Universidad Autónoma Metropolitana – Azcapotzalco

Licenciatura en Ingeniería en Computación

Reporte de Proyecto Tecnológico

# Sistema web para gestionar incidentes delictivos

Presenta:
Raúl Alberto Ruvalcaba Flores
2133000167

Asesores:
José Alejandro Reyes Ortiz
Doctor en Ciencias de la Computación
Departamento de Sistemas

Angeles Belém Priego Sánchez
Doctora en Ciencias del Lenguaje
Departamento de Sistemas

Trimestre 2017 – Primavera

21 de Julio del 2017

# Declaratorias

Yo, José Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

José Alejandro Reyes Ortiz

Yo, Ángeles Belém Priego Sánchez, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Angeles Belém Priego Sánchez

Yo, Raúl Alberto Ruvalcaba Flores, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Raúl Alberto Ruvalcaba Flores

# Resumen

El presente reporte tiene como objetivo principal presentar el desarrollo de una aplicación web para la gestión de incidentes delictivos con publicaciones de usuarios autentificados por la misma aplicación, estos incidentes delictivos son organizados en cinco categorías:

- Homicidio
- Suicidio
- Robo o Asalto
- Violación
- Explotación sexual

A esta aplicación web se le dio el nombre de ReportIt. ReportIt sirve para que la sociedad tenga un sitio que los apoye a geolocalizar los delitos; se pretende que el usuario (ciudadano), se registre y comience a realizar publicaciones con ayuda de la geolocalización sobre los incidentes delictivos de los cuales ha sido víctima o testigo. De igual forma ReportIt permite, en caso de no tener cuenta en el sistema, usuario no autentificado, poder realizar consultas de incidentes delictivos y estar informado por propia seguridad y la de los seres queridos.

El desarrollo de aplicaciones web está muy avanzado y existen diferentes lenguajes de programación y framework's de trabajo que facilitan el desarrollo de la aplicación a la hora de codificar las aplicaciones. En nuestro caso, ReportIt fue desarrollado con:

- ASP.NET MVC framework, que, si bien ya lo menciona, este es un framework de aplicaciones web que utiliza la arquitectura modelo-vista-controlador (MVC), de esta forma el desarrollador consigue ver que las aplicaciones web están compuestas de tres funciones: El modelo, la vista y el controlador.

Al tratarse de un framework muy utilizado para el desarrollo web, existe mucha información en la web que facilita la creación de tu aplicación. Existen desventajas a la hora de utilizar este framework, esto debido a que ASP .NET no realiza el mapeo de tu base de datos como bien lo realiza Hibernate[1], pero de igual forma es una herramienta que se adaptó y que facilitó el desarrollo de ReportIt.

La IDE[2] utilizada para el desarrollo de ReportIt fue Visual Studio 2015 la cual cuenta con grandes ventajas por la rapidez con la que se puede desarrollar software y por la diversidad de servicios que se proporcionan en diferentes lenguajes como lo es el depurador de aplicaciones, que en nuestro caso fue crucial utilizarlo.

---

[1] Hibernate: Es una herramienta de mapeo objeto-relacional para la plataforma Java y .NET, facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación esto mediante archivos declarativos (XML).

[2] IDE: Entorno de desarrollo integrado, es un entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

En este reporte se detallará la metodología utilizada para el desarrollo de ReportIt la cual es una metodología con la que se busca estructurar en el desarrollo del proyecto. La metodología a grandes rasgos utilizada para el desarrollo de ReportIt fue la siguiente: Generar el modelo de clases, con este generar el modelo entidad – relación de la base de datos, codificar los objetos en C#, conectar la base de datos y de ahí en adelante las funciones fueron de vista controlador. En los capítulos que se verán más adelante se detallará la metodología propuesta.

Para el desarrollo de este proyecto se usaron cookies para un manejador de sesiones, se utilizaron envíos de email para bienvenida, y olvido de contraseña.

Resultados Obtenidos:

ReportIt: una aplicación para gestión de incidentes delictivos.

1. Consultas de incidentes delictivos:
   Usuarios no autentificados en ReportIt son usuarios que tienen la posibilidad de realizar consultas a las publicaciones de los incidentes delictivos y visualizar éstos en un mapa. De igual forma, tiene la facilidad de entrar a ver detalles de cada una de estas publicaciones y visualizar al autor de dichas publicaciones.

   Para los usuarios que si están registrados en ReportIt cuentan con el beneficio del uso de búsquedas avanzadas con geolocalización, mostrando todas las publicaciones de incidentes delictivos, sucedidos en un radio de 40 km desde su punto de ubicación.

2. Gestión de un perfil:
   Este servicio es únicamente para usuarios registrados ya que ellos tienen la posibilidad de acceder a sus perfiles, modificar su información, fotos de perfil, ubicación y nombre, o eliminar su cuenta de ReportIt.

3. Gestión de incidentes delictivos:
   Se trata de un servicio exclusivamente para usuarios registrados, ya que solo ellos pueden realizar publicaciones de incidentes delictivos y tener todo el manejo de estas publicaciones. El manejo de las publicaciones incluye: modificar la ubicación, la fecha del incidente, la descripción del incidente y/o el tipo de incidente suscitado.

# Indice

# Indice de figuras

# CAPÍTULO 1. Acercamiento teórico

## 1.1 Introducción

En la ciudad de México, a diario hay incidentes violentos, tales como: robos, asaltos, asesinatos, entre otros, y conforme han pasado los años éstos han ido incrementando. Si tan solo se compara la tasa nacional, por cada 100 mil habitantes de víctimas de homicidio doloso[3], de enero del 2017 contra el promedio del año 2016, se vio un aumento significativo del 11.46% [1]. Estas estadísticas nos llevan a preocuparnos por la inseguridad que existe en México.

En este reporte se desarrolla una aplicación web denominada ReportIt para gestionar incidentes delictivos[4] en México. Esto es con el fin de contribuir con la sociedad para una mejor seguridad.

Por tal motivo, esta aplicación se realizó con el objetivo de ayudar a los habitantes a realizar, principalmente, publicaciones y consultas de incidentes delictivos geolocalizados en México. Estos incidentes incluyen los siguientes tipos:

- Homicidio
- Secuestro
- Robo o asalto
- Violación
- Explotación sexual

Los incidentes, anteriormente mencionados, podrán ser gestionados. Es decir, son consultados por cualquier ciudadano y publicados, eliminados o editados, solo por aquellos usuarios autentificados en la aplicación. Esta gestión de los incidentes es únicamente con el propósito de informar a la sociedad sobre lo que ocurre diariamente en la ciudad.

[3] Se denomina homicidio doloso a un subtipo del delito consistente en matar a alguien sin que concurran las circunstancias de alevosía, precio o ensañamiento [2].

[4] Por gestión de incidentes delictivos se entenderá la publicación, eliminación y/o modificación de incidentes delictivos, además de la búsqueda de estos mismos por medio de los parámetros de los incidentes (fecha, ubicación, palabra clave y tipo de delito).

# 1.2 Antecedentes

A continuación, se presentan algunos trabajos relacionados que dan tratamiento al problema de la inseguridad por medio de algoritmos que realizan minería de textos de páginas web, sistemas de información o aplicaciones móviles donde se realizan publicaciones y se dan atención a estas publicaciones. Algunos otros trabajos, son sistemas de información en los que se realizan publicaciones de otros temas de interés como médicos y se analizan todos los datos obtenidos después de un periodo de tiempo.

## 1.2.1 Proyectos Terminales

- Sistema de información para el análisis de la seguridad social o pública a través de periódicos [3].

En el proyecto descrito en [3], el objetivo es analizar periódicos; el análisis se realiza mediante la conversión de noticias de una página web en archivos de textos. A partir de los archivos se aplica minería de textos y entonces realizar el análisis, para así de esta manera obtener datos adicionales y clasificados.

Si se compara ReportIt con este proyecto, existe la semejanza del problema central a tratar, los incidentes delictivos que ocurren a diario, pero en diferencia el proyecto [3] realiza minería de textos de páginas web, clasifica los datos y los muestra en una interfaz gráfica.

## 1.2.2 Artículos

- Extracting Violent Events From On-Line News for Ontology Population [4].

En el artículo [4], se desarrolló un sistema de extracción de eventos violentos con el cual llena la base de datos de conocimiento de incidentes violentos. Este sistema extrae automáticamente eventos relacionados con seguridad ciudadana de noticias en línea con un algoritmo de bootstrapping[5].

En este artículo se extrae datos de eventos de seguridad mediante un algoritmo de bootstrapping y con esto llena una base de datos, no se realiza una aplicación para visualizar la información en una interfaz de usuario.

[5] *Bootstrapping* es un método para derivar estimaciones robustas de errores estándar e intervalos de confianza para estimaciones tales como la media, la mediana, la proporción, el coeficiente de correlación o el coeficiente de regresión.

- The Impact of a Web-based Reporting System on the Collection of Medication Error Occurrence Data [5].

El artículo descrito en [5], examina la ocurrencia de errores de medicación, estos datos se dividieron en dos bloques, el primero, los datos se analizaron en papel y el segundo, los datos los datos se registraron en un sistema web. Categorizaron cuatro áreas de errores: aumento en el número total de errores de medicación publicados, aumento en el número de errores de medicación interceptados, aumento en el número documentado de errores de medicación atribuidos por el médico y por último mejora en la calidad y especificidad de los datos.

Se encuentra relación del articulo [5] y ReportIt ya que en los dos sistemas web su función principal es el poder realizar publicaciones, en caso de ReportIt no se realiza un análisis de los datos que se van publicando, sino únicamente se publican. Se difiere en los problemas a resolver debido a que este artículo se enfoca en errores médicos no en inseguridad ciudadana.

### 1.2.3 Software

- Alertux [6]

Es una aplicación móvil y web que recopila, organiza y distribuye alertas generadas por la sociedad. Se comparte el objetivo de ayudar a la sociedad evitando lugares peligrosos y contribuir con el desempeño de las autoridades.

- Mi policía [7]

Aplicación por parte de la policía de la Ciudad de México, que sirve para realizar llamados de emergencia y así conocer la ubicación automáticamente y dar una eficiente respuesta a la emergencia. Esta aplicación únicamente genera publicaciones al igual que ReportIt.

# 1.3 Justificación

Un 57% de los habitantes de México cuentan con un teléfono inteligente [8], su gran mayoría en la Ciudad de México. Por tanto, es una ciudad que se encuentra en uso constante de los servicios proporcionados por internet, como lo son, por ejemplo, el uso de las redes sociales, las aplicaciones, entre otros, y principalmente, el acceso a la información. Es por ello que se necesita una aplicación para generar una sociedad más participativa y tener impacto sobre la misma poniendo a su alcance información sobre la seguridad en el país, así como la posibilidad de publicar cualquier incidente delictivo.

Por lo tanto, en este proyecto se desarrolla una aplicación donde los usuarios puedan, entre otras funcionalidades, publicar y consultar información sobre incidentes delictivos de una manera fácil. Con esta aplicación y el uso adecuado de la información publicada por los usuarios, se espera que los niveles de delincuencia sean de dominio público, y que la sociedad sea consciente de las zonas con mayores incidentes delictivos.

Hagamos énfasis en el tema del uso del internet, desde su aparición, todo lo que use este servicio se ha ido revolucionando conforme pasa el tiempo a tal grado que si comparamos el valor al mercado de una empresa como lo es Alphabet Inc. que es una empresa filial de Google, CapitalG, Nest Labs, entre otras empresas (valor al mercado de 554,000 millones de dólares[9]), contra el de una automotriz fuerte como lo es la Ford (valor al mercado de 44.9 millones de dólares [10]), podemos darnos cuenta del gran impacto que tiene una empresa de software en la sociedad, es por ello que las aplicaciones ya sean móviles o web, son relevantes dentro de esta época.

# 1.4 Objetivos

## 1.4.1 Objetivo general

Desarrollar una aplicación web para gestionar incidentes delictivos en México.

## 1.4.2 Objetivos específicos

- Diseñar un modelo de clases para dar estructura a la aplicación e implementar una base de datos relacional para la persistencia de información de la aplicación.
- Desarrollar un módulo de aplicación para crear, consultar, eliminar y modificar perfiles de usuarios.
- Desarrollar un módulo de aplicación para publicar, eliminar y modificar incidentes delictivos basados en ubicación, tiempo e información del incidente.
- Desarrollar un módulo de aplicación para la búsqueda de incidentes delictivos basada en cuatro parámetros: ubicación, fecha, palabra clave y tipo de delito.

# CAPÍTULO 2. Marco teórico

En este capítulo se proporcionará conocimiento para desarrollar aplicaciones web. Se explicará sobre todos los servicios y herramientas utilizadas para poder desarrollar ReportIt como lo son el uso de: los framework´s, la IDE Visual Studio y los servicios que se ocuparon de ésta, los lenguajes de programación, las librerías usadas, los manejos de sesiones con cookies, el uso de la geolocalización en mapas, entre otros servicios y herramientas.

## 2.1 Conector de MySQL .NET

Es un controlador ADO .NET[6] completamente administrado por MySQL. En las versiones más recientes, a partir de la versión 6.7, ya no se incluye el conector con la IDE Visual Studio, ahora esa funcionalidad estará disponible por separado y se podrá instalar en Windows directamente. En el siguiente capítulo, se explicará la manera en la que se realizó la conexión con la aplicación web. En la liga [14] encontraráel conector presentado.

## 2.2 Framework

### 2.2.1 ¿Qué es un framework?

Es un patrón o esqueleto para el desarrollo y/o la implementación de una aplicación [11]. Esta es una definición genérica debido a que un framework de igual forma puede serlo, por ejemplo: el paradigma MVC es un framework debido a que divide nuestra aplicación en tres componentes, el modelo, la vista y el controlador.

Los frameworks no necesariamente tienen que estar ligados a un lenguaje de programación, aunque así lo parezca. Eso si, cuanto más detallado este un framework, más necesidad tendrá de ceñirse a un lenguaje de programación.

### 2.2.2 Ventajas de usar un framework

[6] ADO .NET es un conjunto de componentes del software que pueden ser usados por los programadores para acceder a datos y a servicios de datos. Es una parte de la biblioteca de clases base que están incluidas en el Microsoft .NET Framework.

- El programador únicamente necesita hacer uso del esqueleto que el framework te proporciona.
- Permite definir y estandarizar estrategias de programación esto debido a que proporciona un esqueleto el cual debe ser llenado. De esta manera, si se está programando en conjunto, los demás programadores podrán entender más rápido lo que se ha hecho.
- Es más fácil encontrar herramientas como librerías adaptadas al framework para facilitar el desarrollo.

### 2.2.3 Conclusión

El uso de un framework para el desarrollo de una aplicación implica el que se debe de tener un proceso de aprendizaje primero para poder utilizar correctamente este framework.

Los frameworks, como se mencionó en las ventajas, al proporcionar una estructura, un esqueleto, nos hace la vida más fácil como programadores Esto debido a que, primero, facilita el desarrollo y segundo, nos facilita la modificación de código independientemente de si el proyecto en conjunto, que es lo más común en el desarrollo de software.

La selección de un frameworkes una tarea difícil, debido a los tantos frameworks que existen y en los diferentes lenguajes. Sin embargo, al final solamente permanecerán los que estén mejor definidos, para la aplicación, incluso si no permanece ningún framework, se puede comenzar a crear uno propio con las necesidades que requiere la aplicación.

## 2.3 Visual Studio 2015

### 2.3.1 ¿Qué es Microsoft Visual Studio 2015?

Es un conjunto de herramientas para crear software, desde la fase de diseño pasando por la fase de proyección de la interfaz de usuario, codificación, pruebas, depuración, análisis de la calidad y el rendimiento del código, implementación en los clientes y recopilación de telemetría [7]de uso. Estas herramientas están diseñadas para trabajar juntas de la forma más eficiente posible y todas se exponen a través del Entorno de desarrollo integrado (IDE) de Visual Studio [12].

[7] La telemetría es un conjunto de datos sobre el uso de nuestra aplicación, que nos va a permitir desde tener estadísticas de uso, hasta poder detectar errores y dónde están ocurriendo [13].

### 2.3.2 ¿Por qué usar Visual Studio como entorno de desarrollo?

Esto debido a que es un IDE muy extensible y gratuito, un entorno de desarrollo muy amigable y completo donde se puede desarrollar aplicaciones modernas para Windows, Android e iOS, además de aplicaciones web y servicios en la nube como Microsoft Azure.

### 2.3.3 Servicios de Visual Studio

Visual Studio permite crear muchos tipos de aplicaciones, desde las más sencillas aplicaciones y juegos de la tienda para clientes móviles, hasta sistemas grandes y complejos para empresas y centros de desarrollo. En Visual Studio se pueden crear:

- Aplicaciones y juegos que se ejecutan no solo en Windows, sino también en Android y en iOS.
- Sitios web y servicios web basados en ASP.NET, JQuery, AngularJS y otros entornos populares.
- Aplicaciones para dispositivos y plataformas tan diversas como Azure, Office, Sharepoint, Hololens, Kinect e Internet solo por nombrar algunos ejemplos.
- Juegos y aplicaciones con gráficos avanzados para una variedad de dispositivos Windows, incluido Xbox, con DirectX.

### 2.3.4 Visual Studio 2015 como un IDE

Visual Studio, como se ha mencionando con anterioridad, es una herramienta fuerte para el desarrollo de aplicaciones. Por todo lo que ésta nos puede dar, a continuación, se verán muchos de los servicios que nos proporciona la IDE para un desarrollo de aplicaciones cómodo y rápido, ver Figura 1, la posibilidad de mantenerte siempre actualizado en las versiones de software que se usa, los servicios personalizados con el inicio de sesión, la posibilidad de trabajo en equipo con versiones, etc.

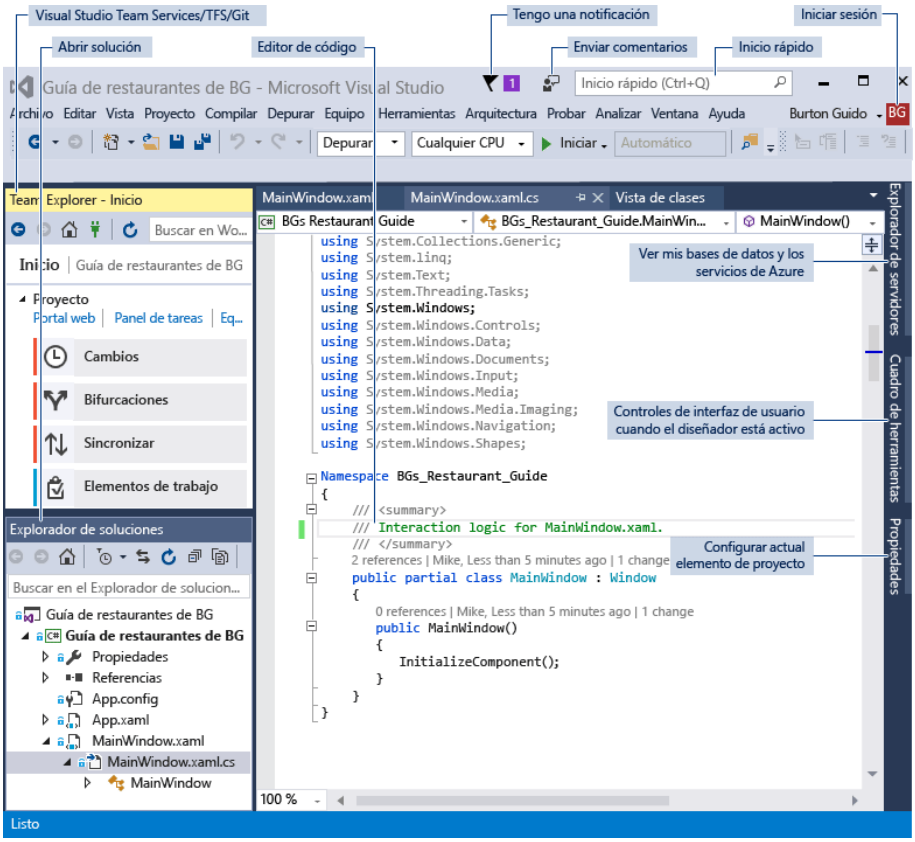


Figura 1. Captura de pantalla de Visual Studio con las secciones resaltadas. Imagen tomada de la referencia [12]

### 2.3.4.1 Inicio de sesión

Al momento de iniciar sesión Visual Studio permite sincronizar sus configuraciones con la de otros dispositivos como las posiciones de las ventanas y conectarse automáticamente a los servicios que pueda necesitar como la suscripción a Microsoft Azure y Visual Studio Team Services, este último se explicará más adelante.

### 2.3.4.2 Mantente actualizado

En el icono de notificaciones de la barra de título, le avisa cuando hay actualizaciones disponibles para Visual Studio o para otros componentes que usted haya instalado. Se puede elegir que acción desea realizar ante estas actualizaciones, si desea omitirlas o hacerlas.

### 2.3.4.3 Realizar búsquedas y obtener ayuda

En la barra de inicio rápido, se puede realizar búsquedas de comandos, las herramientas o características de Visual Studio si no conoce la ubicación del menú o el método abreviado del teclado solo escríbalo y Visual Studio le proporcionara un vínculo.

También, Visual Studio, cuenta con un sitio web de documentación técnica llamado MSDN, dentro de Visual Studio, presiona la tecla F1 y lo llevará a la página de ayuda de MSDN de la ventana activa. De igual manera si coloca el símbolo de intercalación en algún lugar del código y presiona F1, esto proporcionará documentación sobre esa clase, método en el que estas posicionado.

### 2.3.4.4 Personaliza tu IDE

Puede personalizar el diseño de las ventanas para que se ajuste a su estilo de desarrollo. Puede acoplar, hacer flotar u ocultar cualquier ventana en cualquier momento, y también puede ejecutar el editor en modo de pantalla completa. Puede crear y guardar varios diseños de ventanas personalizados que muestren solo las ventanas que necesita para contextos específicos. Por ejemplo, puede crear un diseño de pantalla completa para que todo lo que vea sea el editor de código. Y puede crear diseños diferentes para la depuración y para las operaciones del equipo.

### 2.3.4.5 Conectarse a Visual Studio Team Services y Team Foundation Server

Visual Studio Team Services (VSTS) es un servicio en la nube para hospedar proyectos de software y que permite la colaboración en los equipos. VSTS admite los sistemas de control de código fuente Git y Team Foundation, así como las metodologías de desarrollo Scrum, CMMI y Agile. El control de versiones de Team Foundation (TFVC) usa un solo repositorio del servidor centralizado para los archivos de seguimiento y de versión. Los cambios locales siempre se protegen en el servidor central, donde otros desarrolladores pueden obtener los cambios más recientes. Team Foundation Server (TFS) 2015 es el centro de administración del ciclo de vida de aplicación de Visual Studio. Permite a todas las partes interesadas en el proceso de desarrollo participar con una única solución. TFS es útil para administrar equipos heterogéneos y también proyectos.

Si tiene una cuenta de Visual Studio Team Services o Team Foundation Server en la red, conéctese a ella en la ventana de Team Explorer, ver Figura 2. Desde esta ventana puede proteger o desproteger código en el control de código fuente, administrar elementos de trabajo, iniciar compilaciones y acceder a los salones y las áreas de trabajo del equipo. Puede abrir Team Explorer desde Inicio rápido, ver Figura 1 o, en el menú principal, en Ver -> Team Explorer o Equipo -> Administrar conexiones [12].

