



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

**Implementación de una aplicación SIG
WEB para la visualización de puntos de
calor detectados por el satélite
geoestacionario GOES-16/ABI**

TESIS

Que para obtener el título de

Ingeniero Geomático

P R E S E N T A

Jesús Hernández Torres

DIRECTOR(A) DE TESIS

Dra. Lilia de Lourdes Manzo Delgado



Ciudad Universitaria, Cd. Mx., 2023



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedico este trabajo a madre y hermanas porque sin ellas no podría haber culminado este logro en mi vida.

“El mundo entero se aparta ante un hombre que sabe adónde va”.

Antoine de Saint-Exupéry

Agradecimientos

- Gracias a mi amada madre, por todo el apoyo y amor dado en el camino. Por todos sus consejos y valores, siempre guiándome por el camino correcto. Sin ella no sería posible este logro. Te amo mamá.
- A mis hermanas, me brindaron el amor de una segunda madre. Siempre han estado al tanto de mi persona y de mi seguridad, las amo.
- A mis sobrinas y sobrino, que me apoyaron en muchas noches de desvelo obsequiándome su compañía y atención.
- A la Dra. Lilia de Lourdes Manzo Delgado, por el apoyo incondicional en el ámbito académico y personal, desde que entré al Instituto de Geografía a realizar mi servicio social. Por su guía, paciencia y tiempo en el desarrollo de este proyecto.
- A mi alma máter, la Universidad Nacional Autónoma de México (UNAM) y mi querida Facultad de Ingeniería (FI), por darme la oportunidad de conocer nuevos lugares, personas que contribuyeron en el proceso de mi formación profesional.
- Al Instituto de Geografía (IG) y al Dr. Jorge Prado Molina.
- A cada uno de los profesores que compartieron conmigo sus experiencias y sabiduría.
- A queridos amigos: Aníbal, Areli, Colvert, Karen, Ricardo, Rocío, Salvador y Uriel. Me hicieron ver la carrera desde otra perspectiva y con ustedes viví momentos que jamás olvidaré. Gracias amigos, son mi familia.
- A mis profesores y sinodales: M.I. Adolfo Reyes Pizano, M.C.T. Maria Elena Osorio Tai, Ing. Neith Moreno Rodríguez y la Ing. Ana Lilia Salas Alvarado, por todo el apoyo y conocimiento brindado en las aulas y haber aceptado ser parte del grupo de sinodales de la presente tesis.
- A mi Fer, gracias por el apoyo incondicional que me brindas día con día. Por darme ánimos en mis etapas de debilidad e inspirarme a seguir adelante.
- A los ingenieros Manuel de Jesús Nava y Fernando García, por compartir sus conocimientos y brindarme su apoyo en mi estancia en Milpillas, Sonora.
- Para finalizar, a todas aquellas personas que pusieron su granito de arena para el desarrollo de este proyecto y muchas veces se mantienen en el anonimato. A toda la comunidad que comparte sus conocimientos en blogs, páginas de internet, videos, etc. Sin ustedes no sería posible este trabajo.

Contenido

Abreviaturas.....	7
1 Introducción.....	9
1.1 Antecedentes.....	9
1.2 Planteamiento del problema.....	9
1.3 Objetivo General.....	11
1.4 Objetivos Particulares.....	11
2 Marco conceptual SIG WEB.....	12
2.1 Arquitectura y componentes.....	14
2.2 Tecnologías para la creación de un SIG WEB.....	15
2.2.1 PostgreSQL.....	16
2.2.2 Servicios WEB Geoespaciales.....	16
2.2.3 Geoserver.....	19
2.2.4 HTML5, CSS3 y JavaScript.....	20
2.2.5 JQuery.....	26
2.2.6 Openlayers3.....	27
2.2.7 Django.....	28
2.2.8 Apache HTTP Server.....	29
2.3 Aplicaciones SIG WEB como alerta de incendios.....	30
2.3.1 Alerta temprana de incendios de la CONABIO.....	31
2.3.2 Sistema de Predicción de Peligro de incendios Forestales de México (CONAFOR- CONACYT 2014-2-252620).....	33
2.3.3 TerraMA2.....	35
2.3.4 FIRMS FIRE MAP.....	37
3 Algoritmo de detección de puntos de calor GOES-16/ABI y sus productos.....	39
3.1 Creación de Umbrales.....	39
3.2 Obtención de puntos de calor.....	40
3.3 Generación de productos.....	42
4 Metodología.....	44

4.1 Base de Datos de puntos de calor GOES-16/ABI.....	45
4.1.1 Creación de Base de Datos espacial y tablas con PostgreSQL.....	46
4.1.2 Alimentación de Base de Datos con Python3.....	51
4.2 Visualización en Geoserver.....	54
4.2.1 Creación de espacio de Trabajo.....	54
4.2.2 Almacén de datos.....	55
4.2.3 Creación de capas y Consultas.....	55
4.2.4 Estilos.....	58
4.3 Desarrollo WEB.....	59
4.3.1 Estructura de directorios.....	60
4.3.2 Página principal.....	62
4.3.3 Visualizador de puntos de calor.....	64
5 Resultados.....	69
5.1 Página principal.....	70
5.1.1 Estadísticas.....	73
5.2 Visualizador puntos de calor GOES-16/ABI.....	75
5.2.1 Interfaz del mapa.....	76
5.2.2 Menú Puntos de Calor.....	77
5.2.3 Capas del visualizador.....	78
Vectoriales.....	78
Mapas base.....	80
Ráster.....	81
5.2.4 Simbología.....	82
5.2.5 Barras de herramientas.....	83
6 Conclusiones.....	91
Anexos.....	92
A. Scripts Python.....	92
B. Scripts Shell.....	107
Referencias.....	108

Índice de Figuras

Figura 2.1.1:Arquitectura SIG WEB.....	14
Figura 2.2.1: Interfaz de Geoserver versión 2.15.2, apartado “Acerca de Geoserver”.....	19
Figura 2.2.2: Esquema de un documento HTML.....	22
Figura 2.2.3: Estructura esencial de una página web.....	22
Figura 2.3.1: Consulta de puntos de calor en portal de Alerta Temprana de CONABIO comprendida entre las fechas del 08-10-20 a 09-10-20.....	33
Figura 2.3.2: Incendios esperados para el 09-10-20 por el Sistema de Predicción de Peligro de Incendios Forestales en México.....	35
Figura 2.3.3: Diagrama de los principales módulos del Sistema de Monitoreo, análisis y alerta (TerraMA2).....	37
Figura 2.3.4: Plataforma SIG WEB de FIRE MAP.....	38
Figura 3.2.1: Máscaras de: A) Manchas Urbanas, B) Suelo desnudo y C) Cuerpos de agua. Generadas por el producto MCD12Q1, clasificación IGBP anual.....	41
Figura 3.2.2: Ejemplo de Pixeles potenciales de T3.9 del 11 de mayo 2020 de las 19:00:00 UTC.	41
Figura 4.1: Metodología dividida en Backend y Frontend.....	45
Figura 4.1.1: Parte del archivo CSV arrojado por el algoritmo correspondiente a 01/10/2020.....	46
Figura 4.1.2: Parte del archivo CSV procesado por la Base de Datos Espacial.....	46
Figura 4.1.3: Información de la tabla puntos_10m en PostgreSQL.....	49
Figura 4.1.4: Información de la tabla puntos_diacom en PostgreSQL.....	50
Figura 4.1.5: Diagrama de copia de archivos CSV entre el servidor servidor2 y servidor1.....	51
Figura 4.2.1: Espacio de trabajo denominado <i>fires</i>	54
Figura 4.2.2: Parámetros de conexión de Geoserver con PostGIS.....	55
Figura 4.2.3: Sentencia CQL y Visualización de la capa Puntos_día en Openlayers3 comprendida de 2021-09-02 al 2021-09-10.....	57
Figura 4.3.1: Árbol de directorios, servidor1.....	60
Figura 4.3.2: Diagrama de directorios, representado por colores. Morado:nombre de la cuenta de usuario, Rojo: proyecto y aplicación de Django, Azul: Directorios, Verde: Archivo ejecutable de Python y Naranja: Archivos en HTML.....	61

Figura 4.3.3: Gráfica de barras de los puntos de calor de los últimos 7 días (20 al 27 de octubre del 2021).....	63
Figura 4.3.4: Despliegue de descarga de puntos de calor de las últimas 24 horas. A partir del 2022-09-20, 17:20 GMT.....	68
Figura 4.3.5: Desarrollo de visualizador de Puntos de calor GOES-16/ABI.....	68
Figura 5.1.1: Página principal, apartado de Algoritmo de puntos de calor. Desplegada desde Google Chrome.....	70
Figura 5.1.2: Botón de acceso a visualizador.....	71
Figura 5.1.3: Página principal, apartado de contacto. Desplegada desde Google Chrome.....	71
Figura 5.1.3: Página principal, apartado de Visualizador de puntos de calor. Desplegada desde Google Chrome.....	71
Figura 5.1.4: <i>Footer</i> de página principal. Desplegada desde Chrome.....	72
Figura 5.1.5: Gráfica de número de puntos de calor por mes en el año 2021. Tomada el 27 de octubre de 2021.....	73
Figura 5.1.6: Gráfica de número de puntos de calor en los últimos 7 días. Tomada el 27 de noviembre de 2021.....	73
Figura 5.1.7: Gráfica con los estados con mayor número de puntos de calor detectados. Tomada el 27 de octubre del 2021.....	74
Figura 5.2.1: Animación de carga para el despliegue de la interfaz gráfica del visualizador de puntos de calor GOES-16/ABI.....	75
Figura 5.2.2: Interfaz del visualizador de puntos de calor, del 17 de agosto del 2022.....	76
Figura 5.2.3: Despliegue de interfaz del visualizador. a) Versión celular y b) Versión tableta.....	77
Figura 5.2.4: Pestañas de menú del Visualizador de Puntos de calor GOES-16/ABI.....	78
Figura 5.2.5: Despliegue de capas vectoriales.....	79
Figura 5.2.6: Puntos de calor del 1 de diciembre 2021 al 10 de diciembre 2021. En los estados de San Luis Potosí, Tamaulipas y Veracruz de Ignacio de la Llave.....	80
Figura 5.2.7: Apartado de Mapas base en visualizador de puntos de calor GOES-16/ABI.....	80
Figura 5.2.8: Despliegue de capas ráster en el visualizador de puntos de calor GOES-16/ABI.....	81
Figura 5.2.9: Representación del apartado de Simbología del visualizador de puntos de calor GOES-16/ABI.....	82

Figura 5.2.10: Barra de herramientas del visualizador de puntos de calor GOES-16/ABI.....	83
Figura 5.2.11: Herramienta de medición, opción distancia.....	84
Figura 5.2.12: Herramienta de medición, opción área.....	84
Figura 5.2.13: Herramienta de conteo de puntos de calor del visualizador de puntos de calor GOES-16/ABI.....	85
Figura 5.2.14: Ejemplo de: a) Conteo de puntos de calor detectados en toda la escena el 22-09-2022 y los puntos de calor contenidos por el rectángulo dibujado por el usuario. b) Rectángulo dibujado que contiene los puntos de calor.....	86
Figura 5.2.15: Herramienta de consulta de información con ventas emergentes para la capa de Áreas Naturales Protegidas 2020.....	87
Figura 5.2.16: Herramienta de gráficas del visualizador de puntos de calor GOES-16/ABI. Gráfica de los últimos 7 días detectados por el algoritmo a partir del 03-10-2022.....	88
Figura 5.2.17: Herramienta de gráficas del visualizador de puntos de calor GOES-16/ABI. Gráfica de dona de los puntos de calor detectados por mes, hasta el 05-10-2022.....	89
Figura 5.2.18: Herramienta de gráficas del visualizador de puntos de calor GOES-16/ABI. Gráfica de los estados con mayor puntos de calor de los últimos siete días detectados por el algoritmo.	89
Figura 5.2.19: Resultado de herramienta de tabla de atributos en visualizador de puntos de calor GOES-16/ABI.....	90

Índice de Tablas

Tabla 2.2.1: Comparación de Free Software, OpenSource y Software de Propietario.....	15
Tabla 2.2.2: Características de Estándares de Servicios Web Espaciales de la OGC.....	18
Tabla 2.2.3: Características principales de CSS, CSS2 y CSS3.....	24
Tabla 2.2.4: Características principales de Openlayers según OSGEO.....	27
Tabla 2.3.1: Información disponible en base de datos por el Sistema de Alerta Temprana CONABIO para territorio mexicano.....	32
Tabla 3.3.1: Atributos del producto de incendios GOES-16/ABI.....	43
Tabla 4.1.1: Campos y tipos de datos de la tabla puntos_10m.....	48

Tabla 4.1.2: Campos y tipos de datos de la tabla puntos_diacom.....	50
Tabla 4.1.3: Scripts que alimentan las tablas de las bases de datos de puntos de calor.....	52
Tabla 4.2.1 : Capas en Geoserver.....	56
Tabla 4.2.2: Formatos de estilos aceptados por Geoserver.....	58
Tabla 4.2.3: Estilos de capas en Geoserver.....	59
Tabla 4.3.1: Tablas para estadísticas.....	63
Tabla 4.3.2: Capas de visualizador de puntos de calor por categoría.....	64
Tabla 4.3.3: Herramientas del visualizador de puntos de calor GOES-16/ABI.....	67

Abreviaturas

ABI	Advanced Baseline Imager	GMT	Greenwich Mean Time
ACID	Atomicity, Consistency, Isolation, Durability	GOES	Geostationary Operational Environmental Satellites
ASF	Apache Software Foundation	HTML	HyperText Markup Language
CDN	Content Delivery Network	HTTP	HyperText Transfer Protocol
CERN	Conseil Européen pour la Recherche Nucléaire	IMS	Internet Map Server
CONABIO	Comisión Nacional para el Conocimiento y Uso de la Biodiversidad	INPE	Instituto Nacional de Pesquisas Espaciais
CONACYT	Consejo Nacional de Ciencia y Tecnología	ISC	Índice de Sequedad del Combustible
CONAFOR	Comisión Nacional Forestal	JSON	JavaScript Object Notation
CQL	Common Query Language	KML	Keyhole Markup Language
CSS	Cascading Style Sheets	LANCE	Land Atmosphere Near-real-time Capability for EOS
CSV	Comma Separated Values	LANOT	Laboratorio Nacional de Observación de la Tierra
DB	Data Base	MBSTYLE	MapBox Style
DOM	Document Object Model	MODIS	Moderate Resolution Imaging Spectroradiometer
DSF	Django Software Foundation	NASA	National Aeronautics and Space Administration
ECMA	European Computer Manufacturers Association	NETCDF	Network Common Data Form
EOS	Earth Observing System	NOAA	National Oceanic and Atmospheric Administration
EPSG	European Petroleum Survey Group	OGC	Open Geospatial Consortium
FIRMS	Fire Information for Resource Management System	OSGEO	Open Source Geospatial Foundation
FSF	Free Software Foundation	PNG	Portable Network Graphics

RGB	Red, Green, Blue
SHP	Shapefile
SIG	Sistemas de Información Geográfica
SLD	Style Layer Descriptor
SMN	Servicio Meteorológico Nacional
SQL	Structured Query Language
TXT	Archivo de texto
UNAM	Universidad Nacional Autónoma de México
USGS	United States Geological Survey
UTC	Universal Time Coordinated
VIIRS	Visible Infrared Imaging Radiometer Suite
W3C	World Wide Web Consortium
WCS	Web Coverage Service
WFS	Web Feature Service
WGS84	World Geodetic System 84
WMS	Web Map Service
WMTS	Web Map Tile Service

1 Introducción

1.1 Antecedentes

Los incendios forestales de gran extensión y duración se han convertido en un fenómeno global de gran relevancia por las afectaciones ambientales y sociales que ocasionan; por lo tanto, cada país debe tener un sistema avanzado para detectar y monitorear los incendios en tiempo casi real. En la actualidad, México cuenta con un sistema de alerta temprana de incendios que proporciona información cada doce horas derivadas de imágenes MODIS¹ y VIIRS²; sin embargo, se requiere contar con un sistema de alarma que proporcione información de incendios con mayor frecuencia. Ante esta situación, el Laboratorio Nacional de Observación de la Tierra (LANOT) del Instituto de Geografía de la UNAM, desarrolló un algoritmo para detectar incendios cada 10 minutos, representados como puntos de calor, utilizando imágenes del satélite geostacionario GOES-16/ABI, cuyos resultados se generan en formatos PNG³, CSV⁴, Shapefile⁵, además de animaciones de puntos de calor detectados cada 24 horas. Considerando los productos mencionados, esta tesis abordará el diseño y la implementación de una aplicación SIG WEB en el LANOT, utilizando herramientas OpenSource, para visualizar, consultar y distribuir información de los puntos de calor GOES-16⁶/ABI a dependencias gubernamentales que tienen a su cargo realizar acciones de prevención y combate de incendios, al igual que a usuarios interesados en el tema.

1.2 Planteamiento del problema

En México, la temporada de incendios ocurre durante la primavera antes de las lluvias de verano entre enero y junio, siendo más intensa entre marzo y mayo (Rodríguez-Trejo, 2014). En los últimos 20 años

1 MODIS (Moderate Resolution Imaging Spectroradiometer). Sensor montado en los satélites Terra y Aqua.

2 VIIRS (Visible Infrared Imaging Radiometer Suite). Sensor montado en la plataforma Suomi-NPP.

3 PNG (Portable Network Graphics). Formato gráfico basado en un algoritmo que permite comprimir un mapa de bits sin pérdidas.

4 CSV (Comma Separated Values). Archivo de texto plano separados por comas.

5 Shapefile es el formato de representación vectorial desarrollado por ESRI.

6 GOES-16 (Geostationary Operational Environmental Satellite). Satélite de órbita geostacionaria lanzado el 19 de septiembre del 2019. Con la misión de obtener información relacionada con fenómenos meteorológicos del continente Americano.

(2000 – 2019) el país ha registrado un promedio anual de 7,878 incendios y 316,615 ha afectadas. En el mismo periodo, los años 2011, 2017 y 2019 presentaron incrementos notables en la superficie de afectación: 954,796 ha, 726,361 ha y 633,678 ha respectivamente (CONAFOR, 2019).

Dada la problemática que representa para el país el tema de los incendios, los tomadores de decisiones deben contar con sistemas de alerta que les permitan monitorear el inicio, avance y humo generado por los incendios. Actualmente el país cuenta con el sistema de alerta implementado por la Comisión Nacional para el Conocimiento y Uso de la Biodiversidad (CONABIO) y con el sistema de predicción de peligros de incendios forestales implementado recientemente por la Universidad Juárez de Durango financiado por la Comisión Nacional Forestal (CONAFOR) (Vega Nieva et al., 2019). Ambos sistemas se basan en los puntos de calor MODIS y VIIRS, detectados con imágenes de satélites polares, que proporcionan información cada 12 horas. Sin embargo, durante los meses de mayor frecuencia de incendios, es necesario contar con un sistema que proporcione información en los horarios que no están cubiertos por dichos satélites. Una alternativa para incrementar la frecuencia del monitoreo es implementar un sistema de monitoreo en tiempo casi real utilizando los puntos de calor GOES-16/ABI que se detectan con imágenes de un satélite geoestacionario en el LANOT.

La implementación de un sistema de monitoreo es una tarea complicada que requiere de al menos un Sistema de Información Geográfica (SIG) para distribuir la información y un navegador en internet (WEB) para que el cliente pueda tener acceso a la información, cuya integración en una aplicación se denomina SIG WEB. En la actualidad existen diversas herramientas OpenSource que agilizan la implementación de las aplicaciones SIG WEB utilizando sistemas a bajo costo, como PostgreSQL, servicios WEB Geoespaciales, lenguaje de programación Python, Geoserver, HTML, CSS3 y JavaScript.

La presente tesis forma parte de un trabajo en conjunto con la detección de puntos de calor por el LANOT, mediante un algoritmo utilizando imágenes GOES-16/ABI, plasmado en la tesis de Colvert Gómez Rubio: “Implementación y programación de un algoritmo de detección de incendios forestales para México utilizando imágenes satelitales GOES-16/ABI con software libre”.

1.3 Objetivo General

- Crear una aplicación SIG WEB utilizando herramientas *OpenSource* con la finalidad de visualizar y descargar información espacial y temporal de los puntos de calor GOES-16/ABI para el Laboratorio Nacional de Observación de la Tierra (LANOT) del Instituto de Geografía de la UNAM.

1.4 Objetivos Particulares

- Explorar las aplicaciones SIG WEB disponibles para otros productos de incendios forestales.
- Definir la estructura de la aplicación, tomando en cuenta los servidores del LANOT y las necesidades de los usuarios.
- Diseñar la estructura, funciones y componentes del interfaz para que el usuario consulte, solicite y descargue diversos productos relacionados con los puntos de calor detectados por algoritmo implementado por el LANOT.

2 Marco conceptual SIG WEB

Para Sastre (2010) un SIG en internet se utiliza para localizar servicios, trazar rutas, buscar direcciones, publicar Atlas electrónicos, identificar características geográficas, notificar fenómenos naturales, acceder a Bases de Datos de Organismos Públicos, análisis demográficos, identificar y visualizar datos procedentes a la teledetección, entre otros.

De acuerdo con Fu (2018), la web eliminó la restricción de la distancia en el ciberespacio, permitiéndoles a los usuarios poder acceder a las aplicaciones SIG de una manera más fácil a nivel mundial. Incluyen diversas tecnologías como lo son: protocolo de transferencia de hipertexto (HTTP, *HyperText Transfer Protocol*), lenguaje marcado de hipertexto (HTML), localizador uniforme de recursos (URL, *Uniform Resource Locator*), JavaScript, biblioteca de gráficos web (WebGL, *Web Graphics Library*), *WebSocket* entre otros.

Normalmente se relaciona el término SIG web con la cartografía web, dado que este añade funcionalidad con los análisis espaciales y técnicas de geoprocésamiento. La tecnología es la base fundamental de los Sistemas de Información Geográfica (SIG) y del internet (web). Por lo tanto, si la tecnología avanza estos muestran crecimiento. En las últimas décadas se han destacado la manera en que se visualizan los datos, hoy en día se pueden observar datos dinámicos casi en tiempo real en mapas interactivos de fácil comprensión. Según Li et al. (2008) las tecnologías se han implementado principalmente para la creación de aplicaciones con fines:

- De acceso y diseminación de datos espaciales
- Exploración y visualización de datos espaciales
- Procesamiento, análisis y modelado de datos espaciales
- Apoyo colaborativo y decisiones espaciales utilizando un SIG
- Integración de servicios geoespaciales en la web basados en procesos y entornos informáticos convencionales y empresariales.

Citando a Fu (2018), el SIG web cuenta con ciertas características que son muy distintivas a su símil de escritorio, dichas características son:

- **Alcance global:** Al encontrarse en la red, se puede visualizar en cualquier parte del mundo que cuente con conexión a ésta. Compartiendo así información con cualquier usuario interesado.
- **Gran cantidad de usuarios:** Todo usuario que sea compatible con la tecnología del SIG podrá acceder a la información.
- **Bajo costo por usuario:** Su costo es mucho menor que el crear una aplicación SIG de escritorio, distribuirla e instalarla para cada usuario. Además existen diversas herramientas libres que permiten crear una plataforma SIG Web que satisfaga las necesidades y gustos del desarrollador o usuario.
- **Mejores capacidades multiplataforma:** Las apps Web fundamentadas en JavaScript principalmente pueden ejecutarse y visualizarse en diversos navegadores de escritorio.
- **Fácil de usar:** Normalmente las aplicaciones SIG Web contienen una interfaz amigable para el usuario, de fácil intuición y gran diseño.
- **Fácil de mantener:** Al estar contenido en la Web, la aplicación se puede mantener y actualizar cuando sea necesario y los usuarios podrán obtener los beneficios que estos conlleven. Dicho de otra manera, si la aplicación necesita de una actualización, ésta se realizará en el servidor donde se encuentre montada y se verá reflejada en cada uno de los usuarios que la visiten sin tener la necesidad de actualizar la app en cada uno de los usuarios.

Con respecto a las funcionalidades que nos brinda un SIG Web destacan, según Fu y Sun (2011):

- **Visualización (mapping) y consultas:** Cuando el usuario requiere información específica ya sea de un fenómeno natural, una dirección, la ruta más corta para llegar a un destino, recursos naturales, etc. Es necesario que el SIG realice una serie de consultas en la base de datos para poder visualizar en el mismo SIG los resultados en pantalla por medio de un mapa.
- **Colectar información espacial:** En el transcurso que ha experimentado la evolución de los SIG Web, una de las características que se han resaltado es la capacidad que han adquirido los usuarios para obtener información geoespacial, procesarla y distribuirla de manera voluntaria en una plataforma SIG Web. Esto es un gran aporte para los usuarios futuros que puedan necesitar de dicha información, alimentando así las bases de datos espaciales de cada aplicación. Un ejemplo son los datos que suben los usuarios al poner a la vista su negocio en la plataforma de GoogleMaps.

- **Diseminar información geoespacial:** La web puede interpretarse como el vínculo entre lo que el usuario busca por necesidad y la información establecida en un servidor que contiene la información. En el caso de los datos espaciales se pueden representar mediante geoportales incentivando la organización de la información.
- **Análisis geoespacial:** Se han desarrollado diversas herramientas por medio de las bibliotecas Openlayers y Leaflet que brindan funciones muy útiles para la interpretación de información, tales como medir distancias, zoom, polígonos de áreas de interés, rutas óptimas, localizar lugares, encontrar la ubicación específica en coordenadas o viceversa, etc.

