

Uso de diagramas de Voronoi para la estimación del punto de vacunación más cercano a la posición georreferencial de una persona

Ricardo J. Castillo López¹, Denis O. Cuyo Ttito², Willians Montañez Choque³, Cesar A. Saire Hanco⁴

^{1,2,3,4}Departamento de Informática

^{1,2,3,4}Universidad Nacional de San Antonio Abad del Cusco, Cusco, Perú

¹101059@unsaac.edu.pe, ²182906@unsaac.edu.pe, ³131612@unsaac.edu.pe, ⁴141158@unsaac.edu.pe

Resumen: A día de hoy el mundo entero sigue siendo afectado por la pandemia ocasionada por el virus del covid 19, esto implica por supuesto al Perú y siendo más específico a la ciudad del Cusco, es por ello que nos hemos propuesto como grupo usar nuestros conocimientos para ayudar en el proceso de vacunación que se está llevando a cabo en nuestra ciudad. Para ello desarrollamos una interfaz a través de la cual una persona común y corriente pueda mediante su ubicación actual ubicar el punto de vacunación más cercano a su disposición.

Index Terms—Voronoi, triangulación, ubicación georreferencial

I. INTRODUCCIÓN

En la coyuntura actual de la pandemia causada por el Covid-19, el Estado ha impulsado una campaña nacional de vacunación. Las personas que se encuentran en el Padrón Nacional de Vacunación, pueden conocer su punto, fecha y hora de vacunación accediendo a la plataforma “Pongo el hombro”, esta plataforma consulta los datos figurados en el DNI para asignar de manera estratégica el punto de vacunación más cercano al lugar actual de residencia. El objetivo general de esta investigación es utilizar datos georreferenciales de una persona para estimar el punto de vacunación más cercano a su ubicación actual a partir de la construcción del diagrama de Voronoi con los datos georreferenciales de los puntos de vacunación. Como consecuencia del resultado de esta investigación se mejorará el proceso de asignación del punto de vacunación más cercano y la distribución de las personas que concurren a estos puntos estará mejor distribuida.

Consideramos que un servicio de este tipo es importante para ayudar a las personas que están próximas a vacunarse contra la Covid-19, de tal forma que con solo brindar su ubicación georreferencial se proporcione el punto de vacunación más cercano y así evitar que esté expuesto al contagio por mucho tiempo durante su traslado. La información proporcionada por el servicio puede ser consultada en cualquier momento y desde cualquier lugar con la garantía de que será confiable siempre que el servicio de geolocalización esté disponible y exista un dispositivo receptor y con acceso a Internet para este propósito.

Para esta investigación es necesario un análisis minucioso de los datos actuales que publica el MINSA (Ministerio de Salud del Perú) puestos a disposición del público en general, esto incluye a todos los puntos de vacunación a nivel nacional. Tomando en cuenta estos datos, y con el objetivo de acelerar el proceso de consulta de la información de los puntos de vacunación, es importante recurrir a uno de los métodos que sean eficientes dentro del campo de la geometría computacional, las estructuras conocidas como diagramas de Voronoi son útiles para el propósito de esta investigación, el

cual permitirá realizar un mapeo de los centros de vacunación en Google Maps. Se propone utilizar el algoritmo de Fortune para generar este diagrama. Luego, con esta información se diseñará un algoritmo para determinar el centro de vacunación más cercano a una ubicación específica. El lenguaje utilizado para implementar el backend y frontend del servicio es Python.

En este trabajo de investigación, el apartado de Diseño de la Arquitectura / Modelo, se explica cómo se construirá el software a partir de requisitos específicos (se necesita conocer la posición georreferencial de la persona cuando haga uso de la aplicación) y cómo tiene que interactuar con el usuario. Una vez que capturan los requisitos se define la arquitectura que se utilizará en la implementación del sistema y se especifica el modelo o patrón de diseño utilizado para la presentación de los datos, en este paso se definen los componentes que van a formar parte del diseño como por ejemplo, los componentes que son visibles al usuario y cómo interactúan con los demás componentes.

II. ESTADO DE ARTE

Los diagramas de Voronoi son ampliamente utilizados en campos como la planificación de rutas, visión computacional, etc. En particular, problemas que corresponden a la estimación del punto más cercano a otro, a partir de la división en regiones de un conjunto de puntos en un plano, de tal forma que cada punto tiene asignado una región de dominio propio, corresponden a la definición de los diagramas de Voronoi. Este tipo de problema tiene múltiples aplicaciones en diferentes campos, por ejemplo en epidemiología se utiliza para detectar el foco de infección de una enfermedad [1]. Una aplicación interesante en el contexto actual de la pandemia del Covid-19 es la estimación del punto de vacunación más cercano a la ubicación de una persona. Al respecto, todavía no existen estudios realizados en Perú, y la utilización de los diagramas de Voronoi para otros propósitos similares es escasa.