En nuestro caso, se utilizó Git como Team Services eso debido a que se requería que el proyecto se mantuviera actualizando en la nube y así no tener incidente de perdida en caso de cualquier incidente durante el periodo de desarrollo.

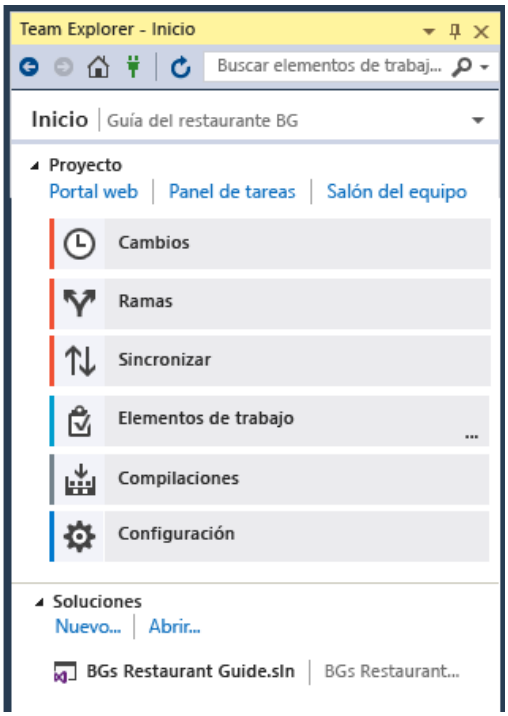


Figura 2. Panel desplegado de Team Services. Imagen obtenida de [12].

## *2.3.4.6 Soluciones y proyectos*

Un proyecto de Visual Studio es una colección de archivos y recursos que, en caso de las aplicaciones, se compilan en un solo archivo ejecutable binario (por ejemplo .exe, DLL, appx, etc.). En el caso de sitios web que no sean ASP.NET, no se genera ningún archivo ejecutable y el proyecto contiene solo los archivos HTML y JavaScript e imágenes. Visual Studio tiene el concepto de solución, que puede contener varios proyectos o sitios web, y es usado cuando estos proyectos o sitios son estrechamente relacionados. Por ejemplo, si tiene un proyecto DLL, puede agregar a la solución un proyecto .exe que carga y usa el archivo DLL.

Una plantilla de proyecto es una colección de archivos de código y opciones de configuración previamente rellenados que permite preparar rápidamente la creación de un tipo específico de aplicación, ver Figura 3. Después de crear un proyecto con una plantilla, puede empezar a escribir su propio código en él, en los archivos proporcionados o en los nuevos archivos que agregue.

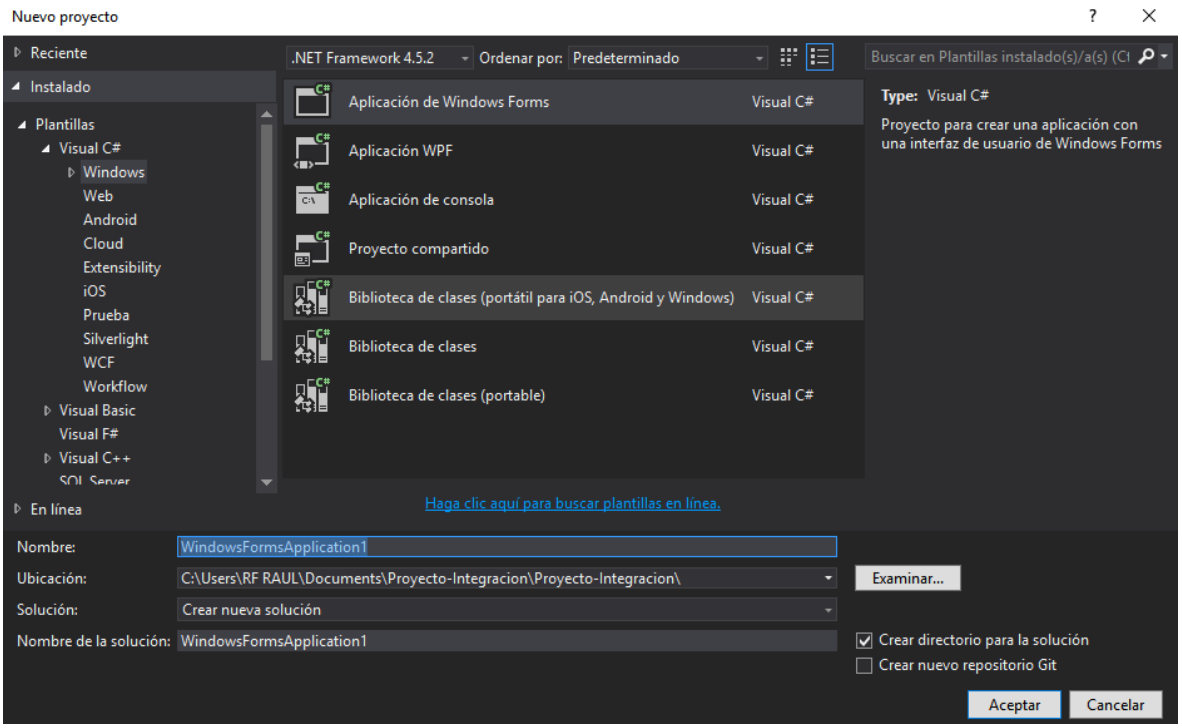


Figura 3. Cuadro desplegado a la hora de crear un proyecto nuevo, la biblioteca de clases es usada para el manejo de clases.

## *2.3.4.7 Escribir código y desplazarse por él y comprenderlo*

Visual Studio incluye editores para C#, C++, Visual Basic, JavaScript, XML, HTML, CSS y F#, así como complementos editores, y compiladores, de terceros para muchos otros lenguajes.

Para editar archivos en un proyecto abierto, haga clic en el nombre de archivo en el Explorador de soluciones. Se colorea el código y puede personalizar la combinación de colores escribiendo "Colores" en el inicio rápido. Puede tener muchas ventanas en las pestañas del editor de texto abiertas a la vez. Puede dividir cada ventana de forma independiente. También puede ejecutar el editor de texto en modo de pantalla completa.

El editor de texto es sumamente interactivo con muchas características de productividad que le ayudarán a escribir código mejor y más rápidamente. Las características varían según el lenguaje y no tiene que usar todas ellas. Escriba "Editor" en Inicio rápido para activar o desactivar características. Algunas de las características de productividad más comunes son:

- Refactorización: cambio inteligente del nombre de las variables, mover líneas seleccionadas de código a una función diferente, mover código a otras ubicaciones, reordenar los parámetros de una función, etc.

- IntelliSense: término que aglutina un conjunto de características muy populares que muestran información de escritura sobre el código directamente en el editor

ver Figura 4 y, en algunos casos, escriben pequeños fragmentos de código automáticamente. Es como tener documentación básica insertada en el editor, lo que evita tener que buscar información de escritura en una ventana de ayuda independiente. Estas características varían según el lenguaje.

- Subrayados ondulados: Avisan de errores o posibles problemas en el código en tiempo real a medida que escribe. Permite detectar errores mucho antes de que el tiempo de compilación o ejecución los detecte, tal y como se muestra en la Figura 5. Si mantiene el mouse sobre la línea ondulada, le dará información adicional sobre el error, también puede aparecer una bombilla del lado izquierdo con sugerencias para corregir el error.

- En el menú contextual del editor de texto, puede invocar la ventana Jerarquía de llamadas para mostrar los métodos que llaman a y son llamados por el método situado por debajo del símbolo de intercalación.

- Una herramienta relacionada, el Examinador de objetos, permite inspeccionar ensamblados .NET o Windows en tiempo de ejecución en el sistema para ver qué tipos contienen y qué métodos y propiedades contienen esos tipos.

- La opción de menú contextual Ir a definición le lleva directamente al lugar donde se definen la función o el objeto. También hay otros comandos de navegación disponibles haciendo clic con el botón secundario en el editor.

- Code Lens permite buscar referencias y cambios en el código, errores vinculados, elementos de trabajo, revisiones de código y pruebas unitarias, todo sin salir del editor.

- La ventana Ojear la definición muestra un método o definición de tipo en línea, sin salir del contexto actual.

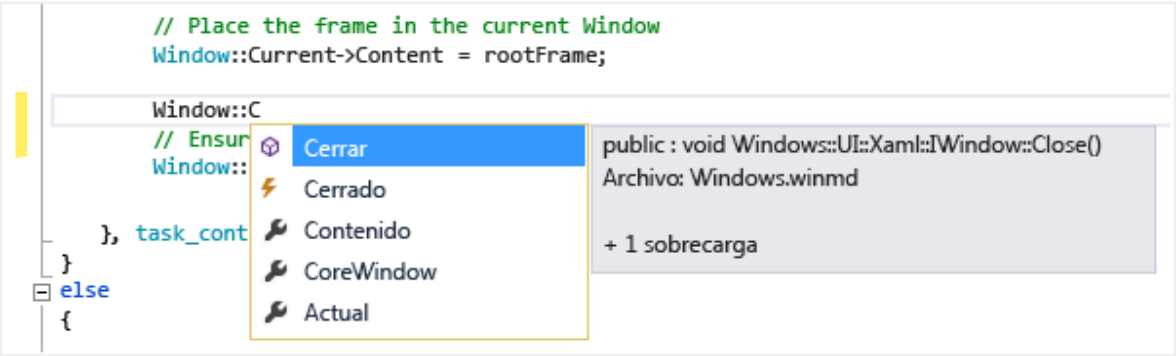


Figura 4. Captura de pantalla que muestra el uso de la configuración IntelliSense. Imagen tomada de [12]

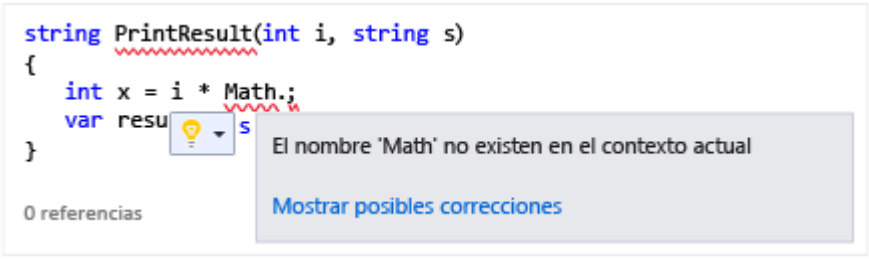


Figura 5. Captura de pantalla que muestra el uso de la configuración, subrayado ondulado, con la función de sugerencia activada.

### *2.3.4.8 Depurar código*

Permite depurar el código que se ejecuta en su proyecto local, en un dispositivo remoto o en un emulador, como los de Android o Windows Phone. Puede ejecutar el código instrucción por instrucción e inspeccionar las variables en cada paso, puede ejecutar paso a paso aplicaciones multiproceso y puede establecer puntos de interrupción que solo se producen cuando se cumple una condición especificada.

### *2.3.4.9 Analizar la calidad y el rendimiento del código*

Visual Studio incluye herramientas eficaces para el análisis estático y en tiempo de ejecución. Las herramientas de análisis estático ayudan a identificar posibles errores de diseño, globalización, interoperabilidad, rendimiento, seguridad y otras categorías. Las pruebas de rendimiento o de generación de perfiles implican medir cómo se ejecuta el programa. A estas herramientas se accede desde el menú “Analizar”.

## 2.4 Lenguajes de programación

### 2.4.1 C#

Es un lenguaje de programación diseñado para crear una variedad de aplicaciones que se ejecutan en .NET Framework. C # es simple, potente, seguro de tipo y orientado a objetos. Las numerosas innovaciones en C# permiten un rápido desarrollo de aplicaciones, al mismo tiempo que conservan la expresividad y elegancia de los idiomas de estilo C.

- .NET Framework

Crea muchos tipos de aplicaciones con .NET, como de nube, IoT (internet de las cosas) y juegos, mediante herramientas multiplataforma gratis. Las aplicaciones se pueden ejecutar en Android, iOS, Linux, Mac OS y Windows [15].

### 2.4.2 Java Script

JavaScript es un lenguaje de programación utilizado para hacer páginas web interactivas. Es lo que da una vida de página, los elementos interactivos y la animación que atraen a un usuario.

Es un lenguaje interpretado, por lo que no se requiere ningún programa especial para crear código utilizable. Un editor de texto sin formato como el Bloc de notas para Windows es todo lo que necesita para escribir JavaScript.

HTML y JavaScript son complementarias. HTML es un lenguaje de marcado diseñado para definir el contenido estático de la página web. Es lo que da a una página web su estructura básica. JavaScript es un lenguaje de programación diseñado para realizar tareas dinámicas dentro de esa página, como animación o un cuadro de búsqueda [16].

### 2.4.3 HTML

HTML es un lenguaje informático diseñado para permitir la creación de sitios web. Estos sitios web pueden ser vistos por cualquier persona conectada a Internet. Es relativamente fácil de aprender y bastante potente en lo que te permite crear. Está constantemente en proceso de revisión y evolución para satisfacer las demandas y exigencias de la creciente audiencia de Internet [17].

## 2.5 Cookies de sesión

Las cookies de sesión permiten que los usuarios sean reconocidos dentro de un sitio web para que cualquier cambio de página o selección de elementos o datos que realice se recuerde de una página a otra sin tener que autenticar o reprocesar cada nueva área que visita [18].

Las cookies son a menudo indispensables para los sitios web que tienen enormes bases de datos, necesitan inicios de sesión, tienen temas personalizables, entre otras características.

### 2.5.1 Clase HttpCookie

Proporciona una manera de seguridad de tipos para crear y manipular cookies HTTP individuales. El manejo de las Cookies funciona similar a un Dictionary en C#, donde se tiene una relación key con su valor correspondiente, de igual modo ese es el funcionamiento que tienen las cookies.

Pasos para comenzar a realizar manejo de sesiones por medio de Cookies desde una clase C# en un navegador:

- Tener tu clase que desees que sea manejador de sesiones.
- Crear un método el cual será el que registre las cookies en tu navegador, con parámetro el valor con el que deseas que sea reconocido tu usuario.
- Instanciar un objeto HtmlCookie y con ayuda del constructor asignar nombre de una nueva cookie
- Le asignamos un valor a es cookie creada
- Se le asigna un tiempo para que expire esa cookie de nuestro navegador.
- Se agrega la nueva cookie creada al navegador por medio de la clase HttpResponse y la propiedad Cookies y el método add. HttpContext.Current.Response.Cookies.Add(cookie instanciada);

### 2.5.2 ¿Por qué usar coockies?

Hay muchas razones para que un sitio web utilice cookies. Cualquier cosa desde que le permite acceder a un área segura de un sitio web, a recordar su nombre o color favorito para la próxima vez que visite el sitio.

No solo le ayuda a proporcionar una mejor experiencia al usuario sino también puede ayudarlo a mejorar su sitio web y servicios proporcionar mayor funcionalidad y servicios, como el color que guardo como predeterminado el usuario.

Aunque las cookies son seguras debido a la encriptación que se tiene no se deben manejar datos sensibles como ubicaciones o contraseñas [19].

## 2.6 Leaflet

Es la biblioteca de JavaScript de código abierto líder para mapas interactivos para móviles; tiene todas las características de mapeo que la mayoría de los desarrolladores necesitan.

Leaflet está diseñado con sencillez, rendimiento y usabilidad en mente. Funciona de manera eficiente en todas las principales plataformas de escritorio y móviles, se puede ampliar con muchos complementos, tiene una API, fácil de usar, bien documentada y un código fuente, sencillo y legible [20].

Es fácil percatarse que en Leaflet los mapas y la mayoría de las cosas funcionan muy diferente a como estamos acostumbrados con Google Maps por ejemplo. Pero a diferencia hay muchas más herramientas que facilitan el desarrollo con Leaflet por ser código abierto.

### 2.6.1 Como comenzar

Para comenzar a utilizar Leaflet dentro de una aplicación web lo primero que se debe descargar. La descarga será una carpeta zip que debe descomprimirse. Una vez realizado esto, aparecerá una carpeta con contenido como se muestra en la Figura 7, ahora únicamente resta incorporar a estas librerías el proyecto.

Para importar el JavaScript que aparece en la carpeta se tiene que agregar en la sección de JavaScript de la aplicación. Posteriormente, se agregael CSS de Leaflet, el cual, de igual manera como se agregó el JavaScript de Leaflet, se tiene que seleccionar la carpeta de CSS de nuestra aplicación e importar el documento.

Incorporación en HTML:

Incorporemos estos dos archivos dentro de nuestro Layout para que sean cargados una vez que nuestra aplicación se ejecute, para importar el css se hace como cualquier otra hoja de estilos como el siguiente ejemplo:

- <link rel="stylesheet" href=https://unpkg.com/Leaflet@1.1.0/dist/Leaflet.css" />

Ahora incorporemos el JavaScript, para hacer esto lo hacemos como cualquier JavaScript, como el ejemplo a continuación:

- <script src="https://unpkg.com/Leaflet@1.1.0/dist/Leaflet.js"></script>

Solo nos falta por incorporar las imágenes png que utiliza Leaflet como marcadores, sombras y labels. Para incorporar a nuestro proyecto, agrégalo en las sección de imágenes y no olvides realizar las modificaciones de las rutas de estas imágenes en la hoja de estilos Leaflet.css.

### 2.6.2 Mapa Sencillo

Para incorporar un mapa sencillo de Leaflet, como el de la Figura 6, dentro de un Html, requerimos de haber hecho previamente el paso anterior, "Como comenzar", después de ello podemos continuar con la incorporación de un mapa dentro de un , centrado en cierta posición, con un marcador y una etiqueta con texto dentro que de una descripción del marcador.

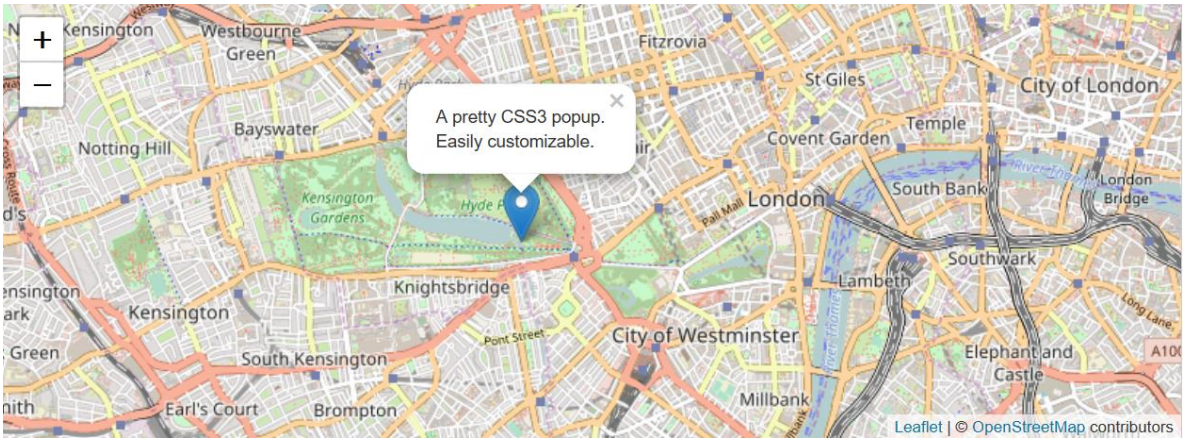


Figura 6. Ejemplo de mapa con marcador en OpenSteetMap usado y manejado por medio de Leaflet.

Paso 1:

Crea un panel o un div donde quiera que se muestre el mapa y date un id igual a “map”, esto más adelante lo usaremos. Es importante que dentro de una hoja de estilos o desde el mismo div, le des un atributo de altura medianamente rectangular.

Paso 2:

Crea un archivo JavaScript, y en él incorporar el código de la Figura 7.

```
var map = L.map('map').setView([51.505, -0.09], 13);

L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
    attribution: '&copy; <a
href="http://osm.org/copyright">OpenStreetMap</a>
contributors'
}).addTo(map);

L.marker([51.5, -0.09]).addTo(map)
    .bindPopup('Mi primer marcador')
    .openPopup();
```

*Figura 7. Código para la implementación de un mapa simple con un marcador y una pequeña descripción.*

## 2.6.3 Plugins Leaflet

Existe una cantidad enorme de plugins para Leaflet, ya que, como lo mencionamos anteriormente, este es un proyecto de código abierto por lo tanto es posible encontrar con mucha facilidad códigos con implementaciones ya hechas o mismas modificaciones dentro los documentos de Leaflet. Los plugins más populares se muestran en la Figura 8.

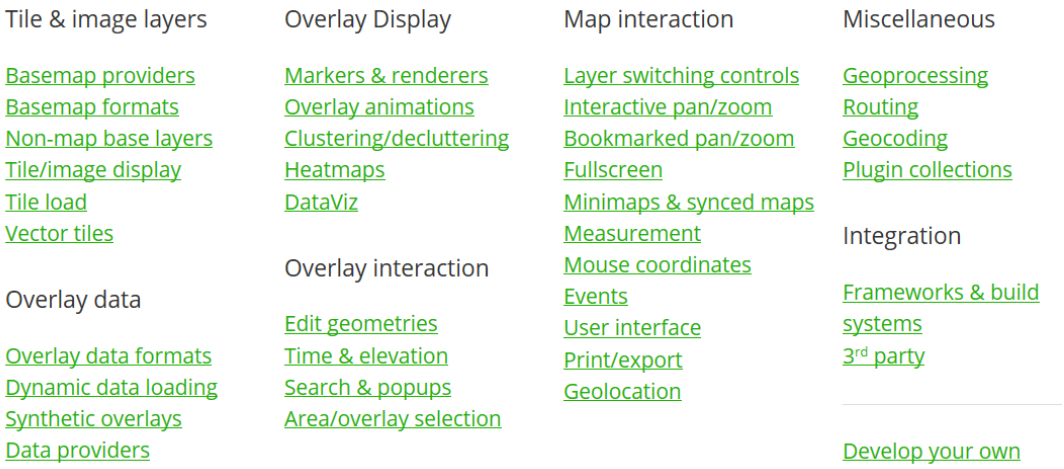


Figura 8. Plugins obtenidos de la página principal de Leaflet.

### 2.6.4 Geolocalización con Leaflet

Para realizar la geolocalización utilizamos la misma Api de Leaflet que nos proporciona un comando en JavaScript que realiza la geolocalización de nuestro dispositivo junto con un conjunto de opciones para dar mejor apariencia[8].

[8] Para un mayor detalle de la geolocalización en Leaflet, consultar [21].

# CAPÍTULO 3. Desarrollo del proyecto

El objetivo de este capítulo es dar la metodología ocupada para llegar a desarrollar una apliación como la nuestra. Dicho capítulo está dividido en cuatro módulos cada uno con diferentes submódulos internos. El primer módulo consta del módelo de clases; en éste diremos que es lo que se consideracomo comienzo y en que ayudará en los siguientes modulos. En el segundo módulo, la base de datos, se pretende que consigas desarrollar toda la parte del servicio de base de datos incluyendo la conexión con la aplicación y los procedimientos almacenados. En el tercer módulo, nos encargaremos de mostrarte los pasos que se siguieron para realizar toda la gestión de un perfil de un usuario. El cuarto módulo, Gestión de pubicaciones de usuarios, de igual modo que en el tercer módulo, te explicaremos la metodológia utilizada para conseguir los objetivos específicos de este proyecto.

