

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Estudio del calor superficial en la Ciudad de México, utilizando técnicas de percepción remota y Sistemas de Información Geográfica

TESIS

Que para obtener el título de

Ingeniero Geomático

P R E S E N T A

Edgar Pascual Zavaleta Martínez

DIRECTORA DE TESIS

Dra. Griselda Berenice Hernández Cruz



Ciudad Universitaria, Cd. Mx., 2026

PROTESTA UNIVERSITARIA DE INTEGRIDAD Y HONESTIDAD ACADÉMICA Y PROFESIONAL
(Titulación con trabajo escrito)

De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado ESTUDIO DEL CALOR SUPERFICIAL EN LA CIUDAD DE MEXICO, UTILIZANDO TECNICAS DE PERCEPCION REMOTA Y SISTEMAS DE INFORMACION GEOGRAFICA que presenté para obtener el titulo de INGENIERO GEOMÁTICO es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Entidad Académica, citando las fuentes de ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia, acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación.

Edgar Pascual Zavaleta Martinez

EDGAR PASCUAL ZAVALETA MARTINEZ
Número de cuenta: 314150938

Para los curiosos por este trabajo, las personas que me han ofrecido su tiempo y apoyo, pero sobre todo a las personas en las cuales vivo.

Agradecimientos

If I have seen further, it is by standing upon the shoulders of giants ...

Isaac Newton probablemente citando a Juan de Salisbury

Quisiera expresar mi sincero agradecimiento a...

A la Doctora Griselda Berenice Hernández Cruz, no puede dimensionar lo mucho que agradezco al no solo haberme invitado a a participar en el proyecto PAPIIT IT101621 cuyo fruto es este trabajo; sino también por todos las experiencias, aprendizajes, apoyos y sobre todo la confianza al guiarme en los misterios de la investigación por medio de su sabiduría y sobre todo mucha paciencia hasta con los trámites de titulación. Usted es parte fundamental de este trabajo al motivarme a no desertar por las palabras que nos dio en una de las pláticas-regaños de la clase de Fotogrametría. Gracias.

Difícilmente yo hubiesa estudiado algo relacionado con las geociencias, de no ser por culpa del Doctor Juan Martín Gómez González, por medio del "Taller de Ciencia para Jóvenes, Proyecto PAPIME-PE100915: Cimiento de Investigadores. Generación 2017". Ese taller me motivo a plantear nuevos desafíos a mi vida, a continuar en aras de la ciencia (claro, desde la Ingeniería) por medio del intercambio entre investigadores/alumnos. Desde el momento en que ingresé al bachillerato de esta universidad me sentí agradecido y al mismo tiempo en deuda con ella. No tenía idea de que podía por ella hacer hasta que escuché sus palabras en el taller: "la Universidad crece así como sus alumnos al realizar trabajos que aportan al conocimiento y la difusión del mismo".

A mis compañeros Pepe y Erick. Me es díficil pensar en esta etapa sin ustedes, la compañia en cada desvelo, proyecto pero sobre todo en gratificación de nuestro esfuerzo, en el crecimiento de cada uno de nosotros, fuimos cómplices y respaldo en esta etapa. Así como a

Beto, Jorge, Miguel y Fernandita que hemos continuado creciendo con amistad.

Elsa, Pale, Sandra, Diego, Reyes, Cecilia, Brocoli y Newton. Nunca habrá las suficientes palabras para cada uno de ustedes por todo lo hecho por mi, por el apoyo, por lo fundamental que han sido y el refugio que llamamos familia. Gracias por su amor, paciencia y cada sacrificio que se ha hecho para que yo pudiera alcanzar esta meta. Este logro es más suyo que mío.

Al proyecto PAPIIT con clave IT101621 “Estudio de la temperatura superficial terrestre (LST), en zonas urbanas y suburbanas de la Ciudad de México mediante técnicas de percepción remota y prototipos de bajo costo.” y a Fundación UNAM en colaboración Grupo BAL por su apoyo durante esta etapa. Y por supuesto a las personas que hicieron, hacen y harán enorme a la UNAM.

Gracias a cada uno de ustedes, así como personas e instituciones que haya olvidado mencionar pero fueron influyentes en mi formación y desarrollo.

Índice

Índice de figuras IX

Índice de tablas XI

1. Anteproyecto 1

2. Marco Conceptual 9
2.1. Temperatura Superficial e Islas de Calor 10
2.2. Percepción Remota 14
2.3. Geo Big Data 16

3. Estimación del calor superficial en la Ciudad de México y espacios conurbados 21
3.1. Características físico-geográficas de los sitios urbanos y semi-urbanos . . . 22
3.2. Estimación del calor superficial 25
3.3. Características de los datos geoespaciales 30
3.4. Procesamiento de imágenes Landsat 33

4. Resultados y Discusión 37
4.1. Caracterización del conjunto de datos 38
4.2. Comportamiento temporal de la LST 39
4.3. Distribución espacial de la temperatura superficial 41

5. Conclusiones 59

Bibliografía 65

Apéndice A. Imágenes obtenidas 71

Apéndice B. Scripts de Google Earth Engine 117

Índice de figuras

2.1. Tipos de Islas de Calor 12
2.2. Espectro electromagnético 15
2.3. Editor de código de Google Earth Engine 19

3.1. Estructura básica del algoritmo para cálculo de LST 31
3.2. Estructura básica del algoritmo SW para cálculo de LST 32
3.3. Extensión de las imágenes Landsat que intersecan con la Ciudad de México y su zona conurbana 33
3.4. Mosaico que contempla la extensión de la Ciudad de México y su zona conurbana 34
3.5. Imagen compuesta RGB, recortada con la Ciudad de México y su zona conurbana 35

4.1. Valor medio de LST por imagen calculada 38
4.2. Valor medio de LST por imagen calculada, con una nubosidad inferior al 80 en la zona de estudio 39
4.3. Diagrama de caja del promedio mensual del LST 39
4.4. Gráfico de barras del valor medio del LST de las imágenes calculadas . . . 40
4.5. Distribución espacial de la temperatura superficial media (LST, K) en la Ciudad de México y zona conurbana 42
4.6. Distribución espacial clasificada de la temperatura superficial media 45
4.7. Mapa para representar las diferentes zonas bajo las cuales se ha dividido el área de estudio. 47
4.8. Mapa que representa los diversos puntos de interés que existen en la zona centro 49
4.9. Mapa que representa los diversos puntos de interés que existen en la zona oriente 52

4.10. Mapa que representa los diversos puntos de interés que existen en la zona oriente 54
4.11. Mapa que representa los diversos puntos de interés que existen en la zona poniente 56
4.12. Mapa que representa los diversos puntos de interés que existen en la zona norte 58

Índice de tablas

2.1. Características de la Percepción Remota 18

3.1. Información respecto a las bandas utilizadas. 26

Capítulo 1

Anteproyecto

El objetivo del presente capítulo es mostrar al lector el bosquejo metodológico y la planificación inicial del presente trabajo, mediante la presentación de una propuesta inicial para la estimación del cálculo de la temperatura de superficie utilizando sensores remotos. De manera sintetizada, se abordan los temas que más adelante se desarrollan como capítulos completos.

Las Islas de Calor Urbanas, "Urban Heat Islands" (UHI por sus siglas en inglés e ICU en español), son aquellas donde hay una mayor temperatura y una mayor urbanización en comparación de las zonas a sus alrededores donde no hay un alto nivel de urbanización. La zona de la Ciudad de México y su área metropolitana no es ajena a este fenómeno y para muestra clara, se podrían mencionar las contingencias ambientales en la Ciudad de México, donde al aumentar la temperatura, se concentra una mayor tasa de reacciones químicas.

Las causas y orígenes de las ICU se han estudiado desde algún tiempo atrás. Realizándose la primera investigación en el siglo XIX por Luke Howard en 1818 [40]. En México es hasta 1895, donde Manuel Moreno y Anda, al estar trabajando en el Servicio Meteorológico Nacional, señala los contrastes de temperatura entre el centro de la Ciudad de México (Palacio Nacional) y el Observatorio de Tacubaya que en su momento era considerado uso de suelo rural [24].

En 2009, un estudio elaborado por [1], apunta que la isla de calor urbana abarcaba la gran mayoría del área, con un centro cálido que varía en función de la época del año, estableciéndose en horarios nocturnos y diurnos.

Existen dos metodologías para el estudio de las ICU. Por un lado, es el método directo (medición en sitio) que se basa en usar como referencia los datos proporcionados por las estaciones climatológicas cercanas realizando métodos de interpolación, mientras por el otro es el uso de la Percepción Remota para valores continuos de temperatura a lo largo del territorio.

Para el primer caso, que es un método que se basa en encontrar las diferencias que hay en la temperatura de la atmósfera, donde interviene la interacción de diversos elementos que existan en la superficie terrestre (el tipo y cobertura del suelo) y en el aire (masas de aire y el vapor de agua). Este método tiene la ventaja de poder realizar mediciones constantes, su desventaja termina siendo que debido a la extensión y diversidad de los materiales que hay en las ciudades resulta complicado obtener resultados precisos.

En el segundo caso, el método recae en el apoyo de la Percepción Remota que al usar imágenes permite obtener la temperatura superficial de toda una "escena" a lo largo de grandes extensiones de territorio; sin embargo, su mayor desventaja recae en que estas observaciones dependerán de las propiedades térmicas del sensor, las resoluciones espaciales y temporales y condiciones del clima que faciliten esta tarea.

Lo ideal para ciudades de gran extensión sería usar el segundo caso por la gran ventaja que da el poder apreciar una mayor cantidad de superficies que coexisten en un sistema urbano. Es por eso que en el presente trabajo se estudiarán las ICU a través de herramientas tecnológicas como método de estudio.

Planteamiento del Problema

Para poder estudiar las ICU se necesita obtener la Temperatura de Superficie Terrestre (LST, Land Surface Temperature por sus siglas en inglés). La LST es una estimación derivada de la medición del nivel de radiación térmica infrarroja que se capta por medio de satélites y es una técnica de Percepción Remota. Su objetivo es tener una estimación que depende de factores como la superficie en la cual incide la radiación, el tipo de suelo y las condiciones atmosféricas.
Es importante el monitoreo de la superficie terrestre debido a que el calor al elevarse afecta patrones meteorológicos y climáticos. Además, es afectada por los gases de efecto invernadero y repercute directamente a los glaciares, las capas de hielo, el permafrost y la vegetación en los ecosistemas de la Tierra [37].
Considerando que la distribución de la población es mayor en áreas urbanas, es fundamental su estudio para apoyar en el desarrollo urbano así como diferentes estrategias que propongan mejorar la calidad de vida de sus habitantes. Por lo anterior, es necesario encontrar una metodología, de bajo costo, que permita el monitoreo de la temperatura en las diferentes zonas que integran una zona metropolitana.

Justificación y relevancia del proyecto

Este proyecto busca contribuir al cumplimiento del Objetivo de Desarrollo Sostenible (ODS) trece: Acción por el Clima, propuesto en el Programa de las Naciones Unidas para el Desarrollo Sostenible aprobado en 2015, donde nuestro país tuvo participación.

Conforme a las necesidades de nuestro país, se creó el Comité Técnico Especializado de los Objetivos de Desarrollo Sostenible de México, que busca especialistas técnicos capaces de desarrollar metodologías efectivas en el tiempo y que sean de bajo costo. En este contexto, la Ingeniería Geomática emerge como un campo crucial, pues coexiste con la información espacial y, entre sus numerosas aplicaciones, incluye el análisis de estos datos. Esto no solo facilita una mejor toma de decisiones, sino que también se alinea perfectamente con los objetivos del comité al ofrecer herramientas y enfoques que contribuyen significativamente

a la consecución de metodologías que aporten a los Objetivos de Desarrollo Sostenible en México.
Esta tesis buscará aplicar técnicas de Percepción Remota a imágenes satelitales usando métodos de bajo costo, permitiendo un método de medición indirecta para después elaborar mapas temáticos que servirán para poder analizar los núcleos urbanos donde existen diferentes temperaturas.

Objetivos

General

Determinar las variaciones de temperatura que se pueden encontrar en la superficie de la Ciudad de México y su área metropolitana, e identificar sitios con un diferencial de temperatura mediante imágenes de satélite.

Particulares

- *Desarrollar una metodología que permita el procesamiento de imágenes satelitales para medir la temperatura superficial de la cobertura terrestre.*
- *Generación de mapas temáticos de temperatura superficial que permitan la caracterización de patrones térmicos espacio temporales en la Zona Metropolitana de la Ciudad de México.*

Hipótesis o preguntas de investigación.

Por medio de herramientas tecnológicas gratuitas y de bajo costo, es posible la identificación de los cambios de temperatura en diferentes coberturas terrestres en la Ciudad de México y su zona metropolitana mediante la aplicación de algoritmos a imágenes satelitales, obteniendo así, metodologías eficientes que permiten el monitoreo continuo de los sitios densamente poblados en el país.

Criterios para la estimación de la temperatura de superficie

Identificación de sitios

Las ICU son creadas cuando se sustituye la superficie de vegetación natural y se reemplaza por superficies impermeables y no reflectantes que terminan absorbiendo un alto porcentaje de la radiación solar, llevando a la formación de "puntos calientes" locales. [45]
Para la delimitación del área de estudio se apoyará en el trabajo [22] que define como 'Zona Metropolitana Ciudad de México', con la clave de metrópoli *09.1.01* con 63 municipios, 50 centrales aquellos donde se ubica la localidad urbana o conurbación principa y 13 exteriores, municipios contiguos a los centrales, cuyas localidades urbanas no están conurbadas a la centralidad de una metrópoli y mantienen un alto grado de integración física y laboral con los municipios centrales. Estos son las 16 alcaldías de la Ciudad de México, 45 municipios del Estado de México y 2 municipios de Hidalgo.
Se descargará los 3 archivos vectorizados georreferenciados estatales en formato shape desde INEGI, debido a que ya están delimitados de manera oficial las entidades municipales. Posteriormente, se generará un nuevo archivo en formato shape que integra ya en un área única la región de estudio.

Pre-Procesamiento de imágenes

Con el área de interés definida que es la ciudad de México y su zona metropolitana, se trabajará con Google Earth Engine. Se trabajará con las colecciones de imágenes disponibles de manera gratuita del satélite Landsat 8. Primeramente, se consulta la Colección 2 de Landsat 8 con reflectancia calibrada de grado 1 en la parte superior de la atmósfera (TOA). Que son imágenes con la más alta calidad de datos disponibles, al ser datos consistentes e intercalibrados, con datos procesados de Terreno de Precisión de Nivel 1 (L1TP). De esta colección se ocupan las bandas 10 y 11 que corresponden a imágenes TIR (Infrarrojo Térmico remuestreadas de 100 metros a 30 m) Posteriormente se consulta la Colección 2 de Landsat 8 de nivel 2 grado 1, de aquí se obtienen los productos de Reflectancia de Superficie (SR) de Landsat 8. Estos datos ya han sido corregidos atmosféricamente. De esta colección se ocuparán las siguientes bandas:

- QA (Control de calidad), para la detección y enmascaramiento de píxeles con baja calidad radiométrica, presencia de nubes, sombras y vapor atmosférico, con el objetivo de reducir errores durante el procesamiento y análisis de las imágenes Landsat.

- B3, B4 y B5 de Landsat para el cálculo de índices de vegetación, debido a su respuesta espectral con las regiones roja e infrarroja cercana del espectro electromagnético.

Finalmente, se consulta la colección de Vapor de Agua por parte de los Centros Nacionales de Predicción Ambiental (NCEP, antes "NMC") y el Centro Nacional de Investigación Atmosférica (NCAR). Esta colección brinda datos con una resolución temporal de 6 horas (0000, 0600, 1200 y 1800 UTC) y una resolución espacial de 2.5 grados. El objetivo es apoyo en el control y corrección de efectos atmosféricos, para simular las condiciones atmosféricas al momento de la captura de las imágenes, Con estos datos se podrán ya realizar la corrección radiométrica, corrección geométrica, corrección atmosférica y realces espectrales a cada imagen suministrada. Esto es con el fin de quitar los efectos atmosféricos, ruidos y crear máscaras que permitan una mejor clasificación de las coberturas terrestres.

Procesamiento de imágenes

Se implementa un algoritmo que obtiene la temperatura superficial, usando las imágenes TIRS (Thermal Infrared Sensor) y el algoritmo "Split Window" propuesto por [19] y que en el capítulo de metodología se desarrollará con detalle. Se utilizará el mapa de coberturas de suelo con los índices de vegetación propuesto en el paso anterior, para determinar la variable de emisividad por cobertura para alimentar al modelo [3] [52]. La variable de vapor de agua, necesaria para el algoritmo, se tomará el valor del vapor de agua que coincida con el área de estudio, si existe más de un pixel dentro del área de estudio se tomará el promedio de todos los pixeles dentro del área.

Adquisición de las imágenes

Considerando que la resolución temporal [1] es de 16 días, podemos obtener una imagen por mes. Sin embargo, el uso de una imagen está bajo las condiciones atmosféricas al momento del paso del satélite. Una zona no debe tener una cobertura de nubosidad mayor al 20 por ciento.
Se creará una colección con las imágenes recopiladas durante el periodo comprendido entre 1 de abril de 2013 al 30 de abril del 2023, que son 10 años de estudio. De este conjunto de datos, se calculará la temperatura media de cada imagen, con el objetivo de analizar los cambios de temperatura asociados a las Islas de Calor Urbano.
De la colección se promediarán las imágenes cuya cobertura de nubosidad sea inferior al

[1] tiempo para que el satélite pase por el mismo lugar

20 %, y se obtendrá una imagen media que refleje el comportamiento de la zona de estudio. El uso del promedio para el periodo previamente establecido tiene el objetivo de reducir posibles sesgos relacionados con la variabilidad en la cantidad de imágenes disponibles entre distintos años, además permite reducir los efectos atmosféricos y temporales particulares de cada escena. De esta manera el análisis se enfoca en obtener la caracterización de patrones térmicos espaciales que existen en el área de estudio.

Cartografía Temática

Se apoyará en una serie de tiempo basada en imágenes satelitales recopiladas durante el periodo establecido anteriormente. Una vez recopiladas las imágenes, se calculará la temperatura media de cada una, donde se reflejará el comportamiento térmico de la zona de estudio. Posteriormente, se identificarán los puntos con las temperaturas más altas y más bajas para un análisis detallado de las variaciones térmicas a lo largo del tiempo.

Capítulo 2

Marco Conceptual

Para el desarrollo del presente trabajo, se enfatiza la necesidad de comprender de manera integral los siguientes conceptos. Estos conceptos actúan como pilares esenciales sobre los cuales se construirá el marco teórico y metodológico de la investigación. En primer lugar, se abordará el concepto de Islas de Calor, el cual proporciona una base sólida para comprender el contexto y los fundamentos del tema de estudio. Seguidamente, se explorará el significado y la aplicación de la Percepción Remota, destacando su relevancia en relación con los objetivos planteados. Además, se apoyará en el Geo Big Data, cuya comprensión resulta crucial para la formulación de hipótesis y la interpretación de los resultados obtenidos. Mediante una comprensión profunda y contextualizada de estos conceptos, se busca establecer un marco conceptual robusto que sirva como guía para el desarrollo y la conclusión exitosa de esta investigación.

2.1. Temperatura Superficial e Islas de Calor

Luke Howard, farmacéutico más conocido por sus aportaciones al campo de la meteorología, se dio cuenta de que en Londres, a inicio del siglo XIX, sucedía un fenómeno peculiar al existir diferencias entre los datos de temperatura que él había registrado en la periferia contra aquellos proporcionados por la Royal Society para la ciudad. Asume que esta diferencia es el calor generado artificialmente en la ciudad por las diferencias de población, las estructuras y la quema de combustible para las industrias [36].

Si bien no está del todo correcto, con esto define en términos prácticos lo que es una isla de calor: la diferencia que hay entre la temperatura de una ciudad y sus alrededores. Este concepto toma el nombre de la similitud que existe en el patrón de las isolíneas, por un lado, la isohipsa en una isla y por otro la isoterma en las ciudades. Incluso en la terminología se toman prestados para describir la forma en la cual se comporta. Siendo la frontera que es el lugar donde existe el cambio un “acantilado”, gran parte del terreno tiene un comportamiento similar recibiendo el nombre de “meseta”, y el área que aglomera una mayor densidad de construcciones (tiende a ser el núcleo comercial de una ciudad) el “pico”, mientras que áreas con una densidad considerable o con edificios altos son llamados “colinas”. Finalmente, lugares donde hay poco o nulo desarrollo urbano, presencia de grandes áreas verdes y cuerpos de agua reciben el nombre de “valles”. Toda la morfología que pueda presentar una isla de calor, está estrechamente relacionada con el grado de construcción urbana y la actividad, teniendo mayor auge cuando no hay mucho viento que pueda distribuir y mezclar el calor [40].

El origen de una isla de calor parte del balance energético aplicado a las zonas urbanas. Esto quiere decir que en este intercambio de energía entre la superficie y la atmósfera terminan

influyendo la infraestructura y los materiales que las componen, la modificación del tipo de suelo y la actividad humana que existe. Este balance de energía es definido por [38] de la siguiente manera:

$$Q^* + Q_F = Q_H + Q_E + \Delta Q_S + \Delta Q_A \ (Wm^{-2}) \tag{2.1}$$

Con Q^* como la radiación neta de ondas (la diferencia entre radiación emitida y absorbida), Q_F la densidad de flujo antropogénico (el calor emitido por las actividades humanas), Q_H la densidad de flujo de calor latente (las diferencias de temperatura entre la superficie y la atmósfera), Q_E densidad de flujo de calor sensible (la consecuencia del transporte del vapor de agua y el calor al acercarse y alejarse de la superficie), ΔQ_S el cambio en el almacenamiento neto de calor (depende de la construcción de materiales, árboles, suelo y el aire), ΔQ_A la energía neta movida por el viento (puede ser positiva o negativa). Finalmente, (Wm^{-2}) es el porcentaje de radiación que cualquier superficie refleja respecto a la radiación que incide sobre ella. Las superficies claras tienen valores de albedo superiores a las oscuras, y las brillantes más que las mates. El albedo medio de la Tierra es aproximadamente 0.3.
Como se puede observar, es un fenómeno complejo, donde la morfología de la superficie genera un albedo [1] más bajo al recibir diversos reflejos de la radiación en las calles, generando un efecto similar al que sucede en un cañón afectando a las calles y de esa manera a la atmósfera urbana. Donde las áreas urbanas están más contaminadas y suelen ser más cálidas en comparación con las áreas rurales.
Las islas de calor se representan en un espectro temporal y espacialmente dinámico, por lo mismo, se sugiere que exista una clasificación con base en la información temporal y espacial que se esté utilizando. La siguiente es una clasificación sencilla sugerida por [39], siendo la figura 2.1 el apoyo gráfico:

Subsuperficial (UHI_{Sub}) Las que se dan en el suelo debajo de la ciudad.

Superficial (UHI_{Sup}) Aquellas donde hay un intercambio directo entre la atmósfera y los materiales sólidos.

Capa de dosel urbana (UHI_{UCL}) Es relativa respecto a la superficie del suelo y el nivel de techo. Recibe su nombre de la similitud con el dosel arbóreo, donde las copas de los árboles cubren y protegen el suelo.

Capa Límite (UHI_{UBL}) Sucede entre el límite de la troposfera y la capa de dosel urbana.

[1] Proviene del latín y quiere decir "blancura". El albedo de una superficie es la fracción de la luz solar incidente que refleja la superficie. Las superficies claras, como la nieve, tienen valores superiores respecto a superficies oscuras, océanos o bosques. No es una propiedad intrínseca, y depende de las distribuciones espectrales y angulares de la luz incidente, que a su vez se rigen por la composición atmosférica y la dirección del haz de luz del sol. [5].

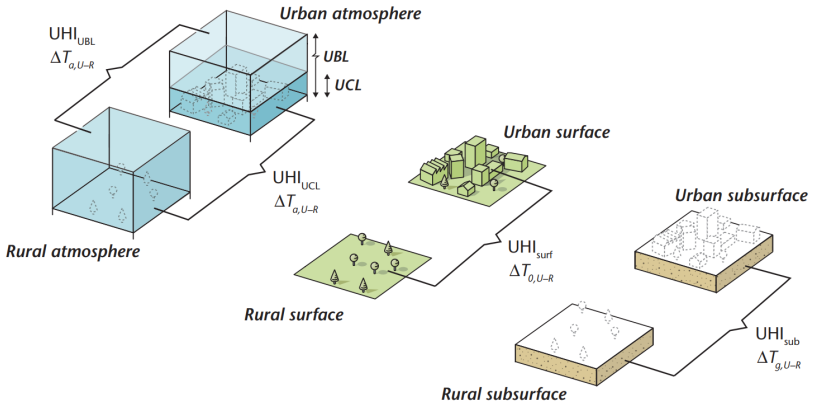


Figura 2.1 En esta imagen se pueden apreciar la posición de las diversas islas de calor que se forman en las zonas urbanas respecto a las zonas rurales: la Urban Boundary Layer (UBL, capa límite urbana) contempla la capa de aire más cercana al suelo en áreas urbanas, influenciada por la actividad humana y las estructuras urbanas; la Urban Canopy Layer (UCL, capa del dosel urbano) comprende las características físicas y biológicas de las áreas urbanas, como edificios y árboles; la superficial se refiere a elementos como pavimentos, edificios, y áreas verdes, que tienen un impacto directo en la absorción y liberación de calor; y la subsuperficial que incluyen el suelo y las capas subterráneas, que pueden afectar la transferencia de calor y la retención de agua en el subsuelo.

Caso de Estudio: Isla de Calor Superficial

Como se ha puntualizado, cada faceta urbana (calle, muros verticales, techos, árboles, etc.) al tener superficie y contacto con el ambiente produce una temperatura superficial, T_O satisfaciendo las combinaciones de radiación, conducción y turbulencias que se lleguen a presentar. Esto es la combinación de un gradiente de temperatura Q_H que genera una densidad de flujo de calor latente liberándose en la atmósfera y en manera similar hay un gradiente de temperatura Q_G, pero este se dirige al sustrato. La ecuación 2.2 representa el balance de energía para una superficie en función del cambio de la T_O en el tiempo, usando una capa de grosor z con una capacidad calorífica C.

$$C\frac{\partial T_0}{\partial t}z = Q* - Q_H - Q_E - Q_G \ (Wm^{-2}) \tag{2.2}$$

Con la ecuación 2.2 se denota la relación que hay en el balance de energía en propiedades térmicas de la materia afectando a T_0. Son cinco propiedades de las superficies que fuertemente toman control en la ecuación 2.2, repercutiendo en T_0. Estas propiedades son: geométricas, radiativas, térmicas, humedad y las aerodinámicas. Que estas propiedades tengan variaciones es la razón de que podamos encontrar diferentes temperaturas superficiales en comparación

con la temperatura del aire, particularmente durante el día.

Propiedades de la superficie

Geométricas Contemplan su orientación (ángulo de pendiente y azimut) en relación con la incidencia del Sol en variación de la apertura del cielo. Por el día se encontrarán mayores temperaturas en donde exista una mayor irradiancia, esto es, lugares donde no exista sombra o donde el ángulo zenital es corto, mientras las T_0 más bajas los veremos en lugares con sombra y ángulos zenitales cortos.

Radiativas Es la capacidad de reflejar la radiación de onda corta (albedo; α) y de onda larga, y para emitir radiación de onda larga (emisividad; ε). Donde exista un albedo (α) bajo hay mayor absorción de onda corta teniendo un T_0 más alto y con un albedo alto una menor absorción y por lo mismo un T_0 más bajo. Mientras, una emisividad (ε) alta favorece la absorción y emisión de radiación de onda larga. Esto sucede cuando una superficie es más caliente que su alrededor, pero al tener un ε alto logrará irradiar calor de mejor manera, logrando refrescarse más rápidamente.

Térmicas Engloba las propiedades térmicas, siendo las principales la conductividad térmica (k) y capacidad calorífica (C) controlando la habilidad de difundir y conducir el calor dentro y fuera del material. Encontraremos un T_0 más alto en el día con las superficies que tengan una inercia térmica (I) baja al resistirse a transferir el calor completamente concentrado el calor en una capa superficial fina (un ejemplo de esto son los techos) que se calientan mucho. En la noche, los techos se vuelven relativamente fríos al tener una capacidad baja en contraste con las superficies pavimentadas.

Humedad Hace referencia a la disponibilidad de humedad respecto a la superficie, sobre la misma y en la vegetación que conforman la manera en la cual se puede disipar el flujo de calor latente (Q_E). Habrá T_0 mayores en cuerpos secos (hormigón, techos) que no permiten una buena evaporación en comparación de las superficies con acceso al agua (suelos húmedos, áreas verdes) donde existirá una T_0 más bajo durante el día al enfriarse cuando se evapora el agua.

Aerodinámica Especialmente la longitud de la rugosidad aerodinámica (Z_0) [2] y la protección contra el viento. Veremos como existen mayores temperaturas donde las superficies sean lisas y resguardadas del viento; serán bajas en aquellas superficies

[2] De manera práctica la rugosidad aerodinámica es la altura teórica donde la velocidad del viento se convierte en cero al no existir obstáculos [53], su similar sería el coeficiente de fricción

rugosas y con protuberancias que permitan la circulación del viento. Las construcciones de estructuras altas, como los rascacielos, son las principales causantes de estas variaciones, generando mezclas casi infinitas en las ciudades.

2.2. Percepción Remota

Es la disciplina que se encarga de adquirir información sin estar en contacto físico con el objeto de estudio. Esta información es adquirida al medir y registrar los cambios que hay en el objeto y sus alrededores, ya sea por medio del campo electromagnético (la energía reflejada o emitida), el sonido (si las ondas son reflejadas o sufren de una perturbación) y por medio de su potencial de gravedad (como se comporta el campo magnético por la presencia de este objeto) [8].

Espectro Electromagnético

La radiación electromagnética, es la manera como fluye la energía en el espacio o en algún medio al fusionarse su forma eléctrica y su forma magnética en ángulos rectos y perpendiculares en dirección del movimiento. Algunas formas en la cual ésta radiación se puede manifestar para nosotros es por medio de la luz visible, ondas de radio y rayos x siendo este tipo de ondas diferentes en su velocidad, que es la relación que existe entre su frecuencia ν (cantidad de veces que se repite) y su longitud λ (distancia que recorre) en un tiempo determinado llamado ciclo.
El espectro electromagnético entonces abarca todas las longitudes y frecuencias de estas ondas que se puede ver en la figura 2.2.

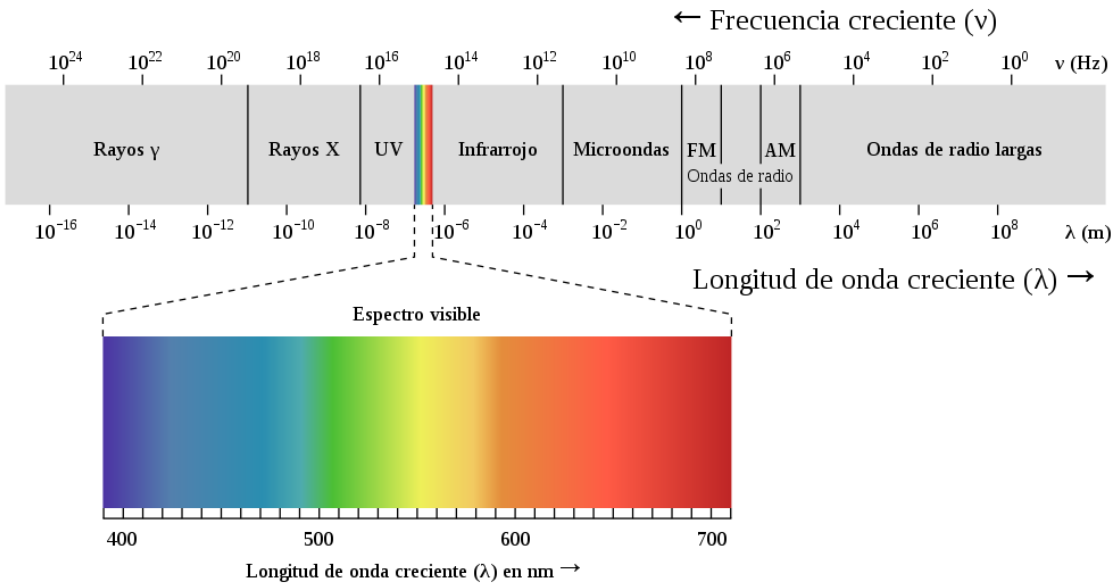


Figura 2.2 En esta imagen podemos ver el espectro electromagnético

Todo cuerpo que tenga una temperatura superior a $-273\,^{\circ}\mathrm{C}$ o $0\ K$ que es el "cero absoluto", irradia energía. Esto genera una interacción de las ondas electromagnéticas con la materia que dependerá de su estructura molecular y atómica. Haciendo posible que la energía sea reflejada, emitida, absorbida o dispersada. La proporción en la cual esto sucede está dado por:

$$\rho_\lambda + \tau_\lambda + \alpha_\lambda = 1 \tag{2.3}$$

Donde ρ_λ es el coeficiente de reflexión, τ_λ el de transmisión y finalmente α_λ el de absorción. El proceso de absorción de energía se da cuando altas frecuencias (la luz, por ejemplo) es convertido a bajas frecuencias (calor). No se puede perder energía absorbida.
Mientras que en la reflexión, siendo el fenómeno más importante para la Percepción Remota, se caracteriza por variar en diversas longitudes de onda para cada objeto, siendo dependiente de un ángulo.

Sensores

Para entonces, nosotros poder medir estas interacciones se necesitan de sensores que se caracterizan primero por registrar e interpretar el flujo de energía que los objetos radian. Se clasifican de acuerdo al tipo de operación en dos: activos y pasivos. Los sensores activos son aquellos que se caracterizan por producir la energía necesaria para posteriormente registrar

la información, mientras que será un sensor pasivo cuando este no sea capaz de producir energía y registre la energía existente en el ambiente.
Como se puede observar en la figura 2.2 el espectro se divide en diferentes regiones o bandas. Para el estudio de estas regiones se hace de la siguiente manera:

Radio Contempla las longitudes mayores a 10 cm (10×10^{-2} m) y frecuencias menores a 10 GHz (10×10^{6} Hz). Esta región usa sensores activos como radares, altímetros, sonar y con menor frecuencia radares pasivos.

Microondas Abarca las longitudes mayores a un 1 mm (10×10^{-4} m) y frecuencias menores a 300 GHz ($3 \cdot 10 \times 10^{10}$ Hz). La mayoría de interacciones existen a nivel molecular. Se usan radiómetros de microondas y sistemas de radar.

Infrarrojo Engloba la región de 1 mm (10×10^{-4} m) a 0.7 μm ($0.7 \cdot 10 \times 10^{-6}$ m), se divide en subregiones llamadas infrarrojo lejano, infrarrojo térmico e infrarrojo cercano. La rotación y vibración molecular toman un papel importante. Se utilizan espectrómetros, radiómetros, polarímetros y láseres.

Luz Visible Se extiende de a 0.7 μm ($0.7 \cdot 10 \times 10^{-6}$ m) a 0.4 μm ($0.4 \cdot 10 \times 10^{-6}$ m). Es la región que nosotros podemos percibir con nuestros propios ojos, contempla del color Rojo al Verde pasando por el Azul. Se usan los mismos instrumentos que en el anterior

Ultravioleta Comprende de 0.4 μm ($0.4 \cdot 10 \times 10^{-6}$ m) a 300 Å ($3 \cdot 10 \times 10^{-8}$ m). Se utiliza más para el estudio de atmósferas planetarias o superficies sin atmósfera debido a que los gases logran opaca en estas ondas.

Rayos X y rayos Γ Contemplan de 300 Å ($3 \cdot 10 \times 10^{-8}$ m) a 3.3 Å ($0.3 \cdot 10 \times 10^{-10}$ m) y para los Rayos Γ todavía menor longitud. Su uso es tiene mayores limitaciones al solo usarse para estudiar la superficie por medio de plataformas o aviones de vuelo bajo.

2.3. Geo Big Data

Con la era digital, el mundo está constantemente cambiando desde la forma de como se hacen las cosas hasta la manera en la cual las ideas son creadas y expresadas. Si bien el origen de la palabra es incierto, la investigación de [30] apunta a que esta palabra nació en el ámbito económico-computacional. Siendo su primera aparición pública de la mano de Erik Larson, al momento de publicar una columna de opinión quejándose de la cantidad de correos promocionales alegando lo siguiente “Los guardianes del big data dicen que lo hacen en beneficio del consumidor, pero los datos tienen una forma distinta a las intenciones

originales". Mientras que Sillicon Graphics fue la empresa que ayudó a la popularización del término para el área científica-tecnológica al referirse a la capacidad de los equipos de cómputo, siendo el fin el mismo: referirse no solo a toda la cantidad de información (data en inglés) sino a las diferentes maneras de trabajar con esta.
Algunas personas sugieren, como Doug Laney [27], que se hablará de Big Data al momento de cumplir por lo menos las siguientes "3 V":

Volumen Hace referencia a la gran cantidad de datos generados, almacenados y disponibles de las diversas fuentes. Si bien la definición es relativa en función de como evoluciona la tecnología, es perceptible al ver que cada vez nuestros dispositivos tienen mayores capacidades, así como con los equipos que son capaces de procesar esa información.

Velocidad Esto es la cantidad de datos que se generan y procesan, pudiendo satisfacer diferentes demandas y desafíos que se encuentran en el camino del crecimiento y el desarrollo. Dos tipos de velocidad se relacionan que son la frecuencia de generación y la frecuencia de manejo, esto cada cuanto se graba y es publicado.

Variedad El tipo y naturaleza de los datos. Abarca datos estructurados y no estructurados que pueden ser generados por humanos o máquinas. La forma más sencilla de separarlos sería por si es cuantitativo o cualitativo.

Específicamente, el Big Data se refiere a una colección de datos, tan grandes y complejos que es difícil emplear algoritmos y métodos tradicionales para su procesamiento. Creando desafíos que van desde la adquisición, el almacenamiento, la búsqueda, el intercambio, la transferencia, el análisis y la visualización de los datos. Recientemente, podemos ver como diversas áreas han mostrado interés en estos datos, siendo una de ellas las geociencias.
La expansión que ha estado teniendo recientemente esta área del conocimiento, ha redefinido la forma en la cual se conoce y entiende al planeta. Esto ha generado una cantidad improcedente de datos geoespaciales que al ser obtenidos datos de diversas fuentes, como lo son la topografía, la Percepción Remota, sensores especializados y recientemente el mapeo móvil, han generado una cantidad de datos diversa que sigue expandiéndose.
Tal vez el área que más llama la atención es justamente en el apartado de la Percepción Remota, al tener características extremadamente concretas y dividas en dos grandes apartados, intrínsecas y extrínsecas [26] como se puede ver en la tabla 2.1.

¿Cómo se procesa la información?

Con la constante evolución del tratamiento de la información, hace que existan diferentes técnicas para procesar y analizar la información, esto debido a que procesar el volumen de

Característica	Tipo	¿Por qué?
Varias Fuentes	Extrínseca	Se utilizan diferentes sensores Hay diferentes tipos de datos
Multiescala	Intrínseca	Considera las resoluciones
Alta Dimensión	Extrínseca	Hay dimensiones espectrales como temporales Por la heterogeneidad de los datos
Estado Dinámico	Intrínseca	La superficie cambia No todos los satélites son geoestacionarios Existen las estaciones y sus respectivos cambios climáticos
Isómero	Extrínseca	Hay diversas formas de representación (raster y vector) Estos datos pueden dar origen a más datos
No Lineal	Intrínseca	No hay mediciones constantes de información La información suele tener "ruido"

Tabla 2.1 En esta tabla se explica las características que convierten a la Percepción Remota en un tópico de alto interés para los especialistas del Big Data de acuerdo a [29].

información generada a una alta velocidad resulta un desafío descomunal para los sistemas actuales. Es ahí donde las soluciones de cómputo tienen que salir por parte de la arquitectura del sistema (son los conceptos o propiedades fundamentales de un sistema en su entorno, incorporados en sus elementos, relaciones y en los principios de su diseño y evolución [18]). Siendo dos las plataformas más dominantes, la primera, Clúster HPC (o supercomputadoras) así como plataformas en la nube.

El sistema clúster HPC se basa en el trabajo colaborativo de varias computadoras como si fueran una sola (SIS), a pesar de ofrecer una tremenda capacidad de cálculo, tienen problemas al momento de cargar, transferir y procesar grandes cantidades de información. [31]

Por otra parte, el sistema basado en computación en la nube, simula ser una computadora normal que tiene diferentes prestaciones partiendo desde el procesador, memoria y almacenamiento hasta a escalar a un sistema completamente personalizable. Algunos ejemplos son Amazon EC2 que brinda una infraestructura y servicio, Microsoft Azure ofrece una plataforma y un servicio y finalmente Google Earth Engine al dar software como parte de un servicio.

Google Earth Engine

GEE se ha vuelto tendencia al ser una plataforma para el análisis científico y la visualización de conjuntos de datos geoespaciales basada en la nube, es fácil de usar, abarcando un gran acervo que concentra datos de más de 40 años de antigüedad de diversos satélites como

Landsat, MODIS, Sentinel entre otros, así como otra información de carácter geoespacial que además se va actualizando a diario.
Para acceder y controlar esta plataforma se necesita acceso a Internet para conectarse a la API (Interfaz de programación de aplicaciones, por sus siglas en inglés) y al IDE (Entorno de desarrollo integrado) permitiendo poder desarrollar código y visualizar datos de manera rápida. Para acceder es necesario registrarse en la página https://earthengine.google.com/ donde después de un proceso de registro el usuario podrá acceder a guías, tutoriales, vídeos, entre otras cosas. Así mismo permite la descarga de datos.
Con GEE es necesario programar, si bien no requiere una gran experiencia previa, se recomienda tener ciertas nociones básicas de programación, sobre todo en el lenguaje de JavaScript (en el cual está basado el entorno) aunque es posible por medio de otros lenguajes como Python. Como usuario, uno puede crear y ejecutar algoritmos en imágenes georreferenciadas y vectores almacenados en la infraestructura de Google, por medio de una Interfaz de Usuario montada en una página web, como se puede ver en la imagen 2.3.

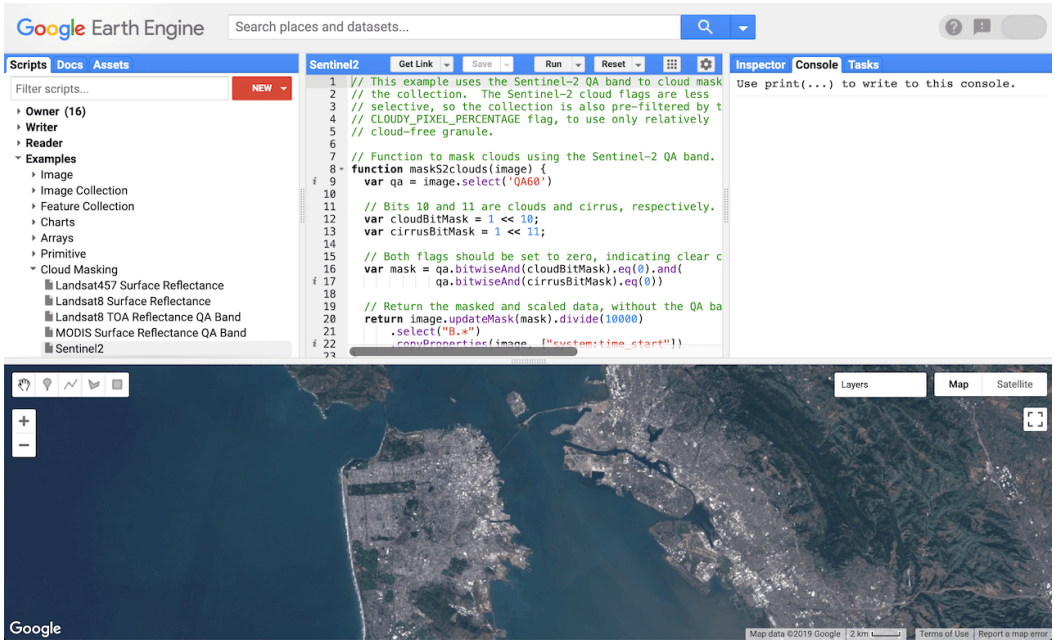


Figura 2.3 Editor de código de Google Earth Engine

El editor de código (Code Editor), es el entorno en el cual se desarrollan las aplicaciones de Earth Engine. En el panel central es donde se escribe el código. Encima del editor hay botones para guardar el script actual (Save), ejecutarlo (Run) y borrar el mapa (Reset). El botón Obtener enlace (Get Link) genera una URL única para el script en la barra de direcciones. El panel izquierdo contiene ejemplos de código, sus scripts guardados, una

referencia, API de búsqueda y un administrador de activos para datos privados. El panel derecho tiene un inspector para consultar el mapa, una consola de salida y un administrador para tareas de ejecución prolongada. El botón de ayuda en la parte superior derecha contiene enlaces a la guía y otros recursos para obtener ayuda. La parte inferior es el mapa en el cual es posible visualizar la información [12].

Capítulo 3

Estimación del calor superficial en la Ciudad de México y espacios conurbados

3.1. Características físico-geográficas de los sitios urbanos y semi-urbanos

Para la humanidad siempre han existido dos grandes interrogantes, ambas partiendo del hecho de conocer su lugar y de esa forma poder descubrir y definir al mundo que lo rodea: el espacio y el tiempo. Como humanidad al tiempo se ha logrado, de cierta manera, asignarle una magnitud física para poder enumerar los acontecimientos del pasado, presente y futuro. Mientras que la forma de entender el espacio ha sido el plantear el lugar donde uno se encuentra, incluyendo la superficie terrestre, el espacio dentro de nuestro planeta, así como fuera.
Normalmente, se asocia con términos geográficos como lo son las formaciones terrestres (continentes, islas), cuerpos de agua (océanos, mares), y conceptos fisiográficos (desierto, pradera, montaña). Por lo mismo que resulta importante describir lo que se ha realizado como humanidad, se ha incluido las áreas ocupadas por los asentamientos humanos (ciudades, pueblos y aldeas) así como las estructuras realizadas por las comunidades [7]. Y es ahí donde surgen ciencias como la Ekística, que tiene como objetivo estudiar los asentamientos humanos. Tal vez la definición más sencilla para entender que es un asentamiento es: el lugar donde la gente vive y trabaja.
Una vez partiendo de la idea de que existen asentamientos, podríamos hablar de que hay dos grandes divisiones: rural y urbana. Pero se debe tomar en cuenta que no es fácil distinguir entre ambos tipos, sobre todo por como cada país asignará en función de sus políticas: unos países usan la estadística de su población para hacer esta clasificación donde existen variaciones, por ejemplo Dinamarca considera 200 habitantes, mientras China requiere 100,000. Algunos delimitan por decisiones administrativas e históricas. Mientras, otros más toman en cuenta el empleo que hay en la región, así como la infraestructura y servicios.
El banco mundial propone medir el "grado de urbanización" distingue tres tipos de asentamientos [28]:

- Ciudades, tienen una población de al menos 50 000 habitantes en celdas de malla contiguas, densamente pobladas (más de 1500 habitantes por kilómetro cuadrado)
- Localidades (o pueblos) y zonas de densidad intermedia, con una población de al menos 5000 habitantes en celdas de malla contiguas cuya densidad es de un mínimo de 300 habitantes por kilómetro cuadrado
- zonas rurales, que están constituidas principalmente por celdas de malla con baja densidad de población.

Para México tenemos como una autoridad en la materia al Instituto Nacional de Estadística y Geografía (INEGI), que propone en el Marco Geoestadístico Nacional la división de las localidades en dos: urbanas y rurales. Las localidades rurales son aquellas que tienen una población mayor o igual a 2 500 habitantes o que sean cabeceras municipales. Mientras, las localidades rurales tienen una población menor a 2 500 habitantes y no son cabeceras municipales. En su nivel base, que son las áreas geoestadísticas básicas, las áreas urbanas se consideran perfectamente delimitadas por calles, avenidas, andadores o cualquier otro rasgo de fácil de identificación en el terreno y cuyo uso del suelo sea principalmente habitacional, industrial, de servicios, comercial, etcétera, solamente se asignan al interior de las localidades urbanas. Mientras las rurales tienen una extensión territorial variable y se caracteriza por el uso del suelo de tipo agropecuario o forestal. Contienen localidades rurales y extensiones naturales como pantanos, lagos, desiertos y otros, delimitada por lo general por rasgos naturales (ríos, arroyos, barrancas, etcétera) y culturales (vías de ferrocarril, líneas de conducción eléctrica, carreteras, brechas, veredas, ductos, límites prediales, etcétera). [6] INEGI a partir de los resultados del Censo de Población y Vivienda 2010 subdivide las localidades urbanas de la siguiente manera [17]:

- Localidades con 5 mil o más habitantes
- Localidades con menos de 5 mil habitantes, conurbadas con una de 5 mil o más habitantes
- Localidades conurbadas entre sí, cuya suma de población fuese 5 mil o más habitantes
- Cabeceras municipales de 2 500 y más habitantes
- Localidades con menos de 5 mil habitantes, amanzanadas ubicadas dentro del perímetro de 3 km de una localidad de cinco mil o más habitantes y que fuesen desarrollos habitacionales urbanizados

Como se puede leer en [35]

> «El término zona metropolitana se acuñó y desarrolló en Estados Unidos a partir de los años veinte del siglo pasado y se utiliza la mayoría de las veces para referirse a una ciudad “grande” cuyos límites rebasan los de la unidad político-administrativa que originalmente la contenía; en el caso de México, dicha unidad es el municipio (Negrete y Salazar, 1986:98-99). En el país, este proceso se inició en la década de los cuarenta en las ciudades de México, Monterrey, Torreón, Tampico y Orizaba (Sobrino, 1993:125) ».

Para la Secretaría de Desarrollo Social (SEDESOL), el Instituto Nacional de Estadística, Geografía e Informática (INEGI) y el Consejo Nacional de Población (CONAPO) en su trabajo publicado en el 2019 [15] indica que los criterios establecidos fueron crear un conjunto de dos o más municipios con una población mayor a 100 mil o más habitantes, que hayan rebasado en actividades y funciones a sus límites (llamados municipios centrales), influyendo directamente a municipios aledaños (municipios exteriores geográficos) existiendo un alto grado de integración socioeconómica (municipios exteriores estadísticos). Así como aquellos reconocidos por los diferentes niveles de gobierno donde está regulado su desarrollo urbano y la ordenación territorial (municipios exteriores de planeación y política urbana).
En el caso de esta Zona Metropolitana se denominó "Valle de México" con la clave 09.01, abarcando las entidades federativas de la Ciudad de México, Hidalgo y México con 76 municipios, 52 centrales, 9 exteriores geográficos-estadísticos y 15 exteriores de planeación y política urbana. Con una superficie de $7,866\ km^2$ y una densidad media urbana de $160.1\ DMU^2(hab/ha)$.
La Ciudad de México y su zona conurbana se encuentra desarrollada en la región centro de la República Mexicana, extendiéndose por la Cuenca de México que tiene un área aproximada de $9,560\ km^2$ con una altitud media de $2,390\ m.s.n.m$ en sus valles donde se concentraban una serie de lagos. Era una depresión cerrada de manera natural, hasta que en el siglo XVIII sufrió de alteraciones, rodeada de montañas de origen volcánico. Geológicamente, el área sur de la Sierra Guadalupe es la región mejor investigada de la Cuenca de México, contempla parte de la Ciudad de México, y se le llama valle de México o valle del Anáhuac, se encuentra parcialmente dividida por varias montañas de menor elevación. Se encuentra en la provincia fisiográfica conocida como Eje Neovolcánico, en la subprovincia de Lagos y Volcanes de Anáhuac, conformada por grandes sierras volcánicas, coladas de lava, conos dispersos, amplios escudos-volcanes de basaltos, depósitos de arenas y cenizas dispersas entre extensas llanuras, destacan amplias cuencas cerradas ocupadas por lagos o por depósitos de lagos antiguos como: Zumpango, Guadalupe, Texcoco y Xochimilco. Tres principales zonas hidrológicas han sido definidas para el Valle de México: la zona lacustre (elevaciones de menor altura), el piedemonte o zona de transición (entre el lecho de los antiguos lagos y las montañas de mayor pendiente) y la zona montañosa. [41]
La posición geográfica y la topografía ocasionan que la Ciudad de México y su zona conurbana sean afectadas por sistemas de circulación atmosférica que definen dos épocas climáticas: la época de lluvias (junio-octubre) donde caen la mayoría de los 700 milímetros de agua de lluvia que caen anualmente en la región concentrándose en unas cuantas tormentas intensas, la seca-fría (noviembre-febrero) y la seca-húmeda (marzo-mayo). Durante la primera, los vientos alisios procedentes del mar coadyuvan a la presencia y aumento de la precipitación,

así como un alto contenido de humedad, los meses invernales en los que dominan los vientos del oeste asociados a las condiciones de sequía o tipo polar y los meses cálidos donde hay características tropicales pero secas. Las altitudes que tiene la Ciudad de México y su zona conurbana mayores a los $2,000$ $m.s.n.m$, en estrecha relación con la temperatura, definen dos zonas térmicas para la Ciudad de México y su zona conurbana: la zona templada, con temperatura media anual entre 12° y 18° C, asociada con altitudes cercanas a $2,000$ $m.s.n.m$ y hasta $2,900$ $m.s.n.m$, y la zona semi-fría, caracterizada por una temperatura media anual entre 5° y 12° C para altitudes mayores entre $2,000$ y $4,000 m.s.n.m$. [16]

3.2. Estimación del calor superficial

Imágenes satelitales Landsat

Landsat es un programa elaborado por el Servicio Geológico de Estados Unidos (USGS) y la Administración Nacional de Aeronáutica y el Espacio (NASA), siendo su principal objetivo la adquisición de imágenes satelitales de la Tierra por medio de satélites. Formando un gran acervo que data desde 1972, con 9 diferentes satélites. Estos satélites llevan montados sensores especializados en captar la radiación térmica infrarroja [1], obteniendo la temperatura de brillo en la capa superior de la atmósfera (TOA). Una de las grandes ventajas de usar estos datos es que son considerados consistentes, tanto en sus mediciones como en sus resoluciones. Incluso Landsat, divide sus colecciones en niveles (Tiers) con base en su calidad. Aquellas que tengan una alta calidad, son clasificados como Tier 1; lo que las hace destacar para poder hacer análisis de tiempo [48].

Se utiliza Landsat 8, a pesar de que al momento de la elaboración del presente trabajo no era la plataforma más reciente, pero si tenía una colección superior de imágenes. Este satélite se encuentra a una altitud de 705 kilómetros, con un ancho de barrido de 185 km, desplazándose de norte a sur sobre la luz solar en una órbita sincrónica con el Sol. Logra completar una órbita cada 99 minutos, logrando 14 al día y una resolución temporal de 16 días. Posee dos sensores: OLI (Operational Land Imager) y TIRS (Thermal Infrared Sensor. OLI (Generador de imágenes terrestres), es un radiómetro de barrido multibanda. Capta el espectro visible, Infrarrojo Cercano (NIR) e Infrarrojo de Onda Corta (SWIR) con una resolución espacial de 30 metros, y una banda pancromática de 15 metros. Mientras, TIRS (Sensor Infrarrojo Térmico) es un radiómetro infrarrojo de dos canales, cuyo objetivo fue obtener

[1] No todos los satélites cuentan con este sensor, es a partir de Landsat 4 que se han empezado a montar sensores térmicos

las características de temperatura de la superficie y estudiar el proceso de transferencia de calor y humedad. Ambos instrumentos fotografían en modo de barrido a lo largo de la trayectoria de la nave, lo que reduce el nivel de distorsión radiométrica en comparación con los instrumentos de barrido transversal utilizados en satélites Landsat anteriores [50].
De este satélite, las bandas que se utilizaron han sido, además de la correspondiente al TIRS, las bandas del rojo y el infrarrojo cercano. Se ocupan las del sensor OLI para derivar el NDVI (Índice de Vegetación de Diferencia Normalizada).

Sensor	Nombre	Descripción	Longitud de onda (μm)	Dataset	Res. Espacial
OLI	4	Roja	0.636 - 0.673	C02/T1_L2	30 m
OLI	5	Infrarrojo Cercano (NIR)	0.851 - 0.879	C02/T1_L2	30 m
TIRS	10	Infrarrojo Térmico 1	10.60 - 11.19	C02/T1_RT_TOA	30 m[a]
TIRS	11	Infrarrojo Térmico 2	11.50 - 12.51	C02/T1_RT_TOA	30 m[b]

Tabla 3.1 Información respecto a las bandas de Landsat 8 que fueron utilizadas.

[a]Originalmente de 100 m, remuestreada a 30 m.
[b]Originalmente de 100 m, remuestreada a 30 m.

Dataset *"LANDSAT/LC08/C02/T1_L2"*

El nombre en el sistema de GEE es *USGS Landsat 8 Level 2, Collection 2, Tier 1*. Este dataset corresponde a la colección Landsat 8 Surface Reflectance o Landsat 8 SR (como suele denominarse). Landsat Surface Reflectance mide la fracción de la radiación solar entrante reflejada desde la superficie de la Tierra hacia el sensor Landsat [47].
Este dataset contiene la reflectancia de la superficie corregida atmosféricamente y la temperatura de la superficie terrestre derivada de los datos producidos por los sensores de Landsat 8. Son 18 bandas, de las cuales 5 corresponden a la parte del espectro visible e infrarrojo cercano, 2 bandas infrarrojas de onda corta (SWIR) procesadas a reflectancia superficial ortorrectificada, y una banda infrarroja térmica (TIR) procesada a temperatura superficial (TS) ortorrectificada. También contienen bandas intermedias utilizadas en el cálculo de los productos TS, así como bandas QA [2].
Los productos Landsat 8 SR se crean con el Código de reflectancia de la superficie terrestre (LaSRC). Todos los productos de la Colección 2 ST se crean con un algoritmo de un solo canal creado conjuntamente por el Instituto de Tecnología de Rochester (RIT) y el Laboratorio

[2]Quality Assessment Band, esta banda proporciona información útil para los usuarios que desean optimizar el valor de los píxeles como un indicador de primer nivel al apuntar qué píxeles pueden verse afectados por las condiciones de la superficie, la contaminación de las nubes o las condiciones del sensor.

de Propulsión a Chorro (JPL) de la Administración Nacional de Aeronáutica y del Espacio (NASA) [14].

Dataset *"LANDSAT/LC08/T1_RT_TOA"*

El nombre en el sistema de GEE es *USGS Landsat 8 Collection 2 Tier 1 and Real-Time data TOA Reflectance*. Los coeficientes de calibración se extraen de los metadatos de la imagen. [13]. Contiene 16 bandas, de las cuales 11 son las usadas por Landsat 8. 2 son QA y las restantes son los ángulos azimutales y zenitales tanto solares como de visión del satélite.

Datos Atmosféricos

Al estar las imágenes en la termosfera, un componente fundamental a considerar son las diversas condiciones que pueden afectar la manera en la cual el sensor realiza sus mediciones. Una de estas condiciones es conocer las características del agua en las capas inferiores de la atmósfera, sobre todo para el uso de imágenes obtenidas por TIRS. GEE contempla los datos obtenidos del vapor de agua total de la columna (TCWV) del Proyecto de Reanálisis NCEP/NCAR (Centro Nacional para la Predicción Ambiental / Centro Nacional para la Investigación Atmosférica) [25]. Su objetivo principal producir nuevos análisis atmosféricos utilizando datos históricos, así como producir análisis del estado atmosférico actual. Utiliza un sistema de análisis/pronóstico de última generación para realizar la asimilación de datos utilizando datos pasados desde 1948 hasta el presente. Los datos tienen una resolución temporal de 6 horas (0000, 0600, 1200 y 1800 UTC) y una resolución espacial de 2,5 grados.

