

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores

(Licentiate Degree Program in Computer Engineering)



Herramienta en ambiente web para la visualización de las corrientes de viento en Costa Rica

(Web environment tool to display Costa Rica's wind flow)

Informe de Trabajo de Graduación para optar por el título de Ingeniero en Computadores con grado académico de Licenciatura

(Report of Graduation Work in fulfillment of the requirements for the degree of Licentiate in Computer Engineering)

Johnny Arturo Romero Lépiz

Cartago, noviembre, 2015

(Cartago, November, 2015)



TEC – Área Ingeniería en Computadores (CE) Acta de Aprobación de Trabajo de Graduación

Con fundamento en lo que establece el "Reglamento de Trabajos Finales de Graduación del Instituto Tecnológico de Costa Rica", el Tribunal Examinador del Trabajo Final de Graduación, nombrado con el propósito de evaluar el proyecto final de graduación.

"Herramienta en ambiente web para la visualización de las corrientes de viento en Costa Rica"


Habiendo analizado el resultado general del trabajo presentado por los estudiantes:


Primer Apellido	Segundo Apellido	Nombre	No. De carné
Romero	Lépiz	Johnny Arturo	200927316


Emite el siguiente dictamen:

<p>APROBADO</p> <p>CALIFICACION: <u>95</u> puntos.</p>	<p><input type="radio"/> REPROBADO</p> <p><input checked="" type="radio"/> SE RECOMIENDA <input type="radio"/> NO SE RECOMIENDA</p> <p>Brindarle una nueva oportunidad para la DEFENSA PUBLICA de su Trabajo Final</p> <p>NUEVA FECHA: _____</p>
--	---

Dando fe de lo aquí expuesto firmamos


Ing. Eduardo A. Canessa Montero, M.Sc.,
Profesor Asesor


Dr. Roberto Pereira Arroyo
Profesor Lector


Máster Isaac Ramírez Herrera
Profesor Lector

19 de Noviembre del 2015

Resumen

El objetivo del presente trabajo es desarrollar una aplicación para la visualización de las corrientes de viento en Costa Rica. Se utilizaron los datos de dirección y rapidez del viento del año 2014, brindados por el Instituto Meteorológico Nacional, los cuales fueron recogidos por veinte estaciones meteorológicas repartidas a lo largo del país. Se desarrolló un algoritmo de interpolación de distancia inversa ponderada, con el fin de crear un campo vectorial que representa las corrientes de viento del país, este fue basado en los datos de las veinte estaciones del Instituto Meteorológico Nacional. Finalmente se creó una aplicación web utilizando HTML, CSS, JavaScript y la biblioteca D3.js, en la cual se realizó la animación del campo vectorial utilizando líneas que se mueven según la dirección y rapidez del viento.

Palabras clave: Estaciones meteorológicas, Interpolación de distancia inversa ponderada, D3.js, Energía Eólica, Zonas Climáticas de Costa Rica, OpenGL, Parallel.js, Sistema de Información Geográfica.

Abstract

The objective of this work is to develop an application to show the Costa Rica's vector wind field. Data of wind speed and direction of the year 2014 were used, this information was provided by the Instituto Meteorológico Nacional and collected by twenty weather stations spread around the country. An interpolation algorithm of inverse distance weighted was developed in order to create a vector field as a representation of the country's wind flow; the data of the stations of the Instituto Meteorológico Nacional were used as initial data and for other sectors of the country, where no stations are found, the interpolation algorithm was used. Finally a web application using HTML, CSS, JavaScript and the D3.js library was developed to show the vector wind field animation, it was performed using lines in motion with the direction and speed of the wind.

Keywords: Weather Stations; Inverse distance weighted interpolation; D3.js; Wind power; Climatic zones of Costa Rica; OpenGL; Parallel.js; Geographic Information System.

ÍNDICE GENERAL

Resumen.....	3
Abstract	4
ÍNDICE GENERAL	5
ÍNDICE DE TABLAS	8
ÍNDICE DE FIGURAS.....	9
1 Introducción	11
1.1 Descripción general del proyecto	11
1.2 Antecedentes	11
1.2.1 Descripción de la empresa	11
1.2.2 Trabajos similares	15
1.3 Planteamiento del problema	18
1.3.1 Contexto del problema.....	18
1.3.2 Situación problemática.....	19
1.3.3 Definición del problema.....	19
1.4 Objetivos del proyecto.....	20
1.4.1 Objetivo general.....	20
1.4.2 Objetivos específicos	20
1.5 Alcances, entregables y limitaciones del proyecto.....	20
1.5.1 Descripción de entregables	21
2 Marco teórico	22
2.1 Interpolación.....	22
2.1.1 Interpolación polinómica	22
2.1.2 Interpolación meteorológica	23
2.2 Sistema de coordenadas cartesianas bidimensionales	25
2.2.1 Campos vectoriales	26
2.2.2 Campo de velocidades	28
2.3 Energía eólica.....	29
2.3.1 Variación de potencia en una turbina de viento	30
2.4 Zonas climáticas de Costa Rica.....	31
2.4.1 Regiones geográfico-climáticas	32
2.5 Desarrollo de software.....	33
2.5.1 Ciclo de vida de un <i>software</i>	33
2.5.2 Modelos de desarrollo de <i>software</i>	33
2.6 Herramientas de Software	34
2.6.1 OpenGL.....	34

2.6.2	OpenSceneGraph	35
2.6.3	OsgEarth.....	35
2.6.4	WebGL.....	35
2.6.5	Data-Driven Documents	36
2.6.6	Web workers	36
2.6.7	Parallel.js.....	36
2.6.8	Semantic UI.....	37
2.6.9	GRASS GIS	37
2.6.10	QGIS	37
3	Desarrollo metodológico.....	38
3.1	Desarrollo de software.....	38
3.2	Interpolación.....	39
3.2.1	Distancia inversa ponderada	39
3.3	Datos iniciales	41
3.4	Parallel.js	43
3.5	D3.js	44
3.5.1	Animación sincrónica	44
3.5.2	Animación por transiciones	45
3.5.3	Animación asincrónica.....	46
3.5.4	Líneas adicionales	48
3.5.5	Canvas.....	50
3.5.6	Acercamiento de imagen.....	50
3.6	Interfaz con el usuario	50
3.7	Topografía	53
3.7.1	Simplificación de geometrías.....	54
3.7.2	Capas adicionales.....	55
4	Análisis de resultados.....	57
4.1	Interpolación por distancia inversa ponderada.....	57
4.1.1	Datos de prueba con Three.js.....	57
4.1.2	Datos de prueba con D3.js	59
4.1.3	Datos reales con D3.js.....	62
4.2	Animación	75
4.2.1	Animación sincrónica	75
4.2.2	Animación por transiciones	77
4.2.3	Animación asincrónica.....	79
4.3	Rendimiento	81
4.3.1	Paralelismo.....	81
4.3.2	Recorte de matriz	83

4.3.3	Simplificación de capas	83
4.3.4	Canvas.....	85
4.3.5	Acercamiento de imagen.....	88
4.4	Velocidad nominal	91
5	Conclusiones	93
5.1	Recomendaciones.....	94
6	Anexos y Apéndices.....	96
	Anexo 1. Código de animación de líneas.....	96
	Anexo 2. Código de D3.js con canvas.	97
	Apéndice 1. Datos de la estación 72163 del mes de septiembre del 2014.....	99
	Apéndice 2. Simplificación de Geometrías.....	100
	Apéndice 3. Datos del tiempo de ejecución de la aplicación con elementos SVG, del mes de enero para el algoritmo de interpolación.....	108
	Apéndice 4. Datos del tiempo de ejecución de la aplicación con elementos SVG, del mes de enero para el algoritmo de interpolación en paralelo.	109
	Apéndice 5. Datos del tiempo de ejecución de la aplicación con canvas, del mes de enero para el algoritmo de interpolación.	110
	Apéndice 6. Datos del tiempo de ejecución de la aplicación con elementos SVG, del mes de enero para el código de carga de datos de las estaciones.....	111
	Apéndice 7. Datos del tiempo de ejecución de la aplicación con elementos SVG y canvas, para el código de inicialización de líneas.	112
	Apéndice 8. Datos del tiempo de ejecución de la aplicación con elementos SVG y canvas, por línea.....	113
	Apéndice 9. Datos del tiempo de ejecución de la aplicación con acercamiento y canvas, por línea.	115
	Apéndice 10. Datos del tiempo de ejecución de la aplicación con elementos SVG, para la carga del mapa y capas sin simplificación.....	117
	Apéndice 11. Datos del tiempo de ejecución de la aplicación con elementos SVG, para la carga de capas con simplificación de geometrías.	118
7	Referencias bibliográficas.....	119

ÍNDICE DE TABLAS

Tabla 3.1. Datos de estaciones meteorológicas.....	41
Tabla 3.2. Estaciones seleccionadas para la realización del proyecto	42
Tabla 3.3. Conversión de punto cardinal a grados	43
Tabla 3.4. Duración de la primera transición según la velocidad del viento	47
Tabla 3.5. Duración de la segunda transición según la velocidad del viento	47
Tabla 3.6. Rangos de tiempos para las transiciones de las líneas	48
Tabla 4.1. Colores según la velocidad del viento	57
Tabla 4.2. Primeros siete días del mes de mayo del 2014 en la estación 94013.....	67
Tabla 4.3. Rangos de grados para los puntos cardinales.....	68
Tabla 4.4. Zonas climáticas por estación.	71
Tabla 4.5. Comparación de promedio de tiempos de ejecución del algoritmo de interpolación sin paralelizar y paralelizado, datos del mes de enero.....	82
Tabla 4.6. Promedio de tiempo de ejecución del código de carga de datos de las estaciones, datos del mes de enero.....	82
Tabla 4.7. Recortes de matriz para los tamaños de 30x30, 40x40 y 60x60.....	83
Tabla 4.8. Tiempo de carga promedio del mapa de Costa Rica en TopoJSON.....	83
Tabla 4.9. Tiempo de carga promedio para las capas del mapa de Costa Rica en TopoJSON.....	84
Tabla 4.10. Promedio de tiempos de ejecución del código de inicialización de líneas con elementos SVG y con canvas.....	86
Tabla 4.11. Promedio de tiempos de ejecución por línea con elementos SVG y con canvas.....	86
Tabla 4.12. Días con vientos con velocidad superior a la velocidad nominal	91

ÍNDICE DE FIGURAS

Figura 1.1. Animated UK Wind Char, mapa de vientos del Reino Unido.....	16
Figura 1.2. Wind Map, mapa de vientos de Estados Unidos	16
Figura 1.3. Air, mapa de vientos de Tokyo.....	17
Figura 1.4. Earth, mapa de condiciones climáticas mundial.....	18
Figura 2.1. Representación gráfica de un campo vectorial concéntrico en un plano.....	27
Figura 2.2. Gráfico de potencia contra velocidad de viento para una turbina de viento estándar	30
Figura 3.1. Diagrama de flujo general de la aplicación de Mapa de vientos de Costa Rica.....	38
Figura 3.2. Diagrama de flujo del algoritmo de interpolación.....	40
Figura 3.3. Diagrama de flujo de la animación.....	45
Figura 3.4. Diagrama de flujo de la animación con líneas adicionales.....	49
Figura 3.5. Secciones de interfaz de usuario creadas en Semantic UI para la aplicación web	51
Figura 3.6. Secciones de interfaz de usuario creadas en Semantic UI para la aplicación web con canvas.....	52
Figura 3.7. Capa vectorial de límites, provincias, cantones y distritos de Costa Rica.....	53
Figura 3.8. Geometría de parte del límite norte de Costa Rica sin simplificar.....	54
Figura 3.9. Geometría de parte del límite norte de Costa Rica simplificada	54
Figura 3.10. Capa de ciudades de Costa Rica	55
Figura 3.11. Capa de volcanes de Costa Rica	56
Figura 4.1. Prueba de ejemplo para el algoritmo de distancia inversa ponderada con valores de prueba diferentes para cada estaciones en Three.js.....	58
Figura 4.2. Prueba de ejemplo para el algoritmo de distancia inversa ponderada con vientos de 13 m/s a 170°	59
Figura 4.3. Prueba de ejemplo para el algoritmo de distancia inversa ponderada con vientos de 2.5 m/s a 10°	60
Figura 4.4. Prueba de ejemplo para el algoritmo de distancia inversa ponderada con valores de prueba diferentes para las estaciones	61
Figura 4.5. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 2	62
Figura 4.6. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 4	63
Figura 4.7. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 6	64
Figura 4.8. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 8	65
Figura 4.9. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 10	66
Figura 4.10. Mapa de vientos de Costa Rica del 2 de enero del 2014, interpolación de distancia inversa ponderada con potencia 4	69
Figura 4.11. Mapa de vientos de Costa Rica del 3 de enero del 2014, interpolación de distancia inversa ponderada con potencia 4	70
Figura 4.12. Mapa de vientos de Guanacaste del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 4	72
Figura 4.13. Mapa de vientos de Guanacaste, datos de prueba	73
Figura 4.14. Mapa de vientos de Costa Rica, datos de prueba, matriz 90x90	74
Figura 4.15. Mapa de viento de Costa Rica con animación sincrónica	76
Figura 4.16. Mapa de viento de Costa Rica con animación sincrónica, al finalizar un ciclo de animación.....	77
Figura 4.17. Mapa de viento de Costa Rica con animación por transiciones	78

ÍNDICE DE FIGURAS

Figura 4.18. Mapa de viento de Costa Rica con animación por transiciones, con líneas de ancho variable y flechas indicadoras de dirección	79
Figura 4.19. Mapa de viento de Costa Rica con animación asincrónica	80
Figura 4.20. Gráfico de tiempo de carga para las capas con geometrías simplificadas del mapa de Costa Rica en TopoJSON	84
Figura 4.21. Gráfico de comparación de tiempos de carga entre las capas sin simplificar y las capas simplificadas	85
Figura 4.22. Mapa de viento de Costa Rica con animación asincrónica y canvas.....	87
Figura 4.23. Gráfico de cambios en las propiedades de las líneas según el tamaño de la imagen. ...	88
Figura 4.24. Gráfico de tiempos de ejecución por línea para canvas con mínimo acercamiento, con máximo acercamiento y para canvas sin acercamiento	89
Figura 4.25. Mapa de viento de Costa Rica con mínimo acercamiento.....	90
Figura 4.26. Mapa de viento de Costa Rica con máximo acercamiento.....	90
Figura 4.27. Mapa de vientos de Costa Rica, el día 9 de abril del 2014.....	92
Figura 6.1. Geometría del límite exterior de Costa Rica sin simplificar	100
Figura 6.2. Geometría del límite exterior de Costa Rica simplificada.....	101
Figura 6.3. Geometría de las provincias de Costa Rica sin simplificar	102
Figura 6.4. Geometría de las provincias de Costa Rica simplificada	103
Figura 6.5. Geometría de los cantones de Costa Rica sin simplificar.....	104
Figura 6.6. Geometría de los cantones de Costa Rica simplificada.....	105
Figura 6.7. Geometría de los distritos de Costa Rica sin simplificar.....	106
Figura 6.8. Geometría de los distritos de Costa Rica simplificada.....	107

1 *Introducción*

1.1 *Descripción general del proyecto*

El proyecto consiste en una aplicación web que utiliza la biblioteca D3.js para la visualización de los datos de las corrientes de viento en Costa Rica del año 2014. En D3.js se carga un mapa de Costa Rica, en formato TopoJSON, para luego superponer una matriz de vectores que representan las corrientes de viento, con su dirección y rapidez.

El valor del proyecto para el Instituto Meteorológico Nacional es que le puede permitir realizar predicciones según los datos gráficos obtenidos de las visualizaciones, ya que si se tiene la visualización de las corrientes de viento, se puede predecir el comportamiento del viento para condiciones climáticas similares. Además también se puede predecir el flujo de agentes contaminantes en el aire, por ejemplo, la ceniza volcánica. Otro uso del proyecto es la utilización de las visualizaciones para la futura colocación de molinos para la obtención de energía eólica.

1.2 *Antecedentes*

1.2.1 *Descripción de la empresa*

El proyecto es un trabajo de colaboración entre el Tecnológico de Costa Rica y el Instituto Meteorológico Nacional.

1.2.1.1 *Tecnológico de Costa Rica*

En este apartado se mencionarán los aspectos más importantes de la historia del Instituto Tecnológico de Costa Rica basado en [1] y [2].

El Tecnológico de Costa Rica es una institución autónoma de educación superior universitaria, dedicada a la docencia, la investigación y la extensión de la tecnología y ciencias conexas para el desarrollo de Costa Rica. En 1971, gracias al presidente de la República, José Figueres Ferrer, y al presidente de la Asamblea Legislativa, Daniel Oduber Quirós, se apoyó la creación de una institución de educación superior siguiendo el modelo del Tecnológico de Monterrey en México. El jueves 10 de junio de 1971 se creó el Instituto Tecnológico de Costa Rica mediante la ley No. 4777.

El primer rector del Tecnológico de Costa Rica fue Vidal Quirós Berrocal, un ingeniero civil con formación en el Tecnológico de Monterrey. El Instituto Tecnológico de Costa Rica estuvo inicialmente instalado en el Edificio Pirie, el cual fue donado por la Municipalidad de Cartago. En 1973, comenzaron las tres primeras carreras del TEC: Ingeniería en Construcción, Ingeniería en Producción Industrial e Ingeniería en Mantenimiento Industrial, seleccionadas por las necesidades en el país. Además de esto se compraron 100 hectáreas de terreno en las cuales se construyó el actual Campus Central.

Ética profesional, honradez, prudencia, amplio conocimiento e interés en la transferencia de conocimiento fueron los valores y conceptos que los docentes dejaron plasmados en los primeros egresados. Al finalizar el año 1982, el TEC pasó de 3 a 16 carreras, de 87 estudiantes a 2 218, y de 2 a 24 edificios. El TEC inició una etapa de cooperación con la Embajada de Alemania. Esto dio inicio a la carrera de Ingeniería Electrónica, con profesores capacitados en Alemania y otros en México.

Con el cambio de rector a Roberto Villalobos Ardón, en 1983, el TEC se integró a los Juegos Deportivos Nacionales, también se creó la Carrera de Ingeniería en Metalurgia y se abre la primera maestría, la Maestría en Computación.

En 1987, durante la gestión del rector Arturo Jofré Vartanián se crearon los primeros centros de investigación y primer laboratorio de computación, se creó la Fundación Tecnológica (FUNDATEC) y se construyó el Centro de Transferencia Tecnológica (CETT) en Zapote.

En 1995, durante la gestión del rector Alejandro Cruz Molina, el enfoque se centró en mejorar la condición socioeconómica de la provincia de Cartago y en apoyar al sector empresarial y el tecnológico de Costa Rica. Entre otros logros, se atrajo inversión privada extranjera al país (Intel).

A partir del 2003, con el rector Eugenio Trejos Benavides se abrieron 5 nuevas carreras y varios programas de posgrados, además se cumplió con la Ley 7600.

En el 2012, durante la gestión del rector Julio Calvo Alvarado, se cambió el nombre del Instituto Tecnológico de Costa Rica para que se conociera públicamente como Tecnológico de Costa Rica, además se decreta al TEC como “Institución Benemérita de la educación, la cultura y la ciencia y la tecnología costarricense”.

El Tecnológico de Costa Rica posee varias sedes alrededor del país, el Campus Central se ubica en Cartago y cuenta con sedes en San Carlos, San José (Centro Académico y el Centro de Transferencia Tecnológica) y Limón.

La estructura organizacional del Tecnológico de Costa Rica se divide en: Asamblea Institucional, máxima autoridad del TEC; Consejo Institucional, órgano directivo superior del TEC; Rector, el rector es el funcionario de más alta jerarquía ejecutiva del TEC; cuatro Vicerrectorías para cumplir con el principal quehacer de la institución, a saber: Docencia, Vida Estudiantil, Administración, Investigación y Extensión.

1.2.1.2 Instituto Meteorológico Nacional

En este apartado se mencionarán los aspectos más importantes de la historia del Instituto Meteorológico Nacional basado en [3] y [4].

El botánico y naturalista danés A.S. Oersted en 1847 dedujo un promedio anual de temperatura de 16 C para Cartago con un mínimo de 13 C en enero, además también documentó la primera medición de lluvia, muy similar al promedio actual, y la temperatura de San José. La Oficina de Estadística incluyó, desde sus orígenes (1861), la recopilación de la información meteorológica como una de sus tareas primordiales. Esta situación motivó a la participación de Costa Rica en la primera Red Internacional de Datos Meteorológicos en 1877.

Para 1887 se establece el primer Observatorio Meteorológico Nacional, bajo la dirección del científico suizo Dr. Henri Pittier Dormond. El 7 de abril de 1888 se publica en La Gaceta la creación del Instituto Meteorológico Nacional y se nombra como Director a Pittier, quien inicia las primeras observaciones sistemáticas de lluvia y de temperaturas en San José, además se comenzó el estudio del clima del país.

La creación de la institución fue impulsada por don Mauro Fernández, abogado, economista y educador, reformador de la enseñanza en Costa Rica, Ministro de Hacienda e Instrucción Pública, alma e inspiración del gobierno de don Bernardo Soto, Presidente de la República entre 1885 y 1889.

Al año de su fundación, el Instituto Meteorológico Nacional pasó a formar parte del Instituto Físico Geográfico, junto con el Observatorio Meteorológico y las estaciones meteorológicas de su dependencia, el Servicio Geográfico, el Museo Nacional y el Herbario Nacional.

En 1936, el Instituto Geográfico se suprimió por falta de presupuesto. Los instrumentos meteorológicos se trasladaron al Departamento Nacional de Agricultura, del Ministerio de Fomento, donde se siguieron haciendo las observaciones meteorológicas.

A partir de 1944 se creó de nuevo el Servicio Meteorológico y Sismológico como dependencia del nuevo Instituto Geográfico Nacional, y se hace un intento por establecer a experimentar en el pronóstico del tiempo a corto plazo. La información se recibía al principio por radio, luego, con el avance de la tecnología, por teletipo.

Hasta 1968 el Servicio Meteorológico y Sismológico desarrolla una actividad polifacética: Meteorología, Sismología, Astronomía, aspectos oceanográficos como las mareas y hasta la hora oficial. A partir de 1970, y con el apoyo del Proyecto Hidrometeorológico Centroamericano, auspiciado por la OMM, se renovó el equipo de medición, se instaló una red básica y 6 estaciones sinópticas con programas de observación de 12 horas al día.

En 1973, mediante la Ley N° 5222 se crea el Instituto Meteorológico Nacional (IMN), adscrito al Ministerio de Agricultura y Ganadería y asume su dirección el Lic. Gerardo Lizano, uno de los primeros graduados en meteorología de la UCR. Posteriormente la Ley N° 7152 de 1990 traslada el IMN al Ministerio de Recursos Naturales, Energía y Minas (MIRENEM), el cual a partir de 1995 pasa a llamarse Ministerio de Ambiente y Energía (MINAE) como una dirección adscrita.

El IMN cuenta entonces con ciento diecisiete años de actividad continua, para el beneficio social y económico del país. Es una institución que se ha preocupado durante este tiempo por el desarrollo profesional y tecnológico con el fin de ser un instituto meteorológico de punta en el ámbito nacional e internacional que cumpla con las exigencias que le impone el Siglo XXI, en campos como la prevención de desastres hidrometeorológicos, la adaptación y mitigación del cambio climático y continuar educando a las nuevas generaciones sobre las aplicaciones de la ciencia meteorológica para el desarrollo de la sociedad costarricense.

El Instituto Meteorológico Nacional se ubica en San José, en Barrio Aranjuez, avenida 9 y calle 17 frente al costado noroeste del Hospital Calderón Guardia.

La estructura organizacional del Instituto Meteorológico Nacional consta del Viceministro de Ambiente, el Consejo Nacional de Meteorología, Oficina de Recursos Humanos, Departamento Administrativo Financiero, Unidad de Planificación, Oficina de Prensa, Unidad de Informática, Departamento de Red Meteorológica y Procesamiento de Datos, Departamento de Meteorología sinóptica y aeronáutica, departamento de climatología e investigaciones aplicadas, departamento de Información y Departamento de Aguas.

1.2.2 Trabajos similares

- Mapas de velocidad media del viento en el centro-este de la provincia de Neuquen [5]

Trabajo argentino donde se crean mapas de velocidad media anual del viento a 10 y 30 metros de altura, utilizando un modelo de diagnóstico que satisface la ecuación de conservación de la masa. Al considerar la altura se hicieron ajustes horizontales y verticales, para los ajustes horizontales de la velocidad se realizó una interpolación de distancia inversa ponderada, para los ajustes verticales se consideró un perfil de viento logarítmico que tiene en cuenta la interpolación horizontal, la rugosidad aerodinámica de la superficie y la estabilidad de la atmósfera.

- Animated UK Wind Chart [6]

Gráfico animado donde se muestra la rapidez y dirección del viento en el Reino Unido. Cuanta mayor velocidad las líneas llegan más lejos. Inspirado en Wind Map de hint.fm. Los datos fueron tomados del servicio de Met Office's DataPoint del Reino Unido y el mapa tomado del tutorial de mapeo de Mike Bostock.

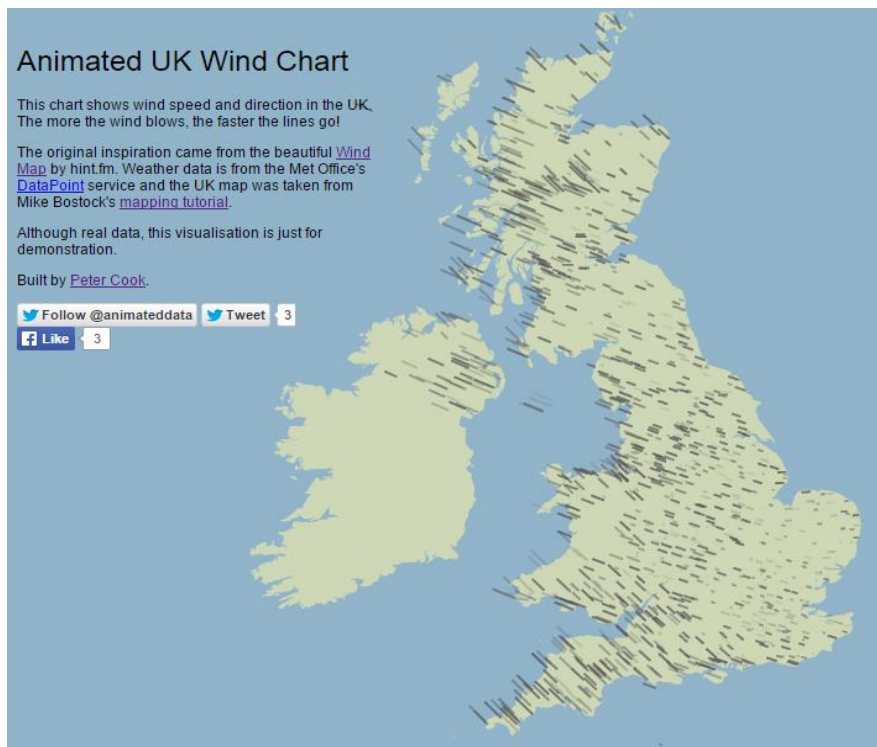


Figura 1.1. Animated UK Wind Char, mapa de vientos del Reino Unido [6]

- Wind Map [7]

Gráfico animado donde se muestra la rapidez y dirección del viento en los Estados Unidos.

Los datos son tomados del National Digital Forecast Database y se refrescan cada hora.



Figura 1.2. Wind Map, mapa de vientos de Estados Unidos [7]

- Air [8]

Gráfico animado donde se muestra la rapidez y dirección del viento en Tokyo. Los datos fueron suministrados por el Gobierno Metropolitano de Tokyo.

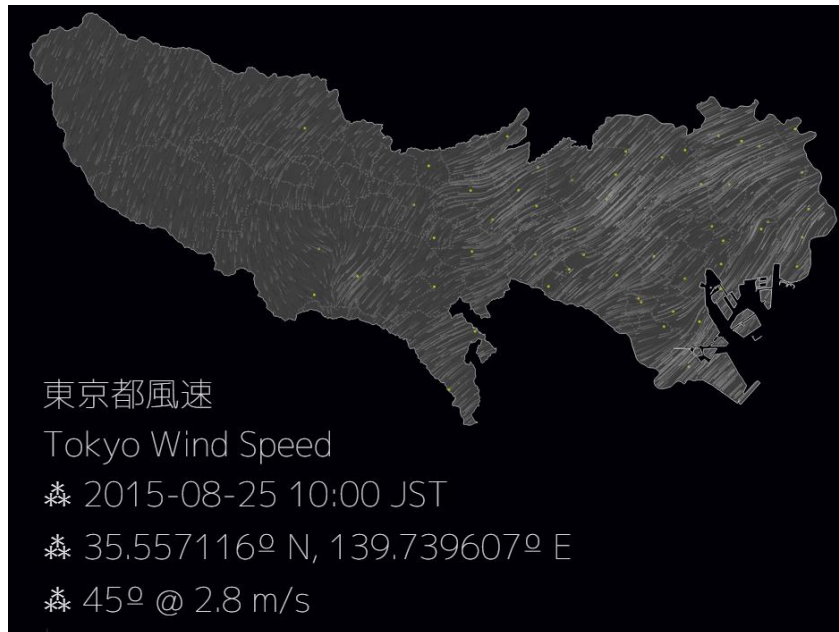


Figura 1.3. Air, mapa de vientos de Tokyo [8]

- Earth [9]

Gráfico animado donde se muestran los datos del viento y otras condiciones climáticas en el mundo. Los datos fueron tomados de Global Forecast System y se actualiza cuatro veces al día. Basado en Air, mapa de vientos de Tokyo y en Wind Map de hint.fm.