## 3.1 Modelo de clases

El modelo de clases es de las primeras cosas que se deben de realizar antes de comenzar a programar, esto debido a que te da una idea más profunda de lo que se puede realizar en la aplicación. Además, te da una imagen de como puedes manejar las vistas, te da la estructura en el proyecto junto con las relaciones entre los mismos objetos. Y claro, lo más importante, te da las herramientas suficientes para modelar tu base de datos y saber como se llevará acabo el manejo de los datos. En la Figura 9, se encuentra el modelo de clases utilizado para desarrollar ReportIt.

Lo primero que debes hacer cuando quieres realizar un modelo de clases es pensar en cuales serán los objetos que participarán en tu aplicación, por ejemplo: un usuario, una publicación, por mencionar algunos objetos. En nuestra aplicación, manejamos seis objetos los cuales detallamos a continuación.

- **Cuenta**:
  Este objeto cuenta con varias propiedades y métodos como se puede ver en el código del Apéndice A.3. Un objeto “Cuenta” consta de dos propiedades, un email y una contraseña; las acciones que puede llegar a realizar este objeto son: Crear una cuenta, modificar una cuenta, seleccionar una cuenta por su email, eliminar una cuenta, agregar un perfil (cada cuenta tiene que tener un perfil asociado), remover perfil, iniciar sesión, cerrar sesión, crear password.

- **Perfil**:

Un objeto “Perfil” consta de varias propiedades debido a que con éste se tiene mucha más interacción dentro de la aplicación. Las propiedades de este objeto son: Id, una UrlImagen, Nombre, una Cuenta, como lo había mencionado en el objeto anterior, una cuenta no puede existir sino tiene un perfil asociado, esto se vera mas adelante en el Modulo 3, y la última de las propiedades es una Ubicación, la Ubicación es importante debido a que con ésta se realizarán las búsquedas avanzadas por ubicación. Este objeto cuenta con varias acciones como lo son: crear un perfil, modificar un perfil, eliminar un perfil, consultar un perfil por su Id, seleccionar un perfil por el Email de la cuenta, modificar imagen, mis reportes, agregar reporte y Remover reporte. El código de este objeto puede consultarse en el Apéndice A.4.

- **Reporte**:
  El código de este objeto se encuentra en el Apéndice A.5. A continuación describiremos sus acciones y propiedades de este objeto.
  Un reporte, cuenta con propiedades como: Id, un Perfil (del usuario que realizó la publicación), una fecha de expedición, una descripción, una Ubicación (usada para la geolocalización de publicaciones dentro de un radio), un incidente (de éste hablaremos muy pronto), una Fecha de ultima modificación. Cuenta de igual forma con acciones como: crear un reporte, modificar un reporte, seleccionar un reporte, visualizar un reporte.

- **Ubicación**:
  La Ubicación, este objeto es uno de los més importantes debido a que con éste se realizan las geolocalizaciones de los incidentes y también, se realizan, las consultas de incidentes que sucedieron cerca de la ubicación de un perfil. En el Apéndice A.6 encontrará el código de este objeto.
  Las propiedades de este objeto, como ya lo han de pensar, son: un Id, la latitud, longitud, una dirección y un código postal, éste último no fue implementado debido a la complejidad que se presentaba a la hora de tratar de extraer el código postal de un String. Sus acciones son el CRUD con sus siglas traducidas al español, crear, seleccionar, modificar y eliminar.

- **Estantería**:
  Este objeto fue creado para, únicamente, realizar búsquedas dentro de la base de datos, en el Apéndice A.7 se encuentra la codificación de este objeto. Como es de parecer, éste no cuenta con propiedades debido a que no las requere, pero sí con muchas acciones, que son nada todas las búsquedas que la aplicación realizada.

- **Base de datos**:
  Este objeto tiene como función hacer la conexión de la base de datos y/o ejecutar consultas, querys, de datos. Esta clase no se encuentra modelada en la Figura 9, debido a que no tiene efecto en la estructura de la aplicación, pero no por ello deja de ser importante.
  Dicho objeto cuenta con dos acciones. En la primera, GetDataSet, se realiza la consulta de datos a la base de datos, este método regresa un objeto del tipo DataSet, el cual se puede ver como una matriz de tres dimensiones donde son varias tablas y cada una de ellas con varias tuplas. En la segunda acción, método QueryCommand, es un método el cual retorna un valor entero diciendo si fue posible o no realizar la actulizacón de un campo o lo que sea que se haya

ejecutado en el QueryEste método es ocupado en todos los casos cuando se inserta en una tabla o cuando se actualizan valores esta misma.

- **Tipo de delito**:
  En este caso, los tipos de delitos son manejados desde la capa de aplicación y no desde la base de datos, es por ellos que los tipos de delitos no son una tabla y mucho menos una clase, sino mas bien es un Enum que contiene tipos de incidentes. Para esta aplicación se manejaron cinco tipos de incidentes: Homicidio con el valor de 1, Suicidio con el valor de 2, Robo o asalto con el valor de 3, Violación con el valor de 4 y finalmente Extorción sexual con el valor de 5. El código de este enum se encuentra en el Apéndice A.2.

Una vez que hemos definido nuestro modelo de clases, aún no podemos estar tan seguros de que ese modelo será el final, sin embargo, debemos comenzar la tarea de codificar cada una de estas clases dentro de nuestro proyecto. Para elloseleccionamos Archivo -> Nuevo -> Poryecto, nos desplegará una ventana donde debemos irnos a las opciones de C# y buscar la biblioteca de clases, no olvider darle un nombre y un directorio antes de guardar.

Por defecto te agregará una clase llamada program.cs, puedes eliminarla, no se usará. Ahora lo que resta es comenzar a crear clases y codificarlas. Es sugerible que lo primero que crees sea la clase que realiza la conexión con la base de datos para así poder implementar todos los métodos sin ninguna restricción.

Para probar el funcionamiento de tus clases es sugerible que agregues a tu proyecto un proyecto de consola, para ello debes dar clic derecho en tu solución **Biblioteca de clases** -> **Agregar** -> **Nuevo proyecto**. Se desplegará una nueva ventana y ahí buscarás en la sección de C# un proyecto que sea consola. Por defecto te crea una clase program.cs, en esta clase será donde implementes tus pruebas. No olvides establecer la consola como proyecto de incio.

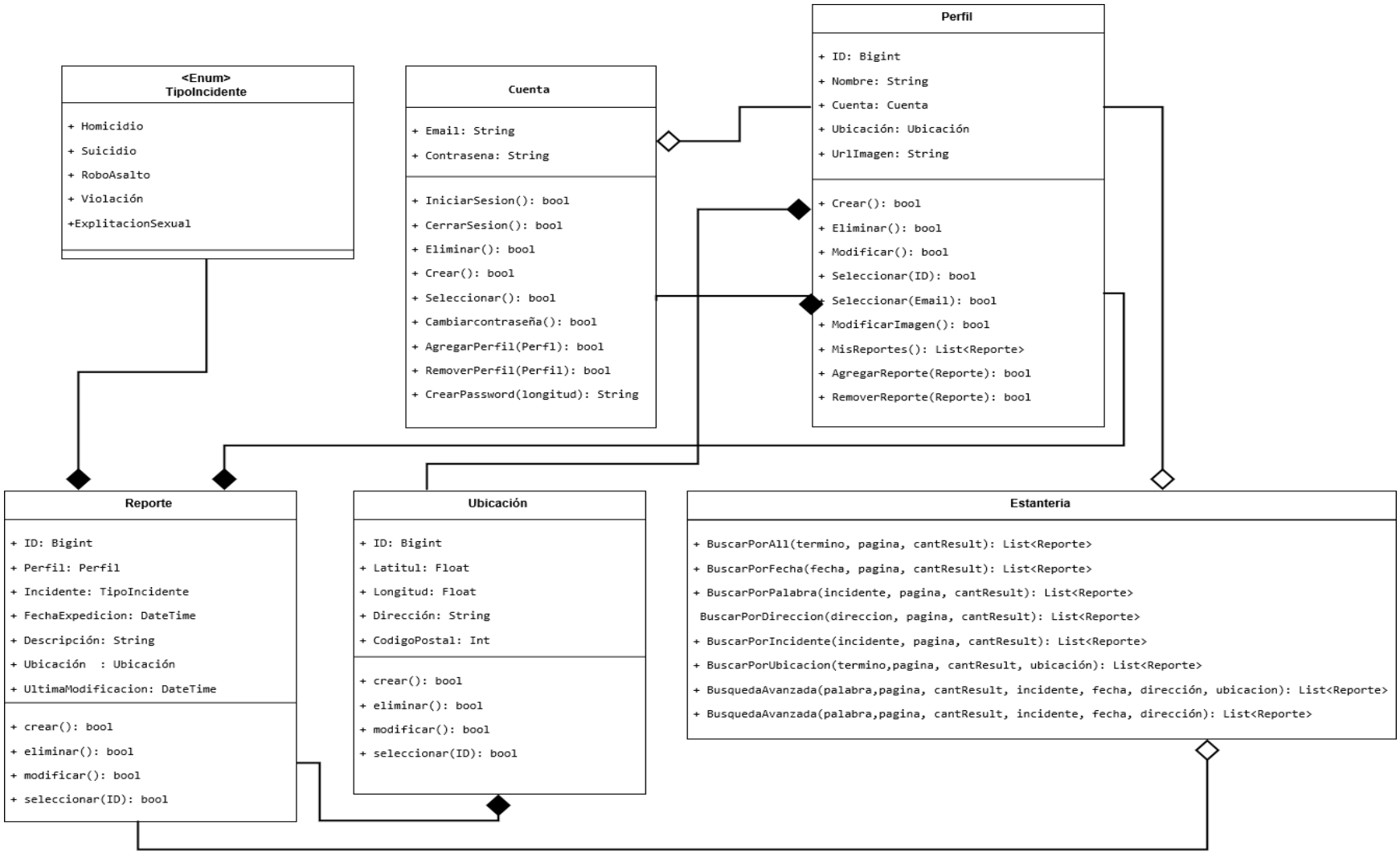


Figura 9. Modelo de clases utilizado para la aplicación de ReportIt.

## 3.2 Base de datos

En este módulo se tratará únicamente el como diseñar la base de datos, que tipo de variables utilizar, como realizar los procedimientos almacenados para realizar consultas, inserciones, eliminaciones y modificacionesAsí como realizar procedimientos almacenados que realicen búsquedas por localización. En la Figura 10, se presenta el modelo entidad – relación ocupado para dar estructura a nuestra base de datos.

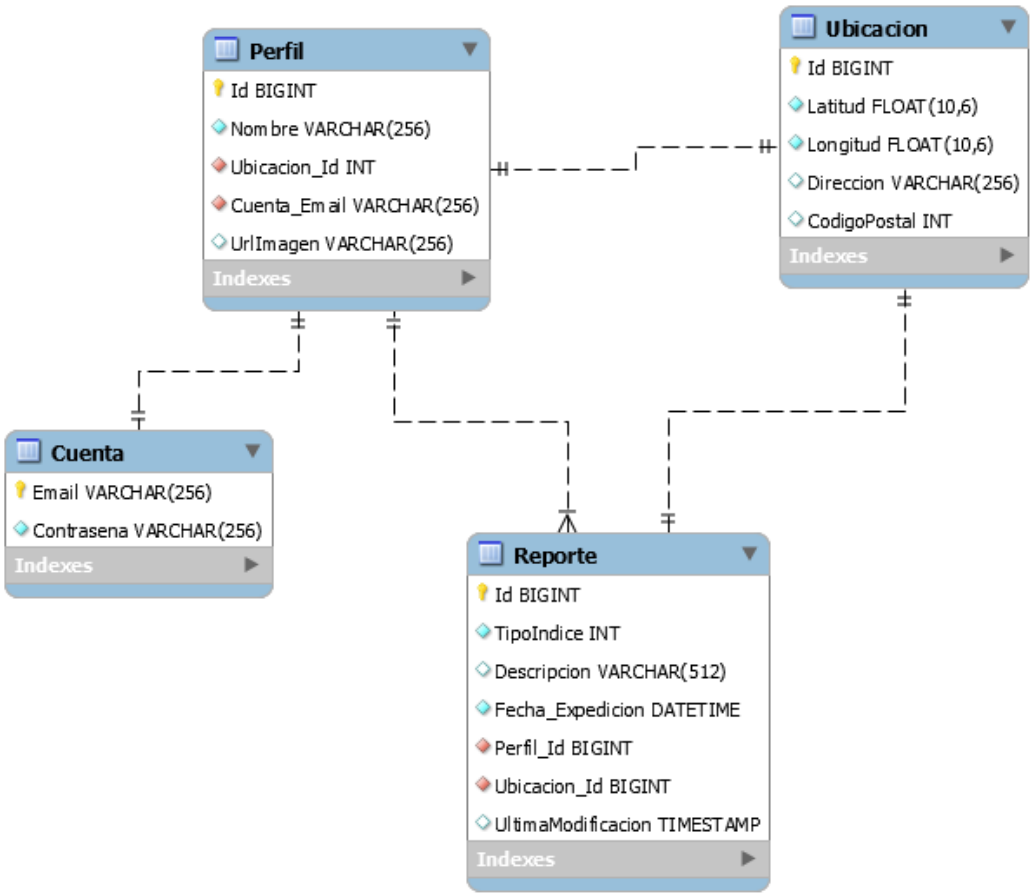


Figura 10. Modelo entidad – relación de la base de datos de nuestra aplicación.

Si se compara el modelo de clases de la Figura 9, al modelo entidad – relación de la Figura 10, podemos percatarnos de ciertas similitudes, es por ello que es importante primero realizar el modelo de clases ya que en este te basarás para realizar tu modelo entidad – relación.

Nuestra base de datos esta diseñada con cuatro tablas: Cuenta, Perfil, Reporte y Ubicación y las relaciones que existen son las siguientes:

- Una cuenta puede tener solamente un perfi y viceversa.
- Un perfil puede tener muchos reportes, pero un reporte no puede tener muchos perfiles.
- Un reporte puede tener solo una ubicación y viceversa.
- Un perfil puede tener solo una ubicación y viceversa.

Se puede apreciar que los tipos de datos manejados para los Id de las tablas son BIGINT, esto pensado a futuro cuando se tengan muchos mas Id sin tener que exceder el tamaño de un INT. De igual modo se utiliza un TIMESTAMP para el atributo de la última modificación de un reporte, ésto para tener control sobre las modificaciones que se realizan. Los DATETIME son ocupados en fechas de expedición de reporte ya que si se requiere se pueden modificar

Como se mencionó, en el módulo anterior, los incidentes serán manejados desde la capa de presentación y en la parte del backend, únicamente manejaremos un INT. Para saber que tipo de incidente fue, esto nos ayuda a ocupar menos espacio en nuestro servidor de base de datos y reducimos el tiempo de consultas e incerciones.

### 3.2.1 Procedimientos almacenados

Los procedimientos amacenados mostrados en el Apéndice B.2, tienen dos guiones medios donde se agrega el delimitador y los simbolos $$ que determinan el límite. Esto debido a que en el Apéndice B.2.10 se muestra un código en XML, el cual utiliza Apache ant y con ello conseguimos ejecutar muchos scripts sin necesidad de estar uno por uno. En este caso, este XML, realiza dos funciones: la primera, elimina todo de la base de datos e incializa la base de datos; la segunda, únicamente ejecuta todos los scripts. Por esta razón, en el proyecto se tiene una carpeta SQL con todo lo relacionado a la base de datos y numerada dependiendo de que tipo de acción tenga en la base de datos. Por ejemplo, los procedimientos amacenados son todos ellos que están numerados con un 400, los que empiezan con 0 son creaciones de tablas y los 900 son inserciones en las tablas, esto se puede ver en la Figura 11. Recordemos que el orden de ejecución es importante debido a las llaves foráneas.

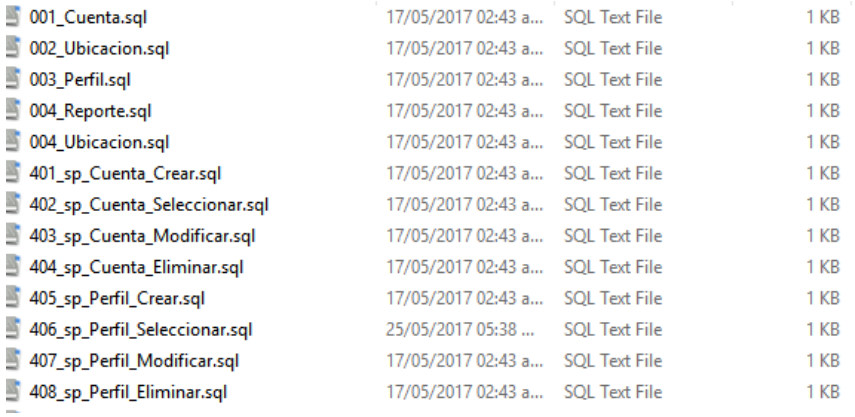


Figura 11 Captura de pantalla que muestra la numeración que se sigue para la ejecución de todos los scripts por medio de Apache Ant.

En cada una de las tablas de la base de datos se crearon sus procedimientos almacenados básicos para realizar el manejo de datos en ellas. Estos procedimientos almacenados cumplen la función del CRUD (Create, Read, Update y Delete), con sus siglas en ingles que significan: Crear (insertar), leer (seleccionar), modificar y eliminar. Los scripts se pueden ver Apéndice B.2.1 al B.2.4, que muestran, únicamente, los cuatro procedimientos almacenados para el manejo de datos de la tabla cuenta debido al gran espacio que ocuparía agregar los demas. La sintaxis que se utiliza para realizar estos procedimientos almacenados son de conocimientos básicos para un programador, es por ello que no se entrará mucho en detalle sobre éstos.

Para los procedimientos almacenados que realizan búsquedas de la clase Estantería, nos adentraremos un poco más, debido a que tienen mayor complejidad.

Los procedimientos almacenados que realizan las bússquedas son los siguientes:

- En todos los atributos:
  Este procedimiento almacenado lo que realiza es una búsqueda de palabras dentro de la descripción, y la dirección del reporte. Además, realiza búsqueda por fecha y por nombre de un perfil.

- En todos los atributos, pero dentro de un radio desde la ubicación de un perfil:
  Al igual que el procedimiento anterior, realiza búsqueda en todos los campos de un reporte y su dirección de éste. Pero adiferencia se tiene que pasar como argumento la latitud y longitud del perfil desde donde se realizará la búsqueda y, es entonces, que se realiza un cálculo con cada uno de las ubicacones de un reporteara determinar si éste está o no dentro de un determinado radio, para nuestra aplicación se uso un radio máximo de 40 Km. La fórmula que realiza el cálculo para determinar si está o no dentro del radio se encuentra en el Apéndice B.2.6.

- Por palabras en descripción, por dirección, por fecha, por tipo de incidente:
  Cada una de estos son procedimientos almacenados de manera diferente, pero más específicos en las funciones para las que fueron creados. Cada uno de los procedimientos almacenados realiza una búsqueda en un diferente atributo dentro de la tabla reporte o ubicación. Los explicamos todos en conjunto debido

a que si entiendes como se realiza la búsqueda en uno de ellos podrás repetirlo en las demas. En el Apéndice B.2.7, se encuentra el códiogo de la búsqueda en la descripción, en este caso se utiliza un SUBSTRING_INDEX para separar lo escrito en palabras y así poder entregar resultados más genéricos.
De la misma manera como se hace la búsqueda en descripción, se realiza con la dirección. En el caso de la fecha, se recibe un varchar con la fecha, que está ya con formato de aaaa-mm-dd y se realiza un CAST AS DATE para buscar, únicamente, por fecha; recordemos que la fecha de expedición es de tipo DATETIME. Para los incidentes se retornan todos los que concuerden con el entero que se busca, correspondiente a un tipo de incidente dentro del enum en nuestro modelo de clases.

- Búsqueda avanzada:
  La búsqueda avanzada es una búsqueda más específica, a las anteriores, debido a que se pueden realizar búsquedas por diferentes campos e incluso por la ubicación de un perfil. El objetivo de este tipo debúsqueda es que se use para buscar reportes de manera más especial. Sepuede realizar la combinación de parámetros de búsqueda como se quiera. Esta búsqueda se encuentra en el Apéndice B.2.7 y B.2.8 con un parámetro más, que es la ubicación del perfil.
  ¿Cómo se realiza la búsqueda avanzada? La búsqueda avanzada, se realiza mediante una serie de combinaciones OR, donde si una tupla concuerda con alguno de los parámetros lo agregará a resultados. Para los campos que no son llenados en la búsqueda avanzada, realiza la búsqueda por nulo. Claro está que debemos asegurarnos de no tener tuplas con campos nulos, es por ellos que estas validaciones deben hacerse desde la capa de presentación.

## 3.3 Gestión de un perfil

### 3.3.1 Email

Email es una clase que ocupamos como un servicio para nuestra aplicación, este servicio consta de envíos de correo electrónico a nuestros usuarios, únicamente, cuando ellos se registran o desean recuperar su contraseña. Para poder recibir el correo electrónico, para cualquiera de los dos casos anteriores, es necesario que se ingrese un correo real, de cualquier tipo, ya que nuestra aplicación no válida nada de eso. Esta clase Email podemos verla en el Apéndice C.1.1.

En el caso de recuperación de contraseña, se genera una nueva contraseña por medio de un método dentro de la clase Cuenta, llamado CreatePassword, este método recibe como parámetro la longitud de la cadena que se generara como contraseña. En sí consiste en seleccionar carácteres de un alfabeto alfanumérico de manera aleatoria, una vez teniendo esa contraseña generada se actualiza su perfil en la base de datos y se envía un correo electrónico con su nueva contraseña generada, el usuario después podrá realizar un cambio de contraseña.

Para mandar un correo electrónico lo realizamos desde una cuenta de Gmail creada especialmente para nuestra aplicación. Para que el servicio de envío de correos funcione debes de tener activado en tu correo electrónico el envío de correos desde aplicaciones externas.

Se utilizó la librería System.Net.Mail la cual tiene una clase MailMessage donde se pueden llenar los atributos. Como parte del mensaje, hacia quien va dirigido el correo electrónico o el asunto del correo, se debe de proporcionar el correo de nuestra aplicación, así como la contraseña y servicios de seguridad si así lo deseamos.

### 3.3.2 Validador

Cuando uno ingresa a nuestra aplicación, se dirige a la vista de ingresar y no ingresa un correo o una contraseña, la clase validadora nos ayuda a determinar si algunos de estos campos son vacíos por medio del método isNullOrEmptyOrWhiteSpace, ya que puede que no sea vacío pero que contenga puros espacios en blanco que finalmente son caracteres también.

Como se explicó, en el párrafo anterior, esta clase es un servicio extra que ayuda a nuestra aplicación a tener un mejor funcionamiento por medio de validaciones.