Emisividad Superficial

En este caso, los algoritmos utilizados para la obtención del LST, requieren valores prescritos de emisividad superficial, por lo que se optó por ocupar la colección ASTER-GED. ASTER-GED es el acrónimo de Base de datos Global de Emisividad del radiómetro de Reflexión y Emisión Térmica Avanzada Espacial. Esta colección fue desarrollada por el Laboratorio de Propulsión Jet (JPL) del Instituto de Tecnología de California. Esta colección incorpora datos de emisividad de 5 bandas en la región TIR [23]. Su derivado son los píxeles de cielo despejado para todos los datos ASTER, que fueron tomados en un periodo de ocho años. Tiene una resolución de espacial de 100 metros y una precisión aproximada de 0.01.
Uno de los desafíos más grandes respecto a este apartado surge de la variabilidad de la cobertura vegetal que puede existir en un periodo anual. Fue necesario aplicar un ajuste derivado de las bandas NDVI de Landsat y ASTER, de esta forma considerar la Fracción de

Cobertura Vegetal (FCV). La FCV propuesta por [2] es la siguiente:

$$FVC = (\frac{NDVI - NDVI_{Descubierto}}{NDVI_{Cubierto} - NDVI_{Descubierto}})^2 \tag{3.1}$$

Como señaló [9] en los estudios previos [51, 44, 43, 49, 20], el valor asignado a $NDVI_{Descubierto} = 0.2$ y $NDVI_{Cubierto} = 0.86$. Siguiendo la recomendación de Jiménez-Muñoz et al. quienes argumentan que para imágenes de alta resolución, los valores de $NDVI_{Cubierto}$ deben ser de 0.8 a 0.9 y no de 0.5 como usualmente se asigna. [20]
Así mismo, se deben considerar los valores de emisividad sobre área con vegetación, aunque estos pueden ser obtenidos usando el método de Cobertura Vegetal [4, 46], definido como:

$$\varepsilon_b = FVC_{\varepsilon_{b,\text{veg}}} + (1 - FVC)\varepsilon_{b,\text{descubierto}} \tag{3.2}$$

Donde $\varepsilon_{b,\text{veg}}$ y $\varepsilon_{b,\text{descubierto}}$ son la emisividad de la vegetación y el suelo desnudo para una banda (b). La emisividad de las superficies con cobertura vegetal, suele tener pequeñas variaciones en la región TIR, y suele usarse $\varepsilon_{b,\text{descubierto}} = 0.99$ [42]. Para cada pixel, la emisividad efectiva ε_b para cada banda Landsat sigue estas reglas:

1. La Fracción de Cobertura Vegetal de Aster, tiene su origen de la Ecuación 3.1.

2. La emisividad del suelo desnudo $\varepsilon_{b,\text{descubierto}}$ para cada banda ASTER es tomada del valor original de la emisividad ε_b, mientras su ASTER FVC usa la ecuación 3.2 con el valor para $\varepsilon_{b,\text{descubierto}} = 0.99$.

3. La emisividad del suelo desnudo para cada banda Landsat TIR ($\varepsilon_{b,\text{descubierto}}$) se deriva de los valores de emisividad del suelo desnudo, ajustando de acuerdo a los coeficientes que ajustan espectralmente a ASTER para Landsat. Estos se pueden ver en la tabla 2 de [32]

4. Los valores de FVC para la imagen de Landsat se calculan a partir de los respectivos valores de NVDI utilizando la ecuación 3.1.

5. Se apoya en la ecuación 3.2 para obtener la emisividad actual superficial para cada banda Landsat TIR.

Estimación de la Temperatura Superficial

La elección del algoritmo LST, suele ser una función semiempírica de las temperaturas de brillo de la parte superior de la atmósfera (TOA) en TIR, está intrínsecamente vinculada

a las características del sensor que se utiliza [33]. Se dividen en dos grandes maneras, la primera es usando la información cuando las mediciones están en una sola banda (ventana única o Mono Window, en inglés), por otro lado, usando una combinación de bandas TIR aprovechando el enfoque generalizado de ventanas divididas (generalized split-windows, GSW). En general, este tipo de algoritmos se basa en una regresión lineal entre las cantidades medidas en la parte superior de la atmósfera y LST, utilizando datos auxiliares como la emisividad espectral, la columna total de vapor de agua (TCWV), el ángulo de visión cenital (ZVA) y la cobertura terrestre.

Algoritmo SMW (Statistical Mono-Window)

El algoritmo utilizado por CM-SAF [3], se basa en una relación empírica entre las temperaturas de brillo TOA en un solo canal TIR y LST y utiliza una regresión lineal simple [11]. El modelo consiste en una linealización de la ecuación de transferencia radiativa que mantiene una dependencia explícita de la emisividad superficial:

$$LST = A_i \frac{Tb}{\varepsilon} + B_i \frac{1}{\varepsilon} + C_i \tag{3.3}$$

Donde T_b es la temperatura de brillo en la TOA para la banda TIR, y ε es la emisividad superficial para la misma banda. Los coeficientes A_i, B_i y C_i son determinados de regresiones lineales donde se simula la transferencia radiativa para 10 clases de TCW, van de 0 a 6 cm con pasos de 0.6 cm. Una diferencia realmente significativa es que no hay coeficientes de ángulo de visión, debido a que Landsat tiene un campo cercano al nadir en comparación de los satélites de Meteosat.

Algoritmo GSW (Split Window)

El algoritmo propuesto por [19] parte de atenuar la radiación para la absorción atmosférica, haciéndola proporcional a la diferencia que existe entre la radiancia de mediciones simultáneas en dos longitudes de onda diferente.

$$\begin{aligned} T_s = {} & T_i + c_1(T_i - T_j) + c_2(T_i - T_j)^2 + c_0 \\ & + (c_3 + c_4 W)(1 - \varepsilon) + (c_5 + c_6 W)\Delta\varepsilon \end{aligned} \tag{3.4}$$

[3] Aplicación de Satélites para la Vigilancia Climática, es una dependencia de la EUMETSAT (Organización Europea para la Explotación de Satélites). Una organización intergubernamental creada a través de una convención internacional acordada por un total actual de 30 Estados miembros europeos, cuyo objetivo es dar seguimiento al tiempo, el clima y el medio ambiente desde el espacio. Entre los satélites que operan están: los satélites geoestacionarios Meteosat y las misiones de Sentinel [10].

Donde T_i y T_j corresponde a la temperatura de brillo en el sensor en las dos bandas (i y j), ε es la emisividad media ($\varepsilon = (\varepsilon_i + \varepsilon_j) \cdot 0.5$), $\Delta\varepsilon$ es la diferencia de emisividades ($\Delta\varepsilon = (\varepsilon_i - \varepsilon_j)$), W es el contenido total de vapor de agua atmosférico y $c_0 - c_6$ son los coeficientes determinados por [21].

3.3. Características de los datos geoespaciales

Procedimiento

Al estar basado GEE en JavaScript, un lenguaje interpretado, ligero y orientado a objetos [34]), se dio prioridad a mantener el código lo más simple posible. Reduciendo su complejidad en lo posible, separando en módulos independientes.

Es importante aclarar que para ambos módulos, hay un procesamiento de depuración del dataset, es decir, se hace una selección de la zona de estudio, que coincida con las fechas planteadas y un filtro de nubes. En este último filtro, es donde apoyados en la banda QA se toman como referencia los píxeles que presentan fallas debido a la nubosidad o donde los datos han sido marcados como inconsistentes y se enmascaran para que no sean contemplados y afecten de alguna manera el estudio. En las figuras 3.1 y 3.2 es bastante notorio el parecido con los procedimientos. Siguen la misma estructura: reciben datos por parte del usuario (región de interés y fechas del estudio). De ahí se ha "fragmentado" en tres bloques: el bloque azul corresponde a la carga de datos atmosféricos. Se solicita el dataset de *"NCEP_RE/surface_wv"*, del cual se extraen el vapor de agua, interpolando los tiempos del modelo respecto a la imagen Landsat; el bloque amarillo, es el que sirve para generar los valores de emisividad partiendo del dataset *"C02/T1_L2"*. Primero genera las zonas de cobertura vegetal, los cuales se combinan con ASTER y de esta forma se obtienen mejores valores de emisividad al tener una parte dinámica. Y finalmente el bloque rosa sirve para las imágenes TIR. Se "combinan" en la ecuación 3.3 y de esta manera se obtiene el LST.

Sin embargo, hay diferencias muy marcadas que deben aclararse en los procedimientos: para el procesamiento de los datos atmosféricos requieren diferentes resultados. Primero se obtiene el valor de W (la ecuación 3.3 explica que es el contenido total de vapor de agua atmosférico), este valor se debe asignar a un valor y de esta reclasificación se asigna los coeficientes A, B y C. La emisividad que se obtiene para MW, es resultado de una convolución obtenida de la emisividad ASTER y las bandas de Landsat que comparten espectro, mientras que ε y $\Delta\varepsilon$ son resultado de calcular la emisividad por banda, respetando el espectro que comparten.

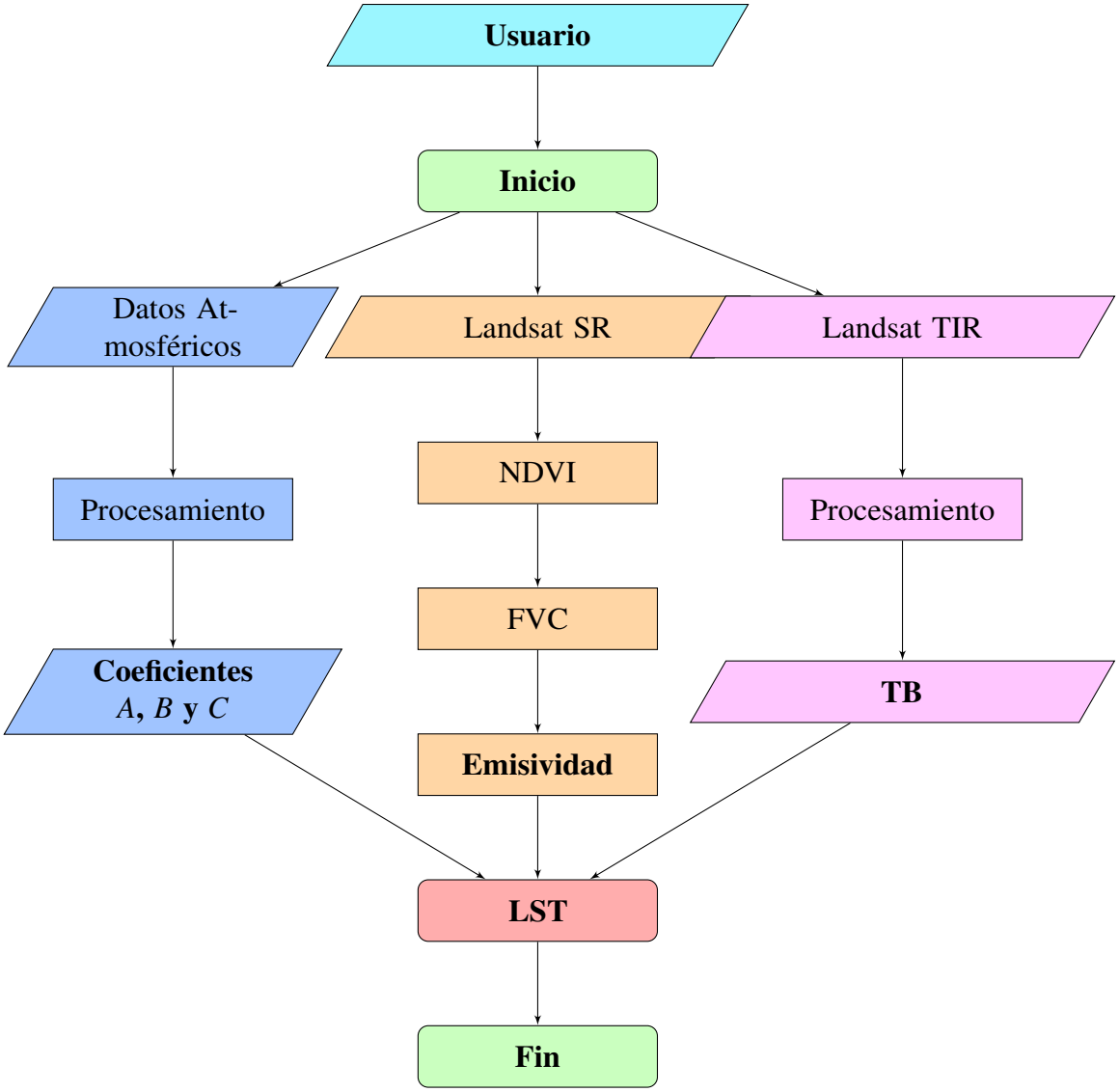


Figura 3.1 Esta imagen es el procedimiento para obtención de imágenes LST, con el algoritmo designado para Mono Window.

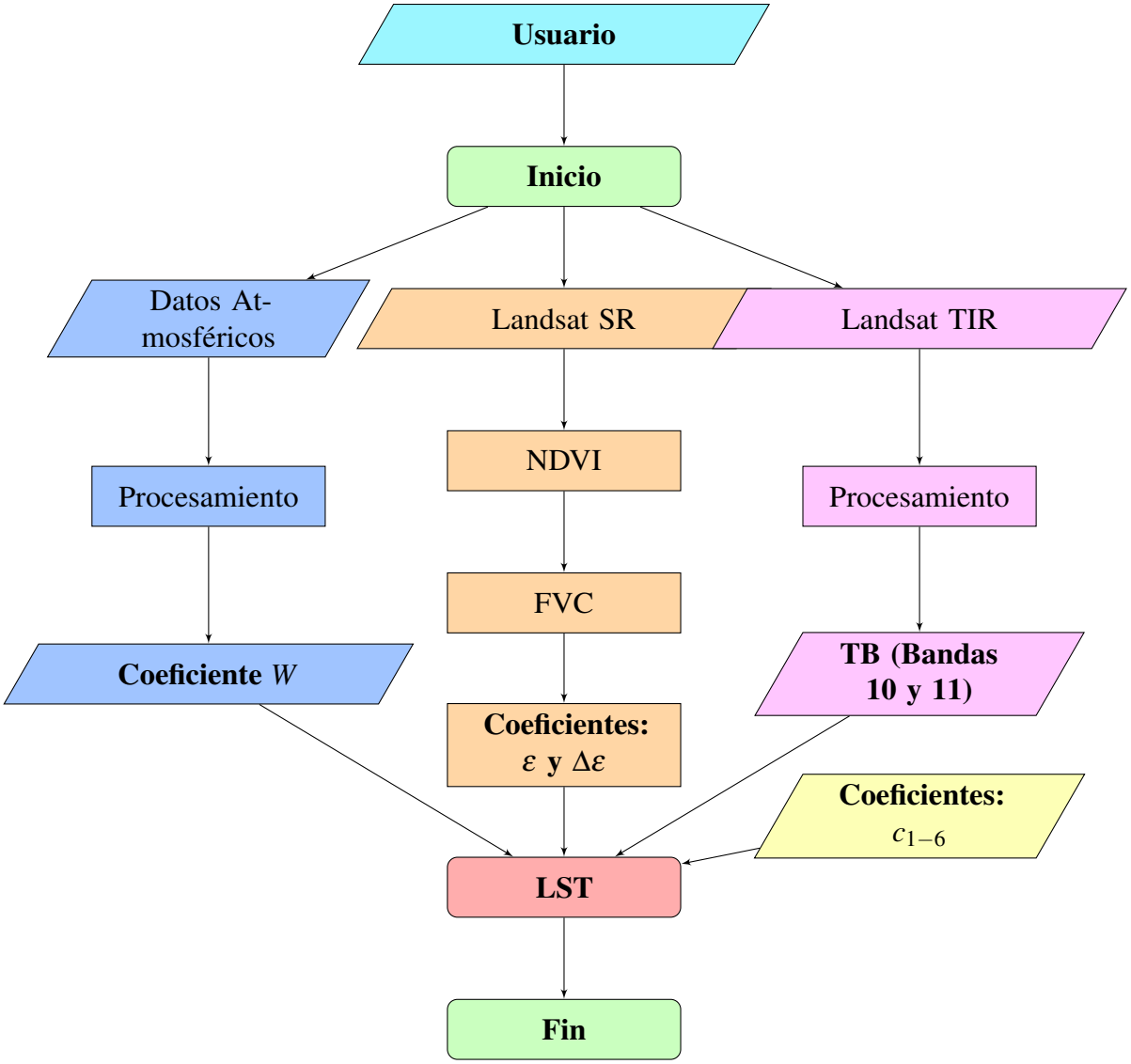


Figura 3.2 Esta imagen es el procedimiento para obtención de imágenes LST, con el algoritmo designado para Split Window.

3.4. Procesamiento de imágenes Landsat

Selección de escenas

Derivado de la extensión territorial de la Ciudad de México y su zona conurbana, fue necesario emplear tres escenas Landsat para cubrir completamente el área de estudio (Figura 3.3). Aunque una de estas escenas incluye únicamente una porción marginal del territorio, las tres corresponden al mismo *path* orbital (path 26) [4], asignado al 26. , lo que permitió mantener un criterio homogéneo de selección.

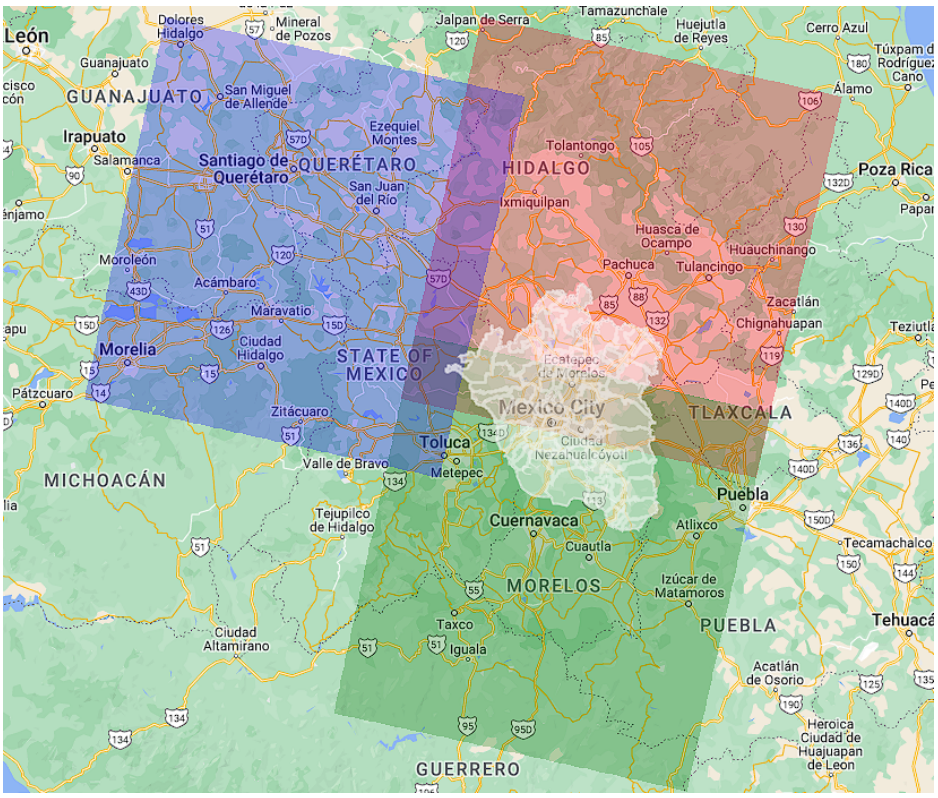


Figura 3.3 Se aprecia que hay 3 cuadros, en este caso es la cobertura espacial de las 3 imágenes que existen y tienen alguna intersección con la Ciudad de México y su zona conurbana. Mientras al centro, de color blanco la Ciudad de México y su zona conurbana

Generación del mosaico

Con el objetivo de reducir el consumo de recursos, se generó un mosaico a partir de las escenas seleccionadas. Dado que la diferencia temporal entre las adquisiciones fue inferior a

[4]Los satélites Landsat tienen un sistema de referencia, conocido como Worldwide Reference System. Este sistema asemeja una malla, el nombre de “path” es la órbita descendente del satélite, mientras “row” es el camino que recorre de norte a sur.

30 segundos respecto a su punto central, se consideró adecuada su integración para conformar una imagen continua del área de estudio (Figura 3.4).

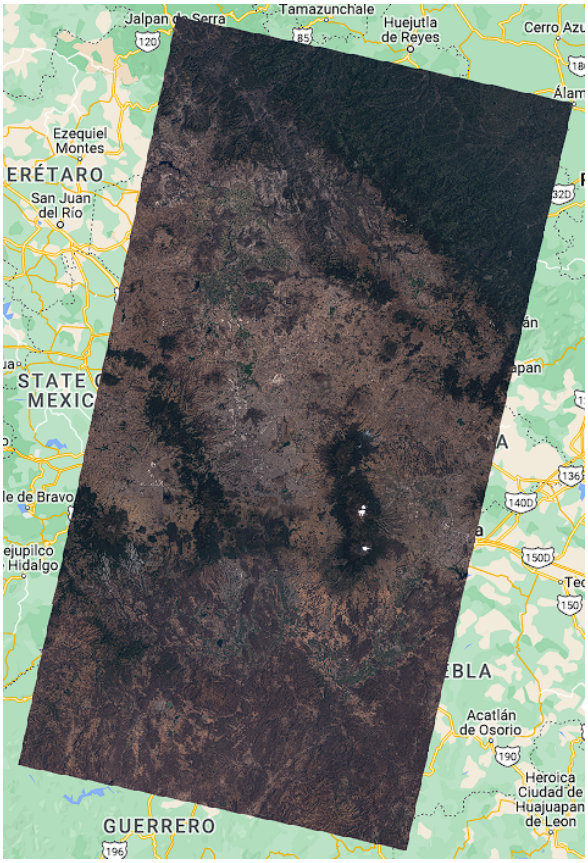


Figura 3.4 Mosaico que contempla la extensión de la Ciudad de México y su zona conurbana

Recorte y evaluación de utilidad

Tras la generación del mosaico, se realizó el recorte conforme al límite del área de estudio. El filtrado de imágenes por porcentaje de nubosidad (menor al 80 %) no se aplicó de manera inicial, ya que la nubosidad reportada en los metadatos no considera si la región de estudio se veía afectada. En su lugar, se calcula el porcentaje efectivo de cobertura libre de nubes mediante el conteo de píxeles posterior al enmascaramiento. En la figura 3.5 se puede observar el caso de una imagen en los metadatos reportaba una nubosidad mayor al 80 %, pero el área de estudio no se ve afectada.



Figura 3.5 Imagen compuesta RGB, recortada con la Ciudad de México y su zona conurbana

Generación de la cartografía temática

Con el fin de obtener una caracterización del patrón térmico durante el periodo de estudio, se generó un mapa temático promedio. Este producto se obtuvo a partir del promedio pixel a pixel de las imágenes que cumplían con los criterios de calidad establecidos.
Los píxeles previamente enmascarados por nubosidad no fueron considerados en el cálculo. En los casos donde el número de observaciones válidas no fue suficiente para estimar una

media representativa, el píxel se clasificó como valor nulo. Esta condición explica la presencia de áreas sin información en la imagen final.

Capítulo 4

Resultados y Discusión

4.1. Caracterización del conjunto de datos

Se obtuvieron un total de 217 imágenes, comprendidas dentro del periodo del 18 de abril de 2013 al 8 de enero de 2023. Como se puede ver en la figura 4.1, hay 7 cuadros en blanco de estos, 2 son debido al traslape del calendario (fin de año) y las otras 5 por no cumplir con las condiciones para ser consideradas parte de la colección de Landsat Tier 1.

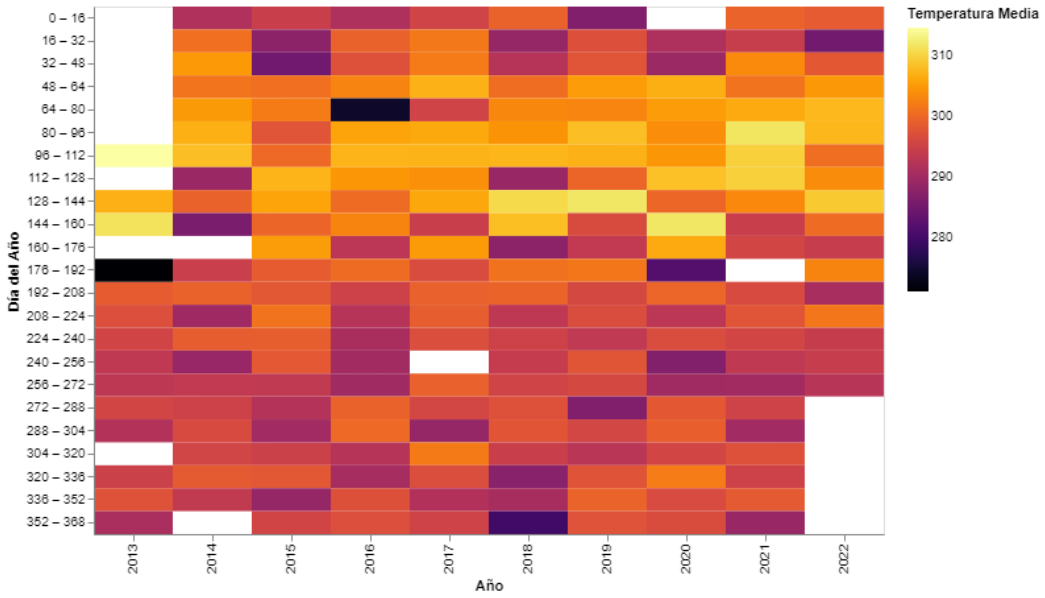


Figura 4.1 Valor medio de LST por imagen calculada

Es notorio que del periodo comprendido entre los 64 a los 160 días (inicios de marzo-inicios de junio) las temperaturas son más elevadas, coincidiendo con la época seca-húmeda que existe en la Ciudad de México y su zona conurbana. Y existen valores de temperatura menores para la temporada seca-fría. Como este gráfico corresponde a todas las imágenes computadas, tiene valores de temperatura inferiores a los 280 Kelvin $\approx 7°C$, y son muy fáciles de distinguir visualmente por su pigmentación oscura, donde resaltan con el amarillo que las rodea. En comparativa con la figura 4.2, donde se obtuvieron 86 imágenes con un porcentaje de nubosidad inferior al 19 % en la región de estudio (aproximadamente el 40 % de las imágenes).

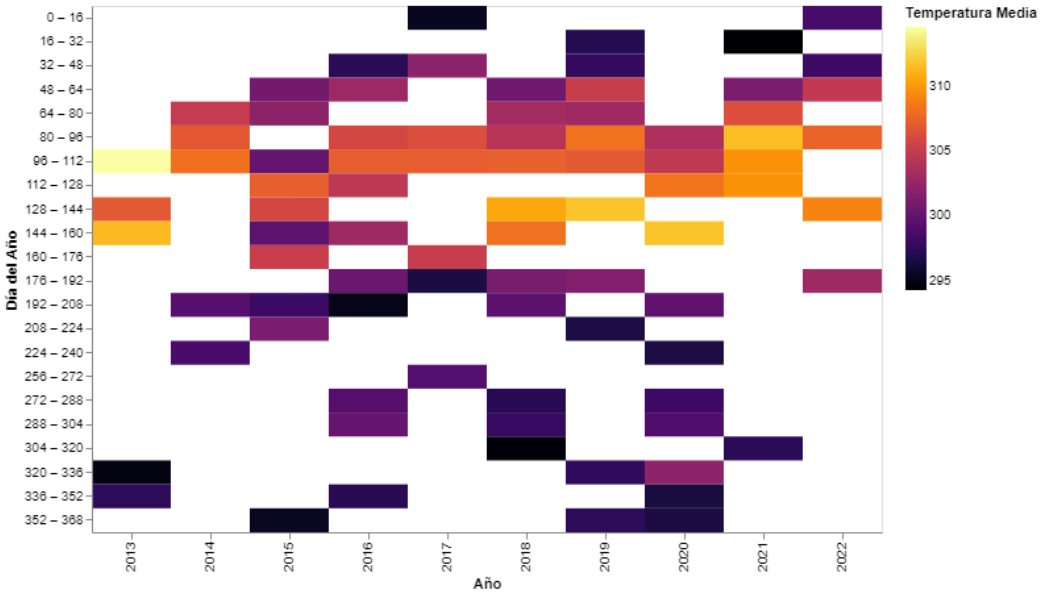


Figura 4.2 Valor medio de LST por imagen calculada, con una nubosidad inferior al 80 en la zona de estudio

4.2. Comportamiento temporal de la LST

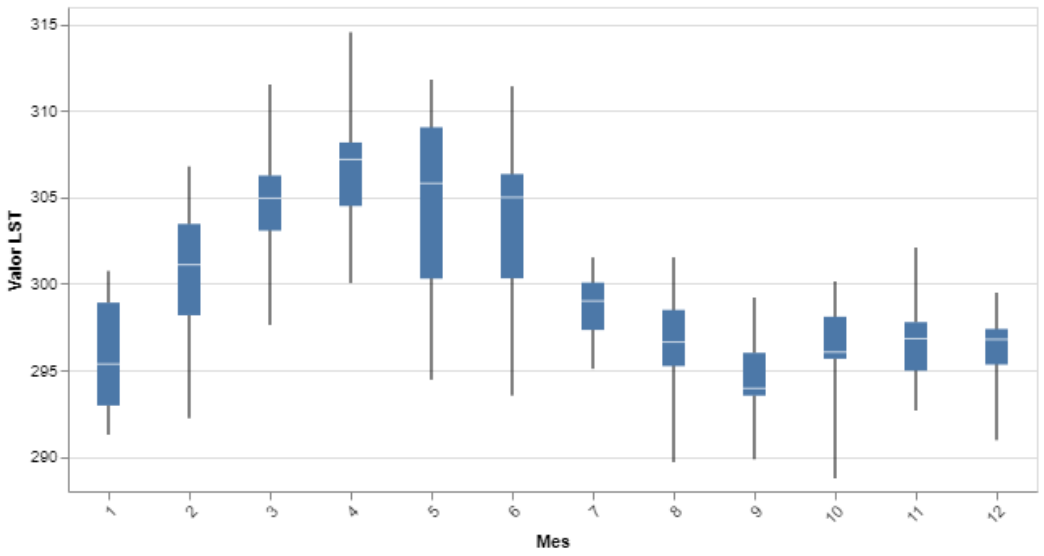


Figura 4.3 Diagrama de caja del promedio mensual del LST

En el diagrama de caja (ver figura 4.3) se pueden observar las siguientes características:

- Durante los meses de julio a diciembre, aunque se registran diferencias de hasta 10 $K \approx 4°C$ en algunos casos, esta caja refleja una estabilidad térmica, indicando que la temperatura se mantiene constante a lo largo de estos meses.

- Los meses 1 y 2, así como los meses 5 y 6, indican una alta variabilidad en los datos. Este patrón refleja las transiciones entre las estaciones más marcadas, específicamente el final del período frío en enero y febrero, así como la disminución de la humedad en junio y julio.

- El mes 4 (abril) destaca por los valores más elevados de la variable analizada, lo cual muestra una correlación con el periodo más cálido del año. En contraste, el mes 9 (septiembre) presenta los valores más bajos, coincidiendo con el final de la época de lluvias.

- Los meses 10, 11 y 12 prácticamente tienen la misma media (296.5 $K \approx 23°C$). Son los meses donde está el otoño.

- El diagrama de caja no presenta valores atípicos, sin embargo, los bigotes se extienden a lo largo de la totalidad del rango, abarcando una diferencia de 13 $K \approx 13°C$ desde el extremo inferior hasta el superior. Exceptuando los meses de junio y julio, cuyos bigotes se extienden a más de 16 $K \approx 16°C$ así como se refleja en el tamaño de la caja.

- Los datos siguen la forma distintiva de un patrón oscilatorio de dinámicas estacionales anuales, en la figura 4.4 muestra el valor medio del LST para cada fecha analizada. Los datos fueron normalizados mediante el centrado respecto a la media global del periodo, permitiendo observar las anomalías térmicas relativas.

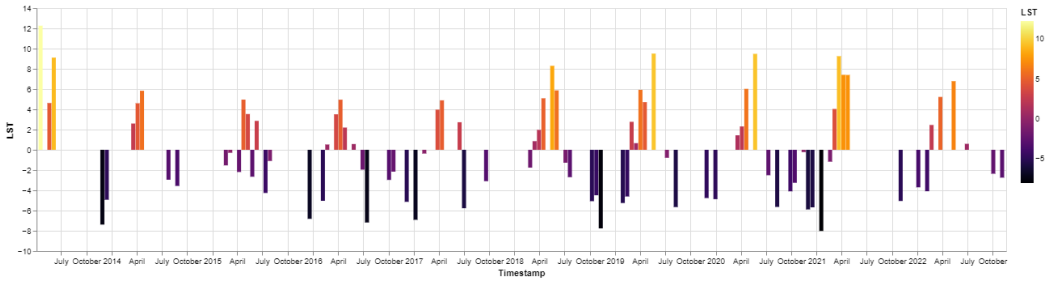


Figura 4.4 Gráfico de barras del valor medio del LST de las imágenes calculadas

4.3. Distribución espacial de la temperatura superficial

Con el objetivo de analizar el comportamiento espacial de la temperatura superficial terrestre (LST), se generó una imagen promedio anual a partir del conjunto de imágenes previamente filtradas bajo el criterio de nubosidad establecido. Esta imagen representa la media térmica por píxel, considerando únicamente aquellos valores válidos en cada escena.

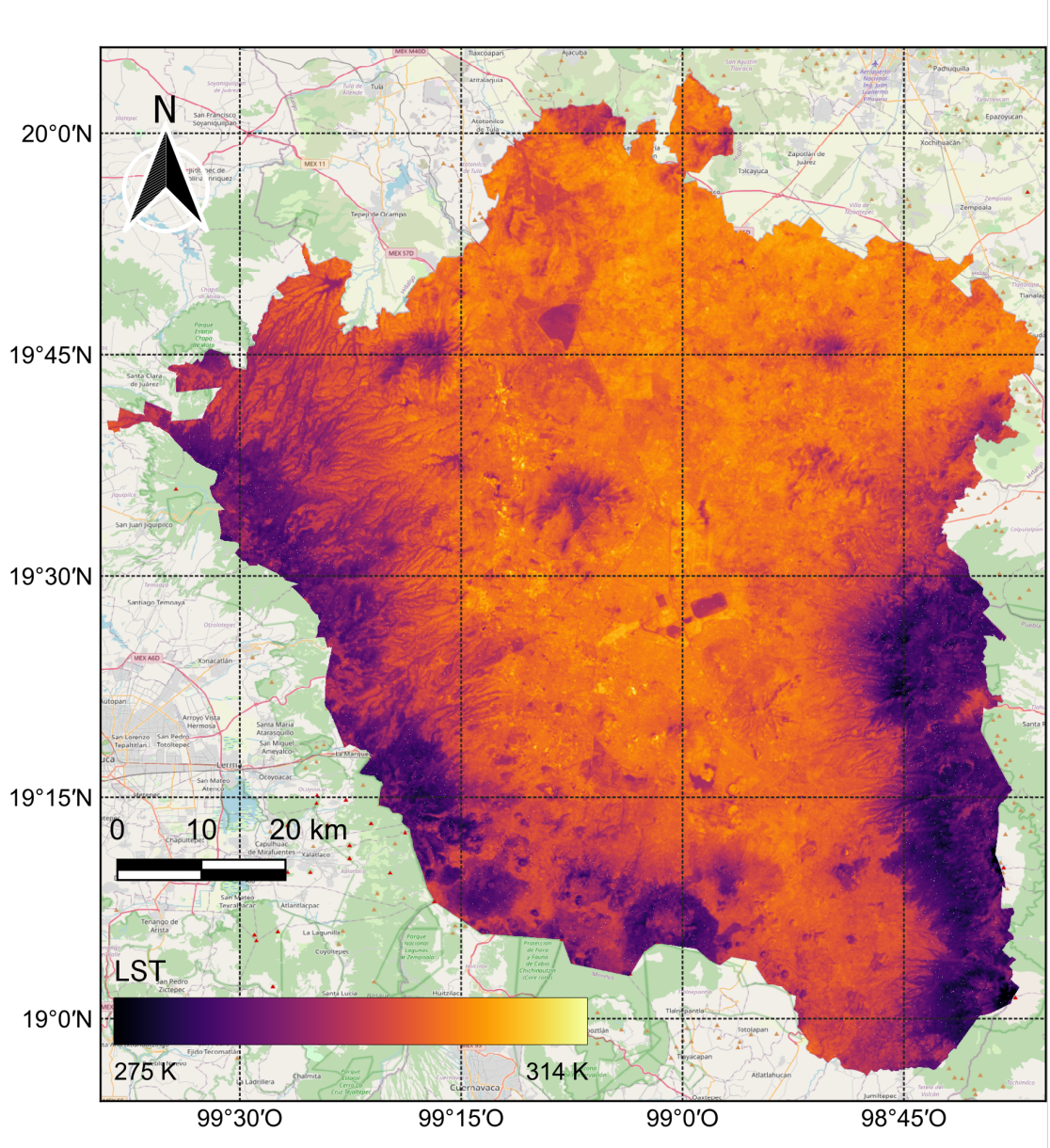


Figura 4.5 Distribución espacial de la temperatura superficial media (LST, K) en la Ciudad de México y zona conurbana. El rango de temperatura observado varía de 275 K a 314 K, equivalente aproximadamente de 2°C a 41°C

La imagen resultante permite identificar la distribución espacial de la temperatura, evidenciando gradientes térmicos bien definidos. Los valores más elevados se concentran en las zonas urbanizadas de alta densidad, mientras que las áreas periféricas y con mayor cobertura vegetal presentan temperaturas relativamente menores.

El rango térmico observado en la imagen media oscila entre 267.367 K (-5.783°C) a los

314.543 K (41.393°C), con una media aproximada de 297 K (23.85°C). Existiendo la temperatura mínima en el pico del volcán Popocatépetl. Las temperaturas menores a los 287 K (13.85°C) (tonalidades moradas) solamente se encuentran en las partes más elevadas de los diferentes cuerpos montañosos. Mientras que los intervalos urbanos a 302 K (28.85°C) y 307 K (33.85°C). Los valores máximos, cercanos a 314 K (40.85°C), se localizan en superficies donde existen grandes extensiones de superficies impermeables, son áreas construidas (asfalto, hormigón, techos) que son superficies con alta absorción y reemisión térmica.
Es importante señalar que algunos píxeles aparecen enmascarados debido a la ausencia de suficientes valores válidos durante el proceso de promediado, lo cual responde al filtrado por nubosidad aplicado previamente. No obstante, estas ausencias no alteran la identificación de los patrones espaciales dominantes.
En términos generales, la imagen promedio anual permite establecer una primera aproximación a la estructura térmica de la Ciudad de México y su zona metropolitana, evidenciando una diferenciación clara entre zonas urbanas consolidadas y áreas con menor grado de urbanización.
La distribución espacial de la temperatura superficial no presenta un comportamiento uniforme. En la porción central del área de estudio se identifica un núcleo térmico de alta intensidad, correspondiente al Aeropuerto Internacional de la Ciudad de México, el cual destaca como uno de los valores más elevados en la imagen promedio.
Este núcleo se encuentra rodeado por regiones con temperaturas relativamente menores que funcionan como contrastes térmicos dentro del sistema metropolitano. Al norte se observa la influencia de la Sierra de Guadalupe; hacia el oriente, la zona lacustre de Texcoco; al sur, el macizo del Tepozteco; y al poniente, la región del Ajusco. Estas áreas, asociadas a mayor cobertura vegetal o características topográficas específicas, presentan valores inferiores en comparación con el núcleo urbano central.
Asimismo, se identifican otros sectores con valores elevados en la porción oriental y en la región norte del área metropolitana, lo que sugiere que el fenómeno térmico no se limita exclusivamente al centro urbano consolidado, sino que responde a patrones de ocupación del suelo y materiales superficiales presentes en distintas zonas.
El análisis comparativo de la colección completa de imágenes generadas para este estudio (véase Apéndice A) permitió identificar patrones espaciales recurrentes a lo largo del periodo de observación. Los cuales son lo siguiente:

- La temperatura es variable en función de la temporada del año. Se encontró que existe un patrón que diferencia los meses cálidos de los meses fríos, ha seguido el patrón que se esperaba de acuerdo a los registros atmosféricos. En la época seca-cálida se

encuentran tanto la mayor cantidad de temperatura como la mayor disparidad entre la región fría y la región cálida ocasional.

  - La región fría corresponde a las zonas donde existe la mayor vegetación, la zona boscosa en las laderas de los diversos cuerpos de montaña que rodean a la zona. Ubicándose en la periferia de la Ciudad de México y su zona conurbana, de Oeste a Este en el Sur, es una zona donde la temperatura se mantiene relativamente estable.
  - La región cálida ocasional, que se da en la región nororiental generada por el uso de suelo agrícola, al ser de riego temporal, su comportamiento es dependiente de las lluvias y de la capacidad del suelo para captar el agua y conservar la humedad.

- La mayoría de nubosidad se encuentra en la región fría. Esto probablemente suceda debido a que es una región montañosa, que actúa como una barrera, evitando que las nubes puedan pasar, así mismo es donde se concentra el uso de suelo forestal.

- En la región norte es donde se encuentran los mayores cambios de temperatura, así como las temperaturas más altas. Las diferencias de temperatura se dan por las características de la región, al ser una combinación entre uso de suelo urbano y uso de agrícola, no existe una planificación territorial y proliferan los asentamientos irregulares con nula presencia de áreas verdes.

- Hay tres cuerpos de agua que son zonas sumidero: el lago Nabor Carrillo, la Laguna de Zumpango y el lago de Regulación Churubusco. Los demás cuerpos de agua no figuran a esta escala. Xochimilco se comporta como una hoya.

- Hay 3 islotes de calor de considerable tamaño generados por los 3 aeropuertos que tiene la Ciudad de México y su zona conurbana. La isla de calor del AICM ha sido constante. La isla de calor correspondiente al NAICM resalta del 2016-02-22 al 2019-05-21; anteriormente era una zona de transición y posteriormente se fusionó con las regiones aledañas. La tercera corresponde con la región del AIFA, aquí la Isla de Calor ha sido constante.

- Existe una clase de collado que se da en la zona oriente de la CDMX en la región de Texcoco. Entre los dos aeropuertos y los cuerpos de agua.

Regionalización Térmica

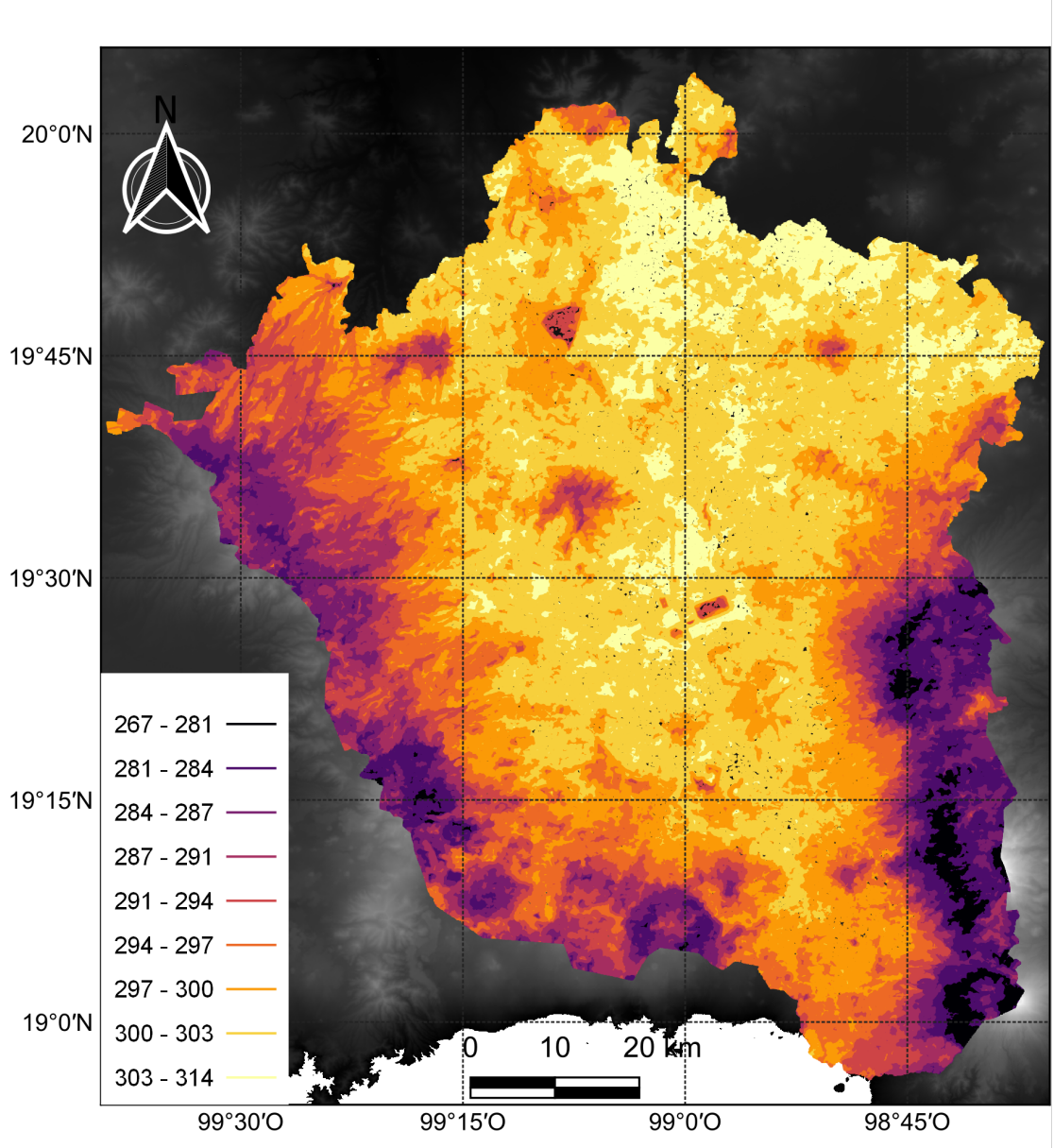


Figura 4.6 Distribución espacial clasificada de la temperatura superficial media (LST, K) en la Ciudad de México y zona conurbana. Para resaltar los contrastes térmicos, los valores fueron agrupados en intervalos irregulares. El rango total de temperatura varía de 267 K a 314 K, equivalentes aproximadamente a $-6°$C y $41°$C respectivamente.

En la figura 4.6 se muestra la distribución espacial de la temperatura superficial media tras el uso del método de clasificación Natural Breaks (Jenks), el cual agrupa los valores

minimizando la varianza interna de cada clase y maximizando la diferencia entre clases. Se observa un patrón característico de la isla de calor urbana, donde las mayores temperaturas se concentran en áreas con mayor densidad urbana, mientras que los valores más bajos están en las zonas periféricas y áreas con mayor cobertura vegetal o elevación del terreno.

Con base en el patrón térmico observado en la Figura 4.6, el área de estudio fue subdividida en cinco zonas de análisis: Centro, Norte, Sur, Oriente y Poniente. Esta delimitación se realizó considerando su ubicación geográfica y los límites administrativos de los municipios y alcaldías que conforman la Ciudad de México y su zona conurbana, con el objetivo de facilitar la comparación espacial de los contrastes térmicos entre áreas con características similares.

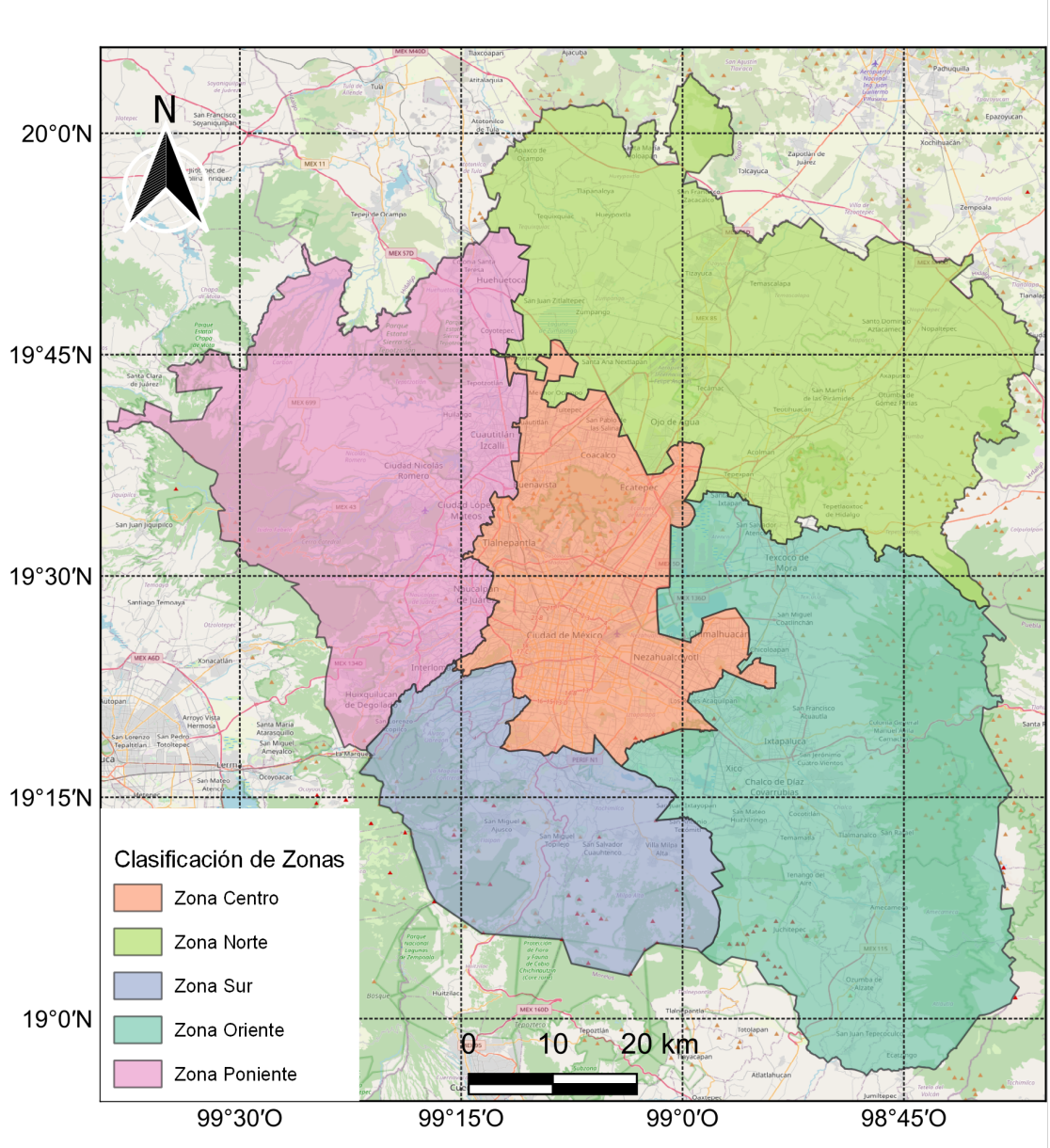


Figura 4.7 Mapa para representar las diferentes zonas bajo las cuales se ha dividido el área de estudio.

Zona Centro

Como característica principal es que se concentra la mayor parte de urbanización dentro de la Ciudad de México y su zona conurbana, por lo mismo es donde se encuentran las temperaturas más altas a nivel urbano. Las alcaldías y municipios que conforman esta zona son:

Ciudad de México Azcapotzalco, Coyoacán, Gustavo A. Madero, Iztacalco, Iztapalapa, Benito Juárez, Cuauhtémoc, Miguel Hidalgo y Venustiano Carranza.

Estado de México Coacalco de Berriozábal, Cuautitlán, Ecatepec de Morelos, Nezahualcóyotl, Tlalnepantla de Baz, Tultepec y Tultitlán.

Si bien es una zona urbana, por la planeación urbana, se encontrarán los picos en áreas de características industriales, estos llegan a esparcirse, afectando las áreas adyacentes. Lo que lo vuelve más interesante es que estas islas no son aisladas y llegan a unirse formando plataformas continentales.

- Hay una isla generada por la Central de Abastos de la Ciudad de México, el calor se extiende hasta 2 km a la redonda. Con una temperatura máxima de 308.81 K (35.66°C).
- Existe una isla focalizada en el Complejo Industrial Tecnológico Iztapalapa. Es suavizada por las áreas verdes cercanas a la zona, encabezadas por los parques: Cuitlahuác, el Parque Recreativo Santa Cruz Meyehualco y el Parque Nacional Cerro de la Estrella. Tiene una temperatura máxima de 315.46 K (42.31°C).
- En lo que se conoce como el centro histórico existen tienen temperaturas muy altas, superiores a los 304 K (30.85°C), con el pico más alto con un valor de 306.85 K (33.7°C) ubicado en el Mercado de Mixcalco.
- Hay una plataforma que tiene una extensión de 20 km, siendo el centro Ciudad Neza. Extendiéndose a los municipios de Chimalhuacán y La Paz, las colonias Vicente Guerrero y Agrícola Pantitlán. Puede pasar desapercibida por su proximidad con el AICM. Esta plataforma es reprimida por los diferentes cuerpos montañosos que se encuentran al sur y en el norte son los cuerpos de agua. Con una temperatura máxima de 307.09 K.
- En la colonia Nueva Industrial Vallejo y se extiende hasta Universidad Autónoma Metropolitana Azcapotzalco. Con temperatura media de 305 K (31.85°C).
- Depósito de Evaporación Solar “El Caracol”, con una temperatura máxima de 307 K (33.85°C).
- Las islas de Tepotzotlán, Subestación Victoria y Tlalnepantla se analizan en la zona Poniente porque comparten características que hacen más fácil su análisis.

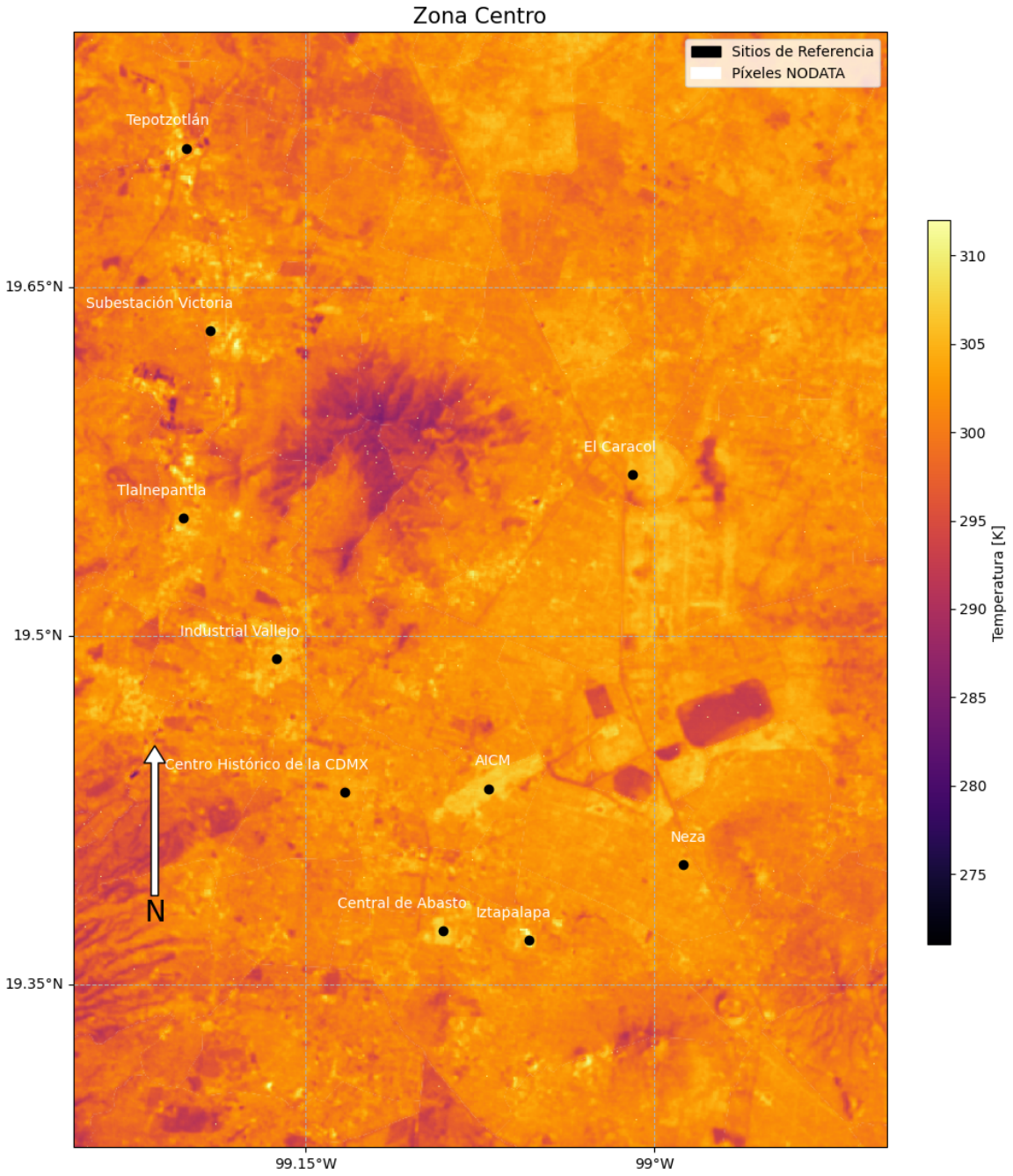


Figura 4.8 Mapa que representa los diversos puntos de interés que existen en la zona centro. La temperatura mínimas medias registradas para esta zona son de 284.97 $K \approx 12°$C en los picos de la Sierra de Guadalupe

Zona Oriente

Esta zona tiene como característica ser una zona de transición, al tener urbanización de baja densidad (edificios menores a 5 pisos) así como estar en las faldas de los volcanes Popocatépetl e Iztaccíhuatl.
La alcaldía y los municipios que conforman esta zona son:

Ciudad de México Tláhuac.

Estado de México Amecameca, Atenco, Atlautla, Ayapango, Cocotitlán, Chalco, Chicoloapan, Ecatzingo, Ixtapaluca, Juchitepec, Ozumba, La Paz, Temamatla, Tenango del Aire, Tepetlixpa, Texcoco, Texcoco, Tlalmanalco y Valle de Chalco Solidaridad

Dentro de esta zona la orografía es parte fundamental al estar rodeada por la parte más alta del relieve de la Ciudad de México y su zona conurbana (los volcanes), y las temperaturas replican ese modelo del terreno montañoso y accidentado. Es donde se encuentran las temperaturas más bajas.

- El punto más cálido de esta zona se encuentra en la plancha de concreto derivada del cancelado Nuevo Aeropuerto Internacional de la Ciudad de México (NAICM) con 308.15 K (35°C), y una media en el área destinada de 306 K (32.85°C). Está termina siendo ahogada por los cuerpos de agua que existen a su alrededor así como el área de reserva natural de Atenco, principalmente por el lago Nabor Carrillo que tiene una temperatura mínima de 294.44 K (21.29°C) (un delta de 13 grados). Sin embargo, la zona inferior a este (la ribera del lago que es suelo desnudo, presenta una temperatura media de 305 K).

- Texcoco es una ICU que tiene características peculiares. Solamente en el centro tiene una temperatura mayor a 305 K (31.85°C), mientras la periferia alberga temperaturas de hasta los 296 K (22.85°C). Creando un terreno escarpado, en la zona. Esto puede ser el equilibrio entre la densidad poblacional, los campos de cultivo y las áreas verdes.

- De Valle de Chalco (temperatura máxima de 307.22 K (34.07°C)) a Chalco (temperatura máxima de 308.69 K (35.54°C)) con una extensión lineal de 12 km. Por sus condiciones, presenta un buen lugar para describir y analizar el fenómeno de las Islas de Calor Urbana en la periferia. Tiene una distribución ideal debido a que al estar rodeada por naturaleza, la "aíslan" haciendo que se concentre el calor en la zona.

