



Santiago Rodas Rodriguez Juan Fernando Martinez	Método de la ingeniería
--	--------------------------------

Fase 1: Identificación del problema

Los Estados Unidos de América es el país con mayor fuerza militar en todo el planeta. Sus diferentes divisiones como la aérea, naval, armada, entre otras, hacen que tenga las últimas tecnologías balísticas y de rastreo satelital. Estos soldados están preparados para manejar diferentes tipos de armas y escabullirse por las selvas africanas o aguantar los calores del Sahara. Asimismo, aprenden a manejar diferentes técnicas de inteligencia como el código Morse, código binario, e inclusive el alfabeto radiofónico.

Sin embargo, hay momentos donde el peligro se encuentra en la línea enemiga, y ellos deben de retroceder a sus bases militares. Pero, durante ese movimiento de despliegue contrario, hay soldados que quedan heridos, o simplemente se esconden para no ser capturados. Es por esa razón, que se necesita de un software que rastree la longitud y latitud exacta, junto a la hora precisa en que el soldado hizo un llamado para ser rescatado.

Tomando en cuenta esta información introductoria, se tiene los siguientes requerimientos funcionales:

1. Codificar el software en el lenguaje de programación C#.
2. Leer y cargar un dataset que ellos brindan al momento de utilizar el programa.
3. Reportar los datos en una tabla y filtrarlos con alguna condición por medio de un ComboBox.
4. Clasificar los tipos de campos como: categórico, cadena o numérico.
5. Marcar o fijar en el mapa las ubicaciones correspondientes de los soldados, logrando así filtrar datos si el usuario lo requiere.
6. Dibujar un polígono sobre el mapa para realizar una búsqueda limitada en una área específica del planeta.
7. Graficar diferentes opciones estadísticas que brinda el sistema para la correcta compilación de datos cuánticos.



Fase 2: Recolección de la información necesaria

Para solucionar este problema y realizar un correcto software, es indispensable conocer algunas herramientas importantes que permitan cumplir con cada uno de los requerimientos funcionales anteriormente mencionados. Algunas son:

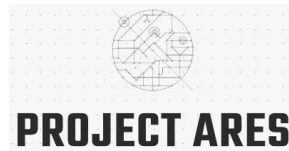
GMap: Es una librería que se basa principalmente en Google Maps y que permite tener mapas con marcadores, calcular rutas rápidas, crear perfiles longitudinales, etc. La documentación es un poco deficiente, pero se tiene un buen catálogo de funcionalidades que reducen el código, y lo hacen más entendible.

Gráficos: Una parte indispensable del programa es generar gráficos estadísticos que permitan entender el problema de manera significativa. Por ende, es de gran importancia reconocer cuales pueden ser algunos tipos de gráficos que podemos utilizar, y cuales son de gran necesidad para el problema planteado.

Lector de archivos: Para utilizar el sistema de forma eficaz, es necesario leer rápidamente un dataset con toda la información existente hasta el momento. De esta manera no solo se evita perder datos significativos, sino que también permite filtrar casillas por cualquier tipo de limitación que el mismo usuario interponga.

Además de esas funcionalidades, se complementa la información con los requerimientos no funcionales encontrados en el sistema.

1. Medir el tiempo de trabajo por medio del programa Process Dashboard para calcular el pago final de ambos estudiantes.
2. Depositar los códigos fuentes, y cualquier otro cambio en el sistema en la plataforma GitHub.
3. Grabar el uso Beta del sistema para mostrar su funcionamiento a las cadenas de mando militares desplegadas por todo el mundo.



Fase 3: Búsqueda de soluciones creativas

Para llevar a cabo esta etapa, se toman en cuenta 3 etapas significativas:

- Analizar los requerimientos indicados para el proyecto, diseñando un problema cuya solución pueda plantearse en términos de lo solicitado en el enunciado.
- Dialogar de manera significativa entre los integrantes del grupo de trabajo para determinar el tema principal del proyecto, claramente modelado como un software futuro.
- Definir los atributos y exigencias propias del problema elegido para iniciar el desarrollo principal.