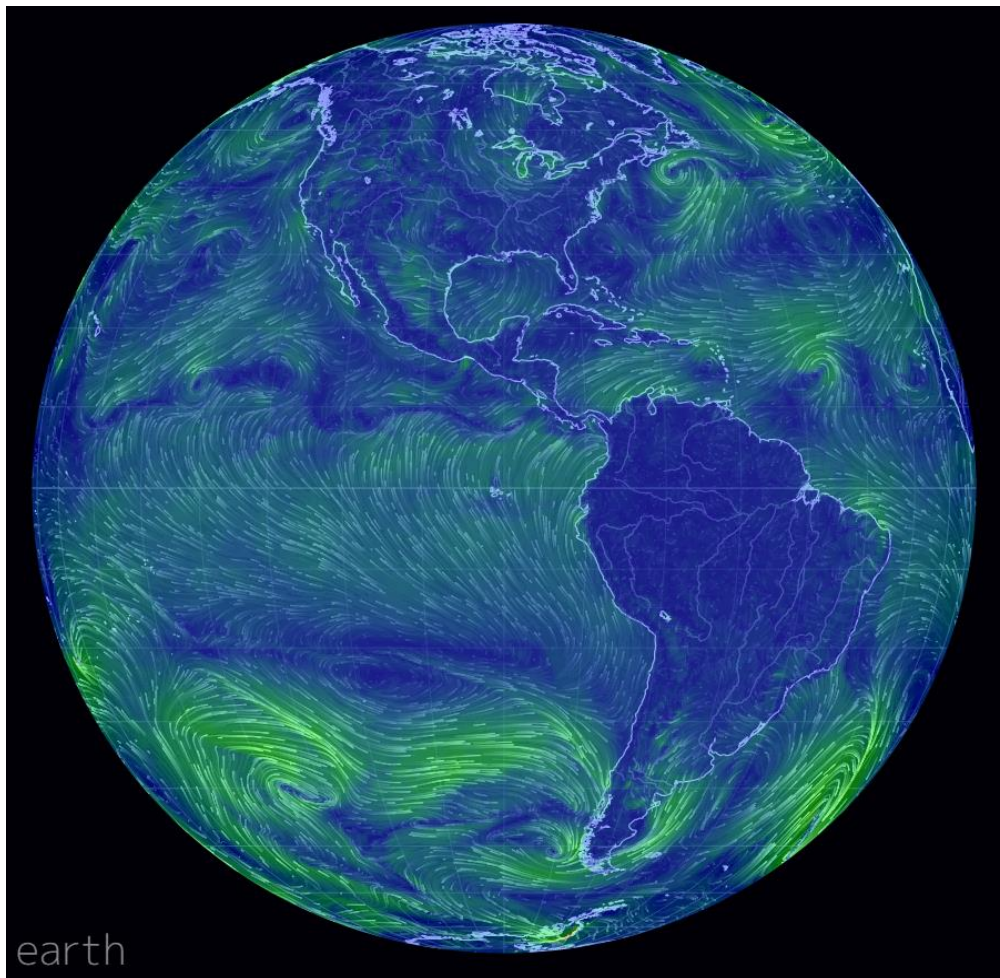


Figura 1.4. Earth, mapa de condiciones climáticas mundial [9]

1.3 Planteamiento del problema

1.3.1 Contexto del problema

Con la reciente actividad volcánica en el país, por parte del volcán Turrialba, se ha encontrado la necesidad de tener una forma de visualizar la forma en la que el viento afecta la distribución de ceniza a lo largo del país. Si se tuviera una manera de visualizar las corrientes de viento, se podría predecir con más exactitud cuáles serían las áreas afectadas por la ceniza.

En Costa Rica existen entidades públicas y privadas a las que esta información les sería de utilidad, por ejemplo al Aeropuerto Juan Santamaría, cuyo servicio se ha visto afectado por la caída de ceniza.

Además de la predicción del flujo de contaminantes a través del aire, una visualización de las corrientes del viento puede ayudar en la colocación de molinos de energía eólica en el país, si se tiene la visualización se puede observar de manera gráfica cuáles son los puntos más adecuados para la colocación de estos molinos. Esta función facilitaría los estudios necesarios para la colocación de molinos en Costa Rica.

1.3.2 Situación problemática

El proyecto busca solucionar la falta de información visual sobre las corrientes de viento en Costa Rica, el Instituto Meteorológico Nacional posee estos datos pero no la manera de visualizarlos. Con una visualización de las corrientes de viento en Costa Rica se podría predecir acontecimientos importantes en el país, el acontecimiento más importante que se busca en el proyecto es sobre el flujo de contaminantes y enfermedades a través del aire que afecten a personas, animales, plantas, estructuras, fuentes de alimento o nacientes de agua.

Este proyecto también podría ser utilizado por otras entidades, además del Instituto Meteorológico Nacional, para su propia utilidad, la idea de que el proyecto esté montado en una aplicación web es con el fin de que la información esté disponible a toda la población.

1.3.3 Definición del problema

La utilidad de la información visual de las corrientes de viento ayuda a instituciones públicas y privadas con predicciones acerca de los movimientos aéreos. Los aeropuertos pueden hacer uso de esta información para saber si la caída de ceniza afectará en ese momento la funcionalidad del aeropuerto. De igual manera el Ministerio de Salud puede lanzar campañas de prevención contra ceniza o enfermedades contagiosas de transmisión aérea. Al Instituto Costarricense de Acueductos y Alcantarillados le podría servir para saber si algún tipo de contaminante fluye hacia las nacientes de agua donde se extrae el agua potable del país.

Otra institución a la cual le podría ser de utilidad la información es al Ministerio de Agricultura y Ganadería para cuidar las plantas y animales de las zonas que pudieran ser afectados por contaminantes que fluyen a través del aire.

De esta manera otras instituciones con funciones similares pueden usar los datos visuales para tomar las medidas necesarias a cada caso.

1.4 Objetivos del proyecto

1.4.1 Objetivo general

Desarrollo de una herramienta en ambiente web para la visualización de las corrientes de viento en Costa Rica.

1.4.2 Objetivos específicos

Desarrollar un algoritmo de interpolación que represente el campo vectorial asociado a las corrientes de viento presentes en el país.

Crear una aplicación web que contenga la visualización del algoritmo de las corrientes de viento en Costa Rica.

1.5 Alcances, entregables y limitaciones del proyecto

La principal limitación del proyecto es el ancho de banda, ya que se tiene una aplicación web que toma los datos de todos los días del 2014 en veinte estaciones del país, hace un algoritmo de interpolación con estos datos para cada día y finalmente toma la matriz resultante para crear la visualización; aunque se tengan equipos con una gran capacidad de procesamiento el ancho de banda podría ser el principal limitante de la aplicación.

Además, se puede inferir del punto anterior, la capacidad de procesamiento del equipo, ya que por la cantidad de datos por procesar, se podría estar necesitando equipo especializado para hacer pruebas.

Otra limitante importante es la obtención de datos, el Instituto Meteorológico Nacional proporciona datos pero tiene condiciones específicas para el tipo y cantidad de datos entregados.

1.5.1 Descripción de entregables

- Bitácora de proyecto: bitácora de las actividades realizadas día a día durante el transcurso del proyecto.
- Algoritmo de interpolación: solución al algoritmo de interpolación necesario para la recreación del campo vectorial de las corrientes de viento en Costa Rica.
- Aplicación web de visualización de corrientes de viento: aplicación final con todas las funciones propuestas.
- Presentación final: presentación de la aplicación terminada a representantes de iReal, e-science, Instituto Meteorológico Nacional y VIE.
- Informe del proyecto: informe al finalizar el proyecto sobre las actividades realizadas y sus resultados.

2 Marco teórico

2.1 Interpolación

La interpolación es un método matemático para obtener datos nuevos a partir de datos conocidos. A continuación se explican los conceptos de interpolación y en el apartado 2.1.1 se profundiza en la interpolación polinómica, ambos basados en [10].

Al trabajar con datos obtenidos de fuentes diversas, sucede que solo se puede obtener una cantidad limitada de datos, por ejemplo al realizar mediciones experimentales. En estos casos se utilizan funciones para aproximar los datos faltantes.

La interpolación trata de resolver dos problemas, con (x_i, f_i) como datos iniciales:

- Encontrar $f(x)$ tal que

$$f(x_i) = f_i \quad i = 0, 1, \dots, n \quad (2.1)$$

Donde se genera una función aproximada con los datos iniciales, en esta se puede evaluar los datos deseados para obtener resultados aproximados a los datos reales.

- Calcular directamente $f(\tilde{x})$ para un punto \tilde{x} distinto de $x_i = 0, 1, \dots, n$

Donde se calcula directamente el valor deseado sin tener una función.

El teorema de Weierstrass asegura que una función continua en un intervalo cerrado es aproximable por una función polinómica, además los polinomios tienen la ventaja de ser integrables o derivables, siendo el método de interpolación polinómica el más utilizado y el más sencillo.

2.1.1 Interpolación polinómica

Un polinomio es una función con un conjunto finito de coeficientes constantes con variables de exponentes enteros positivos, unidas por operaciones aritméticas como se muestra en (2.2).

$$P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (2.2)$$

En una interpolación polinómica con (x_i, f_i) como datos iniciales, se construye un polinomio como el anterior de grado igual o menor que la cantidad de datos iniciales, con los datos iniciales y el polinomio se resuelve el sistema de ecuaciones lineales:

$$a_0 + a_1x_0 + \dots + a_nx_0^n = f_0 \quad (2.3)$$

$$a_0 + a_1x_1 + \dots + a_nx_1^n = f_1 \quad (2.4)$$

$$a_0 + a_1x_n + \dots + a_nx_n^n = f_n \quad (2.5)$$

Donde si todos los x_i son distintos entre sí, entonces el sistema tiene una única solución.

2.1.2 Interpolación meteorológica

Los métodos de interpolación polinómica son útiles en casos de una variable y en el caso de condiciones meteorológicas la interpolación se da sobre un plano, teniendo como valores iniciales (x_i, y_i, f_i) .

2.1.2.1 Interpolación espacial

Para datos geográficos la mejor interpolación es la interpolación espacial, la principal razón es que en esta se utilizan dos o más dimensiones. La visualización de la elevación, las precipitaciones, la temperatura, el viento, entre otras condiciones, se aprecia mejor en varias dimensiones, por lo que los métodos de interpolación comunes no son efectivos para este tipo de datos.

2.1.2.2 Interpolación global y local

Los métodos de interpolación se clasifican en globales y locales. Los métodos globales se caracterizan por utilizar todos los datos iniciales disponibles y permite visualizar tendencias en estos. Los métodos locales en cambio solo utilizan datos de una zona específica con el fin de lograr una interpolación precisa.

En la interpolación espacial de datos geográficos se utilizan principalmente los siguientes métodos de interpolación:

2.1.2.3 Vecino natural

En este algoritmo a los valores iniciales se les asigna un área en el plano según la distancia entre los otros puntos iniciales. Al buscar la interpolación de un nuevo punto, se buscan las áreas más cercanas a este, y se calcula una ponderación del valor de los puntos de las áreas correspondientes, esta ponderación dependerá del área asignada inicialmente a los puntos [11].

2.1.2.4 Distancia inversa ponderada

En este algoritmo se pondera un valor respecto a los valores iniciales y la distancia de estos al punto que se quiere interpolar. Para que este algoritmo se pueda utilizar los puntos deben ser dependientes de la ubicación, ya que se propone que la influencia de los puntos disminuye entre más distancia haya desde su ubicación hasta el punto deseado [12].

La función de interpolación por distancia inversa ponderada es el siguiente [13]:

$$\tilde{z}_j = \frac{\sum_{i=0}^n \frac{z_i}{d_{ij}^\beta}}{\sum_{i=0}^n \frac{1}{d_{ij}^\beta}} \quad (2.6)$$

Donde \tilde{z}_j es el valor buscado en el punto j , z_i el valor inicial en el punto i , d_{ij}^β la distancia ponderada entre el punto buscado j y el punto inicial i , β la potencia.

En esta función observamos que el valor del nuevo punto es la sumatoria de las influencias de los puntos iniciales entre la distancia a que se encuentran estos puntos.

La influencia de los puntos se puede controlar con el parámetro de potencia, donde cuanto más alta sea la potencia, la influencia es más fuerte en los puntos cercanos. Con valores bajos cada punto tiene una mayor influencia en puntos más lejanos.

2.1.2.5 *Kriging*

A diferencia de los algoritmos anteriores el Kriging no es un algoritmo determinístico (caracterizados por utilizar los valores circundantes para realizar la interpolación), el Kriging es un método geoestadístico donde se consideran las relaciones estadísticas entre los puntos, por lo que agrega precisión a las predicciones.

Para utilizar el Kriging primero se debe realizar una variografía en el plano, en esta se calcula la diferencia cuadrada entre los valores de las distancias iniciales en cada punto, esto genera un diagrama de varianza empírico. El segundo paso es ajustar un modelo que se adapte al diagrama de varianza encontrado, este puede ser circular, esférico, exponencial o lineal, este ajuste se realiza para otorgarle una función a la distribución de puntos del diagrama de varianza [14].

La función de interpolación por Kriging es la siguiente [13]:

$$\tilde{z}_j = \sum_{i=0}^n z_i \lambda_i \quad (2.7)$$

Donde \tilde{z}_j es el valor buscado en el punto j , z_i el valor inicial en el punto i , λ_i la distancia ponderada ajustada por el modelo de varianza en i .

2.2 *Sistema de coordenadas cartesianas bidimensionales*

En este apartado se explicarán los conceptos de sistema de coordenadas basados en [15], [16] y [17].

Los sistemas de coordenadas son referencias para realizar estudios sobre objetos físicos, estos estudios buscan describir las propiedades y los estados presentes en dichos objetos, y se representan por medio de valores medibles. Estos valores pueden ser escalares o vectoriales, los valores escalares consisten de un número y su respectiva unidad (ejemplo: masa, presión), los datos vectoriales además de un número y su respectiva unidad (magnitud), cuentan con dirección y sentido (orientación), como por ejemplo la velocidad y el desplazamiento.

En un sistema de coordenadas cartesianas se tiene un punto de origen y dos ejes, horizontal y vertical, que pasan por el punto de origen y son perpendiculares entre sí, por lo que el plano queda dividido en cuatro cuadrantes. Para la representación de magnitudes escalares se representan como puntos en el sistema de coordenadas, para la representación de magnitudes vectoriales estos puntos además deben tener una dirección.

En este proyecto las ecuaciones más importantes del manejo de sistemas de coordenadas cartesianas son la distancia entre puntos (2.8) y la conversión de ángulos a coordenadas cartesianas (2.9).

La siguiente es la ecuación para el cálculo de la distancia dados dos puntos:

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.8)$$

Donde d es la distancia entre puntos, P es un punto, x es la componente horizontal de un punto y y es la componente vertical de un punto. Esta ecuación es utilizada en el cálculo de la interpolación.

La siguiente ecuación sirve para la conversión de los valores de un vector (magnitud y ángulo) en coordenadas cartesianas:

$$\begin{cases} x = r \cos \theta \\ y = r \sen \theta \end{cases} \quad (2.9)$$

Donde x es la componente horizontal del punto, y es la componente vertical del punto, r es la magnitud del vector y θ es el ángulo del vector. Esta ecuación se utiliza para cambiar el ángulo de calculado en la interpolación, junto con la magnitud, a coordenadas cartesianas.

2.2.1 Campos vectoriales

Un campo físico se refiere a una región del espacio donde se presentan u observan propiedades físicas. Inicialmente el físico Faraday utilizó el concepto de campo para representar fuerzas electromagnéticas, en la actualidad se utiliza este concepto para describir todas las interacciones de la naturaleza.

Los campos físicos se clasifican en escalares y vectoriales. En los campos escalares las interacciones dependen de la posición de los objetos físicos, mientras que en los campos vectoriales la interacción no solo depende de la posición sino que también de la dirección de los objetos físicos. Estos objetos representados por posición y dirección son llamados vectores. Los campos vectoriales además definen las líneas de campo, las cuales son tangentes a los vectores y ayudan en la visualización de los campos que representan.

Para representar un campo vectorial en un plano se utilizan un conjunto de flechas, donde cada flecha corresponde a un vector con origen en un punto del plano, esta representación es análoga para un campo vectorial es el espacio. En la Figura 2.1 se muestra la representación gráfica de un campo vectorial en un plano.

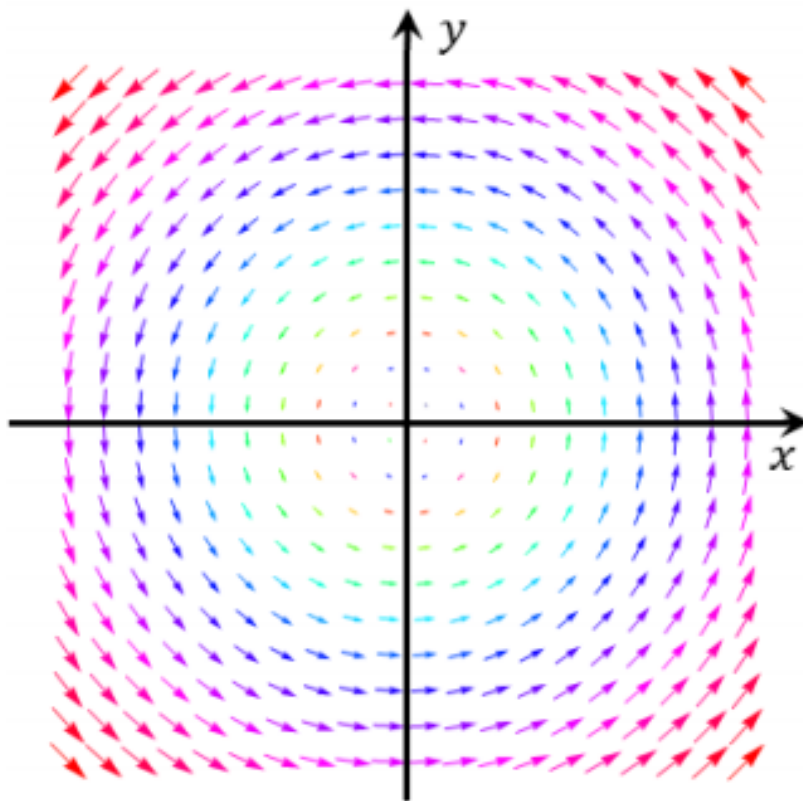


Figura 2.1. Representación gráfica de un campo vectorial concéntrico en un plano [16]

2.2.2 Campo de velocidades

Un campo de velocidades es un campo vectorial utilizado para la descripción de fluidos, en este se muestra la rapidez y la dirección con las que los elementos del fluido pasan a través del espacio. En los campos de velocidades, los vectores forman líneas de flujo, similares a las líneas de campo, estas representan las trayectorias seguidas por las partículas. Las líneas de flujo se utilizan para visualizar el movimiento del fluido alrededor de objetos, por ejemplo el viento pasando a través de un sistema montañoso.

El campo de velocidades es un campo de posición y tiempo, en este se indica que en cada punto del espacio y en cada instante, se tiene un valor asociado para la velocidad de una partícula posicionada en ese sitio en ese momento. La función que describe el campo de velocidad, por ende, debe representar la rapidez del movimiento y la orientación en el espacio, además debe ser capaz de dar las componentes de la velocidad a lo largo de cualquier dirección en el espacio. En las siguientes ecuaciones se describen las funciones del campo de velocidad para las componentes de la velocidad.

$$V_x = V_x(x, y, t) \quad (2.10)$$

$$V_y = V_y(x, y, t) \quad (2.11)$$

Donde V_x representa la componente horizontal de la velocidad, V_y representa la componente vertical de la velocidad, x, y representa la localización en el espacio y t representa el instante de tiempo.

En este proyecto se tiene una representación de un campo de velocidades del viento, donde la localización en el espacio es sobre la superficie de Costa Rica (ver apartado 3.7) y el instante de tiempo son los días del año 2014. Se utiliza una medida de instante de tiempo diaria por restricciones con la obtención de datos (ver apartado 3.3). El campo de velocidades del viento se construye con datos suministrados por el Instituto Meteorológico Nacional de rapidez y dirección del viento.

2.3 Energía eólica

En este apartado se explicarán los conceptos de energía eólica basados en la referencia [18].

El viento al estar en movimiento posee energía cinética, esta energía cinética es la que se aprovecha para hacer girar los rotores de las turbinas de viento. La energía cinética se puede calcular con (2.12)

$$E_c = \frac{1}{2}mv^2 \quad (2.12)$$

Donde E_c es la energía cinética, m es la masa del viento y v es la velocidad del viento.

Con esto se puede calcular la potencia generada por una corriente de viento pasando por una superficie.

$$P = \frac{1}{2}\rho Av^3 \quad (2.13)$$

Donde P es la potencia del viento, ρ es la densidad del viento, A es la superficie y v la velocidad del viento.

Al incluir un rotor de una turbina de viento, además se tiene que considerar el rendimiento del motor, por lo que tenemos:

$$P = \left(\frac{1}{2}\rho Av^3\right) C_p \quad (2.14)$$

Donde C_p es el coeficiente de potencia, el cual determina el rendimiento del rotor.

2.3.1 Variación de potencia en una turbina de viento

En este apartado se explicarán los conceptos de turbinas de viento basado en [19].

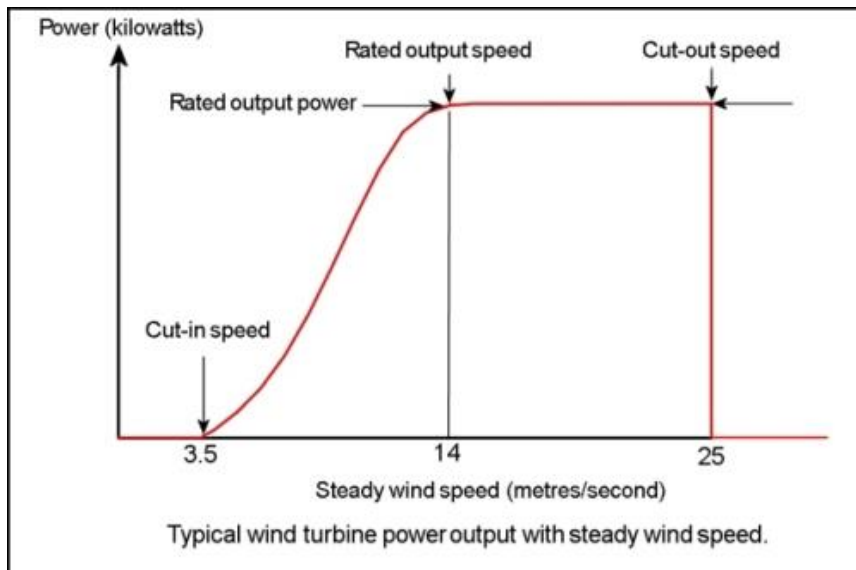


Figura 2.2. Gráfico de potencia contra velocidad de viento para una turbina de viento estándar [19]

2.3.1.1 Velocidad de corte inferior (Cut-in speed)

En velocidades muy bajas el viento no tiene suficiente energía como para superar las pérdidas de las turbinas. La velocidad de corte inferior es la velocidad a la cual la energía del viento es suficiente para comenzar a mover la turbina, este valor es generalmente entre 3 y 4 metros por segundo.

2.3.1.2 Velocidad nominal y potencia nominal (Rated output power and rate output wind speed)

La velocidad nominal es el punto en el que la turbina alcanza la potencia nominal, la potencia nominal es el límite de energía que el generador eléctrico, de la turbina, es capaz de producir. La velocidad nominal se alcanza generalmente entre los 12 y 17 metros por segundo, a velocidades mayores la turbina está diseñada para limitarse a su potencia máxima.

2.3.1.3 Velocidad de corte superior

Al alcanzar la velocidad de corte superior hay un riesgo de daño en el motor, se utiliza un sistema de frenado para detener al rotor. La velocidad de corte superior se alcanza generalmente a los 25 metros por segundo.

2.4 **Zonas climáticas de Costa Rica**

En este apartado se comentará sobre las zonas climáticas de Costa Rica, basado en [20].

Costa Rica posee una distribución variada de montañas, valles y llanuras que definen las condiciones climatológicas del país. Los sistemas montañosos y los vientos provenientes del noreste (alisios) dividen al país en tres regiones climáticas: la región tropical húmeda del Atlántico, ubicada en el norte y en el atlántico del país, la región central intermontana, ubicada en el valle central, y la región tropical del Pacífico, ubicada en el Pacífico norte, central y sur.

Considerando las características físico-geográficas del país y las regiones climáticas, Costa Rica se divide en siete regiones geográfico-climáticas con sus subregiones. Esta clasificación se basa en los trabajos de investigación realizados por Coen (1967), la cual toma en cuenta los siguientes aspectos para la clasificación:

- Provincias térmicas: Se basa en la temperatura media anual (tma). Microtermal, tma menor a 10°C. Mesotermal, tma entre 10°C a 22°C. Megatermal, tma mayor a 22°C.
- Provincias pluviométricas: Se basa en los totales de lluvia anual (tlla). Moderadamente lluviosa, tlla entre 1000 y 2000 milímetros. Lluviosa, tlla entre 2000 y 4000 milímetros. Excesivamente lluviosa, tlla superior a 4000 milímetros.
- Régimen de lluvia anual: Zonas donde todos los meses hay clima con estación seca, clima estacional o clima lluvioso.

2.4.1 Regiones geográfico-climáticas

- Región Pacífico Norte: Comprende la provincia de Guanacaste y los cantones de Esparza y Montes de Oro de la provincia de Puntarenas; y los cantones de Orotina y San Mateo de la provincia de Alajuela. Se encuentra en la ladera opuesta a los vientos alisios de la Cordillera Volcánica Norte. Se subdivide en: Subregión Occidental de la Península de Nicoya, Subregión Central del Pacífico Norte, Subregión base y faldas de las cordilleras de Guanacaste y Tilarán y Subregión cuenca baja de los ríos Barranca y Grande de Tárcoles.
- Región Pacífico Central: Comprende el poblado de Tinamaste, Valle de Parrita, Quepos y Manuel Antonio. La Fila Costera impide la llegada de los vientos alisios. Se subdivide en: Subregión Valle de Parrita, Subregión Cuenca del Río Naranjo, Subregión Cuenca del Río Barú.
- Región Pacífico Sur: Comprende todo el Valle del General, la Península de Osa, Valle de Coto Colorado, Valle de Coto Brus y Golfito. Se subdivide en: Subregión Valle del General y de Coto Brus, Subregión Valle de Diquís, Subregión Valle Coto Colorado, Subregión Península de Osa y Subregión faldas del Pacífico de la Cordillera de Talamanca.
- Región Montañosa del Sur: ubicada al sur del Valle Intermontano Central y al norte del Pacífico Central. Se subdivide en: Subregión cuenca Alta de los Ríos Turrubares y Tulín, y el Valle de Candelaria y Subregión cuenca Alta del Río Pirrís.
- Región Valle Intermontano Central: Comprende de los Montes del Aguacate pasando por el Alto de Ochomogo hasta Juan Viñas, además de la Cordillera Volcánica Central. Se subdivide en: Subregión Valle Central Occidental, Subregión Valle Central Oriental y Subregión Faldas de la Cordillera Volcánica Central.
- Región Norte: Entre la Cordillera Volcánica Central, la Cordillera de Guanacaste, la Cordillera de Tilarán y el Río Chirripó. Se subdivide en: Subregión faldas Orientales de las Cordilleras de Guanacaste y de Tilarán, Subregión faldas del Norte de la Cordillera Volcánica Central, Subregión llanuras de los Guatusos, Subregión llanuras de San Carlos y Subregión llanura de Sarapiquí.
- Región Atlántica: Comprende toda la provincia de Limón y la parte oriental de la provincia de Cartago. Se subdivide en: Subregión cuenca de los ríos Macho, Grande de Orosi y Pejibaye, Subregión faldas del Atlántico del Macizo Irazú-Turrialba y de la Cordillera de Talamanca - Valle del Reventazón, Subregión Llanura de Tortuguero, Subregión Llanuras de Santa Clara y Matina, y Cuenca del Río Banano y Subregión al Sur de la Cuenca de Río Banano.

2.5 *Desarrollo de software*

En este apartado se explicarán los conceptos de los modelos de desarrollo de *software* basados en [21].

2.5.1 *Ciclo de vida de un software*

El ciclo de vida de un *software* es el conjunto de fases que este atraviesa desde que se plantea hasta que se deja de utilizar. Su función es determinar el orden de las fases del desarrollo, establecer los criterios para la transición entre fases, definir los objetivos de cada fase, describir las actividades y el esquema que se utilizarán como base para planificar, organizar, coordinar y desarrollar el *software*.

El concepto de fase dentro del ciclo de vida de un *software* se refiere al conjunto de actividades necesarias para cumplir con un objetivo del proyecto. Las fases además pueden tener entregables como requisito de finalización, y un entregable es un producto intermedio o final que permite evaluar el progreso del proyecto. La cantidad de fases y entregables dependen del modelo de desarrollo de *software* escogido.

2.5.2 *Modelos de desarrollo de software*

Un modelo de desarrollo de *software* es el encargado de definir el ciclo de vida de un proyecto de *software*, en él se describen las fases principales del proceso de desarrollo y define las actividades dentro de cada fase.

Modelo en cascada: se tiene un orden riguroso de las fases del proyecto, estas se da de manera secuencial, sin posibilidad de regresar a fases anteriores. Estas fases son: especificación de requerimientos, diseño, implementación, pruebas, instalación y mantenimiento. Este modelo es fácil de implementar pero se debe utilizar en proyectos estables y predecibles, debido a que su estructura rigurosa no permite cambios inesperados.

Modelo en V: se propone que cada fase del proyecto tenga su propia fase de verificación y validación. Estas fases son: especificación de requerimientos con validación del sistema, diseño del sistema con verificación del sistema, diseño del *software* con verificación del *software* y codificación. Este modelo también es fácil de implementar y tiene mejores resultados que el modelo de cascada, pero al igual que este no permite cambios inesperados en fases avanzadas del proyecto.

Modelo iterativo: consiste en utilizar varias veces el modelo en cascada, se presentan versiones del producto al cliente y se corrigen los errores en cada nueva iteración. Este modelo solventa los problemas de rigidez de los modelos anteriores pero presenta problemas al definir el alcance del proyecto, los requerimientos no están realmente definidos en el inicio del proyecto.

