

**MÉTODO DE LA INGENIERÍA**  
**PROYECTO DE CURSO**

Nicolás Penagos - Alejandro Parra - Christian Sanchez - Juan Pablo Herrera  
Marzo 2020

Universidad ICESI  
Ingeniería de Sistemas  
Proyecto Integrador I

## **Fase 1**

### **Identificación del problema**

#### **Descripción del contexto problemático (Causas y síntomas) :**

Asociación Ambiente y Sociedad (AAS) es una organización no gubernamental que se encarga de generar cambios positivos en las regulaciones, políticas y la toma de ausencia de regulaciones sobre la fabricación y consumo de vehículos a combustión. Para promover regulaciones más estrictas sobre los vehículos a combustión, AAS decidió crear una campaña de concientización sobre la contaminación del aire en Colombia.

Como parte de su investigación para la campaña, AAS ha obtenido una base de datos pública con varios datos sobre la calidad del aire en diferentes departamentos y municipios en Colombia. Debido a la gran cantidad de datos almacenados en la base de datos los analistas de la organización han sido incapaces de poder procesar y analizar adecuadamente la información almacenada. Es de gran importancia lograr un manejo adecuado de los datos para la organización debido a que se desea mostrar información concreta y relevante para su campaña.

Con lo planteado anteriormente AAS ha solicitado la creación de una aplicación que le permite recoger datos específicos a partir de ciertos criterios, y mostrarlos para poder ser analizados por el usuario.

#### **Identificación de necesidades:**

Las siguientes fueron las necesidades a solucionar identificadas:

- Se debe buscar datos por criterios especificados en la base de datos
- Se requiere traer los datos desde la base de datos al usuario
- Se debe mostrar los datos al usuario de manera ordenada
- Se debe demostrar los datos de una manera visual e interactiva
- Se debe permitir al usuario navegar a través de los datos por medio de una línea de tiempo.

### **Definición del problema:**

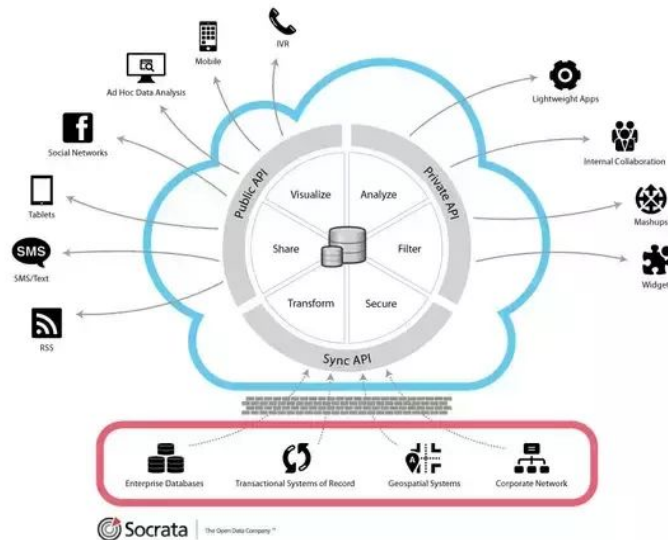
Actualmente la AAS cuenta con una extensa base de datos que le suministra información detallada sobre la calidad del aire en Colombia. Sin embargo, la organización no cuenta con una herramienta que permita analizar y visualizar dicha información para obtener una correcta interpretación y así establecer las regulaciones, políticas y decisiones necesarias en torno a las prácticas medioambientales se deben seguir.

## **Fase 2**

### **Recopilacion de informacion necesaria**

#### **I. Socrata open data API (SODA):**

Socotra es una compañía privada que trabaja junto a gobiernos, NGOs y organizaciones sin ánimo de lucro con el fin de hospedar cientos de catálogos de datos abiertos para estas entidades. Socrata implementa SODA en cada uno de los catálogos con el fin de permitir a cualquier usuario un mejor acceso a los datos.