Esta clase validadora consta de tres tipos de validaciones, la primera validación es ejemplificada en el primer párrafo, la segunda es para validar si el email que se está ingresando, ya sea para autentificarse en nuestra aplicación o para registrarse, no son válidos; esto se realiza gracias a una cadena que nos da el formato que debe tener el correo, esta cadena se compara con el correo ingresado y si éste cumple con el formato del método retornará un valor booleano de true. Con el tercer método, se validan las extensiones de las imágenes que son modificadas en detalles de un perfil, ya que se pueden agregar archivos de cualquier tipo siempre y cuando pesen menos de 10Mb. Esta tercera validación, consiste en extraer la extensión del archivo a descargar y lo compara con extensiones de imágenes (.jpg, .png, etc.). Si este archivo es una imagen, se descargará y se usará como foto de perfil, en caso contrario no se realizará ninguna acción sobre su imagen. Esta clase validadora se puede ver en el Apéndice C.1.2.

### 3.3.3. Manejador de sesiones

En la actualidad no hay aplicaciones que no ocupen cookies para dar mejores servicios y nuestra aplicación no es la excepción. Como recordaremos del capítulo 2, una cookie puede ser vista como un Diccionario, que consiste de una key y un value.

El manejador de sesiones es una clase que realiza todo el manejo de las cookies del navegador donde se esté visualizando nuestra aplicación, el cual realiza:

- Ingresos: se crea una cookie con un valor del email de usuario que quiere ingresar o se haya registrado.
- Registrar perfil: agrega el Id de un perfil a una cookie con key Perfil.
- Salir: consiste en modificar los valores de email e Id de un perfil por vacíos.

- Cuenta activa, retorna un objeto de tipo Cuenta con el email que está registrado en esta cookie.
- Perfil activo: retorna un objeto de tipo perfil, este objeto es seleccionado con el Id del perfil que este en la cookie.

Con el manejador de sesiones podemos dar una mejor experiencia a nuestro usuario ya que podemos dar un servicio más personalizado, mostrándole que conocemos su nombre de usuario y con ello puede acceder a todo lo relacionado a su cuenta.

## 3.3.4 Geolocalización

La primera vista que se mostrará, cuando ingresas a nuestra aplicación, será la vista de inicio, esta vista da una introducción al servicio y objetivos que tenemos dentro de nuestra aplicación. En dicha vista se mostrará el mapa de últimas modificaciones, como el la Figura 12, este es un mapa de Leaflet y está geolocalizado pero esta vez es por parte del servicio de Leaflet.

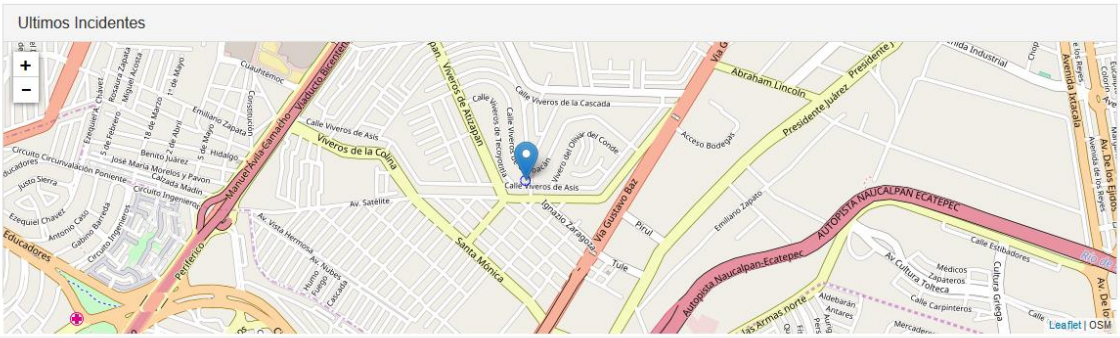


Figura 12. Mapa de últimos reportes de la vista inicio con marcador geolocalizado a la posición de un dispositivo.

La geolocalización es de lo más importante de este proyecto ya que todos los reportes de incidentes delictivos son localizados en un mapa, se les asigna una dirección y esta dirección es obtenida precisamente por la clase de Geolocation. Esta clase implementa varios métodos y en general los más importantes son obtener nombre de la ciudad, nombre de la región, la ip del dispositivo, y un método nombrado getDataRow.

Todos los métodos implementados dependen del método getDataRow, ya que este le pasa una ubicación con lalitud y longitud y realiza una búsqueda en Google mediante la siguiente url:

http://maps.googleapis.com/maps/api/geocode/xml?latlng=lat,lon&sensor=false.

La extracción de los datos se realiza visualizando los resultados de la url anterior en un XML y este XML se lee por medio de un DataSet.Finalmente, se retorna el renglón cero que es el que contiene toda la información en un objeto tipo DataRow.

Cuando realizamos la selección de una ubicación, ya sea de un perfil o un reporte, le damos el clic al botón de confirmar, en ese momento se hace una llamada a un método Post dentro de un controlador y un método específico, el cual realiza la búsqueda de la dirección por medio de la clase Geolocation y la latitud y la longitud del marcador colocado en la vista. El resultado de este método de búsqueda es asignado a la dirección del objeto ubicación y es modificado para la persistencia de datos. Es así como se conocen las direcciones de los marcadores en los mapas.

Puede implementarse una base de datos más normalizada para las búsquedas y es una de las cosas que se podrían mejorar en las siguientes versiones del proyecto.

## 3.3.5 Layout

Desde cualquier vista de nuestra aplicación puedes visualizar que el contenido de las Figuras 13 y 14, siempre aparecerán, esto debido a que son el layout que manejamos. El Layout consta de ser la base de nuestra aplicación, metafóricamente podemos hablar de que es el esqueleto de nuestra aplicación, ya que en este todo se maneja, se importan las hojas de estilo (como Bootstrap) y JavaScripts (como Leafelt), para el uso dentro de todas más demás vistas.


Figura 13. Header de nuestra aplicación: Barra superior (Menú), una introducción y un buscador filtrado.


Figura 14. Footer de nuestra aplicación: nombre de nuestra aplicación, año de desarrollo, país origen y este con una liga hacia un acerca del desarrollador de nuestra aplicación.

Layout es una vista que encontraras en el apéndice C.2.1 y no es una vista parcial como el Header, Figura 13, o el Footer, Figura 14, esta es una vista normal y requiere de la implementación de un RenderBody(), que es un método que indica que en esa sección será el contenido que estará cambiando, es por ello que decimos que el layout es nuestro esqueleto porque todo se va a manejar desde una sola vista pero modificando el cuerpo de esta por otras vistas que podrían verse como parciales pero no se implementan como tal.

El header consta de tres secciones:

- Una barra superior (navbar), nuestro menú, en donde se pueden realizar diferentes acciones dependiendo si estás o no autentificado. En la Figura 15, se muestra el menú en el caso de no estar autentificado y se puede ver que no puedes realizar reportes, ver las opciones de un perfil, ni nombre de un perfil como en la Figura 13.
- Una introducción a nuestra aplicación, esta de igual forma depende de si está o no autentificado el usuario ya que se personaliza. Si se compara la introducción del header de la Figura 13 y la Figura 15 se nota que en la vista donde no hay usuario autentificado se invita a registrarse por medio de una liga. Y en el otro caso, donde el usuario esta autentificado, se invita a visualizar su perfil.
  En la introducción se tiene como objetivo dar a conocer que es lo que se realiza en nuestra página y en que les puede beneficiar.
- Un buscador filtrado, este buscador se quedó como buscador principal y tiene cinco filtros, como se explicó en la sección 3.2.1 de este capítulo, pero a nivel de base de datos, el primero de ellos realiza una búsqueda en todos los parámetros de un reporte incluyendo el nombre de un perfil por si se desea conocer reportes de una persona. El segundo filtro, por dirección. El tercero, por una fecha mediante la visualización de un calendario. El cuarto por palabras dentro de las descripciones y por último por tipos de incidentes, este último despliega una dropdown como el de la Figura 15.


Figura 15. Header de nuestra aplicación con barra superior sin un usuario autentificado.


Figura 16. Buscador principal con la opción de filtro por dirección activada.

Para que se muestre, únicamente, una opción de filtrado dentro del buscador principal, se utilizaron cinco vistas parciales, una por cada filtro. Estas vistas se colocaron dentro de diferentes collapse y cada que uno de ellos se mostraba y los demás se escondían, ver Figura 15 y 16, este efecto se realizó en un JavaScript mostrado en el Apéndice C.2.4.

El Footer, es una vista parcial de las menos importantes que encontrarás en nuestra aplicación ya que únicamente consta con una liga hacia la vista “Acerca de”.

### 3.3.6 Controlador básico (HomeController)

Todos los controladores son una clase como las del modelo de clases, pero a diferencia, los controladores incorporan la interfaz controller para el manejo las vistas. Existen dos tipos de métodos que se pueden implementar dentro de una clase controladora: Get y Post.

Los métodos Get son aquellos métodos que te redirigen a otra vista, son métodos de tipo ActionResult y siempre retornan una vista y posiblemente un modelo también. Más adelante se explicará que es un modelo y cómo funciona en una vista. En general, un método Get se ejecuta siempre que deseas pasar de una vista a otra. Por ejemplo, cuando le damos clic a ingresar en el menú, nos manda a una vista diferente, esa acción es gracias a estos métodos tipo Get.

Métodos Post, los métodos post son los métodos que tienen efecto sobre los datos que se manejan dentro de la aplicación. Como por ejemplo, cuando estamos creando una cuenta nueva y damos clic en crear, este clic realiza la ejecución de un método, el método se encuentra dentro de una clase controladora y lo que realiza es la ejecución de otro método, pero de la clase Cuentahace que los datos ingresados, desde la vista, sean guardados en la base de datos haciéndolos persistentes en el sistema.

Los modelos de las vistas son objetos que usaran como estructura dentro de la vista, por ejemplo, en la vista detalles de reporte, se le pasa como modelo un objeto, este objeto tiene todos sus atributos llenados y es por ello que podemos mostrarlos en la vista detalles de un perfil. Para pasar un modelo a una vista se utiliza la siguiente sentencia: return View(model).

En la clase HomeController, se ejecuta el método Get para visualizar las vistas: Inicio, Faq, About, búsqueda avanzada. Y métodos Post como: Búsqueda avanzada, búsqueda general y resultados de búsqueda. Esta clase puede ser apreciada en el Apéndice C.3.

Para recuperar datos de la vista y usarlos dentro de una clase controladora hay dos opciones, la primera es que le coloquemos el atributo name y le demos un nombre de un atributo de una clase como por ejemplo “Nombre” y dentro de los argumentos del método en la clase controladora agregar un objeto de tipo Perfil y automáticamente te ligará el atributo “Nombre” con el objeto Perfil, así de fácil es desarrollar con .Net, la otra manera es agregar un objeto tipo FormCollection, que abstractamente puede verse como un Dictionary.

### 3.3.7 Acerca de (“About”)

Vista Acerca de, es una vista creada para conocer más acerca del desarrollador de este proyecto de integración por parte de la Universidad Autónoma Metropolita, una vista que solamente gente curiosa encontrará ya que no es algo que todos los usuarios están buscando. En la Figura 17 se encuentra esta vista y en el Apéndice C.2.5 se encuentra la pequeña porción de código para su implementación.


Figura 17. Vista Acerca de, que muestra el desarrollador de esta aplicación y una liga hacia su linkedin.

## 3.3.8 Inicio

Vista de inicio, esta vista no estuvo dentro de los objetivos del proyecto, pero se consideró que era importante para introducir nuestra aplicación.

Dicha vista consta de dos partes, la primera, ver Figura 18, es el mapa de últimos reportes, el cual está pensado para buscar los últimos diez reportes realizados. Esta sección puede mejorarse ya que en un principio se pensó los diez ultimos reportes más cercanos a una ubicación, pero debido a la falta de tiempo esto no fue posible implementar. La segunda parte, ver Figura 19, se muestran imágenes flotadas y los objetivos de la aplicación, junto con una liga para invitar a que te unas a nuestra aplicación y ayudes a la sociedad a mejorar la seguridad de todos y así hacer una sociedad más participativa

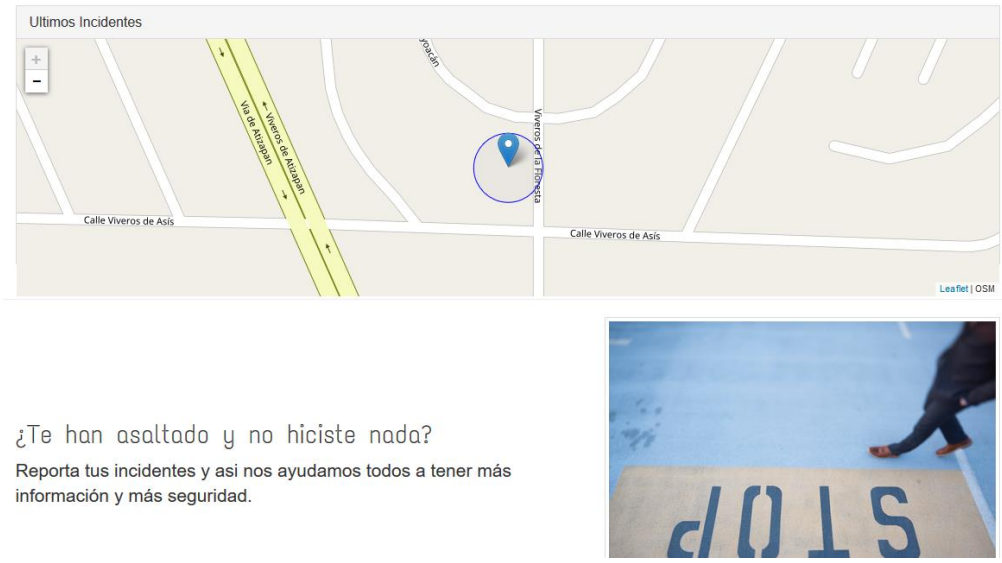


Figura 18. Vista inicio (Primera parte).


Figura 19. Vista inicio (Segunda parte).

### 3.3.9 Preguntas frecuentes (Frequently Asked Questions - FAQ)

Se desarrolló esta vista que al igual que la vista de inicio, no se plantearon en los objetivos específicos. Sin embargo, ésta es de gran utilidad cuando un usuario es nuevo dentro de nuestra aplicación y no sabe cómo funciona correctamente.

El objetivo de esta vista es ayudar a los usuarios con respuestas de las preguntas más frecuentes que se harían dentro de nuestra aplicación. Esto para saber, por ejemplo, cómo realizar una búsqueda avanzada, cuántos reportes puedo realizar, dónde realizar un reporte, entre otras cuestiones.

Para ingresar a esta vista no es requerimiento estar autentificado ya que las consultas de reportes son libres para todo el que visite nuestra aplicación. En la Figura 20, se tienen una gran cantidad de preguntas y respuestas las cuales puedes consultar desde nuestro menú dando clic en “Faq”. La implementación de esta vista se encuentra en el Apéndice C.5.


Figura 20. Vista de preguntas frecuentes.

## 3.3.10 Controlador de cuenta (CuentaController)

Esta clase controladora se encarga de registrar cookies cuando un usuario ingresa y limpia éstas mismas cuando termina su sesión. Ejecuta métodos Get para dirigir a vistas como ingresar, cambiar contraseña y crear cuenta, y métodos Post como, cambiar contraseña, crear, eliminar, olvide contraseña e ingresar.

El método eliminar borra todo lo relacionado a esa cuenta de la base de datos ya que así está definido desde la creación de las tablas, eliminación en cascada.

Los métodos Post, ingresar o crear, registran las cookies en el navegador para su uso Destaquemos que estas cookies se les dio un tiempo de caducidad de un día. Cuando todo se ejecutó correctamente dentro de estos métodos, las consultas o inserciones a la base de datos fueron correctas, retorna la vista de inicio para comenzar a navegar en nuestra aplicación.

El método Post, olvide contraseña, genera una nueva contraseña con el método de la clase cuenta CreatePassword. Además, modifica la cuenta dentro de la base de datos y envía un mensaje de correo electrónico con la nueva contraseña para que el usuario pueda recuperar el acceso y posteriormente modificar su contraseña.

## 3.3.11 Controlador de perfil (PerfilController)

Clase controladora de un perfil complementa la clase controladora de cuentas, debido a que están fuertemente ligadas una a la otra. Desde el controlador de cuenta se puede eliminar una cuenta, o crearla, a la vez el perfil será creado y eliminado. Es

por ello, que en esta clase se implementarán métodos como modificar perfil o el visualizar perfil. La clase controladora de perfiles se encuentra en el Apéndice C.7.

Se encuentra la implementación de métodos Get para ver detalles de un perfil, modificar perfil, visualizar todos los reportes de un perfil paginados y métodos Post para cargar una imagen de un perfil y para modificar un perfil.

Para realizar la modificación de una imagen de perfil se debe acceder a detalles de perfil y esta opción únicamente se habilitará para aquellos usuarios que corresponda el perfil. Se utilizó un JavaScript, el cual descargá los archivos que se seleccionen desde el explorador de archivos de Windows. Es importante que, en la caja de textos, donde deseas implementar que la descarga de archivos, en el atributo class agregues la propiedad filefield. El JavaScript implementado para este servicio se encuentra en el Apéndice C.7.1.

### 3.3.12 Registrar ubicación en Mapa (Leaflet)

El registrar ubicaciones es utilizado en dos vistas, la vista crear perfil o en la vista nuevo reporte, en dichas vistas se presentan mapas como el de la Figura 21, donde se coloca un marcador, en primera instancia, en tu ubicación actual.

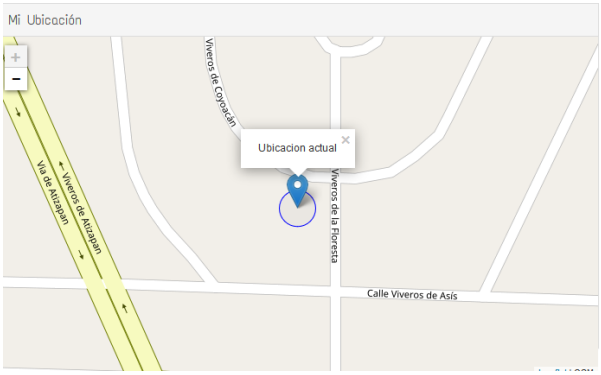


Figura 21. Ubicación actual de un dispositivo que usa nuestra aplicación.

Este marcador con tu ubicación actual por defecto puede ser modificado dando clic en cualquier otra parte del mapa como se muestra en la Figura 22.

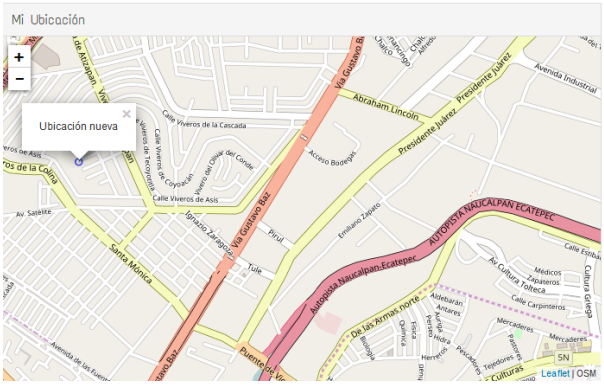


Figura 22. Ubicación modificada en mapa.

Para la geolocalización se utilizó el servicio de Leaflet el cual obtiene la ubicación actual (latitud y longitud) mediante el servicio de geolocation api. Es así como se coloca el marcador inicial con la leyenda “Ubicación actual”. En caso de que se diera clic en alguna posición diferente a la por defecto, se crea un evento el cual es un objeto que contiene una latitud y una longitud, y se modifica las coordenadas del marcador y circulo que indican tu posición actual.

El circulo que se encuentra junto al marcador indica la precisión que se tiene de tu ubicación, es por ello que en algunas de las veces la api de geolocalización no consige obtener la ubicación precisa y se muestra un circulo un poco grande. En estos casos es recomendable refrescar la pantalla y así conseguir una mejor precisión. El JavaScript que controla toda la funcionalidad del mapa se encuentra en el Apéndice C.8.

### 3.3.13 Crear cuenta

Vista crear cuenta, tiene ese nombre debido a que lo primero que se debe de crear es una cuenta debido a la relación que existe en el modelo de clases y la base de datos. En esta vista se tienen que ingresar cuatro atributos, el correo electrónico, una contraseña, un nombre de usuario y tu ubicación seleccionando en el mapa con un clic donde deseas colocar el marcador, ver Figura 23 y24.

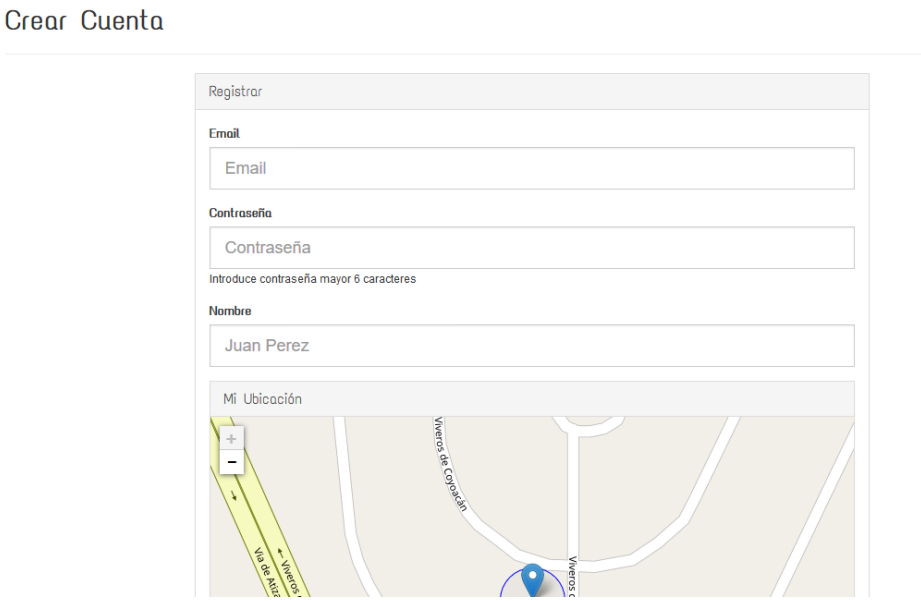


Figura 23. Vista Crear Perfil (Cuenta). Primera parte.

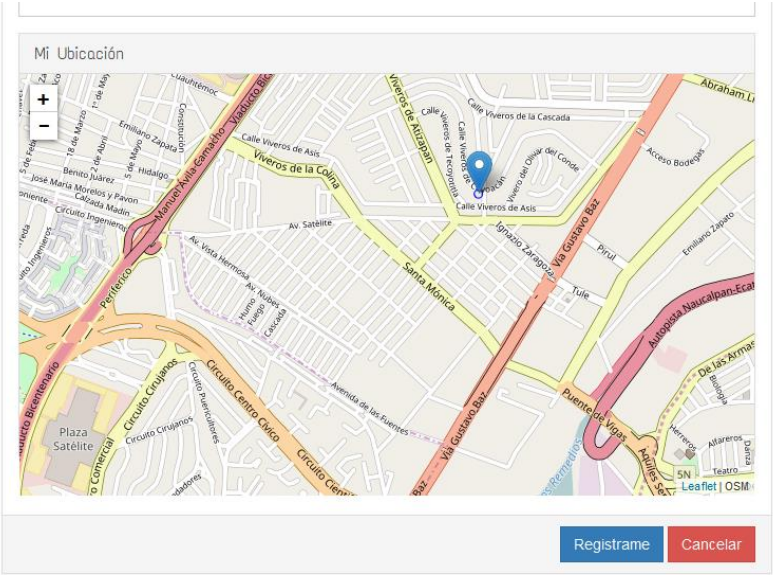


Figura 24. Vista Crear Perfil (Cuenta). Segunda parte.

