

INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO



Computational Geometry

Proyecto

"Obtención de rutas a partir de un diagrama de
Voronoi"

Profesor: Rosaura Palma Orozco

Alumno: Santiago Ruiz Javier Guadalupe

Grupo: 3CM16

Fecha de entrega: 9 de enero de 2023

Índice

Introducción.....	4
Diagramas de Voronoi	4
Propiedades básicas de un diagrama de Voronoi	4
Grafos.....	5
Algoritmo de Dijkstra.....	6
Desarrollo.....	7
Lenguaje de programación y plataforma utilizada	7
Descripción general de la aplicación	7
Pruebas y resultados.....	11
Zona G.....	11
Zona B.....	12
Conclusiones.....	13
Referencias bibliográficas	13

Índice de ilustraciones

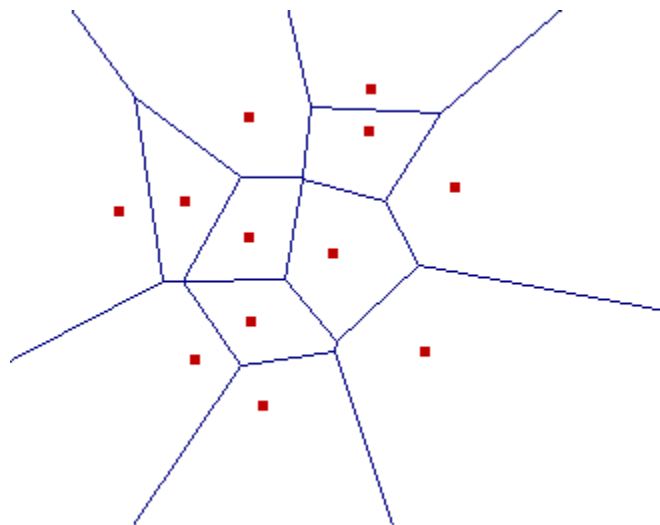
Ilustración 1: Voronoi.java	7
Ilustración 2: Mapa.java	8
Ilustración 3: Es_Aqui.java	8
Ilustración 4: D.java (ruta 1)	9
Ilustración 5: D.java (ruta 2)	10
Ilustración 6: Menu.java	10
Ilustración 7: G.java (ruta 1)	11
Ilustración 8: G.java (ruta 2)	11
Ilustración 9: B.java (ruta 1)	12
Ilustración 10: B.java (ruta 2)	12

Introducción

Diagramas de Voronoi

Los diagramas de Voronoi son una de las estructuras fundamentales dentro de la Geometría Computacional, de alguna forma ellos almacenan toda la información referente a la proximidad entre puntos. Son numerosísimas sus aplicaciones.

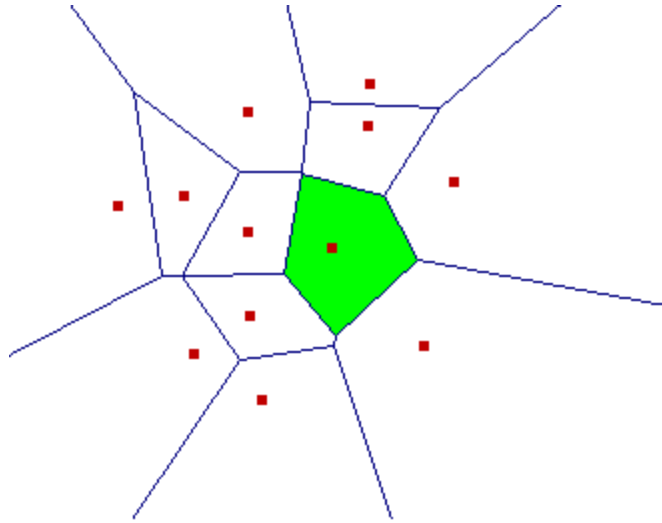
La idea del diagrama de Voronoi se basa fundamentalmente en la proximidad. Suponemos dado un conjunto finito de puntos en el plano $P = \{p_1, \dots, p_n\}$ (con n mayor o igual que dos) y a cada p_j le asociamos aquellos puntos del plano que están más cerca o igual suya que de cualquier otro de los p_i con i distinto de j . Todo punto del plano queda así asociado a algún p_i , formándose conjuntos que recubren a éste. Existirán puntos que disten lo mismo de dos elementos de P y que formarán la frontera de cada región. Los conjuntos resultantes forman una teselación del plano, en el sentido de que son exhaustivos (todo punto del plano pertenece a alguno de ellos) y mutuamente excluyentes salvo en su frontera. Llamamos a esta teselación Diagrama de Voronoi plano (denotado $\text{Vor}(P)$). A cada una de las regiones resultantes las llamaremos regiones de Voronoi o polígonos de Voronoi (denotado $\text{Vor}(p_i)$). Los puntos del conjunto reciben el nombre de generadores del diagrama.