- En los poblados de Ozumba-Tepetlixpa se encuentran un fenómeno parecido al anterior pero en una escala inferior. Sin embargo, lo interesante sucede en la parte superior de

estos poblados, en la Plaza de Cobro – No. 192 Ozumba con temperaturas superiores a los 300 K (26.85°C), mientras en estos poblados la media es de 296 K (22.85°C).

- Amecameca no tiene un efecto marcado derivado de una ICU, sino derivado de la agricultura que se practica en la zona. Creando un delta de hasta 4 K (4°C) con el área suburbana y el área agrícola.
- Río Frío es una zona de transición. Al estar en medio de montañas, destaca la manera en la cual se extiende sobre la zona por el contraste de los 297 K (24.85°C) que tiene de promedio la zona con los 284 K (10.85°C) que tiene el área que lo rodea.
- Las temperaturas más bajas se encuentran en los picos de los volcanes Popocatépetl 270. 33 K (-2.82°C) e Iztaccíhuatl 271.76 K (-1.39°C).

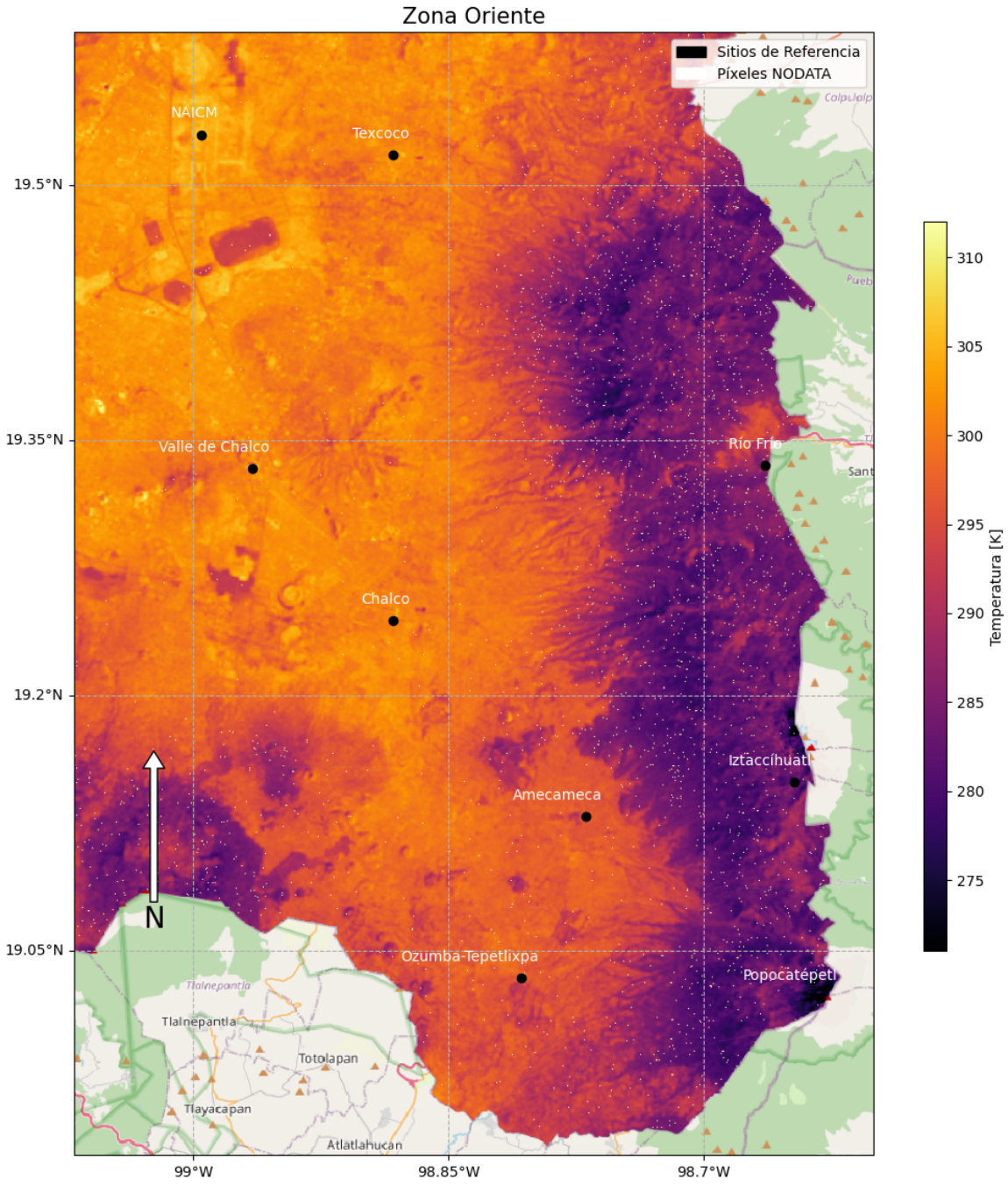


Figura 4.9 Mapa que representa los diversos puntos de interés que existen en la zona oriente. La temperatura mínimas medias registradas para esta zona son de 270. 33 $K \approx -3°$C en el pico del volcán Popocatépetl.

Zona Sur

Es una zona que tiene muy marcadas las diferencias entre las áreas urbanas (presenta alta densidad, y principalmente baja densidad) y las áreas naturales que la rodean como lo son el Parque Nacional Desierto de los Leones, el Parque Nacional Cumbres del Ajusco y el Parque Nacional El Tepozteco.
Las alcaldías que conforman esta zona son:

Ciudad de México Cuajimalpa de Morelos, La Magdalena Contreras, Milpa Alta, Álvaro Obregón, Tlalpan, Coyoacán [1] y Xochimilco.

En general la zona se presenta como una temperatura variante produciendo una sensación de terreno escarpado. Los picos existen en las grandes infraestructuras recreativas (plazas comerciales y estadios). Destaca una plataforma que abarca desde el Pedregal de Santo Domingo, hasta Paseos de Taxqueña.

- Hay un pico menor ubicado en la Plaza San Jerónimo, contrasta al tener una temperatura de hasta 306.09 K (32.94°C) mientras la zona aledaña tiene una media de 300 K (26.85°C).
- Existe una isla cuya temperatura más alta es de 308.47 K (35.32°C) ubicada en la plaza Exhibimex.
- Una plataforma tiene el centro en el Pedregal de Santo Domingo y llega a Coapa, extendiéndose hasta 10 km a la redonda. Siendo mitigada por Ciudad Universitaria, el Heroico Colegio Militar y principalmente por el Parque Nacional Bosque de Tlalpan y Xochimilco. Concentra dos picos aislados: en la plaza Gran Sur con 313.07 K (39.92°C) de máxima y en el Estadio Azteca el pico es de 309.05 K (35.9°C). Tiene una media de 302 K (28.85°C).
- La temperatura más baja se encuentra en los puntos más altos del Parque Nacional Cumbres del Ajusco, con sumideros de hasta 279 K (5.85°C) y media de 284 K (10.85°C).
- Las áreas suburbanas de la zona presentan una temperatura media de 298 K (24.85°C), esto incluye tanto la parte donde existen localidades urbanas, carreteras así como las faldas de los diversos cuerpos montañosos que hay en la zona.

[1] También se pone aquí a esta alcaldía debido a que comparte características con la descripción de los casos de estudio

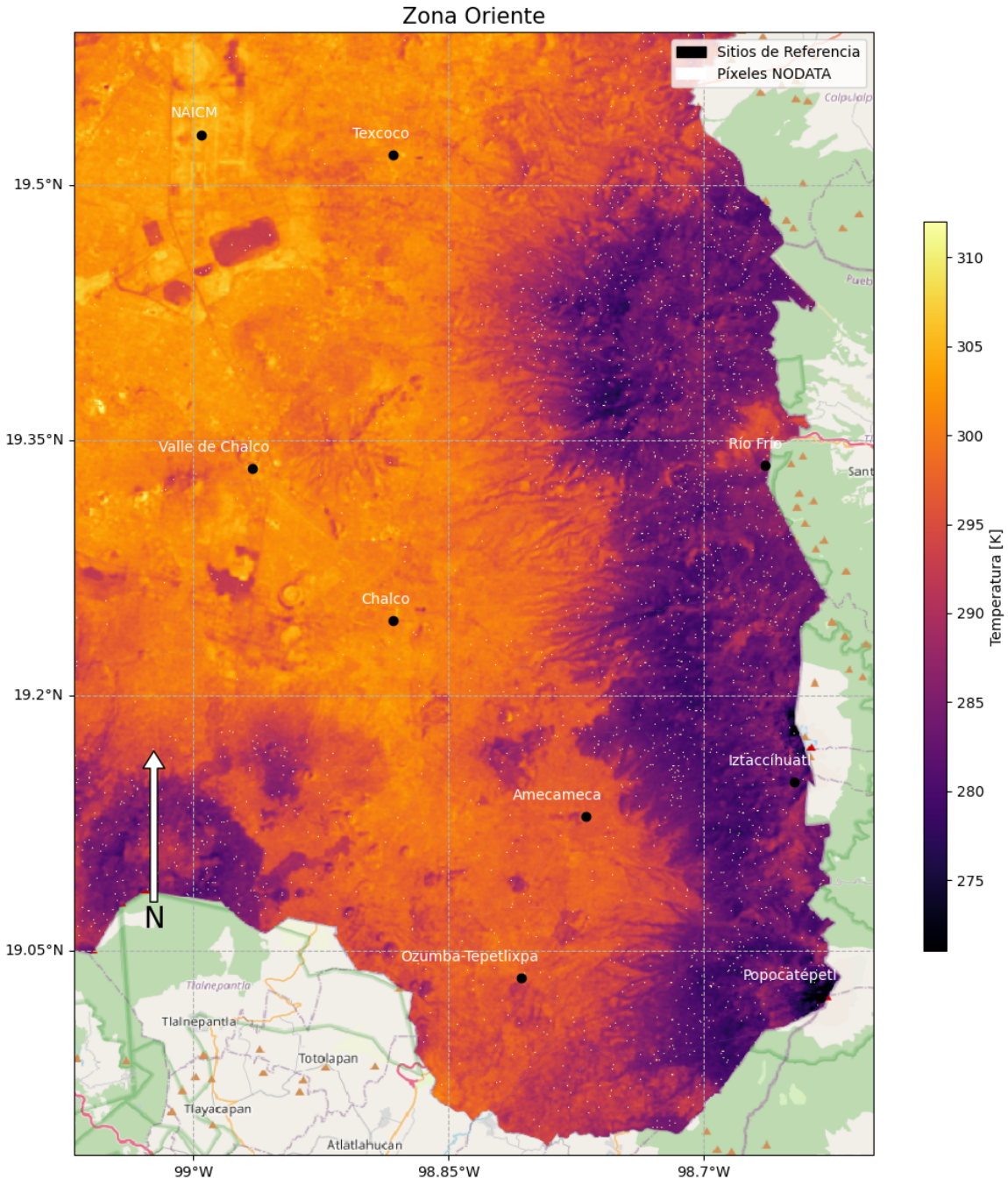


Figura 4.10 Mapa que representa los diversos puntos de interés que existen en la zona oriente. La temperatura mínimas medias registradas para esta zona son de 278.14 K (4.99°C) en el cerro Santo Tomás.

Zona Poniente

Es una zona que tiene áreas urbanas, suburbanas, y principalmente grandes áreas industriales, así como áreas naturales en los bordes que tienen un nivel considerable de erosión. Los municipios que conforman esta zona son:

Estado de México Atizapán de Zaragoza, Coyotepec, Huehuetoca, Huixquilucan de Degollado, Isidro Fabela, Jilotzingo, Naucalpan de Juárez, Nicolás Romero, Tepotzotlán, Villa del Carbón y Cuautitlán.

Hay una especie de cordillera que se forma en zonas adyacentes a la carretera de la Carretera Federal 57 de Cuota (la que conecta con Querétaro) y con una extensión lineal de 28 km y una perpendicular desde 1 km hasta los 3 en su tramo más industrializado. Así mismo, en el extremo poniente de esta zona se encuentra una cadena de "depresiones" en el área de montaña del Parque Nacional Cumbres Sierra Nevada.

- El primer punto de interés de esta cordillera está en el municipio de Naucalpan de Juárez, cerca del Palacio Municipal, está el punto más caliente de esta zona con 309.28 K (36.13°C). Tiene una extensión de más de 4 km con una media de 302 K (28.85°C).

- El segundo punto de interés está en el municipio de Tlalnepantla en la colonia Centro Industrial Tlalnepantla y otro en la colonia Industrial Barrientos, con una temperatura media de 304 K (30.85°C) y la máxima de 312.53 K (39.85°C). Tiene una extensión de 5 km con una media de 303 K (29.85°C).

- El tercer punto se encuentra ubicada en las naves industriales que existen en la zona de Tultitlán y Cuautitlán Izcalli, siendo su pico la subestación Victoria con 314.19 K (41.04°C). Tiene una extensión de 5 km con una media de 303 K (29.85°C).

- El último pico está ubicado en la zona cercana a la Caseta Tepotzotlán, en la zona de alto flujo vial que se forma por los intercambios de la autopista y las carreteras, tiene una extensión de 5 km, dos picos con temperaturas máximas de 313 K (39.85°C) y una temperatura media de 304 K (30.85°C).

- La temperatura más baja registradas se encuentra en la cadena de colinas presente en el Parque Nacional Cumbres Sierra Nevada con una media de 285 K (11.85°C). Tiene una extensión de 28.4 km.

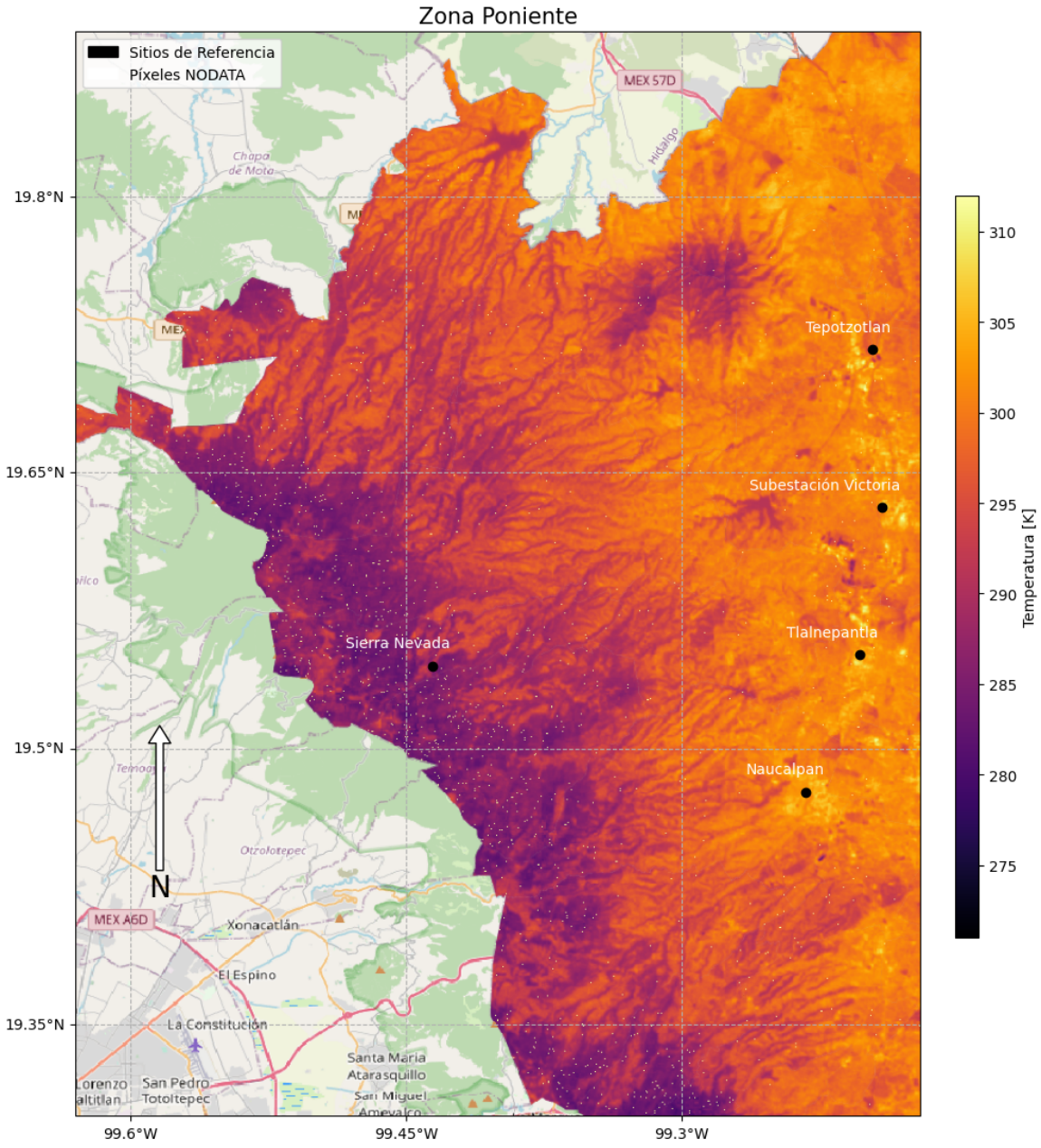


Figura 4.11 Mapa que representa los diversos puntos de interés que existen en la zona poniente. Si bien la zona presenta erosión del suelo (lo que provoca un efecto similar a ver una cuenca hidrográfica) se puede apreciar a simple vista el impacto de las actividades antropogénicas sobre la temperatura superficial.

Zona Norte

Es una zona que en su gran mayoría concentra áreas de uso agrícola, no tiene a diferencia de las anteriores áreas protegidas y las áreas se podrían clasificar como áreas suburbanas de

baja densidad (construcciones menores a 10 pisos). Los municipios que conforman esta zona son:

Estado de México Tonanitla, Zumpango, Tezoyuca, Tequixquiac, Tepetlaoxtoc, Teotihuacán, Teoloyucan, Temascalapa, Tecimac, San Martín de las Pirámides, Otumba, Nopaltepec, Nextlalpan, Melchor Ocampo, Jaltenco, Hueypoxtla, Chiconcuac, Chiautla, Axapusco, Apaxco y Acolman

Hidalgo Tizayuca

En algunos casos esta zona podría considerarse un valle con una de media de 302 K, es una zona que no presenta cambios abruptos. Lo que lo vuelve interesante es que las temperaturas máximas medias existen en el uso de suelo agrícola o en el suelo desnudo/erosionado.

- Hay variaciones en la parte superior del complejo militar y aeropuerto de Santa Lucía, es un pequeño monte de 305 K (31.85°C) media con un radio de 6 km y el pico existe de 309.77 K (36.62°C) de máximo.
- En el poblado de Zumpango, cerca del Centro Universitario UAEM Zumpango (5 km al Noreste de Zumpango) con un pico de 308.37 K (35.22°C) y una radio de 3 km, con una media de 306 K (32.85°C). Por su cercanía con Santa Lucía, podría parecer que están fusionadas
- Cerca del poblado de Santa María Ajoloapan, sucede otra meseta con una media de 305 K (31.85°C) y un radio de 5 km. No se extiende al poblado
- Cerca del poblado de Ixtlahuaca de Cuauhtémoc, sucede otra meseta con una media de 305 K (31.85°C) y un radio de 4 km. No se extiende al poblado.
- Hay una depresión de media de 292 K (18.85°C) en el lago de Zumpango.
- La temperatura más baja se encuentra en el pico del Cerro Gordo con 287.77 K (14.62°C). Creando una depresión que se extiende hasta 5 km en radio. Haciendo que la zona tenga una temperatura media de 295 K (21.85°C).

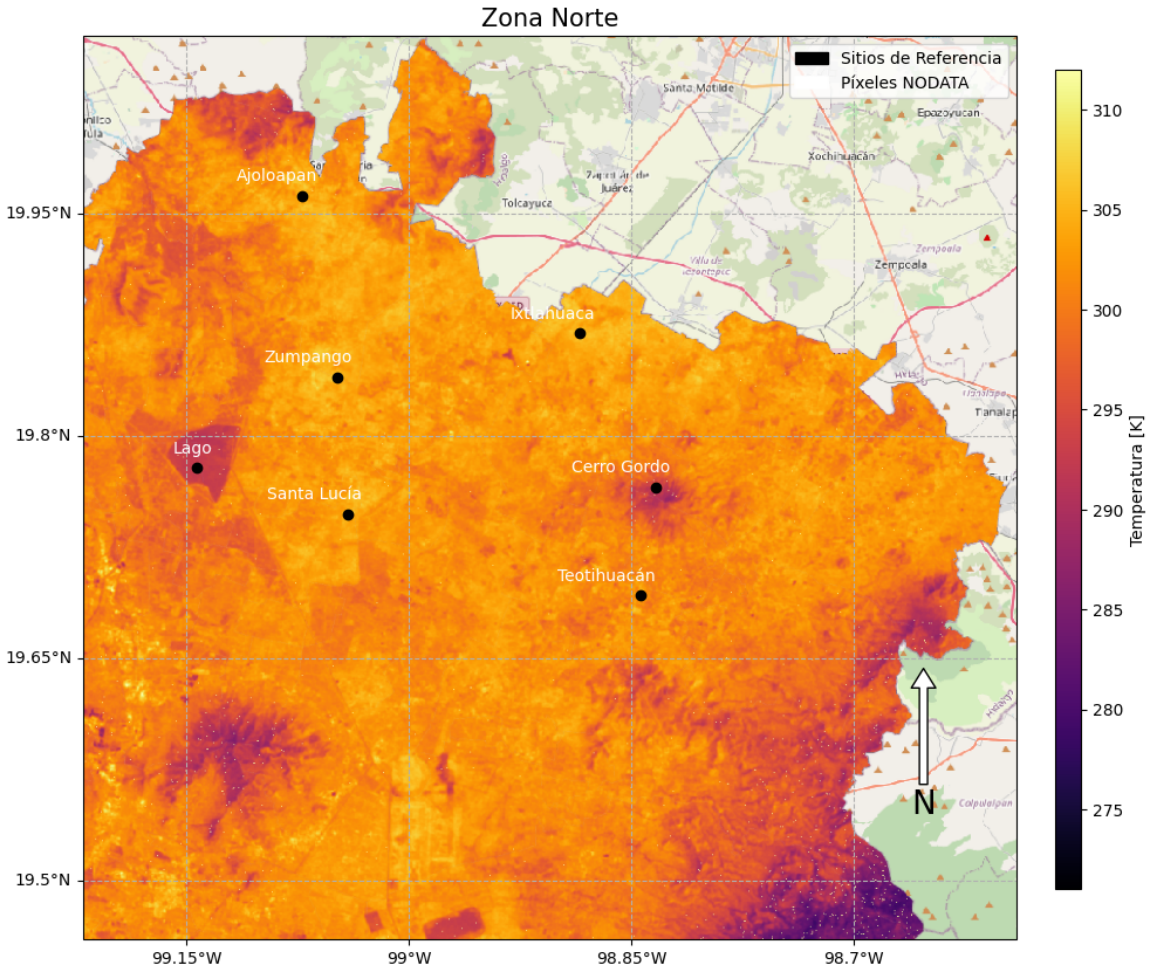


Figura 4.12 Mapa que representa los diversos puntos de interés que existen en la zona norte. La zona no es tan útil considerando que no hay ICU por la densidad poblacional y las temperaturas más altas se encuentran en áreas donde el suelo está muy erosionado.

Capítulo 5

Conclusiones

Este proyecto ha representado un esfuerzo significativo en el descubrimiento y análisis de peculiaridades fundamentales. La contribución esencial de Ermida y sus colegas [9] ha servido como cimiento fundamental para este estudio, especialmente en la aplicación de Google Earth Engine (GEE). Asimismo, se destaca la influencia del algoritmo propuesto por Jiménez-Muñoz [19], que ha desempeñado un papel crucial en nuestras investigaciones al lograrse adaptar de manera exitosa a GEE.

La asombrosa utilidad de Google Earth Engine (GEE) ha dejado una profunda impresión en mí, especialmente en lo que respecta a la facilidad de acceso a la información. En contraste con los desafíos comunes asociados con trabajos de Percepción Remota, como la necesidad de vastos recursos de almacenamiento y capacidad de procesamiento, GEE ha demostrado ser excepcionalmente eficiente. Su acceso a un extenso conjunto de datos disponible para los usuarios y la capacidad de ejecución en la nube, desde cualquier dispositivo con conexión a Internet, han simplificado considerablemente la obtención y manipulación de información.

No obstante, es importante señalar que, en ciertas instancias, nos enfrentamos a limitaciones de procesamiento al utilizar tanto la aplicación GEE y aplicaciones externas como Colab, superando en algunas ocasiones los límites establecidos por Google. Estas experiencias resaltan la necesidad continua de abordar y superar desafíos técnicos para aprovechar al máximo las capacidades de estas herramientas innovadoras.

Se ha cumplido de manera satisfactoria el objetivo general planteado al inicio de este trabajo. A través del análisis de diversas imágenes, se logró identificar la temperatura superficial, revelando una correlación estrecha con los factores estacionales. La extensa cobertura geográfica de la Ciudad de México y su zona conurbana permitió observar el fenómeno de las islas de calor en diversas regiones con usos del suelo variados, abarcando áreas urbanas, suburbanas y naturales.

Hablar de la elaboración del algoritmo es una tarea que se aborda con relativa facilidad, pero su ejecución presenta desafíos significativos. Nuestro objetivo fue concebir una metodología que no solo fuera comprensible en cierta medida, sino también fácil de replicar. Esta ambiciosa meta se logró al aprovechar y emplear diversas funciones, diseñadas de manera que fueran accesibles para una amplia gama de usuarios.

La estimación de la temperatura superficial a lo largo de la última década reveló resultados que, no se anticipaban originalmente. Inicialmente se esperaban hallar patrones fácilmente analizables donde se pudiese identificar, interpretar y analizar los datos. Sin embargo, no se obtuvieron datos con estas características porque no se logró observar una disminución drástica en el tamaño de las islas de calor en la mancha urbana, y su desarrollo adoptó formas caprichosas que desafiaron las expectativas tradicionales descritas en la bibliografía consultada.

En lugar de encontrar un efecto reducido por el entorno, se manifestó una similitud con una plataforma continental, que se le atribuye el adjetivo cálida, cuya expansión variaba según la temporada. Estas "plataformas" presentaban características inesperadas, acentuandose en grandes extensiones de concreto, techos de metal e incluso áreas aparentemente no afectadas a gran escala por actividades antropogénicas. Las ICU que se alinearon con la descripción teórica se localizaron en las zonas suburbanas de la región oriental del Estado de México. Este fenómeno resulta previsible considerando que, a pesar del desarrollo urbano agresivo, estas áreas siguen envueltas por espacios naturales en los distintos cuerpos montañosos presentes en el Valle del Anáhuac.
En comparación con otros estudios realizados en la misma zona, se identificaron notables y significativas discrepancias probablemente derivado de la magnitud y distribución de la LST. Esta discrepancia no permite establecer concluyentemente la existencia de una Isla de Calor Urbana bajo las condiciones y periodo analizados. En primer lugar, no muestran la Isla de Calor Urbana ICU centralizada claramente delimitada, típicamente ubicada en el centro de la Ciudad de México, observándose, en cambio, múltiples patrones térmicos heterogéneos y zonas con temperaturas superficiales relativamente altas distribuidas en distintos sectores del área de estudio. Esta variación podría atribuirse a tres posibles causas:

- Los estudios anteriores están limitados al análisis de un área considerablemente más pequeñas en comparación con la extensión abarcada en este trabajo.
- Podrían emplear metodologías de estudio diferentes, siendo crucial recordar que esta investigación se centra en las islas de calor superficiales en contraste con estudios previos que se han enfocado en aspectos atmosféricos.
- La diversidad en los usos del suelo puede explicar las diferencias observadas. En este estudio, los picos de las Islas de Calor se acentuaron en áreas donde prevalecen características similares, como el tipo de construcción, acabados y características del suelo, extendiéndose a lo largo de grandes extensiones.

Cada método de estudio posee sus propias ventajas y desventajas. En términos generales, se podría resumir que la búsqueda de una mayor precisión conlleva la compensación de una reducida ventana de tiempo disponible para el análisis.
Una hipótesis inicial, planteaba encontrar una mayor temperatura en la zona poniente, derivada del cruce de los rayos solares al amanecer hasta el momento de la captura de la imagen (considerando que el sol se eleva por el oriente), fue refutada por los resultados del estudio. Sorprendentemente, se demostró que la región oriental es la que presenta temperaturas más elevadas.

En cuanto a la influencia de la nubosidad en la temperatura, se plantea la pregunta de si esta variación se debe a condiciones atmosféricas, como una mayor humedad en el suelo, o si, al medir la temperatura superficial, la sombra proporcionada por las nubes evita el calentamiento directo de la superficie. Aunque se requiere una evaluación más detallada, los resultados sugieren una posible relación entre la temperatura superficial y las condiciones atmosféricas.
Continuar con esta línea de investigación sería sumamente enriquecedor. Agregar más factores al estudio permitiría una comprensión más completa de los fenómenos observados. Aquí están algunas sugerencias para expandir la investigación:

Uso de datos topográficos Esto permitiría evaluar la influencia de la altitud en el desarrollo y distribución de las Islas de Calor. La topografía puede desempeñar un papel crucial en la retención del calor y la formación de patrones térmicos.

Información precisa de Uso de Suelo Integrar mapas detallados de uso de suelo proporcionaría una visión más precisa de cómo la ocupación del espacio, especialmente en áreas industriales en lugar de urbanas, afecta la formación de las Islas de Calor. Esto podría ayudar a distinguir entre áreas con diferentes características de construcción y materiales.

Uso de cartas geológicas y de tipo de suelo Esto sería valioso para comprender en qué medida estas variables afectan la creación, contención y expansión de las Islas de Calor. Diferentes tipos de suelo tienen propiedades térmicas distintas que podrían influir en los patrones de temperatura.

Desarrollo de Módulos Adicionales Explorar la posibilidad de desarrollar módulos adicionales, especialmente uno que facilite la identificación del material registrado por el sensor, sería beneficioso. Esto permitiría un análisis más detallado de las características de la superficie y su contribución a las variaciones de temperatura.

Estas adiciones fortalecerían la investigación al proporcionar una visión más completa y detallada de los factores que influyen en la formación de las Islas de Calor.
Es relevante destacar los desafíos significativos que surgieron durante el desarrollo de la investigación, siendo el principal obstáculo la identificación de las Islas de Calor. La zona norte, en particular, planteó dificultades considerables debido a la presencia ocasional de islas de calor cálidas. Además, se encontraron complicaciones relacionadas con las herramientas de análisis utilizadas. Aunque Google Earth Engine (GEE) ofrece funciones para evaluar, visualizar y manipular datos, la elección de herramientas más robustas, aunque beneficiosas en ciertos aspectos, generó inconvenientes al cambiar entre plataformas y códigos. Estos

desafíos subrayan la importancia de la selección cuidadosa de herramientas y metodologías para garantizar la eficiencia y coherencia en el análisis de datos.

La aplicación de filtros posteriores para descartar la presencia de nubosidad demostró ser una decisión acertada. En el proceso convencional, se descartan las imágenes con nubes al principio del análisis, antes de comenzar el trabajo. Sin embargo, al aplicar este enfoque, donde se descarta en función de la zona, se obtuvo un resultado más robusto. En las primeras pruebas realizadas en 2015, bajo el parámetro de descartar en relación con la nubosidad, se redujo a solo 6 imágenes. En cambio, al seguir el modelo de descartar en función de la zona, el resultado fue de 10 imágenes. Este logro fue posible gracias a las capacidades de cómputo eficientes de Google Earth Engine (GEE).

Mi última observación sería reducir el tamaño de las áreas de estudio para analizar y registrar con mayor detalle las anomalías. Al enfocarse en áreas más pequeñas, se puede obtener una visión más precisa de las variaciones y peculiaridades dentro de la región de estudio. La identificación de píxeles con altos valores en zonas templadas, como construcciones en parques o fábricas/almacenes identificados por techos de metal en áreas residenciales, proporcionaría una comprensión más detallada de las fuentes de calor específicas.

Esto debería crear una investigación más a fondo para entender las razones detrás de las regiones cálidas ocasionales, ya sea debido a condiciones propias de la región, actividades agrícolas u otros factores, lo cuál permitiría una comprensión más completa de las dinámicas térmicas en estas áreas particulares. Este enfoque más detallado contribuiría a una interpretación más precisa de las Islas de Calor y sus causantes en la región de estudio.

Bibliografía

[1] Ballinas Oseguera, M. d. J. (2011). *Mitigación de la isla de calor urbana : estudio de caso de la Zona Metropolitana de la Ciudad de México*. PhD thesis, Universidad Nacional Autónoma de México.

[2] Carlson, T. N. and Ripley, D. A. (1997). On the relation between NDVI, fractional vegetation cover, and leaf area index. *Remote Sensing of Environment*, 62(3):241–252.

[3] Caselles, E., Abad, F. J., Valor, E., and Caselles, V. (2011). *Automatic Generation of Land Surface Emissivity Maps*. IntechOpen. Publication Title: Climate Change - Research and Technology for Adaptation and Mitigation.

[4] Caselles, V., Valor, E., Coll, C., and Rubio, E. (1997). Thermal band selection for the PRISM instrument: 1. Analysis of emissivity-temperature separation algorithms. *Journal of Geophysical Research: Atmospheres*, 102(D10):11145–11164. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/97JD00344.

[5] Coakley, J. (2003). REFLECTANCE AND ALBEDO, SURFACE. In *Encyclopedia of Atmospheric Sciences*, pages 1914–1923. Elsevier.

[6] de Estadística y Geografía (INEGI), I. N. (2010). Manual de cartografía censal.

[7] Dutta, B., Giunchiglia, F., and Maltese, V. (2011). A Facet-Based Methodology for Geo-Spatial Modeling. In Claramunt, C., Levashkin, S., and Bertolotto, M., editors, *GeoSpatial Semantics*, Lecture Notes in Computer Science, pages 133–150, Berlin, Heidelberg. Springer.

[8] Elachi, C. and Van Zyl, J. (2006). *Introduction to the physics and techniques of remote sensing*. Wiley series in remote sensing. Wiley-Interscience, Hoboken, N.J, 2nd ed edition. OCLC: ocm61309501.

[9] Ermida, S. L., Soares, P., Mantas, V., Göttsche, F.-M., and Trigo, I. F. (2020). Google Earth Engine Open-Source Code for Land Surface Temperature Estimation from the Landsat Series. *Remote Sensing*, 12(9):1471. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[10] EUMETSAT (2020). Who we are.

[11] Freitas, S., Trigo, I., Macedo, J., Barroso, C., Silva, R., and Perdigao, R. (2012). Land surface temperature from multiple geostationary satellites. *International Journal of Remote Sensing*, 34:1–18.

[12] Google Developers (2021a). Get Started with Earth Engine.

[13] Google Developers (2021b). USGS Landsat 8 Collection 2 Tier 1 and Real-Time data TOA Reflectance.

[14] Google Developers (2021c). USGS Landsat 8 Level 2, Collection 2, Tier 1.

[15] Guerrero Elemen, C. A., Benítez Matías, J., Vielma Orozco, E., and Covarrubias Gaitán, F. (2018). *Delimitación de las zonas metropolitanas de México 2015*. Instituto Nacional de Estadística y Geografía, 1 edition.

[16] Instituto Nacional de Estadística de Geografía e Informática, Secretaría del Medio Ambiente, and Gobierno del Distrito Federal (2005). *Estadísticas del medio ambiente del Distrito Federal y Zona Metropolitana 2002*. INEGI ; Secretaría del Medio Ambiente, Gobierno del Distrito Federal, Aguascalientes, México; México D.F. OCLC: 689486398.

[17] Instituto Nacional de Estadística y Geografía (INEGI) (2019). Mexico - Características del Entorno Urbano y Localidad.

[18] ISO/IEC/IEEE (2011). ISO/IEC/IEEE Systems and software engineering – Architecture description. *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*, pages 1–46. Conference Name: ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000).

[19] Jimenez-Munoz, J.-C. and Sobrino, J. A. (2008). Split-Window Coefficients for Land Surface Temperature Retrieval From Low-Resolution Thermal Infrared Sensors. *IEEE Geoscience and Remote Sensing Letters*, 5(4):806–809.

[20] Jiménez-Muñoz, J. C., Sobrino, J. A., Plaza, A., Guanter, L., Moreno, J., and Martinez, P. (2009). Comparison Between Fractional Vegetation Cover Retrievals from Vegetation Indices and Spectral Mixture Analysis: Case Study of PROBA/CHRIS Data Over an Agricultural Area. *Sensors*, 9(2):768–793. Number: 2 Publisher: Molecular Diversity Preservation International.

[21] Jiménez-Muñoz, J. C., Sobrino, J. A., Skoković, D., Mattar, C., and Cristóbal, J. (2014). Land Surface Temperature Retrieval Methods From Landsat-8 Thermal Infrared Sensor Data. *IEEE Geoscience and Remote Sensing Letters*, 11(10):1840–1843. Conference Name: IEEE Geoscience and Remote Sensing Letters.

[22] Jiménez Uribe, Rodrigo, Romo Viramonte, Raúl, Barrón López, Eric Alan, Benítez Villegas, Israel, Villasana Ocampo, Diana, Torres Vázquez, Héctor, Lomelí Covarrubias, Álvaro, Aguilar García, Francisco Javier, Bolaños Juárez, Ana Laura, Aréizaga Gutiérrez, Omar Alejandro, Tuyub Poot, Marcos Daniel, Ramírez Cañas, Xochitl Berenice, Olaya Contreras, Hiram, Rodríguez Abreu, Mauricio, Esparza Ríos, Luis Gerardo, Schmitz Basañez, Carole Odette, Martínez Topete, Claudio, López Condado, Pedro Alain, and Esparza Juárez, Armando (2023). *Metrópolis de México 2020*. Secretaría de Desarrollo Agrario, Territorial y Urbano, 1 edition.

[23] JPL, N. (2014). ASTER Global Emissivity Dataset, 100-meter, HDF5. Type: dataset.

[24] Jáuregui Ostos, E. (1992). Capítulo 15. La isla de calor urbano de la Ciudad de México a finales del siglo XIX. *Boletín de Investigaciones Geográficas*, 26:31–40.

[25] Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., Zhu, Y., Chelliah, M., Ebisuzaki, W., Higgins, W., Janowiak, J., Mo, K. C., Ropelewski, C., Wang, J., Leetmaa, A., Reynolds, R., Jenne, R., and Joseph, D. (1996). The NCEP/NCAR 40-Year Reanalysis Project. *Bulletin of the American Meteorological Society*, 77(3):437–472. Publisher: American Meteorological Society Section: Bulletin of the American Meteorological Society.

[26] Kitchin, R. and McArdle, G. (2016). What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets. *Big Data & Society*, 3(1):2053951716631130. Publisher: SAGE Publications Ltd.

[27] Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. *META group research note*, 6(70):1. Publisher: Stanford.

[28] Lewis Dijkstra, Ellen Hamilton, Somik Lall, and Sameh Wahba (2020). How do we define cities, towns, and rural areas?

[29] Liu, P. (2015). A survey of remote-sensing big data. *Frontiers in Environmental Science*, 3:45.

[30] Lohr, S. (2013). The Origins of 'Big Data': An Etymological Detective Story. Cad: 1 Section: Technology.

[31] Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., and Jie, W. (2014). Remote sensing Big Data computing: challenges and opportunities. *Future Generation Computer Systems*, 51.

[32] Malakar, N. K., Hulley, G. C., Hook, S. J., Laraby, K., Cook, M., and Schott, J. R. (2018). An Operational Land Surface Temperature Product for Landsat Thermal Data: Methodology and Validation. *IEEE Transactions on Geoscience and Remote Sensing*, 56(10):5717–5735. Conference Name: IEEE Transactions on Geoscience and Remote Sensing.

[33] Martins, J. P. A., Trigo, I. F., Bento, V. A., and Da Camara, C. (2016). A Physically Constrained Calibration Database for Land Surface Temperature Using Infrared Retrieval Algorithms. *Remote Sensing*, 8(10):808. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

[34] MDN (2022). ¿Qué es JavaScript? - Aprende sobre desarrollo web.

[35] Mexico, de Población (Mexico), C. N., and Instituto Nacional de Estadística, G. e. I. M., editors (2004). *Delimitación de las zonas metropolitanas de México*. Secretaría de Desarrollo Social : Consejo Nacional de Población ; Instituto Nacional de Estadística, Geografía e Informática, México, D.F. : Aguascalientes, Ags, 1. ed edition. OCLC: ocm58974467.

[36] Mills, G. (2008). Luke Howard and The Climate of London. *Weather*, 63(6):153–157.

[37] NASA Earth Observations (NEO) (2021). Land Surface Temperature. Publisher: NASA Earth Observatory.

[38] Oke, T. (1988). The urban energy balance. *Progress in Physical Geography: Earth and Environment*, 12(4):471–508. Publisher: SAGE Publications Ltd.

[39] Oke, T. R., Cermak, J. E., Davenport, A. G., Plate, E. J., and Viegas, D. X. (1995). The Heat Island of the Urban Boundary Layer: Characteristics, Causes and Effects. In *Wind Climate in Cities*, NATO ASI Series, pages 81–107. Springer Netherlands, Dordrecht.

[40] Oke, T. R., Mills, G., Christen, A., and Voogt, J. A. (2017). *Urban Climates*. Cambridge University Press, Cambridge.

[41] on the Mexico City Water Supply, J. A. C., editor (1995). *Mexico City's water supply: improving the outlook for sustainability*. National Academy Press, Washington, D.C.

[42] Peres, L. and DaCamara, C. (2005). Emissivity maps to retrieve land-surface temperature from MSG/SEVIRI. *IEEE Transactions on Geoscience and Remote Sensing*, 43(8):1834–1844. Conference Name: IEEE Transactions on Geoscience and Remote Sensing.

[43] Prihodko, L. and Goward, S. N. (1997). Estimation of air temperature from remotely sensed surface observations. *Remote Sensing of Environment*, 60(3):335–346.

[44] Ren, H., Liu, R., Qin, Q., Fan, W., Yu, L., and Du, C. (2017). Mapping finer-resolution land surface emissivity using Landsat images in China. *Journal of Geophysical Research: Atmospheres*, 122(13):6764–6781. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/2017JD026910.

[45] Rosenzweig, C., Solecki, W., Parshall, L., Gaffin, S., Lynn, B., Goldberg, R., Cox, J., Hodges, S., and College, H. (2006). MITIGATING NEW YORK CITY'S HEAT ISLAND WITH URBAN FORESTRY, LIVING ROOFS, AND LIGHT SURFACES. *86th AMS Annual Meeting*, page 5.

[46] Rubio, E., Caselles, V., and Badenas, C. (1997). Emissivity measurements of several soils and vegetation types in the 8–14, m Wave band: Analysis of two field methods. *Remote Sensing of Environment*, 59(3):490–521.

[47] Survey, U. S. G. (2015a). Landsat surface reflectance data. Technical Report 2015-3034, U.S. Geological Survey. ISSN: 2327-6932 Publication Title: Fact Sheet.

[48] Survey, U. S. G. (2015b). Landsat—Earth observation satellites. Technical Report 2015-3081, U.S. Geological Survey. ISSN: 2327-6932 Publication Title: Fact Sheet.

[49] Tang, R., Li, Z.-L., and Tang, B. (2010). An application of the Ts–VI triangle method with enhanced edges determination for evapotranspiration estimation from MODIS data in arid and semi-arid regions: Implementation and validation. *Remote Sensing of Environment*, 114(3):540–551.

[50] U.S. Geological Survey (2021). Landsat 8: Satellite Imagery, Overview, And Characteristics.

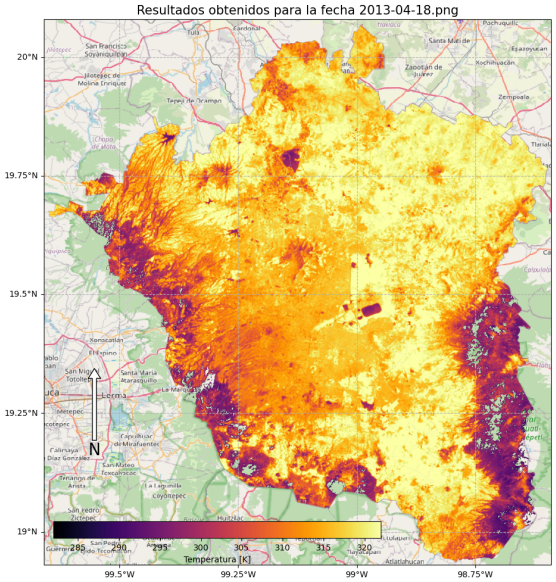
[51] Wang, F., Qin, Z., Song, C., Tu, L., Karnieli, A., and Zhao, S. (2015). An Improved Mono-Window Algorithm for Land Surface Temperature Retrieval from Landsat 8 Thermal Infrared Sensor Data. *Remote Sensing*, 7(4):4268–4289. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[52] Yu, X., Guo, X., and Wu, Z. (2014). Land Surface Temperature Retrieval from Landsat 8 TIRS—Comparison between Radiative Transfer Equation-Based Method, Split Window Algorithm and Single Channel Method. *Remote Sensing*, 6(10):9829–9852. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

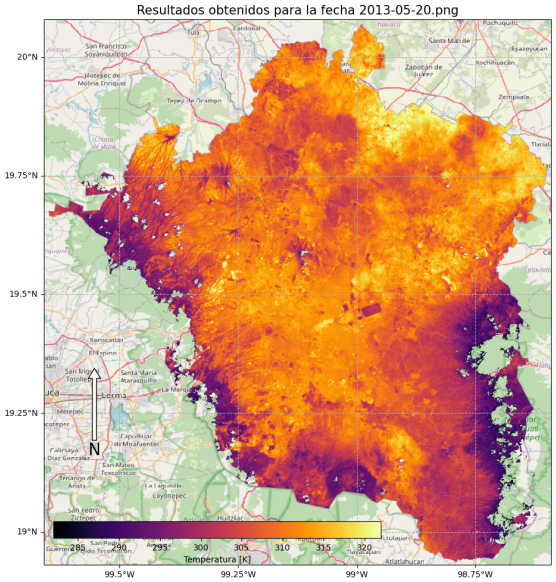
[53] Zhang, F., Sha, M., Wang, G., Li, Z., and Shao, Y. (2017). Urban Aerodynamic Roughness Length Mapping Using Multitemporal SAR Data. *Advances in Meteorology*, 2017:e8958926. Publisher: Hindawi.