Para el funcionamiento del mapa, se utilizó lo contenido en la sección anterior, donde se explica cómo es el manejo de eventos clic en el mapa y como se obtiene la ubicación actual por defecto.

Cuando se termina de llenar los campos y se da clic en el botón crear, este ejecuta el método Post dentro de la clase controladora de cuenta, este verifica que ninguno de los datos sea vacío, crea una cuenta, crea una ubicación, crea el perfil con

una imagen de defecto, y se crean las cookies para el email activo y el perfil registrado, y así poder usarlos desde nuestra aplicación.

Dentro del método Post crear cuenta, se envía un mensaje de correo electrónico cuando todo lo del párrafo anterior se tiene permanente en la base de datos. El envió de correo electrónico, es únicamente para dar la bienvenida a un usuario más dentro de nuestra aplicación, el formato del correo es simple y se puede apreciar en la Figura 25. Esta vista crear cuenta puede apreciarse en el Apéndice C.9.

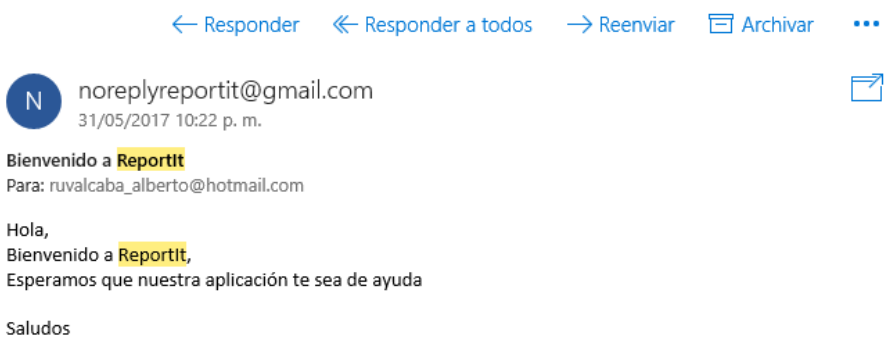


Figura 25. Correo recibido por parte de nuestra aplicación como bienvenida.

## 3.3.14 Visualizar ubicación en Mapa (Leaflet)

La visualización de un marcador dentro de un mapa es la parte más sencilla de implementación de mapas dentro de esta aplicación. El JavaScript utilizado para la visualización de mapas está en el Apéndice C.10. Consiste en centrar el mapa en una latitud y longitud, a una a una altura considerable y colocar un marcador y su círculo en esa misma localización.

En las vistas detalles de perfil y de reportes son las únicas vistas que ocupan este servicio de mapa como se muestra en la Figura 26.

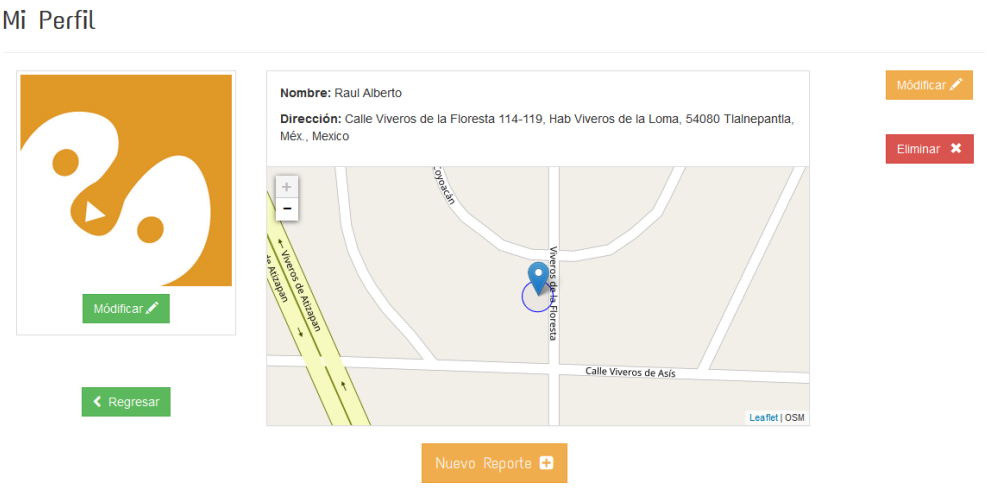


Figura 26. Vista detalles de perfil, con imagen por defecto y un mapa para visualizar una ubicación.

El marcador dentro del mapa visualizar, no es modificable como en el caso de crear una cuenta, pero si puedes manipular el mapa para alejarte o colocarte en diferentes posiciones.

## 3.3.15 Detalles Perfil

Vista detalles perfil es una vista que utiliza el perfil activo para saber si a la vista detalles perfil que estas ingresando es tuya, en caso de que, si sea tu perfil, podrás visualizar todos los botones como se muestra en la Figura 26, donde puedes modificar tu imagen de perfil, modificar tu perfil, eliminar cuenta y por ende todo lo relacionado a esa cuenta, y un botón regresar que tiene dos funcionalidades dependiendo de si estás en tu perfil o no. En caso de estar en tu propio perfil el retornar te regresa a tus reportes, en caso contrario te hace un Windows back, regresando solo a la vista anterior que tenías en tu navegador.

Dentro de esta vista se encuentra un botón nuevo reporte, que te direcciona a la vista crear reporte, esta implementación del botón en esa posición fue únicamente para facilitar la navegación dentro de nuestra aplicación.

En la Figura 27, se muestra que al final de esta vista se encuentran los últimos reportes que ha realizado el perfil, en caso de que sean más de cuatro puedes dirigirte a ver todos sus reportes que tiene ese perfil. Si le das clic a alguno de esos reportes te mandará a detalles del reporte, esta última vista mencionada se explicará en posteriores secciones.


Figura 27. Vista detalles perfil en la sección de ultimos reportes de un perfil.

## 3.3.16 Modificar ubicación en Mapa (Leaflet)

La vista modificar perfil y modificar reporte son las únicas vistas que cuentan con el servicio de modificación de ubicación dentro de un mapa, ver Figura 28, y en si consiste en pintar un marcador en una posición específica, y esta puede ser modificable dando clic en cualquier otra posición del mapa. El servicio de modificación de ubicación en mapa se encuentra en el Apéndice C.12.

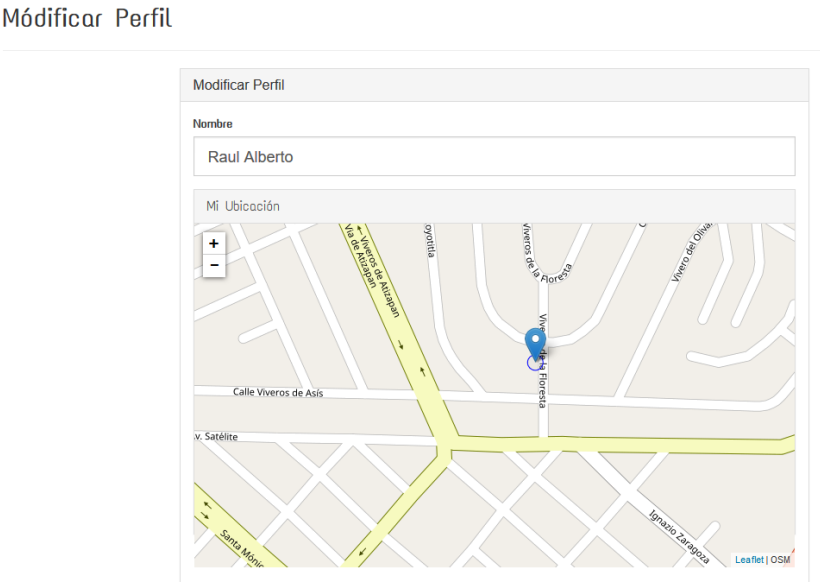


Figura 28. Vista modificar perfil.

De igual modo que en el registrar ubicación en mapa, se dan manejos a eventos clic que suceden dentro del mapa, cada que se da un clic, se crea un objeto con una

posición diferente y muchos atributos más. A nosotros solo nos interesa la latitud y longitud donde se dio clic para actualizar la posición del marcador en el mapa.

### 3.3.17 Modificar y eliminar perfil

Desde la vista detalles de un perfil, y únicamente si al perfil que se está accediendo es el mismo perfil que se encuentra activo, puedes acceder a la vista modificar perfil o eliminar tu perfil. Si nos damos cuenta en la Figura 28, solo se modifican dos atributos de un perfil, su nombre y su ubicación ya que la imagen se puede modificar dentro de la vista detalles de un perfil. La modificación perfil se encuentra en el Apéndice C.13.

Para la modifiacion de una ubicación se utiliza el mismo JavaScript que se explicó en la sección 3.3.16.

Si se da clic en el botón eliminar, desplegara un modal que pregunta si estás seguro de eliminar tu cuenta como se muestra en la Figura 29. Al dar clic en eliminar dentro del modal ejecutas un método Post dentro de la clase controladora de cuenta y este elimina de la base de datos la cuenta y todo lo relacionado a esta.

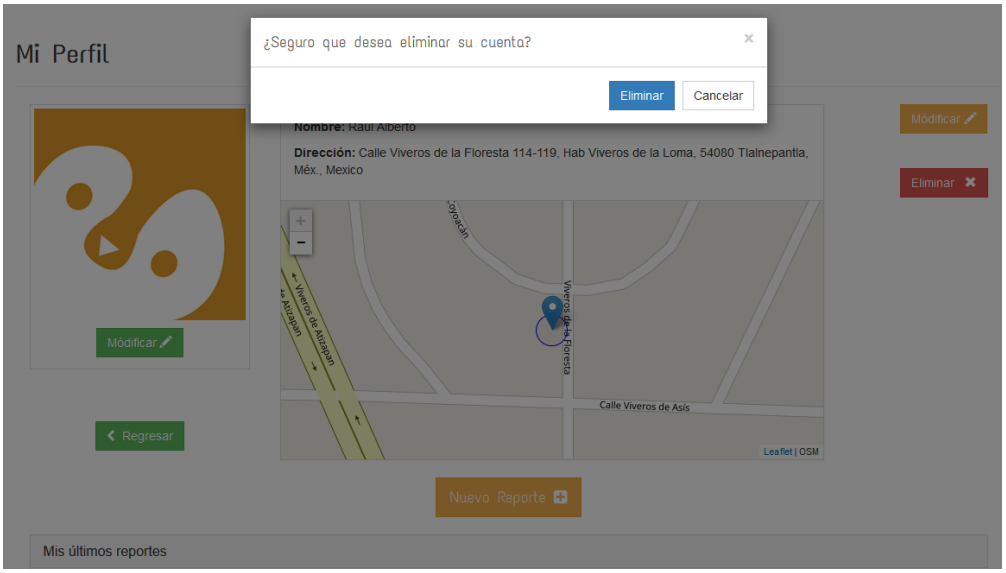


Figura 29. Vista detalles perfil con modal eliminar cuenta activado.

## 3.4 Gestión de un reporte

### 3.4.1 Controlador de reporte (ReporteController)

Clase controladora de un reporte, como cualquier otra clase controladora maneja métodos Get y Post. Las vistas que contiene este controlador son: Detalles de un reporte, nuevo reporte, modificar reporte.

Dentro de detalles de un reporte, si eres el perfil que realizó la publicación del reporte te permitirá realizar la ejecución de métodos Post como eliminar reporte o acceder a la vista modificar reporte pasando como modelo de la vista el reporte que se desea modificar y realizar la modificación con el método Post correspondiente.

Para la creación de un reporte no se utiliza modelo debido a que no se tiene un esqueleto del cual obtener datos para visualizarlos en la vista crear reporte. Como bien lo dice es crear un reporte, por lo tanto, todos los campos se encontrarán vacíos. En el Apéndice D.1.

### 3.4.2 Enum

Esta clase consta de ser un servicio para nuestra aplicación. Nos ayuda en el manejo del enum TipoIncidente ya que este es utilizado en listas desplegables dentro de vistas como crear reporte o modificar reporte.

Las listas desplegables que se utilizaron dentro de la aplicación despliegan el nombre de cada uno de los enumeradores del enum TipoIncidente, para conseguir que esto suceda fue necesario el servicio de esta clase. De igual modo se ocupó para las vistas en donde se arrojan los resultados, para indicar cuál fue el enumerador ocupado para realizar la búsqueda y no muestra el enumerador como tal, sino, el nombre que se le dio al enumerador.

Esta clase consta de varios métodos, pero los más importantes son:

- Parse, recibe el nombre del enumerador y retorna un objeto de TipoIncidente con el enumerador seleccionado.
- GetDisplayValue, recibe un enumerador de TipoIncidente y retorna el nombre de ese enumerador.
- GetNames, retorna una lista de tipo String con todos los nombres de los enumeradores del enum TipoIncidente.

### 3.4.3 Visualización de reportes en un mapa (JavaScript)

Este JavaScript realiza el manejo del mapa en vistas como mis reportes o resultados de una búsqueda ya sea una búsqueda general o avanzada. Su objetivo es geolocalizar la ubicación del dispositivo que está usando nuestra aplicación y pintar

marcadores de diferentes colores y leyendas dependiendo del tipo de delito que se desee presentar en el mapa.

Para saber en que posición se tiene que colocar el marcador se lee de la vista cajas de texto escondidas que contienen latitud, longitud y tipo de incidente y simplemente con un switch dependiendo del tipo de incidente se sabe de que color tiene que ser el marcador, en que posición debe estar y que es lo que debe decir la leyenda de ese marcador.

Para realizar que los marcadores tuvieran un icono dentro, un color especifico y una leyenda debajo del como los de la Figura 30, se utilizó un plugin de Leaflet, este plugin consigue hacer que los marcadores sean de colores y con un icono de Awesome Font, se puede encontrar como Leaflet AwesomeMarkers, y es de código abierto. Para las leyendas que aparecen debajo de los marcadores únicamente se utilizó el JavaScript de Leaflet que nos proporciona este tipo de funcionalidades.

El JavaScript utilizado para estas funcionalidades en el mapa se encuentra en el Apéndice D.3.

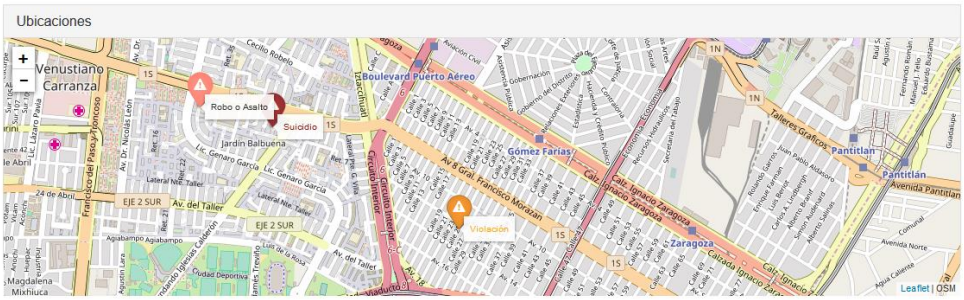


Figura 30. Mapa con reportes geolocalizados en un mapa.

## 3.4.4 Crear reporte

La implementación de esta vista consistió en el uso de datepicker el cual despliega un calendario como el de la Figura 31, una lista desplegable de tipos de incidentes delictivos, ver Figura 32, y un mapa para realizar el registro de la ubicación del incidente. La implementacion de esta vista se encuentra en el Apéndice D.4.

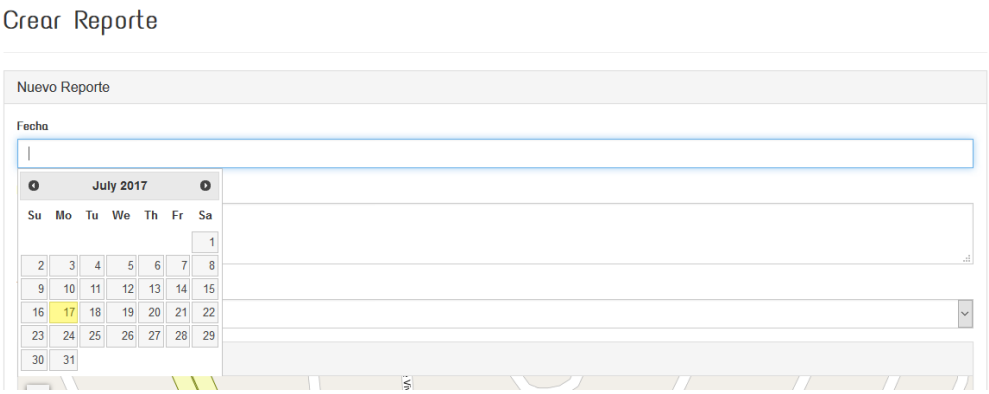


Figura 31. Vista crear reporte, con calendario desplegado.

Datepicker es una librería de bootstrap que nos da la posibilidad de desplegar un calendario, seleccionar una fecha y esta fecha darle un formato y escribirla en algun lugar de nuestro Html.

Para la implentación del calendario desplegale fue necesario agregar la siguiente linea de codigo:

```
$("#fecha").datepicker({
                dateFormat: 'yy-mm-dd'
});
```

Nuestra caja de texto donde incorporaremos el calendario tiene un Id = “fecha”, y con ayuda de JQuery cosneguimos hacer que este se despliegue y no solo eso sino que se le da un formato para su manejo. El formato fue requerido debido a que simplicaba las incersiones en la base de datos.

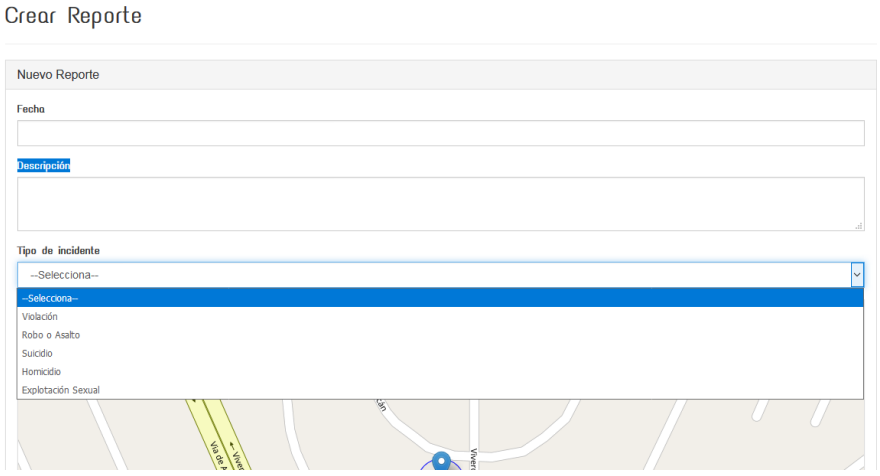


Figura 32. Vista crear reporte con dropdown tipo de incidente desplegado.

Para la implementación de la lista desplegable se utilizo un EnumDropDownListFor, que automaticamente el html helper nos entrega una lista desplegable con los nombres de los enumerables. El metodo EnumDropDownListFor

tiene sobrecarga y en uno de estos metodos se tiene la posibilidad de agregar un valor inicial para mostrar en la lista desplegable, en nuestro caso se utilizo la cadena “— Selecciona--”.

El mapa utiliza el JavaScript implementado en la sección 3.3.12 de este reporte, donde se tiene un manejo de eventos clic en el mapa y una geolocalización por defecto, ver Figura 33, la geolocalización se realiza mediante geolocation api, servicio proporcionado por Leaflet.

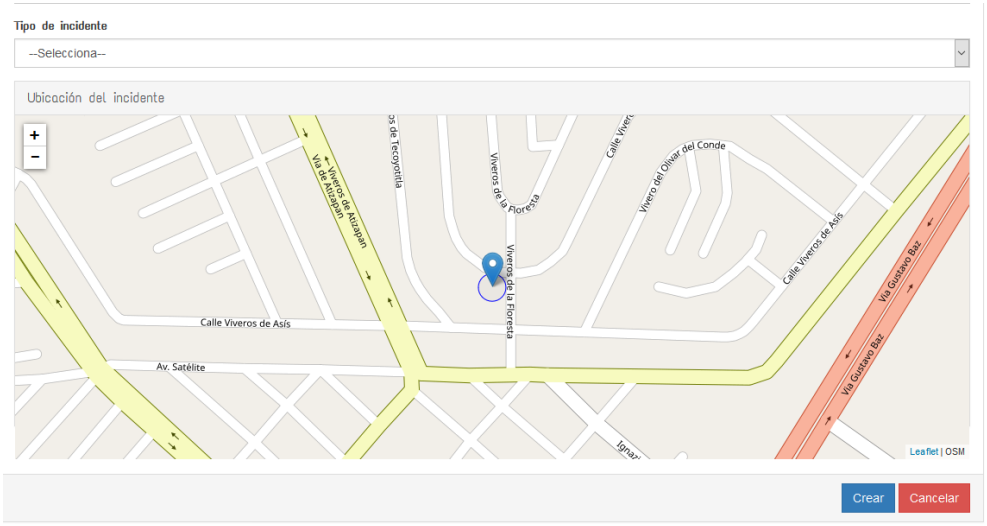


Figura 33. Mapa para registrar una ubicación de un nuevo reporte.

El objetivo de este mapa es colocar el marcador en la posición donde sucedió el incidente y así poder realizar un registro de un nuevo incidente con una localización y posteriormente realizar consultas cobre ubicación de reportes y obtener los más cercanos dentro de un radio.

## 3.4.5 Detalles reporte

Cualquier usuario aun no esté autentificado puede acceder a esta vista para visualizar con más detalle un reporte. Como se muestra en la Figura 34, se tiene una liga del perfil que realizó la publicación del incidente, una fecha, un tipo de delito, una dirección, la descripción del incidente y la ubicación. Esta vista podrás apreciarla en el Apéndice D.5.

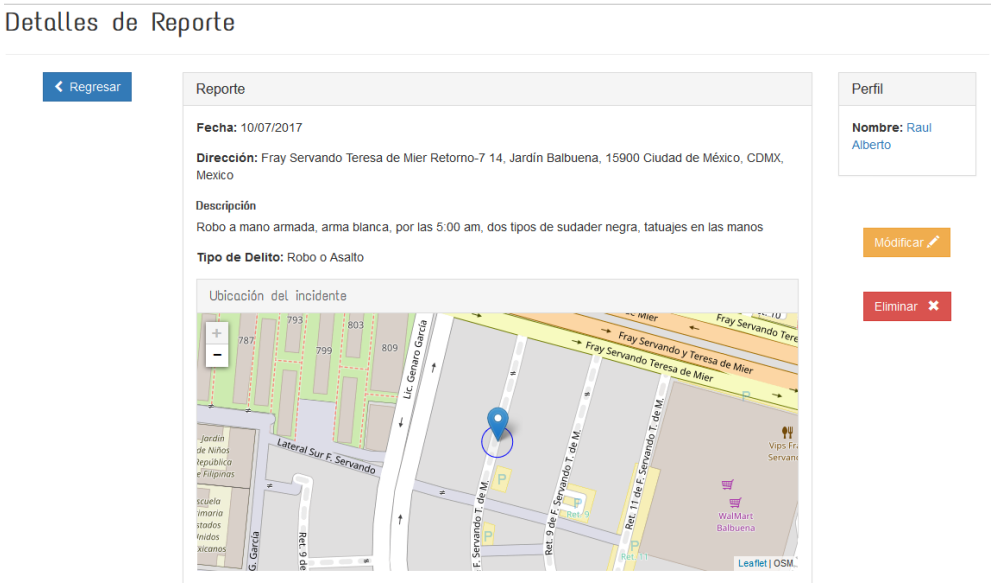


Figura 34. Vista detalles de un reporte.

