



**UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**SISTEMA WEB PARA LA VISUALIZACIÓN
DE CALIDAD DEL AIRE**

FERNANDO RENÉ GÓMEZ SEPÚLVEDA

Profesores Guía: BENJAMÍN INGRAM
FEDERICO MEZA

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Junio, 2017

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Curicó, 2019

Dedicado a mi familia y amigos.

AGRADECIMIENTOS

En primer lugar agradecer a mi padre Carlos Gómez, mi madre Carmen Sepúlveda y a mis hermanos Carlos y Pablo, por el esfuerzo y apoyo en todos estos años de estudios.

Agradezco a cada una de las personas que creyeron en mí y que me acompañaron en este camino. Familia, amigos, compañeros y también a las personas que ya no están en este mundo.

También agradecer a los profesores guías, ya que tuve la suerte de contar con dos profesores, Benjamín Ingram y Federico Meza. Agradecer por la gran disposición y la ayuda durante el desarrollo de este proyecto.

Finalmente me gustaría agradecer a mi médico que me acompañó y ayudó por varios años Dr. Augusto Aspillaga.

RESUMEN

La calidad del aire constituye uno de los temas ambientales que más directamente afectan a la población. Pese a los esfuerzos y a los distintos instrumentos utilizados, el país aún no cumple con los estándares establecidos en las normas de calidad.

Múltiples estudios nacionales e internacionales han mostrado que existe una asociación entre el nivel de concentración de contaminantes como material particulado (MP), ozono (O_3), dióxido de azufre (SO_2) y dióxido de nitrógeno (NO_2) y la incidencia de muertes prematuras y varias enfermedades cardiorrespiratorias, tanto en niños como en adultos. Además, existe evidencia de efectos ambientales, tales como disminución de visibilidad, daños a los materiales e impactos en la flora y fauna. El material particulado es el contaminante que más significativamente ha sido asociado a eventos de mortalidad y morbilidad en la población.

El objetivo de este proyecto de título, es implementar un prototipo de un sistema web basado en un Sistema de Información Geográfica (SIG), que sirva de ayuda y que facilite a los habitantes y a los fiscalizadores la toma de decisiones con respecto a los niveles de contaminación presente en su localidad. El enfoque de esta memoria está puesto en el procesamiento de la información sobre la contaminación del aire, y a partir de ésta, generar imágenes que visualmente interpreten los niveles de contaminación en un mapa. Dicho sistema web fue construido mediante las tecnologías JavaScript, HTML PHP y el uso del programa estadístico R. El procedimiento utilizado para el desarrollo y construcción, fue siguiendo el proceso que describe la metodología de desarrollo de software ágil SCRUM. Finalmente en base a lo construido se realizan pruebas que permiten validar que dicho sistema web cumple con lo estipulado en los objetivos.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VII
Índice de Tablas	IX
Resumen	x
1. Introducción	11
1.1. Contexto del Proyecto	11
1.2. Trabajo Relacionado	12
1.3. Definición del Problema	12
1.4. Objetivos	13
1.5. Alcances	14
1.6. Trabajo Complementario	14
2. Marco Teórico	15
2.1. Contaminación Atmosférica	15
2.1.1. Material Particulado (MP)	15
2.1.2. MP 2.5	16
2.1.3. MP 10	16
2.2. Sistema de Información Geográfica	17
2.2.1. Modelos de Datos Espaciales	18
2.3. Interpolación Espacial	21
2.3.1. Métodos de Interpolación	22
2.4. Arquitectura Cliente-Servidor	23
2.4.1. Base de Datos Geográfica	23
2.4.2. PostgreSQL PostGIS	23

2.4.3.	MapServer	24
2.4.4.	R	24
2.4.5.	Openlayers	25
2.5.	Sistemas Existentes	26
2.5.1.	Sistema de Información Nacional de Calidad del Aire (SINCA)	26
3.	Metodología	27
3.1.	Scrum	27
3.1.1.	Características de Scrum	28
3.1.2.	Equipos Scrum	28
3.1.3.	Eventos	28
3.2.	Aplicación de Metodología	29
3.2.1.	Etapas de Desarrollo	29
4.	Diseño del Sistema	32
4.1.	Tecnologías Involucradas	32
4.1.1.	Herramientas	32
4.1.2.	Lenguajes de Programación	33
4.2.	Historias de Usuario	35
4.3.	Diseño de la Arquitectura y Procesos	36
4.3.1.	Arquitectura	37
4.3.2.	Proceso de Generación de Imagen	38
4.3.3.	Proceso de Visualización	40
4.4.	Diseño de la Base de Datos	40
4.4.1.	Modelo Entidad-Relación y Relacional	41
4.4.2.	Entidades	43
4.5.	Diseño de la Interfaz de Usuario	46
4.5.1.	Mocukp Inicial	46
5.	Desarrollo del Sistema	49
5.1.	Instalación y Configuración	49
5.1.1.	Características del Servidor	49
5.1.2.	PostgreSQL/PostGIS	50
5.1.3.	MapServer	51
5.2.	Implementación de la Base de Datos	53

5.2.1.	Procesamiento de Datos	54
5.2.2.	Generación de Imágenes	55
5.3.	Implementación del Cliente Web	63
5.3.1.	Interfaz	64
5.3.2.	Implementación Mapa base	65
5.3.3.	Controles de Mapa	67
5.3.4.	Implementación de los Gráficos	69
6.	Pruebas	72
6.1.	Funcionalidad	72
6.1.1.	Datos de Prueba	72
6.1.2.	Comparación	74
6.1.3.	Resultado	77
6.2.	Evaluación de Usabilidad	77
6.2.1.	Encuesta de Usabilidad	77
6.2.2.	Resultados	78
7.	Conclusiones	83
7.1.	Trabajo Futuro	84
	Glosario	85
	Bibliografía	86
	Anexos	
A:	Archivo Mapfile	89
A.1.	Capa MP10	89
A.2.	Capa MP2.5	91
B:	Base de Datos SQL	94
B.1.	Base de datos	94
B.2.	Vista Parcial	96
B.3.	Trigger	96

C: Script en R	98
C.1. Interpolación IDW	98
C.2. Interpolación Kriging	99
D: Encuesta de Usabilidad	101

ÍNDICE DE FIGURAS

	página
2.1. Comparación entre MP 2.5 y 10. Fuente: https://es.wikipedia.org/wiki/PM10	17
2.2. Elementos que forman el sistema SIG. Fuente: http://volaya.github.io/libro-sig/chapters/Introduccion_fundamentos.html	18
2.3. Modelos raster y vectorial. Fuente: http://geoservice.igac.gov.co/contenidos_telecentro/fundamentos_sig/cursos/sem_2/uni2/index.php?id=2	19
2.4. Representación vectorial y ejemplos particulares de cada una de ellas con atributos asociados. Fuente: http://volaya.github.io/libro-sig/chapters/Tipos_datos.html	20
2.5. Representación raster. Fuente: http://geoservice.igac.gov.co/contenidos_telecentro/fundamentos_sig/cursos/sem_2/uni2/index.php?id=24	21
2.6. Openlayers	25
2.7. SINCA. Fuente: http://sinca.mma.gob.cl/index.php/	26
4.1. Modelo de la arquitectura usada.	37
4.2. Proceso para la generación de imagen.	39
4.3. Proceso de generación y visualización de mapas e imágenes.	40
4.4. Modelo entidad-relación	42
4.5. Modelo Relacional base de datos	43
4.6. Mockup inicial del sistema.	47
5.1. Imagen .TIF resultado de interpolación IDW.	57
5.2. Gráfico del Semi-Variograma.	59
5.3. Gráfico del SemiVariograma, Modelo Exponencial.	60
5.4. Gráfico del SemiVariograma, Modelo Esférico.	61
5.5. Gráfico del SemiVariograma, Modelo Lineal.	62
5.6. Imagen .TIF resultado de interpolación Kriging.	63
5.7. Página inicial del sistema.	64
5.8. Interfaz del gráfico de datos.	65
5.9. Interfaz de Controles de Mapa.	68
5.10. Interfaz gráfico.	71

6.1. Comparación entre imágenes IDW.	75
6.2. Comparación entre imágenes Kriging.	76
6.3. Nivel de conocimiento de informática.	79
6.4. Gráfico Pregunta de Interfaz.	80
6.5. Gráfico Pregunta sobre Imagen.	80
6.6. Gráfico Pregunta sobre Interacción.	81

ÍNDICE DE TABLAS

	página
5.1. Parte de la tabla Excel de registros.	54
5.2. Parte de la tabla Excel de estaciones.	55
6.1. Datos de Prueba.	74

1. Introducción

1.1. Contexto del Proyecto

Los efectos en la salud provocados por la contaminación atmosférica cuentan con evidencia científica cada vez más irrefutable, que proviene de una variedad de estudios epidemiológicos realizados tanto en el contexto poblacional como ocupacional, además de estudios toxicológicos y de exposición controlada en seres humanos. En especial para el material particulado, se ha acumulado evidencia que apoya su relación causal con mortalidad prematura de causa cardiovascular, respiratoria y cáncer pulmonar, así como un sinnúmero de efectos en la morbilidad: aumento de las hospitalizaciones por cardiopatía coronaria, insuficiencia cardíaca, asma bronquial, enfermedad pulmonar obstructiva crónica (EPOC), entre otras [17].

El material particulado es una mezcla de partículas líquidas y sólidas, de sustancias orgánicas e inorgánicas, que se encuentran en suspensión en el aire. El material particulado forma parte de la contaminación del aire. Su composición es muy variada y podemos encontrar, entre sus principales componentes, sulfatos, nitratos, el amoníaco, el cloruro sódico, el carbón, el polvo de minerales, cenizas metálicas y agua. Dichas partículas además producen reacciones químicas en el aire.

Durante los últimos años en la región del Maule, se ha percibido un aumento en las alertas con respecto a la contaminación del aire. Esto se debe, entre otras causas al uso indebido de distintos sistemas de calefacción. En la actualidad, el monitoreo de estas mediciones y generación de alertas son realizadas por el Ministerio del Medio Ambiente, utilizando una red de estaciones en todo Chile. El sistema actual para la visualización de estos datos es básico. [11].

El problema del sistema de monitoreo existente es el bajo análisis que se puede realizar a los datos solamente con la información que se entrega al usuario, por lo

cual se propone la implementación de un Sistema de Información Geográfica para la representación y visualización de estos datos.

Un Sistema de Información Geográfica o SIG (GIS en su acrónimo inglés, Geographic Information Systems) es un sistema o software especializado que permite al usuario crear consultas interactivas, integrar, analizar y representar de una forma simple y eficiente información geográfica referenciada asociada a un territorio, conectando mapas con bases de datos[12].

El uso de estos sistemas permite facilitar la visualización de datos en un mapa, con el objetivo de reflejar fenómenos geográficos de cualquier tipo, desde un mapa de carreteras, hasta la identificación de parcelas o densidad de población, entre otros, facilitando la labor de quien deba tomar decisiones respecto a dichos datos.

1.2. Trabajo Relacionado

El principal sistema de información geográfica para visualizar los datos de la calidad de aire es la página del Ministerio de Medio Ambiente llamada Sistema de Información Nacional de Calidad del Aire (SINCA), cuyo objetivo es poner a disposición información de calidad de aire de todo el país, buscando mejorar gradualmente el conocimiento, la vigilancia y la gestión de la calidad del aire que respiramos [11].

Otro ejemplo es World Air Quality Index, un proyecto de empresa social se inició en 2007. Su misión es promover el conocimiento de la contaminación del aire y proporcionar información de la calidad del aire unificada, para todo el mundo, en tiempo real a través de sus sitios [15].

Ambos ejemplos son muy parecidos en su implementación, pues muestran los datos de la contaminación y un historial de las mediciones obtenidas. Esto lo hacen de forma muy básica, mostrando las estaciones con sus valores actuales de medición, World Air Quality Index se basa de los datos obtenidos por SINCA con respecto a Chile.

1.3. Definición del Problema

A nivel nacional hoy en día no existe un sistema de información geográfica que contenga los datos de contaminación de una forma que se pueda ver claramente los lugares o algunos puntos en concreto y la forma como varían las mediciones

de la contaminación. Esto trae un problema tanto para las personas que viven en la ciudad como también para las entidades encargadas de fiscalizar a los posibles contaminadores.

Mencionado lo anterior, actualmente se necesita un modelo de estos datos para que sea más fácil el estudio y el análisis de la situación actual. Con esto se puede tomar diferentes tipos de acciones para poder controlar y reducir estos niveles de contaminación, como también mejorar las fiscalizaciones realizadas cuando se esté en presencia de alguna alerta.

Las motivaciones que impulsaron el desarrollo de este proyecto son dos: la importancia de analizar y detectar cuándo los niveles de contaminación sobrepasen la norma establecida, ya que con esto se pueden tomar diferentes tipos de acciones para que mejore esta condición; y poder informar a la población de lo que está pasando en su localidad, sobre todo para los deportistas que sin saberlo pueden estar en peligro al practicar su deporte.

1.4. Objetivos

Este proyecto se guió por los siguientes objetivos.

Objetivo General

- Desarrollar un prototipo funcional de un sistema web para la visualización de la calidad del aire.

Objetivos Específicos

- Diseñar y seleccionar la arquitectura necesaria para implementar el sistema de información geográfica.
- Diseñar base de datos para el sistema.
- Crear datos y cargarlos en la base de datos.
- Investigar métodos de interpolación automática para la generación de mapas.
- Construir un prototipo funcional para el sistema.
- Evaluar el sistema de visualización.

1.5. Alcances

- Este trabajo se limita a la ciudad de Curicó.
- Este trabajo no dispondrá de un registro de usuario.
- Este trabajo se limita al desarrollo de un prototipo, esto quiere decir que no se espera tener el sistema web en un servidor de producción.
- En cuanto a los datos, no serán reales si no que generados por una herramienta de simulación y con un mínimo de 50 estaciones.
- Este trabajo no estará optimizado para dispositivos móviles.

1.6. Trabajo Complementario

Este proyecto se complementa con otra memoria, ambas memorias se realizaron paralelamente. La memoria complementaria se basa en la elección de una estación de medición para la contaminación del aire, éste proyecto generará datos de los distintos contaminantes y los insertará en la base de datos, la cual fue diseñada entre ambos estudiantes para lograr el correcto funcionamiento de los dos sistemas.

2. Marco Teórico

En este capítulo se describe el marco teórico necesario para el entendimiento de todo el desarrollo del proyecto, así como para la interpretación de resultados y la formulación de conclusiones.

2.1. Contaminación Atmosférica

Los efectos en la salud provocados por la contaminación atmosférica cuentan con evidencia científica cada vez más irrefutable, que proviene de una variedad de estudios epidemiológicos realizados tanto en el contexto poblacional como ocupacional, además de estudios toxicológicos y de exposición controlada en seres humanos. En especial para el material particulado, se ha acumulado evidencia que apoya su relación causal con mortalidad prematura de causa cardiovascular, respiratoria y cáncer pulmonar, así como un sinnúmero de efectos en la morbilidad: aumento de las hospitalizaciones por cardiopatía coronaria, insuficiencia cardíaca, asma bronquial, enfermedad pulmonar obstructiva crónica (EPOC), entre otras [17].

La contaminación atmosférica puede ser por varios tipos de contaminantes. Para este proyecto el material particulado (MP) será el único contaminante que va ser implementado, pues es el que afecta de mayor manera a nuestro cuerpo, además de ser uno de los contaminantes más relevantes en nuestra región y país. Este tipo de material puede darse en diferentes tamaños. En este proyecto trabajamos con el MP10 y el MP2.5.

2.1.1. Material Particulado (MP)

Es una mezcla de partículas líquidas y sólidas suspendidas en el aire que difieren en tamaño, composición y origen. El tamaño de las partículas suspendidas

en la atmósfera varía en más de cuatro órdenes de magnitud, desde unos pocos nanómetros a decenas de micrómetros, en la Fig. 2.1 se puede apreciar la comparación de tamaños. El Material Particulado conviene clasificarlo por sus propiedades aerodinámicas (Diámetro Aerodinámico), dado que éstas son un factor decisivo para el transporte y la remoción de las partículas desde el aire. También, son determinantes para el sistema respiratorio y están asociadas con la composición química y las fuentes de esas partículas. Cuando se habla del tamaño de una partícula, se está hablando de su diámetro aerodinámico.

2.1.2. MP 2.5

Corresponde a las partículas de diámetro menor a $2.5 \mu\text{m}$. Representa una mezcla compleja de sustancias orgánicas e inorgánicas. Estas partículas pueden ser generadas de forma natural o por la acción del hombre. Son más perjudiciales que las MP10, ya que su tamaño es menor por lo que penetran de mayor manera en el sistema respiratorio, provocando un mayor daño.

2.1.3. MP 10

Corresponde a las partículas de diámetro menor a $10 \mu\text{m}$. Representa una mezcla compleja de sustancias orgánicas e inorgánicas. Estas partículas pueden ser generadas de forma natural o por la acción del hombre.

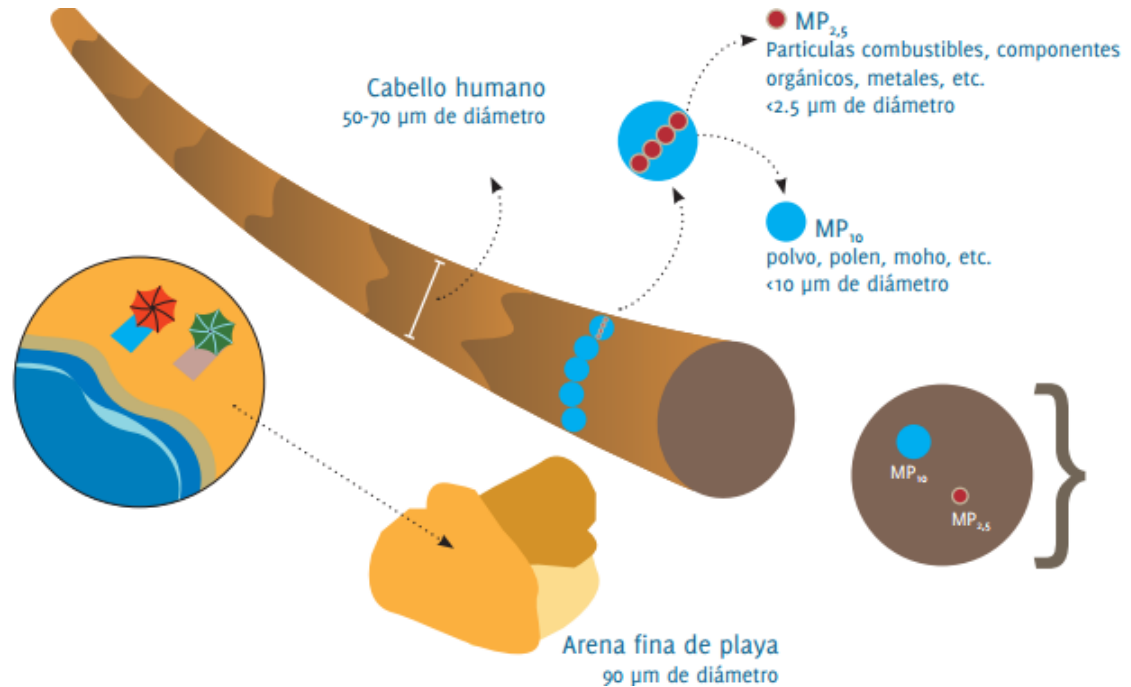


Figura 2.1: Comparación entre MP 2.5 y 10. Fuente: <https://es.wikipedia.org/wiki/PM10>

2.2. Sistema de Información Geográfica

La mayor parte de la información que manejamos en cualquier tipo de disciplina está geo referenciada. Es decir, que se trata de información a la cual puede asignarse una posición geográfica, y es por tanto información que viene acompañada de otra información adicional relativa a su localización. Un Sistema de Información Geográfica (SIG) es fundamentalmente una herramienta para trabajar con información geo referenciada, una definición en la que pueden entrar un gran número de tecnologías y de otros elementos no tecnológicos [7].

Un SIG es un sistema complejo que está compuesto de cinco elementos esenciales, los cuales están fuertemente relacionados entre sí, como se aprecia en la Fig. 2.2.