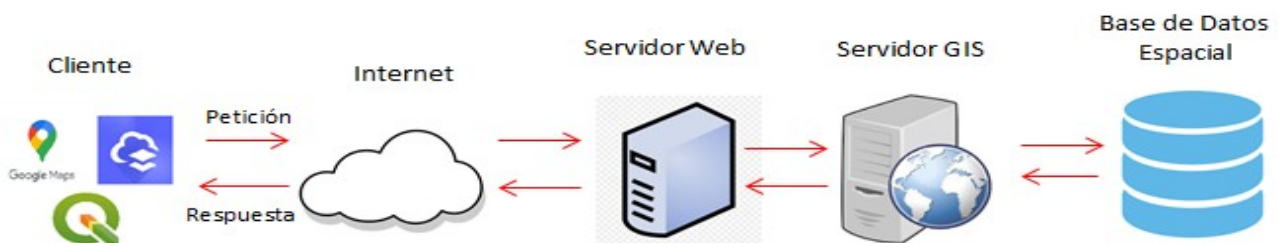
2.1 Arquitectura y componentes

Alesheikh et al. (2002) menciona a la arquitectura de un SIG Web como un flujo de información descrito de la siguiente manera:

- **Cliente (C):** En él se comienza el ciclo, el cliente (navegador, buscador Web, Aplicación móvil) envía una petición al servidor Web a través del internet vía HTTP. De igual manera muestra los resultados enviados por el Servidor Web.
- **Servidor Web (SW):** Reenvía las peticiones hechas por el cliente al Servidor SIG. También hace lo propio con los resultados que arroja el servidor SIG enviándolos al cliente.
- **Servidor SIG (SG):** Recibidas las peticiones del Servidor Web, el Servidor SIG la procesa con la base de datos geográfica (DB SIG); la petición puede incluir desde una consulta, una visualización o un análisis de los datos. Los datos obtenidos, se envían de nuevo al Servidor Web.

Figura 2.1.1

Arquitectura SIG WEB



2.2 Tecnologías para la creación de un SIG WEB

Se pueden encontrar infinidad de aplicaciones tecnológicas empleadas para la construcción de un Sistema de Información Geográfica WEB. Estas aplicaciones pueden provenir desde un software libre (*Free Software*), uno de código abierto (*OpenSource*) o de un propietario (Licencia). A pesar de que los softwares libres y de código abierto pueden tener ciertas similitudes como el dar mayor libertad al desarrollo y transparencia, difieren en ciertos aspectos. La siguiente (Tabla 2.2.1) presenta un resumen de las características principales de los tres tipos de softwares.

Tabla 2.2.1

Comparación de Free Software, OpenSource y Software de Propietario.

Software Libre (Free Software)	Código Abierto (OpenSource)	Software de propietario (Licencia)
	Basada por la Open Source	
Cimentada por la Free Software Foundation (FSF) en 1985, tiene las siguientes libertades:	Initiative (OSI) contiene en su ética los siguientes requisitos:	Quien es el responsable de su desarrollo tiene los derechos exclusivos de su código fuente y comercialización. Los usuarios tienen acceso a su versión en binario.
<ul style="list-style-type: none"> • Ejecutar el programa • Cambiar el programa • Redistribuir copias • Distribuir copias de sus versiones modificadas a terceros 	<ul style="list-style-type: none"> • Libre redistribución • Inclusión de su código fuente • Licencias con permisos para modificaciones y derivados • Libre distribución de licencia 	Para su compra es necesario pagar una licencia la cual tiene limitada su distribución.

Nota: Tomada de (Yanza, 2013)

Como se mencionó anteriormente, existen diversas tecnologías que integran un Web SIG, divididas entre softwares, lenguajes de programación, servidores SIG, APIS, Bibliotecas, etc. En los siguientes subcapítulos mencionaremos los utilizados en esta tesis para realizar la aplicación.

2.2.1 PostgreSQL

Sistema de base de datos (DB, por sus siglas en Inglés) relacional de objetos de código abierto. Sus orígenes remontan en 1986 por parte de la Universidad de California en Berkeley por el proyecto Postgres. Tiene una ampliación del lenguaje SQL⁷ puesto que se combinó con diversas funciones que almacenan y escalan información.

Se ejecuta en cualquier sistema operativo (Windows, Linux, OS, etc) cumpliendo con el ACID⁸ desde el 2001. Contiene un complemento potente para el manejo de base de datos espaciales. POSTGIS⁹.

2.2.2 Servicios WEB Geoespaciales

Al inicio de la era del SIG en internet, sus tecnologías fueron desarrolladas de una manera individual, esto repercutió en el manejo de información (principalmente al compartirla) y las funciones que estas desarrollaban. Ante esta situación el *Open Geospatial Consortium* (OGC)¹⁰, presentó estándares de Servicios Web Espaciales.

Para (Li et al., 2011), la organización de los servicios web es un desafío que incluye la selección de dichos servicios, la manera en cómo se relacionan e interactúan y el cómo se colocan en flujos de trabajo que brindan soluciones para usuarios, organizaciones y aplicaciones.

7 SQL (*Structured Query Language*). Lenguaje de programación diseñado para administrar y recuperar información de sistemas de gestión de bases de datos racionales.

8 ACID (*Atomicity, Consistency, Isolation, Durability*). Es un conjunto de propiedades de las transacciones de base de datos destinadas a garantizar la validez de los datos.

9 POSTGIS. Extensión de base de datos espacial para la base de datos relacional de objetos PostgreSQL.

10 OGC (*Open Geospatial Consortium*). Organización fundada en 1994, con la finalidad de llevar la información geográfica a un estándar mundial. Realizan estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica. Realiza acuerdos con las diversas empresas del sector que fomenten el intercambio de información geográfica en beneficio del usuario.

<https://live.osgeo.org/archive/10.5/es/standards/standards.html>

La OGC contiene el estándar OGC *Web Services Context Document* (OWS Context) que permite que un conjunto de recursos de información configurados de cierta forma, se intercambien entre aplicaciones principalmente como una colección de servicios, (OGC, s.f.). Tomando en cuenta su desarrollo en línea respalda casos de uso como la distribución de resultados de búsqueda, el intercambio de recursos como lo son:

- **Web Feature Service (WFS):** Servicio de publicación de objetos es un estándar que describe los datos georreferenciados de carácter vectorial. Permite actualizar, consultar, insertar y eliminar objetos geográficos. Define tres operaciones: *GetCapabilities*, devuelve los metadatos del nivel de servicio; *DescribeFeatureType*, describe la estructura del tipo de feature; *GetFeature*, devuelve el feature en formato GML (*Geography Markup Language*).
- **Web Map Service (WMS):** Servicio de publicación de mapas que contiene una interfaz sencilla HTTP. Se genera a partir de datos georreferenciados procesandolos en imágenes con formatos PNG, GIF o JPEG. Este servicio se establece por tres operaciones: *GetCapabilities*, retorna los metadatos del nivel de servicio; *GetMap*, devuelve un mapa con parámetros geográficos establecidos; *GetFeatureInfo*, muestra la información de los datos mostrados en el mapa. La norma que lo avala es ISO 19128.
- **Web Map Tile Service (WMTS):** Servicio de mapas en web por teselas, nos permite almacenar los datos recientes leídos en su visualización, formando una consulta de información por medio del caché. Es así como se agiliza su despliegue en una solicitud futura. El servicio funciona con un modelo de teselas (*Tiling Model*) parametrizando el contenido y poder mostrar las peticiones del usuario en un conjunto discreto de valores con imágenes prerrenderizadas (*Tiles*).
- **Web Covarage Service (WCS):** Servicio de coberturas en web, ofrece acceso a coberturas espaciales(formato tipo celda – ráster). Dichos formatos contienen valores para determinar las propiedades de localización, pueden contener imágenes de satélite, fotografías aéreas, modelos digitales de elevación entre otros. Establece tres operaciones: *GetCapabilities*, devuelve los metadatos del nivel del servicio; *DescribeCoverage*, descripción detallada de las coberturas; *GetCoverage*, obtiene una cobertura.

Tabla 2.2.2

Características de Estándares de Servicios Web Espaciales de la OGC.

Estándar	Entradas	Salidas	Permisos al Cliente	Operaciones
<i>Web Map Service (WMS)</i>	Mapas creados por un dataset de carácter ráster o vectorial	Imagen con extensión: - JPEG - GIF - PNG - Entre otros	-Solicitud	- Retorna los metadatos de nivel de servicio. - Devuelve un mapa con parámetros geográficos ya establecidos. - Muestra información de los datos.
<i>Web Feature Service (WFS)</i>	Dataset de carácter vectorial (puntos, líneas, polígonos)	Devuelve el resultado en formato: <i>Geography Markup Language (GML)</i>	- Solicitud - Consulta - Manipulación de información (Insertar, actualizar, eliminar)	- Retorna los metadatos de nivel de servicio. - Describe la estructura del tipo de feature - Devuelve el feature en formato GML
<i>Web Coverage Service (WCS)</i>	Dataset de carácter ráster.	Imágenes en código binario: - GeoTIFF ¹¹ - NetCDF ¹² - Entre otros.	- Solicitud - Consulta	- Devuelve los metadatos de nivel de servicio. - Descripción detallada de la cobertura. - Muestra la cobertura

Nota: Tomada de (Yanza, 2013)

11 GeoTIFF. Es un formato de intercambio de imágenes ráster georreferenciadas. Se basa en el formato TIFF y pertenece como estándar para la OGC. <https://earthdata.nasa.gov/esdis/eso/standards-and-references/geotiff>

12 NetCDF (*Network Common Data Form*). Se refiere a un formato de archivo destinado al almacén de datos científicos multidimensionales.

<https://desktop.arcgis.com/es/arcmap/10.3/manage-data/netcdf/what-is-netcdf-data.htm>

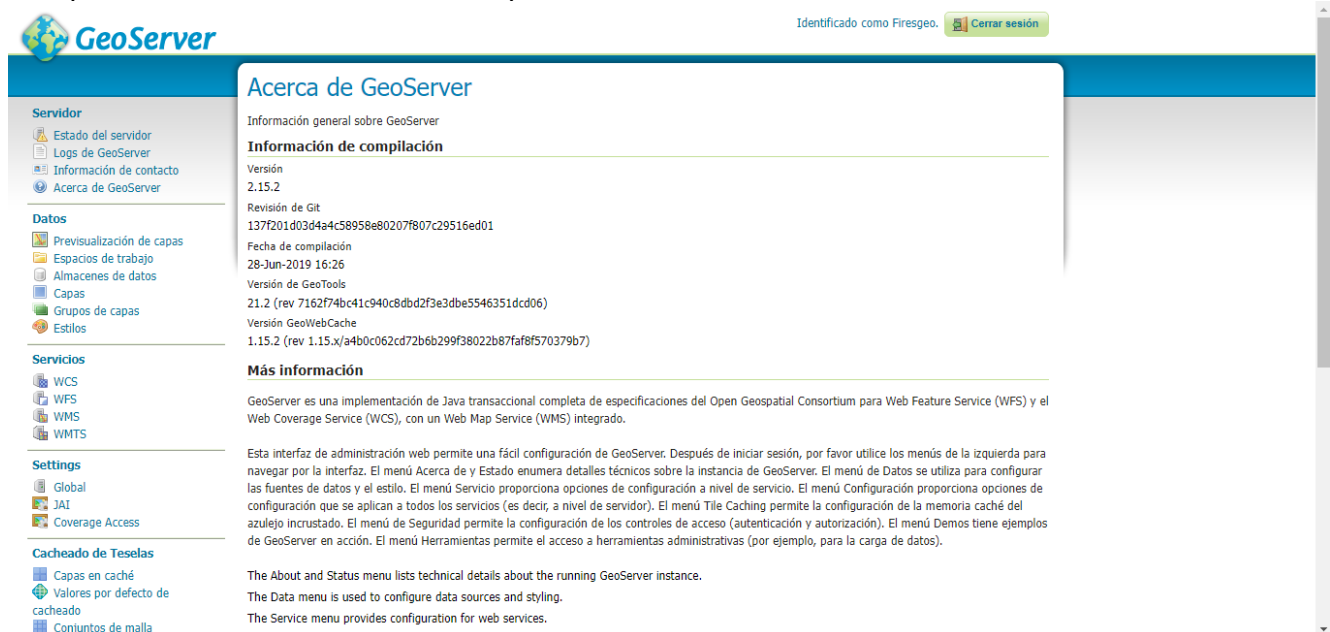
2.2.3 Geoserver

La importancia de un Servidor Web Map o en inglés *Internet Map Server* (IMS) radica en el interés de proveer cartografía en la red. Geoserver es un servidor web de código abierto que permite la publicación de datos espaciales en internet (Mappinggis, s.f.). Implementa protocolos establecidos por la OGC como lo son *Web Feature Service* (WFS), *Wep Map Service* (WMS) y *Web Coverage Service* (WCS). De igual forma cuenta con formatos y opciones de publicación adicionales como lo son los servicios de procesamiento web (WPS) y el Servicio de Mosaicos de Mapas Web (WMTS).

Desarrollado en el lenguaje de JAVA, cuenta con una interfaz muy amigable para su configuración y manejo (Figura 2.2.1). En ella se puede observar un menú con diversos apartados (Servidor, Datos, Servicios, *Settings*, Cacheado de Teselas, Seguridad, Demos y Herramientas) los cuales nos permitirán realizar las tareas que ofrece el servidor.

Figura 2.2.1

Interfaz de Geoserver versión 2.15.2, apartado “Acerca de Geoserver”.



2.2.4 HTML5, CSS3 y JavaScript

Es necesario señalar que para el desarrollo y diseño de una página o aplicación web, no se cuenta con un lenguaje de programación único. Es decir, es necesario un conjunto de ellos. En esta sección se desglosarán en este apartado los 3 lenguajes fundamentales para la creación de la aplicación SIG WEB desarrollada en esta tesis. Se considera desde la creación de la estructura de la página por medio del lenguaje HTML5 y su asignación de estilos con CSS3 hasta los procesos de interacción de acción y respuesta con el usuario por parte del lenguaje de programación JavaScript.

HTML y HTML5

El lenguaje marcado de hipertexto (HTML, por sus siglas en inglés) es el esqueleto de una página web. Su uso es por medio de etiquetas, organizando así los diferentes tipos de contenido y los fines de cada uno, por lo que no es un lenguaje de programación lógico.

Su misión es ser interpretado por un navegador y mostrar la página que solicita el servidor, si este contiene un error, se visualizará lo que el explorador de internet haya entendido. Su estructura es por medio de etiquetas, las cuales conforman un estándar para cada objeto de la página web; como el cuerpo del documento.

Fue en el *Conseil Européen pour la Recherche Nucléaire* (CERN, por sus siglas en francés) en Ginebra, Suiza año 1991, donde Tim Berners-Lee creó el HTML, con la finalidad de permitir acceso a la información entre investigadores en diversas partes del mundo, de una manera muy peculiar; el poder crear referencias cruzadas entre un artículo y otro. Es decir, el tejer una red de información con los documentos. (Tabarés, 2012)

HTML mostró una limitación en la forma de representar la información de un documento y la manera de cómo organizarlo en sus etiquetas, por lo que diversas compañías invirtieron en el desarrollo de

nuevos lenguajes y programas para ampliar el catálogo de características a las redes. Java¹³ y Flash¹⁴ fueron adoptados como los *plugins*¹⁵ (complementos) que cambiarían la internet, generando una lucha silenciosa entre ellos . A pesar de su gran impacto inicial de Java y Flash, fue JavaScript quien permitió a los desarrolladores innovar en sus páginas web. Dicho esto, fue como comenzó la era del nuevo mundo en la red conocido como “Web 2.0”.

HTML5 fue el sucesor de HTML, mostrando tres características esenciales: su estructura, estilo y funcionalidad. Dicho de otra manera, HTML mostraría un cambio al mundo del internet por su compactación en visualización web. Es por ello que muchos autores y desarrolladores establecen que HTML es una combinación de HTML, CSS y Javascript.

“HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red” Gauchat (2012).

La estructura HTML está dividida en dos secciones principales (Figura 2.2.2):

- **Cabeza (<HEAD>).** En ella se encuentra la información, como el título de la página (<TITLE>), palabras clave (utilizadas por el motor de búsqueda), información general acerca del documento, vinculación de archivos externos en con estilos y códigos de Javascript, imágenes , etc. Estos atributos a excepción del título y algunos íconos o imágenes, son invisibles para el usuario.
- **Cuerpo (<BODY>).** Abarca todo lo relacionado con el contenido del documento. Es la parte principal de la organización de este, y representa su parte visible.

13 Java: Lenguaje de programación orientado a objetos y plataforma informática comercializada desde 1995.

14 Flash: Tecnología utilizada para crear animaciones gráficas vectoriales independientes del navegador.

15 *Plugins*: Aplicación que añade funcionalidad adicional a un programa, aplicación o página web.

Figura 2.2.2

Esquema de un documento HTML

```

1 <html>
2   <head>
3     <title> Título de la página </title>
4   </head>
5   <body>
6     |   |   Contenido de la página
7   </body>
8 </html>
9

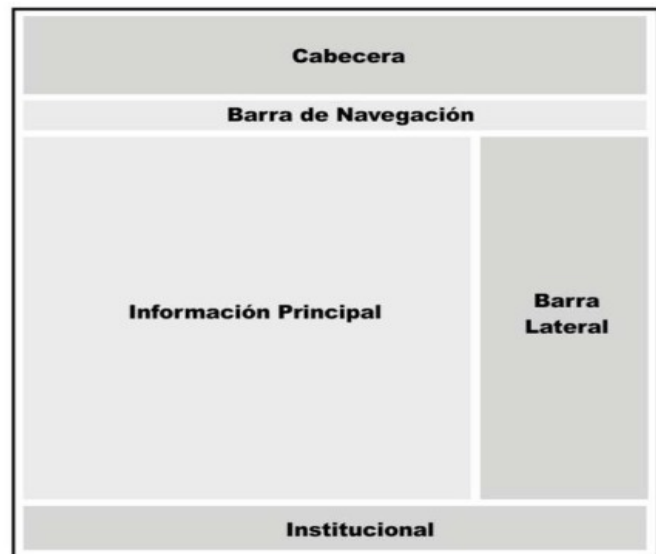
```

La organización de una página web en HTML (en el *body*) varía dependiendo el diseño propuesto por su creador, así como con las necesidades del usuario que interactuará con ella. A pesar de ello, se ha constituido plantillas en las cuales se consideran algunas partes esenciales en la visualización de una página, las cuales son:

- Cabecera (*header*)
- Barra de Navegación (*nav*)
- Sección de Información (*section*)
- Barra Lateral
- Barra institucional (*footer*)

Figura 2.2.3

Estructura esencial de una página web.



Fuente: Gouchat, 2012.

CSS y CSS3

Las hojas de estilo en cascada o *Cascading Style Sheets* (CSS), representa al lenguaje de estilos utilizado para el formato y apariencia de un documento de marcaje (HTML, XML, etc). Nace de la necesidad de manejar en estancias diferentes la estructura y contenido de una página web con su presentación. Dicho de otra manera, CSS se encarga de la parte visual y estilos de lo impreso en pantalla.

La primera especificación publicada por el *World Wide Web Consortium* (W3C) data en diciembre de 1996 y su objetivo fue unificar la sintaxis y el modo de definir una hoja de estilos. (Puig, 2013) menciona las principales características mencionadas en esta especificación:

- Prioridad en el tipo de letra y su estilo
- Colores de los textos y de los fondos
- Atributos del texto (espacio entre caracteres, palabras y líneas)
- Alineación de tablas, bloques de texto, imágenes, párrafos.
- Márgenes, filetes y posición de los elementos.
- Definición de los elementos mediante identificadores (**id**) y clases (**class**)

Posteriormente se agregaron diversas funciones a las hojas de estilo tanto en su segunda revisión como en tercera. Vale la pena decir que estas funciones eliminaron las limitaciones que se tenían en la primera revisión, como lo fueron las superposiciones y establecimientos de espacios en el documento por parte de la segunda revisión (CSS2) y los selectores, unidades de medida ,etc. Establecidos en la tercera revisión (CSS3). En la Tabla (2.2.3) se muestran las características principales entre revisiones CSS.

Tabla 2.2.3

Características principales de CSS, CSS2 y CSS3

CSS (1996)	CSS2 (1998)	CSS3 (2005)
<ul style="list-style-type: none"> • Propiedades y estilo de letra • Colores en texto y fondos • Atributos del texto • Alineación de tablas, bloques de texto, imágenes , párrafos. • Márgenes externos e internos, filetes y posición de los elementos. • Definición de los elementos con identificadores y clases. 	<ul style="list-style-type: none"> • Definición de posiciones de forma absoluta, relativa o fija. • Profundidad de elementos con (z -index) • Soportes para formatos de voz y textos bidireccionales. 	<ul style="list-style-type: none"> • Uso de módulos para partir el contenido y realizar una carga más rápida de la página web • Nuevos selectores • 16 nuevas pseudo-clases • Nuevos estilos y propiedades en modelos de caja y línea • Nuevos estilos y propiedades de imágenes y de imágenes de fondo. • Definición de número y ancho de columnas en el contenido. • Módulos de texto, color, fuentes, etc. • MediaQueries

Nota: Tomada de (Puig, 2013).

JAVASCRIPT

Es un lenguaje de programación creado por Brendan Eich en los años noventa. Tiene como función crear el dinamismo en las páginas web; dicho de otra manera, incorporar efectos , animaciones y funciones que permitan una mayor interacción con el usuario. Javascript es un lenguaje interpretado, por lo cual no se necesita compilar el programa para poder ejecutarlo, es por ello que cualquier navegador puede mostrar el resultado de un script de dicho lenguaje (Eguíluz, 2009).

Eich tras plantear solución a las páginas complejas y red de velocidad limitada, incorporó su lenguaje “LiveScript” al navegador *Netscape Navigator 2.0* desarrollado por la empresa *Netscape Corporation* en 1995. Sin embargo, *Netscape* con el tiempo firmó una alianza con *Sun Microsystems* para el desarrollo del lenguaje mencionado. Para ello justo antes del lanzamiento de *Netscape* el nombre de

LiveScript cambio al de JavaScript, por fines de *marketing* dado que Java tenía un poder comercial en ese tiempo. Por lo dicho anteriormente, JavaScript no pertenece en lo absoluto a Java.

En 1997 se mandó la especificación JavaScript 1.1 a la *European Computer Manufacturers Association* (ECMA¹⁶, por sus siglas en inglés) con el fin de estandarizar el lenguaje. Por lo que se creó el primer estándar **ECMA – 262**, en el que se definió el lenguaje **ECMAScript**. “De hecho, JavaScript no es más que la implementación que realizó la empresa *Netscape* del estándar ECMAScript.”, Eguíluz (2009).

JavaScript revolucionó la internet y su interacción usuario-servidor, para lo que se crearon diversas bibliotecas cuyo objetivo es obtener una API (*Application Programming Interface*). Para (Puig, 2013) estas bibliotecas pueden ofrecernos solución a los dos problemas principales en el desarrollo web:

1. **Interacción con el DOM**¹⁷: Seleccionar, añadir, modificar y borrar nodos.
2. **Interacción con el usuario**: Acciones de ratón , teclado, etc.

En el desarrollo de la aplicación SIG Web, de esta tesis, se ocuparon dos bibliotecas:

1. **JQuery**: para el entorno dinámico de la página, principalmente en el aspecto de animación y despliegue del menú.
2. **Openlayers3**: Para la creación y despliegue del mapa y datos geográficos.

16 ECMA (*European Computer Manufacturers Association*). Asociación industrial fundada en 1961, dedicada a la estandarización de sistemas de información y comunicación. <https://www.ecma-international.org/>

17 DOM. (*Document Object Model*). API para documentos HTML y XML. Proporciona una representación estructural del documento, permitiendo la modificación de su contenido o su presentación visual. <https://developer.mozilla.org/es/docs/DOM>

2.2.5 JQuery

Biblioteca de Javascript gratuita, pequeña, rápida y funcional (JQuery, s.f.). Facilita la selección de elementos HTML, su creación de animaciones y efectos, control de eventos y la manipulación con Ajax. Destaca por su funcionamiento en múltiples navegadores en particular la característica de poder reemplazar funciones HTML5 en exploradores que no estén preparador para soportarlas.

Existen dos maneras para poder ocupar la biblioteca de JQuery, la primera es por medio de la descarga del archivo de Javascript por medio de su página oficial y la segunda por medio del CDN¹⁸ (*Content Delivery Network*) de Google.

1. Enlace de la biblioteca JQuery por medio del archivo Javascript

```
<script type="text/javascript" src="jquery.js"></script>
```

2. Enlace de la biblioteca por medio de CDN

```
<script type="text/javascript"
src = "http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js"></script>
```

18 CDN (*Content Delivery Network*) . Red de distribución de contenidos de nivel global, con servidores geolocalizados e manera óptima para mejorar los tiempos de descarga.

2.2.6 Openlayers3

Biblioteca de Javascript de código abierto, utilizada en la publicación de mapas en la web. Puede mostrar diversas herramientas como la impresión de mosaicos, datos vectoriales, marcadores, etc. Según la OSGEO¹⁹ en su página oficial menciona que OpenLayers cuenta con las siguientes características principales (Tabla 2.2.4):

Tabla 2.2.4

Características principales de Openlayers según OSGEO

Capas	Controles e Interacciones	Personalización y Estilo	Superposiciones	Eventos	Más
- Manejo de imágenes ráster (WMS/WMTS, fuentes XYZ, etc) - Manejo de capas vectoriales (WFS, KML, GeoJSON, GML, etc) - Mapas de calor - Renderizados de capas vectoriales a ráster	- Overview map - Zoom slider - Scale - Desplazamiento de mapa - Rotación - Selección de funciones -Etc	- Diseño y estilo de puntos, líneas, polígonos e íconos - Personalización de botones con CSS.	- Renderizado de elementos DOM en cualquier lugar del mapa - Mezcla de capacidades HTML5 con el mapeo	- Creación de controles o interacciones personalizadas - Adjuntar funciones <i>listener</i> para asignar eventos.	- Soporte de navegador móvil - Reproyección ráster - Análisis de ráster -Renderizado de mapas con Canvas y WebGL.

Nota: Tomada de (OSGeoLive, s.f.)