Cada nueva aplicación de los diagramas de Voronoi, requiere una variación diferente y el uso de algoritmos para su construcción que varían en complejidad. En la literatura se han propuesto estrategias eficientes como el método divide y vencerás, el algoritmo de Fortune, el concepto de dualidad a partir del grafo obtenido por el método de la triangulación de Delaunay, entre otros. Estudios más recientes han realizado aplicaciones respecto al problema de estimación del punto más cercano a otro, sobre la base de estos algoritmos donde se proponen mejoras importantes.

Elijah Smith, Christian Trefftz, Byron DeVries, A DIVIDE AND CONQUER ALGORITHM FOR COMPUTING VORONOI DIAGRAMS, Grand Valley University 2020.

El algoritmo divide y vencerás se utiliza para generar diagramas de Voronoi. Este enfoque calcula las posiciones

de las esquinas de un cuadrante sin la necesidad de calcular las posiciones de cada punto. Si las esquinas pertenecen a una misma región, no se subdivide el cuadrante, pero si no pertenecen, el cuadrante es subdividido en cuadrantes más pequeños. Este proceso se repite recursivamente hasta que se genere el diagrama total. Este algoritmo demuestra ser mejor que el enfoque Naive con una eficiencia de $O(n \log(n))$

Weizhen Wang, Chenglei Yang, Zheng Sun, *POINT LOCATION IN VORONOI DIAGRAMS OF POLYGONS*, Shandong University 2012.

Se ha tratado el problema de encontrar la región que contiene la ubicación un punto específico en el diagrama de Voronoi de un polígono, para ello se utilizó un método refinado basado en la triangulación de Delaunay y una estructura de datos que se utiliza para representar los límites de cada nodo en el diagrama y a partir de información realizar un seguimiento que permite la localización de un determinado punto en una región en particular. La complejidad temporal calculada para este algoritmo es de $O(\log(n))$ mientras que la complejidad espacial es de $O(n)$ que representa la memoria requerida para el procesamiento total.

Neil Sarnak, Robert E. Tarjan, *PLANAR POINT LOCATION USING PERSISTENT SEARCH TREES*, Communications of the ACM 1996.

Se presenta una solución al problema de la ubicación de un punto en un plano subdividido en regiones poligonales utilizando árboles de búsqueda persistentes, un tipo de árbol que difiere del árbol ordinario en que después de una operación de inserción o eliminación, todavía es posible acceder a la versión anterior del árbol. El algoritmo desarrollado para la ubicación del punto es una aplicación de este tipo de estructura de datos. Con este método la solución al problema se realiza en un tiempo de procesamiento $O(\log(n))$ y tiene una complejidad espacial $O(n)$ superior a los métodos tradicionales.

Gustavo D. Buzai, *ÁREAS DE INFLUENCIA DE LOS CENTROS DE ATENCIÓN PRIMARIA DE SALUD (CAPS) EN LA CIUDAD DE LUJÁN MEDIANTE POLÍGONOS DE VORONOI-THIESSEN*, Investigaciones y Ensayos Geográficos 2016

En este trabajo se demuestra la importancia de la construcción de áreas de influencia para hacer un estudio del alcance de los centros de salud. en este caso se realizó el estudio del alcance espacial de los Centros de Atención Primaria de salud (CAPS) que consta de 17 puntos, para los cuales se usaron los diagramas de Voronoi también llamados polígonos de Thiessen para así determinar su alcance tanto territorial como en densidad poblacional encontrándose así diferencias significativas en la densidad poblacional respecto a los centros de atención, por lo que se reafirma la importancia del uso de los diagramas de Voronoi para un óptimo alcance de este tipo de servicios.

Fernando Martinez Santa, Edwar Jacinto Gomez, Fredy Hernan Martinez Sarmiento, *USING THE DELAUNAY TRIANGULATION AND VORONOI DIAGRAMS FOR NAVIGATION IN OBSERVABLE ENVIRONMENTS*, Revista Tecnura 2014

Se utilizan diagramas de Voronoi y el método de la triangulación de Delaunay para el control de la trayectoria de un robot utilizando sensores para el estudio del terreno. En

este caso se realiza el desarrollo de un robot cuyo objetivo es llegar de un lugar a otro sin chocar con los obstáculos en el camino los cuales estarían representados por puntos que servirán para trazar los diagramas de Voronoi de manera que el robot pueda determinar un camino libre de obstáculos. Terminado el prototipo se observó la alta eficiencia de los algoritmos que utilizan el método de triangulación.

Edgar F. Arcos Hurtado, Jesus A. López, *CONTROL CO-OPERATIVE STRATEGY BASED ON IDEAL FREE DISTRIBUTION AND LOAD BALANCING FOR APPLICATIONS OF SURVEILLANCE AND MONITORING*, Prospectiva 2018

Se realizó una comparación de dos terrenos de cultivos en los cuales se repartieron diferentes robots los cuales son encargados de verificar la integridad de las plantaciones. En uno de los terrenos se repartieron de manera común y en la otra se particionó el terreno utilizando algoritmos para la construcción de diagramas de Voronoi. Luego de soltar un hongo fitopatígeno que infecto a las plantas e hizo que los robots entraran en acción, se concluyó que el terreno particionado tuvo mejores resultados en el control de la enfermedad causada por el hongo.

