<i>Pruebas del sistema</i>	88
13.Tabla	<i>Prueba no funcional 1</i>	89
14.Tabla	<i>Prueba no funcional 2</i>	89
15.Tabla	<i>Prueba no funcional 3</i>	90
16.Tabla	<i>Prueba no funcional 4</i>	91
17.Tabla	<i>Prueba no funcional 5</i>	92
18.Tabla	<i>Prueba no funcional 6</i>	92
19.Tabla	<i>Prueba 1 caja negra</i>	93
20.Tabla	<i>Prueba 2 caja negra</i>	94
21.Tabla	<i>Prueba 3 caja negra</i>	94
22.Tabla	<i>Prueba 4 caja negra</i>	95
23.Tabla	<i>Prueba 5 caja negra</i>	95
24.Tabla	<i>Prueba 6 caja negra</i>	96
25.Tabla	<i>Compatibilidad de navegadores</i>	107
26.Tabla	<i>Resultados pruebas no funcionales</i>	108
27.Tabla	<i>Resultados pruebas de caja negra</i>	110
28.Tabla	<i>Resultados pruebas de stress JMeter</i>	112

RESUMEN

Este proyecto propone un sistema web para brindar datos meteorológicos usando una Api, capaz de transmitir los datos a los usuarios gratuita y fácilmente. El sistema logra ofrecer información atmosférica histórica de las ciudades de Ecuador, también permite observar el clima de todos los países del mundo facilitando descargar los datos en diferentes tipos de formatos. El objetivo primordial para este proyecto técnico es la elaboración de un sistema web, el cual permita obtener datos detallados, útiles y eficaces que proporcione información meteorológica, el mismo que contiene una base de datos (DB) y algunos procesos de Extracción, transformación y carga (ETL). Los cuales se exhibirán datos de la temperatura, humedad, precipitación atmosférica, nubosidad y velocidad del viento. En el primer capítulo se describe el problema de la falta de sistemas web donde faciliten datos meteorológicos, haciendo énfasis en la necesidad de una estrategia para solucionar el mismo. El segundo capítulo refiere al marco teórico y las características de todos los elementos integrados en el sistema web realizado. Se anuncia el objetivo principal que persigue realizar el sistema web con éxito para la facilidad de obtener datos meteorológicos. El tercer capítulo expone el funcionamiento de sistema mediante diagramas de secuencia y por último tenemos las conclusiones y recomendaciones del sistema web.

ABSTRAC

This project proposes a web system to provide meteorological data using an API, capable of transmitting the data to users freely and easily. The system manages to offer historical atmospheric information of the cities of Ecuador, it also allows observing the climate of all the countries of the world, facilitating the downloading of data in different types of formats. The main objective of this project is the creation of a system that allows obtaining detailed and useful information on meteorological data, this system is formed by a database and a set of ETL processes, in charge of displaying data on temperature, precipitation, humidity, cloud cover and wind speed and a web system to visualize them. The first chapter describes the problem of the lack of web systems where they provide meteorological data, emphasizing the need for a strategy to solve it. The second chapter refers to the theoretical framework and the characteristics of all the elements integrated in the web system carried out. The main objective that the web system pursues successfully for the ease of obtaining meteorological data is announced. The third chapter exposes the operation of the system through sequence diagrams and finally we have the conclusions and recommendations of the web system.

1 INTRODUCCIÓN

1.1 ANTECEDENTES

De acuerdo con (Campoverde, 2018) Su estudio tuvo como objetivo general analizar el calor Urbano del cantón Cuenca en épocas secas y húmedas en condiciones extremas en los años 2015 hasta 2017 basado en datos meteorológicos. A través de esta investigación la cual utiliza una metodología cuantitativa ha permitido desarrollar aspectos estadísticos para el análisis de los datos. Como resultados de esta investigación en los años 2015 al 2017, arroja una distribución de la onda de calor atravesada en Cuenca, de la misma manera los valores de ICU ascendente. Concluyendo que de forma global la revisión de estos datos meteorológicos será fundamental para prevenir y reducir riesgos en el medio ambiente.

Como lo hace notar (Marcos, 2017) Su estudio implementó una interfaz web la cual permite la visualización de información meteorológica. Para recolectar datos basándose en las observaciones de los diferentes comportamientos naturales, se utilizó una metodología cuantitativa. Los resultados de este proyecto son el ingreso a su sitio web, el almacenamiento de información de una forma correcta y la presentación visual de graficas meteorológicas. Concluyendo con el desarrolló un sistema SIG WEB que permitirá una mejora en el proceso de análisis de los datos obtenidos.

Desde la posición de (Aris Munandar, 2017) Su estudio tuvo como objetivo general el diseño de un sistema el cual monitoree el clima basado en un aplicativo móvil. La metodología que se utiliza es cualitativa ya que se hace un acopio de datos basados en la visualización de conductas del medio ambiente. En los resultados el sistema ha sido probado mediante un dispositivo móvil con una resolución de pantalla de 720 x 1280 pixeles, 5,5 pulgadas y sistema Android. Presentando: muestras de todas las medidas actuales; parámetros de las condiciones climáticas; muestras de gráficos de los cambios de temperatura. Concluyendo que

el sistema puede mostrar la información meteorológica mediante una aplicación móvil, mostrando los datos meteorológicos.

Tal como afirma (Kumar, 2020) Su estudio tuvo como objetivo proporcionar una solución óptima para el seguimiento de las condiciones climáticas a un local, que cuente con un bajo costo basado en internet de las cosas (IoT). En el desarrollo de esta investigación se utiliza la metodología cuantitativa mediante características observables y cuantificables, usando métodos estadísticos el análisis de la información. Como resultado da que el sistema pudo funcionar de una manera continua por 10 horas sin presentar ningún error. Este sistema fue capaz de obtener muchos datos meteorológicos confiables y eficaces. Entonces el sistema funciona satisfactoriamente, aunque es de bajo costo y potencia optimizada. Concluyendo que gracias a este trabajo se pudo verificar que el monitoreo del clima es una perspectiva importante en estos días y que también a un nivel bien localizado para su uso en cualquier campo. La tecnología IoT nos ha ayudado a diseñar un sistema de bajo costo para monitorear las condiciones climáticas de forma continua.

A juicio de (Limlahapun, 2009) El objetivo general de este desarrollo es crear un sistema web de alerta y monitoreo de inundaciones con la utilización de datos meteorológicos e hidrológicos. Se define parámetros de recaudación, manipulación, detección, verificación y recalificación de información para dar a conocer mensajes urgentes. En el desarrollo de esta investigación se utiliza la metodología cuantitativa debido a que tiene una orientación estadística para el estudio de esta información, estos ayudaran a evitar catástrofes naturales. Se obtuvo como resultado una interfaz web de fácil uso que permite al usuario mantener una comunicación eficiente y eficaz. Este sistema es de fácil acceso a consultas y Datos GIS para cualquier persona con acceso a la web.

En consecuencia, interactividad del mapeo espacial web estará compuesto por las herramientas de funciones personalizadas como acercar, alejar, información de consulta, etc. Concluyendo que el sistema de alerta temprana propuesto es la transformación de la comunicación tradicional en un sistema basado en la web, que es más eficaz porque tiene un mejor sistema de gestión de la información meteorológica.

1.2 PROBLEMA DE ESTUDIO

El desarrollo de este proyecto es importante puesto que los desastres climatológicos en todo el mundo tienen como consecuencia daños generales en propiedades, lesiones físicas y con 9000 muertes en el año 2019. Dicho número ha disminuido a relación del año 2010 con 295.000 muertes al año por catástrofes naturales. Gracias al avance de la tecnología la cual ha permitido predecir alguno de estos desastres a tiempo y se tomaron las medidas necesarias. Por tal motivo la recolección de esta información es vital. Se requiere de una entrega de *datos meteorológicos* efectiva y oportuna. Estos pueden funcionar como una estrategia de prevención para dichos problemas. Una solución a estos problemas es la implementación de algún tipo de sistema de comunicación por internet el cual puede funcionar como una alerta. Un *sistema web* es fundamental en la recolección de datos meteorológicos, sirve para medir y registrar variables climáticas, haciéndolas accesibles al público en general (Limlahapun, 2009).

Las tendencias mundiales modernas dentro del cambio climático en las urbanizaciones, de igual manera, en la creciente demanda de la población por un entorno urbano seguro y confortable. Es fundamental para los avances de investigación dentro del campo de la meteorología incluyendo el desarrollo de servicios a las diferentes ciudades del mundo como a su población sobre las condiciones ambientales (Abbatea, 2013). “*Las tendencias recientes en las variaciones de las condiciones climáticas son drásticas en todo el mundo y, además, su creciente imprevisibilidad es una gran preocupación. Las soluciones existentes son altamente globales y son inaccesibles para el hombre común*” (Kumar, 2020). “*Datos meteorológicos son necesarios alrededor del mundo para lograr obtener información de condiciones ambientales del todo el mundo. Para brindar de esta manera datos globales que pueden ayudar a predecir desastres ambientales*” (Abbatea, 2013)

En Latinoamérica cada año, mueren alrededor de 100.000 personas por desastres ambientales. Por esta razón, los eventos climáticos extremos tienen un costo de 485 mil millones en pérdidas materiales. En algunos años a futuro estas cifras podrían aumentar en consecuencia al calentamiento global dado que aumenta la frecuencia e intensidad de los desastres naturales (Turner, 2017) . En los países de Latinoamérica, el clima es más difícil de predecir, puesto que, no es fácil acceder a consultas de los datos meteorológicos (Turner, 2017). Algunos de estos desastres se pueden prevenir al contar con la información necesaria en páginas web que estén al alcance de cualquier persona.

En el Ecuador existe una institución pública denominada Instituto Nacional De Meteorología e Hidrología – “INAMHI” la cual *“opera y mantiene la infraestructura nacional de estaciones meteorológicas e hidrológicas: recopila, estudia, procesa, publica, y difunde la información hidrometeorológica”* (INAMHI, 2021). Es el organismo científico en el Ecuador de la generación y difusión de la información hidrometeorológica (INAMHI, 2021). Dentro del país es la única institución que cuenta con un sistema web para la presentación de datos meteorológicos. Por consiguiente, es importante tener la información meteorológica en el país con fácil accesibilidad. Puesto que no es posible por falta de sistemas web.

El presente proyecto de investigación responderá a la presente problemática: ¿Existe correlación en variedad de formatos en datos climáticos y falta de sistemas web meteorológicos, impedimento en el uso de los datos ambientales y escasez de desarrollo de software meteorológico Open Source? el problema que incita al desarrollo de este proyecto es la variedad de formatos en la que se maneja la información meteorológica. Para la determinación de estos se utilizan varias medidas por ende la obtención de datos tiene diferentes formatos como pueden ser: JSON, XML, XLSX, información no estructurada en textos planos, entre otros. Dicho problema radica en la falta de sistemas web que brinden una unificación de estos para poder generalizar el manejo de ellos. Este proyecto técnico es

necesario, pues los datos son de suma importancia para la medición, predicción, prevención para el futuro. Otro problema que ínsita a desarrollar este sistema es la falta de software meteorológico que sean open source dentro del país los cuales permitan estudiar y modificar el código para poder brindar algún tipo de avance o enfoque específico. El principal beneficio de este desarrollo será tener un acceso a dichos datos en el cual no importe el formato en el que se manejen los mismos dejando al alcance de cualquier persona y una manera más sencilla de obtener los datos; el sistema web que se realizará será open source facilitando a cualquier usuario poder modificar o estudiar el código a su conveniencia. Este proyecto ayudara a la comunidad científica y universitaria, para posibles investigaciones con los datos históricos meteorológicos que se encontraran en una base de datos brindando una unificación de formatos para su correcta administración.

Las causas que originaron el problema: i) falta de sistemas web meteorológico en el país. Esta situación afecta directamente a la obtención de datos y con ello un manejo incorrecto de los mismos. ii) Dificultad del manejo de datos meteorológicos. Por lo tanto, el presente proyecto se encargará de reunir información en una base de datos debido a que se uniformaran los diferentes formatos de los datos. iii) Escasez de desarrollos web meteorológicos Open Source. Provoca un difícil acceso a los datos ambiental.

Los efectos que produce este problema: i) La imperfección en el manejo de la información. Genera una ii) Variedad de formatos en información meteorológica. También genera iii) Un déficit al momento de consulta de datos meteorológicos. Mismos que cuentan con un formato no estructurado de información, puesto que dificulta su uso. Todos estos efectos se podrán resolver con la implementación de un sistema web meteorológico, el cual permitirá acceder a dichos datos de una forma más sencilla, mejorar la gerencia de datos, logrando tener un histórico de ellos sin ningún costo.

Como una solución a la problemática se propone el desarrollo de un sistema web mediante la utilización de un API. Permitirá una mejora en accesibilidad a los datos ambientales del País. Para obtener la recolección de datos meteorológicos en una base para su presentación, donde se encontrará referencias climáticas. Los principales beneficios de este proyecto serán el almacenamiento de información ambiental y adquirir estos de forma gratuita. El mejor uso de los mismos tendrá una forma más sencilla para su administración.

1.3 JUSTIFICACIÓN DEL PROBLEMA

Este proyecto tiene por objetivo establecer el uso y recepción de datos meteorológicos mediante un API en un sistema web. El cual se ejecutará de una manera autónoma sin ninguna dependencia de formatos ambientales tradicionales.

Actualmente no se cuenta con un sistema web que permita alojar y presentar la información obtenida de un API. Por consiguiente, contara con datos históricos, para brindar información que se podría usar en un futuro en predicciones meteorológicas.

Cada año, mueren alrededor de 100.000 personas por desastres ambientales, la mayoría de la población antes mencionada se encuentra en Latinoamérica (Turner, 2017). Considerando que, en su mayoría, las ciudades del país no cuentan con informes de datos en un sistema web. En efecto, el proyecto contribuirá al país, la cual poseerá una red segura de datos que no comprometerá la información obtenida. Conforme a lo expuesto es oportuno que contar con un sistema web que presente los datos meteorológicos sin importar su formato.

Ciertamente al observar la falta de sistemas web meteorológico en el país se considera vital contar con una recolección de datos obtenidos desde un API. Al momento se tiene dificultad del manejo de datos meteorológicos, la cual puede llegar a ser importante en registros históricos.

Por ende, se desarrollará un sistema web que recolecte los datos de un API. Estos presentarán beneficios sociales al país y público en general que necesite la información meteorológica. Además, se obtendrá beneficios, por consiguiente, se realizará un mejor uso y recolección de los datos meteorológicos, mismos permitirán almacenar datos históricos que puedan ser usados a futuro.

1.4 OBJETIVOS

1.4.1 Objetivo General:

Elaborar un sistema web open source que recolecte datos meteorológicos de múltiples formatos a través de un API.

1.4.2 Objetivos específicos:

- Analizar los requerimientos de un sistema web que proyecte datos meteorológicos de diferentes formatos.
- Diseñar en base a la metodología escogida los artefactos UML necesarios para la construcción de un sistema.
- Desarrollar un sistema web que recopile datos meteorológicos mediante un API.
- Evaluar el sistema web meteorológico y el cálculo computacional mediante métricas.

1.5 MARCO METODOLÓGICO

1.5.1 Metodologías Ágiles

Son metodologías de desarrollo las cuales aplican el método de operación en el entorno del proyecto, de esta forma se consigue mejor de tiempos y resistencia en la revelación, a fin de adaptar el plan y su progreso a las particularidades del marco. Las empresas que utilizan esta metodología obtienen resolver sus ideas de modo adaptable, autónomo y firme logrando la reducción de costos y aumentar su rendimiento (Vlaanderen, 2011).

SCRUM es la metodología ágil la cual nos servirá para la implementación de este proyecto técnico para direccionar la elaboración del sistema web metodológico mediante la utilización de una interfaz de programación de aplicaciones (API). El termino ágil es utilizado por los especialistas en desarrollo de software como una mejora de procesos en el desarrollo. La característica principal era la poca flexibilidad y la cantidad excesiva de documentación. La agilidad de este enfoque ha permitido distribuir de una manera continua el sistema de software con un diseño de iteraciones más veloces (Vlaanderen, 2011).

1.Tabla

Cuadro comparativo de metodologías ágiles:

Metodologías de desarrollo de software			
Criterios	Rup	XP	Scrum
Tipo de Framework	Verificación del planteamiento, creación e información	Enfocado en la adaptabilidad, con una mejor elasticidad,	Dirección e implementación de software, enfocado en

	de sistemas orientados a objetos	dinamismo y funcionalidad	procesos activos y continuos
Tipo de Revisión	La metodología se repite una o más veces en cada fase, afinando los objetivos en el proceso. Si no se completa una fase, se inicia la siguiente.	Como mínimo, se debe incorporar una vez al día y se deben realizar pruebas cuando se complete el proceso.	Se requiere una revisión diaria y debe incluir los siguientes elementos: trabajo completado el día anterior; trabajo programado para completarse; y tareas que pueden o no completarse.
Objetivos	Orientado a objetos, estableciendo los principios, plantillas y ejemplos para todos los procesos de desarrollo de software	Fundamentada en dar importancia a trabajos con resultados directos: <ul style="list-style-type: none"> ➤ Beneficios al cliente ➤ Actividades grupales ➤ Basarse en aspectos: valor, durabilidad, calidad y objetivos 	Diseñado para aplanar complicados: <ul style="list-style-type: none"> ➤ Respuestas inmediatas ➤ Requerimientos variables ➤ Cambios continuos.

Tipos de Desarrollo	Desarrollo continuo en crecimiento por etapas: <ul style="list-style-type: none"> ➤ Inicio ➤ Desarrollo ➤ Construcción ➤ Transición 	Ligera y flexible. Desarrollo por etapas <ul style="list-style-type: none"> ➤ Estrategias planificadas ➤ Diseño ➤ Codificación ➤ Pruebas 	<ul style="list-style-type: none"> ➤ Desarrollo sencillo. ➤ Domino empírico y flexible a la transformación del desarrollo.
Tipo de Proyecto	Dirigido a proyectos grandes y a largo plazo	Se utiliza para mínimos proyectos al tener una certeza del valor fijo del proyecto	Recomendado para las mejoras rápidas y organizaciones que no dependen de una fecha limite

(Calan, 2017)

Elaborado por Ken Constante Metodologías Ágiles

El desarrollo inicia mediante la definición de los requerimientos funcionales, mediante el agrupamiento de los miembros del grupo de trabajo e interesados, los cuales brindan su apoyo en los diferentes Sprints para lograr obtener un diagrama de casos de uso de la funcionalidad del sistema (Calan, 2017).

2.Tabla

Cuadro comparativo de metodologías ágiles:

Características	Metodología		
	XP	SCRUM	RUP
Todos los miembros implicados tienen el conocimiento sobre lo que ocurre en el proyecto, esto hace que el proyecto sea entendido por todos y así tener una visión global de cómo se implementara el sistema web para la obtención de datos meteorológicos en el Ecuador.		x	
Desarrollo interactivo e incremental	x	x	x
Habitualmente tendrá una inspección para detectar posibles problemas, esto se hará para que el plan fluya de mejor manera y el equipo trabaje de forma organizada y así alcanzar el objetivo.		x	
Integración del equipo de programación con el cliente.	x		
Esta característica nos ayuda a que debemos entender que el proyecto tiene requisitos cambiantes y debemos adaptarnos para innovar, ya que la		x	

complejidad y la flexibilidad en este plan son fundamentales.			
Es un tanto complejo ya que necesita obtener resultados de datos meteorológicos pronto, poder realizar cambios y que tenga innovación.		x	x
Metodología que nos ayudara al análisis del diseño del sistema web basado en la estabilidad y flexibilidad basada en un proceso interactivo e incremental.	x	x	x

(Calan, 2017)

Elaborado por Ken Constante, Características Adaptadas a la metodología Scrum.

Esta metodología permite empezar proyectos complejos que exigen una flexibilidad y una rapidez fundamental al instante de ejecutar los resultados. La táctica ira orientada a resolver y ordenar los errores que se puedan causar en desarrollos bastante largos, a través de, reuniones frecuentes para afirmar el acatamiento de los objetivos establecidos (Calan, 2017)

2 CAPITULO I: MARCO TEÓRICO

2.1 METEOROLOGÍA

Es una ramificación de la física que estudia la atmósfera y los fenómenos físicos que tienen lugar en su interior y que son los determinados del llamado tiempo meteorológico. La meteorología hace posible prevenir lugares donde se encuentran los fenómenos físicos y así eludirlos (Abelladan, 2020).

Para el meteorólogo Luis Mashiques Zaragoza la meteorología al ser una de las ciencias más difíciles de comprender y predecir, puesto que tiene una gran influencia en el diario vivir del ser humano y a la vez menciona que es una ciencia que ha permitido que el ser humano la desarrolle gracias a sus experiencias meteorológicas como son saber sobre los ámbitos naturales que lo rodean para que de esta manera se pueda prevenir climas extremos y desastres naturales (Turner, 2017).

❖ **Ramificación**

➤ **Meteorología sinóptica.**

La cual se enfoca a la predicción del clima, utilizando la observación continua, constante y en tiempo real del medio ambiente, recolectando datos en mapas geográficos y aprendiendo el modo más idóneo de presentación de los datos para el uso en general (Uriarte, 2020).

➤ **Micrometeorología.**

Se diferencia de la sinóptica en que su dirección es mucho más completa, y el deseo de ponderar el clima global, se entiende como un sistema de movimiento y efecto a nivel mundial (Uriarte, 2020).

➤ **Mesometeorología.**

Análisis de la índole meteorológica de mediana escala, es decir, en la membrana intermedia de la atmósfera (desde varios kilómetros y los 2000 km de altitud) (Uriarte, 2020).

➤ **Hidrometeorología**

Es una perspectiva meteorológica que se centra en la dinámica del agua en la atmósfera (gaseosa, líquida o sólida): lluvia, nieve, vapores atmosféricos, etc. (Uriarte, 2020).

2.1.1 Métricas Meteorológicas

➤ **Temperatura**

Es una puntuación física que describe el calor y el frío, es ajustado a la energía cinética del término medio de una masa. Los termómetros se utilizan para calcular la temperatura. Estos pueden ser tipificados a una de tres escalas (Meteoblue, 2006).

- Escala Fahrenheit (°F), que es especialmente conocido en EE.UU.
- Escala Celsius (°C), que es la más empleado comúnmente.
- Escala Kelvin (°K), muestra la unidad SI para la misma.

La medida de la temperatura se realiza por lo general en un sitio y elevación determinada, 2 metros del piso, si están en correlación con las directrices de la Organización Meteorología Mundial (Meteoblue, 2006).

➤ **Precipitación**

Estudia la aceleración del agua de la superficie del territorio, con estructura de lluvia (líquida), nieve (cristales), hielo o granizo (agua congelada) y rocío (condensación en las superficies. (Meteoblue, 2006).

- La velocidad convectiva cae en modo de lluvia con una ligera alteración en la fuerza y una demarcación determinada durante un tiempo relativamente corto, precisamente porque las nubes convectivas fija la extensión horizontal (Meteoblue, 2006).
- La rapidez orográfica se desploma cuando el aire promovido por el viento está forzado a ascender en la longitud de las laderas de las formaciones terrestres amplias, como los bosques (Meteoblue, 2006).
- La celeridad estratiforme producida por los métodos frontales, que por lo común se reparte en la lluvia similar sobre un área mayor (Meteoblue, 2006).

La precipitación puede ser medida por tres métodos:

1. El pluviómetro
2. La partición de la precipitación en la atmosfera se puede cuantificar empleando un radar
3. El manejo de depósitos, zanjas, medición de drenaje o desbordamiento.

Las mediciones se expresan en milímetros, centímetros o pulgadas; 1mm corresponde a 1 litro de agua por metro cuadrado. (Meteoblue, 2006).

➤ **Humedad**

La humedad relativa es el tanto por ciento de congestión de un cuerpo característico del aire a una temperatura especial. La humedad del aire está a la voluntad de la temperatura y la presión del volumen de aire analizado. Como la unidad de humedad es por ciento, varía entre 0 y 100% (Meteoblue, 2006).

$E(T_d)$ = presión de vapor saturado en el punto de rocío T_d (normalmente en $g/(m^3)$)

$E(T)$ = presión de vapor cargado a la (real) temperatura del aire T (generalmente, en $g/(m^3)$).

La humedad específica la cantidad de agua que traslada por el aire, es importante para resolver el progreso de las nubes y el factor precipitación (Meteoblue, 2006).

➤ **Capa de Nubes**

Las nubosidades se presentan en porcentaje (%) de la máxima cobertura de nubes.

Normalmente las diferentes nubosidades se agrupan de la siguiente manera del 0 al 25%, del 25% al 50%, etc. (Meteoblue, 2006).

Donde el 0% quiere decir que no existe ninguna nube visible en el cielo. El 50% representa que la mitad del cielo está cubierta de nubes. (Meteoblue, 2006).

Capa de nubes agregadas se la puede representar de acuerdo con la WMO como:

- Nubes bajas: de 0 a 2 km (0-2 km en el Ecuador)
- Nubes medias: de 2 a 7 km (2-8 km en el Ecuador)
- Nubes altas: de 5 a 13 km (6-18 km en el Ecuador)

(Meteoblue, 2006).

La medida de la nubosidad es muy ardua en la práctica. Los dispositivos más empleados son las cámaras de los satélites, que generan una visión general de la nubosidad, pero no pueden discernir bien entre bajo, medio y nubes altas (Meteoblue, 2006).

➤ **Velocidad del Viento**

La media de todas las velocidades del viento considerado para cada intervalo de cada duración desde el periodo anterior. El viento es la orientación del aire en dirección con una velocidad determinada. El viento se determina a 10 metros sobre el suelo. (Meteoblue, 2006).

El sentido del viento se fija como el rumbo de donde sopla el viento, por ejemplo, el viento del norte viene del norte y está soplando hacia el sur. Sin embargo, el sentido

del viento está representada por una barba que señala el curso en la que sopla el viento. En climatología es común el uso de alegorías eólicas, para indicar la dirección del viento, junto con su velocidad (Meteoblue, 2006).

El sentido del viento se da en el Norte (N), Sur (S), Este (E), y Oeste (W), o en grados de cimut del norte (0-360°) (Meteoblue, 2006).

Para la celeridad del viento, hay diferentes unidades de medida:

- Kilometros por hora (km/h): $1\text{km/h} = 0.27778\text{ m/s}$
- Metros por segundo (m/s): $1\text{ m/s} = 3.6\text{ km/h}$
- Nudos (kn): $1\text{ kn} = 1.852\text{ km/h}$

(Meteoblue, 2006).

2.2 DATOS METEOROLÓGICOS

Los datos meteorológicos son aquellas estadísticas e información que ayudan a los meteorólogos a entender de una mejor manera la complejidad de un fenómeno climático a través de la medición de ciertos parámetros físicos y con la ayuda de ciertos instrumentos que permiten realizar predicciones meteorológicas (Lopez, 2012).

De esta manera con este proyecto queremos facilitar la obtención de datos meteorológicos mediante un sistema web utilizando una API para alcanzar la información que necesitamos y en los formatos que nos sea más fácil utilizar (Lopez, 2012).

Estas estadísticas se las puede conocer a través de estaciones meteorológicas correspondiente a diferentes asociaciones nacionales e internacionales repartidas globalmente. El autor Ignacio Zúñiga López menciona que estos antecedentes deben ser concretos pues de lo contrario pueden provocar equivocaciones importantes a la hora de generar computo necesarios (Lopez, 2012).

2.2.1 *Formatos de Recolección Datos*

➤ **.XLM**

La necesidad de crear grandes cantidades de datos dinámicos y complejos por un lado y la necesidad de procesarlos y publicarlos por otro, han provocado la enunciación de un formato de datos universal: XML (Genexus, s.f.).

Este sitio web brindara información sobre datos meteorológicos; para permitirse dar estos servicios a los clientes, este plan va a introducir este formato ya que es uno de los más usados por los usuarios (Genexus, s.f.)

➤ **.CSV**

Es un fichero de valores separados por comas. Además, conocido como dígitos separados por caracteres o archivo definido por comas. Estos CSV se distinguen de los planes conservadores de hojas de cálculo porque solo tienen los medios de tener una sola hoja sin los valores de celda, columna o fila guardados (WikiRyte, 2020).

Recolectar datos meteorológicos en este tipo de fichero es útil, ya que sirve para manejar gran cantidad de datos en formatos de tabla, también es muy sencillo de usar. Es fácil de visualizar ya que los datos solo están separados por punto y coma (WikiRyte, 2020).