Únicamente los usuarios que están viendo sus propios reportes podrán visulizar los botones de modificar o eliminar un reporte, en caso contrario estos serán escondidos en la capa de presentación y desde el método que realiza el Get para entregar las vista modificar reporte, o el método Post que realiza la eliminación harán una validación de si el Id del perfil que está activo es el mismo que el Id del perfil que realizo la publicación. Los métodos se encuentran en la clase controladora de reportes en el Apéndice D.1.

## 3.4.6 Modifiicar reporte

Modificar reporte, es una vista a la cual se le transfiere como modelo de la vista, un objeto de tipo reporte, este reporte es transferido desde la vista detalles. Esta vista se encuentra en el Apéndice D.6.

El flujo que se sigue para que, en esta vista se use como modelo el objeto reporte, es:

- Se da clic en modificar reporte, este ejecuta el método Get modificar y se pasa como argumento el Id del reporte.
- Selecciona el reporte y se genera un objeto completo de un reporte, incluyendo ubicación.
- Se verifica si el perfil que está tratando de acceder a la vista modificar es el mismo que esta autentificado dentro la aplicación.
- En caso de que sea cierto retornará la vista con el objeto reporte como argumento.
- Dentro de la vista modificar se debe especificar que se utiliza el modelo Reporte como estructura de la vista, esto se hace colocando la siguiente instrucción en la parte superior de la vista:

@model Proyecto_Integracion.Models.Reporte

Esta vista tiene la misma estructura que la vista crear reporte ya que, se tiene una fecha, una descripción, un tipo de incidente y una ubicación por modificar, ver Figura 35. La ubicación es modificable debido al JavaScript de la sección 3.3.16, pág. 44.

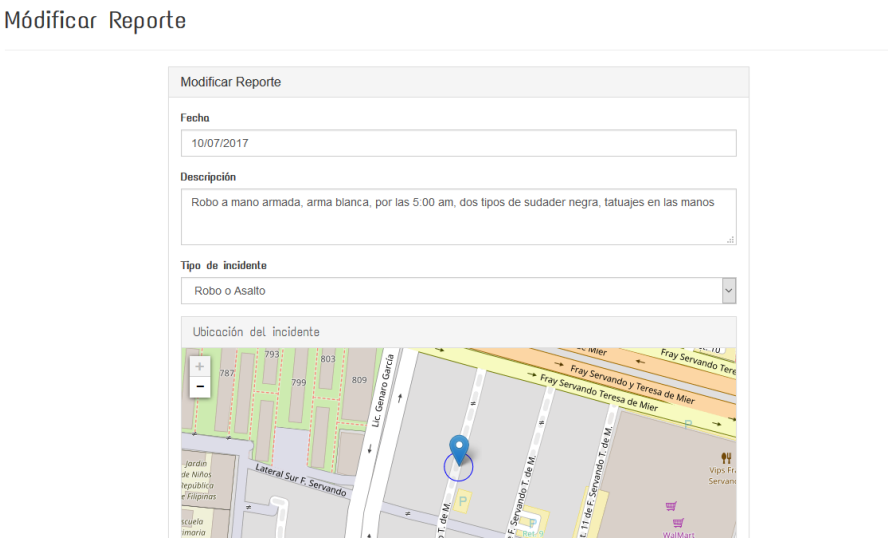


Figura 35. Vista modificar reporte con ubicación modificable en un mapa.

## 3.4.7 Eliminar reporte

El metodo eliminar dentro de la clase controladora de reportes se ejecuta una vez que se da clic en el botón eliminar dentro de detalles perfil. Como se mencionó en la vista detalles de un reporte, este botón únicamente se mostrará cuando el perfil autentificado sea el mismo que realizó el reporte.

Al hacer clic para eliminar un reporte, se desplegará el modal, ver la Figura 36, donde podrás confirmar la eliminación del reporte mediante la ejecución del metodo eliminar dentro de la clase reporte. La implementación del modal se encuentra junto con la vista detalles de reporte en el Apéndice D.5.

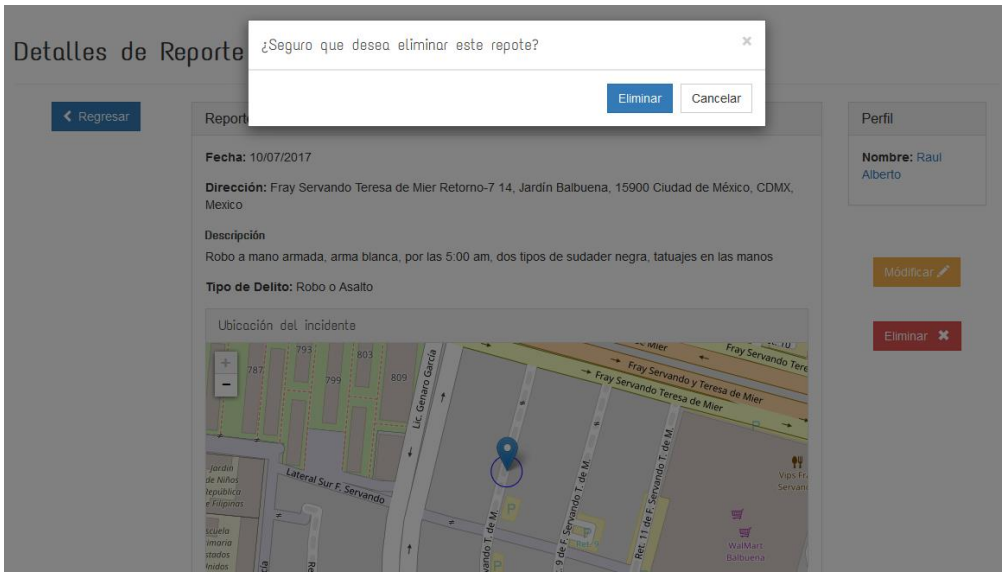


Figura 36. Modal eliminar reporte desplegado dentro de vista detalles de un reporte.

## 3.4.8 Búsqueda general (Búsqueda en Layout)

En el Header del Layout se encuentra un buscador (Figura 15), el buscador principal de nuestra aplicación. Este cuenta con cinco diferentes filtros y cada uno de estos es soportado por un método diferente dentro de la clase estantería y por ende un procedimiento almacenado diferente a nivel base de datos.

Para la implementación de este método fue requerido dividir, en vistas parciales cada uno de los filtros. En todas las vistas parciales se hace ejecución del mismo método, pero este recibe diferente filtro, indicando que tipo de búsqueda debe realizar. Por ejemplo, para realizar búsqueda por dirección se debe colocar el filtro con valor de uno.

Los resultados obtenidos del método ejecutado dentro de la estantería, es usado como modelo dentro de los resultados de la búsqueda. Para conseguir que sea modelo de la vista es necesario agregar la siguiente instrucción en la parte superior:

@model IEnumerable<Proyecto_Integracion.Models.Reporte>

Por medio de un foreach, conseguimos recorrer todo el arreglo, que es usado como modelo, y así conseguimos mostrar los resultados paginados de diez en diez.

La vista donde se muestran los resultados de esta búsqueda se encuentra en el Apéndice D.7.

### *3.4.8.1 Lista de resultados de búsqueda general*

Desde el buscador de la Figura 15, se realiza una búsqueda por cualquiera de los filtros que este buscador contiene, los resultados son consultados desde la vista resultados de búsqueda general. En si, los resultados de busqueda general y busqueda avanzada son mostrados en listas similares como se muestra en la Figura 37.

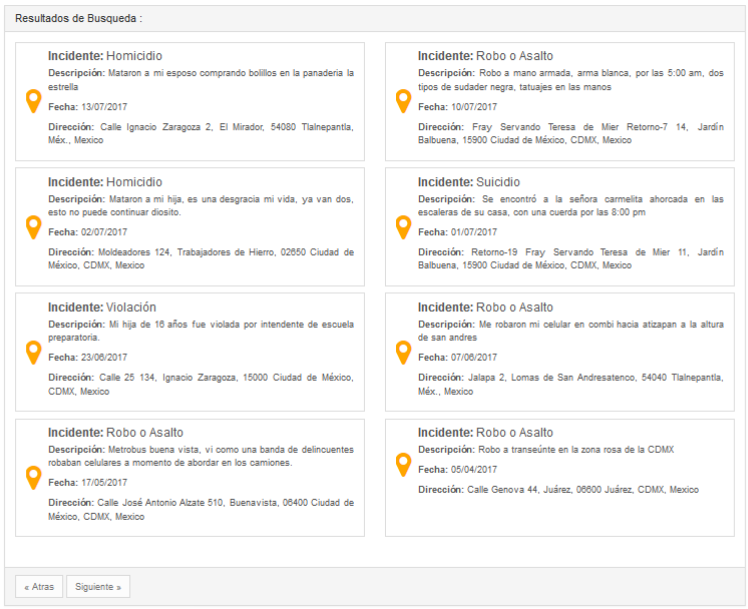


Figura 37. Resultados de una búsqueda desde el Layout.

Para la visualización de reportes en modo de lista, se realiza un foreach que recorre todo el arreglo de reportes, cada reporte es generado con los mismos atributos y mismo diseño. Es por ello que la estructura de los reportes son iguales solo que el contenido cambia.

## 3.4.9 Búsqueda avanzada

Desde el menú se puede accesar a la busquda avanzada, que, consiste en ser una busqueda mas especifica. Como se muestra en la Figura 38, se tiene varios parametros para realizar la busqueda, en caso de que no se llene uno de estos campos, no se tomara encuenta para realizar la busqud, eso significa que se pueden realizar las busquedas con la combinacion de parametros que el usuario guste.

Siempre se ejecuta el mismo metodo Post (Busqueda General) Para hacer las busquedas independentemente de la combinación de parametros, eso si, si se ejecuta el metodo y no se llenan algunos campos se mandaran como campos vacias, dentro del metodo se convertiran en nulos y es entonces que ejecuta el procedimiento almacenado pero con argumentos nulos. Esto tiene desventajas debido a que si se encontrase algun atributo en una tabla donde el campo es nulo lo retornaría, es por

ello que se requiere de una validación para el registro de reportes y en la creación de cuentas no se envien campos vacios.

Cuando el usuario esta autentificado se muestra la opción de realiza la busqueda por la ubicación del perfil, a unos 40 Km de radio. Esto es conseguido gracias a que se tienen la implementacion del metodo busqueda avanzada con una ubicación, si no hay cuenta autentifiada, estos valores de ubicación como latitud o longitud se mandaran nulos y el metodo no intentará hacer la busqueda porque no se checó el checkbox de “buscar por mi ubicación”. Estos metodos que realizan las busquedas avazadas ejecutan procedimientos alamcenados del Apéndice B.2.9.

La vista busqueda avanzada se encuentra en el Apéndice D.8.

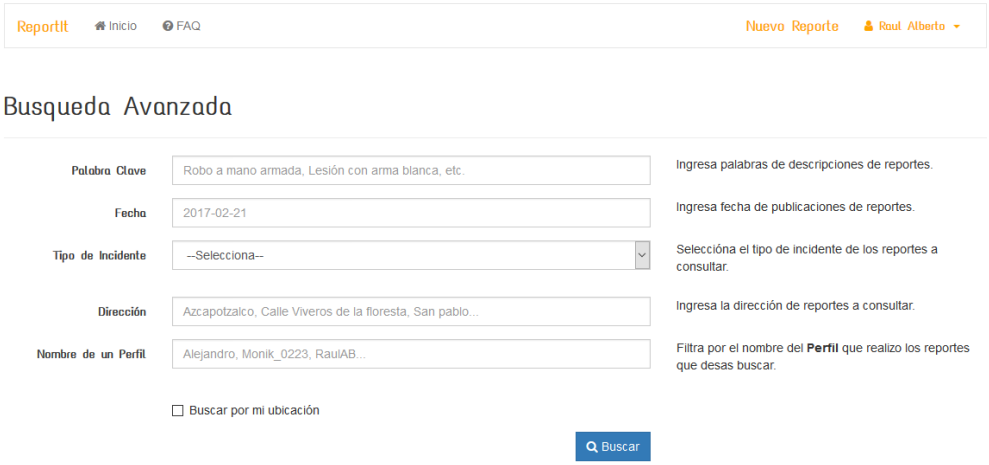


Figura 38. Vista para realizar una búsqueda avanzada.

## 3.4.10 Visualización de resultados de una búsqueda en un mapa

Cada marcador tiene su color y leyenda especifico para cada tipo de incidente, ver Figura 39. Para conseguir realizar esto, desde el JavaScript se realiza un switch con el tipo de incidente para saber que color y leyenda debe llevar el marcador. Para geolocalizar el marcador, desde las listas de resultados se extrae información que se

encuentra escondida, como latitud, longitud y tipo de incidente de cada uno de los reportes.

Unicamente se mostrarán como máximo diez marcadores ya que, la lista de resultados se encuentra paginada de diez en diez y la información que se ocupa para agregar marcadores al mapa es extraida de la lista.

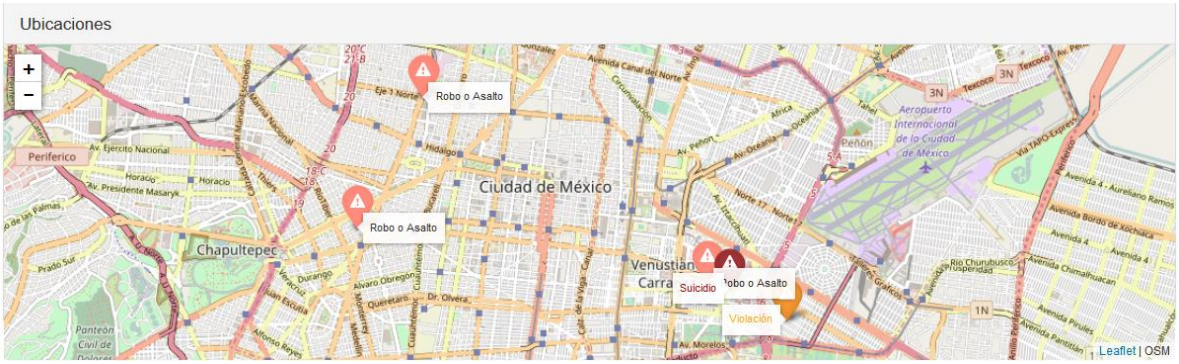


Figura 39. Resultados geolocalizados en un mapa de una búsqueda avanzada por nombre de un perfil.

# CAPÍTULO 4. Resultados

El objetivo de este capítulo es mostrar el funcionamiento correcto de nuestra aplicación en las diferentes etapas de desarrollo con respecto a los objetivos. Se presentan tres pruebas para cada módulo de la aplicación.

## 4.1 Pruebas

En esta sección se prentan las pruebas de cada módulo, mostrando flujos dentro de la aplicación para conseguir realizar las gestiones.

### 4.1.1 Prueba del módulo de gestión de perfiles

Creación de una cuenta, ver Figura 40.

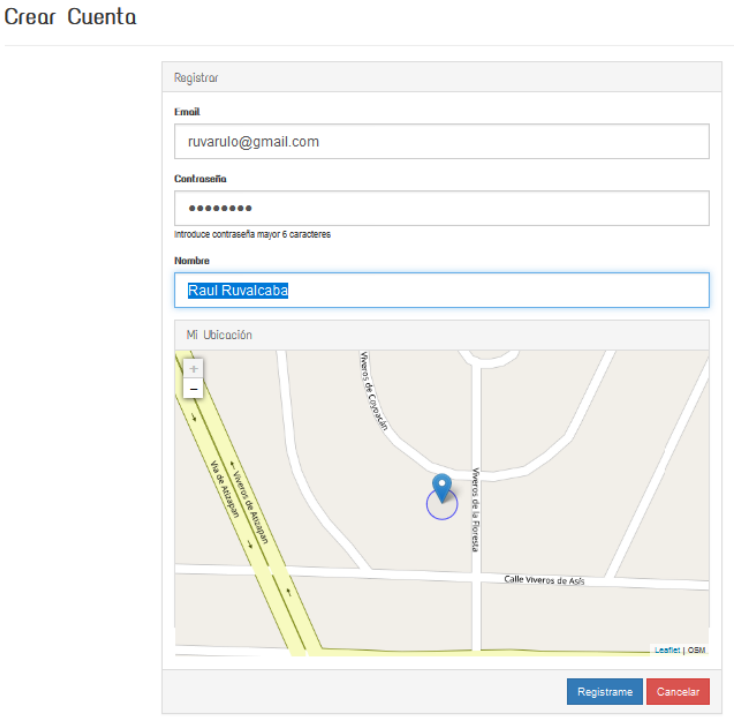


Figura 40. Creación de una cuenta.

Una vez creada la cuenta, este nos retorna al inicio, para consultar nuestro perfil, debemos dar clic en el menú de opciones de un perfil, y seleccionar la opción Mi perfil, El menú de opiones aparace en la Figura 41.


Figura 41. Menú de un perfil.

La vista retornada al dar clic en la opción de Mi Perfil es la vista detalles de perfil, esta la podemos apreciar en la Figura

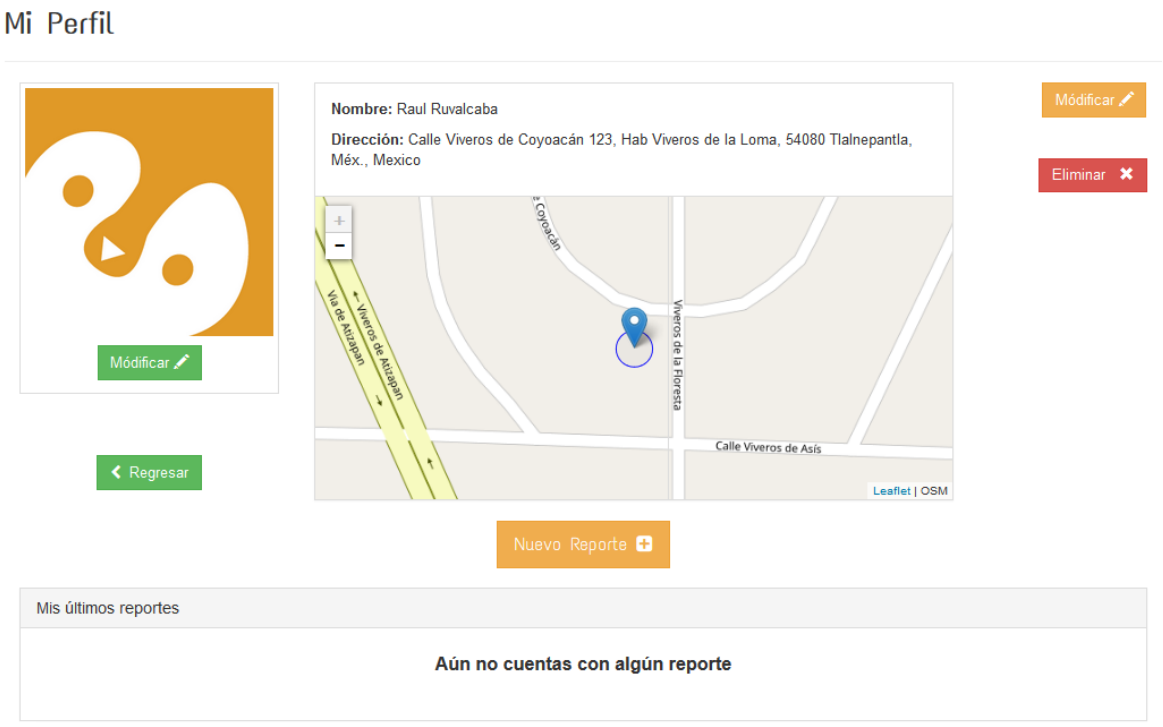


Figura 42. Vista detalles del perfil recien creado.

Desde esta vista podemos realizar las dos partes que nos faltan para tener la gestión completa de un perfil, la modificación y eliminación.

Para modificar el perfil recién creado, damos clic en el botón naranja modificar de la parte izquierda. Nos retornara a la vista modificar perfil, ver Figura 43.

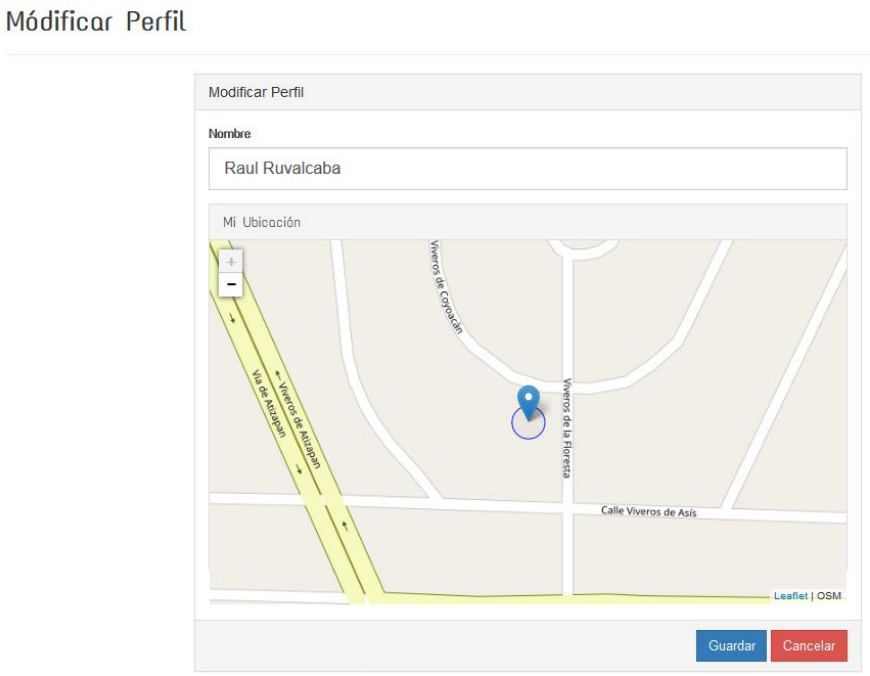


Figura 43. Vista modificar perfil.

Modificaremos el nombre del perfil por Raul Ruvalcaba2 y la ubicación por cualquier otro punto. El resultado de esta modificación se aprecia en la Figura 44.