Y. Pérez, L. González, et al., *PROPUESTAS PARA EL ANÁLISIS GEOESPACIAL EN ESTUDIOS SALUBRISTAS*, Scielo.sld.cu, 2021

Se mencionan cuatro métodos para el análisis geoespacial para estudios salubristas, la georreferenciación, la estratificación de territorios, la regionalización y la geosimulación.

La regionalización de servicios se basa en los diagramas de Voronoi (DV) y se hace una generalización de los Diagramas de Voronoi Ponderados multiplicativos (DVMP) porque en la regionalización hay variables que inciden en el servicio. Se propone un algoritmo basado en vectores para generar DV y DVPM para la integración con sistemas de información geográficos. Con esta propuesta, se realizó una estratificación tomando como variables, las 10 principales causas de muerte en un determinado año. Para la estratificación, el algoritmo que se utiliza es el K-Means, ya que es uno de los algoritmos más utilizados y con mejores resultados en este campo.

La regionalización utilizando Diagramas de Voronoi Ponderados permite mantener y gestionar servicios desde su diseño.

F. Rezende, R. Almeida and F. Nobre, *DIAGRAMAS DE VORONOI PARA A DEFINIÇÃO DE ÁREAS DE ABRANGÊNCIA DE HOSPITAIS PÚBLICOS NO MUNICÍPIO DO RIO DE JANEIRO*, Cadernos de Saúde Pública

Este estudio se realizó con los 21 hospitales generales públicos de Río de Janeiro, los cuales se consideraron como puntos generadores para los Diagramas de Voronoi. Primero se realizó la implementación de los diagramas de Voronoi no ponderados, como resultado algunas regiones tenían áreas de influencia similares, pero las características de sus unidades de salud eran diferentes. Esto se mejoró con la implementación de diagramas de Voronoi ponderados en base a la tasa anual de ingreso.

Al aplicar diagramas de Voronoi ponderados, una ventaja es la accesibilidad a los datos, que definen el poder de atracción de una unidad. Y una desventaja es la dificultad de obtener mapas digitalizados, o cuando estos ya existen, mapas con unidades de salud igualmente georreferenciadas. Otro, son las barreras geográficas o urbanas, las redes de tráfico. Después de

la construcción de las áreas de cobertura, se puede calcular la población inscrita para cada unidad de salud. También realizar estudios que llevan a una mejor asignación y uso de recursos.

Annisa Annisa, Leni Angraeni, LOCATION SELECTION QUERY IN GOOGLE MAPS USING VORONOI-BASED SPATIAL SKYLINE (VS2) ALGORITHM, IPB Univresity 2020.

Puesto que Google Maps permite la búsqueda de una ubicación bastante popular o alguien puede buscar la ubicación más cercana, dada la circunstancias es necesario agregar más ubicaciones y dar al usuario más opciones, para esto se usó el algoritmo de Spatial Skiline (VS2) basado en los diagramas de Voronoi y el API de Google Maps para obtener información de las coordenadas deseadas. Esta investigación fue exitosa, con los datos obtenidos de los puntos de interés del API de Google Maps e implementar el algoritmo Espatial Skuline, pero como se ingresaron muchos datos más la complejidad aumentó por lo tanto se recomienda investigar otros algoritmos más óptimos y tal vez una aplicación móvil.

Ickjai Lee, Kyungmi Lee, Christopher Torpelund, RASTER VORONOI TESSELLATION AND ITS APLICATION TO EMERGENCY MODELING, James Cook University 2011.

Existen muchos sistemas informáticos que proveen mapas como Google, Google Earth o la Nasa, pero pocas tienen un sistema de alerta, debido al gran coste social, económico y ambiental, con el fin de reducir estos costos se implementa un sistema de mapas basado en diagramas de Voronoi de y Minkowski. Con este sistema será posible incluso predecir las emergencias o desastres y ayudar a tomar decisiones con un enrutamiento. El sistema propuesto de desarrollo en el lenguaje de programación Java que permite consulta de vecinos, consulta de distritos, optimización de ubicación y enrutamiento para diversos toma de decisiones. Para mayor eficacia aún se debe probar con más datos reales de diferentes circunstancias.

Marchena Pejerrey David Jesús, DESARROLLO DE UN SISTEMA PARA LA OPTIMIZACIÓN DE RUTAS DE TRABAJO UTILIZANDO EL ALGORITMO DE DIJKSTRA Y DIAGRAMA DE VORONOI, Tesis, Universidad Señor de Sipán 2015.