Fuente: <https://dev.socrata.com>

## II. JSON:

El formato JSON (JavaScript Object Notation) se utiliza para el intercambio liviano de datos. el formato permite facilidad al humano para escribirlo y leerlo, y a la máquina para analizar y generar.

JSON puede ser construido de dos formas:

- Una colección de pares nombre/valor que al analizar se toma como objetos.
- una lista ordenada de valores que es tomada como un arreglo, vector o lista.

```
{
  "Title": "The Cuckoo's Calling",
  "Author": "Robert Galbraith",
  "Genre": "classic crime novel",
  "Detail": {
    "Publisher": "Little Brown",
    "Publication Year": 2013,
    "ISBN-13": 9781408704004,
    "Language": "English",
    "Pages": 494
  },
  "Price": [
    {
      "type": "Hardcover",
      "price": 16.65,
    },
    {
      "type": "Kindle Edition",
      "price": 7.03,
    }
  ]
}
```

Diagram illustrating the structure of a JSON object with annotations:

- Object Starts**: Points to the opening curly brace {.
- Object Starts**: Points to the opening curly brace of the "Detail" object {.
- Value string**: Points to the string value "Little Brown".
- Value number**: Points to the numeric value 2013.
- Object ends**: Points to the closing curly brace of the "Detail" object }.
- Array starts**: Points to the opening square bracket [ of the "Price" array.
- Object Starts**: Points to the opening curly brace of the first object in the "Price" array {.
- Object ends**: Points to the closing curly brace of the first object in the "Price" array }.
- Object Starts**: Points to the opening curly brace of the second object in the "Price" array {.
- Object ends**: Points to the closing curly brace of the second object in the "Price" array }.
- Array ends**: Points to the closing square bracket ] of the "Price" array.
- Object ends**: Points to the closing curly brace } of the main JSON object.

Fuente: <https://www.json.org/json-en.html>

## III. Newtonsoft - Json.NET:

Json.NET es un framework de alto rendimiento que tiene como fin facilitar el manejo de archivos JSON. Este framework permite convertir la convertir entre objetos .NET y JSON

<https://www.newtonsoft.com/json>

#### IV. SODA.NET:

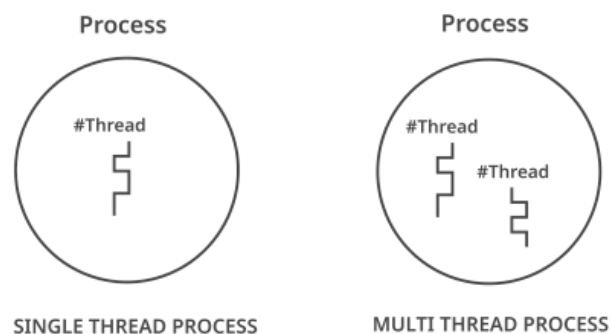
SODA.NET es una libreria que permite el acceso a la API para proyectos .NET (4.5 para arriba) y que se puede encontrar como un paquete NuGet

<https://github.com/CityofSantaMonica/SODA.NET>

#### V. Concurrencia

En términos simples, concurrencia significa hacer más de una cosa al mismo tiempo. En informática, la concurrencia es utilizada cada vez que se requiere que una aplicación realice una tarea mientras se está ejecutando alguna otra en paralelo.

Una de las formas más comunes de concurrencia es el “Multithreading” o Multihilos, esto se refiere al uso de múltiples hilos en la ejecución de un proceso. El uso de múltiples hilos permite que se lleve a cabo un procesamiento en paralelo dividiendo grandes cantidades de trabajo que es ejecutado de manera concurrente. El procesamiento en paralelo maximiza el uso de los múltiples cores de un procesador. Teniendo en cuenta que la mayoría de los CPUs (Unidad Central de Procesamiento) modernos tienen más de un core, el procesamiento en paralelo permite que todo el trabajo se divida en partes masomenos equivalentes entre los diferentes cores del procesador. Con esto se mejora el tiempo de ejecución de una tarea y no se deja ningún core sin trabajar.