Para tomar en cuenta las etapas anteriormente mencionadas, se utiliza la herramienta [Brainwriting](#) como método principal al momento de generar ideas creativas:

1. Realizar una interfaz gráfica cómoda con el usuario, permitiendo así no solo tener una correcta visualización del programa, sino que también sea eficaz al momento de utilizar el software. Ganando tiempo, y sobre todo, salvando vidas.
Comentario: No es específico en cuanto al diseño del programa o el backend.
2. Implementar un mapa que sea agradable a la vista del usuario, y que al mismo tiempo combine con las tonalidades de color vistas en el modelo base del programa.
Comentario: Al igual que el punto anterior no es específico
3. Dibujar un polígono de 4 lados iguales desde una coordenada inicial que digite el usuario. Claro está, con la idea de limitar una área de búsqueda específica del programa. Se dibuja la figura en el mapa de manera automática por cada marcador existente, tomando como centro la ubicación de los soldados. Se toma una distancia a los lados de la figura de 500 m (poner diferencia en latitud y longitud) y aplicando una fórmula matemática se hallan las coordenadas de cada vértice.
Comentario: Es una buena idea ya que evita poner campos adicionales en la interfaz gráfica.
4. Desarrollar una ventana emergente que permita seleccionar un tipo de gráfico para mostrar automáticamente en esa misma. Además, es importante destacar que tanto la tabla como el mapa serán visualizadas en un solo formulario, facilitando el filtrado y acelerando algunos componentes del sistema.
Comentario: Buena idea, ya que está tomando en cuenta la eficacia del usuario.
5. Almacenar las capas en una lista. Es decir, cada capa, desde los puntos hasta los polígonos, será guardada en el sistema para tener un registro de cada movimiento realizado.
Comentario: Se puede realizar, pero realmente no lo veo necesario.



6. Guardar la información obtenida del dataset en un datatable para evitar la creación de un objeto adicional, ya que este es necesario para la visualización de los datos.

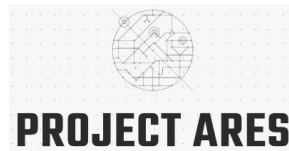
Comentario: ¡Muy buena idea, felicitaciones!

7. En el área del mapa, sería una buena estrategia dibujar un polígono directamente con el mouse, facilitando el proceso y volviéndolo más eficaz.

Comentario: Genial, es una excelente idea.

8. Separar en pestañas la tabla, los gráficos y el mapa.

Comentario: Puede ser una alternativa

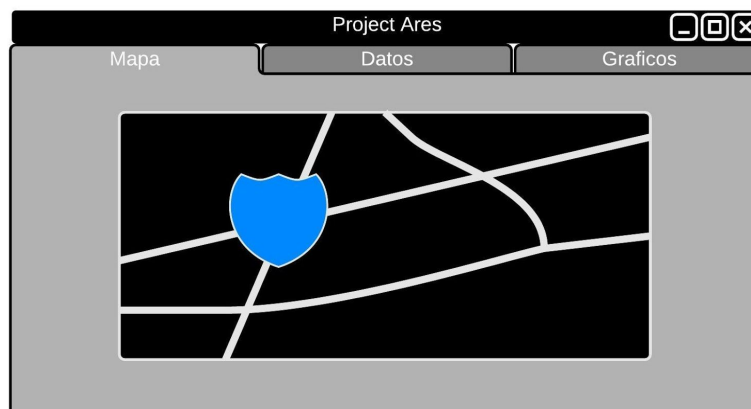


Fase 4: Pasar de la idea principal al diseño preliminar

- Se descarta la idea de guardar las capas en una lista, primero porque no es estrictamente necesario, y segundo porque solo se requieren 2: una para los puntos y otra para los polígonos.
- También se tiene en cuenta cada idea de la interfaz gráfica, pero no son útiles al momento de pensarla o crearla, ya que en pocas palabras solo debemos diseñar algo útil y claro al mismo tiempo.
- Ambas ideas en relación a los polígonos deberán de ser evaluadas en la siguiente fase, debido a que tienen una utilidad elevada. Lo mismo ocurre con el dataset, y en cierta parte con la ui diseñada.