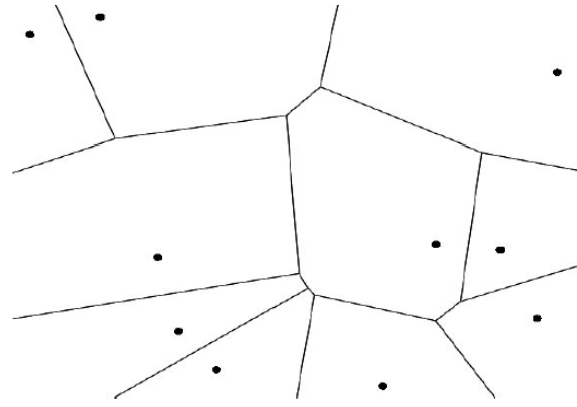
Debido al coste en las rutas no optimas se desarrolló un software utilizando el algoritmo de Dijkstra y la diagrama de Voronoi en el departamento de Micromedición de la empresa EPSEL S.A. Tomando en cuenta todo el sistema de rutas de trabajo para aplicar los algoritmos mencionados, en el desarrollo de este software se utilizó la metodología Extreme Programing e interactuar con los datos obtenidos y lograr el objetivo. Se logró optimizar el costo del proyecto y se pudo determinar de manera satisfactoria las rutas óptimas para los trabajadores para que lo operarios tomen en cuenta su trabajo geográficamente.

III. DIAGRAMAS DE VORONOI

A. INTRODUCCIÓN

1) Definición

Dado un conjunto de puntos P_1, \dots, P_n en el plano, denominaremos diagrama de Voronoi a la subdivisión del plano en subregiones tales que la región i es el conjunto de puntos más cercanos a P_i que a cualquiera de los $P_j, j \neq i$.

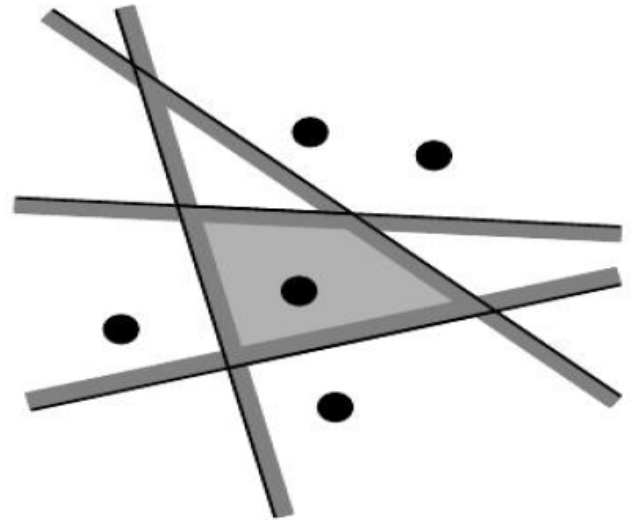


2) Propiedades

- El conjunto de puntos equidistantes de dos dados es una línea recta.
- El conjunto de puntos equidistantes de tres o más dados es un punto (si existe).
- Las componentes conexas del complementario del conjunto de puntos que equidistan de dos o más P_i son las regiones buscadas.

3) Regiones

Fijado un punto P_i , consideremos para cada $P_j, j \neq i$ la recta r_j formada por los puntos que equidistan de P_i y P_j . La recta r_j divide al plano en dos semiplanos, en uno de los cuales está P_i , al que llamaremos π_j .



La región del diagrama de Voronoi que contiene a P_i puede obtenerse como la intersección de todos los semiplanos π_j .

$$V_i = \bigcap \pi_j$$

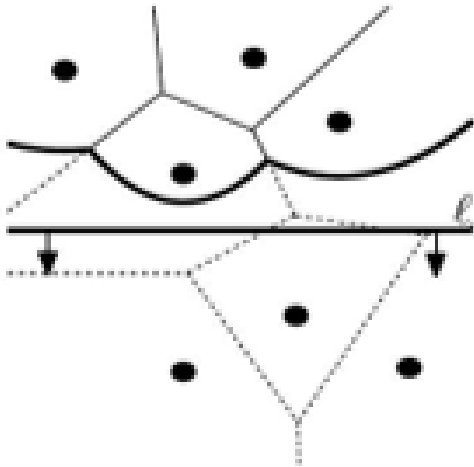
Como los semiplanos son convexos y la intersección de convexos es convexa, entonces cada región del diagrama de Voronoi es convexa.

B. ALGORITMO DE FORTUNE

1) Beach Line

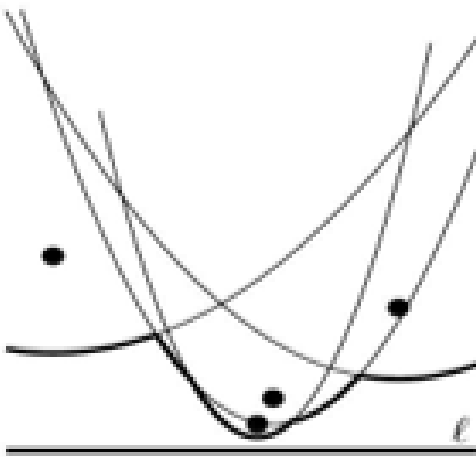
El algoritmo de Fortune es una variación de los algoritmos tipo "sweep line" aplicado al cálculo del diagrama de Voronoi. El problema al barrer el plano con una línea es que la parte del diagrama que cae por encima de la línea puede modificarse por puntos que están debajo. Para evitar este problema se

construye la “beach line”, el conjunto de puntos que equidistan de la “sweep line” y algún p_i por encima de la “sweep line”.