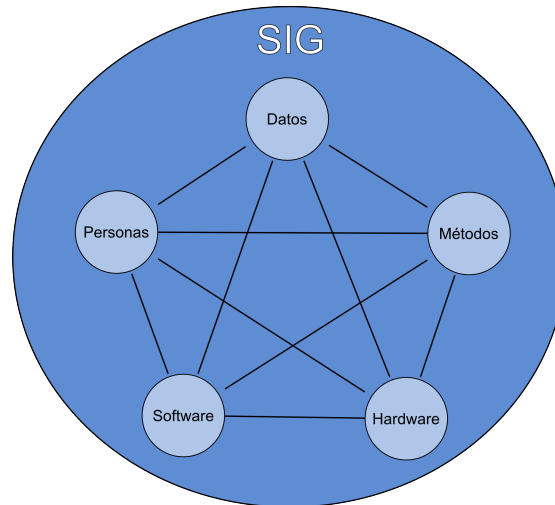


Figura 2.2: Elementos que forman el sistema SIG. Fuente: http://volaya.github.io/libro-sig/chapters/Introduccion_fundamentos.html

- **Datos:** Los datos son la materia prima necesaria para el trabajo en un SIG, y los que contienen la información geográfica vital para la propia existencia de los SIG.
- **Métodos:** Un conjunto de formulaciones y metodologías a aplicar sobre los datos.
- **Software:** Es necesaria una aplicación informática que pueda trabajar con los datos e implemente los métodos anteriores.
- **Hardware:** El equipo necesario para ejecutar el software.
- **Personas:** Las personas son las encargadas de diseñar y utilizar el software, siendo el motor del SIG.

2.2.1. Modelos de Datos Espaciales

Los modelos geográficos nos ofrecen una concepción particular del espacio geográfico y sus atributos. En base a ellos, el siguiente paso es reducir las propiedades de

dichos modelos a un conjunto finito de elementos, de tal modo que el registro de dichos elementos sirva para almacenar la realidad que los modelos geográficos describen. Para ello, empleamos los modelos de representación, también denominados modelos de datos. Para un mismo tipo de información existen diversas alternativas en cuanto a la forma de materializar la realidad y plasmar el modelo geográfico concreto. Estas formas las podemos clasificar en dos grupos principales: modelo de representación raster y modelo de representación vectorial, como se aprecia en la Fig. 2.3 [8].

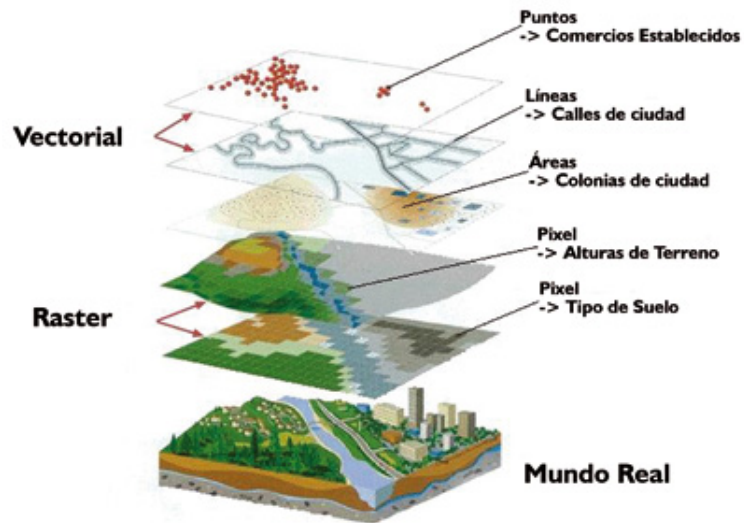


Figura 2.3: Modelos raster y vectorial. Fuente: http://geoservice.igac.gov.co/contenidos_telecentro/fundamentos_sig/cursos/sem.2/uni2/index.php?id=2

Modelo Vectorial

Es un sistema de información geográfica que utiliza vectores definidos por pares de coordenadas relativas a algún sistema cartográfico para la descripción de los objetos geográficos. Es importante aclarar que modelo de datos hace referencia a la conceptualización del espacio mientras que estructura de datos se refiere a la implementación de esa conceptualización. En la Fig. 2.4 se aprecia los diferentes tipos

de primitivas que representan el modelo vectorial. El modelo de datos vectorial usa el sistema de coordenadas X, Y, además de representaciones geométricas de punto, línea y polígono de acuerdo con el número de dimensiones y sus propiedades [6].


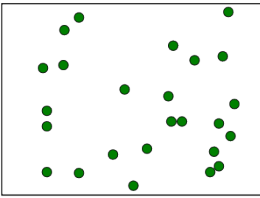

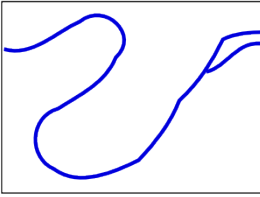

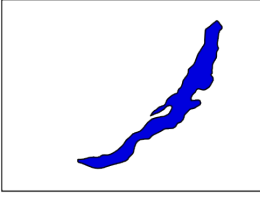
<i>Primitiva</i>	<i>Entidad espacial</i>	<i>Representación</i>	<i>Atributos</i>																		
Puntos			<table border="1"> <thead> <tr> <th>ID</th> <th>Altura</th> <th>Diámetro Normal</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>17.5</td> <td>35</td> </tr> <tr> <td>2</td> <td>22</td> <td>45.6</td> </tr> <tr> <td>3</td> <td>15</td> <td>27.2</td> </tr> <tr> <td>4</td> <td>19.7</td> <td>36.1</td> </tr> <tr> <td>...</td> <td></td> <td></td> </tr> </tbody> </table>	ID	Altura	Diámetro Normal	1	17.5	35	2	22	45.6	3	15	27.2	4	19.7	36.1	...		
ID	Altura	Diámetro Normal																			
1	17.5	35																			
2	22	45.6																			
3	15	27.2																			
4	19.7	36.1																			
...																					
Líneas			<table border="1"> <thead> <tr> <th>Ancho máx(m)</th> <th>Calado máx(m)</th> <th>Longitud(km)</th> </tr> </thead> <tbody> <tr> <td>15</td> <td>4.3</td> <td>35</td> </tr> <tr> <td>6.3</td> <td>3.9</td> <td>5.2</td> </tr> </tbody> </table>	Ancho máx(m)	Calado máx(m)	Longitud(km)	15	4.3	35	6.3	3.9	5.2									
Ancho máx(m)	Calado máx(m)	Longitud(km)																			
15	4.3	35																			
6.3	3.9	5.2																			
Polígonos			<table border="1"> <thead> <tr> <th>Superficie(km)²</th> <th>Profundidad máx(m)</th> </tr> </thead> <tbody> <tr> <td>31494</td> <td>1637</td> </tr> </tbody> </table>	Superficie(km) ²	Profundidad máx(m)	31494	1637														
Superficie(km) ²	Profundidad máx(m)																				
31494	1637																				

Figura 2.4: Representación vectorial y ejemplos particulares de cada una de ellas con atributos asociados. Fuente: http://volaya.github.io/libro-sig/chapters/-Tipos_datos.html

Modelo Raster

En el modelo raster, la zona de estudio se divide de forma sistemática en una serie de unidades mínimas, denominadas habitualmente celdas. Para cada una de éstas se recoge la información pertinente que la describe. La característica principal del modelo raster, y que le confiere gran parte de sus propiedades más interesantes, especialmente de cara al análisis, es su sistematicidad. La división del espacio en unidades mínimas se lleva a cabo de forma sistemática de acuerdo con algún patrón,

de tal modo que existe una relación implícita entre las celdas, ya que éstas son contiguas entre sí, cubren todo el espacio, y no se solapan. Por tanto, la posición de una celda depende de las restantes, para así conformar en conjunto toda la malla regular que cumple las anteriores características. Dicho de otro modo, el orden propio de las celdas, presente gracias a la división sistemática realizada, aporta un elemento adicional que las relaciona entre sí [16]. En la Fig. 2.5 se puede ver la representación de una imagen raster.

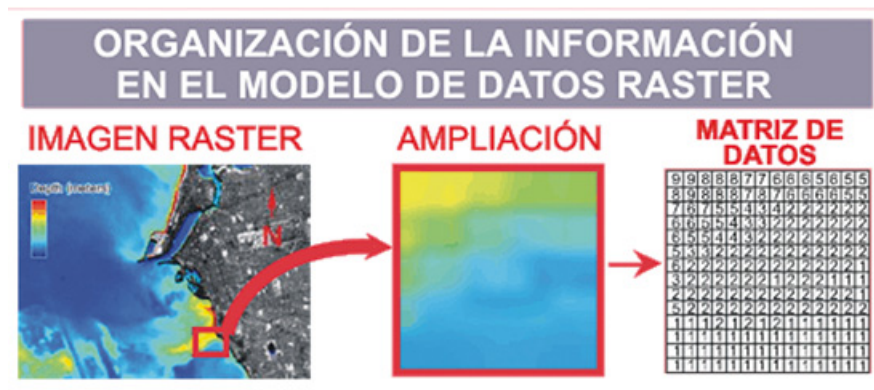


Figura 2.5: Representación raster. Fuente:http://geoservice.igac.gov.co/contenidos_telecentro/fundamentos_sig/cursos/sem_2/uni2/index.php?id=24

2.3. Interpolación Espacial

Interpolación espacial es el proceso de manipular información espacial para extraer información nueva y significativa a partir de los datos originales. Usualmente el análisis espacial se realiza con un SIG, que proporciona herramientas de análisis espacial para calcular estadísticas de las entidades y realizar actividades de geo procesamiento como la interpolación de datos. En hidrología es posible que los usuarios enfatizen la importancia del análisis del terreno y del modelado hidrológico modelar el movimiento del agua por sobre y dentro de la tierra. En la gestión de la vida salvaje, los usuarios están interesados en funciones analíticas que tratan con las ubicaciones de puntos de la vida salvaje y sus relaciones con el medio ambiente. Cada

usuario tendrá diferentes cosas en las que está interesado dependiendo de la clase de trabajo que realicen [3].

2.3.1. Métodos de Interpolación

Mediante interpolación espacial usamos puntos con valores conocidos para estimar valores desconocidos en puntos para los que no tenemos información. En nuestro caso las mediciones de contaminación se hacen en varias estaciones de medición en la ciudad de Curicó, pero no se puede tener en cada lugar una estación, por lo que se tiene que interpolar los datos generados y así lograr estimar esos datos faltantes. Por este motivo existen diferentes tipos de interpolación que nos ayudan a estimar y generar superficies estadísticas.

Inverso a la Distancia (IDW)

En el método de interpolación IDW, los puntos de muestreo se ponderan durante la interpolación de tal manera que la influencia de un punto en relación con otros disminuye con la distancia desde el punto desconocido que se desea crear. La ponderación es asignada a los puntos de muestreo mediante la utilización de un coeficiente de ponderación que controla cómo la influencia de la ponderación decae mientras la distancia hacia el punto nuevo se incrementa. Mientras más grande sea el coeficiente de ponderación menor será el efecto que los puntos tendrán si están lejos del punto desconocido durante el proceso de interpolación. Conforme el coeficiente se incrementa, el valor de los puntos desconocidos se aproxima al valor del punto de observación más cercano [3].

Kriging Ordinario

Kriging es un procedimiento geo estadístico avanzado que genera una superficie estimada a partir de un conjunto de puntos dispersados con valores z . Más aún que con otros métodos de interpolación, se debe realizar una investigación profunda del comportamiento espacial del fenómeno representado por los valores z antes de seleccionar el mejor método de estimación para generar la superficie de salida. Kriging presupone que la distancia o la dirección entre los puntos de muestra reflejan una correlación espacial que puede utilizarse para explicar la variación en la superficie [5].

2.4. Arquitectura Cliente-Servidor

La arquitectura cliente-servidor se caracteriza por dividir los sistemas en dos partes. Por un lado tenemos el cliente que se ejecuta en los terminales de los usuarios. Los servidores son los encargados de proveer los servicios necesarios para que los clientes trabajen y entreguen la información requerida por éstos. Estas entidades se encuentran separadas de forma lógica y se comunican a través de una red de comunicaciones, para trabajar de forma conjunta. El esquema básico es que un cliente realiza una petición de un servicio, el servidor recibe y procesa dicha petición y retorna la respuesta solicitada.

2.4.1. Base de Datos Geográfica

Una Base de Datos Geográfica (BDG) es un conjunto de datos geográficos organizados de tal manera que permiten la realización de análisis y la gestión del territorio dentro de aplicaciones de Sistemas de Información Geográfica. Además, una BDG se utiliza de soporte para la implantación de servicios geográficos relacionados con las Infraestructuras de Datos Espaciales (IDE), y su contenido es la base fundamental en los procesos de producción cartográficos.

La espina dorsal de una BDG es el modelo de datos, que consiste en la formalización conceptual (descripción) de las entidades geográficas del mundo real con el objeto de realizar una abstracción que permita satisfacer unas necesidades de información. La implementación del modelo debe de facilitar la explotación y optimizar el almacenamiento para conseguir el mejor rendimiento en las consultas [4].

2.4.2. PostgreSQL PostGIS

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto de mejor desempeño y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afecta al resto y el sistema puede continuar funcionando [14].

PostGIS es una extensión que convierte el sistema de base de datos PostgreSQL

en una base de datos espacial. La combinación de ambos es una solución perfecta para el almacenamiento, gestión y mantenimiento de datos espaciales. Debido a que está construido sobre PostgreSQL, PostGIS hereda automáticamente sus características, así como los estándares abiertos [13].

2.4.3. MapServer

MapServer es un motor de renderizado de datos geográficos de código abierto escrito en C. Más allá de exploración de datos SIG, MapServer permite crear mapas de imágenes geográficas, es decir, los mapas que pueden dirigir contenido al usuario [1].

Algunas de sus principales características son:

- Soporte para la visualización y consulta de cientos de formatos raster, vectorial y base de datos.
- Capacidad de ejecutarse en varios sistemas operativos como Windows, Linux, Mac OS X, entre otros.
- Soporte de los populares lenguajes de scripts y ambientes de desarrollo, como PHP, Python, Perl, Ruby, Java y .NET.
- Representación de alta calidad.
- Salida de aplicaciones totalmente personalizables.

2.4.4. R

R es una implementación de software libre del lenguaje S pero con soporte de alcance estático. Se trata de uno de los lenguajes más utilizados en investigación por la comunidad estadística, siendo además muy popular en el campo de la minería de datos, la investigación biomédica, la bioinformática y las matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes bibliotecas o paquetes con funcionalidades de cálculo o para graficar. Además, R puede integrarse con distintas bases de datos y existen bibliotecas que facilitan su utilización desde lenguajes de programación interpretados, como Perl y Python [2].

2.4.5. Openlayers

Openlayers es una librería javascript que se ha posicionado como la mejor alternativa a google maps a la hora de incluir un componente de tipo mapa en cualquier página web, con georreferencias. Lo más interesante es que incluso si disponemos de la cartógrafa de Google, Microsoft (MSN Virtual Earth) o Yahoo, también es posible usar OpenLayers como visor porque es totalmente compatible con tecnologías propietarias, además de cumplir con los estándares WMS y WFS. Al ser una librería del lado del cliente, es un visor de mapas en javascript, la descarga de éstos se realiza directamente desde el navegador a través de Ajax, no genera tráfico en el servidor, los mapas se descargan directamente del servidor de mapas que suele ser una pieza diferenciada a nivel de sistemas. OpenLayers permite sobreponer distintas capas sobre una básica, añadir indicadores o puntos en el mapa con leyendas, así como polígonos y proporciona su propio API para dibujarlos de una manera sencilla [9]. En la Fig. 2.6 se aprecia un mapa generado por Openlayers.



Figura 2.6: Openlayers

2.5. Sistemas Existentes

2.5.1. Sistema de Información Nacional de Calidad del Aire (SINCA)

El Sistema de información Nacional de Calidad del Aire (SINCA) del Ministerio de Medio Ambiente pone a su disposición información de calidad de aire de todo el país, buscando mejorar gradualmente el conocimiento, la vigilancia y la gestión de la calidad del aire que respiramos [11].

Este es un sitio web que contiene la información de la calidad del aire en las distintas regiones y ciudades del país, Como se puede ver en la Fig. 2.7 su visualización es simple. Además, este sistema tiene una sección de estadísticas, que entrega información más detallada de los distintos tipos de contaminantes como también la posibilidad de graficar estos datos.

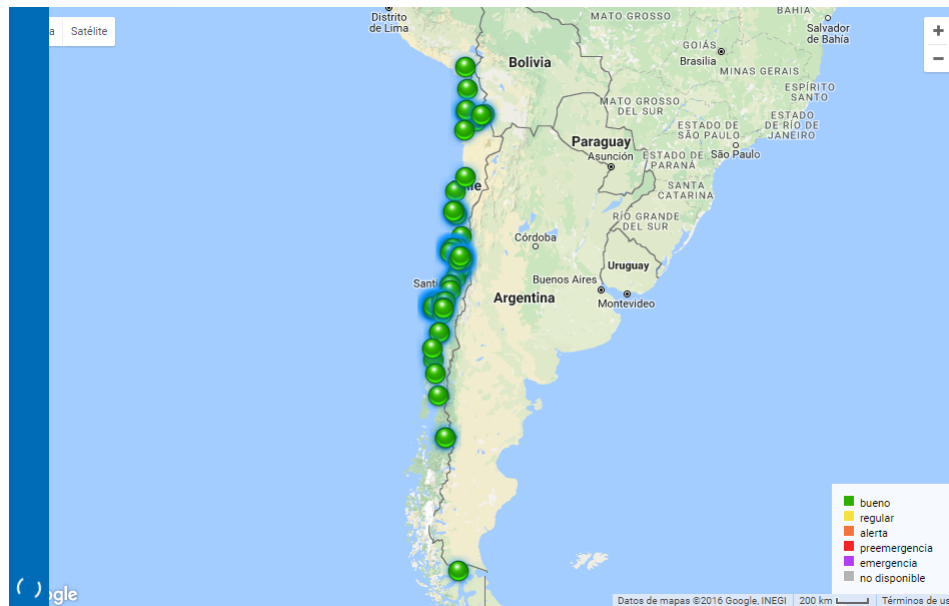


Figura 2.7: SINCA. Fuente: <http://sinca.mma.gob.cl/index.php/>

3. Metodología

En este capítulo se explica la forma en que se afrontó el problema a resolver. Para la realización de este proyecto fue primordial la elección de una metodología que facilitara tanto el diseño como el desarrollo del sistema. Es por esto que basándose en el objetivo principal que era la realización de un prototipo funcional, se consideró importante tener una metodología de desarrollo bien definida desde un comienzo para poder estructurar, planificar y controlar el desarrollo del software. Además, al tratarse de un prototipo para el cual los requisitos y soluciones iban a ir evolucionando a lo largo del tiempo, se buscó utilizar una que permitiera un alto grado de flexibilidad.

Para la elección de la metodología se tomaron algunas variables en cuenta, una de ellas es la del tiempo para la realización del prototipo, ya que se contaba con un plazo determinado el cual era escaso y no se podía modificar. Otra variable es la flexibilidad y posibles cambios en los requisitos durante el desarrollo del proyecto. Teniendo en cuenta estas principales consideraciones, para la realización de este proyecto se decidió utilizar como base la metodología ágil Scrum.

3.1. Scrum

Scrum es una metodología ágil de desarrollo, aunque surgió como modelo para el desarrollo de productos tecnológicos. También se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software. Es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto [10].

3.1.1. Características de Scrum

El marco de trabajo Scrum se basa en los Equipos Scrum, roles y eventos. Cada uno de estos componentes dentro del marco de trabajo sirve a un propósito específico y son esenciales para el éxito del proyecto.

3.1.2. Equipos Scrum

El equipo Scrum está compuesto por el dueño del producto (Product Owner), el equipo de desarrollo y un Scrum Master. Cada equipo es auto organizado y multi-funcional. En Scrum cada equipo entrega productos de forma iterativa e incremental. A continuación se presentan los integrantes de un equipo de Scrum:

- **Dueño del Producto (Product Owner):** Es el representante de todas las personas interesadas en los resultados del proyecto. Es el responsable de gestionar la Lista del Producto (Product Backlog), que consiste en una lista ordenada de todo lo que será necesario en el producto y es la única fuente de requisitos.
- **Equipo de Desarrollo (Development Team):** Son los encargados de producir y entregar un incremento del producto terminado al final de cada Sprint.
- **Scrum Master:** Es el responsable de que Scrum sea entendido y adoptado por todos los miembros del equipo. Es un líder que está al servicio del equipo y es el que hace que las reglas se cumplan.