19 OSGEO (*The Open Source Geospatial Foundation*) . Organización sin ánimo de lucro cuya misión es la promoción del desarrollo colaborativo de tecnologías geoespaciales abiertas siendo una organización inclusiva y dedicada a una filosofía abierta y participativa y al desarrollo dirigido mediante una comunidad participativa . <https://www.osgeo.org/>

2.2.7 Django

Django es un *framework* web desarrollado en Python²⁰ por Django Software Foundation (DSF)²¹. Tiene como finalidad la creación de sitios/aplicaciones web seguros y de fácil mantenimiento. Es gratuito y de código abierto, cuenta con documentación en línea (<https://docs.djangoproject.com/en/4.0/>) que permite al desarrollador consultarla para un mejor manejo.

Para (Mozilla Foundation, s.f.), Django facilita la creación de una aplicación web con las siguientes características:

- **Versátil:** Se puede construir casi cualquier sitio web (contenidos, redes sociales, sitios de noticias, etc). Devuelve contenido en variedad de contenidos incluyendo los formatos RSS feeds, JSON, XML, HTML, etc.
- **Seguro:** Proporciona una manera confiable de la administración de usuarios y contraseñas y evitar información de sesión en *cookies*. Las contraseñas se pueden enviar en un *hash* que permite encriptarlas disminuyendo la posibilidad de un ataque al a cuenta.
- **Escalable:** Basado en una arquitectura “*share-nothing*”²², puede escalar para aumentar el tráfico al agregar *hardware* en cualquier nivel.
- **Mantenible:** Cuenta con principios de mantener un código reutilizable y ligero. Promueve la agrupación de módulos por medio de aplicaciones.
- **Portable:** Al estar escrito en Python, permite una gran versatilidad con respecto a plataformas que lo pueda ejecutar, así como diversos sistemas operativos.

20 Python. Lenguaje de programación de alto nivel con enfoque multiparadigma, admitiendo construcciones de programación procedimentales, orientadas a objetos y algunas funcionales.

21 DFS (Django Software Foundation). Organización de código abierto sin fines de lucro, cuyo objetivo es apoyar, promover y avanzar en el marco web de Django.

22 *Share-nothing*. Tipo de arquitectura en la que cada nodo es independiente y autosuficiente.

2.2.8 Apache HTTP Server

Software de servidor web destinado a crear una implementación de código fuente robusta, de grado comercial pero gratuita de un servidor HTTP (Apache Software Foundation, s.f). Fue desarrollado por un conjunto de usuarios voluntarios y forma parte del proyecto ASF ²³(*Apache Software Foundation*, por sus siglas en inglés).

Entre sus ventajas se puede encontrar, los parches de seguridad que son actualizados frecuentemente brindando mayor confianza al desarrollador, a demás de una estructura en módulos que permite activarlos o desactivarlos a conveniencia. Apache HTTP *Server* es multiplataforma por lo que se puede correr en servidores con sistemas operativos Windows o Linux, por otro lado una de sus desventajas son los problemas de estabilidad con un aforo de más de 10 000 visitantes (Dinahosting, s.f.).

²³ ASF (*Apache Software Foundation*). Fundación estadounidense establecida en 1999, financiada por donaciones individuales. Supervisa más de 350 proyectos de código abierto incluyendo Apache HTTP *Server*. <https://www.apache.org/foundation/>

2.3 Aplicaciones SIG WEB como alerta de incendios

Los sistemas de información geográfica en la red han sido de gran utilidad en nuestros días, principalmente para la visualización y manejo de información de fenómenos naturales. Los incendios, principalmente los forestales, han provocado grandes desastres en los últimos años propiciando cambios radicales en los ecosistemas. Diversas empresas tanto privadas como gubernamentales han desarrollado varios programas para su prevención, localización y reparación de los daños causados.

A nivel mundial, se cuenta con varias aplicaciones web donde se visualizan los puntos de calor obtenidos por imágenes satelitales, uno de ellos es el desarrollado por la NASA²⁴, *Fire Information for Resource Management System* (FIRMS). En México, la CONABIO²⁵ creó un sistema de alerta temprana en el cual se pueden visualizar focos de calor y CONAFOR mediante el proyecto CONAFOR-CONACYT 2014-2-252620, implementó el Sistema de Predicción de Peligro de Incendios Forestales de México. (Vega Nieva et al, 2016)

24 NASA (*National Aeronautics and Space Administration*). Agencia del gobierno de estadounidense encargado del programa espacial civil, así como la investigación aeronáutica y aeroespacial.

<https://www.nasa.gov/>

25 CONABIO (Comisión Nacional para el Conocimiento y Uso de la Biodiversidad).

2.3.1 Alerta temprana de incendios de la CONABIO

En el año de 1999 la CONABIO implementó el programa “*Detección de puntos de calor mediante técnicas de percepción remota*”, en respuesta a los incendios forestales catastróficos registrados en la temporada de 1998, donde se registraron 14,445 incendios afectando 849,632 hectáreas.

Las imágenes utilizadas para el estudio serían las del satélite DMSP-OLS (*Defense Meteorological Satellites Program – Operational Linescan System*), proporcionadas por NGDC-NOAA (*National Geophysical Data Center – National Oceanic and Atmospheric Administration*). Sin embargo, se tenía acceso a las imágenes tres días después de su registro por lo que el sistema mudó a la utilización de rásters proveniente de AVHRR (*Advance Very High Resolution Radiometer*) de la serie de satélites de la NOAA, facilitadas por el Instituto de Geografía (IG) de la UNAM.

Para el año del 2000 la CONABIO adquirió una estación de recepción de imágenes AVHRR, para reducir el tiempo de respuesta. A la par en portal (versión 2000) incrementaba sus funciones desplegando información en forma de calendario. Fue la primera vez que el portal abría acceso a todo el público.

Para su tercera versión (v. 2009) incrementó la organización, presentación y acceso a la información. En su cuarta y más actual versión (v. 2014) el portal migró a ser un sistema de alerta temprana el cual tiene como objetivo:

“Contar con un sistema de alerta temprana, monitoreo e impacto de los incendios forestales, mediante información satelital e información geográfica, que contribuya en la prevención y combate de incendios forestales para la conservación de la biodiversidad y la salvaguarda de la vida humana.” (CONABIO, 2022)

Este sistema contiene como primer cambio la visualización de la información en forma espacial y no tabular, además de agregar los puntos de calor detectados mediante las imágenes VIIRS. Los puntos de calor que despliega pertenecen a tres tipos sensores AVHRR, MODIS y VIIRS mismos que se distribuyen del 2000 a la fecha.

Tabla 2.3.1

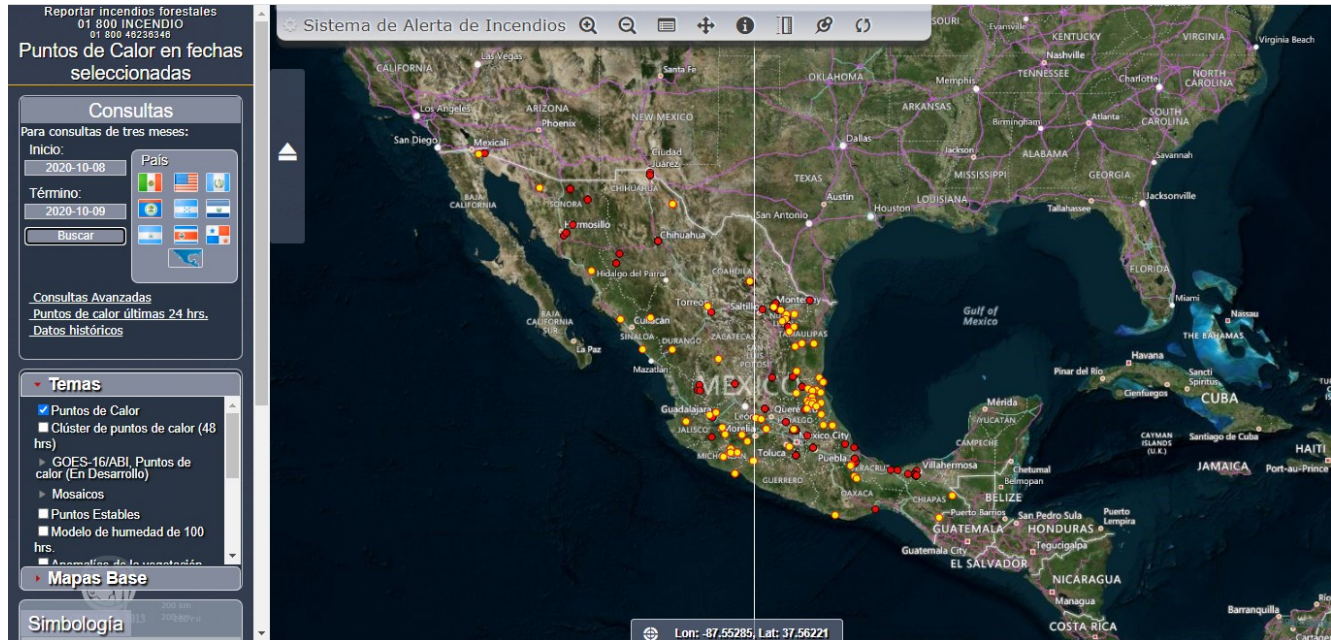
Información disponible en base de datos por el Sistema de Alerta Temprana CONABIO para territorio mexicano.

Tipo de Imagen	Año	Periodo
AVHRR	2000	Enero - Agosto
AVHRR	2001	Enero - Julio (diurnas) Enero - Agosto (nocturnas)
AVHRR	2002	Enero - Octubre
AVHRR	2003	Enero – Agosto (diurnas) Enero – Agosto (nocturnas)
MODIS	2004 - 2013	Enero - Diciembre
MODIS	2014	Enero
VIIRS	2014 - fecha	Enero - Diciembre

Nota: Tomada de (CONABIO, 2022).

Figura 2.3.1

Consulta de puntos de calor en portal de Alerta Temprana de CONABIO comprendida entre las fechas del 08-10-20 a 09-10-20.



Nota: Se consideran puntos diurnos los comprendidos en el intervalo de 6:00 – 17:59 y puntos nocturnos entre las 18:00 (día anterior) a las 5:59 hora local CDMX.

2.3.2 Sistema de Predicción de Peligro de incendios Forestales de México (CONAFOR-CONACYT 2014-2-252620)

Desarrollado en el proyecto CONAFOR-CONACYT 2014-2-252620, el Sistema de Predicción de Peligro de incendios Forestales es una herramienta para la toma de decisiones en el trato de prevención y combate de incendios en la república mexicana. (*Sistema de Predicción de Peligro de Incendios Forestales de México. Acerca del Sistema*, s. f.) creado por la Universidad de Juárez en Durango en colaboración de diversas instituciones nacionales e internacionales (Vega Nieva et al., 2016), el sistema evalúa diversas características y condiciones del ecosistema para determinar la probabilidad que se presente un incendio. Entre los factores que este considera se encuentran: las condiciones de sequedad del combustible, peligro de incendio asociado, variables meteorológicas en casi tiempo real,

mapas de vegetación y factores humanos (poblados y carreteras). Su finalidad consiste en apoyar el proceso de toma de decisiones en la prevención y lucha contra incendios que realiza la CONAFOR.

La información ofrecida por el sistema se representa en niveles de índices de peligro de incendio, proporcionando:

- Evolución de la sequedad del combustible.
- Número de incendios esperados.
- Localización aproximada de los incendios.
- Comportamiento esperado del incendio.

Información de los puntos activos (CONABIO), información meteorológica (CONABIO y SMN) e incendios activos (CONAFOR) son los insumos para el cálculo de índices de sequedad del combustible, peligro de incendio, peligro meteorológico y los incendios esperados por entidad federativa (estado).

- **Índice de Sequedad del Combustible (ISC):** Obtenida por mapas de temperatura, humedad y precipitación además de compuestos semanales de índice de vegetación de diferencia normalizada (NDVI, por sus siglas en inglés), arroja las condiciones de estrés hídrico diarias de los combustibles.
- **Índice de Peligro Meteorológico:** Combina el índice de sequedad del combustible (ISC) con el tipo de vegetación, ecorregión y la presencia de puntos de calor previos.
- **Peligro de incendio:** Muestra la probabilidad de ocurrencia de un combate de incendio, es decir de todas las áreas donde el peligro meteorológico sea alto y estén con cierta cercanía a las localidades y vías de acceso.
- **Número esperado de incendios por estado:** Muestra el pronóstico de los incendios esperados por estado.

Figura 2.3.2

Incendios esperados para el 09-10-20 por el Sistema de Predicción de Peligro de Incendios Forestales en México.



2.3.3 TerraMA²

TerraMA² es una plataforma computacional diseñada por *Instituto Nacional de Pesquisas Espaciais* (INPE) capaz de analizar, monitorear y alertar sobre riesgos ambientales. Integrada al campo del *Open Source*, está formada principalmente por la biblioteca geográfica TerraLib (*TERRAMA²*, s. f.). Compatible con los estándares marcados por la OGC, en su cuarta generación TerraMA² presenta avances en:

- Almacenamiento y acceso a datos geospaciales mediante estándares de servicios Web (WMS, WCS, WFS)
- Uso de bases de datos distribuidas para datos dinámicos y estáticos.
- Soporte para las diferentes arquitecturas en el almacenamiento de datos.
- Ejecución de servicios locales o remotos

- Administración de usuarios y control de proyectos mediante una interfaz WEB
- Visor Web actualizado
- Análisis de guiones con Python.

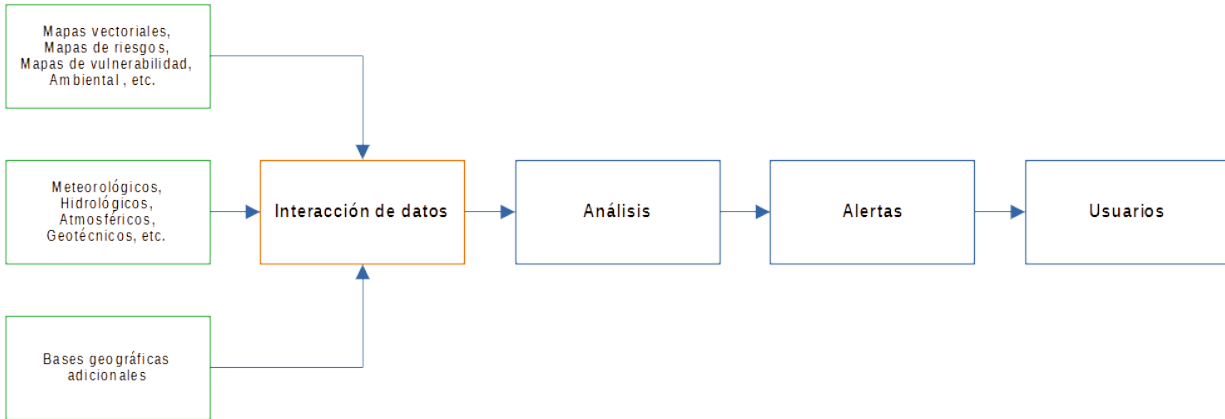
Necesita fundamentalmente de 3 insumos comprendidos por: datos dinámicos de entrada de carácter espacial relacionados principalmente con el contexto ambiental (meteorológicos, hidrológicos, atmosféricos, geotécnicos, etc), el objeto monitoreado (mapas vectoriales, mapa de riesgo, mapas de vulnerabilidad) y datos adicionales como sus bases cartográficas. Posteriormente se procesa los datos con un análisis computacional comprendido por un lenguaje de programación con la finalidad de obtener un sistema de alerta que estará a disposición de los múltiples usuarios beneficiados.

Esta plataforma cuenta con tres niveles de usuarios:

- **Administradores de la Plataforma:** Se refiere al personal que instala, administra y configura la plataforma. Son encargados de definir el espacio donde se crearán las bases de datos y rutas del TerraMA² en el ordenador. A demás de administrar a los usuarios.
- **Operadores de la Plataforma:** Son usuarios que administran la plataforma mediante la visualización web para fines prácticos de sus proyectos sin embargo no cuentan con privilegios de administrador.
- **Clientes de Alerta:** Personal que pueden realizar acciones preventivas para la pérdida de datos. Reciben las alertas en tiempo real y pueden monitorear los procesos y resultados. También recibe notificaciones por correo electrónico o mensajes.

Figura 2.3.3

Diagrama de los principales módulos del Sistema de Monitoreo, análisis y alerta (TerraMA2).



Nota: Adaptado de (INPE & DPI, 2018)

2.3.4 FIRMS FIRE MAP

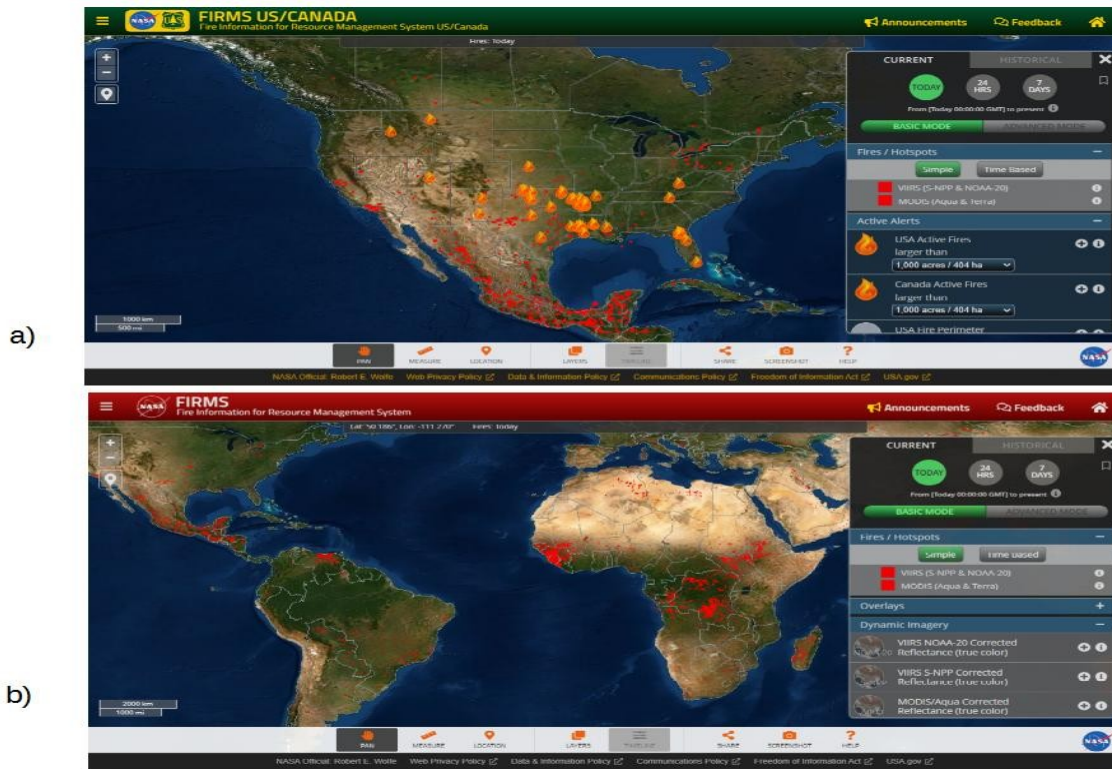
El sistema de información sobre incendios para gestión de recursos (FIRMS, por sus siglas en inglés), proporciona datos de incendios activos para aplicaciones y monitoreo casi reales. El sistema distribuye datos casi en tiempo real, dentro de las 3 horas posteriores a la observación satelital desde el espectroradiómetro de imágenes de resolución moderada (MODIS, por sus siglas en inglés) a bordo de los satélites Aqua y Terra, y el conjunto de radiómetros de imágenes infrarrojas visibles (VIIRS, por sus siglas en inglés) a bordo del S-NPP y NOAA 20 (formalmente conocido como JPSS-1), FIRMS, s.f.

Los datos de incendios activos, representados como puntos de calor (hotspots), se visualizan en la sección de FIRMS FIRE MAP (Figura 2.3.4) o en WORLDVIEW de la NASA. Dicha información se entrega como alertas por correo electrónico, también se puede descargar en sitio WEB en los siguientes formatos: SHP, KML, TXT, WMS.

FIRMS es parte de los servicios operacionales terrestre y atmosférica que realiza EOS (LANCE) de la NASA.

Figura 2.3.4

Plataforma SIG WEB de FIRE MAP.



Nota: a) FIRE MAP US/CANADA de FIRMS del 18/04/2022. b) FIRE MAP GLOBAL de FIRMS del 18/04/2022.

3 Algoritmo de detección de puntos de calor GOES-16/ABI y sus productos

Tras la oleada de incendios que se han suscitado los últimos años a nivel mundial, se han desarrollado de mano de la tecnología diversos métodos para su prevención, detección y combate. Tal es el caso del algoritmo creado por el LANOT perteneciente al Instituto de Geografía de la UNAM, que tiene como finalidad poder detectar incendios en tiempo casi real por medio de imágenes satelitales, provenientes del satélite de órbita geoestacionaria, GOES-16.

Para su desarrollo fue necesario contemplar diversos factores, como las manchas urbanas, puntos estables de calor (como refinerías, campos geotérmicos, industrias, etc), cuerpos de agua, cambios climáticos, etc. Dado que se manejan imágenes en una resolución temporal muy corta. se captarán los cambios de los factores mencionados. Por ello (Manzo Delgado, 2022), lo cataloga como un algoritmo de multiumbral dinámico.

Son tres fases distintivas en el proceso de obtención de un punto de calor por el algoritmo. El primero, tiene como propósito obtener el umbral máximo (óptimo) para cada horario correspondiente a una imagen GOES 16 ($T_{3,9}$ ²⁶ y ΔT ²⁷). El segundo, obtiene la matriz de pixeles con potencial a puntos de calor mediante la multiplicación de bandas reclasificadas por los umbrales encontrados y las máscaras, eliminado así la posibilidad de ser puntos estables de calor. Para terminar, la matriz de pixel potencial obtenida se somete a una prueba contextual que confirme que el valor de temperatura de brillo de cada pixel sea superior a la de sus colindantes, afirmando ser un punto de calor (Gómez Rubio, 2022).

3.1 Creación de Umbrales

Se obtienen los umbrales óptimos para cada fecha y horario en $T_{3,9}$ y ΔT , mediante la creación de eventos FIRMS. Estos, son tratados por medio de geoprocursos hasta la obtención de polígonos que muestren las zonas de posibles puntos de calor, mismas que cumplan, un área mayor de 1 000 metros cuadrados, cuenten con más de dos puntos de calor en su interior y que estos tengan más de dos horas

26 Banda 7 del sensor *Advanced Baseline Imager* (ABI) con longitud de onda central de 3.9 μm .

27 Diferencia de las bandas 7 y 14 (3.9 y 11.2 μm) del sensor *Advanced Baseline Imager* (ABI).

de actividad. Esto confirma que los puntos de calor contenidos en los polígonos sean los más extensos y representativos.

Una vez obtenidos los eventos FIRMS, se sobreponen en las imágenes para la obtención de sus estadísticas zonales guardando solamente los valores máximos registrados. Puesto que las imágenes GOES-16/ABI se obtienen cada 10 minutos, se generan al final eventos FIRMS de 144 horarios en un día para $T_{3,9}$ y 144 para ΔT . Este proceso se realiza una vez al día, para cada hora del día, para después realizar una interpolación *spline* para encontrar los valores de cada 10 minutos (Gómez Rubio, 2022). Cada evento contiene al menos dos valores de temperatura de brillo por lo que se toma el valor máximo por evento, quedando solamente un valor por cada horario (10 minutos).

3.2 Obtención de puntos de calor

Este proceso abarca la preparación de imágenes $T_{3,9}$ y ΔT y recortarlas a una escena que abarque al territorio mexicano, países colindantes centroamericanos y caribe. A demás de reclasificarlas con respecto a los umbrales obtenidos anteriormente. Debido a que obtener los umbrales de la imagen actual es imposible, se toma el umbral máximo de los 10 días anteriores del horario de prueba. En dicha reclasificación se asigna el valor de 1, al pixel candidato a ser punto de calor y con 0 al que no.

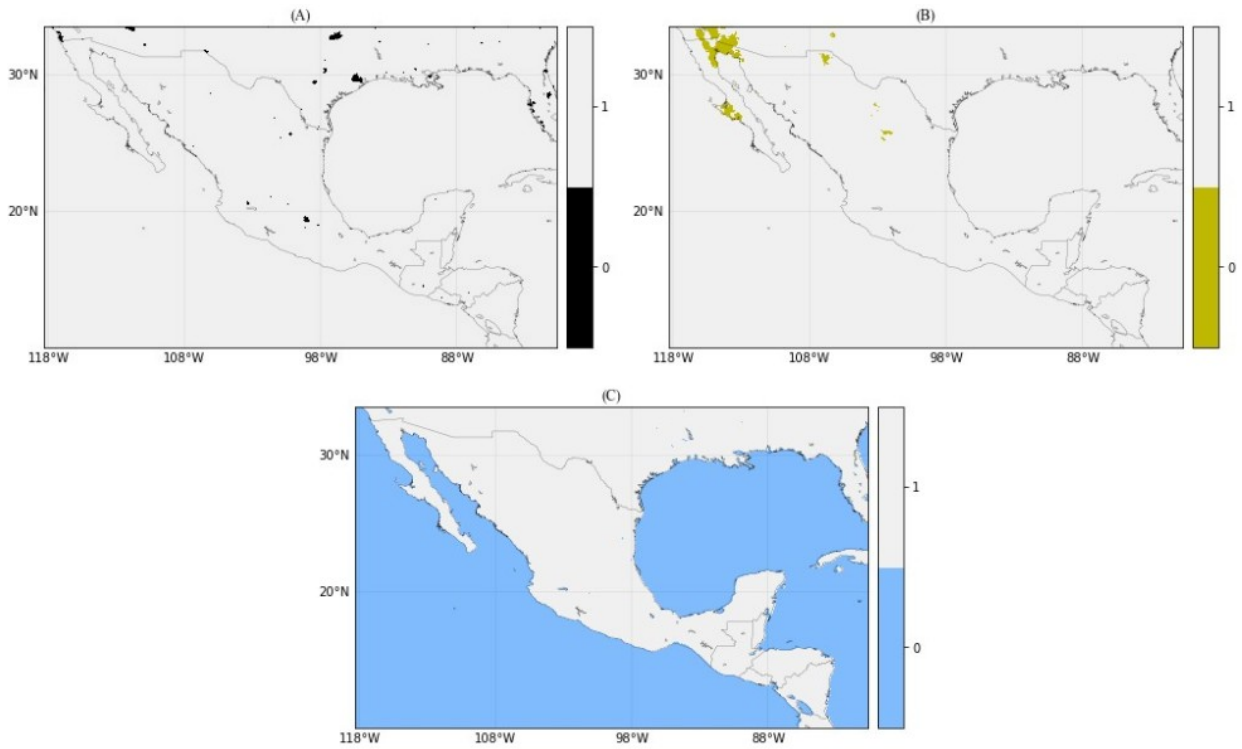
Posteriormente, se multiplica con máscaras asociadas a cuerpos de agua, suelo desnudo y manchas urbanas, para eliminar posibles errores asociados a ellas. Dichas máscaras son matrices booleanas, donde las represente el 0 y lo que no 1 (Gómez Rubio, 2022) (Figura 3.2.1). Estas fueron generadas del producto “*Collection 6 Modis Land Cover (MCD12Q1)*”²⁸ descargado del explorador de USGS²⁹: <https://earthexplorer.usgs.gov/>. Aquellos puntos que lleguen a contener el valor de 1 son los que se puede considerar como pixeles potenciales (Figura 3.2.2).