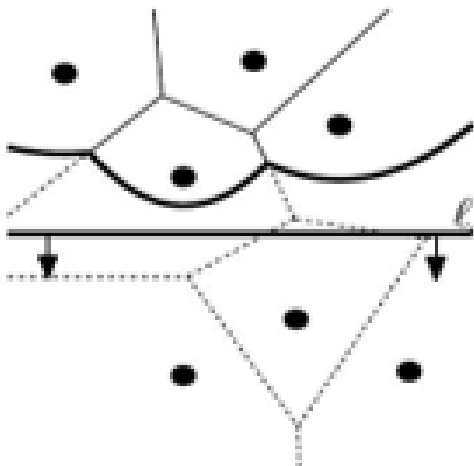


2) Estructura de Beach Line

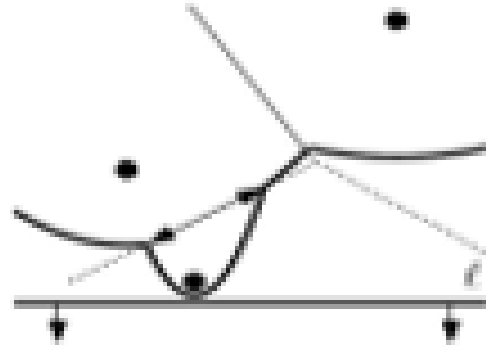
Para cada punto p_i por encima de la “sweep line”, el conjunto de puntos que equidistan de p_i y de la “sweep line” es una parábola. Si construimos todas las parábolas correspondientes a puntos p_i por encima de la “sweep line” y tomamos el ínfimo de la coordenada y , obtendremos la “beach line”. Por tanto, estará formada por arcos de parábola.



Cada dos arcos de parábola se cortan en un punto que equidista de dos p_i (y de la “sweep line”), luego será parte de una arista del diagrama de Voronoi.

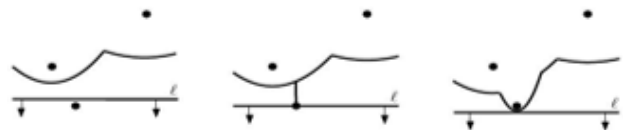


Los vértices del diagrama se corresponden con puntos donde coinciden las distancias de tres puntos. Cuando la “beach line” llega a un punto de este tipo, lo que ocurre es que un arco de parábola desaparece.



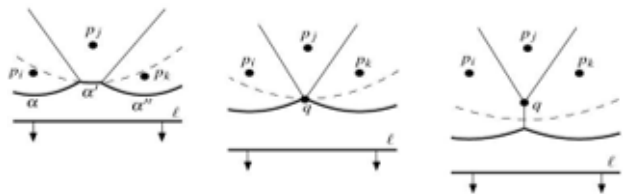
3) Eventos

Igual que en el algoritmo de intersección de segmentos, no guardaremos la “beach line” sino que sólo miraremos cuando cambia su topología (cuándo aparece o desaparece un arco de parábola). A los puntos en los que cambia la topología los denominaremos eventos. Los eventos se guardarán en una lista de eventos, ordenada por la coordenada y de la posición de la “sweep line” que hace que ocurra el evento. Existen únicamente dos tipos de eventos: la aparición de un nuevo arco o la desaparición de un arco. Aparición de arcos Un nuevo arco únicamente puede aparecer cuando un nuevo vértice toca la “sweep line”. Cada arco lo marcaremos con el índice del vértice que lo ha generado.



4) Desaparición de arcos

Como vimos anteriormente, un arco sólo desaparece en los vértices del diagrama de Voronoi. Es decir, en un punto q que equidista de tres p_i cuyos arcos son consecutivos en la “beach line”.



5) Descripción global

El esquema del algoritmo de Fortune es similar al de los algoritmos de “sweep line”:

- Generamos una lista con todos los eventos que se conocen en el instante inicial. En este caso todos los puntos p_i Se ordenan según la ordenada y .
- Se genera una estructura para guardar la “beach line”. En este caso generaremos una matriz, aunque la estructura más eficiente es un árbol de búsqueda binario.
- Se elige el primer evento de la lista. Se elimina de la lista y se procesa.
- Repetimos el paso anterior hasta que no queden eventos.

6) Manejo de eventos

Para terminar el algoritmo anterior sólo nos falta decidir cómo se procesa cada uno de los eventos. Evento de creación: Se coloca un nuevo arco en la “beach line”. Para ello miramos qué arco cae en la vertical del punto que provocó el evento. Se marca el arco con el índice del punto que lo crea. Se miran las dos nuevas ternas que se han generado en la “beach line” para ver si se produce algún evento circular (es decir, se calcula el centro de la circunferencia que pasa por los tres puntos). En dicho caso se almacena el evento. Para los eventos circulares, hay que almacenar a qué altura de la “sweep line” se producirán. La altura es la coordenada y del centro de la circunferencia menos su radio. Evento circular. Se elimina el arco correspondiente. Se comprueba si las nuevas ternas de puntos de la “beach line” producen algún evento circular. Se elimina cualquier evento circular que estuviese en alguna de las aristas que se han cortado (y en la dirección adecuada). Se añade el punto de intersección al diagrama de Voronoi.