3.1.3. Eventos

Scrum utiliza cuatro eventos formales, contenidos dentro del Sprint. Un sprint es el corazón de Scrum, consiste en un periodo de tiempo en el cual se crea un incremento del producto, el cual debe estar terminado y utilizable. Por cada Sprint se deben realizar los siguientes eventos:

- **Reunión de Planificación del Sprint (Sprint Planning Meeting):** Se planifica el trabajo y cuál será el objetivo a realizar durante el Sprint.
- **Scrum Diario (Daily Scrum):** Es una reunión breve en donde los integrantes del equipo sincronizan sus actividades, y se crea un plan para las siguientes 24 horas.

- **Retrospectiva del Sprint (Sprint Retrospective):** Es una instancia en la cual el equipo de Scrum se inspecciona a sí mismo para crear un plan de mejoras que serán abordadas durante el siguiente Sprint.
- **Revisión del Sprint (Sprint Review):** Se realiza al final de cada Sprint para inspeccionar el incremento realizado.

3.2. Aplicación de Metodología

Para este proyecto se utilizó la metodología ágil Scrum. Se eligió un plan de trabajo con Sprints de una semana de duración, en la cual se realizaban los eventos principales de Scrum. Para el Sprint Planning se determinaba cuáles serían las historias de usuario a abordar durante la iteración en la cual se iba a trabajar. Los Daily Scrum, por motivos de tiempo y disponibilidad, no se realizaban diariamente pero existían momentos dentro del Sprint para informar las tareas que se realizaban, así como los planes. Por último, en cuanto al Sprint Review, éste se realizaba semanalmente al terminar la iteración, en donde se presentaba y analizaba el incremento realizado. Todo esto era gestionado por el equipo de Scrum, cuya composición era la siguiente:

- **Product Owner:** Este rol fue protagonizado por el profesor guía. Se realizaban reuniones entre profesor y desarrollador para definir cómo abordar las historias de usuario.
- **ScrumMaster:** Este rol fue protagonizado por el profesor guía, el cual se encargaba de ir verificando el trabajo realizado, de entregar soporte en caso de problemas. El trabajo principal de un ScrumMaster es eliminar los obstáculos que impiden que el desarrollador alcance el objetivo.
- **Equipo de Desarrollo:** Este rol fue realizado por el estudiante. Es él quien tiene la responsabilidad de entregar el trabajo terminado.

3.2.1. Etapas de Desarrollo

Para cumplir con la metodología se dividió el proyecto en varias etapas. Las etapas de desarrollo fueron repartidas de una manera tal que permitiera realizar

una etapa de diseño lo más sólida posible sin perder agilidad, de tal forma que los Sprints de Scrum estuvieran enfocados más en la implementación que en el diseño. Las etapas de desarrollo del proyecto eran las siguientes:

Etapa Base

- **Análisis:** En esta etapa se define el problema a resolver y se identifican todos los aspectos generales del proyecto junto con las entidades que interactúan con el sistema. Además, se describe el problema y su contexto, junto con el objetivo general y los específicos.
- **Requerimientos:** Para esta etapa del desarrollo, se definen los requisitos principales que debe incluir el sistema, los que pueden ser funcionales o no funcionales.
- **Diseño arquitectura:** En esta etapa se desarrolla el diseño de la primera versión de la arquitectura del sistema, identificando las diferentes interacciones que existirán entre los componentes.

Etapa de Diseño

- **Modelado de la base de datos:** En esta etapa se desarrolla el modelo inicial de la base de datos geoespacial mediante un diagrama entidad relación. Además, se carga la base de datos con valores de prueba.
- **Desarrollo del prototipo:** Para esta etapa se implementa el primer prototipo que servirá de base para los Sprints posteriores, el cual crecerá de manera iterativa. En este punto se necesita una implementación básica de la base de datos con valores de prueba junto con la puesta en marcha del servidor de mapas.

Etapa de Desarrollo

- **Implementación :** En esta etapa se van desarrollando iterativamente las funcionalidades necesarias del sistema, las cuales deben ser agregadas al prototipo inicial. Además debido a la utilización de una metodología ágil, en esta etapa se va actualizando de ser necesario, el modelo de la base de datos, la arquitectura

del sistema o alguna tecnología web, para adaptarse a los cambios que implica generar una funcionalidad nueva.

Etapa de Pruebas

- **Pruebas del sistema:** Después de terminar la etapa de desarrollo y completado la totalidad de las funcionalidades y objetivos del sistema, se debe realizar las pruebas de funcionamiento del sistema completo. Cabe señalar que aplicando Scrum además se deben realizar pruebas en cada Sprint de las funcionalidades implementadas.

4. Diseño del Sistema

En este capítulo se describe todos los aspectos relevantes en el diseño del proyecto para el sistema desarrollado. En primer lugar se presentan las herramientas y tecnologías utilizadas. Luego, se presentan los requerimientos funcionales del sistema en forma de historias de usuario. Posteriormente se describe el diseño de la arquitectura, la base de datos y la interfaz del prototipo.

4.1. Tecnologías Involucradas

En esta sección se describen cada una de las tecnologías necesarias para el desarrollo del prototipo. Estas se encuentran divididas en herramientas y lenguajes de programación.

4.1.1. Herramientas

- **Programa R¹**: R es un entorno de software libre para la informática estadística y los gráficos. Compila y ejecuta en una amplia variedad de plataformas UNIX, Windows y MacOS. Utilizado en el proyecto para realizar las interpolaciones de los datos y para la generación de imágenes raster.
- **PuTTY²**: es un cliente SSH y telnet, desarrollado originalmente por Simon Tatham para la plataforma Windows. PuTTY es un software de código abierto disponible con código fuente y desarrollado y soportado por un grupo de voluntarios.

¹<https://www.r-project.org/>

²<http://www.putty.org/>

- **Sublime Text**³: Es un editor de texto, especialmente para crear código, utilizado para programar todo el sitio web.
- **QGIS**⁴: Es un Sistema de Información Geográfica de código libre, que permite manejar formatos raster (TIFF, GeoTIFF, JPG, entre otros) y vectoriales (Shapefiles, GRASS GIS, Mapinfo, entre otros.) mediante bibliotecas de GDAL y además trabajar con bases de datos mediante su soporte para la extensión espacial PostGIS.
- **PgAdminIII**⁵: Es una herramienta de administración y desarrollo para la base de datos PostgreSQL y que además provee de una interfaz visual para el manejo de ésta.
- **Apache HTTP Server**⁶: Es un servidor web HTTP de código abierto que funciona tanto en sistemas UNIX como Windows. Utilizado en el proyecto para alojar el sitio web.
- **MapServer**⁷: Es un entorno de desarrollo en código abierto para la creación de aplicaciones SIG en Internet/Intranet con el fin de visualizar, consultar y analizar información geográfica a través de la red.

4.1.2. Lenguajes de Programación

- **PHP**⁸: Es un lenguaje interpretado de alto nivel embebido en páginas HTML y que es ejecutado en el lado del servidor. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML. Esta herramienta es utilizada en este proyecto para comunicar el cliente con la base de datos, ya que provee un conector para trabajar con PostgreSQL.
- **JavaScript**⁹: Es un lenguaje de programación interpretado, orientado a objetos, débilmente tipado, que es usado del lado del cliente en una aplicación

³<http://www.sublimetext.com/>

⁴<http://www.qgis.org/>

⁵<http://www.pgadmin.org/>

⁶<http://httpd.apache.org/>

⁷<http://mapserver.org/>

⁸<https://www.php.net/>

⁹<https://www.javascript.com/>

web, el cual permite el manejo del documento HTML. Esencial para la construcción de sitios web, utilizado para el control de eventos y hacer el sitio web más dinámico con ayuda de JQuery, especialmente para el uso de notificaciones del sistema. El uso más común de JavaScript es implementar funciones que son incluidas en páginas HTML, como por ejemplo, cargar elementos nuevos o enviar datos al servidor sin la necesidad de recargar la página, o también para la validación de los valores de las entradas de un formulario web. Para este proyecto Javascript es utilizado para implementar funciones que trabajan en conjunto con Openlayers y en la validación de formularios.

- **Ajax**¹⁰: Proviene del acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML). AJAX es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones son ejecutadas por el lado del cliente mientras mantienen comunicación de manera asíncrona con el servidor. De esta forma es posible realizar cambios en una página sin necesidad de recargarla. En este proyecto se utilizó AJAX junto con PHP para desplegar datos provenientes de la base de datos en forma de tablas de manera dinámica, además también se utilizó para obtener los datos para graficar.
- **Bootstrap**¹¹: Es un framework o conjunto de herramientas para realizar el diseño de sitios y aplicaciones web. Bootstrap contiene plantillas de diseño con tipografía, botones, menús de navegación, formularios y otros elementos basados en HTML y CSS. Además posee extensiones de JavaScript para manejar los elementos anteriormente señalados. Para este proyecto se utilizó la versión 3.3.5 y un Template llamado Material Dashboard, desarrollado con Bootstrap para el desarrollo de la interfaz.
- **CSS**¹²: Proviene del acrónimo de Cascading Style Sheets. Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación (fuentes, colores, espaciado) de los documentos definidos con HTML. En este proyecto se utilizó para manejar la presentación de la interfaz de la aplicación web.
- **jQuery**¹³: Es una librería escrita en JavaScript que permite tener el control del

¹⁰<http://www.w3schools.com/ajax/>

¹¹<http://getbootstrap.com/>

¹²<http://www.w3schools.com/css/>

¹³<http://jquery.com/>

documento HTML, animaciones, manejo de eventos y uso de Ajax3. Por ende, en la construcción del sitio web es muy utilizado para el control de eventos realizados por el usuario y hacer peticiones al servidor de manera asincrónica.

- **Openlayers**¹⁴: Es una librería JavaScript de código abierto que permite operar y visualizar mapas en un navegador web. Esta herramienta provee un API para trabajar con diferentes fuentes de información geográfica en la web, entre los cuales se encuentran: Web Map Services (WMS), Web Features Services (WFS), datos vectoriales (GeoJSON, KML, GML, entre otros) y mapas comerciales (Google Maps, Bing, OpenStreetMap, entre otros). Se utilizó la versión 4.1.1 y esta herramienta es utilizada en este proyecto para realizar todo el manejo de la visualización de mapas en el cliente web, además del manejo de capas de mapas.
- **Highcharts**¹⁵: Es una biblioteca de gráficos escrita en JavaScript puro, que ofrece una manera fácil de agregar gráficos interactivos a un sitio web o aplicación web. Utilizada para visualizar los datos en forma de gráficos de una manera más simple.

4.2. Historias de Usuario

Para la confección de requisitos del proyecto se utilizó las historias de usuario. El motivo es simple: sólo hay que expresar a grandes rasgos en qué consiste el requisito, no es necesario un nivel de detalle muy elevado, con lo cual no es mucho el gasto de tiempo para generar estos requisitos. Esto va acorde a la metodología ágil seleccionada que no ve la captura de requisitos como algo rígido y tedioso.

Una historia de usuario es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos, acompañadas de las discusiones con los usuarios y las pruebas de validación. Cada historia de usuario debe ser limitada, por lo que debería poderse escribir sobre una nota adhesiva pequeña.

¹⁴<https://openlayers.org/>

¹⁵<https://www.highcharts.com/>

Para la captura se describen los requisitos del sistema a desarrollar en forma de historias de usuario. Estos requisitos fueron obtenidos a partir reuniones con el profesor guía que es el Product Owner del proyecto. A continuación se presenta un listado de requisitos que se plantearon previamente para el proyecto:

1. Los usuarios quieren poder visualizar concentraciones de contaminación del aire de la ciudad de Curicó mediante una imagen en el mapa.
2. Los usuarios quieren poder visualizar estaciones de medición de la contaminación en la ciudad.
3. Los usuarios quieren poder visualizar de forma continua las imágenes generadas.
4. Los usuarios quieren poder visualizar un resumen del día, mostrando porcentajes de contaminación con respecto a la ciudad.
5. Los usuarios quieren poder seleccionar una estación de medición y ver datos con respecto a ésta.
6. Los usuarios quieren poder visualizar la información de cada estación en forma de gráfico, éstas pueden ser en años, meses, semanas o días.
7. Los usuarios quieren poder controlar la visibilidad de las imágenes generadas.
8. Los usuarios quieren poder descargar la información de la contaminación de la ciudad en algún formato estándar.

4.3. Diseño de la Arquitectura y Procesos

En este apartado se presenta el diseño de la arquitectura seleccionada para la implementación de este proyecto, además de una explicación de los elementos que la componen y cómo éstos trabajan en conjunto para poder lograr un correcto funcionamiento del sistema. Además de los diferentes procesos de los datos y también para la visualización de éstos que son necesarios para el prototipo.

4.3.1. Arquitectura

Para la arquitectura del proyecto, la seleccionada fue Cliente-Servidor debido a que la mayoría de las operaciones se centran en el proceso de generación y visualización de mapas. En este proceso se trabaja bastante con peticiones a servicios web ya sean WMS o WFS. Por otro lado, la arquitectura Cliente-Servidor está enfocada a trabajar con aplicaciones distribuidas como es el caso de este proyecto, en donde el cliente es una implementación web y el servidor es utilizado para responder las peticiones de los clientes mediante servicios web. En la siguiente Fig. 4.1 se puede apreciar la interacción de las diferentes capas de la arquitectura.

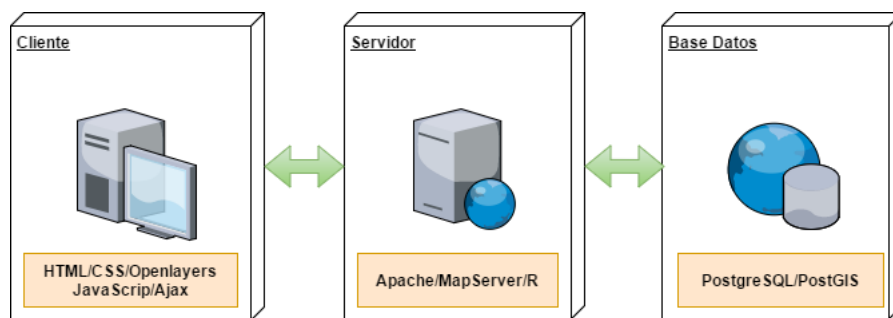


Figura 4.1: Modelo de la arquitectura usada.

- Cliente:** En esta capa se localiza todo lo relacionado a la aplicación web. Para el diseño de la interfaz de la página se utiliza HTML y CSS. En cuanto al manejo y visualización de los mapas se utiliza Openlayers y para el comportamiento dinámico de la aplicación web se utiliza JavaScript y Ajax. Esta capa es la encargada de interactuar con el usuario directamente.
- Servidor:** En esta capa se encuentra el servidor de mapas compuesto por MapServer que es el encargado de recibir y procesar las peticiones de los usuarios, además de obtener los datos desde la base de datos y por último de estilizar los datos para su presentación. También se encuentra un servidor web de apache con el módulo de PHP que es utilizado para alojar la aplicación web y PHP es utilizado para realizar consultas directas a la base de datos. Por último el

programa R que se encarga de la interpolación de los datos y de generar las imágenes raster de las mediciones de contaminación .

- **Base de Datos:** En esta capa se encuentra la base de datos en PostgreSQL junto con la extensión para manejar datos espaciales PostGIS. Además, tenemos todos los datos necesarios para la aplicación y también la imagen raster generada.

4.3.2. Proceso de Generación de Imagen

Como el proyecto se realiza con una base de datos que contiene los datos en bruto de las mediciones de contaminación, se debe realizar el proceso de generación de la imagen Raster la cual contendrá la contaminación de una zona y fecha antes seleccionada. Esta se va renovando y generando cada una hora ya que se estarán ingresando datos nuevos cada hora en el sistema. Para el proceso de la generación de imagen, se creó una secuencia para la imagen del tipo raster. Esta secuencia tiene tres etapas las cuales se realizan en la capa del servidor web. Se utilizan las diferentes herramientas para lograr con el objetivo: el programa R, extensión de PostGIS (raster2pgsql) y también la extensión de PostgreSQL (psql). En la siguiente Fig. 4.2 se puede apreciar la interacción de las diferentes etapas para la generación de la imagen.

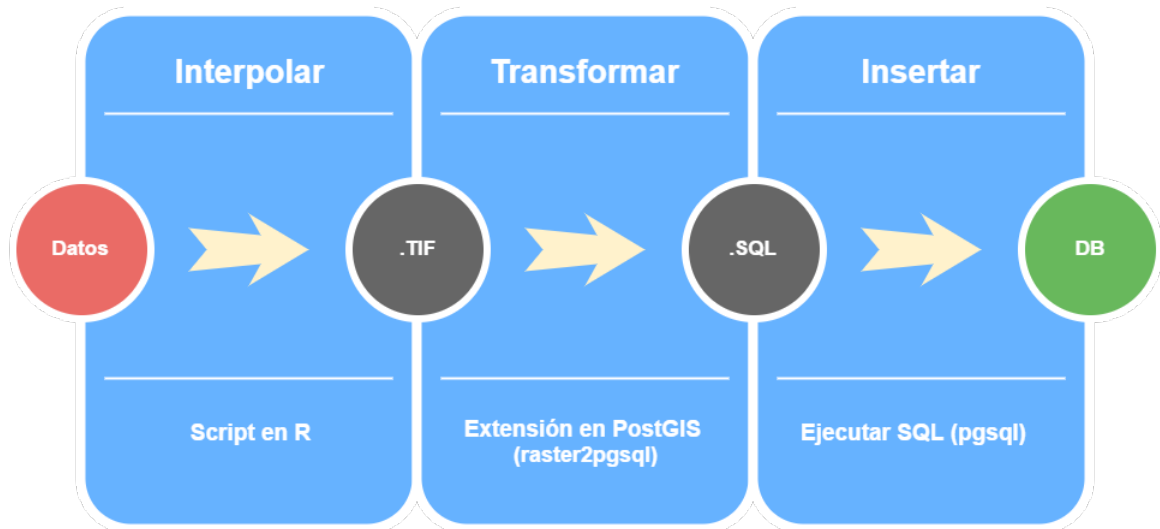


Figura 4.2: Proceso para la generación de imagen.

- Interpolar:** Esta es la primera etapa en la cual se realiza la interpolación de los datos. Todo se ejecuta mediante un script en el programa R, que se encarga de leer los datos directamente desde PostgreSQL dependiendo del tipo de contaminación y una fecha en particular. Luego, con los datos se realiza la interpolación y se genera una imagen de tipo raster como extensión .TIF.
- Trasformar:** En esta etapa se realiza la transformación de la imagen raster .TIF a una consulta en .SQL. Esto es necesario porque para guardar esta imagen se tiene que modificar en una consulta SQL. Para poder insertar una imagen raster en PostGIS utilizamos la extensión raster2pgsql la cual toma la imagen generada por el programa R y genera un archivo SQL listo para su ejecución e inserción de la imagen en la base de datos.
- Insertar:** En esta última etapa se realiza la inserción del archivo .SQL en la base de datos, se toma la consulta en .SQL y mediante la extensión pgsql de PostgreSQL se inserta en la base de datos de manera automática. Con esto se termina la secuencia para la generación de una imagen desde las mediciones de contaminación de la base de datos.

4.3.3. Proceso de Visualización

Para poder visualizar la imagen raster de los datos de contaminación, se pasa por varias etapas. Cada una de ellas se produce entre la interacción entre el cliente y el servidor. En la siguiente Fig. 4.3 se puede apreciar el proceso estándar para la generación y visualización de datos en forma de mapas.

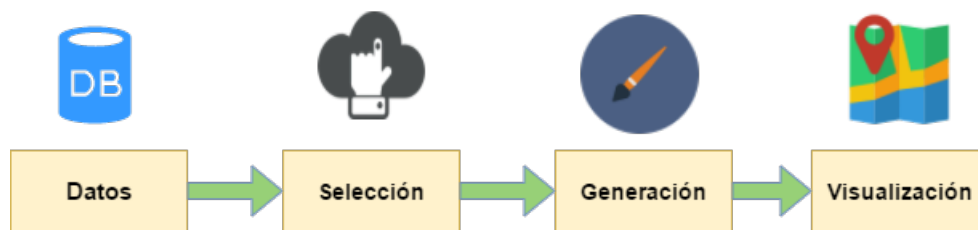


Figura 4.3: Proceso de generación y visualización de mapas e imágenes.

- **Datos:** Esta es la primera etapa en la cual tenemos los datos de la contaminación en formato raster en la base de datos.
- **Selección:** En esta etapa se realiza el proceso de selección que depende de la imagen de la contaminación (PM 2.5 y PM 10) que desea visualizar el usuario, como también del tipo de interpolación con la que generó la imagen (IDW y Kriging).
- **Generación:** En esta etapa se realiza un proceso de generación de la capa, que consiste en darle un formato y el estilo con el cual se presentará al usuario.
- **Visualización:** En esta última etapa se realiza la visualización que corresponde a la presentación de estos datos con sus estilos correspondientes en sistema.