28 Producto obtenido de datos de reflectancia MODIS Aqua y Terra, que proporciona tipos de cobertura terrestre en intervalos anuales (2001 – 2019).

29 USGS (*United States Geological Survey*). Servicio Geológico de los Estados Unidos.

Figura 3.2.1

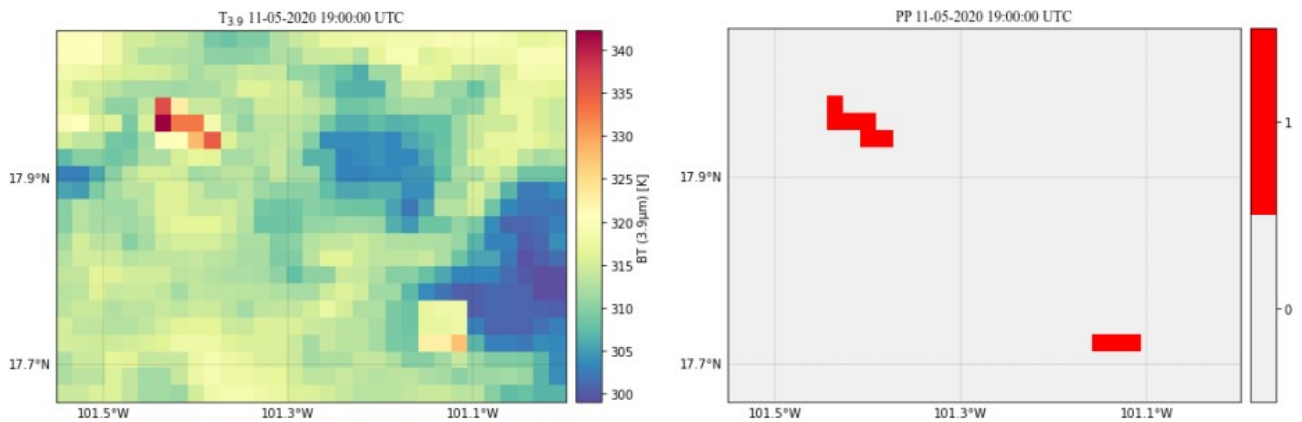
Máscaras de: A) Manchas Urbanas, B) Suelo desnudo y C) Cuerpos de agua. Generadas por el producto MCD12Q1, clasificación IGBP anual.



Nota: Tomada de (Gómez Rubio, 2022)

Figura 3.2.2

Ejemplo de Pixeles potenciales de T3.9 del 11 de mayo 2020 de las 19:00:00 UTC.



Nota: Tomada de (Gómez Rubio, 2022)

Por último la prueba de contexto, se basa en someter cada pixel potencial a una prueba condicional que radica principalmente en funciones estadísticas. Esta prueba, confirma que la temperatura de brillo del pixel sea mayor a la de sus vecinos. Es por ello, que se toma un *kernel* de 7x7 donde el pixel potencial es el núcleo de la matriz y tendrá que cumplir con la siguiente condición:

$$\underline{(PP3.9 > (mean(K3.9) + 2.5std(K3.9))) \& (PP3.9 - 11.2 > (mean(K3.9 - 11.2) + 2.5std(K3.9 - 11.2)))}$$

Dónde:

PP3.9 : *Temperatura de brillo del pixel potencial de la banda 7.*

mean(k3.9) : *Media aritmética de la temperatura de brillo del kernel 7x7 de la banda 7.*

std(k23.9) : *Desviación estándar de la temperatura de brillo del kernel 7x7 de la banda 7.*

PP3.9-11.2 : *Temperatura de brillo del pixel potencial de la diferencia de bandas 7-14.*

mean (k3.9-11.2) : *Media aritmética de la temperatura de brillo del kernel 7x7 de la diferencia de bandas 7-14.*

std(k3.9 – 11.2) : *Desviación estándar de la temperatura de brillo del kernel 7x7 de la diferencia de bandas 7-14.*

3.3 Generación de productos

Aquellos pixeles que pasen la prueba de contexto, se les denomina como puntos de calor. Mismos que se les asigna información relevante, tal es el caso como la temperatura de brillo en la B7 como en la diferencia de bandas 7-14, fecha y hora de detección del punto de calor, información espacial como su longitud y latitud, entre otras (Tabla 3.3.1). Los resultados se representan en formatos CSV y *Shapefile*.

Tabla 3.3.1

Atributos del producto de incendios GOES-16/ABI.

Nombre	Descripción	Rango de datos validos
Lon	Longitud del punto de calor	-119° , -80°
Lat	Latitud del punto de calor	10° , 34°
Satélite	Satélite proveedor de la imagen	-
BT_c07	Temperatura de brillo en banda 7	-
BT_c14	Temperatura de brillo en banda 14	-
Dif_c07-c14	Diferencia de temperatura de brillo banda7-14	-
Fecha	Fecha de detección del punto de calor	-
Hora	Hora de detección del punto de calor	-
Land_Cover	Cubierta terrestre correspondiente al punto de calor (datos de clasificación IGBP anual)	1 , 17
Estado	Estado donde se localiza el punto de calor	-
País	País correspondiente al punto de calor	-
ANP	Área Natural Protegida donde se localiza el punto de calor, en caso de estarlo (datos CONABIO)	-
Freq	Frecuencia en la que se detectó durante el día el punto de calor	1 , 144
Freq_norm	Frecuencia normalizada. Porcentaje de detecciones durante el día del mismo punto de calor	0 – 100%

Nota: Tomada de (Gómez Rubio, 2022).

4 Metodología

La metodología se organizó en una serie de pasos certeros y concisos . En su desarrollo se contó como insumo a los archivos CSV arrojados por **el algoritmo de detección de puntos de calor GOES-16/ABI**.

Se desarrolló una página previa al visualizador con la finalidad de introducir al usuario al proyecto de puntos de calor y así poder familiarizarse con el visualizador y sus funciones. Dentro de esta página se muestran gráficas relacionadas con las estadísticas obtenidas por los datos de los puntos de calor.

Los procesos realizados se dividieron en dos secciones: *Backend* y *Frontend*. La primera representa aquellos pasos relacionados con el desarrollo de la base de datos, manejo del servidor y creación de scripts para la aplicación en conjunto de los mismos, comúnmente se le conoce como la parte encargada de aquello que no se ve en una aplicación web. Por el contrario, el Frontend se refiere a la parte que el usuario visualiza en la aplicación, es otras palabras el aspecto visual del sitio web; menú, controles, información, mapa, etc.

Procesos en Backend:

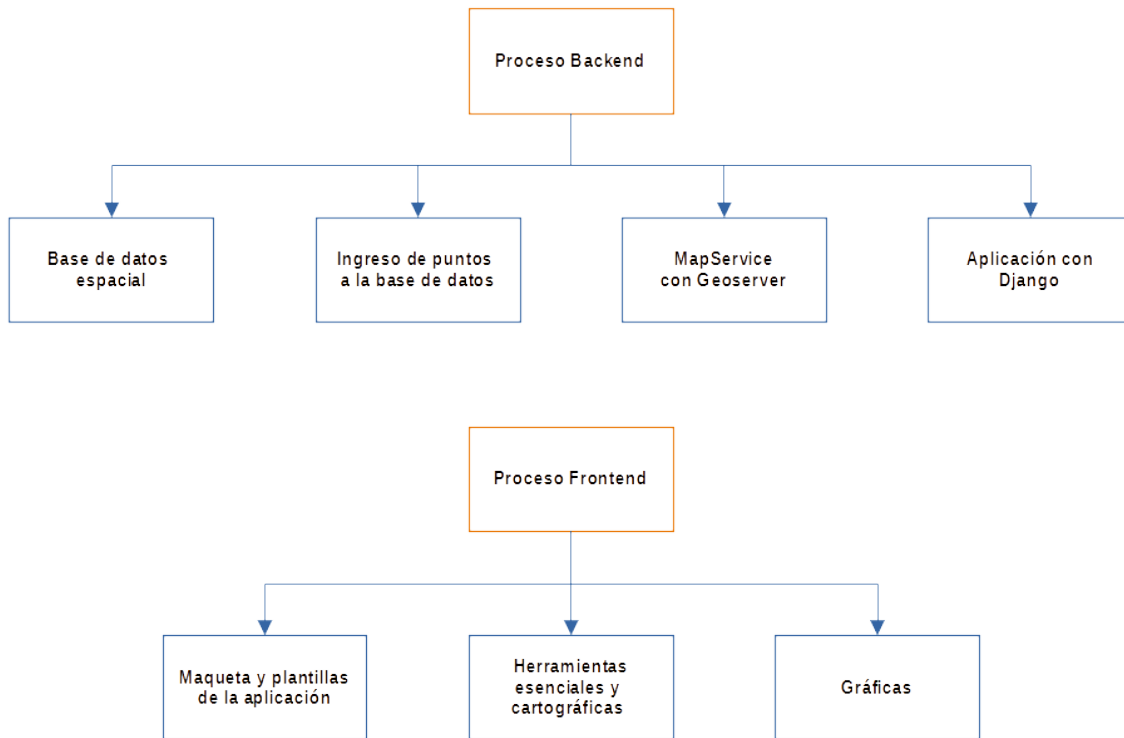
1. Creación de Base de Datos Espacial
2. Alimentación de la Base de Datos Espacial con los archivos CSV
3. Visualización de los datos en Geoserver y obtención de *MapService*
4. Creación de aplicación con Django

Procesos en Frontend:

5. Maquetación y creación de plantillas (principal y visualizador) por medio de HTML5 y CSS3.
6. Implementación herramientas cartográficas y esenciales del funcionamiento de la aplicación por medio de la biblioteca Openlayers3 y javascript.
7. Desarrollo de maquetas gráficas correspondientes a las estadísticas de los incendios en periodos de tiempo.

Figura 4.1

Metodología dividida en Backend y Frontend.



4.1 Base de Datos de puntos de calor GOES-16/ABI

El manejo de datos es muy importante para la representación de los puntos de calor, debido a que tener un orden y control sobre ellos eficiencia el manejo para su visualización, consulta y descarga. Para ello se creó una base de datos en el software PostgreSQL correspondiente a los puntos de calor GOES – 16 /ABI.

Cuenta con los campos contenidos en los archivos CSV arrojados por el algoritmo, agregándoles el campo de identificación (ID) siendo su clave primaria y el campo de geometría. El campo de identificación se actualiza cada que se cargue un nuevo punto a la base de datos, esto garantiza una actualización continua de los registros existentes. En otras palabras cada punto de calor cuenta con un

ID que lo identifica. El campo de geometría es indispensable para la visualización de los puntos, puesto que les da la forma geométrica a los datos (puntos, línea o polígono) así como su ubicación geográfica. Este último campo es el encargado de crear la base de datos de índole espacial.

En la figura 4.1.1 y 4.1.2 se puede observar la diferencia entre el archivo proveniente del algoritmo y el procesado por la base de datos espacial, en el cual se pueden observar los campos de identificación (id) y geometría (geom).

Figura 4.1.1

Parte del archivo CSV arrojado por el algoritmo correspondiente a 01/10/2020.

lon	lat	satelite	bt_c07	bt_c14	dif_c07c14	fecha	hora	land_cover	estado	pais	anp
-83.04954	33.220402	Goes-16	326.27	297.89	28.38	01/10/2020	17:40:00	Woody Savannas		Estados Unidos	
-83.03154	33.220402	Goes-16	326.27	297.89	28.38	01/10/2020	17:40:00	Deciduous Broadleaf Forests		Estados Unidos	
-83.01354	33.220402	Goes-16	333.09	298.13	34.96	01/10/2020	17:40:00	Mixed Forests		Estados Unidos	
-83.01354	33.220402	Goes-16	328.59	298.25	30.34	01/10/2020	18:00:00	Mixed Forests		Estados Unidos	

Figura 4.1.2

Parte del archivo CSV procesado por la Base de Datos Espacial.

id	lon	lat	satelite	bt_c07	bt_c14	dif_c07c14	fecha	hora	land_cover	estado	pais	anp	geom
421878	-83.04954	33.220402	Goes-16	326.27	297.89	28.38	01/10/2020	17:40:00	Woody Savannas		Estados Unidos		POINT (-83.049539807205 33.2204023911273)
421879	-83.03154	33.220402	Goes-16	326.27	297.89	28.38	01/10/2020	17:40:00	Deciduous Broadleaf Forests		Estados Unidos		POINT (-83.031539807205 33.2204023911273)
421880	-83.01354	33.220402	Goes-16	333.09	298.13	34.96	01/10/2020	17:40:00	Mixed Forests		Estados Unidos		POINT (-83.013539807205 33.2204023911273)
421895	-83.01354	33.220402	Goes-16	328.59	298.25	30.34	01/10/2020	18:00:00	Mixed Forests		Estados Unidos		POINT (-83.013539807205 33.2204023911273)

4.1.1 Creación de Base de Datos espacial y tablas con PostgreSQL

Para la creación de la base de datos se ocupó el servidor1 de LANOT con un sistema operativo GNU/LINUX, el cual tiene montado el software de PostgreSQL. Se creó un usuario el cual tiene los privilegios para crear y actualizar la base de datos. Posteriormente se creó nuestra base de datos.

El usuario se creó tecleando el comando:

CREATE USER fires_user WITH PASSWORD 'password';

La base de datos se nombro “fires_db”, en ella guardamos las tablas relacionadas con los puntos de calor, y se creó tecleando desde consola:

CREATE DATABASE fires_db WITH OWNER fires;

Para acceder a la base de datos creada, se tecléa la siguiente sentencia desde la consola de linux:

psql -U fires_user -d fires_db -p 5433 -h localhost;

Donde:

- **fires_user** : significa el nombre de usuario que contiene la base de datos
- **fires_db** es la base de datos creada
- **5433**: es el puerto en el que se encuentra vinculado PostgreSQL
- **localhost**: es el host o anfitrión. Común mente es el IP del servidor o si la conexión se realiza desde el mismo servidor se puede ocupar “localhost” como en este caso.

Una vez escrito el comando, escribiremos el *password* relacionado con el usuario. Dentro de la base de datos, creamos una extensión a PostGis mediante (OSGeoLive, s.f.):

CREATE EXTENSION postgis;

CREACIÓN DE TABLAS

El visualizador cuenta con dos capas relacionadas con los puntos de calor. La primera se refiere aquellos detectados las últimas 24 horas, mientras que la segunda, nos permite desplegar los puntos por medio de una consulta temporal, es decir, en intervalos de días. Por lo que fue necesario crear dos tablas en la base de datos, una por capa.

La primer tabla está relacionada con la capa de los puntos de calor de las últimas 24 horas y se actualiza cada 10 minutos (tiempo en que se obtiene un nuevo producto del algoritmo), contiene los campos mostrados en la Tabla 4.1.1 y se nombró **puntos_10m**.

Tabla 4.1.1

Campos y tipos de datos de la tabla puntos_10m.

Nombre del Campo	Tipo de dato*
id	serial
X	float
Y	float
lon	float
lat	float
satelite	varchar
bt_c07	float
bt_c14	float
dif_c07c14	float
fecha	date
hora	time
land_cover	varchar
estado	varchar
pais	varchar
anp	varchar
geom	geometry(POINT, 4326)

Descripción del tipo de dato:

- **serial**: Es el identificador en número único
- **float**: Representa el tipo de dato de números con punto decimal
- **varchar**: Cadena de caracteres sin espacios al final
- **date**: Tipo de dato en fechas con formato ‘AAAA-MM-DD’
- **time**: Tipo de dato en hora con formato ‘hh:mm:ss’
- **geometry(POINT, 4326)** : representa el tipo de dato ocupado por PostGis para la visualización del dato en forma de punto con proyección WGS84.

Para la realización de esta tabla se usó la siguiente sentencia:

```
CREATE TABLE puntos_actual( id serial not null, lon float8, lat float8,
    satelite varchar(7), bt_c07 float8, bt_c14 float8, dif_c07c14 float8,
    fecha date, hora time, land_cover varchar(40), estado varchar(35),
    pais varchar(25), anp varchar(20), geom geometry(POINT, 4326));
```

Donde:

- *not null*: Menciona que el número identificador no contendrá el cero, por lo que la cuenta comenzará con el número 1.
- *float8*: Valor doble flotante en PostgreSQL.
- *varchar(n)*: n significa el número de caracteres que tendrá la cadena.

Figura 4.1.3

Información de la tabla *puntos_10m* en PostgreSQL

Column	Type	Collation	Nullable	Default
<i>id</i>	integer		not null	nextval('puntos_10m_id_seq'::regclass)
<i>x</i>	double precision			
<i>y</i>	double precision			
<i>lon</i>	double precision			
<i>lat</i>	double precision			
<i>satelite</i>	character(7)			
<i>bt_c07</i>	double precision			
<i>bt_c14</i>	double precision			
<i>dif_c07c14</i>	double precision			
<i>fecha</i>	date			
<i>hora</i>	time without time zone			
<i>land_cover</i>	character varying(40)			
<i>estado</i>	character varying(35)			
<i>pais</i>	character varying(40)			
<i>anp</i>	character varying(20)			
<i>geom</i>	geometry(Point, 4326)			

La segunda tabla está vinculada al producto de puntos de calor que agrupados por día. Esto nos permite conocer la frecuencia en la que un punto se detectó en un mismo lugar en un día. Para ello, además de mantener la misma estructura de la tabla anterior se agregaron dos atributos más: *freq* y *freq_norm* (Tabla 4.1.2). Esta tabla contiene el nombre de *puntos_diacom*.

Tabla 4.1.2

Información de la tabla `puntos_10m` en PostgreSQL

Nombre del Campo	Tipo de dato*
id	serial
X	float
Y	float
lon	float
lat	float
satelite	varchar
bt_c07	float
bt_c14	float
dif_c07c14	float
fecha	date
hora	time
land_cover	varchar
estado	varchar
pais	varchar
anp	varchar
freq	int
freq_norm	float
geom	geometry(POINT, 4326)

Figura 4.1.4

Información de la tabla `puntos_diacom` en PostgreSQL.

```

Table "public.puntos_diacom"
 Column | Type          | Collation | Nullable | Default
-----|-----|-----|-----|-----
 id     | integer       |           | not null | nextval('puntos_dia_id_seq'::regclass)
 x      | double precision
 y      | double precision
 lon    | double precision
 lat    | double precision
 satelite | character varying(7)
 bt_c07 | double precision
 bt_c14 | double precision
 dif_c07c14 | double precision
 fecha  | date
 hora   | time without time zone
 land_cover | character varying(40)
 estado | character varying(35)
 pais   | character varying(40)
 anp    | character varying(20)
 freq   | integer
 freq_norm | double precision
 geom   | geometry(Point, 4326)
    
```

4.1.2 Alimentación de Base de Datos con Python3

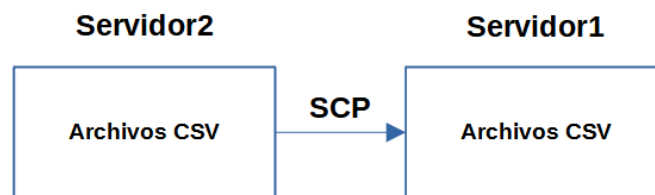
La integración de los puntos de calor a las tablas mencionadas con anterioridad, se realiza por medio de scripts desarrollados en Python, esto nos permite automatizar el proceso y lograr así que se ejecute en momento determinado con un CRON³⁰.

TRASPASO DE ARCHIVOS

Es necesario mandar los archivos del servidor2 (ejecuta el algoritmo de incendios y alberga sus productos) al servidor1. Este último alberga la base de datos así como el Geoserver, por lo que es conveniente tener los archivos CSV también en este servidor. Se crearon dos directorios, **Resultados** para los productos obtenidos cada 10 minutos y **Resultados_dia** para los acumulados por día.

Figura 4.1.5

Diagrama de copia de archivos CSV entre el servidor servidor2 y servidor1



Para copiar los últimos CSV generados se desarrolló un script en *Shell* para copiar los archivos que se generan cada 10 minutos (en el directorio de Resultados) y cada 24 horas (en el directorio de Resultados_dia). El nombre del script es: **cp_last_csv_servidor2_servidor1.sh** y se encuentra en el directorio de **scripts**.

³⁰ Administrador de procesos en segundo plano (también llamados demonios) que permite ejecutarlos en un tiempo determinado

ALIMENTACIÓN DE BASE DE DATOS CON PYTHON3

Se desarrollaron 4 scripts en Python destinados a esta actividad. A pesar de que cada uno ejecuta tareas diferentes, los 4 tienen una misma finalidad: el poder ingresar información a las tablas de las bases de datos y escribir en un archivo de texto los archivos ya agregados. Este archivo de texto, sirve como bitácora y permite al mismo script validar que archivos están dentro de la base de datos y que archivos aún no se han podido anexar a ella.

En la Tabla 4.1.3 se muestra el nombre del script, su objetivo y en que horario se tiene que ejecutar.

Tabla 4.1.3

Scripts que alimentan las tablas de las bases de datos de puntos de calor

Script	Objetivo	Ejecución
Conectdb.py	Agregar todos los puntos de calor albergados en los archivos CSV de la carpeta Resultados, a la tabla puntos_actual y escribe el nombre de los archivos en el documento datos_servidor2.txt . Se ejecutará solamente una vez.	1 vez
Conectdb1.py	Agregar los puntos de calor del último archivo CSV de la carpeta Resultados a la tabla puntos_actual y escribe el nombre del archivo en el documento datos_servidor2.txt .	Cada 10 minutos
Conectdb_dia.py	Agregar todos los puntos de calor albergados en los archivos CSV de la carpeta Resultados_dia, a la tabla puntos_diacom y escribe el nombre de los archivos en el documento datos_diaservidor2.txt . Se ejecutará solamente una vez.	1 vez
Conectdb_dia1.py	Agregar los puntos de calor del último archivo CSV de la carpeta Resultados_dia a la tabla puntos_diacom y escribe el nombre del archivo en el documento datos_diaservidor2.txt .	Cada 24 horas

Los módulos ocupados para estos programas son:

- Psycopg2: Adaptador de base de datos PostgreSQL más popular en Python. “Sus principales características son la implementación completa de la especificación Python DB API 2.0 y la seguridad de subprocesos (varios pueden compartir la misma conexión). Fue diseñado para aplicaciones con múltiples subprocesos que crean y destruyen muchos cursores y hacen una gran cantidad de "INSERT" o "UPDATE" simultáneos.”(*psycopg2 - Adaptador de base de datos Python-PostgreSQL psycopg2 2.8.6*, s. f.)
- Os: “Este módulo proporciona una forma portátil de utilizar la funcionalidad dependiente del sistema operativo. ” (*Python Software Foundation*, s. f.) Unas de sus funcionalidades es poder leer, escribir archivos, manipulación de rutas, creación de archivos y directorios temporales, etc.
- Sys: “Este módulo proporciona acceso a algunas variables utilizadas o mantenidas por el intérprete y a funciones que interactúan fuertemente con el intérprete. Siempre está disponible.”(*sys- Parámetros y funciones específicos del sistema*, s. f.)

4.2 Visualización en Geoserver

Geoserver fungirá como servidor de datos geográficos y poder visualizar los puntos de calor además obtener los servicios WEB espaciales necesarios principalmente los *Web Map Service* (WMS) y *Web Feature Service* (WFS).

4.2.1 Creación de espacio de Trabajo

Para el manejo de Geoserver es necesario crear un espacio de trabajo, para poder facilitar el manejo de capas, en su mayoría el grupo de capas agregadas en los espacios de trabajo tienen un fin en común.

El espacio de trabajo se denominó: *fires*.

Figura 4.2.1

Espacio de trabajo denominado fires.

Editar espacio de trabajo
 Editar un espacio de trabajo existente

Nombre

URI del espacio de nombres

El URI del espacio de nombres asociado con este espacio de trabajo

Espacio de trabajo por defecto
 Isolated Workspace

Configuración

Habilitado

Servicios

WCS
 WFS
 WMS
 WMTS

Guardar **Cancelar**

4.2.2 Almacén de datos

Un almacén de datos contiene la información para realizar el vínculo con la fuente de datos geográficos, en este caso la base de datos espacial. En ella se da la oportunidad de seleccionar el origen de los mismos recaudando la información necesaria para conectar a la base de datos con Geoserver (figura 4.2.2).