7) Complejidad

El número de eventos es proporcional al número de puntos, n , (se puede probar que el número de intersecciones del diagrama es proporcional a n). En cada evento el coste de procesarlo (si se emplea un árbol de búsqueda binario) es $\log(n)$, dando un tiempo total de $n\log(n)$.

8) GeoVoronoi

geovoronoi ayuda a generar regiones de Voronoi para datos geográficos, por ejemplo, coordenadas de universidades públicas en un país determinado. Esto, a su vez, puede usarse para estimar algún tipo de “cobertura”. Sin embargo, el uso no se limita a los datos geográficos. Este paquete le permite generar regiones Voronoi finitas dentro de cualquier forma poligonal circundante válida.

La función principal de este paquete, `voronoi_regions_from_coords()` toma una lista de coordenadas y calcula las regiones de Voronoi a partir de ellas usando SciPy. En los bordes, estas regiones van al infinito. A continuación, podemos tomar la forma del área circundante (por ejemplo, la forma de un país como polígono) para cortar las regiones de Voronoi para que encajen en la forma proporcionada, haciendo que las regiones en los bordes sean finitas. geovoronoi utiliza bien proporcionado para estas operaciones. Además, el paquete implementa algunas funciones para trazar fácilmente las regiones de Voronoi resultantes.

a) Uso: Tiene un área geográfica que contiene algunos puntos para los que desea generar regiones de Voronoi. Esta área geográfica es un shapely Polygon/MultiPolygon object (que, por ejemplo, obtuvo de un archivo GeoJSON que cargó con GeoPandas o Fiona). Los N puntos que tiene están en forma de una matriz $N \times 2$ NumPy, o una lista de objetos Point bien formados (se pueden convertir con las funciones `coords_to_points` y `points_to_coords`). Tanto los puntos como el área geográfica circundante deben estar en el mismo CRS (coordinate reference system). Se muestra un ejemplo con algunas coordenadas de Italia.

```
array([[1690891.43454513, 4865911.53550427],
       [1303898.2749075 , 5398659.4816214 ],
       [1379407.32051822, 5701267.51923313],
       [1703402.05850744, 4916559.63783754],
       ...
       ])
```

41 random points and their Voronoi regions in Italy



IV. LOCALIZACIÓN DE UN PUNTO EN UN PLANO

A. Método de Kirkpatrick

El método de Kirkpatrick para la localización de un punto es un método basado en la jerarquía de triangulaciones. El polígono inicial define el nivel más bajo de la jerarquía y los niveles superiores se construyen como triangulaciones aproximadas del siguiente nivel inferior. Una precondition es que el nivel mínimo tiene que ser una triangulación (envolvente convexa) para un conjunto de puntos N , esta triangulación tendrá a lo mucho $3|V| - 6$ aristas, donde V es el conjunto de vértices. La triangulación inicial es inscrita en un triángulo delimitador, de manera que el nivel mínimo consta solo de este triángulo. Un tiempo de ejecución $O(\log(N))$ sólo está garantizado con esta condición previa.

Sea la secuencia de triangulaciones S_1, S_2, \dots, S_n donde S_1 es la triangulación inicial de vértices. El método para obtener la triangulación S_i a partir de S_{i-1} se describe a continuación:

- 1) Un conjunto de vértices no adyacentes y no limítrofes se eliminan de S_{i-1}
- 2) Los polígonos que surgen de la eliminación de los vértices se vuelven a triangular y los nuevos triángulos se añaden a S_i
- 3) Se hace una referencia a partir de un triángulo recién creado en S_i hacia uno eliminado en S_{i-1} si ambos se cruzan.
- 4) i se incrementa y el algoritmo de repite desde el paso 1 hasta que quede un único triángulo, es decir se llega a la triangulación de S_n

La secuencia de triangulaciones se almacena en una estructura de árbol denominada grafo de búsqueda acíclico dirigido

(Search-Directed Acyclic Graph - SDAG) cuyos nodos son triángulos. La localización de un punto en esta estructura procede de acuerdo al siguiente algoritmo: Sea P el punto del cual se desea conocer su ubicación. Inicialmente se verifica si está incluido en la triangulación S_n . La prueba de la inclusión del punto en un triángulo se puede realizar en un tiempo $O(1)$. Si el punto está incluido en el triángulo de la raíz, entonces se verifica su inclusión en uno de sus nodos hijo. Este proceso se realiza hasta llegar al triángulo que pertenece.

Algoritmo:

Si P no está incluido en el triángulo de la raíz S_n , entonces:

Retornar P , es decir, no está incluido en el polígono inicial.