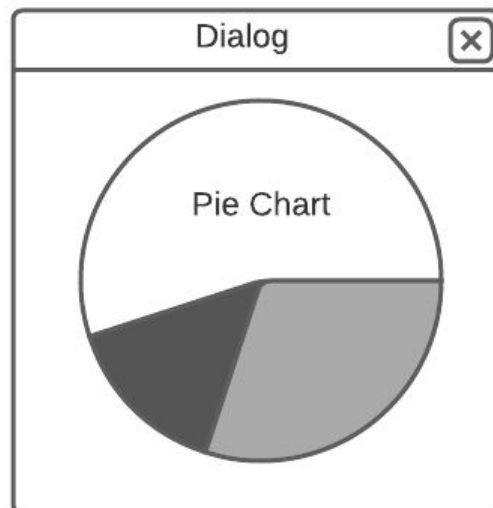
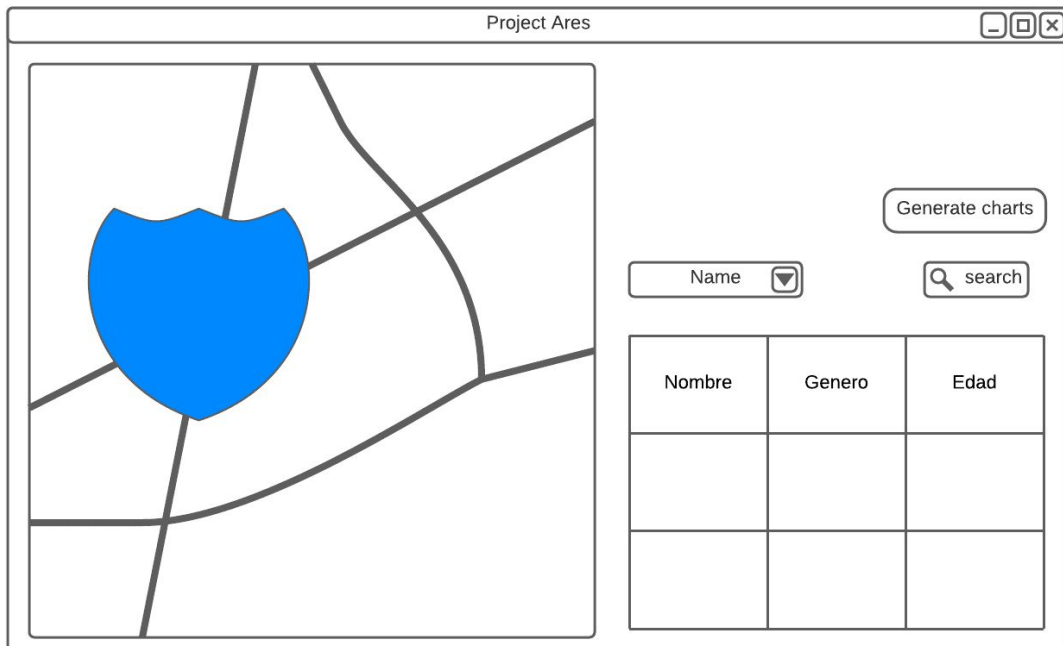
Modelo 1: Basado en la idea de pestañas.

S	M	T	W	Th	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			





Modelo 2: Basado en la idea de una sola ventana principal



La primera decisión importante que se toma, es el uso de un dataset militar con un total de 1000 datos utilizables en el sistema. Con las funciones preliminares de Visual Studio, se espera leer los datos de manera progresiva para ser mostrados en la tabla circundante, y filtrarlos por medio de alguna acción especial.



Además de eso, también se toma la decisión de implementar todo el código tomando en cuenta la orientación a objetos, ya que esta nos brinda grandes beneficios al momento de utilizar la información y/o tener un código mucho más estructurado.

Por último, se toma en cuenta el [proyecto personal anteriormente creado](#) para tomar algunas funcionalidades básicas del sistema, como la lectura, filtración y muestra de los datos existentes en un dataset. Con esto, no solo se espera reducir los tiempos de trabajo, sino también mejorar las habilidades computacionales y algorítmicas en cada uno de los integrantes del grupo de trabajo.

Statistical reports

Select the file to load the data.