Apéndice A

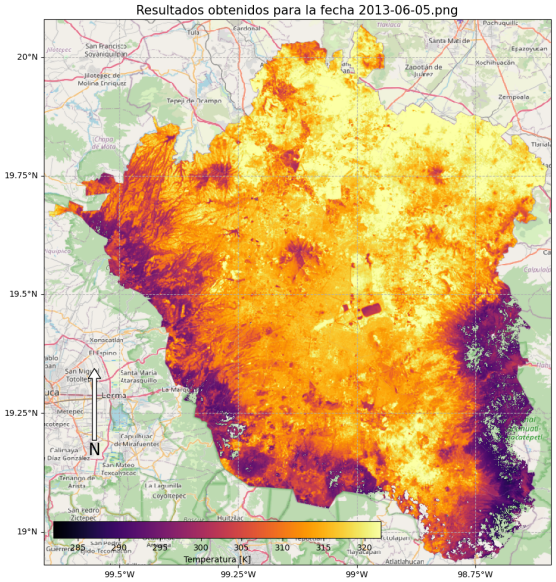
Imágenes obtenidas



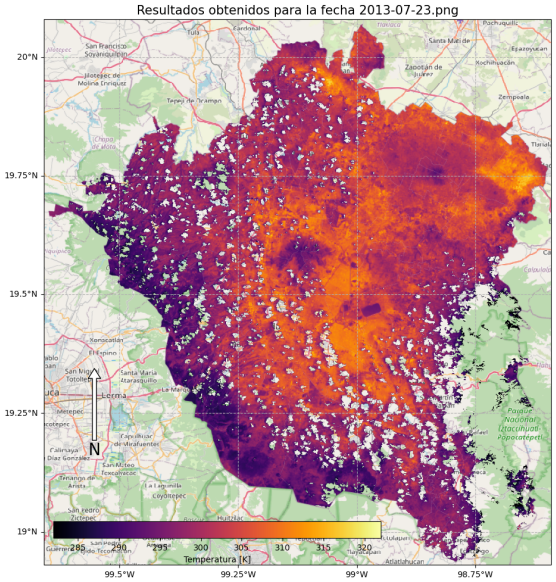
(a) 2013-04-18



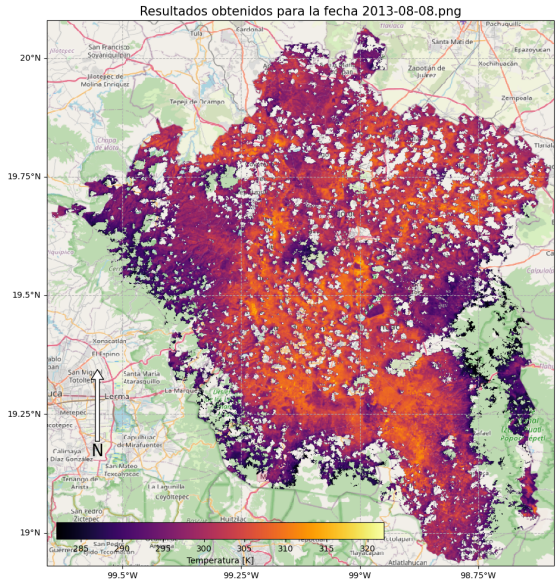
(b) 2013-05-20



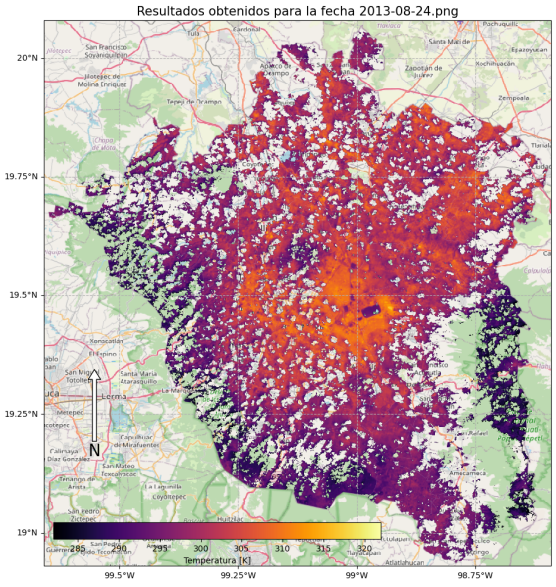
(c) 2013-06-05



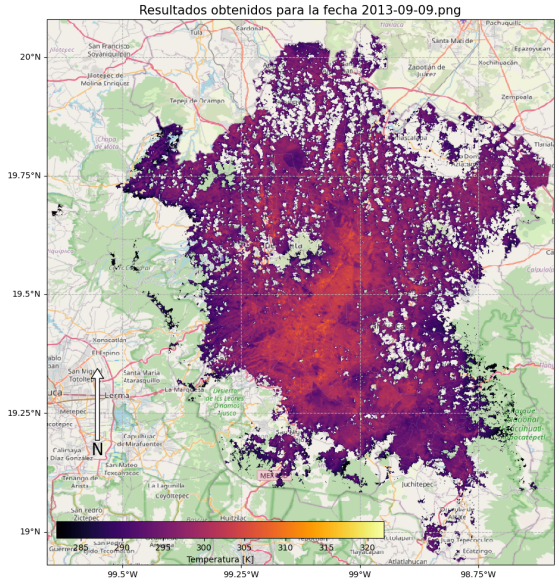
(d) 2013-07-23



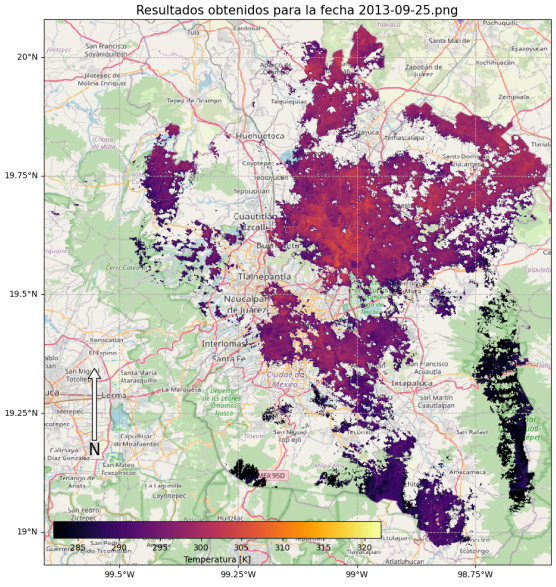
(a) 2013-08-08



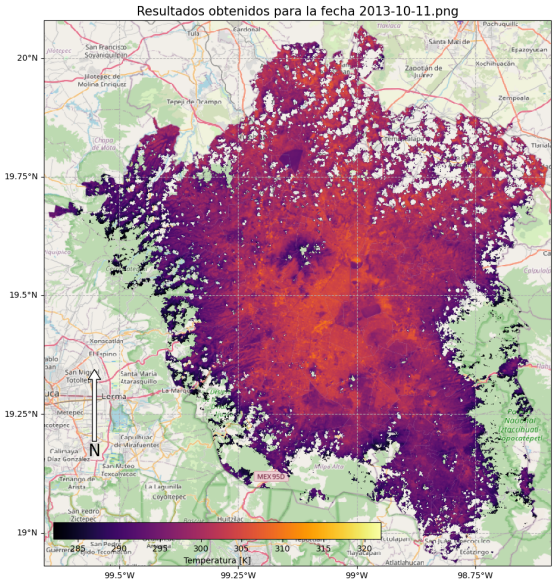
(b) 2013-08-24



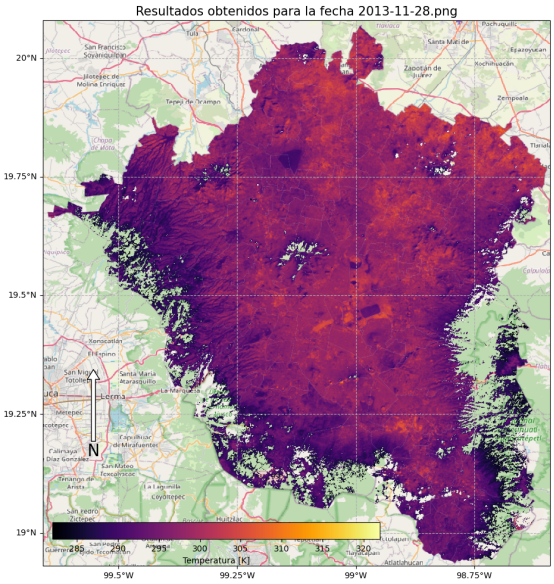
(c) 2013-09-09



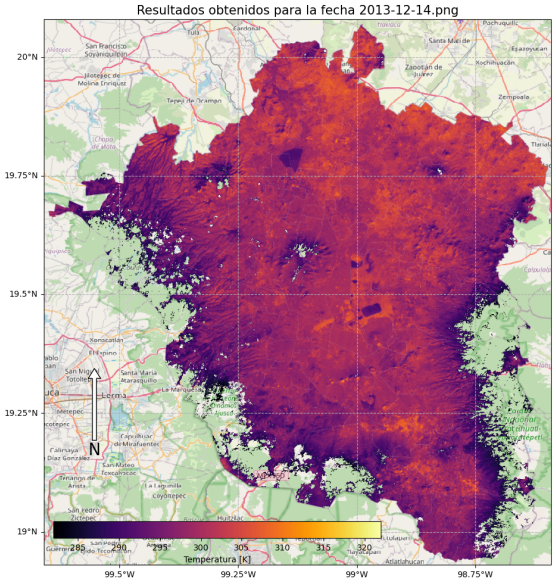
(d) 2013-09-25



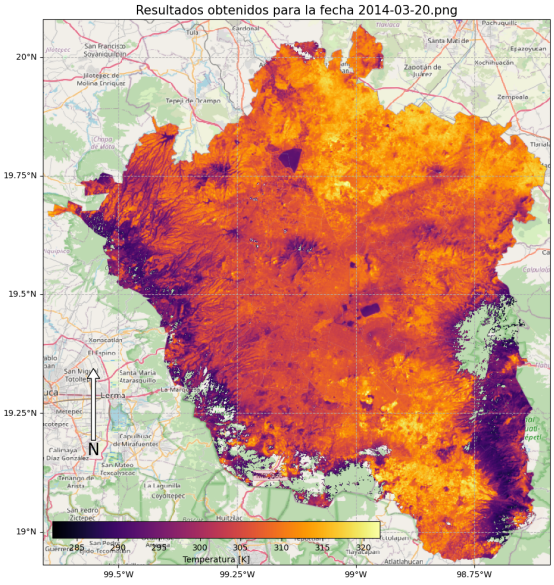
(a) 2013-10-11



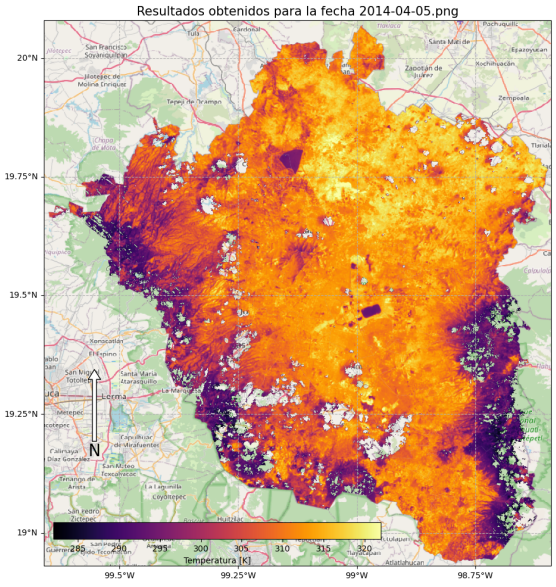
(b) 2013-11-28



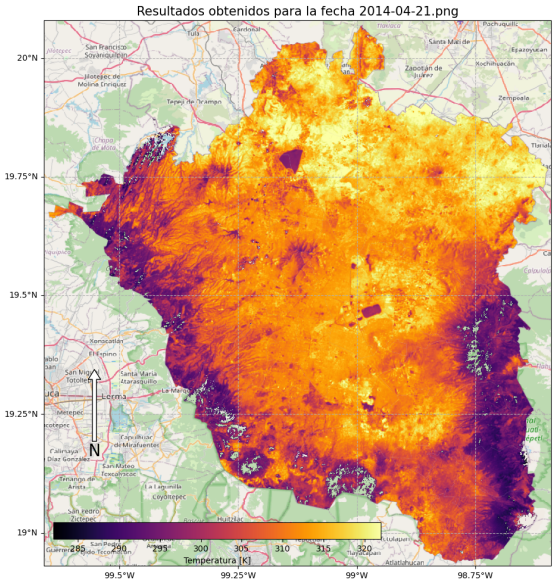
(c) 2013-12-14



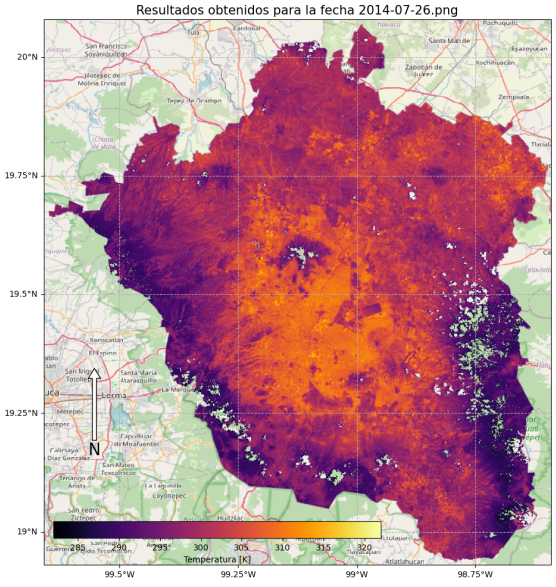
(d) 2014-03-20



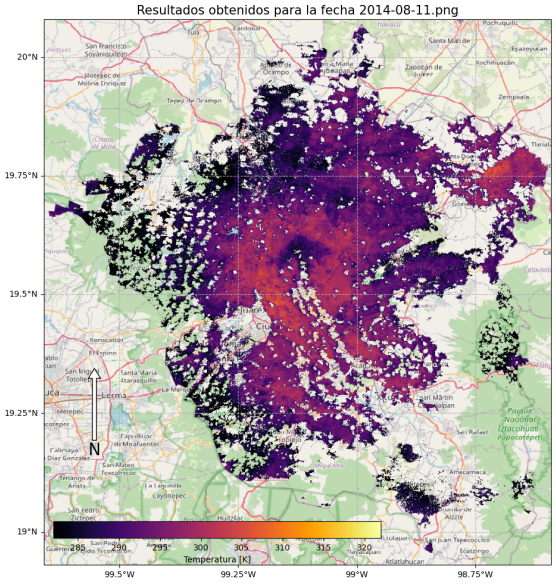
(a) 2014-04-05



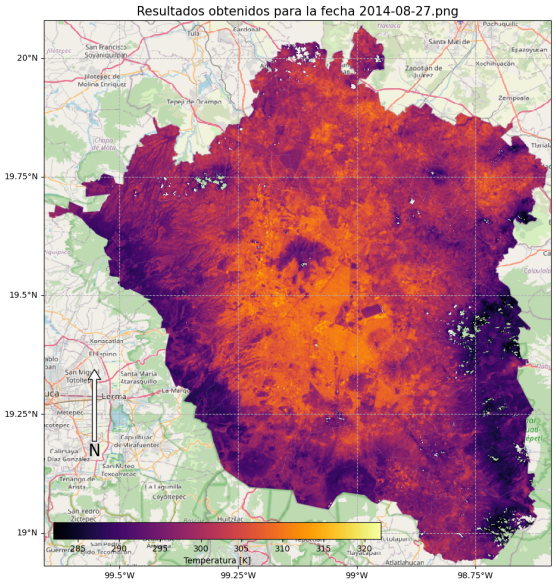
(b) 2014-04-21



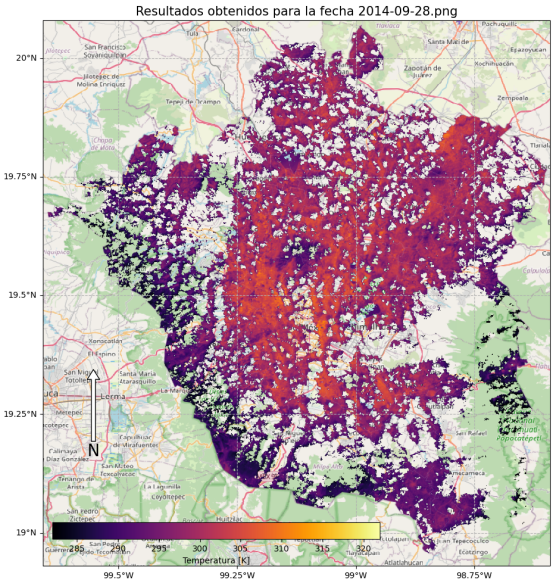
(c) 2014-07-26



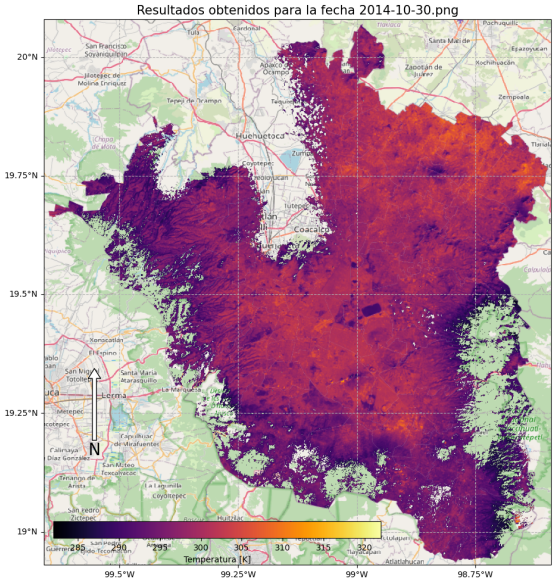
(d) 2014-08-11



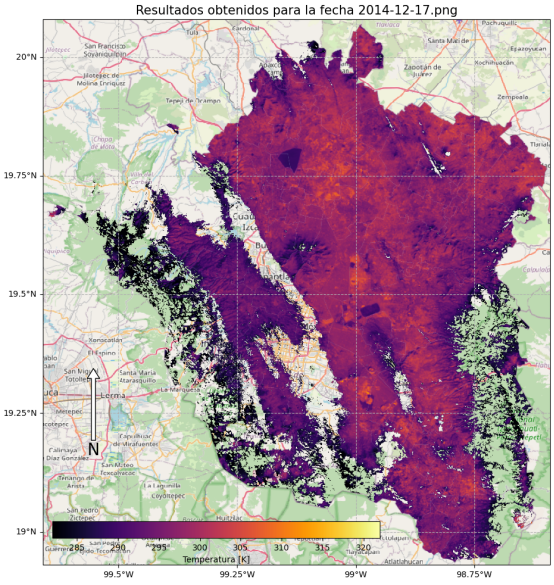
(a) 2014-08-27



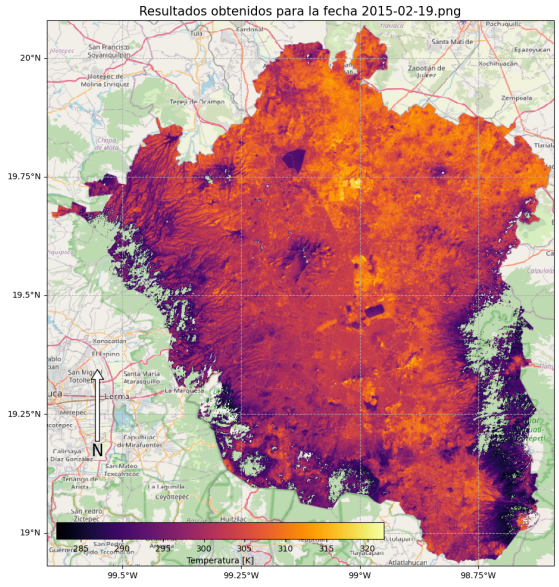
(b) 2014-09-28



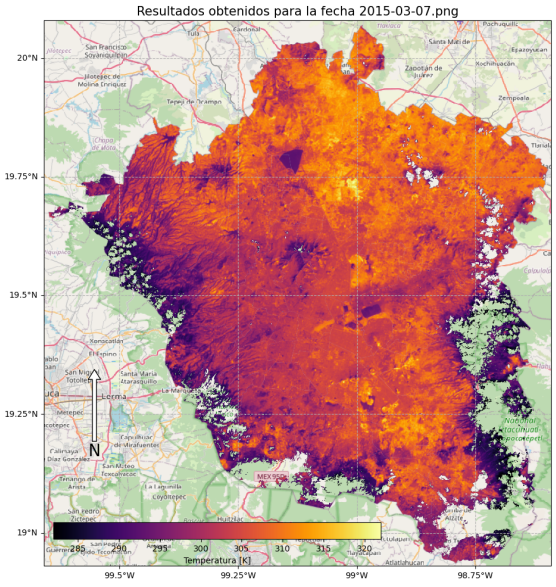
(c) 2014-10-30



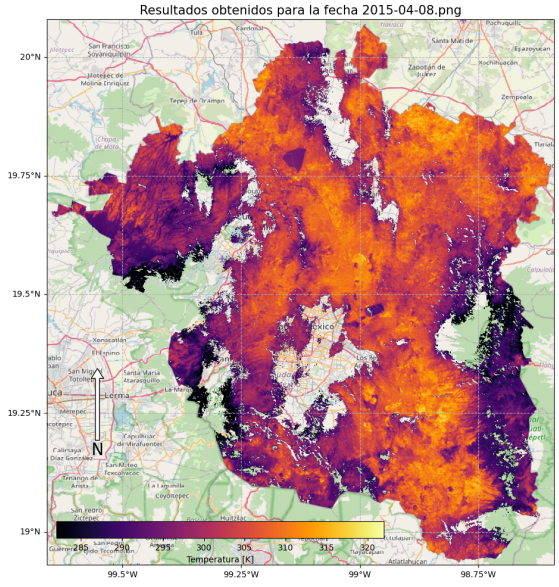
(d) 2014-12-17



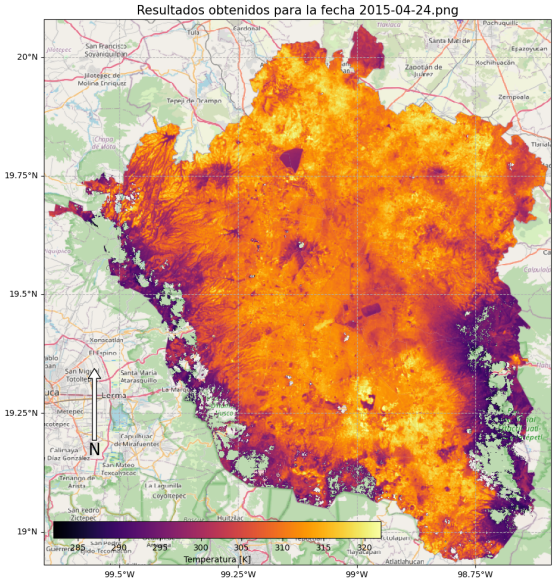
(a) 2015-02-19



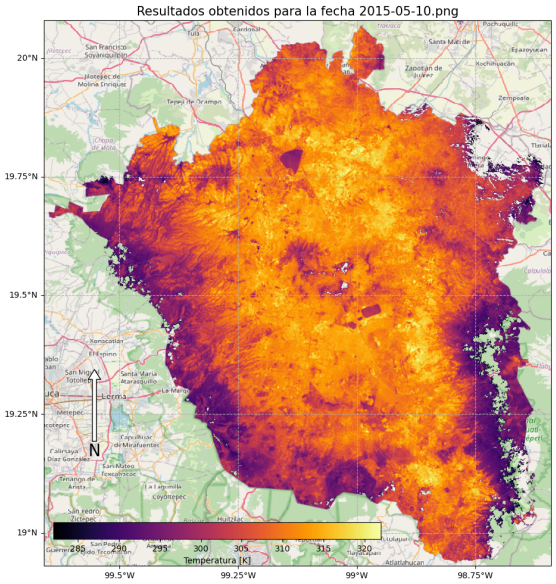
(b) 2015-03-07



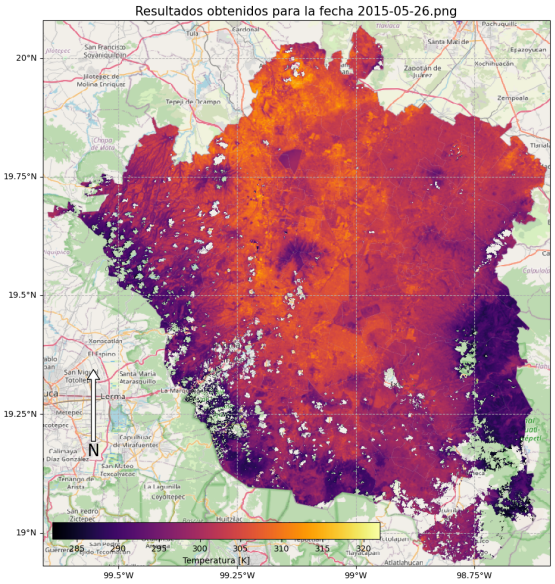
(c) 2015-04-08



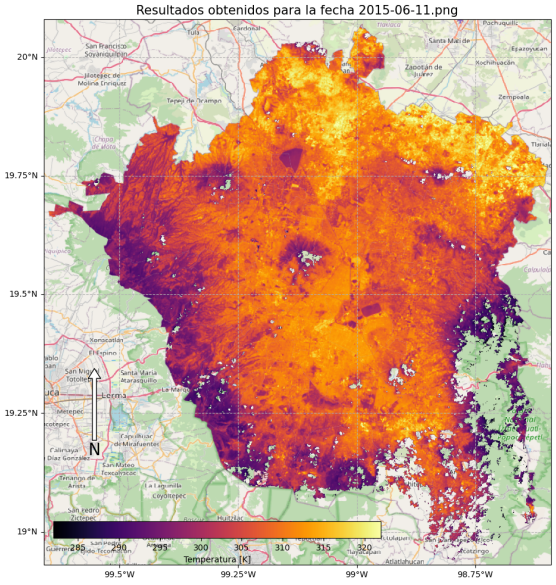
(d) 2015-04-24



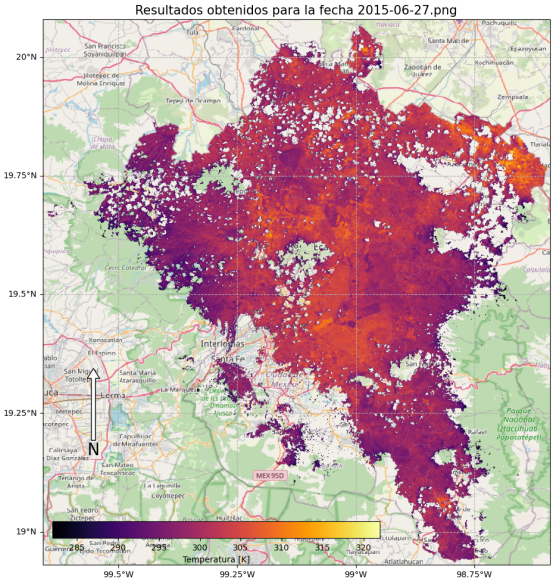
(a) 2015-05-10



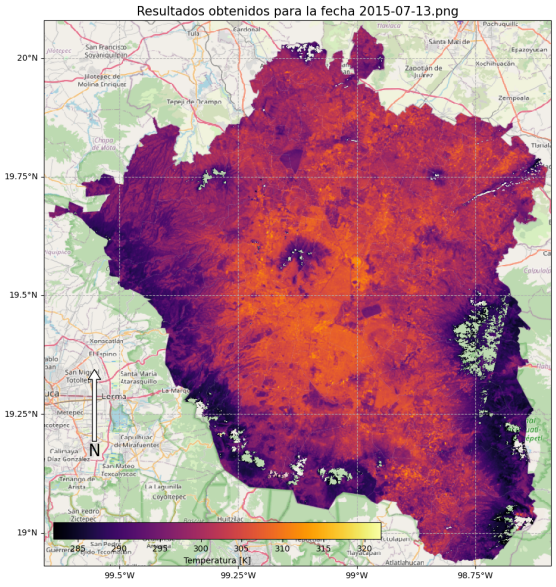
(b) 2015-05-26



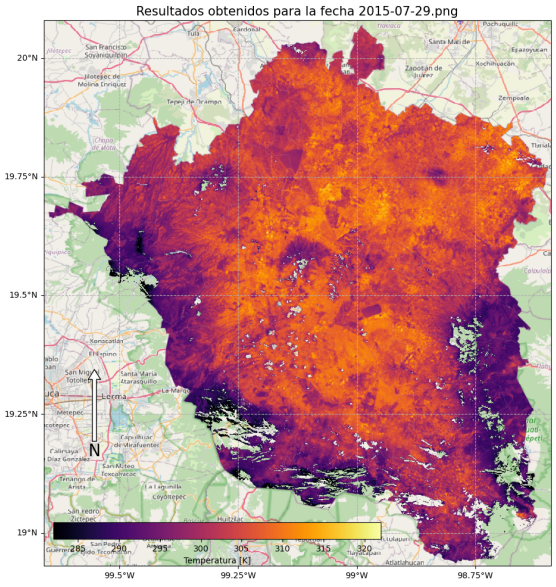
(c) 2015-06-11



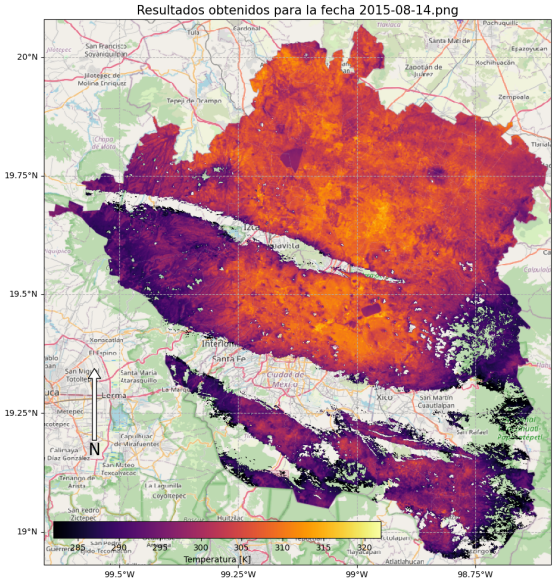
(d) 2015-06-27



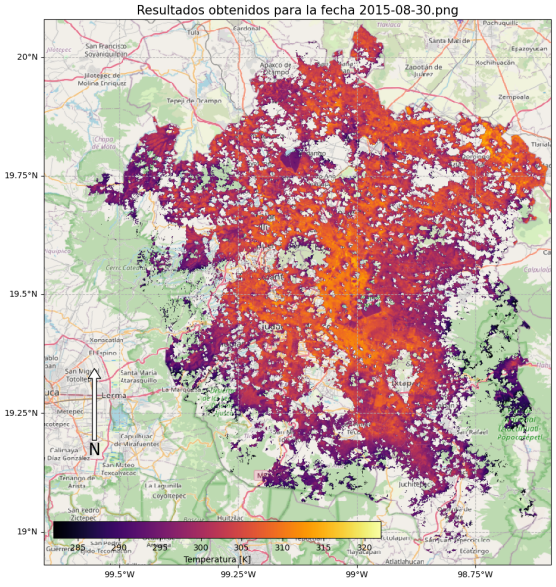
(a) 2015-07-13



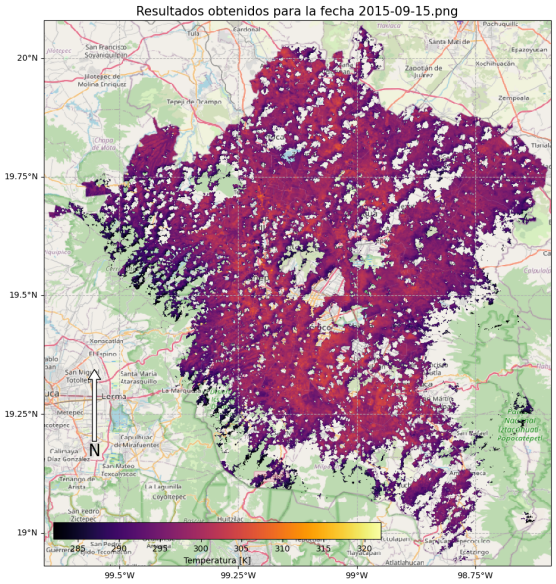
(b) 2015-07-29



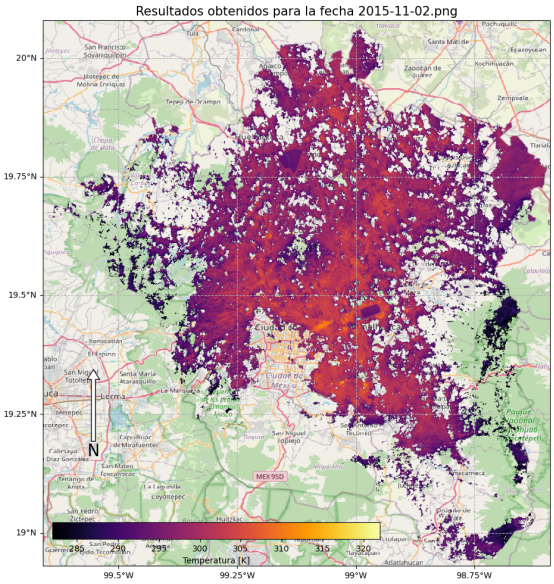
(c) 2015-08-14



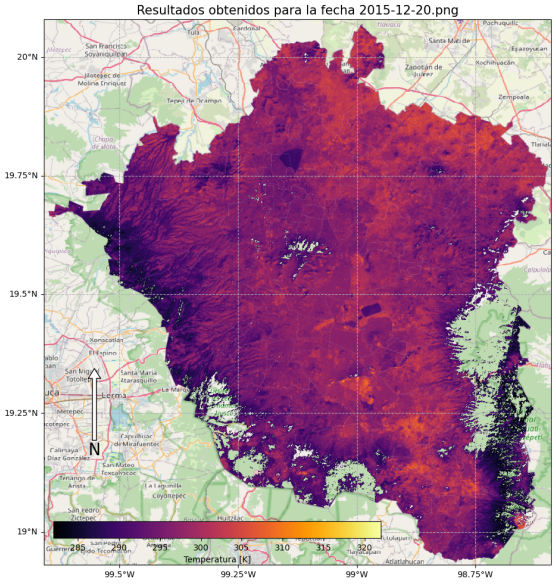
(d) 2015-08-30



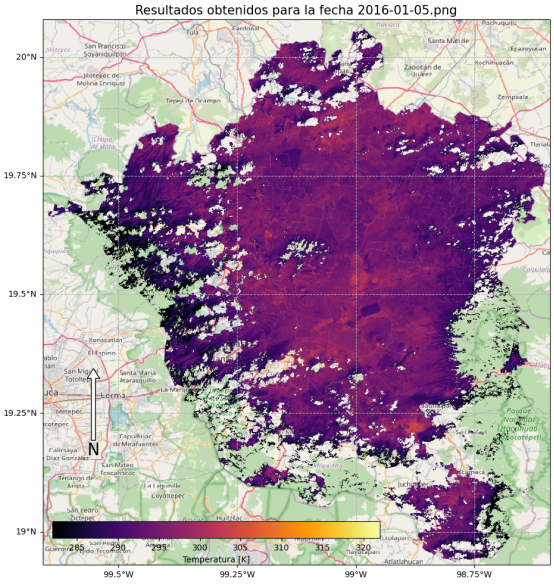
(a) 2015-09-15



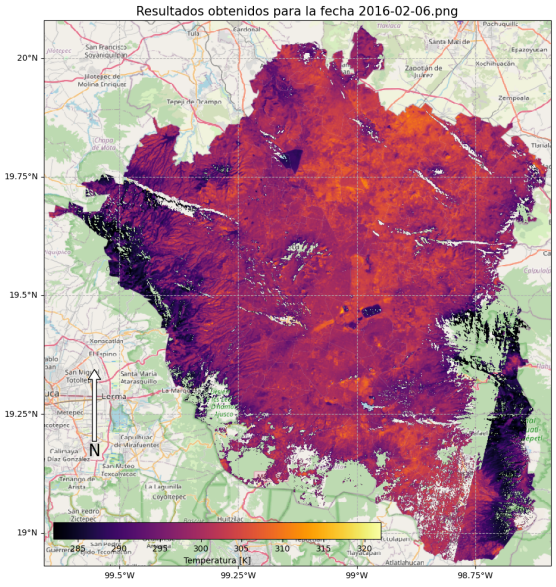
(b) 2015-11-02



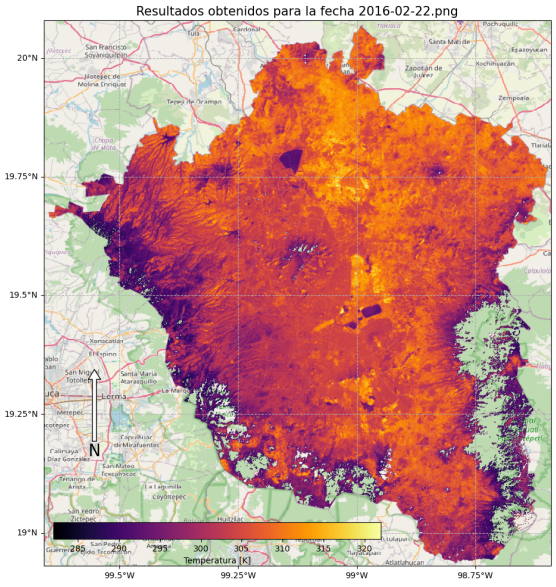
(c) 2015-12-20



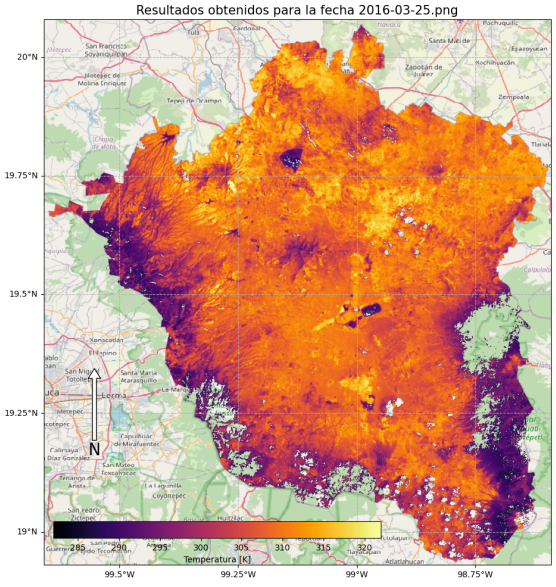
(d) 2016-01-05



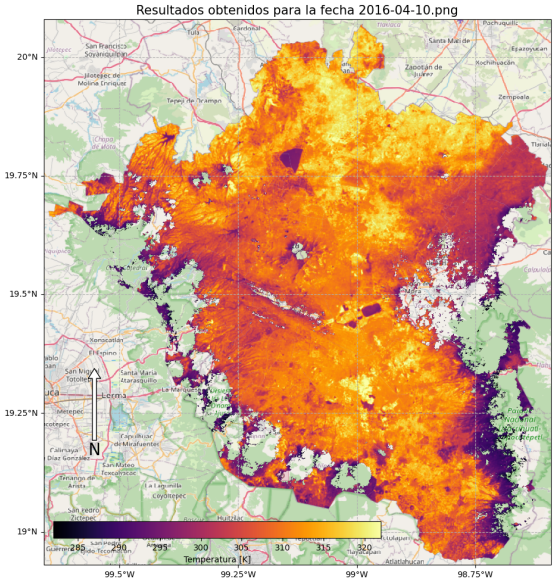
(a) 2016-02-06



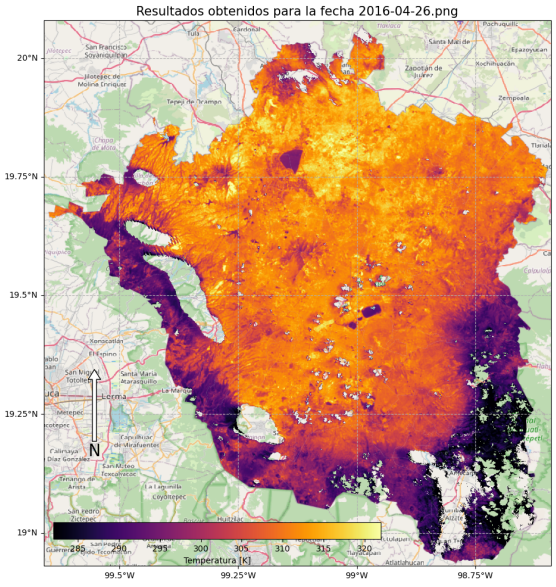
(b) 2016-02-22



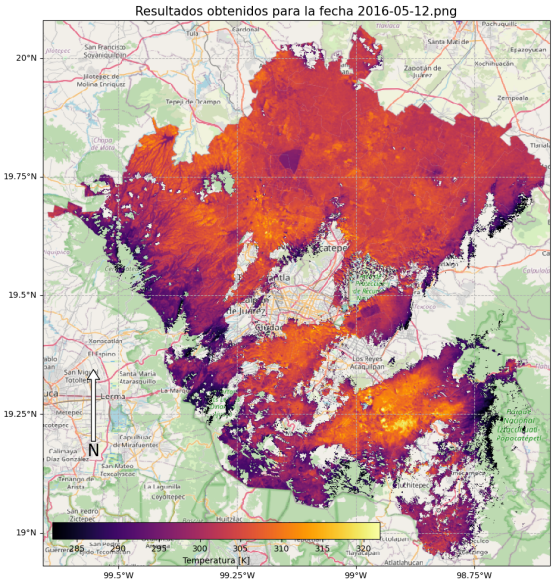
(c) 2016-03-25



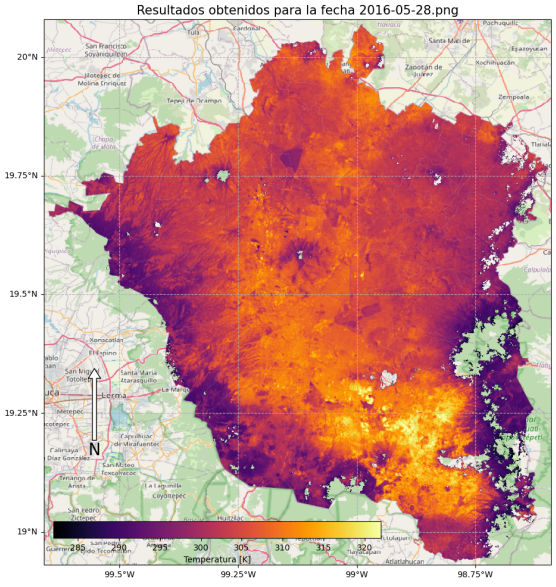
(d) 2016-04-10



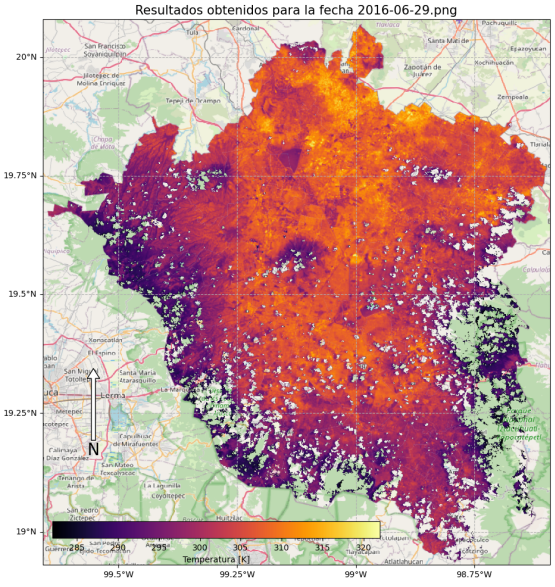
(a) 2016-04-26



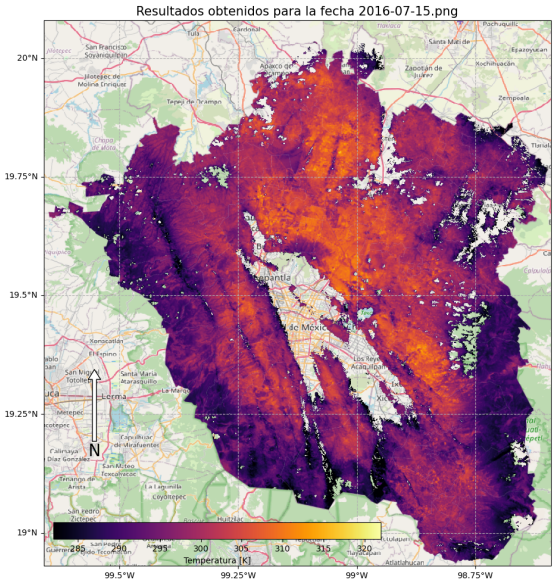
(b) 2016-05-12



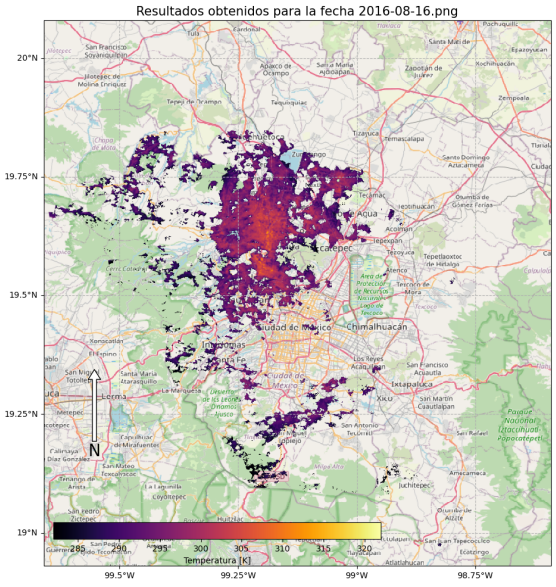
(c) 2016-05-28



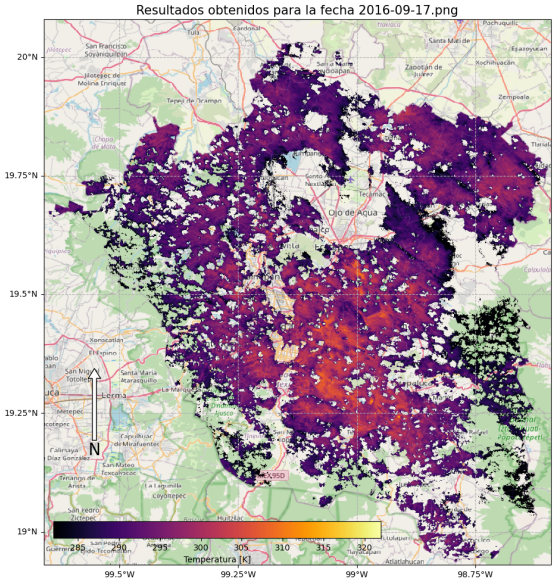
(d) 2016-06-29



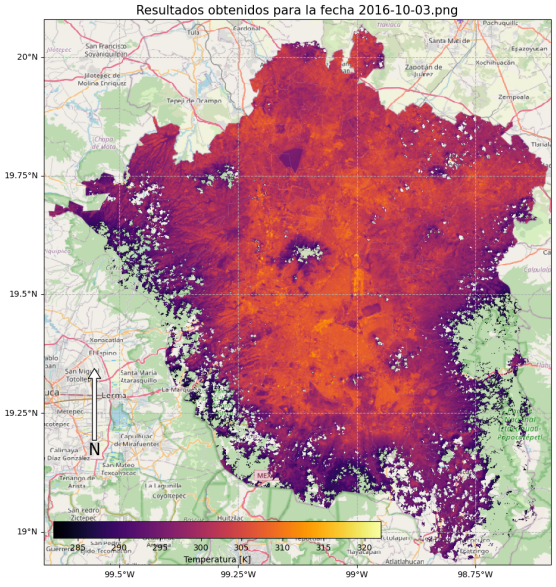
(a) 2016-07-15



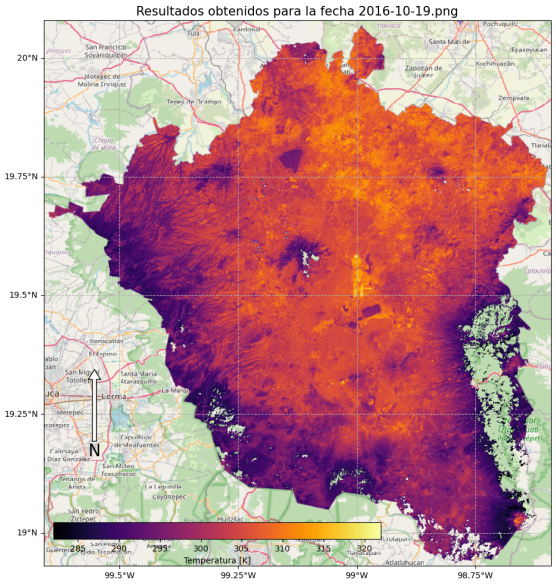
(b) 2016-08-16



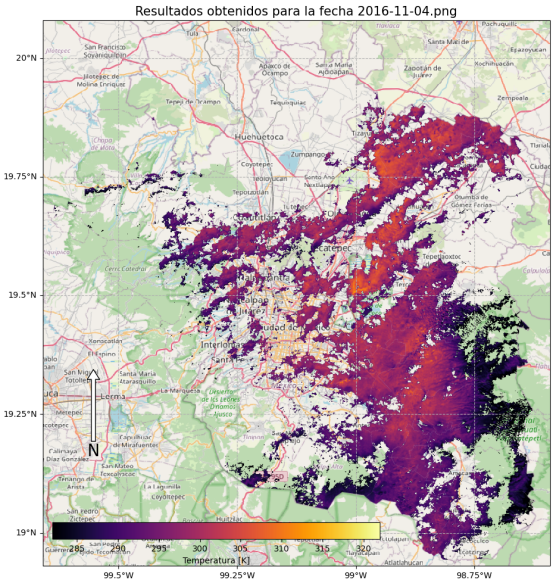
(c) 2016-09-17



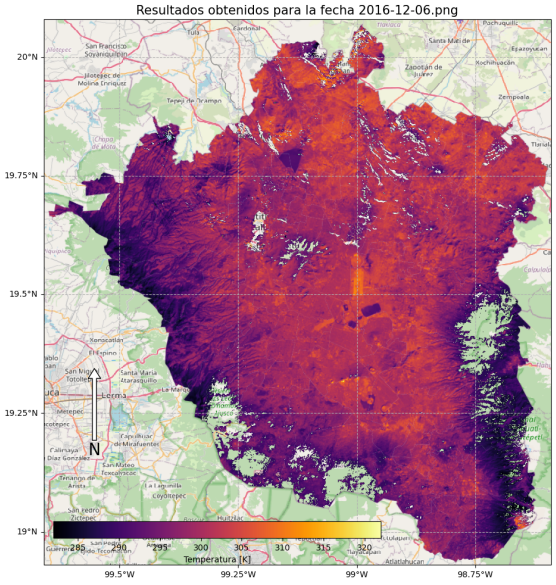
(d) 2016-10-03



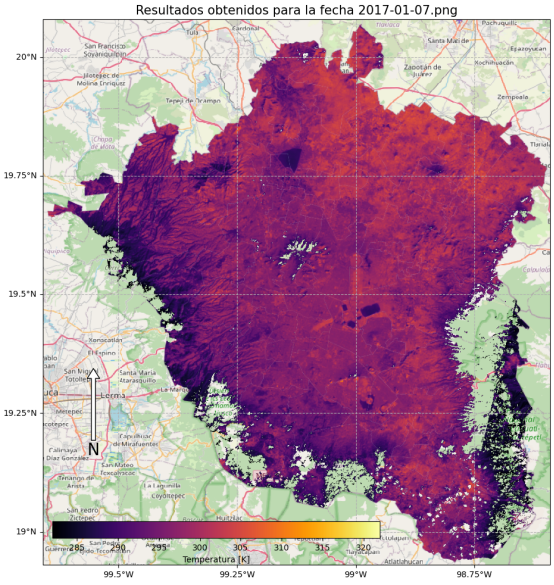
(a) 2016-10-19



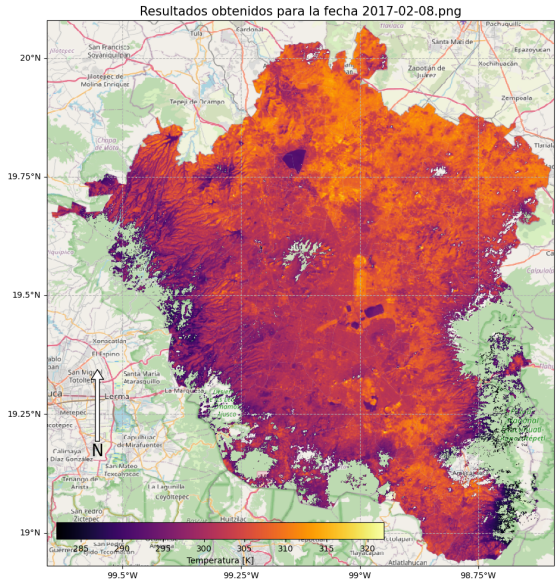
(b) 2016-11-04



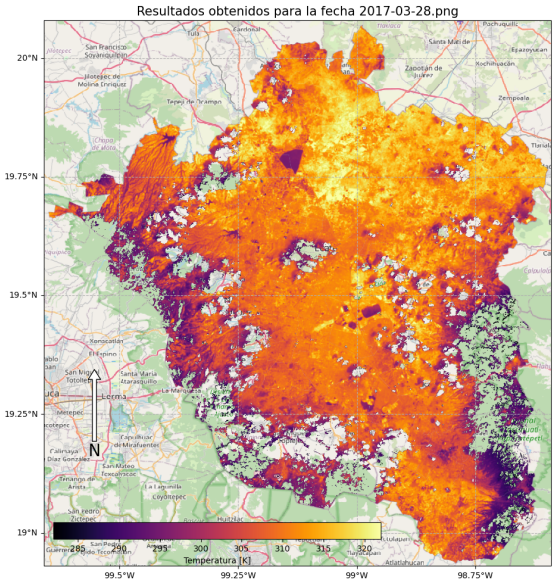
(c) 2016-12-06



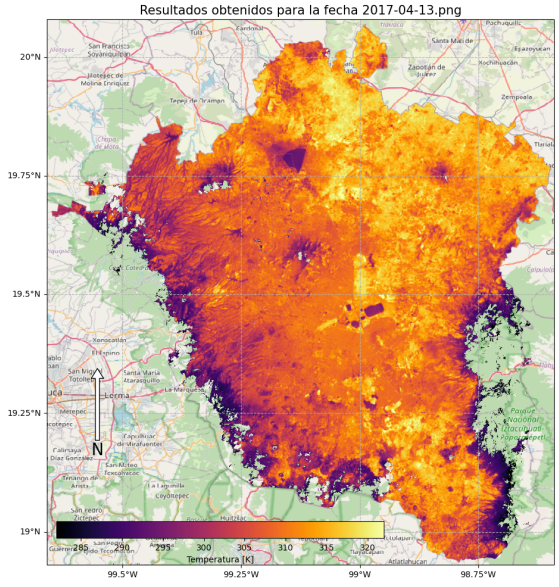
(d) 2017-01-07



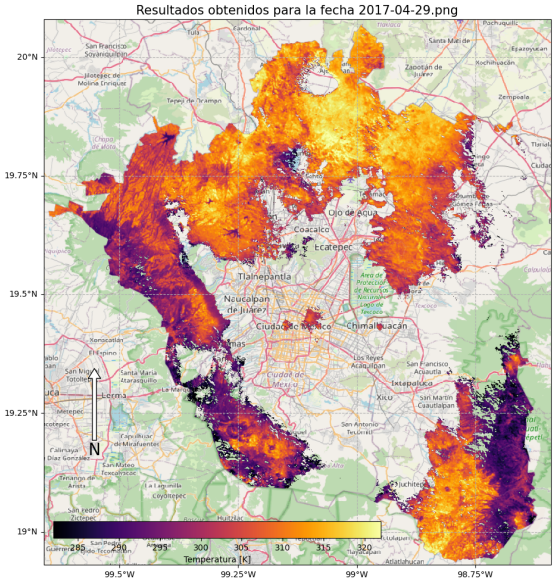
(a) 2017-02-08



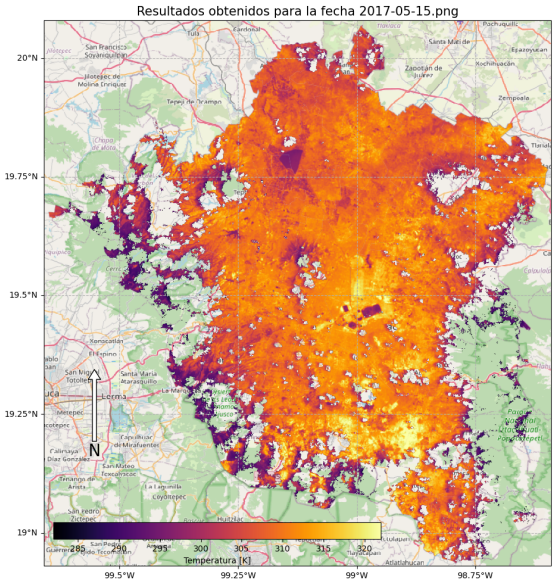
(b) 2017-03-28



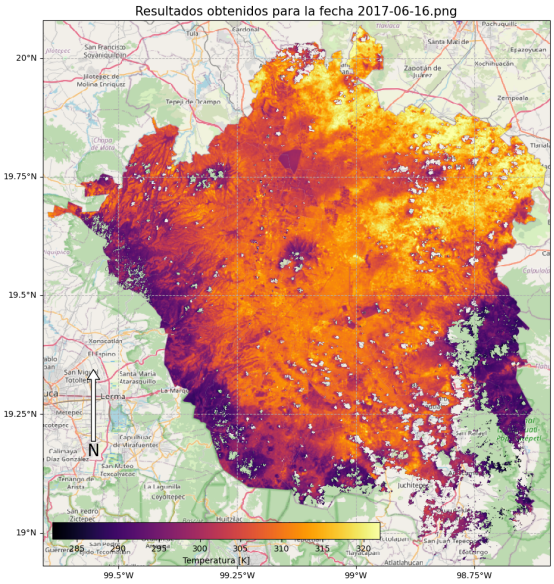
(c) 2017-04-13



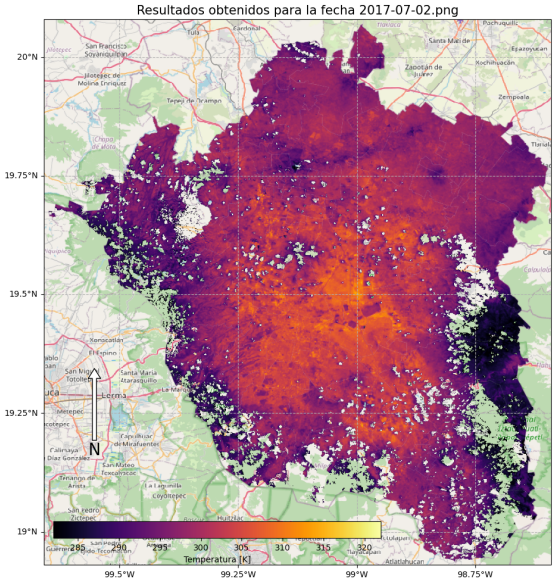
(d) 2017-04-29



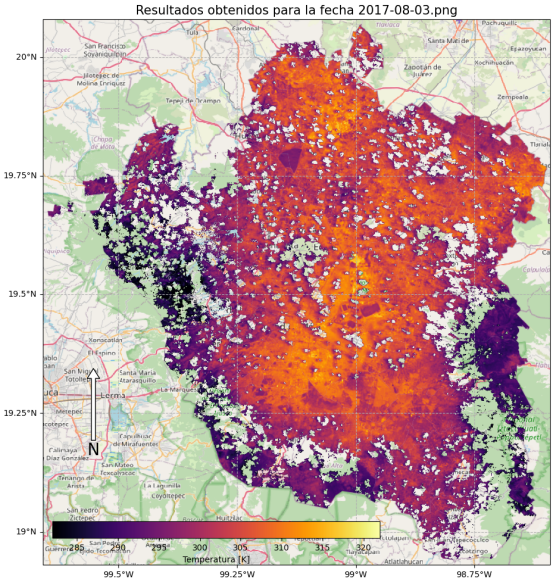
(a) 2017-05-15



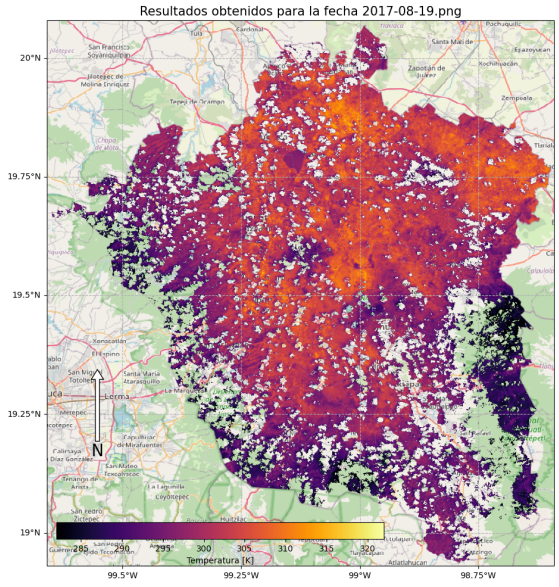
(b) 2017-06-16



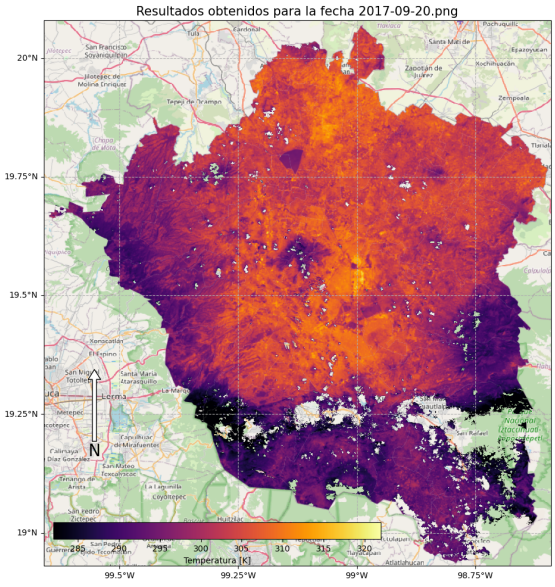
(c) 2017-07-02



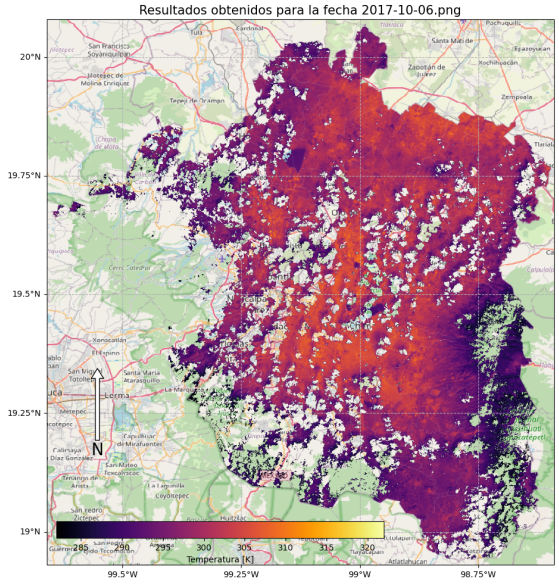
(d) 2017-08-03



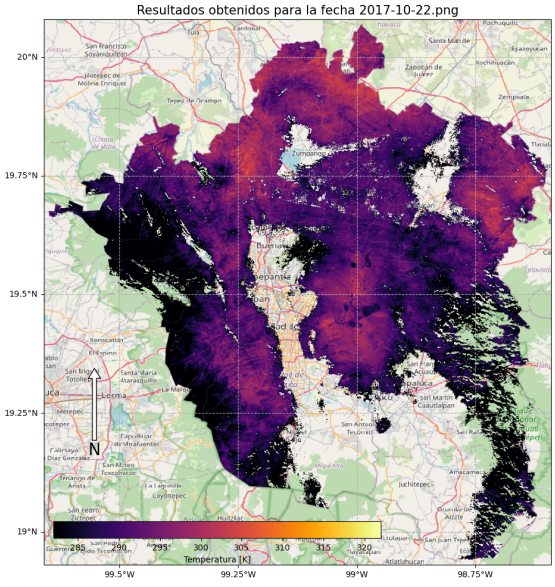
(a) 2017-08-19



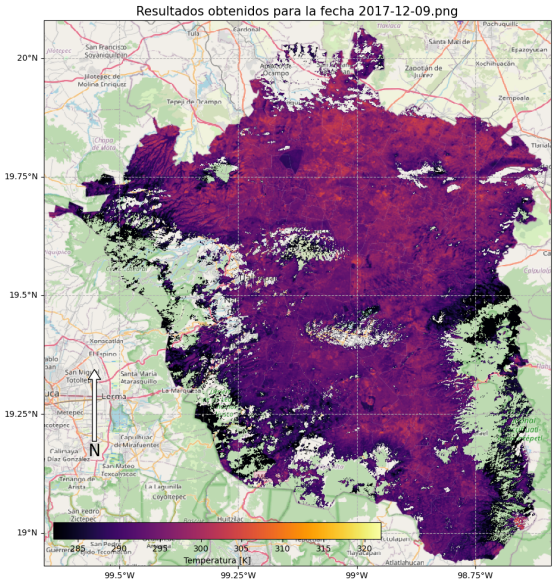
(b) 2017-09-20



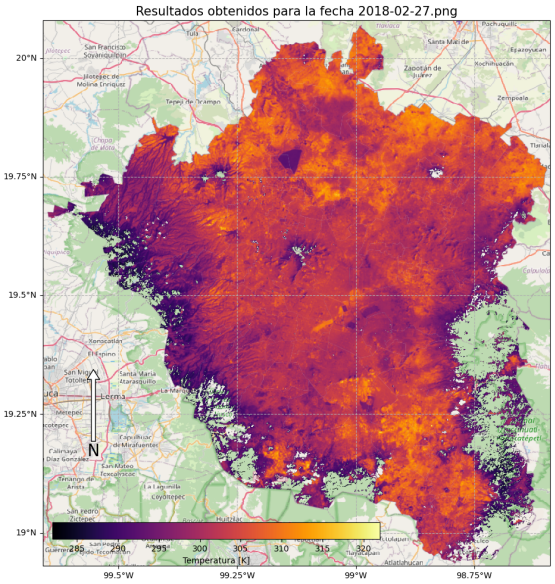
(c) 2017-10-06



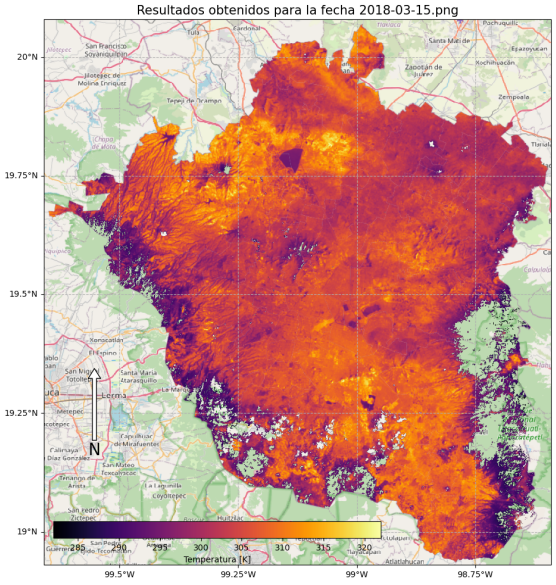
(d) 2017-10-22



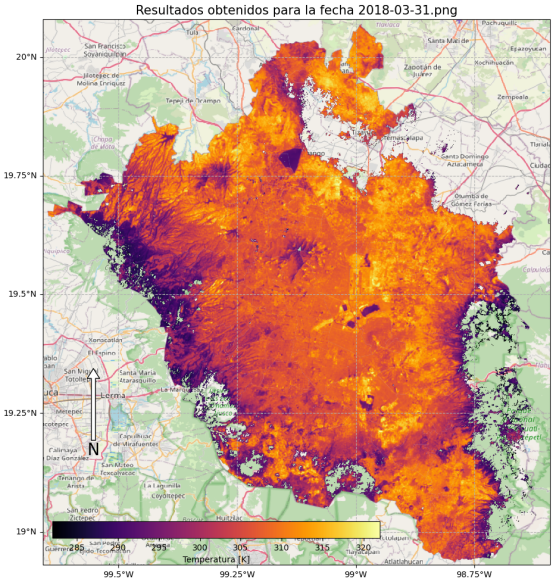
(a) 2017-12-09



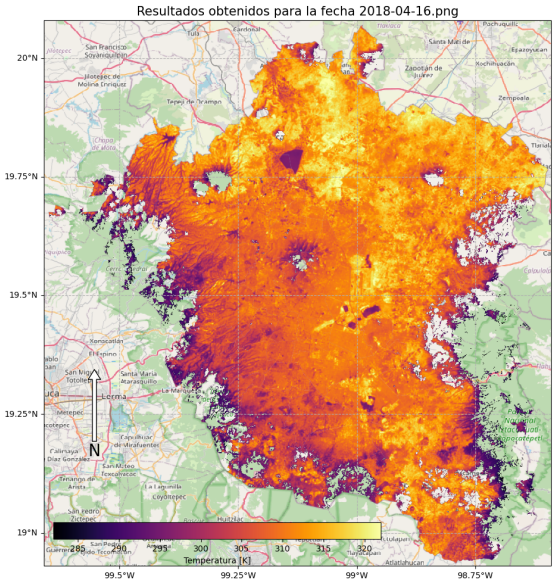
(b) 2018-02-27



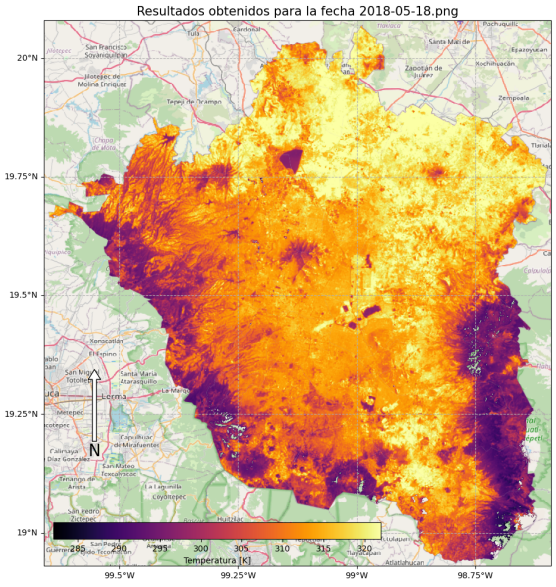
(c) 2018-03-15



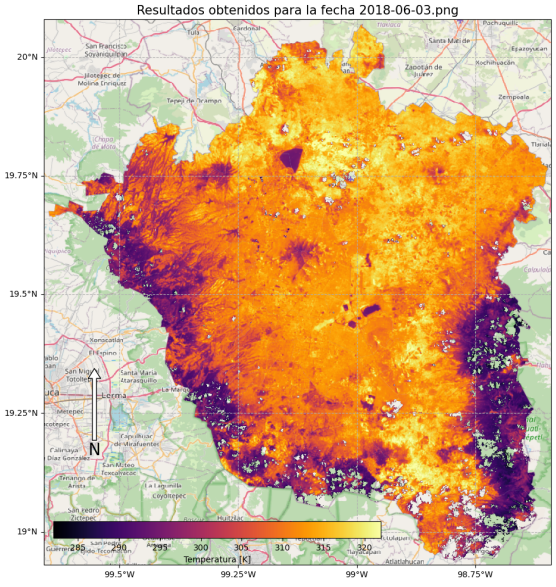
(d) 2018-03-31



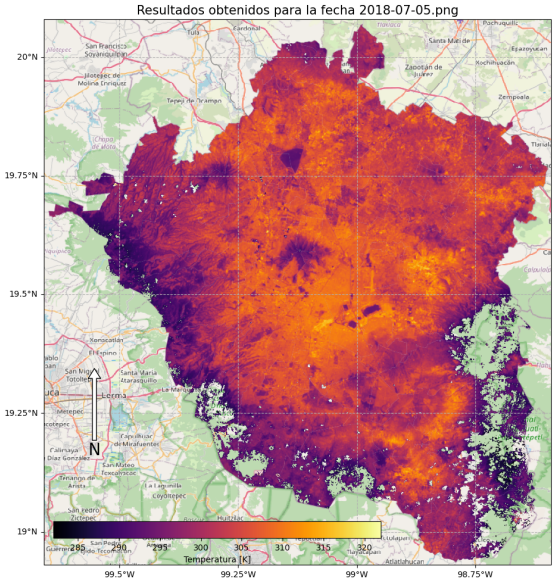
(a) 2018-04-16



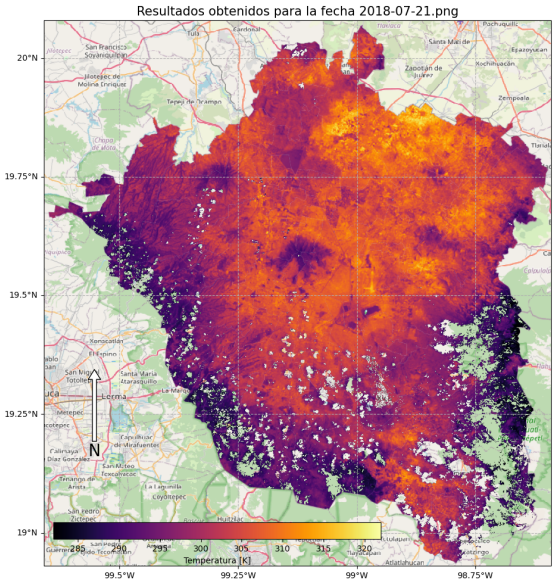
(b) 2018-05-18



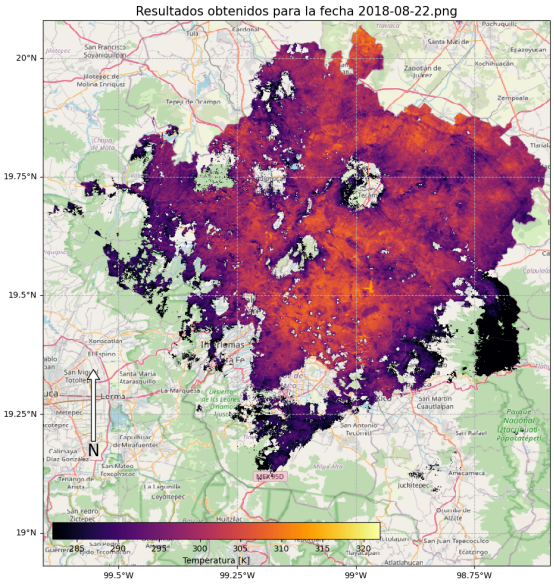
(c) 2018-06-03



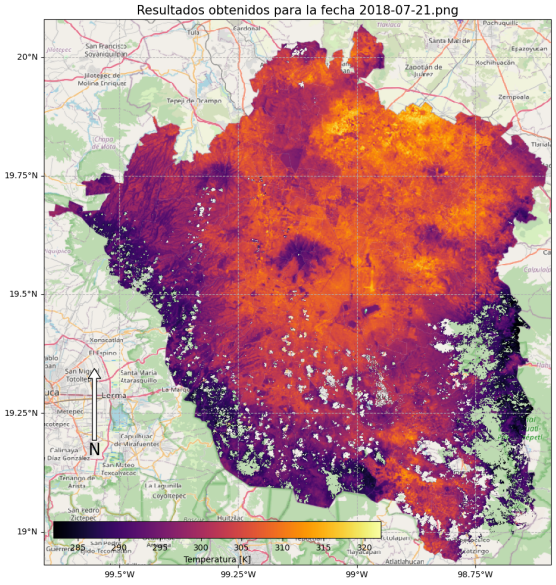
(d) 2018-07-05



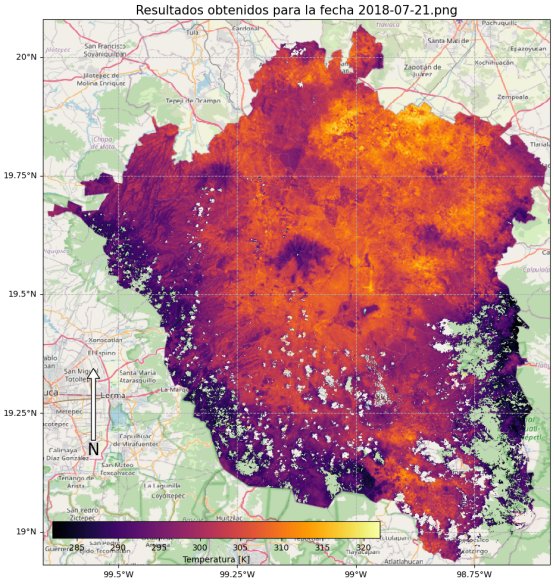
(a) 2018-07-21



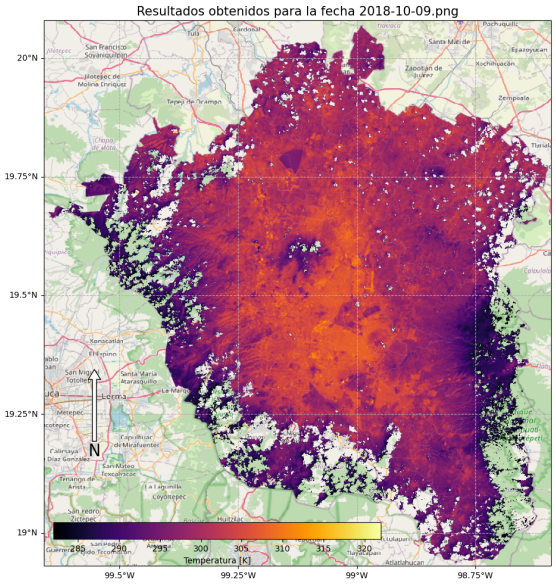
(b) 2018-08-22



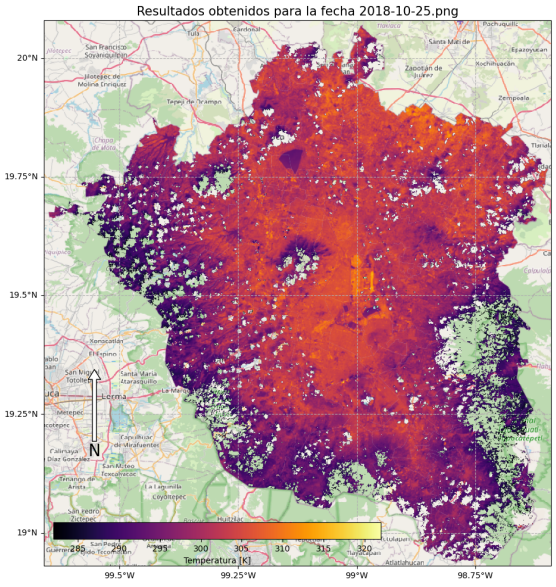
(c) 2018-09-07



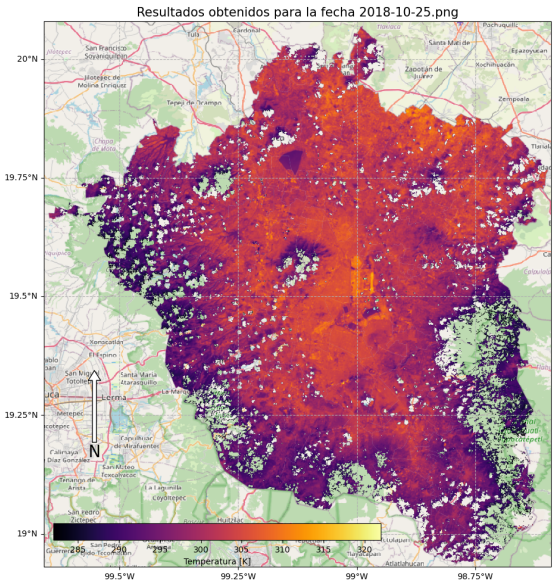
(d) 2018-09-23



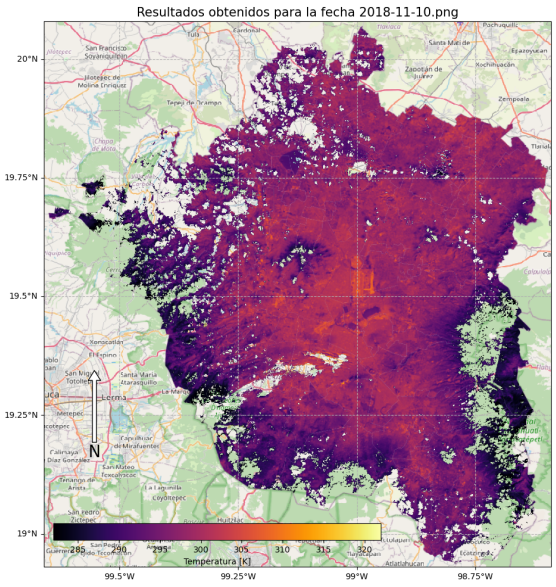
(a) 2018-10-09



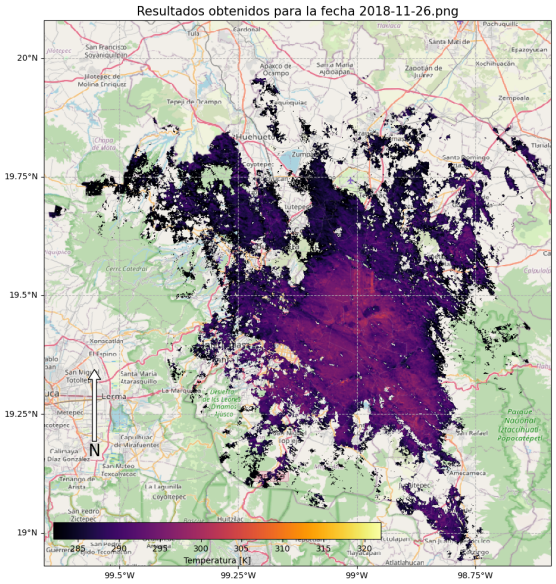
(b) 2018-10-25



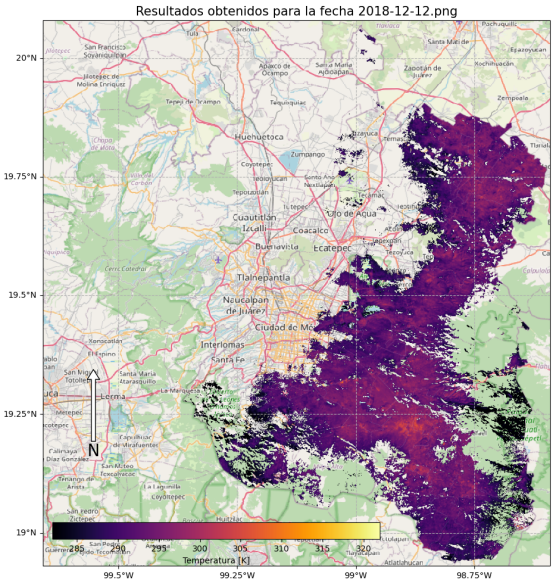
(c) 2018-10-25



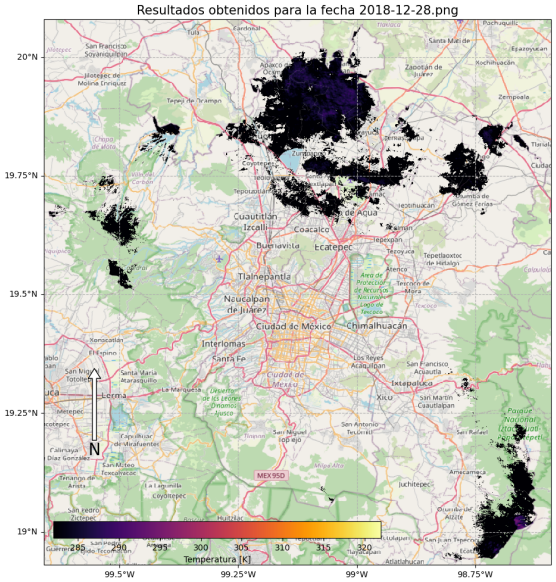
(d) 2018-11-10



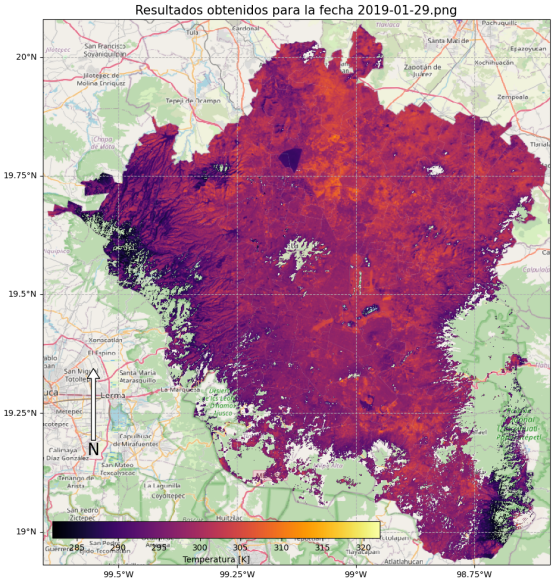
(a) 2018-11-26



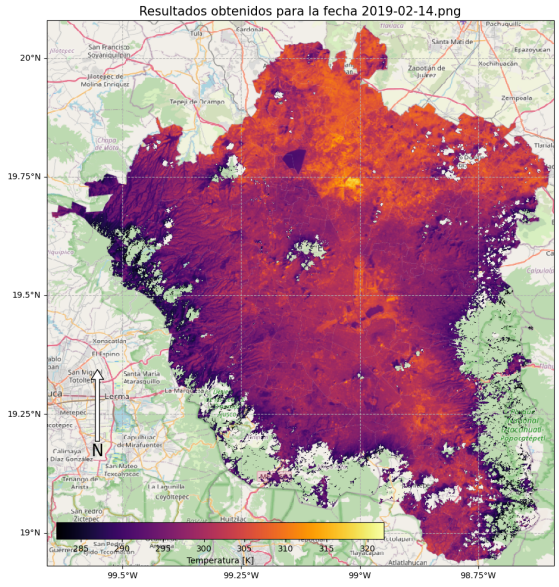
(b) 2018-12-12



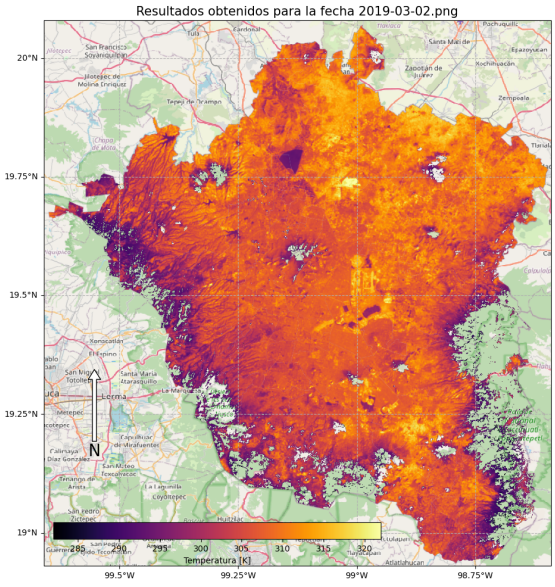
(c) 2018-12-28



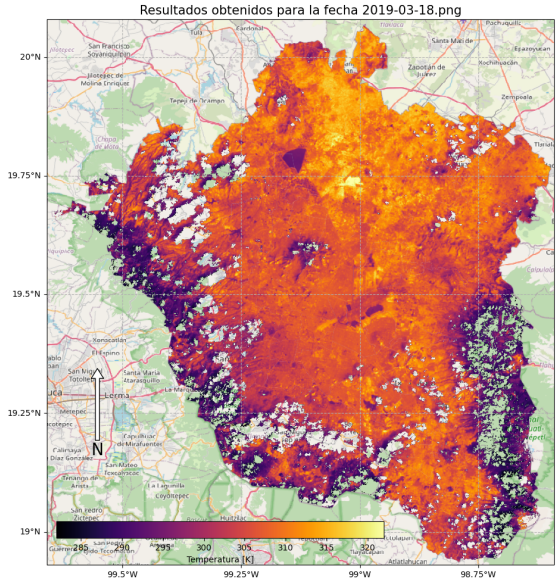
(d) 2019-01-29



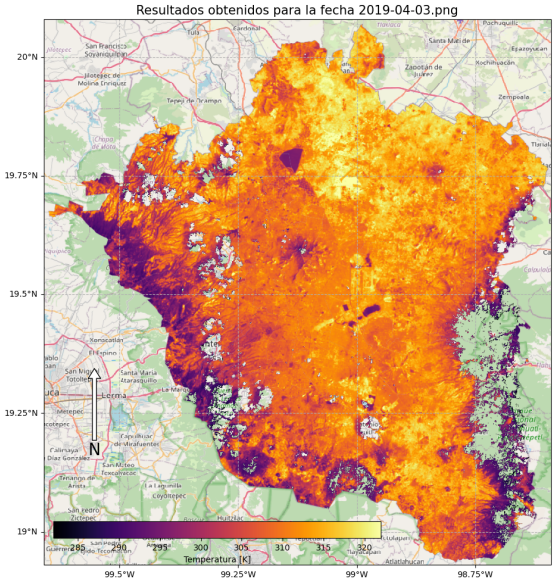
(a) 2019-02-14



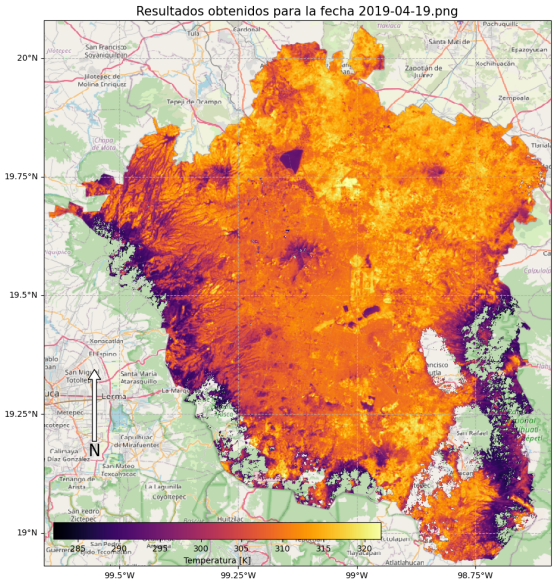
(b) 2019-03-02



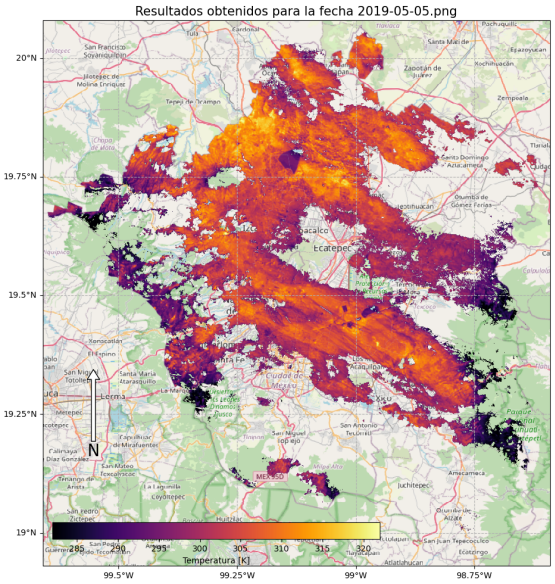
(c) 2019-03-18



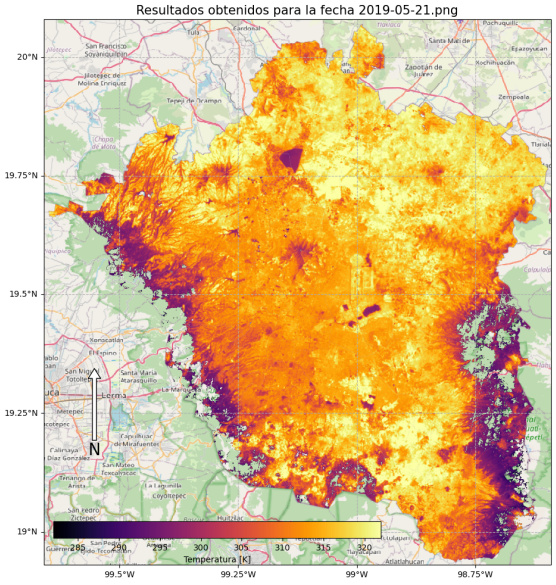
(d) 2019-04-03



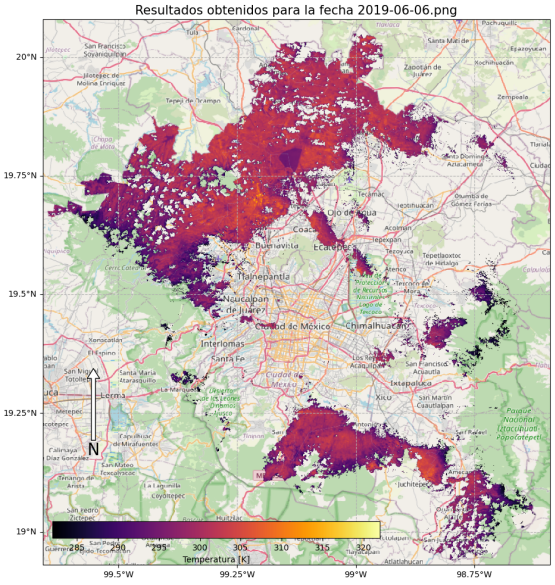
(a) 2019-04-19



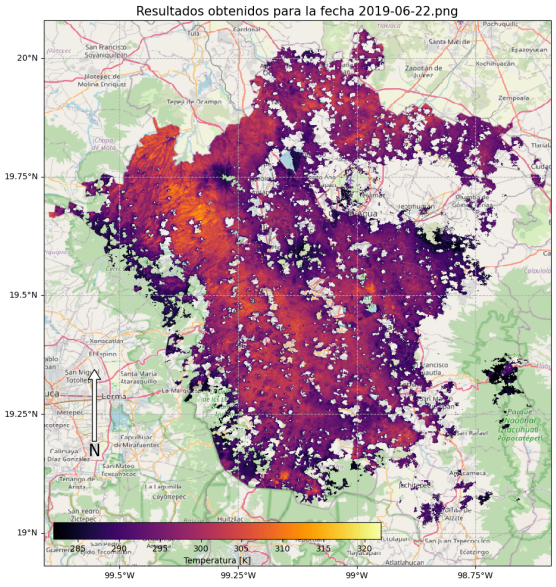
(b) 2019-05-05



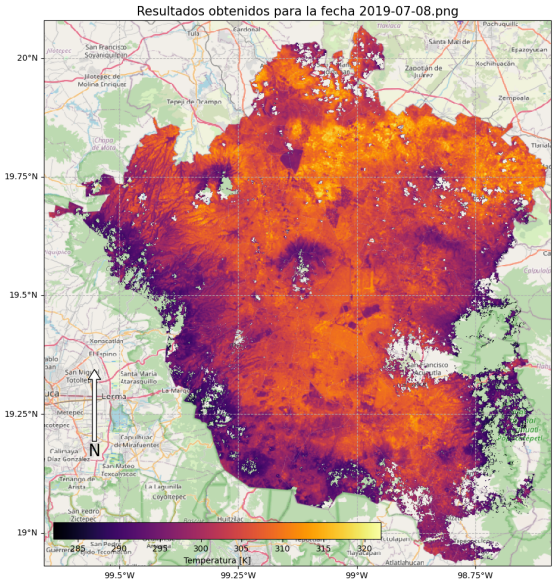
(c) 2019-05-21



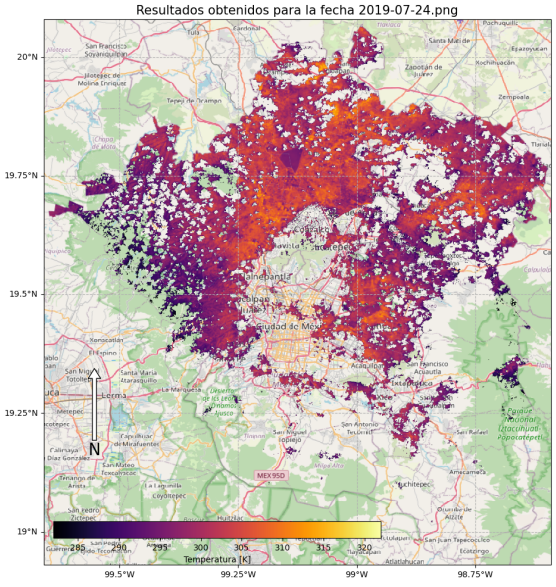
(d) 2019-06-06



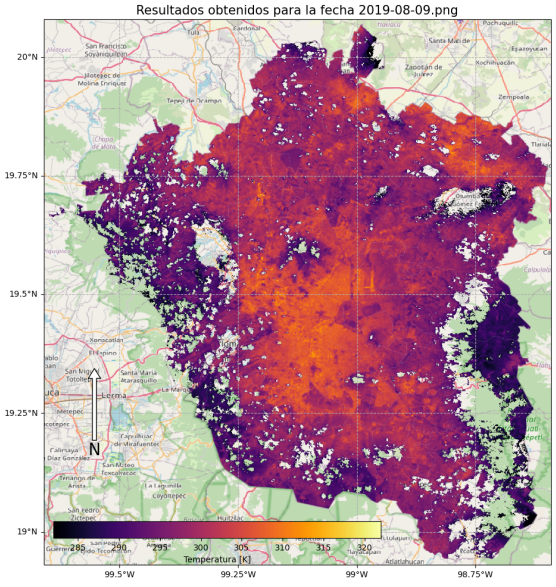
(a) 2019-06-22



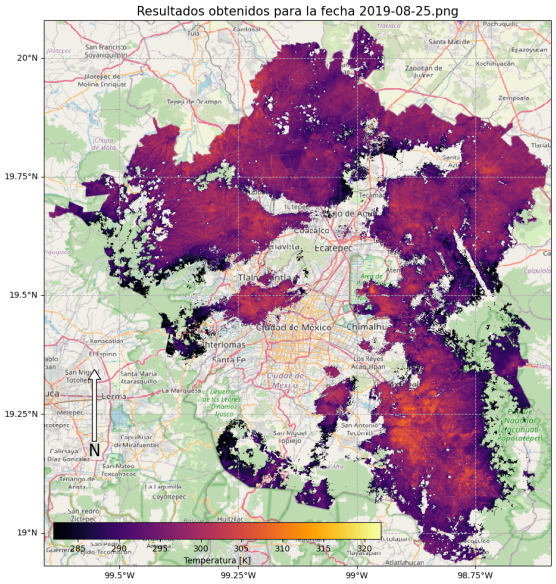
(b) 2019-07-08



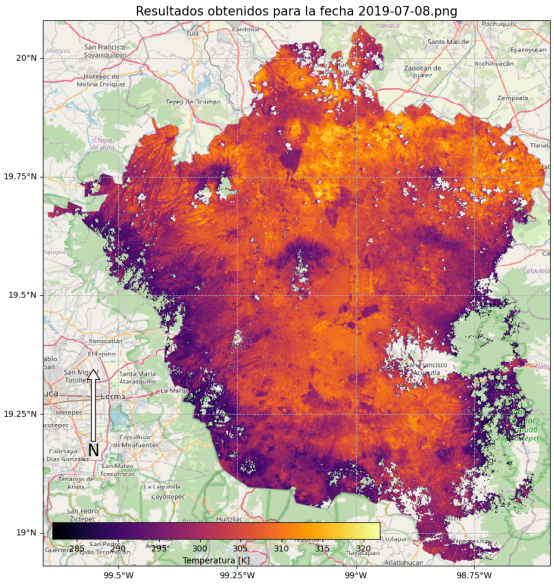
(c) 2019-07-24



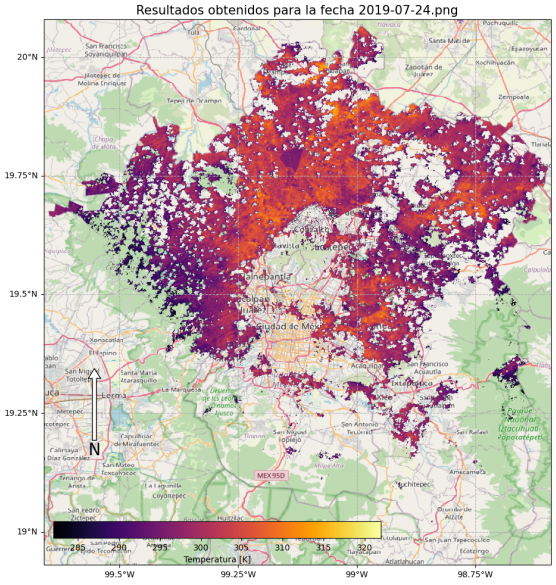
(d) 2019-08-09



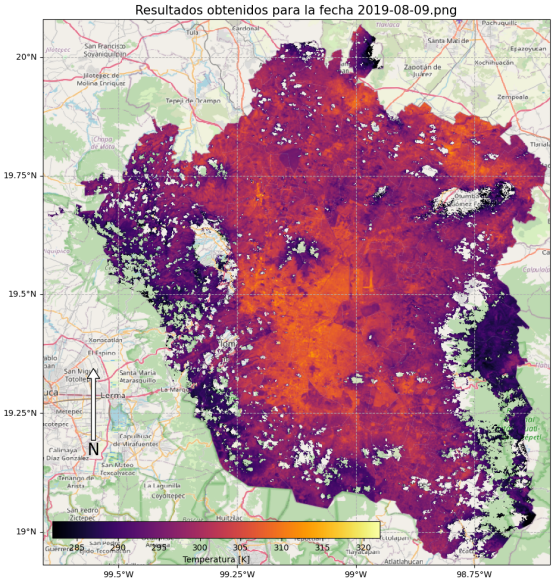
(a) 2019-08-25



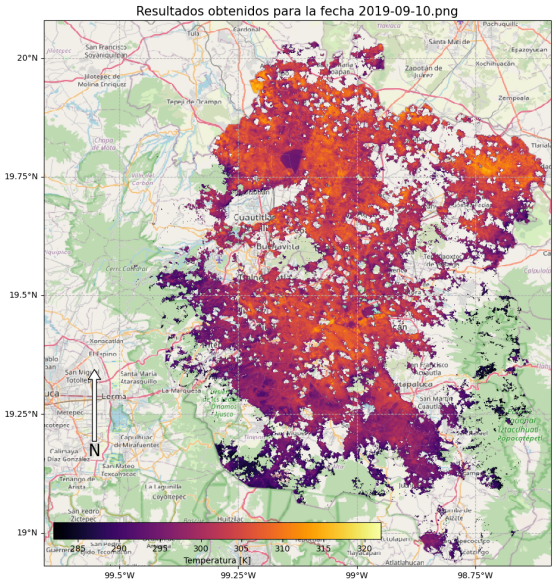
(b) 2019-07-08



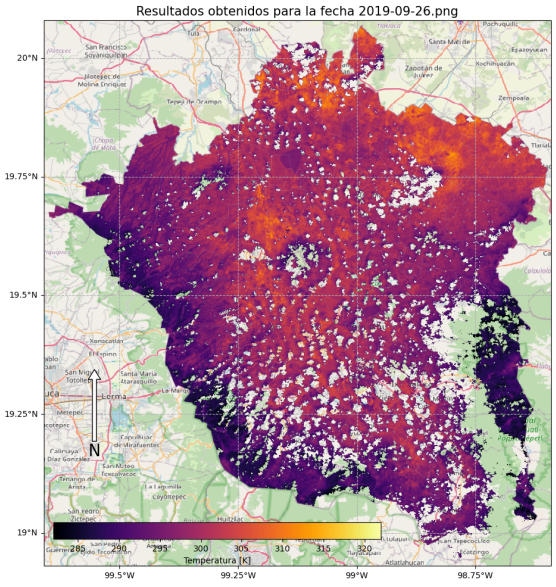
(c) 2019-07-24



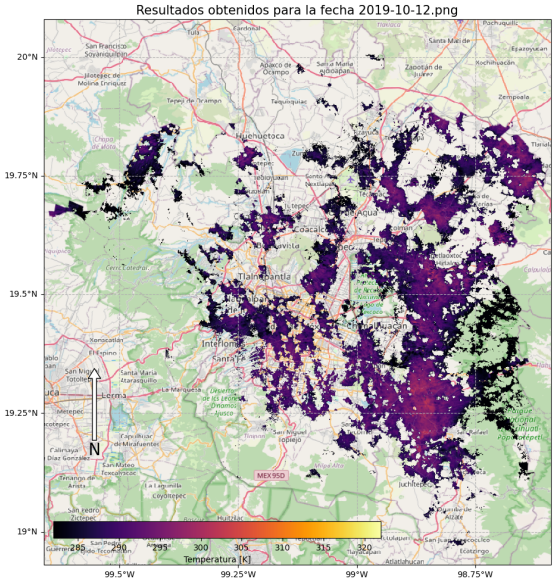
(d) 2019-08-09



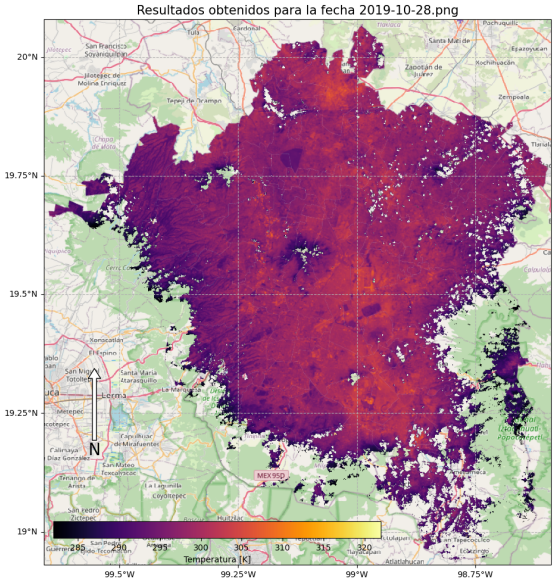
(a) 2019-09-10



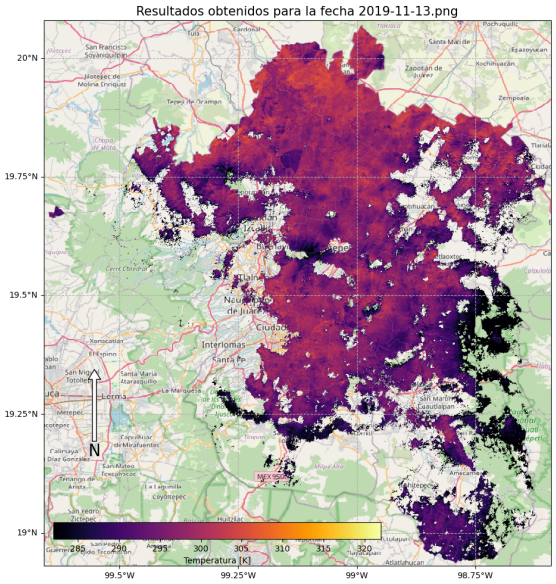
(b) 2019-09-26



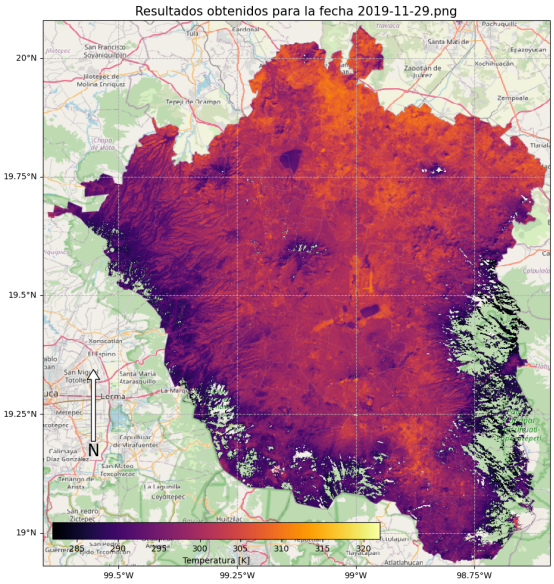
(c) 2019-10-12



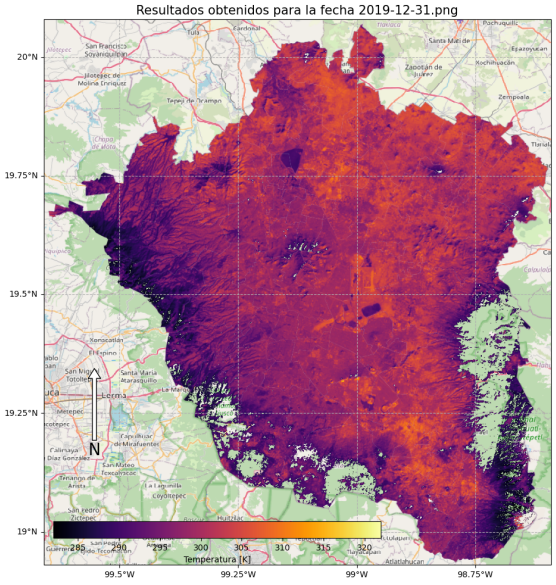
(d) 2019-10-28



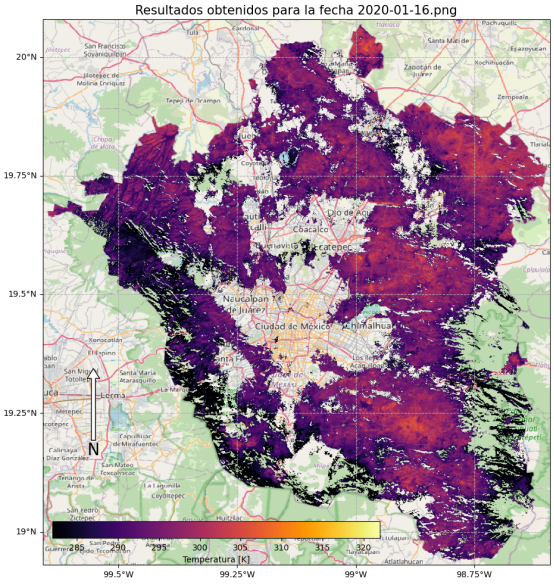
(a) 2019-11-13



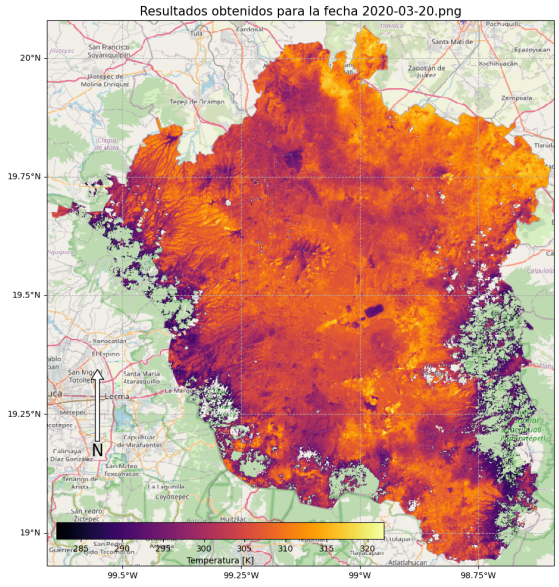
(b) 2019-11-29



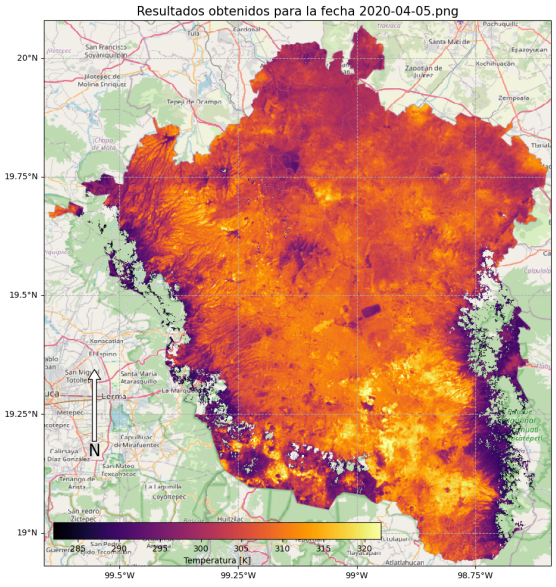
(c) 2019-12-31



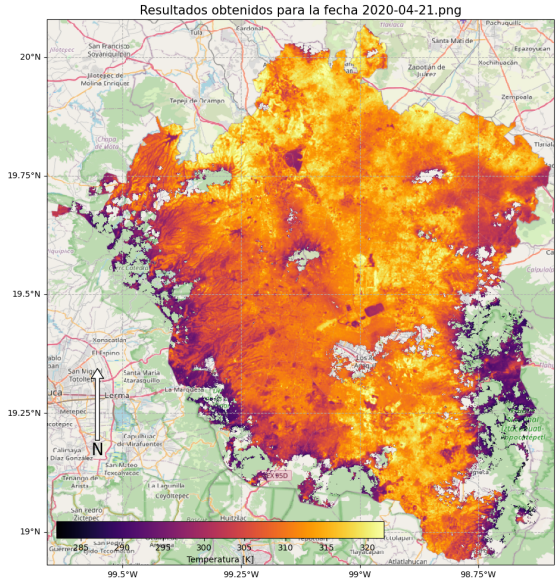
(d) 2020-01-16



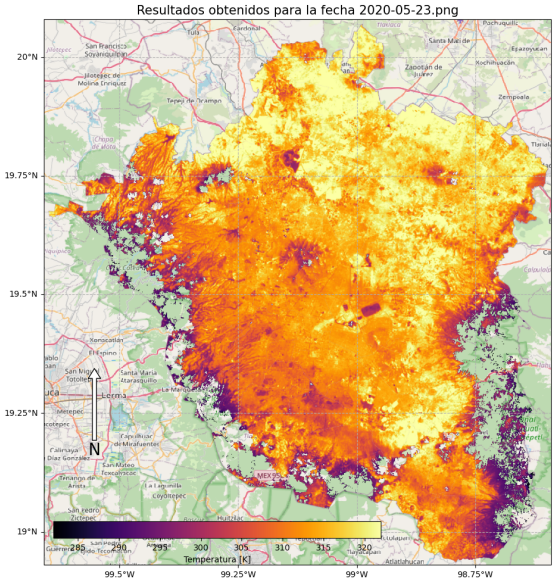
(a) 2020-03-20



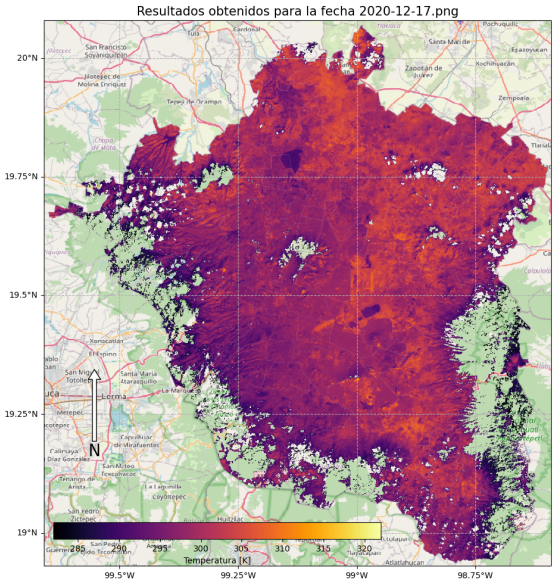
(b) 2020-04-05



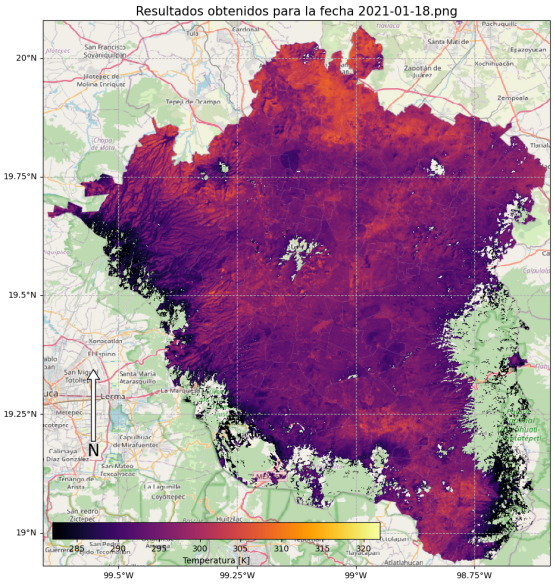
(c) 2020-04-21



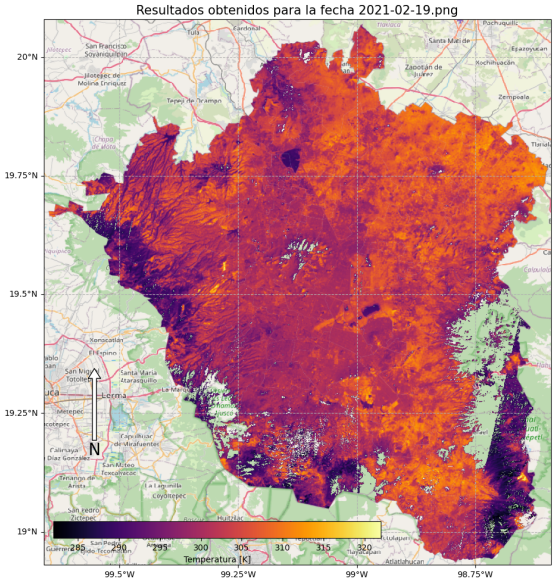
(d) 2020-05-23



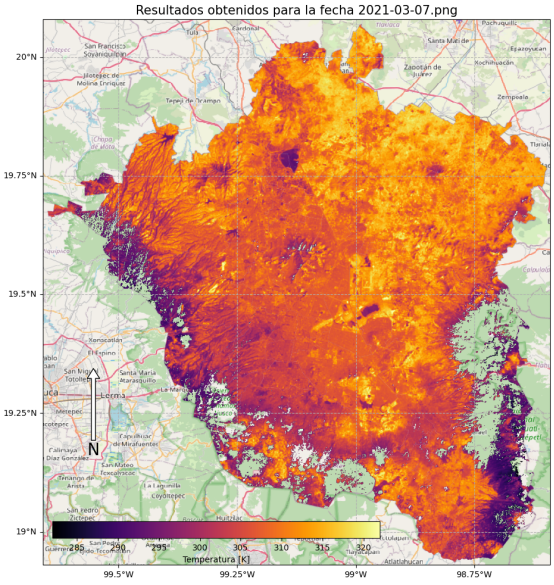
(a) 2020-12-17



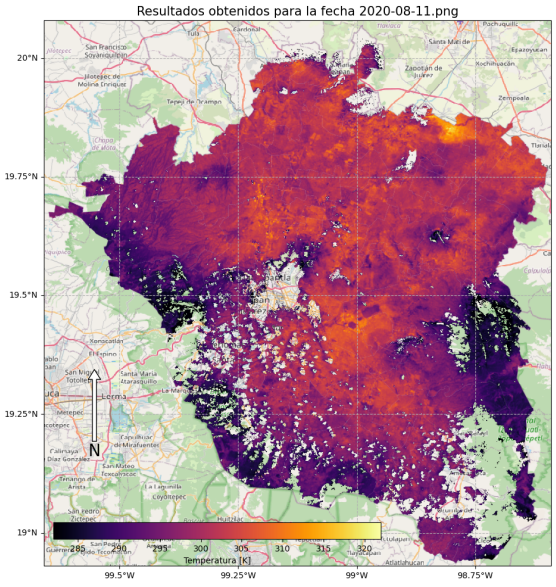
(b) 2021-01-18



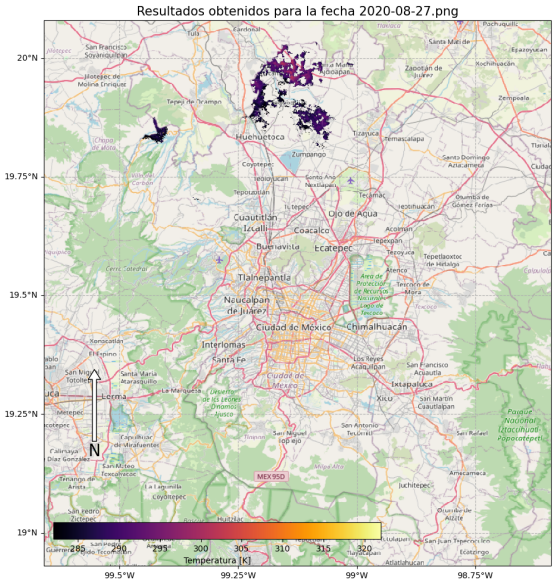
(c) 2021-02-19



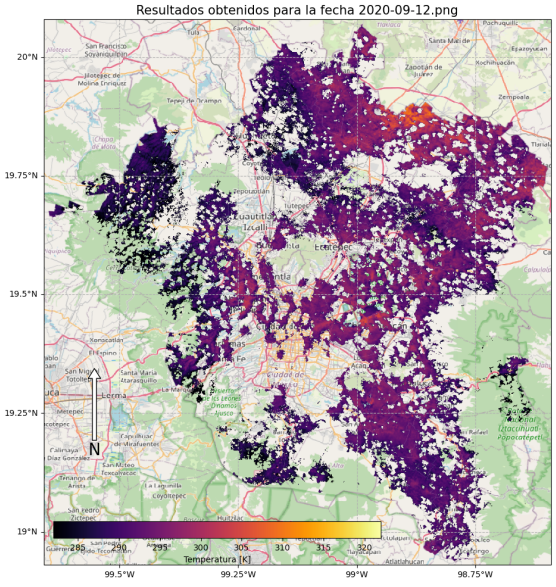
(d) 2021-03-07



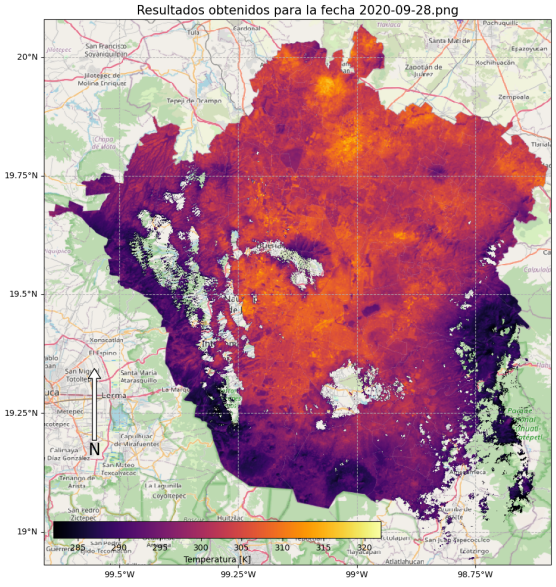
(a) 2020-08-11



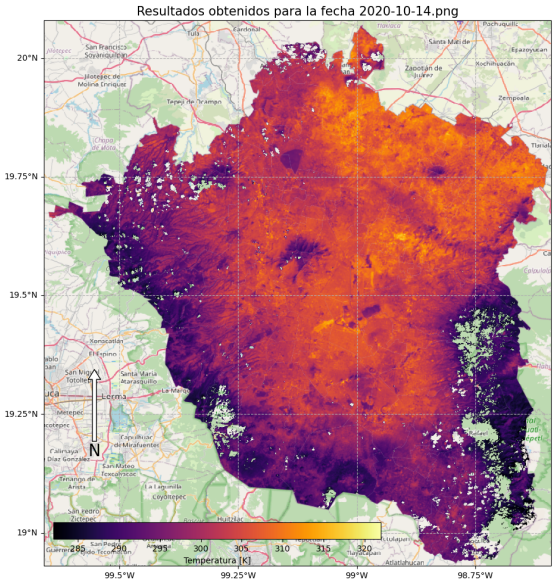
(b) 2020-08-27



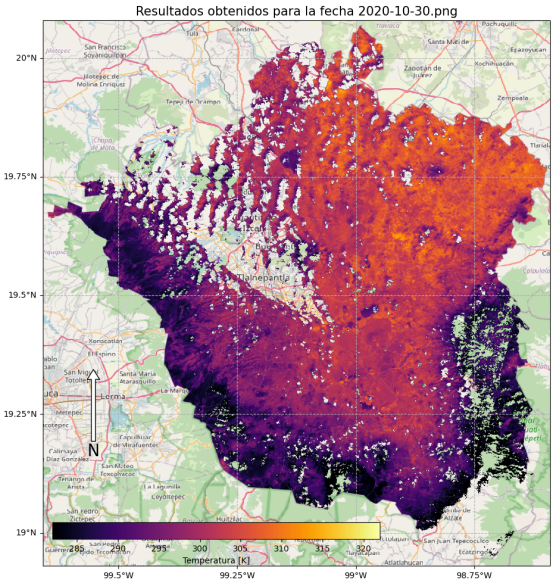
(c) 2020-09-12



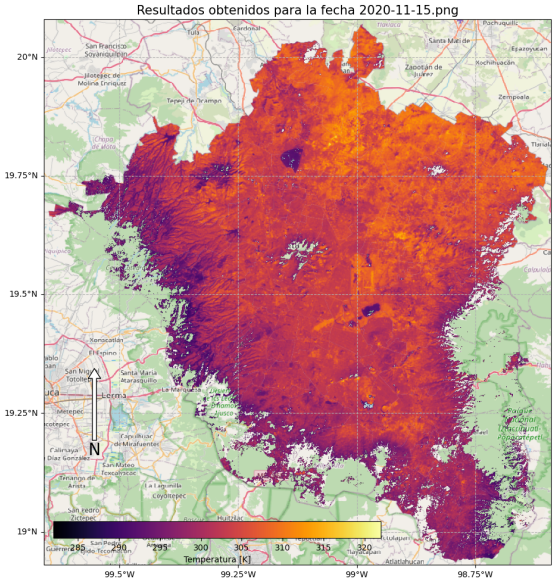
(d) 2020-09-28



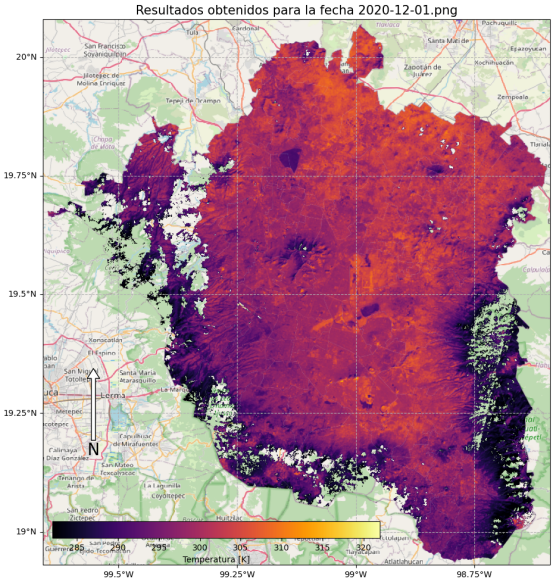
(a) 2020-10-14



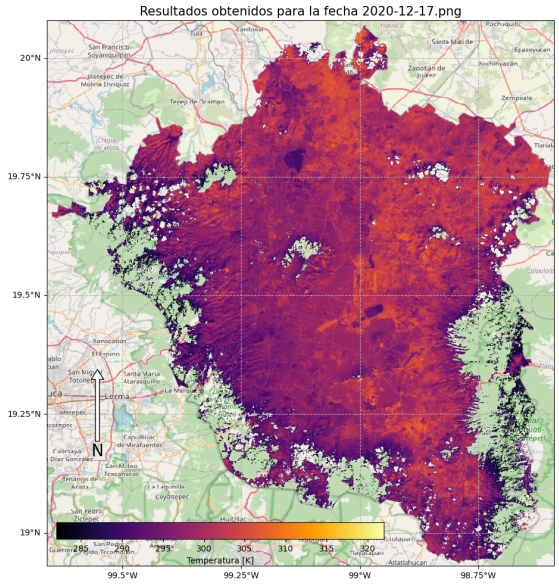
(b) 2020-10-30



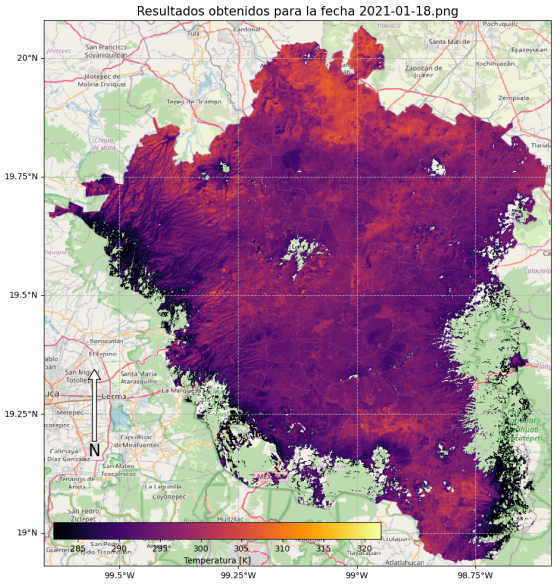
(c) 2020-11-15



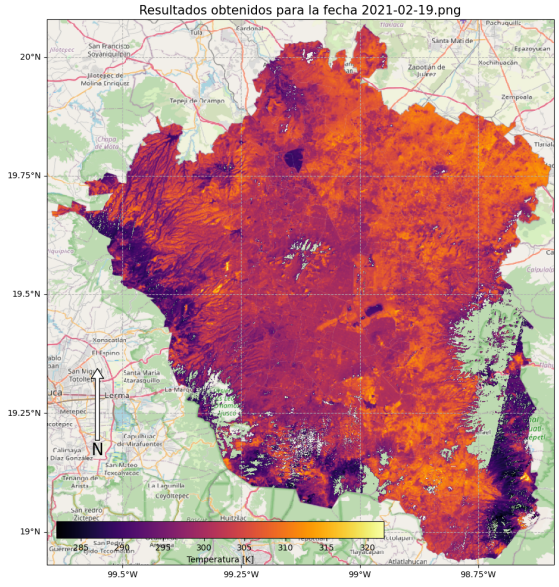
(d) 2020-12-01



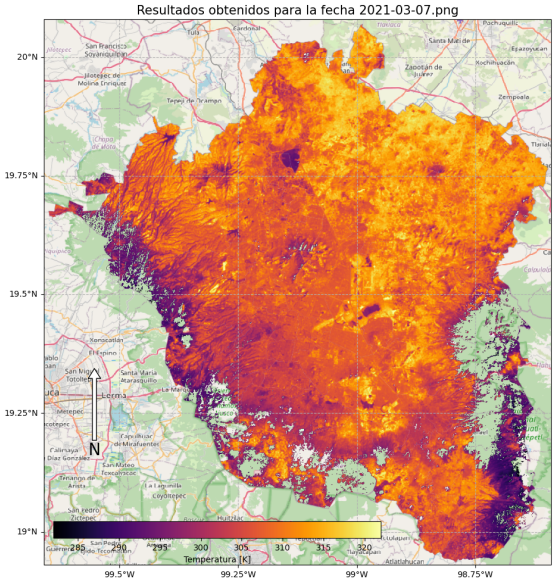
(a) 2020-12-17



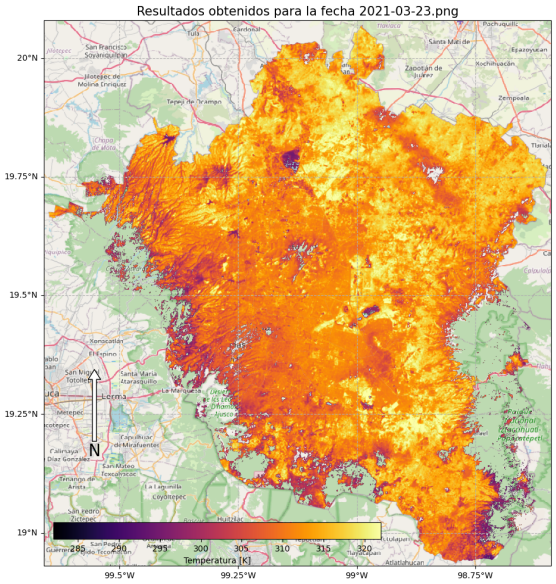
(b) 2021-01-18



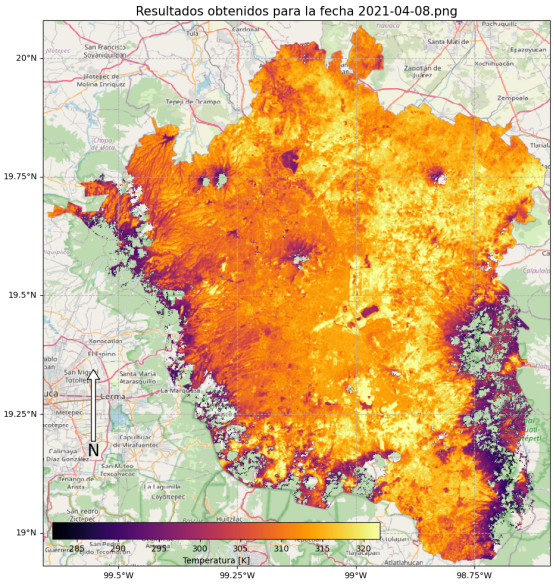
(c) 2021-02-19