➤ **.XSL**

El .xsl amplitud del archivo no está de primera mano asociada con la única aplicación. Esto se debe a que es más acreditado como un formato de fichero de concepto de los estilos de texto, gráficos y tablas, así como otras XML objetos de documento. .xsl archivos son un poco lo mismo con los archivos .css que la multitud de los documentos HTML usan y hay una gran porción de la aplicación que se utilizan para abrir estos archivos ya que este formato es acorde con casi todas las hojas XML estilos (Reviversoft, 2021).

El tamaño de archivo también se afilia con el estilo flexible hoja idioma para variaciones. Así este formato reúne datos meteorológicos ya que tiene hojas de estilo apoyándose principalmente en XML directores. Estos ficheros son muy prácticos ya que son muy alcanzables, ya que no es sólo una única aplicación que se puede utilizar para abrirlos (Reviversoft, 2021).

➤ **.JSON**

JSON es un lenguaje usado para el trueque de datos entre sistemas, está centrado en la notación de los literales de objeto de JavaScript. Básicamente usa el mismo dígito o forma con la que se escriben los objetos JavaScript en el código, los literales de objeto, con algunas limitaciones y añadidos extra (A, 2020).

Se trata de un formato para proteger e intercambiar datos meteorológicos que cualquier persona pueda observar. Los archivos json tienen solo texto y utilizan la extensión. json, también ejerce para relación entre el sistema web y los clientes que ven información meteorológica en formato JSON (A, 2020).

➤ **.TEXT**

Estos .text archivos se pueden crear, abrir o ver y modificar el uso de cualquier aplicación de procesamiento de texto o palabra para sistemas basados en Mac y programas de edición de texto conocido para componentes apoyados en Windows de Microsoft, como Microsoft WordPad y Microsoft Bloc de notas. El formato del texto se utiliza sobre todo cuando no es necesario el contenido de texto para ser implementado con atributos de diseño de página y elementos de formato (Reviversoft, 2021).

Utilizar este formato es útil dentro de la recolección de datos meteorológicos ya que los .text es tener archivos de texto en tamaños más pequeños, por lo que subir estos

documentos de texto a un servidor Web o adjuntar estos archivos a mensajes de correo electrónico será más rápido y fácil para los usuarios (Reviversoft, 2021).

2.3 API

Estas posibilitan que sus artículos y servicios se relacionen con otros, sin la necesidad de saber cómo están desarrollados. Esto reduce el progreso de las aplicaciones y permite escatimar tiempo. En este proyecto utilizara un API ya que le otorgan flexibilidad para las pruebas que se necesitara para la recolección de datos ayudándonos también simplificando el diseño, la administración y el uso del sistema web, proporcionando oportunidades de innovación, lo cual es ideal al momento de diseñar el API meteorológica que nos permita obtener la información en diferentes tipos de formatos (RedHat, s.f.).

2.3.1 Ventajas y Desventajas

Ventajas

➤ Agilidad o Rapidez

En este proyecto un Api ayudara a ahorrar tiempo en la realización de la actividad ya que el desarrollador no puede integrar cosas como mapas, fotos y servicios de autenticación en minutos sin crearlos desde cero (Tecnoaporte, 2018).

➤ Mejora Continua

Como los grandes proyectos no pueden ser desplazadas a través de la competencia, un API nos ayudara con en forma continua innovando y mejorando la seguridad de los servicios y brindar un sistema web que refleje datos meteorológicos correctos y que el usuario pueda obtener con facilidad (Tecnoaporte, 2018).

Desventajas

➤ **Cambios**

Si se utiliza un servicio de terceros en su desarrollo, debe considerarse que puede haberse notado que su implementación o comportamiento al usar la API ha cambiado por razones de seguridad. A partir de este cambio, es posible que pierda el acceso de la API. También hay situaciones en las que se ejecuta de forma gratuita cuando accede a la API y que los desarrolladores decidan en qué momento deberá ser pagado (Tecnoaporte, 2018).

➤ **Control de disponibilidad**

Es cierto que usar API lleva mucho tiempo para desarrollar aplicaciones web, pero sigue siendo un servicio de terceros incontrolable, y si se cae por alguna razón, de repente funciona y puede ser muy difícil. La solución no está bajo nuestro control. En cualquier caso, la única opción es esperar a que el desarrollador de la API resuelva cualquier problema relacionado con esta operación (Tecnoaporte, 2018).

➤ **Conocimiento**

Otro inconveniente que puede ocurrir al implementar un servicio de terceros como API por primera vez es que una falta de atención o el mismo conocimiento puede indicar un retraso en la misma tarea, no en todas las API. Son fáciles de implementar y no todo el mundo necesita los mismos requisitos para su necesidad (Tecnoaporte, 2018).

2.4 API METEOROLÓGICA

Son semejantes a las API de mapas en términos de incorporación e inconstancia de las fuentes de datos. Cada servicio recopila, integra y procesa el clima y otros datos destacados y después presenta acceso a ellos a través de API de manera gratuita o con algún costo. Hay varios proveedores de servicios de este tipo, cada uno con una totalidad única de capacidades, costos

variables y grados de confiabilidad variables. Algunos todavía se dirigen a mercados o comunidades específicas con características únicas para aplicaciones agrícolas y monitoreo de las propiedades del aire (EBERGEMENT, 2021).

2.4.1 API Weatherstack

Este proyecto utilizara esta API ya que es un servicio web completo que proporciona información meteorológica con graficas que ayudaran a entender mejor el tiempo que hará no solo en el día actual sino en los siguientes, pudiendo observar la evolución a través de una línea de tiempos bien estructurada y completamente interactiva (Hidalgo, 2017).

➤ **Ventajas**

- **Fuentes de datos confiables**

Impulsado por una sólida columna vertebral de fuentes de datos, nuestra API de datos meteorológicos viene con el más alto nivel de confiabilidad, consistencia y precisión (Weatherstack, 2021).

- **Respuesta ultrarrápida**

Los datos meteorológicos se entregan en formato JSON ligero para garantizar un alto nivel de velocidad y compatibilidad con cualquier lenguaje de programación (Weatherstack, 2021).

- **Infraestructura escalable**

Nuestra API REST está respaldada por una infraestructura de nube escalable construida y mantenida por una capa, capaz de manejar miles de millones de solicitudes por día (Weatherstack, 2021).

- **Búsqueda de ubicación flexible**

Se pueden buscar millones de ubicaciones por nombre de ciudad o región, código postal, dirección IP o incluso utilizando coordenadas de latitud y longitud (Weatherstack, 2021).

- **Seguridad a nivel de banco**

Cualquier tipo de dato que es recibido y enviado por la API Weatherstack cuentan con la protección del protocolo HTTPS (SSL) el cual es un estándar de la empresa. (Weatherstack, 2021).

- **Amplia documentación de API**

Una API es tan buena como su documentación, por lo que una serie de ejemplos de código interactivo en varios idiomas lo están esperando (Weatherstack, 2021).

➤ **Datos Meteorológicos y el API**

- **¿Quién ejecuta el API?**

Esta API fue creada y se da mantenimiento por Apilayer, esta empresa se encuentra en Londres, Viena, Reino Unido y Australia. Apilayer es una empresa la cual está encargada de los productos APPI y SaaS los cuales son las más usadas en el mundo, incluyendo currencykayer, eversing, currencylayer. (Weatherstack, 2021).

- **¿De dónde provienen los datos?**

Esta API cuenta con licencias de las mejores y más grandes estaciones alrededor del mundo, las cuales se supervisan a detalle para tener una garantía de coherencia y la precisión necesaria de los datos las 24 horas (Weatherstack, 2021).

- **Actualización de los datos**

Los datos devueltos por el API, así como los datos de pronóstico del tiempo, siempre contienen la información meteorológica más actualizada en el momento actual (Weatherstack, 2021).

- **Funciones del API**

- Clima Actual
- Clima Histórico
- Series de tiempo histórico
- Pronostico del Tiempo
- Búsqueda de ubicación

(Weatherstack, 2021)

2.5 OPEN SOURCE SOFTWARE

Es software el cual brinda a los usuarios la potestad y libertad de ejecutar el programa para cualquier tipo de propósito que tenga, realizar estudios y permisos de modificación del programa. También puede redistribuir copias del programa ya sea original o modificado sin tener inconvenientes de regalías a los desarrolladores pasados. El open source software al pasar los años se ha convertido en un éxito mundial, transformándose de una realidad oscura a convertirse en la moda actual en muy poco tiempo. Hoy en día existen muchos tipos de licencias Open Source, pero existen dos principales licencias que son GPL (licencia publica general) y BSD (distribución del sistema Berkeley) (Wong, 2004).

2.5.1 Ventajas y Desventajas

Ventajas

La principal ventaja del software de código abierto es que tiene la capacidad de distribuir, estudiar y publicar el código origen a través de un sistema. El código abierto, por otra parte, mejora la colaboración entre usuarios. Esta característica demuestra la velocidad y variabilidad del desarrollo múltiples instrumentos. Por arquetipo, los beneficiarios de un sistema particular pueden hacer perfeccionamiento, corregir errores y mejorar el rendimiento del servicio gracias a los miembros de la comunidad, revistas (GlosarioTic, 2015).

Desventajas

Si bien los beneficios de usar el software de código libre son importantes, hay una sucesión de desventajas que se deben considerar. Sin quejas ni asistencia técnica, la mayoría de los programas de código abierto no tiene una empresa que pueda patrocinar o hacer afirmaciones. Por ejemplo, cuando surge un altercado, la compañía se ve forzada a indagar soluciones en la sociedad o brindar soluciones que salen del equipo de TIC.

Esto puede significar n crecimiento y costo económico inesperado (GlosarioTic, 2015).

2.5.2 Requisitos del software open source

- Libre redistribución
- Código fuente
- Trabajos derivados
- Integridad del código fuente del autor
- Sin discriminación

- Distribución de la licencia
- La licencia no debe restringir otro software
- La licencia debe ser tecnológicamente virtual

2.5.3 *Licencias de Open Source*

En casi todas las licencias Open Source se puede asegurar su instalación en diferentes ordenadores, acceso al código origen, recopilar y cambiar, las diferencias fundamentales son que se puede hacer con el código origen.

➤ **Licencia Apache**

Permite que el código fuente que este bajo esta licencia se pueda modificar difundir o desarrollar bajo otros tipos de licencia Open Source o también de código privado o propietario, por lo cual esta no tiene restricciones para la distribución del código para nuevos productos. La única restricción que tiene es que se debe citar al autor del código. Este tipo de licencia es considerada como la más permisiva ya que no impone condiciones a las personas que la usan ya que estas pueden ser usuarios finales o empresarios que utilizan el código para la creación de nuevos productos bajo una licencia Apache. El objetivo es aumentar el uso de la misma, ya sea que este producto tenga un costo o no, difundir las fuentes o no (Hierro, 2017).

➤ **Licencia GPL**

Es una licencia de derechos de autor es bastante recurrente en el software libre, esta da una garantía a los usuarios finales. La libertad del software para ser usado, estudiado, compartido, copiado y modificado. Sus principales propósitos son: manifestar que el software que tiene esta licencia es libre, y proteger el código gracias a la utilización del copyleft el cual restringe a nuevos usuarios por cada vez que el código es modificado, difundido o ampliado (Hierro, 2017).

➤ **Licencia Affero**

Es una licencia de código abierto que puede considerarse como una alteración de la licencia GPL pero que además requiere que, si se usa en una aplicación web, se repartan las fuentes de esa nueva adaptación.

Ambas interpretaciones de Affero GPL fueron bosquejo para cerrar una laguna de suministrador de servicios de aplicaciones (ASP) discernir en la GPL ordinaria, donde, al utilizar pero no distribuir el software, las normas de copyleft no se activan. Cada versión se diferencia de la explicación de GNU GPL en la que se basa en tener una orden adicional que aborda el uso de software a través de una cadena informática . Esta disposición exige que el código fuente completo esté disponible para cualquier cliente de la red de las tareas con licencia AGPL, generalmente una aplicación web . (Hierro, 2017).

➤ **Licencia BSD**

Esta licencia prohíbe un poco menos el uso de las fuentes, que se puede usar en un proyecto que no utilicen licencias Open Source. La cual está basada en una variante del sistema Unix, ya que no obliga a la difusión del código de los nuevos desarrollos, también porque es una licencia permisiva y no impone condiciones. (Hierro, 2017).

➤ **Licencia para usar**

La licencia que se ocupara para la realización de este sistema web mete reológico es Apache ya que permite la utilización de estos desarrollos para su redistribución o modificación de otros que pueden ser Open Source o de código propietario, ya que

no se obliga a la distribución del código fuente de los nuevos desarrollos, también porque es una licencia permisiva y no impone condiciones. (Hierro, 2017).

2.6 HERRAMIENTAS DE DESARROLLO

2.6.1 *Tecnologías de BackEnd*

Estas técnicas son las que se usan en el lado del servidor, para tramitar las diferentes solicitudes de indagación que le llegan y para diligenciar base de datos instaladas en los mismos. Los datos una vez tratados van de regreso al dispositivo para que sean observados en el artilugio a través de las tecnologías FrontEnd. Las tecnologías más usadas en el BackEnd son PHP, Java, .NET, Python, MySQL, etc., (Ribas, 2010).

2.6.2 *Tecnologías FrontEnd*

Estas tecnologías son las que se utiliza al lado del cliente, son las herramientas la cuales se utiliza para poder hacer una conexión *en* algún tipo de servidor a través de internet. Estas tecnología y lenguajes se utilizan hoy *en día* en diferentes tipos de Navegadores, pero no todos estos lo implementan a su máxima capacidad. Las *tecnologías más utilizadas* para el desarrollo FrontEnd son HTML, CSS, jQuery, JavaScript, Bootstrap, Ajax, Angular, etc., (Ribas, 2010).

2.7 SISTEMA WEB

2.7.1 *Características*

Se deben utilizar adecuadamente diversas herramientas (HTML, CSS, ASP.NET u otros lenguajes de servidos, AJAX, etc.) para lograr una navegación con las siguientes características:

- ❖ Incluye datos en múltiples formatos, incluidos texto, sonido, video, movimiento e imágenes.
- ❖ Tiene que ver con el estilo de los datos.

- ❖ Proporciona una aplicación interactiva.
- ❖ Tiene direcciones a otros sitios y páginas web.

(BeeDIGITAL, 2018).

2.7.2 *Ventajas y Desventajas*

Ventajas

- ❖ Es una cuestión de rapidez y economía, porque la inversión en desarrollo es menor y se necesita menos tiempo.
- ❖ No es necesario crear una versión diferente para cada familia de dispositivos.
- ❖ No es necesario realizar actualizaciones periódicas ni permitir que los usuarios apliquen cambio. Cuando se realiza una variación, los usuarios de internet volverán a abrir el sitio web solo para buscar nuevas versiones y proporcionar notificaciones y advertencias

(BeeDIGITAL, 2018)

Desventajas

- ❖ Debe organizar los sitios web de sus destinatarios para que puedan utilizar el servicio sin perder el tiempo.
- ❖ Los usuarios suelen necesitar más integración en sus aplicaciones web que en las aplicaciones móviles para utilizar las herramientas.
- ❖ El rendimiento está limitado por la respuesta del navegador en el que se ejecuta la aplicación.
- ❖ Es incompatible con el dispositivo que se está utilizando, lo que resulta en una experiencia de usuario más pobre que la aplicación original.
- ❖ La funcionalidad de línea se ignora en las aplicaciones web

(BeeDIGITAL, 2018).

2.8 SOFTWARE

Software es un vocablo informático que hace alusión a un programa de mecanización, así como información, métodos y modelos que permiten realizar distintas labores en un sistema informático. Por lo general se emplea este término para a contar de una forma muy genérica al proceso de un artilugio informático, sin embargo, el software abarca todo aquello que es intangible en un sistema computacional (Significados, 2014).

2.8.1 *Características*

- ❖ Cumple con el objetivo del proyecto
- ❖ Es fácil de usar y de aprender a modificarlo
- ❖ Es resistente a ataques extremos y es seguro
- ❖ Puede ser modificado por otros desarrolladores
- ❖ Se puede ocupar en diversos equipos y formatos

(Significados, 2014).

2.8.2 *Ventajas y Desventajas*

Ventajas

- ❖ Cualquiera puede obtener el software de forma gratuita bajo términos de la licencia.
- ❖ Es gratuito
- ❖ Existe libertad de conocimiento y cooperación entre los usuarios, lo que puede aumentar la creatividad.
- ❖ Tiene una rápida corrección de errores a través de servicios comunitarios en línea y acceda al código fuente.
- ❖ Absoluta libertad de proveedores. Los usuarios tienen control total y son libres de administrar su propio desarrollo y actividades (Significados, 2014).

Desventajas

- ❖ Pierde problemas para intercambiar archivos, errores o datos.
 - ❖ Sera más difícil cargar y migrar daros para el usuario en su conjunto
 - ❖ Desconocimiento. El usuario medio conoce el soporte de Microsoft, lo que aumenta el coste del aprendizaje.
 - ❖ Falta de garantía. El software gratuito no se hace responsable de ningún daño
 - ❖ La configuración requiere conocimientos previos del sistema operativo
- (Significados, 2014).

2.9 MANGODB ATLAS

La base de datos que se utilizará para el almacenamiento de datos del proyecto será MongoDB Atlas, ya que esta es un tipo de base NoSQL y es más orientada a documentos Open Source, la forma de guardar los datos ya no es en tablas como lo era tradicionalmente si no que hoy por hoy se usa una estructura BSON el cual es similar a JSON. Al ser una base de código abierto está presente para sistemas operativos como lo son Windows, GNU/Linux, Os X y Solaris y es usado en múltiples proyectos o en grandes empresas como es el caso de MTV Network, Craigslist, BCI o Fourquare. (Robledano, 2019).

2.9.1 Características

- ❖ **Consultas ad hoc.** Mediante la utilización de MongoDB podemos utilizar todo tipo de consultas. Se puede hacer un filtrado por campos, o consultas de tipo rangos y expresiones regulares. También estas consultas pueden utilizarse para restituir algún tipo de campo en específico del documento, pero al mismo tiempo puede ser una función JavaScript definida por el usuario (Robledano, 2019).
- ❖ **Almacenamiento de archivos.** MongoDB puede ser utilizado como un sistema de archivos y también incluye la repartición oficial, la que permite manejar archivos y

contenidos mediante el aprovechamiento de la magnitud para el balance de carga y replicación de datos (Robledano, 2019).

2.9.2 Ventajas y Desventajas

➤ **Ventajas**

- ❖ Motores de acaparamiento integrado
- ❖ Menor tiempo de sobre ponerse ante fallos
- ❖ Comprobación de documentos (Robledano, 2019).

➤ **Desventajas**

- ❖ No es un desenlace conveniente para aplicaciones de transigencia complejas
- ❖ No tiene una sustitución para las soluciones de patrimonio.
- ❖ Aún es una tecnología inexperta (Robledano, 2019).

2.10 NODE JS

Es un entorno de tiempo, practico en JavaScript de ahí que utiliza archivos con extensión .js.

En este la ejecución se cuenta con lo necesario para elaborar un programa escrito bajo lenguaje JavaScript. Cuenta con muchos beneficios y a la vez soluciona muchos problemas (Lucas, 2019).

2.10.1 Características

❖ Asíncrono y controlado por eventos

Todas las API de la biblioteca de Node.js están bien y no se eliminarán. Esto significa que el servidor raíz de Node.js no esperaba que la API recuperara los

datos. Una vez que se llama al servidor, pasa a la siguiente API y el sistema de notificación Node.js permite que el servidor reciba respuestas de llamadas API anteriores (Lucas, 2019).

❖ **Sin almacenamiento en búfer**

La aplicación Node.js nunca almacena datos. Esta aplicación proporciona datos solo en fragmentos (Lucas, 2019).

❖ **Procesos dentro de un solo hilo el cual contiene un bucle de eventos**

Node.js usa un modelo de un solo subproceso para organizar eventos. El sistema de eventos permite que el servidor responda sin problemas, lo que le da al servidor más flexibilidad que los servidores tradicionales que imponen restricciones en el procesamiento de solicitudes (Lucas, 2019).

Node.js usa un hilo y el mismo programa puede cumplir con requisitos más altos que los servidores tradicionales como Apache HTTP server (Lucas, 2019).

2.10.2 Ventajas y Desventajas

Ventajas

Es de Código Abierto, por lo que puede elegir módulos de la biblioteca para simplificar su trabajo (Lucas, 2019).

La Comunidad de desarrolladores de Node JS se expande todos los días (Lucas, 2019).

❖ Es una de las formas más utilizadas en la actualidad, superando el entorno y el lenguaje conocido como PHP o C (Lucas, 2019).

Desventajas

A diferencia de otros lenguajes actualizados, esto aumentara el tiempo de aprendizaje para los programadores acostumbrados a otros sistemas. Sin embargo, es fácil para aquellos que ya conocen JavaScript (Lucas, 2019).

❖ Su funcionalidad proviene de elementos incompatibles, por lo que no tiene el mismo control informático que otros lenguajes, como Java (Lucas, 2019).

2.10.3 Módulos de Node.js

Otra idea interesante a considerar en Node.js son los problemas de diseño. En general, cuando hablando de productos, estamos hablando de recursos más simples o menos simples, o características complejas que incluyen código JavaScript reutiliza aplicación (Lucas, 2019).

2.11 METODOLOGÍA SCRUM

2.11.1 Características

Vamos a descarta algunas de las características de Scrum.

➤ Tiene ciclos o Sprint

En el desarrollo del trabajo diario, o gestión de la visión, algunos productos se describen al inicio del proyecto y son prometedores. Mientas tanto se analiza sus requisitos, diseño, implementación, las pruebas y finalmente el despliegue (Abelladan, 2020).

➤ Scrum trabaja con roles

Los equipos se coordinan directamente y lideran directamente entre grupos. Esto significa que la propia organización gestiona y dirige el proyecto. Además de la otra

serie de reuniones descritas durante el curso, tenemos reuniones periódicas con el equipo para explorar el progreso del proyecto (Abelladan, 2020).

2.11.2 Ventajas y Desventajas

Ventajas

➤ Gestión de las expectativas del usuario

La participación de los clientes en cada paso del proceso y recomendar una solución.

En esencia, todo el proceso se realiza para este tipo de evaluación colaborativa (Españon, 2019).

➤ Resultados Anticipados

Obtendrá una lista de resultados en cada etapa del proceso. Por lo tanto, los clientes no tienen que esperar hasta el final para ver resultados (Españon, 2019).

➤ Flexibilidad y adaptación a los contextos

Apto para todo entorno, región y unidad de trámite. En definitiva, no es una especialidad en ningún campo (Españon, 2019).

➤ Gestión Sistemática de Riesgo

También resuelve problemas que ocurren durante el proceso de implementación y pueden afectar el rendimiento cuando ocurren. Esto es posible porque la firma del grupo de trabajo se puede implementar (Españon, 2019).

Desventajas

➤ Funciona más que nada con equipos reducidos

Por ejemplos, grandes compañías deben subdividirse en grupos con objetivos específicos. De lo contrario, realmente perderá el impacto técnico (Españon, 2019).

- **Requiere una exhaustiva definición de las tareas y sus plazos**

Scrum se perderá si estos aspectos no están bien explicados. Recuerde que las asignaciones de trabajos en cada nivel, que son exactamente las mismas (Españon, 2019).

- **Exige que quienes la utilicen cuenten con una alta cualificación o formación**

El éxito de Scrum depende de la práctica adquirida por los expertos del equipo con muchos años de experiencia (Españon, 2019).

2.11.3 Tres Pilares de Scrum

1. Figura

Pilares de Scrum.



Elaborado por Ken Constante

- **Trasporencia**

A manera de Scrum, todos los involucrados están al tanto de lo que está sucediendo y como va en el proyecto. Esto condujo a una mayor comprensión del proyecto (Abelladan, 2020).

➤ **Inspección**

Scrum fomenta una serie de revisiones y desarrollos arqueológicos para identificar y corregir diferencias no deseadas. El Escaneo se realiza durante la ejecución del programa, scrum diario, ejecución y reparación (Abelladan, 2020).

➤ **Adaptación**

Cuando algo cambia, el equipo se ajustará para lograr el objetivo del éxito. Esta es la clave del éxito en el proyecto complejo donde la flexibilidad, la creatividad, el ingenio y la resiliencia son esenciales, junto con las necesidades cambiantes o la definición de su significado de manera efectiva (Abelladan, 2020).

2.11.4 Etapas de Scrum

Para comprender mejor como se manejan estos procesos, es importante saber cuáles son los pasos principales. Scrum es una compañía de producción basada en carreras bien diseñada. Cada proyecto promueve proyectos pequeña escala que ayudan a maximizar el impacto del proyecto más grande. Los principales pasos Scrum (Conexionesan, 2018).

➤ **Planificación del sprint**

Comprender el significado del sprint como pequeñas tareas en un gran proyecto tiene un propósito específico. Por ejemplo, el primer paso puede ser determinar cuánto se gastará el presupuesto, por lo que necesitará un equipo de expertos y experiencia en el campo de la economía (Conexionesan, 2018).

La primera reunión del grupo determinara áreas como desempeño, misión, riesgo de ejecución y tiempo de entrega. Luego se realizó una reunión entre el equipo y el gerente del proyecto para explicar cómo avanza cada intermedio. Aquí se evalúan cosas como evaluaciones, decisiones y actualizaciones (Conexionesan, 2018).

➤ **Etapa de Desarrollo**

Si la ejecución está en curso, el gerente debe asegurarse de que no haya cambios de última hora que puedan afectar al objetivo. Además, asegura que se cumplan las normativas especificadas en ese momento (Conexionesan, 2018).

➤ **Revisión de Sprint**

Al final de la brecha de desarrollo, los resultados se pueden analizar y evaluar. Si es necesario, todo el equipo trabajara en conjunto para encontrar áreas que necesiten cambios. En este caso, se fomenta la colaboración y las ideas entre todos. Estos puntos son:

- ❖ Colabora entre equipos, supervisores, gerentes y productores.
- ❖ La investigación externa se apoya como un proceso completo.
- ❖ Los grupos de trabajo abordan las deficiencias de desarrollo en curso.
- ❖ A partir de ahí, de este punto se puede regresar a la etapa de planificación y evaluación como un plan de mejora para su próxima carrera.
- ❖ Hasta ahora, esta revisión describe como el producto pudo crear valor adicional.
- ❖ Examine las capacidades, las horas de trabajo y otros datos de su equipo para descubrir que se necesita mejorar (Conexionesan, 2018).

➤ **Retroalimentación**

Los resultados se pueden traer para recibir ideas no solo de los profesionales del proyecto, sino también de las personas que usan lo que quieren lograr directamente. En otras palabras, clientes potenciales. Las lecciones aprendidas en este nivel harán que los juegos posteriores sean más efectivos y funcionales (Conexionesan, 2018)

3 CAPITULO 2: ANÁLISIS Y DISEÑO

3.1 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

La siguiente tabla nos muestra algunas de las características de sitios web meteorológicos que utilizaron un API, dándonos información de datos que se serán utilizadas en este proyecto.

3.Tabla

Análisis comparativo

Funcionalidades	INAHMI http://186.42.174.236/InamhiEmas/	Tropical Tidbits https://www.tropicaltidbits.com/	TuTiempo.net https://www.tutiempo.net/ecuador.html	AccuWeather https://www.accuweather.com/es/ec/
Consultar el clima actual de cualquier ciudad de mundo mediante un API		x	x	x
Permite ingresar sin necesidad de un login	x	x	x	x
Presentar datos históricos de las ciudades del Ecuador.			x	x

Consulta de datos con filtros de ciudad de ecuador, fecha inicio, fecha final, hora inicio y hora final	x		x	
Presentar una gráfica de la curva de los datos filtrados.	x	x	x	
Permite descargar los datos en diferentes formatos	x			x

Elaborado por: Ken Constante, Análisis comparativo de funcionalidades de sitios enfocados a datos meteorológicos.

4.Tabla

Requerimientos Funcionales

CÓDIGO	DESCRIPCIÓN
RF1	Consultar el clima actual de cualquier ciudad de mundo mediante un API.
RF2	Almacenar datos meteorológicos de todas las ciudades del Ecuador obtenidas desde un API en una base NoSQL (MongoDB Atlas).