Modelo de desarrollo incremental: se basa en la construcción de prototipos, al igual que en el modelo iterativo se utiliza el modelo en cascada de forma repetitiva. En este modelo se requiere la generación de *software* funcional en las primeras etapas del proyecto, por lo que no es fácil de implementar, además los requerimientos tampoco están definidos en su totalidad al inicio del proyecto.

Modelo en espiral: se definen ciclos con cuatro fases cada uno, estas fases son: fijar objetivos, análisis de riesgo, desarrollo y validación y planificación. En el análisis de riesgo se analizan riesgos potenciales y se proponen soluciones a estos, en la planificación se hace un recuento de las actividades realizadas y se planifican las actividades siguientes. Este modelo no es fácil de implementar y genera una alta cantidad de trabajo adicional, pero reduce los riesgos del proyecto, mejora la calidad y se tiene un desarrollo junto con un mantenimiento del *software*.

En el apartado 3.1 se explica cómo se desarrollaron los modelos de cascada y en espiral para la realización del proyecto.

2.6 Herramientas de Software

2.6.1 OpenGL

OpenGL es un entorno de desarrollo portable para aplicaciones gráficas interactivas en 2D y 3D. Fomenta la innovación y acelera el desarrollo de aplicaciones mediante la incorporación de un amplio conjunto de funciones de renderizado, mapeado de texturas y efectos especiales. Proporciona a los desarrolladores de *software* el acceso a primitivas geométricas e imágenes, listas de visualización, transformaciones de modelado, iluminación y texturizado, “anti-aliasing”, mezclas, y muchas otras características [22].

El especificación estándar de OpenGL tiene enlaces de lenguaje para C, C ++, Fortran, Ada y Java.

2.6.2 OpenSceneGraph

OpenSceneGraph es un conjunto de herramientas de alto rendimiento para gráficos 3D, utilizado en aplicaciones de simulación visual, juegos, realidad virtual, visualización científica y modelado. Escrito en C++ y OpenGL, es de código abierto y multiplataforma, funciona en Windows, OSX, GNU / Linux, IRIX, Solaris, HP-UX, AIX, sistemas operativos FreeBSD y teléfonos móviles y tabletas que usen OpenGL [23].

2.6.3 OsgEarth

OsgEarth es un kit de desarrollo de *software* geoespacial, escrito en C++. Utiliza OpenSceneGraph para la visualización de mapas y datos geoespaciales [24].

2.6.4 WebGL

WebGL es una especificación estándar multiplataforma para interfaces de programación de aplicaciones (API), se utiliza para crear gráficos 3D en un navegador Web. Está basado en OpenGL por lo que ofrece familiaridad con la API de OpenGL. Tiene una estrecha integración con el contenido HTML, incluyendo la composición de capa, la interacción con otros elementos HTML, y el uso de los mecanismos de control de eventos estándar HTML. Posee aceleración de *hardware* de gráficos 3D para el entorno del navegador y un entorno de programación que simplifica la creación de prototipos 3D [25].

Algunas bibliotecas que usan WebGL son las siguientes:

- three.js: Biblioteca liviana y sencilla para creación de gráficos en 3D [26].
- OSGJS: basado en OpenSceneGraph, permite usar una caja de herramientas similar a la de OpenSceneGraph para interactuar con WebGL a través de JavaScript [27].
- GLGE: Permite el acceso directo a OpenGL y el uso de aceleración de gráficos [28].
- SceneJS: motor de visualización 3D [29].
- SpiderGL: Biblioteca de JavaScript para desarrollar aplicaciones de gráficas con renderizado en tiempo real [30].

Además de estas bibliotecas, están las que también integran funcionalidades geoespaciales como:

- OpenWebGlobe: kit de desarrollo de *software* para crear aplicaciones en un globo virtual. Puede procesar gran cantidad de datos geoespaciales [31].
- Glob3m: motor de globo virtual 3D, multiplataforma, disponible en dispositivos móviles y en los navegadores web HTML5 [32].
- WebGLEarth: API JavaScript de globo virtual realizado con la tecnología HTML5 y el canvas de WebGL [33].

2.6.5 Data-Driven Documents

Data-Driven Documents también conocido como D3.js, es una biblioteca de JavaScript para manipular documentos basados en datos, ayuda a llevar los datos a HTML, SVG y CSS. A diferencia de las bibliotecas anteriores, D3.js no está basado en OpenGL y se caracteriza por enlazar datos al modelo de objetos del documento (DOM por sus siglas en inglés, es un conjunto estándar de objetos para representar documentos de HTML y XML). El objetivo principal de D3.js es tener una manipulación eficiente de documentos basados en datos, además busca ser rápido y soportar gran cantidad de datos y comportamientos para la interacción y animación [34].

2.6.6 Web workers

Un *web worker* es un código en JavaScript que se ejecuta en segundo plano, de forma independiente a otros comandos, sin afectar el rendimiento de la página. Al ejecutar comandos en una página HTML, la página no responde hasta que finalice la secuencia de comandos. Se puede seguir haciendo acciones en la página mientras que el *web worker* se ejecuta en segundo plano [35].

2.6.7 Parallel.js

Parallel.js es una biblioteca para el procesamiento multi-núcleo en JavaScript. Se creó con la intención de solventar el hecho de que JavaScript funciona con un solo hilo y que la tecnología en las computadoras avanza incrementando el número de núcleos rápidamente. Parallel.js da acceso al procesamiento multi-núcleo utilizando *web workers* [36].

2.6.8 Semantic UI

Semantic UI es una plataforma de trabajo que utiliza HTML, JavaScript y CSV para crear páginas web de manera rápida y sencilla. Utiliza una sintaxis similar al lenguaje natural como sustantivos, adjetivos, orden de palabras y pluralidad para unir conceptos intuitivamente [37].

2.6.9 GRASS GIS

GRASS GIS (por sus siglas en inglés *Geographic Resources Analysis Support System* y *Geographic Information System*) es sistema de información geográfica gratis y de código abierto, usado para el manejo de datos geoespaciales, análisis, procesamiento de imágenes, producción de mapas, modelado espacial y visualización [38].

2.6.10 QGIS

QGIS es un sistema de información geográfica de código abierto usado para visualizar, gestionar, editar y analizar datos geoespaciales y diseñar mapas imprimibles [39]. QGIS tiene funciones similares a GRASS GIS pero estas están como opciones en la interfaz, en GRASS GIS las funciones se realizan por medio de comandos, siendo este más complicado de utilizar. Además en QGIS se puede instalar un complemento en el cual se pueden añadir las funciones por comandos de GRASS GIS.

3 Desarrollo metodológico

3.1 Desarrollo de software

Para el desarrollo del proyecto, se utilizó un modelo de desarrollo de *software* en cascada para la parte del algoritmo de interpolación y un modelo en espiral para la parte de la animación. En la Figura 3.1 se observa el diagrama de flujo general de la aplicación.

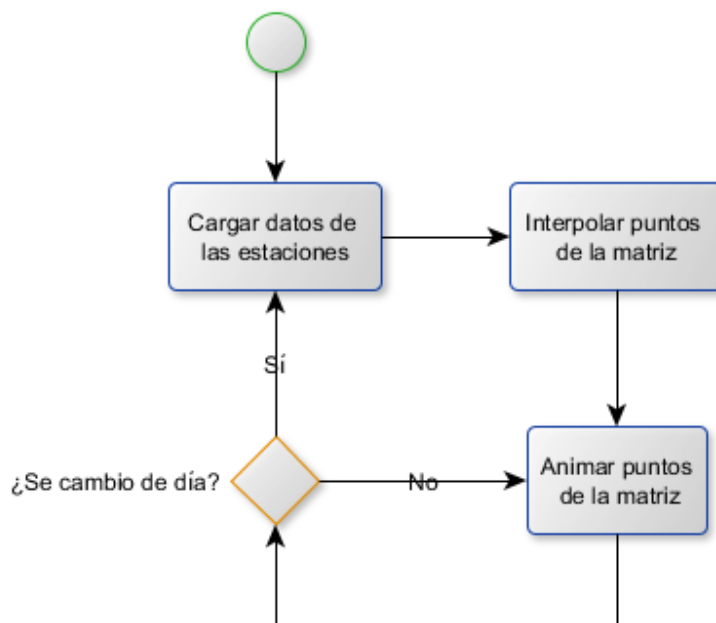


Figura 3.1. Diagrama de flujo general de la aplicación de Mapa de vientos de Costa Rica

El desarrollo de los dos primeros bloques, carga de datos e interpolación, se desarrolló con un modelo en cascada, donde se tomaron los requerimientos, se propuso un diseño, se desarrolló el algoritmo y finalmente se hicieron pruebas. En el apartado 3.1 se explica el proceso de desarrollo del algoritmo de interpolación y en el apartado 3.2 se explica el proceso de carga de datos.

Para el bloque de animación no se tenían claros los requerimientos necesarios para obtener una aplicación que satisficiera al cliente, por lo que se optó por un modelo en espiral. En el modelo en espiral primero se fijan los objetivos, luego se analizan los riesgos de estos objetivos, se desarrollan y finalmente se planifica una siguiente iteración. Utilizar el modelo en espiral permitió empezar el desarrollo del proyecto sin tener una visión clara del producto final. En cada iteración se desarrollaba una versión de la aplicación y, con ayuda del cliente, se planificaban nuevos objetivos hasta alcanzar un punto óptimo de satisfacción.

Con el modelo de espiral se obtuvieron cinco versiones de la aplicación, los cambios de versión se dan al variar totalmente el concepto de la animación o al cambiar el uso de una tecnología. Entre versiones también se dan iteraciones para ajustes de rendimiento y de interfaz. La primera versión de la aplicación se realizó con la biblioteca Three.js, en el apartado 4.1.1 se abarcan los detalles de esta animación. Para las tres versiones siguientes se cambió a la biblioteca D3.js y se diferencian por el tipo de animación, animación sincrónica (apartado 3.5.1), animación por transiciones (apartado 3.5.2) y animación asincrónica (3.5.3). En la última versión se cambia la tecnología de objetos SVG a canvas (apartado 3.5.5), obteniendo finalmente una aplicación a satisfacción del cliente.

3.2 Interpolación

Considerando los algoritmos de interpolación espacial más frecuentemente utilizados, vecino natural, distancia inversa ponderada y Kriging, se escogió el algoritmo interpolación basado en los siguientes criterios. Respecto al algoritmo de Kriging, este proyecto contempló realizar una visualización de datos y no una simulación, para poder realizar una simulación se necesitaría un modelo de distribución de vientos de Costa Rica, el algoritmo de Kriging necesita este modelo estadístico de datos para funcionar, por lo que este algoritmo se descartó al salirse del ámbito del proyecto. El algoritmo de vecinos naturales utiliza un fundamento parecido al del distancia inversa ponderada pero conlleva un cálculo adicional de áreas en los puntos conocidos, por lo que aumenta en complejidad respecto al algoritmo de distancia inversa ponderada que se basa únicamente en la distancia entre los puntos.

3.2.1 Distancia inversa ponderada

Se escogió como algoritmo de interpolación el algoritmo de distancia inversa ponderada, este se desarrolló y probó inicialmente en C++ para luego ser trasladado a JavaScript donde se realizó la aplicación final. El algoritmo consta de una matriz de puntos donde se desplegarán los datos de velocidad y dirección de viento de cada día del 2014. Como datos iniciales se tienen los datos de veinte estaciones distribuidas a lo largo del país (ver Tabla 3.2).

En la Figura 3.2 se observa el diagrama de flujo del algoritmo de interpolación por distancia inversa ponderada.

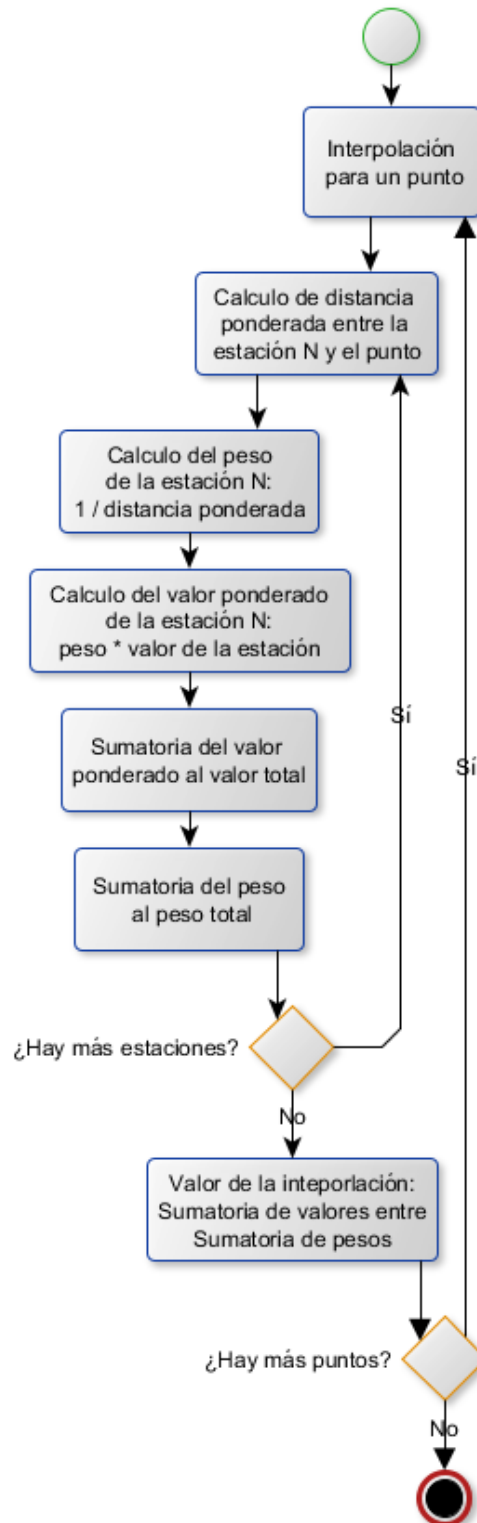


Figura 3.2. Diagrama de flujo del algoritmo de interpolación

3.3 Datos iniciales

Para la realización del algoritmo de interpolación se utilizaron, como datos iniciales, los datos entregados por el Instituto Meteorológico Nacional.

El Instituto Meteorológico Nacional entregó datos diarios de rapidez y dirección de viento en Costa Rica del año 2014, de treinta y cuatro estaciones meteorológicas repartidas a lo largo del país. Las estaciones meteorológicas se listan en la Tabla 3.1, esta tabla contiene el número de la estación, el nombre, la latitud norte, la longitud oeste y la altitud en metros sobre el nivel de mar.

Tabla 3.1. Datos de estaciones meteorológicas

Número	Estación	Latitud Norte	Longitud Oeste	Altitud (m.s.n.m)
69679	Upala, Aut.	10° 52' 51"	85° 04' 21"	60
69727	Adifort, La Fortuna San Carlos, Aut	10° 28' 04"	84° 38' 50"	266
72149	San José, Pinilla, Aut.	10° 15' 36"	85° 50' 16"	15
72161	Santa Rosa, Gte. Aut.	10° 50' 00"	85° 37'00"	285
72163	Santa Rosa, Cc. Aut.	10° 50' 28"	85° 37'10"	315
73123	ITCR, Cartago, Aut.	09° 51' 08"	83° 54' 31"	1360
73137	Volcán Irazú, Aut.	09° 58' 08"	83° 50' 21"	3331
73143	Volcán Turrialba, Aut	10° 01' 09"	83° 45' 23"	3343
73145	Earth, Cc, Aut.	10° 12' 35"	83° 35' 39"	30
73147	Guápiles, Aut.	10° 12' 41,8"	83° 46' 18"	253
73149	Maravilla , Juan Viñas, Aut	09° 53' 46"	83° 44' 53"	1181
74051	Aerop. Liberia Oeste 07, Aut.	10° 35' 20,4"	85° 33' 07,7"	89
74061	El Corral, Palo Verde Cc. Aut.	10° 20' 51"	85° 21' 04"	10
74063	Mangarica, Llano La Cruz, Liberia, Aut.	10° 36' 20,7"	85° 27' 09"	135
81005	Aerop. Limón, Aut.	09° 57' 44"	83° 01' 29"	5
84139	CIGEFI, Aut.	09° 56' 11"	84° 02' 43"	1210
84141	IMN, Aranjuez, Aut.	09° 56' 17"	84° 04' 11"	1181
84169	Aerop. Juan Santamaria Media Pista, Aut.	09° 59' 28"	84° 10' 52"	913
84181	Iztaru, Aut.	09° 53' 22"	83° 58' 08"	1680
84187	Fabio Baudrit, Aut.	10° 00' 18"	84° 15' 56"	840
84195	Aerop. Pavas Oeste, Aut.	09° 57' 23,68	84° 08' 17,29	997
84197	Santa Bárbara, Aut.	10° 02' 00"	84° 09' 57"	1070
84203	Zapote, San José, Aut.	09° 55' 22"	84° 03' 33"	1192
84209	Universidad Para La Paz, Cc, Aut.	09° 55' 13"	84° 16' 22"	825
84215	Cerro Burio, Aserrí, Aut.	09° 50' 25,3"	84° 06' 45,6"	1811
84219	Cerro Chitaria, Santa Ana, Aut.	09° 53' 30,1"	84° 11' 37,3"	1717
84231	Cerro Cedral, Escazú, Aut.	09° 51' 41,1"	84° 08' 45,2"	2255
84237	Ciudad Judicial, San Joaquín De Flores, Aut.	10° 00' 24,57	84° 09' 50,17	1024
85023	Manzanillo, Aut.	09° 37' 53,25	82° 39' 39,41	80
88047	La Lucha 2, Aut	09° 44' 12"	84° 00' 02"	1740
94013	Cerro Buenavista, Cc. Aut	09° 33' 36"	83° 45' 13"	3400
98095	Altamira, Aut.	09° 01' 45"	83° 00' 28"	1370

En la apéndice 1 se muestra un ejemplo del formato de los datos en cada estación, se tiene el año, el mes, el día, la dirección predominante durante el día y la velocidad en metros por segundo.

Además, en el apéndice 1 se muestra el mes de septiembre de la estación 72163, donde se puede observar que en los días 27 y 28 el dato de la velocidad del viento no existe, esta falta de datos de rapidez y dirección del viento se puede encontrar en la mayoría de estaciones.

Para solucionar esta falta de datos en las estaciones, como primer paso se escogieron las veinte estaciones con la mayor cantidad de datos, en la Tabla 3.2 se puede observar las estaciones seleccionadas.

Tabla 3.2. Estaciones seleccionadas para la realización del proyecto

Número	Estación	Latitud Norte	Longitud Oeste	Altitud (m.s.n.m)
69679	Upala, Aut.	10° 52' 51"	85° 04' 21"	60
72149	San José, Pinilla, Aut.	10° 15' 36"	85° 50' 16"	15
73137	Volcán Irazú, Aut.	09° 58' 08"	83° 50' 21"	3331
73145	Earth, Cc, Aut.	10° 12' 35"	83° 35' 39"	30
73147	Guápiles, Aut.	10° 12' 41,8"	83° 46' 18"	253
73149	Maravilla , Juan Viñas, Aut	09° 53' 46"	83° 44' 53"	1181
74051	Aerop. Liberia Oeste 07, Aut.	10° 35' 20,4"	85° 33' 07,7"	89
74061	El Corral, Palo Verde Cc. Aut.	10° 20' 51"	85° 21' 04"	10
84139	CIGEFI, Aut.	09° 56' 11"	84° 02' 43"	1210
84141	IMN, Aranjuez, Aut.	09° 56' 17"	84° 04' 11"	1181
84169	Aerop. Juan Santamaria Media Pista, Aut.	09° 59' 28"	84° 10' 52"	913
84187	Fabio Baudrit, Aut.	10° 00' 18"	84° 15' 56"	840
84195	Aerop. Pavas Oeste, Aut.	09° 57' 23,68	84° 08' 17,29	997
84197	Santa Bárbara, Aut.	10° 02' 00"	84° 09' 57"	1070
84203	Zapote, San José, Aut.	09° 55' 22"	84° 03' 33"	1192
84215	Cerro Burío, Aserrí, Aut.	09° 50' 25,3"	84° 06' 45,6"	1811
84219	Cerro Chitaría, Santa Ana, Aut.	09° 53' 30,1"	84° 11' 37,3"	1717
84231	Cerro Cedral, Escazú, Aut.	09° 51' 41,1"	84° 08' 45,2"	2255
94013	Cerro Buenavista, Cc. Aut	09° 33' 36"	83° 45' 13"	3400
98095	Altamira, Aut.	09° 01' 45"	83° 00' 28"	1370

Como segundo paso se aplicó el algoritmo de interpolación de distancia inversa ponderada descrito en el apartado 3.1 para obtener los datos faltantes de la velocidad del viento. En el caso de la dirección del viento, se utilizó la función de moda para obtener la dirección predominante durante todo el año, y esta fue utilizada para darle dirección a los datos faltantes.

En el tercer paso se convirtieron las direcciones en puntos cardinales a grados, como se observa en la Tabla 3.3.

Tabla 3.3. Conversión de punto cardinal a grados

Punto cardinal	Grados (°)
Este	0
Noreste	45
Norte	90
Noroeste	135
Oeste	180
Suroeste	225
Sur	270
Sureste	315

El cuarto paso fue pasar los datos al formato JSON para utilizarlos de manera sencilla en la aplicación web. Para realizar este proceso se utilizó la aplicación web Mr. Data Converter [40]. la cual transforma datos con formato de tabla en datos con formato JSON. Los datos en formato JSON se clasificaron por estación y se añadieron a la aplicación.

3.4 *Parallel.js*

Parallel.js es una biblioteca de JavaScript que utiliza *web workers* para el manejo de múltiples núcleos.

La implementación de Parallel.js se realizó con el fin de mejorar el rendimiento de la aplicación. En el apartado 1.5 se expresa cómo la capacidad de procesamiento es una limitante del proyecto.

Se investigó la implementación de Parallel.js utilizando una computadora con cuatro núcleos en el código de la interpolación, sin embargo los *web workers* de esta herramienta no tienen acceso a los objetos del DOM, por lo que no se puede generar una animación en paralelo.

3.5 D3.js

La biblioteca D3.js posee facilidades para el manejo y animación de datos, además de la capacidad de cargar capas de mapas de forma sencilla y con un formato ligero como lo es TopoJSON.

3.5.1 Animación sincrónica

La primera animación en D3.js fue una animación sincrónica. Para esta animación se utilizaron líneas que se mueven a una distancia variable, hacia una dirección variable. Las líneas son blancas y de un pixel de ancho, estas líneas toman los datos de la matriz creada por la interpolación, colocando la misma cantidad de líneas como datos. Se utilizan los valores de velocidad en m/s para determinar el largo de la línea, cuanto más velocidad más larga es la línea. Se utilizan los valores de dirección para calcular la posición final de la línea, el ángulo se pasa a radianes (las funciones matemáticas de JavaScript utilizan radianes), se calculan los valores para el eje horizontal (x) y vertical (y) utilizando (2.9). Las líneas se colocaron a quince píxeles entre ellas, en una matriz de 30x30, y se hace una transición de un segundo donde las líneas se extienden de un punto hasta la posición calculada con los valores de la interpolación. Seguidamente se genera una segunda transición de medio segundo donde la línea se hace transparente y desaparece. Al terminar la animación esta se repite recursivamente. Para la realización de esta animación se tomó como referencia la animación de P. Cook (ver Anexo 1).

Esta animación es sincrónica porque todas las líneas aparecen y desaparecen al mismo tiempo sin importar la velocidad del viento. La dirección del viento se percibe con las líneas moviéndose hacia la dirección calculada y la velocidad del viento se percibe con el largo de las líneas. En la Figura 3.3 se muestra el diagrama de flujo del algoritmo de animación para la animación sincrónica, por transiciones y asincrónica.

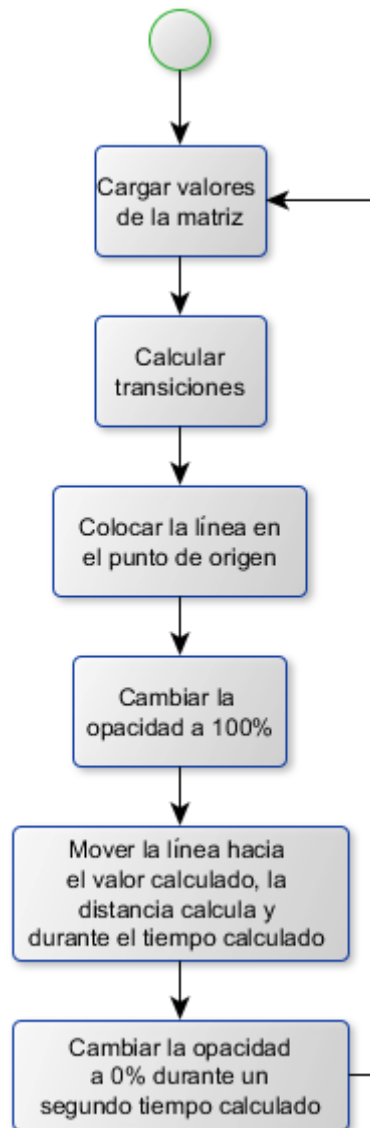


Figura 3.3. Diagrama de flujo de la animación

3.5.2 Animación por transiciones

En la animación de transiciones se utilizan los mismos principios que en la animación sincrónica, con la diferencia de que las líneas no tienen ciclos recursivos. Se tiene una matriz de 60x60 con siete píxeles entre las líneas, se hace una transición de un segundo donde las líneas se extienden hasta la posición calculada con los valores de la interpolación. La segunda transición no se ejecuta en esta animación, las líneas quedan estáticas hasta que se cambie el día, al cambiar el día los valores de la interpolación cambian y las líneas desde su posición actual hacen la transición al nuevo punto.

Se añadieron flechas para mejorar la percepción de dirección de las líneas, al no moverse constantemente la dirección no siempre es clara. La percepción de la velocidad se mejoró cambiando el ancho de las líneas, a mayor velocidad las líneas son más anchas y largas.

3.5.3 Animación asincrónica

En la animación asincrónica se retoma la animación constante presentada en la animación sincrónica. De igual manera se utilizan los valores de velocidad en m/s para determinar el largo y ancho de la línea, cuanto más velocidad, más larga y ancha es la línea. Se utilizan los valores de dirección para calcular la posición final de la línea. Las líneas se colocaron a once píxeles entre ellas, en una matriz de 40x40, y se hace una transición de tiempo variable donde las líneas se extienden de un punto hasta la posición calculada con los valores de la interpolación. Seguidamente se genera una segunda transición de tiempo variable donde la línea se hace transparente y desaparece. Al terminar la animación esta se repite recursivamente.

Los tiempos variables son la clave para animación asincrónica, en (3.1) se muestra la fórmula para calcular el tiempo de la primera transición y en (3.2) la fórmula para calcular el tiempo de la segunda transición.

$$\text{Duración } A = 1000 * 1.2833 - v * 33 \quad (3.1)$$

$$\text{Duración } B = 500 * 2.066 - v * 33 \quad (3.2)$$

Donde v es la velocidad calculada de la interpolación.

En la Tabla 3.4 se muestran los resultados de (3.1) para el rango de velocidades de los vientos en Costa Rica. Este rango fue extraído de los datos de las estaciones entregadas por el Instituto Meteorológico Nacional.

Tabla 3.4. Duración de la primera transición según la velocidad del viento

Velocidad (m/s)	Duración (ms)
0	1283
1	1250
2	1217
3	1184
4	1151
5	1118
6	1085
7	1052
8	1019
9	986
10	953
11	920
12	887
13	854
14	821
15	788
16	755

En la Tabla 3.5 se muestran los resultados de (3.2) para el rango de velocidades de los vientos en Costa Rica.

Tabla 3.5. Duración de la segunda transición según la velocidad del viento

Velocidad (m/s)	Duración (ms)
0	1033
1	1000
2	967
3	934
4	901
5	868
6	835
7	802
8	769
9	736
10	703
11	670
12	637
13	604
14	571
15	538
16	505

De las Tablas 3.4 y 3.5 se puede observar que una línea con una velocidad de 16 m/s dura 1260 ms completando un ciclo, mientras que una línea con una velocidad de 1 m/s dura 2250 ms para completar un ciclo. Este efecto asincrónico mejora la percepción de la velocidad, del flujo y de los patrones del viento.

La decisión de utilizar los valores de (3.1) y (3.2) fue tomada por el cliente luego de presentarle diferentes tiempos para las transiciones de las líneas. En la Tabla 3.6 se muestran los rangos de tiempo propuestos para las transiciones de las líneas.

Tabla 3.6. Rangos de tiempos para las transiciones de las líneas

Prueba	Tiempo mínimo primer transición	Tiempo máximo primero transición	Tiempo mínimo segunda transición	Tiempo máximo segunda transición
1	505	1033	255	783
2	755	1283	505	1033
3	1755	2283	755	1283
4	755	1283	755	1283
5	505	1033	755	1283

El cliente señaló que la prueba 2 de la Tabla 3.6 ofrecía la mejor percepción de las corrientes de vientos, las pruebas siguientes generaban lapsos donde el viento no se encontraba presente y en la prueba 1 el viento no se logra apreciar por la rapidez de la transición.

3.5.4 Líneas adicionales

Para mejorar la percepción del flujo y de los patrones del viento, se decidió agregar líneas dinámicas dependiendo de la velocidad del viento. Las líneas adicionales duplican a la línea original y se colocan diagonalmente. En la Figura 3.4 se muestra el diagrama de flujo de la animación con líneas adicionales.