Figura 4.2.2

Parámetros de conexión de Geoserver con PostGIS

Parámetros de conexión

host *

port *

database

schema

user *

passwd

Espacio de nombres *
<http://www.openplans.org/topp>

4.2.3 Creación de capas y Consultas

En el visualizador se ocuparon 8 capas, tres relacionadas directamente con el producto de puntos de calor (Puntos de calor de las últimas 24 horas, Puntos de calor por día, Mapa de calor de los puntos de calor de las últimas 24 horas), tres respecto a imágenes satelitales (Compuesto color pseudo-verdadero

GOES 16/ABI³¹ de día y Banda 13³² con luces de ciudad de noche, Compuesto *Fire Temperature* GOES/ABI³³ y Color Verdadero VIIRS³⁴) y dos capas complementarias (Línea de costa con estados de la república mexicana y Áreas Naturales Protegidas). (Tabla 4.2.1)

Tabla 4.2.1

Capas en Geoserver.

Capa	Nombre de la capa en Geoserver
Puntos de los últimas 24 horas	Puntos_10m
Mapa de calor de puntos últimas 24 horas	Calor_10m
Puntos recabados por día	Puntos_diacom
Áreas Naturales Protegidas (2020)	ANP2020
Línea de costa y estados de la república mexicana	Coastline_mexicoMerc
Imagen GOES-16 RGB(Color pseudo-verdadero) actual	Truecolor_GOES
Imagen GOES-16 RGB(Fire Temperature) actual	Fires_GOES
Imagen VIIRS RGB(True color) actual	Truecolor_VIIRS

31 Compuesto color pseudo-verdadero GOES-16/ABI. Se generará utilizando las bandas 1,2 y 3 (0.47 μm , 0.64 μm y 0.86 μm) del satélite GOES-16 sensor ABI. Para la simulación del canal verde se utilizan datos de Himawari-8 AHI con longitud de onda 0.51 μm .

https://www.star.nesdis.noaa.gov/GOES/documents/QuickGuide_CIRA_Geocolor_20171019.pdf

32 Banda 13 de GOES-16/ABI. Ubicada en la longitud de onda de 10.3 μm , tiene como característica la baja sensibilidad a la absorción de vapor de agua, por lo que permite generar una clasificación de la temperatura de nubes.

https://www.star.nesdis.noaa.gov/GOES/documents/ABIQuickGuide_Band13.pdf

33 Compuesto *Fire Temperature* GOES-16/ABI. Compuesto RGB en orden de longitud de onda (3.9, 2.2, 1.6), permite interpretar incendios de baja intensidad (colores rojos) a los de alta (colores blancos y brillantes).

https://www.star.nesdis.noaa.gov/GOES/documents/QuickGuide_Fire_Temperature_RGB.pdf

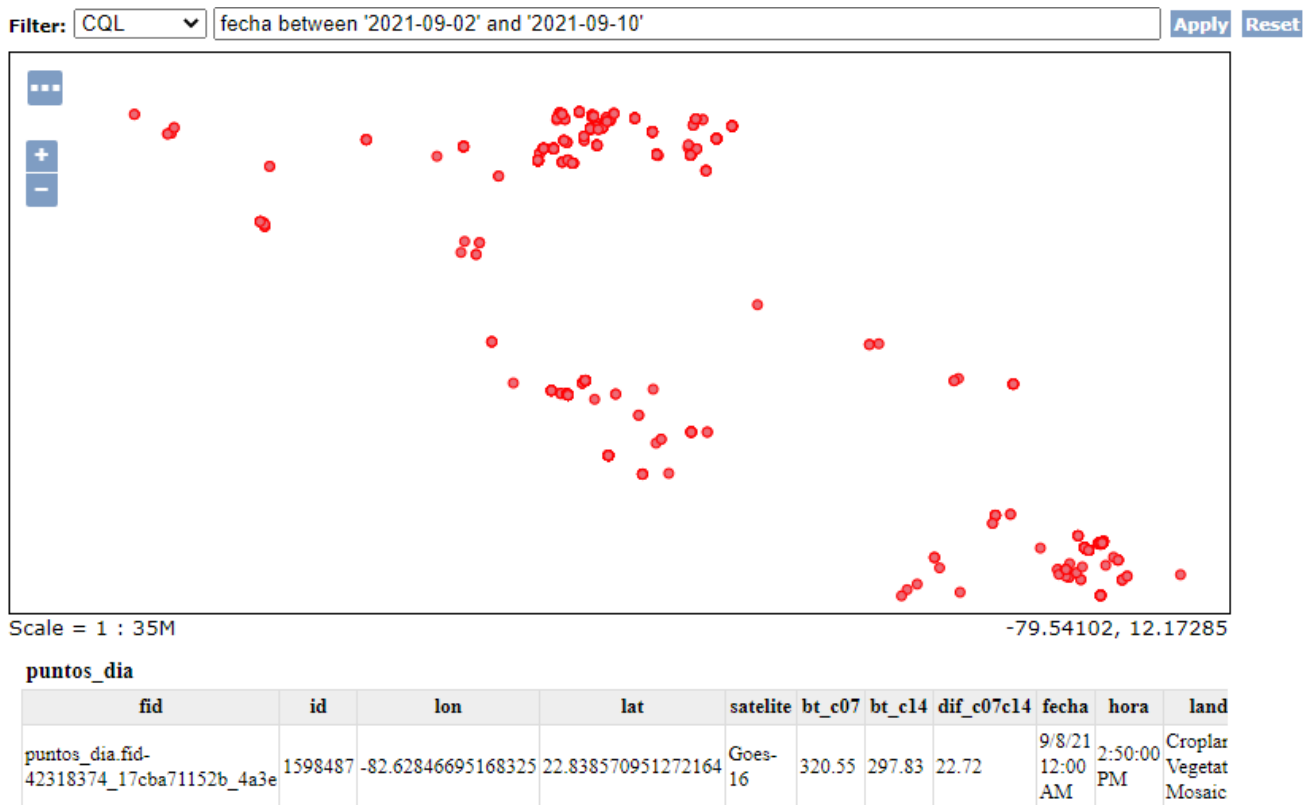
34 Compuesto color verdadero VIIRS. Compuesto RGB de las bandas M5(0.672 μm), M4(0.555 μm) y M3 (0.488 μm) del Sensor VIIRS.

<https://coastwatch.noaa.gov/cw/satellite-data-products/imagery/true-color-viirs-snpp.html>

Dividiendo las capas vectoriales por su actualización se tomarían las relacionadas con los puntos de calor como Dinámicas mientras que las estáticas, son las áreas naturales protegidas y línea de costa con estados de la república mexicana. Aquellas capas denominadas dinámicas son capaces de cambiar por medio de consultas a sus atributos, para este fin Geoserver nos permite realizar consultas CQL³⁵ (Figura 4.2.3).

Figura 4.2.3

Sentencia CQL y Visualización de la capa Puntos_día en Openlayers3 comprendida de 2021-09-02 al 2021-09-10



³⁵ CQL (*Common Query Language*) es un lenguaje de consulta creado por OGC para la especificación *Catalog Web Services*.

4.2.4 Estilos

Los estilos representan la parte visual de los datos espaciales puesto que controlan su apariencia. Geoserver cuenta con 4 diversos formatos de estilos (Tabla 4.2.2)

Tabla 4.2.2

Formatos de estilos aceptados por Geoserver.






Estilo	Descripción
Descriptor de capa con estilo (SLD, siglas en inglés)	Estándar OGC para el estilo geoespacial y disponible por defecto en Geoserver.
Hojas de estilo en cascada (CSS, por sus siglas en inglés)	Tiene una sintaxis similar con el CSS convencional, está disponible a través de una extensión.
YSLD	Basado en YAML, es un formato equivalente al SLD y está disponible mediante la extensión yslid.
MBStyle	Sintaxis basada en JSON, disponible por la extensión mbstyle.

Nota: Tomada de (Geoserver, s.f.)

Se desarrollaron 3 estilos en formato SLD, los cuales se asignaron a los datos de los Puntos de calor y las Áreas Naturales Protegidas. Para **Puntos_10m** se determinó un polígono en forma circular con borde amarillo y relleno rojo mientras que para **Puntos_díacom** el borde es rojo y el relleno rosa. En cuanto al mapa de calor, se tomó una paleta de colores fríos a cálidos, esta se ajustará dependiendo el mayor número de puntos en escena (Tabla 4.2.3).

Tabla 4.2.3

Estilos de capas en Geoserver.

Capa/Estilo	Descripción	Visualización
Puntos_10m / P_hoy	Circulo con borde amarillo y relleno rojo	
Puntos_diacom/ Consulta	Circulo con borde rojo y relleno rosado	
Calor_10m/fires_heatmap	Azul < 10% 10% < Rojo < 50% 50% < Amarillo < 100%	
ANP2020/ANP	Polígono Verde con transparencia en su relleno	
Coastline_mexicoMerc/ Coastline_white	Línea blanca	

4.3 Desarrollo WEB

Con el propósito de no cargar de información el visualizador, se decidió realizar una página previa a este; misma que contiene información relacionada con el proyecto del algoritmo de detección de incendios, así como contenido del visualizador que podría servir al usuario de manera introductoria. De la misma forma, incluye un apartado donde se despliegan en forma de gráficas de barras, estadísticas relacionadas con los puntos de calor detectados, como lo son:

- Estados de la república con mayor números de puntos de calor detectados.
- Número de puntos de calor detectados en los últimos 7 días.
- Número de puntos de calor detectados por mes.

Por otra parte, la página destinada al visualizador cuenta con herramientas desarrolladas (en su parte lógica) en lenguaje javascript, mismas que permiten al usuario navegar dentro de la interfaz gráfica y obtener información de las capas que se despliegan en el mapa.

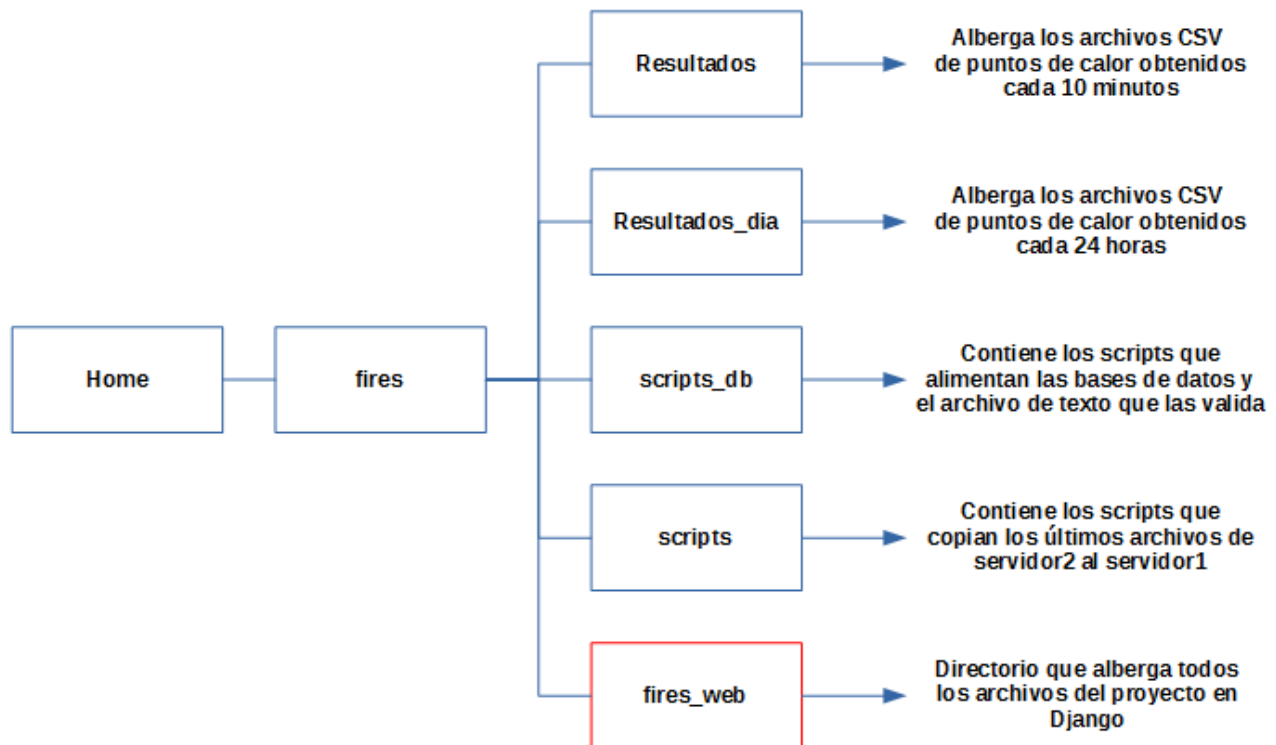
4.3.1 Estructura de directorios

Django cuenta con una organización de directorios y archivos que permite un fácil despliegue y manejo de la información. Es por ello que se creó un proyecto con el nombre de “Fires_web” en el que contiene toda la aplicación. Se crearon tres directorios, además del homónimo al proyecto, mismos que albergaran la siguiente información (Figura 4.3.2).

- **Fires_web** : Todos los archivos relacionados al proyecto creados por Django (view, urls, etc).
- **Static** : Los archivos estáticos de la aplicación. Este directorio a su vez contiene las carpetas de img, js, css.
- **Puntos_calor**: Carpeta creada por Django al momento de establecer la aplicación de puntos_calor.
- **Templates**: Archivos HTML que albergan la parte visual de la aplicación.

Figura 4.3.1

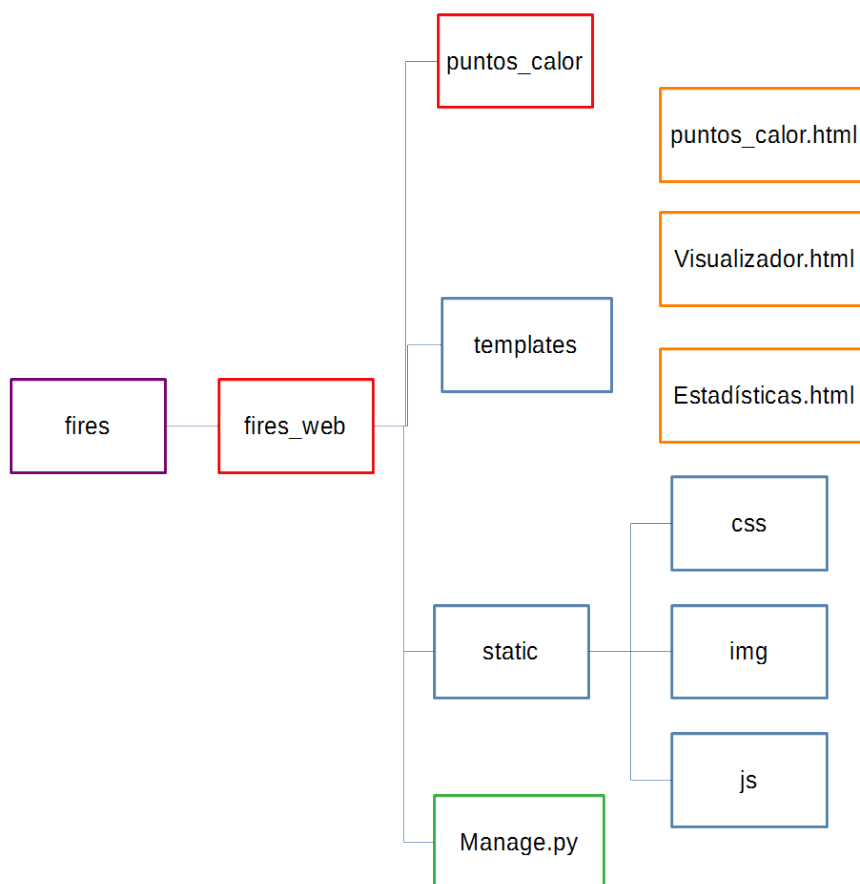
Árbol de directorios, servidor1.



También se cuenta con el archivo ejecutable con el que se ejecutará la aplicación, este cuenta con el nombre de **manage.py**. Al ser un archivo ejecutable de Python, basta con teclear en consola “**python manage.py runserver**” para poder ejecutar nuestra aplicación. Esto es recomendable para aplicaciones en desarrollo, debido que en consola nos podrá mostrar todos los errores (en caso de existir), conexiones remotas, líneas de código ejecutadas, etc. Para aplicaciones en ejecución se recomienda utilizar un servidor WEB como en este caso, Apache.

Figura 4.3.2

Diagrama de directorios, representado por colores. Morado: nombre de la cuenta de usuario, Rojo: proyecto y aplicación de Django, Azul: Directorios, Verde: Archivo ejecutable de Python y Naranja: Archivos en HTML.



4.3.2 Página principal

Está dividida en tres segmentos importantes, anclados³⁶ en la barra de navegación:

- **Algoritmo de puntos de calor:** Contiene información relevante al algoritmo de puntos de calor desarrollado en el LANOT.
- **Visualizador de puntos de calor:** Alberga una pequeña introducción al visualizador así como un carrusel con imágenes representativas del mismo.
- **Estadísticas:** Muestra estadísticas de los puntos de calor en gráficas de barras.

En el footer, se muestra conexión con a otros proyectos que desarrolla el Laboratorio Nacional de Observación de la Tierra y sitios que podrían causar interés al usuario por su relación con el tema de incendios en la república mexicana (Alerta temprana de puntos de calor CONABIO, Sistema de Predicción de Peligro de Incendios Forestales en México).

Estadísticas

Para la obtención de estadísticas se realizaron 3 tablas en la base de datos (Tabla 4.3.1), que recabarán información relacionada con los puntos de calor y así ser mostrada por la aplicación. Para ello se implementó un script en Python (**estadisticasp.py**) que nos permite actualizar las tablas cada día por medio de un Cron en Linux. Para el despliegue de las gráficas en la aplicación se utilizó la biblioteca de javascript “Morris” (<https://morrisjs.github.io/morris.js/index.html>).

³⁶ Elemento HTML (<a>) que crea enlaces a otras páginas, archivos o ubicaciones dentro de la misma página.

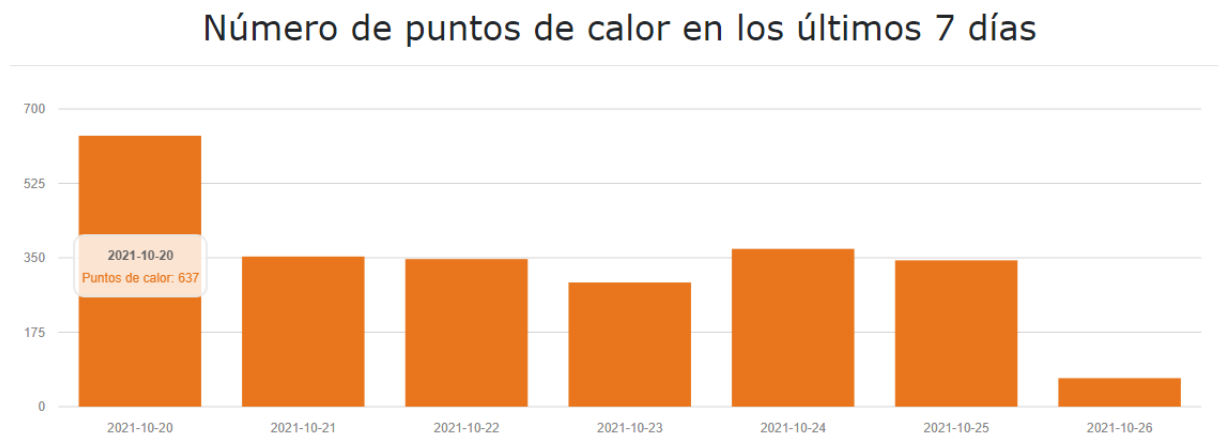
Tabla 4.3.1

Tablas para estadísticas.

Nombre de la tabla	Atributos	Descripción
estados_pc	estado, puntos_count	Mostrar los 10 estados de la república mexicana con mayor número de puntos de calor detectados.
pc_mes	mes, puntos_count	Mostrar los puntos de calor detectados por mes.
pc_sietedias	dia, puntos_count	Mostrar los puntos de calor detectados en los últimos siete días.

Figura 4.3.3

Gráfica de barras de los puntos de calor de los últimos 7 días (20 al 27 de octubre del 2021).



4.3.3 Visualizador de puntos de calor

Para el visualizador se desarrolló una estructura que cumpla los objetivos generales de esta tesis. Por lo que se podría categorizar en tres funciones: visualización de información cartográfica , descarga de puntos de calor y manipulación de información. Estas tres más la plantilla base desarrollada en HTML es como se obtiene el visualizador de puntos de calor (Figura 4.3.5).

Visualización cartográfica

En ella se recae la acción de plasmar la información cartográfica en un mapa. En nuestro caso, el objeto de estudio abarca el territorio nacional mexicano, parte del sur de Estados Unidos de América y parte del Caribe. Por lo que, nuestra escena abarca los siguientes límites longitud mínima (-140), longitud-máxima (-51), latitud mínima (1) y latitud máxima (37). En coordenadas geográficas (WGS84, EPSG:4326).

Las capas se categorizaron en: vectoriales, ráster y mapas base. Las vectoriales (Tabla 4.3.2) cuentan con cinco elementos (el mapa de calor de puntos de calor de las últimas 24 horas, a pesar de no tener una geometría se consideró vectorial por su relación con los puntos de calor), mientras que las ráster y mapas base solo con tres cada una. GOES-16/ABI Color verdadero (Pseudo-verdadero), GOES-16/ABI Fire temperature y VIIRS Color verdadero para las capas ráster y Mapa base Gris Argis Online, Mapa base satelital de Google y Mapa base de OpenStreetMap para el grupo de mapas base.

Tabla 4.3.2

Capas de visualizador de puntos de calor por categoría

Capa	Geometría	Atributos
Puntos de calor últimas 24 horas	Puntos	Número identificador, Coordenada en X, Coordenada en Y, Longitud, Latitud, Satélite, Temperatura de brillo banda 7, Temperatura de brillo banda 14, Diferencia tem, brillo banda 7-14, Fecha de detección,

Puntos de calor consulta	Puntos	Hora GMT de detección, Cobertura terrestre, Estado, País, Área Natural Protegida 2020 Número identificador, Coordenada en X, Coordenada en Y, Longitud, Latitud, Satélite, Temperatura de brillo banda 7, Temperatura de brillo banda 14, Diferencia tem, brillo banda 7-14, Fecha de detección, Hora GMT de detección, Cobertura terrestre, Estado, País, Área Natural Protegida 2020, Frecuencia en que se detectó un punto de calor, Frecuencia normalizada
Mapa de calor de los puntos de calor últimas 24 horas	N/A	N/A
Estados y línea de costa	Líneas	N/A
Áreas Naturales Protegidas 2020	Polígonos	Identificador del área, Nombre del área, CAT_DECRET, CAT_MANEJO, Estados, Municipios, Región, Superficie, Superficie terrestre, Superficie marina, PRIM_DEC, ULT_DOF, PCM1, SINAP

Manipulación de la información

Para la manipulación de datos se crearon diversas herramientas (Tabla 4.3.3) que nos permitieran realizar:

- Consulta de atributos (en tabla de atributos para puntos de calor y en ventana emergente para Áreas Naturales Protegidas 2020).
- Despliegue de capas.
- Botones de *zoom* al mapa.
- Despliegue de gráficas de estadísticas.
- Conteo de puntos de calor encerrados en un polígono rectangular.
- Medición de distancias y áreas.
- Barra de búsqueda de lugares (*geocoding*³⁷).
- Escala gráfica
- Recuadro de latitud y longitud.
- Consultas a los puntos de calor por fechas.
- Despliegue de simbología al mapa.

37 Geocoding: Proceso de transformar la descripción de una ubicación en coordenadas y asignarle un lugar en la superficie terrestre. <https://desktop.arcgis.com/en/arcmap/latest/manage-data/geocoding/what-is-geocoding.htm>

Tabla 4.3.3

Herramientas del visualizador de puntos de calor GOES-16/ABI.