RF3	Presentar datos históricos con filtros de ciudad de Ecuador, fecha inicio, fecha final, hora inicio y hora final
RF4	Presentar una gráfica de la curva de la temperatura, sensación térmica, precipitación, humedad, capa de nubes, velocidad de viento filtrada.
RF5	Almacenar información desde archivos con los siguientes formatos (csv, json, xlsx, xml, txt) dentro de la base.
RF6	Descargar datos gráficos meteorológicos en formato PNG, JPGE, PDF, SVG, CSV, XLS

Elaborado por Ken Constante RF.

5.Tabla

Requerimientos NO Funcionales

CÓDIGO	DESCRIPCIÓN
RNF1	Crear una interfaz intuitiva para el usuario.
RNF2	Utilizar modelo MVC (modelo, vista, controlador) para manejar la arquitectura del sistema.
RNF3	Utilizar MongoDB Atlas como una base no relacional.

RNF4	El sistema debe ser escalable para que una estación meteorológica pueda adecuarlo a sus necesidades y siga funcional.
RNF5	El sistema debe tener un correcto rendimiento para una mejor experiencia de usuario.
RNF6	Mantener la confiabilidad de la información presentada.
RNF7	Tener una disponibilidad permanente para una mejor interacción con los usuarios
RNF8	Cualquier usuario con conexión a internet puede ingresar al sitio web
RNF9	Utilizar Node.js como backend en el desarrollo del sistema.
RNF10	El sistema debe funcionar en los navegadores web (Google Chrome, Firefox, Microsoft Edge)

Elaborado por Ken Contante, RNF.

3.2 HISTORIAS DE USUARIO

6.Tabla

Historia de Usuario 1

Código Requerimiento	RF1
Actor	Usuario
Flujo Normal	<ol style="list-style-type: none">1. El usuario accede al portal web2. La aplicación carga los datos correspondientes en la página principal3. El usuario elige la ciudad de su interés4. La aplicación muestra el clima actual de la ciudad elegida indicando la temperatura, sensación térmica, precipitación, humedad, capa de nubes, velocidad de viento.
Flujo Alternativo	<ol style="list-style-type: none">2.a La aplicación no puede conectarse con el servidor web y emite un mensaje de información al usuario indicando el problema4.a La aplicación tiene problemas de conexión con el API, el sistema emite un mensaje de advertencia al usuario

Elaborado por Ken Constante, Historia de Usuario 1.

7.Tabla

Historia de Usuario 2

Código Requerimiento	RF2
Actor	Administrador
Flujo Normal	<ol style="list-style-type: none">1. El administrador entra a la URL del sistema.2. El administrador investiga la información meteorológica en una API.3. Descarga los datos de dicha API4. El administrador del sitio web carga los datos meteorológicos de cada ciudad en el sistema
Flujo Alternativo	<ol style="list-style-type: none">3.a La API utilizada no permite descargar los datos meteorológicos4.a El administrador tiene problemas con subir los datos al sitio web

Elaborado por Ken Constante, Historia de Usuario2.

8.Tabla

Historia de Usuario 3

Código Requerimiento	RF3
Actor	Usuario
Flujo Normal	<ol style="list-style-type: none">1. El usuario accede al portal web2. La aplicación carga los datos correspondientes en la página principal3. El usuario elige la ciudad de su interés4. El usuario elige la fecha y hora de inicio y fin de su interés.5. El usuario elige el tipo de dato meteorológico que desea consultar.6. La información carga los datos meteorológicos en la página principal.
Flujo Alternativo	<ol style="list-style-type: none">2.a La aplicación no puede conectarse con el servidor web y emite un mensaje de información al usuario indicando el problema3.a El usuario no elige una ciudad de Ecuador6.a La aplicación no tiene conexión para mostrar los datos meteorológicos correspondientes.

Elaborado por Ken Constante, Historia de Usuario3.

9.Tabla

Historia de Usuario 4

Código Requerimiento	RF4
Actor	Usuario
Flujo Normal	<ol style="list-style-type: none">1. El usuario accede al portal web2. La aplicación carga los datos correspondientes en la página principal3. El usuario elige el tipo de dato meteorológico del que desea la gráfica4. La aplicación carga la información y muestra en la página principal
Flujo Alternativo	<ol style="list-style-type: none">2.a La aplicación no puede conectarse con el servidor web y emite un mensaje de información al usuario indicando el problema3.a El usuario desea una sola gráfica con todos los datos meteorológicos4.a La aplicación tiene problemas de conexión a internet y no carga la información requerida

Elaborado por Ken Constante, Historia de Usuario 4.

10.Tabla

Historia de Usuario 5

Código Requerimiento	RF5
Actor	Usuario
Flujo Normal	<ol style="list-style-type: none">1. El usuario accede al portal web2. La aplicación carga los datos correspondientes en la página principal3. El usuario elige la opción de subir información4. La aplicación muestra la ventana, para elegir el archivo que sea subir5. El usuario selecciona el archivo en formato (csv, json, xlsx, xml, txt)
Flujo Alternativo	<ol style="list-style-type: none">2.a La aplicación no puede conectarse con el servidor web y emite un mensaje de información al usuario indicando el problema5.a El usuario selecciona un archivo diferente a los permitidos, el sistema le mostrara un mensaje de error de formato seleccionado

Elaborado por Ken Constante, Historia de Usuario5.

11.Tabla

Historia de Usuario 6

Código Requerimiento	RF6
Actor	Usuario
Flujo Normal	<ol style="list-style-type: none">1. El usuario accede al portal web2. La aplicación carga los datos correspondientes en la página principal3. El usuario elige el tipo de dato que desea4. La aplicación carga la gráfica del tipo de dato seleccionado5. El usuario selecciona el tipo de formato en el que desea descargar la gráfica de datos meteorológicos6. La aplicación descarga en el ordenador la gráfica en el formato seleccionado
Flujo Alternativo	<ol style="list-style-type: none">2.a La aplicación no puede conectarse con el servidor web y emite un mensaje de información al usuario indicando el problema5.a La aplicación no tiene el formato que el usuario desea descargar

Elaborado por Ken Constante, Historia de Usuario6.