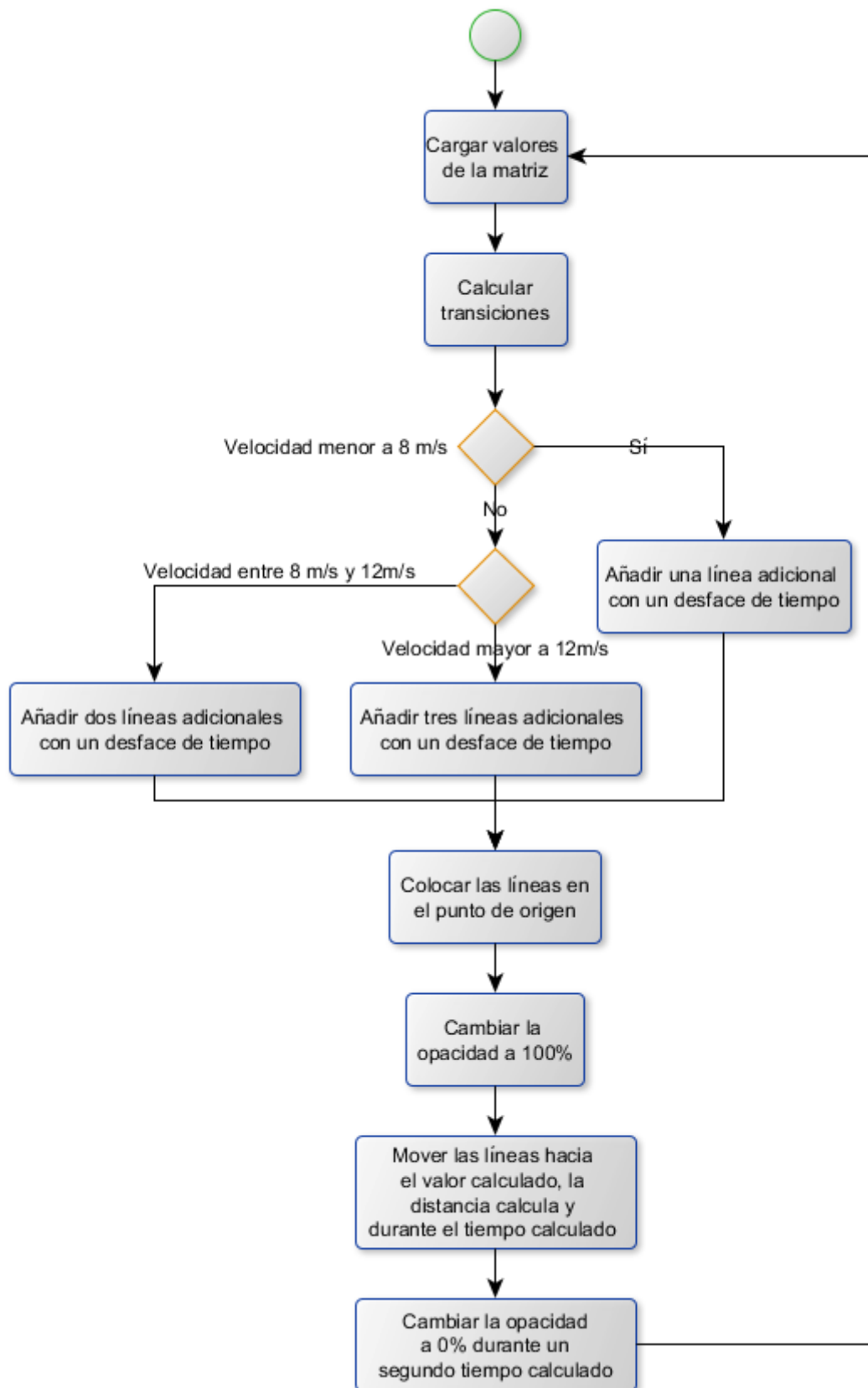


Figura 3.4. Diagrama de flujo de la animación con líneas adicionales

3.5.5 Canvas

La biblioteca D3.js trabaja comúnmente con objetos SVG, los objetos SVG generan una gran carga en términos de rendimiento. Con una matriz de 40x40 se tienen 1600 objetos SVG por lo que el rendimiento se ve afectado. Al utilizar canvas no se tienen objetos, en vez de esto se dibujan las figuras deseadas en la aplicación. Para la realización de este cambio se tomó como referencia el código de M. Bostock (ver Anexo 2).

Con el aumento del rendimiento al utilizar el canvas, se añadió una línea adicional aun en velocidades menores a 4 m/s. Al todas las líneas tener al menos una línea adicional se mejora la percepción de fluidez del viento.

3.5.6 Acercamiento de imagen

Finalmente, para obtener una mejor percepción de la animación se requirió crear un método para acercar la imagen del mapa de vientos de Costa Rica. Al cambiar los objetos de SVG a canvas las capas de información geográfica no se pueden cargar, por lo que se cambió el mapa vectorial de Costa Rica por una imagen del mapa de Costa Rica.

Al acercar la imagen se cambia la separación, el largo y el ancho de las líneas, además se añadió una segunda línea adicional para todas las velocidades menores a 12 m/s. Esta línea adicional se añadió para mejorar la percepción de densidad aun cuando se hiciera un acercamiento de imagen.

3.6 Interfaz con el usuario

La interfaz con el usuario fue creada con la biblioteca de Semantic UI. En la Figura 3.5 se puede observar la interfaz de usuario con sus secciones para la aplicación creada en D3.js con elementos SVG.

Mapa de vientos de Costa Rica

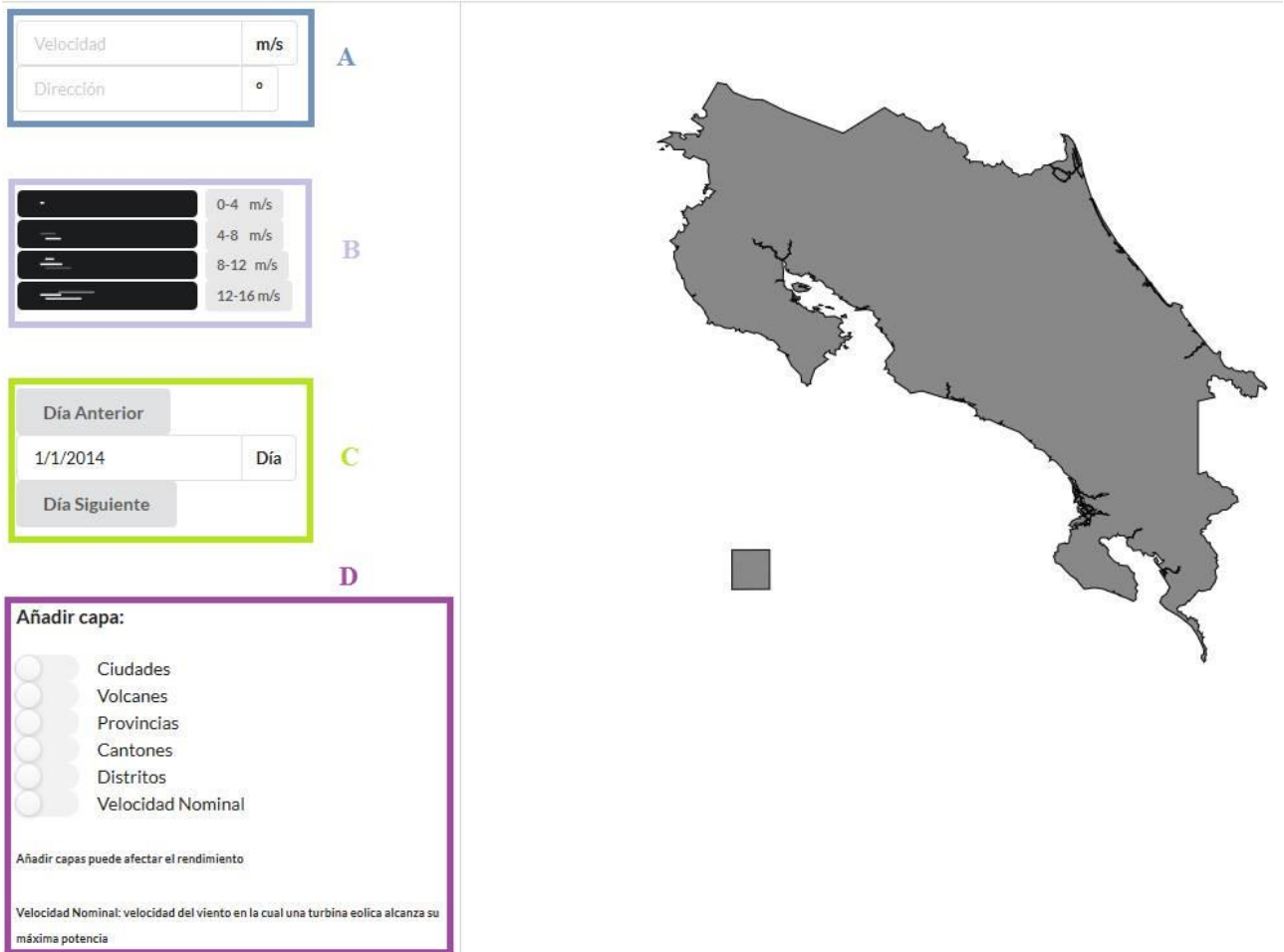


Figura 3.5. Secciones de interfaz de usuario creadas en Semantic UI para la aplicación web

Se añadió una cuadrícula para separar los diferentes paneles de la interfaz, el panel superior contiene el título de la aplicación, el panel izquierdo contiene ayudas visuales y opciones para animación y el panel derecho contiene la animación.

En la parte superior del panel izquierdo, en la sección A, se tienen dos campos de ayuda, estos son utilizados para visualizar la rapidez y la dirección del viento donde se encuentre la punta del ratón.

Debajo de los campos de rapidez y dirección de viento, en la sección B, se encuentra una escala, esta es utilizada para representar la intensidad del viento y está formada de cuatro elementos para diferenciar las diferentes intensidades.

En el panel izquierdo, en la sección C, también se encuentra un campo para indicar la fecha de los datos que actualmente están siendo visualizados. Se puede presionar el botón “Día Anterior” para retroceder un día, o presionar el botón “Día Siguiente” para avanzar un día.

Finalmente, en la sección D se encuentra un selector de capas, donde se pueden añadir capas adicionales al mapa de Costa Rica.

En la Figura 3.6 se puede observar la interfaz de usuario actualizada al cambiar elementos SVG por canvas.

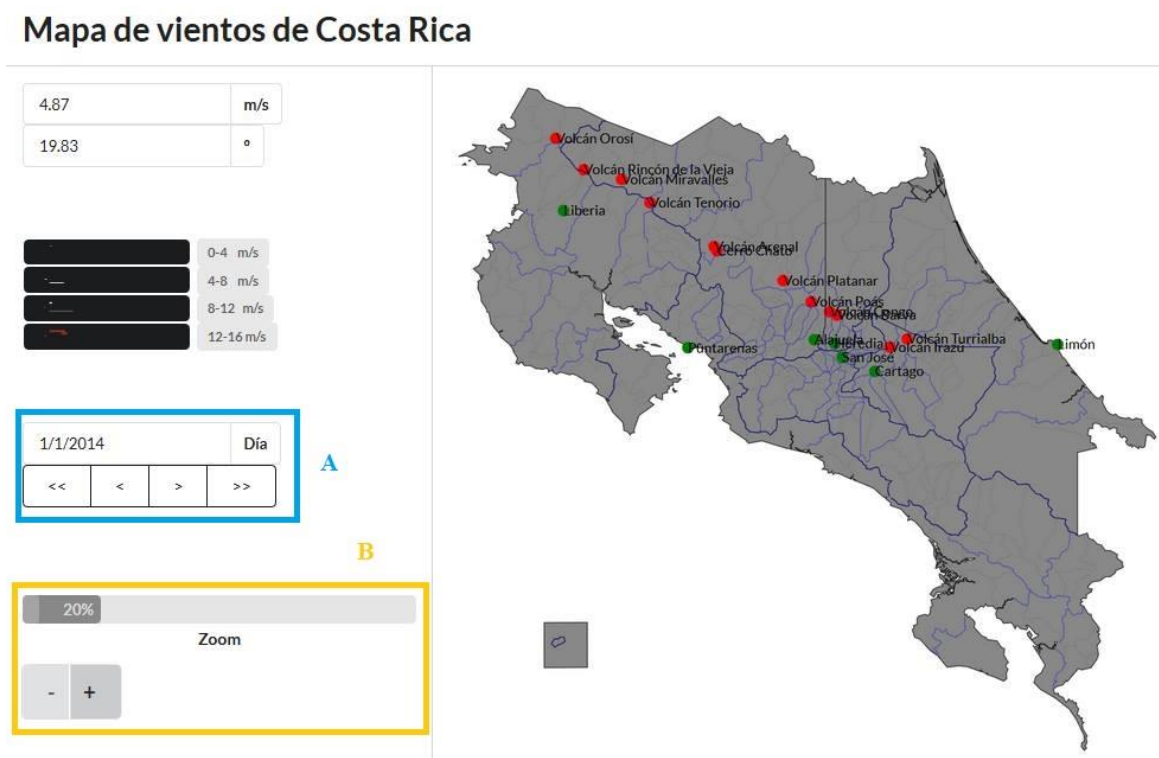


Figura 3.6. Secciones de interfaz de usuario creadas en Semantic UI para la aplicación web con canvas

En la sección A se puede observar que se cambiaron los controladores de cambio de día, de izquierda a derecha se tiene: retroceder 15 días, retroceder 1 día, avanzar 1 día y avanzar 15 días.

En la sección B se añadió un controlador para el acercamiento, el botón con el símbolo “+” acerca el mapa y el botón con el símbolo de “-” aleja el mapa.

Se puede observar que el selector de capas ya no se encuentra en esta versión de la aplicación, esto es debido al cambio de elementos SVG a canvas, donde estas capas ya no se pueden utilizar.

3.7 Topografía

El dominio de visualización de la aplicación abarca desde 82.50°O a 86.05°O y desde 7.91°N a 11.22°N, correspondientes a los 51 100 km² de la superficie de Costa Rica. Se utilizan datos de límites, provincias, cantones y distritos cargados como una capa vectorial transformados a TopoJSON para su utilización web. Estos datos se obtuvieron del profesor Armando Arce del Instituto Tecnológico de Costa Rica, de su repositorio en Github (Figura 3.7) [41].

Los valores de altura en Costa Rica no serán considerados para este proyecto.



Figura 3.7. Capa vectorial de límites, provincias, cantones y distritos de Costa Rica

3.7.1 Simplificación de geometrías

La gran cantidad de geometrías del mapa de Costa Rica afecta el rendimiento de la aplicación, por lo que se efectuó una simplificación de geometrías en las capas del mapa. La simplificación de geometrías reduce el detalle de los límites, cambiando geometrías puntiagudas por geometrías más lisas.

Para realizar la simplificación de geometrías se utilizó el programa QGIS, con este se cargó el archivo TopoJSON y se procedió a realizar el proceso de simplificación, las capas resultantes fueron guardadas como GeoJSON y transformadas a TopoJSON con la aplicación web GeoJSON-TopoJSON [42] para su utilización en el proyecto.

Las siguientes figuras representan un ejemplo de simplificación de geometrías.

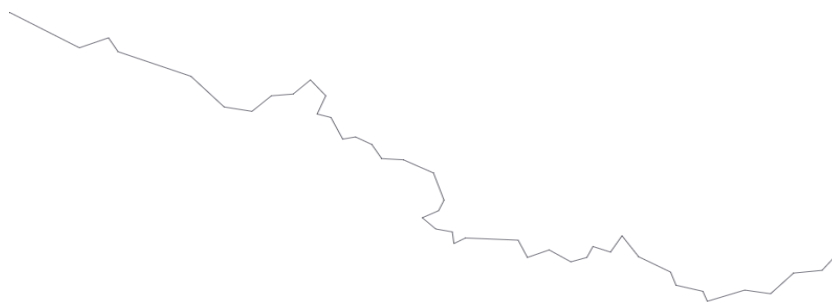


Figura 3.8. Geometría de parte del límite norte de Costa Rica sin simplificar

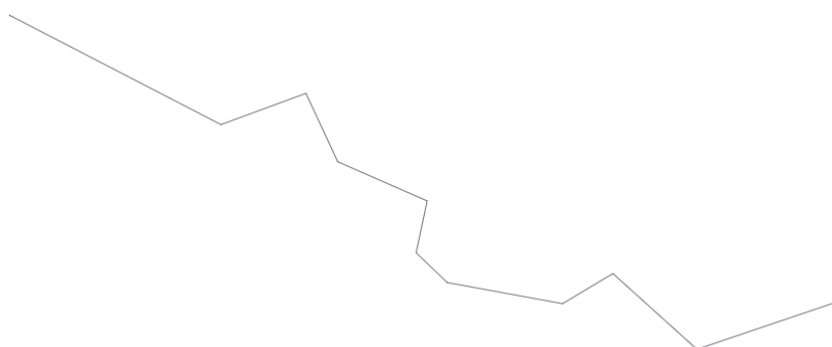


Figura 3.9. Geometría de parte del límite norte de Costa Rica simplificada

Se simplificaron las geometrías de la capa del límite exterior de Costa Rica, de la capa de provincias, de la capa de cantones y de la capa de distritos (ver apéndice 2).

3.7.2 Capas adicionales

Además de las capas de límites de Costa Rica se añadieron dos capas adicionales, estas capas son: capa de ciudades y capa de volcanes.

La capa de ciudades contiene puntos localizados en las capitales de las provincias de Costa Rica como se puede ver en la Figura 3.10. Esta capa, además de contener los puntos, contiene una etiqueta con el nombre de cada ciudad para ayudar en su identificación.

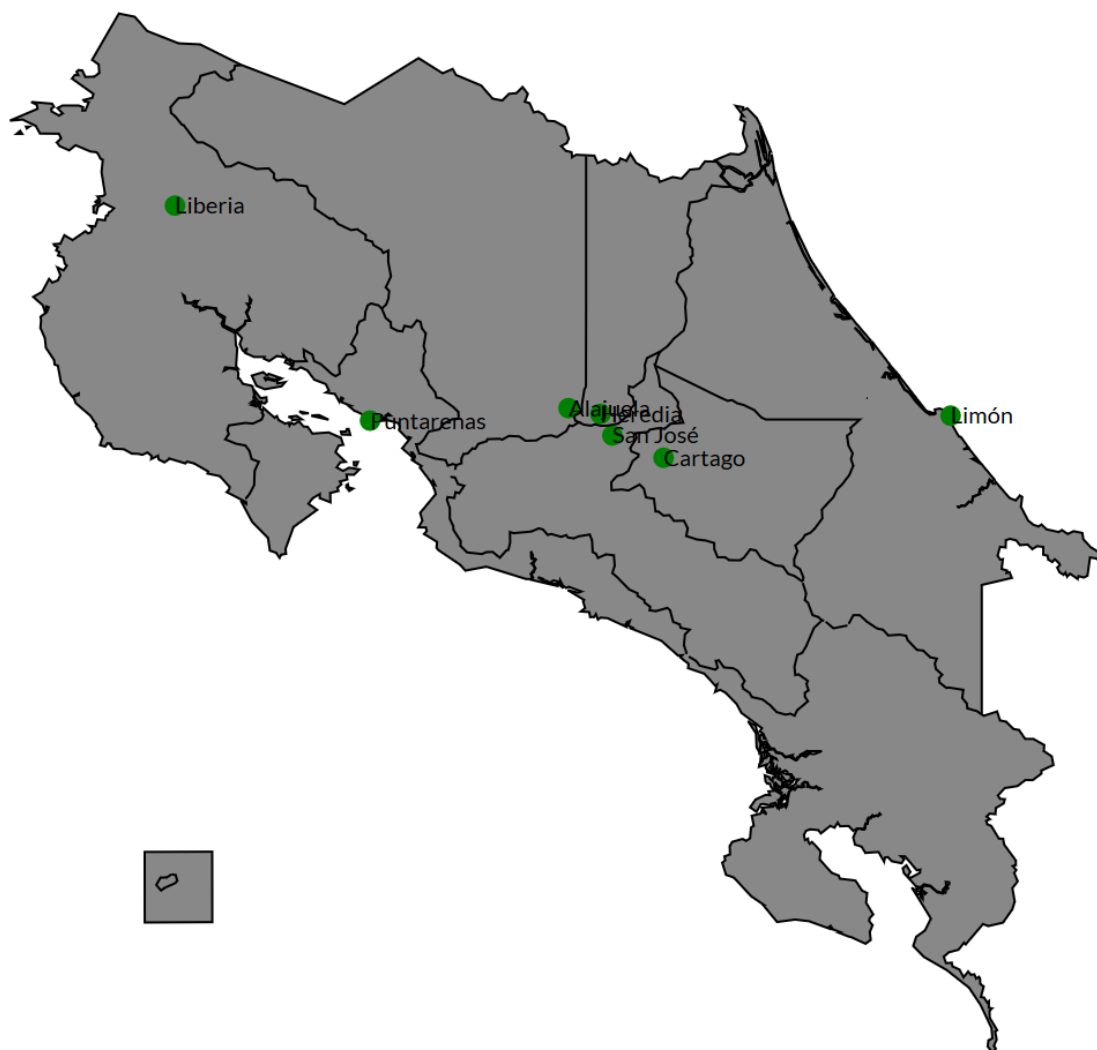


Figura 3.10. Capa de ciudades de Costa Rica

La capa de volcanes contiene puntos localizados en los volcanes de Costa Rica como se puede ver en la Figura 3.11. Esta capa, además de contener los puntos, contiene una etiqueta con el nombre de cada volcán para ayudar en su identificación.



Figura 3.11. Capa de volcanes de Costa Rica

4 Análisis de resultados

4.1 Interpolación por distancia inversa ponderada

4.1.1 Datos de prueba con Three.js

Como se explicó en el apartado 3.2, se utilizó el algoritmo de distancia inversa ponderada para la realización del proyecto. Inicialmente se implementó en JavaScript y se animó con Three.js. En Three.js se tiene una matriz de flechas que cambian de color según la rapidez del viento y de dirección según la dirección del viento. En la Tabla 4.1 se muestran los colores según el rango de velocidad del viento.

Tabla 4.1. Colores según la velocidad del viento

Color	Código hexadecimal	Velocidad mínima (Km/h)	Velocidad máxima (Km/h)
Azul	#0000FF	0	6
Azul Claro	#0099FF	7	12
Celeste	#00CCFF	13	18
Aguamarina	#00FFCC	19	24
Lima	#00FF00	25	30
Verde	#99FF33	31	36
Amarillo	#FFFF00	37	42
Oro	#FFCC00	43	48
Naranja	FF6600	49	54
Rojo	FF0000	55	60

Se presentarán ejemplos del funcionamiento del algoritmo de interpolación animado en Three.js. La Figura 4.1 representa el primer ejemplo, en este se tienen treinta estaciones repartidas de manera aleatoria en la matriz con valores de rapidez y dirección igualmente aleatorias.

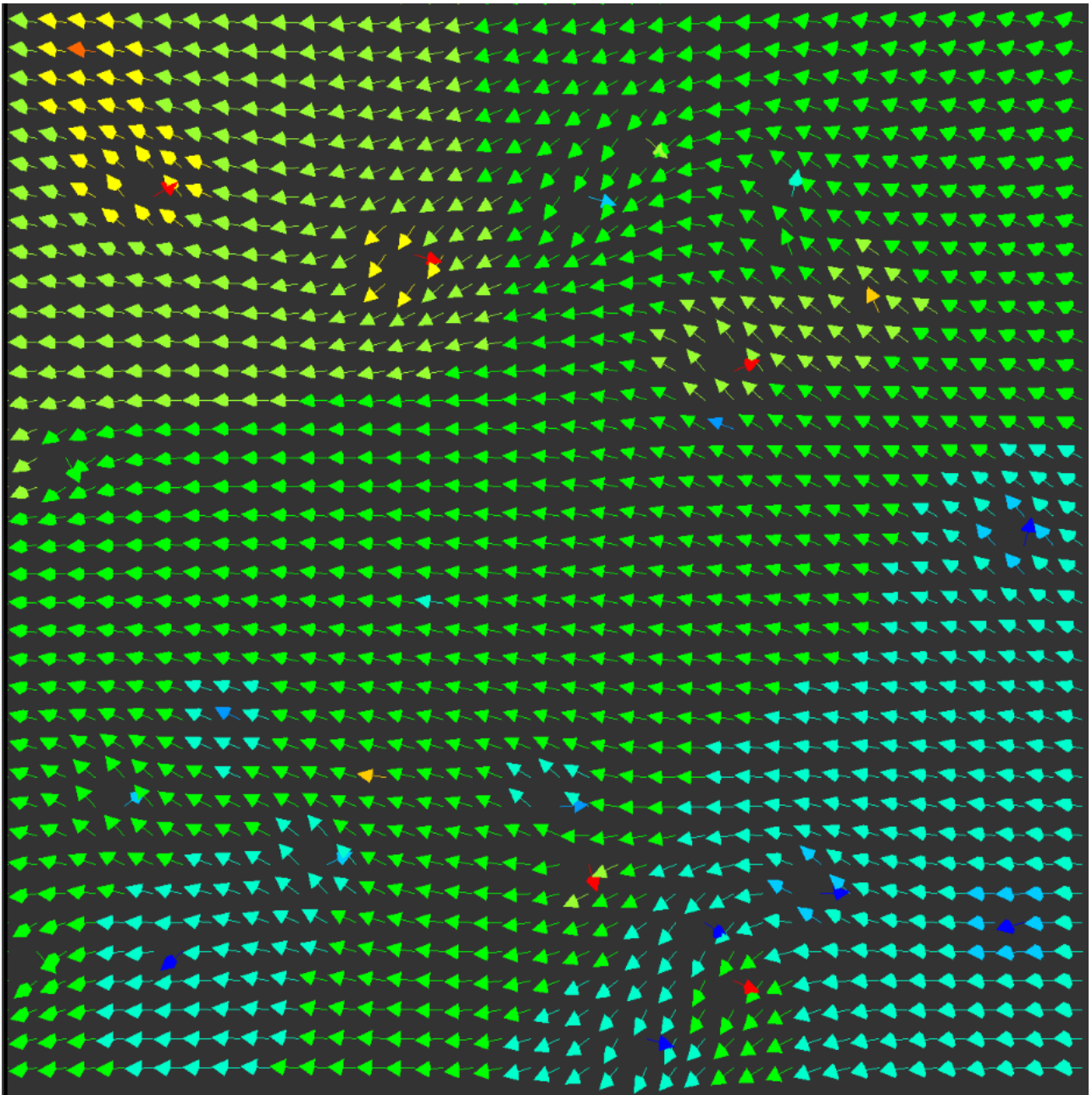


Figura 4.1. Prueba de ejemplo para el algoritmo de distancia inversa ponderada con valores de prueba diferentes para cada estaciones en Three.js

En la Figura 4.1 se observa la ejecución del algoritmo de interpolación. Se tienen sectores “azules” donde la velocidad del viento es baja y va cambiando de color a verde conforme la velocidad va subiendo a los sectores “rojos”. La dirección de las flechas va siguiendo un patrón general, desviándose un poco alrededor de las estaciones con una dirección diferente al patrón.

4.1.2 Datos de prueba con D3.js

Como se explicó en el apartado 3.5, se utilizó la biblioteca D3.js para la realización del proyecto. Al implementar el algoritmo de distancia inversa ponderada en JavaScript y animarlo con D3.js se obtuvieron los siguientes resultados.

La Figura 4.2 representa el resultado del algoritmo al escoger veinte puntos aleatorios, simbolizando las estaciones, y asignándoles una corriente de viento con una rapidez de 13 metros por segundo y una dirección de 170° .



Figura 4.2. Prueba de ejemplo para el algoritmo de distancia inversa ponderada con vientos de 13 m/s a 170°

Con el ejemplo de la Figura 4.2 se puede observar el funcionamiento del algoritmo de distancia inversa ponderada. Se tienen veinte puntos repartidos sin ningún patrón definido, por lo que se puede concluir que el algoritmo no depende de la posición de las estaciones. También se puede observar que todas las líneas tienen el mismo patrón, el mismo largo y la misma dirección, al no haber cambios entre los valores de las estaciones, cuando se realiza la ponderación se obtienen los mismos números previamente asignados a las estaciones. La Figura 4.3 representa otro ejemplo cambiando la rapidez a 2.5 metros por segundo y la dirección a 10° .



Figura 4.3. Prueba de ejemplo para el algoritmo de distancia inversa ponderada con vientos de 2.5 m/s a 10°

De igual manera se puede observar en la Figura 4.3 que los patrones de las líneas son equivalentes. Al utilizar valores iniciales de igual magnitud, los valores de los puntos calculados por el algoritmo resultan ser los mismos que los valores de los puntos iniciales.

En la Figura 4.4 se utilizan valores diferentes para la rapidez y dirección de las estaciones, teniendo veinte estaciones con posición, rapidez del viento y dirección del viento aleatorias.



Figura 4.4. Prueba de ejemplo para el algoritmo de distancia inversa ponderada con valores de prueba diferentes para las estaciones

En la Figura 4.4 se puede notar un patrón de líneas diferente a lo largo del mapa, por ejemplo en la zona norte de Costa Rica se observan líneas largas con dirección Este, en el Pacífico Central se observan líneas medianas con dirección Sureste y en el Pacífico Sur se observan líneas cortas con dirección Este. El algoritmo de interpolación suaviza los cambios entre los valores de las estaciones por lo que el flujo de las líneas no cambia abruptamente. En el Pacífico Sur se encuentran dos líneas cortas que representan estaciones con valores muy diferentes a las líneas creadas por el algoritmo de interpolación, esta diferencia es la que genera que las líneas del Pacífico Central al Pacífico Sur se hagan más cortas y cambien de dirección Sureste a dirección Este.

4.1.3 Datos reales con D3.js

Al obtener y limpiar los datos reales del Instituto Meteorológico Nacional se integraron al proyecto. En la Figura 4.5 se tienen los datos del 1 de enero del 2014 animados en D3.js.