4.4. Diseño de la Base de Datos

En esta sección se presenta el diseño de la base de datos para el sistema que es necesario para la implementación de este proyecto. Se mostrará mediante un esquema

entidad-relación y un modelo relacional, junto con una explicación de cada una de las entidades que la componen y cómo éstas se relacionan entre sí.

4.4.1. Modelo Entidad-Relación y Relacional

En la Fig. 4.4 podremos observar el modelo entidad-relación y en la Fig. 4.5 el modelo relacional de la base de datos, basados en los requisitos del sistema anteriormente obtenido.

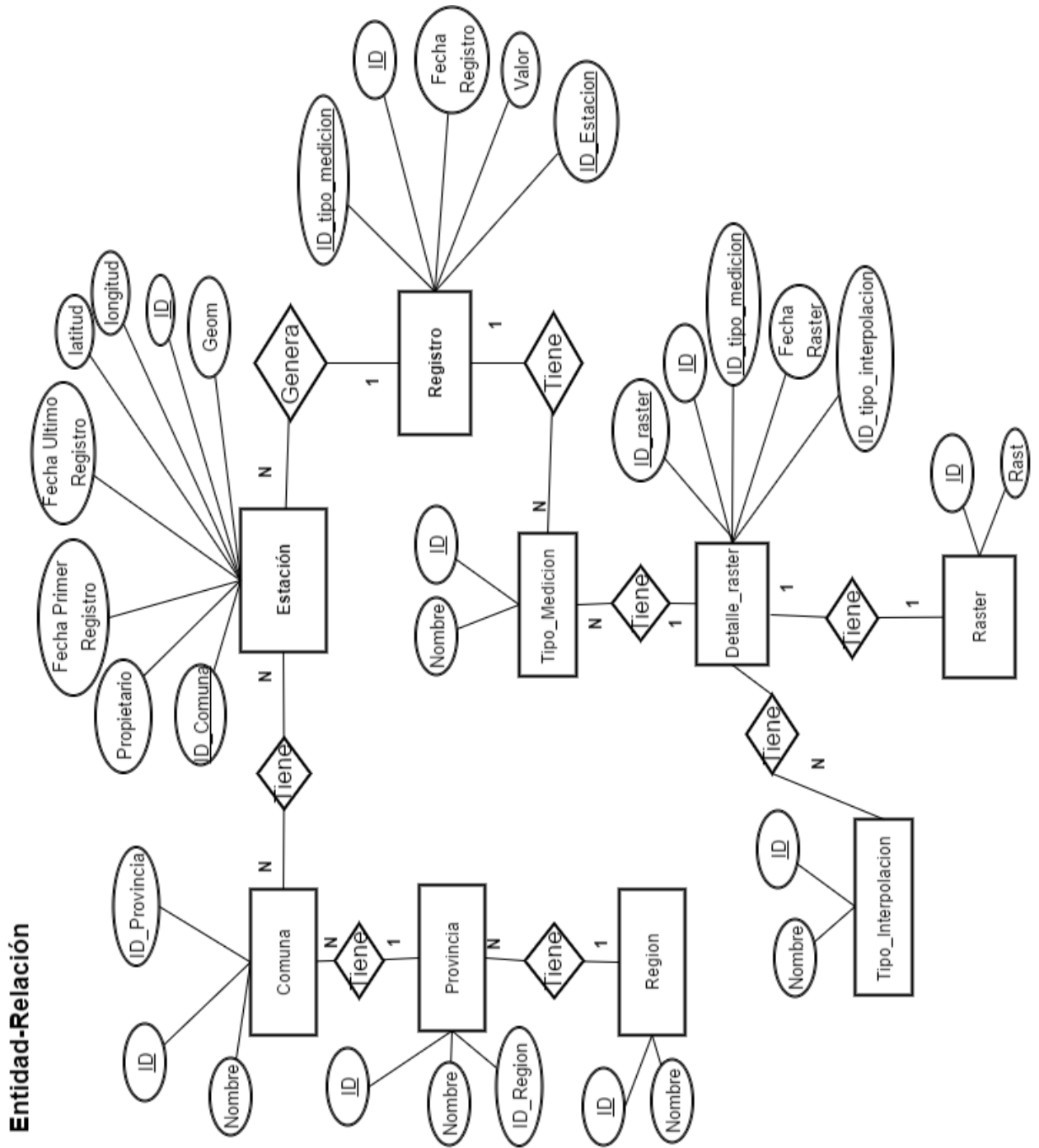


Figura 4.4: Modelo entidad-relación

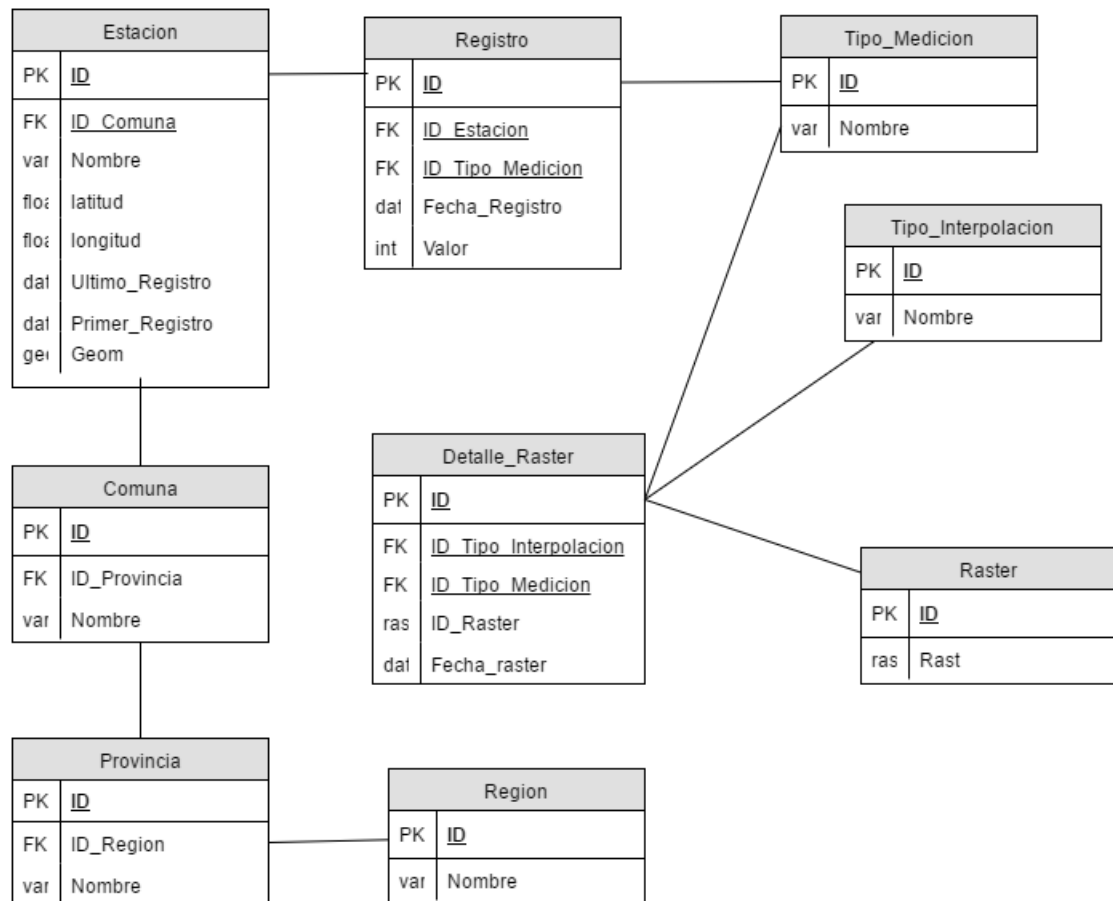
Modelo Relacional

Figura 4.5: Modelo Relacional base de datos

4.4.2. Entidades

En esta sección se definirá las entidades de la base de datos con una breve descripción de cada uno de sus atributos.

- **Registro**: Entidad que almacena los registros proporcionados por las estaciones de medición. Esta entidad contiene las mediciones las cuales pueden ser de

diferentes tipos. Esta entidad es generada por cada estación de medición.

- ID: Identificador del registro.
 - ID_Estacion: Clave foránea de la estación a cual corresponde el registro.
 - ID_tipo_medicion: Clave foránea del tipo de medición.
 - fecha_registro: Fecha del registro.
 - valor: Valor de la contaminación para este registro.
- **Estación:** Entidad que almacena los datos de cada estación de la ciudad de Curicó, además de sus geometrías. Esta entidad contiene la posición exacta de la estación de medición como también las fechas del último y primer registro creado.
 - ID: Identificador de la estación.
 - Propietario: Corresponde al nombre del encargado de la estación.
 - fecha_primer_registro: Fecha del primer registro.
 - fecha_ultimo_registro: Fecha del último registro.
 - latitud: Coordenada geográfica correspondiente a la latitud de la estación.
 - longitud: Coordenada geográfica correspondiente a la longitud de la estación.
 - geom: Del tipo geometría, representa el punto donde se encuentra la estación en el mapa.
- **Tipo Medición:** Entidad que almacena los tipos de mediciones posibles para cada registro y detalle del raster.
 - ID: Identificador del tipo de medición.
 - nombre: Nombre del tipo de medición.
- **Tipo interpolación:** Entidad que almacena los tipos de interpolaciones posibles para cada raster generado en el sistema.
 - ID: Identificador del tipo de Interpolación.
 - nombre: Nombre del tipo de Interpolación.

- **Raster:** Entidad que almacena los raster generados en el sistema. Esta entidad sólo contiene los datos del tipo raster, los cuales contiene la información de la contaminación.
 - ID: Identificador del raster.
 - rast: Del tipo Raster, representa la contaminación en el mapa.
- **Detalle raster:** Entidad que almacena los detalles de los raster generados. Esta entidad contiene todos los datos de información del raster.
 - ID: Identificador del detalle raster.
 - ID_raster: Clave foránea del raster de cual pertenece el detalle raster.
 - ID_tipo_medicion: Clave foránea del tipo de medición del raster.
 - ID_tipo_interpolacion: Clave foránea del tipo de interpolación del raster.
 - fecha_raster: Fecha en la cual fue creado el raster.
- **Comuna:** Entidad que almacena el nombre de la comuna.
 - ID: Identificador del detalle de la comuna.
 - nombre: Nombre de la comuna.
 - ID_provincia: Clave foránea de la provincia a cual pertenece la comuna.
- **Provincia:** Entidad que almacena el nombre de la provincia.
 - ID: Identificador del detalle de la Provincia.
 - nombre: Nombre de la Provincia.
 - ID_region: Clave foránea de la región a cual pertenece la provincia.
- **Región:** Entidad que almacena el nombre de la Región.
 - ID: Identificador del detalle de la región.
 - nombre: Nombre de la región.

4.5. Diseño de la Interfaz de Usuario

En esta sección se enseña un diseño básico de la interfaz para el sistema. A pesar de tratarse de un prototipo y centrarse principalmente en el proceso de visualización de los datos de la contaminación, la idea fue realizar una interfaz sencilla y simple de utilizar y que también permitiera la posibilidad de ir agregando funcionalidades a futuro.

4.5.1. Mocukp Inicial

El Mockup inicial del prototipo proporciona al menos una parte principal de las funcionalidades del sistema y permite realizar diferentes pruebas del diseño. En la Fig. 4.6 se puede apreciar el Mockup realizado inicialmente, que con el tiempo fue cambiando en el desarrollo del sistema.

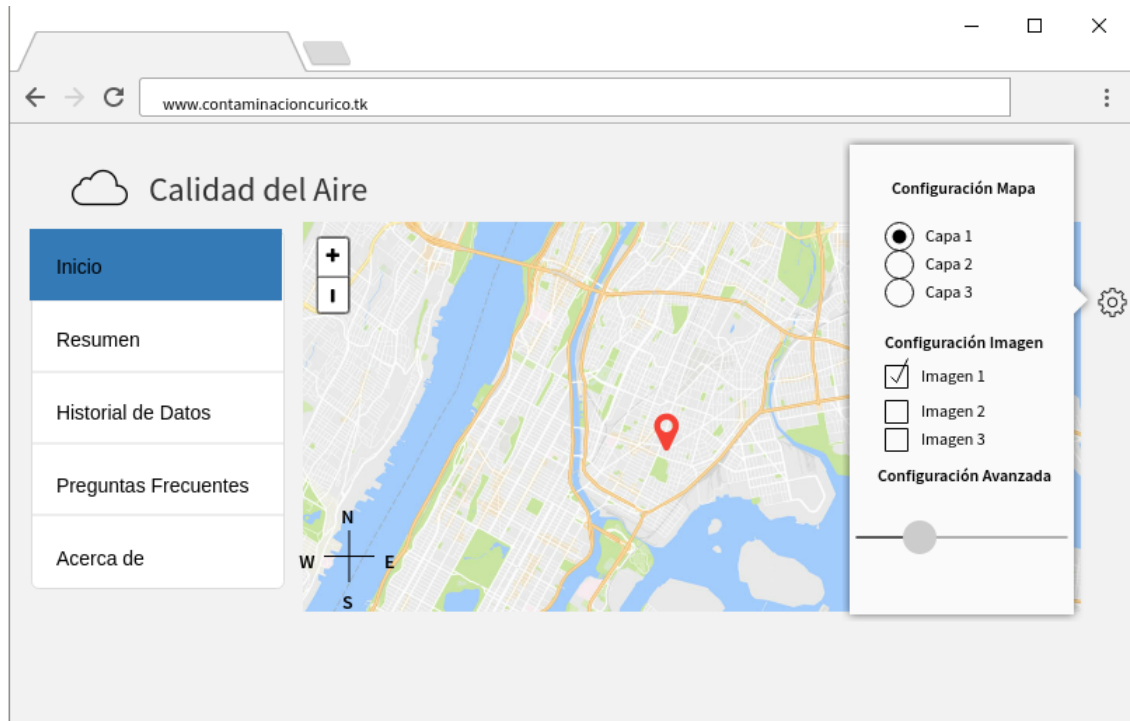


Figura 4.6: Mockup inicial del sistema.

La interfaz del Mocukp está dividida en tres partes principales que se especifican a continuación:

- **Sector de Navegación:** En esta parte de la interfaz es donde se encuentra la barra con las distintos secciones que contiene la página para navegar. Se encuentra al lado izquierdo y es de un considerable tamaño para que sea claro en donde se encuentra actualmente el usuario.
- **Sector Mapa:** En esta parte de la interfaz es en donde se muestra el mapa principal del sistema, en el cual se despliegan las imágenes generadas a partir de las consultas de los usuarios y se encuentra en la parte central de la pantalla.
- **Sector de Control:** En esta parte de la interfaz es en donde se encuentran las herramientas de selección de capas de mapas, junto con los controles básicos

para manejar dichas capas. Se encuentra en la parte derecha de la pantalla.

5. Desarrollo del Sistema

En este capítulo se describe la solución del problema, se realiza un acabado análisis a todos los aspectos presentes en el proceso de desarrollo del prototipo del sistema web. En primer lugar, se describe el proceso de instalación y configuración de las diferentes herramientas y tecnologías utilizadas. Luego se describe el proceso de la implementación de la base de datos y de cómo procesar los datos para insertarlos en ésta. A continuación se especifica el proceso de generación de las imágenes mediante el programa R. Luego, se efectúa la implementación del cliente junto con su interfaz y sus principales componentes.

5.1. Instalación y Configuración

Para el proyecto se trabajó con un servidor propio, al cual se le instalaron todos los componentes necesarios para la ejecución del sistema. A continuación se explicará cómo fue el proceso de instalación y las configuraciones necesarias para trabajar con las herramientas.

5.1.1. Características del Servidor

Para el servidor se utilizó un ordenador portátil antiguo que es bastante básico, pero para realizar el proyecto cumplía completamente. Además de poder instalar lo necesario también fue posible configurar todo de forma que el servidor solamente quedara funcionando para este único proyecto. A continuación se describirá de forma general las características más importantes de este servidor:

- Modelo: Compaq C700 notebook.

- CPU: Intel T1400 1.7GHz.
- RAM: 2 GB.
- Disco: 80 GB.
- Sistema Operativo: Ubuntu 16.04.
- Año: 2010.

5.1.2. PostgreSQL/PostGIS

A continuación se explicará la instalación y configuración de la base de datos PostgreSQL y el complemento PostGIS en el servidor.

Instalación

Para la instalación de PostgreSQL en nuestro servidor con Ubuntu primero actualizamos los paquetes que ya están instalados, luego utilizamos los paquetes de PostgreSQL, que se pueden instalar fácilmente utilizando el sistema de paquetería apt. Podemos instalar el paquete PostgreSQL y un paquete contrib que agrega algunas funciones y utilerías adicionales. Además después de instalar PostgreSQL instalamos la extensión de PostGIS. Esto se realiza de forma automática, se descarga e instala la base de datos en nuestro sistema.

```
1 $ sudo apt-get update
2 $ sudo apt-get install postgresql postgresql-contrib
3 $ sudo apt-get install -y postgis postgresql-9.5.5-postgis-2.2.1
```

La versión finalmente instalada de PostgreSQL es la 9.5.5 y para la extensión PostGIS la 2.2.1.

Configuración

Ahora que ya está instalado PostgreSQL y su complemento espacial PostGIS, se procede a crear una base de datos espacial para el sistema. Para esto utilizamos la consola en Linux con el siguiente código:

```
1 $psql -U nombreUsuario nombreBaseDatos
2 $psql -U nombreUsuario -d nombreBaseDatos -c "CREATE EXTENSION postgis;"
3 $psql -U nombreUsuario -d nombreBaseDatos -c "CREATE EXTENSION
   postgis_topology;"
```

Una vez creada la base de datos espacial revisamos su correcta creación, mediante pgAdmin III conectándonos desde el sistema Windows al servidor, verificando la existencia de las tablas con la información espacial las cuales son: geometry columns que corresponde al soporte para datos de tipo geométrico y spatial ref sys que consiste en aproximadamente tres mil sistemas de referencia espacial.

5.1.3. MapServer

A continuación se explicará la instalación y configuración del servidor de mapas MapServer.

Instalación

Se explicará como instalar Mapserver 7.0.2 en el servidor con sistema Ubuntu, todo esto se realiza mediante la conexión desde Windows al servidor con la herramienta PuTTY. Lo primero a realizar es instalar los requisitos previos que necesitan para MapServer, es decir el servidor web Apache y PHP. Luego al igual que para la base de datos utilizamos el sistema de paquetería apt para instalar MapServer y por último se debe habilitar CGI en Apache, esto se realiza de la siguiente forma:

```
1 $ sudo apt-get install apache2
2 $ sudo apt-get install cgi-mapserver mapserver-bin
3 $ sudo a2enmod cgi sudo service apache2 restart
```

Una vez finalizada la instalación revisamos que el servidor esté funcionando correctamente ingresando en nuestro navegador web a la dirección `http://localhost/cgi-bin/mapserv?` y si la instalación fue exitosa se debe poder visualizar en la pantalla el mensaje “No query information to decode. QUERY_STRING is set, but empty.”.

Configuración

Para configurar MapServer se necesita de un archivo Mapfile, este archivo de configuración lo usa MapServer para convertir los datos geospaciales en imágenes o datos vectoriales. Los propósitos principales son: definir las capas que pueden generar, obtener los datos necesarios y cómo dibujar las capas.

A continuación se puede observar una parte del código del archivo Mapfile que contiene la creación de la capa para la contaminación MP2.5 :

```

1 LAYER
2   NAME pm25
3   METADATA
4       "wms_srs" "EPSG:32719"
5       "wms_name" "pm25"
6       "wms_server_version" "1.1.1"
7       "wms_format" "image/png"
8       "wms_enable_request" "*"
9       "wms_transparent" "true"
10  END
11
12  TYPE raster
13  OPACITY 70
14  STATUS ON
15  VALIDATION
16      'tipointer' '^ [0-9]{1,100}$'
17  END
18  #Consulta para obtener los datos.
19  DATA "PG:host='localhost' port='5432' dbname='proyecto' user='
20      postgres' password='password' schema='public' table='vista_
21      raster' where='id_tipo_interpolacion=%tipointer%AND id_tipo_
22      medicion=1 AND ( fecha_registro BETWEEN %fechaini%AND %
23      fechafinal%)' mode='2'"
24
25  CLASS
26  EXPRESSION ([pixel] >= 0 AND [pixel] <50)
27  STYLE
28  COLORRANGE 255 255 255 "#14D80D"
29  DATARANGE 0 50
30  RANGEITEM "pixel"
31  END #STYLE
32  END #end class

```

```
29  
30     PROCESSING "SCALE=AUTO"  
31  
32 END
```

La principal propiedad que tiene nuestro Mapfile para mostrar los datos está definida por:

- **LAYER:** Define el acceso y la forma de mostrar un conjunto de datos espaciales.