Herramienta	Descripción	Ícono
Búsqueda de lugares	Interpreta la descripción de una ubicación para verla reflejada en el mapa	
Consulta de atributos	Muestra los atributos de la capa de ANP 2020 con ventana emergente	
Consulta temporal de puntos de calor	Permite realizar consultas temporales de los puntos de calor	
Conteo de puntos de calor	Permite contabilizar los puntos de calor dentro de un recuadro dibujado por el usuario y a su vez muestra sus atributos	
Escala gráfica	Escala gráfica del mapa	
Gráficas de estadísticas de puntos de calor	Nos muestra las estadísticas relacionadas con los puntos de calor	
Latitud y Longitud	Muestra la posición del puntero (en coordenadas geográficas)	
Medir (área/distancia)	Permite realizar mediciones de distancias y áreas	
Simbología	Despliega la leyenda relacionada con las capas activas	
Tabla de atributos	Muestra los atributos de los puntos de calor	
Zoom	Acerca o leja el mapa	

Descarga de puntos de calor

El visualizador cuenta con un botón a un costado de las capas de puntos de calor, el cual despliega un menú mostrando los formatos en los que se puede descargar los puntos de calor (Geojson y CVS) y a la fecha correspondiente a los puntos de calor. (Figura 4.3.4)

Figura 4.3.4

Despliegue de descarga de puntos de calor de las últimas 24 horas. A partir del 2022-09-20, 17:20

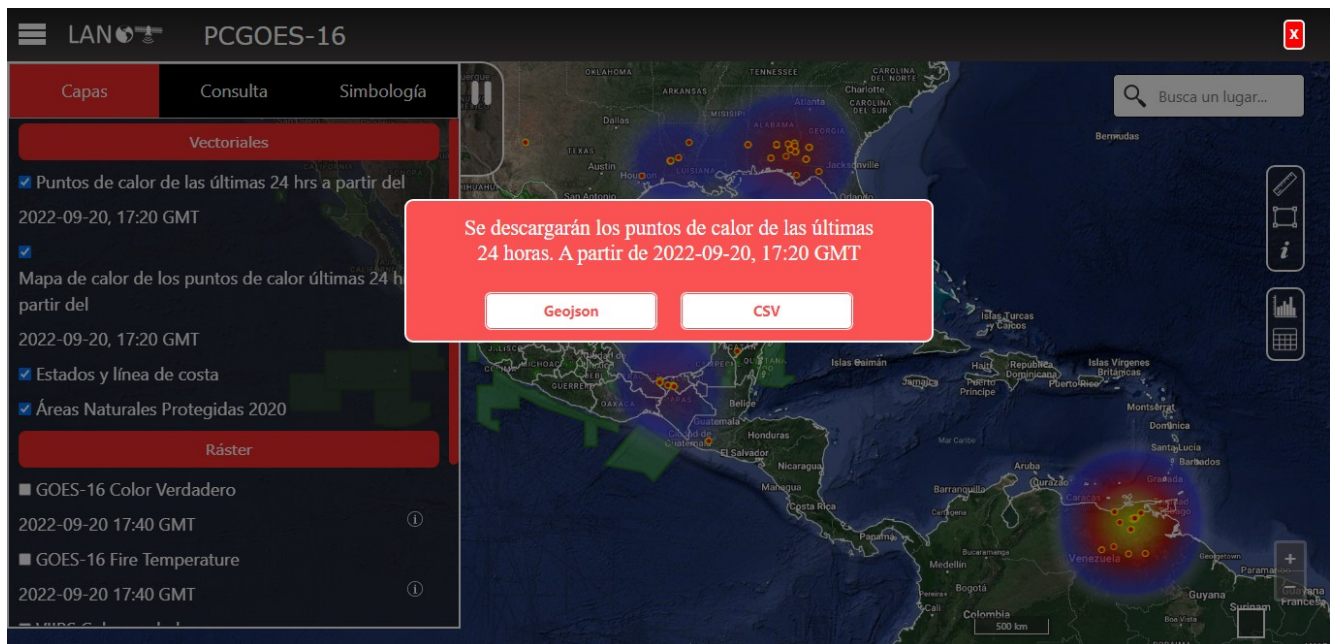
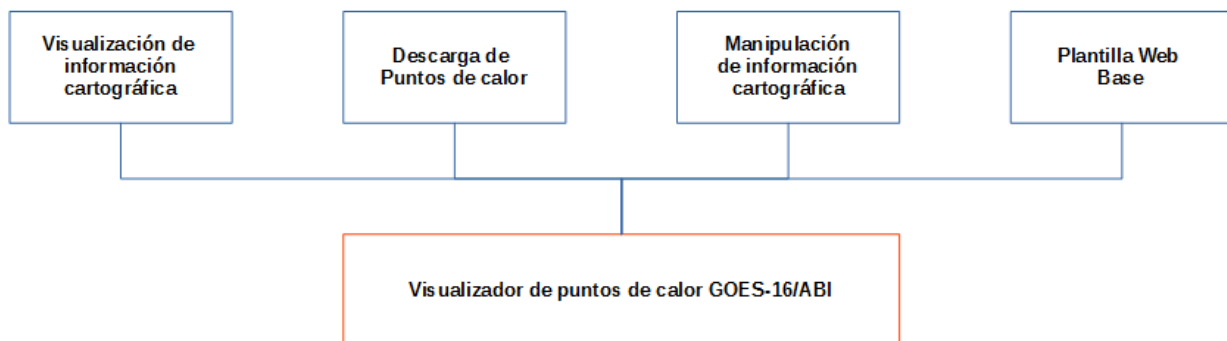


Figura 4.3.5

Desarrollo de visualizador de Puntos de calor GOES-16/ABI.



5 Resultados

El trabajo mostrado en esta tesis, dio como resultado una aplicación SIG Web que permite al usuario navegar en una interfaz intuitiva y conocer sobre producto del algoritmo de detección de puntos de calor GOES-16/ABI desarrollado por el LANOT, tanto de una manera teórica por medio de una página principal al proyecto, como práctica al desplegar la información espacial (capas) en el visualizador.

Cabe mencionar que la plantilla mostrada en el apartado 5.1 (Página principal), ha sido modificada por el LANOT recientemente (04/10/2022) debido a la actualización de su portal. A pesar de eso, esta no modificó en absoluto su contenido textual informativo y su estructura, exceptuando la parte de contacto.

5.1 Página principal

La página principal³⁸ se encuentra en la dirección http://sargazo.lanot.unam.mx/lanot/puntos_calor/, y muestra información relacionada con el algoritmo y las imágenes mostradas en el visualizador, así como las estadísticas relacionadas con los puntos de calor representadas en gráficas de barras.

Figura 5.1.1

Página principal, apartado de Algoritmo de puntos de calor. Desplegada desde Google Chrome.



Debajo de la información relacionada con el Visualizador se muestra un carrusel de imágenes representativas del mismo. De igual forma se tiene un acceso directo a el, por medio de un botón (Figura 5.1.2).

³⁸ Las figuras que se muestran, se propusieron inicialmente, sin embargo fueron modificadas por el LANOT recientemente, tras la actualización de su portal.

Figura 5.1.2

Botón de acceso a visualizador.



Figura 5.1.3

Página principal, apartado de contacto. Desplegada desde Google Chrome.



Figura 5.1.4

Footer de página principal. Desplegada desde Chrome.



5.1.1 Estadísticas

Se muestran 3 gráficas representativas de los puntos de calor:

- Número de puntos de calor por mes (año 2021) . Figura (5.1.6)
- Número de puntos de calor en los últimos 7 días (del 20 a 27 de octubre 2021). Figura (5.1.7)
- Estados con mayor puntos de calor detectados desde que se tuvieron registro. Figura(5.1.8)

Figura 5.1.5

Gráfica de número de puntos de calor por mes en el año 2021. Tomada el 27 de octubre de 2021.

Número de puntos de calor por mes (Año 2021)

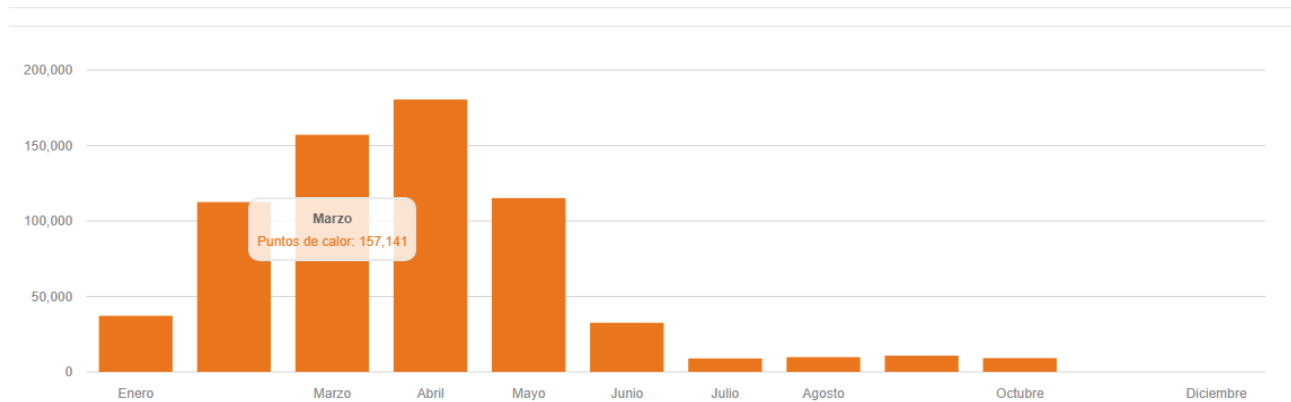


Figura 5.1.6

Gráfica de número de puntos de calor en los últimos 7 días. Tomada el 27 de noviembre de 2021.

Número de puntos de calor en los últimos 7 días

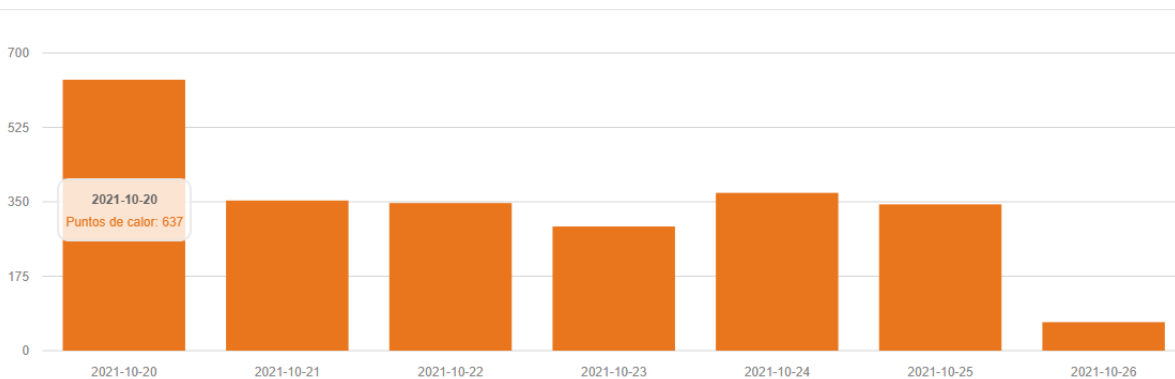
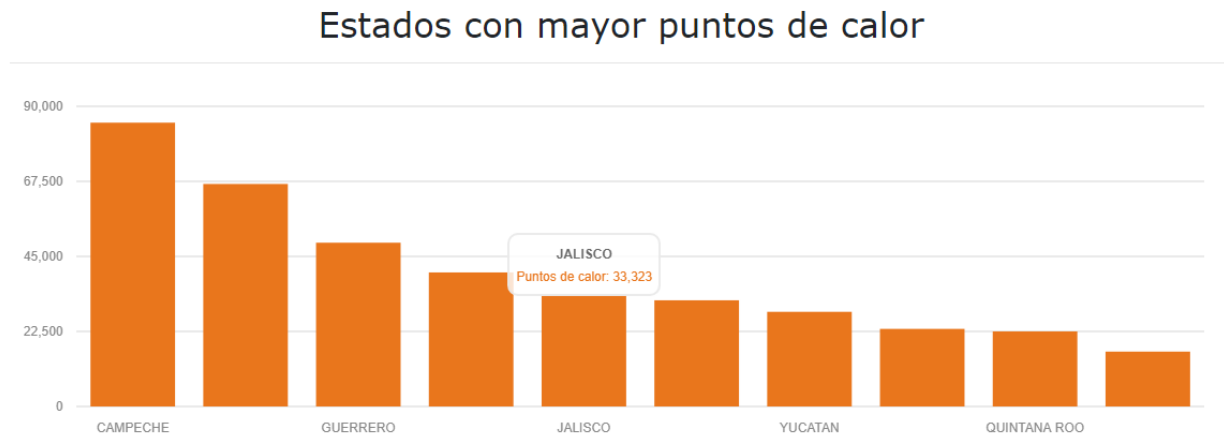


Figura 5.1.7

Gráfica con los estados con mayor número de puntos de calor detectados. Tomada el 27 de octubre



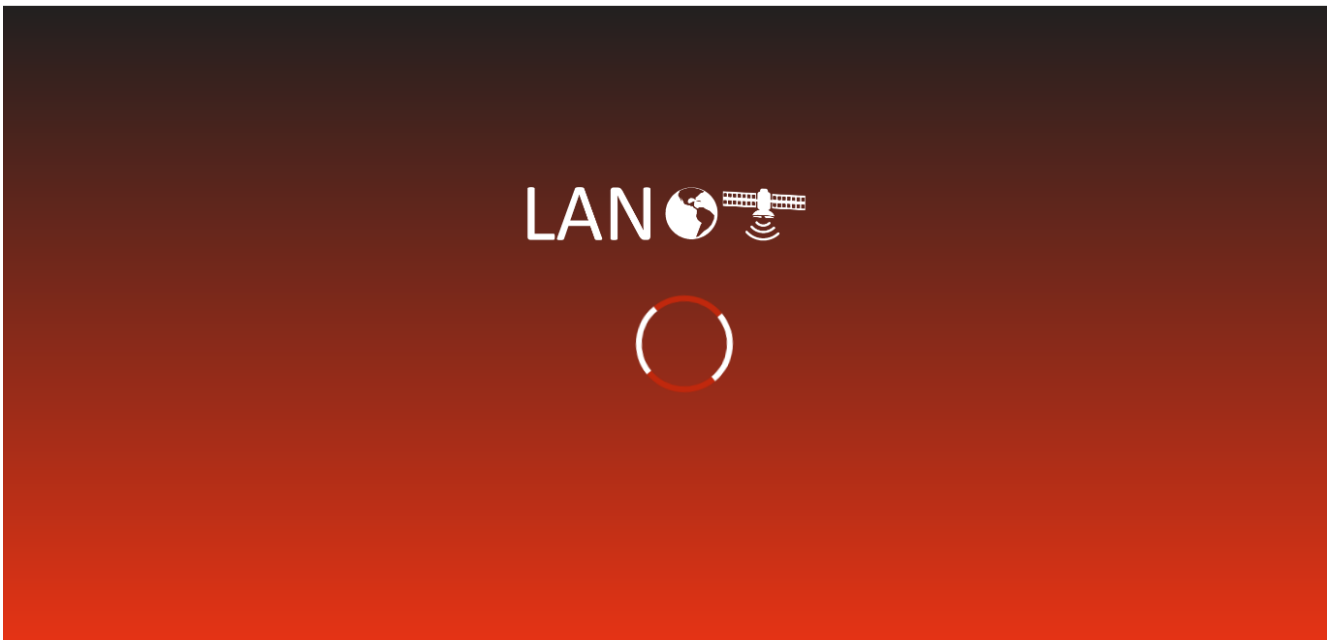
5.2 Visualizador puntos de calor GOES-16/ABI

Se puede ingresar al visualizador de puntos de calor GOES-16/ABI de dos formas, la primera es por medio del botón ubicado en la zona media de la página principal del proyecto (apartado 5.1) y la segunda es por medio de la dirección http://sargazo.lanot.unam.mx/lanot/puntos_calor/visualizador/.

Al cargar la página se muestra una animación para una mejor espera del usuario. Tiene como fondo un degradado vertical del negro al rojo y en el centro el logo del LANOT con una rueda con movimientos en sentidos de las manecillas del reloj.

Figura 5.2.1

Animación de carga para el despliegue de la interfaz gráfica del visualizador de puntos de calor GOES-16/ABI.

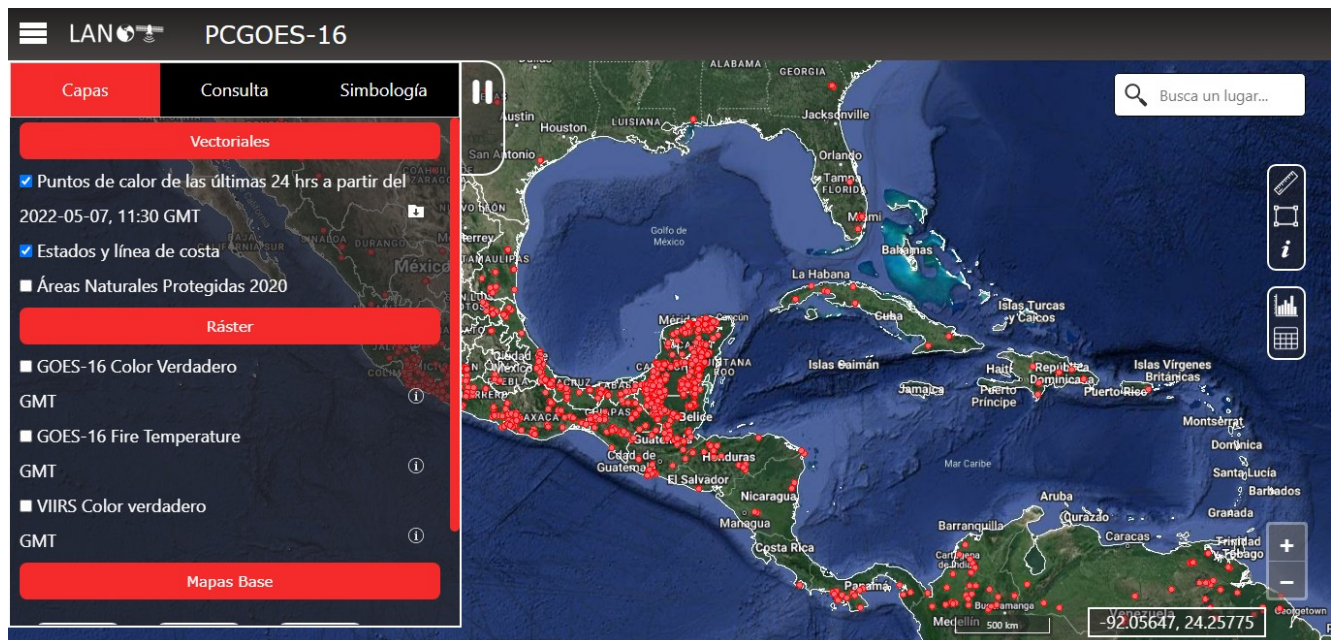


5.2.1 Interfaz del mapa

En el despliegue de la interfaz gráfica del visualizador en un dispositivo de escritorio, se puede observar de lado izquierdo el recuadro de menú con cuatro pestañas (Capas, Consulta y Simbología). Por la parte superior derecha se encuentra la barra de búsqueda de lugares y por debajo de él las barras de herramientas. Del mismo lado en su parte inferior se ubican los botones de *zoom*, coordenadas de navegación y la barra de escala gráfica (Figura 5.2.2).

Figura 5.2.2

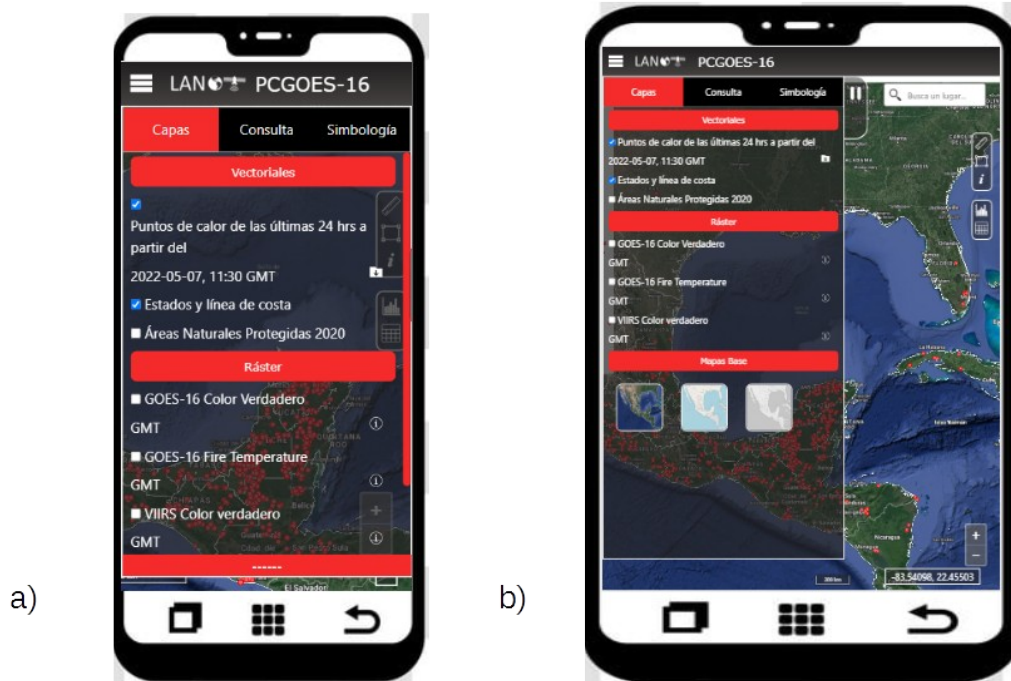
Interfaz del visualizador de puntos de calor, del 17 de agosto del 2022.



En cuanto corresponde a las tabletas, estas conservan las mismas dimensiones que la versión de escritorio. Por el contrario, los dispositivos móviles al cargar el visualizador, lo primero que se observa es el menú abarcar toda la pantalla. Este en su parte inferior contiene una pestaña que al apretarla permite contraerlo y permitir así al usuario navegar en el visualizador.

Figura 5.2.3

Despliegue de interfaz del visualizador. a) Versión celular y b) Versión tableta



5.2.2 Menú Puntos de Calor

El menú se concentra en 3 pestañas: Capas, Consulta y Simbología. La primera despliega todas las capas que alberga el visualizador, tomando las 3 categorías mencionadas anteriormente (vectoriales, ráster y mapas base). La segunda pestaña se refiere a la consulta temporal de puntos de calor, el usuario puede seleccionar la fecha de inicio y de término de los puntos de calor a querer desplegar. Por último, la tercer pestaña (simbología) muestra al usuario la leyenda relacionada con la capa que esté activa.

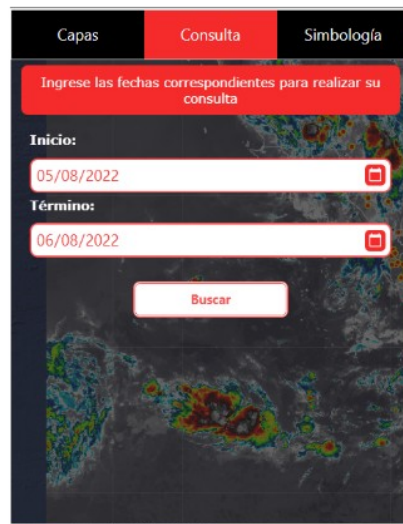
Figura 5.2.4

Pestañas de menú del Visualizador de Puntos de calor GOES-16/ABI.

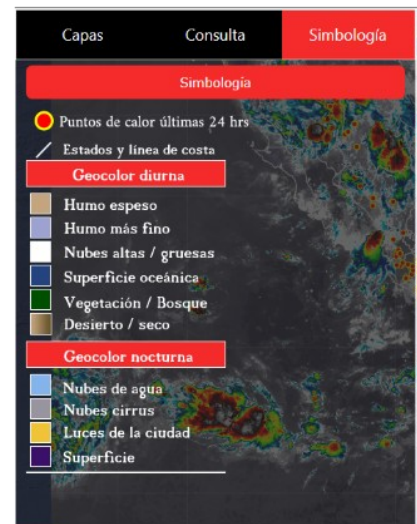
a)



b)



c)



Nota: a) Pestaña de Capas, b) Pestaña de Consulta de puntos de calor, c) Pestaña de Simbología

5.2.3 Capas del visualizador

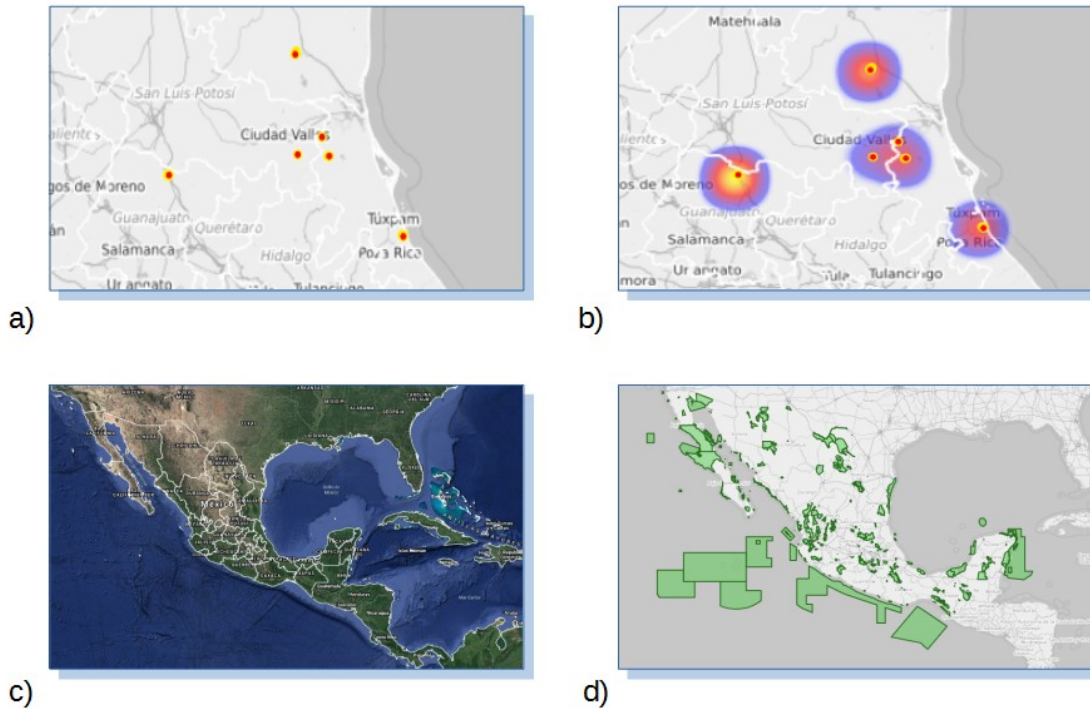
Vectoriales

Al desplegar el menú de capas del visualizador se mostrará en la parte superior el apartado destinado a las capas de carácter vectorial (Figura 5.2.5):

- Puntos de calor de las últimas 24 horas, a partir de la última detección.
- Mapa de calor de los puntos de calor de las últimas 24 horas, a partir de la última detección.
- Línea de costa y estados de la república mexicana.
- Áreas Naturales Protegidas (INEGI 2020)

Figura 5.2.5

Despliegue de capas vectoriales.



Nota: a) Puntos de calor de las últimas 24 horas a partir del 15 de diciembre 2021 a las 19:10 GMT. En los estados de Guanajuato, San Luis Potosí, Tamaulipas y Veracruz de Ignacio de la Llave. b) Mapa de calor de los puntos de calor de las últimas 24 horas a partir del 15 de diciembre 2021 a las 19:10 GMT. En los estados de Guanajuato, San Luis Potosí, Tamaulipas y Veracruz de Ignacio de la Llave. c) Línea de costa y estados de la república mexicana. d) Áreas Naturales protegidas de la república mexicana, INEGI 2020.

Realizada una consulta de puntos de calor por fechas, se desplegará una capa extra en el apartado de capas vectoriales: Puntos de calor del “fecha de inicio” al “fecha de término” (Figura 5.2.6).