Figura 4.5. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 2

En la Figura 4.5 se muestran varios cambios respecto a la aplicación en el apartado 4.1.2, se triplicó el tamaño de la matriz, pasando de 20x20 a 60x60, se redujo el ancho de las líneas y las líneas representado a las estaciones, se cambiaron a color azul para facilitar su visualización. Respecto a la interpolación se puede observar igualmente que en las Figuras 4.1 y 4.4 existe un patrón general de intensidad y dirección del viento que ligeramente cambia según las estaciones. En los siguientes apartados se explicarán aspectos importantes del algoritmo de interpolación.

4.1.3.1 *Potencia*

En el apartado 2.1.2.4, donde se explica el algoritmo de distancia inversa ponderada, se menciona sobre la potencia en la ecuación. La potencia afecta la influencia de los valores iniciales en los valores interpolados, si la potencia es alta la influencia de los valores iniciales cercanos es más fuerte, si la potencia es baja la influencia de los valores iniciales lejanos se incrementa.

En la Figura 4.5 se tiene el algoritmo de interpolación de distancia inversa ponderada con una potencia de 2, este valor de potencia crea el patrón general también visto en las Figuras 4.1 y 4.4. Como se tiene una potencia de valor bajo la influencia de cada estación es similar en las líneas interpoladas, cambiando ligeramente cerca de las estaciones con valores lejos del promedio.

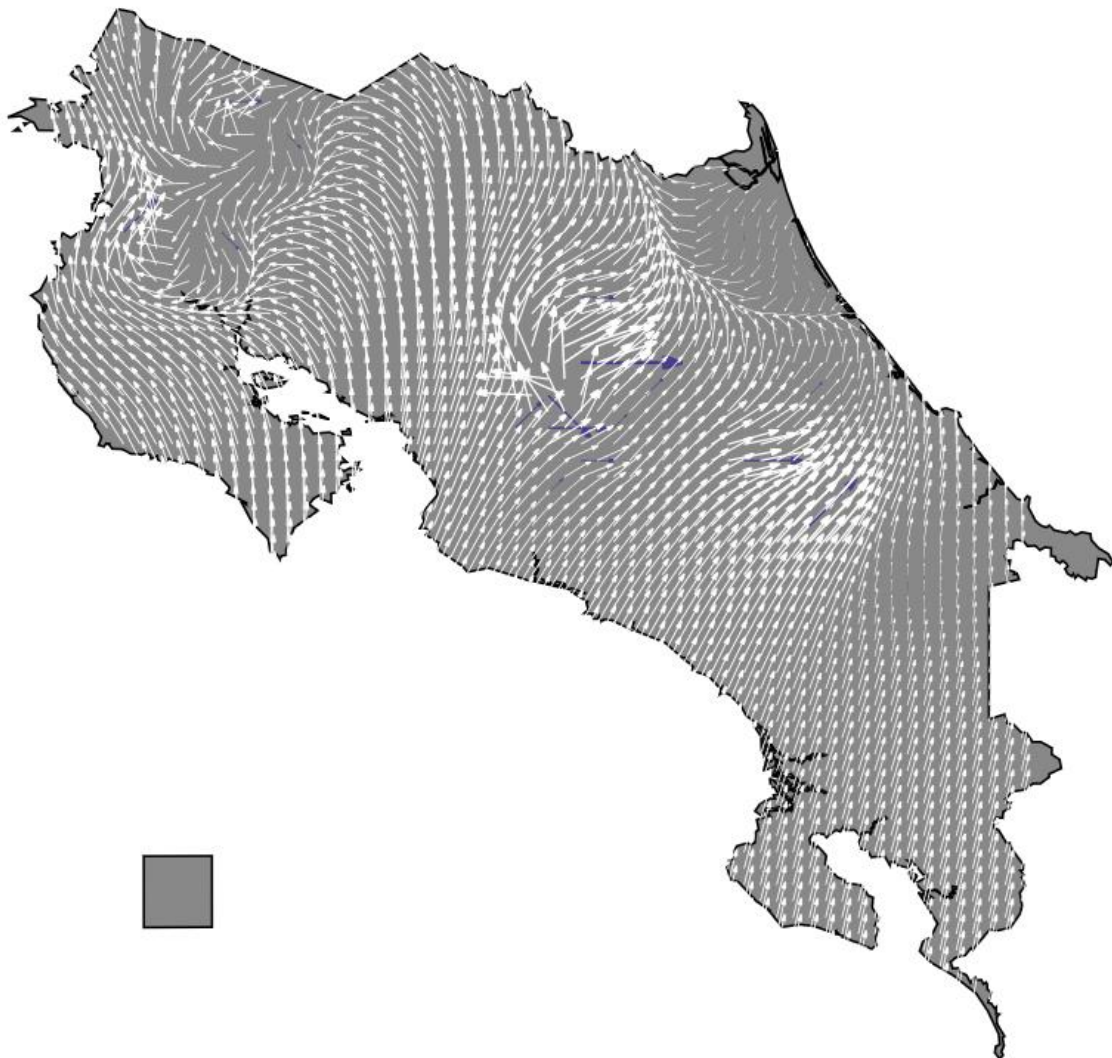


Figura 4.6. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 4

En la Figura 4.6 se cambia únicamente la potencia del algoritmo de interpolación subiendo a un valor de 4. Se puede observar que en las estaciones se tiene una misma rapidez y dirección del viento que en la Figura 4.5, pero el patrón general se sustituye con patrones por sectores alrededor de las estaciones. En las Figuras 4.7, 4.8 y 4.9 se mostrará el algoritmo al subir la potencia a 6, 8 y 10 respectivamente.



Figura 4.7. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 6



Figura 4.8. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 8

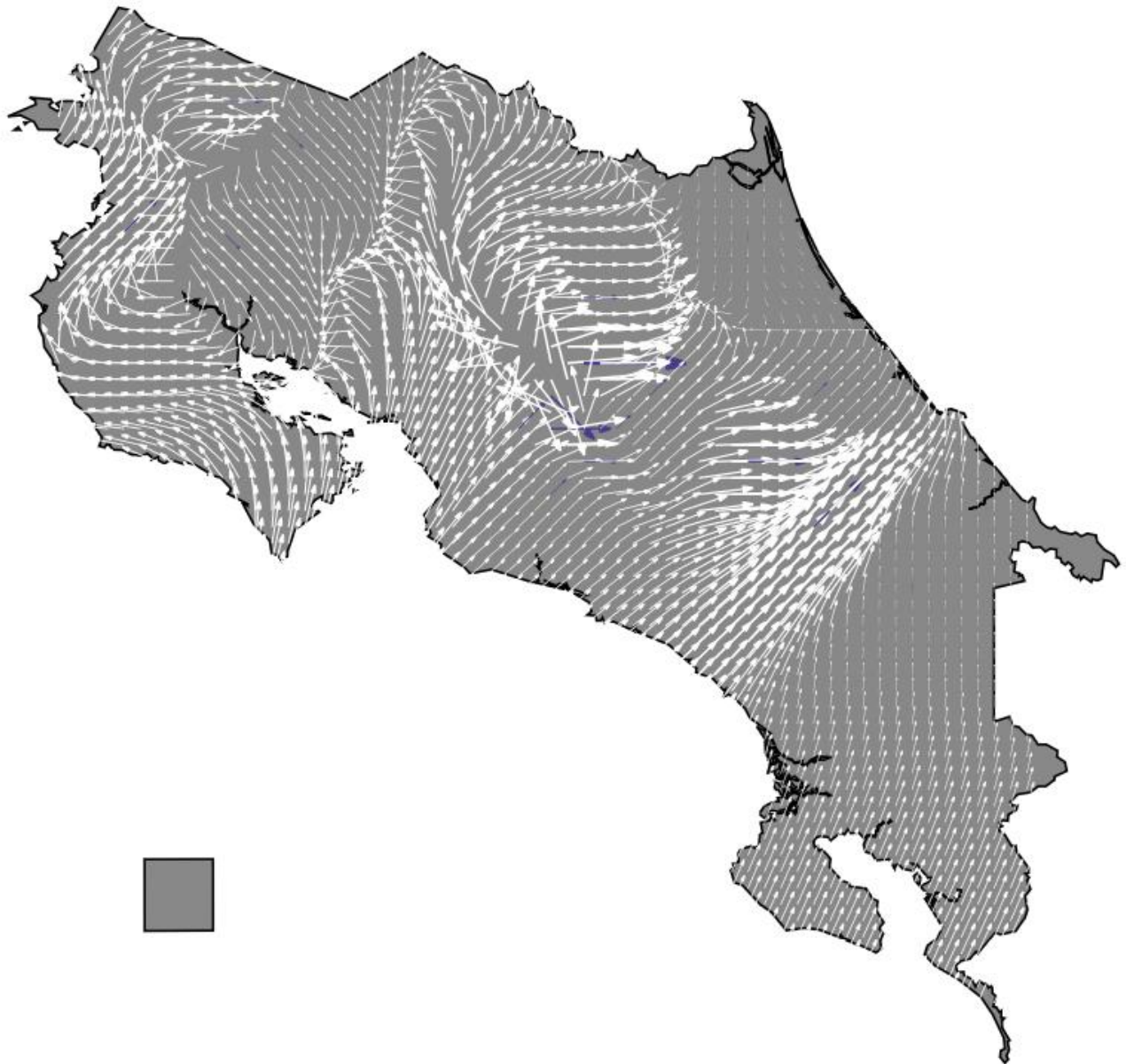


Figura 4.9. Mapa de vientos de Costa Rica del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 10

En las Figuras 4.7, 4.8 y 4.9 se aprecia que el aumento de potencia aumenta la influencia de las estaciones cercanas, las líneas siguen el patrón de las estaciones más influyentes según su cercanía.

El aumento del valor de la potencia trae consigo como consecuencia que la suavidad de los cambios de valores entre las estaciones se vea afectada. En la Figura 4.5, con potencia 2, se observa que los cambios entre valores ocurren con suavidad, mientras que en las Figuras 4.6, 4.7, 4.8 y 4.9, con potencias de 4, 6, 8 y 10 respectivamente, se observan cambios abruptos entre las estaciones.

Finalmente, para encontrar la potencia correcta se tendría que realizar un modelo estadístico de los vientos de Costa Rica, en una etapa futura del proyecto se puede retomar este aspecto para mejorar la visualización de los vientos. La potencia de valor 4 fue escogida para el proyecto por la razón de que en la visualización con potencia 4 los vientos fluyen con más naturalidad alrededor de los estaciones; con potencia 2 hay un patrón general a lo largo del mapa y con potencias más altas se crean cambios más abruptos entre valores.

4.1.3.2 Formato de datos

En el apartado anterior se explicó cómo la potencia afecta la visualización del algoritmo de interpolación. Se observó cómo las potencias altas crean cambios repentinos entre los valores iniciales, pero la potencia no es la única que genera este tipo de cambios. En la Tabla 4.2 se pueden observar los datos entregados por el Instituto Meteorológico Nacional de los primeros siete días del mes de mayo del 2014 en la estación 94013.

Tabla 4.2. Primeros siete días del mes de mayo del 2014 en la estación 94013

Año	Mes	Día	Direc. Pred.	Velocidad (m/s)
2014	5	1	Noroeste	3.83
2014	5	2	Norte	1.90
2014	5	3	Noroeste	3.12
2014	5	4	Este	2.12
2014	5	5	Este	2.02
2014	5	6	Sureste	2.93
2014	5	7	Noreste	2.21

Los datos de dirección del viento del Instituto Meteorológico Nacional se describen como “dirección predominante”, lo que implica que cambios de dirección durante el transcurso del día no son considerados y no se escoge como dirección final la dirección promedio durante el día. Además se puede observar de la Tabla 4.2 que la dirección se representa como los puntos cardinales, teniendo 45° de diferencia entre cada punto cardinal. En la segunda fila de la Tabla 4.2 tenemos que la velocidad predominante es Norte, si se realiza un cambio a grados, como se especifica en la Tabla 3.5, se tendría 90°, considerando la diferencia entre puntos cardinales de 45°, realmente se tendrían $90^{\circ} \pm 22,5^{\circ}$, en la Tabla 4.3 se observan los rangos de grados para los puntos cardinales.

Tabla 4.3. Rangos de grados para los puntos cardinales

Punto Cardinal	Grado Mínimo	Grado Medio	Grado Máximo
Este	337,5	0	22,5
Noreste	22,5	45	67,5
Norte	67,5	90	112,5
Noroeste	112,5	135	157,5
Oeste	157,5	180	202,5
Suroeste	202,5	225	247,5
Sur	247,5	270	292,5
Sureste	292,5	315	337,5

Para el proyecto se escogió utilizar el grado medio, el cual estaba especificado en los datos entregados por el Instituto Meteorológico.

Teniendo en consideración los rangos de grados para los puntos cardinales, si se observan las diferencias de dirección entre las filas de la Tabla 4.2, se puede notar que hay una gran diferencia de valores entre un día y otro. Por ejemplo, del día 3 de mayo al 4 de mayo hay un rango de 90° a 180° en un solo día.

Estos cambios bruscos de valores en las estaciones también crean pérdida de suavidad, en la Figuras 4.10 y 4.11 se mostrará el cambio de suavidad causado por pasar del 2 de enero al 3 de enero del 2014.



Figura 4.10. Mapa de vientos de Costa Rica del 2 de enero del 2014, interpolación de distancia inversa ponderada con potencia 4



Figura 4.11. Mapa de vientos de Costa Rica del 3 de enero del 2014, interpolación de distancia inversa ponderada con potencia 4

En la Figura 4.10 se observa en la zona central del país, que las líneas cerca de la estación al norte tienen un cambio brusco, al cambiar de día, en la Figura 4.11 la estación cambia de dirección y el flujo de las líneas se suaviza.

En conclusión, para mejorar la visualización de los vientos de Costa Rica se necesitan valores más exactos de dirección en las estaciones.

4.1.3.3 Posición de estaciones

La posición de las estaciones también afecta la forma en la que el algoritmo de interpolación efectúa sus cálculos. En la Tabla 4.4 se muestran las zonas climáticas de las estaciones escogidas.

Tabla 4.4. Zonas climáticas por estación.

Número	Estación	Zona Climática
69679	Upala, Aut.	Norte
72149	San José, Pinilla, Aut.	Pacífico Norte
73137	Volcán Irazú, Aut.	Valle Central
73145	Earth, Cc, Aut.	Atlántica
73147	Guápiles, Aut.	Atlántica
73149	Maravilla , Juan Viñas, Aut	Atlántica
74051	Aerop. Liberia Oeste 07, Aut.	Pacífico Norte
74061	El Corral, Palo Verde Cc. Aut.	Pacífico Norte
84139	CIGEFI, Aut.	Valle Central
84141	IMN, Aranjuez, Aut.	Valle Central
84169	Aerop. Juan Santamaria Media Pista, Aut.	Valle Central
84187	Fabio Baudrit, Aut.	Norte
84195	Aerop. Pavas Oeste, Aut.	Valle Central
84197	Santa Bárbara, Aut.	Norte
84203	Zapote, San José, Aut.	Valle Central
84215	Cerro Burío, Aserrí, Aut.	Montañosa del Sur
84219	Cerro Chitaría, Santa Ana, Aut.	Valle Central
84231	Cerro Cedral, Escazú, Aut.	Valle Central
94013	Cerro Buenavista, Cc. Aut	Atlántica
98095	Altamira, Aut.	Atlántica

Se tienen 8 estaciones en el Valle Central, 5 en el Atlántico, 3 en el Pacífico Norte, 3 en la zona Norte, 1 en la región Montañosa del Sur, 0 en el Pacífico Central y 0 en el Pacífico Sur.

En las Figuras del apartado 4.1.3 se ha mostrado la distribución de las estaciones a lo largo del país, como se muestra en la Tabla 4.4 las estaciones están centradas principalmente en el Valle Central. Esta concentración de estaciones en el Valle Central genera que en los lugares lejanos a este los valores de la interpolación no sean tan certeros.

Como solución a este problema se necesitaría una lista de estaciones más variada en localizaciones, considerando las estaciones meteorológicas que el Instituto Meteorológico Nacional tenga disponibles.

4.1.3.4 Cantidad de estaciones

La cantidad de estaciones también es un factor importante a la hora de la interpolación. Actualmente se cuenta con veinte estaciones que se utilizaron y doce estaciones extra con datos faltantes. Con una cantidad mayor de estaciones, distribuidas como se explica en el apartado 4.1.3.3, el algoritmo de interpolación mejoraría los valores creados a lo largo del mapa.

4.1.3.5 Cantidad de valores

Actualmente se cuenta con una matriz de 60x60 para todo el país, en las Figuras 4.2, 4.3 y 4.4 se muestra el algoritmo con un matriz de 20x20 y la densidad de líneas no es lo suficiente para tener una buena visualización de los vientos de Costa Rica. Con la matriz de 60x60 las corrientes de vientos son más claras y se puede determinar con una mejor claridad las condiciones de viento en una localización específica. En las Figuras 4.12 y 4.13 se mostrará un ejemplo de este apartado.

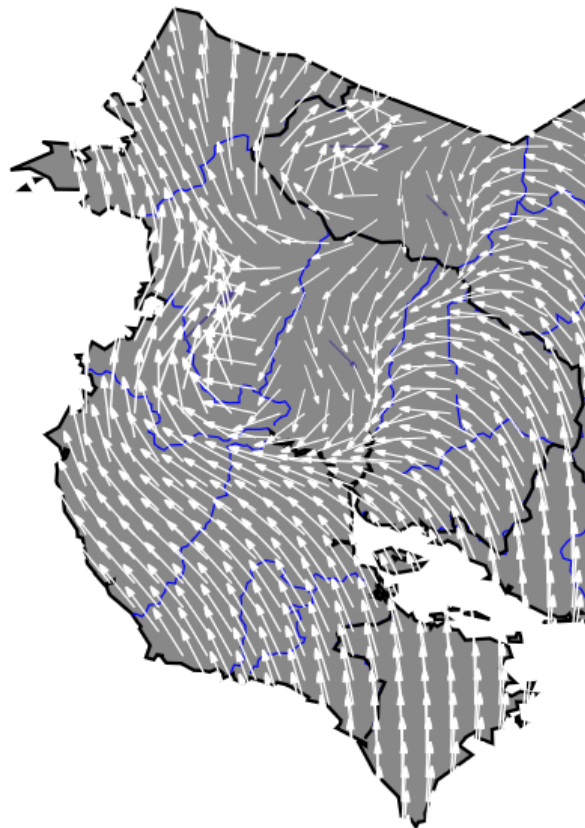


Figura 4.12. Mapa de vientos de Guanacaste del 1 de enero del 2014, interpolación de distancia inversa ponderada con potencia 4

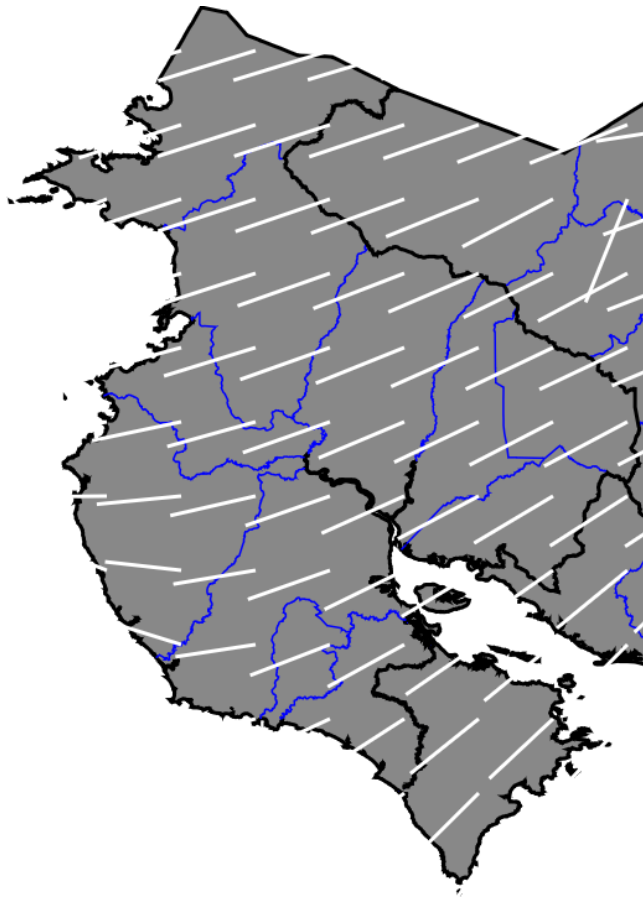


Figura 4.13. Mapa de vientos de Guanacaste, datos de prueba

En las Figuras 4.12 y 4.13 se muestra el mapa de Guanacaste, si se toma el ejemplo específico del cantón de Nicoya en la Figura 4.12 se puede encontrar al menos 30 líneas indicando las condiciones de viento del cantón, en la Figura 4.13 solo encontramos 6 líneas para todo el cantón.

Aumentar la cantidad de líneas favorece a la visualización, pero al aumentar la cantidad de líneas se debe aumentar el tamaño del mapa, esto se debe a que las líneas no se distinguen en una gran cantidad. En la Figura 4.14 se muestra un ejemplo de la matriz con 90x90 líneas usando datos de prueba.

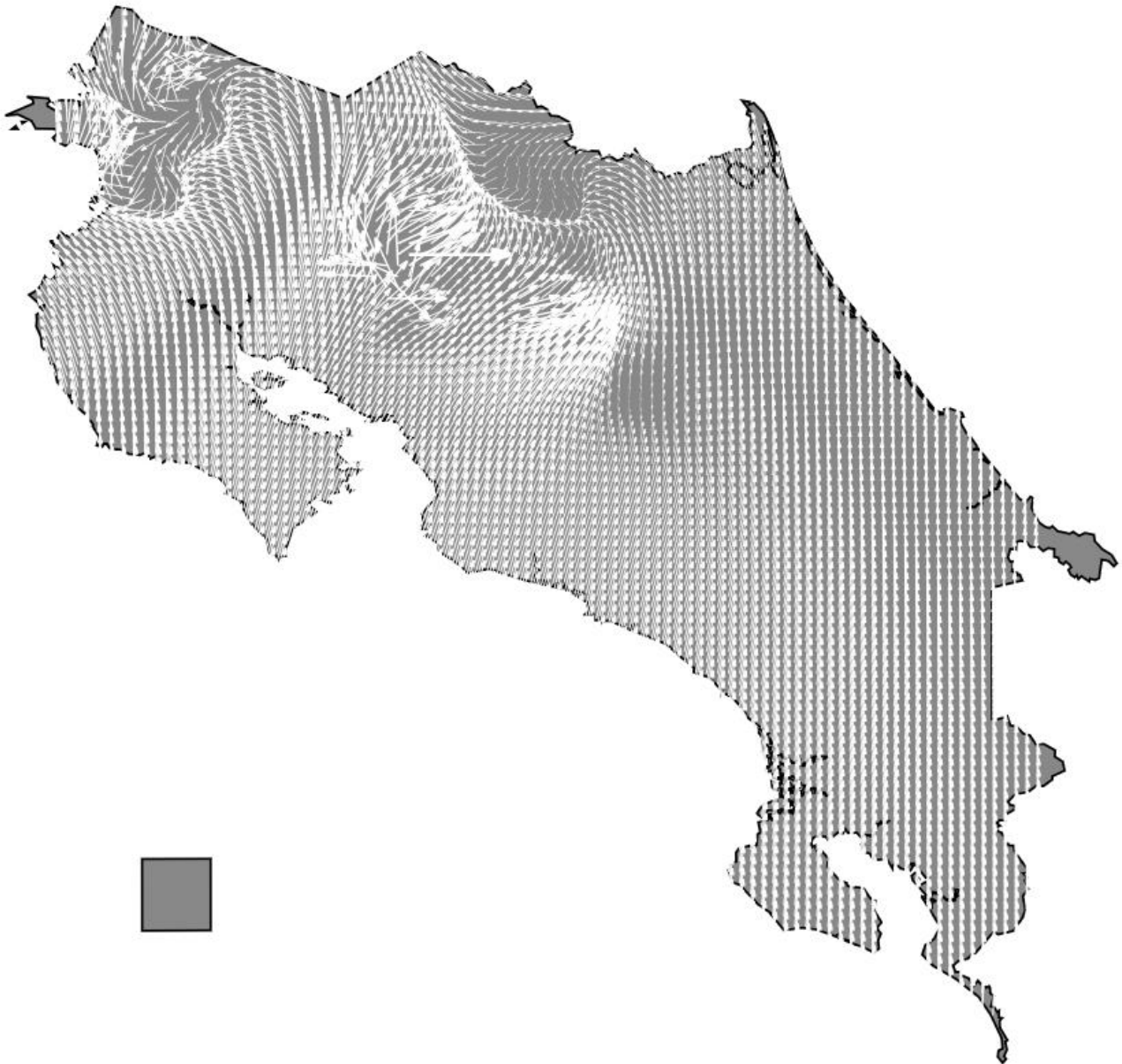


Figura 4.14. Mapa de vientos de Costa Rica, datos de prueba, matriz 90x90

En la Figura 4.14 se pierde el inicio y final de las líneas, por lo que no se puede decir en qué dirección se encuentra el viento.

Para aumentar el tamaño de la matriz se tiene que aumentar el tamaño del mapa, para este proyecto se escogió una matriz de 40x40 por razones de rendimiento.

4.1.3.6 *Altura*

Las diferencias de altura entre las estaciones también afectan la visualización, si se observan las Figuras dentro del apartado 4.1.3 se puede notar que entre las estaciones en el Valle Central hay cambios bruscos, estos cambios son generados por las diferencias de altura en las estaciones, a diferente altura las condiciones de viento cambian.

En una próxima iteración del proyecto se podría tomar este aspecto junto con el modelo estadístico del viento en Costa Rica para mejorar la visualización y el algoritmo de interpolación.

4.2 *Animación*

Las animaciones implementadas no poseen criterios cuantificables de comparación, se fueron modificando a criterio del cliente. El juicio del cliente se basa en su percepción del flujo del viento.

4.2.1 **Animación sincrónica**

La animación sincrónica fue la primera animación implementada, en esta todas las líneas tienen un mismo tiempo de ciclo. En la Figura 4.15 se muestra la aplicación con animación sincrónica.

Mapa de vientos de Costa Rica

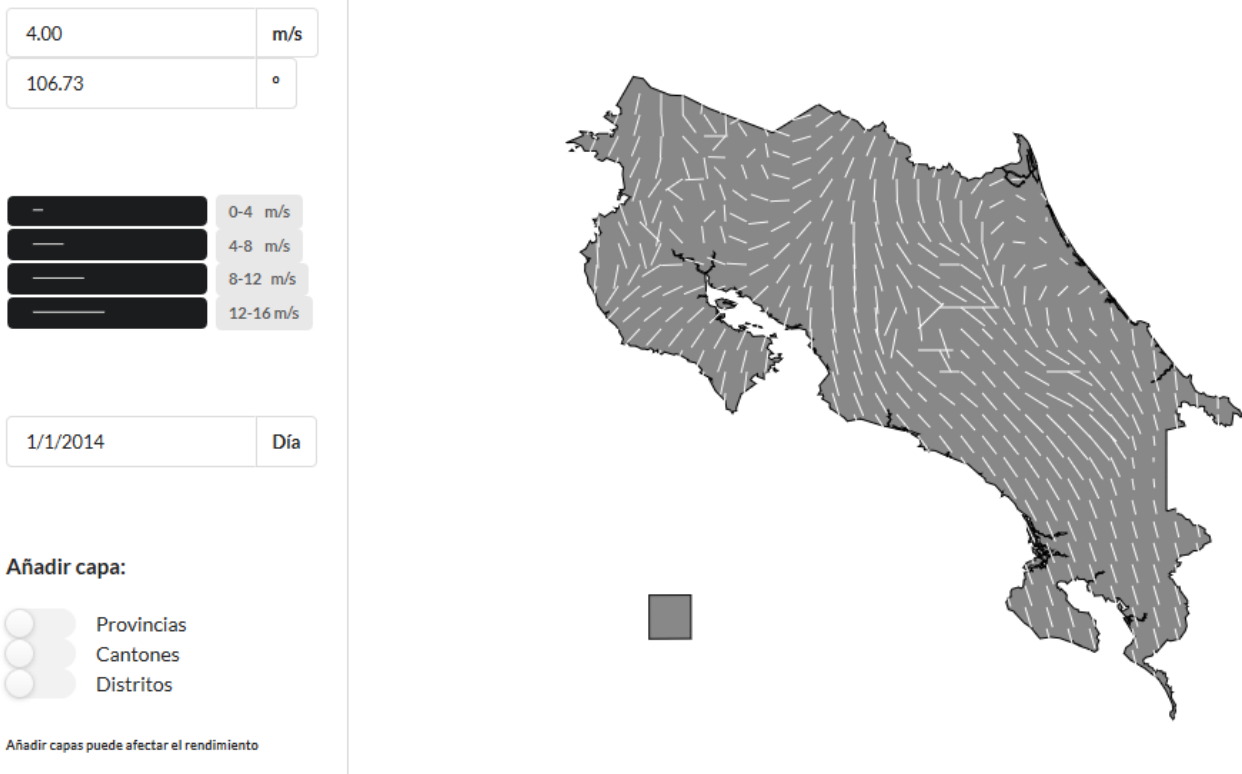


Figura 4.15. Mapa de viento de Costa Rica con animación sincrónica

En esta animación se utiliza una matriz de 30x30, el rendimiento se ve afectado por la animación constante, las líneas duran apareciendo 1000 ms y desapareciendo 500 ms, el largo de las líneas es igual a la velocidad calculada multiplicada por 3.6 y se tiene una separación de 15 píxeles entre cada línea.

El juicio emitido por el cliente sobre esta animación se basa en la falta de percepción de flujo, ya que hay momentos en el que viento desaparece totalmente, en la Figura 4.16 se observa esta condición. También se refirió a la separación de las líneas, las cuales no aparentaban continuidad en las corrientes de viento. Finalmente, manifestó que en esta propuesta de animación, se daba la impresión que los vientos aparecían en los orígenes de las líneas y desaparecían a los pocos kilómetros, por lo que recomendó utilizar líneas estáticas que solo se animarán al pasar de día, donde los valores de las estaciones cambiaban.