Si miramos un poco más de cerca el objeto LAYER, veremos que contiene una serie de propiedades y objetos. Entre ellos están:

- **STATUS:** La capa puede ser dibujada (ON), no puede ser dibujada (OFF) o es siempre dibujada (DEFAULT).
- **TYPE:** El tipo de geometría que MapServer debería usar para dibujar los datos en la imagen. En nuestro mapfile, los datos serán dibujados como raster.
- **DATA:** El nombre del archivo que usa la capa para los datos, en nuestro caso es una consulta directamente a la base de datos.
- **CLASS:** Determina cómo dibujar la capa (estilo) en el mapa. En nuestro caso se definieron varios tipos de CLASS para poder distinguir los diferentes valores de la contaminación en una misma imagen.

Para más información sobre el archivo completo del Mapfile revisar el Anexo A.

5.2. Implementación de la Base de Datos

A continuación se detallará los pasos que se realizaron para procesar los datos en bruto y llevarlos a nuestra base datos espacial, además de la implementación de la generación de la imagen en el servidor dependiendo el tipo de interpolación.

5.2.1. Procesamiento de Datos

Lo primero fue crear la base de datos para el sistema, que se basa del diseño que se muestra en la Sección 4.4.1. del Capítulo 4. Para ello se creó un archivo .SQL, donde se incluyen todas las entidades, atributos y relaciones de cada clase. Para conocer más de este archivo .SQL revisar el Anexo B.

Los datos fueron generados por una simulación basada en 58 estaciones de medición dentro de la ciudad de Curicó. Esta simulación fue realizada por el profesor guía y se crearon dos archivos Excel: los registros de la contaminación de un día completo , con registros cada una hora; y el otro archivo eran las 58 estaciones con sus coordenadas geográficas.

Estos archivos se utilizaron de base para adaptarlos a nuestro diseño de datos y sistema, para ello mediante un programa básico en Java se agregaron datos necesarios, como diferentes ID o nombres ficticios.

En el Cuadro 5.1 se puede apreciar parte de la tabla de registros, la cual contiene el formato de los datos generados. Cada registro tiene un id, fecha, id_estacion, tipo_medición y valor de la contaminación.

Registros				
id_registro	id_estacion	id_tipo_medición	fecha_registro	valor
1	1	1	2017-05-16 00:00:00	63.0302391052
2	1	1	2017-05-16 01:00:00	58.6373100281
3	1	1	2017-05-16 02:00:00	60.8756866455
4	1	1	2017-05-16 03:00:00	56.4422874451
5	1	1	2017-05-16 04:00:00	61.9681663513
6	1	1	2017-05-16 05:00:00	47.0621032715

Cuadro 5.1: Parte de la tabla Excel de registros.

En el Cuadro 5.2 se puede apreciar parte de la tabla de estaciones, la cual contiene las 58 estaciones creadas. Cada estación tiene su longitud y latitud para su ubicación en el mapa, además de un nombre y un id de la comuna a la cual pertenece.

Estaciones				
id_estacion	id_comuna	nombre	longitud	latitud
1	7301	Estación 1	-71.2051917128	-34.974101859

Sigue en la página siguiente.

id_estacion	id_comuna	nombre	longitud	latitud
2	7301	Estación 2	-71.2069427594	-34.9702503902
3	7301	Estación 3	-71.2431617749	-34.9819552734
4	7301	Estación 4	-71.2392910404	-34.9815022123
5	7301	Estación 5	-71.2373556732	-34.9842960493
6	7301	Estación 6	-71.2422401715	-34.9895059232

Cuadro 5.2: Parte de la tabla Excel de estaciones.

Para insertar estos archivos en la base de datos se utilizó la herramienta PgAdmin, en la cual nos conectamos a la base de datos del servidor y mediante los siguientes comandos cargamos los archivos:

```

1 ->COPY "estacion" (id_estacion ,id_comuna ,propietario ,lon ,lat ) FROM '/
    datosEstaciones.csv' WITH DELIMITER ',' csv header encoding 'ISO
    -8859-1';
2 ->COPY "registro" (id_registro ,id_estacion ,id_tipo ,fecha_registro ,valor
    ) FROM '/datosRegistros.csv' WITH DELIMITER ',' csv header encoding
    'ISO-8859-1';

```

5.2.2. Generación de Imágenes

En esta sección veremos cómo se realizó la confección de las imágenes raster. Para ello nos basamos en el diseño realizado en la Sección 4.3.2. del Capítulo 4 el cual cuenta con tres procesos para la generación de la imagen e inserción en la base de datos. Explicaremos este proceso para cada uno de los dos tipos de interpolación que el sistema tiene.

Script de Interpolación IDW

La primera etapa la generación de la imagen raster es la interpolación, la cual se logra por medio de un script para el programa R. En esta sección podremos observar la interpolación IDW, la cual por medio de su principal función `gstat::idw()` genera los nuevos datos interpolados. Además se realiza una consulta directa a la base de

datos por medio de un SQL que se pasa en los argumentos del Script. Para ello fue necesario crear el siguiente Script:

```

1 # conexion a la base de datos.
2 BD <- odbcConnect("proyecto")
3 # Recoleccion de argumentos.
4 args <- commandArgs(TRUE)
5 # Recoleccion de datos.
6 datos=sqlQuery(BD, args)
7 # Trasformacion de los datos.
8 coordinates(datos) <- c("lon", "lat")
9 proj4string(datos) <- CRS("+init=epsg:4326")
10 datos <- spTransform(datos, CRS("+init=epsg:32719"))
11 # Ajustando tamaño de celda.
12 grd=spsample(datos, type = "regular", cellsize = c(100,100))
13 gridded(grd)=TRUE
14 # Interpolacion IDW.
15 salida <-gstat::idw(valor ~ 1, datos, newdata=grd, idp=2)

```

Para generar la imagen raster en el programa R se utiliza la función raster. Luego con esta imagen le damos un formato con la función writeRaster(), la cual necesita un nombre y el formato en cual se necesita la imagen, en nuestro caso el formato .TIF. Para esto utilizamos los siguientes comandos:

```

1 # Configuracion de proyeccion.
2 proj4string(salida) <- CRS("+init=epsg:32719")
3 r=raster(salida["var1.pred"])
4 # Creacion archivo raster .tif
5 writeRaster(r,"archivo_idw.tif", "GTiff", overwrite=TRUE)
6 # salir de R.
7 q()

```

Como resultado de la interpolación y generación de imagen tenemos un archivo .TIF, que mediante la herramienta QGIS podemos visualizar dentro de un mapa. En la Fig.5.6 se puede apreciar el resultado de la interpolación IDW para una hora y un tipo de material particulado.

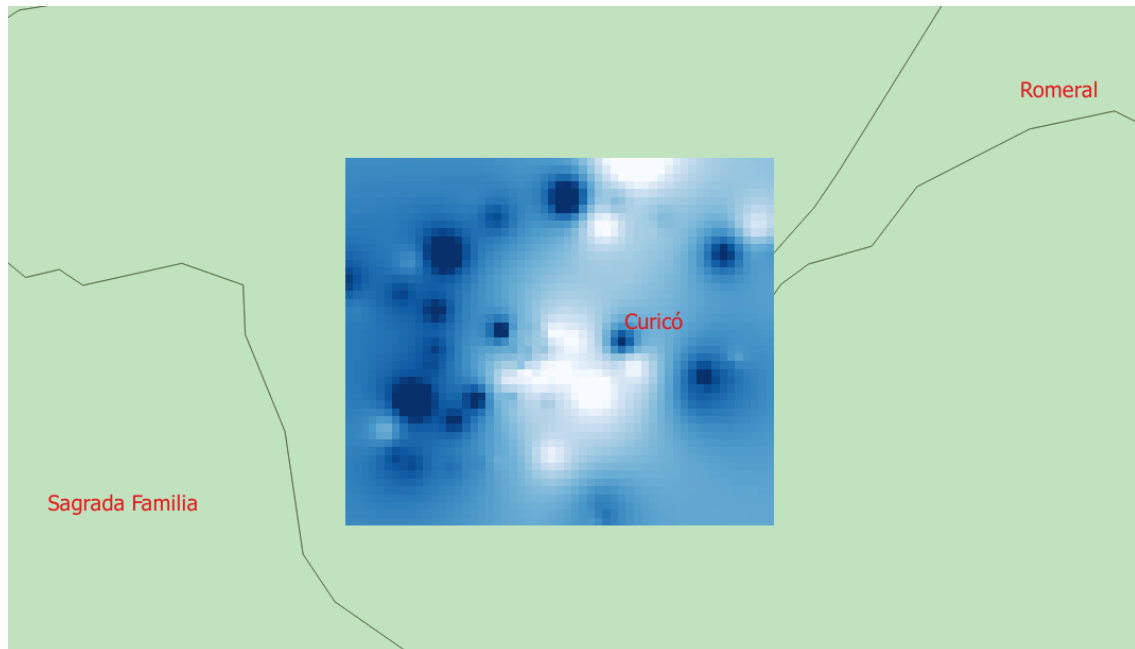


Figura 5.1: Imagen .TIF resultado de interpolación IDW.

Script de Interpolación Kriging

Para el Script de la interpolación Kriging, se utilizó el programa R al igual que para IDW. Para métodos de prueba en este caso la fecha es algo fijo dado por los datos que ya están cargados, pero la idea es que mediante un demonio Cron en el servidor ejecute cada una hora el Script.

Primero se realiza una consulta a la base de datos por los valores de los registros en una fecha en particular y un tipo de contaminación. Esto se realiza con la siguiente consulta:

```
1 SELECT registro.valor , estacion.lat ,estacion.lon FROM registro INNER
   JOIN estacion ON registro.id_estacion=estacion.id_estacion WHERE
   fecha_registro = '2017-05-18 00:00:00' and id_tipo = 2
```

Ahora a diferencia de IDW, Kriging requiere la construcción de un modelo de semivariograma para describir el patrón de auto-correlación espacial para la variable de la contaminación. Para ello se realiza en R el siguiente código:

```
1 # Transformacion de los datos.
2 coordinates(datos) <- c("lon", "lat")
3 proj4string(datos) <- CRS("+init=epsg:4326")
4 datos <- spTransform(datos, CRS("+init=epsg:32719"))
5 # Creacion del semi-variograma.
6 empvario <- variogram(valor ~ 1, datos)
```

En la Fig. 5.2 se aprecia el gráfico del semivariograma resultante del código anterior, en el se puede ver la autocorrelación espacial de los puntos de los datos. Con la información que entrega el diagrama del semivariograma, podemos estimar el umbral, el rango y el nugget a utilizar para nuestro semivariograma modelo final.

En este caso, el rango es el punto del eje de la distancia donde el semivariograma comienza a nivelar, que está alrededor del valor del último retraso, el que es cercano a 2300, por lo que utilizaremos este valor. El umbral es el punto del eje Y donde el semivariograma comienza a nivelarse que es de alrededor de 320. Para el nugget parece ser alrededor de 80, por lo que se ve dónde empiezan los valores en el eje Y en el semivariograma.

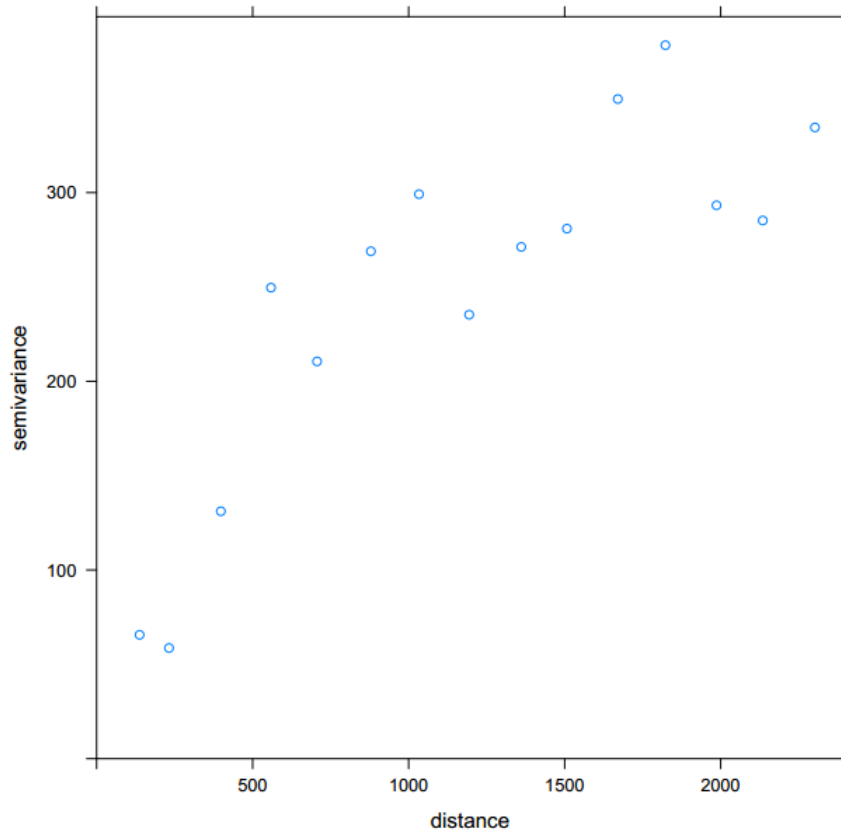


Figura 5.2: Gráfico del Semi-Variograma.

Usando esta información obtenida desde el semivariograma, generaremos un semivariograma modelo con la función `vgm()` en `gstat`. Esto se realiza con el siguiente código en el programa R:

```
1 variofit = fit.variogram(empvario, vgm(psill=320, "Exp", nugget=80,
  range=2300))
```

En la Fig. 5.3 se puede apreciar el gráfico del semivariograma modelo resultante del código anterior. Se utilizó el modelo exponencial (Exp) como base y parece ser un buen ajuste, pero probamos con otros para verificar si es el mejor.

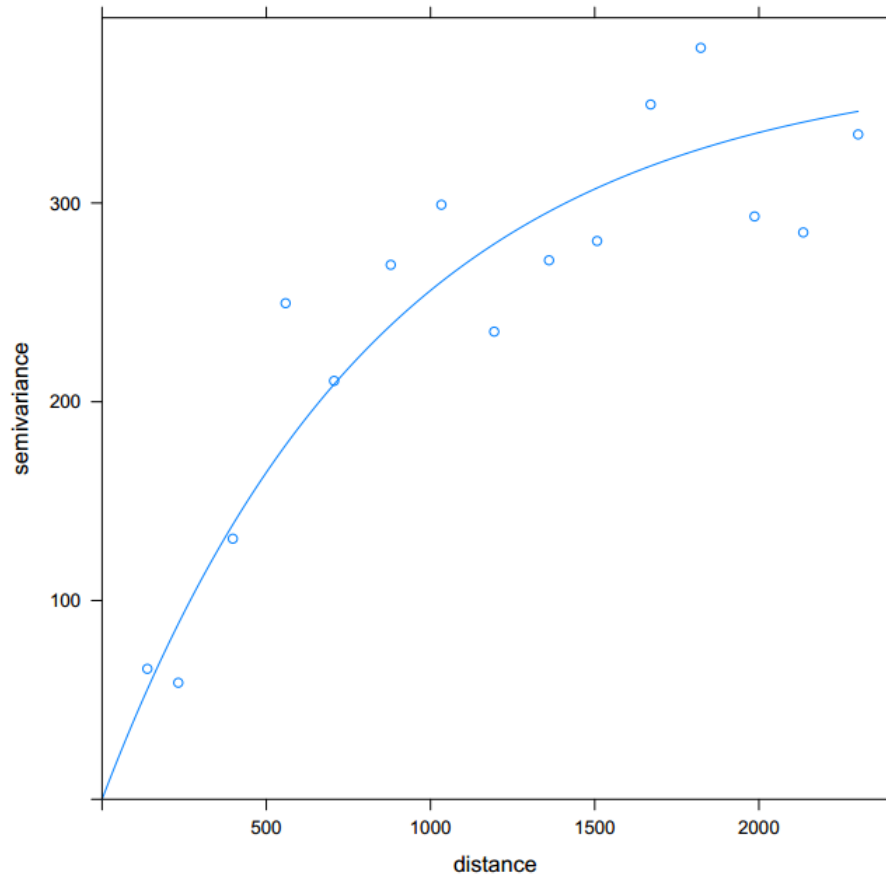


Figura 5.3: Gráfico del SemiVariograma, Modelo Exponencial.

Otro modelo para el semivariograma que podría adaptarse a nuestros datos, es el modelo esférico (Shp). Esto se realiza con el siguiente código en el programa R:

```
1 variofit = fit.variogram(empvario, vgm(psill=320, "Sph", nugget=80,
  range=2300))
```

En la Fig. 5.4 se puede apreciar el gráfico del semivariograma en el cual se utilizó el modelo esférico como base. Se ve claramente que no es mejor que el modelo exponencial, sobre todo en la parte superior del gráfico, donde realiza una curva que no se ajusta a los datos.

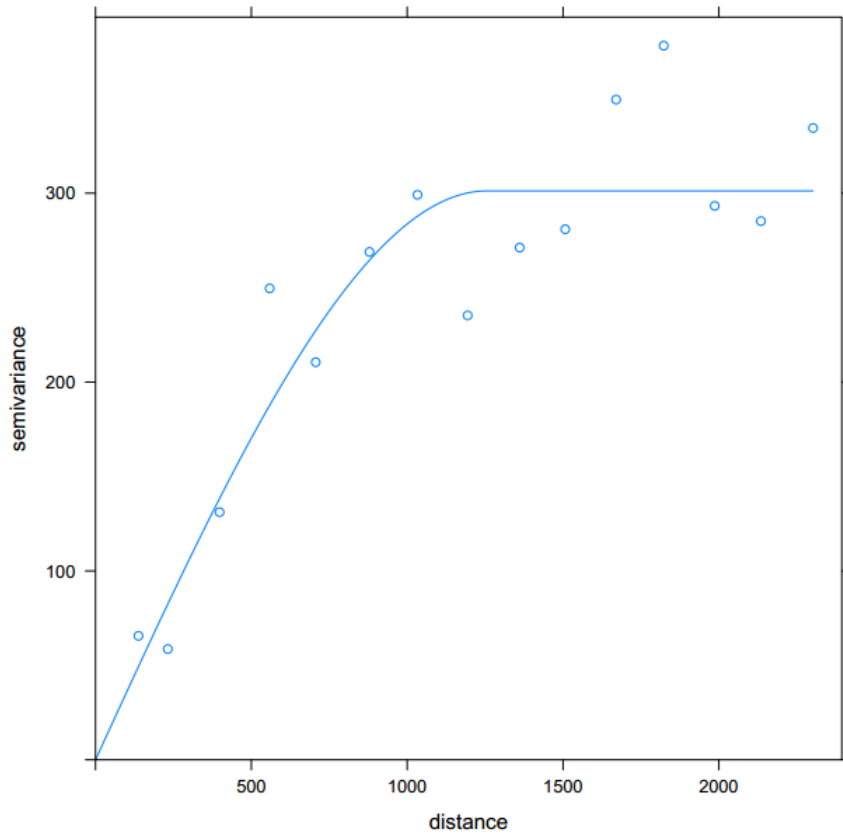


Figura 5.4: Gráfico del SemiVariograma, Modelo Esférico.

Por último probaremos como modelo en el semivariograma al modelo lineal (Lin). Esto se realiza con el siguiente código en el programa R:

```
1 variofit = fit.variogram(empvario, vgm(psill=320, "Lin", nugget=80,
   range=2300))
```

En la Fig. 5.5 se aprecia el gráfico del semi-variograma en el cual se utilizó el modelo lineal como base. Al igual que el esférico no se ajusta correctamente a los datos. Entre los tres modelos el mejor es el exponencial.