Caso contrario:

$\text{padre} := \text{triángulo de la raíz}$

Mientras el nodo padre tiene hijos hacer:

Para cada i –ésimo hijo del padre hacer:

Si P está incluido en el i –ésimo hijo entonces:

$\text{padre} = i$ –ésimo hijo

Terminar bucle para

Retornar padre

B. Retriangulación

Los polígonos que surgen de la eliminación de un vértice tienen una propiedad especial: tienen la forma de una estrella. El algoritmo utilizado para volver a triangular estos polígonos se describe a continuación: Sea p un punto arbitrario del polígono P con N vértices:

- 1) A partir de p buscar dos aristas consecutivas que forman un ángulo menor que $180 - 360/N$ sustituyendo p con el siguiente punto del polígono.
- 2) Generar el triángulo $\langle p, p+1, p+2 \rangle$ y disminuir N en 1.
- 3) Repetir desde el paso 1 para $p = p+2$ hasta que N sea 3.

C. Eficiencia

La eficiencia del método de localización de puntos de Kirkpatrick depende en gran medida de la elección del conjunto de puntos que se eliminarán durante el paso de construcción de la jerarquía. Sea P un polígono en forma de estrella con K vértices, cuyo punto central ha sido eliminado. Antes de eliminar este punto el polígono está conformado por K triángulos y el grado del punto central es también K . Después de eliminar este punto se crean $K-2$ triángulos en el proceso de retriangulación del polígono.

La elección de la constante K es fundamental para el rendimiento del algoritmo de Kirkpatrick, su valor no puede ser menor que 3, porque el grado mínimo de cualquier punto interior en una triangulación es 3, sin embargo, no hay límite superior para este valor. Una adecuada elección de K hace que se seleccionen la mayoría de triángulos durante el paso de construcción de la jerarquía y el rendimiento incrementa. Una elección inadecuada hace que existan triángulos que se conserven para la siguiente jerarquía y en consecuencia disminuye el rendimiento del algoritmo de localización del punto. En aplicaciones típicas se ha utilizado un valor para K en el rango de 6 a 8, lo cual ha conducido a buenos resultados.

V. DISEÑO DE LA ARQUITECTURA/MODELO

A. Patrón de diseño

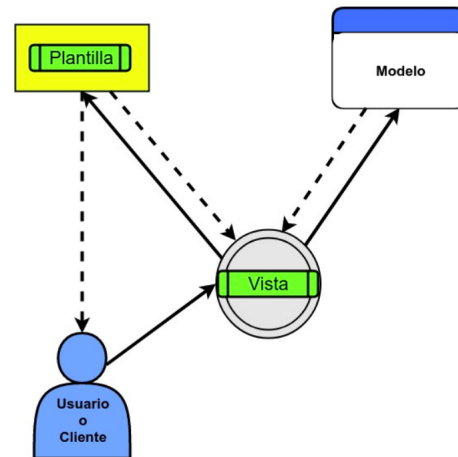
Para el desarrollo de esta aplicación se utiliza Django, que es un framework de desarrollo web.

El patrón de que se usa Django es el de Modelo-Vista-Template. MVT (Model View Template) es un patrón de diseño de software. Es una colección de tres componentes importantes: Vista de modelo y Plantilla.

M significa "Model" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.

T significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.

V significa "View" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelos y las plantillas.



B. Diseño de la lógica de negocio

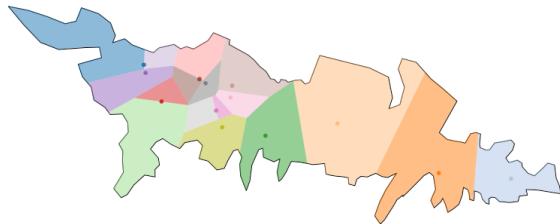


para empezar a analizar los datos con el diagrama de Voronoi, es necesario importa en formato CSV los puntos georreferenciales de los puntos de vacunación de determinada localidad.

como el proyecto se está desarrollando en Python, es aplicará algunas librerías para el tratamiento de estos datos, utilizando la librería “pandas” importaremos los datos del CSV, luego con “geopandas” importamos el mapa de dicha localidad (tipo-polígono) en SHP y convertirlo en un polígono que pueda entender “geovoronoi”



una vez hecha la importación del SHP y ubicado los puntos (coordenadas) aplicamos Voronoi y obtenemos el diagrama.

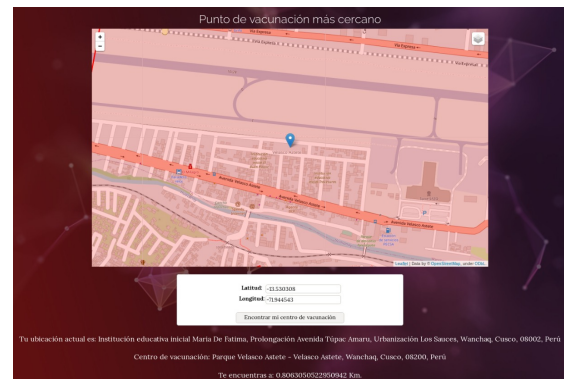
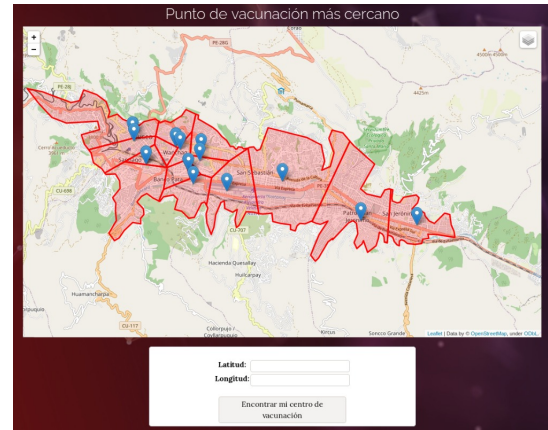


como proceso final, exportamos en geoJson y construir el interfaz con Django, otra librería de Python.