Mapa de vientos de Costa Rica



Figura 4.16. Mapa de viento de Costa Rica con animación sincrónica, al finalizar un ciclo de animación

4.2.2 Animación por transiciones

Siguiendo las recomendaciones del cliente, se implementó una animación por transiciones, en esta solo se presenta la animación al cambiar de día, moviendo las líneas a su nueva posición en vez de moverlas desde el punto de origen. En la Figura 4.17 se puede observar la aplicación con animación con transiciones.

Mapa de vientos de Costa Rica



Figura 4.17. Mapa de viento de Costa Rica con animación por transiciones

En esta animación se utiliza una matriz de 60x60, se aprovecha la ligereza de la animación, las líneas duran 1000 ms haciendo una transición y esperan un mínimo de 500 ms entre transiciones, el largo de las líneas es igual a la velocidad calculada multiplicada por 3.6 y se tiene una separación de 7 píxeles entre cada línea. Además se añadieron colores para las líneas que representan a las estaciones.

De igual manera el juicio emitido por el cliente se basa en la falta de percepción del flujo del viento, pese a que se corrigen los aspectos mencionados en la animación sincrónica, el hecho de que se tengan líneas estáticas la mayor parte del tiempo no genera la apreciación de las corrientes de viento deseada. Otro efecto de las líneas estáticas es la falta de percepción de la dirección, solo mientras se genera la transición se tiene una idea clara de la dirección de las líneas.

Las siguientes soluciones se plantearon para solventar los problemas de la animación por transiciones: regresar a la animación continua pero de manera asincrónica, cambiar el ancho según la velocidad y mejorar el rendimiento. Además se recomendó agregar flechas de indicación de dirección. En la Figura 4.18 se muestra la aplicación con animación por transiciones por flechas y cambio de ancho según la velocidad.

Mapa de vientos de Costa Rica

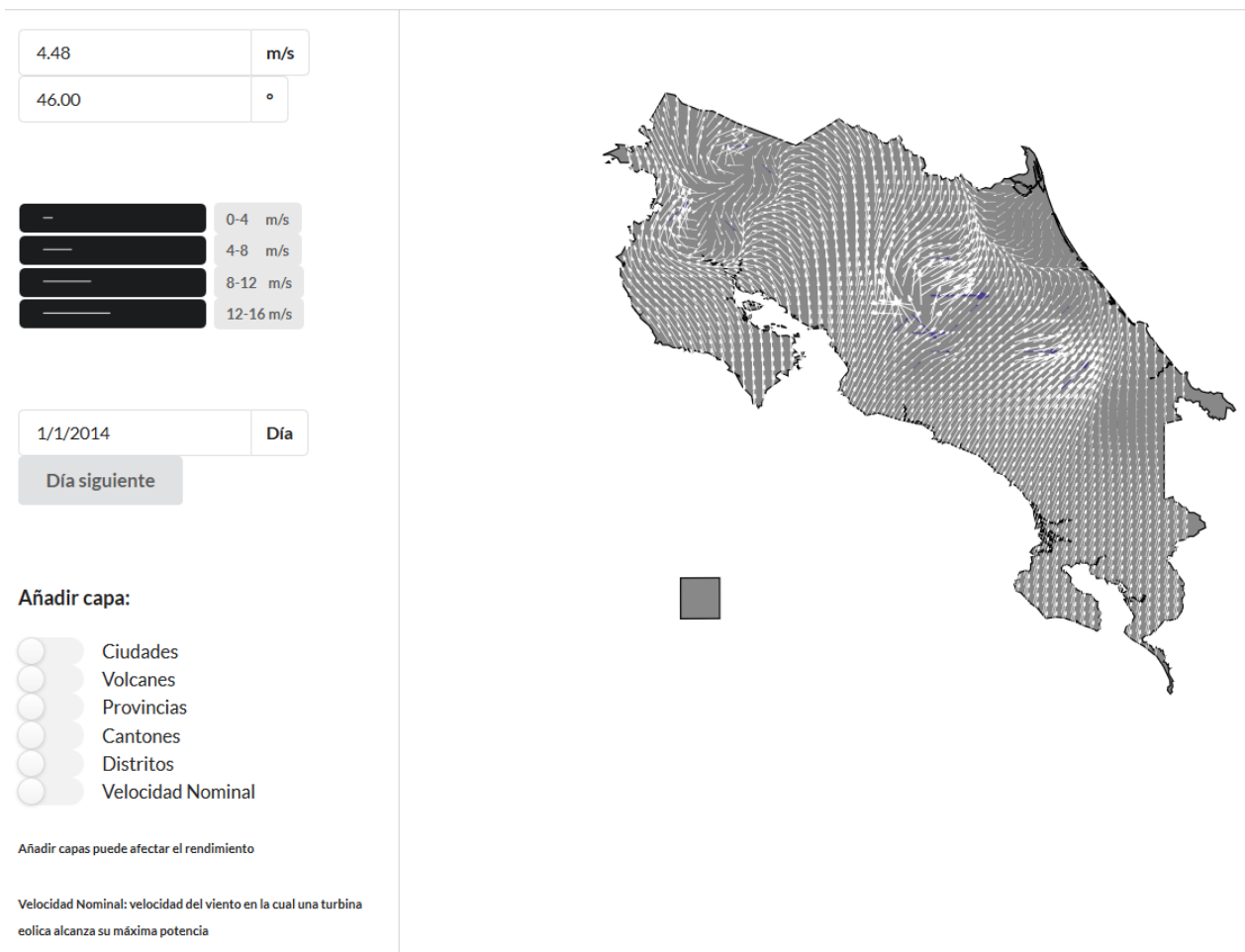


Figura 4.18. Mapa de viento de Costa Rica con animación por transiciones, con líneas de ancho variable y flechas indicadoras de dirección

4.2.3 Animación asincrónica

En la animación asincrónica se aplican los cambios propuestos para mejorar la percepción de la animación. Se utiliza una animación similar a la animación sincrónica, con la diferencia de que las transiciones tienen duración variable según la velocidad del viento. En las Tablas 3.6 y 3.7 se puede observar el cambio de duración según la velocidad. En la Figura 4.19 se muestra la aplicación con animación asincrónica con líneas adicionales.

Mapa de vientos de Costa Rica

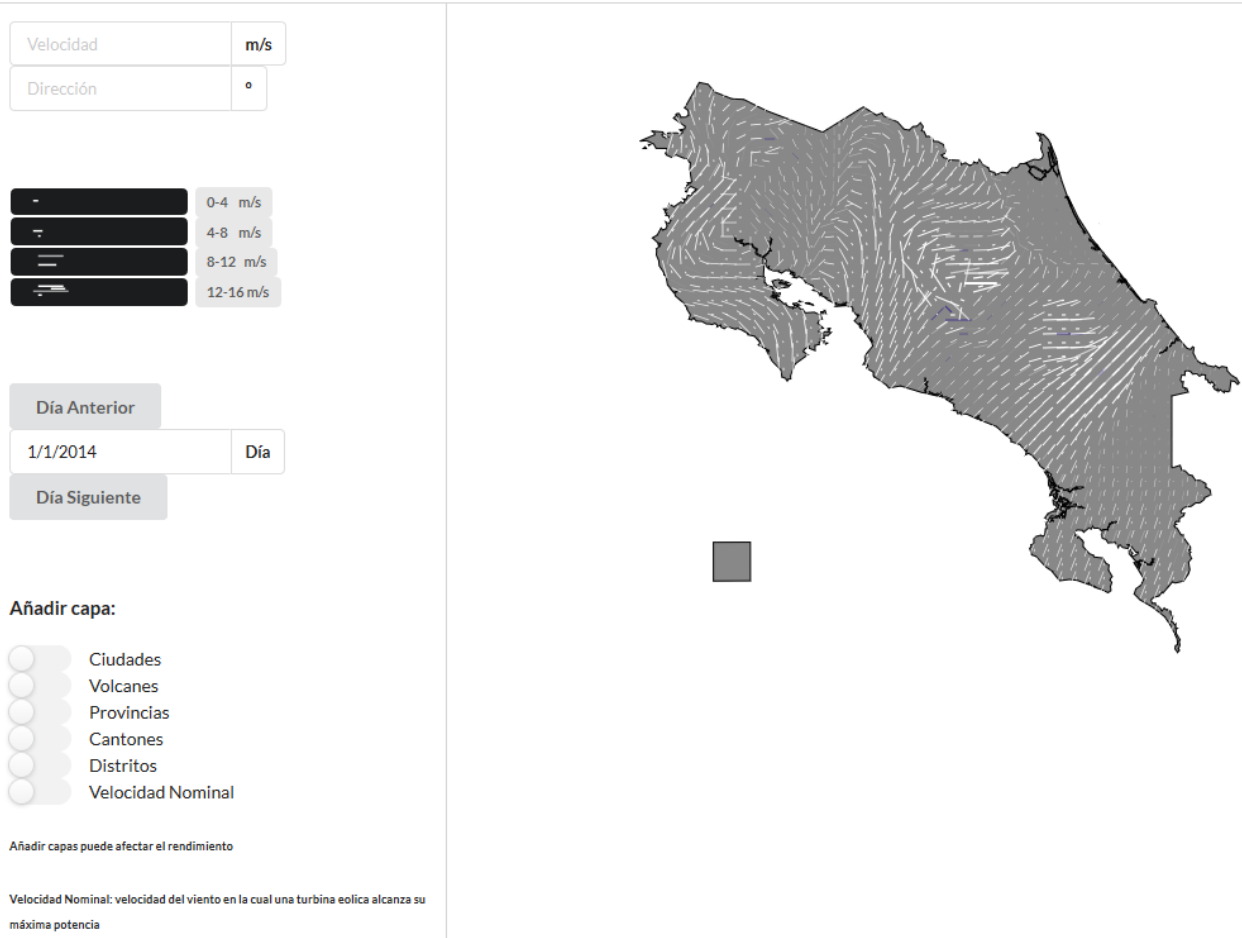


Figura 4.19. Mapa de viento de Costa Rica con animación asincrónica

En esta animación se utiliza una matriz de 40x40, el rendimiento se ve afectado por la animación constante, las líneas duran un tiempo variable apareciendo y otro tiempo variable desapareciendo, el largo de las líneas es igual a la velocidad calculada multiplicada por 2, el ancho es igual, velocidad calculada dividida por 6 y se tiene una separación de 11 píxeles entre cada línea. Además se añadieron líneas adicionales (ver apartado 3.5.4) con una separación en el eje horizontal y vertical de 4 píxeles.

Respecto a las animaciones anteriores, la animación asincrónica logra solventar los problemas encontrados, se mejora la percepción de la velocidad con el largo, con el ancho, con la duración del ciclo y con la cantidad de líneas, se mejora la percepción de la dirección con la animación de las líneas creciendo en la dirección calculada. También se mejora la percepción del flujo de viento con las líneas adicionales, ya que no se crean los espacios donde las líneas desaparecen al terminar su ciclo (ver Figura 4.16). Además con las líneas adicionales se crean campos de densidad alta, simulando grandes corrientes de viento.

El juicio del cliente para esta animación se basó en el rendimiento y en la cantidad de líneas, al aumentar la matriz de tamaño el rendimiento se vería afectado, y con una matriz más pequeña regresarían los problemas de percepción presentados en la Figura 4.15.

4.3 Rendimiento

Para mejorar la percepción del cliente de las corrientes de viento se aumentó la cantidad de líneas, pero al hacerlo se vio afectado el rendimiento. A continuación se explican las soluciones implementadas para mejorar el rendimiento.

4.3.1 Paralelismo

En apartado de alcances, entregables y limitaciones (apartado 1.5), se explica cómo el rendimiento de la aplicación es un gran limitante previsto desde el inicio del proyecto, y como posible solución se propone computación de alto rendimiento.

Usando los perfiladores de los navegadores web se descubrió que la mayor carga de rendimiento está en la animación (ver Tabla 4.6), por lo que era lo primero que se debía ser paralelizado. Investigando las bibliotecas de JavaScript se descubrió que el paralelismo no se utiliza, debido a que JavaScript trabaja en un solo hilo, pero aun así se encontró que los *web workers* podían ser utilizados para crear paralelismo y que la biblioteca Parallel.js simplificaba el uso de *web workers* para esta funcionalidad.

Utilizar Parallel.js desde un inicio conllevaba una restricción, los *web workers* no tienen acceso al DOM, por lo que paralelizar la animación no se podía llevar a cabo. En la Figura 3.1 se muestra el diagrama de flujo de la aplicación en general, en esta se aprecia que las partes fundamentales de la aplicación son la animación y el algoritmo de interpolación, por lo que se implementó Parallel.js para acelerar el código del algoritmo de interpolación. En la Tabla 4.5 se resumen los datos obtenidos de los tiempos de ejecución del algoritmo de interpolación sin paralelizar y el algoritmo de interpolación paralelizado (ver Apéndices 3 y 4 para datos completos).

Tabla 4.5. Comparación de promedio de tiempos de ejecución del algoritmo de interpolación sin paralelizar y paralelizado, datos del mes de enero

Algoritmo	Promedio (ms)	Máximo (ms)	Mínimo (ms)
Sin paralelizar	1,65	2,78	1,16
Paralelizado	1,68	3,20	1,18

El tiempo de ejecución en el algoritmo paralelizado es prácticamente igual al algoritmo sin paralelizar, los algoritmos fueron probados con matrices de 40x40 recortadas en los bordes para simular de mejor manera la forma de Costa Rica, generando una matriz de 1086 elementos. Se concluye que paralelizar el algoritmo de interpolación no contribuye a la mejora de rendimiento.

Se exploraron otros sectores del código que pudieran ser paralelizados, en la Tabla 4.6 se muestra el promedio del tiempo de ejecución para el código de carga de datos de las estaciones desde JSON a la aplicación (ver Apéndice 6 para datos completos), y el promedio del tiempo de ejecución para el código de inicializar las líneas (ver Apéndice 7 para datos completos).

Tabla 4.6. Promedio de tiempo de ejecución del código de carga de datos de las estaciones, datos del mes de enero

Código	Promedio (ms)	Máximo (ms)	Mínimo (ms)
Carga de Datos	2,80	7,85	2,42
Inicialización de líneas	257,43	272,33	241,20

En la Tabla 4.6 se observa que el tiempo de carga de datos es 69.7% más que el tiempo del algoritmo de interpolación. Si se comparan estos datos con los datos del código de inicialización de líneas se obtiene que el tiempo de ejecución del código de inicialización de líneas es 156.02 veces más que el algoritmo de interpolación y 91.94 veces más que el código de carga de datos. Se concluye que el paralelismo al no poder utilizarse para la animación no es una herramienta efectiva para mejorar el rendimiento en este proyecto.

4.3.2 Recorte de matriz

Como se concluyó en el apartado anterior, la principal causa de bajo rendimiento es la animación. En este apartado y en los siguientes se explicarán los cambios a la animación efectuados para mejorar el rendimiento.

La primera solución fue el recorte de matrices, se recorta la matriz de manera que se parezca a la forma que se tiene en Costa Rica. Las esquinas superior derecha e inferior izquierda son eliminadas. En la Tabla 4.7 se muestran los cortes para las matrices de tamaño 30x30, 40x40 y 60x60.

Tabla 4.7. Recortes de matriz para los tamaños de 30x30, 40x40 y 60x60

Tamaño de Matriz	Inicio del corte superior (eje horizontal)	Inicio del corte inferior (eje vertical)	Total sin cortes	Total con cortes	Porcentaje de corte
30x30	17	11	900	599	66,56%
40x40	24	14	1600	1086	67,88%
60x60	35	22	3600	2495	69,31%

Como se observa en la Tabla 4.7, el recorte de matrices reduce hasta un 33.41% la carga de la animación para el caso de una matriz de 30x30, 32.12% para una matriz de 40x40 y 30.69% para una matriz de 60x60.

4.3.3 Simplificación de capas

Se midió el tiempo de carga de la capa del mapa de Costa Rica en TopoJSON, la Tabla 4.8 muestra el promedio de los datos de esta medición (ver Apéndice 10 para datos completos).

Tabla 4.8. Tiempo de carga promedio del mapa de Costa Rica en TopoJSON

Capa	Promedio (ms)	Máximo (ms)	Mínimo (ms)
Mapa Completo	361,41	369,53	353,73

La carga del mapa de Costa Rica tiene un peso considerable en el rendimiento, para mejorar estos tiempos se decidió separar las capas en límite exterior, provincias, cantones y distritos, utilizando la capa de límite exterior como la capa básica y las otras capas como variables. En la Tabla 4.9 se muestra el promedio del tiempo de carga por capa (ver Apéndice 10 para datos completos).

Tabla 4.9. Tiempo de carga promedio para las capas del mapa de Costa Rica en TopoJSON

Capa	Promedio (ms)	Máximo (ms)	Mínimo (ms)
Límite exterior	64,99	66,70	64,06
Provincias	12,42	13,00	12,29
Cantones	54,29	55,77	53,09
Distritos	238,33	243,68	234,42

En la Tabla 4.9 se puede observar que la capa de distritos es la capa con más tiempo de carga, esto es debido a que tiene más geometrías, a diferencia de la capa de provincias, la cual tiene pocas geometrías y estas son simples.

La segunda solución para mejorar el tiempo de carga del mapa de Costa Rica fue la simplificación de geometrías (ver apartado 3.7.1). En la Figura 4.20 se muestra el tiempo de carga por capa simplificada, y en la Figura 4.21 la mejoría (ver Apéndice 11 para datos completos).

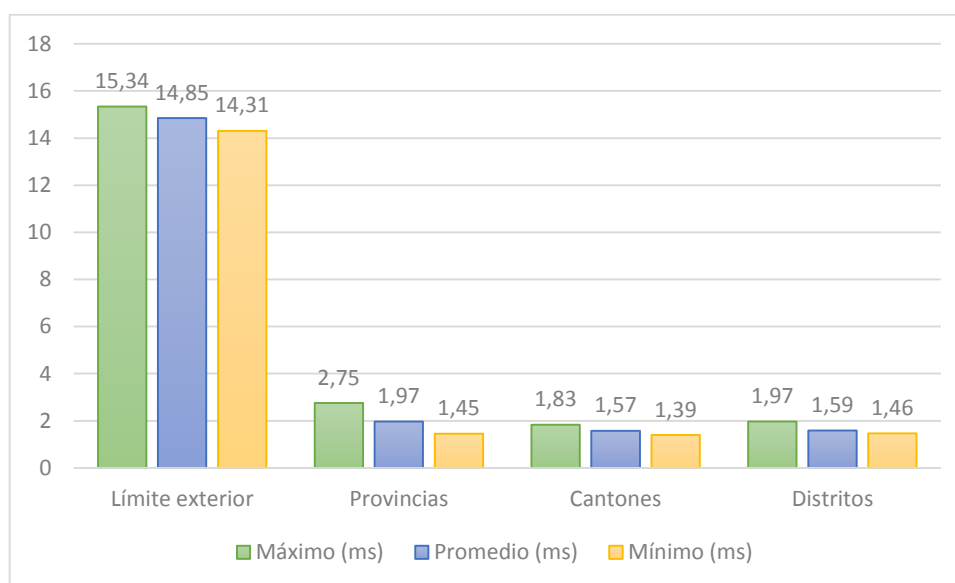


Figura 4.20. Gráfico de tiempo de carga para las capas con geometrías simplificadas del mapa de Costa Rica en TopoJSON

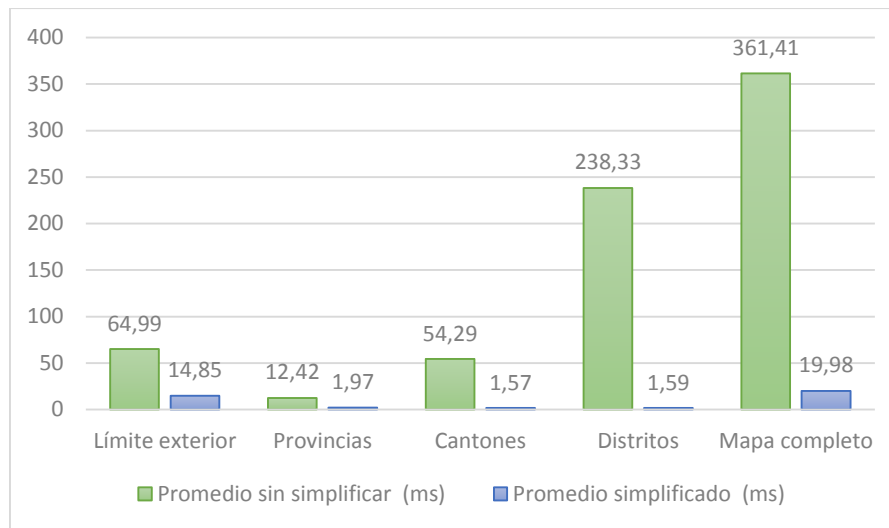


Figura 4.21. Gráfico de comparación de tiempos de carga entre las capas sin simplificar y las capas simplificadas

En las Figuras 4.20 y 4.21 se observa la gran mejoría otorgada por la simplificación de geometrías. A la capa de límite exterior se le aplicó el proceso de simplificación de geometrías en menor medida, la razón de esto es conservar el aspecto de los límites de Costa Rica, aun así el tiempo se redujo a un 22.85% del tiempo sin simplificar. El tiempo de carga de la capa de distritos se logró reducir a un 0.67% de su tiempo original. En total se da una reducción del tiempo de carga al 5,53% del tiempo inicial al utilizar todas las capas, de todas maneras se dejó la capa de límite exterior como capa básica y la demás capas como opcionales.

4.3.4 Canvas

Durante una investigación se descubrió que la biblioteca D3.js tiene problemas de rendimiento por diversas razones, a continuación se hace una lista de ellas y sus soluciones [43]:

- Crear muchos elementos SVG: el rendimiento de los elementos SVG es limitado, para solucionar esto se recomienda remover los grupos de elementos, utilizar canvas en vez de SVG y reducir la cantidad de datos.
- Usar bucles con “setInterval/setTimeout”: estas funciones obligan al programador a saber cuándo se tiene que redibujar, por lo que puede generar bajo rendimiento o salto de datos, para solucionar esto se recomienda utilizar la función “d3.timer”

- Redibujar elementos que no han cambiado: es un gasto de tiempo dibujar elementos que no han cambiado, por lo que se recomienda actualizar selectivamente.

- Datos obtenidos de un servidor: la cantidad de datos obtenidos en cada intervalo puede afectar el rendimiento, se recomienda utilizar bibliotecas adicionales para manejar los datos entrantes.

De estos problemas y soluciones solo el primer problema está presente en el proyecto. El redibujado selectivo no es una opción ya que se necesita que las líneas se redibujen por ciclos para tener la perspectiva del flujo del viento.

Al tener muchos elementos SVG se decidió cambiar estos por un canvas, el rendimiento del canvas depende del tamaño del canvas y no de la cantidad de elementos. En la Tabla 4.10 se muestra la comparación del tiempo de ejecución del código de inicialización de líneas utilizando elementos SVG y canvas (ver Apéndice 7 para datos completos).

Tabla 4.10. Promedio de tiempos de ejecución del código de inicialización de líneas con elementos SVG y con canvas

Formato	Promedio (ms)	Máximo (ms)	Mínimo (ms)
SVG	257,43	272,33	241,20
Canvas	241,60	243,82	235,76

Como se observa en la Tabla 4.10 no hay una mejora significativa con el tiempo de inicialización de líneas (en el Apéndice 5 se observa un caso similar con el algoritmo de interpolación). En la Tabla 4.11 se muestra el tiempo ejecución promedio por línea para elementos SVG y canvas (ver Apéndice 8 para datos completos).

Tabla 4.11. Promedio de tiempos de ejecución por línea con elementos SVG y con canvas

Formato	Promedio (ms)	Máximo (ms)	Mínimo (ms)	Total (ms)
SVG	0,28	0,47	0,13	305,71
SVG con líneas adicionales	0,41	0,62	0,38	448,95
Canvas	0,18	0,31	0,17	198,20

En la aplicación con canvas todas las líneas tienen al menos una línea adicional (ver apartado 3.5.4), mientras que en la aplicación con elementos SVG las líneas con velocidad menor a 4 m/s segundo no tienen línea adicional. La fila SVG con líneas adicionales se refiere a que se modificó la aplicación con elementos SVG para que todas las líneas tuvieran al menos una línea

adicional y tener una comparación con igualdad de condiciones. En la columna total se utiliza la referencia de que se tiene una matriz recortada de 40x40 y se multiplica el promedio de tiempo por la cantidad de elementos en la matriz.

En la Tabla 4.11 sí se logra observar la mejoría de rendimiento otorgada por el canvas, teniendo que los elementos SVG tienen un tiempo de ejecución 1,54 veces más que el tiempo de ejecución del canvas, y en igualdad de condiciones los elementos SVG tienen un tiempo de ejecución 2.27 veces mayor que el canvas.

Pese a las mejoras otorgadas por el canvas en rendimiento, este método afecta la funcionalidad del manejo de mapas de D3.js. Al colocar el canvas no se pueden colocar las capas en TopoJSON, por lo que la visualización en canvas es más sencilla con respecto a la información geográfica. En la Figura 4.22 se muestra la aplicación con canvas.

Mapa de vientos de Costa Rica

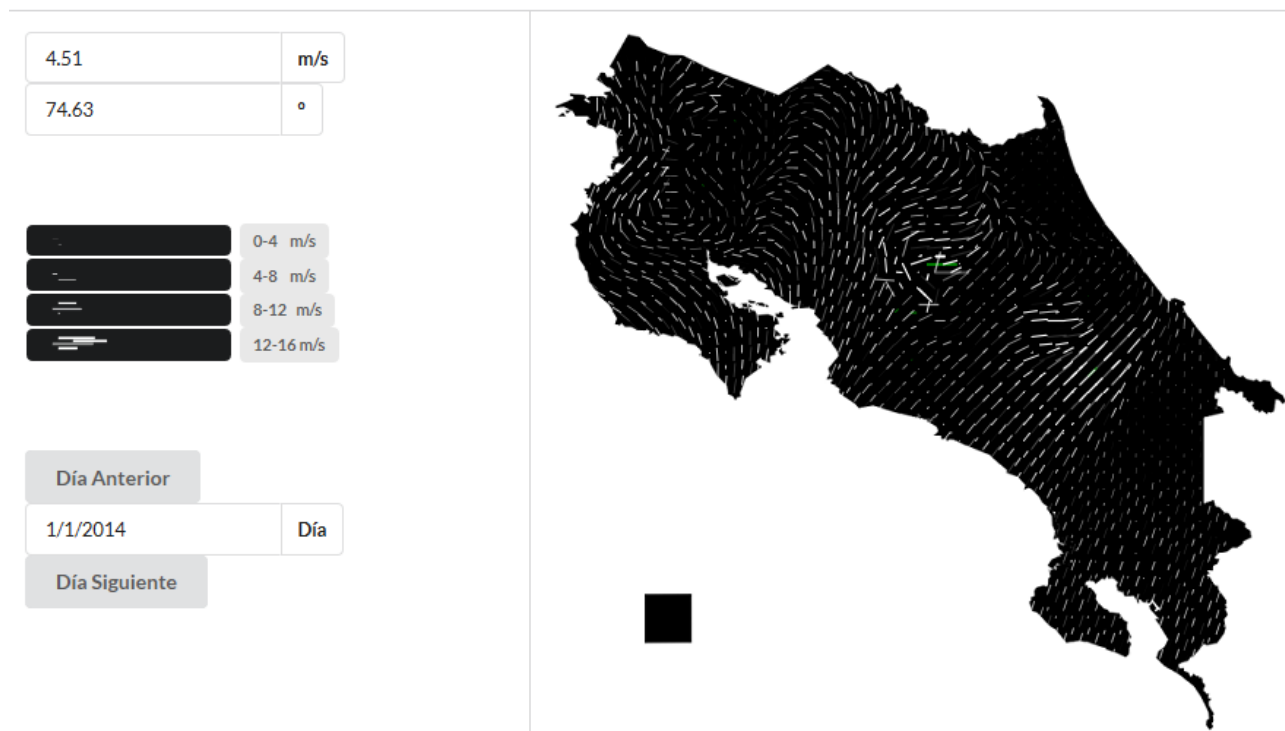


Figura 4.22. Mapa de viento de Costa Rica con animación asincrónica y canvas

En la Figura 4.22 se puede observar que el panel de capas ya no se encuentra y además en vez de mapa TopoJSON se utiliza una imagen con la silueta de Costa Rica como representación.

4.3.5 Acercamiento de imagen

Para realizar el acercamiento de imagen se realizaron varios cambios en los valores de las líneas para mantener la percepción de los vientos lo más similar posible a la solución anterior. En la Figura 4.23 se resumen los cambios a las líneas según se aumenta el tamaño de la imagen.