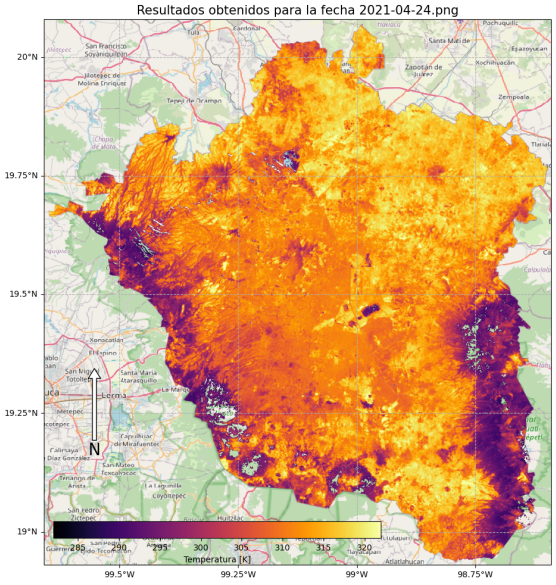
(d) 2021-03-07



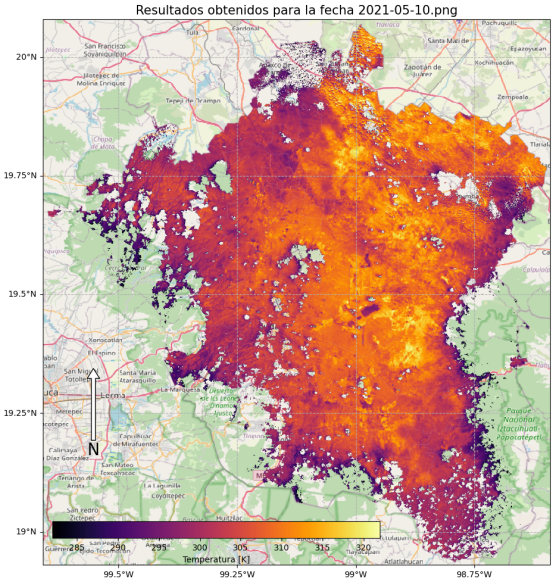
(a) 2021-03-23



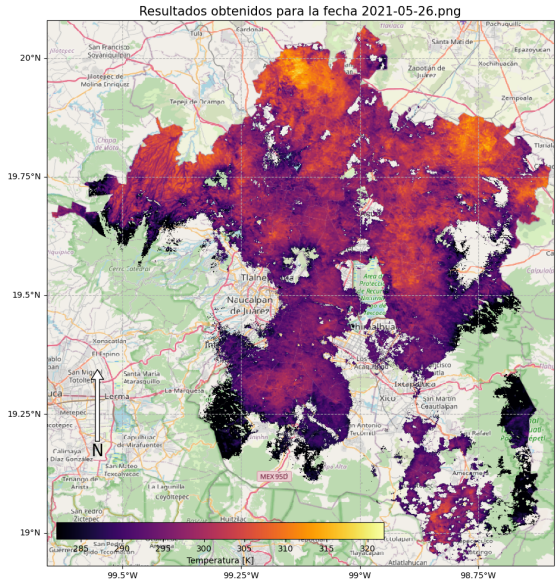
(b) 2021-04-08



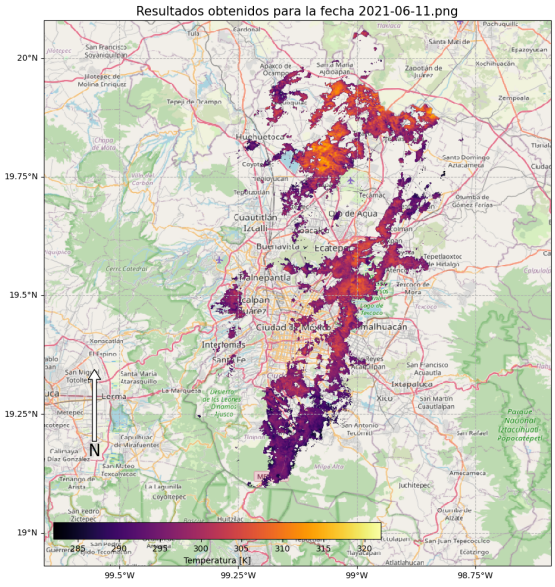
(c) 2021-04-24



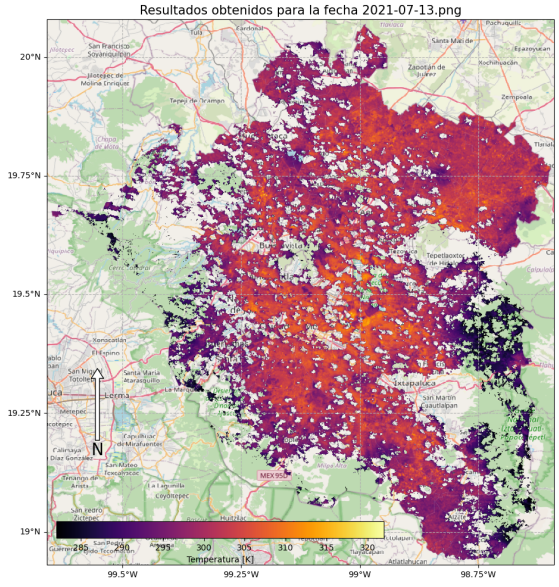
(d) 2021-05-10



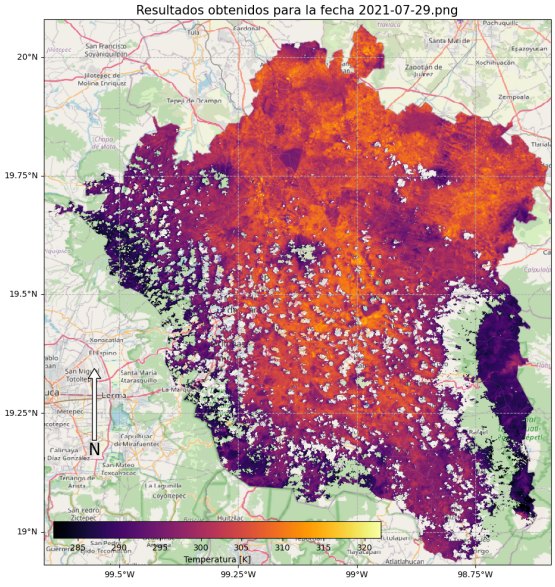
(a) 2021-05-26



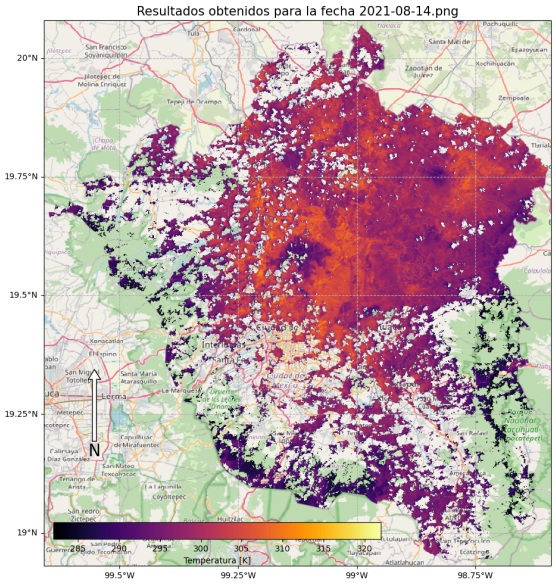
(b) 2021-06-11



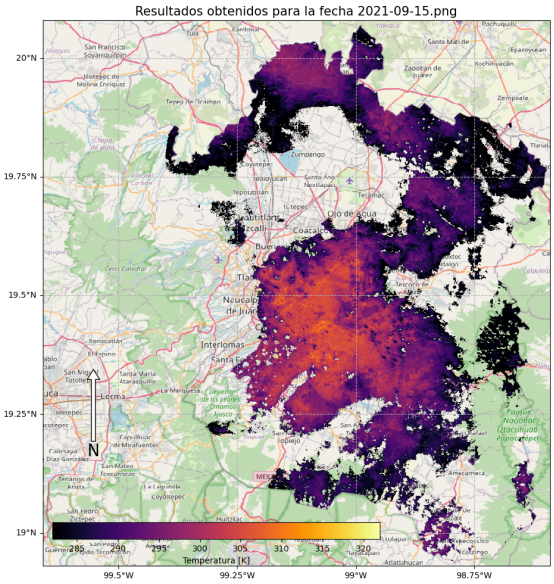
(c) 2021-07-13



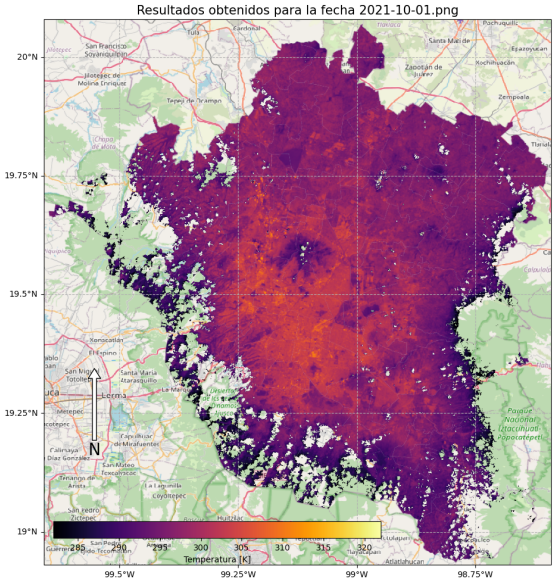
(d) 2021-07-29



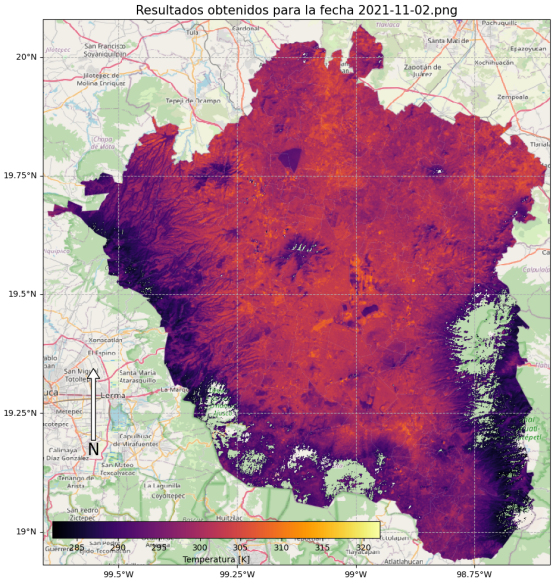
(a) 2021-08-14



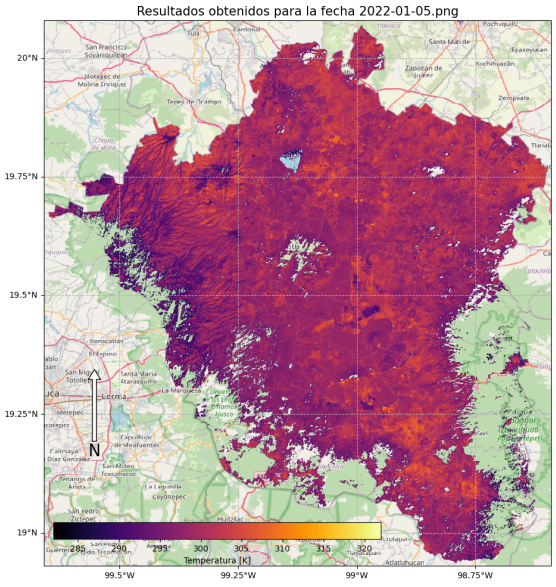
(b) 2021-09-15



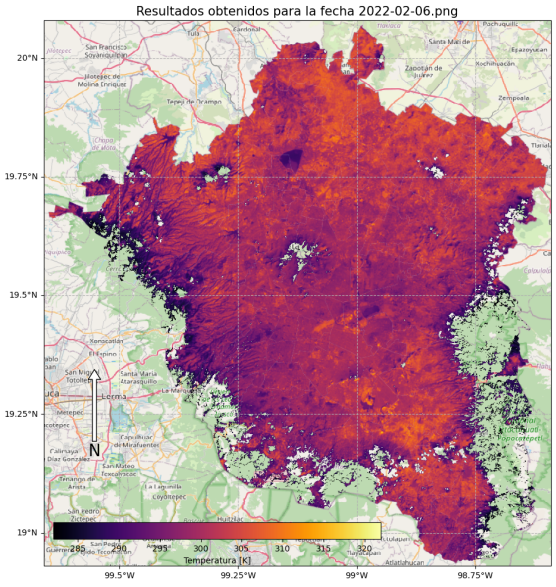
(c) 2021-10-01



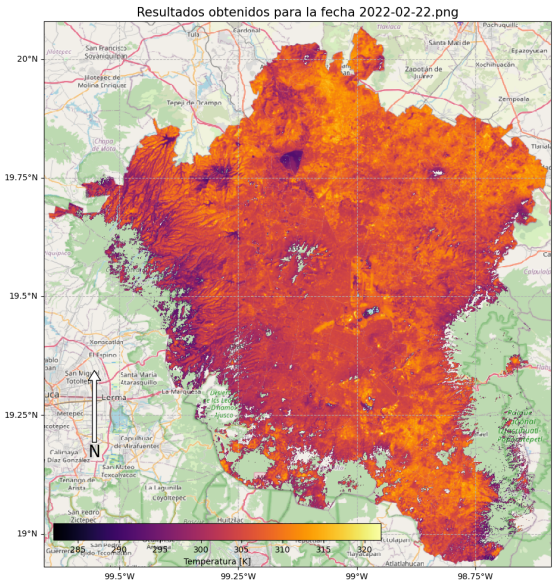
(d) 2021-11-02



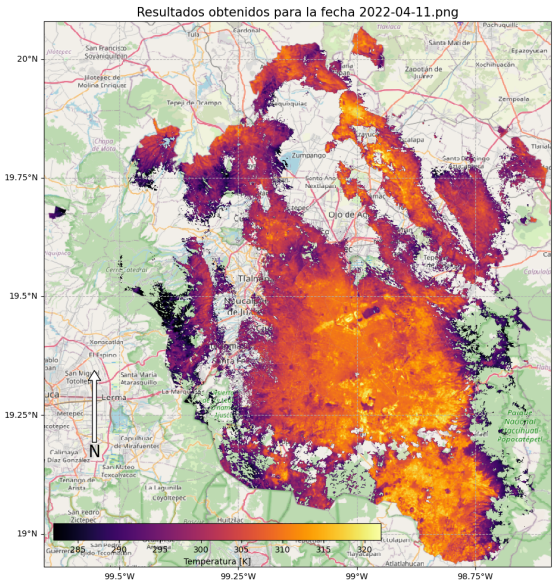
(a) 2022-01-05



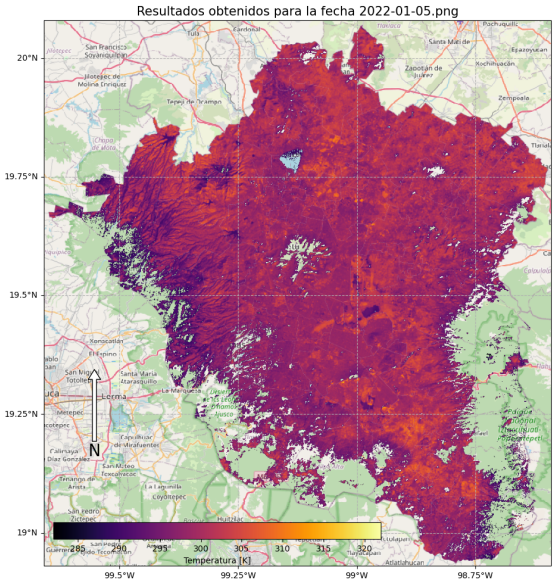
(b) 2022-02-06



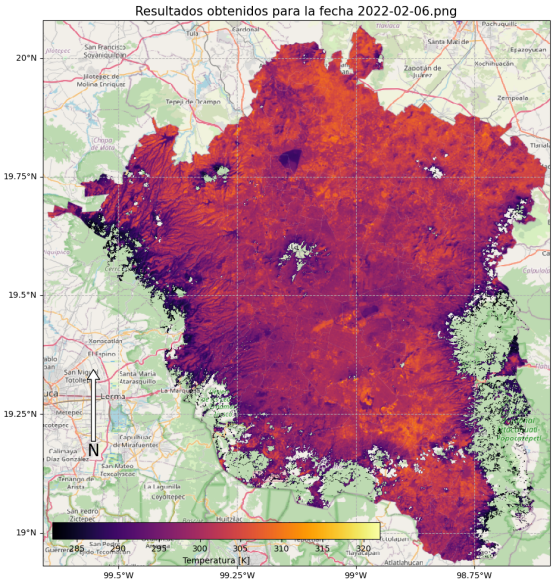
(c) 2022-02-22



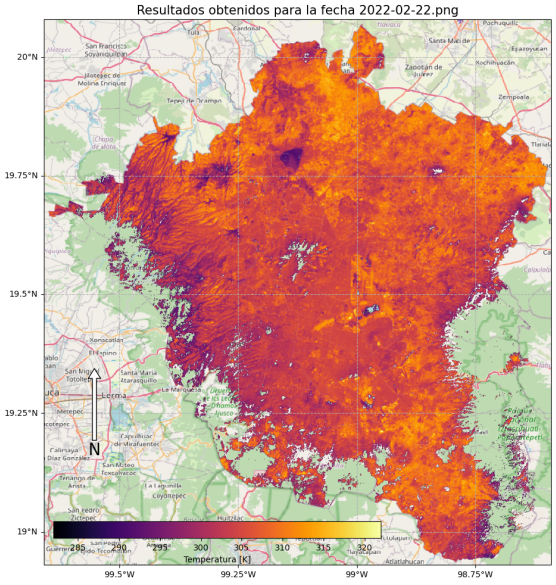
(d) 2022-04-11



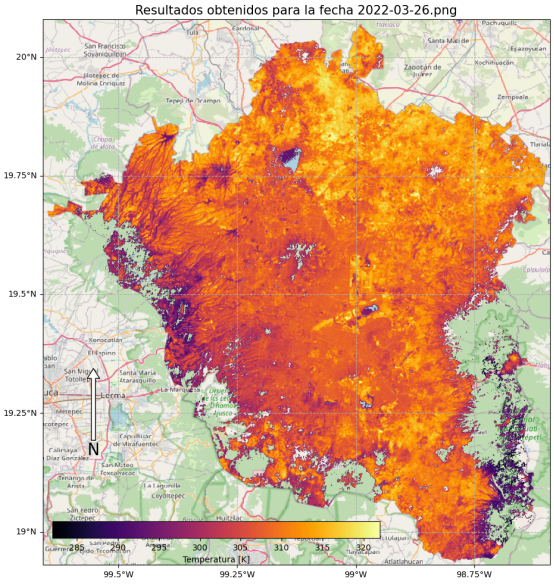
(a) 2022-01-05



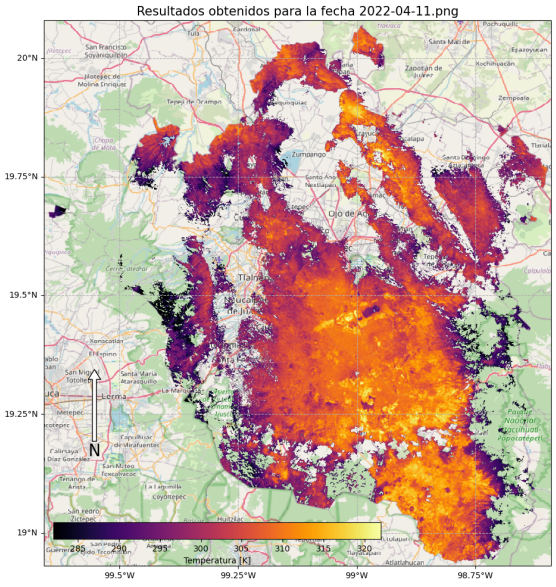
(b) 2022-02-06



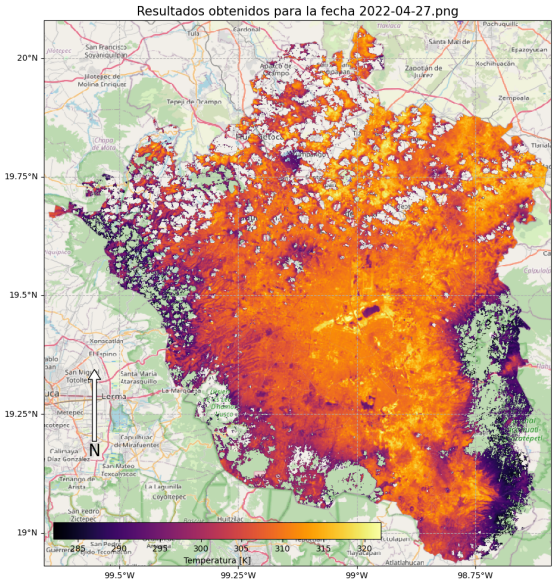
(c) 2022-02-22



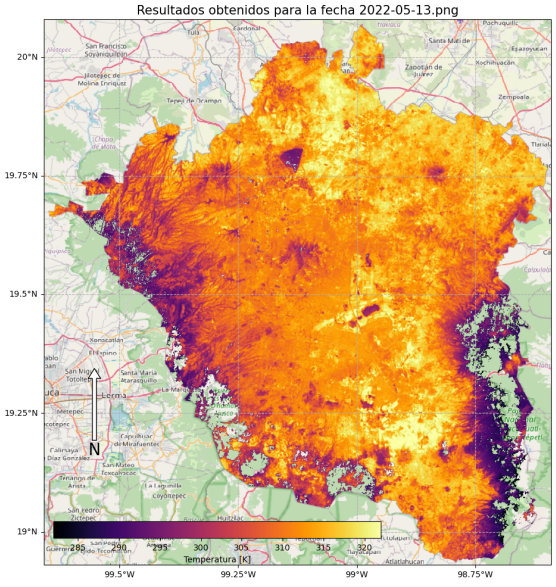
(d) 2022-03-26



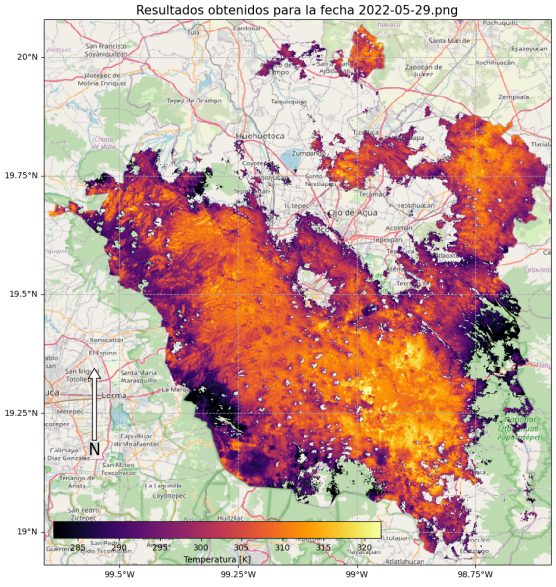
(a) 2022-04-11



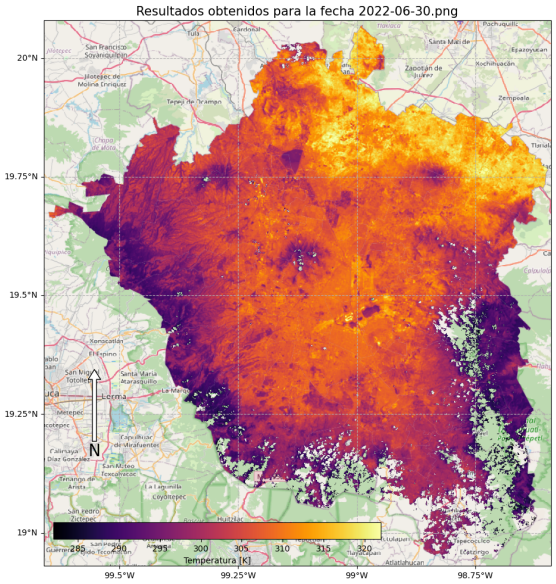
(b) 2022-04-27



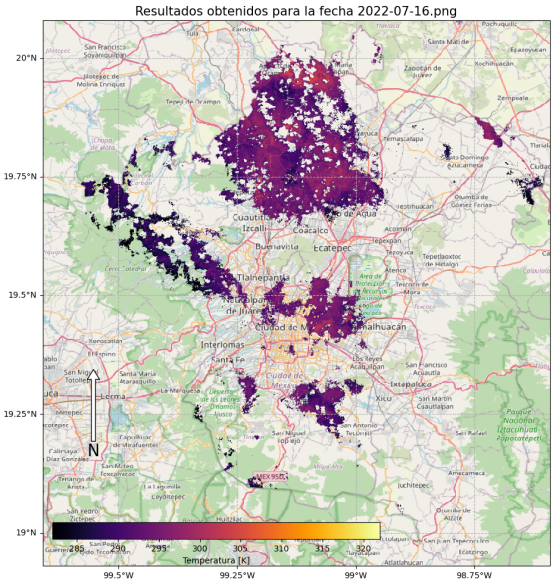
(c) 2022-05-13



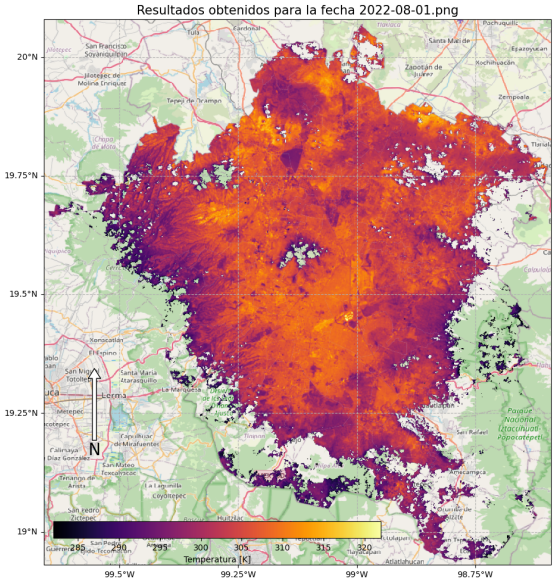
(d) 2022-05-29



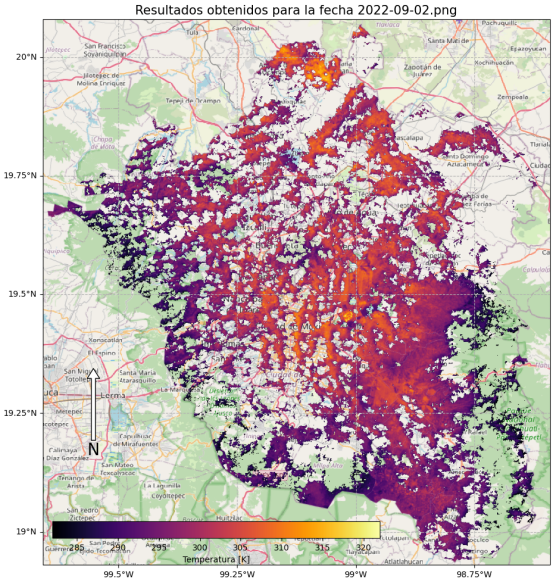
(a) 2022-06-30



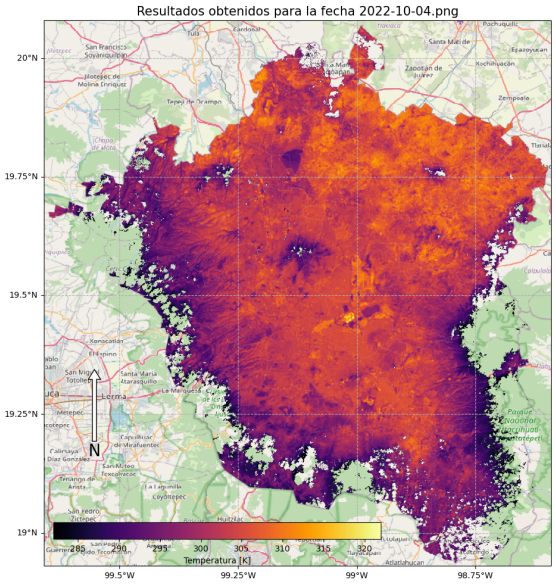
(b) 2022-07-16



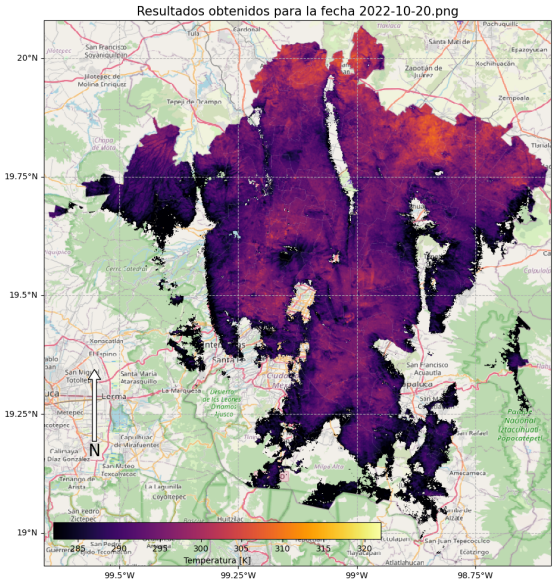
(c) 2022-08-01



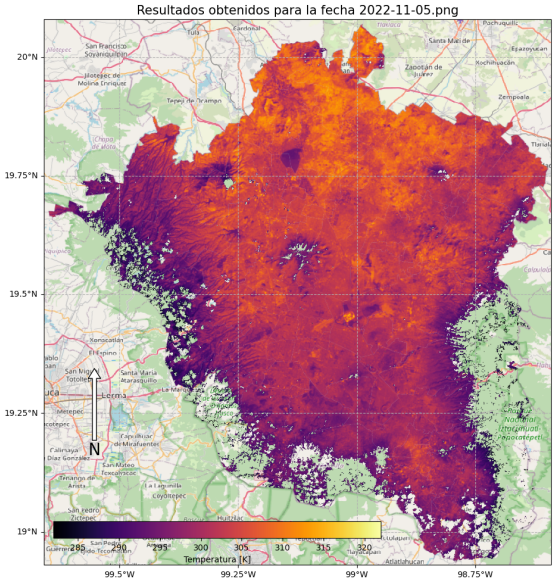
(d) 2022-09-02



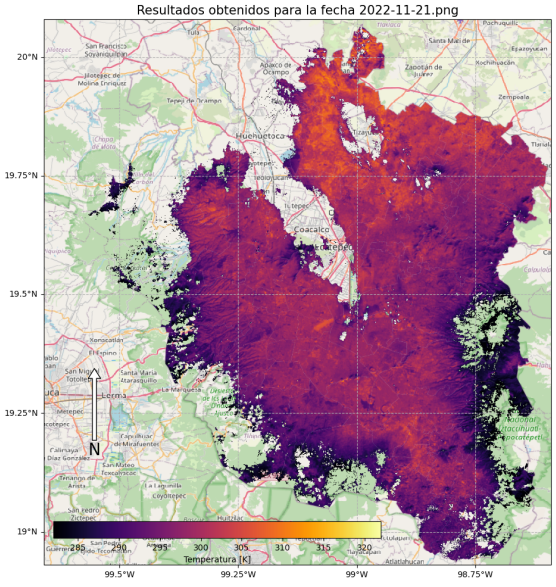
(a) 2022-10-04



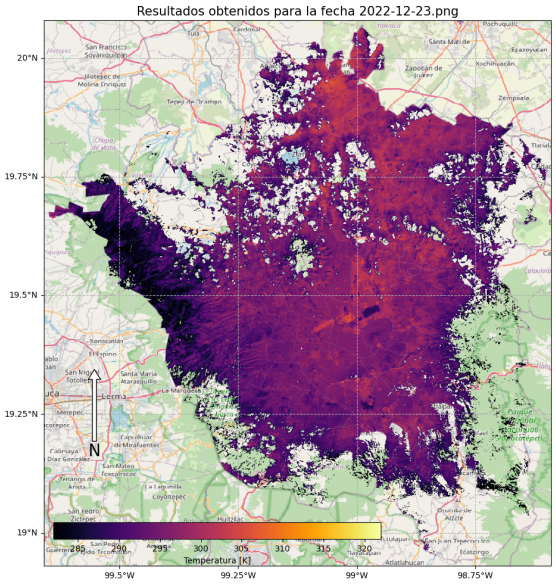
(b) 2022-10-20



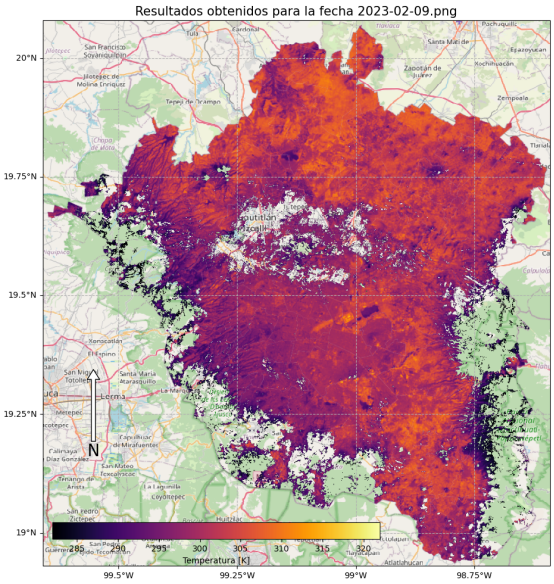
(c) 2022-11-05



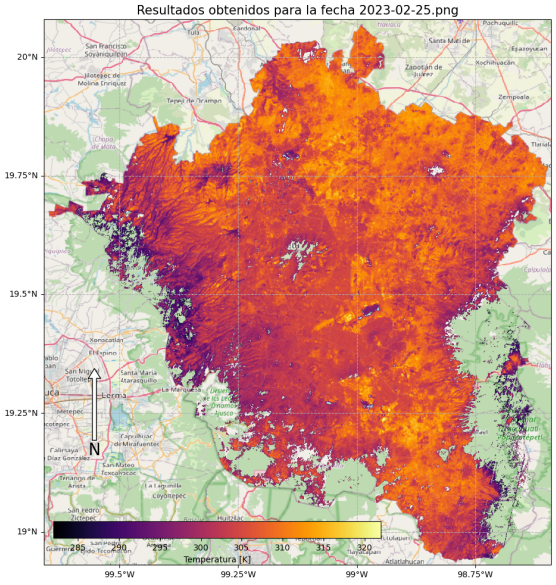
(d) 2022-11-21



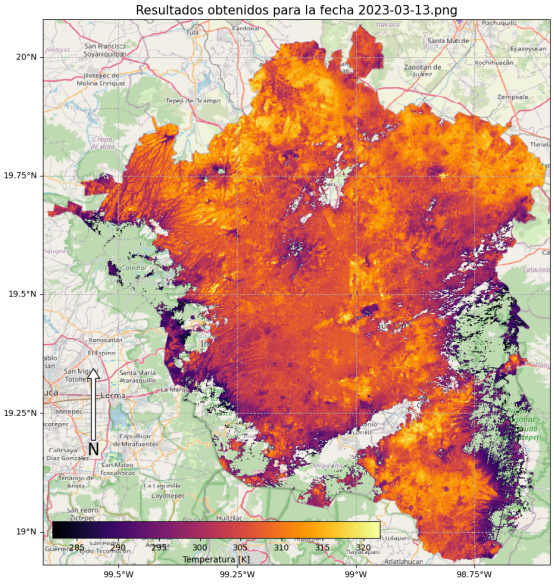
(a) 2022-12-23



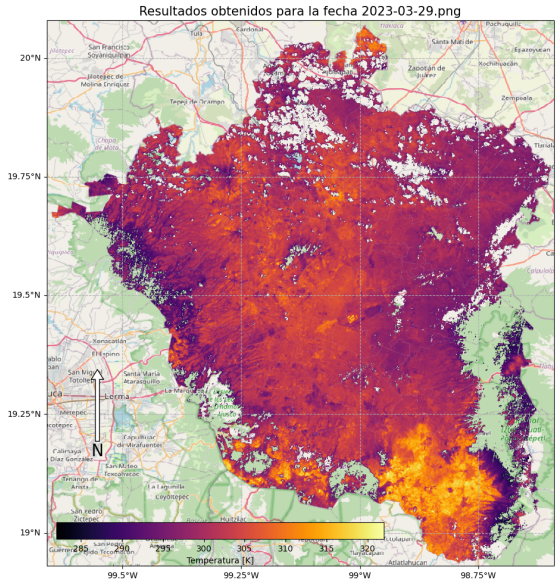
(b) 2023-02-09



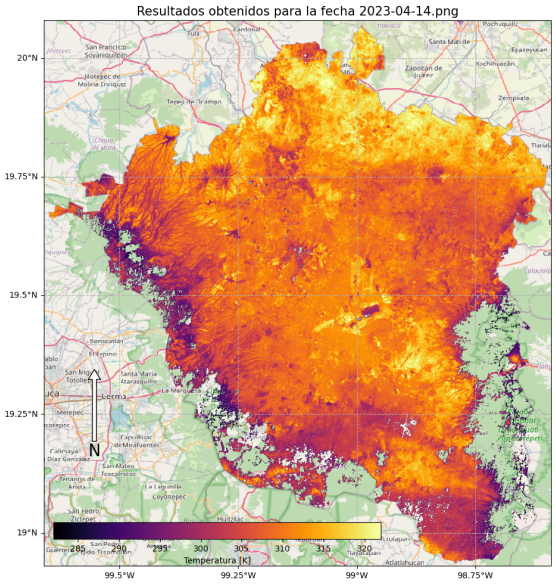
(c) 2023-02-25



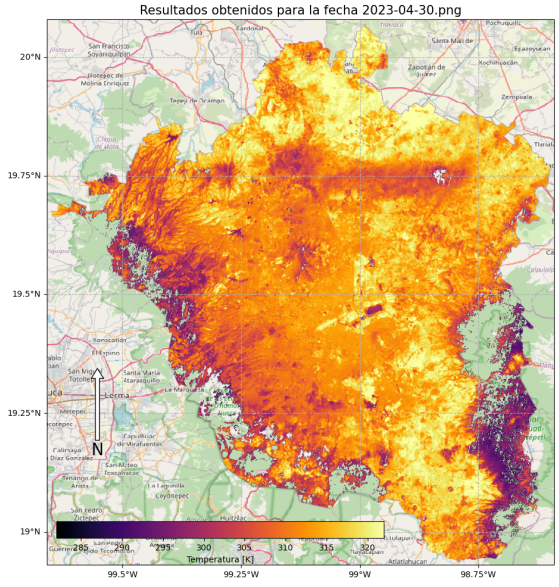
(d) 2023-03-13



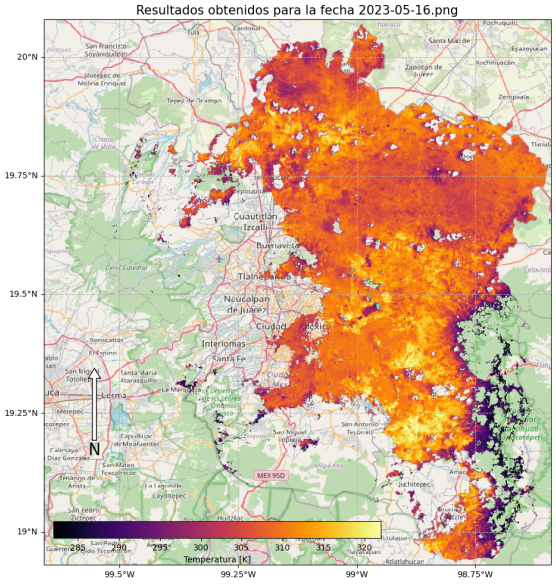
(a) 2023-03-29



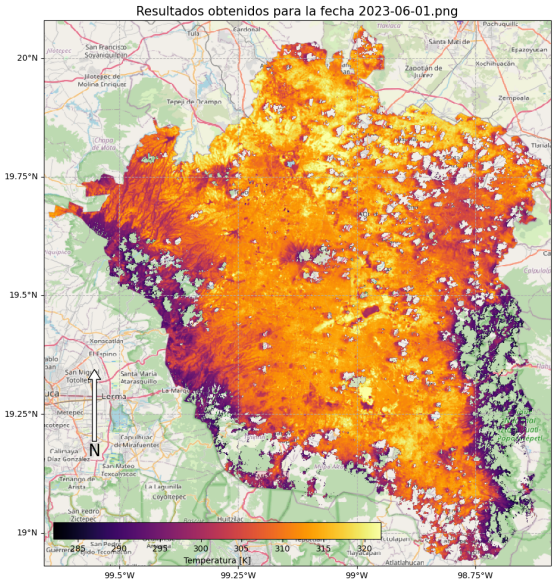
(b) 2023-04-14



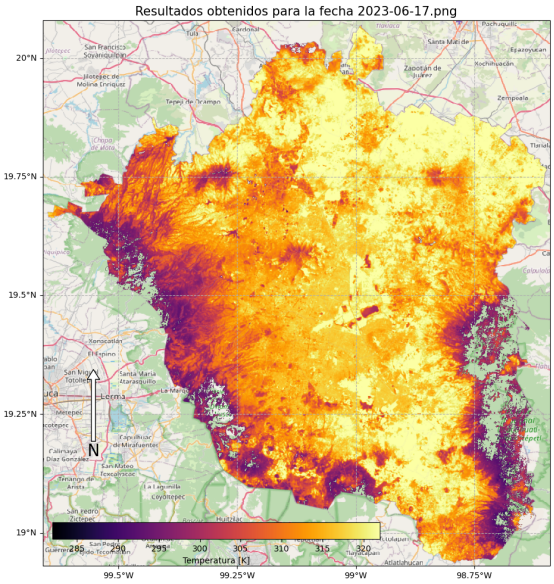
(c) 2023-04-30



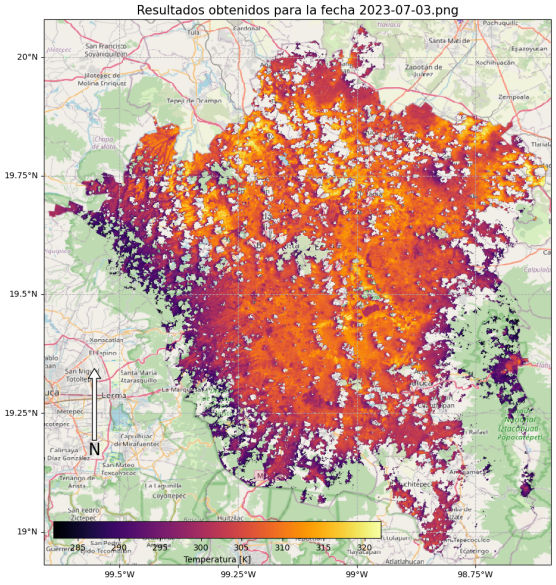
(d) 2023-05-16



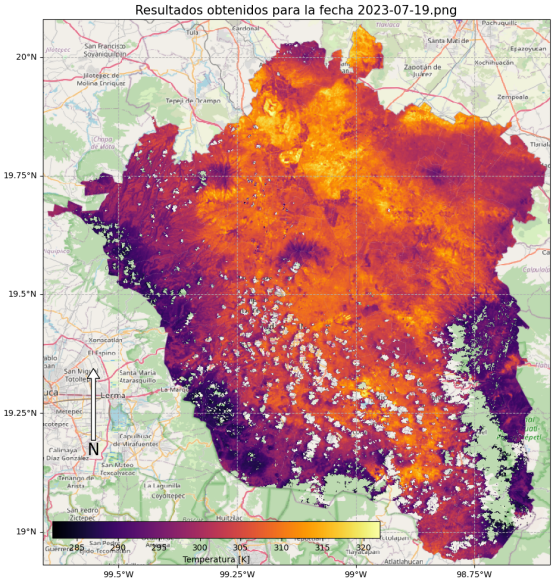
(a) 2023-06-01



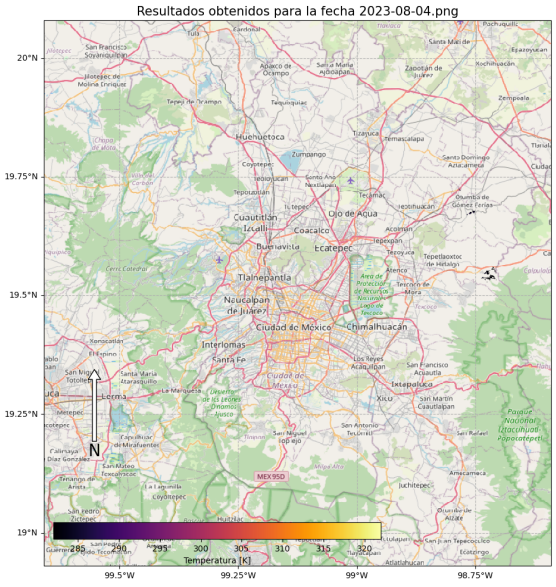
(b) 2023-06-17



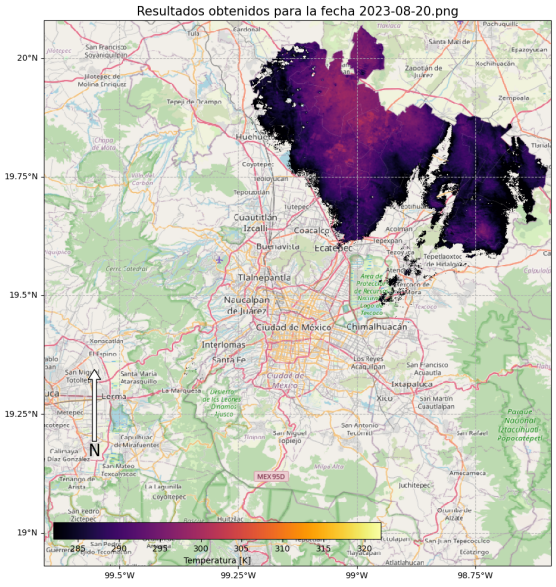
(c) 2023-07-03



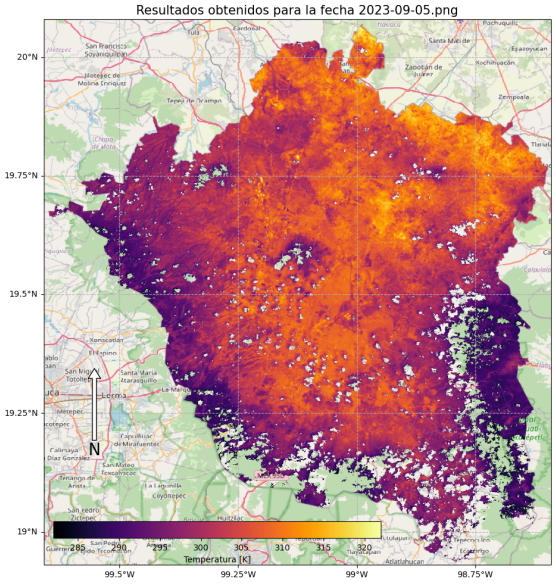
(d) 2023-07-19



(a) 2023-08-04



(b) 2023-08-20



(c) 2023-09-05

Apéndice B

Scripts de Google Earth Engine

En esta sección de apéndice, se presentan los diversos códigos de scripts utilizados en la interfaz de Google Earth Engine que se utilizaron para llevar a cabo los diferentes análisis geoespaciales relevantes en esta investigación. Estos scripts proporcionan una base técnica para la implementación y ejecución de diversas tareas dentro del contexto de este trabajo.

Es importante tener en cuenta que, si bien los lectores no necesitan un profundo conocimiento previo de programación o Google Earth Engine para comprender la investigación, se asume que tienen cierta familiaridad con los conceptos básicos de análisis geoespacial y programación.

Cada código de script se presenta con un título descriptivo que refleja su función principal en el contexto del análisis. Junto con el código en sí, se proporcionan comentarios explicativos para resaltar los pasos clave y facilitar la comprensión de la lógica detrás de cada implementación.

Los códigos de scripts presentados en este apéndice representan un componente esencial para la reproducibilidad de los resultados y la validación de los hallazgos. Además, demuestran la aplicación práctica de los conceptos teóricos en un entorno real de análisis geoespacial.

A través de esta sección, se pretende proporcionar una guía detallada que permita a otros investigadores y profesionales utilizar los códigos de scripts como punto de partida para sus propias investigaciones y aplicaciones prácticas en el campo de la geomática.

Scripts Principales

SplitWindow

```
1 /*
```

```
2  * Este archivo es el principal para calcular los valores de LST, por
       medio del algoritmo de "Split-window" elaborado por Jiménez Muñ
     os et. al. En este script solamente es donde ocurre la
     representación visual.
3  * Se divide en 3 grandes bloques:
4  * 1) Importación de los scripts.
5  * 2) Realización del cálculo del LST
6  * 3) Representación visual
7 */
8
9 /*
10 * BLOQUE I: Importación de los scripts que hacen el cálculo
11 * -swLST: Algoritmo Principla
12 * -mosaicDate: Unión de imágenes
13 * -cloudcoverroi: Cálculo de nubosidad
14 * -iMaxMin: Obtiene valores máximos y mínimos
15 * -contour: Genera curvas de nivel
16 */
17 var swLST = require('users/edgarpzavaleta/Tesis:Modules/SWAlgorithm')
     ;
18 var mosaicDate = require('users/edgarpzavaleta/Tesis:Help/datemosaic'
     );
19 var cloudcoverroi = require('users/edgarpzavaleta/Tesis:Modules/
     CloudCoverROI');
20 var iMaxMin = require('users/edgarpzavaleta/Tesis:Help/minmax');
21 var countour = require('users/edgarpzavaleta/Tesis:Help/countours');
22
23 /*
24 * BLOQUE II: Realización del cálculo del LST.
25 * Esta parte requiere 4 datos de entrada.
26 * -geometry: geometría que sirve para seleccionar la zona de estudio
       .
27 * -satellite: versión de satélite de Landsat. Si bien es  compatible
       desde Landsat 4, está pensado y  optimizado para Landsat 8.
     Landsat 9 no está desarrollado
28 * startDate: fecha inicial
29 * endDate: fecha final.
30 * En esta parte pueden surgir errores por asignar una fecha de
     manera errónea. Esto incluye poner fechas inexistentes (31 de
     abril), o  escoger la fecha de manera precisa. Si se conoce el dí
     a se debe de poner un día antes como inicial y un día después
     como final.
31 * Posterior a este cálculo se filtran las imágenes por su "path" y
     se unen por fecha. Esto se hizo por las particularidades de la
```

```
    región. En caso de ser que no exista algún inconveniente,
    comentar la sección A y des-comentar la sección B. Se agrega el
    porcentaje de nubosidad como si fuesen metadatos.
32  * Después se queda solamente con las imágenes LST. Se seleccionan
    aquellas con base en su porcentaje de nubosidad.
33  */
34  var geometry = ee.FeatureCollection("users/edgarpzavaleta/ZMVMWGS84")
    ;
35  var satellite = 'L8';
36  var startDate = '2014-01-01';
37  var endDate = '2014-12-31';
38  var landsatColl = swLST.collection(satellite, startDate, endDate,
    geometry);
39
40  //Sección A
41  landsatColl = landsatColl.filter(ee.Filter.eq('WRS_PATH', 26));//
    Filtro por "path". Comentar en caso de desconocer o querer una
    zona más grande
42  var mosaicColl = mosaicDate.collection(landsatColl); // Unión de imá
    genes
43  mosaicColl = mosaicColl.map(cloudcoverroi.fclip(geometry)); //
    Recorte de zona
44  //Sección B
45  //var mosaicColl = landsatColl.map(cloudcoverroi.fclip(geometry)); //
    Recorte de zona
46
47  var LSTColl = mosaicColl.select('LST');
48  LSTColl = LSTColl.map(cloudcoverroi.percCover(geometry)); // Calculo
    de Nubosidad
49  LSTColl = LSTColl.filter(ee.Filter.gte('IMAGE_PERC_COVER', 80)); //
    Porcentaje de Imagen
50  // print(LSTColl);// Imágenes
51
52  /*
53   * BLOQUE III: Visualización del cálculo del LST.