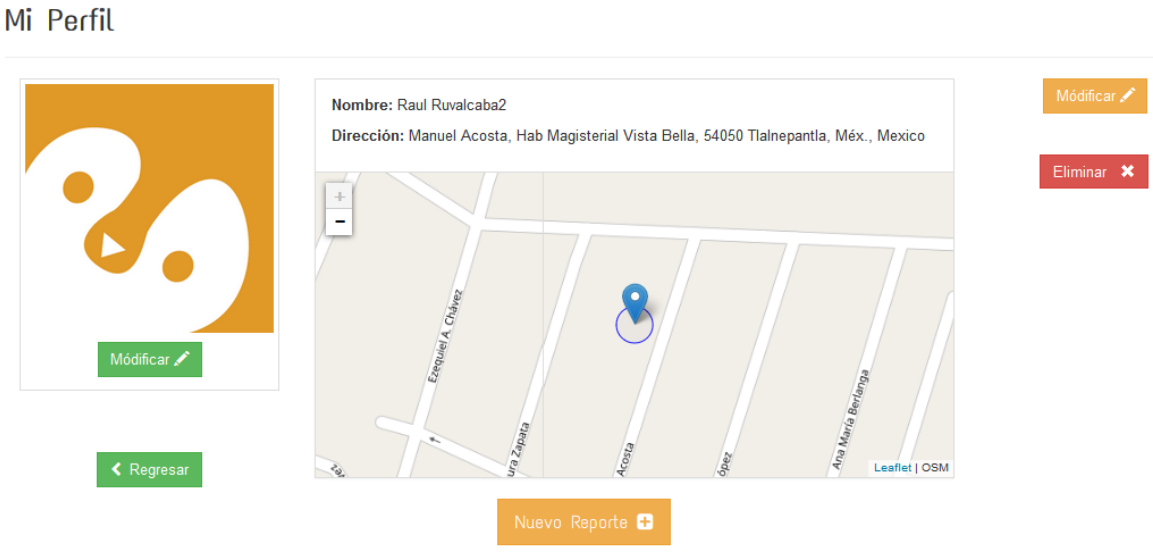


Figura 44. Vista detalles de perfil modificada.

Ahora damos clic en el botón eliminar para visualizar el modal que pregunta si estamos seguros de eliminar nuestra cuenta, ver Figura 45. La eliminaremos y la volveremos a crear para la siguiente sección de gestión de un reporte.

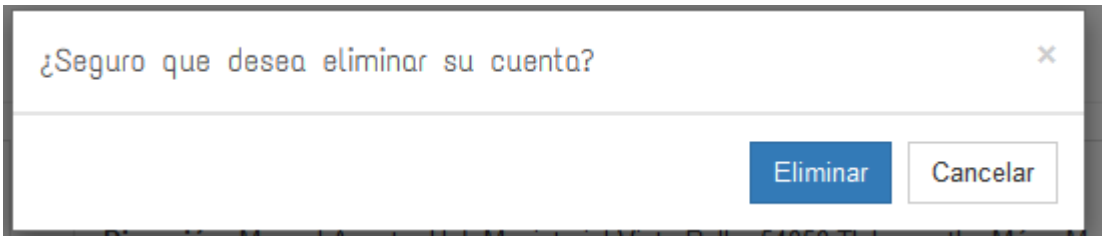


Figura 45. Modal eliminar cuenta.

Damos clic en eliminar y automáticamente nos retorna a la vista de incio.

Con esto acabamos esta sección de pruebas para la gestión de un perfil, comprobando el funcionamiento de todas las etapas que esta incluye.

### 4.1.2 Pruebas del módulo de gestión de reportes

Creamos de nuevo la cuenta de Raul Ruvalcaba y nos dirigimos a crear un nuevo reporte, este se encuentra en la barra menú del lado izquierdo, ver Figura 46.

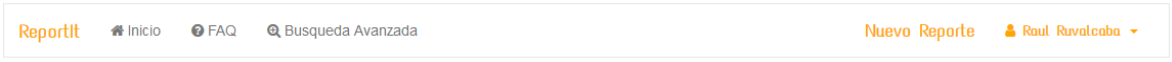


Figura 46. Menú de navegación usuario Raul Ruvalcaba autentificado.

Creamos el reporte como en la Figura 47.

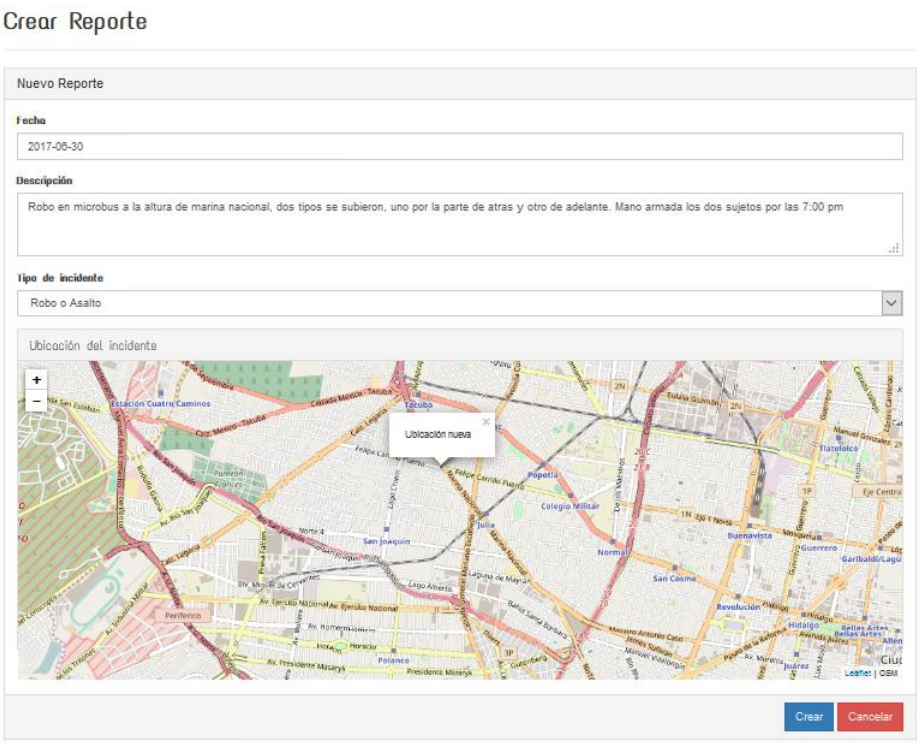


Figura 47. Creación de un reporte por parte del usuario Raul Ruvalcaba.

Después de crear el reporte nos regresa a nuestro perfil donde podemos ver los últimos cuatro reportes que hemos realizado, en nuestro caso, como nuestra cuenta es nueva, es nuestro primer reporte, ver Figura 48.

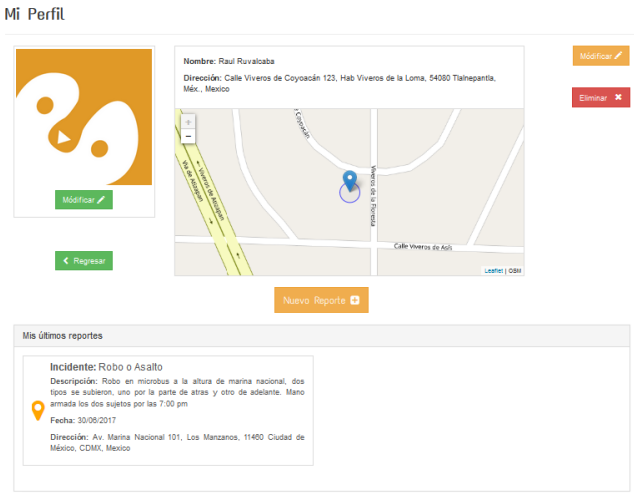


Figura 48. Cuenta recién creada con el nuevo reporte realizado.

Seleccionamos nuestro reporte desde nuestro perfil, y entramos a visualizar los detalles del reporte. Desde esta página podemos eliminar o modificar nuestro reporte, ver Figura 49.

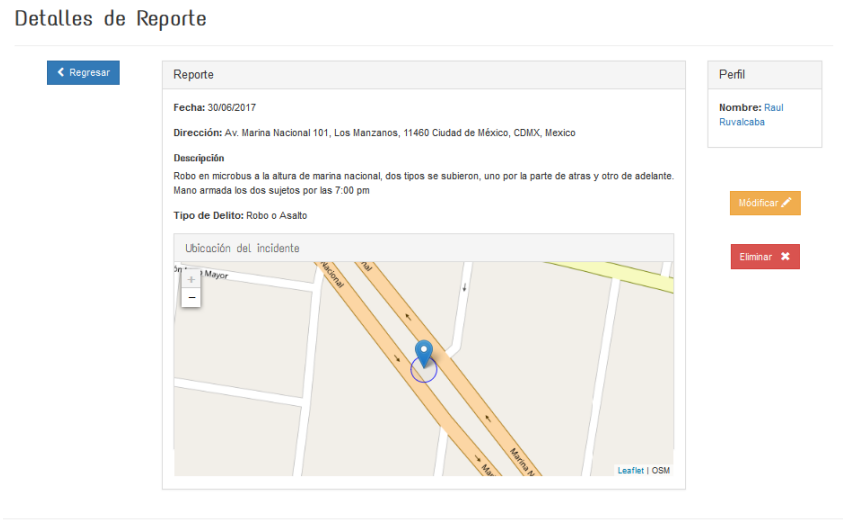


Figura 49. Detalles de reporte recientemente creado.

Ahora modificaremos el reporte con una fecha nueva y una ubicación diferente, verifica la modificación de los campos en la Figura 50.

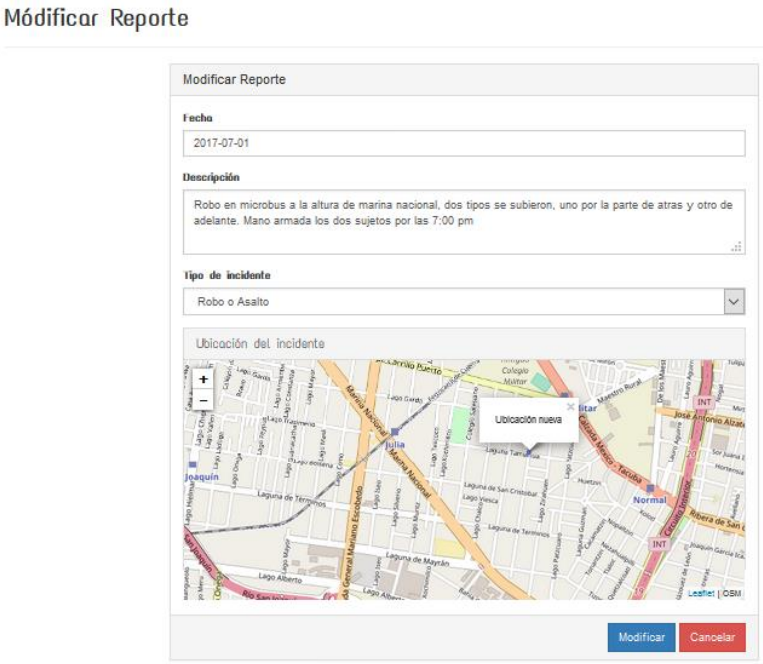


Figura 50. Modificar último reporte del usuario Raul Ruvalcaba.

Damos clic en Modificar y visualizaremos la modificación en la vista detalles del reporte como la Figura 51.

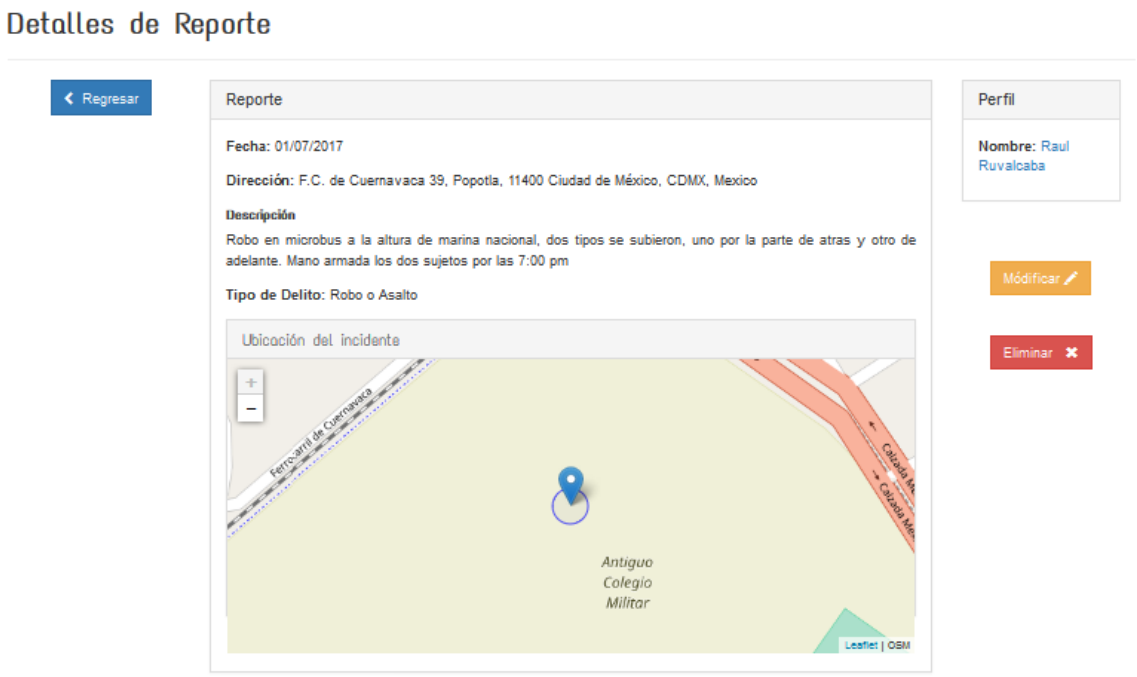


Figura 51. Reporte modificado en fecha y ubicación.

La siguiente etapa es la etapa de las consultas de reportes. Nos dedicaremos a buscar nuestro reporte desde dos buscadores, primero el buscador general y segundo el buscador avanzado.

Realizamos una búsqueda en nuestro buscador general por todos los atributos de un reporte, ver Figura 52.


Figura 52. Buscador general con opción de búsqueda en todos los campos. Se busca el término microbús.

Los resultados de la búsqueda solo es nuestro reporte creado anteriormente ya que no hay otro donde se haya escrito el término microbús, ver los resultados de la búsqueda en la Figura 53.

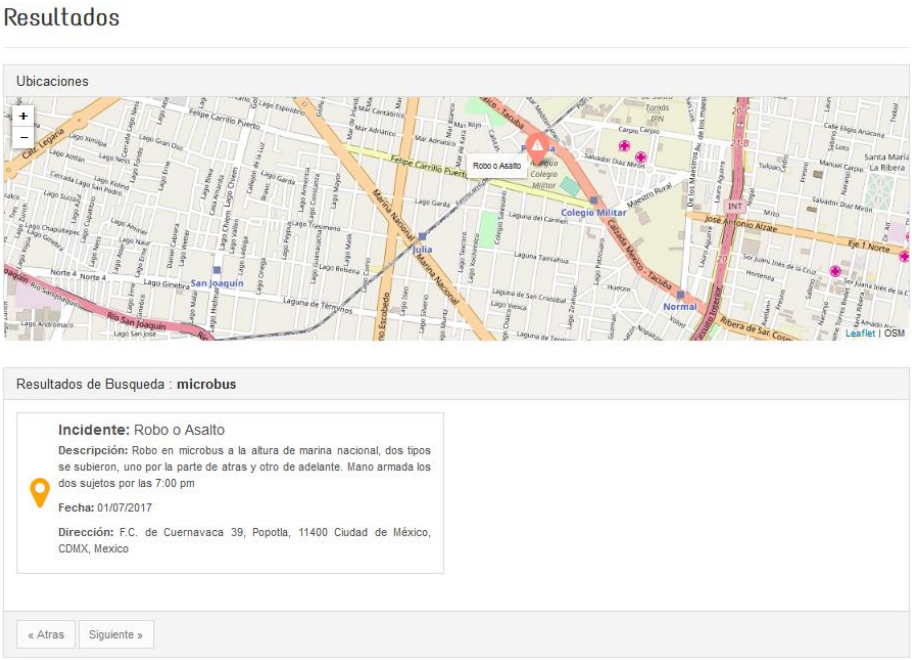


Figura 53. Resultados de búsqueda por todo con el término "microbús".

Realizamos una búsqueda por dirección buscando la dirección “CDMX”, ver Figura 54.


Figura 54. Búsqueda por dirección "CDMX".

Resultados de la búsqueda son mostrados en la siguiente Figura 55.

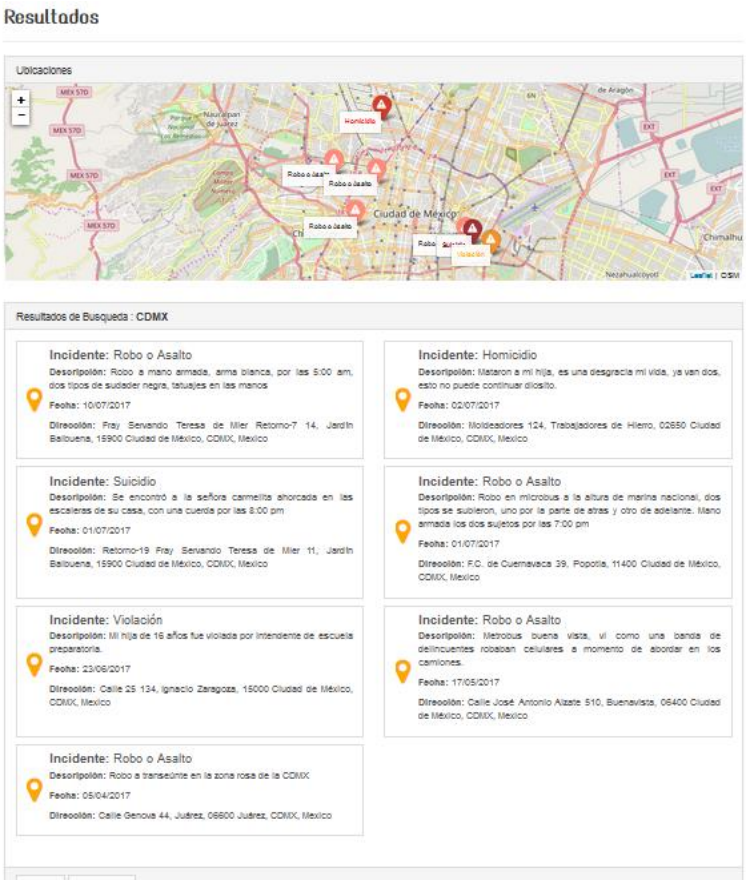


Figura 55. Resultados de la búsqueda por la dirección "CDMX".

Realizamos la búsqueda por la fecha que colocamos en el reporte creado al principio de este capítulo 2017-07-01, ver Figura 56.

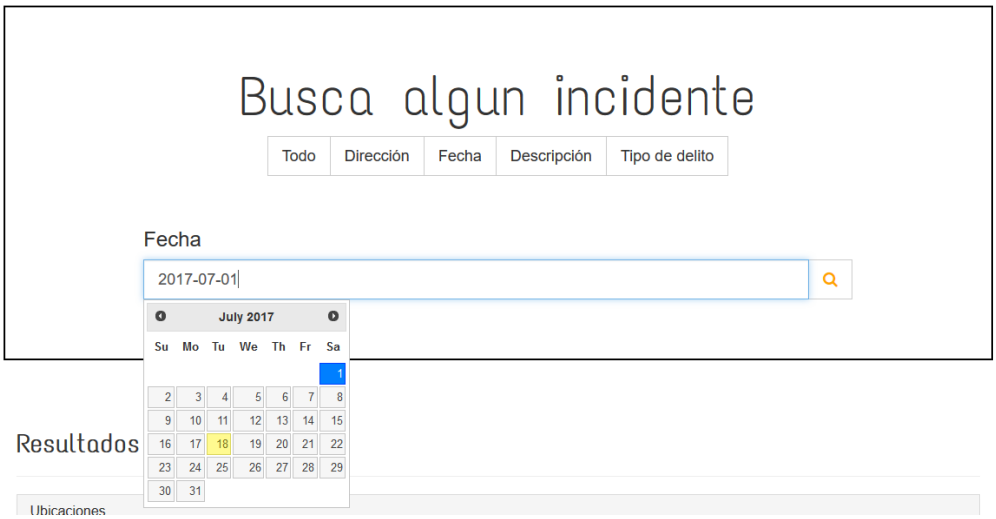


Figura 56. Búsqueda por la fecha "2017-07-01".

Los resultados de la búsqueda los apreciamos en la Figura 57.

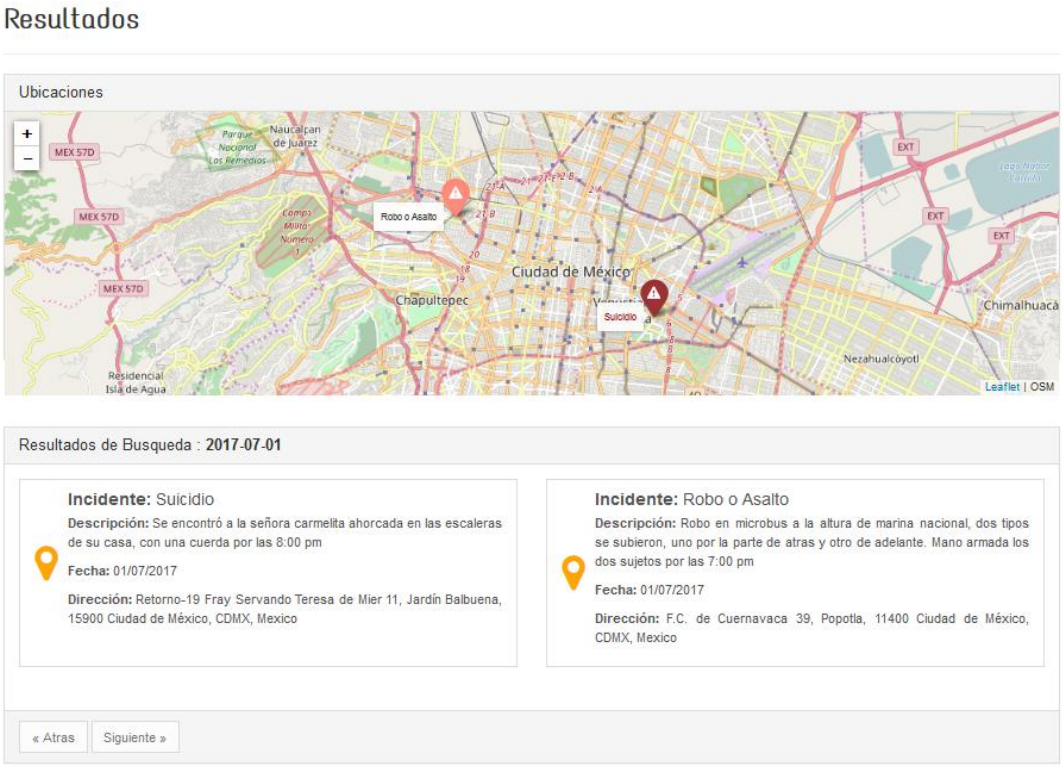


Figura 57. Resultados de la búsqueda por fecha "2017-07-01"

Se encontraron dos reportes que se realizaron ese mismo día, uno es un suicidio y el otro un robo.