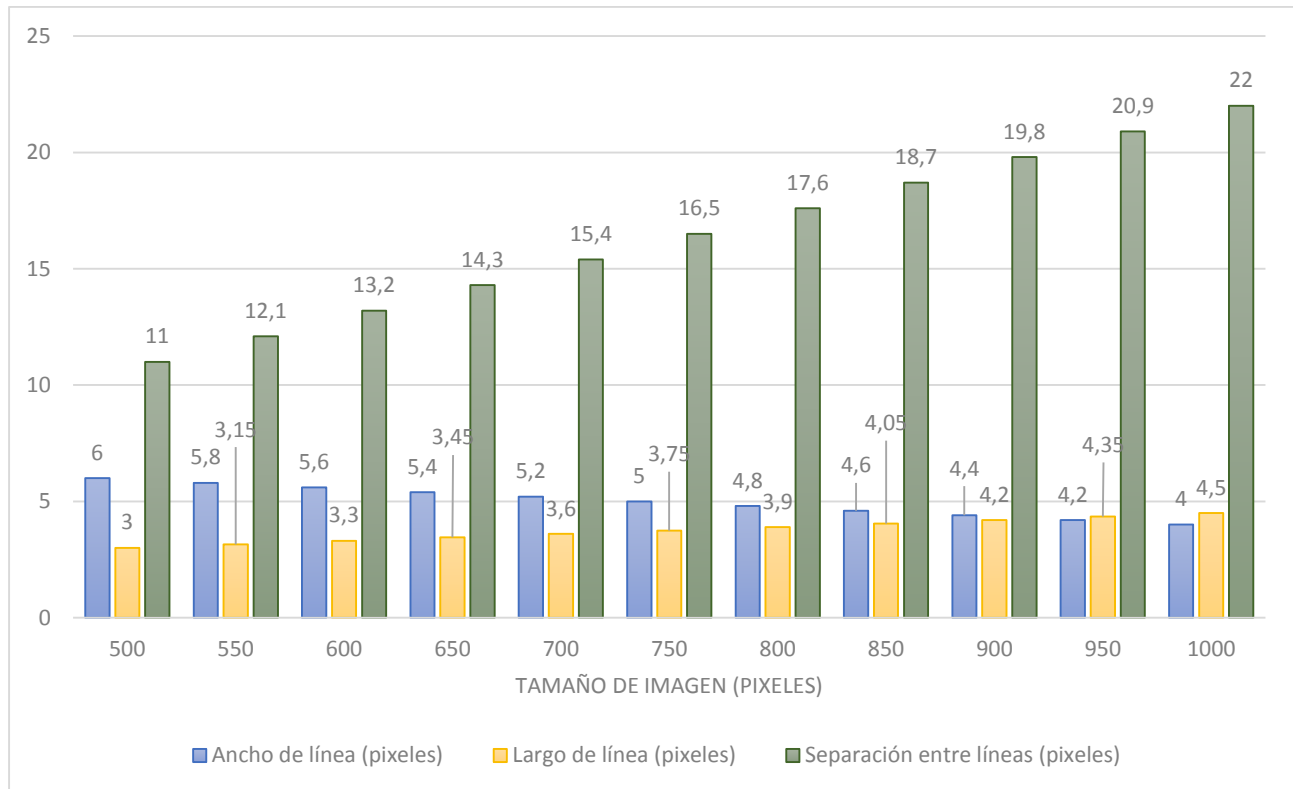


Figura 4.23. Gráfico de cambios en las propiedades de las líneas según el tamaño de la imagen.

En la Figura 4.23 se observa el aumento de la separación de las líneas del doble en el máximo acercamiento, el ancho cambia inversamente ya que este valor se divide con la velocidad para obtener el ancho variable y el largo aumenta solo un 50% de su valor original ya que se afecta la percepción de la visualización con largos de línea de valores altos.

Al cambiar los valores de las líneas y al añadir dos líneas adicionales para todas las líneas con valor de menos de 12 m/s, se aumenta la carga de rendimiento de la aplicación, en la Figura 4.24 se muestran los tiempos de ejecución por línea para el máximo acercamiento, el mínimo acercamiento y el canvas sin acercamiento (ver Apéndice 9 para datos completos).

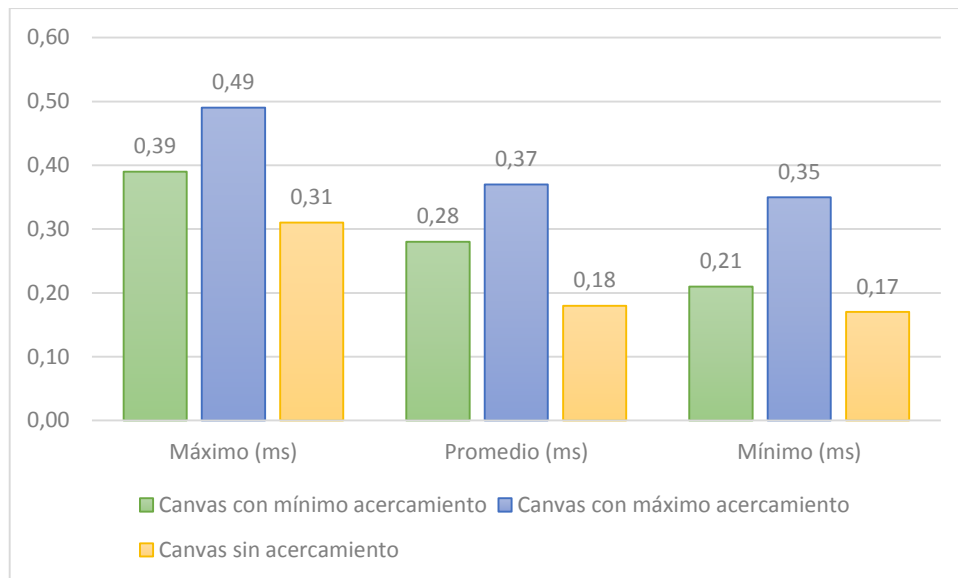


Figura 4.24. Gráfico de tiempos de ejecución por línea para canvas con mínimo acercamiento, con máximo acercamiento y para canvas sin acercamiento

En la Figura 4.24 se observa que la carga de rendimiento aumenta a un 101,97% con el máximo acercamiento, para solucionar este aumento en la carga de rendimiento se modificó otro valor de las líneas para cambiar la percepción de la animación. El parámetro modificado fue el *easing* de la animación, este cambia la velocidad de la animación para crear efectos más realistas, en las versiones anteriores se tenía un *ease* lineal por lo que se cambió a un *ease* cúbico para tener un velocidad mayor al inicio de la animación de las líneas y una velocidad reducida al finalizar la animación. Este efecto mejora la percepción del rendimiento de la aplicación ya que las líneas tienen una animación visualmente más rápida.

En la Figura 4.25 se muestra la aplicación con mínimo acercamiento y en la Figura 4.26 con máximo acercamiento.

Mapa de vientos de Costa Rica

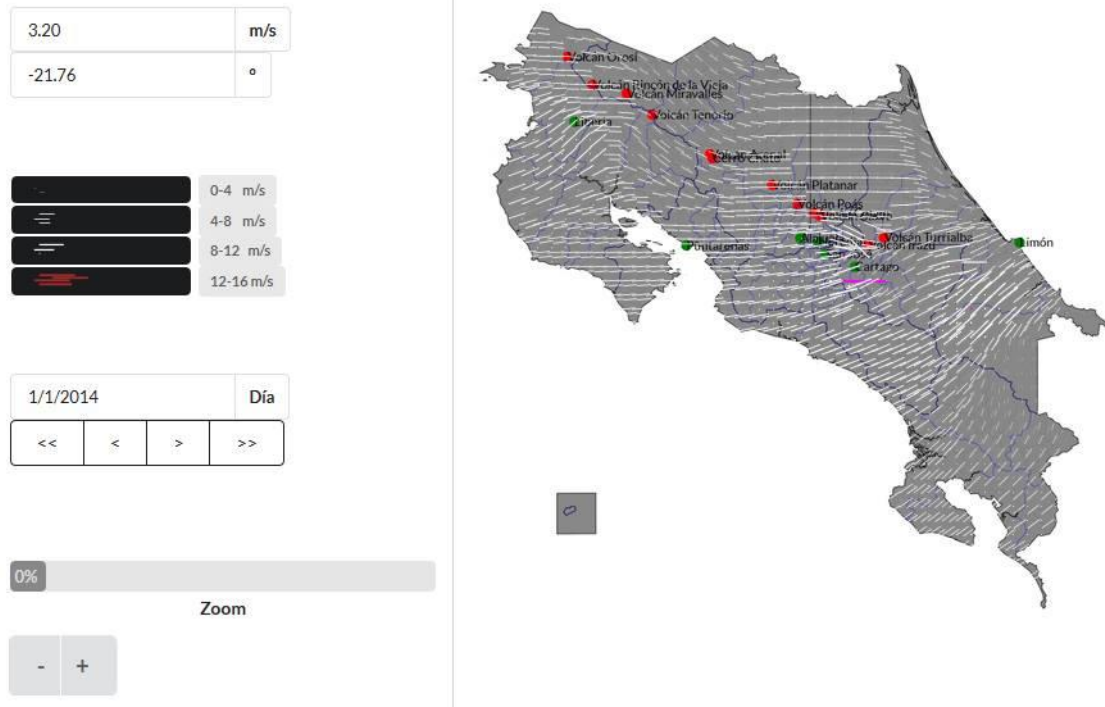


Figura 4.25. Mapa de viento de Costa Rica con mínimo acercamiento

Mapa de vientos de Costa Rica

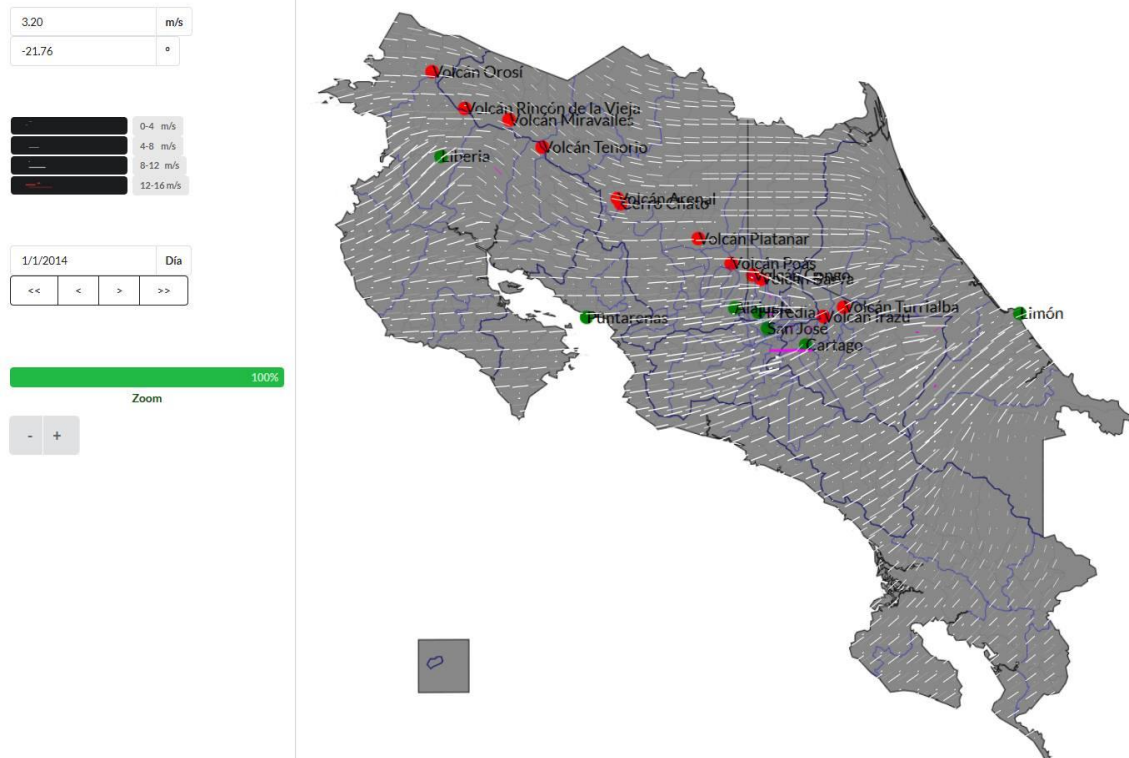


Figura 4.26. Mapa de viento de Costa Rica con máximo acercamiento

En las Figuras 4.25 y 4.26 se puede observar que se cambió la silueta del mapa de Costa Rica por una imagen con las capas previamente puestas.

Finalmente, mejorar el rendimiento es importante, ya que mejora la perspectiva de la aplicación en general, el canvas logra el mejor rendimiento con matrices de 40x40, pero se pierde información geográfica. La aplicación con elementos SVG no tiene un rendimiento tan bueno como la aplicación con canvas, pero posee ayudas visuales sobre información geográfica específica, no solamente la información encontrada en el panel de capas de la aplicación, sino también la información geográfica adicional que se desee agregar al mapa, con el fin de realizar estudios específicos en materia de vientos.

4.4 Velocidad nominal

En el apartado 2.3.1 se explica el funcionamiento de una turbina en relación con la velocidad del viento. En la aplicación se eligió que las velocidades donde se alcanza la velocidad nominal se tornen rojas, como representación. El estudio de las locaciones donde se presenta la velocidad nominal otorga oportunidades de colocación de turbinas de eólicas. En la Tabla 4.12 se muestra la lista de días donde se dan vientos que alcanzan la velocidad nominal.

Tabla 4.12. Días con vientos con velocidad superior a la velocidad nominal

Lista de días con vientos superiores a la velocidad nominal
9/4/2014
19/7/2014
21/7/2014
4/11/2014
27/11/2014
28/11/2014
25/12/2014
26/12/2014
30/12/2014

En la Figura 4.27 se muestra el día 9 de abril del 2014 donde se presentan vientos con velocidades superiores a la velocidad nominal.

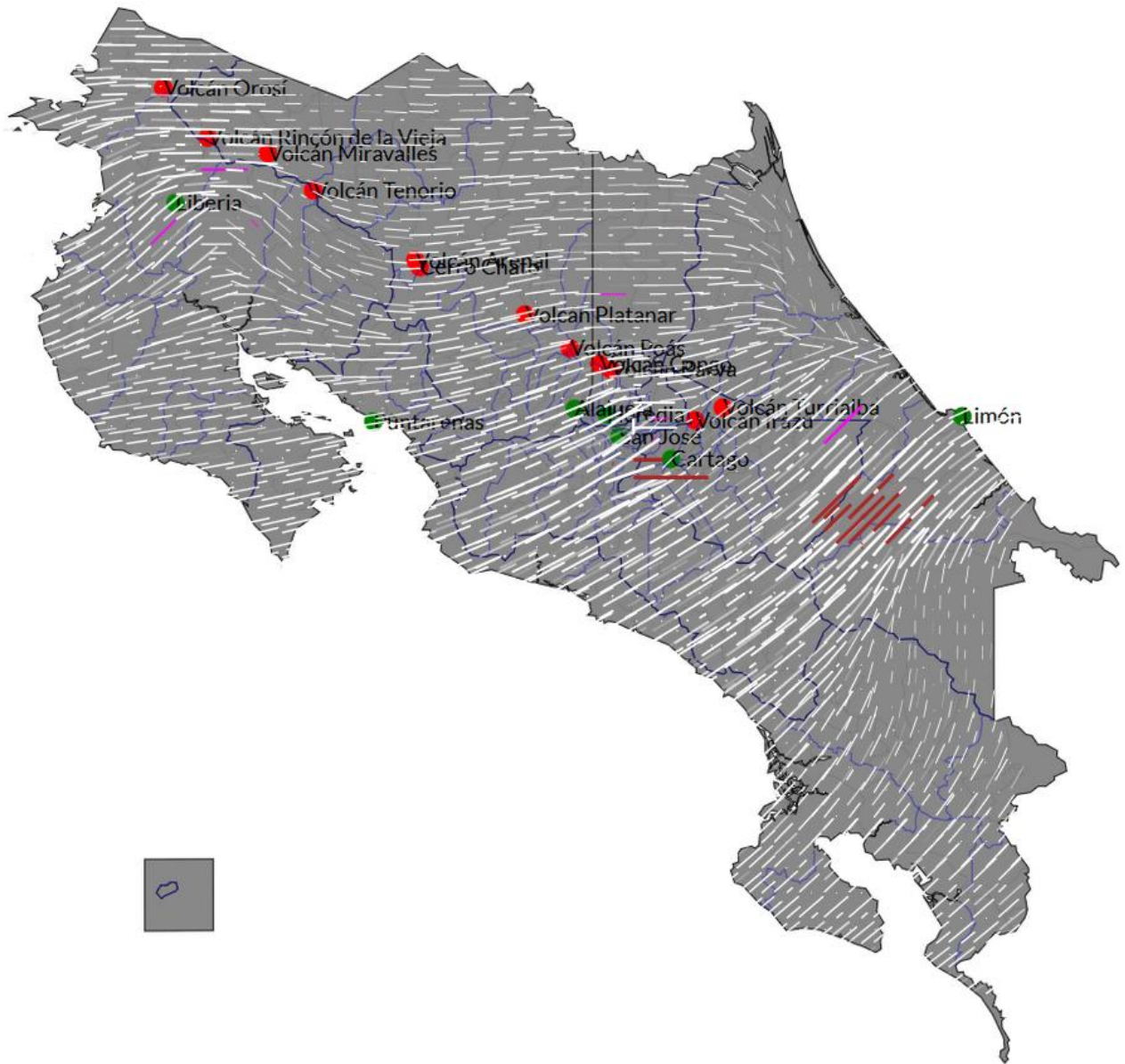


Figura 4.27. Mapa de vientos de Costa Rica, el día 9 de abril del 2014

5 Conclusiones

- Utilizar un modelo de desarrollo de *software* en espiral mitigó la falta de requerimientos presente al inicio del proyecto, las iteraciones crearon cinco versiones diferentes de la aplicación, en las cuales se hicieron cambios hasta obtener una aplicación a satisfacción del cliente.
- El modelo de desarrollo en cascada fue más eficiente para el algoritmo de interpolación, el desarrollo de este algoritmo fue estable y predecible, además la validación del cliente se realizó por medio de la animación.
- Existen diferentes factores que afectan el comportamiento del algoritmo de interpolación, estos son:
 - Potencia: la potencia afecta si la visualización tiene cambios bruscos o suaves, crear un modelo estadístico de los vientos de Costa Rica ayudaría a encontrar el valor de potencia necesario para una tener visualización certera.
 - Formato de datos: el formato de datos es importante, cuanto más exactos sean, mejor será la interpolación, datos con mucha incertidumbre crean cambios bruscos en la interpolación.
 - Posición de estaciones: los datos iniciales concentrados dejan sectores donde el algoritmo de interpolación se vuelve poco preciso. Datos iniciales distribuidos a lo largo del mapa mejoraría la visualización y el algoritmo de interpolación.
 - Cantidad de estaciones: aumentar la cantidad de estaciones también favorece en la precisión del algoritmo de interpolación.
 - Cantidad de valores: aumentar el tamaño de la matriz favorece en la visualización ya que se pueden observar las condiciones de viento en todas localizaciones con claridad.
 - Altura: Considerar la altura de los puntos iniciales evita cambios bruscos en el algoritmo de interpolación.
- Una mala animación no permite que se aprecien visualmente los buenos resultados de la interpolación.
- Se demuestra la importancia de los equipos de trabajo de múltiples áreas, ya que aunque las mediciones fuesen de carácter cualitativo, las observaciones del cliente fueron certeras para obtener una animación apropiada.

- La percepción del flujo de vientos se logra utilizando la animación asincrónica, la densidad de líneas y un buen rendimiento.
- El paralelismo en JavaScript no es una tecnología muy desarrollada, por lo que su uso actual no es tan eficiente para mejorar el rendimiento de aplicaciones.
- Cambio en aspectos como el tamaño de la matriz, el tipo de animación y simplificar las capas que se cargan, fueron suficientes para lograr un mejor tiempo de ejecución, lo que a su vez mejora el rendimiento de la animación.
- El rendimiento de la aplicación se puede mejorar al sacrificar características de D3.js como lo son los elementos SVG y el manejo de información geográfica.
- La aplicación muestra el comportamiento de las corrientes de viento en Costa Rica, esta información además se complementa con las capas de información geográfica añadidas, por lo que se puede ver cómo afecta el viento en las provincias, en los cantones, en los distritos, en las ciudades y en los volcanes.
- Se puede observar en cuáles zonas el viento llega a velocidades donde una turbina eólica alcanza su máxima potencia, logrando así un conocimiento de lugares posibles para realizar estudios especializados para la colocación de estas turbinas.
- Finalmente, el campo vectorial creado por la animación de los vientos muestra posibles resultados del flujo de partículas a través del aire, tales como la ceniza o contaminantes.

5.1 Recomendaciones

- Seguir con el uso de un modelo de desarrollo de *software* dinámico, como lo es el modelo en espiral, es importante en futuras iteraciones del proyecto. Además se recomienda tener objetivos claros para que la primera fase de requerimientos arroje resultados más precisos, sin tener que recurrir a varios ciclos de iteraciones. También es recomendada la creación de instrumentos para el análisis cualitativo, este ayuda a la toma de requerimientos en proyectos con características similares a este.

- Respecto al tema del algoritmo de interpolación para un futuro trabajo sobre este proyecto, se recomendaría trabajar sobre el modelo estadístico de los vientos en Costa Rica, esto con el fin de encontrar el valor de potencia exacto y tener una interpolación clara según las diferencias de altura a lo largo del país. También se recomienda aumentar la cantidad de estaciones, obtener datos más precisos y aumentar el tamaño del mapa y de la matriz. Estos aspectos no solo mejorarían el algoritmo de interpolación sino también la visualización de la aplicación.
- Para mejorar la percepción del flujo de las corrientes de viento, además de aumentar el tamaño del mapa y de la matriz, se debe aumentar la densidad de líneas (como al añadir líneas adicionales) y calibrar aspectos de la animación para que se adecuen a estos cambios. El rendimiento también es un aspecto importante para la percepción, por lo que se debe mejorar junto con los aspectos antes mencionados.
- Para trabajos futuros, siguiendo el tema de la animación y del rendimiento, se puede migrar la aplicación totalmente a una tecnología WebGL, ya que en este tipo de tecnologías la animación es acelerada por *hardware*, por lo que se obtiene un rendimiento mucho mayor. Al igual que el canvas, utilizar una tecnología WebGL eliminaría el manejo de mapas y de información geográfica que se encuentra en D3.js, por lo que se recomienda tener claro los objetivos antes de migrar la aplicación.
- Para simular la función de capas de D3.js en canvas o con alguna tecnología WebGL, se recomienda el uso de imágenes que representen estas capas, con la capacidad de colocarse o borrarse en memoria dinámicamente.
- El uso de información geográfica adicional mejora la usabilidad de la aplicación, añadir capas como ríos o montañas ayudaría a personas e instituciones que necesiten ese tipo de información.
- El tiempo real es un aspecto que también se puede retomar para un trabajo futuro, pese a que el Instituto Meteorológico Nacional tiene estaciones automáticas, estos datos no están disponibles para el uso público.

6 Anexos y Apéndices

Anexo 1. Código de animación de líneas.

Para la realización del algoritmo de animación de líneas se tomó como referencia en el siguiente código [44].

```
function lineAnimate(selection) {  
  selection  
    .attr({x1: 200, x2: 200})  
    .attr('y1', function(d) {return d;})  
    .attr('y2', function(d) {return d;})  
    .style('opacity', 0.5)  
    .transition()  
      .ease('linear')  
      .duration(1000)  
      .delay(function(d) {return d*10;})  
      .attr('x2', 500)  
    .transition()  
      .duration(1000)  
      .style('opacity', 0)  
    .each('end', function() {d3.select(this).call(lineAnimate)});  
}  
  
d3.select('svg')  
  .selectAll('line')  
  .data([0, 5, 10, 15, 20])  
  .enter()  
  .append('line')  
  .call(lineAnimate);
```


Anexo 2. Código de D3.js con canvas.

Para la realización del cambio de elementos SVG a canvas se tomó como referencia en el siguiente código [45].

```
<!DOCTYPE html>
<meta charset="utf-8">
<title>Custom Elements</title>
<body>
<script src="https://cdnjs.cloudflare.com/ajax/libs/d3/3.5.5/d3.min.js"
charset="utf-8"></script>
<script>

// Register the "custom" namespace prefix for our custom elements.
d3.ns.prefix.custom = "http://github.com/mbostock/d3/examples/dom";

var width = 960,
    height = 500;

// Add our "custom" sketch element to the body.
var sketch = d3.select("body").append("custom:sketch")
    .attr("width", width)
    .attr("height", height)
    .call(custom);

// On each mouse move, create a circle that increases in size and fades away.
d3.select(window).on("mousemove", function() {
    sketch.append("custom:circle")
        .attr("x", d3.event.clientX)
        .attr("y", d3.event.clientY)
        .attr("radius", 0)
        .attr("strokeStyle", "red")
        .transition()
        .duration(2000)
        .ease(Math.sqrt)
        .attr("radius", 200)
        .attr("strokeStyle", "white")
        .remove();
});

function custom(selection) {
    selection.each(function() {
        var root = this,
            canvas = root.parentNode.appendChild(document.createElement("canvas")),
            context = canvas.getContext("2d");

        canvas.style.position = "absolute";
        canvas.style.top = root.offsetTop + "px";
        canvas.style.left = root.offsetLeft + "px";

        // It'd be nice to use DOM Mutation Events here instead.
        // However, they appear to arrive irregularly, causing choppy animation.
        d3.timer(redraw);

        // Clear the canvas and then iterate over child elements.
        function redraw() {
            canvas.width = root.getAttribute("width");
            canvas.height = root.getAttribute("height");
            for (var child = root.firstChild; child; child = child.nextSibling)
                draw(child);
        }
    });
}
```

```
}

// For now we only support circles with strokeStyle.
// But you should imagine extending this to arbitrary shapes and groups!
function draw(element) {
  switch (element.tagName) {
    case "circle": {
      context.strokeStyle = element.getAttribute("strokeStyle");
      context.beginPath();
      context.arc(element.getAttribute("x"), element.getAttribute("y"),
element.getAttribute("radius"), 0, 2 * Math.PI);
      context.stroke();
      break;
    }
  }
}
});
};

</script>
```

Apéndice 1. Datos de la estación 72163 del mes de septiembre del 2014

Año	Mes	Día	Direc. Pred.	Velocidad (m/s)
2014	9	1	Este	2.12
2014	9	2	Este	2.23
2014	9	3	Este	1.37
2014	9	4	Este	1.35
2014	9	5	Este	1.58
2014	9	6	Este	1.94
2014	9	7	Este	1.41
2014	9	8	Este	1.55
2014	9	9	Suroeste	1.37
2014	9	10	Este	0.96
2014	9	11	Este	1.34
2014	9	12	Este	1.13
2014	9	13	Sureste	1.26
2014	9	14	Sureste	1.13
2014	9	15	Suroeste	1.35
2014	9	16	Este	2.02
2014	9	17	Este	1.74
2014	9	18	Este	1.17
2014	9	19	Este	1.42
2014	9	20	Este	1.02
2014	9	21	Sureste	1.08
2014	9	22	Noreste	1.15
2014	9	23	Suroeste	0.98
2014	9	24	Sureste	1.19
2014	9	25	Sureste	1.21
2014	9	26	Suroeste	1.01
2014	9	27	Norte	
2014	9	28	Este	
2014	9	29	Sureste	1.03
2014	9	30	Este	1.08

Apéndice 2. Simplificación de Geometrías

Límite exterior de Costa Rica:



Figura 6.1. Geometría del límite exterior de Costa Rica sin simplificar



Figura 6.2. Geometría del límite exterior de Costa Rica simplificada

Provincias de Costa Rica:



Figura 6.3. Geometría de las provincias de Costa Rica sin simplificar



Figura 6.4. Geometría de las provincias de Costa Rica simplificada

Cantones de Costa Rica:



Figura 6.5. Geometría de los cantones de Costa Rica sin simplificar



Figura 6.6. Geometría de los cantones de Costa Rica simplificada.

Distritos de Costa Rica:

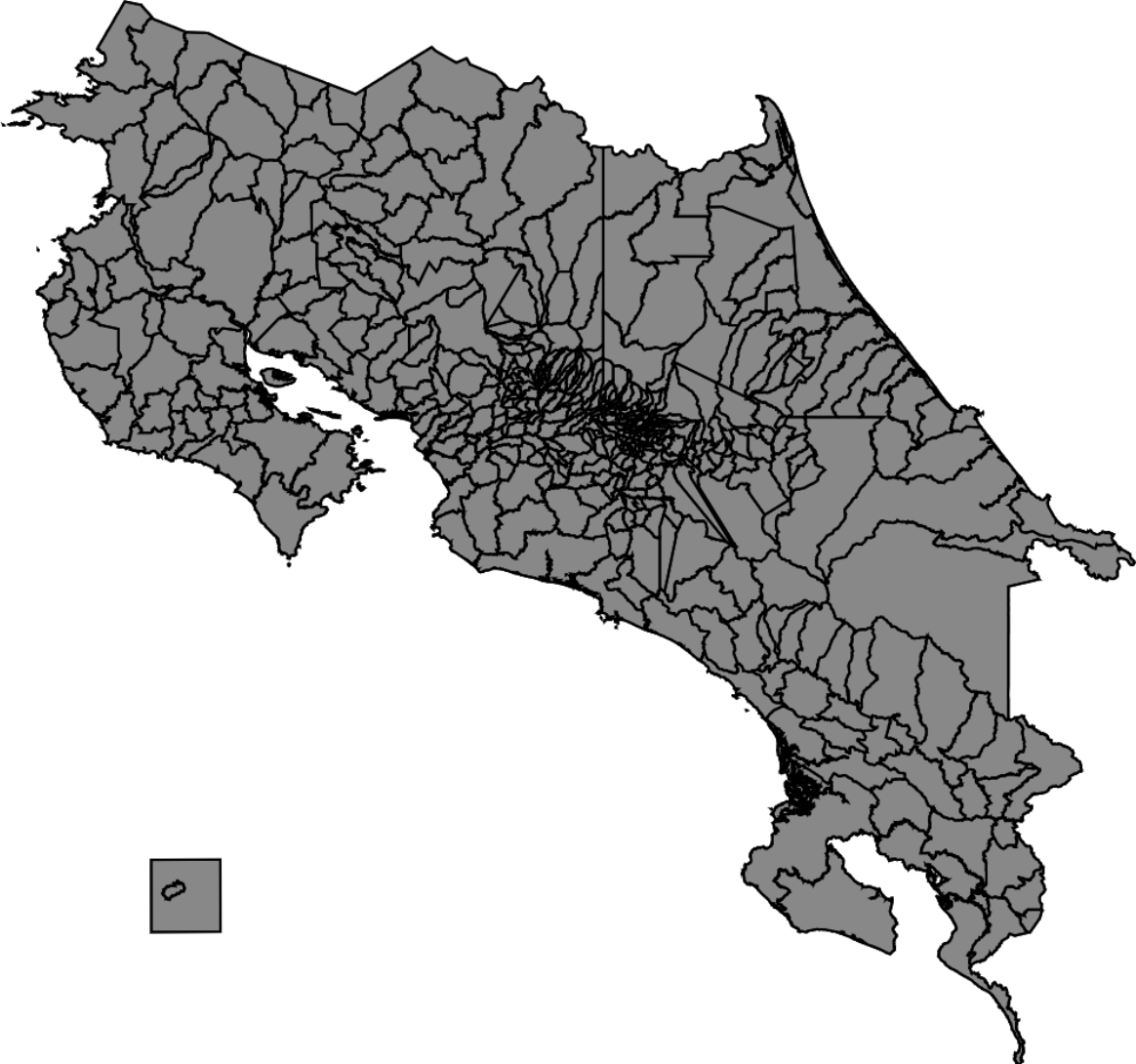


Figura 6.7. Geometría de los distritos de Costa Rica sin simplificar.