Propiedades básicas de un diagrama de Voronoi

Lema 1: La intersección de los semiplanos $h(p_1, p_j)$ es $\text{Vor}(p_1)$.

A partir del Lema 1 podemos deducir que las regiones de Voronoi serán conjuntos convexos al ser intersección de convexos (semiplanos). En la figura se ve el diagrama de Voronoi que ya representamos anteriormente y una de sus regiones que se observa que es convexo.



Proposición 1: Una región de Voronoi es no acotada si y sólo si su generador se encuentra en la frontera de la envolvente convexa.

Proposición 2:

- Los bordes de una región de Voronoi son rectas infinitas si y sólo si todos los puntos de P descansan sobre una misma recta.
- El borde de Voronoi entre dos generadores es una semirrecta si y sólo si P no es colineal y los generadores son consecutivos en la frontera de la envolvente convexa de P .
- El borde de Voronoi entre dos generadores es segmento de recta finito si y sólo si P es no colineal y al menos uno de los dos generadores está en el interior de la envolvente convexa de P .

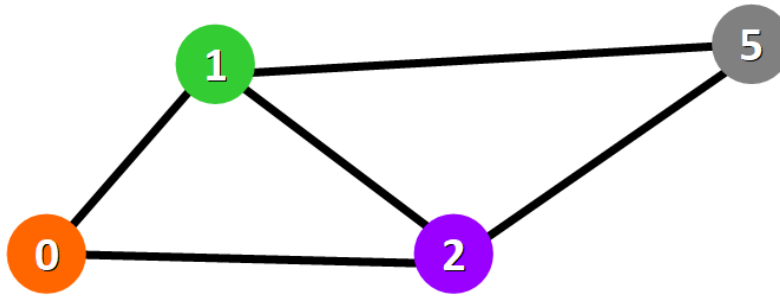
Proposición 3: Dado un diagrama de Voronoi $\text{Vor}(P)$ generado por un conjunto de puntos P en el plano, se cumple:

- Un punto q es vértice de $\text{Vor}(P)$ si y sólo si el círculo máximo vacío centrado en q contiene tres o (en el caso de tratarse de un diagrama degenerado) más generadores en su frontera.
- La bisectriz entre dos generadores define un borde de $\text{Vor}(P)$ si y sólo si existe un punto q sobre dicha bisectriz tal que el círculo máximo vacío centrado en q contiene solamente a estos dos generadores en su frontera.

Grafos

Los grafos son estructuras de datos usadas para representar "conexiones" entre pares de elementos.

- Estos elementos se llaman nodos. Representan objetos reales, personas o entidades.
- Las conexiones entre los nodos se llaman aristas o arcos.

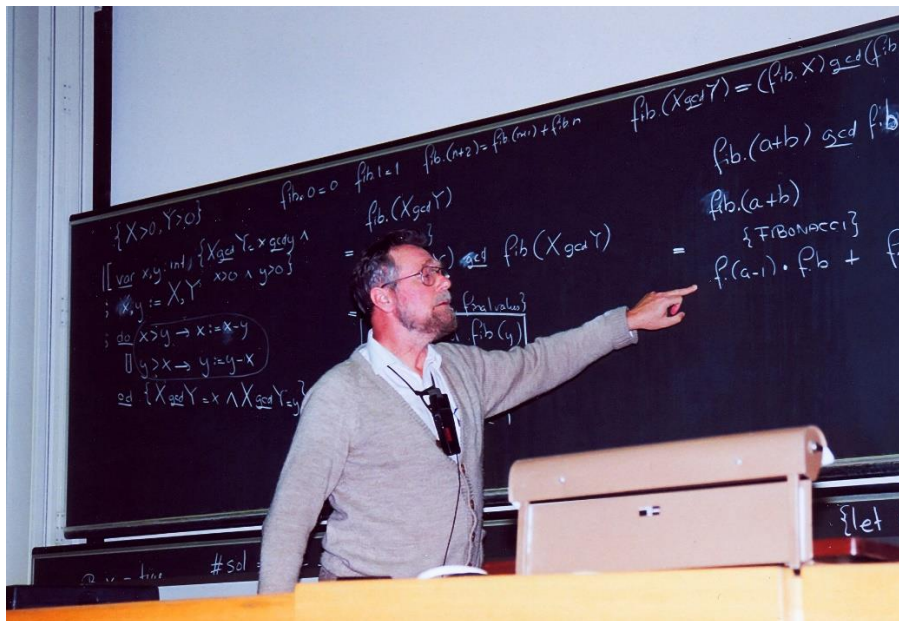


Algoritmo de Dijkstra

Con el algoritmo de Dijkstra, puedes encontrar la ruta más corta o el camino más corto entre los nodos de un grafo. Específicamente, puedes encontrar el camino más corto desde un nodo (llamado el nodo de origen) a todos los otros nodos del grafo, generando un árbol del camino más corto.