Figura 5.2.6

Puntos de calor del 1 de diciembre 2021 al 10 de diciembre 2021. En los estados de San Luis Potosí, Tamaulipas y Veracruz de Ignacio de la Llave.



Mapas base

La estructura de los mapas base se conformó por medio de imágenes relacionadas con el tipo de mapa a representar. Estas a su vez son botones que al generar un clic cambiará el mapa base en el visualizador (Figura 5.2.7).

Figura 5.2.7

Apartado de Mapas base en visualizador de puntos de calor GOES-16/ABI.

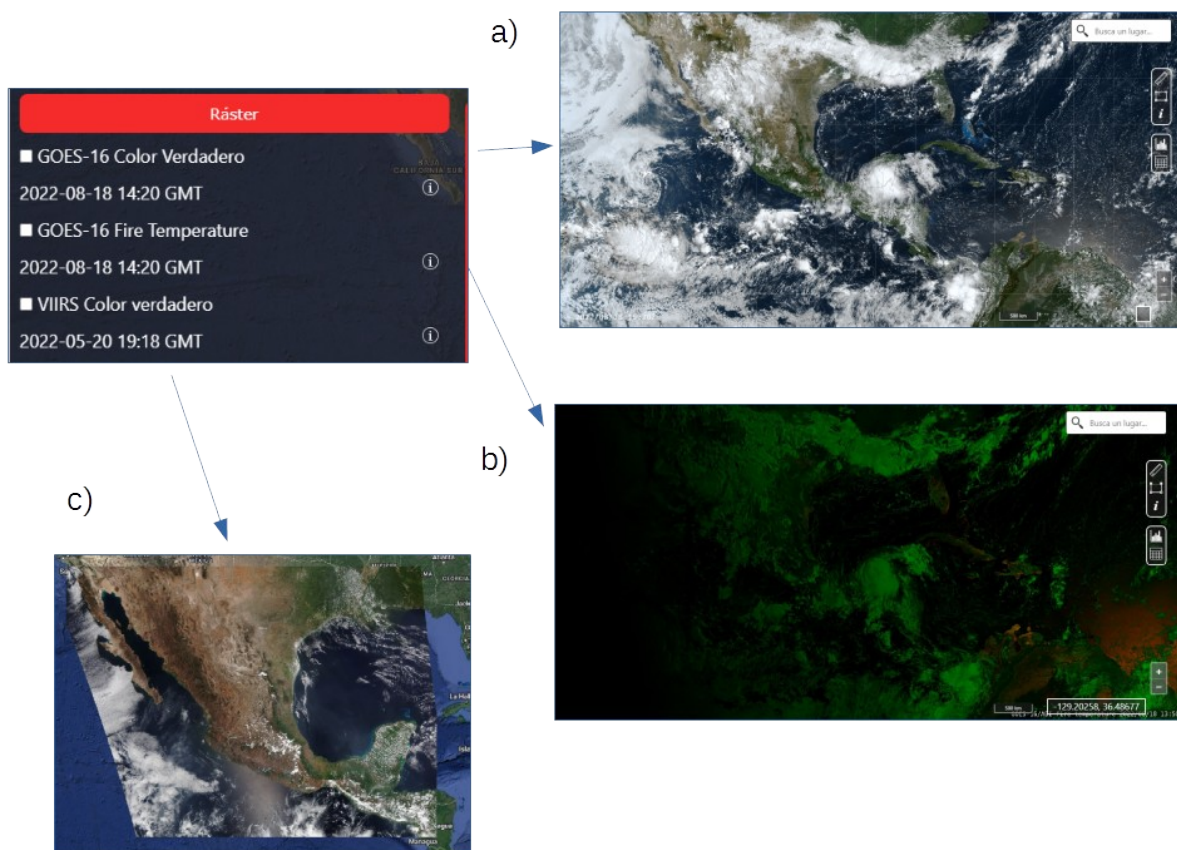


Ráster

Las capas tipo ráster pertenecen a las imágenes satelitales, en este visualizador se mostrarán solo de los sensores ABI y VIIRS. (Figura 5.2.8)

Figura 5.2.8

Despliegue de capas ráster en el visualizador de puntos de calor GOES-16/ABI.



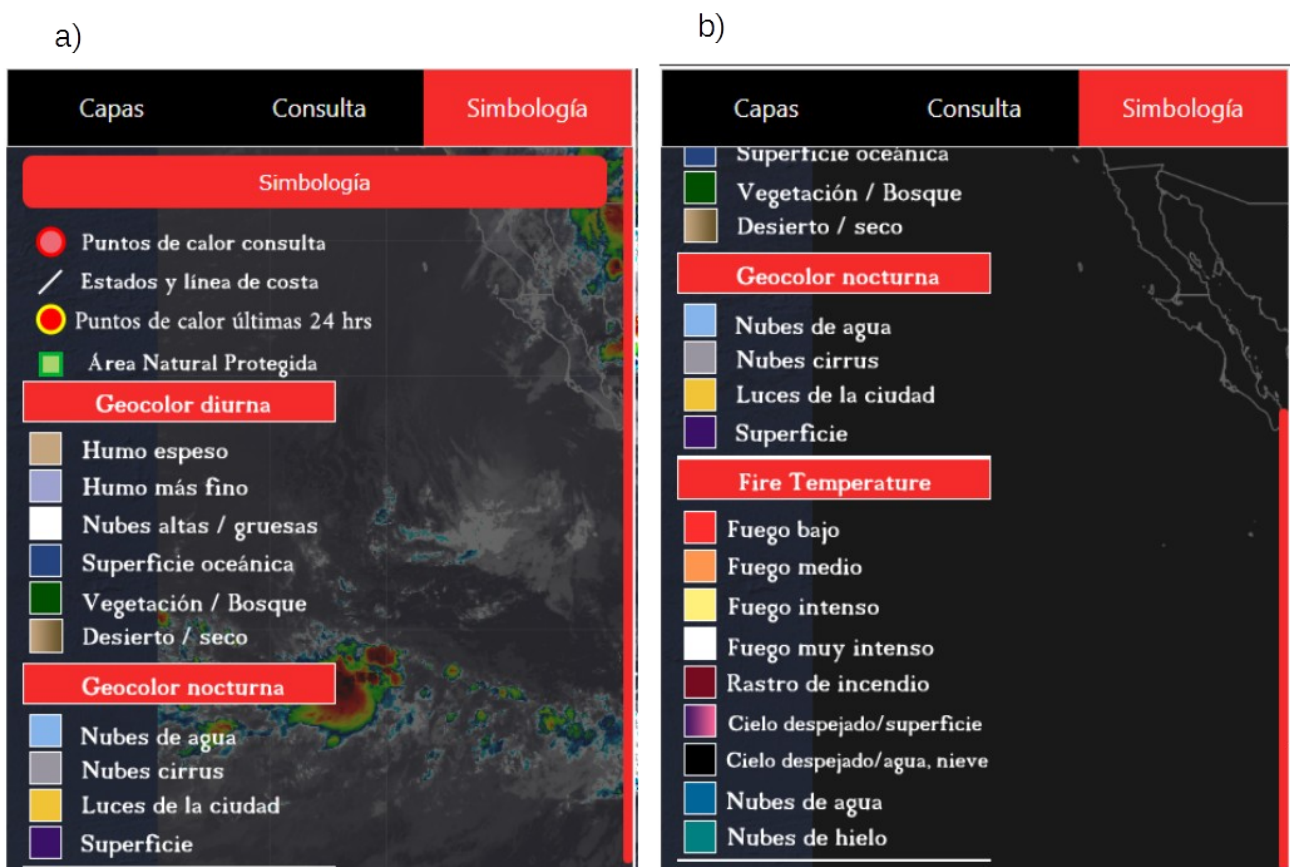
Nota: a) GOES-16 Color Verdadero del 18-08-2022 a las 14:20 GMT. b) GOES-16 Fire temperature del 18-08-2022 a las 14:20 GMT. c) VIIRS Color verdadero del 20-05-2022 19:18 GMT

5.2.4 Simbología

La pestaña de simbología muestra las leyendas de las capas activas en el visualizador. En el caso de las imágenes satelitales correspondientes al sensor ABI, se tomaron las clasificaciones de mostradas en la guía rápida de los productos.

Figura 5.2.9

Representación del apartado de Simbología del visualizador de puntos de calor GOES-16/ABI.



Nota: a) Apartado de simbología de puntos de calor, estados, línea de costa, área natural protegida y Geocolor Diurna / nocturna. Tomadas de la guía rápida del producto Geocolor (Color pseudoverdadero):

https://rammb.cira.colostate.edu/training/rmtc/docs/QuickGuides/QuickGuide_CIRA_Geocolor_es.pdf

b) Apartado de simbología del compuesto Geocolor nocturno y Fire temperature. Tomada de la guía rápida del producto Fire temperature:

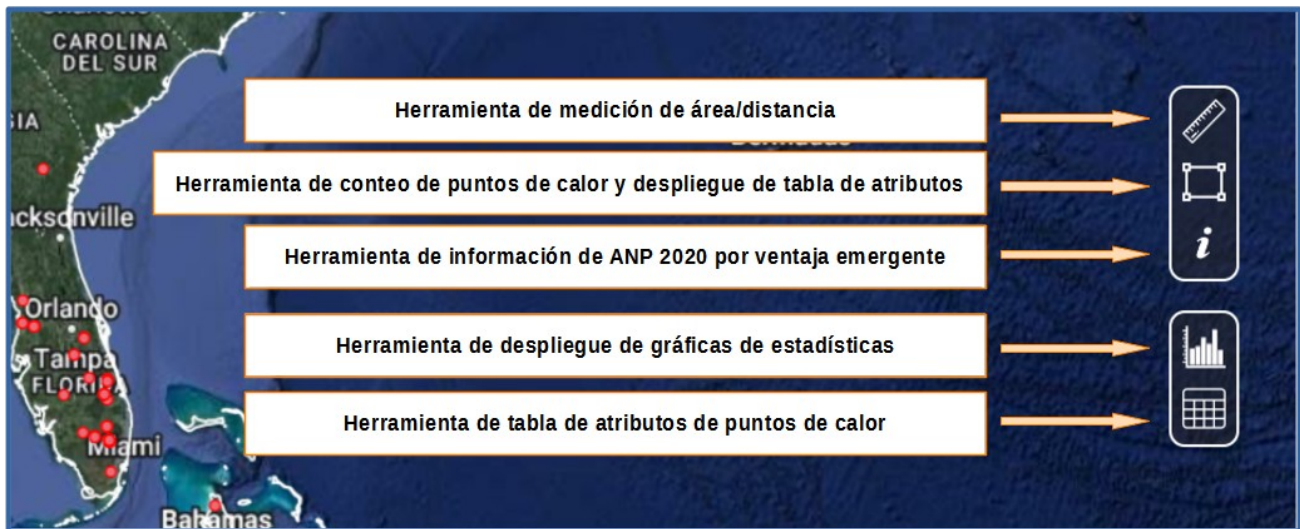
https://rammb.cira.colostate.edu/training/visit/quick_guides/Fire_Temperature_RGB.pdf

5.2.5 Barras de herramientas

Esta barra se ubica del lado derecho del visualizador y contiene 5 herramientas que permiten al usuario interactuar con la información cartográfica plasmada en el mapa (Figura 5.2.11).

Figura 5.2.10

Barra de herramientas del visualizador de puntos de calor GOES-16/ABI



Herramienta de medición área/distancia

Basada en el módulo “*Sphere*” de Openlayers, esta herramienta nos permite realizar mediciones esféricas sobre el mapa, utilizando los métodos *getArea* y *getLength* (Openlayers, s.f).

Figura 5.2.11

Herramienta de medición, opción distancia.



Figura 5.2.12

Herramienta de medición, opción área.



Herramienta de conteo de puntos de calor y atributos

Muestra en pantalla el número de puntos de calor detectados en toda la escena y los contenidos en un rectángulo, dibujado por el usuario en el mapa (Figura 5.2.14). Adicionalmente, se despliega su tabla de atributos (Figura 5.2.13). Esta última es posible desactivarla con el icono de la tabla de atributos.

Figura 5.2.13

Herramienta de conteo de puntos de calor del visualizador de puntos de calor GOES-16/ABI.

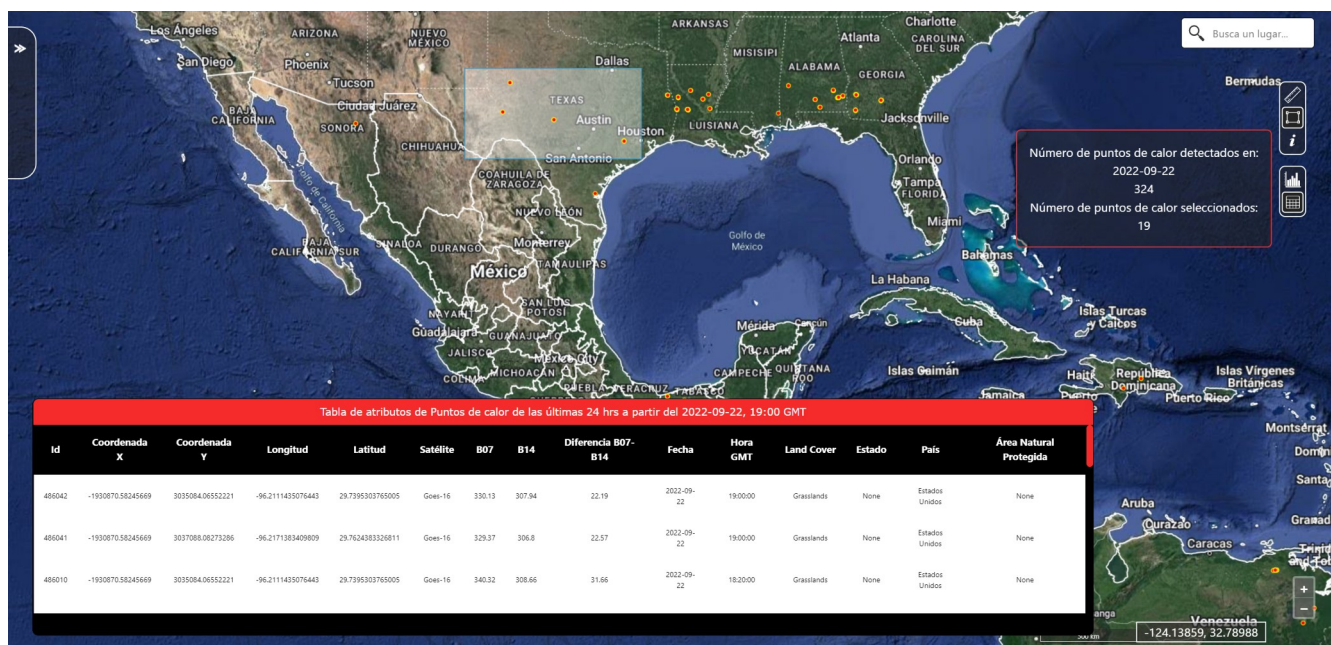
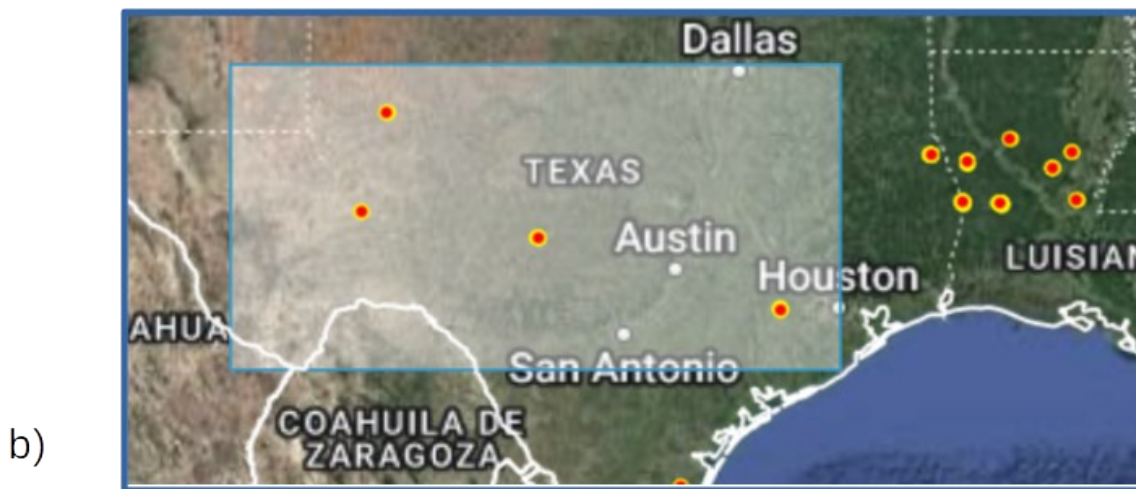
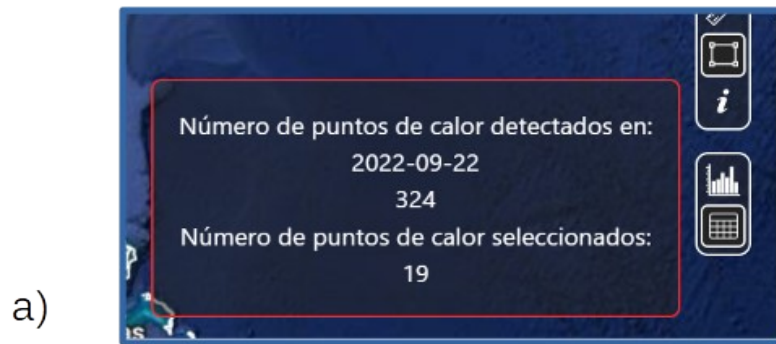


Figura 5.2.14

Ejemplo de: a) Conteo de puntos de calor detectados en toda la escena el 22-09-2022 y los puntos de calor contenidos por el rectángulo dibujado por el usuario. b) Rectángulo dibujado que contiene los puntos de calor.

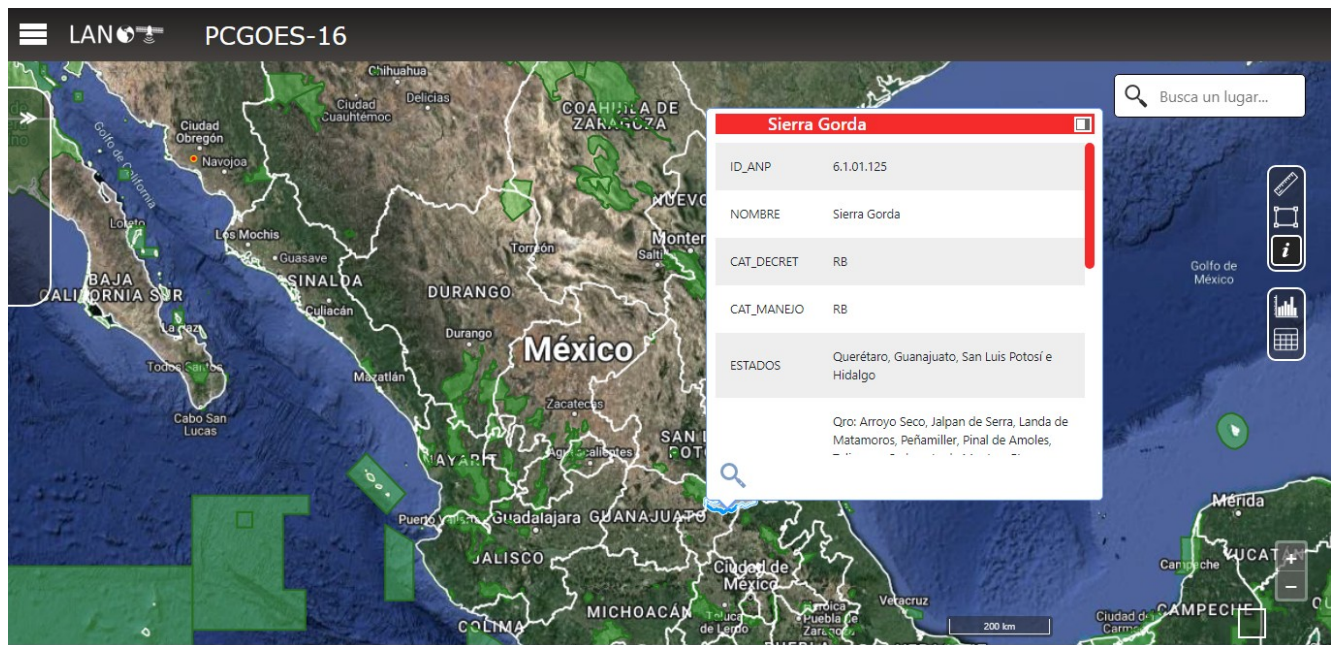


Herramienta de consulta de atributos por ventana emergente

Al activar esta herramienta se puede obtener la información de la capa de Áreas Naturales Protegidas 2020, en forma de venta emergente. Solo es necesario activar la capa y dar clic en un polígono relacionado a esta (Figura 5.2.15).

Figura 5.2.15

Herramienta de consulta de información con ventas emergentes para la capa de Áreas Naturales Protegidas 2020.



Herramienta de gráficas de estadísticas

La herramienta de visualización de gráficas, despliega un recuadro con las tres gráficas mencionadas en el apartado 5.1.1. Cuenta con un menú donde el usuario selecciona la gráfica a mostrar o en su defecto, observar las tres. Se realizó un cambio en la representación de la gráfica de puntos de calor por mes del año 2022, pasando de barras a gráfica de dona (Figura 5.2.17).

Figura 5.2.16

Herramienta de gráficas del visualizador de puntos de calor GOES-16/ABI. Gráfica de los últimos 7 días detectados por el algoritmo a partir del 03-10-2022.

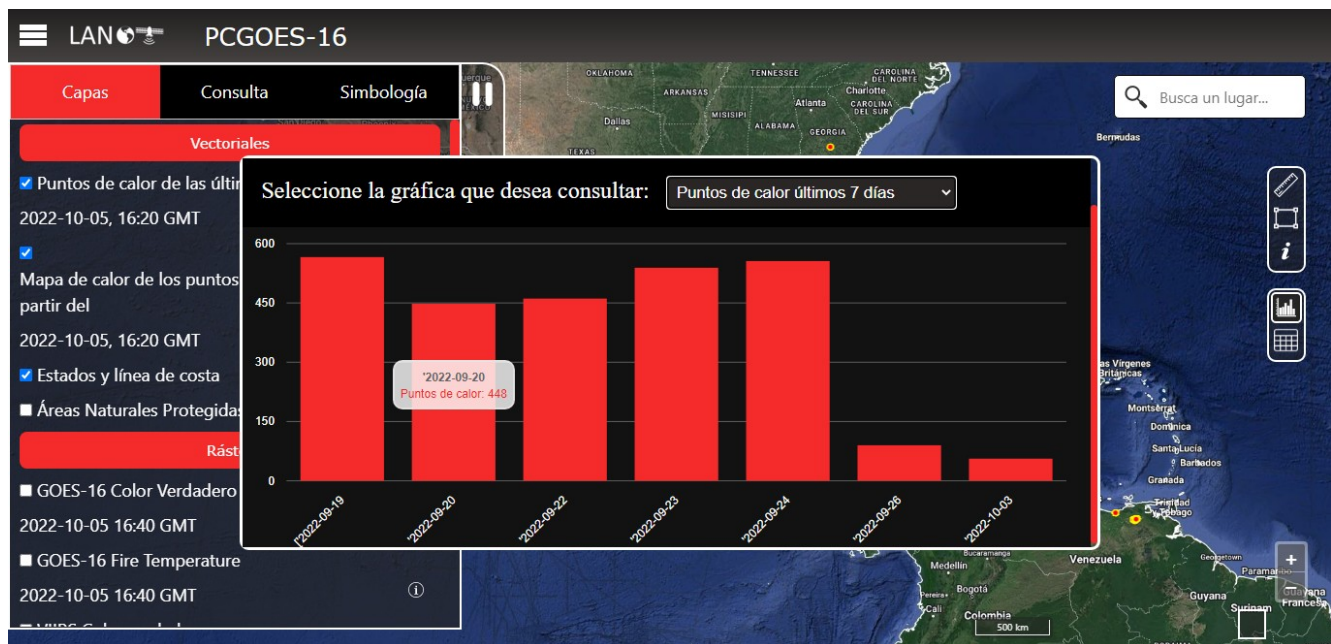


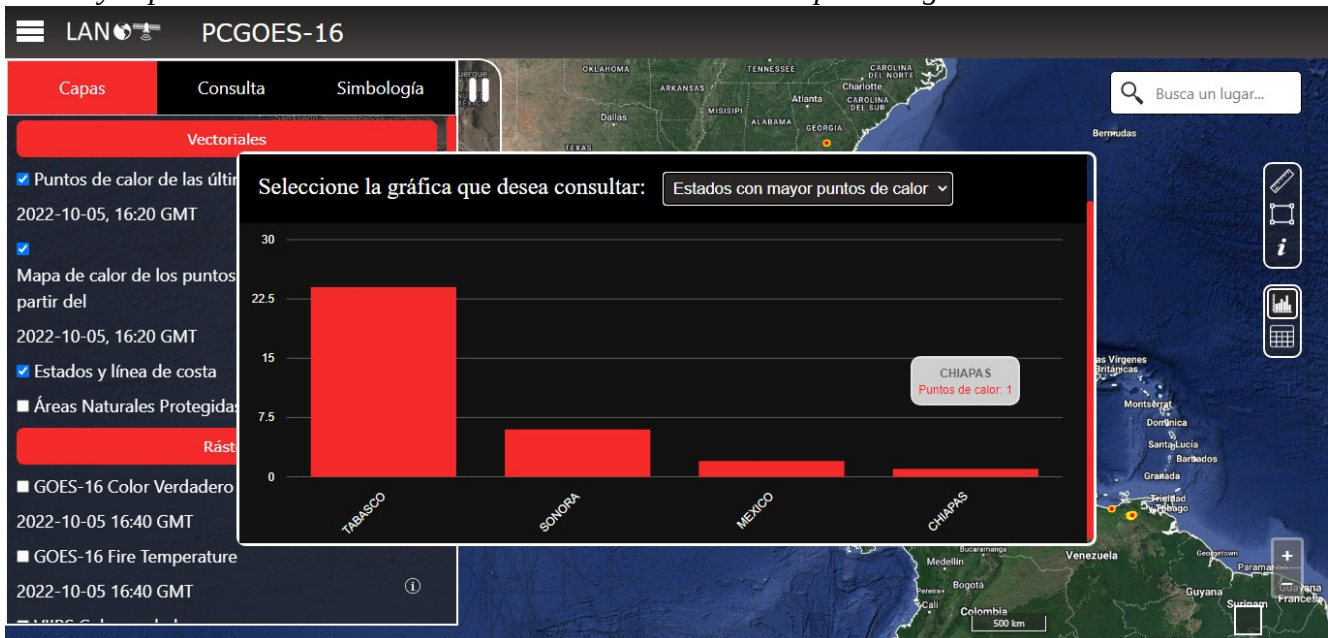
Figura 5.2.17

Herramienta de gráficas del visualizador de puntos de calor GOES-16/ABI. Gráfica de dona de los puntos de calor detectados por mes, hasta el 05-10-2022.