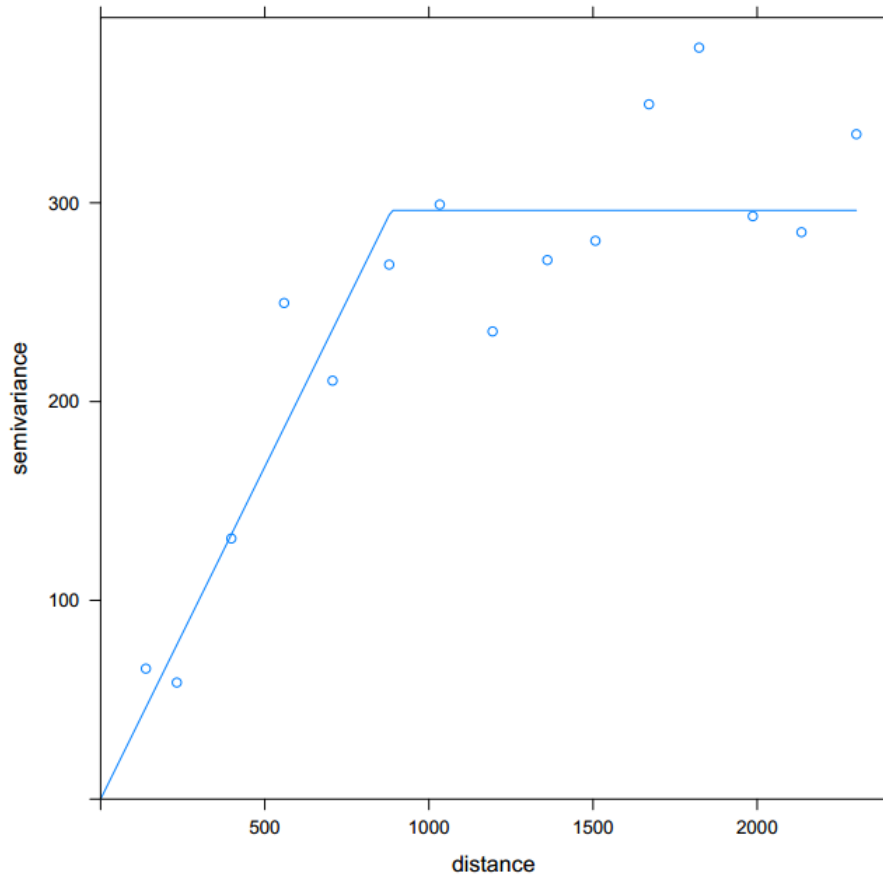


Figura 5.5: Gráfico del SemiVariograma, Modelo Lineal.

Ahora, como sabemos que el mejor modelo base para nuestro semivariograma es el exponencial procederemos a realizar la interpolación de los datos, mediante la función `krige()` en `gstat` junto con el semivariograma del modelo que acabamos de realizar para generar una superficie Kriging ordinaria y también generar la imagen raster `.TIF` en el servidor. Esto se realiza con el siguiente código en el programa R:

```

1 # Interpolacion kriging ordinario.
2 salida <- krige(valor~1, datos, grd, variofit)
3 proj4string(salida) <- CRS("+init=epsg:32719")
4 r=raster(salida["var1.pred"])
5 # creacion archivo raster .TIF .
6 writeRaster(r,"archivo_kriging.tif", "GTiff", overwrite=TRUE)

```



```
7 #salir de R.  
8 q()
```

Como resultado de esta interpolación tenemos un archivo .TIF. Mediante la herramienta QGIS podemos visualizar la imagen dentro del mapa. En la Fig.5.6 se puede apreciar el resultado de la interpolación Kriging ordinario.

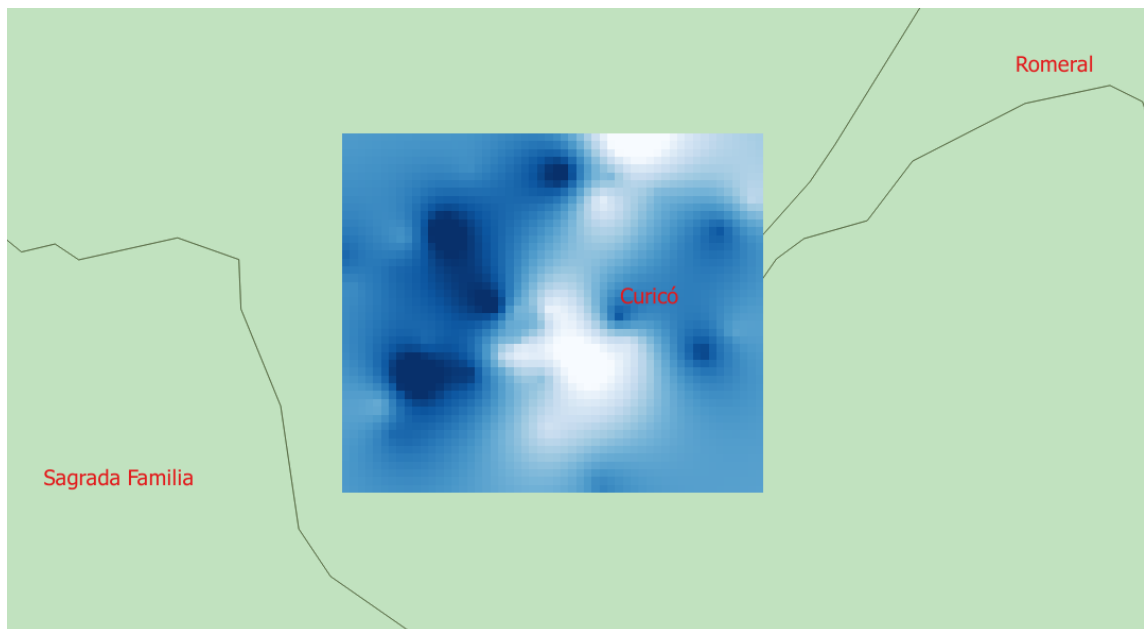


Figura 5.6: Imagen .TIF resultado de interpolación Kriging.

Para más información sobre el archivo completo del Script de las interpolaciones revisar el Anexo C.

5.3. Implementación del Cliente Web

En las secciones anteriores se describió el proceso para almacenar los datos en la base de datos y generar las imágenes mediante el programa R por parte del servidor. A continuación se explicará el proceso final que consiste en cómo presentar las imágenes generadas en el cliente web.

5.3.1. Interfaz

Para la implementación del cliente web, se utilizó un template de Bootstrap y utilizando solamente HTML y CSS para la parte gráfica del sistema. Este template es open source basado en Bootstrap llamado Material Dashboard¹. Para controlar los formularios o las acciones se utilizó únicamente JavaScript. El objetivo principal es que cualquier persona que desee utilizar como base este proyecto pueda elegir el lenguaje que más le acomode.

En la Fig. 5.7 se puede ver el diseño final de la página inicial del sistema, en donde destacan tres zonas: a la izquierda, la zona de navegación, en el centro la zona de mapas y a la derecha la zona de controles.

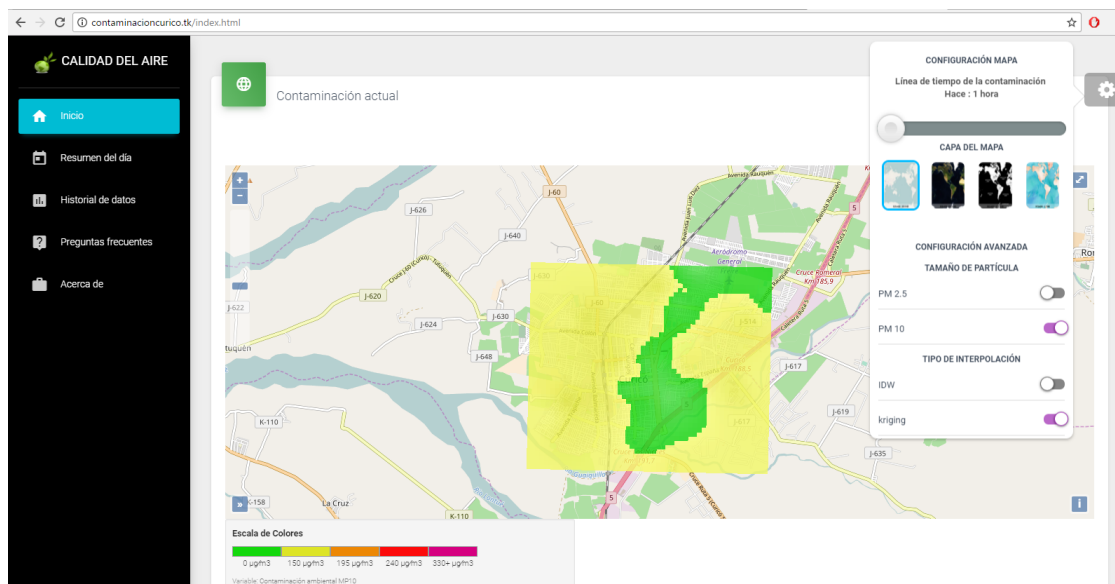


Figura 5.7: Página inicial del sistema.

Otra parte importante en la interfaz, es la visualización de los datos en los gráficos. Para esto se creó la sección de visualización de gráficos mediante un modal, el cual contiene una fecha de inicio y final para los datos a graficar, además de tener las opciones de descarga de estos datos seleccionados en ese tiempo.

¹<https://www.creative-tim.com/>

En la Fig. 5.8 se puede apreciar un gráfico de líneas generado para la estación 1. El gráfico es de tipo línea ya que muestra una serie de datos como un conjunto de puntos conectados mediante una sola línea. Esto sirve para mostrar al usuario cómo la varía la contaminación en un período continuo de tiempo.

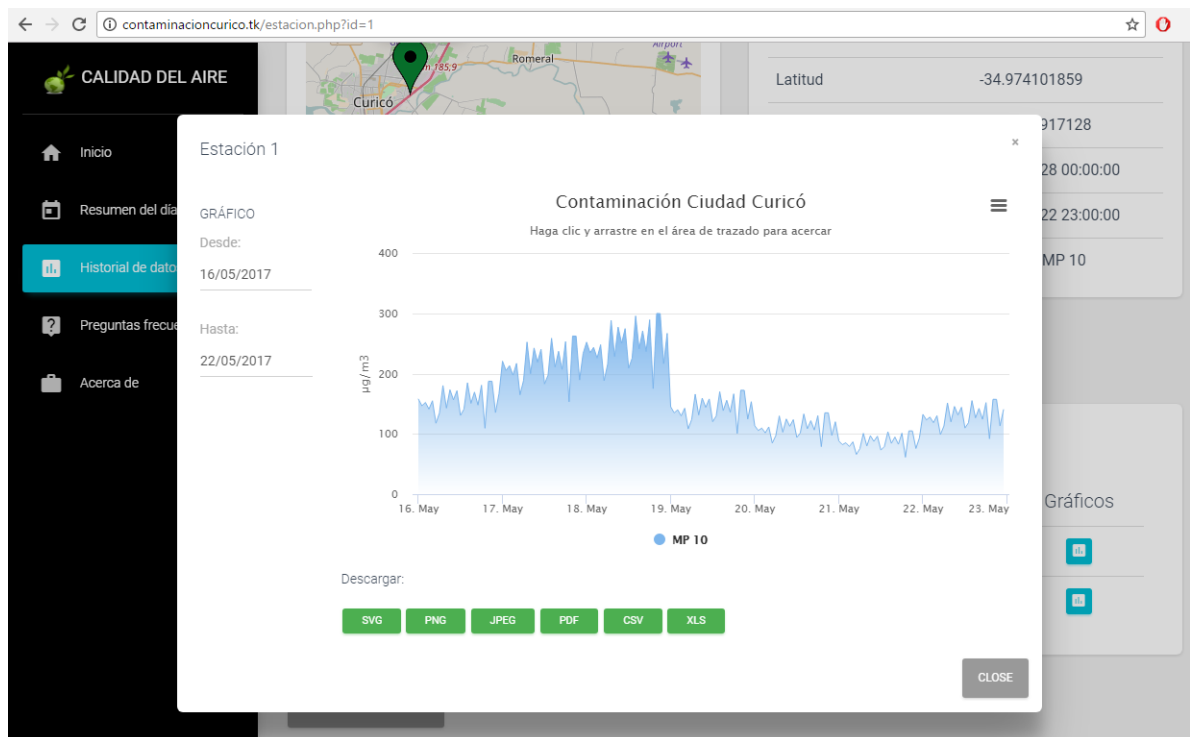


Figura 5.8: Interfaz del gráfico de datos.

5.3.2. Implementación Mapa base

La implementación completa de la visualización de mapas se realizó mediante Openlayer, el cual sólo necesita un div en HTML para poder cargar los mapas en la página. Para incluir un mapa en una página web se deben seguir los siguientes pasos:

- **Incluir librería de Openlayer**

```

1 <link rel="stylesheet" href="https://openlayers.org/en/v4.1.1/css/
  ol.css" type="text/css">
2 <script src="https://openlayers.org/en/v4.1.1/build/ol.js"></
  script>

```

- **Crear un div en el HTML de la página el cual será el contenedor de los mapas**

```

1 <div id="map" class="map" style="width: 100%; height: 100%"></div>

```

- **Mediante JavaScript generar el o los mapas que se deseen cargar en el contenedor.**

```

1 var map = new ol.Map({
2     interactions: ol.interaction.defaults({mouseWheelZoom:
3         false}),
4     controls: ol.control.defaults().extend([
5         new ol.control.FullScreen({
6             source: 'fullscreen'
7         })
8     ]),
9     layers:
10    [mapas1, mp10],
11    target: 'map',
12    view: new ol.View({
13        center: ol.proj.transform([-71.2329, -34.9780], '
14            EPSG:4326', 'EPSG:3857'),
15        zoom: 13
16    })
17 });

```

En el JavaScript anterior podemos identificar los siguientes componentes:

- **Interactions:** En este lugar se realizan las interacciones que tendrá el mapa, en este caso se desactivó la función del zoom con la rueda del mouse.
- **Control:** Estos son los controles que se tienen sobre el mapa, se agregó el control de maximizar el mapa.

- **Layers:** Es un arreglo que es utilizado para definir la lista de las capas que estarán disponibles en el mapa al iniciar, el sistema comienza con un mapa básico y la contaminación MP10.
- **Target:** Es donde se le da el nombre al Id del contenedor donde se cargará el mapa.
- **View:** En este lugar se define el tipo de proyección en la que se trabajará, el centro del mapa, la resolución y el nivel de zoom del mapa.

5.3.3. Controles de Mapa

En esta sección se definen los controles creados en este proyecto para el manejo de las capas en el mapa. En la Fig. 5.9 se puede apreciar los 4 controles que existen sobre el mapa, los que aparecen al lado derecho del sitio web en la página principal.

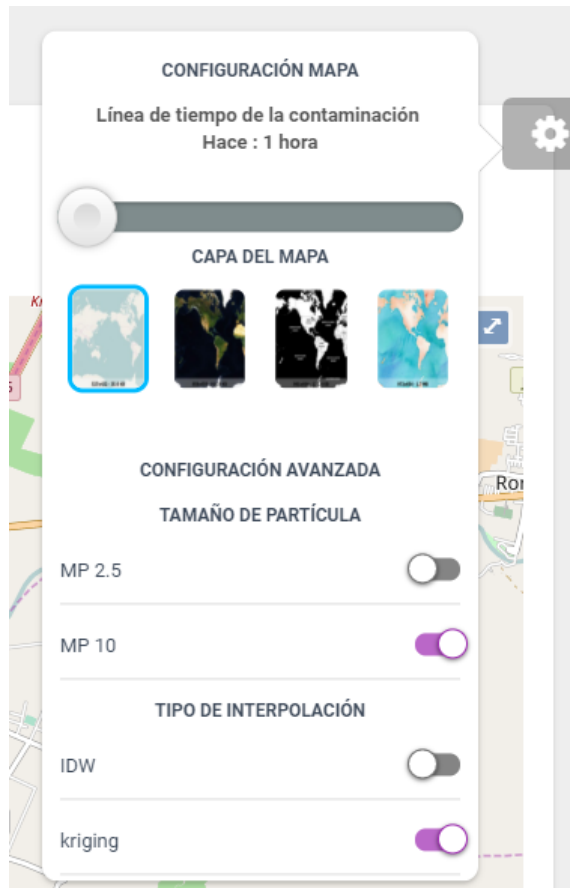


Figura 5.9: Interfaz de Controles de Mapa.

Configuración Mapa

Este control sirve para modificar la capa de la contaminación que actualmente se está visualizando en el mapa. Este control es un rango que varía de 1 a 24, que corresponde a las horas en que se puede ver la contaminación hacia atrás en el tiempo. Es de mucha ayuda para ver cómo ha ido evolucionando la contaminación en la ciudad a través del tiempo.

Capa del Mapa

Este control actúa sobre la capa base del mapa, contiene 4 imágenes en las cuales se refleja cómo es la capa elegida.

Tamaño de Partícula

Este control actúa sobre la capa de contaminación que se está visualizando, contiene dos selectores, cada uno para un tipo de tamaño de partícula con respecto al material particulado (MP2.5 y MP10).

Tipo de Interpolación

Este control actúa sobre la capa de contaminación que se está visualizando, contiene dos selectores, cada uno para un tipo de interpolación de los datos (IDW y Kriging).

5.3.4. Implementación de los Gráficos

Para la implementación de los gráficos se utilizó la librería de JavaScript Highcharts, la cual mediante Ajax logra la obtención de los datos del servidor. Para incluir un gráfico en una página web se deben seguir los siguientes pasos:

- **Incluir librería de Highcharts**

```
1 <script src="https://code.highcharts.com/highcharts.js"></script>
2 <script src="https://code.highcharts.com/modules/exporting.js"></script>
```

- **Crear un div en el HTML de la página el cual será el contenedor del gráfico**

```
1 <div class="highcharts-container" id="container" style="min-width:
   300px; height: 400px; margin: 0 auto"></div>
```

- **Mediante JavaScript generar el gráfico para cargar en el contenedor.**

```
1 $.ajax({
2   data: {"id" : id, "idTipo":idTipo, "inicial" : desde, "final":
   hasta},
```

```
3     type: "POST", dataType: "json", url: "registros.php",
4   })
5   .done(function( datos, textStatus, jqXHR ) {
6     chart = Highcharts.chart('container', {
7       chart: {
8         zoomType: 'x'
9       },
10      title: {
11        text: 'Contaminacion Ciudad Curico'
12      },
13      xAxis: {
14        type: 'datetime',
15
16        labels: {
17          overflow: 'justify'
18        }
19      },
20      series: [{
21        type: 'area',
22        name: nombreTipo,
23        data: datos
24      }],
25      navigation: {
26        menuItemStyle: {
27          fontSize: '10px'
28        }
29      }
30    });
```

En el JavaScript anterior podemos identificar algunos componentes, como es el caso del Ajax el cual necesita el Id de la estación, el tipo de medición, la fecha inicial y la fecha final seleccionada para poder graficar. Con esto se realiza la consulta a la base de datos para luego crear el gráfico de Highchart con los datos entregados por el Ajax.

Como resultado de lo anterior, podemos ver la Fig. 5.10 la cual tiene en el centro el gráfico, a su lado izquierdo se encuentra la zona para seleccionar fecha inicial y final, Además en la parte baja de la página se encuentra la zona de descarga de los datos visualizados.



Figura 5.10: Interfaz gráfico.

6. Pruebas

En este capítulo se presentan las pruebas realizadas al sistema. Se desarrollaron dos tipos de pruebas: una que se centra en la funcionalidad de generación de imágenes raster; y la otra que se basa en una evaluación de usabilidad realizada al sistema web por los usuarios.

6.1. Funcionalidad

En esta sección se verificarán las funcionalidades principales y su correcto funcionamiento. Para ello se desarrollaron pruebas funcionales al sistema. El procedimiento que se utilizó consiste en probar la funcionalidad de generación de imagen, mediante comparación entre la imagen raster que realiza el sistema web, y la imagen generada por la herramienta QGIS, para lo cual se aplicaron las siguientes consideraciones:

- Se utilizaron datos simulados de tamaño MP2.5.
- Se utilizaron ambas interpolaciones (IDW y Kriging).

En este documento se presentan los resultados de las pruebas para la generación de imágenes de la contaminación de tamaño MP2.5, pero también es válido para el otro tipo de contaminante, sólo se debe proceder de forma similar.

6.1.1. Datos de Prueba

Para las pruebas realizadas se utilizaron los datos cargados para un mismo día y a la misma hora. Los valores de estos datos se pueden apreciar en el Cuadro 6.1, donde aparece el id de la estación y el valor correspondiente a la contaminación.