*Figura 1: Simple ilustración de procesamiento multihilo.*

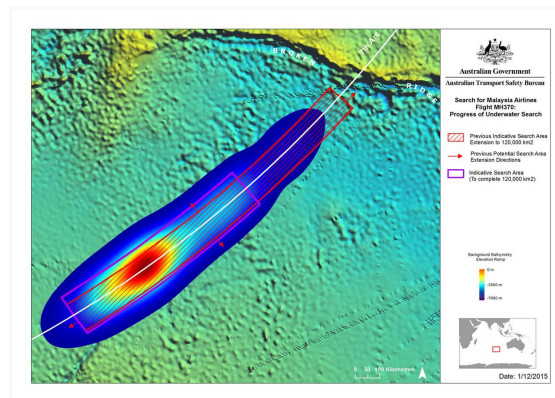
*Fuente:* <http://www.pcnltx.tech/what-is-multithreading/>

## VI. Mapa de Calor

Un mapa de calor es una representación gráfica donde valores individuales que están contenidos en una matriz son representados por colores.

Hay diferentes tipos de mapas de calor, entre los más comunes se encuentran:

- Los mapas de calor de biología se usan típicamente en biología molecular para representar el nivel de expresión de muchos genes en varias muestras comparables (por ejemplo, células en diferentes estados, muestras de diferentes pacientes) a medida que se obtienen de microarrays de ADN.
- El mapa de árbol es una partición jerárquica 2D de datos que se asemeja visualmente a un mapa de calor.
- Un diagrama de mosaico es un mapa de calor en mosaico para representar una tabla de datos bidireccional o superior. Al igual que con los mapas de árbol, las regiones rectangulares en un diagrama de mosaico están organizadas jerárquicamente. Esto significa que las regiones son rectángulos en lugar de cuadrados.
- Una visualización de función de densidad es un mapa de calor para representar la densidad de puntos en un mapa. Le permite a uno percibir la densidad de los puntos independientemente del factor de zoom.



*Un mapa de calor, encima de un mapa batimétrico en color, que indica la ubicación probable del vuelo 370 de Malaysia Airlines perdido, basado en un análisis del método bayesiano de posibles rutas de vuelo de la aeronave.*

*Fuente:*

[https://en.wikipedia.org/wiki/Heat\\_map#/media/File:MH370\\_location\\_probability\\_heat\\_map\\_per\\_DST\\_Group\\_analysis.jpg](https://en.wikipedia.org/wiki/Heat_map#/media/File:MH370_location_probability_heat_map_per_DST_Group_analysis.jpg)

## **Estado del Arte :**

Diversas soluciones se han dado antes para este tipo de problemas. Las siguientes son algunas de estas:

- **World Bank Group Finances App**

Esta aplicación creada por el banco mundial y utilizando el API de Socrata, es capaz de recoger una gran cantidad de datos sobre las economías a nivel mundial y permite ordenar y categorizar los datos.

- **Strava**

Es una aplicación que muestra un mapa de calor a nivel mundial, además de manejar una base de datos de temperaturas a nivel mundial.

- **Weather Forecast.com**

Este sitio web contiene una base de datos climatológica a nivel mundial que permite procesar y analizar patrones climáticos, para producir predicciones sobre el clima y mostrarlo en un mapa.

### **Fase 3**

#### **Búsqueda de soluciones creativas**

Se utilizó la técnica de lluvia de ideas para generar propuestas para resolver el problema definido anteriormente, estas ideas se dividieron en dos categorías:

##### **Ideas para cargar la base de datos al programa de computador:**

1. Implementar SODA en el catálogo de la base de datos para que así sea posible acceder únicamente a ciertas porciones de la base de datos por medio de consultas o queries evitando que el programa colapse.
2. Exportar la base de datos en un archivo .json y utilizar los respectivos métodos de serialización y deserialización para así facilitar el intercambio de datos entre el dataset y el aplicativo.
3. Utilizar la concurrencia y el procesamiento en paralelo para cargar la base de datos utilizados múltiples hilos en donde cada hilo cargará un rango de datos.