Este algoritmo es usado por los dispositivos GPS para encontrar el camino más corto entre la ubicación actual y el destino del usuario. Tiene amplias aplicaciones en la industria, especialmente en aquellas áreas que requieren modelar redes.

Fue creado y publicado por el Dr. Edsger W. Dijkstra,



Desarrollo

Lenguaje de programación y plataforma utilizada

Para la realización de este proyecto se decidió utilizar el lenguaje de programación **Java** y el entorno de desarrollo **NetBeans 14**.

Descripción general de la aplicación

La aplicación presenta el mapa una escuela al cual se la ha obtenido su diagrama de Voronoi. Cada sitio de Voronoi de cada zona es considerado una cafetería (iconos rosa) a la cual se quiere llegar. Para cumplir este objetivo es necesario pasar por diversas ubicaciones (iconos azules), la aplicación le indica al usuario cuales son las ubicaciones que conforman la ruta mas corta para llegar a la cafetería.

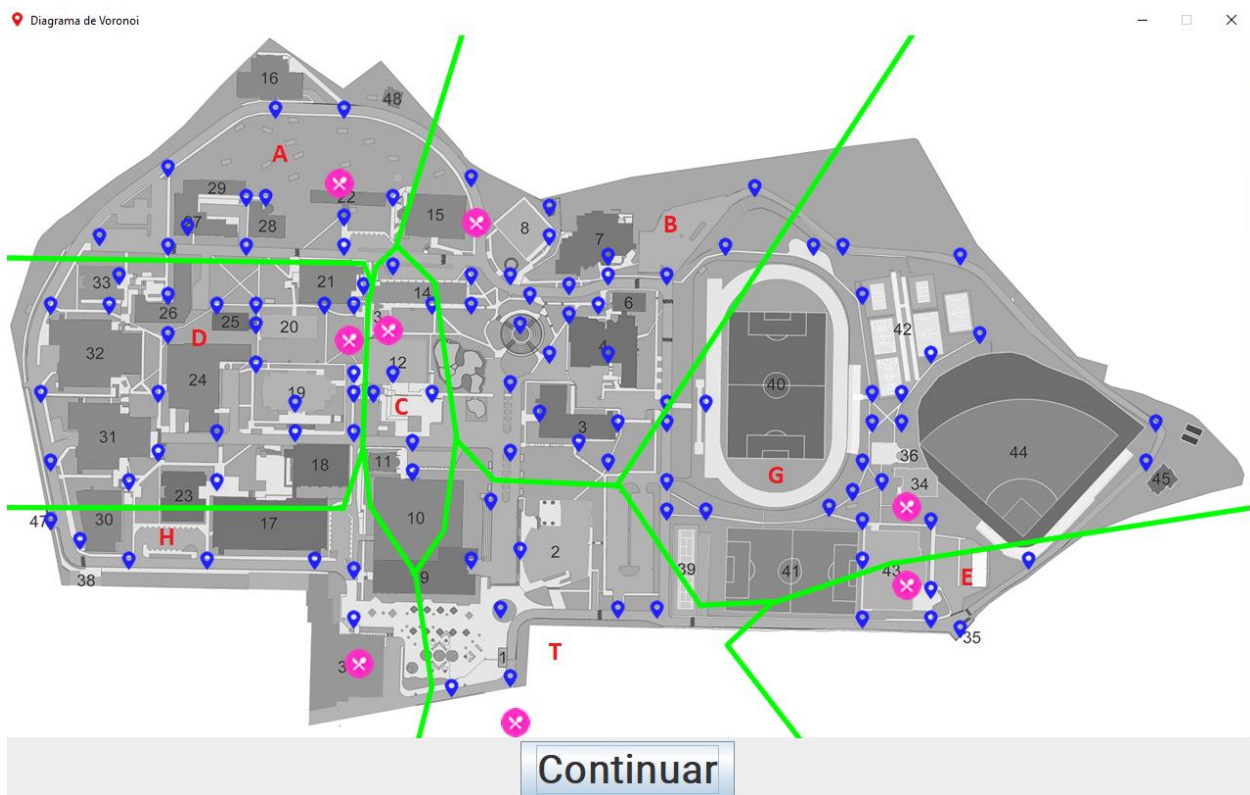


Ilustración 1: Voronoi.java

Una vez pulsado el botón “Continuar” el usuario debe seleccionar en qué lugar se encuentra (iconos negros).