Como se mencionó anteriormente en el documento, las estaciones cargadas a la base de datos son 58.

id_estacion	valor	id_estacion	valor
1	157.575597763	30	138.379983902
2	141.87311172475	31	115.64218521125
3	178.927822113	32	171.212978363
4	152.903213501	33	136.51803016675
5	150.6263542175	34	134.13855552675
6	153.99703025825	35	151.754770279
7	149.711637497	36	154.1570949555
8	126.76901817325	37	155.32931327825
9	132.34201431275	38	167.3695755005
10	149.5404815675	39	170.95306396475
11	131.79521560675	40	152.3973083495
12	151.9360637665	41	102.2917842865
13	167.11376190175	42	181.04185104375
14	160.59757232675	43	157.66834259025
15	181.366329193	44	161.346797943
16	159.14789199825	45	162.840938568
17	168.868045807	46	165.11955261225
18	160.93997955325	47	174.95237350475
19	167.86462783825	48	165.53199768075
20	171.3652420045	49	163.0481529235
21	186.42555236825	50	156.24302864075
22	153.79512786875	51	138.1113910675
23	167.60362625125	52	150.85695266725
24	167.549610138	53	130.75140953075
25	163.245124817	54	168.31426620475
26	170.58151245125	55	152.30432510375
27	174.77930068975	56	130.0646305085
28	160.02180099475	57	134.49154853825

Sigue en la página siguiente.

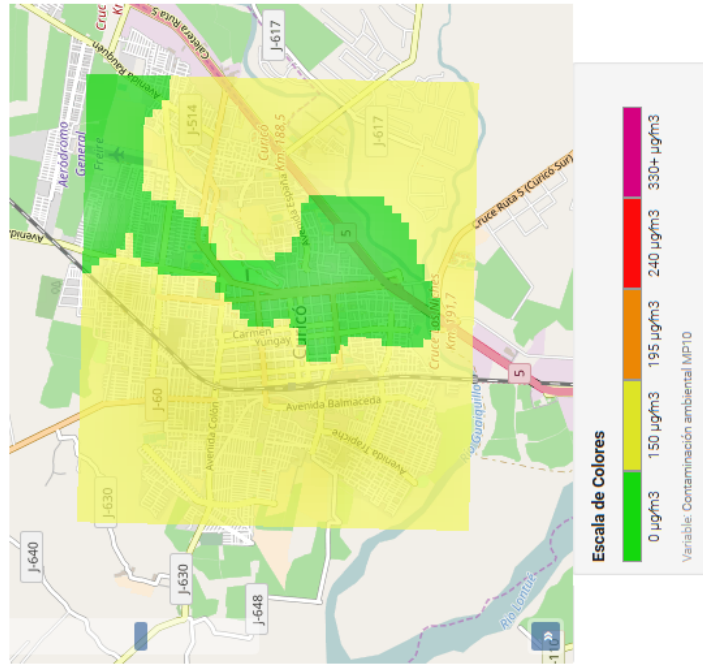
id_estacion	valor	id_estacion	valor
29	154.43017005925	58	149.035987854

Cuadro 6.1: Datos de Prueba.

6.1.2. Comparación

Para las pruebas se utilizaron los datos cargados de un mismo día y a la misma hora. Para comparar se crearon dos imágenes por interpolación: una imagen es generada con la herramienta QGIS y la otra con el sistema web. La idea es poder verificar si la imagen generada por el prototipo se asemeja a la realizada por la herramienta.

En la Fig. 6.1 se puede apreciar la comparación de imágenes para el caso de la interpolación IDW, donde la imagen de color azul es la de QGIS y la otra imagen es del sistema web. En la Fig. 6.2 se muestra la comparación para la interpolación Kriging. Al igual que en la figura anterior, la imagen de color azul es la de QGIS.



IDW

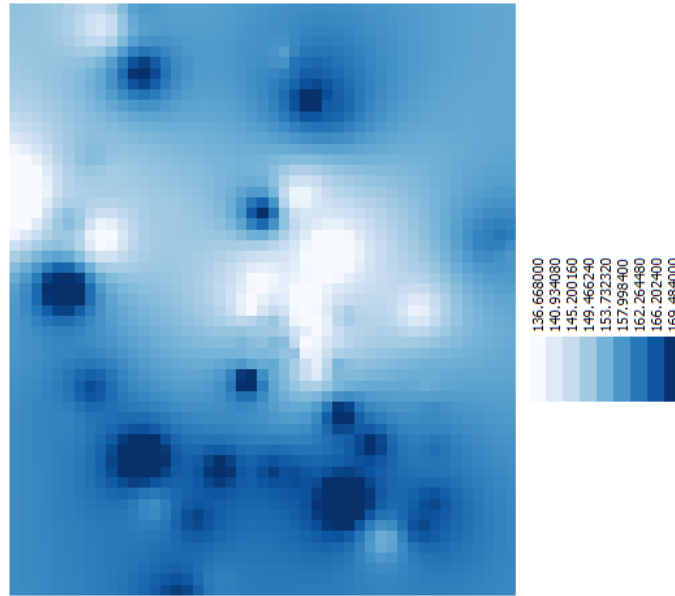
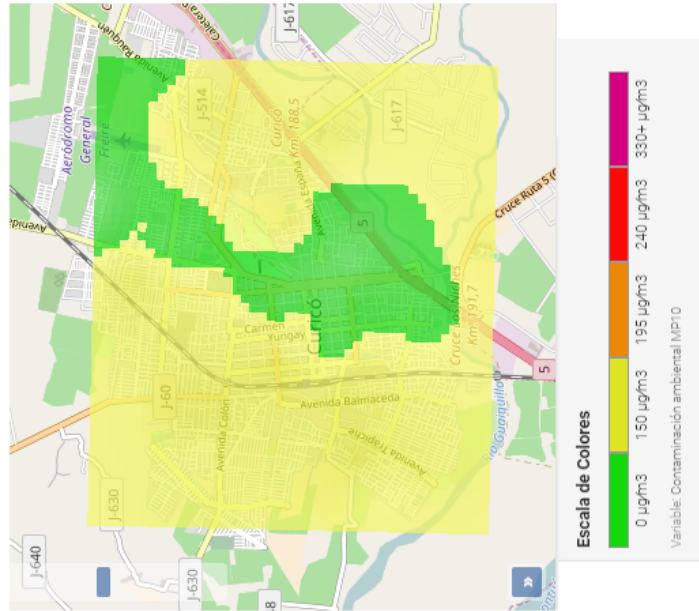


Figura 6.1: Comparación entre imágenes IDW.



Kriging

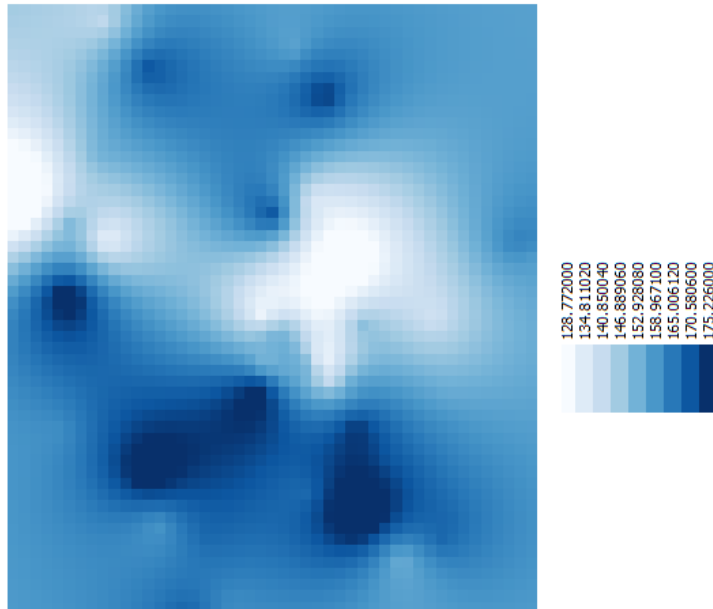


Figura 6.2: Comparación entre imágenes Kriging.

6.1.3. Resultado

En ambas figuras se puede ver que los resultados son similares. Por parte del sistema web no se aprecia mucho el cambio entre las dos interpolaciones, en cambio entre las generadas por la herramienta QGIS se visualiza una pequeña modificación. En la imagen de la interpolación de Kriging se aprecia una mayor suavidad entre celdas, no se producen cambios de colores tan bruscos en comparación a la imagen de la interpolación IDW.

En cambio en las imágenes generadas por el sistema esta variación generada entre interpolaciones no es visible ya que los rangos de colores son menores, por lo tanto no es posible distinguir estos cambios. Estos rangos de colores están fijados por la norma ambiental del país, en donde se indica por medio de un color en particular la contaminación que afecta una zona.

6.2. Evaluación de Usabilidad

La realización de la evaluación de usabilidad sobre el diseño implementado es beneficioso para conocer el grado en el que los usuarios son capaces de usar el sistema y si éste se adapta al modelo mental de uso que tienen los usuarios. Se aplicó una encuesta de usabilidad para poder evaluar a los usuarios.

6.2.1. Encuesta de Usabilidad

La encuesta realizada para este proyecto fue diseñada para obtener datos útiles para conocer la experiencia del usuario en aquellas áreas que podría haber más problemas para el usuario o cuya decisión de diseño no fue evaluada previamente a la implementación. Orientar las preguntas de la encuesta en esta dirección permite que podamos conocer la experiencia del usuario con las partes más complejas y evaluar si el equipo ha sido capaz de hacer un diseño apropiado cuando no ha podido evaluar con usuarios reales los elementos a diseñar.

Acorde con lo descrito anteriormente se creó una encuesta (véase Anexo D) que evaluase los siguientes elementos de la plataforma.

- Navegación por la plataforma.
- Visualización del mapa.

- Control sobre el mapa.
- Control sobre el gráfico.
- Botones de acción.

La evaluación de los elementos citados anteriormente se realizó mediante escalas de diferenciación semántica. Estas escalas ofrecen al usuario una lista de pares de adjetivos opuestos entre sí, de forma que el usuario indique, mediante la escala ofrecida, el grado en el que cada par de elementos describe mejor el elemento evaluado. La lista ofrecida al usuario contenía los siguientes pares de conceptos:

- Muy Difícil – Muy Fácil.
- Difícil de entender - Fácil de entender.
- Confuso – Muy claro.
- Demasiado lento – Muy rápido.
- Muy malo – Excelente.

6.2.2. Resultados

En esta sección se estudiarán los resultados obtenidos en la encuesta, la cual fue aplicada a 18 personas de manera online y de forma presencial. Estas personas eran de diversas capacidades de manejo con la informática. A continuación se mostrarán los resultados de las preguntas que son las de mayor importancia dentro del diseño del sistema.

Preguntas

- Para saber el tipo de usuario que realizó la encuesta, se creó la siguiente pregunta:

- ¿Qué nivel de conocimientos de informática tiene usted?

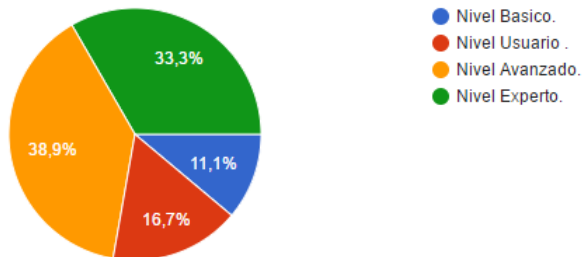


Figura 6.3: Nivel de conocimiento de informática.

En la Fig. 6.3 se puede ver que la mayoría de los usuarios encuestados tiene un nivel avanzado de conocimiento sobre la informática, esto es porque gran parte de los usuarios encuestados son estudiantes de ingeniería e ingenieros ya egresados, también existen en menor cantidad encuestados de nivel básico y usuario.

- Otra parte importante dentro de la interfaz del sistema es si el usuario tiene la capacidad de saber de qué se trata el sitio web. Para ello se creó la siguiente pregunta:
 - **En la página principal del sitio observe la interfaz y responda:**
¿Con la información que se ofrece en la pantalla, es posible saber de qué se trata el sitio web?

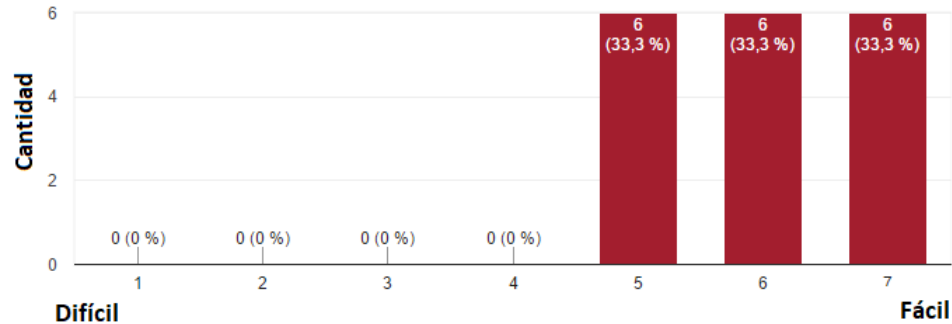


Figura 6.4: Gráfico Pregunta de Interfaz.

En la Fig. 6.4 se aprecia que la mayoría de los usuarios encuestados se inclinó por la opción más cercana a fácil. Esto quiere decir que entendieron de qué se trataba el sitio web con sólo mirar la interfaz principal, aunque no se entregue mucha información en la página principal es intuitivo el propósito del sitio, esto es principalmente por el mapa y la imagen de la contaminación dentro de éste.

- Para saber si el usuario entendió sobre la imagen de la contaminación que se muestra en el mapa, se realizó la siguiente pregunta:
 - ¿Queda claro a partir del mapa el nivel de contaminación que existe en la ciudad?

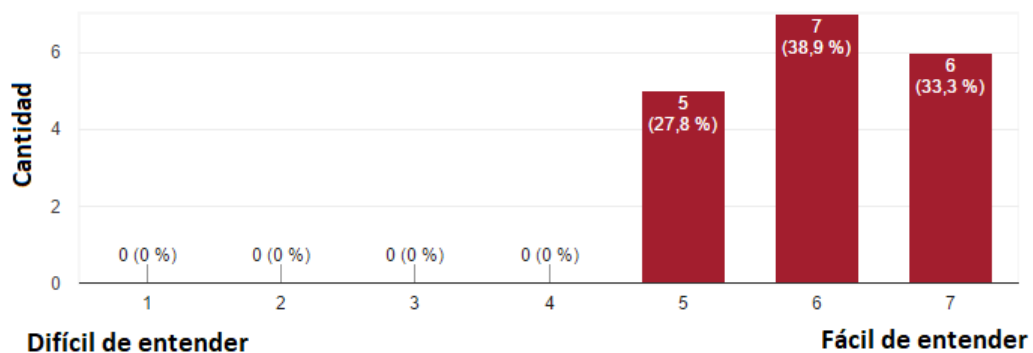


Figura 6.5: Gráfico Pregunta sobre Imagen.

En la Fig. 6.5 se puede deducir que la mayoría de los usuarios encuestados se inclinó por la opción más cercana a fácil de entender. Esto quiere decir que se logró el objetivo de informar sobre la contaminación que existe en la ciudad por medio de la imagen generada. El usuario con sólo mirar el mapa puede saber la contaminación que afecta a un lugar.

- Para saber cómo el usuario interactúa con el sistema web, se creó una pregunta pero antes tiene que realizar una tarea en particular, para ello se realizó la siguiente pregunta:
 - **Ahora debe realizar una tarea en particular: En historial de datos, se quiere que pueda ver los datos para la estación 5, luego graficar la medición de MP10 con fecha entre 20-05-2017 hasta el 22-05-2017. Además, descargar los datos en algún formato. ¿Qué tan fácil o difícil fue realizar la tarea antes mencionada?**

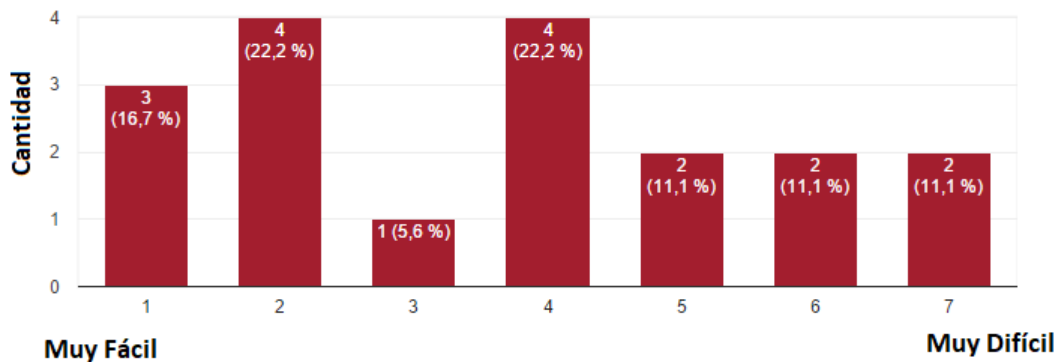


Figura 6.6: Gráfico Pregunta sobre Interacción.

En la Fig. 6.6 se puede apreciar las diferentes opiniones de los usuarios encuestados. La tarea que se indicó realizar es la más difícil en todo el sitio web. El gráfico demuestra que existe una tendencia de la mayoría de los usuarios que al realizar la tarea no les fue tan complicado, pensando que es la tarea más difícil. Se entiende que los usuarios de conocimiento básico de la informática encuentren un poco más complicada la ejecución de esta tarea.

- La pregunta final de esta encuesta se relaciona a la percepción que tuvo el usuario hacia la utilidad que entrega el sistema, se realizó la siguiente pregunta:
 - **Respecto a la utilidad que ofrece el sitio web, ¿Qué es lo que más te llamó la atención positivamente o negativamente?**

Como esta pregunta necesita de una respuesta escrita, se seleccionó las respuestas tanto negativas como positivas más repetidas en la encuesta. Estas respuestas son:

- La posibilidad de visualizar la contaminación sobre el mapa de la ciudad.
- Se ve ordenada y fácil de utilizar.
- En la versión móvil, no se puede ver el detalle de una estación.
- Cambiaría de ubicación ciertos elementos

En las respuestas positivas se destaca en gran cantidad lo fácil de utilizar el sitio, como también la funcionalidad de poder visualizar la contaminación en el mapa, esto llama la atención de los usuarios lo simple y funcional del sitio web. Entre las pocas respuestas negativas podemos decir hablan sobre el funcionamiento del sistema en un aparato móvil y de algunos componentes que no fueron del agrado de los usuarios. Cabe destacar que el sistema web no está optimizado completamente para una versión móvil ya que esto se encuentra en los alcances del proyecto.

7. Conclusiones

El desarrollo de este proyecto permitió aplicar el conocimiento del área de computación, a una que no tiene mucha relación, como es el caso de visualización para la calidad del aire, mediante la implementación de un SIG que facilita la labor de los habitantes y fiscalizadores de la ciudad, a tomar decisiones sobre algo tan importante como es la contaminación que afecta gravemente a la ciudad, además que son perjudiciales tanto para las personas como los animales.

Se logró demostrar que es posible implementar un sistema específico que dado un conjunto de datos, éstos pueden ser georreferenciados y procesados para generar imágenes y que éstos también sean fácilmente visualizados mediante una aplicación web. Además, la visualización en forma de mapas permite una fácil interpretación de cada uno de los datos asociados a la contaminación. Por la parte tecnológica, quedó demostrado que es posible implementar un SIG que genera imágenes mediante el programa estadístico R, relegando casi la totalidad del trabajo al servidor y la base de datos. Esto permite que pueda ser utilizado por cualquier persona sin la necesidad de poseer equipos con una gran capacidad de cómputo y que mediante un navegador web pueda visualizar la totalidad de los datos. Además, como fue posible realizar este proyecto con la contaminación de material particulado, es posible llevar este proyecto a cualquier otro contaminante, incluso no ligado al estudio de la calidad del aire. Esto es posible mediante algunos ajustes mínimos para procesar y visualizar cualquier tipo de datos existentes en una base de datos.

En cuanto a los objetivos, éstos se cumplieron completamente, desde la implementación de la infraestructura necesaria como la arquitectura y base de datos, hasta la generación y visualización de las imágenes en el mapa. Además, se agregó una sección que no estaba considerada en el sitio pero trabajando en el sistema se dio la

posibilidad de informar de alguna forma al usuario. Esta es la sección de preguntas frecuentes.

Por último cabe destacar que fue de mucha satisfacción el desarrollo de este proyecto, en el cual se puso en práctica diversas áreas, conocimientos y tecnologías aprendidas durante toda la carrera, como es el caso de trabajar con bases de datos, servidores y lenguajes de computación, junto con metodologías y técnicas para el desarrollo de software.