##### **Ideas para la creación de la aplicación:**

1. Crear una aplicación que utilice mapas un mapa geográfico en donde se pueda navegar por los datos a través de las diferentes ubicaciones en el mapa.
2. Crear una línea de tiempo en donde se pueda avanzar y retroceder por los datos a través de los diferentes años en los cuales recopilamos los datos.
3. Utilizar un mapa de calor para representar los datos de la calidad del aire para hacerlo de una manera fácil e interactiva para el usuario.
4. Implementar diferentes filtros para que usuario pueda realizar búsquedas personalizadas sobre la base de datos.
5. Implementar paginación para que el usuario pueda visualizar la base de datos por partes en lugar de tener los 15 millones de datos en un mismo lugar.



## **Fase 4**

### **Transformación de las ideas a los diseños preliminares**

#### **Alternativa 1 : Implementar SODA**

El API de Soda permite el acceso a contenidos de bases de datos externas. Al manejar una base de datos de manera remota esta tiene varios beneficios como: no consume mucho espacio dentro del computador de los usuarios. Además el API de SODA permite hacer filtros para la información con el fin de solamente mostrar la información que cumpla con los parámetros de entrada que de el usuario. La limitación más grande que posee esta alternativa sería la necesidad de una red de internet para conectarse a la base de datos y ejecutar todas las operaciones.

#### **Alternativa 2 : Exporta la Base de datos como .json**

Al exportar la base de datos como una archivo .json, nos permite una lectura y filtración de la información sin la necesidad de utilizar una red internet. Además de poder utilizar los criterios de filtración para obtener la información que desea analizar el usuario. Pero esta tiene la restricción de tener que guardar en extenso archivo en el computador del usuario.

#### **Alternativa 3 : Concurrencia**

Al programar una aplicación utilizando vario hilos para cargar diferente partes de la base de datos, tendremos que explorar a más fondo el manejo de hilos dentro del programa. Además debido a la naturaleza de manejar varios hilos este requiere más memoria para el procesamiento de todos los hilos. Debido a la gran cantidad de datos y la cantidad de hilos necesarios para procesarlos esta alternativa se descarta.

#### **Alternativa 4 : Mapa Geográfico**

Para implementar esta alternativa se necesitaría la ayuda de una librería externa que permite el manejo de mapas para una aplicación. Además deja abierta el manejo y lectura de la base de datos. Debido a la falta de especificación con respecto al manejo de datos y el tiempo adicional que se requiere para encontrar y manejar adecuadamente una librería de mapas, esta alternativa se descarta.

#### **Alternativa 5 : Línea de Tiempo**

Esta alternativa consiste en poder crear una línea de tiempo con todos los datos y mostrarla al usuario. Esto le facilita el análisis de los datos y muestra de manera continua el cambio de los datos mediante que pasa el tiempo. La dificultad de esta alternativa es el manejo de las grandes cantidades de datos, debido a la gran magnitud de información en la base de datos, es muy in conveniente leer todo el archivo y luego ubicarlo en una línea de tiempo. Por estas razones esta alternativa queda descartada.

#### **Alternativa 6 : Mapa de Calor**

Esta alternativa parecida a la alternativa 4 también requiere del uso de una librería externa que permita el uso de mapas para la aplicación. La diferencia más notoria sería que : se requiere una librería que permita la creación de mapas de calor utilizando datos externos. Que en este caso serían los ofrecidos en la base de datos. La dificultades de esta alternativa se encuentran cuando se toma en cuenta la investigación y la práctica necesaria para crear un mapa de calor lo suficientemente adecuado para el análisis de parte del usuario. Por esta razón se descarta esta alternativa.