3.3 DIAGRAMAS DE CASO DE USO

2. Figura

Diagrama de caso 1: RF1

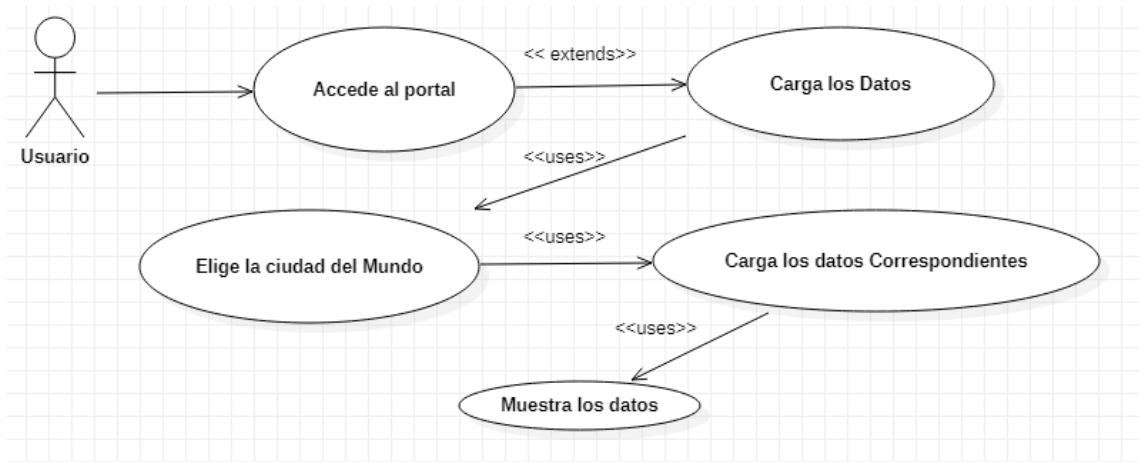


Diagrama de caso de uso RF1. Elaborado por Ken Contante

3. Figura

Diagrama de caso 2: RF2

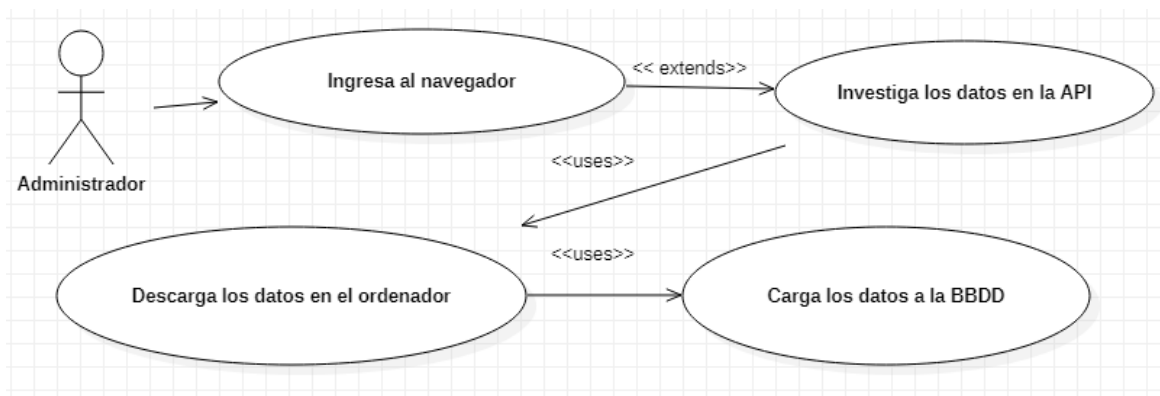


Diagrama de caso de uso RF2. Elaborado por Ken Contante

4. Figura

Diagrama de caso 3: RF

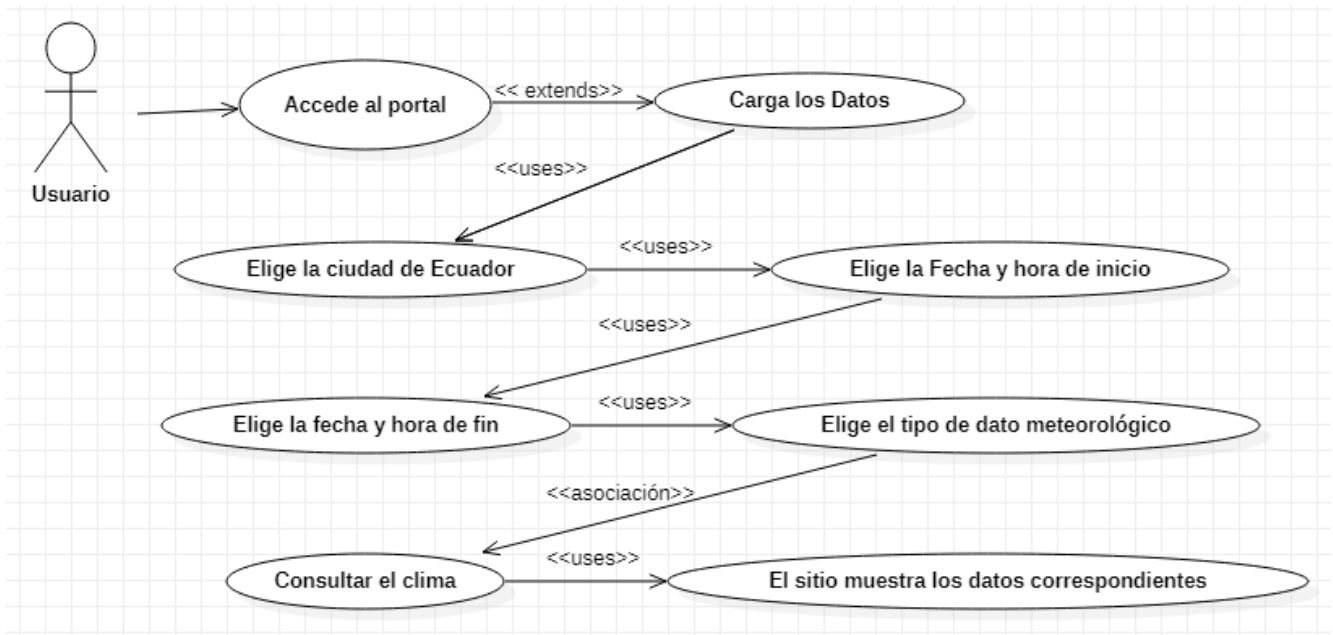


Diagrama de caso de uso RF3. Elaborado por Ken Contante

5. Figura

Diagrama de caso 4: RF4

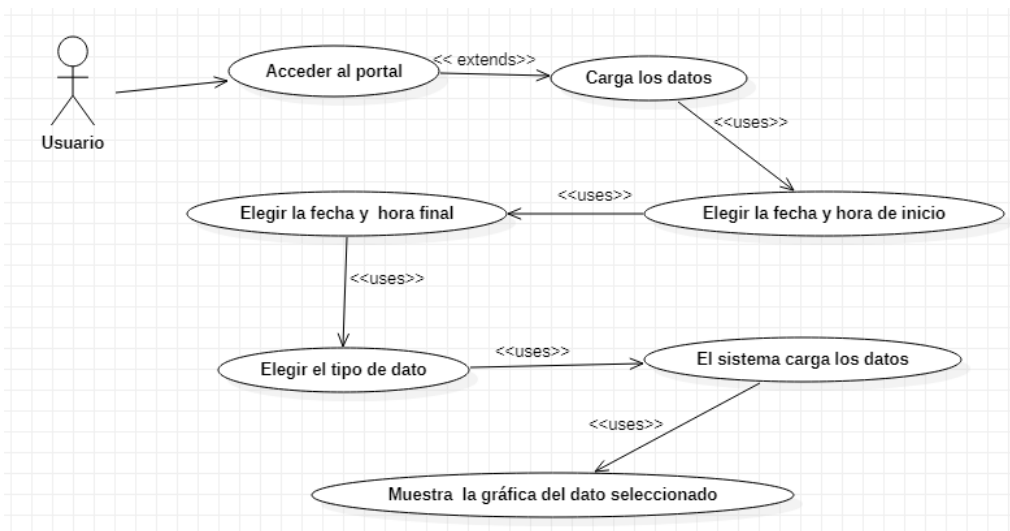


Diagrama de caso de uso RF4. Elaborado por Ken Contante

6. Figura

Diagrama de caso 5: RF5

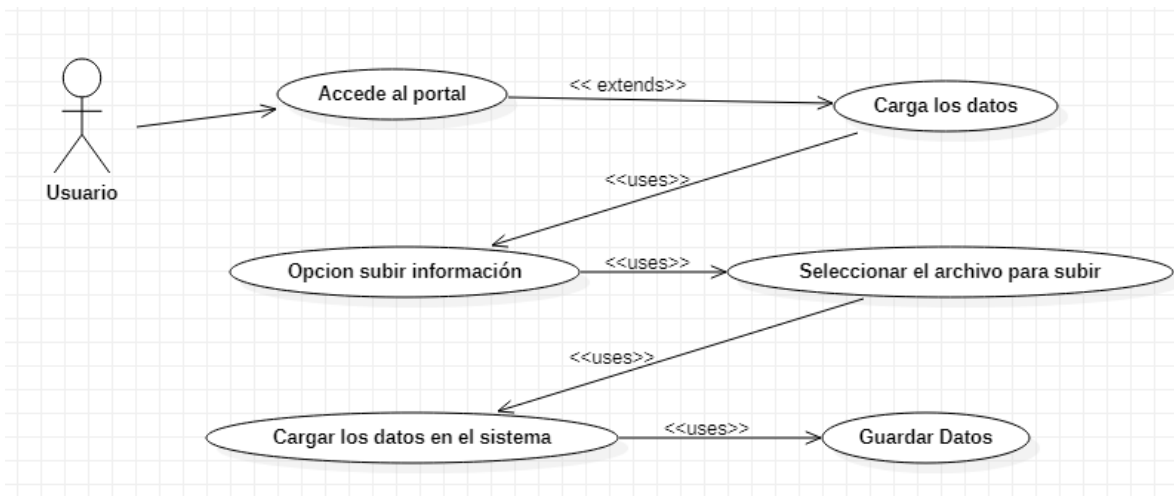


Diagrama de caso de uso RF5. Elaborado por Ken Contante

7. Figura

Diagrama de caso 6: RF6

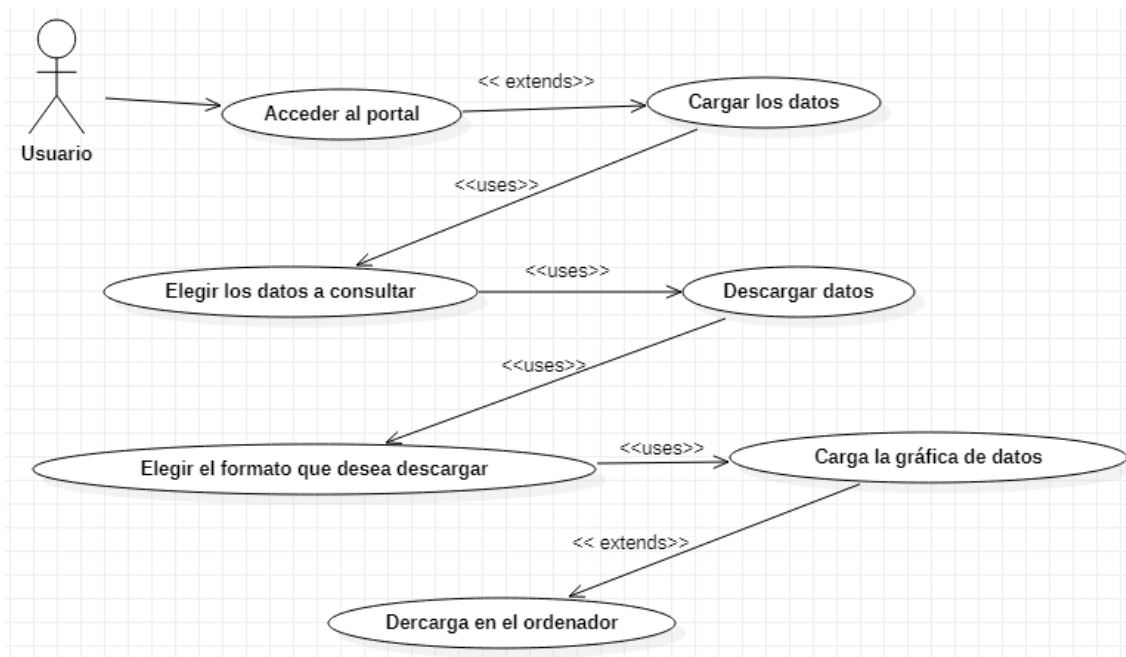


Diagrama de caso de uso RF6. Elaborado por Ken Contante

3.4 DIAGRAMA DE SECUENCIAS

8. Figura

Diagrama de Secuencia 1

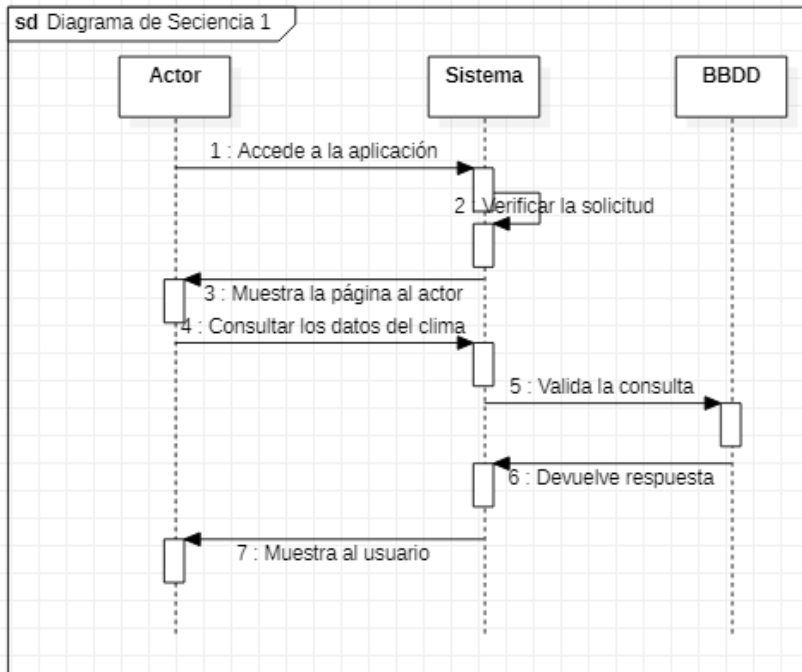


Diagrama de Secuencia1. Elaborado por Ken Constante

9. Figura

Diagrama de Secuencia 2

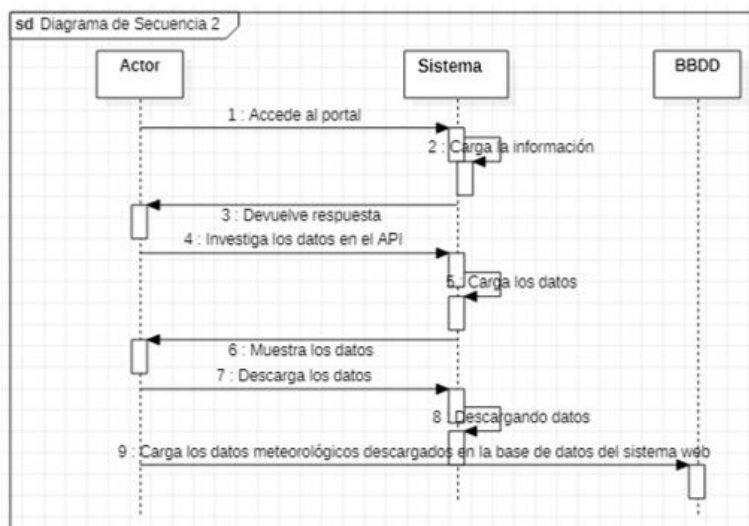


Diagrama de Secuencia2. Elaborado por Ken Constante

10. Figura

Diagrama de Secuencia 3

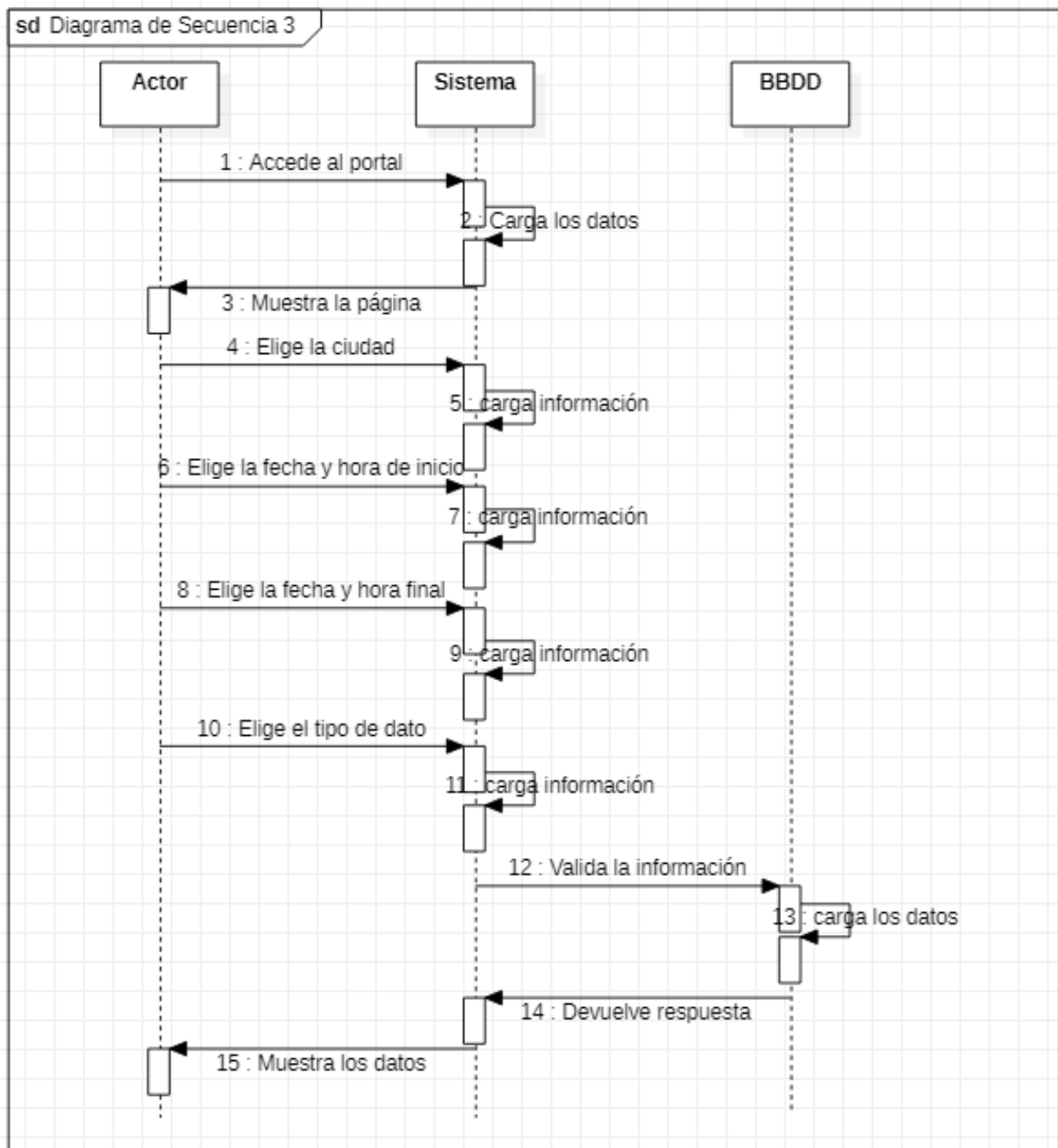


Diagrama de Secuencia3. Elaborado por Ken Constante

11. Figura

Diagrama de secuencia 4

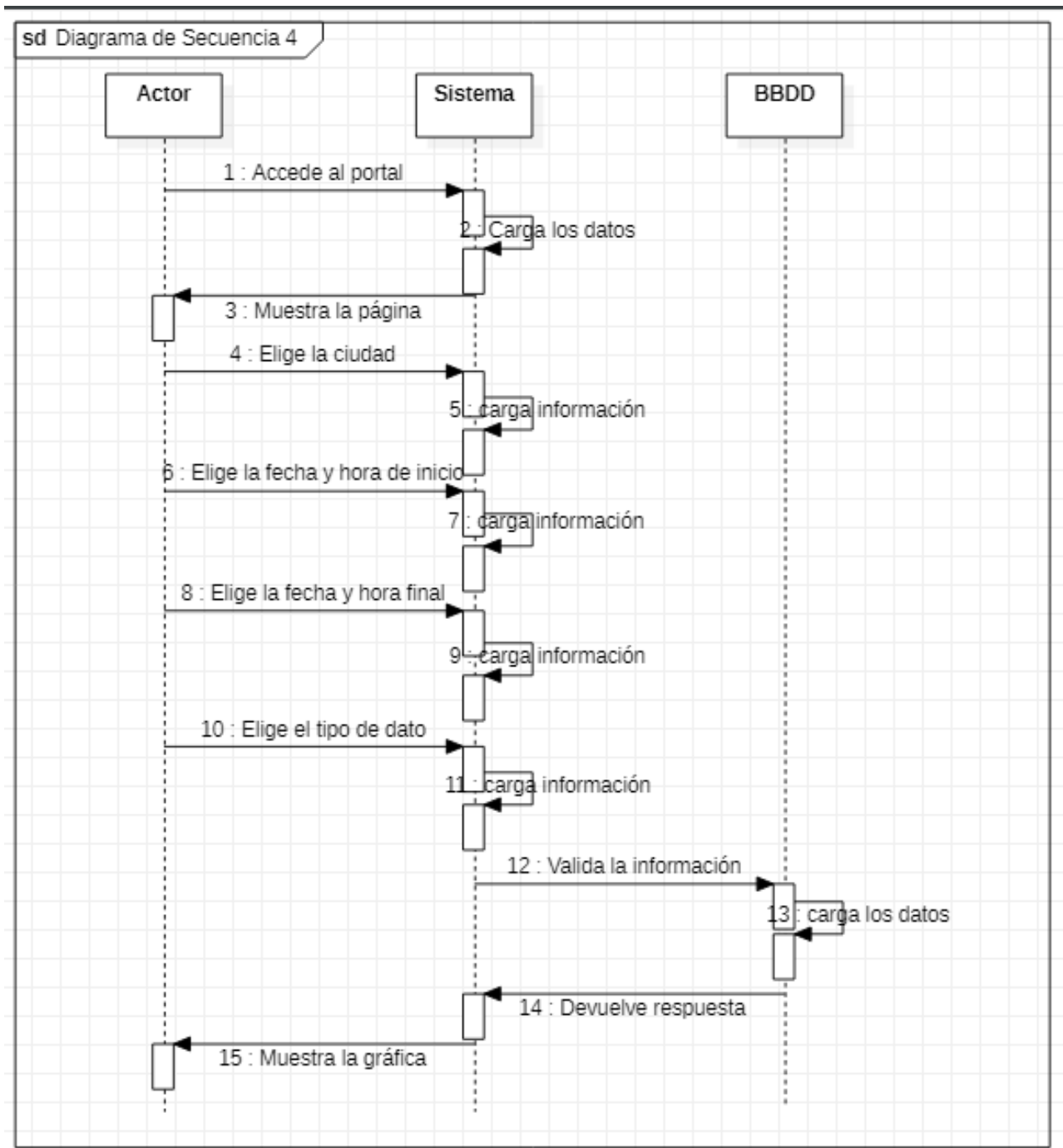


Diagrama de Secuencia4. Elaborado por Ken Constante

12. Figura

Diagrama de Secuencia 5

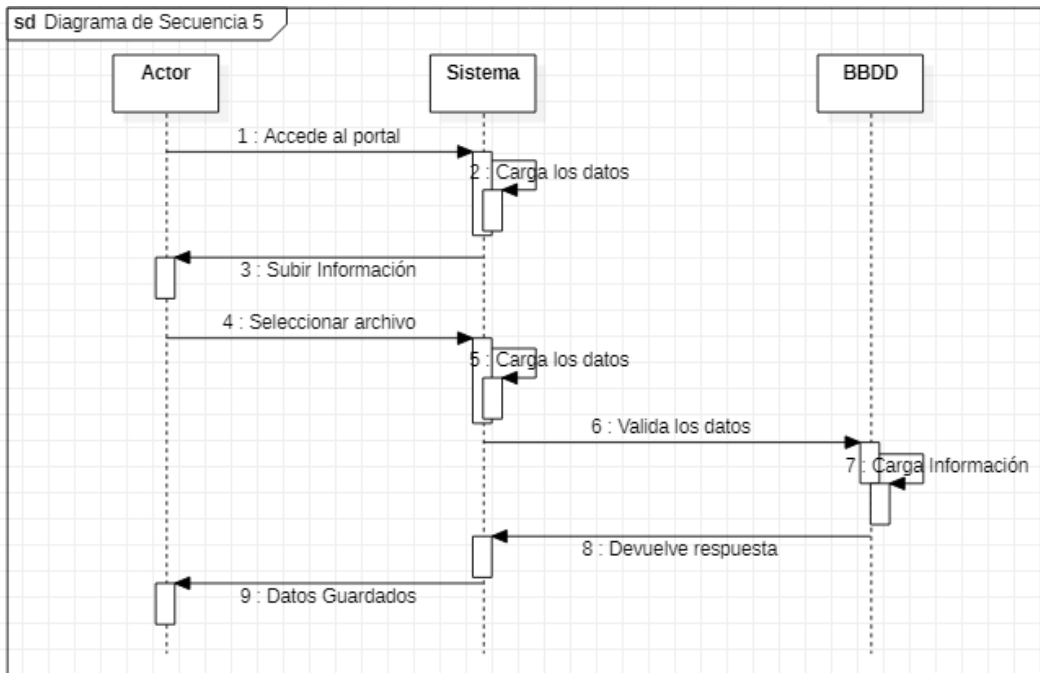


Diagrama de Secuencia5. Elaborado por Ken Constante

13. Figura

Diagrama de Secuencia 6

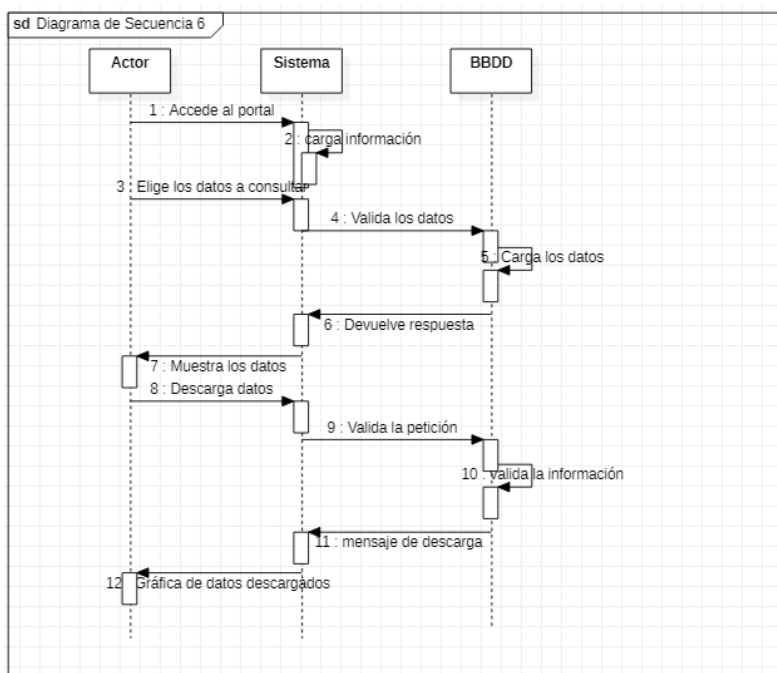


Diagrama de Secuencia6. Elaborado por Ken Constante

3.5 MODELO CONCEPTUAL DE LA BBDD

14. Figura

Diagrama de la base de datos.

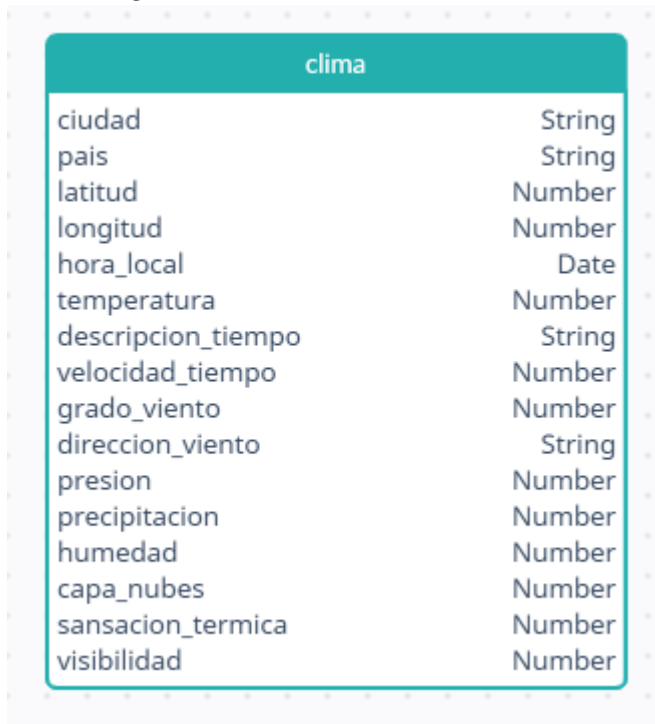


Diagrama de la base de datos para la tabla 'clima'. La tabla contiene los siguientes campos y tipos de datos:

clima	
ciudad	String
pais	String
latitud	Number
longitud	Number
hora_local	Date
temperatura	Number
descripcion_tiempo	String
velocidad_tiempo	Number
grado_viento	Number
direccion_viento	String
presion	Number
precipitacion	Number
humedad	Number
capa_nubes	Number
sansacion_termica	Number
visibilidad	Number

Diagrama de la base de datos. Elaborado por Ken Constante

4 CAPUTULO 3: CONSTRUCCIÓN Y PRUEBAS

En este apartado se va a aclarar los aspectos principales que corresponden al aplicativo, la programación, los métodos, las configuraciones y el funcionamiento correcto.

4.1 ARQUITECTURA DE LA PROPUESTA

4.1.1 *Modelo Cliente-Servidor*

Es uno de los modelos más utilizados para servicios y protocolos de internet está formado por, una parte del servidor y la otra parte por el cliente donde se espera las respuestas del servidor el cual actúa como un depósito de datos y también como un sistema de gestión de aplicaciones y bases de datos.

El sistema se creó en NodeJS, el cual se encarga de gestionar y utilizar la base NoSQL MongoDB y para la creación y diseño de la página web se utiliza Bootstrap.

Para la ejecución del sistema y con las siguientes características el sistema trabaja de manera correcta:

Equipo:

- Computador: escritorio/portátil
- Procesador: Intel CORE i3
- Memoria: 8 GB en RAM
- Disco: 1000 GB
- Sistema: Windows 10 pro-64 bits

El aplicativo funciona por medio de un API el cuál con un apikey permite al aplicativo extraer los datos: ciudad, temperatura, fecha, térmica, humedad, icono, descripción, velocidad del viento, presión, precipitación, capa de nubes y estos almacenarlos en una

variable para posteriormente usarlos en la pantalla principal como resultado, de esta manera mostrando al usuario el resultado de su consulta.

15. Figura

Diagrama modelo cliente-servidor.

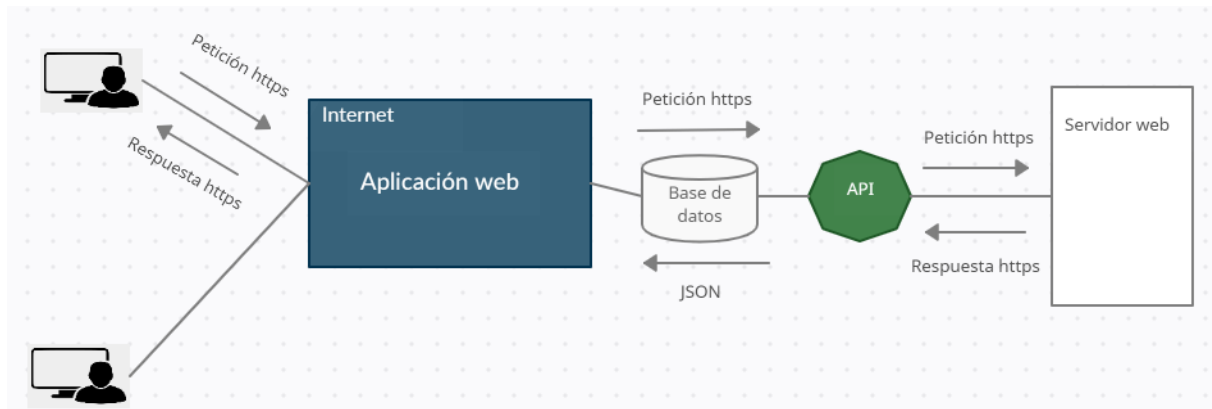


Diagrama modelo cliente-servidor. Elaborado por Ken Constante

En el diagrama se puede observar cómo está levantada la arquitectura, al tratarse de una aplicación bajo Node.js se estableció una conexión remota con Heroku para levantar la aplicación, así la base de datos recibe los datos del API para guardarlos en variables y mediante los métodos establecidos en el back-end plasmarlos en el front-end.

El sistema cuenta con dos módulos: módulo de usuario y de administrador, el módulo del usuario siendo la interacción de este con el sistema y el de administrador presentando el nivel de jerarquía que tiene con el sistema, a continuación, se procede a describir los dos módulos detalladamente.

4.1.2 Módulo usuario

Consulta por ciudad: Aquí el usuario consulta datos meteorológicos de acuerdo con el clima de su interés.

Consulta por ciudad, fecha y hora de inicio, tipo de dato meteorológico: Aquí el usuario consulta los datos meteorológicos de una manera más específica ingresando variables adicionales de su interés.

Consulta gráfica por tipo de dato meteorológico: Aquí el usuario puede consultar una gráfica del tipo de dato meteorológico en específico.

Subir información: Aquí el usuario puede subir un archivo en los formatos establecidos (csv, json, xlsx, xml, txt).

Descarga gráfica de datos meteorológicos: Aquí el usuario puede descargar la información consultada seleccionando unos parámetros de su interés.

4.1.3 Modulo web

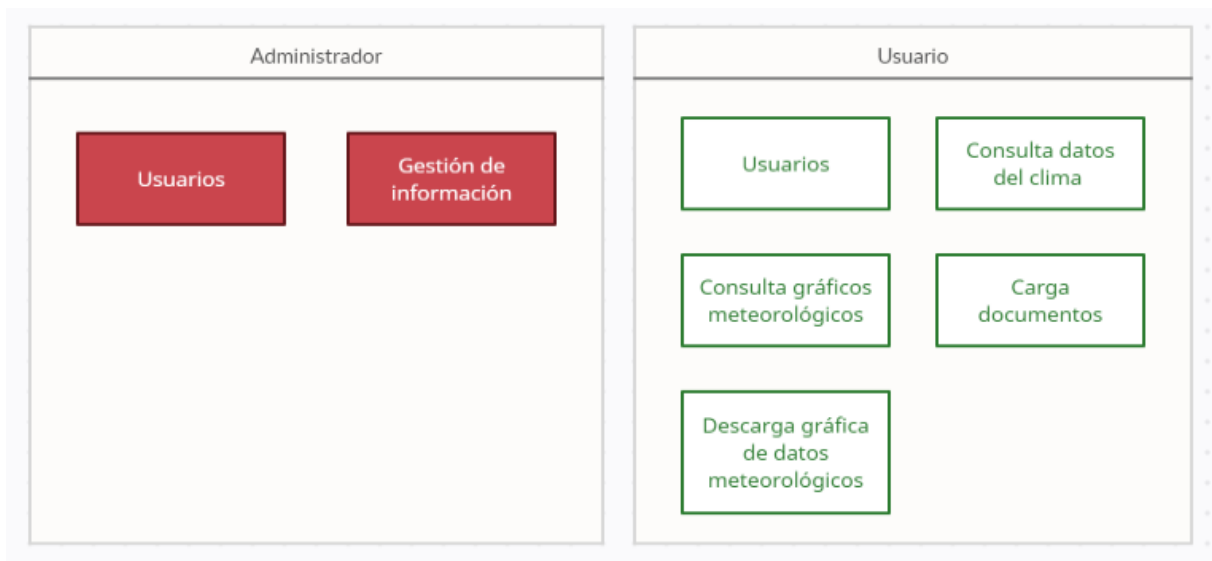
Administrador: En este modelo se permite la dirección de la información, alusivo a las variables del clima.

4.1.4 Diagrama de bloques general

Con relato a las solicitudes y los esquemas de casos de uso se puede definir al procesamiento en dos módulos: administración y usuario. Cada uno de estos módulos tiene pequeños funcionamientos planteados en la configuración de la ejecución.

16. Figura

Descripción de diagrama de bloques



Descripción de diagrama de bloques. Elaborado por Ken Constante

4.2 DISEÑO DE INTERFACES

La interfaz gráfica de este procedimiento meteorológico está trazada de manera que se presente el clima amistoso y de una forma fácil. Implantar ciertos principios básicos como son la sencillez de uso, la facilidad y la seguridad en cada parte del sistema, también se indaga en una idea de tener transparencia visual en cada sector gráfico, es decir colocando de manera valida los textos y botones que se presentan en la interfaz, ya que si no se tiene en cuenta estos aspectos la vista del usuario termina sobre cargado y no es del gusto del usuario, la idea es cautivar al beneficiario con un diseño muy favorable para él.

4.2.1 Diseño de interfaces del usuario web

Para el diseño de la interfaz web se determinó la presentación del clima con los datos climáticos muy bien representados con el objetivo de mostrar la información centralizada y que el navegante capte estos datos de manera correcta y eficiente.

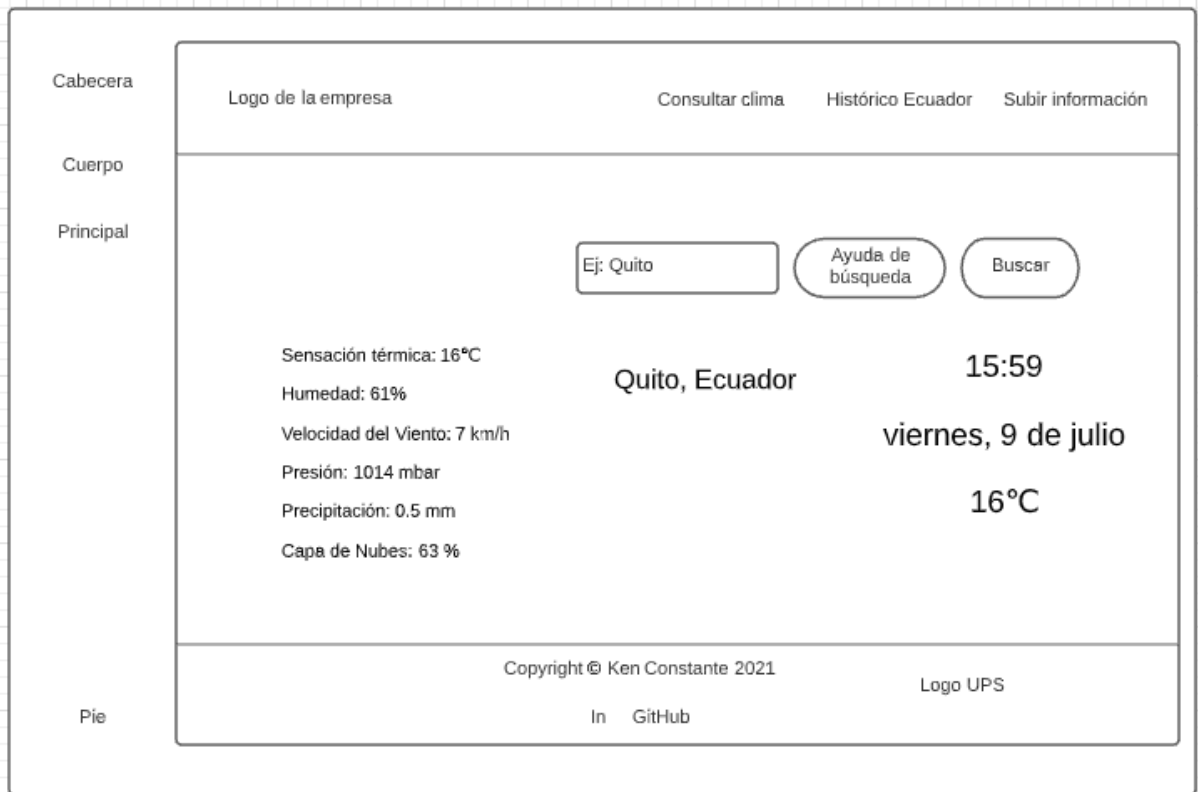
4.2.2 *Diseño interfaz principal*

Pantalla principal (Consultar clima):

- **Menú (header):** Consta de 3 botones, uno para realizar la consulta del clima, otro para ver los datos detalladamente y otro para poder subir nuestra información a la página.
- **Campo buscar:** Aquí se ingresa el parámetro para consultar el clima ejemplo: Quito.
- **Botón buscar:** Este apartado inicializará la búsqueda del parámetro ingresado en el campo del texto.
- **Botón ayuda a buscar:** Este botón activa la opción del autocompletado en el campo buscar.
- **Datos meteorológicos:** Aquí se presenta los datos Sensación térmica, Humedad, Velocidad del Viento, Presión, Precipitación, Capa de Nubes.
- **Ubicación (Por defecto):** En este apartado se muestra en una etiqueta la ubicación en donde estas situado, Quito, Ecuador.
- **Hora, fecha y grados centígrados:** Aquí encontramos la información acerca de la hora, fecha y los grados centígrados del lugar donde estemos en este momento.

17. Figura

Interfaz pantalla principal (consultar clima)



Interfaz pantalla principal (consultar clima). Elaborado por Ken Constante

4.2.3 Diseño interfaz consulta

Pantalla principal (Histórico Ecuador):

- **Menú (header):** Consta de 3 botones, uno para realizar la consulta del clima, otro para ver los datos detalladamente y otro para poder subir nuestra información a la página.
- **Ubicación (Por defecto):** En este apartado se muestra en una etiqueta la ubicación en donde estas situado, Quito, Ecuador.
- **Campo ciudad:** Aquí seleccionaremos una ciudad de nuestro interés, para utilizarla como parámetro de búsqueda.
- **Fecha, hora inicio:** En este apartado seleccionaremos la fecha y la hora de inicio en la que queremos consultar el clima.

- **Fecha, hora final:** Por este lado seleccionaremos la fecha y hora de fin para seleccionar un parámetro de inicio y fin en la búsqueda.
- **Botón buscar:** Con el botón buscar daremos la funcionalidad a esta búsqueda, inicializándola.
- **Botón ayuda a buscar:** Este botón activa la opción del autocompletado en el campo buscar.
- **Label con los parámetros ingresados:** Este es un mensaje que nos mostrara como queda la información con los parámetros ingresados.
- **Spinner con los tipos de datos:** Esta nos brindará una lista de opciones de las cuales podemos sacar un gráfico referido a esos datos.
- **Gráfico de datos:** Posteriormente seleccionaremos un tipo de datos de nuestro interés para imprimir en su parte de abajo un gráfico con información acerca de ese dato.
- **Media de datos:** Se presenta una estadística de los datos del histórico en un cuadro con sus respectivas etiquetas.
- **Boton descargar:** Este botón inicializa la acción de descargar estos datos.
- **Datos Históricos:** Aquí se muestra todos los datos que están en el histórico de este sistema para verlos detalladamente.

18. Figura

Interfaz pantalla principal (Histórico Ecuador)

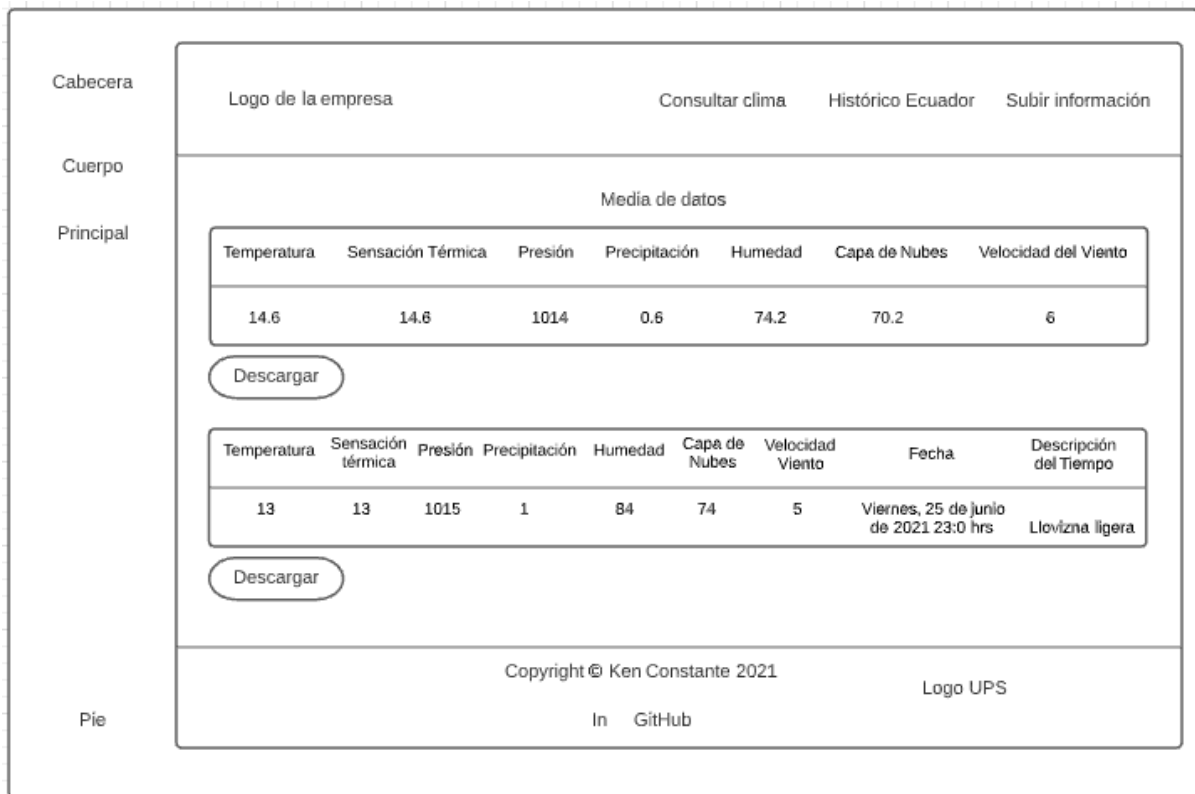
The wireframe shows a web interface with the following layout:

- Cabecera:** Logo de la empresa, Consultar clima, Histórico Ecuador, Subir información.
- Cuerpo Principal:**
 - Consulta de Datos Históricos
 - Quito, Ecuador
 - Ciudades: Seleccione una Ciudad del Ecuador
 - Fecha y Hora de Inicio: dd/mm/aaaa --:--
 - Fecha y Hora Final: dd/mm/aaaa --:--
 - Buscar
 - Desde 6/25/2021, 7:00:00 PM hasta 6/25/2021, 11:00:00 PM
 - Tipo de datos: Selecciona una opción
 - Gráfico
- Pie:** Copyright © Ken Constante 2021, Logo UPS, In GitHub.

Interfaz pantalla principal (Histórico Ecuador). Elaborado por Ken Constante

19. Figura

Interfaz pantalla principal (Histórico Ecuador)



Interfaz pantalla principal (Histórico Ecuador). Elaborado por Ken Constante

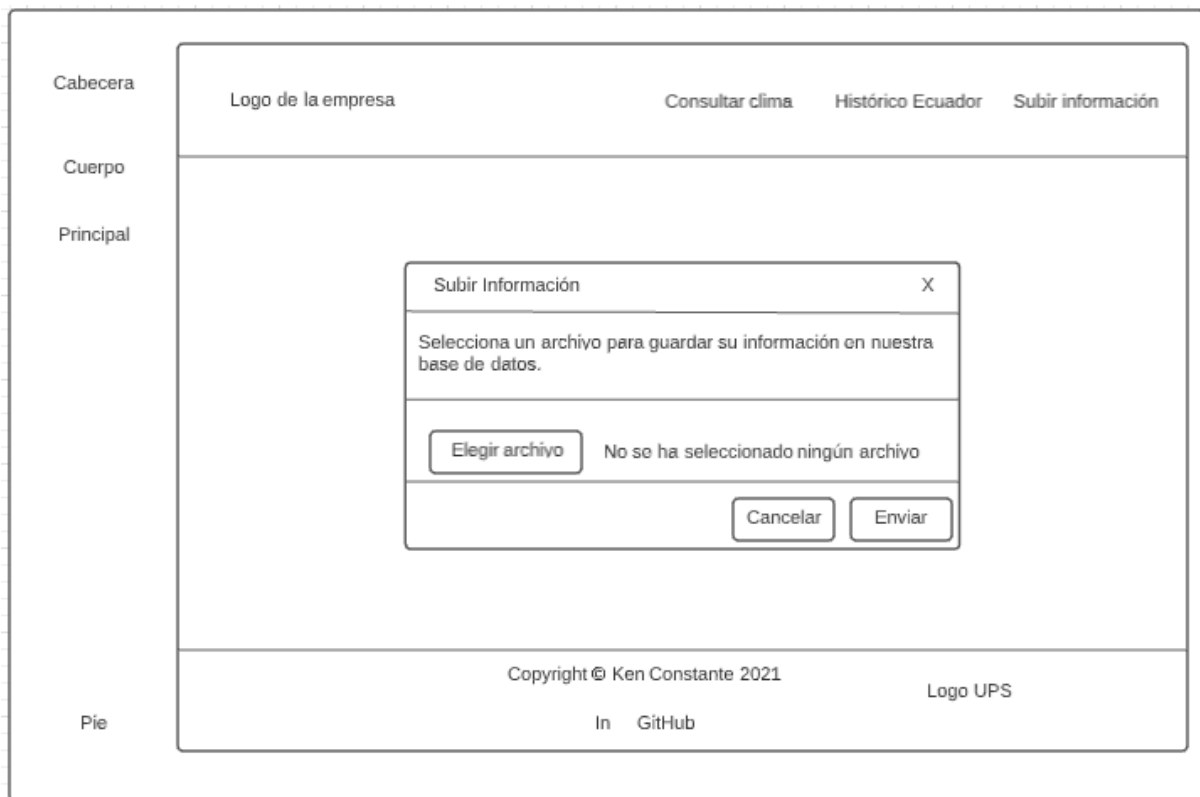
4.2.4 Diseño interfaz subir información

Pantalla principal (Subir información):

- **Menú (header):** Consta de 3 botones, uno para realizar la consulta del clima, otro para ver los datos detalladamente y otro para poder subir nuestra información a la página.
- **Botón Subir información:** Al accionar este botón con un clic izquierdo, este desplegará una ventanita a modo mensaje de manera que le muestre al usuario la opción para subir la información y almacenarla en la base de datos.
- **Botón elegir archivo:** Con este botón inicializamos el explorador de archivos de nuestra computadora para seleccionar el archivo a subir.