Figura 5.2.18

Herramienta de gráficas del visualizador de puntos de calor GOES-16/ABI. Gráfica de los estados con mayor puntos de calor de los últimos siete días detectados por el algoritmo.



Herramienta de tabla de atributos

La tabla de atributos solo se desplegará para puntos de calor (últimas 24 hrs o por consulta temporal) y será activada con el símbolo de tabla a un costado derecho de la pantalla.

Figura 5.2.19

Resultado de la herramienta de tabla de atributos en visualizador de puntos de calor GOES-16/ABI.

a)

Tabla de atributos de Puntos de calor de las últimas 24 hrs a partir del 2022-09-20, 19:00 GMT											
Id	Coordenada X	Coordenada Y	Longitud	Latitud	Satélite	B07	B14	Diferencia B07-B14	Fecha	Hora GMT	Land Cover
484588	1231468.5759415	1047098.99256214	-63.6323224091073	9.57193970313712	Goes-16	302.05	292.26	9.79	Sept. 19, 2022	4 a.m.	Grasslands

b)

Tabla de atributos de Puntos de calor del 2022-08-05/2022-08-06											
Id	Coordenada X	Coordenada Y	Longitud	Latitud	Satélite	B07	B14	Diferencia B07-B14	Fecha	Hora GMT	Land Cover
3646331	-2463939.16048832	2674360.96760607	-101.495382709769	25.9320340878414	Goes-16	325.81	292.98	32.83	2022-08-05	00:00:00	Savannas

Nota: a) Tabla de atributos de puntos de calor GOES-16/ABI últimas 24 horas a partir del 20-09-2022, 19:00 GMT.

b) Tabla de atributos de puntos de calor GOES-16/ABI del 05-08-2022 al 06-08-2022.

6 Conclusiones

Gracias a los avances tecnológicos, se ha mostrado un crecimiento en la cartografía digital, llegando al punto de generar Sistemas de Información Geográfica Web, que permiten al usuario desplegar y manipular información espacial sin necesidad de contar con un software de escritorio especializado.

En la presente tesis se desarrolló un visualizador SIG WEB donde se pueden desplegar los puntos de calor detectados en tiempo casi real por el LANOT mediante un algoritmo, a demás de poder realizar consultas temporales de los mismos. Además, permite la descarga de los mismos en formatos CSV y GeoJson.

También, despliega las capas de Áreas Naturales Protegidas (2020), mapas de calor de los puntos de calor de los últimas 24 hrs, línea de costa, estados de la república mexicana, compuestos de imágenes GOES-16 (color pseudo-verdadero, *Fire Temperature*) y color verdadero de imágenes VIIRS. Adicionalmente, proporciona al usuario diversas herramientas para conocer las características de los puntos de calor de manera individual, en grupo o región, a la vez que puede analizar algunas estadísticas de la tendencia del número de puntos de calor a nivel nacional y estatal.

El visualizador se encuentra en los servidores del Laboratorio Nacional de la Tierra del Instituto de Geografía, desde mediados del año 2022.

Anexos

A. Scripts Python

Script de Python “conectdb.py”

```
# -*- coding: utf-8 -*-
"""
-----
Este script agregará los puntos de calor detectados cada diez minutos por el
algoritmo de puntos de calor GOES-16/ABI a la tabla puntos_10m
-----
"""
import psycopg2
import os

#Creamos una función que agregará los valores a la BD provenientes de los CSV
def faltantes(n):
    path1 = path + "/" + n
    #print(path1)
    #Ejecutamos la transacción de copiar los puntos de los CSV a BD
    cur.execute("COPY puntos_10m (x,y,lon, lat, satellite, bt_c07, bt_c14, dif_c07c14, fecha, hora,
land_cover, estado, pais, anp) FROM '%s' delimiter ',' csv header" % (path1))
    #Ejecutamos la actualización de los datos espaciales con la columna geom
    cur.execute("UPDATE puntos_10m SET geom = ST_GeometryFromText('POINT (' || lon || ' ' ||
lat || ')', 4326);")
    #Ejecutamos la selección de los datos actualizados
    cur.execute("SELECT * FROM puntos_10m")

    #Devuelve todos los registros de los movimientos
    rows = cur.fetchall()

#Ciclo for para imprimir en pantalla todos los resultados
#for r in rows:
    #print(r)
    #dt = r[3]
    #n = dt.strftime("%Y%j%H%M")
```

```
#l = "incendios" + n + ".csv"

#Se registrará el archivo en el texto datos_kawak
f = open('/home/fires/scripts_db/datos_kawak.txt','a')
f.write(n+"\n")
f.close()
#Se realizan los movimientos en la base de datos
conect.commit()

#Creamos la función ciclo para iterar
def ciclo(lista, var):
    for n in lista:
        for i in var:
            if i == n:
                print("\nYa esta el archivo: " +n+ " en la base de datos.\n")
                break
            else:
                print("\nEl archivo " +n+ " aun no se encuentra en la base de datos.")
                faltantes(n)
        print("\nSe agrego el archivo: " +n+ " en la base de datos, tras no encontrarse.\n")
        continue

#Abrimos conexión con la base de datos

conect = psycopg2.connect(
    host = "localhost",
    database = "fires",
    user = "fires",
    password = "password",
    port = 5433
)

#Abrimos cursor

cur = conect.cursor()

#Ruta de los Resultados obtenidos por el Algoritmo de puntos de calor GOES-16/ABI
path = "/var/www/html/hotspots/csv_10min"
```

```
#Se organizan los archivos, tomando en cuenta solo los que se generaron en el servidor Kawak
lista = os.listdir(path)
lista.sort()
lista = lista[5911:]
#lista1 = lista[609:]
#print(lista1)

#Se abre el archivo datos_kawak y se obtiene
f= open('/home/fires/scripts_db/datos_kawak.txt','r')
var = f.read()
var = var.split("\n")
#var3 = var2[1]
#f.close()
#print(var)
f.close()

if lista[-1] == var[-2]:
    print("Termino el programa")
    print(lista[-1])
    print(var[-2])
else:
    ciclo(lista,var)
#f.close()

#cerramos cursor
cur.close()

#cerramos conexión
conect.close()
```

Script de Python “conectdb1.py”

```
# -*- coding: utf-8 -*-
"""
-----
Este script agregará los últimos puntos de calor detectados cada diez minutos por el
algoritmo de puntos de calor GOES-16/ABI a la tabla puntos_10m
-----
"""
import psycopg2
import os
import datetime
import sys

#Creamos función que agregará valores a la BD provenientes de los CSV

def actualizar(path1):
    #Ejecutamos la selección
    cur.execute("COPY puntos_10m (x,y,lon, lat, satellite, bt_c07, bt_c14, dif_c07c14, fecha, hora,
land_cover, estado, pais, anp) FROM '%s' delimiter ';' csv header" % (path1))

    #Ejecutamos la actualización de los datos espaciales en la columna geom
    cur.execute("UPDATE puntos_10m SET geom = ST_GeometryFromText('POINT (' || lon || ' ' || lat || '),
4326);")

    #Ejecutamos la selección de los datos actualizados
    cur.execute("SELECT * from puntos_10m")

    #Todos los registros de los movimientos
    row = cur.fetchall()

    #Se realizan los movimientos en la base de datos
    conect.commit()

#Abrimos conexión con la base de datos

conect = psycopg2.connect(
    host = "localhost",
```

```
database = "fires",
user = "fires",
password = "password",
port = 5433
)

#Abrimos cursor

cur = conect.cursor()

#Ruta de los Resultados obtenidos por el Algoritmo de puntos de calor GOES-16/ABI
path = "/var/www/html/hotspots/csv_10min"

#Se organizan los archivos
archivos = os.listdir(path)
archivos.sort()
archivo = archivos[-1]

#Se crea la ruta con el archivo a ingresar
path1 = path + "/" + archivo
print (path1)

#Se abre el archivo de texto que llevará los registros de los CSV ingresados a la BD
f = open('/home/fires/scripts_db/datos_kawak.txt','r')
var = f.read()
archivo2 = var.split("\n")[-2]
f.close()

print ("archivo: ",archivo)
print ("archivo2: ",archivo2)
print (len(archivo))
print (len(archivo2))
print (archivo == archivo2)
#Sentencia que determina si se actualizará la BD
if archivo != archivo2:
    try:
        actualizar(path1)
        f = open("/home/fires/scripts_db/datos_kawak.txt","a")
        f.write(archivo + "\n")
```

```
        f.close()
        print ("Se agregó archivo: "+archivo+".")
#Se realiza los movimientos en la base de datos
#conect.commit()
except Exception as e:
    print(f'Ocurrió un error en la transacción: {e}')

#Cerramos cursor
cur.close()
#Cerramos conexión
conect.close()
elif archivo == archivo2:
    print("Ya se encuentra el archivo: "+archivo+" en la base de datos.")
    #conect.commit()
    #cerramos cursor
    cur.close()

#Cerramos conexión
conect.close()
sys.exit()
```

Script de Python “conectdb_dia.py”

```
# -*- coding: utf-8 -*-
"""
-----
Este script agregará los puntos de calor detectados cada día por el
algoritmo de puntos de calor GOES-16/ABI a la tabla puntos_diacom
-----
"""
import psycopg2
import os
#import datetime

#Creamos una función que agregará los valores a la BD provenientes de los CSV
def faltantes(n):
    path1 = path + "/" + n
    print(path1)

    #Ejecutamos la transacción de copiar los puntos de los CSV a BD
    cur.execute("COPY puntos_diacom (x,y,lon, lat, satellite, bt_c07, bt_c14, dif_c07c14, fecha, hora,
land_cover, estado, pais, anp, freq, freq_norm) FROM '%s' delimiter ',' csv header" % (path1))

    #Ejecutamos la actualización de los datos espaciales con la columna geom
    cur.execute("UPDATE puntos_diacom SET geom = ST_GeometryFromText('POINT (' || lon || ' ' ||
lat || '), 4326);")

    #Ejecutamos la selección de los datos actualizados
    cur.execute("SELECT * FROM puntos_diacom")

    #Devuelve todos los registros de los movimientos
    rows = cur.fetchall()

    #Se registrará el archivo en el texto datos_kawak
    f = open('/home/fires/scripts_db/datos_kawak_dia.txt','a')
    f.write(n+"\n")
    f.close()
    #Se realizan los movimientos en la base de datos
    conect.commit()
```

#Creamos la función ciclo para iterar

```
def ciclo(lista, var):
    for n in lista:
        print(var)
        for i in var:
            if i == n:
                print("\nYa esta el archivo: " +n+ " en la base de datos.\n")
                break
            else:
                print("\nEl archivo " +n+ " aun no se encuentra en la base de datos.")
                faltantes(n)
        print("\nSe agregó el archivo: " +n+ " en la base de datos, tras no encontrarse.\n")
        continue
```

#Abrimos conexión con la base de datos

```
conect = psycopg2.connect(
    host = "localhost",
    database = "fires",
    user = "fires",
    password = "password",
    port = 5433
)
```

#Abrimos cursor

```
cur = conect.cursor()
```

#Ruta de los Resultados obtenidos por el Algoritmo de puntos de calor GOES-16/ABI

```
path = "/var/www/html/hotspots/csv_dia"
```

#Se organizan los archivos, tomando en cuenta solo los que se generaron en el servidor Kawak

```
lista = os.listdir(path)
```

```
lista.sort()
```

```
lista=lista[1:]
```

#Se abre el archivo datos_kawak y se obtiene


```
f= open('/home/fires/scripts_db/datos_kawak_dia.txt','r')
var = f.read()
var = var.split("\n")
#var3 = var2[1]
#f.close()
#print(var)
f.close()
```

```
if lista[-1] == var[-2]:
    print("Termino el programa")
    print(lista[-1])
    print(var[-2])
else:
    ciclo(lista,var)
#f.close()
```

```
#cerramos cursor
cur.close()
```

```
#cerramos conexión
conect.close()
```

Script de Python “conectdb_dia1.py”

```
# -*- coding: utf-8 -*-
"""
-----
Este script agregará los últimos puntos de calor detectados cada día por el
algoritmo de puntos de calor GOES-16/ABI a la tabla puntos_diacom
-----
"""
import psycopg2
import os
import datetime
import sys

#Creamos función que agregará valores a la BD provenientes de los CSV

def actualizar(path1):
    #Ejecutamos la selección
    cur.execute("COPY puntos_diacom (x,y,lon, lat, satellite, bt_c07, bt_c14, dif_c07c14, fecha, hora,
land_cover, estado, pais, anp, freq, freq_norm) FROM '%s' delimiter ',' csv header" % (path1))

    #Ejecutamos la actualización de los datos espaciales en la columna geom
    cur.execute("UPDATE puntos_diacom SET geom = ST_GeometryFromText('POINT (' || lon || ' ' || lat ||
)'), 4326);")

    #Ejecutamos la selección de los datos actualizados
    cur.execute("SELECT * from puntos_diacom")

    #Todos los registros de los movimientos
    row = cur.fetchall()

    #Se realizan los movimientos en la base de datos
    conect.commit()

#Abrimos conexión con la base de datos

conect = psycopg2.connect(
    host = "localhost",
```

```
database = "fires",
user = "fires",
password = "password",
port = 5433
)

#Abrimos cursor

cur = conect.cursor()

#Ruta de los Resultados obtenidos por el Algoritmo de puntos de calor GOES-16/ABI
path = "/var/www/html/hotspots/csv_dia"

#Se organizan los archivos
archivos = os.listdir(path)
archivos.sort()
archivo = archivos[-1]

#Se crea la ruta con el archivo a ingresar
path1 = path + "/" + archivo
print (path1)

#Se abre el archivo de texto que llevará los registros de los CSV ingresados a la BD
f = open('/home/fires/scripts_db/datos_kawak_dia.txt','r')
var = f.read()
archivo2 = var.split("\n")[-2]
f.close()

print ("archivo: ",archivo)
print ("archivo2: ",archivo2)
print (len(archivo))
print (len(archivo2))
print (archivo == archivo2)

#Sentencia que determina si se actualizará; la BD
if archivo != archivo2:
    try:
        actualizar(path1)
        f = open("/home/fires/scripts_db/datos_kawak_dia.txt","a")
```

```
f.write(archivo + "\n")
f.close()
print ("Se agrego archivo: "+archivo+".")
```

```
#Se realiza los movimientos en la base de datos
```

```
except Exception as e:
    print(f'Ocurrió un error en la transacción: {e}')
```

```
#Cerramos cursor
cur.close()
```

```
#Cerramos conexión
conect.close()
```

```
elif archivo == archivo2:
    print("Ya se encuentra el archivo: "+archivo+" en la base de datos.")
```

```
#cerramos cursor
cur.close()
```

```
#Cerramos conexión
conect.close()
sys.exit()
```

Script de Python “acestadisticaspc.py”

```
# -*- coding: utf-8 -*-
"""
-----
Este script actualizará las tablas relacionadas con las bases de datos de fires
con la finalidad que en la página web se puedan mostrar los datos más recientes.
-----
"""

import psycpg2

#Creamos función de conexión a la base de datos fires
def conectdb():
    #Abrimos conexión con la base de datos
    global conect
    conect = psycpg2.connect(
        host = "localhost",
        database = "fires",
        user = "fires",
        password = "password",
        port = 5433
    )

    #Abrimos cursor
    global cur
    cur = conect.cursor()

#Creamos función que obtiene las fechas de los últimos 7 días de la tabla puntos_diacom
def ultfecha():
    #Se abre el archivo datos_kawak y se obtiene
    f= open('/home/fires/scripts_db/datos_kawak_dia.txt','r')
    var = f.read()
    var = var.split("\n")
    f.close()
    var = var[-8:-1]
    global fechadate
    fechadate=[]
    for i in var:
        fecha=i[9:17]
```

```

fechad=fecha[0:4]+'-'+fecha[4:6]+'-'+fecha[6:]
fechadate.append(fechad)

```

#Creamos función para actualizar tabla de pc_sietedias

```

def actpc_sietedias():
    conectdb()
    ultfecha()
    cur.execute("truncate table pc_sietedias restart identity;")
    for i in fechadate:
        cur.execute("insert into pc_sietedias(fecha,puntos_calor) select fecha ,count(*) from puntos_diacom
where fecha = date'%s' - interval '0 day' group by fecha;" % (i))
        cur.execute("select * from pc_sietedias;")
        #Devuelve todos los registros de los movimientos
        rows = cur.fetchall()
        print(rows)
        #Se realizan los movimientos en la base de datos
        conect.commit()
        #cerramos cursor
        cur.close()

    #close the conection
    conect.close()

```

#Creamos función para actualizar tabla de estados_pc

```

def actestados_pc():
    conectdb()
    ultfecha()
    cur.execute("truncate table estados_pc;")
    fechaultima = fechadate[-1]
    cur.execute("insert into estados_pc select Estado, count(id) from puntos_diacom where fecha between
date'%s' - interval '7 day' and date'%s' group by Estado order by count desc FETCH FIRST 11 ROWS
ONLY;" % (fehaultima, fechaultima))
    cur.execute("select * from estados_pc;")
    #Devuelve todos los registros de los movimientos
    rows = cur.fetchall()
    print(rows)
    #Se realizan los movimientos en la base de datos
    conect.commit()
    #cerramos cursor

```

```
cur.close()

#close the conection
conect.close()

#Creamos función para actualizar tabla de pc_mes
def actpc_mes():
    conectdb()
    ultfecha()
    i = 1
    while i < 13:
        cur.execute("select count(*) from puntos_diacom where date_part('month', fecha)='%s' and
date_part('year', fecha)= 2022" % (i))
        pc_calor=cur.fetchone()
        # import pdb; pdb.set_trace()
        cur.execute("UPDATE pc_mes SET puntos_calor= %s where id = '%s'" % (pc_calor[0], i))
        i = i+1
    cur.execute("select * from estados_pc;")
    #Devuelve todos los registros de los movimientos
    rows = cur.fetchall()
    print(rows)
    #Se realizan los movimientos en la base de datos
    conect.commit()
    #cerramos cursor
    cur.close()

#close the conection
conect.close()

#Ejecutamos funciones
actpc_sietedias()
actestados_pc()
actpc_mes()
```

B. Scripts Shell

Script en Shell “cp_last_csv_KAWAK_CUMULUS.sh”

```
server="fires@IPservidor2"  
dirIn="/home/fires/resultados/csv_10min/"  
dirOut="/home/fires/Resultados/"
```

```
NC=`ssh $server 'ls -t '$dirIn' | head -1`
```

```
echo NC
```

```
scp $server:$dirIn$NC $dirOut
```


Referencias

- Alesheikh, A., Helali, H., & Behroz, H. (2002). *Web GIS: Technologies and Its Applications*.
<https://www.isprs.org/proceedings/XXXIV/part4/pdfpapers/422.pdf>
- Apache Software Foundation. (s.f.). *Apache HTTP Server Project*.
https://httpd.apache.org/ABOUT_APACHE.html
- Comisión Nacional para el Conocimiento y Uso de la Biodiversidad [CONABIO]. (2022). *Puntos de calor detectados con imágenes de satélite 08-10-2020*. <http://incendios1.conabio.gob.mx/08-09-2020>
- Comisión Nacional Forestal [CONAFOR]. (2019). *Programa de manejo del fuego Centro nacional de manejo del fuego. Cierre de la temporada 2019*.
https://www.gob.mx/cms/uploads/attachment/file/522446/Cierre_de_la_Temporada_2019.pdf
- Dinahosting. (s.f.). *¿Qué es Apache y para qué sirve?*. <https://dinahosting.com/ayuda/que-es-apache-y-para-que-sirve/>
- Eguíluz, J. (2009). *Introducción a JavaScript*. <http://www.librosweb.es/javascript>
- FIRMS. (s. f.). *Fire Information for Resource Management System*.
<https://firms.modaps.eosdis.nasa.gov/>
- Fu, P. (2018). *Getting to know WEB GIS*. Esri Press.
- Fu, P., & Sun, J. (2011). *Web GIS: Principles and Applications*. ESRI Press.
- Gauchat, J. (2012). *El gran libro de HTML5, CSS3 y Javascript (Primera)*. Marcombo.
- Geoserver. (s.f.). *Styles*. <https://docs.geoserver.org/stable/en/user/styling/webadmin/index.html#styles-page>
- Gómez Rubio, C. (2022). *Implementación y programación de un algoritmo de detección de incendios forestales para México utilizando imágenes satelitales GOES16/ABI con software libre [Tesis de licenciatura, UNAM]*. Tesis UNAM.
- INPE, & DPI. (2018). *Manual do Usuário TERRAMA² Monitorio, Análise e Alerta Versão 4.0.6*.
http://www.terrama2.dpi.inpe.br/wp-content/uploads/2018/08/Manual_Usuario_TERRAMA2_v4_Cap1e2.pdf
- JQuery (s.f.). *What is JQuery?*. <https://jquery.com/>

- Li, S., Dragičević, S., & Veenendaal, B. (2011). *Advances in Web-based GIS, Mapping Services and Applications*. CRC Press.
- Li, Z., Chen, J., & Baltsavias, E. (Eds.). *Web Mapping/GIS services and applications*. En Li, Z., Chen, J., & Baltsavias, E. (Eds.), *Advances in photogrammetry remote sensing and spatial information science: 2008 ISPRS congress book* (pp. 335-354). CRC Press.
- Manzo-Delgado L.L., Morelli F, Victorino P., Gomez-Rubio C., Mendoza-Castillo U.J., y Aguilar-Sierra A. (2022). *Detection of vegetation fires in Mexico using GOES-16/ABI images: algorithm description and preliminary assessment*. *International Journal of Remote Sensing*. ISSN. En revisión.
- Mappinggis. (s.f.). *Geoserver para novatos*. <https://mappinggis.com/2022/06/geoserver-para-novatos/>
- Mozilla Foundation. (s.f.). *Introducción a Django*. <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>
- OGC. (s.f.). *OGC Web Services Context Document (OWS Context)*. <https://www.ogc.org/standards/ows>
- OpenLayers. (s.f.). *Ol/Sphere*. https://openlayers.org/en/latest/apidoc/module-ol_sphere.html
- OSGeoLive. (s.f.). *Openlayers*. https://live.osgeo.org/es/overview/openlayers_overview.html
- OSGeoLive. (s.f.). *Inicio Rápido de PostGIS*. https://live.osgeo.org/es/quickstart/postgis_quickstart.html
- Puig, J. C. (2013). *CSS3 y Javascript avanzado*. Universitat Oberta de Catalunya.
- Python Software Foundation. (s.f.). *psycopg2 2.9.3*. <https://pypi.org/project/psycopg2/>
- Python Software Foundation. (s.f.). *os-Micellaneous operating system interfaces*. <https://docs.python.org/3/library/os.html>
- Rodríguez-Trejo, D. A. (2014). *Incendios de vegetación: su ecología, manejo e historia*. Vol 1. Colegio de Postgraduados.
- Python Software Foundation. (s.f.). *sys-System-specific parameters and functions*. <https://docs.python.org/3/library/sys.html>
- Sastre, P. (2010). *Sistemas de Información Geográfica (SIG) Técnicas básicas para estudios de biodiversidad*. Instituto Geológico y Minero de España. <https://geoinnova.org/wp-content/uploads/2018/08/Sistemas-de-Informaci%C3%B3n-Geogr%C3%A1fica-SIG-T%C3%A9cnicas-b%C3%A1sicas-para-estudios-de-biodiversidad.pdf>

-
- Tabarés, R. (2012). *El inicio de la Web: Historia y cronología del hipertexto hasta HTML 4.0 (1990-99)*. Ediciones Universidad de Salamanca (España), Vol5 (n° 1), 57-82.
- Vega Nieva, D. J. Nava-Miranda M.G, CallerosFlores E., López-Serrano P.M., Briseño-Reyes J., Flores Medina F., López-Sánchez C., Corral-Rivas J.J., González-Cabán A., Alvarado-Celestino E., Cruz I., Cuahutle M., Ress R., Setzer A., Morelli F., Pérez-Salicrup D., Jardel-Pelaez E., Cortes-Montaña C., Vega J.A. & Jiménez E. (2019). *Desarrollo de un Sistema Nacional de Peligro de Incendios Forestales Para México*. Memorias del quinto simposio internacional sobre políticas, planificación y economía de los incendios forestales: servicios ambientales e incendios forestales. Gen. Tech. Rep. PSW-GTR-261, (Español). Albany, CA: Departamento de Agricultura de los EE.UU., Servicio Forestal, Estación de Investigación del Pacífico Suroeste: 42-52.
https://www.fs.usda.gov/psw/publications/documents/psw_gtr261esp/psw_gtr261esp_042.pdf
- Yanza, A. (2013). Lineamientos para el desarrollo de aplicaciones SIG WEB [Tesis de Magíster, Universidad EAFIT Escuela de Ingeniería].
https://repository.eafit.edu.co/bitstream/handle/10784/1280/yanzahurtado_andreaviviana_2013.pdf?sequence=1