7.1. Trabajo Futuro

En esta sección se presenta un listado con funcionalidades para realizar a futuro. Cabe mencionar que estas funcionalidades no fueron implementadas a lo largo de esta memoria, ya que la cantidad de tiempo que requiere implementarlas demanda mucho más tiempo que el asignado para realizar este trabajo. Es por ello que se implementaron las funcionalidades primordiales y las que permiten cumplir con todos los objetivos planteados al inicio de este trabajo.

- Optimizar sitio web para dispositivos móviles.
- Integración con las redes sociales.
- Implementar un sistema de registro y autenticación de usuarios para mejorar la seguridad del sistema.
- Implementar un sistema de alerta que avise al usuario sobre la contaminación (Correo o SMS).

Glosario

Interpolación: Es la obtención de nuevos puntos partiendo del conocimiento de un conjunto discreto de puntos.

Semivariograma: Representa la autocorrelación espacial de los puntos de muestra medidos.

Raster: El mundo es representado a través del uso de imágenes en las que cada celda (píxel) se le asigna algún valor, lo cual sirve para representar atributos de los objetos.

PostgreSQL: Es un sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD.

PostGIS: Añade soporte para objetos geográficos a la base de datos PostgreSQL.

SIG: Son sistemas de información que permiten obtener, manipular, analizar y visualizar datos georreferenciados.

Script: Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

SQL: Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

MP2.5: Son partículas de diámetro aerodinámico inferior o igual a los 2,5 micrones, es decir, son 100 veces más delgadas que un cabello humano.

MP10: Son partículas de diámetro menor o igual a 10 micrones. Por su tamaño, el MP10 es capaz de ingresar al sistema respiratorio del ser humano.

Bibliografía

- [1] Acerca de mapserver. <http://mapserver.org/es/about.html>, Consultado el 30 de Octubre de 2016.
- [2] Acerca de r. <https://www.r-project.org/about.html>, Consultado el 30 de Octubre de 2016.
- [3] Análisis espacial (interpolación). http://docs.qgis.org/2.6/es/docs/gentle_gis_introduction/spatial_analysis_interpolation.html, Consultado el 20 de Octubre de 2016.
- [4] Concepto base de datos geográfica. <https://www.ign.es/ign/layoutIn/actividadesBDGintro.do>, Consultado el 25 de Octubre de 2016.
- [5] Conceptos del conjunto de herramientas interpolación de ráster. <https://pro.arcgis.com/es/pro-app/tool-reference/3d-analyst/how-kriging-works.htm>, Consultado el 20 de Octubre de 2016.
- [6] Fundamentos de sistemas unidad 2 de información geográfica. http://geoservice.igac.gov.co/contenidos_telecentro/fundamentos_sig, Consultado el 07 de Octubre de 2016.
- [7] Introducción. ¿qué es un sig?. http://volaya.github.io/libro-sig/chapters/Introduccion_fundamentos.html, Consultado el 15 de Octubre de 2016.
- [8] Modelos para la información geográfica. http://volaya.github.io/libro-sig/chapters/Tipos_datos.html, Consultado el 16 de Octubre de 2016.
- [9] Openlayers. <http://openlayers.org/>, Consultado el 30 de Octubre de 2016.

- [10] Procesos de software. <https://procesosdesoftware.wikispaces.com/METODOLOGIA+SCRUM>, Consultado el 07 de Septiembre de 2016.
- [11] Sistema de información nacional de calidad del aire. <http://sinca.mma.gob.cl/>, Consultado el 27 de Agosto de 2016.
- [12] Sistemas de información geográfica - tipos y aplicaciones. <http://sig.cea.es/SIG>, Consultado el 28 de Agosto de 2016.
- [13] Sobre postgis. <http://mappinggis.com/2012/09/por-que-utilizar-postgis/>, Consultado el 30 de Octubre de 2016.
- [14] Sobre postgresql. http://www.postgresql.org.es/sobre_postgresql, Consultado el 25 de Octubre de 2016.
- [15] World air quality index. <http://aqicn.org>, Consultado el 26 de Agosto de 2016.
- [16] Víctor Olaya. Modelo ráster. In *Sistemas de Información Geográfica*, pages 86–87, October 2014.
- [17] Dr. Claudio Vargas R. Efectos de la fracción gruesa (PM10-2.5) del material particulado sobre la salud humana. (3):1–30, 2011.

ANEXOS

A. Archivo Mapfile

En el presente Anexo se encuentra el detalle del archivo Mapfile que genera los mapas del proyecto, se crearon dos capas en el archivo, cada una para el tipo de material particulado (MP2.5 y MP10).

A.1. Capa MP10

Para la generación de la imagen de la contaminación MP10 en el mapa de la ciudad, se creó una capa con el nombre de mp10, ésta contiene la consulta hacia la base de datos. También los diferentes estilos de color para poder diferenciar en la imagen la cantidad de contaminación que existe.

```
1 LAYER
2   NAME mp10
3     METADATA
4       "wms_srs" "EPSG:32719"
5       "wms_name" "mp10"
6       "wms_server_version" "1.1.1"
7       "wms_format" "image/png"
8       "wms_enable_request" "*"
9       "wms_transparent" "true"
10    END
11
12   TYPE raster
13   OPACITY 70
14   STATUS ON
15   VALIDATION
16     'tipointer' '^([0-9]{1,100})$'
```

```
17 END
18 #Consulta para obtener los datos.
19 DATA "PG:host='localhost' port='5432' dbname='proyecto' user='
    postgres' password='password' schema='public' table='vista_
    raster' where='id_tipo_interpolacion=%tipointer%AND id_tipo_
    medicion=2 AND ( fecha_registro BETWEEN %fechaini%AND %
    fechafinal%)' mode='2'"
20
21 CLASS
22     EXPRESSION ([pixel] >= 0 AND [pixel] <150)
23     STYLE
24         COLRRANGE 255 255 255 "#14D80D"
25         DATARANGE 0 150
26         RANGEITEM "pixel"
27     END #STYLE
28 END #end class
29
30 CLASS
31     EXPRESSION ([pixel] >= 150 AND [pixel] < 195)
32     STYLE
33         COLRRANGE "#FBFF75" "#DDE426"
34         DATARANGE 150 195
35         RANGEITEM "pixel"
36     END #STYLE
37 END #end class
38
39 CLASS
40     EXPRESSION ([pixel] >= 195 AND [pixel] < 240)
41     STYLE
42         COLRRANGE "#FFC475" "#EC8704"
43         DATARANGE 195 240
44         RANGEITEM "pixel"
45     END #STYLE
46 END #end class
47
48 CLASS
49     EXPRESSION ([pixel] >= 240 AND [pixel] < 330)
50     STYLE
51         COLRRANGE "#FF683A" "#FC0A0A"
52         DATARANGE 240 330
53         RANGEITEM "pixel"
```

```

54     END #STYLE
55 END #end class
56
57 CLASS
58     EXPRESSION ([pixel] >= 330 AND [pixel] < 400)
59     STYLE
60         COLORRANGE "#FF6BC4" "#D50080"
61         DATARANGE 330 400
62         RANGEITEM "pixel"
63     END #STYLE
64 END #end class
65
66 PROJECTION
67     "init=epsg:32719"
68 END
69
70 PROCESSING "SCALE=AUTO"
71
72 END

```

A.2. Capa MP2.5

Para la generación de la imagen de la contaminación MP2.5 en el mapa de la ciudad, se creó una capa con el nombre de mp25, igual que para la capa anterior esta contiene la consulta hacia la base de datos. También tiene los diferentes estilos de color para poder diferenciar en la imagen la cantidad de contaminación que existe.

```

1 LAYER
2     NAME mp25
3     METADATA
4         "wms_srs" "EPSG:32719"
5         "wms_name" "mp25"
6         "wms_server_version" "1.1.1"
7         "wms_format" "image/png"
8         "wms_enable_request" "*"
9         "wms_transparent" "true"
10    END

```

```
11
12 TYPE raster
13 OPACITY 70
14 STATUS ON
15 VALIDATION
16     'tipointer' '^ [0-9]{1,100}$'
17 END
18 #Consulta para obtener los datos.
19 DATA "PG:host='localhost' port='5432' dbname='proyecto' user='
    postgres' password='password' schema='public' table='vista_
    raster' where='id_tipo_interpolacion=%tipointer%AND id_tipo_
    medicion=1 AND ( fecha_registro BETWEEN %fechaini%AND %
    fechafinal%)' mode='2'"
20
21 CLASS
22     EXPRESSION ([pixel] >= 0 AND [pixel] < 50)
23     STYLE
24         COLRRANGE 255 255 255 "#14D80D"
25         DATARANGE 0 50
26         RANGEITEM "pixel"
27     END #STYLE
28 END #end class
29
30 CLASS
31     EXPRESSION ([pixel] >= 50 AND [pixel] < 80)
32     STYLE
33         COLRRANGE "#FBFF75" "#DDE426"
34         DATARANGE 50 80
35         RANGEITEM "pixel"
36     END #STYLE
37 END #end class
38
39 CLASS
40     EXPRESSION ([pixel] >= 80 AND [pixel] < 110)
41     STYLE
42         COLRRANGE "#FFC475" "#EC8704"
43         DATARANGE 80 110
44         RANGEITEM "pixel"
45     END #STYLE
46 END #end class
47
```

```
48 CLASS
49     EXPRESSION ([pixel] >= 110 AND [pixel] < 170)
50     STYLE
51         COLRRANGE "#FF683A" "#FC0A0A"
52         DATARANGE 110 170
53         RANGEITEM "pixel"
54     END #STYLE
55 END #end class
56
57 CLASS
58     EXPRESSION ([pixel] >= 170 AND [pixel] < 350)
59     STYLE
60         COLRRANGE "#FF6BC4" "#D50080"
61         DATARANGE 170 250
62         RANGEITEM "pixel"
63     END #STYLE
64 END #end class
65
66 PROJECTION
67     "init=epsg:32719"
68 END
69
70 PROCESSING "SCALE=AUTO"
71
72 END
```

B. Base de Datos SQL

En el presente Anexo se encuentra el detalle de los archivos SQL utilizados en la creación de la base de datos para el sistema.

B.1. Base de datos

En esta sección se presenta el archivo principal de la base de datos, ésta contiene todas las entidades y los atributos definidos.

```
1 CREATE database contaminacion ;
2 CREATE EXTENSION postgis;
3
4 drop table if exists estacion cascade;
5 CREATE table if not exists estacion
6 (
7     id_estacion bigint ,
8     id_comuna bigint not null ,
9     propietario varchar (100) not null ,
10    lat double precision not null ,
11    lon double precision not null ,
12    fecha_primer_registro timestamp null ,
13    fecha_ultimo_registro timestamp null ,
14    PRIMARY KEY (id_estacion) ,
15    FOREIGN KEY (id_comuna) REFERENCES comuna("COMUNA_ID" )
16 );
17
18 drop table if exists tipo cascade;
```



```
19 CREATE table if not exists tipo_medicion
20 (
21     id_tipo_medicion bigint ,
22     nombre varchar (100) not null ,
23     PRIMARY KEY (id_tipo)
24 );
25 drop table if exists tipo_interpolacion cascade;
26 CREATE table if not exists tipo_interpolacion
27 (
28     id_tipo_interpolacion bigint ,
29     nombre varchar (100) not null ,
30     PRIMARY KEY (id_tipo)
31 );
32
33 drop table if exists rast cascade;
34 CREATE table if not exists rast
35 (
36     id_raster SERIAL PRIMARY KEY,
37     rast raster
38 );
39 drop table if exists detalle_raster cascade;
40 CREATE table if not exists detalle_raster
41 (
42     id_detalle SERIAL PRIMARY KEY,
43     id_tipo_medicion bigint not null ,
44     id_tipo_interpolacion bigint not null ,
45     id_raster bigint not null ,
46     fecha_raster timestamp not null ,
47     FOREIGN KEY (id_tipo_medicion) REFERENCES tipo_medicion(id_tipo_
         medicion),
48     FOREIGN KEY (id_tipo_interpolacion) REFERENCES tipo_interpolacion(
         id_tipo_interpolacion),
49     FOREIGN KEY (id_raster) REFERENCES rast(id_raster)
50 );
51
52 drop table if exists registro cascade;
53 CREATE table if not exists registro
54 (
55     id_registro SERIAL PRIMARY KEY,,
56     id_estacion bigint not null ,
57     id_tipo bigint not null ,
```

```

58     fecha_registro timestamp not null ,
59     valor double precision not null ,
60     PRIMARY KEY (id_registro) ,
61     FOREIGN KEY (id_estacion) REFERENCES estacion(id_estacion) ,
62     FOREIGN KEY (id_tipo) REFERENCES tipo(id_tipo)
63 );

```

B.2. Vista Parcial

Para la implementación completa de la base de datos se necesitó de una vista parcial, esto es porque al generar una imagen raster a .SQL se crea el archivo con una tabla definida que para nuestro sistema es demasiado básica, por lo tanto se creó otra tabla con los detalles del raster y con la vista parcial logramos juntar en una sola tabla todo el raster.

```

1 CREATE VIEW vista_raster
2 AS SELECT rast.id_raster ,detalle_raster.id_detalle , rast.rast ,
3     detalle_raster.fecha_raster ,detalle_raster.id_tipo_interpolacion
4 FROM detalle_raster
5 INNER JOIN rast
6 ON detalle_raster.id_raster=rast.id_raster ;

```

B.3. Trigger

Para la implementación completa de la base de datos se necesitó de un Trigger. La finalidad de este Trigger es para cuando se inserta o actualiza un nuevo registro en una estación de medición, verifica si la fecha de este nuevo registro es mayor o menor a la que tiene, si fuera mayor reemplaza el atributo fecha de último registro de la estación. En el caso contrario no realiza ninguna acción.

```

1 CREATE OR REPLACE FUNCTION estacion_ultimoregistro(id_estacion_1 bigint
2     , fecha TIMESTAMP )
3 RETURNS boolean AS

```

```
3 $func$
4 BEGIN
5 IF (SELECT fecha_ultimo_registro FROM estacion WHERE estacion.id_
      estacion = id_estacion_1)
6     <
7     ( fecha)
8     THEN
9         RETURN TRUE;
10    END IF;
11    RETURN FALSE;
12 END
13 $func$ LANGUAGE plpgsql STABLE;
14
15 CREATE OR REPLACE FUNCTION check_estacion_ultimo_registro () RETURNS
      TRIGGER AS
16 $$
17 BEGIN
18     IF estacion_ultimoregistro(NEW.id_estacion , NEW.fecha_registro)
19         THEN
20             UPDATE estacion
21             SET fecha_ultimo_registro = NEW.fecha_registro
22             WHERE estacion.id_estacion = NEW.id_estacion;
23         RETURN NEW;
24     ELSE
25         RETURN NEW;
26     END IF;
27 END;
28 $$ LANGUAGE plpgsql;
29
30 CREATE TRIGGER check_ultimo_registro AFTER INSERT OR UPDATE
31     ON registro FOR EACH ROW
32     EXECUTE PROCEDURE check_estacion_ultimo_registro ();
```

C. Script en R

En el presente Anexo se encuentra el detalle de los Script para el programa R, existen dos, cada uno para un tipo de interpolación. Estos se ejecutan en el servidor mediante otro Script que utiliza los Script de las interpolaciones.

C.1. Interpolación IDW

```
1 # Librerias utilizadas.
2 library(gstat)
3 library(sp)
4 library(raster)
5 library(rgdal)
6 library(RODBC)
7 # Conexion a base de datos.
8 BD <- odbcConnect("proyecto")
9 ## Recoleccion de argumentos.
10 args <- commandArgs(TRUE)
11 # Recoleccion de datos.
12 datos=sqlQuery(BD, args)
13 # Transformacion de los datos.
14 coordinates(datos) <- c("lon", "lat")
15 proj4string(datos) <- CRS("+init=epsg:4326")
16 datos <- spTransform(datos, CRS("+init=epsg:32719"))
17 # Ajustando tamaño de celda.
18 grd=spsample(datos, type = "regular", cellsize = c(100,100))
19 gridded(grd)=TRUE
```

```

20 # Interpolacion IDW.
21 salida <-gstat::idw(valor~ 1, datos , newdata=grd , idp=2)
22 # Configuracion de proyeccion.
23 proj4string(salida) <- CRS("+init=epsg:32719")
24 r=raster(salida["var1.pred"])
25 # Creacion archivo raster .Tif
26 writeRaster(r,"archivo_idw.tif", "GTiff", overwrite=TRUE)
27 # salir de R.
28 q()

```

C.2. Interpolación Kriging

```

1 # Librerias utilizadas.
2 library(gstat)
3 library(sp)
4 library(raster)
5 library(rgdal)
6 library(RODBC)
7 # Conexion a base de datos.
8 BD <- odbcConnect("proyecto")
9 ### Recoleccion de argumentos.
10 args <- commandArgs(TRUE)
11 # Recoleccion de datos.
12 datos=sqlQuery(BD, args)
13 # Transformacion de los datos.
14 coordinates(datos) <- c("lon", "lat")
15 proj4string(datos) <- CRS("+init=epsg:4326")
16 datos <- spTransform(datos, CRS("+init=epsg:32719"))
17 # Creacion del variograma.
18 empvario <- variogram(valor~1,datos)
19 variofit = fit.variogram(empvario, vgm(440, "Exp", nugget=400, range
    =2300))
20 # Ajustando tamaño de celda.
21 grd=spsample(datos, type = "regular", cellsize = c(100,100))
22 gridded(grd)=TRUE
23 # Interpolacion kriging ordinario.
24 salida <- krige(valor~1, datos, grd, variofit)

```

```
25 proj4string(salida) <- CRS("+init=epsg:32719")
26 r=raster(salida["var1.pred"])
27 # creacion archivo raster .TIF .
28 writeRaster(r,"archivo_kriging.tif", "GTiff", overwrite=TRUE)
29 #salir de R.
30 q()
```

D. Encuesta de Usabilidad

En el presente Anexo se encuentra el detalle de las preguntas realizadas en la encuesta de usabilidad.

1. ¿Qué nivel de conocimientos de informática tiene usted?

- a) Nivel Basico.
- b) Nivel Usuario.
- c) Nivel Avanzado.
- d) Nivel Experto.

2. En la página principal del sitio observe la interfaz y responda: ¿Con la información que se ofrece en la pantalla, es posible saber de qué se trata el sitio web? Fácil

Difícil.....1.....2.....3.....4.....5.....6.....7.....Fácil

3. ¿Queda claro a partir del mapa el nivel de contaminación que existe en la ciudad?

Difícil de entender.....1.....2.....3.....4.....5.....6.....7.....Fácil de entender

4. La organización de la información en la pantalla cree que es:

Confusa.....1.....2.....3.....4.....5.....6.....7.....Muy Claro

5. Ahora inspeccione el sitio web y revise todas las categorías que le interesen. ¿En la portada y las demás páginas la forma en que se navega por el sitio se distingue fácilmente?

Difícil.....1.....2.....3.....4.....5.....6.....7.....Fácil

6. ¿Cómo encuentra que fue la navegación entre las distintas secciones del sitio web?

Demasiado lento.....1.....2.....3.....4.....5.....6.....7.....Muy rápido

7. La combinación de colores del sitio web le parece:

Muy mala.....1.....2.....3.....4.....5.....6.....7..... Excelente

8. Ahora debe realizar una tarea en particular: En historial de datos, se quiere que pueda ver los datos para la estación 5, luego graficar la medición de MP10 con fecha entre 20-05-2017 hasta el 22-05-2017. Además, descargar los datos en algún formato. ¿Qué tan fácil o difícil fue realizar la tarea antes mencionada?

Muy Fácil.....1.....2.....3.....4.....5.....6.....7.....Muy Difícil

9. Cómo encuentra que fue el gráfico para mostrar los resultados de los datos de la tarea anterior?

Difícil.....1.....2.....3.....4.....5.....6.....7..... Fácil

10. ¿Le parece bien todas las opciones de formatos disponibles para descargar los datos?

- a) Me parece bien.
- b) Faltan algunos.
- c) Muchos (sobran).

11. Ahora debe realizar otra tarea: En la pagina principal, en la zona de configuración del mapa, se quiere que pueda ver la contaminación de hace 3 horas en la ciudad y que esta sea del tipo MP2.5. ¿Qué tan fácil o difícil fue realizar la tarea antes mencionada?

Difícil.....1.....2.....3.....4.....5.....6.....7.....Fácil

12. ¿Cómo encuentra la sección derecha de configuración del mapa en la tarea anterior?

Muy Fácil.....1.....2.....3.....4.....5.....6.....7..... Muy Difícil

13. Respecto a la utilidad que ofrece el sitio web. ¿Qué es lo que más te llamó la atención positivamente o negativamente?

R/.....