20. Figura

Interfaz pantalla principal (Histórico Ecuador)



Interfaz pantalla principal (Histórico Ecuador). Elaborado por Ken Constante

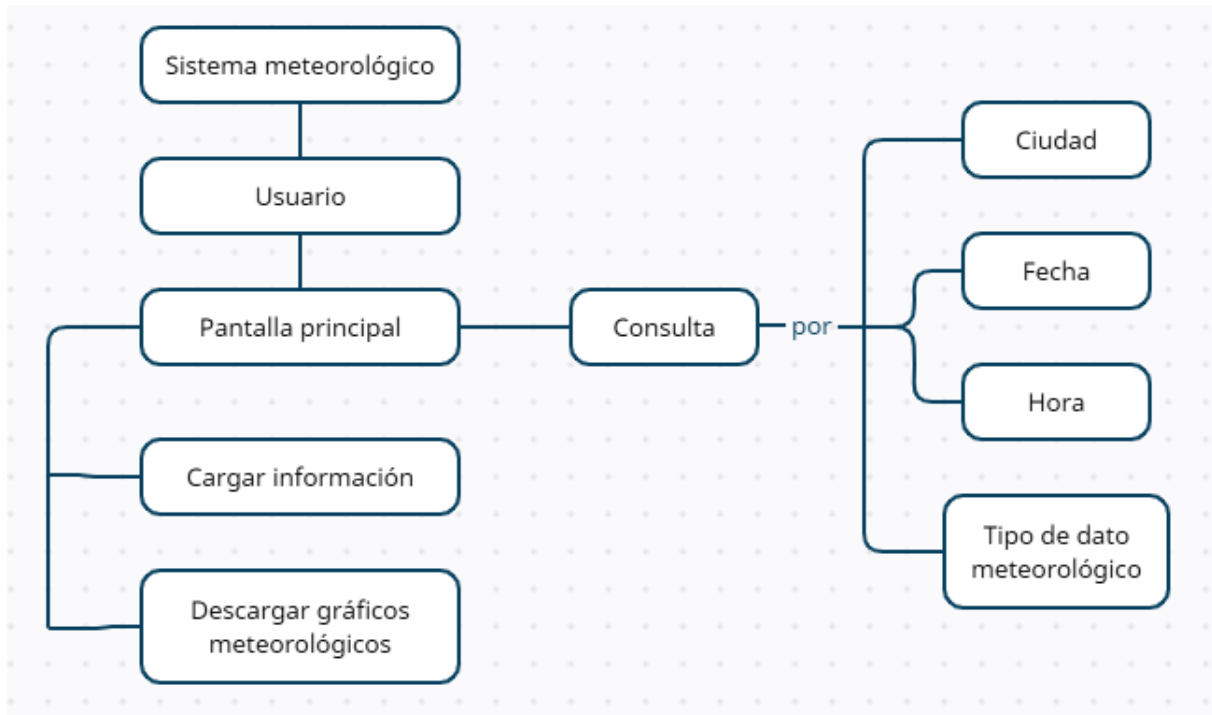
4.3 DIAGRAMA DE NAVEGACIÓN

El mapa de navegación que presenta el aplicativo, es muy sencillo, se mostrará a continuación como navegaría un usuario común y corriente para consultar los datos del clima, los gráficos del clima, para cargar documentos y para descargar los gráficos con los datos meteorológicos.

De esta manera se crea una orientación al usuario para que su recorrido por la estructura propuesta sea muy amigable y sencilla de usar.

21. Figura

Mapa de navegación del aplicativo



Mapa de navegación del aplicativo. Elaborado por Ken Constante

4.4 ARQUITECTURA DEL SISTEMA

El sistema meteorológico en cuestión está formado por dos complementos:

- Heroku: Plataforma de despliegue
- Base de datos: Almacenamiento de la información

4.4.1 Heroku

Es una plataforma alojada en la nube la cual ofrece servicios denominados como “PaaS” o “Platform as a service”, la facilidad de uso y versatilidad en diferentes desarrollos la ha hecho acreedora a una gran popularidad en los últimos años. Las compañías han seleccionado a Heroku como PaaS por su facilidad y fuerte enfoque para diferentes tipos de despliegue de aplicaciones. También permite un mejor manejo de los servidores y configuraciones de este, administración y escalamiento. A esta plataforma solo se le brinda el lenguaje o la base que se utiliza.

4.4.2 Base de datos No SQL: MongoDB Atlas Data base

El modelo de datos a implementarse en este sistema meteorológico almacenará los datos de ciudad, temperatura, fecha, térmica, humedad, icono, descripción, velocidad del viento, presión, precipitación, capa de nubes, extraídos de una API que proporcionará toda esta información, dado que se trata de una base de datos NoSQL es una base distribuida, basada en documentos y que su uso es moderno y dedicado a aplicaciones con almacenamiento en la nube.

4.5 DIAGRAMA DE DESPLIEGUE

4.5.1 Como se comunica la aplicación entre capas

Es un diagrama el cual cuenta con una estructura la cual nos muestra la arquitectura del sistema desde un punto de vista orientado al despliegue de los implementos del software.

22. Figura

Diagrama de despliegue entre arquitecturas

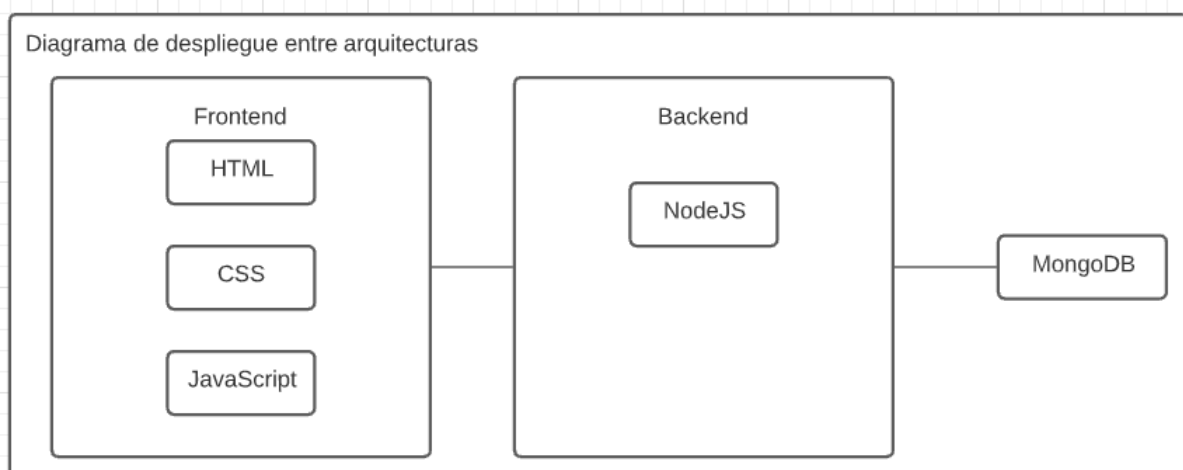


Diagrama de despliegue entre arquitecturas. Elaborado por Ken Constante

Gracias al gráfico de despliegue de arquitecturas se puede apreciar, se distribuye en tres secciones llamadas nodos, el primero es el frontend o la vista, el segundo el backend donde está desarrollado bajo Node.js y la base de datos MongoDB, que en conjunto permiten desplegar la aplicación.

23. Figura

Diagrama de despliegue entre redes

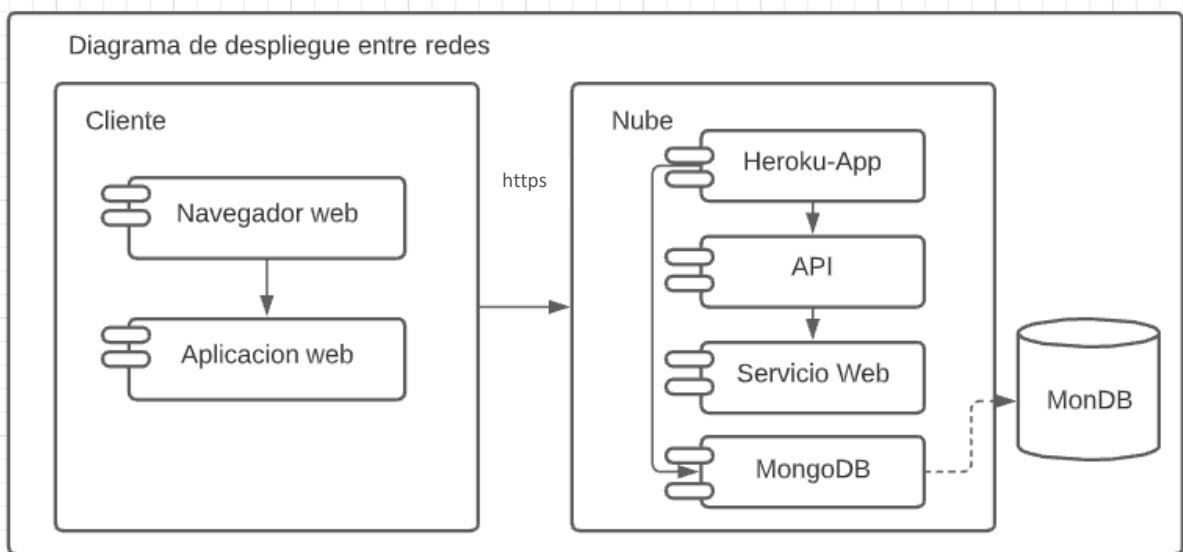


Diagrama de despliegue entre redes. Elaborado por: Ken Constante

El diagrama de despliegue entre redes permite dar a conocer un enfoque más detallado de cómo se realiza el despliegue en la arquitectura cliente-servidor, principalmente el cliente está formado por el navegador web donde se aloja la aplicación por la cual ingresará las peticiones https el cliente, por donde se conectará a la nube para acceder al servidor donde esta alojada la aplicación remotamente, es decir, donde las peticiones serán atendidas por medio de la aplicación con los recursos del API.

4.6 ALGORITMOS

Para sistemas meteorológicos se utilizan algoritmos de ramificación, esto como objetivo de realizar una demostración de forma más fácil, de esta forma se contemplará todo el proceso que realizará el sistema web en sus diferentes etapas.

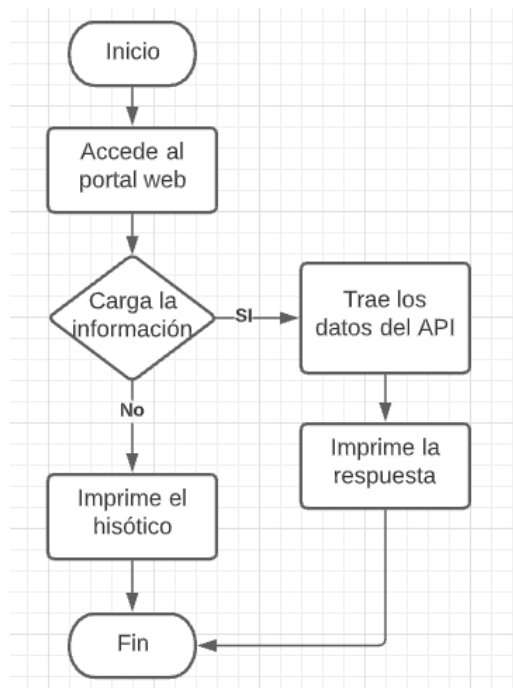
- **Algoritmos de ramificación**

Este tipo de algoritmo está representado por una condición, si una condición es verdadera, el algoritmo cumplirá una sentencia, si la condición es falsa, el algoritmo cumplirá una sentencia diferente, por tal motivo se usará algoritmos de ramificación para explicar los métodos utilizados dentro del sistema.

- **Algoritmo de la aplicación web**

24. Figura

Algoritmo de la aplicación web



Algoritmo de la aplicación web. Elaborado por Ken Constante

Para realizar un pedido en la aplicación web, primero se ingresa la información en el campo ciudad posteriormente la aplicación llama al procedimiento `getClima ()` el cuál mediante una señal espera la respuesta por parte del API para traer la información a la base de datos y guardarlos en variables para usarlos en la interfaz, de acuerdo con el factor que ingreso el usuario.

En el siguiente bloque de código se obtendrá la ciudad y los ítems que cargan con la vista principal, se situó en el `server.js`.

25. Figura

Bloque de código aplicación web principal

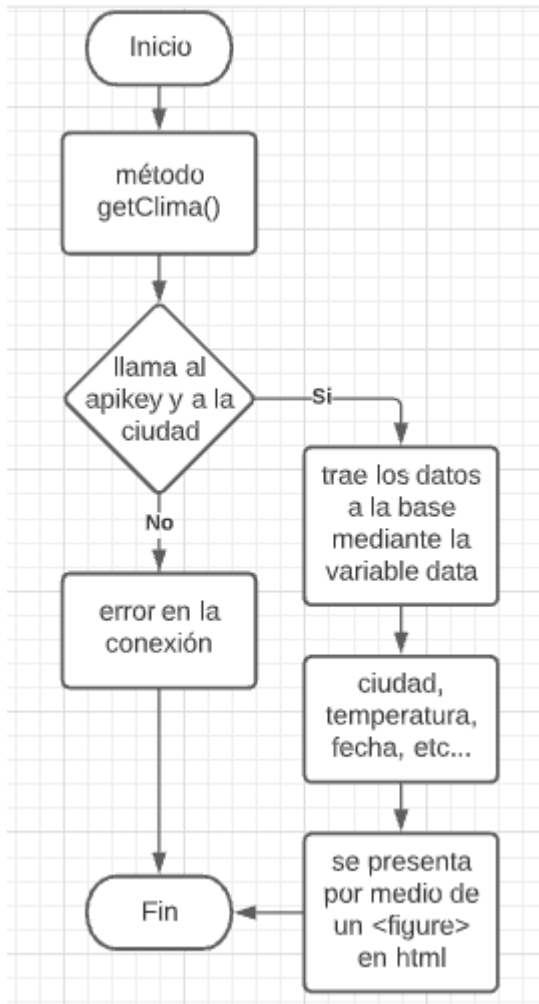
```
const apikey = '32030e015cdf9bf639d86120f39a4dda';
let data;
const getClima = async(ciudad) => {
  try {
    const ciudadURI = encodeURI(ciudad);
    const respuesta = await axios.get(`http://api.weatherstack.com/current?access_key=${apikey}&query=${ciudadURI}`);
    data = {
```

Bloque de código aplicación web principal. Elaborado por Ken constante

- *Algoritmo de la consulta del clima por ciudad, fecha y hora en la aplicación web*

26. Figura

Algoritmo de la consulta del clima por ciudad, fecha y hora en la aplicación web



Elaborado por Ken Constante

Con la implementación de una búsqueda más precisa se implementó la búsqueda con la ciudad por hora y fecha, el cuál sigue el proceso del algoritmo anterior ya que, el mismo método `getClima ()` llama al ápice con la ciudad y los parámetros adicionales para revisar los datos almacenados en la base y verlos representados mediante un `<figure>` en HTML de acuerdo al tipo de datos, como se puede ver en el siguiente bloque de código.

27. Figura

Bloque de código búsqueda por clima, hora y fecha

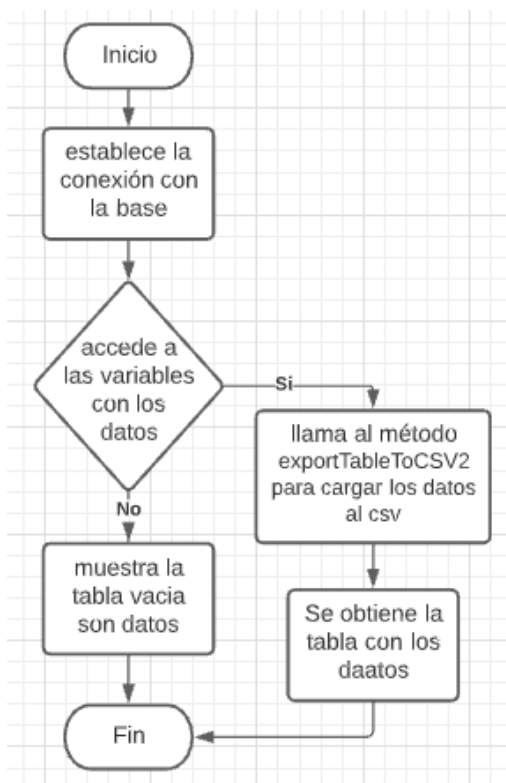
```
app.post('/consultaBase', (req, res) => {  
  let inicio = req.body.fecha_inicio;  
  let final = req.body.fecha_final;  
  ciudad_q = req.body.provincia;  
  if (ciudad_q === "Selecciona una Ciudad del Ecuador") {  
    ciudad_q = "Quito";  
  }  
  Clima.find({  
    ciudad: ciudad_q,  
    hora_local: {  
      $gte: inicio,  
      $lte: final  
    }  
  })  
})
```

Bloque de código búsqueda por clima, hora y fecha. Elaborado por Ken Constante

- *Algoritmo de la consulta del clima para descargar los datos de la página web*

28. Figura

Algoritmo de la consulta del clima para descargar los datos de la página web



Algoritmo de la consulta del clima para descargar los datos de la página web. Elaborado por

Ken Constante

Para lograr la descarga de la información que se obtiene del sistema, se implementó una funcionalidad la cual permite al usuario que guarde los datos seleccionados. Esto funciona gracias a un método de exportar la tabla de los datos presentados en un formato CSV, con la utilización de un button el cual es el encargado de llamar al método mediante un click, como se manifiesta en el siguiente bloque.

29. Figura

Bloque de código descargar datos del clima (HTML)

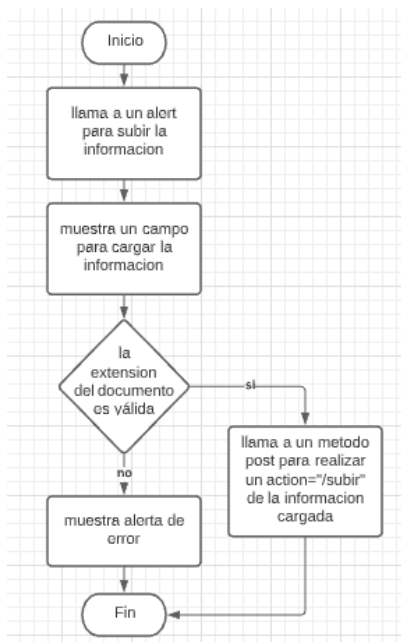
```
<button class="btn btn-success"
        onclick="exportTableToCSV2('{{titulo}}_medias.csv')">Descargar</button>
</div>
</div>
```

Bloque de código descargar datos del clima (HTML). Elaborado por Ken Constante

- *Algoritmo de la consulta del clima para subir datos en la página web*

30. Figura

Algoritmo de la consulta del clima para subir datos en la página web



Algoritmo de la consulta del clima para subir datos en la página web. Elaborado por Ken Constante

Para montar información a la base de datos de la aplicación, el sistema lleva a cabo una opción que inicia un procedimiento post para recibir los datos en los formatos establecidos csv, json, xml, txt y xls, esto el sistema lo carga por medio de un action="" /subir"" desde el alert hacia la base.

31. Figura

Bloque de código subir información a la aplicación web

```
//Metodo para subir archivo
app.post('/subir', upload.single('archivo'), (req, res) => {
  filename = req.file.filename;
  if (getFileExtension(filename) === 'csv') {
    subircsv(filename);
  }
  if (getFileExtension(filename) === 'json') {
    subirjson(filename);
  }
  if (getFileExtension(filename) === 'xml') {
    subirxml(filename);
  }
  if (getFileExtension(filename) === 'txt') {
    subirtxt(filename);
  }
  if (getFileExtension(filename) === 'xlsx') {
    subirxlsx(filename);
  }
});
```

Bloque de código subir información a la aplicación web. Elaborado por Ken Constante

4.7 PRUEBAS

En esta sección se explicará las pruebas que se realizarán al sistema para seguir la evaluación final del sistema web del clima en las cuales se definirá:

- Pruebas de rendimiento (Pingdom Tools y Apache JMeter).
- Los resultados detallados mediante las pruebas efectuadas.
- Pruebas de caja negra y caja de stress. (Apache JMeter)

4.7.1 Plan de pruebas

Aquí se procederá a explicar los enfoques de los diferentes tipos de prueba para saber si satisfacen con el correcto funcionamiento en la web, para determinar si la aplicación responde bien en todas sus fases de prueba.

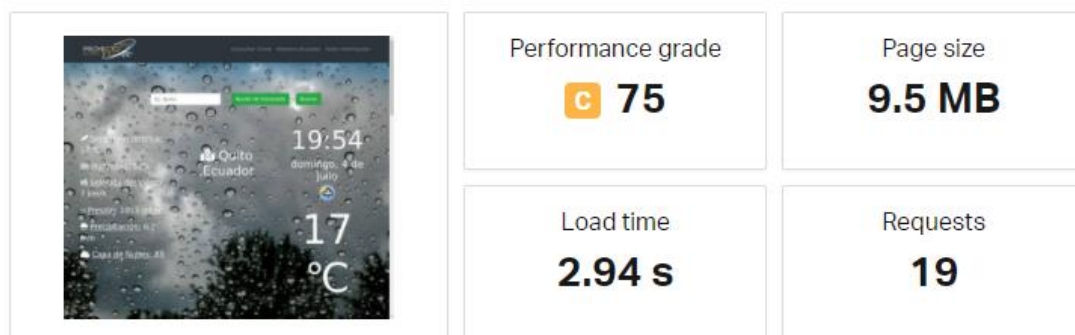
4.7.2 Pingdom Website Speed Test

Esta página es una herramienta para analizar el rendimiento en la web, la página ofrece un monitoreo desde 70 lugares diferentes, esta herramienta permite analizar el rendimiento y las interacciones en la web para mejorar la experiencia final del usuario.

A continuación, se presentará los resultados analizados con esta página:

32. Figura

Prueba de rendimiento



Rendimiento de la aplicación. Elaborado por Ken Constante

Los resultados arrojados por la página en cuestión de rendimiento son muy buenos, puesto que en general le da una calificación de 75, lo cual resulta en un rendimiento bueno, en un uso promedio regular, y en un navegador Microsoft Edge el cuál es manipulado por múltiples usuarios.

Con un rendimiento de carga de 2,98 segundos lo cual dice que la página se carga muy rápido con sus componentes, también nos arroja un peso bastante pequeño para tener un sistema en

la web y 19 peticiones de las cuales 18 están en estado ok y 1 nada más en estado no encontrado.

33. Figura

Rendimiento de la página

Mejorar el rendimiento de la página

GRADO	SUGERENCIA	
F 0	Comprimir componentes con gzip	▼
F 0	Agregar encabezados Expires	▼
C 72	Realizar menos solicitudes HTTP	▼
B 90	Reducir las búsquedas DNS	▼
UN 100	Evite el vacío src o href	▼
UN 100	Poner JavaScript en la parte inferior	▼
UN 100	Reducir el número de elementos DOM	▼

Rendimiento de la página. Elaborado por Ken Constante

En las sugerencias a mejorar que califica la herramienta, por lo que se puede ver en la imagen existen 4 puntos en estado verde lo cual es muy satisfactorio en calificación y nada más 2 en estado rojo que son complementos adicionales que mejoraran el rendimiento de la página, y una en naranja que notifica que es mejor realizar menos peticiones.

34. Figura

Peticiones de la página web

Response codes

RESPONSE CODE	RESPONSES
200 OK	18
404 Not Found	1

Peticiones de la página web. Elaborado por Ken Constante

Con respecto a las peticiones realizadas que encontró este sistema de test, se realizó una evaluación que resulto muy buena, donde se atienden prácticamente todas las peticiones realizadas y solo una no se pudo encontrar, lo cual dice de forma correcta que el sistema funciona bien en todo su apartado.

4.7.3 Apache JMeter

JMeter es un proyecto de Apache el cuál se usará para realizar pruebas de caja y stress para la aplicación del clima y analizar y medir el rendimiento de los servicios que brinda este sistema, con una variedad de situaciones hasta observar errores en el sistema.

12.Tabla

Pruebas del sistema

Tipo de pruebas	Descripción	Herramienta utilizada
Pruebas no funcionales	Las pruebas no funcionales como tales garantizan que el funcionamiento sea correcto, por lo tanto, ayudan a tener una percepción del usuario con el sistema en los siguientes aspectos: Rendimiento, Mantenibilidad, Usabilidad, Disponibilidad y Seguridad	Para este caso no es necesario utilizar una clase de software, ya que en este apartado se hace una revisión de los requerimientos no funcionales de forma manual.
Pruebas de rendimiento en la web	Las pruebas de rendimiento web aseguran si la aplicación tiene un comportamiento adecuado	Pingdom Tools.

	cuando se suba a producción.	
--	------------------------------	--

Tabla de pruebas del sistema. Elaborado por Ken Constante

4.7.4 Pruebas no funcionales

13.Tabla

Prueba no funcional 1

PRNFU-001	
No. Caso de prueba	1
Nombre	Rendimiento
Descripción	Calcular el tiempo de respuesta del sistema en la ejecución de la página principal
Procedimiento de prueba	1. Se procedió a ejecutar la página web en un computador con las características antes mencionadas
Resultado	La aplicación se demora 4 segundos aproximadamente en cargar su página principal evaluado con Pingdom tools.

Prueba no funcional 1. Elaborado por Ken Constante

14.Tabla

Prueba no funcional 2

PRNFU-002	
No. Caso de prueba	2
Nombre	Fiabilidad

Descripción	Revisar si el sistema arroja resultados correctos y que su funcionalidad no varíe por alguna circunstancia especial
Procedimiento de prueba	Se manejo el sistema en todas sus opciones múltiples veces para confirmar que los resultados no varías y también se verifique que sean correctos
Resultado	La aplicación arroja resultados precisos de acuerdo con las múltiples peticiones del usuario evaluado con Apache JMeter

Prueba no funcional 2. Elaborado por Ken Constante

15.Tabla

Prueba no funcional 3

PRNFU-003	
No. Caso de prueba	3
Nombre	Disponibilidad
Descripción	Estimar el tiempo que el aplicativo puede trabajar sin problema
Procedimiento de prueba	Dado que el sistema este alojado en servidores de Heroku, se pudo corroborar de que no existió problema alguno sobre el tiempo de ejecución de la aplicación

Resultado	La aplicación respondió excelentemente durante el uso que se le dio, ningún error desafortunado no anomalía en el sistema
------------------	---

Prueba no funcional 3. Elaborado por Ken Constante

16.Tabla

Prueba no funcional 4

PRNFU-004	
No. Caso de prueba	4
Nombre	Mantenibilidad
Descripción	Presentar el esfuerzo que se puede requerir para mantener una versión del software en funcionamiento, dado que se presente un error
Procedimiento de prueba	El sistema está en su versión estable, a menos de que exista algún problema con la conectividad a internet, se presentará alguna anomalía en el sistema, por tanto, mientras el API funcione correctamente no existiría un problema en la carga de datos de la página
Resultado	El sistema responde correctamente

Prueba no funcional 4. Elaborado por Ken Constante

17.Tabla

Prueba no funcional 5

PRNFU-005	
No. Caso de prueba	5
Nombre	Portabilidad
Descripción	Definir en qué plataformas el sistema funciona totalmente correcto
Procedimiento de prueba	El sistema como tal, se probó en navegadores como Mozilla Firefox, Google Chrome y Microsoft Edge los cuáles son los más habituales y comerciales en la industria y todos respondieron correctamente
Resultado	El sistema funciona en cualquier navegador correctamente

Prueba no funcional 5. Elaborado por Ken Constante

18.Tabla

Prueba no funcional 6

PRNFU-006	
No. Caso de prueba	7
Nombre	Usabilidad
Descripción	Establecer si los usuarios presentan una facilidad para interactuar con el sistema
Procedimiento de prueba	El sistema presenta una interfaz con etiquetas llamativas en colores bajos para

	que la lectura del usuario no se canse, y también botones y gráficos muy bien posicionados para que sea fácil y cómoda la experiencia del usuario
Resultado	El sistema se probó con múltiples usuarios, los cuáles dieron una respuesta positiva ante el sistema

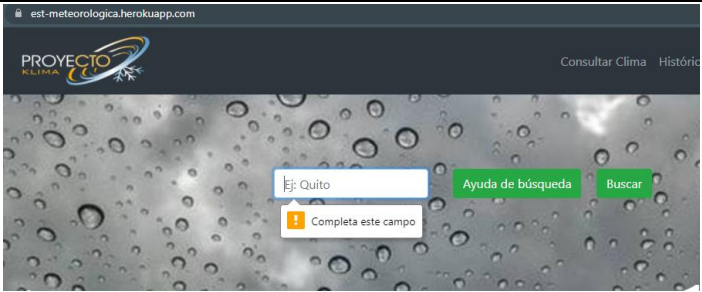
Prueba no funcional 6. Elaborado por Ken Constante

4.7.5 Prueba de Caja Negra

Es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o procesos de ejecución internos en el software, para este sistema se aplicará las pruebas de caja negra bajo la técnica de casos de uso.