54   * En esta parte es donde se puede visualizar las imágenes. Se han
    creado dos paletas de colores tomando como referencia "lava" e "
    inferno" de Matplotlib.
55   * De nuevo existen dos forma de ejecución, la primera es escoger la
    imagen y la segunda obtener una imagen media. En ambos casos se
    calculan los valores máximos y mínimos,  para que de forma automá
    tica asigne a la imagen. Centra en la ZMVM y muestra la imagen
56   * Posterior a esto se divide en otros dos procesos.
```

```
57  * Proceso A) Obtener máximos y mínimos de manera automática. Sirve
        cuando se hace el primer acercamiento y de esta manera ejecutar
        el proceso B. Es un proceso mucho más lento y puede llegar a
        trabarse.
58  * Proceso B) Obtiene imagen e isotermas. Es hasta 15 veces más rá
        pido.
59  */
60
61  // Paletas de colores
62  var lava = ['000004', '1B0C41', '4A0C6B', '781C6D', '52C60', 'CF4446'
        , 'ED6925', 'FB9B06', 'F7D13D', 'FCFFA4'];
63  var inferno = ['000003', '160b39', '410967','6a176e', '932567', '
        ba3655', 'dc5039', 'f2751a', 'f6d542','fcfea4'];
64
65  Map.addLayer(geometry.draw('red'),{'opacity': 0.3},'Zona
        Metropolitana',0);
66  Map.centerObject(geometry,10);
67
68  // Sección A: Primera imagen o fecha exacta
69  var image1 = LSTColl.first().clip(geometry);
70  // Sección B: Imagen media
71  // var image1 = LSTColl.mean().clip(geometry);
72
73  // Proceso A: Máximos y mínimos
74  /*
75  var maxmin = iMaxMin.MinMax(image1, 'LST', geometry, 0, 100);
76  print(maxmin);
77  maxmin.evaluate(function (LSTmaxmin) {
78    var BandCompViz = {
79      bands:["LST"],
80      min: LSTmaxmin.min,
81      max: LSTmaxmin.max,
82      palette: lava
83    };
84    Map.addLayer(image1, BandCompViz, 'LST');
85  });
86  */
87
88  // Proceso B: Imagen y Curvas de nivel
89  var contourlines = countour.countours(270, 370, 2, image1);
90  var viz = {bands:["LST"],
91             min: 282,
92             max: 322,
93             palette: inferno};
```

```
Map.addLayer(image1, viz, "LST");
Map.addLayer(contourlines, {min: 282, max: 322, palette:['000003'],
    opacity:0.3}, "Isoterma");
```

MonoWindow

```
/*
  Este archivo es el principal para calcular los valores de LST, por
     medio del algoritmo "Mono-window" En este script solamente es
     donde ocurre la representación visual. Se divide en 3 grandes
     bloques:
  1) Importación de scripts adicionales.
  2) Realización del cálculo del LST
  3) Representación visual
*/

/*
  BLOQUE I: Importación de los scripts que hacen el cálculo
  -landsatLST: Algoritmo Principla
  -mosaicDate: Unión de imágenes
  -cloudcoverroi: Cálculo de nubosidad
  -iMaxMin: Obtiene valores máximos y mínimos
  -contour: Genera curvas de nivel
  -cloudcoverroi: Cálculo de nubosidad
*/

var landsatLST = require('users/edgarpzavaleta/Tesis:Modules/
    Landsat_LST');
var mosaicDate = require('users/edgarpzavaleta/Tesis:Help/datemosaic'
    );
var countour = require('users/edgarpzavaleta/Tesis:Help/countours');
var cloudcoverroi = require('users/edgarpzavaleta/Tesis:Modules/
    CloudCoverROI');
var iMaxMin = require('users/edgarpzavaleta/Tesis:Help/minmax');

/*
 BLOQUE II: Realización del cálculo del LST.
 Esta parte requiere 4 datos de entrada.
 - geometry: geometría que sirve para seleccionar la zona de estudio.
 - satellite: versión de satélite de Landsat. Si bien es compatible
     desde Landsat 4, está pensado y optimizado para Landsat 8.
     Landsat 9 no está desarrollado
 - startDate: fecha inicial.
 - endDate: fecha final.
```

```
31  - use_ndvi: Recibe un dato booleano (true or false), si debe
        considerar el ndvi.
32  En esta parte pueden surgir errores por asignar una fecha de manerra
        erronea. Esto incluye poner fechas inexistentes (ëjemplo: 31 de
       abril), o escoger la fecha de manera precisa. Si se conoce el día
        se debe de poner un día antes como inicial y un día después como
        final.
33  Posterior a este calculo se filtran las imágenes por su "path" y se
       unen por fecha. Esto se hizo por las particularidades de la regió
       n. En caso de que no exista algún inconveniente comentar la secci
       ón A y descomentar la sección B. Se agrega el porcentaje de
       nubosida como metadato.
34  Después se queda solamente con las imágenes LST. Se seleccionan
       aquellas con base en su porcentaje de nubosidad.
35 */
36 var geometry = ee.FeatureCollection("users/edgarpzavaleta/ZMVMWGS84")
       ;
37 var satellite = 'L8';
38 // Estas fechas corresponden a la época húmeda para la ciudad de Mé
       xico del año 2018
39 var date_start = '2018-05-15';
40 var date_end = '2018-05-30';
41 var use_ndvi = true;
42
43 var landsatColl = landsatLST.collection(satellite, date_start,
       date_end, geometry, use_ndvi);
44
45 //Sección A
46 var landsatColl = landsatColl.filter(ee.Filter.eq('WRS_PATH', 26));//
        Filtro por "path"
47 var mosaicColl = mosaicDate.collection(landsatColl); // Union de imá
       genes
48 mosaicColl = mosaicColl.map(cloudcoverroi.fclip(geometry)); //
       Recorte de zona
49
50 //Sección B
51 //var mosaicColl = landsatColl.map(cloudcoverroi.fclip(geometry)); //
        Recorte de zona
52 var LSTColl = mosaicColl.select('LST');
53 LSTColl = LSTColl.map(cloudcoverroi.percCover(geometry)); // Calculo
       de Nubosidad
54 LSTColl = LSTColl.filter(ee.Filter.gte('IMAGE_PERC_COVER', 80)); //
       Porcentaje de Imagen
55
```

```
/*
 BLOQUE III: Visualización del cálculo del LST.
 En esta parte es donde se puede visualizar las imagánes. Se han creado dos paletas de colores tomando como referencia "lava" e "inferno" de Matplotlib.
 De nuevo existen dos forma de ejecución, la primera es escoger la imagen y la segunda obtener una imagen media. En ambos casos se calculan los valores máximos y mínimos, para que de forma automática asigne a la imagen. Centra en la ZMVM y muestra la imagen
 Posterior a esto se divide en otros dos procesos.
 Proceso A) Obtener máximos y míimos de manera automática. Sirve cuando se hace el primer acercamiento y de esta manera ejecutar el proceso B. Es un proceso mucho más lento y puede llegar a trabarse.
 Proceso B) Obtiene imagen e isotermas. Es hasta 15 veces más rápido.
*/

var lava = ['000004', '1B0C41', '4A0C6B', '781C6D', 'A52C60','CF4446','ED6925','FB9B06', 'F7D13D','FCFFA4'];
// var inferno = ['000003', '160b39', '410967','6a176e', '932567', 'ba3655', 'dc5039', 'f2751a', 'f6d542','fcfea4'];

Map.addLayer(geometry.draw('red'),{'opacity': 0.3},'Zona Metropolitana',0);
Map.centerObject(geometry,10);

// Sección A: Primera imagen o fecha exacta
// var image1 = LSTColl.first().clip(geometry);
// Sección B: Imagen media
var image1 = LSTColl.mean().clip(geometry);

// Proceso A: Máximos y mínimos
/*
var maxmin = iMaxMin.MinMax(image1, 'LST', geometry, 0, 100);
print(maxmin);
maxmin.evaluate(function (LSTmaxmin) {
  var BandCompViz = {
    bands:["LST"],
    min: LSTmaxmin.min,
    max: LSTmaxmin.max,
    palette: lava
  };
  Map.addLayer(image1, BandCompViz, 'LST');
});
```

```
89 */
90
91 // Proceso B: Imagen y Curvas de nivel
92 var contourlines = countour.countours(275, 335, 2, image1);
93 var viz = {bands:["LST"],
94             min: 296,
95             max: 327,
96             palette: lava};
97 Map.addLayer(image1, viz, "LST Ajustada"); //31 grados
98 Map.addLayer(image1.select('LST'),{min: 279, max:335, palette:lava},
     'LST 100%',0);  //56 grados
99 Map.addLayer(contourlines, {min: 275, max: 330, palette:['000003'],
     opacity:0.3}, "Isotermas");
```