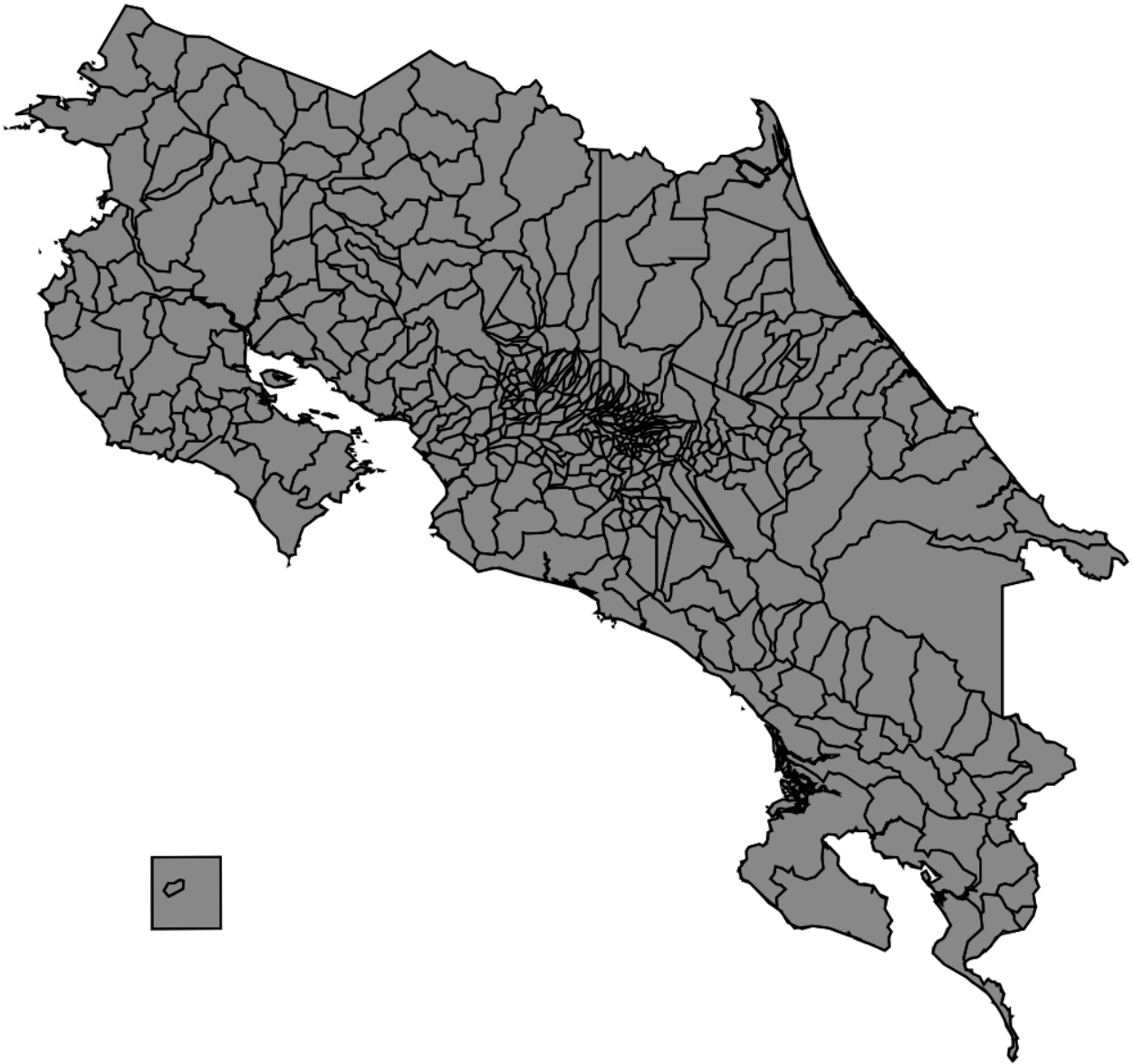


Figura 6.8. Geometría de los distritos de Costa Rica simplificada.

Apéndice 3. Datos del tiempo de ejecución de la aplicación con elementos SVG, del mes de enero para el algoritmo de interpolación.

Día	Prueba 1 (ms)	Prueba 2 (ms)	Prueba 3 (ms)	Prueba 4 (ms)	Prueba 5 (ms)
01/01/2014	2,5	2,5	2,44	2,72	2,78
02/01/2014	1,98	2,14	2,11	2,22	2,2
03/01/2014	1,62	1,67	1,88	1,64	1,61
04/01/2014	2,58	1,47	1,91	1,51	1,8
05/01/2014	2,58	1,69	2,08	1,66	1,53
06/01/2014	1,82	1,5	1,45	1,69	1,4
07/01/2014	1,44	1,4	1,22	1,31	2,72
08/01/2014	1,94	2,39	1,18	1,51	1,39
09/01/2014	1,44	1,16	1,47	1,49	1,35
10/01/2014	1,53	1,39	1,82	1,46	2,29
11/01/2014	1,38	1,3	1,48	1,43	1,21
12/01/2014	1,18	1,44	1,36	1,36	2,49
13/01/2014	1,45	1,33	1,35	1,6	2,4
14/01/2014	2,01	1,28	1,38	1,97	1,38
15/01/2014	1,6	1,37	1,46	1,38	1,38
16/01/2014	1,3	1,53	1,56	1,38	1,79
17/01/2014	1,46	1,54	1,57	1,42	1,5
18/01/2014	1,42	1,41	1,39	2,1	2,48
19/01/2014	1,67	1,4	1,37	1,54	1,44
20/01/2014	1,55	1,42	1,4	1,39	2,19
21/01/2014	2,08	1,29	1,7	1,33	1,49
22/01/2014	1,28	1,36	1,37	1,49	1,47
23/01/2014	2,38	1,28	1,72	2,6	1,4
24/01/2014	1,42	1,39	1,92	1,47	1,33
25/01/2014	2,2	1,95	1,47	1,5	1,43
26/01/2014	1,53	2,07	1,52	1,42	2,14
27/01/2014	2,35	1,46	1,33	1,3	1,37
28/01/2014	1,33	1,34	1,33	1,5	1,32
29/01/2014	1,49	1,47	1,34	2,22	1,38
30/01/2014	2,4	1,38	1,64	1,48	1,3
31/01/2014	1,67	1,41	1,39	1,65	2,78

Apéndice 4. Datos del tiempo de ejecución de la aplicación con elementos SVG, del mes de enero para el algoritmo de interpolación en paralelo.

Día	Prueba 1 (ms)	Prueba 2 (ms)	Prueba 3 (ms)	Prueba 4 (ms)	Prueba 5 (ms)
01/01/2014	2,47	2,88	2,58	2,81	2,49
02/01/2014	2,03	2,71	2,38	1,93	1,79
03/01/2014	1,8	1,8	1,65	2,09	2,18
04/01/2014	2,03	1,88	1,75	1,89	1,54
05/01/2014	2,14	1,95	1,47	1,5	1,75
06/01/2014	2,67	2,39	2,73	2,85	2,7
07/01/2014	1,4	1,42	1,5	1,62	1,63
08/01/2014	1,41	1,67	1,63	1,32	1,38
09/01/2014	1,25	1,49	1,29	1,38	1,46
10/01/2014	1,52	1,44	1,22	1,54	1,27
11/01/2014	1,69	1,71	1,74	2,04	3,2
12/01/2014	1,45	1,42	2,3	1,49	1,4
13/01/2014	1,46	1,53	1,93	1,8	1,37
14/01/2014	1,52	1,59	2,01	1,31	1,43
15/01/2014	1,45	1,46	2,07	1,45	1,45
16/01/2014	1,73	1,35	2,04	1,84	1,67
17/01/2014	1,54	1,71	1,53	2,21	1,29
18/01/2014	1,34	1,38	1,75	1,51	1,33
19/01/2014	1,47	1,31	1,42	1,28	1,27
20/01/2014	1,43	1,68	1,63	1,21	1,76
21/01/2014	1,71	2,47	1,66	1,81	2,17
22/01/2014	1,61	1,68	1,39	1,42	1,75
23/01/2014	1,32	1,36	1,65	1,54	1,46
24/01/2014	1,45	2,55	1,57	1,5	1,4
25/01/2014	1,4	1,36	1,46	1,46	1,64
26/01/2014	1,76	1,84	2,1	1,49	1,72
27/01/2014	1,53	1,47	1,36	1,34	1,51
28/01/2014	1,52	1,36	1,4	1,49	1,42
29/01/2014	1,4	1,33	1,39	1,21	1,4
30/01/2014	1,49	1,26	1,41	1,18	1,43
31/01/2014	1,95	1,65	1,48	1,71	1,76

Apéndice 5. Datos del tiempo de ejecución de la aplicación con canvas, del mes de enero para el algoritmo de interpolación.

Día	Prueba 1 (ms)	Prueba 2 (ms)	Prueba 3 (ms)	Prueba 4 (ms)	Prueba 5 (ms)
01/01/2014	2,13	2,88	2,63	2,92	2,85
02/01/2014	2,08	2,06	2,26	2,01	2,22
03/01/2014	1,56	1,58	1,56	1,64	1,78
04/01/2014	1,38	1,28	1,55	1,34	1,68
05/01/2014	1,74	1,61	1,72	1,58	1,43
06/01/2014	1,48	1,44	1,37	1,39	1,11
07/01/2014	1,34	1,43	1,97	1,41	1,57
08/01/2014	1,34	1,12	1,72	1,28	1,34
09/01/2014	1,35	1,34	1,6	2,32	1,25
10/01/2014	1,43	1,22	1,36	1,38	1,13
11/01/2014	1,28	1,48	1,49	1,4	1,17
12/01/2014	1,29	1,4	1,43	1,32	1,57
13/01/2014	1,27	1,26	1,37	1,35	1,44
14/01/2014	1,27	1,69	1,38	1,33	1,33
15/01/2014	1,37	1,27	1,38	1,25	1,49
16/01/2014	1,29	1,4	1,35	1,29	1,48
17/01/2014	1,49	1,2	1,4	1,32	2,18
18/01/2014	1,32	1,18	1,37	1,39	1,29
19/01/2014	1,44	1,3	1,37	1,3	1,38
20/01/2014	1,39	1,46	1,34	1,28	1,71
21/01/2014	1,44	1,23	1,4	1,33	1,27
22/01/2014	1,4	1,36	1,27	1,35	1,3
23/01/2014	1,15	1,38	1,29	1,47	1,35
24/01/2014	1,13	1,34	1,39	1,33	1,32
25/01/2014	1,32	1,42	1,37	1,39	1,1
26/01/2014	1,39	1,39	1,41	1,4	1,13
27/01/2014	1,39	1,27	1,31	1,3	1,97
28/01/2014	1,88	1,31	1,24	1,61	1,53
29/01/2014	1,35	1,45	1,4	1,48	1,19
30/01/2014	1,27	1,45	1,43	1,77	1,13
31/01/2014	1,38	1,48	1,32	1,76	1,1

Apéndice 6. Datos del tiempo de ejecución de la aplicación con elementos SVG, del mes de enero para el código de carga de datos de las estaciones.

Día	Prueba 1 (ms)	Prueba 2 (ms)	Prueba 3 (ms)	Prueba 4 (ms)	Prueba 5 (ms)
01/01/2014	3,05	3,11	3,75	3,11	3,42
02/01/2014	2,84	4,85	2,97	3,38	2,89
03/01/2014	2,99	2,8	2,72	2,85	2,81
04/01/2014	2,88	2,66	2,54	2,78	2,63
05/01/2014	2,8	2,51	3,16	2,53	2,5
06/01/2014	3,13	2,53	2,77	2,88	2,52
07/01/2014	2,52	2,98	2,94	2,79	3,09
08/01/2014	2,59	2,58	2,66	2,66	2,53
09/01/2014	3,05	2,47	2,5	2,6	3,06
10/01/2014	2,69	2,42	2,57	2,57	2,9
11/01/2014	2,67	2,5	3,09	3,13	2,73
12/01/2014	2,48	2,64	2,59	2,63	2,6
13/01/2014	2,54	3,15	2,69	3,2	2,64
14/01/2014	2,62	3,22	2,47	2,91	2,52
15/01/2014	3,33	3,28	2,5	2,78	3,19
16/01/2014	2,94	2,82	2,51	2,83	2,53
17/01/2014	2,82	2,82	3	2,7	2,49
18/01/2014	2,72	2,72	3,31	2,66	2,58
19/01/2014	2,54	2,58	3,22	2,99	2,56
20/01/2014	2,54	2,77	2,88	2,66	2,58
21/01/2014	3,02	2,7	2,67	2,55	3,22
22/01/2014	2,75	3,1	2,53	2,52	3,24
23/01/2014	3,38	2,5	2,58	2,55	2,84
24/01/2014	3,49	2,76	2,62	2,62	3,89
25/01/2014	3,24	2,58	3,17	3,2	2,65
26/01/2014	2,72	2,51	2,59	3,21	2,63
27/01/2014	2,78	2,47	2,48	2,95	2,6
28/01/2014	2,9	2,42	2,5	2,79	3,19
29/01/2014	2,56	2,5	2,49	2,68	2,54
30/01/2014	2,58	2,51	2,6	2,56	2,52
31/01/2014	3,07	2,63	3,13	2,53	2,6

Apéndice 7. Datos del tiempo de ejecución de la aplicación con elementos SVG y canvas, para el código de inicialización de líneas.

Prueba	Elementos SVG (ms)	Canvas (ms)
1	251,99	242,86
2	261,79	238,35
3	247,61	235,76
4	241,20	242,94
5	272,33	243,66
6	246,87	240,67
7	257,04	242,30
8	272,07	243,33
9	257,27	243,82
10	266,14	242,35

Apéndice 8. Datos del tiempo de ejecución de la aplicación con elementos SVG y canvas, por línea.

Prueba	Elementos SVG (ms)	SVG con líneas adicionales (ms)	Canvas (ms)
1	0,26	0,41	0,18
2	0,26	0,41	0,18
3	0,26	0,41	0,18
4	0,26	0,41	0,17
5	0,26	0,43	0,23
6	0,26	0,41	0,17
7	0,28	0,40	0,17
8	0,26	0,41	0,17
9	0,26	0,41	0,22
10	0,28	0,42	0,17
11	0,29	0,41	0,18
12	0,27	0,41	0,17
13	0,28	0,40	0,23
14	0,26	0,45	0,17
15	0,26	0,44	0,18
16	0,28	0,42	0,17
17	0,26	0,40	0,21
18	0,26	0,40	0,18
19	0,27	0,59	0,17
20	0,41	0,47	0,18
21	0,41	0,55	0,22
22	0,43	0,45	0,17
23	0,45	0,41	0,17
24	0,29	0,41	0,21
25	0,27	0,43	0,18
26	0,28	0,40	0,17
27	0,47	0,41	0,18
28	0,28	0,42	0,17
29	0,28	0,39	0,18
30	0,27	0,39	0,19
31	0,26	0,42	0,17
32	0,26	0,41	0,18
33	0,27	0,39	0,17
34	0,28	0,40	0,18
35	0,27	0,39	0,17
36	0,26	0,41	0,17
37	0,27	0,39	0,17
38	0,29	0,39	0,19
39	0,26	0,39	0,18
40	0,25	0,43	0,18
41	0,27	0,45	0,17
42	0,27	0,39	0,17
43	0,27	0,39	0,18
44	0,25	0,39	0,18
45	0,26	0,41	0,31
46	0,26	0,39	0,18
47	0,25	0,38	0,18
48	0,26	0,41	0,18
49	0,26	0,43	0,17
50	0,26	0,40	0,18
51	0,26	0,39	0,18
52	0,25	0,38	0,21
53	0,26	0,43	0,19
54	0,27	0,46	0,18
55	0,26	0,38	0,17
56	0,26	0,47	0,18

57	0,26	0,38	0,17
58	0,25	0,41	0,17
59	0,28	0,39	0,17
60	0,26	0,39	0,22
61	0,26	0,39	0,17
62	0,27	0,43	0,17
63	0,25	0,41	0,17
64	0,26	0,38	0,22
65	0,26	0,39	0,17
66	0,25	0,45	0,18
67	0,26	0,43	0,17
68	0,26	0,39	0,18
69	0,28	0,38	0,17
70	0,41	0,39	0,22
71	0,25	0,40	0,18
72	0,27	0,43	0,18
73	0,43	0,38	0,18
74	0,32	0,38	0,17
75	0,27	0,41	0,17
76	0,28	0,43	0,17
77	0,26	0,41	0,17
78	0,37	0,40	0,17
79	0,32	0,39	0,17
80	0,32	0,43	0,18
81	0,28	0,44	0,18
82	0,28	0,39	0,21
83	0,28	0,38	0,18
84	0,28	0,38	0,18
85	0,28	0,43	0,23
86	0,46	0,62	0,18
87	0,33	0,45	0,18
88	0,29	0,39	0,21
89	0,27	0,39	0,17
90	0,37	0,41	0,18
91	0,34	0,39	0,17
92	0,28	0,38	0,18
93	0,14	0,44	0,17
94	0,13	0,38	0,18
95	0,13	0,39	0,18
96	0,13	0,39	0,18
97	0,38	0,47	0,17
98	0,29	0,45	0,18
99	0,28	0,38	0,18
100	0,28	0,38	0,18

Apéndice 9. Datos del tiempo de ejecución de la aplicación con acercamiento y canvas, por línea.

Prueba	Canvas con mínimo acercamiento	Canvas con máximo acercamiento	Canvas
1	0,22	0,38	0,18
2	0,22	0,37	0,18
3	0,22	0,37	0,18
4	0,22	0,37	0,17
5	0,22	0,37	0,23
6	0,22	0,36	0,17
7	0,23	0,36	0,17
8	0,22	0,36	0,17
9	0,22	0,37	0,22
10	0,22	0,36	0,17
11	0,22	0,37	0,18
12	0,24	0,36	0,17
13	0,24	0,37	0,23
14	0,26	0,37	0,17
15	0,25	0,37	0,18
16	0,21	0,36	0,17
17	0,22	0,36	0,21
18	0,22	0,37	0,18
19	0,21	0,36	0,17
20	0,21	0,37	0,18
21	0,29	0,47	0,22
22	0,22	0,36	0,17
23	0,22	0,37	0,17
24	0,23	0,39	0,21
25	0,26	0,36	0,18
26	0,23	0,38	0,17
27	0,23	0,37	0,18
28	0,21	0,36	0,17
29	0,22	0,36	0,18
30	0,22	0,36	0,19
31	0,24	0,37	0,17
32	0,22	0,36	0,18
33	0,22	0,47	0,17
34	0,32	0,37	0,18
35	0,23	0,36	0,17
36	0,24	0,36	0,17
37	0,23	0,36	0,17
38	0,22	0,37	0,19
39	0,23	0,37	0,18
40	0,22	0,36	0,18
41	0,22	0,36	0,17
42	0,22	0,36	0,17
43	0,23	0,36	0,18
44	0,22	0,36	0,18
45	0,22	0,36	0,31
46	0,23	0,42	0,18
47	0,23	0,35	0,18
48	0,22	0,44	0,18
49	0,22	0,36	0,17
50	0,23	0,36	0,18
51	0,22	0,36	0,18
52	0,22	0,35	0,21
53	0,22	0,37	0,19
54	0,22	0,37	0,18
55	0,22	0,37	0,17
56	0,31	0,36	0,18
57	0,32	0,35	0,17
58	0,39	0,36	0,17

59	0,34	0,37	0,17
60	0,39	0,36	0,22
61	0,34	0,36	0,17
62	0,37	0,36	0,17
63	0,36	0,36	0,17
64	0,24	0,36	0,22
65	0,23	0,35	0,17
66	0,22	0,36	0,18
67	0,22	0,36	0,17
68	0,23	0,36	0,18
69	0,22	0,36	0,17
70	0,22	0,35	0,22
71	0,23	0,37	0,18
72	0,22	0,36	0,18
73	0,22	0,37	0,18
74	0,22	0,36	0,17
75	0,22	0,36	0,17
76	0,23	0,37	0,17
77	0,23	0,36	0,17
78	0,23	0,36	0,17
79	0,23	0,35	0,17
80	0,22	0,36	0,18
81	0,22	0,36	0,18
82	0,22	0,36	0,21
83	0,23	0,35	0,18
84	0,22	0,36	0,18
85	0,23	0,37	0,23
86	0,22	0,38	0,18
87	0,23	0,38	0,18
88	0,22	0,35	0,21
89	0,23	0,40	0,17
90	0,23	0,37	0,18
91	0,22	0,36	0,17
92	0,22	0,37	0,18
93	0,26	0,38	0,17
94	0,22	0,36	0,18
95	0,23	0,36	0,18
96	0,22	0,36	0,18
97	0,22	0,49	0,17
98	0,27	0,36	0,18
99	0,27	0,37	0,18
100	0,22	0,36	0,18

Apéndice 10. Datos del tiempo de ejecución de la aplicación con elementos SVG, para la carga del mapa y capas sin simplificación.

Prueba	Mapa completo (ms)	Límite exterior (ms)	Provincias (ms)	Cantones (ms)	Distritos (ms)
1	355,68	66,15	12,43	54,35	241,89
2	367,61	64,91	12,80	55,26	240,87
3	360,14	66,70	12,14	54,25	243,68
4	353,73	64,47	13,00	53,82	238,90
5	360,11	65,88	12,43	53,55	239,87
6	359,95	64,22	12,09	53,53	234,42
7	357,26	64,06	12,47	53,78	234,43
8	369,53	64,16	12,55	55,77	238,40
9	369,13	64,17	12,20	55,52	235,15
10	360,98	65,21	12,12	53,09	235,65

Apéndice 11. Datos del tiempo de ejecución de la aplicación con elementos SVG, para la carga de capas con simplificación de geometrías.

Prueba	Límite exterior (ms)	Provincias (ms)	Cantones (ms)	Distritos (ms)
1	15,34	2,42	1,69	1,97
2	14,31	1,45	1,54	1,56
3	15,32	1,87	1,43	1,49
4	15,06	1,83	1,40	1,74
5	14,67	2,75	1,67	1,61
6	15,00	1,57	1,39	1,53
7	14,69	1,81	1,45	1,55
8	14,44	1,88	1,60	1,51
9	14,65	2,10	1,68	1,46
10	15,00	2,00	1,83	1,50

7 Referencias bibliográficas

- [1] Tecnológico de Costa Rica, “Presentación.” [En línea]. Disponible en: <http://www.tec.ac.cr/eltec/Paginas/default.aspx>.
- [2] Tecnológico de Costa Rica, “Acerca Del Tec.” [En línea]. Disponible en: <http://www.tec.ac.cr/eltec/Paginas/acercaDelTec.aspx>.
- [3] J. Fallas Sojo, “BREVE HISTORIA DEL INSTITUTO METEOROLÓGICO NACIONAL (1888-2005),” [En línea]. Disponible en: <http://www.imn.ac.cr/sobreimn/historia.html>.
- [4] Instituto Meteorológico Nacional, “Sobre IMN.” [En línea]. Disponible en: <http://www.imn.ac.cr/sobreimn/index.html>.
- [5] C. Palese, J. Gardiol, y J. Lassig, “MAPAS DE VELOCIDAD MEDIA DEL VIENTO EN EL CENTRO-ESTE DE LA PROVINCIA DE NEUQUEN.” FI-UNC, 2010.
- [6] P. Cook, “Animated UK Wind Chart.” [En línea]. Disponible en: <http://charts.animateddata.co.uk/ukwind/>.
- [7] HINT.FM, “Wind Map.” [En línea]. Disponible en: <http://hint.fm/wind/>.
- [8] C. Beccario, “air,” 2014. [En línea]. Disponible en: <https://github.com/cambecc/air>.
- [9] C. Beccario, “earth,” 2015. [En línea]. Disponible en: <https://github.com/cambecc/earth>.
- [10] Blanes Zamora, Sergio, Ginestar Peiró, Damián, y Roselló, Ferragud, María Dolores. Introducción a los métodos numéricos para ecuaciones diferenciales. España: Editorial de la Universidad Politécnica de Valencia, 2013. ProQuest ebrary.
- [11] R. Sibson, “A Brief Description of Natural Neighbor Interpolation,” en *Interpolating Multivariate Data*, New York: John Wiley & Sons, 1981.
- [12] D. F. Watson y G. M. Philip, ““A Refinement of Inverse Distance Weighted Interpolation,”” en *Geoprocessing*, 1985.
- [13] A. M. Felicísimo, “Tema 1 Conceptos Construcción del MDE la interpolación.” [En línea]. Disponible en: <http://www6.uniovi.es/~feli/CursoMDT/Tema1/Page16.html>.

- [14] M. A. Oliver, “Kriging: A Method of Interpolation for Geographical Information Systems,” en *International Journal of Geographic Information Systems*, 1990.
- [15] A. Beléndez, J. Bernabeu, y C. Pastor, “Campos escalares y vectoriales,” en *Magnitudes, vectores y campos*, Universidad Politécnica de Valencia, 1988.
- [16] Facultad de Ciencias Exactas, “ANÁLISIS MATEMÁTICO II.” Universidad Nacional de La Plata, 2014.
- [17] Grupo de Investigación Gestión del Ambiente para el Bienestar Social, “Cinemática: El campo de velocidad.” [En línea]. Disponible en: <http://fluidos.eia.edu.co/fluidos/cinematica/cinematica.html>.
- [18] WindPower Program, “Wind turbine power output variation with steady wind speed.” [En línea]. Disponible en: http://www.wind-power-program.com/turbine_characteristics.htm.
- [19] Agencia Insular de Energía de Tenerife, “INFORMACIÓN GENERAL SOBRE ENERGÍA EÓLICA.”
- [20] J. Solano y R. Villalobos, “REGIONES Y SUBREGIONES CLIMÁTICAS DE COSTA RICA.” Instituto Meteorológico Nacional.
- [21] Laboratorio Nacional de Calidad del Software, “INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA.” INTECO, 2009.
- [22] The Khronos Group, “OpenGL Overview.” [En línea]. Disponible en: <https://www.opengl.org/about/>.
- [23] D. Burns y R. Osfield, “Features.” [En línea]. Disponible en: <http://www.openscenegraph.org/index.php/about/features>.
- [24] Pelican mapping, “osgEarth.” [En línea]. Disponible en: <http://osgearth.org/>.
- [25] The Khronos Group, “Getting Started.” [En línea]. Disponible en: https://www.khronos.org/webgl/wiki/Getting_Started.
- [26] Mr.doob, “three.js.” [En línea]. Disponible en: <https://github.com/mrdoob/three.js>.
- [27] C. Pinson, “OSG.JS WebGL framework.” [En línea]. Disponible en:

<https://github.com/cedricpinson/osgjs>.

- [28] P. Brunt, “What is GLGE?” [En línea]. Disponible en: <http://www.glge.org/>.
- [29] XeoLabs, “SceneJS 3D Engine for the web.” [En línea]. Disponible en: <http://scenejs.org/>.
- [30] M. Di Benedetto y M. Potenziani, “Welcome to SpiderGL.org.” [En línea]. Disponible en: <http://spidergl.org/>.
- [31] M. Christen, “About.” [En línea]. Disponible en: <http://www.openwebglobe.org/>.
- [32] Go Software, “Glob3 Mobile: a 3D virtual globe for web browsers y mobile devices.” [En línea]. Disponible en: <http://ami.dis.ulpgc.es/glob3m/>.
- [33] Klokian Technologies, “WebGL Earth JavaScript API.” [En línea]. Disponible en: <http://www.webglearth.org/>.
- [34] M. Bostok, “Data-Driven Documents,” 2015. [En línea]. Disponible en: <http://d3js.org/>.
- [35] W3Schools, “HTML5 Web Workers,” 1999–2015. [En línea]. Disponible en: http://www.w3schools.com/html/html5_webworkers.asp.
- [36] A. Savitzky, “Parallel.js,” 2015. [En línea]. Disponible en: <http://adambom.github.io/parallel.js/>.
- [37] Semantic Org, “Semantic UI.” [En línea]. Disponible en: <http://semantic-ui.com/>.
- [38] GRASS Development Team, “GRASS GIS,” 1998–2015. [En línea]. Disponible en: <https://grass.osgeo.org/>.
- [39] QGIS Development Team, “Descubre que hace de QGIS el SIG líder de Código Abierto para escritorio,” 2002–2015. [En línea]. Disponible en: <http://www.qgis.org/es/site/about/index.html>.
- [40] S. Carter, “Mr. Data Converter,” 2014. [En línea]. Disponible en: <https://shancarter.github.io/mr-data-converter/>.
- [41] A. Arce, “Costa Rica TopoJSON.” [En línea]. Disponible en: <https://gist.github.com/arce/9357998>.

[42] J. Paine, “GeoJSON-TopoJSON,” 2014. [En línea]. Disponible en: <http://jeffpaine.github.io/geojson-topojson/>.

[43] P. Le Bek, “Speeding Up D3.js: A Checklist.” [En línea]. Disponible en: <https://www.safaribooksonline.com/blog/2014/02/20/speeding-d3-js-checklist/>.

[44] P. Cook, “Making the animated UK wind chart.” [En línea]. Disponible en: <http://prcweb.co.uk/making-the-uk-wind-chart/>.

[45] M. Bostock, “DOM-to-Canvas using D3,” 2011. [En línea]. Disponible en: <http://bl.ocks.org/mbostock/1276463>.