19.Tabla


Prueba 1 caja negra

Condición	Problema	Solución
Realizar una consulta de clima	Si el usuario no ingresa ningún parámetro y realiza la búsqueda	

Prueba 1 caja negra. Elaborado por Ken Constante

20.Tabla


Prueba 2 caja negra

Condición	Problema	Solución
Ingresar a la página principal de la aplicación	El API tiene problemas con los datos	

Prueba 2 caja negra. Elaborado por Ken Constante

21.Tabla


Prueba 3 caja negra

Condición	Problema	Solución
Realizar una consulta a la aplicación	El sistema pierde conexión para mostrar los datos requeridos	

Prueba 3 caja negra. Elaborado por Ken Constante

22.Tabla


Prueba 4 caja negra

Condición	Problema	Solución
Realizar una consulta de datos históricos	Problema con el API, el sistema muestra un mensaje al usuario	

Prueba 4 caja negra. Elaborado por Ken Constante

23.Tabla

Prueba 5 caja negra

Condición	Problema	Solución
Se requiere subir un archivo a la aplicación	Se intenta subir una extensión diferente a las permitidas con el sistema	

Prueba 5 caja negra. Elaborado por Ken Constante

24.Tabla

Prueba 6 caja negra

Condición	Problema	Solución																								
Se desea descargar información histórica de la aplicación	Se desea una extensión de documentos que el sistema no tiene	 <p>The screenshot shows a web interface for 'PROYECTO' with weather data. A 'Descargas' (Downloads) menu is open, listing several files for download. The data table below the menu includes columns for Temperature, Sensation, Pressure, Precipitation, Humidity, Cloud Cover, Wind Speed, and Date.</p> <table border="1"> <thead> <tr> <th>Temperatura °C</th> <th>Sensación Térmica °C</th> <th>Presión mbar</th> <th>Precipitación mm</th> <th>Humedad %</th> <th>Capa de Nubes %</th> <th>Velocidad Viento km/h</th> <th>Fecha</th> </tr> </thead> <tbody> <tr> <td>14.8</td> <td>14.8</td> <td>1013.6</td> <td>0.58</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>14</td> <td>14</td> <td>1014</td> <td>0.9</td> <td>77</td> <td>73</td> <td>3</td> <td>Viernes, 9 de julio de 2021 2:10 hrs</td> </tr> </tbody> </table>	Temperatura °C	Sensación Térmica °C	Presión mbar	Precipitación mm	Humedad %	Capa de Nubes %	Velocidad Viento km/h	Fecha	14.8	14.8	1013.6	0.58					14	14	1014	0.9	77	73	3	Viernes, 9 de julio de 2021 2:10 hrs
Temperatura °C	Sensación Térmica °C	Presión mbar	Precipitación mm	Humedad %	Capa de Nubes %	Velocidad Viento km/h	Fecha																			
14.8	14.8	1013.6	0.58																							
14	14	1014	0.9	77	73	3	Viernes, 9 de julio de 2021 2:10 hrs																			

Prueba 6 caja negra. Elaborado por Ken Constante

Las pruebas descritas se realizaron en función de los casos de uso, especificando su entrada y salida correspondiente, para las 6 pruebas realizadas se detalló las extensiones en cada punto, expresando así los sucesos en caso de una anomalía en el sistema.

4.7.6 Prueba de Caja y Stress

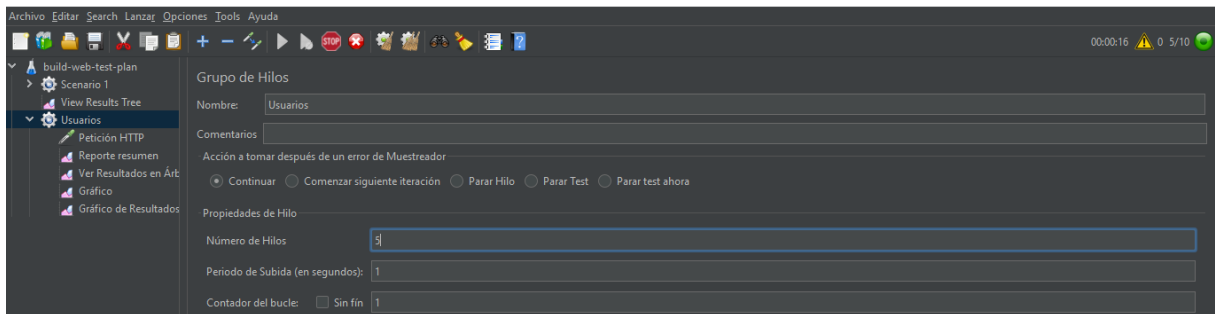
Se procede a probar con una determinada cantidad de usuarios para ver cuál es el rendimiento que nos arroja a medida que se sobrecarga al sistema, primero se inicializará estas pruebas con los siguientes usuarios:

- Prueba con 10 usuarios que realizarán una petición cada usuario en un margen de tiempo de 1 segundo.

Se crean parámetros en el JMeter:

35. Figura

Parámetros con 10 usuarios en JMeter



Parámetros con 10 usuarios en JMeter. Elaborado por Ken Constante

Para inicializar la prueba se ingresa 10 usuarios en estado online, es decir, que todos realizaran una petición en 1 minuto, así se establece los parámetros iniciales para ver cómo se comporta el sistema con un tráfico un poco bajo o normal.

36. Figura

Resultados en tabla con 10 usuarios en JMeter

The screenshot shows the 'Ver Resultados en Árbol' (View Results in Tree) window in JMeter. It displays a table of test results for 5 samples. The table columns are: Muestra #, Tiempo de comie..., Nombre del hilo, Etiqueta, Tiempo de Muestr..., Estado, Bytes, Sent Bytes, Latency, and Connect Time(ms).

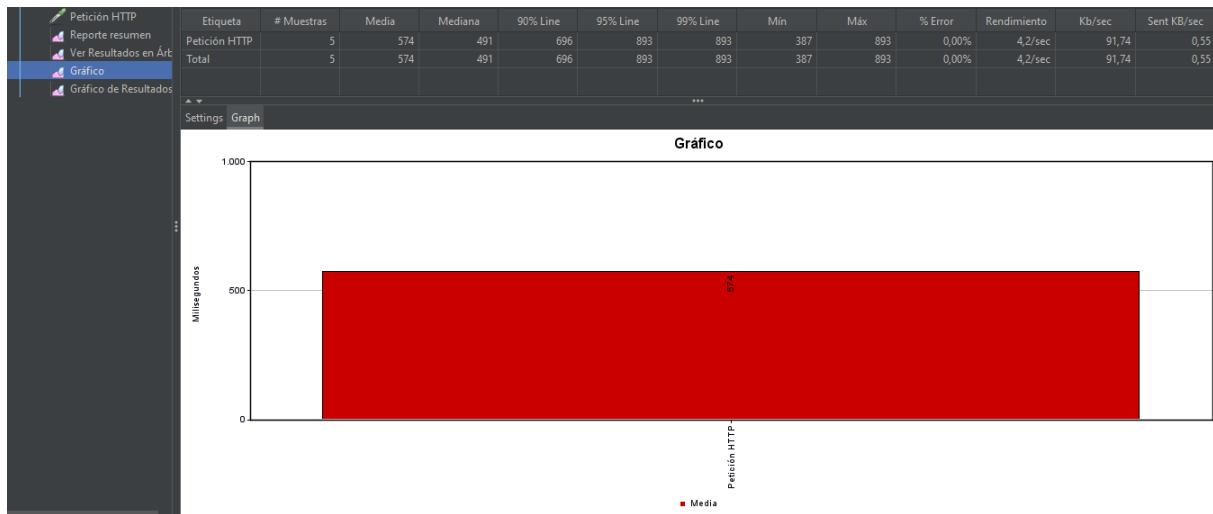
Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Muestr...	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	13:51:47.130	Usuarios 2-3	Petición HTTP	491	✓	22265	133	392	100
2	13:51:46.929	Usuarios 2-2	Petición HTTP	696	✓	22265	133	597	100
3	13:51:46.733	Usuarios 2-1	Petición HTTP	893	✓	22265	133	795	262
4	13:51:47.331	Usuarios 2-4	Petición HTTP	404	✓	22265	133	305	100
5	13:51:47.531	Usuarios 2-5	Petición HTTP	387	✓	22265	133	288	96

Resultados en tabla con 10 usuarios en JMeter. Elaborado por Ken Constante

Los resultados arrojados por el sistema JMeter son muy buenos puesto que la aplicación respondió correctamente a todas las peticiones realizadas, y ninguna tuvo alguna dificultad los tiempos de carga de los usuarios fue con una latencia de 400-600 milisegundos muy rápido.

37. Figura

Gráfica con 10 usuarios en JMeter

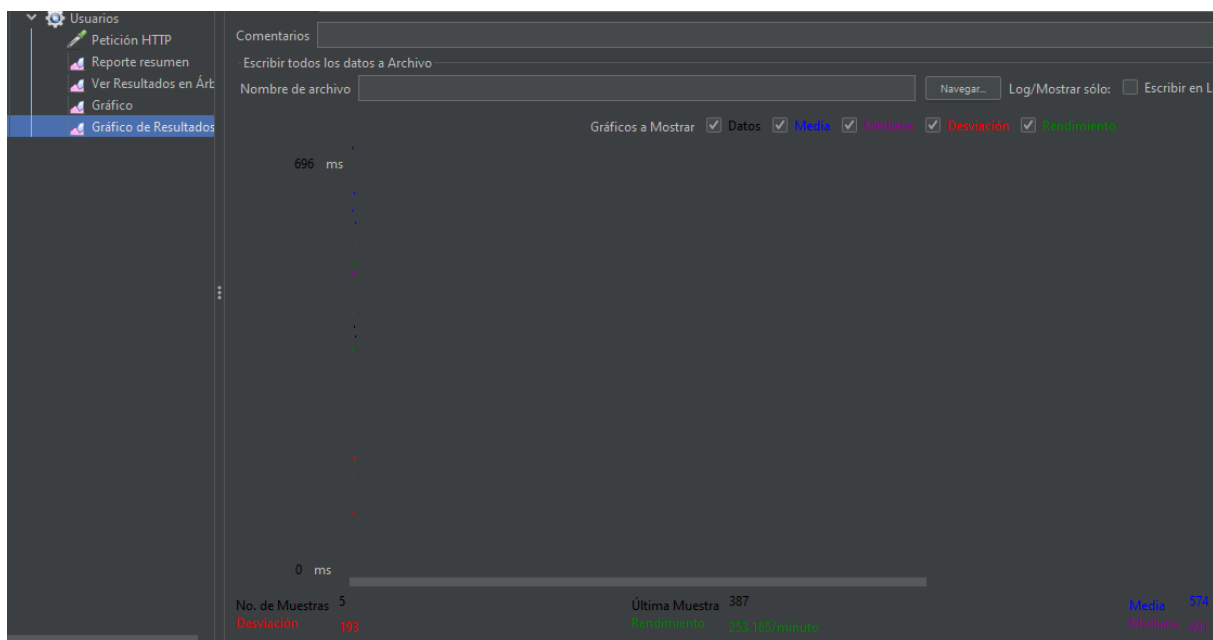


Gráfica con 10 usuarios en JMeter. Elaborado por Ken Constante

En la gráfica se puede observar la media en milisegundos que le toma a un grupo de 10 usuarios trabajar con el sistema a la misma vez, realizando peticiones por minuto, la media es de 500 milisegundos de latencia un tiempo muy bueno.

38. Figura

Gráfica de datos del resultado con 10 usuarios JMeter



Gráfica de datos del resultado con 10 usuarios JMeter. Elaborado por Ken Constante

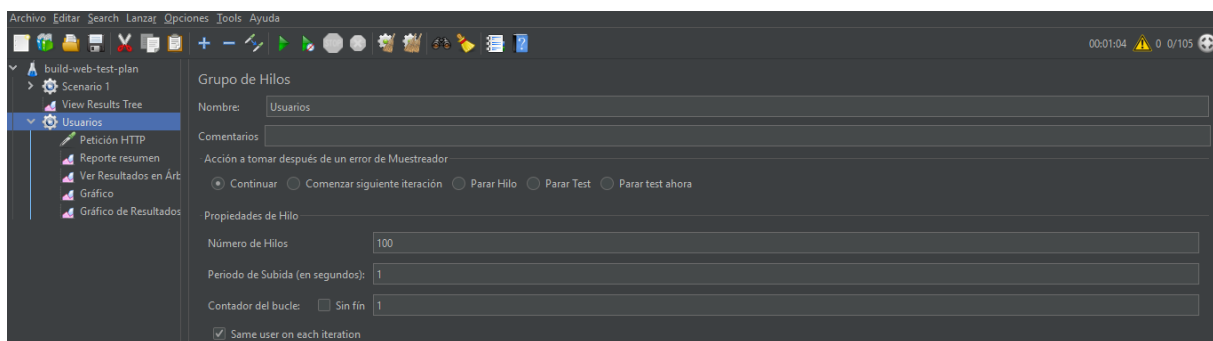
No se puede observar claramente el gráfico de tráfico del sistema, pero si se realiza un pequeño zoom se puede observar unos puntos de color, verde, azul, rojo y morado los cuales solo les puede observar el punto de partida, puesto que es una prueba muy sencilla para el sistema y casi que ni se observa la carga que tiene.

- Prueba con 100 usuarios que realizaran una petición cada usuario en un margen de tiempo de 1 segundo

Se establece los parámetros en el JMeter:

39. Figura

Parámetros con 100 usuarios en JMeter

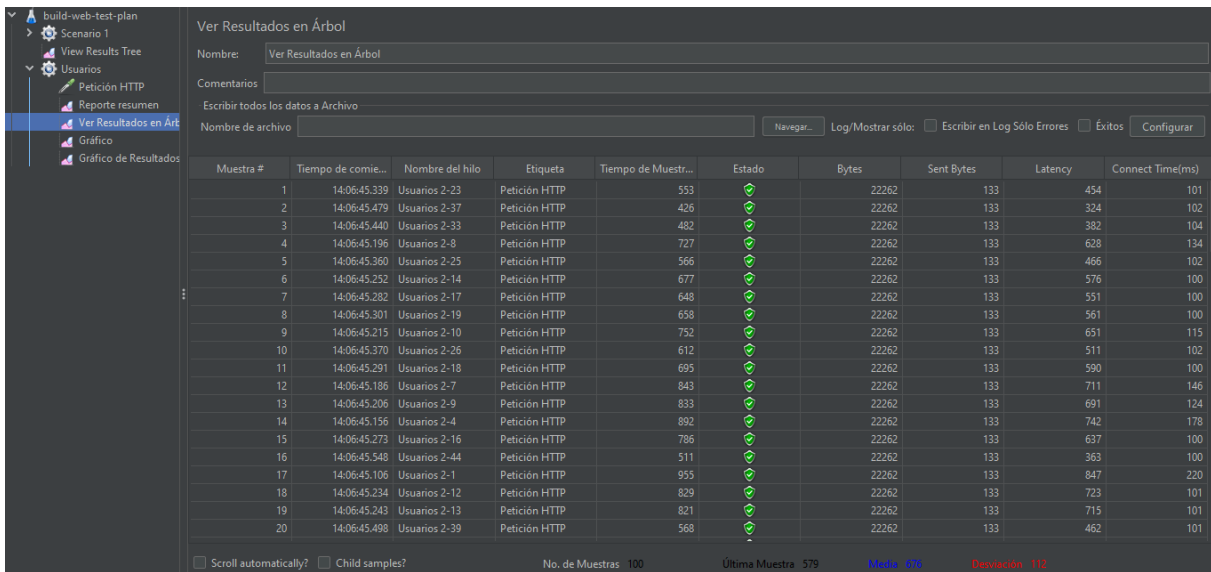


Parámetros con 100 usuarios en JMeter. Elaborado por Ken Constante

Para esta prueba ya se proporciona una cantidad de usuarios un poco más prolongada, se establece en el parámetro de usuarios la cantidad de 100 usuarios que realizaran 1 petición cada uno, para ver cómo se comporta el sistema.

40. Figura

Resultados en tabla con 100 usuarios en JMeter



Ver Resultados en Árbol

Nombre: Ver Resultados en Árbol

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos Configurar

Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Muestr...	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	14:06:45.339	Usuarios 2-23	Petición HTTP	553	✓	22262	133	454	101
2	14:06:45.479	Usuarios 2-37	Petición HTTP	426	✓	22262	133	324	102
3	14:06:45.440	Usuarios 2-33	Petición HTTP	482	✓	22262	133	382	104
4	14:06:45.196	Usuarios 2-8	Petición HTTP	727	✓	22262	133	628	134
5	14:06:45.360	Usuarios 2-25	Petición HTTP	566	✓	22262	133	466	102
6	14:06:45.252	Usuarios 2-14	Petición HTTP	677	✓	22262	133	576	100
7	14:06:45.282	Usuarios 2-17	Petición HTTP	648	✓	22262	133	551	100
8	14:06:45.301	Usuarios 2-19	Petición HTTP	658	✓	22262	133	561	100
9	14:06:45.215	Usuarios 2-10	Petición HTTP	752	✓	22262	133	651	115
10	14:06:45.370	Usuarios 2-26	Petición HTTP	612	✓	22262	133	511	102
11	14:06:45.291	Usuarios 2-18	Petición HTTP	695	✓	22262	133	590	100
12	14:06:45.186	Usuarios 2-7	Petición HTTP	843	✓	22262	133	711	146
13	14:06:45.206	Usuarios 2-9	Petición HTTP	833	✓	22262	133	691	124
14	14:06:45.156	Usuarios 2-4	Petición HTTP	892	✓	22262	133	742	178
15	14:06:45.273	Usuarios 2-16	Petición HTTP	786	✓	22262	133	637	100
16	14:06:45.548	Usuarios 2-44	Petición HTTP	511	✓	22262	133	363	100
17	14:06:45.106	Usuarios 2-1	Petición HTTP	955	✓	22262	133	847	220
18	14:06:45.234	Usuarios 2-12	Petición HTTP	829	✓	22262	133	723	101
19	14:06:45.243	Usuarios 2-13	Petición HTTP	821	✓	22262	133	715	101
20	14:06:45.498	Usuarios 2-39	Petición HTTP	568	✓	22262	133	462	101

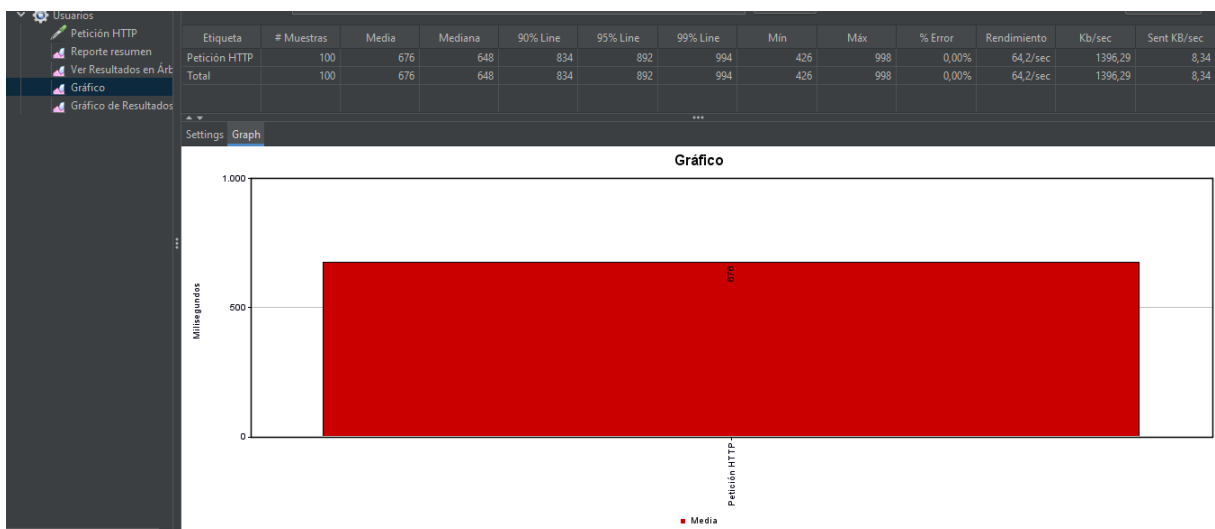
Scroll automatically? Child samples? No. de Muestras: 100 Última Muestra: 579 Media: 676 Desviación: 112

Resultados en tabla con 100 usuarios en JMeter. Elaborado por Ken Constante

Los resultados con 100 usuarios realizando una petición por minuto, fue muy buena obteniendo 100 peticiones en estado ok, sin que ninguna se cayera o tuviera alguna clase de falla, con una carga un poco más prolongada el sistema respondió muy bien.

41. Figura

Gráfica con 100 usuarios en JMeter

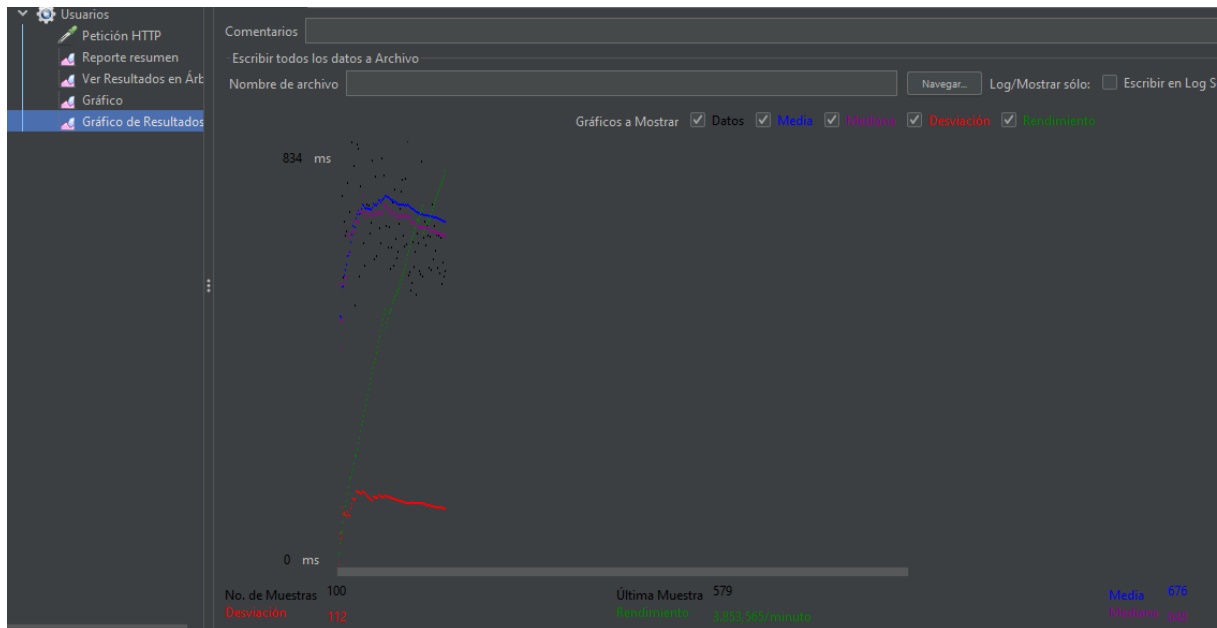


Gráfica con 100 usuarios en JMeter. Elaborado por Ken Constante

El sistema arroja como resultado con 100 usuarios un tiempo de 676 milisegundos de latencia en promedio de respuesta para cada usuario, un tiempo bueno con diferencia a la cantidad de 10 usuarios que sigue siendo muy bueno.

42. Figura

Gráfica de datos del resultado con 100 usuarios JMeter



Gráfica de datos del resultado con 10 usuarios JMeter. Elaborado por Ken Constante

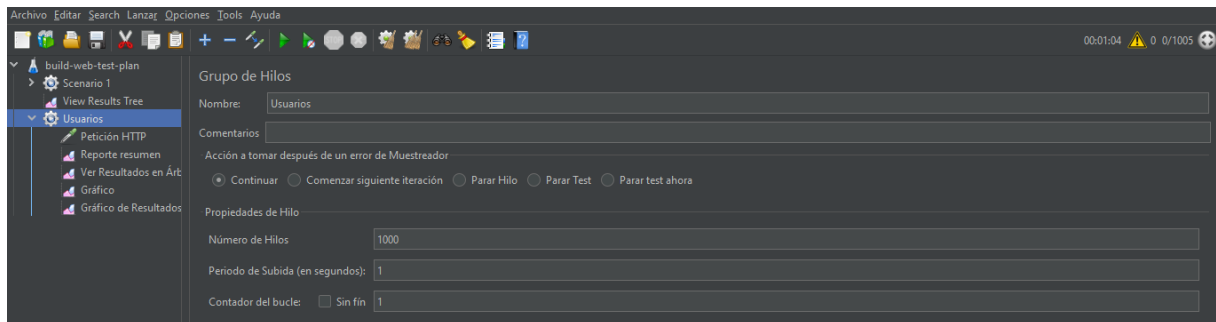
La prueba realizada con 100 usuarios ya es un parámetro un poco más regular para el sistema realizando 100 peticiones por segundo y respondiendo exitosamente en un promedio de 0,676 milisegundos de latencia, sin tener error en ningún como se puede observar en la gráfica el sistema se mantiene sin tener un pique hacia abajo por rendimiento.

- Prueba con 1000 usuarios que realizaran una petición cada usuario en un margen de tiempo de 1 segundo

Se establece los parámetros en el JMeter:

43. Figura

Parámetros con 1000 usuarios en JMeter



Parámetros con 1000 usuarios en JMeter. Elaborado por Ken Constante

Se establece un parámetro inicial de 1000 usuarios que realizaran una petición cada uno en un segundo, esperando que el sistema responda o ya presente algún error en su carga.

44. Figura

Resultados en tabla con 1000 usuarios en JMeter

The screenshot shows the "Ver Resultados en Árbol" (View Results in Tree) window in JMeter. The table displays the results of 100 samples, all of which were successful (OK). The columns include "Muestra #" (Sample #), "Tiempo de comie..." (Start time), "Nombre del hilo" (Thread name), "Etiqueta" (Label), "Tiempo de Muestr..." (Sample time), "Estado" (Status), "Bytes", "Sent Bytes", "Latency", and "Connect Time(ms)".

Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Muestr...	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
81	14:19:50.294	Usuarios 2-240	Petición HTTP	4235	OK	22265	133	4132	121
82	14:19:50.266	Usuarios 2-136	Petición HTTP	4266	OK	22265	133	4163	133
83	14:19:50.303	Usuarios 2-167	Petición HTTP	4230	OK	22265	133	4128	127
84	14:19:50.346	Usuarios 2-98	Petición HTTP	4189	OK	22265	133	4091	114
85	14:19:50.161	Usuarios 2-71	Petición HTTP	4378	OK	22265	133	4280	148
86	14:19:50.290	Usuarios 2-143	Petición HTTP	4258	OK	22265	133	4155	125
87	14:19:50.299	Usuarios 2-157	Petición HTTP	4254	OK	22265	133	4152	130
88	14:19:50.307	Usuarios 2-151	Petición HTTP	4246	OK	22265	133	4141	160
89	14:19:50.308	Usuarios 2-185	Petición HTTP	4247	OK	22265	133	4147	159
90	14:19:50.306	Usuarios 2-174	Petición HTTP	4252	OK	22265	133	4152	161
91	14:19:50.361	Usuarios 2-312	Petición HTTP	4198	OK	22265	133	4098	116
92	14:19:50.301	Usuarios 2-162	Petición HTTP	4259	OK	22265	133	4159	129
93	14:19:50.296	Usuarios 2-146	Petición HTTP	4266	OK	22265	133	4168	119
94	14:19:50.344	Usuarios 2-192	Petición HTTP	4219	OK	22265	133	4118	116
95	14:19:50.297	Usuarios 2-155	Petición HTTP	4270	OK	22265	133	4174	118
96	14:19:50.352	Usuarios 2-224	Petición HTTP	4216	OK	22265	133	4115	109
97	14:19:50.276	Usuarios 2-137	Petición HTTP	4293	OK	22265	133	4193	124
98	14:19:50.302	Usuarios 2-255	Petición HTTP	4271	OK	22265	133	4172	128
99	14:19:50.342	Usuarios 2-225	Petición HTTP	4233	OK	22265	133	4134	118
100	14:19:50.297	Usuarios 2-153	Petición HTTP	4280	OK	22265	133	4181	118

Resultados en tabla con 1000 usuarios en JMeter. Elaborado por Ken Constante

Sorprendentemente el sistema atendió correctamente las peticiones, obteniendo un resultado de 1000 peticiones en estado ok, es decir todos los usuarios obtuvieron su respuesta.