C. Interfaz de Usuario: Prototipo



D. Interfaz de Usuario: Implementación



VI. CONCLUSIONES

Se ha desarrollado un interfaz gráfica para estimar el punto de vacunación más cercano a la posición georreferencial de una persona (latitud y longitud), para ello fue necesario distintas herramientas para análisis de datos georreferenciales como: Google Earth y ARGIS Map, generar los mapas y luego exportarlo en un formato ideal para Python. El aplicativo web se ha desarrollado utilizando el framework Django de Python y librerías correspondientes que implementan algoritmos para generar el diagrama de Voronoi (Pandas, GeoPandas y Geovoronoi) sobre el mapa geográfico de la ciudad del Cusco y para estimar el área poligonal al cual pertenece un punto específico, a partir de esta información se obtiene los datos del centro de vacunación más cercano al usuario.

Se concluye entonces que el desarrollo del proyecto fue satisfactorio, se logró el objetivo principal de la investigación. Puede ser mejorado para ser aplicado en cualquier ciudad del mundo para resolver problemas similares.

REFERENCES

- [1] C. Expósito, "Los diagramas de Voronoi, la forma matemática de dividir al mundo". [Online]. Available: <https://mediateca.educa.madrid.org>. [Accessed: 10- Jul- 2021].
- [2] E. Smith, C. Trefftz, et al. "A Divide-and-Conquer Algorithm for Computing Voronoi Diagrams." *IEEE International Conference on Electro Information Technology: IEEE International Conference on Electro Information Technology* vol. 2020 (2020): 10.1109/eit48999.2020.9208270. doi:10.1109/eit48999.2020.9208270
- [3] W. Wang, C. Yang, et al. "Point location in Voronoi diagrams of polygons," 2012 *International Symposium on Information Technologies in Medicine and Education*, 2012, pp. 469-472, doi: 10.1109/ITIME.2012.6291344.
- [4] N. Sarnak, R. E. Tarjan. "Planar point location using persistent search trees." *Commun. ACM* 29, 7 (July 1986), 669–679. DOI: <https://doi.org/10.1145/6138.6151>
- [5] E. F. Arcos Hurtado, J. A. López, "Control cooperative strategy based on ideal free distribution and load balancing for applications of surveillance and monitoring", *Prospectiva*, Vol XVI, N° 2, 59-67, 2018.
- [6] G. D. Buzai, "Áreas de influencia de los centros de atención primaria de salud (CAPS) en la ciudad de Luján mediante polígonos de Voronoi-Thiessen", *Investigaciones y Ensayos Geográficos*, XIII, 13, 12-26, 11-2016.
- [7] F. Martinez Santa, E. Jacinto Gomez, F. Martinez, "Using the Delaunay Triangulation and Voronoi diagrams for navigation in observable environments", *Revista Tecnura*, Edición especial, 81-87, 05-2014.
- [8] D. J. Marchena, "Desarrollo de un sistema para la optimización de rutas de trabajo utilizando el algoritmo de Dijkstra y diagramas de Voronoi". Tesis, Facultad de Ingeniería, Arquitectura y Urbanismo, USS, Pimentel, Perú, 2015.
- [9] L. A. Annisa Annisa, "Location Selection Query in Google Maps using Voronoi-based Spatial Skyline (VS2) Algorithm", *Jurnal Online Informatika*, vol. VI, n° 1, pp. 25-32, 2020.
- [10] K. L. C. T-B. Ickjai Lee, "Raster Voronoi Tessellation and Its Application to Emergency Modeling", *Geo-spatial Information Science*, vol. XIV, n° 4, pp. 235-245, 2011.
- [11] Y. Pérez Betancourt, L. González Polanco, J. Febles Rodríguez and A. Cabrera Campos, "Propuestas para el análisis geoespacial en estudios salubristas.", *Scielo.sld.cu*, 2021. [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S2227-18992018000200004&lng=es&nrm=iso&tlng=en. [Accessed: 09- Jul- 2021].
- [12] F. Rezende, R. Almeida and F. Nobre, "Diagramas de Voronoi para a definição de áreas de abrangência de hospitais públicos no Município do Rio de Janeiro", *Cadernos de Saúde Pública*, vol. XVI, n° 2, pp. 467-475, 2000. Available: https://www.researchgate.net/publication/26359385_Diagramas_de_Voronoi_para_a_definicao_de_areas_de_abrangencia_de_hospitais_publicos_no_Municipio_do_Rio_de_Janeiro. [Accessed 10 July 2021].