Ilustración 2: Mapa.java

A partir de aquí pueden ocurrir tres cosas:

- 1) Que el usuario desee regresar a ver el diagrama de Voronoi pulsando el botón “Regresar”.
- 2) Dar clic en una cafetería (iconos rosas), esto desplegará el siguiente mensaje:

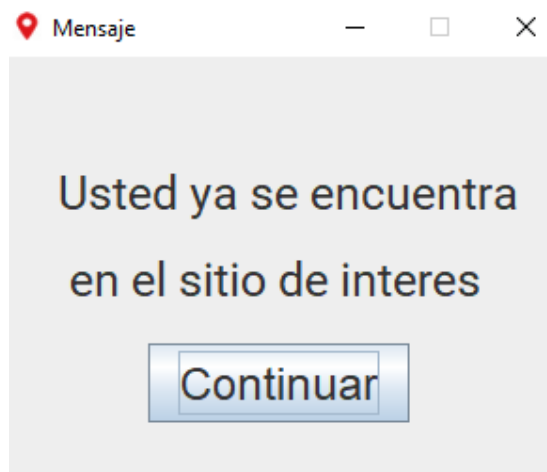


Ilustración 3: Es_Aqui.java

- 3) Seleccionar correctamente una ubicación, esto desplegará una pantalla con la zona de voronoi a la cual pertenece esa ubicación, además de mostrar la ruta más corta que debe seguir el usuario para llegar a la cafetería iluminada de color rojo.



Ilustración 4: D.java (ruta 1)

Una vez posicionado en la ventana de la zona de Voronoi es posible dar clic a otra zona y volver a obtener la nueva ruta más corta.



Ilustración 5: D.java (ruta 2)

Finalmente el usuario puede repetir el proceso dando clic en el botón “Salir” el cual lo llevara nuevamente a la pantalla inicial.

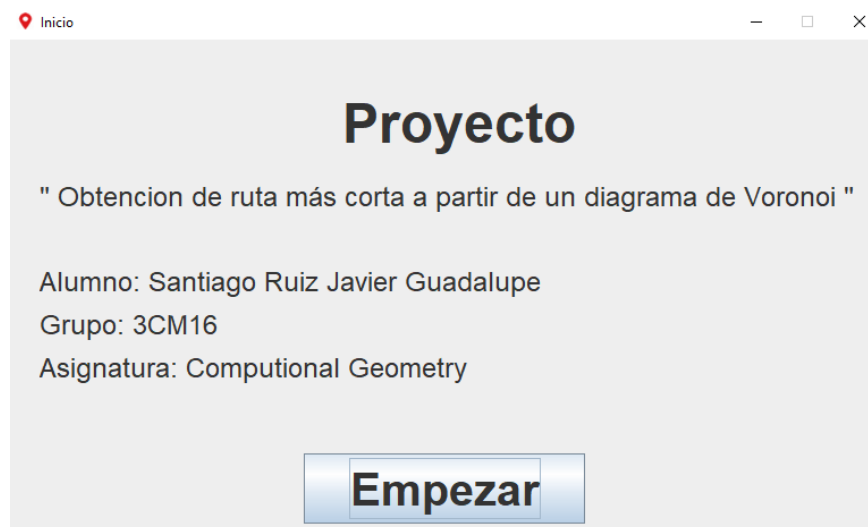


Ilustración 6: Menu.java

Pruebas y resultados

La ubicaciones iniciales seleccionadas están sombreadas de amarillo.

Zona G



Ilustración 7: G.java (ruta 1)



Ilustración 8: G.java (ruta 2)

Zona B



Ilustración 9: B.java (ruta 1)



Ilustración 10: B.java (ruta 2)

Conclusiones

La idea principal del proyecto se logro concretar al 100%. En algunas ocasiones los botones para continuar entre las pantallas y los de selección de ubicación parecen no funcionar pero basta con volver a dar clic.

Me pareció muy interesante las múltiples aplicaciones que tienen los diagramas de Voronoi y como estos nos pueden proporcionar diversa información dependiendo el contexto en donde se apliquen.

Referencias bibliográficas

- [Geometría Computacional, 2019] Anónimo. Tema 3: Diagramas de Voronoi
<http://asignatura.us.es/fgcitig/contenidos/gctem3ma.htm>
- [FreeCodeCamp, 2022] Estefanía Cassingena Navone. Algoritmo de la ruta más corta de Dijkstra - Introducción gráfica y detallada
<https://www.freecodecamp.org/espanol/news/algoritmo-de-la-ruta-mas-corta-de-dijkstra-introduccion-grafica/>