45. Figura

Gráfica con 1000 usuarios en JMeter

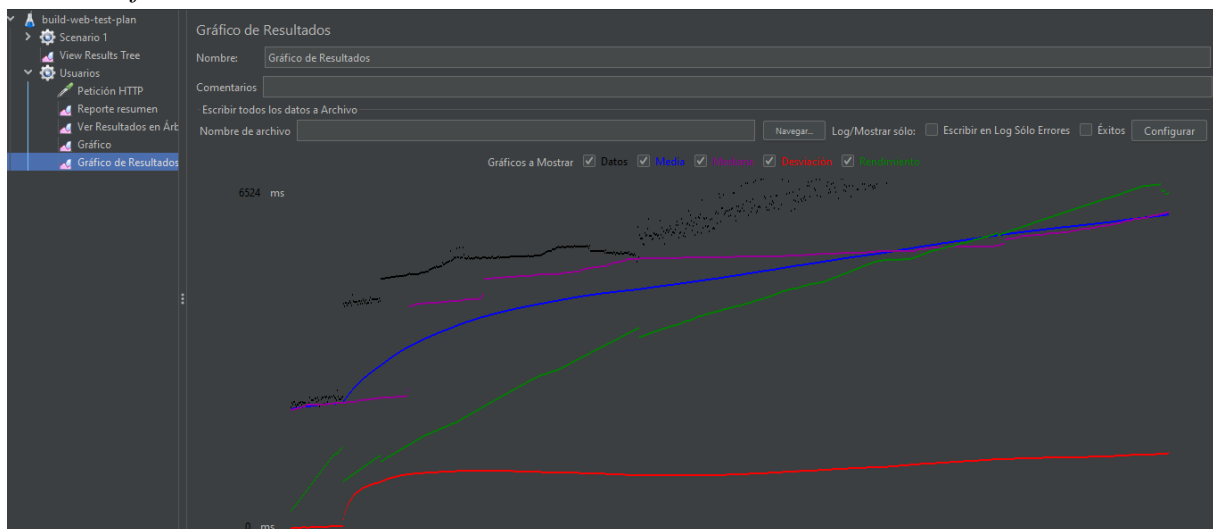


Gráfica con 1000 usuarios en JMeter. Elaborado por Ken Constante

Sin embargo, ya se notó un cambio muy llamativo, puesto que el tiempo de respuesta en promedio fue de 5,868 milisegundos de latencia, lo cual varía demasiado con las anteriores pruebas realizadas.

46. Figura

Gráfica de datos del resultado con 1000 usuarios en JMeter



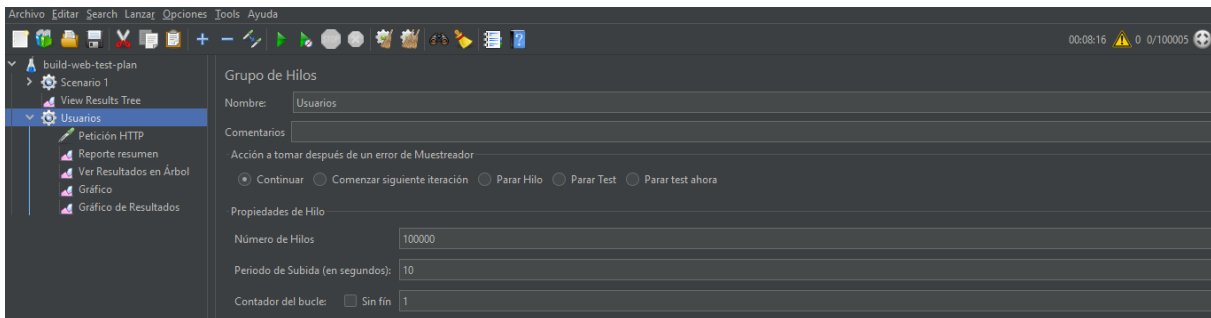
Gráfica de datos del resultado con 1000 usuarios en JMeter. Elaborado por Ken Constante

La prueba realizada con 1000 usuarios tiene una exigencia ya pronunciada hacia el sistema, observando como atiende 1000 peticiones en un lapso de un segundo en nada más que 5,868 segundos lo cual está bastante bien teniendo en cuenta la cantidad de usuarios y peticiones en un segundo, la gráfica crece, pero a pasos lentos lo cual denota que ya le cuesta atender a cada usuario.

Para 1000 usuarios realizando 10 peticiones en un lapso de 10 segundos el resultado fue malo, pues aquí ya se obtuvo una serie de errores que en parte es por el equipo en el cual se está probando, pero también ya se presentan errores de carga, donde el servidor logra colapsar por lapsos.

47. Figura

Parámetros con 10000 peticiones en JMeter



Parámetros con 10000 peticiones en JMeter. Elaborado por Ken Constante

Se inicializa la prueba final de carga y stress con 10000 usuarios, esto ya demando una exigencia potente al sistema y si no se colapsa, ya es muy satisfactorio.

48. Figura

Resultados en tabla con 10000 peticiones en JMeter

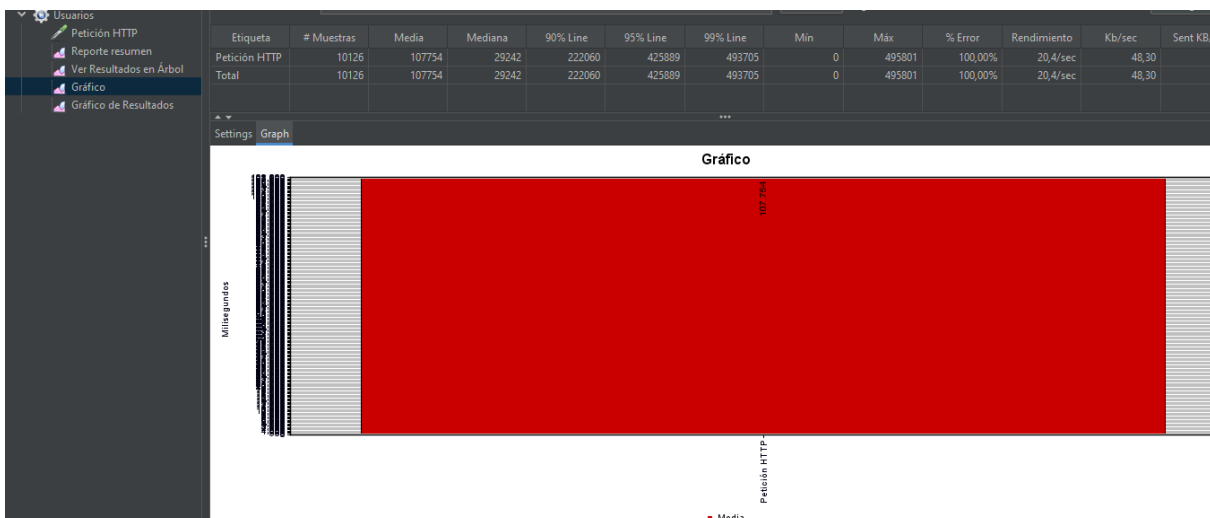
Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Muestr...	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	14:40:04.441	Usuarios 2-14544	Petición HTTP	3	✖	2326	0	0	3
2	14:40:04.442	Usuarios 2-14545	Petición HTTP	2	✖	2326	0	0	2
3	14:40:04.442	Usuarios 2-14543	Petición HTTP	2	✖	2326	0	0	2
4	14:40:04.442	Usuarios 2-14939	Petición HTTP	2	✖	2326	0	0	2
5	14:40:04.434	Usuarios 2-14917	Petición HTTP	4	✖	2326	0	0	4
6	14:40:04.436	Usuarios 2-14432	Petición HTTP	2	✖	2326	0	0	2
7	14:40:04.425	Usuarios 2-14859	Petición HTTP	2	✖	2326	0	0	2
8	14:40:04.425	Usuarios 2-15003	Petición HTTP	2	✖	2326	0	0	2
9	14:40:04.469	Usuarios 2-14779	Petición HTTP	5	✖	2326	0	0	5
10	14:40:04.467	Usuarios 2-14576	Petición HTTP	7	✖	2326	0	0	7
11	14:40:04.469	Usuarios 2-14580	Petición HTTP	5	✖	2326	0	0	5
12	14:40:04.469	Usuarios 2-14569	Petición HTTP	5	✖	2326	0	0	5
13	14:40:04.468	Usuarios 2-14573	Petición HTTP	7	✖	2326	0	0	7
14	14:40:04.465	Usuarios 2-14575	Petición HTTP	10	✖	2326	0	0	10
15	14:40:04.467	Usuarios 2-14771	Petición HTTP	8	✖	2326	0	0	8
16	14:40:04.465	Usuarios 2-14838	Petición HTTP	10	✖	2326	0	0	10
17	14:40:04.465	Usuarios 2-14770	Petición HTTP	10	✖	2326	0	0	10
18	14:40:04.465	Usuarios 2-14837	Petición HTTP	10	✖	2326	0	0	10
19	14:40:04.465	Usuarios 2-14926	Petición HTTP	10	✖	2326	0	0	10
20	14:40:04.464	Usuarios 2-14752	Petición HTTP	11	✖	2326	0	0	11

Resultados en tabla con 10000 peticiones en JMeter. Elaborado por Ken Constante

Como se tenía previsto el sistema empezó a arrojar cargas de error en los usuarios que se contabilizaban como 2000 para arriba, dado que se prueba con hilos el rendimiento no se establece un orden, es decir, el hilo que en la primera prueba si recibió respuesta, no se asegura que vuelva a recibir respuesta en una segunda prueba.

49. Figura

Gráfica con 10000 peticiones en JMeter

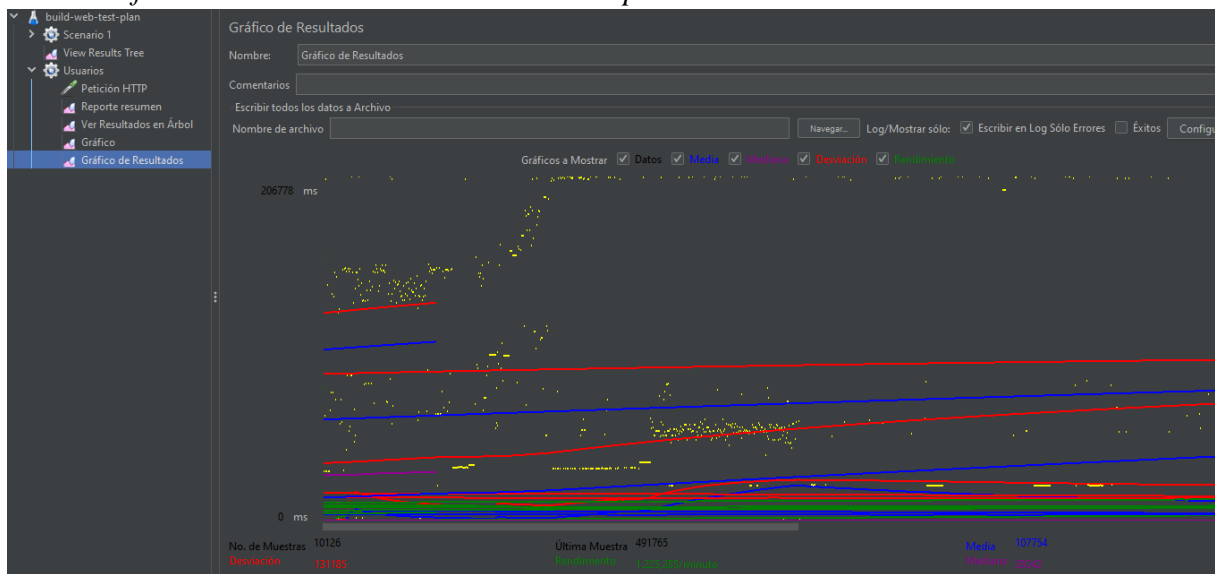


Gráfica con 10000 peticiones en JMeter. Elaborado por Ken Constante

El tiempo promedio de respuesta para esta prueba fue de 115.000 milisegundos de latencia, por petición, es decir que ya se demora una cantidad considerable de tiempo en responder, y a lapsos se cae el servidor por el gran número de peticiones.

50. Figura

Gráfica de datos del resultado con 10000 peticiones en JMeter



Gráfica de datos del resultado con 10000 peticiones en JMeter. Elaborado por Ken

Constante

La gráfica de rendimiento ya se ve totalmente rota, porque al pasar de los 2000 usuarios el sistema ya no responde bien, anulando miles de peticiones y generando una gráfica de rendimiento muy inestable donde no se concreta un buen trabajo casi en ninguna circunstancia.

4.8 ANÁLISIS DE RESULTADOS

Una vez realizadas las pruebas no funcionales, de caja negra y stress, se procederá a detallar los resultados evaluados en cada una de ellas, adicionalmente se realizó una prueba de compatibilidad de los navegadores para encontrar anomalías en algún tipo de navegador.

25.Tabla

Compatibilidad de navegadores

Navegador	Versión	Acciones	Resultado
Microsoft Edge	Versión 91.0.864.67	1. Ingresar al enlace	100% ok
		2. Carga de los componentes principales	100% ok
		3. Despliegue del menú	100% ok
		4. Navegación	100% ok
		5. Manipulación del sistema	100% ok
		6. Salir del sistema	100% ok
Google Chrome	Versión 91.0.4472.124	Ingresar al enlace	100% ok
		Carga de los componentes principales	100% ok
		Despliegue del menú	100% ok
		Navegación	100% ok
		Manipulación del sistema	100% ok
		Salir del sistema	100% ok
Mozilla Firefox	Versión 89.0.2	Ingresar al enlace	100% ok
		Carga de los componentes principales	80% plugin con fallas en la fecha
		Despliegue del menú	100% ok
		Navegación	100% ok
		Manipulación del sistema	100% ok

		Salir del sistema	100% ok
--	--	-------------------	---------

Compatibilidad de navegadores. Elaborado por Ken Constante

Como se puede observar se está evaluando en los 3 navegadores más comunes de eso para los usuarios, y en todos los resultados fueron muy buenos en cuestión de rendimiento, solamente en el navegador Mozilla Firefox, la aplicación presenta problemas de compatibilidad en el plugin de la fecha, no carga correctamente, pero por lo demás el rendimiento y la portabilidad muy buena.

4.8.1 Resultados pruebas no funcionales

Los resultados para estas pruebas fueron muy buenos, evaluando así su rendimiento como aplicación en la web, su velocidad de carga y evaluando si es que se cumplen todas las peticiones realizadas al sistema, a continuación, se detallará los resultados de estas pruebas en su totalidad.

26.Tabla

Resultados pruebas no funcionales

Caso de prueba	Perfil	Resultado	Observación	Estado
Rendimiento	Sistema web	La aplicación se demora 4 segundos aproximadamente en cargar su página principal	La aplicación tiene un rendimiento bueno	100% ok
Fiabilidad	Sistema web	La aplicación arroja resultados	La aplicación responde	100% ok

		precisos de acuerdo con las múltiples peticiones del usuario	excelentemente a todas las peticiones hechas por el usuario	
Disponibilidad	Sistema web	La aplicación respondió excelentemente durante el uso que se le dio, ningún error desafortunado no anomalía en el sistema	La aplicación respondió muy bien en toda su demanda	100% ok
Mantenibilidad	Sistema web	El sistema responde correctamente	La aplicación está en una versión muy estable	100% ok
Portabilidad	Sistema web	El sistema funciona en cualquier navegador correctamente	La aplicación funciona en todos los ambientes muy bien	100% ok

Usabilidad	Sistema web	Los usuarios dieron una respuesta positiva ante el sistema	La aplicación tiene buen feedback	100% ok
-------------------	--------------------	--	-----------------------------------	----------------

Resultados pruebas no funcionales. Elaborado por Ken Constante

4.8.2 Resultados pruebas de caja negra

Con la demostración se obtuvo un efecto muy bueno, el cual fue el fiado con la corriente alterna, la respuesta del sistema estuvo a la altura, a continuación, se especificarán los beneficios de estas pruebas.

27.Tabla

Resultados pruebas de caja negra

Prueba	Perfil	Flujo alternativo	Resultado	Estado
Prueba 1	Sistema web	El sistema autocompleta la búsqueda cuando se ingresa un parámetro no valido	El sistema cumple con la respuesta	100% ok
Prueba 2	Sistema web	El API presenta problemas de funcionamiento	El sistema emite la respuesta	100% ok

Prueba 3	Sistema web	El sistema pierde conexión con los datos	El sistema emite la respuesta	100% ok
Prueba 4	Sistema web	El sistema no tiene datos para el gráfico	El sistema emite la respuesta	100% ok
Prueba 5	Sistema web	Se intenta subir un formato no válido al sistema	El sistema cumple con la respuesta	100% ok
Prueba 6	Sistema web	El sistema descarga en los datos válidos	El sistema cumple con la respuesta	100% ok

Resultados pruebas de caja negra. Elaborado por Ken Constante

4.8.4 RESULTADOS DE PRUEBAS DE STRESS JMETER

Las pruebas de stress se realizaron con JMeter y se obtuvo resultados muy buenos sometiendo al sistema a una presión de trabajo muy alta, llegando a colapsar y saturando el servidor como era de esperar luego de una demanda muy alta, a continuación, se presentará los resultados detallados.

28.Tabla

Resultados pruebas de stress JMeter

Número de usuarios	Peticiones	Tiempo	Resultado
10	1 petición cada uno	500 milisegundos de latencia en la respuesta	Rendimiento satisfactorio
100	1 petición cada uno	676 milisegundos de latencia en la respuesta	Rendimiento satisfactorio
1000	1 petición cada uno	5,868 milisegundos de latencia en la respuesta	Rendimiento bueno
1000	10 peticiones cada uno en un lapso de 10 segundos	115.000 milisegundos de latencia en la respuesta	Rendimiento muy malo, debido a que se demora 1 minuto con 55 segundos y colapsa

Resultados pruebas de stress JMeter. Elaborado por Ken constante

Las pruebas de stress se realizaron con la finalidad de saturar totalmente al servidor del sistema web, sobrecargándolo de manera que se llegue a caer o simplemente no responda más, se evidencio que al pasar de 1000 usuarios la aplicación colapso.

Dado que es una aplicación alojada en un servidor free de Heroku que proporciona 512 de RAM y con límite de almacenamiento de 150MB, se pudo demostrar que trabaja muy bien

hasta los 1000 usuarios, arrojando una latencia mínima en cada respuesta, se estableció que es justificable que pasados los 1000 usuarios la aplicación deje de responder.

CONCLUSIONES

- Con la implementación de una ayuda de búsqueda en el parámetro para ingresar el clima se consiguió automatizar el funcionamiento de la página, dado que como el API cuenta con miles de registros de ubicaciones, un problema era la carga de todos los registros con un nombre en común para la aplicación.
- La aplicación web fue desarrollada bajo JavaScript puesto que en un lenguaje muy robusto y bastante rápido por cómo se ejecuta de forma inmediata en el navegador, así mismo se utilizó Node JS para la construcción del backend.
- Con la arquitectura utilizada (cliente-servidor) se obtuvo un nivel de seguridad bastante bueno, ya que se cuenta con un sistema de autenticación por parte del administrador en el API para una correcta funcionalidad de la información que se obtiene.
- Gracias a las pruebas no funcionales se pudo lograr la evaluación del funcionamiento del sistema, debido al cumplimiento de todos los requerimientos establecidos, y cumpliendo con el objetivo general al que fue enfocado.
- Las pruebas de stress demostraron que la aplicación bajo presión de más de 1000 usuarios empezaba a presentar anomalías, peticiones no atendidas, colapso del servidor, pero al mismo tiempo demostró que con una carga normal y un poco exigente respondía muy bien, tomándose un poco de tiempo, pero al final atendiendo peticiones de todos los usuarios.
- A pesar de que el servidor remoto Heroku donde esta alojada la aplicación, es de servicio gratis, se determinó un funcionamiento correcto, en etapas de exigencia normal y con carga pesada, la aplicación está muy bien optimizada lo cual es crucial en un servicio en internet.

RECOMENDACIONES

- El aplicativo web puede ser complementado con un apartado de pronóstico del clima, para que quede una aplicación bastante completa, debido a que cuenta con datos estadísticos se puede elaborar un apartado para predecir en base a los datos históricos.
- Para la personalización también se puede implementar un apartado donde al usuario se le permita obtener temas para su consulta del clima, puesto que la atención del usuario es un tema muy importante cuando se trata de un sitio web, es crucial desarrollar estrategias para captar la atención del usuario.
- Se puede recomendar la idea de implementar un tipo de opción para observar las estaciones de la luna, ya que la aplicación web está orientada al clima, se podría también observar un poco de información acerca de la Luna.
- Se recomienda hacer la carga de la información a la base, con los formatos establecidos, debido a que estos son los principales que se utilizan a nivel mundial, también puede ocasionar una mala carga de la información dentro de la base.
- Se recomienda utilizar el botón de ayuda de búsqueda al ingresar parámetros en el apartado de búsqueda, si es que no se tiene el lugar muy claro o muy bien escrito, por ejemplo, si se va a consultar el clima de la capital de Australia, se ingresa solo Sid... y el botón brindará una lista de lugares con esas coincidencias en este caso Sídney, Australia, logrando así una búsqueda exacta del lugar de interés.
- Para tener una experiencia muy buena, se recomienda realizar una búsqueda a detalle con los parámetros que se le brinda al usuario por medio de la aplicación, para así que disfrute de una aplicación en su máximo esplendor y si desea también descargar los datos para realizar algún tipo de análisis.
- Si el usuario desea realizar una consulta más exhaustiva acerca de un sitio en específico, se le aconseja no utilizar una fecha de inicio muy ambigua, puesto que como la base de

datos está en constante actualización en periodos de tiempo considerables, puede ocasionar un resultado malo de la búsqueda, mostrando datos vacíos sobre esa ciudad.

Bibliografía

Artículos

- Aris Munandar, H. F. (2017). Design of Real-time Weather Monitoring System. *IEEE*, 4.
- Campoverde, A. S. (2018). Analysis of the urban heat island in the Andean environment of Cuenca-Ecuador. *Scopus*, 14.
- Kumar, D. (2020). Low Cost IoT enabled Weather Station. *IEEE*, 7.
- Marcos, C. R. (11 de 09 de 2017). Implementación De Una Interfaz SIG Web Para El Despliegue De Pronósticos Meteorológicos Dirigido A La Subdirección De Aviación Civil Del Ecuador. *Universidad De Guayaquil.Facultad De Ciencias Matemáticas Y Físicas.Carrera De Ingeniería En Sistemas Computacionales*, 146.
- Turner, M. (2017). Utilizing a Network of Wireless Weather Stations to Forecast Weather in Developing Countries. *IEEE*, 3.
- Uriarte, J. M. (13 de 04 de 2020). "Meteorología". Obtenido de <https://www.caracteristicas.co/meteorologia/>.

Sitios Web

- A, D. (11 de 12 de 2020). *Hostinger.es*. Obtenido de <https://www.hostinger.es/tutoriales/que-es-json>
- Abbatea, S. (2013). Deploying a communicating Automatic Weather Station on an Alpine glacier. *IEEE*, 8.
- Abelladan, E. (20 de 05 de 2020). *WAM*.
- BeeDIGITAL, E. (05 de 10 de 2018). *Ventajas y Desventajas sistema web*.
- Calan, F. (17 de 11 de 2017). *Unknown Metodologias Agiles*.
- Conexionesan. (10 de 10 de 2018). *Etapas Scrum*.
- EBERGEMENT. (03 de 02 de 2021). *EBERGEMENTWEB*.
- Españon, A. P. (03 de 12 de 2019). *Drew*. Obtenido de <https://blog.wearedrew.co/ventajas-y-desventajas-de-la-metodologia-scrum>
- Genexus. (s.f.). *Formato XML*.
- GlosarioTic. (15 de 09 de 2015). *Tic.portal*.
- Hidalgo, J. (11 de 03 de 2017). *www.hastnew*.
- Hierro, J. (17 de 02 de 2017). *Biblogstecarios*.
- INAMHI. (12 de 03 de 2021). *Instituto Nacional del Meteorología e Hidrología*. Obtenido de nstituto Nacional del Meteorología e Hidrología: <https://www.serviciometeorologico.gob.ec/>
- Kumar, D. (2020). Low Cost IoT enabled Weather Station. *IEEE*, 7.
- Limlahapun, P. (2009). Flood monitoring and early warning system integrating object extraction tool on web-based. *IEEE*, 5.
- Lucas, J. (04 de 09 de 2019). *OpenWebinar Node js*.

Marcos, C. R. (11 de 09 de 2017). Implementación De Una Interfaz SIG Web Para El Despliegue De Pronósticos Meteorológicos Dirigido A La Subdirección De Aviación Civil Del Ecuador. *Universidad De Guayaquil.Facultad De Ciencias Matemáticas Y Físicas.Carrera De Ingeniería En Sistemas Computacionales*, 146.

Meteoblue. (2006). *Variables Meteorológicas* . Obtenido de <https://content.meteoblue.com/es/especificaciones/variables-meteorologicas/temperatura>

RedHat. (s.f.). *API*.

Reviversoft. (2021). *.xsl- .TEXT*.

Ribas, J. (2010). *Disseny Producte*. Obtenido de <https://dissenyproducte.blogspot.com/2017/10/tecnologias-frontend-y-backend-en-el.html>

Robledano, A. (28 de 10 de 2019). *OpenWbinar*. Obtenido de MongoDB.

Significados. (06 de 05 de 2014). Obtenido de <https://www.significados.com/software/>

Tecnoaporte. (07 de 08 de 2018). *Venytajas y Desventajas de un API*.

Turner, M. (2017). Utilizing a Network of Wireless Weather Stations to Forecast Weather in Developing Countries. *IEEE*, 3.

Uriarte, J. M. (13 de 04 de 2020). *"Meteorología"*. Obtenido de <https://www.caracteristicas.co/meteorologia/>.

Vlaanderen, K. J. (2011). *The agile requirements refinery: Applying SCRUM principles to software product management. Information and Software Technology*. Obtenido de <https://doi.org/10.1016/j.infsof.2010.08.004>

Weatherstack. (2021). Obtenido de <https://weatherstack.com/>

WikiRyte. (2020). *CSV*.

Libros

Lopez, I. Z. (2012). *Meteorología y Climatología*. Universidad Nacional de Educación a Distancia.

Wong, K. (2004). *Free / Open Source Software*. Kuala Lumpur: UNDP-APDIP.

ANEXOS

MANUAL TÉCNICO

El manual describe los pasos necesarios para realizar el deploy mediante la herramienta HEROKU, cualquier persona que este familiarizada con proyectos web, bajo cualquier lenguaje de backend, tiene la capacidad de realizar el despliegue de su aplicación en la plataforma de HEROKU.

Es valioso tener en cuenta que en esta guía se realiza una mención especial a las peculiaridades mínimas en las cuales se puso a prueba el instrumento, dado que HEROKU facilita todas las herramientas para extender un proyecto paso a paso, lo único invariable para el manejo, es una conexión a internet permanente, puesto que se trabaja en la nube.

REQUERIMIENTOS MÍNIMOS DE HARDWARE

- Procesador: Intel Core i3
- Memoria RAM: 8 Gigabytes (GB)
- Disco Duro: 1000Gb.