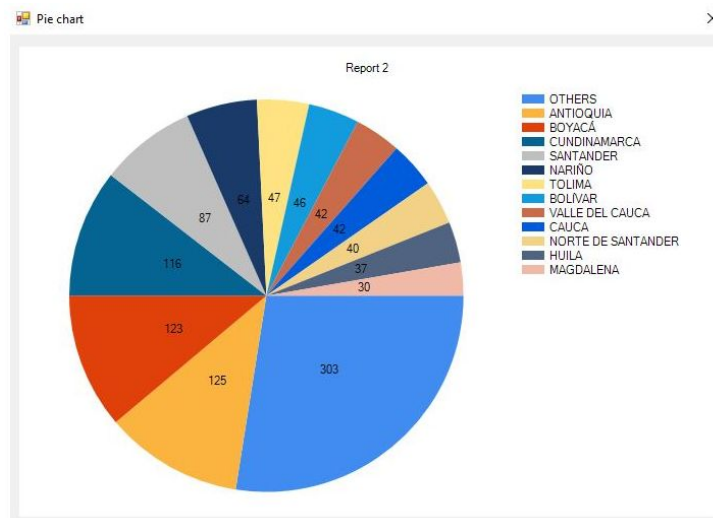
Search

Filter data

Pie chart

C:\Users\Santiago\Desktop\2021 - T1\1. Proyecto integrador\1. Tareas\Tarea 1\statistical-reports\data\data_data.csv

Código Departamento	Código Municipio	Nombre Departamento	Nombre Municipio	Tipo: Municipio / Isla / Área no municipalizada
05	05001	ANTIOQUIA	MEDELLÍN	Municipio
05	05002	ANTIOQUIA	ABEJORRAL	Municipio
05	05004	ANTIOQUIA	ABRUJUIT	Municipio
05	05021	ANTIOQUIA	ALEJANDRIA	Municipio
05	05030	ANTIOQUIA	AMAGÁ	Municipio
05	05031	ANTIOQUIA	AMALFI	Municipio
05	05034	ANTIOQUIA	ANDES	Municipio
05	05036	ANTIOQUIA	ANGELÓPOLIS	Municipio
05	05038	ANTIOQUIA	ANGOSTURA	Municipio
05	05040	ANTIOQUIA	ANORÍ	Municipio
05	05042	ANTIOQUIA	SANTA FE DE ANTIOQUIA	Municipio
05	05044	ANTIOQUIA	ANZÁ	Municipio
05	05045	ANTIOQUIA	APARTADÓ	Municipio
05	05051	ANTIOQUIA	APBULETES	Municipio
05	05055	ANTIOQUIA	ARBOLEA	Municipio
05	05059	ANTIOQUIA	ARMENIA	Municipio
05	05079	ANTIOQUIA	BARBOSA	Municipio
05	05086	ANTIOQUIA	BELMIRA	Municipio
05	05088	ANTIOQUIA	BELLO	Municipio
05	05091	ANTIOQUIA	BETANIA	Municipio
05	05093	ANTIOQUIA	BETULIA	Municipio





Fase 5: Evaluación y selección de la solución

- Para mayor facilidad del usuario, se decide tomar la idea de manejar toda la información relacionada del sistema en una sola ventana principal. Es decir, la interfaz grafica sera basada en el modelo 2 anteriormente presentado: mostrando el mapa en el lado izquierdo, junto a la tabla y otras opciones en el lado derecho. Además, al momento de generar los gráficos estadísticos se visualizarán desde una ventana externa.
- Para guardar y almacenar la información se decide utilizar un arreglo de listas, ya que sería mucho más eficiente al momento de filtrar los datos, ahorrando tiempos de sistema y mermando la complejidad temporal del software. Asimismo, es más competente, ya que obviamente se requiere modificar el datatable cuando se realice cualquier tipo de acción.

Por último, se decide utilizar las siguientes herramientas:

Visual Studio: Es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador la creación de software, permitiéndonos desarrollar aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET, algunos lenguajes que podemos encontrar son: Visual Basic, Visual C# y Visual C + +.

GitHub: Es un sitio web y un servicio en la nube que ayuda a los desarrolladores a almacenar y administrar su código, al igual que llevar un registro y control de cualquier cambio sobre este código.

LucidChart: Es una herramienta de diagramación basada en la web, que permite a los usuarios colaborar y trabajar juntos en tiempo real, creando diagramas de flujo, organigramas, esquemas de sitios web, diseños UML, mapas mentales, prototipos de software y muchos otros tipos de diagrama.

Google-Drive: Permite crear carpetas para almacenar y subir archivos de cualquier tipo. Además, logra producir y modificar documentos en línea, manejando diferentes formatos de procesador de textos, planillas de cálculo, editor de diapositivas, etc. Por último, también permite elaborar formularios para encuestas o exámenes.



Fase 6: Preparación de reportes, planos y especificaciones

- [Diagrama de clases.](#)
- [Diagrama de objetos.](#)

Fase 7: Implementación del diseño

- [Repositorio en GitHub.](#)