### **Alternativa 7: Implementar filtros**

La implementación de filtros para la búsqueda de la información, consistiría en tener que investigar todas las categorías de información que ofrece la base de datos con el fin de poder darle la opción al usuario de poder clasificar y ver la información en específico que necesite el usuario. Además el manejo de estos filtros puede ser implementado de varias maneras, o implementando el API de Soda o manejando los datos directamente desde el computador del usuario.

### **Alternativa 8: Paginación de Datos**

Esta alternativa consiste en limitar la cantidad de datos que el usuario puede ver, parecido a las técnicas usadas en programas para leer libros como kindle, o pdf viewer. Este permite cargar la base de datos por partes y no toda de una lo que permite un mejor manejo de datos y no ocupa tanta memoria en el computador del usuario.

## **Fase 5**

### **Evaluación y selección de la mejor solución**

A continuación se definirán los siguientes criterios de evaluación relevantes para la elección de las soluciones que se implementaran en el proyecto final.

- **Facilidad de implementación:** Este aspecto será calificado de 1 a 5 en donde una calificación baja se le asignará una solución con una elevada dificultad de implementar y cuyos conocimientos necesarios de desarrollo se encuentren por fuera del entendimiento y habilidades del equipo de trabajo. Por otro lado, una calificación alta se le otorgará a una solución que se encuentre dentro de los temas de dominio y habilidades de desarrollo del equipo de trabajo.
- **Relevancia para la solución del problema:** Este criterio se refiere a que tanto aporta la propuesta de solución al cumplimiento de los requerimientos funcionales del problema. Se evaluará con una calificación de 1 a 5 en donde una calificación baja significa que la solución aporta poco o nada en la consecución de algún requerimiento funcional. Por otro lado, se le asignará una calificación alta a las propuestas que aporten significativamente en el cumplimiento de alguno o varios requerimientos funcionales.
- **Eficiencia de solución:** Este criterio trata acerca de la complejidad de ejecución de la propuesta con el manejo de grandes volúmenes de datos, en donde una alta complejidad implicaría grandes tiempos de ejecución que pueden causar problemas en la estabilidad del programa. Se le dará una calificación baja a soluciones que cuenten

con una gran complejidad temporal y se la asignará una calificación alta a soluciones eficientes con una baja complejidad temporal.

Evaluación de las soluciones:

<b>PROPUESTA</b>	<b>FACILIDAD DE IMPLEMENTACIÓN</b>	<b>RELEVANCIA PARA LA SOLUCIÓN</b>	<b>EFICIENCIA DE LA SOLUCIÓN</b>	<b>TOTAL</b>
Implementar Soda	4	5	5	<b>14</b>
Exportar base de datos en archivo .Json	4	4	3	<b>11</b>
Concurrencia	5	2	2	<b>9</b>
Mapa geográfico	4	5	4	<b>13</b>
Línea de tiempo	3	4	3	<b>10</b>
Mapa de calor	2	5	3	<b>10</b>
Implementar filtros	5	4	4	<b>13</b>
Paginación de datos	5	2	2	<b>9</b>

**Tabla 1:** Evaluación de alternativas propuestas.

### **Métodos de análisis de datos**

De acuerdo con la tabla 1, las alternativas con calificaciones más bajas o un puntaje menor 10 serán descartadas. Por lo tanto, se decide no implementar la paginación de datos ni las implementaciones concurrentes por considerarse que su aporte es suficiente para que valga la pena la inversión de tiempo y esfuerzo que conlleva su implementación. Por otro lado se decide implementar el resto de alternativas las cuales son implementar la tecnología soda,

exportar los datos de la base de datos en archivos en formato .json, presentar los datos a través de un mapa geográfico y un mapa de calor y por último implementar un sistema de búsqueda por filtros.

## Referencias

- Stephen, C. Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programing (2019). Págs 1 - 3.
- Heat Map. [https://en.wikipedia.org/wiki/Heat\\_map](https://en.wikipedia.org/wiki/Heat_map)
- SODA.NET: <https://github.com/CityofSantaMonica/SODA.NET>
- JSON: <https://www.json.org/json-es.html>