REQUERIMIENTOS MÍNIMOS DE SOFTWARE

- Privilegios de administrador
- Sistema Operativo: Windows8/Windows10

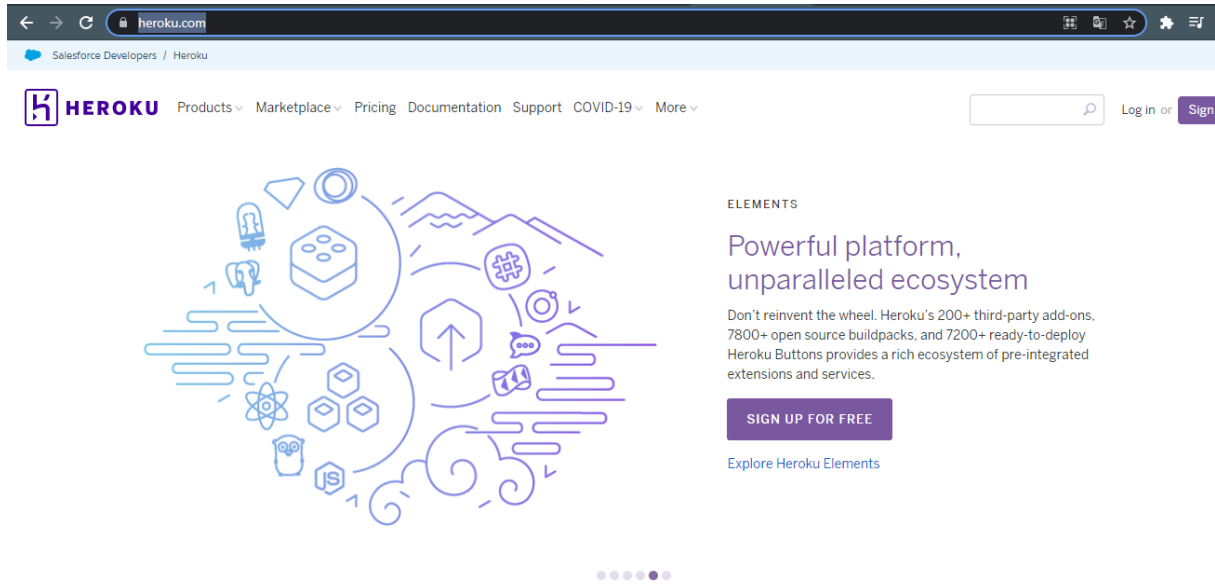
Como se mencionó con anterioridad la aplicación web está desarrollada bajo NodeJS, así que se demostrará a continuación paso a paso como se realizó el despliegue en la plataforma de HEROKU para un desarrollo web con backend en NodeJS.

Instalación de HEROKU

HEROKU como tal es una plataforma web que permite usar su página web para realizar el deploy, entonces no hay necesidad de instalar una interfaz como tal para la manipulación de la

plataforma, ahora si, como primer paso se ingresa al siguiente enlace <https://www.heroku.com/> que se dirige al apartado web de HEROKU, como se puede observar.

Enlace de HEROKU



Enlace de HEROKU. Elaborado por Ken Constante

El siguiente paso es ingresar a la cuenta de HEROKU, si eres un usuario nuevo como este es el caso, ingresarás a la opción sign up, para realizar el registro correspondiente y crear la cuenta en HEROKU, de la siguiente manera.

Registro cuenta en HEROKU

signup.herokuapp.com

Free account

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

Your app platform

A platform for apps, with app management & instant scaling, for development and production.

Deploy now

Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.

First name *
Ken

Last name *
Constante


Email address *
Email address

Company name
UPS

Role *
Student

Country *
Ecuador

Primary development language *
Node.js

I'm not a robot 
reCAPTCHA
Privacy · Terms

CREATE FREE ACCOUNT

Registro cuenta en HEROKU. Elaborado por Ken Constante

Luego de los procesos comunes al crear una cuenta nueva en cualquier plataforma, es decir, verificación del correo, validación de la cuenta, al ingresar ya con las credenciales registradas, la plataforma mostrara una página de esta manera.

Página principal HEROKU

Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Personal

New

Filter apps and pipelines

Página principal HEROKU. Elaborado por Ken Constante


Creación de la aplicación web

Para crear el sistema web, dirigirse al apartado New, el cual este situado en la parte superior derecha de la cuneta la cual se ha creado se selecciona “crear una app” y se llena los campos correspondientes, le damos un nombre al proyecto y una región donde se alojara el sistema web, por ultimo seleccionamos en el botón crear aplicación.

Creación de la aplicación web


Create New App

App name

proyecto-klima is available

Choose a region

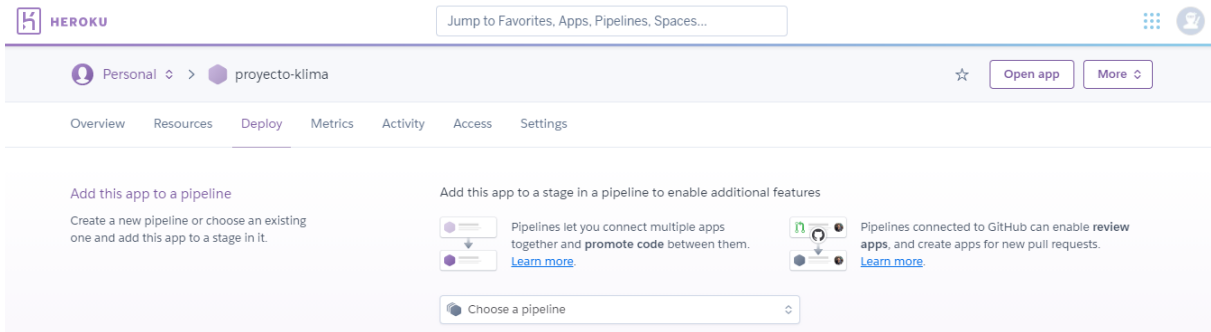
Add to pipeline...

Create app

Creación de la aplicación web. Elaborado por Ken Constante

A continuación, HEROKU mostrara la interfaz con los pasos para realizar el deploy de la aplicación web.

Interfaz de la aplicación en HEROKU

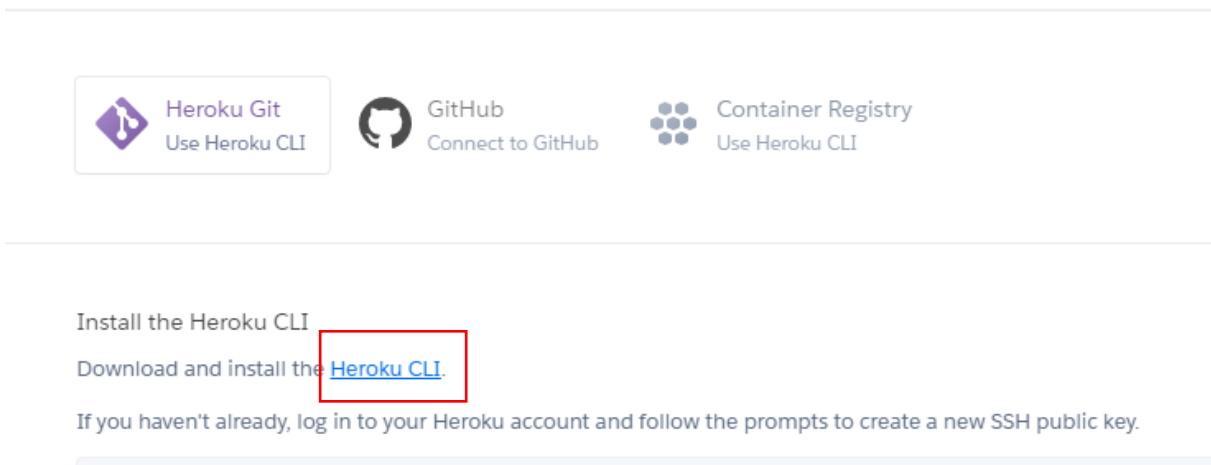


Interfaz de la aplicación en HEROKU. Elaborado por Ken Constante

Instalación de HEROKU CLI

Para cargar un proyecto en HEROKU se necesita instalar HEROKU CLI el cual esta en la misma interfaz que quedo previamente.

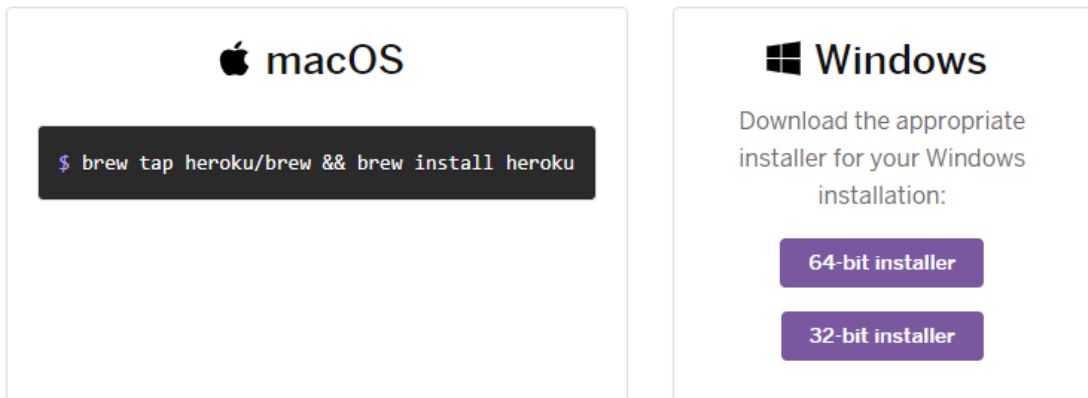
Enlace HEROKU CLI



Enlace HEROKU CLI . Elaborado por Ken Constante

Al dirigirse al enlace de HEROKU CLI, se mostrara una serie de opciones de instalacion para los diferentes sistemas operativos que son MacOS, Windows y Ubuntu.

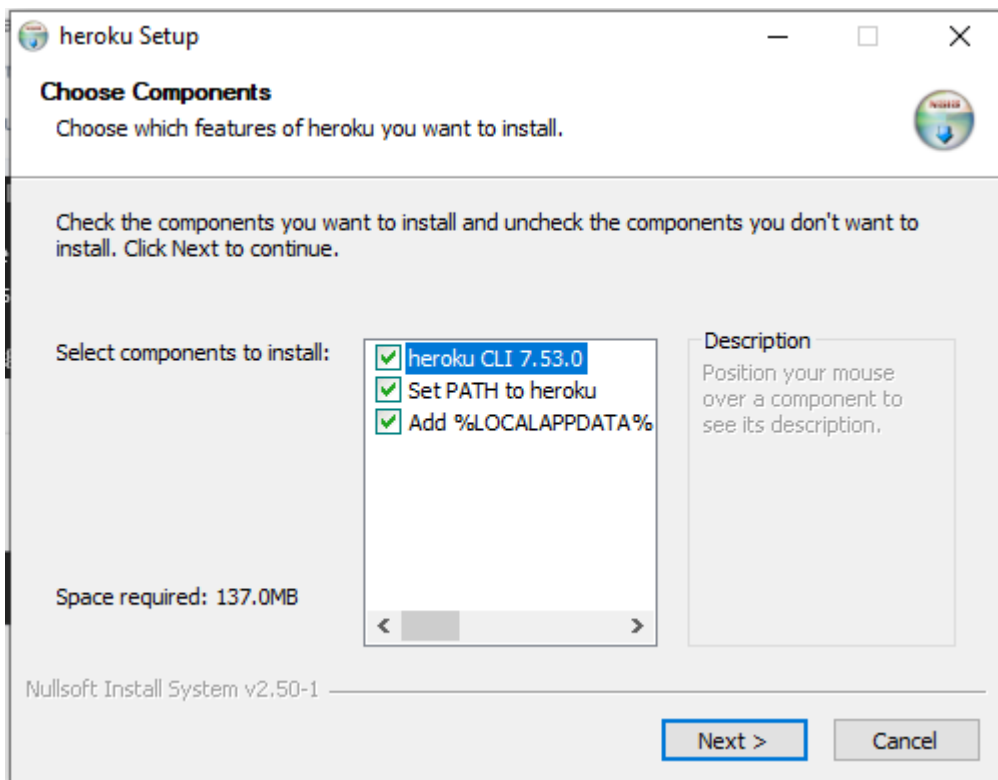
Instalacion de HEROKU CLI en Windows



Instalacion de HEROKU CLI en Windows. Elaborado por Ken Constante

Seleccionamos el sistema operativo de nuestro pc y realizamos la instalación de HEROKU, en la primera ventana del instalador solo se seleccionará la opción siguiente.

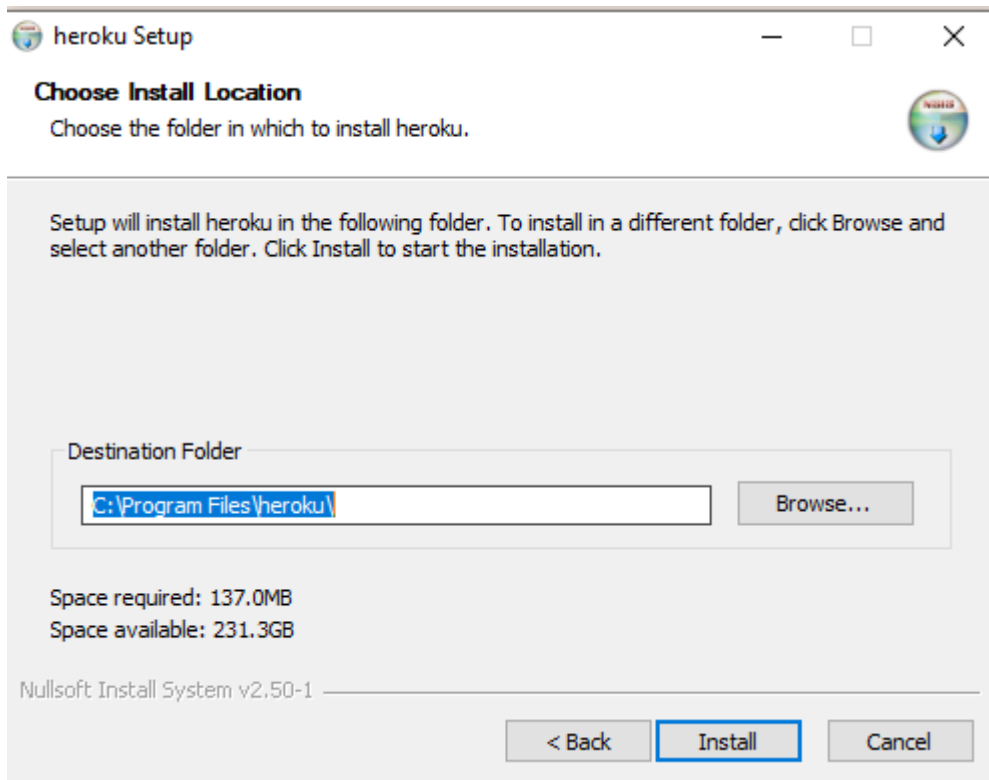
Instalación de HEROKU componentes



Instalación de HEROKU componentes. Elaborado por Ken Constante

Posteriormente se seleccionará el lugar de instalación de la herramienta para iniciar la instalación de HEROKU.

Instalación HEROKU localización



Instalación HEROKU localización. Elaborado por Ken Constante

Una vez que se tiene HEROKU en el pc, hay que moverse a la interfaz de la elaboración de la aplicación y observar los pasos para cargar mediante el terminal de Windows la aplicación en cuestión.

Pasos del deploy en HEROKU

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/  
$ git init  
$ heroku git:remote -a proyecto-klima
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```



You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Pasos del deploy en HEROKU. Elaborado por Ken Constante

Una vez identificado los pasos para realizar el deploy, se abre el terminal de Windows y

```
> heroku login|
```

simplemente se realiza los pasos en consola, primero:

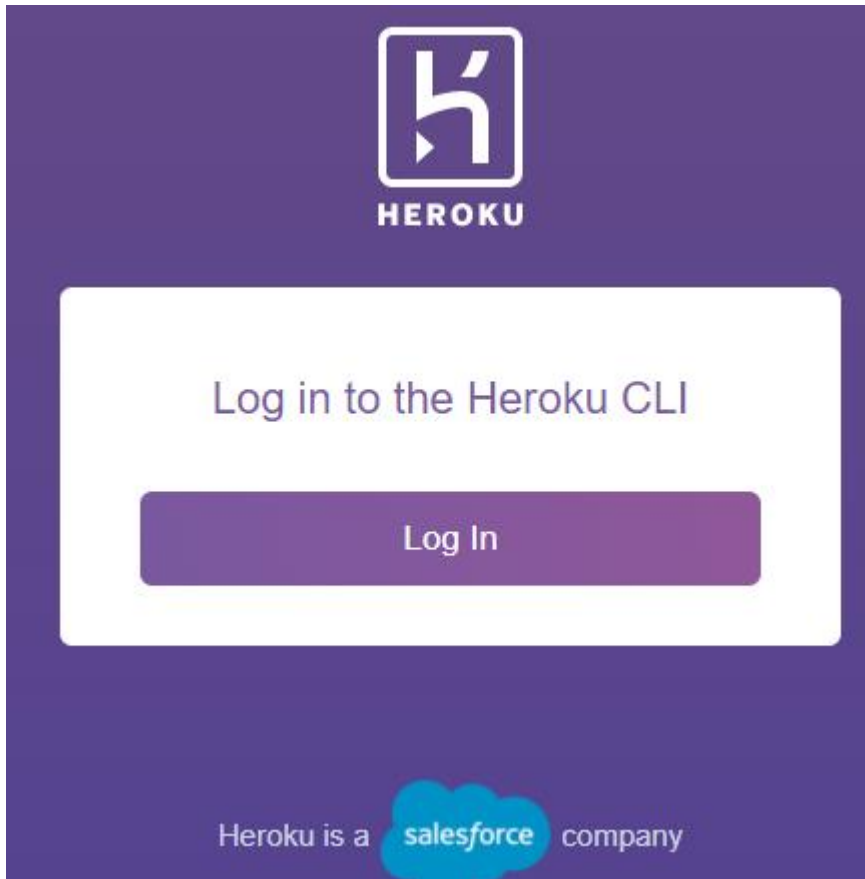
Al realizar este comando en la consola se desplegará la siguiente instrucción.

```
heroku: Press any key to open up the browser to login or q to exit:
```

Se digita cualquier tecla y automáticamente se abrirá el navegador predeterminado para

realizar el login correspondiente de HEROKU, se presiona Login y ya está.

Login en terminal de HEROKU



Login en terminal de HEROKU. Elaborado por Ken Constante

Creación de un nuevo repositorio y deploy

Ahora se dirige a la siguiente instrucción dada por HEROKU la cual es acceder al directorio del proyecto para levantarlo.

```
cd app-klima|
```

Posteriormente la siguiente instrucción inicializar git con la siguiente instrucción.

```
git init|
```

Y finalmente se ejecuta la última instrucción dada por HEROKU que es para conectar con la aplicación creada en la plataforma de HEROKU.

```
heroku git:remote -a proyecto-klima
```

Para realizar el deploy se realiza solo el ingreso de las instrucciones dadas por la propia plataforma de HEROKU.

Primero se añade al proyecto.

```
git add .
```

Luego se realiza el commit.

```
git commit -am "make it better"
```

Y finalmente el push para terminar con el deploy y generar la aplicación web.

```
git push heroku master
```

Una vez realizado estos pasos se puede observar la aplicación creada.

Deploy en HEROKU

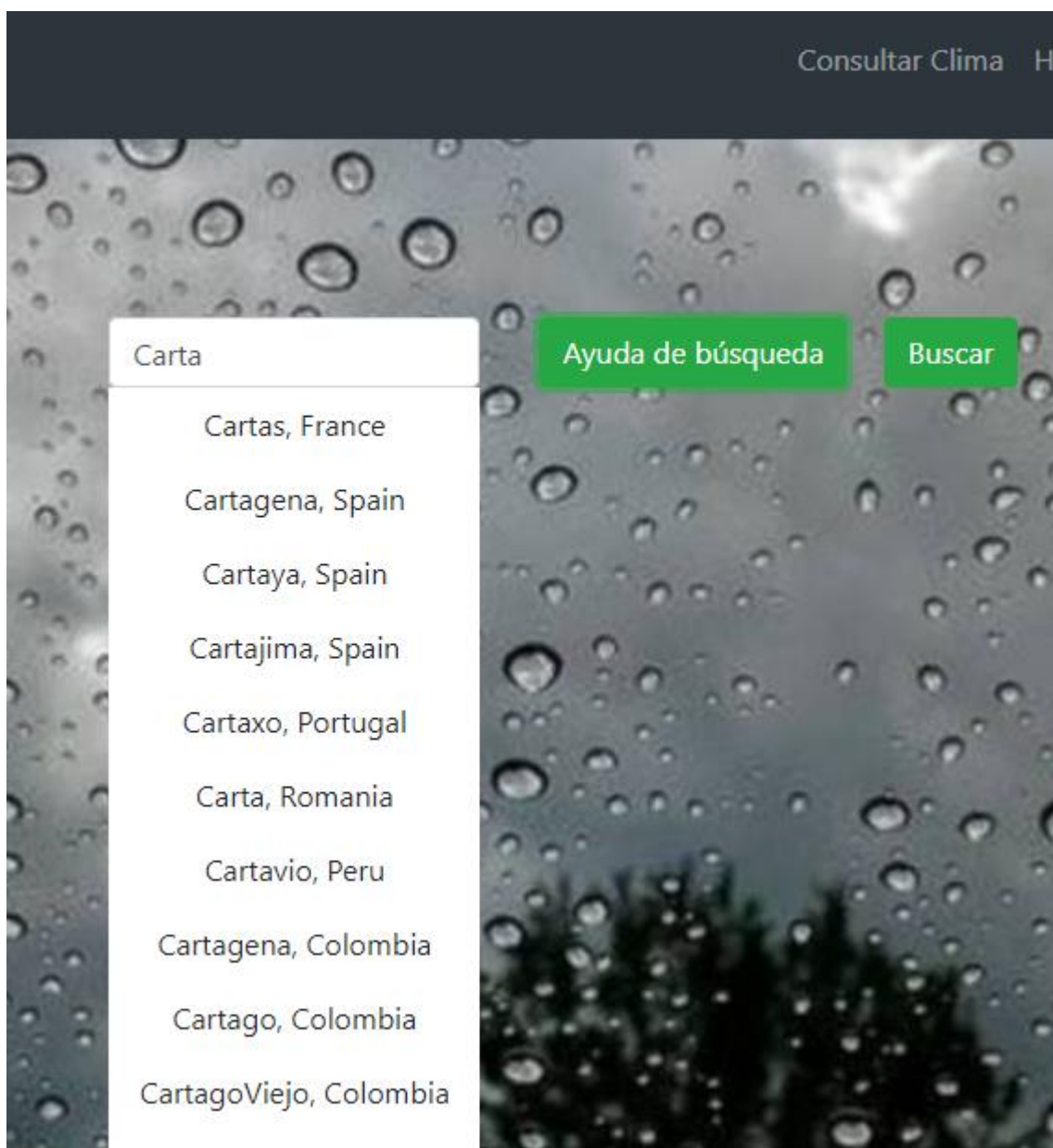


Deploy en HEROKU. Elaborado por Ken Constante

Búsqueda del clima

La aplicación web es muy sencilla de usar debido a que presenta campos de texto con labels que describen las funcionalidades de la aplicación y como iniciar una nueva búsqueda, como primer paso es ingresar una ciudad en la búsqueda del clima, si se tiene algún problema con el sitio en específico se puede ingresar parte del parámetro y seleccionar la opción ayuda de búsqueda.

Búsqueda clima



Búsqueda clima. Elaborado por Ken Constante

Luego se realiza la búsqueda del sitio elegido y listo se obtiene el clima de la ciudad de interés.

Búsqueda exitosa del clima



Búsqueda exitosa del clima. Elaborado por Ken Constante

Para consultar datos históricos la interfaz basta con solo bajar un poco la página hasta llegar a la opción de Consulta datos históricos y llenaron los campos correspondientes para observar una gráfica de los parámetros ingresados.

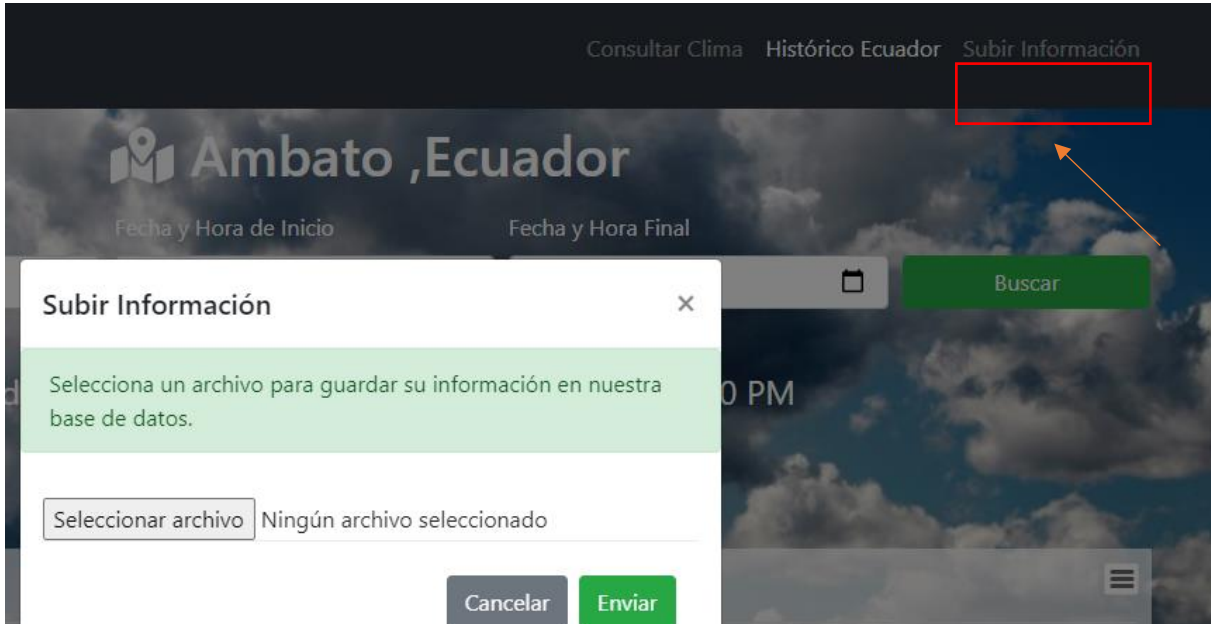
Clima y grafica de los datos históricos



Clima y grafica de los datos históricos. Elaborado por Ken Constante

Para finalizar, para cargar información a la base de la página web, basta con seleccionar la opción subir información que está situada en el header del sistema.

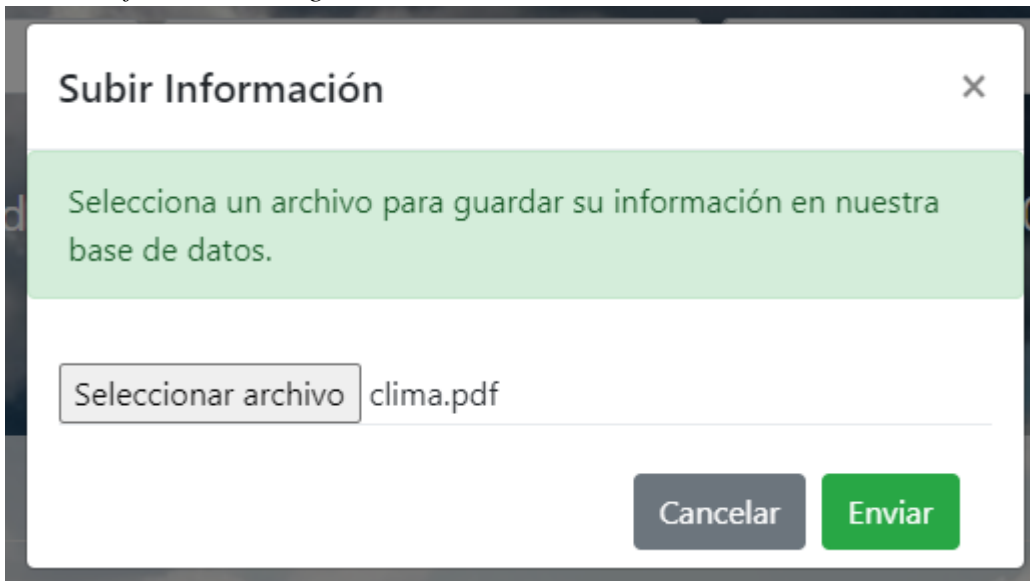
Opción subir información a la página web



Opción subir información a la página web. Elaborado por Ken constante

Y listo se carga la información y posterior a eso se realiza inicializa la carga con el botón enviar, la página cargara la información en cuestión.

Información cargada en la base



Información cargada en la base. Elaborado por Ken Constante