Exportación Anual

```
1  /*
2    Este archivo es el principal para calcular los valores de LST, por
       medio del algoritmo de "Split-window" elaborado por Jiménez Muñ
       os et. al. En este script calcula por intervalos de tiempo, por
       el volumen de imágenes se recomienda máximo tres años.
3    Se divide en 4 grandes bloques:
4    1) Importación de los scripts.
5    2) Realización del cálculo del LST
6    3) Exportación
7    4) Representación visual
8  */
9
10 /*
11   BLOQUE I: Importación de los scripts que hacen el cálculo
12     -swLST: Algoritmo Principal
13     -mosaicDate: Unión de imágenes
14     -cloudcoverroi: Cálculo de nubosidad
15     -iMaxMin: Obtiene valores máximos y mínimos
16     -contour: Genera curvas de nivel
17 */
18 var swLST = require('users/edgarpzavaleta/Tesis:Modules/SWAlgorithm')
     ;
19 var mosaicDate = require('users/edgarpzavaleta/Tesis:Help/datemosaic'
     );
20 var cloudcoverroi = require('users/edgarpzavaleta/Tesis:Modules/
     CloudCoverROI');
21 var iMaxMin = require('users/edgarpzavaleta/Tesis:Help/minmax');
22 var countour = require('users/edgarpzavaleta/Tesis:Help/countours');
```

```
23
24 /*
25   BLOQUE II: Realización del cálculo del LST.
26   Esta parte requiere 4 datos de entrada.
27     -geometry: geometría que sirve para seleccionar la zona de
          estudio.
28     -satellite: versión de satélite de Landsat. Si bien es compatible
           desde Landsat 4, está pensado y optimizado para Landsat 8.
          Landsat 9 no está desarrollado
29     -startDate: fecha inicial
30     -endDate: fecha final.
31   En esta parte pueden surgir errores por asignar una fecha de manera
        errónea. Esto incluye poner fechas inexistentes (ejemplo: 31 de
        abril), o escoger la fecha de manera precisa. Si se conoce el d
       ía se debe de poner un día antes como inicial y un día después
       como final.
32   Posterior a este cálculo se filtran las imágenes por su "path" y se
        unen por fecha. Esto se hizo por las particularidades de la
       región. En caso de que no exista algún inconveniente, comentar
       la sección A y descomentar la sección B. Se agrega el porcentaje
        de nubosidad como metadatos.
33   Después se queda solamente con las imágenes LST. Se seleccionan
       aquellas con base en su porcentaje de nubosidad. Y se obtiene la
        imagen media
34 */
35 var geometry = ee.FeatureCollection("users/edgarpzavaleta/ZMVMWGS84")
     ;
36 var satellite = 'L8';
37 var startDate = '2023-01-01';
38 var endDate = '2023-12-31';
39
40 var landsatColl = swLST.collection(satellite, startDate, endDate,
     geometry);
41
42 //Sección A
43 landsatColl = landsatColl.filter(ee.Filter.eq('WRS_PATH', 26)); //
     Filtro por "path"
44 var mosaicColl = mosaicDate.collection(landsatColl); // Union de imá
     genes
45 mosaicColl = mosaicColl.map(cloudcoverroi.fclip(geometry)); //
     Recorte de zona
46
47 //Sección B
```

```
48 //var mosaicColl = landsatColl.map(cloudcoverroi.fclip(geometry)); //
       Recorte de zona
49
50 var LSTColl = mosaicColl.select('LST');
51 LSTColl = LSTColl.map(cloudcoverroi.percCover(geometry)); // Calculo
     de Nubosidad
52 LSTColl = LSTColl.filter(ee.Filter.gte('IMAGE_PERC_COVER', 80)); //
     Porcentaje de Imagen
53 var meanLST = LSTColl.mean();
54
55 /*
56 BLOQUE III: Exportación.
57   Aquí hay dos maneras:
58     1 Guardarlo como un Asset (colección en la nube, disponible para
         GEE)
59     2 Guardarlo en Google Drive (colección privada)
60 */
61
62 Export.image.toAsset({
63   image: meanLST,
64   description: 'myExportImageTask',
65   assetId: 'projects/ee-edgarpzavaletatesislst/assets',
66   region: geometry,
67   scale: 30
68 });
69
70 /*
71 Export.image.toDrive({
72   image: image,
73   description: 'image_export',
74   folder: 'ee_demos',
75   scale: 30,
76 });
77 */
78
79 // Si la exportación tiene más de 1e8 píxeles, establezca "maxPixels"
       a un valor más alto.
80 Export.image.toAsset({
81   image: meanLST,
82   description: 'myExportImageTask',
83   assetId: 'projects/ee-edgarpzavaletatesislst/LST2013',  // TODO:
       cambiar ubicación
84   region: geometry,
85   scale: 30
```

```
86 });
87
88 /*
89 BLOQUE IV: Representación Visual
90   En este bloque se puede ver las diferentes imágenes en secuencia
       como si fueran una tira de película
91 */
92
93 /*
94 // Definir argumentos de visualización.
95 var visArgs = {
96   min: 282,
97   max: 322,
98   palette: ['000003', '160b39', '410967','6a176e', '932567', 'ba3655
       ', 'dc5039', 'f2751a','f6d542','fcfea4']
99 };
100
101 // Definir una función para convertir una imagen en una imagen de
    visualización RGB y copiarla de la imagen original a la imagen RGB
    .
102 var visFun = function(img) {
103   return img.visualize(visArgs).copyProperties(img, img.propertyNames
       ());
104 };
105
106 // Asignar sobre la colección de imágenes para convertir cada imagen
    en una visualización RGB utilizando la función de visualización
    definida previamente.
107 var ndviColVis = LSTColl.map(visFun);
108
109 /*
110 // Definir argumentos para los parámetros de la función de animación.
111 var filmArgs  = {
112   dimensions: 128,
113   region: geometry,
114   crs: 'EPSG:3857',
115   min: 282,
116   max: 322,
117   palette: ['000003', '160b39', '410967','6a176e', '932567', 'ba3655
       ', 'dc5039', 'f2751a','f6d542','fcfea4']
118 };
119
120 // Imprimir una URL que producirá la tira de película cuando se
    acceda.
```

```
print(LSTColl.getFilmstripThumbURL(filmArgs));
*/
```

LST Media Decenal

```
/*
  Este script fue utilizado para calcular el valor medio del LST
     decenal, derivado de las datos anuales que se obtuvieron por
     medio del algoritmo "Split-window" elaborado por Jiménez Muños
     et. al. En el script Imagen_Individual
*/

// Se importa la geometría de la ZMVM
var geometry = ee.FeatureCollection("users/edgarpzavaleta/ZMVMWGS84")
    ;

// Se importan las imágenes ya promediadas
var image2013 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2013");
var image2014 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2014");
var image2015 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2015");
var image2016 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2016");
var image2017 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2017");
var image2018 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2018");
var image2019 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2019");
var image2020 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2020");
var image2021 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2021");
var image2022 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2022");
var image2023 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
    LST_2023");

// Se convierten la imágenes en una collección
var col = ee.ImageCollection.fromImages(
  [image2013, image2014, image2015, image2016, image2017, image2018,
     image2019, image2020,
```

```
  image2021, image2022, image2023]
);

// Se obtiene la media
var meanLST = col.mean();

// Paleta de colores para la imagen
var inferno = ['000003', '160b39', '410967','6a176e', '932567', '
    ba3655', 'dc5039', 'f2751a','f6d542','fcfea4'];

// Configuración de visualización para la imagen
var viz = {bands:["LST"],
            min: 282,
            max: 305,
            palette: inferno};

Map.addLayer(meanLST, viz, "LST");
Map.addLayer(geometry.draw('red'),{},'Zona Metropolitana',0);
Map.centerObject(geometry,10);

//Obtención de isoterma decenal
var countour = require('users/edgarpzavaleta/Tesis:Help/countours');
var contourlines = countour.countours(280, 310, 2, meanLST);
Map.addLayer(contourlines, {min: 282, max: 305, palette:['000003'],
    opacity:0.3}, "Isoterma Decenal");

// Exportación para Drive
Export.image.toDrive({
  image: meanLST,
  description: 'image_export_maxpixels',
  folder: 'Tesis',
  region: geometry,
  crs: 'EPSG:4326',
  scale: 30,
  maxPixels: 1e13
});
```

HotSpots

```
/*
  Este script fue utilizado para identificar y marcar las áreas con
     temperaturas significativamente más altas en comparación con su
     entorno.
*/
```

```
4
5  // Importación de archivos previamente calculados
6  var image = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
       LST_2013"),
7      image2 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
           LST_2014"),
8      image3 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
           LST_2015"),
9      image4 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
           LST_2016"),
10     image5 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
           LST_2017"),
11     image6 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
           LST_2018"),
12     image7 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
           LST_2020"),
13     image8 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
           LST_2021"),
14     image9 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
           LST_2022"),
15     image10 = ee.Image("projects/ee-edgarpzavaletatesislst/assets/
           LST_2023");
16
17 // Creación de una colección (agrupación de imágenes) de los archivos
       previamente calculados
18 var temperatureCollection = ee.ImageCollection([image, image2, image3
       , image4, image5, image6, image7, image8, image9, image10]);
19
20 // Calculo de la media y desviación estándar
21 var meanImage = temperatureCollection.mean();
22 var stdDevImage = temperatureCollection.reduce(ee.Reducer.stdDev());
23
24 // Calculo de los valores z
25 /*
26 Nota:
27 Los valores z miden cuántas desviaciones estándar es el valor de
       temperatura de un píxel de la temperatura media para su ubicación
       a lo largo del tiempo.
28 Está dada por: z = (x - mu) / sigma, donde x es el valor de
       temperatura, mu es la temperatura media y sigma es la desviación
       estándar de temperaturas para esa ubicación a lo largo del tiempo.
29 */
30 var zScoreImage = temperatureCollection.map(function(image) {
31   return image.subtract(meanImage).divide(stdDevImage);
```

```
});

// Valor de umbral z para la detección
/*
Nota:
El umbral es una técnica de segmentación para separar un objeto considerado como primer plano de su fondo.
Un valor de umbral sirve para determinar a partir de cuanto (por ejemplo, z > 2) se considera un punto caliente. Una puntuación z mayor que el umbral indica que la temperatura en ese píxel es significativamente más alta que el promedio.
*/
var threshold = 1;

// Creación de una máscara binaria
/*
Nota:
Crear una máscara binaria sirve para distinguir las áreas con temperaturas significativamente más altas con un valor de 1 y las áreas con temperaturas promedio o por debajo como 0. Esto en función del valor asignado en la parte anterior
*/
var hotspotMask = zScoreImage.map(function(zScore) {
  return zScore.gt(threshold);
});

// Combinación de todas las ímagenes en una colección para obtener e resultado final
var finalHotspotMask = hotspotMask.reduce(ee.Reducer.anyNonZero());

// Obtención de una imagen en la colección, para usarlo como punto de partida para el enmascaramiento
var sampleImage = image8;

// Aplicación de la máscara binaria sobre la imagen en la colección
var maskedTemperature = sampleImage.updateMask(finalHotspotMask);

// Parámetros de visualización para mostrar la imagen
var visualizationParams = {
  min: 290, // Valor mínimo a representar
  max: 300, // Valor máximo a representar
   palette: ['1d4877', '1b8a5a', 'fbb021','f68838']
};

```

```
var showHotspotMask = finalHotspotMask.updateMask(finalHotspotMask.
    neq(1));
Map.addLayer(showHotspotMask,{palette: ['ee3e32']}, 'Hotspots');
Map.addLayer(maskedTemperature, visualizationParams, 'Temperatura
    Enmascarada');
```

Scripts Modulares

Porcentaje de Nubosidad en la Zona

```
/*
Esta función devuelve el porcentaje de cobertura de nubosidad que
    tiene una imagen, la imagen tiene que estar enmascarada.

ENTRADAS:
    - ROI: una geometría, la región de interés
    - scale: la escala nominal en metros de la proyección a trabajar.

SALIDAS:
    - <ee.Image>: la imagen con el porcentaje de cobertura por
        nubosidad
*/

exports.fclip = function(roi) {
  var wrap = function(image){
    return image.clip(roi);
  };
  return wrap;
};

exports.percCover = function(roi) {
  var wrap = function(image){
    // Cálculo del número de píxeles totales
    var totPixels = ee.Number(ee.Image(1).reduceRegion({
      reducer: ee.Reducer.count(),
      geometry: roi,
      scale: 10,
      maxPixels: 999999999,
    }).values().get(0));

    // Cálculo la cantidad real de píxeles dentro de la región de
        interés
    var actPixels = ee.Number(image.reduceRegion({
```

```
      reducer: ee.Reducer.count(),
      scale: 10,
      geometry: roi,
      maxPixels: 999999999,
    }).values().get(0));

    // Cálculo el porcentaje de cobertura
    var percCover = actPixels.divide(totPixels).multiply(100).round()
        ;
    // Regresa el porcentaje como un número de salida
    return image.set('IMAGE_PERC_COVER', percCover);
  };
  return wrap;
};
```

Corrección de Cobertura ASTER

```
/*
Esta función aplica una corrección de la cobertura vegetal a la
    emisividad ASTER para obtener un componente de emisividad del
    suelo desnudo en cada píxel.

ENTRADAS:
  -imagen: <ee.Imagen> Se requiere una imagen para recortar los datos
      ASTER a la geometría de la imagen; usar la imagen ASTER
     completa compromete el rendimiento
SALIDAS:
  - <ee.Imagen> Emisividad del suelo desnudo de la banda deseada
*/

// Obtiene la emisividad ASTER AG100: Conjunto de datos de emisividad
    global ASTER V003 de 100 metros
var aster = ee.Image("NASA/ASTER_GED/AG100_003");

// Obtiene el valor ASTER FVC (fracción de cobertura vegetal) del
    NDVI
var aster_ndvi = aster.select('ndvi').multiply(0.01);

var aster_fvc = aster_ndvi.expression('((ndvi-ndvi_bg)/(ndvi_vg -
    ndvi_bg))**2',
  {'ndvi':aster_ndvi,'ndvi_bg':0.2,'ndvi_vg':0.86});
aster_fvc = aster_fvc.where(aster_fvc.lt(0.0),0.0);
aster_fvc = aster_fvc.where(aster_fvc.gt(1.0),1.0);

```

```
// Exporta la emisividad del terreno desnudo para cada banda
exports.emiss_bare_band10 = function(image){
  return image.expression('(EM - 0.99*fvc)/(1.0-fvc)',{
    'EM':aster.select('emissivity_band10').multiply(0.001),
    'fvc':aster_fvc})
    .clip(image.geometry());
};

exports.emiss_bare_band11 = function(image){
  return image.expression('(EM - 0.99*fvc)/(1.0-fvc)',{
    'EM':aster.select('emissivity_band11').multiply(0.001),
    'fvc':aster_fvc})
    .clip(image.geometry());
};

exports.emiss_bare_band12 = function(image){
  return image.expression('(EM - 0.99*fvc)/(1.0-fvc)',{
    'EM':aster.select('emissivity_band12').multiply(0.001),
    'fvc':aster_fvc})
    .clip(image.geometry());
};

exports.emiss_bare_band13 = function(image){
  return image.expression('(EM - 0.99*fvc)/(1.0-fvc)',{
    'EM':aster.select('emissivity_band13').multiply(0.001),
    'fvc':aster_fvc})
    .clip(image.geometry());
};

exports.emiss_bare_band14 = function(image){
  return image.expression('(EM - 0.99*fvc)/(1.0-fvc)',{
    'EM':aster.select('emissivity_band14').multiply(0.001),
    'fvc':aster_fvc})
    .clip(image.geometry());
};
```

Cálculo del FVC

```
/*
Esta función calcula la fracción de cobertura vegetal a partir del
   NDVI. La función NDVI debe ser llamada antes de esta.

ENTRADAS:
  - <ee.Image>: la imagen procesada por el NDVI
```

```
6
7  SALIDAS:
8    - <ee.Image>: la imagen obtenida del FVC
9  */
10
11 exports.addBand = function(landsat){
12   var wrap = function(image){
13
14     var ndvi = image.select('NDVI');
15
16     // Cálculo del FVC
17     var fvc = image.expression('((ndvi-ndvi_bg)/(ndvi_vg - ndvi_bg))
           **2',
18       {'ndvi':ndvi,'ndvi_bg':0.2,'ndvi_vg':0.86});
19     fvc = fvc.where(fvc.lt(0.0),0.0);
20     fvc = fvc.where(fvc.gt(1.0),1.0);
21
22     return image.addBands(fvc.rename('FVC'));
23   };
24   return wrap;
25 };
```

Cálculo del LST

```
1  /*
2  Se puede llamar a esta función para calcular y aplicar dos algoritmos
       de ventana: unica (Mono o mw) y dividida (Split o sw)
3
4  ENTRADAS:
5    -landsat: <string> la identificación del satélite Landsat. Entradas
         válidas: 'L4', 'L5', 'L7' y 'L8'
6    - image: <ee.Image> imagen para la cual calcular el LST
7  SALIDAS:
8    - image: <ee.Image> 'LST': land surface temperatura
9  */
10
11
12
13 // Function to create a lookup between two columns in the feature
       collection
14 var get_lookup_table = function(fc, prop_1, prop_2) {
15   var reducer = ee.Reducer.toList().repeat(2);
16   var lookup = fc.reduceColumns(reducer, [prop_1, prop_2]);
17   return ee.List(lookup.get('list'));
```

```
18 };
19
20 // Algoritmo de ventana unica (Mono o MW)
21 exports.mw = function(landsat){
22     // Coeficientes para el algoritmo Mono-Window
23     var coefficients = require('users/sofiaermida/landsat_smw_lst:
       modules/SMW_coefficients.js');
24
25     // Función para crear una búsqueda entre dos columnas en la
       colección de características
26     var get_lookup_table = function(fc, prop_1, prop_2) {
27       var reducer = ee.Reducer.toList().repeat(2);
28       var lookup = fc.reduceColumns(reducer, [prop_1, prop_2]);
29       return ee.List(lookup.get('list'));
30     };
31
32     var wrap = function(image){
33
34     // Selecciona los coeficientes del algoritmo
35     var coeff_SMW = ee.FeatureCollection(ee.Algorithms.If(landsat==='
       L4',coefficients.coeff_SMW_L4,
36                                         ee.Algorithms.If(landsat==='
                                              L5',coefficients.
                                              coeff_SMW_L5,
37                                         ee.Algorithms.If(landsat==='
                                              L7',coefficients.
                                              coeff_SMW_L7,
38                                         coefficients.coeff_SMW_L8))))
                                              ;
39
40     // Crea búsquedas para los coeficientes del algoritmo.
41     var A_lookup = get_lookup_table(coeff_SMW, 'TPWpos', 'A');
42     var B_lookup = get_lookup_table(coeff_SMW, 'TPWpos', 'B');
43     var C_lookup = get_lookup_table(coeff_SMW, 'TPWpos', 'C');
44
45     // Asigna coeficientes a la imagen utilizando la posición del
       contenedor TPW (Total Precipitable Water)
46     var A_img = image.remap(A_lookup.get(0), A_lookup.get(1),0.0,'
       TPWpos').resample('bilinear');
47     var B_img = image.remap(B_lookup.get(0), B_lookup.get(1),0.0,'
       TPWpos').resample('bilinear');
48     var C_img = image.remap(C_lookup.get(0), C_lookup.get(1),0.0,'
       TPWpos').resample('bilinear');
49
```

```
50      // Selecciona la banda TIR
51      var tir = ee.String(ee.Algorithms.If(landsat==='L8','B10',
52                          ee.Algorithms.If(landsat==='L7','B6_VCID_1',
53                          'B6')));
54      // Calcula el LST
55      var lst = image.expression(
56        'A*Tb1/em1 + B/em1 + C',
57           {'A': A_img,
58            'B': B_img,
59            'C': C_img,
60            'em1': image.select('EM'),
61            'Tb1': image.select(tir)
62           }).updateMask(image.select('TPW').lt(0).not());
63
64
65      return image.addBands(lst.rename('LST'));
66    };
67    return wrap;
68  };
69
70  // Algoritmo de ventana dividida (Split o SW)
71  exports.sw = function(landsat){
72      var wrap = function(image){
73      // Expresión para calcular el LST
74      var lst = image.expression(
75        'Ti + c1*(Ti - Tj) + c2*(Ti-Tj)** + c0 + (c3+c4*w)*(1-e) + (c5
             + (c6*w))*dE',{
76        'c0': -0.268,
77        'c1': 1.378,
78        'c2': 0.183,
79        'c3': 54.30,
80        'c4': -2.238,
81        'c5': -129.20,
82        'c6': 16.4,
83        'w': image.select('TPW'),
84        'r': 0.984,
85        'Ti': image.select('B10'),
86        'Tj': image.select('B11'),
87        'e':  (image.select('e_i').add(image.select('e_j'))).multiply
             (0.5),
88        'dE': (image.select('e_i').subtract(image.select('e_j')))
89      }).updateMask(image.select('TPW').lt(0).not());
90      return image.addBands(lst.rename('LST'));
91    };
```

```
92   return wrap;
93 };
```

Colección del LST Mono Window

```
1  /*
2  Esta función permite seleccionar datos de imágenes Landsat, para
       proceder a aplicar máscaras de nubes, calcular índices (NDVI y FVC
       ), y realizar cálculos relacionados con la temperatura de
       superficie terrestre (LST).
3  Selecciona los datos de Landsat en función de las entradas del
       usuario y realiza el cálculo LST.
4
5  USA:
6    - NCEP_TPW.js
7    - cloudmask.js
8    - compute_NDVI.js
9    - compute_FVC.js
10   - compute_emissivity.js
11   - SMWalgorithm.js
12
13 ENTRADAS:
14   - landsat: <string> Identificación del satélite Landsat: 'L4', 'L5
         ', 'L7' y 'L8'
15   - date_start: <string> Fecha de inicio de la colección Landsat
         siguiendo el formato: AAAA-MM-DD
16   - date_end: <string> Fecha final de la colección Landsat siguiendo
         el formato: AAAA-MM-DD
17   - geometry: <ee.Geometry> Región de Interés
18   - use_ndvi: <boolean> Evaulua si se debe considerar el NDVI.  Si es
          verdadero, los valores NDVI se utilizan para obtener una
         emisividad dinámica; Si es falso, la emisividad es obtenido
         directamente de ASTER
19
20 SALIDAS:
21   - <ee.ImageCollection> Una colección de imágenes con las bandas:
22     - Bandas originales LANDSAT: todas de SR excepto las bandas TIRS
           (de TOA)
23     - cloud masked: Nubes enmascaradas
24     - 'NDVI': Índice de vegetación normalizado
25     - 'FVC': Fracción de cobertura vegetal [0-1]
26     - 'TPW': Agua precipitable total [mm]
27     - 'EM': Emisividad superficial para la banda TIR
28     - 'LST': Temperatura de la superficie terrestre
```

```
29 */
30
31 // DECLARACIÓN DE MÓDULOS
     ------------------------------------------------------------
32 // Nubes enmascaradas
33 var cloudmask = require('users/edgarpzavaleta/Tesis:Modules/cloudmask
     ');
34 // Índice de vegetación normalizado
35 var ndvi = require('users/edgarpzavaleta/Tesis:Modules/NDVI');
36 // Fracción de cobertura vegetal
37 var fvc = require('users/edgarpzavaleta/Tesis:Modules/
     FractionofVegetationCover');
38 // Agua precipitable total
39 var tpw = require('users/edgarpzavaleta/Tesis:Modules/
     TotalPrecipitableWater');
40 // Emisividad superficial
41 var se = require('users/edgarpzavaleta/Tesis:Modules/
     SurfaceEmissivity');
42 // Temperatura de la superficie terrestre
43 var lst = require('users/edgarpzavaleta/Tesis:Modules/
     LandSurfaceTemperature');
44 //
     ------------------------------------------------------------------------------------------

45
46 var landsatDictionary = ee.Dictionary({
47   'L4': {
48     'TOA': ee.ImageCollection('LANDSAT/LT04/C01/T1_TOA'),
49     'SR': ee.ImageCollection('LANDSAT/LT04/C01/T1_SR'),
50     'TIR': ['B6',]
51   },
52   'L5': {
53     'TOA': ee.ImageCollection('LANDSAT/LT05/C01/T1_TOA'),
54     'SR': ee.ImageCollection('LANDSAT/LT05/C01/T1_SR'),
55     'TIR': ['B6',]
56   },
57   'L7': {
58     'TOA': ee.ImageCollection('LANDSAT/LE07/C01/T1_TOA'),
59     'SR': ee.ImageCollection('LANDSAT/LE07/C01/T1_SR'),
60     'TIR': ['B6_VCID_1','B6_VCID_2'],
61   },
62   'L8': {
63     'TOA': ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA'),
64     'SR': ee.ImageCollection('LANDSAT/LC08/C02/T1_L2'),
```

```
    'TIR': ['B10','B11'],
    'VISW': ['SR_B2','SR_B3','SR_B4','SR_B5','QA_PIXEL']
  }
});

//
    ------------------------------------------------------------------------------

exports.collection = function(landsat, start, end, geometry, use_ndvi
    ){
  // Carga de TOA Radiancia/Reflectancia
  var collectionDict = ee.Dictionary(landsatDictionary.get(landsat));

  var landsatTOA =  ee.ImageCollection(collectionDict.get('TOA')).
      filter(ee.Filter.date(start, end)).filterBounds(geometry);

  var landsatSR = ee.ImageCollection(collectionDict.get('SR')).filter
      (ee.Filter.date(start, end)).filterBounds(geometry).map(
      cloudmask.sr).map(ndvi.addBand(landsat)).map(fvc.addBand(landsat
      )).map(tpw.mw).map(se.mw(landsat,use_ndvi));

  // Combina colecciones de todos los canales de la colección de
      reflectancia de superficie
  // Excepto canales tir: de la colección TOA seleccione bandas TIR
  var tir = ee.List(collectionDict.get('TIR'));
  var visw = ee.List(collectionDict.get('VISW'))
    .add('NDVI')
    .add('FVC')
    .add('TPW')
    .add('TPWpos')
    .add('EM');

  var landsatALL = (landsatSR.select(visw).combine(landsatTOA.select(
      tir), true));

  // Cálculo del LST
  var landsatLST = landsatALL.map(lst.mw(landsat));
  return(landsatLST);
};
```

Cálculo del NDVI

```
/*
```

```
2  Se puede llamar a esta función para calcular y aplicar dos algoritmos
       de ventana: unica (Mono o mw) y dividida (Split o sw)
3
4  ENTRADAS:
5    -landsat: <string> la identificación del satélite Landsat. Entradas
         válidas: 'L4', 'L5', 'L7' y 'L8'
6    - image: <ee.Image> imagen para la cual calcular el LST
7  SALIDAS:
8    - image: <ee.Image> 'LST': land surface temperatura
9  */
10
11
12
13 // Function to create a lookup between two columns in the feature
       collection
14 var get_lookup_table = function(fc, prop_1, prop_2) {
15   var reducer = ee.Reducer.toList().repeat(2);
16   var lookup = fc.reduceColumns(reducer, [prop_1, prop_2]);
17   return ee.List(lookup.get('list'));
18 };
19
20 // Algoritmo de ventana unica (Mono o MW)
21 exports.mw = function(landsat){
22     // Coeficientes para el algoritmo Mono-Window
23     var coefficients = require('users/sofiaermida/landsat_smw_lst:
           modules/SMW_coefficients.js');
24
25     // Función para crear una búsqueda entre dos columnas en la
           colección de características
26     var get_lookup_table = function(fc, prop_1, prop_2) {
27       var reducer = ee.Reducer.toList().repeat(2);
28       var lookup = fc.reduceColumns(reducer, [prop_1, prop_2]);
29       return ee.List(lookup.get('list'));
30     };
31
32     var wrap = function(image){
33
34     // Selecciona los coeficientes del algoritmo
35     var coeff_SMW = ee.FeatureCollection(ee.Algorithms.If(landsat==='
           L4',coefficients.coeff_SMW_L4,
36                                         ee.Algorithms.If(landsat==='
                                               L5',coefficients.
                                               coeff_SMW_L5,
```

```
37                                          ee.Algorithms.If(landsat==='
                                               L7',coefficients.
                                               coeff_SMW_L7,
38                                          coefficients.coeff_SMW_L8))))
                                               ;
39
40      // Crea búsquedas para los coeficientes del algoritmo.
41      var A_lookup = get_lookup_table(coeff_SMW, 'TPWpos', 'A');
42      var B_lookup = get_lookup_table(coeff_SMW, 'TPWpos', 'B');
43      var C_lookup = get_lookup_table(coeff_SMW, 'TPWpos', 'C');
44
45      // Asigna coeficientes a la imagen utilizando la posición del
            contenedor TPW (Total Precipitable Water)
46      var A_img = image.remap(A_lookup.get(0), A_lookup.get(1),0.0,'
            TPWpos').resample('bilinear');
47      var B_img = image.remap(B_lookup.get(0), B_lookup.get(1),0.0,'
            TPWpos').resample('bilinear');
48      var C_img = image.remap(C_lookup.get(0), C_lookup.get(1),0.0,'
            TPWpos').resample('bilinear');
49
50      // Selecciona la banda TIR
51      var tir = ee.String(ee.Algorithms.If(landsat==='L8','B10',
52                          ee.Algorithms.If(landsat==='L7','B6_VCID_1',
53                          'B6')));
54      // Calcula el LST
55      var lst = image.expression(
56        'A*Tb1/em1 + B/em1 + C',
57           {'A': A_img,
58            'B': B_img,
59            'C': C_img,
60            'em1': image.select('EM'),
61            'Tb1': image.select(tir)
62           }).updateMask(image.select('TPW').lt(0).not());
63
64
65      return image.addBands(lst.rename('LST'));
66    };
67    return wrap;
68  };
69
70  // Algoritmo de ventana dividida (Split o SW)
71  exports.sw = function(landsat){
72      var wrap = function(image){
73      // Expresión para calcular el LST
```

```
74      var lst = image.expression(
75        'Ti + c1*(Ti - Tj) + c2*(Ti-Tj)** + c0 + (c3+c4*w)*(1-e) + (c5
             + (c6*w))*dE',{
76        'c0': -0.268,
77        'c1': 1.378,
78        'c2': 0.183,
79        'c3': 54.30,
80        'c4': -2.238,
81        'c5': -129.20,
82        'c6': 16.4,
83        'w': image.select('TPW'),
84        'r': 0.984,
85        'Ti': image.select('B10'),
86        'Tj': image.select('B11'),
87        'e':  (image.select('e_i').add(image.select('e_j'))).multiply
             (0.5),
88        'dE': (image.select('e_i').subtract(image.select('e_j')))
89      }).updateMask(image.select('TPW').lt(0).not());
90      return image.addBands(lst.rename('LST'));
91    };
92    return wrap;
93  };
```

Cálculo del LST Split Window

Cálculo de la Emisividad Superficial

```
1  /*
2  Esta función calcula la emisividad de la superficie para Landsat
       requiere valores de FVC: Compute_FVC
3
4  ref: Malakar, N.K., Hulley, G.C., Hook, S.J., Laraby, K., Cook, M.,
       Schott, J.R., 2018. An Operational Land Surface Temperature
       Product for Landsat Thermal Data: Methodology and Validation. IEEE
        Trans. Geosci. Remote Sens. 56, 5717  5735  .  https://doi.org
       /10.1109/TGRS.2018.2824828
5
6  USA:
7    - ASTER_bare_emiss.js
8
9  ENTRADAS:
10   - landsat: <string> Identificación del satélite Landsat: 'L4', 'L5
         ', 'L7' y 'L8'
```

```
11    - use_ndvi: <boolean> Evalua si se debe considerar el NDVI.  Si es
          verdadero, los valores NDVI se utilizan para obtener una
          emisividad dinámica; Si es falso, la emisividad es obtenido
          directamente de ASTER
12
13  SALIDAS:
14    - <ee.Image> la imagen de entrada con 1 nueva banda: 'EM'.
          Emisividad superficial de la banda TIR
15  */
16  // Función importada desde el algoritmo de Sofia Armida et al.
17  var asterCorrection = require('users/sofiaermida/landsat_smw_lst:
        modules/ASTER_bare_emiss.js');
18
19  // Esta función calcula la emisividad de la banda Landsat TIR para
        convolucionar las bandas 13 y 14 de ASTER, y obtener la emisividad
         de un solo canal
20  exports.mw = function(landsat, use_ndvi){
21    var wrap = function(image){
22
23      var c13 = ee.Number(ee.Algorithms.If(landsat==='L4',0.3222,
24                                ee.Algorithms.If(landsat==='L5',-0.0723,
25                                ee.Algorithms.If(landsat==='L7',0.2147,
26                                0.6820))));
27      var c14 = ee.Number(ee.Algorithms.If(landsat==='L4',0.6498,
28                                ee.Algorithms.If(landsat==='L5',1.0521,
29                                ee.Algorithms.If(landsat==='L7',0.7789,
30                                0.2578))));
31      var c = ee.Number(ee.Algorithms.If(landsat==='L4',0.0272,
32                                ee.Algorithms.If(landsat==='L5',0.0195,
33                                ee.Algorithms.If(landsat==='L7',0.0059,
34                                0.0584))));
35
36      // Obtiene la emisividad ASTER y convoluciona a la banda Landsat
37      var emiss_bare = image.expression('c13*EM13 + c14*EM14 + c',{
38        'EM13':asterCorrection.emiss_bare_band13(image),
39        'EM14':asterCorrection.emiss_bare_band14(image),
40        'c13':ee.Image(c13),
41        'c14':ee.Image(c14),
42        'c':ee.Image(c)
43        });
44
45      // Calcula la emisividad dinámica para Landsat
46      var EMd = image.expression('fvc*0.99+(1-fvc)*em_bare',
47        {'fvc':image.select('FVC'),'em_bare':emiss_bare});
```

```
48
49      // Calcula la emisividad directamente desde ASTER sin corrección
            de vegetación
50      var aster = ee.Image("NASA/ASTER_GED/AG100_003")
51        .clip(image.geometry());
52      var EM0 = image.expression('c13*EM13 + c14*EM14 + c',{
53        'EM13':aster.select('emissivity_band13').multiply(0.001),
54        'EM14':aster.select('emissivity_band14').multiply(0.001),
55        'c13':ee.Image(c13),
56        'c14':ee.Image(c14),
57        'c':ee.Image(c)
58        });
59
60      // Selecciona qué emisividad generar según la selección del
            usuario
61      var em = ee.Image(ee.Algorithms.If(use_ndvi,EMd,EM0));
62
63      return image.addBands(em.rename('EM'));
64    };
65    return wrap;
66  };
67
68  // Esta función calcula la emisividad de los datos ASTER con banda 10
        y banda 11
69  exports.sw = function(landsat){
70    var wrap = function(image){
71      var fvc = image.select('FVC');
72      var em13 = asterCorrection.emiss_bare_band13(image);
73      var em14 = asterCorrection.emiss_bare_band14(image);
74
75      // Calcula la emisividad dinámica para Landsat em13
76      var eI = image.expression('fvc*0.99+(1-fvc)*em13',{'fvc':fvc,'
            em13':em13}).rename('e_i');
77
78      // Calcula la emisividad dinámica para Landsat em14
79      var eJ = image.expression('fvc*0.99+(1-fvc)*em14',{'fvc':fvc,'
            em14':em14}).rename('e_j');
80
81      return image.addBands([eI, eJ]);
82    };
83    return wrap;
84  };
```

Cálculo del Agua Precipitable Total

```
/*
Esta función compara los datos de vapor de agua atmosférico del reanálisis del NCEP con cada imagen Landsat.
Los valores TPW se interpolan desde los tiempos del modelo de 6 horas hasta el tiempo de la imagen.
Hay dos versiones, para cada algoritmo de cálculo por ventana (dividida y única)


ENTRADAS:
  - image: <ee.Image> Imagen para la cual interporal el valor TPW, necesita la propiedad de imagen 'system: time_start'
SALIDAS:
  - <ee.Image> la imagen de entrada con 2 nuevas bandas:
    -'TPW': Valores totales del agua precipitable
    -'TPWpos': Índice para el LUT de los coeficientes del algoritmo SMW
*/

// Ventana Única
exports.mw = function(image){

  // Primero selecciona el día de interés
  var date = ee.Date(image.get('system:time_start'));
  var year = ee.Number.parse(date.format('yyyy'));
  var month = ee.Number.parse(date.format('MM'));
  var day = ee.Number.parse(date.format('dd'));
  var date1 = ee.Date.fromYMD(year,month,day);
  var date2 = date1.advance(1,'days');

  // Calcula la diferencia horaria con respecto a las imágenes Landsat.
  var datedist = function(image){
    return image.set('DateDist',
      ee.Number(image.get('system:time_start'))
      .subtract(date.millis()).abs());
  };

  // Carga la recopilación de datos atmosféricos.
  var TPWcollection = ee.ImageCollection('NCEP_RE/surface_wv').filter(ee.Filter.date(date1.format('yyyy-MM-dd'), date2.format('yyyy-MM-dd'))).map(datedist);
```

```
35
36    // Selecciona los dos tiempos de modelo más cercanos
37    var closest = (TPWcollection.sort('DateDist')).toList(2);
38
39    // Evalua si hay datos atmosféricos en el día deseado. Si no, crea
          una imagen TPW con valores no realistas estos luego se
          enmascaran en la función SMWalgorithm (evita errores)
40    var tpw1 = ee.Image(ee.Algorithms.If(closest.size().eq(0), ee.Image
          .constant(-999.0), ee.Image(closest.get(0)).select('pr_wtr') ));
41    var tpw2 = ee.Image(ee.Algorithms.If(closest.size().eq(0), ee.Image
          .constant(-999.0), ee.Algorithms.If(closest.size().eq(1), tpw1,
          ee.Image(closest.get(1)).select('pr_wtr') )));
42
43    var time1 = ee.Number(ee.Algorithms.If(closest.size().eq(0), 1.0,
          ee.Number(tpw1.get('DateDist')).divide(ee.Number(21600000)) ));
44    var time2 = ee.Number(ee.Algorithms.If(closest.size().lt(2), 0.0,
          ee.Number(tpw2.get('DateDist')).divide(ee.Number(21600000)) ));
45
46    var tpw = tpw1.expression('tpw1*time2+tpw2*time1',{
47      'tpw1':tpw1,'time1':time1, 'tpw2':tpw2,'time2':time2}).clip(image
            .geometry());
48
49    // Los coeficientes SMW están agrupados por valores TPW. Encuentra
          el contenedor de cada valor TPW
50    var pos = tpw.expression(
51      "value = (TPW>0 && TPW<=6) ? 0" +
52      ": (TPW>6 && TPW<=12) ? 1" +
53      ": (TPW>12 && TPW<=18) ? 2" +
54      ": (TPW>18 && TPW<=24) ? 3" +
55      ": (TPW>24 && TPW<=30) ? 4" +
56      ": (TPW>30 && TPW<=36) ? 5" +
57      ": (TPW>36 && TPW<=42) ? 6" +
58      ": (TPW>42 && TPW<=48) ? 7" +
59      ": (TPW>48 && TPW<=54) ? 8" +
60      ": (TPW>54) ? 9" +
61      ": 0",{'TPW': tpw})
62      .clip(image.geometry());
63
64    // Agrega el TPW a la imagen como banda
65    var withTPW = (image.addBands(tpw.rename('TPW'),['TPW'])).addBands(
          pos.rename('TPWpos'),['TPWpos']);
66
67    return withTPW;
68  };
```

```
69
70 // Ventana Dividida
71 exports.sw = function(image){
72
73   // Primero selecciona el día de interés
74   var date = ee.Date(image.get('system:time_start'));
75   var year = ee.Number.parse(date.format('yyyy'));
76   var month = ee.Number.parse(date.format('MM'));
77   var day = ee.Number.parse(date.format('dd'));
78   var date1 = ee.Date.fromYMD(year,month,day);
79   var date2 = date1.advance(1,'days');
80
81   // Calcula la diferencia horaria con respecto a las imágenes
       Landsat.
82   var datedist = function(image){
83     return image.set('DateDist',
84       ee.Number(image.get('system:time_start'))
85       .subtract(date.millis()).abs());
86   };
87
88   // Carga la recopilación de datos atmosféricos
89   var TPWcollection = ee.ImageCollection('NCEP_RE/surface_wv')
90                       .filter(ee.Filter.date(date1.format('yyyy-MM-dd
                            '), date2.format('yyyy-MM-dd')))
91                       .map(datedist);
92
93   // Selecciona los dos tiempos de modelo más cercanos
94   var closest = (TPWcollection.sort('DateDist')).toList(2);
95
96   // Evalua si hay datos atmosféricos en el día deseado. Si no, crea
       una imagen TPW con valores no realistas estos luego se
       enmascaran en la función SMWalgorithm (evita errores)
97   var tpw1 = ee.Image(ee.Algorithms.If(closest.size().eq(0), ee.Image
       .constant(-999.0),ee.Image(closest.get(0)).select('pr_wtr') ));
98   var tpw2 = ee.Image(ee.Algorithms.If(closest.size().eq(0), ee.Image
       .constant(-999.0), ee.Algorithms.If(closest.size().eq(1), tpw1,
       ee.Image(closest.get(1)).select('pr_wtr') )));
99
100  var time1 = ee.Number(ee.Algorithms.If(closest.size().eq(0), 1.0,
       ee.Number(tpw1.get('DateDist')).divide(ee.Number(21600000)) ));
101  var time2 = ee.Number(ee.Algorithms.If(closest.size().lt(2), 0.0,
       ee.Number(tpw2.get('DateDist')).divide(ee.Number(21600000)) ));
102
103  var tpw = tpw1.expression('tpw1*time2+tpw2*time1',{
```

```
104       'tpw1':tpw1, 'time1':time1, 'tpw2':tpw2, 'time2':time2}).clip(
              image.geometry());
105
106     // Agrega el TPW a la imagen como banda
107     return image.addBands(tpw.resample('bicubic').rename('TPW') );
108   };
```

Enmascaramiento de nubes

```
1   /*
2   Esta función enmascara las nubes y la sombra de las nubes utilizando
        la banda Calidad.
3   ENTRADAS:
4     -image: <ee.Image> Imagen para la cual las nubes están enmascaradas
5   SALIDAS:
6     -image: <ee.Image> Imagen con máscara actualizada
7   */
8
9   // Máscara de nubes para datos TOA
10  exports.toa = function(image) {
11    var qa = image.select('QA_PIXEL');
12    var mask = qa.bitwiseAnd(1 << 3).eq(0);
13    return image.updateMask(mask);
14  };
15
16  // Máscara de nubes para datos SR
17  exports.sr = function(image) {
18    var qa = image.select('QA_PIXEL');
19    var mask = qa.bitwiseAnd(1 << 3)
20       .or(qa.bitwiseAnd(1 << 4));
21    return image.updateMask(mask.not());
22  };
```

Scripts de Apoyo

Obtención de Isolíneas

```
1   /*
2   Esta función extrae isolíneas que toman referencia los valores
        proporcionados por el usuario.
3
4   ENTRADA:
5     -start: número. Valor de inicio
6     -end: número. Valor de finalización
```

```
  -jump: salto. Valor que define el intervalo estático para generar
     las isolíneas
  -input: Imagen. Imagen de la cual extraerá las isolíneas
*/

exports.countours = function(start, end, jump, input){
  start = ee.Algorithms.If( ee.Number(start).round().gt(ee.Number(
     start)),
      (ee.Number(start).subtract(1)).round(),
      ee.Number(start).round());
  end = ee.Number(end).round();
  var image = input.select('LST');
  var lines = ee.List.sequence(start, end, jump);
  var countourlines = lines.map(function(line){
    var mycontour = image.convolve(ee.Kernel.gaussian(5, 3))
    .subtract(ee.Image.constant(line)).zeroCrossing()
    .multiply(ee.Image.constant(line)).toFloat();
    return mycontour.mask(mycontour);
  });
  var contourlines = ee.ImageCollection(countourlines).mosaic();
  return(contourlines);
};
```

Mosaico por fecha

```
/*
Esta función crea mosaicos (unión de una o más imágenes) por fecha
ENTRADA:
  -imcol: Colección de imágenes.
SALIDA:
  -mosaic_imlist: Colección de imagénes, agrupadas por fecha ("YYYY-
     MM-dd").
*/
exports.collection = function (imcol) {
    var imlist = imcol.toList(imcol.size());
    var unique_dates = imlist.map(function(im){
      return ee.Image(im).date().format("YYYY-MM-dd");
    }).distinct();
    var mosaic_imlist = unique_dates.map(function(d){
      d = ee.Date(d);
      var im = imcol.filterDate(d, d.advance(1, "day")).mosaic();

      return im.set(
          "system:time_start", d.millis(),
```

```
            "system:id", d.format("YYYY-MM-dd"));
    });

    return ee.ImageCollection(mosaic_imlist);
};
```

Obtención de Mínimos y Máximos

```
/*
Esta función obtiene los valores mínimos y máximos de una imagen.
ENTRADA:
  -collection: colección de imágenes
  -band: nombre de la banda en la cual se requiere buscar datos min/
     max
  -geometry: geometría para el límite
  -a,b: valores opcionales para reducción a un percentil
SALIDA:
  -vmin: Valores mínimos
  -vmax: Valores máximos

¿Cómo utilizar esta función?
-Por solo obtener los valores. LLamar a la función con
  var iMaxMin = require('users/edgarpzavaleta/Tesis:Help/minmax');
  var min = ee.Number(ee.List(maxmin.vmin)); \\ Valor mínimo
  var max = ee.Number(ee.List(maxmin.vmax)); \\ Valor máximo
-Mapeo dinámico. LLamar a la función con
  var maxmin = iMaxMin.getMaxMin(collection, 'Band Name', geometry,
     lp, hp);
  var min = ee.Number(ee.List(data.vmin));
  var max = ee.Number(ee.List(data.vmax));

  var viz = {bands:["Band Name"], min: min.getInfo(), max: max.
     getInfo()};
  print(viz);
  Map.addLayer(nombre_de_la_imagen, viz, "Estirado a valores mínimos
     máximos");
*/

exports.getMaxMin = function(collection, band, geometry, a, b){
  var data = collection.select(band).reduceRegion({
    reducer: ee.Reducer.percentile([a, b]),
    geometry: geometry,
    scale: 30,
    bestEffort: true
```

```
  });
  var min = data.get(''+band+'_p'+a).getInfo();
  // Borrar la siguiente línea si se busca el valor completo, en este
      caso se redondea al valor más bajo
  min = ee.Algorithms.If( ee.Number(min).round().gt(ee.Number(min)),
      (ee.Number(min).subtract(1)).round(),
      ee.Number(min).round());

  var max = data.get(''+band+'_p'+b).getInfo();
  // Borrar la siguiente línea si se busca el valor completo
  max = ee.Number(max).round();
  return {vmin: min, vmax: max};
};

exports.MinMax = function(collection, band, geometry, a, b){
  var MinMax = collection.select(band).reduceRegion({
    reducer: ee.Reducer.percentile([a, b]),
    geometry: geometry,
    scale: 30,
    bestEffort: true
  });

  var min = MinMax.get(''+band+'_p'+a);
  var max = MinMax.get(''+band+'_p'+b);

  var dictionary = ee.Dictionary({
    min: min,
    max: max
  });

  return (dictionary);
};
```

Cálculo de Porcentajes de Cobertura

```
// Cuenta los píxeles en la imagen aoi
var totPixels = ee.Number(ee.Image(1).reduceRegion({
    reducer: ee.Reducer.count(),
    geometry: geometry,
    scale: 10,
    maxPixels: 999999999,
  }).values().get(0));

```

```
9  // Cuenta los píxeles distintos de cero/nulos en la imagen dentro del
       aoi
10   var actPixels = ee.Number(image1.select('LST').reduceRegion({
11     reducer: ee.Reducer.count(),
12     scale: 10,
13     geometry: geometry,
14     maxPixels: 999999999,
15   }).values().get(0));
16
17 // Calcula el porcentaje de cobertura
18 var pcPix = actPixels.divide(totPixels).multiply(100);
19 print(pcPix);
20 print(totPixels);
```

1 Configuración Python

1.1 Configuración de GEE

1. Importar la biblioteca de Earth Engine.
2. Autenticar el acceso (verificación de registro y acceso a la cuenta de Google).
3. Inicializar la API

[ ]:
```
import ee
ee.Authenticate()
ee.Initialize()
```

1.2 Configuración de Python

1. Importar la librería *pandas*. Permite el análisis de datos (incluida la estructura de datos DataFrame)
2. Importar la librería *altair*. Para la visualización declarativa (utilizada para gráficos)
3. Importar la librería *numpy*. Para procesamiento de matrices (utilizado para la regresión lineal)
4. Importar la librería *folium*. Mapa web interactivo
5. Importar la librería *pprint*. Embellece las tablas

[ ]:
```
import pandas as pd
import altair as alt
import numpy as np
import folium
import pprint
```

2 Módulos para Cálculo de LST

2.1 Filtro de Nubes

Esta función enmascara las nubes y la sombra de las nubes usando la banda de Calidad

[ ]:
```
def cloudmask_toa (image):
  # Cloudmask for TOA data
  qa = image.select('QA_PIXEL')
  mask = qa.bitwiseAnd(1 << 3).eq(0)
  return image.updateMask(mask);

def cloudmask_sr (image):
  # Cloudmask for SR data
  qa = image.select('QA_PIXEL');
  mask = qa.bitwiseAnd(1 << 3) or (qa.bitwiseAnd(1 << 4))
  return image.updateMask(mask.Not())
```

2.2 Cálculo NDVI

Esta función calcula los valores NDVI

[ ]:
```
def ndvi (image):
  # Choose bands
  nir = image.select('SR_B5')
  red = image.select('SR_B4')
  # Compute NDVI
  return image.addBands(image.expression('(nir-red)/(nir+red)',{
      'nir': nir.multiply(0.0000275).add(-0.2),
      'red': red.multiply(0.0000275).add(-0.2)
  }).rename('NDVI'))
```

2.3 Cálculo FVC

Esta función calcula la Fracción de la Cobertura Vegetal del NDVI, requiere que exista la función NDVI

[ ]:
```
def fvc (image):
  ndvi = image.select('NDVI')
  fvc = image.expression('((ndvi-ndvi_bg)/(ndvi_vg - ndvi_bg))**2',
                         {'ndvi':ndvi,'ndvi_bg':0.2,'ndvi_vg':0.86})
  fvc = fvc.where(fvc.lt(0.0),0.0)
  fvc = fvc.where(fvc.gt(1.0),1.0)
  return image.addBands(fvc.rename('FVC'));
```

2.4 Cálculo Vapor de Agua

[ ]:
```
def tpw (image):
  date = ee.Date(image.get('system:time_start'))
  year = ee.Number.parse(date.format('yyyy'))
  month = ee.Number.parse(date.format('MM'))
  day = ee.Number.parse(date.format('dd'))
  date1 = ee.Date.fromYMD(year,month,day)
  date2 = date1.advance(1,'days')

  # Compute the time difference from landsat images
  def funcdatedist(image):
    return image.set('DateDist',
      ee.Number(image.get('system:time_start'))
      .subtract(date.millis()).abs())
  datedist = funcdatedist

  # Load atmospheric data collection
  TPWcollection = ee.ImageCollection('NCEP_RE/surface_wv').filter(ee.Filter.
↪date(date1.format('yyyy-MM-dd'), date2.format('yyyy-MM-dd'))).map(datedist)

  # Select the two closest model times
  closest = (TPWcollection.sort('DateDist')).toList(2);
```

```
  #Check if there is atmospheric data in the wanted day. If not creates a TPW␣
↪image with non-realistic values. These are then masked  to prevents errors
  tpw1 = ee.Image(ee.Algorithms.If(closest.size().eq(0), ee.Image.constant(-999.
↪0),ee.Image(closest.get(0)).select('pr_wtr') ))
  tpw2 = ee.Image(ee.Algorithms.If(closest.size().eq(0), ee.Image.constant(-999.
↪0),
                        ee.Algorithms.If(closest.size().eq(1), tpw1,
                        ee.Image(closest.get(1)).select('pr_wtr') )))

  time1 = ee.Number(ee.Algorithms.If(closest.size().eq(0), 1.0,
                        ee.Number(tpw1.get('DateDist')).divide(ee.
↪Number(21600000)) ))
  time2 = ee.Number(ee.Algorithms.If(closest.size().lt(2), 0.0,
                        ee.Number(tpw2.get('DateDist')).divide(ee.
↪Number(21600000)) ))

  tpw = tpw1.expression('tpw1*time2+tpw2*time1',
                            {'tpw1':tpw1,
                            'time1':time1,
                            'tpw2':tpw2,
                            'time2':time2
                            }).clip(image.geometry());
  return image.addBands(tpw.resample('bicubic').rename('TPW') )
```

2.5 Cálculo Emisividad

Se divide en dos partes

Esta función aplica una corrección de cobertura vegetal a la emisividad ASTER para obtener un componente de emisividad de suelo desnudo en cada píxel

[ ]:
```
# Bare ground emissivity functions for each band
def e_emiss_bare_band13(image):
  # Get ASTER emissivity
  aster = ee.Image("NASA/ASTER_GED/AG100_003")
  # Get ASTER FVC from NDVI
  aster_ndvi = aster.select('ndvi').multiply(0.01);

  aster_fvc = aster_ndvi.expression('((ndvi-ndvi_bg)/(ndvi_vg - ndvi_bg))**2',␣
↪{'ndvi':aster_ndvi,'ndvi_bg':0.2,'ndvi_vg':0.86});
  aster_fvc = aster_fvc.where(aster_fvc.lt(0.0),0.0)
  aster_fvc = aster_fvc.where(aster_fvc.gt(1.0),1.0)
  return image.expression('(EM - 0.99*fvc)/(1.0-fvc)',{
      'EM':aster.select('emissivity_band12').multiply(0.001),
      'fvc':aster_fvc}).clip(image.geometry())

def e_emiss_bare_band14(image):
```

```
  # Get ASTER emissivity
  aster = ee.Image("NASA/ASTER_GED/AG100_003")
  # Get ASTER FVC from NDVI
  aster_ndvi = aster.select('ndvi').multiply(0.01);

  aster_fvc = aster_ndvi.expression('((ndvi-ndvi_bg)/(ndvi_vg - ndvi_bg))**2',␣
↪{'ndvi':aster_ndvi,'ndvi_bg':0.2,'ndvi_vg':0.86});
  aster_fvc = aster_fvc.where(aster_fvc.lt(0.0),0.0)
  aster_fvc = aster_fvc.where(aster_fvc.gt(1.0),1.0)
  return image.expression('(EM - 0.99*fvc)/(1.0-fvc)',{
      'EM':aster.select('emissivity_band12').multiply(0.001),
      'fvc':aster_fvc}).clip(image.geometry())
```

Esta función calcula la emisividad de la superficie para Landsat

[ ]:
```
def se(image):
  fvc = image.select('FVC');
  em13 = e_emiss_bare_band13(image)
  em14 = e_emiss_bare_band14(image)
  # Compute the dynamic emissivity for Landsat em13d
  eI = image.expression('fvc*0.99+(1-fvc)*em13',{'fvc':fvc,'em13':em13}).
↪rename('e_i')
  # Compute the dynamic emissivity for Landsat em14d
  eJ = image.expression('fvc*0.99+(1-fvc)*em14',{'fvc':fvc,'em14':em14}).
↪rename('e_j')
  return image.addBands([eI, eJ]);
# emiss_bare_band11 = e_emiss_bare_band11()
```

2.6 Cálculo LST

[ ]:
```
def lst(image):
  lst = image.expression(
      'Ti + c1*(Ti - Tj) + c2*(Ti-Tj)** + c0 + (c3+c4*w)*(1-e) + (c5 +␣
↪(c6*w))*dE',{
      'c0': -0.268,
      'c1': 1.378,
      'c2': 0.183,
      'c3': 54.30,
      'c4': -2.238,
      'c5': -129.20,
      'c6': 16.4,
      'w': image.select('TPW'),
      'r': 0.984,
      'Ti': image.select('B10'),
      'Tj': image.select('B11'),
      'e':  (image.select('e_i').add(image.select('e_j'))).multiply(0.5),
      'dE': (image.select('e_i').subtract(image.select('e_j')))
```

```
    })
  lst = lst.updateMask(image.select('TPW').lt(0).Not()).rename('LST')
  return image.addBands(lst)
```

2.7 Algoritmo LST

[ ]:
```
landsatDictionary = ee.Dictionary({
  'L4': {
    'TOA': ee.ImageCollection('LANDSAT/LT04/C01/T1_TOA'),
    'SR': ee.ImageCollection('LANDSAT/LT04/C01/T1_SR'),
    'TIR': ['B6',]
  },
  'L5': {
    'TOA': ee.ImageCollection('LANDSAT/LT05/C01/T1_TOA'),
    'SR': ee.ImageCollection('LANDSAT/LT05/C01/T1_SR'),
    'TIR': ['B6',]
  },
  'L7': {
    'TOA': ee.ImageCollection('LANDSAT/LE07/C01/T1_TOA'),
    'SR': ee.ImageCollection('LANDSAT/LE07/C01/T1_SR'),
    'TIR': ['B6_VCID_1','B6_VCID_2'],
  },
  'L8': {
    'TOA': ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA'),
    'SR': ee.ImageCollection('LANDSAT/LC08/C02/T1_L2'),
    'TIR': ['B10','B11'],
    'VISW': ['SR_B2','SR_B3','SR_B4','SR_B5','QA_PIXEL']
  }
})
```

Este invoca a los módulos y genera la imagen

[ ]:
```
def collection(landsat, start, end, geometry):
  collectionDict = ee.Dictionary(landsatDictionary.get(landsat))
  landsatTOA =  ee.ImageCollection(collectionDict.get('TOA')).filter(ee.Filter.
↪date(start, end)).filterBounds(geometry)
  landsatSR = ee.ImageCollection(collectionDict.get('SR')).filter(ee.Filter.
↪date(start, end)).filterBounds(geometry).map(cloudmask_sr).map(ndvi).map(fvc).
↪map(tpw).map(se)

  # Combine collections all channels from surface reflectance collection
  # Except tir channels: from TOA collection select TIR bands
  tir = ee.List(collectionDict.get('TIR'))
  visw = ee.List(collectionDict.get('VISW')).add('NDVI').add('FVC').add('TPW').
↪add('e_i').add('e_j')

  landsatALL = (landsatSR.select(visw).combine(landsatTOA.select(tir)))
```

```
  landsatLST = landsatALL.map(lst)
  return(landsatLST)
```

3 Módulos Adicionales

3.1 Mosaico por fechas

Une las dos imágenes, y la almacena en una sola

[ ]:
```
def mosaicbydate (imcol):
  imlist = imcol.toList(imcol.size());
  def ud (im):
    return ee.Image(im).date().format("YYYY-MM-dd")
  unique_dates = imlist.map(ud).distinct()
  def mi (d):
    d = ee.Date(d)
    im = imcol.filterDate(d, d.advance(1, "day")).mosaic()
    return im.set(
          "system:time_start", d.millis(),
          "system:id", d.format("YYYY-MM-dd"));
  mosaic_imlist = unique_dates.map(mi)
  return ee.ImageCollection(mosaic_imlist);
```

3.2 Recorte de Zona

Corta la región de interés

[ ]:
```
def clipImage (imcol):
  return imcol.clip(geometry)
```

3.3 Clasificación de Nubosidad

Esta función computa la cantidad de nubosidad dentro de la región de estudio

[ ]:
```
def cloudcover(image):
  def totPixels(image):
    totPixels = ee.Number(ee.Image(1).reduceRegion(
      reducer= ee.Reducer.count(),
      scale= 10,
      geometry= geometry,
      maxPixels= 999999999,
    ).values().get(0))
    return totPixels
  def actPixels(image):
    actPixels = ee.Number(image.reduceRegion(
      reducer= ee.Reducer.count(),
      scale= 10,
```

```
      geometry= geometry,
      maxPixels= 999999999,
    ).values().get(0))
    return actPixels
  totPixels = totPixels(image)
  actPixels = actPixels(image)
  percCover = actPixels.divide(totPixels).multiply(100).round();
  return image.set('percCover', percCover);
```

3.4 Reducción de región

La reducción de los píxeles que intersecan la región de interés de una estadística se realizará varias veces. Definir una función reutilizable que pueda realizar la tarea para cada conjunto de datos. La función acepta argumentos como escala y método de reducción para parametrizar la operación para cada análisis en particular.

[ ]:
```
def create_reduce_region_function(geometry,
                                  reducer=ee.Reducer.mean(),
                                  scale=4000,
                                  crs='EPSG:4326',
                                  bestEffort=True,
                                  maxPixels=1e9):#,
                                  #tileScale=4):
  """Creates a region reduction function.

  Creates a region reduction function intended to be used as the input function
  to ee.ImageCollection.map() for reducing pixels intersecting a provided region
  to a statistic for each image in a collection. See ee.Image.reduceRegion()
  documentation for more details.

  Args:
    geometry:
      An ee.Geometry that defines the region over which to reduce data.
    reducer:
      Optional; An ee.Reducer that defines the reduction method.
    scale:
      Optional; A number that defines the nominal scale in meters of the
      projection to work in.
    crs:
      Optional; An ee.Projection or EPSG string ('EPSG:5070') that defines
      the projection to work in.
    bestEffort:
      Optional; A Boolean indicator for whether to use a larger scale if the
      geometry contains too many pixels at the given scale for the operation
      to succeed.
    maxPixels:
      Optional; A number specifying the maximum number of pixels to reduce.
```

```
    tileScale:
      Optional; A number representing the scaling factor used to reduce
      aggregation tile size; using a larger tileScale (e.g. 2 or 4) may enable
      computations that run out of memory with the default.

  Returns:
    A function that accepts an ee.Image and reduces it by region, according to
    the provided arguments.
  """

  def reduce_region_function(img):
    """Applies the ee.Image.reduceRegion() method.

    Args:
      img:
        An ee.Image to reduce to a statistic by region.

    Returns:
      An ee.Feature that contains properties representing the image region
      reduction results per band and the image timestamp formatted as
      milliseconds from Unix epoch (included to enable time series plotting).
    """

    stat = img.reduceRegion(
        reducer=reducer,
        geometry=geometry,
        scale=scale,
        crs=crs,
        bestEffort=bestEffort,
        maxPixels=maxPixels)#,
        #tileScale=tileScale)

    return ee.Feature(geometry, stat).set({'millis': img.date().millis()})
  return reduce_region_function
```

3.5 De GEE a Python (Cambio de formato)

El resultado de la función de reducción de región anterior aplicada a una ee.ImageCollection produce una ee.FeatureCollection. Estos datos deben transferirse al kernel de Python, pero las colecciones de funciones serializadas son grandes y difíciles de manejar. Este paso define una función para convertir la colección de características en un ee.Dictionary donde las claves son nombres de propiedades de características y los valores son listas correspondientes de valores de propiedades, que los pandas pueden manejar fácilmente.

[ ]:
```
# Define a function to transfer feature properties to a dictionary.
def fc_to_dict(fc):
  prop_names = fc.first().propertyNames()
```

```
  prop_lists = fc.reduceColumns(
      reducer=ee.Reducer.toList().repeat(prop_names.size()),
      selectors=prop_names).get('list')

  return ee.Dictionary.fromLists(prop_names, prop_lists)
```

3.6 Mapa interactivo

Permite generar un mapa interactivo por medio de Folium

[ ]:
```
# Define a method for displaying Earth Engine image tiles to folium map.
def add_ee_layer(self, ee_image_object, vis_params, name):
    """Adds a method for displaying Earth Engine image tiles to folium map."""
    map_id_dict = ee.Image(ee_image_object).getMapId(vis_params)
    folium.raster_layers.TileLayer(
        tiles=map_id_dict['tile_fetcher'].url_format,
        attr='Map Data &copy; <a href="https://earthengine.google.com/">Google␣
 ↪Earth Engine</a>',
        name=name,
        overlay=True,
        control=True
    ).add_to(self)

# Add EE drawing method to folium.
folium.Map.add_ee_layer = add_ee_layer
```

4 Cálculo de Islas de Calor

[ ]:
```
geometry = ee.FeatureCollection("users/edgarpzavaleta/ZMVMWGS84")
satellite = 'L8'
startDate = '2013-03-18'
endDate = '2023-01-29'
landsatColl = collection(satellite, startDate, endDate, geometry)
landsatColl = landsatColl.filter(ee.Filter.eq('WRS_PATH', 26))

#Mosaic by date
mosaic = mosaicbydate(landsatColl).map(clipImage);
```

5 Análisis de Datos

5.1 Importación de Datos

1. Carga las imágenes del mosaico
2. Carga la geometría (es un poco redundante, se optimizará a futuro)
3. Calcula el porcentaje de nubes dentro de la región
4. Selecciona las imágenes que tengan un porcentaje mayor al 20, 50 y 80 en cuanto a “limpieza”

[ ]:
```
lst = mosaic.select('LST')
aoi = geometry.geometry()
lstcc = lst.map(cloudcover)
lstcc20 = lstcc.filter(ee.Filter.gt('percCover', 20))
lstcc50 = lstcc.filter(ee.Filter.gt('percCover', 50))
lstcc80 = lstcc.filter(ee.Filter.gt('percCover', 80))
```

El primer renglón corresponde a la cantidad total de imágenes (recordar que ya une las dos imágenes de Landsat) P.D: Si no se requiere calcular o ver diferencia, mejor no ejecutar. Son funciones que consumen tiempo y recursos

[ ]:
```
print(lstcc50.size().getInfo())
```

```
159
```

5.2 Reducción de Datos

1. Crea una función de reducción de región.
2. Asigna la función sobre la última colección de imágenes para reducir cada imagen.
3. Filtra las entidades resultantes que tengan valores calculados nulos (ocurre cuando todos los píxeles en un AOI están enmascarados).

[ ]:
```
#reduce_lst = create_reduce_region_function(
#  geometry=aoi, reducer=ee.Reducer.mean(), scale=5000, crs='EPSG:4326')

#lst_stat_fc = ee.FeatureCollection(lst.map(reduce_lst)).filter(
#    ee.Filter.notNull(lst.first().bandNames()))

reduce_lst = create_reduce_region_function(
  geometry=aoi, reducer=ee.Reducer.mean(), scale=10000)

lst_stat_fc = ee.FeatureCollection(lstcc50.map(reduce_lst)).filter(
    ee.Filter.notNull(lstcc.first().bandNames()))
```

5.3 Transferencia de servidor a cliente (o el paso de la muerte)

ee.FeatureCollection debe convertirse en un diccionario y transferirse al kernel de Python. 1. Se aplica la función fc_to_dict para convertir de ee.FeatureCollection a ee.Dictionary. 2. Se llama a getInfo() en ee.Dictionary para transferir los datos del lado del cliente.

P.D: no tener miedo, es normal que el proceso sea lento

[ ]:
```
lst_dict = fc_to_dict(lst_stat_fc).getInfo()
```

El resultado es un diccionario de Python. Se imprime una pequeña parte para ver cómo el nuevo estilo.

[ ]:
```
print(type(lst_dict), '\n')
for prop in lst_dict.keys():
    print(prop + ':', lst_dict[prop][0:3] + ['...'])
```

```
<class 'dict'>

LST: [314.5259684748315, 306.8866007880906, 311.37556420329764, '...']
millis: [1366243200000, 1369008000000, 1370390400000, '...']
system:index: ['0', '1', '2', '...']
```

Se convierte el diccionario de Python en un DataFrame de pandas.

[ ]:
```
lst_df = pd.DataFrame(lst_dict)
display(lst_df)
print(lst_df.dtypes)
```

```
           LST         millis system:index
0   314.525968  1366243200000            0
1   306.886601  1369008000000            1
2   311.375564  1370390400000            2
3   294.884405  1385596800000           10
4   297.339211  1386979200000           11
..         ...            ...          ...
81  307.493984  1648252800000          192
82  309.046344  1652400000000          195
83  302.860403  1656547200000          197
84  299.908759  1664841600000          203
85  299.521324  1667606400000          205

[86 rows x 3 columns]
```

```
LST             float64
millis            int64
system:index     object
dtype: object
```

5.3.1 Exportación e Importación de Dataframe de Google Drive

Esta parte es para ahorrar el "paso de la muerte", calculada desde el 2013-03-18 hasta el 2023-02-01. Sirve para almacenar, cargar y ejecutar archivos y de esta manera ahorra tiempo en su ejecución.

[ ]:
```
from google.colab import drive

drive.mount('/content/drive')
root_path = '/content/drive/My Drive/Tesis/'
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).

#Para exportar archivos tipo csv de un dataframe a Drive
df.to_csv(root_path+'df_lst.csv',  encoding='utf-8', index=False)
```

```
#Para importar archivos tipo csv a un dataframe desde Drive
df=pd.read_csv(root_path+'/data.csv')
```

```
[ ]: lst_df.to_csv(root_path+'df_lst_cc_80.csv',  encoding='utf-8', index=False)
```

Aquí se importan 4 archivos, el orden de los dataframes es el siguiente:

1. Todos las imágenes (218 imágenes)
2. Imágenes con un porcentaje de nubosidad menor al 20% (86 imágenes)
3. Imágenes con un porcentaje de nubosidad menor al 80% (194 imágenes)
4. Imágenes con un porcentaje de nubosidad inferior al 50% (159 imágenes)

```
[ ]: lst_df = pd.read_csv(root_path+'/df_lst.csv')            #218 imágenes
     lst20_df = pd.read_csv(root_path+'/df_lst_cc_20.csv')   #194 imágenes
     lst50_df = pd.read_csv(root_path+'/df_lst_cc_50.csv')   #159 imágenes
     lst80_df = pd.read_csv(root_path+'/df_lst_cc_80.csv')   #86 imágenes
```

5.4 Adición de columnas

Se agregan columnas de fecha derivadas de los milisegundos de la columna de época de Unix. La biblioteca pandas proporciona funciones y objetos para las marcas de tiempo y el objeto DataFrame permite una fácil mutación.

Se define una función para agregar variables de fecha al DataFrame: año, mes, día y día del año (DOY).

```
[ ]: # Function to add date variables to DataFrame.
     def add_date_info(df):
       df['Timestamp'] = pd.to_datetime(df['millis'], unit='ms')
       df['Year'] = pd.DatetimeIndex(df['Timestamp']).year
       df['Month'] = pd.DatetimeIndex(df['Timestamp']).month
       df['Day'] = pd.DatetimeIndex(df['Timestamp']).day
       df['DOY'] = pd.DatetimeIndex(df['Timestamp']).dayofyear
       return df
```

```
[ ]: lst_df = add_date_info(lst_df)
     lst_df.tail(5)
```

```
[ ]:
```

	LST	millis	system:index	Timestamp	Year	Month	Day	DOY
212	289.960541	1445040000000	212	2015-10-17	2015	10	17	290
213	290.138536	1472688000000	213	2016-09-01	2016	9	1	245
214	294.474057	1496188800000	214	2017-05-31	2017	5	31	151
215	289.971017	1634428800000	216	2021-10-17	2021	10	17	290
216	294.163926	1655164800000	217	2022-06-14	2022	6	14	165

```
[ ]: #lst80_df = add_date_info(lst80_df)
     #lst50_df = add_date_info(lst50_df)
     lst20_df = add_date_info(lst20_df)
```

[ ]:
```
lst20_df
```

[ ]:

	LST	millis	system:index	Timestamp	Year	Month	Day	DOY
0	314.525968	1366243200000	0	2013-04-18	2013	4	18	108
1	306.886601	1369008000000	1	2013-05-20	2013	5	20	140
2	311.375564	1370390400000	2	2013-06-05	2013	6	5	156
3	294.884405	1385596800000	10	2013-11-28	2013	11	28	332
4	297.339211	1386979200000	11	2013-12-14	2013	12	14	348
..	...	...	...	...	...	...	...	...
81	307.493984	1648252800000	192	2022-03-26	2022	3	26	85
82	309.046344	1652400000000	195	2022-05-13	2022	5	13	133
83	302.860403	1656547200000	197	2022-06-30	2022	6	30	181
84	299.908759	1664841600000	203	2022-10-04	2022	10	4	277
85	299.521324	1667606400000	205	2022-11-05	2022	11	5	309

[86 rows x 8 columns]

5.5 Limpieza

Aquí se borran las columnas que no serán utilizadas, así como el cambio de nombre de algunas columnas

[ ]:
```
# Function to erase some variables of a DataFrame.
def erase_df_vaiables(df):
  df = df.drop(columns=['millis', 'system:index'])
  return df
```

[ ]:
```
lst_df = lst_df.drop(columns=['millis', 'system:index'])
lst_df.head(5)
```

[ ]:

	LST	Timestamp	Year	Month	Day	DOY
0	314.525968	2013-04-18	2013	4	18	108
1	306.886601	2013-05-20	2013	5	20	140
2	311.375564	2013-06-05	2013	6	5	156
3	298.695594	2013-07-23	2013	7	23	204
4	296.876009	2013-08-08	2013	8	8	220

[ ]:
```
#lst80_df = erase_df_vaiables(lst80_df)
#lst50_df = erase_df_vaiables(lst50_df)
lst20_df = erase_df_vaiables(lst20_df)
```

[ ]:
```
pd.options.display.max_rows = None
lst20_df
```

[ ]:

	LST	Timestamp	Year	Month	Day	DOY
0	314.525968	2013-04-18	2013	4	18	108
1	306.886601	2013-05-20	2013	5	20	140
2	311.375564	2013-06-05	2013	6	5	156

3	294.884405	2013-11-28	2013	11	28	332
4	297.339211	2013-12-14	2013	12	14	348
5	304.862685	2014-03-20	2014	3	20	79
6	306.863747	2014-04-05	2014	4	5	95
7	308.099496	2014-04-21	2014	4	21	111
8	299.310905	2014-07-26	2014	7	26	207
9	298.702227	2014-08-27	2014	8	27	239
10	300.733281	2015-02-19	2015	2	19	50
11	301.976169	2015-03-07	2015	3	7	66
12	300.076445	2015-04-08	2015	4	8	98
13	307.225005	2015-04-24	2015	4	24	114
14	305.805161	2015-05-10	2015	5	10	130
15	299.611134	2015-05-26	2015	5	26	146
16	305.124985	2015-06-11	2015	6	11	162
17	298.003200	2015-07-13	2015	7	13	194
18	301.170328	2015-07-29	2015	7	29	210
19	295.450804	2015-12-20	2015	12	20	354
20	297.226531	2016-02-06	2016	2	6	37
21	302.793517	2016-02-22	2016	2	22	53
22	305.779892	2016-03-25	2016	3	25	85
23	307.218665	2016-04-10	2016	4	10	101
24	304.467506	2016-04-26	2016	4	26	117
25	302.841503	2016-05-28	2016	5	28	149
26	300.313085	2016-06-29	2016	6	29	181
27	295.086780	2016-07-15	2016	7	15	197
28	299.293925	2016-10-03	2016	10	3	277
29	300.108197	2016-10-19	2016	10	19	293
30	297.129402	2016-12-06	2016	12	6	341
31	295.357459	2017-01-07	2017	1	7	7
32	301.896226	2017-02-08	2017	2	8	39
33	306.237735	2017-03-28	2017	3	28	87
34	307.157225	2017-04-13	2017	4	13	103
35	304.994159	2017-06-16	2017	6	16	167
36	296.501679	2017-07-02	2017	7	2	183
37	299.172688	2017-09-20	2017	9	20	263
38	300.506266	2018-02-27	2018	2	27	58
39	303.121204	2018-03-15	2018	3	15	74
40	304.236365	2018-03-31	2018	3	31	90
41	307.355565	2018-04-16	2018	4	16	106
42	310.564161	2018-05-18	2018	5	18	138
43	308.135809	2018-06-03	2018	6	3	154
44	300.990022	2018-07-05	2018	7	5	186
45	299.567737	2018-07-21	2018	7	21	202
46	297.193733	2018-10-09	2018	10	9	282
47	297.781727	2018-10-25	2018	10	25	298
48	294.507534	2018-11-10	2018	11	10	314
49	297.014080	2019-01-29	2019	1	29	29

50	297.656603	2019-02-14	2019	2	14	45
51	305.035340	2019-03-02	2019	3	2	61
52	302.925020	2019-03-18	2019	3	18	77
53	308.190083	2019-04-03	2019	4	3	93
54	306.977140	2019-04-19	2019	4	19	109
55	311.773455	2019-05-21	2019	5	21	141
56	301.480723	2019-07-08	2019	7	8	189
57	296.613693	2019-08-09	2019	8	9	221
58	297.504267	2019-11-29	2019	11	29	333
59	297.397377	2019-12-31	2019	12	31	365
60	303.704358	2020-03-20	2020	3	20	80
61	304.577523	2020-04-05	2020	4	5	96
62	308.291085	2020-04-21	2020	4	21	112
63	311.748308	2020-05-23	2020	5	23	144
64	299.755802	2020-07-10	2020	7	10	192
65	296.626589	2020-08-11	2020	8	11	224
66	298.170140	2020-09-28	2020	9	28	272
67	299.013777	2020-10-14	2020	10	14	288
68	302.044497	2020-11-15	2020	11	15	320
69	296.377177	2020-12-01	2020	12	1	336
70	296.587608	2020-12-17	2020	12	17	352
71	294.251207	2021-01-18	2021	1	18	18
72	301.097450	2021-02-19	2021	2	19	50
73	306.305123	2021-03-07	2021	3	7	66
74	311.514938	2021-03-23	2021	3	23	82
75	309.677343	2021-04-08	2021	4	8	98
76	309.661988	2021-04-24	2021	4	24	114
77	297.210626	2021-11-02	2021	11	2	306
78	298.566282	2022-01-05	2022	1	5	5
79	298.178516	2022-02-06	2022	2	6	37
80	304.719601	2022-02-22	2022	2	22	53
81	307.493984	2022-03-26	2022	3	26	85
82	309.046344	2022-05-13	2022	5	13	133
83	302.860403	2022-06-30	2022	6	30	181
84	299.908759	2022-10-04	2022	10	4	277
85	299.521324	2022-11-05	2022	11	5	309

5.6 Primer acercamiento

1. Se describe el tipo de datos
2. Se conoce el valor máximo del LST
3. Se conoce el valor mínimo del LST

[ ]:
```
lst_df.dtypes
```

[ ]:
```
LST             float64
millis            int64
```

```
system:index      int64
dtype: object
```

[ ]:
```
lst20_df.dtypes
```

[ ]:
```
LST                    float64
millis                   int64
system:index             int64
Timestamp       datetime64[ns]
Year                     int64
Month                    int64
Day                      int64
DOY                      int64
dtype: object
```

[ ]:
```
lst_df['LST'].max()
```

[ ]:
```
314.5259684748315
```

[ ]:
```
lst_df['LST'].min()
```

[ ]:
```
271.21120037280207
```

[ ]:
```
lst20_df['LST'].max()
```

[ ]:
```
314.5259684748315
```

[ ]:
```
lst20_df['LST'].min()
```

[ ]:
```
294.25120725368856
```

¿En qué fecha tenemos el resultado con mayor temperatura promedio? En mayo 18 de 2013 Temperatura mínima: 274 Temperatura máxima: 381

[ ]:
```
print(lst20_df[lst20_df.LST == lst20_df.LST.max()])
```

```
          LST         millis  system:index  Timestamp  Year  Month  Day  DOY
0  314.525968  1366243200000             0 2013-04-18  2013      4   18  108
```

¿En qué fecha tenemos el resultado con menor temperatura promedio? En enero 18 de 2021 Temperatura mínima: 245 Temperatura máxima: 315

[ ]:
```
print(lst20_df[lst20_df.LST == lst20_df.LST.min()])
```

```
           LST         millis  system:index  Timestamp  Year  Month  Day  DOY
71  294.251207  1610928000000           167 2021-01-18  2021      1   18   18
```

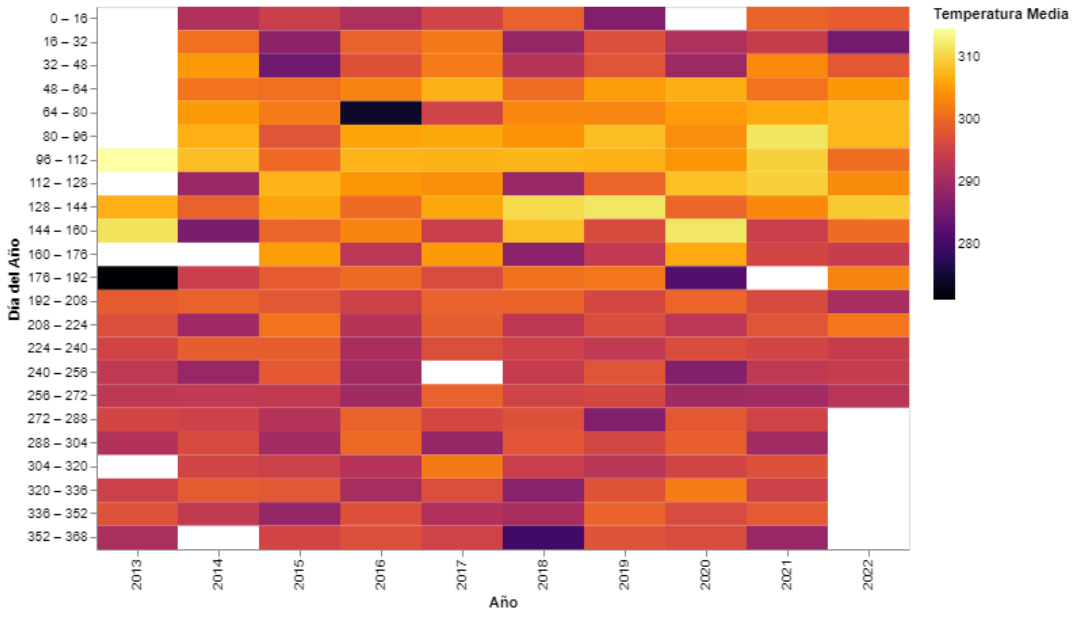
5.7 Recursos Gráficos

5.7.1 Mapa de Calor

Se grafican los datos LST como un mapa de calor de calendario. Se establece el año de observación como la variable del eje "x", el mes como el eje "y" y el valor LST como color.

[ ]:
```
        alt.Chart(lst_df).mark_rect().encode(
            x=alt.X('Year:O', title='Año'),
            y=alt.Y('DOY:O', title='Día del Año', bin=alt.Bin(extent=[0, 366],␣
↪step=16)),
            color=alt.Color(
                'LST:Q', scale=alt.Scale(scheme="inferno"), title='Temperatura␣
↪Media'),
            tooltip=[
                alt.Tooltip('Year:O', title='Año'),
                alt.Tooltip('LST:Q', title='Valor LST'),
                alt.Tooltip('DOY', title='Día del Año'),
            ]
        ).properties(width=600, height=400)
```

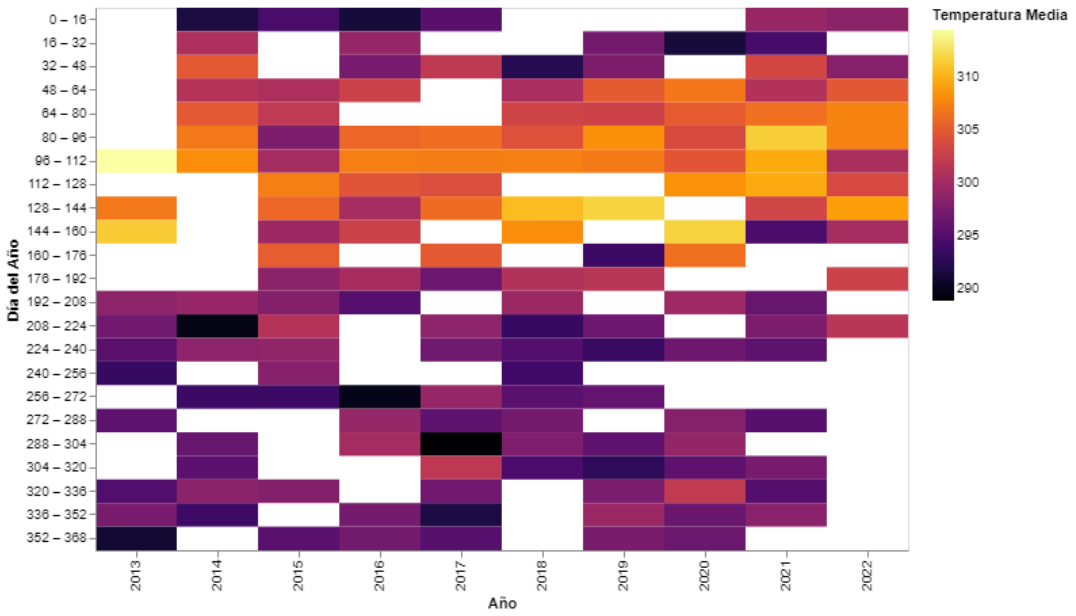
[ ]:
```
␣
```



[ ]:
```
alt.Chart(lst50_df).mark_rect().encode(
    x=alt.X('Year:O', title='Año'),
    y=alt.Y('DOY:O', title='Día del Año', bin=alt.Bin(extent=[0, 366], step=16)),
    color=alt.Color(
        'LST:Q', scale=alt.Scale(scheme="inferno"), title='Temperatura Media'),
    tooltip=[
        alt.Tooltip('Year:O', title='Año'),
```

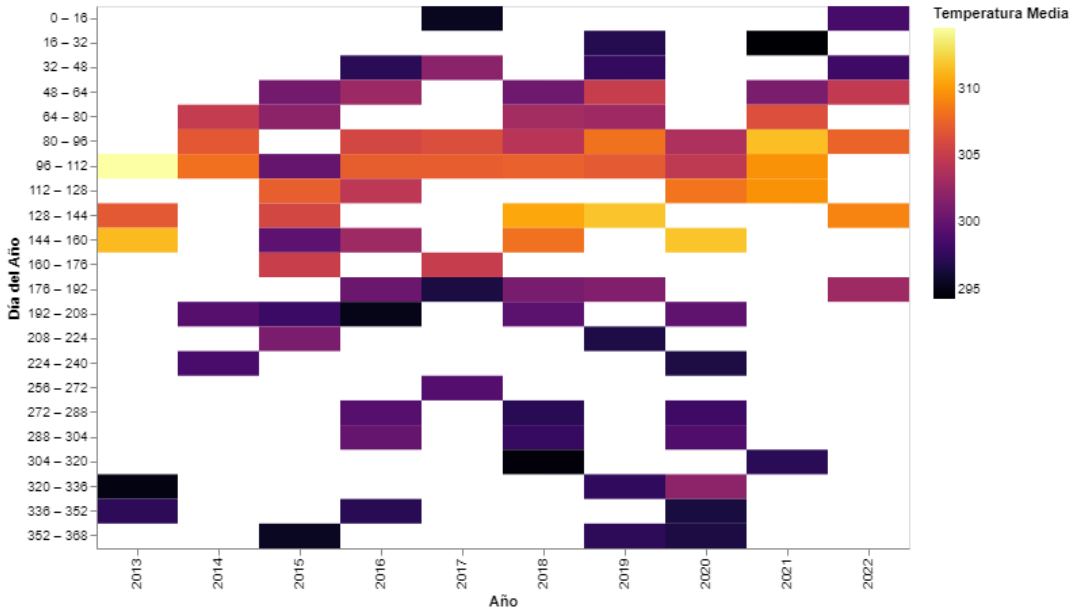
```
        alt.Tooltip('LST:Q', title='Valor LST'),
        alt.Tooltip('DOY', title='Día del Año'),
    ]
).properties(width=600, height=400)
```

[ ]:



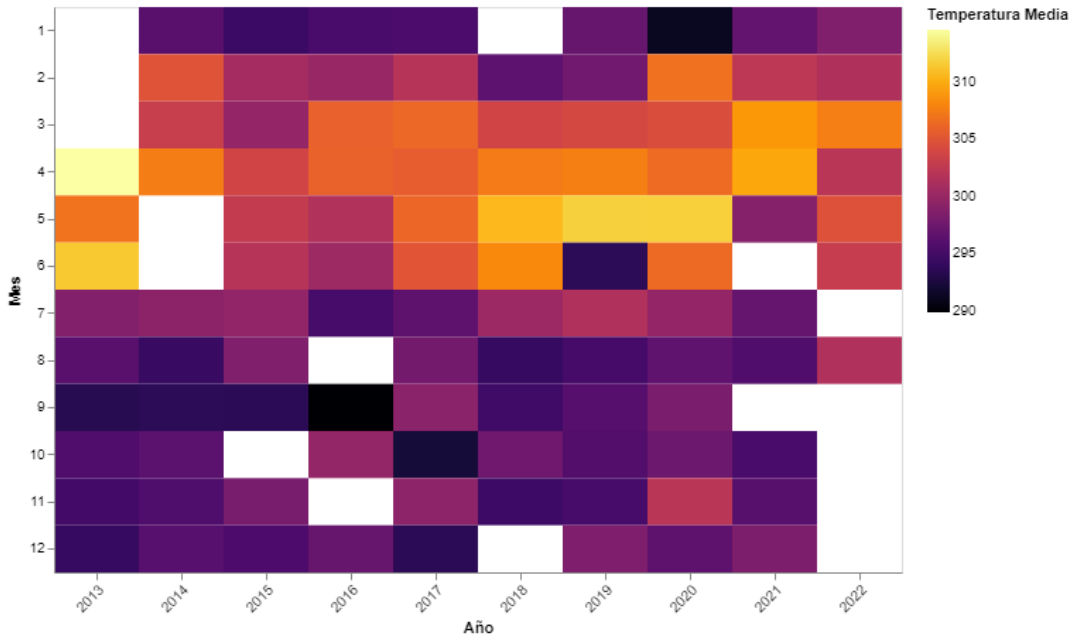
[ ]:
```
alt.Chart(lst80_df).mark_rect().encode(
    x=alt.X('Year:O', title='Año'),
    y=alt.Y('DOY:O', title='Día del Año', bin=alt.Bin(extent=[0, 366], step=16)),
    color=alt.Color(
        'LST:Q', scale=alt.Scale(scheme="inferno"), title='Temperatura Media'),
    tooltip=[
        alt.Tooltip('Year:O', title='Año'),
        alt.Tooltip('LST:Q', title='Valor LST'),
        alt.Tooltip('DOY', title='Día del Año'),
    ]
).properties(width=600, height=400)
```

[ ]:



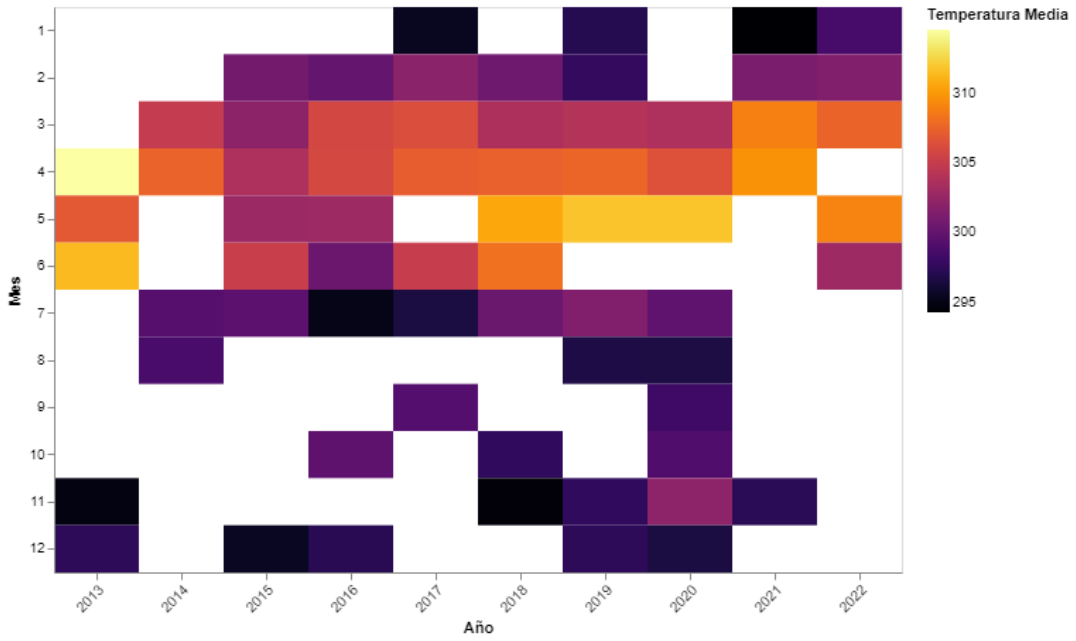
[ ]:
```
alt.Chart(lst50_df).mark_rect().encode(
    x=alt.X('Year:O',
            axis=alt.Axis(
                labelAngle=-45,
                title='Año',
    )),
    y=alt.Y('Month:O', title='Mes'),
    color=alt.Color(
        'mean(LST):Q', scale=alt.Scale(scheme="inferno"), title='Temperatura␣
 ↪Media'),
    tooltip=[
        alt.Tooltip('Year:O', title='Año'),
        alt.Tooltip('Month:Q', title='Mes'),
        alt.Tooltip('mean(LST):Q', title='LST'),
    ]
).properties(width=600, height=400)
```

[ ]:



[ ]:
```
alt.Chart(lst80_df).mark_rect().encode(
    x=alt.X('Year:O',
            axis=alt.Axis(
                labelAngle=-45,
                title='Año',
    )),
    y=alt.Y('Month:O', title='Mes'),
    color=alt.Color(
        'mean(LST):Q', scale=alt.Scale(scheme="inferno"), title='Temperatura␣
 ↪Media'),
    tooltip=[
        alt.Tooltip('Year:O', title='Año'),
        alt.Tooltip('Month:Q', title='Mes'),
        alt.Tooltip('mean(LST):Q', title='LST'),
    ]
).properties(width=600, height=400)
```

[ ]:



5.7.2 Gráfico de barras

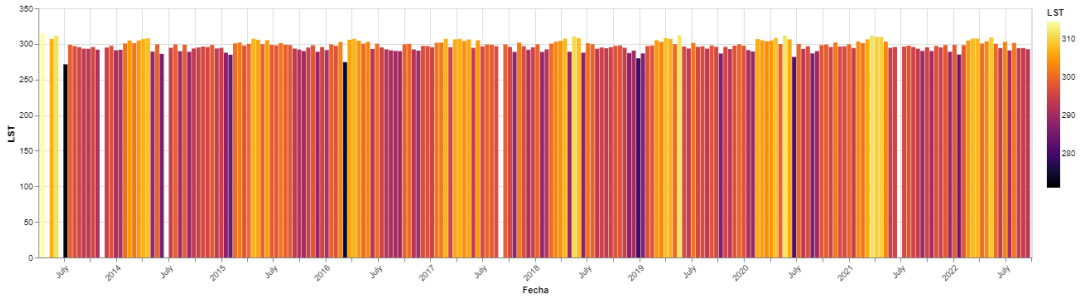
Se grafican las series temporales de LST como un gráfico de barras para interpretar más fácilmente los valores absolutos y compararlos a lo largo del tiempo. Aquí, la marca de tiempo de observación se representa en el eje x y el LST se representa tanto en el eje y como en el color. Dado que cada observación de LST tiene una marca de tiempo única que se puede trazar en el eje x, no es necesario agregar valores de LST como en el gráfico anterior. Se agrega información sobre herramientas al gráfico. Nota: pasando el mouse sobre las barras para revelar los valores de cada variable.

[ ]:
```
lst_df['LST'].mean()
```

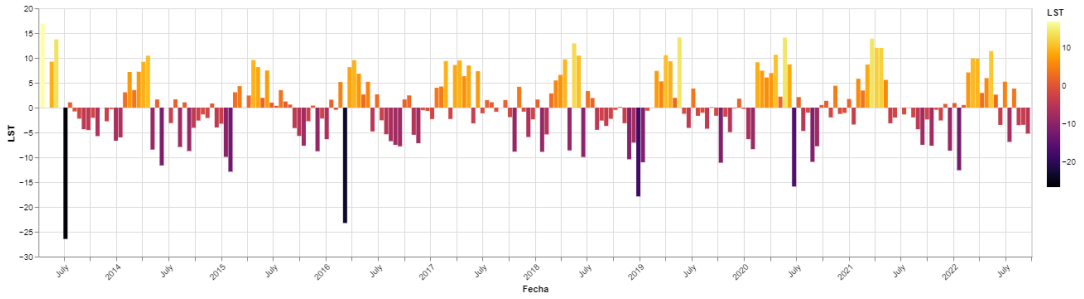
[ ]:
```
297.6472306639522
```

[ ]:
```
alt.Chart(lst_df).mark_bar().encode(
    x=alt.X('Timestamp:T',
            axis=alt.Axis(
                labelAngle=-45,
                title='Fecha')),
    y='LST:Q',
    color=alt.Color(
        'LST:Q', scale=alt.Scale(scheme='inferno')),
    tooltip=[
        alt.Tooltip('Timestamp:T', title='Fecha'),
        alt.Tooltip('LST:Q', title='LST')
    ]).properties(width=1200, height=300)
```

[ ]:



[ ]:
```
alt.Chart(lst_df.eval("LST = LST -  297.647231")).mark_bar().encode(
    x=alt.X('Timestamp:T',
            axis=alt.Axis(
                labelAngle=-45,
                title='Fecha')),
    y='LST:Q',
    color=alt.Color(
        'LST:Q', scale=alt.Scale(scheme='inferno')),
    tooltip=[
        alt.Tooltip('Timestamp:T', title='Fecha'),
        alt.Tooltip('LST:Q', title='LST')
    ]).properties(width=1200, height=300)
```

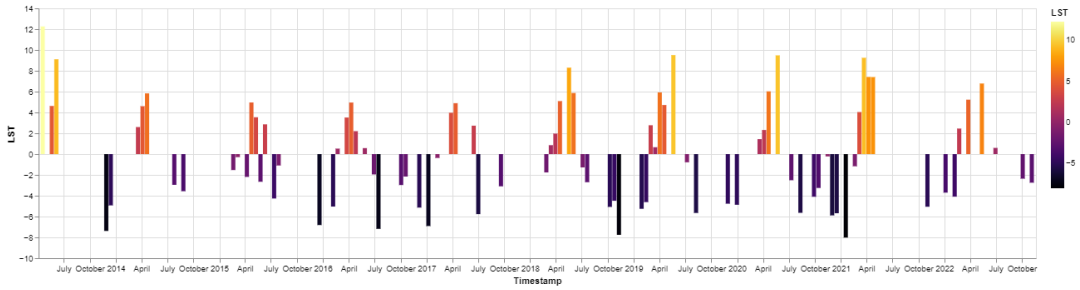
[ ]:


[ ]:
```
lst20_df['LST'].mean()
```

[ ]:
```
302.26793196097566
```

[ ]:
```
alt.Chart(lst20_df.eval("LST = LST -  302.26793196097566")).mark_bar().encode(
    x='Timestamp:T',
    y='LST:Q',
    color=alt.Color(
        'LST:Q', scale=alt.Scale(scheme='inferno')),
    tooltip=[
        alt.Tooltip('Timestamp:T', title='Date'),
        alt.Tooltip('LST:Q', title='LST')
    ]).properties(width=1200, height=300)
```

[ ]:

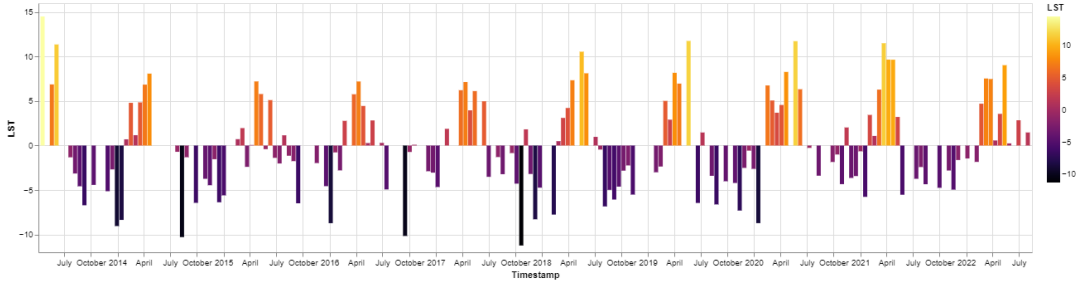


[ ]:
```
lst50_df['LST'].mean()
```

[ ]:
```
300.02117695120324
```

[ ]:
```
alt.Chart(lst50_df.eval("LST = LST -  300.02117695120324")).mark_bar().encode(
    x='Timestamp:T',
    y='LST:Q',
    color=alt.Color(
        'LST:Q', scale=alt.Scale(scheme='inferno')),
    tooltip=[
        alt.Tooltip('Timestamp:T', title='Date'),
        alt.Tooltip('LST:Q', title='LST')
    ]).properties(width=1200, height=300)
```

[ ]:



[ ]:
```
lst80_df['LST'].mean()
```

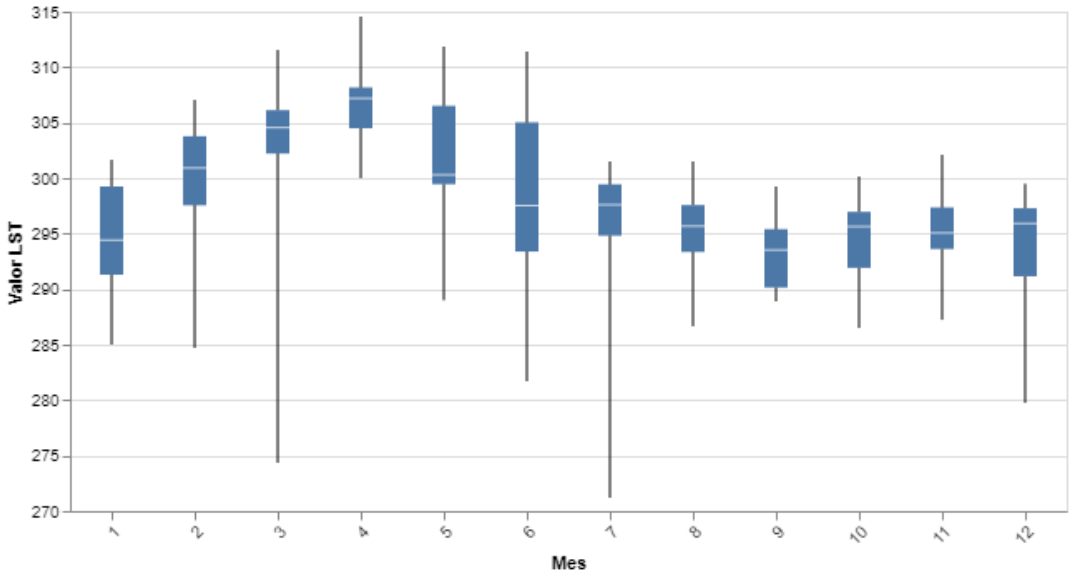
[ ]:
```
302.26793196097566
```

[ ]:
```
alt.Chart(lst80_df.eval("LST = LST -  302.32871507463216")).mark_bar().encode(
    x='Timestamp:T',
    y='LST:Q',
    color=alt.Color(
        'LST:Q', scale=alt.Scale(scheme='inferno')),
    tooltip=[
        alt.Tooltip('Timestamp:T', title='Fechas'),
        alt.Tooltip('LST:Q', title='LST')
    ]).properties(width=1200, height=300)
```

5.7.3 Diagrama de caja

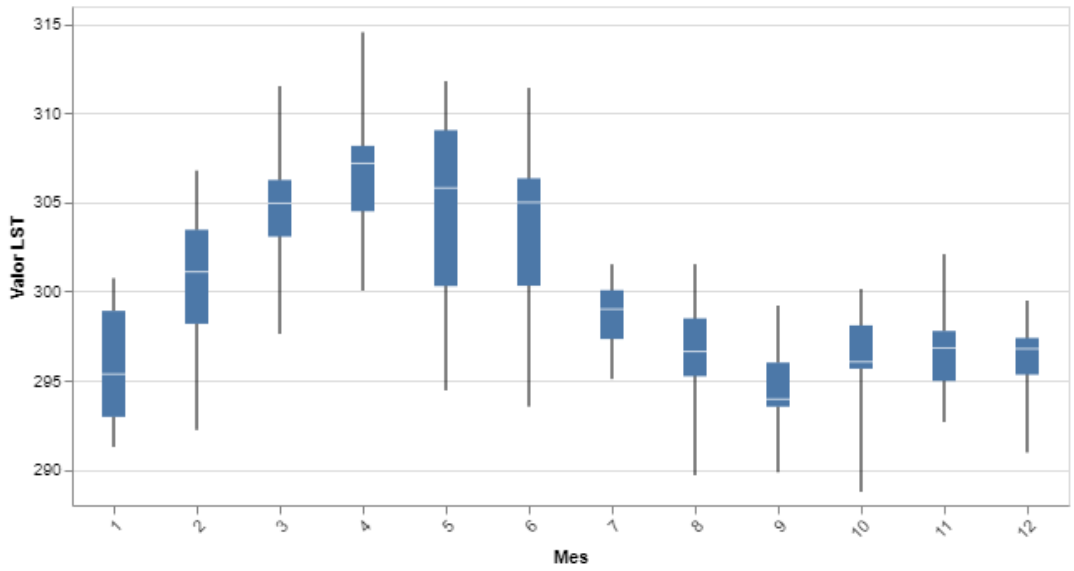
[ ]:
```
alt.Chart(lst_df).mark_boxplot(extent='min-max').encode(
    x=alt.X('Month:O',
            axis=alt.Axis(
                labelAngle=-45,
                title='Mes',
                #values=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
                #labelExpr="datum.value == 1 ? 'Enero' : datum.value == 2 ?␣
↪'Febrero' "),
    )),
    y=alt.Y('LST:Q', title='Valor LST', scale=alt.Scale(zero=False)),
).properties(width=600, height=300)
```

[ ]:

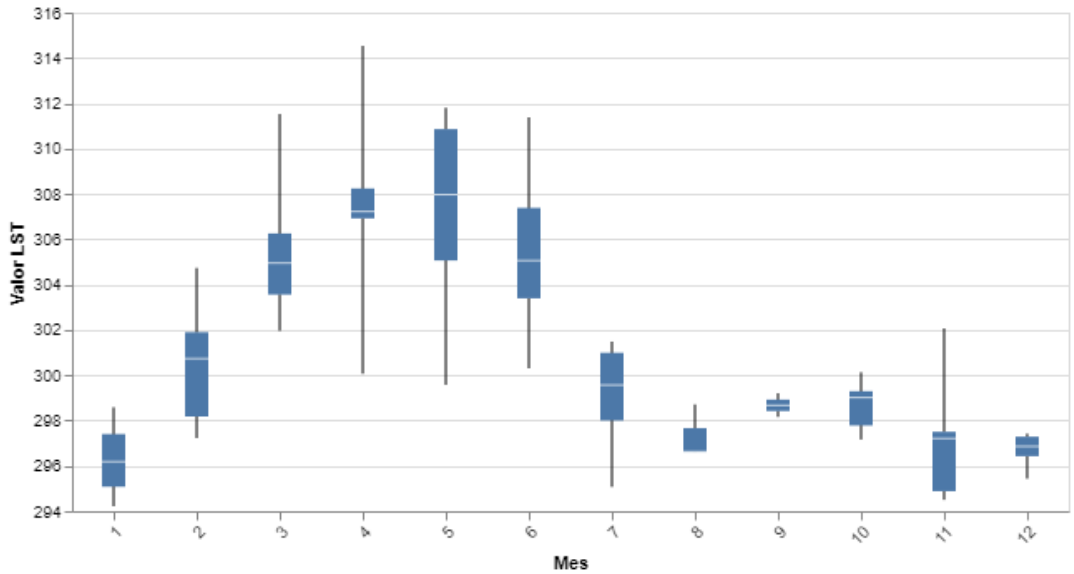


[ ]:
```
alt.Chart(lst50_df).mark_boxplot(extent='min-max').encode(
    x=alt.X('Month:O',
            axis=alt.Axis(
                labelAngle=-45,
                title='Mes',
                #values=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
                #labelExpr="datum.value == 1 ? 'Enero' : datum.value == 2 ?␣
↪'Febrero' "),
    )),
    y=alt.Y('LST:Q', title='Valor LST', scale=alt.Scale(zero=False)),
).properties(width=600, height=300)
```

[ ]:

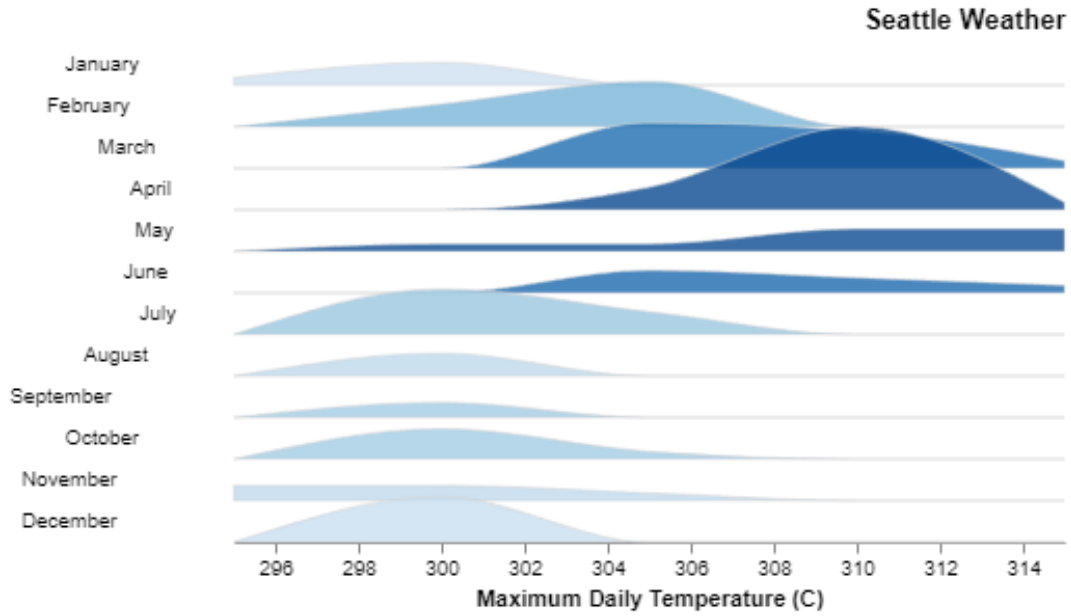


[ ]:
```
alt.Chart(lst80_df).mark_boxplot(extent='min-max').encode(
    x=alt.X('Month:O',
            axis=alt.Axis(
                labelAngle=-45,
                title='Mes',
                #values=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
                #labelExpr="datum.value == 1 ? 'Enero' : datum.value == 2 ?
 ↪'Febrero' "),
    )),
    y=alt.Y('LST:Q', title='Valor LST', scale=alt.Scale(zero=False)),
).properties(width=600, height=300)
```

[ ]:


[ ]:
```
step = 20
overlap = 1

alt.Chart(lst80_df, height=step).transform_timeunit(
    Month='month(Timestamp)'
).transform_joinaggregate(
    mean_temp='mean(LST)', groupby=['Month']
).transform_bin(
    ['bin_max', 'bin_min'], 'LST'
).transform_aggregate(
    value='count()', groupby=['Month', 'mean_temp', 'bin_min', 'bin_max']
).transform_impute(
    impute='value', groupby=['Month', 'mean_temp'], key='bin_min', value=0
).mark_area(
    interpolate='monotone',
    fillOpacity=0.8,
    stroke='lightgray',
    strokeWidth=0.5
).encode(
    alt.X('bin_min:Q', bin='binned', title='Maximum Daily Temperature (C)'),
    alt.Y(
        'value:Q',
        scale=alt.Scale(range=[step, -step * overlap]),
        axis=None
    ),
    alt.Fill(
        'mean_temp:Q',
        legend=None,
        #scale=alt.Scale(domain=[40, 5], scheme='redyellowblue')
    )
).facet(
    row=alt.Row(
        'Month:T',
        title=None,
        header=alt.Header(labelAngle=0, labelAlign='right', format='%B')
    )
).properties(
    title='Seattle Weather',
    bounds='flush'
).configure_facet(
    spacing=0
).configure_view(
    stroke=None
).configure_title(
    anchor='end'
)
```

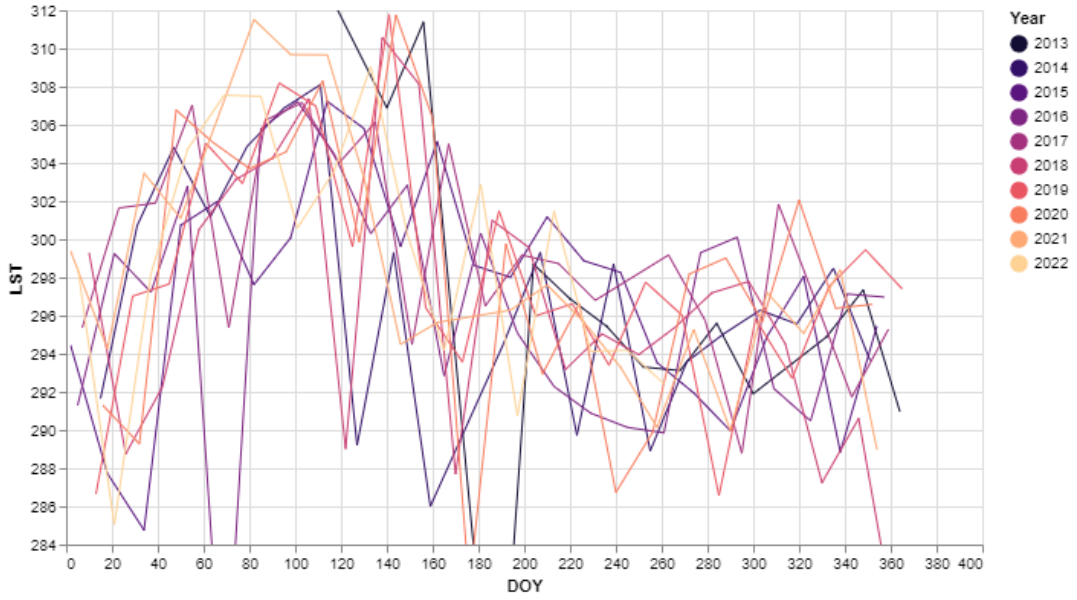
[ ]:



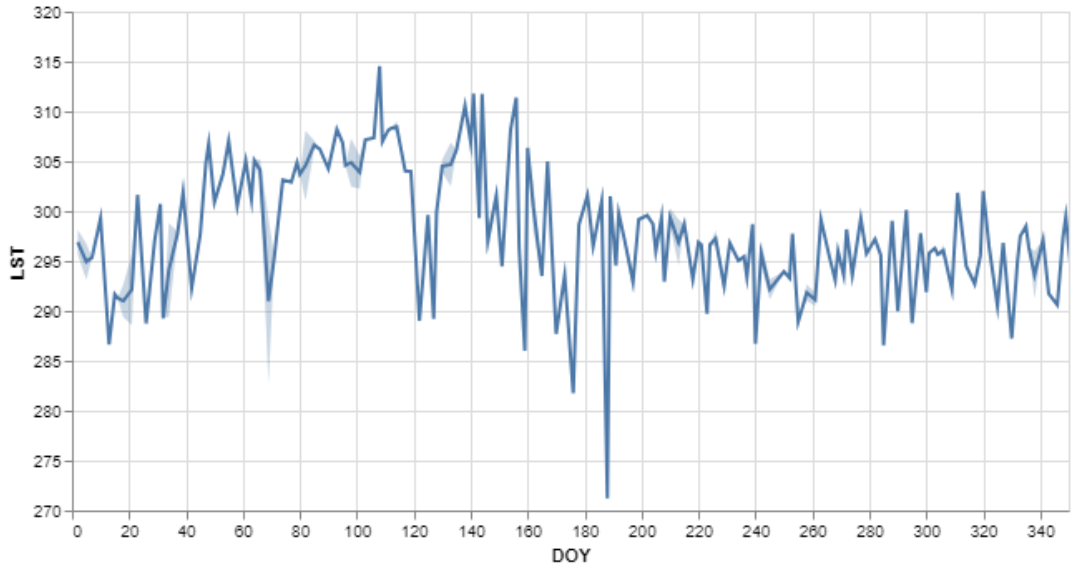
[ ]:

```
highlight = alt.selection(
    type='single', on='mouseover', fields=['Year'], nearest=True)

base = alt.Chart(lst_df).encode(
    x=alt.X('DOY:Q', scale=alt.Scale(domain=[0, 353], clamp=True)),
    y=alt.Y('LST:Q', scale=alt.Scale(domain=[285, 311])),
    color=alt.Color('Year:O', scale=alt.Scale(scheme='magma')))

points = base.mark_circle().encode(
    opacity=alt.value(0),
    tooltip=[
        alt.Tooltip('Year:O', title='Year'),
        alt.Tooltip('DOY:Q', title='DOY'),
        alt.Tooltip('LST:Q', title='LST')
    ]).add_selection(highlight)

lines = base.mark_line().encode(
    size=alt.condition(~highlight, alt.value(1), alt.value(3)))

(points + lines).properties(width=600, height=350).interactive()
```

[ ]:



[ ]:
```
base = alt.Chart(lst_df).encode(
    x=alt.X('DOY:Q', scale=alt.Scale(domain=(0, 350))))

line = base.mark_line().encode(
    y=alt.Y('median(LST):Q', scale=alt.Scale(domain=(273, 320))))

band = base.mark_errorband(extent='iqr').encode(
    y='LST:Q')

(line + band).properties(width=600, height=300).interactive()
```

[ ]:


[ ]:
```
lst80_df
```

[ ]:
```
           LST  Timestamp  Year  Month  Day  DOY
0   314.525968 2013-04-18  2013      4   18  108
1   306.886601 2013-05-20  2013      5   20  140
2   311.375564 2013-06-05  2013      6    5  156
3   294.884405 2013-11-28  2013     11   28  332
4   297.339211 2013-12-14  2013     12   14  348
..         ...        ...   ...    ...  ...  ...
79  298.178516 2022-02-06  2022      2    6   37
80  304.719601 2022-02-22  2022      2   22   53
81  307.493984 2022-03-26  2022      3   26   85
82  309.046344 2022-05-13  2022      5   13  133
83  302.860403 2022-06-30  2022      6   30  181

[84 rows x 6 columns]
```

6 Visualización de Imágenes

[ ]:
```
lstcc80
```

[ ]:
```
lc_img = lstcc80.select('LST').first()
lc_vis_params = {
    'min': 263,
    'max': 351,
    'palette': ['000004', '1B0C41', '4A0C6B', '781C6D',␣
 ↪'A52C60','CF4446','ED6925','FB9B06','F7D13D','FCFFA4']
}
```

[ ]:
```
# Create a folium map object
lat, lon = 19.332, -99.192
my_map = folium.Map(location=[lat, lon], zoom_start=10)

# Add the land cover to the map object.
my_map.add_ee_layer(lc_img, lc_vis_params, 'Land Surface Temperature')

# Add a layer control panel to the map.
my_map.add_child(folium.LayerControl())

# Display the map.
my_map
```

[ ]:
```
<folium.folium.Map at 0x7f3980f709d0>
```