Realizamos una búsqueda por palabras en la descripción “mano armada”, visualiza la búsqueda en la Figura 58.


Figura 58. Búsqueda en descripción de un reporte por el termino "mano armada".

Resultados de la búsquedason mostrados en la Figura 59.

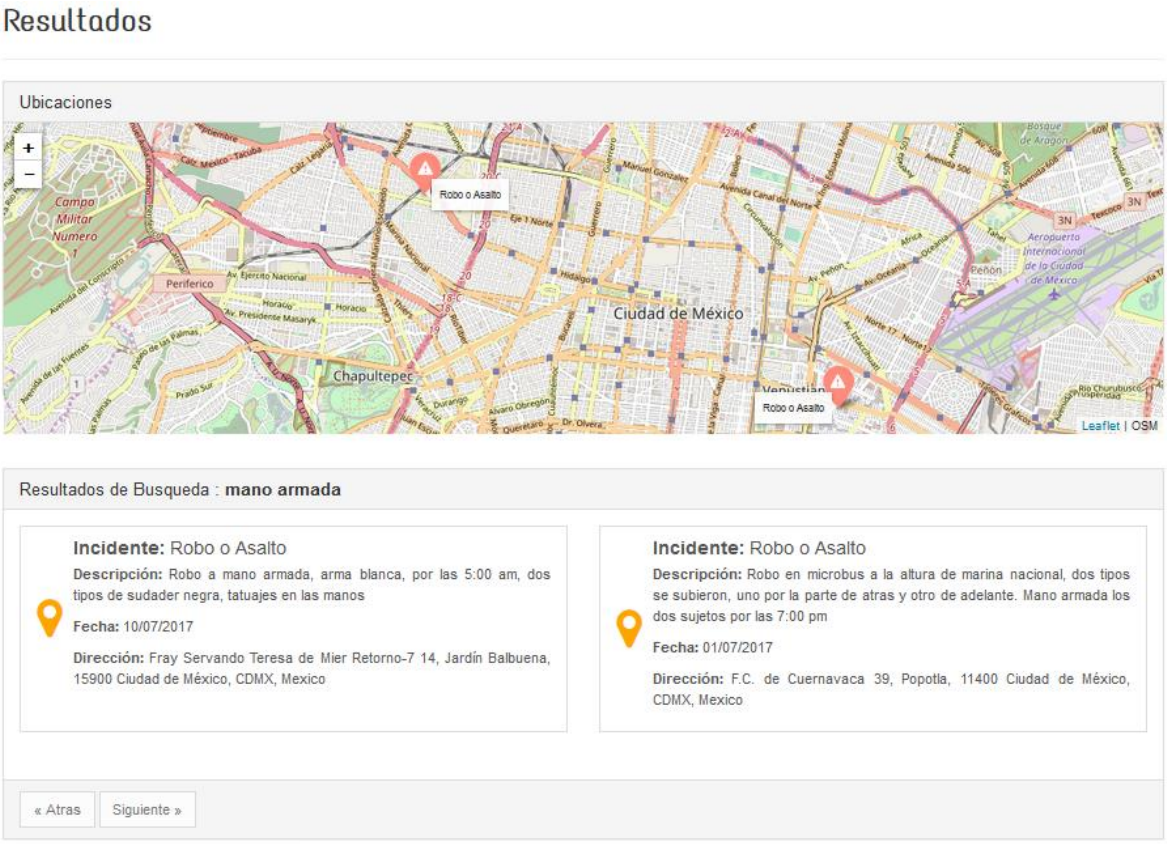


Figura 59. Resultados de la búsqueda por "mano armada" en la descripcón de los reportes.

Realizamos una búsqueda por tipo de delito y buscamos todos los robos o asaltos que se han cometido, visualiza la búsqueda en la Figura 60.


Figura 60. Búsqueda por tipo de incidente robo o asalto.

Resultados de búsqueda por tipo de incidente robo o asalto en la Figura 61.

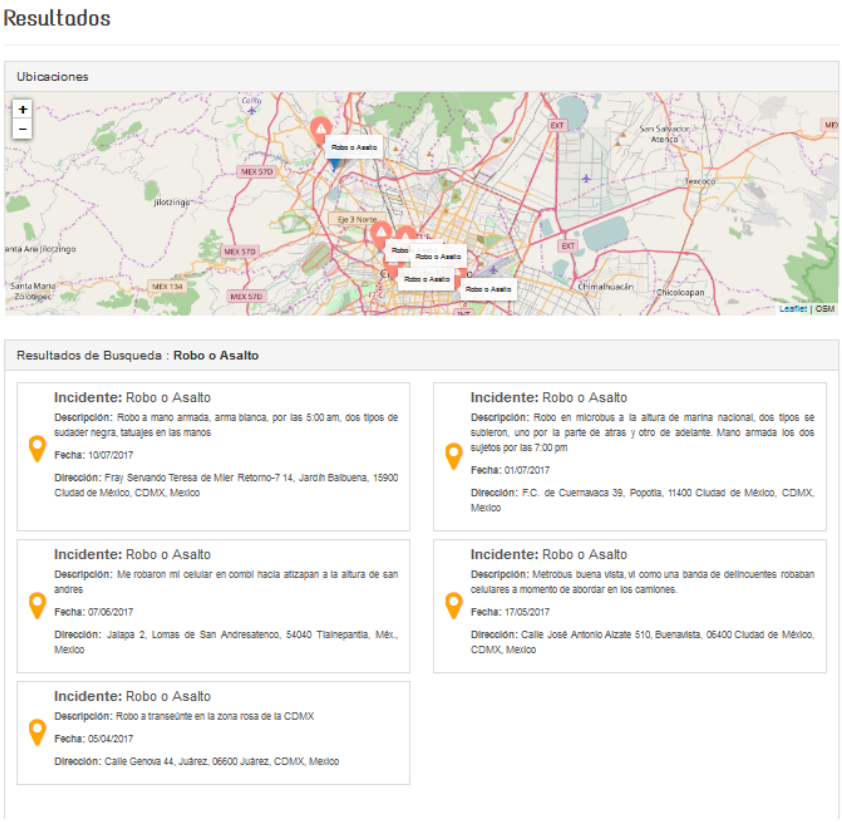


Figura 61. Resultados de búsqueda por tipo de incidente.

Realizamos una búsqueda avanzada buscando por nuestro nombre del perfil que recién creamos para estas pruebas, tipo de incidente violación y por la ubicación del perfil, ver Figura 62.

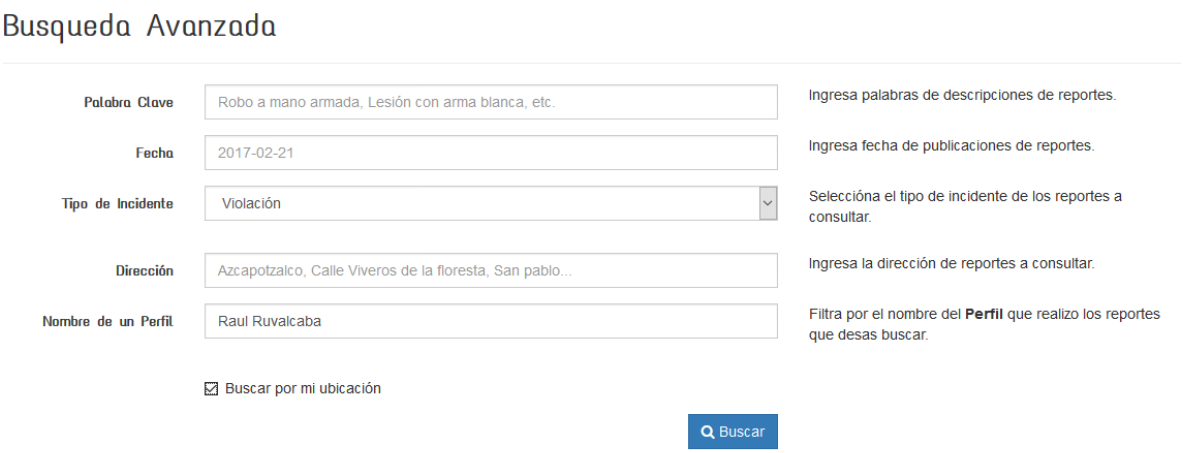


Figura 62. Búsqueda avanzada por nombre de un perfil, un tipo de incidente y cerca de la ubicación del perfil.

Resultados de la búsqueda avanzada en la Figura 63.

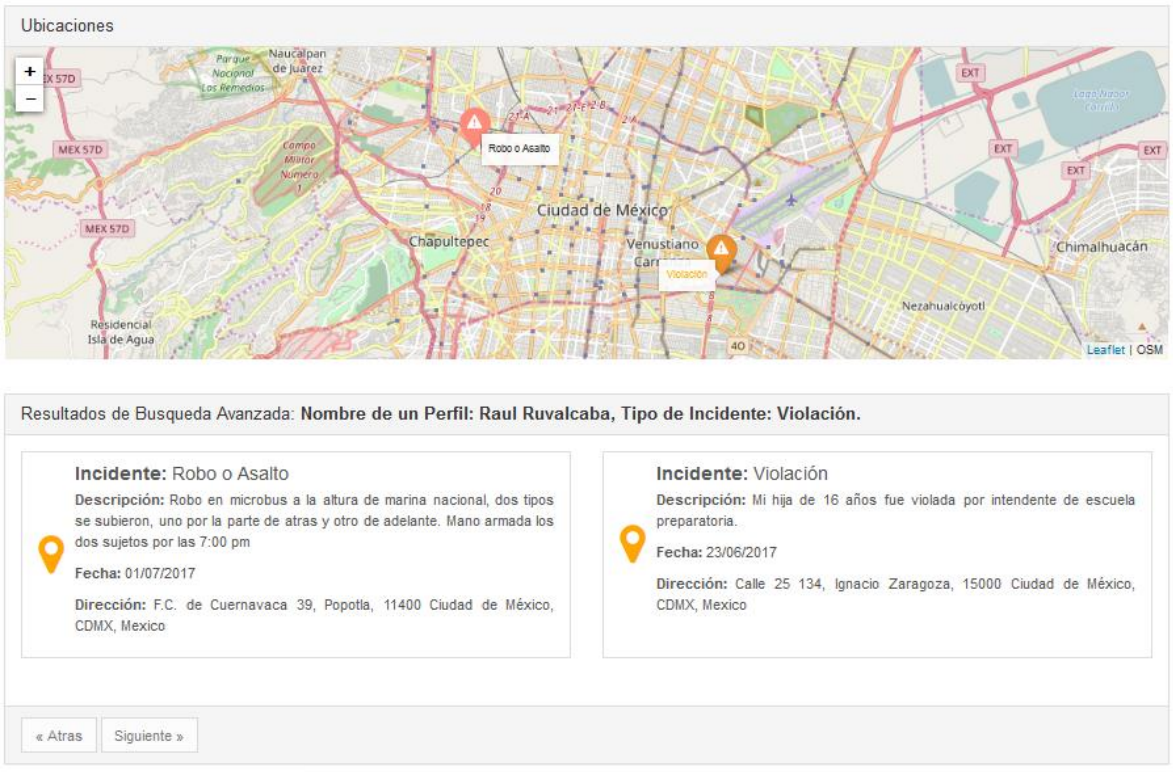


Figura 63. Resultados de la búsqueda avanzada por ubicación

## 4.2 Análisis y Discusión de resultados

En esta sección se analizarán los resultados más específicos dentro de la aplicación. Algunos de estos resultados no están contemplados en las etapas del proyecto, pero sirvieron para dar una mejor satisfacción al cliente.

Envió de correo electrónico de bienvenida al momento de crear la cuenta de Raul Ruvalcaba ver Figura 64.


Figura 64. Correo recibido por parte de la aplicación, dando la bienvenida.

Cuando no recordamos nuestra contraseña podemos recuperarla dando nuestro correo electrónico en el modal desplegado como el de la Figura 65 y esperamos a que se genere una nueva contraseña y sea enviada a nuestro correo.

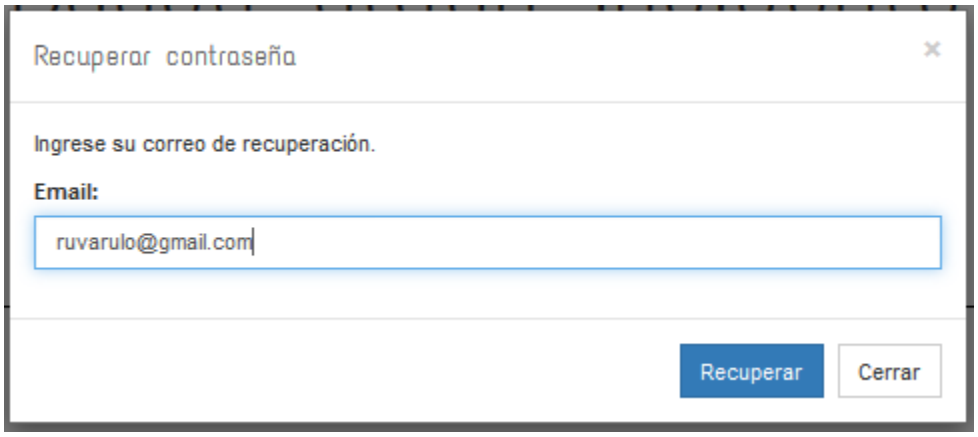


Figura 65. Modal para recuperación de contraseña.

Correo recibido con nueva contraseña en la Figura 66.


Figura 66. Correo recibido con nueva contraseña.

Podemos ingresar a la aplicación con la nueva contraseña generada y posteriormente cambiar de contraseña en la correspondiente vista, ver Figura 67.


Figura 67. Cambio de contraseña, pero las contraseñas no coinciden.

Cambio de contraseña con las contraseñas validadas, ver Figura 68. Ya no aparece el mensaje de “contraseñas no coinciden”.


Figura 68. Cambio de contraseña con nuevas contraseñas ingresadas y validadas.

Al dar clic en cambiar, se asignará la contraseña la cuenta.

Se puede visualizar los reportes que tiene un perfil, ingresando desde el menú desplegable del lado izquierdo y seleccionando mis reportes, ver menú desplegable en la Figura 69.


Figura 69. Menú de opciones para un perfil.

Como la cuenta de Raul Ruvalcaba solo cuenta con un reporte, este es el único que se visualizará en la página, vista mis reportes en la Figura 70.

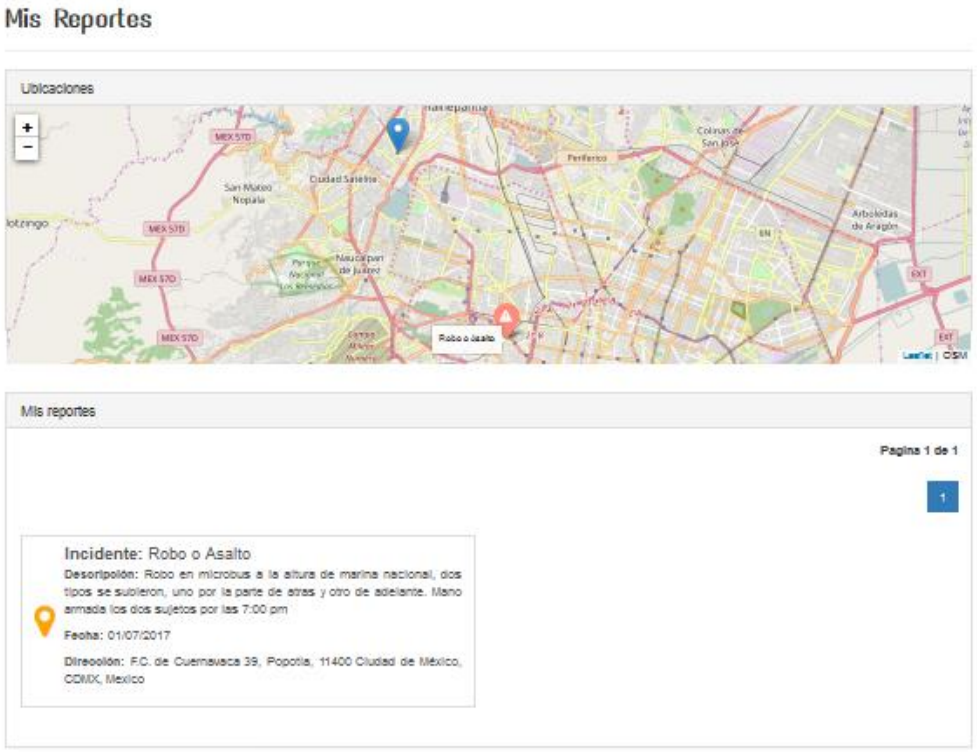


Figura 70. Reportes paginados y localizados en un mapa de un perfil.

Modificaremos la imagen del perfil recién creado en el modal desplegado, este modal lo apreciaras en la Figura 71.

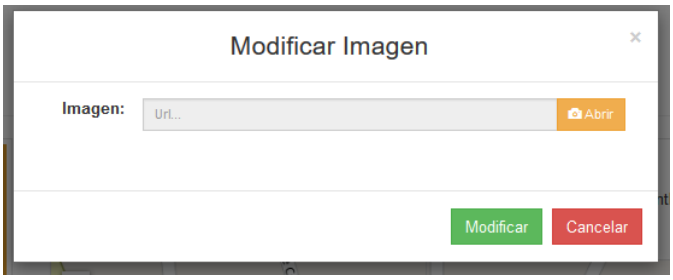


Figura 71. Modal desplegado para exportar la nueva imagen de perfil.

Seleccionamos la imagen que deseamos usar como foto de perfil, explorador de archivos en la Figura 72.

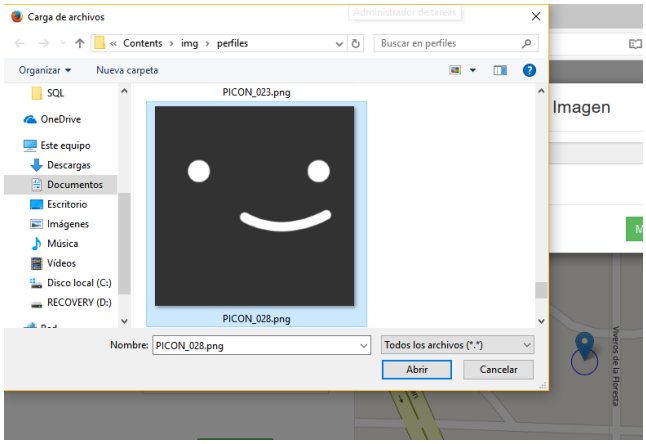


Figura 72. Explorador de archivos para búsqueda de imagen a exportar.

Le damos clic en modificar y es entonces que la imagen se descargará y se actualizará, ver Figura 73.

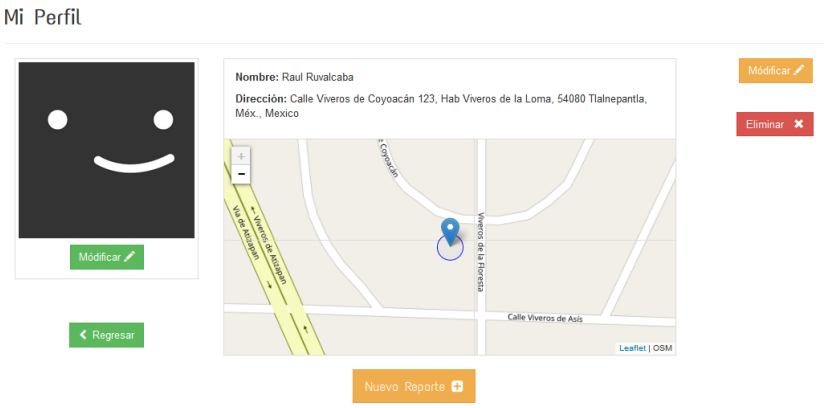


Figura 73. Imagen modificada del perfil de Raul Ruvalcaba.

Todas las pruebas realizadas en este capítulo fueron correctas ya que en todas ellas se cumplio su función de gestionar reportes y perfiles. Es necesario aclarar que para el funcionamiento de los

mapas en toda nuestra aplicación es requerido que nos permita accesar a su ubicación desde su dispositivo y que tenga internet en ese momento.

# Conclusiones

Para el desarrollo de este proyecto se presentaron problemas medianamente grandes ya que no se tienen los conocimientos previos de todas las herramientas, librerías y demás que se pueden utilizar y que facilitan el trabajo. Como por ejemplo, para realizar la lista desplegable de tipos de incidentes se dieron mil y unas vueltas por diferentes caminos para encontrar finalmente la sentencia de código que solucionaba el problema. Sin embargo, por falta de experiencia se perdieron varios días tratando de solucionarlo e incluso se programaron cosas que nunca funcionaron.

Se cumplió el objetivo de desarrollar una aplicación web para la gestión de reportes y perfiles con la ayuda de herramientas como asp .net. Se tuvieron muchos errores durante el desarrollo de la aplicación ya que no se tenían buenas bases en el lenguaje de programación y las herramientas de desarrollo que se iban a ocupar, pero finalmente se llegó a concluir obteniendo resultados satisfactorios.

Existen muchas cosas que aún se pueden mejorar dentro de la aplicación y se pudieron tratar durante este desarrollo, pero por falta de tiempo esto no se hizo. Las posibles mejoras que yo le haría a la aplicación serían:

- Validaciones de campos vacíos para creación de reportes.
- Validaciones de que el correo con el que se registra sea un correo real.
- Modales que muestren información o errores como," El correo o contraseña ingresados no son correctos".
- Haría búsquedas de reportes cercanos a un dispositivo sin tener que estar registrado. Que obtenga la ubicación del usuario.
- Pedir permisos para el uso de su ubicación.
- Agregaría más atributos a un perfil para una posible minería de datos, como sexo, edad, entre otros.
- Agregaría un buscador dentro de los mapas para localizar ubicaciones más rápido.

Y muchas cosas más, que si bien, ya solo son detalles más sencillos, aun así, son importantes y dan una mejor sensación al usar la aplicación y claro calidad de servicio.

El desarrollar este proyecto me deja con una sensación de que pude haber hecho más por él, haber explotado más mi conocimiento. Pero también está la otra parte del tiempo, tres meses para desarrollar una aplicación de este tipo es un tiempo justo y es por ello por lo que se quedan muchas cosas por mejorar en las próximas versiones.

# Referencias bibliográficas


[1] Novedades Observatorio Nacional Ciudadano, “Tendencia por entidad federativa”, [En línea]. Disponible en: Onc.org.mx, 2017.

[2] Diccionario de la Real Academia, “Edición del Tricentenario”, http://www.rae.es/, 2017.

[3] E. F. Rosas Coronado, “Sistema de información para el análisis de la seguridad social o pública a través de periódicos”, proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2014.

[4] J. Piskorski, H. Tanev, y P. Oezden, "Extracting Violent Event From On-Line News for Ontology Population", Research Center of the European Commission, vol. 1, pp. 2 - 4, 2006.

[5] K. Henriksen, JB. Battles, ES. Marks, et al., “The Impact of a Web-based Reporting System on the Collection of Medication Error Occurrence Data”, Department of HIM, University of Mississippi Medical Center, vol. 3, pp 1, 2005.

[6] Alertux, “Alertas Ciudadanas que salvan vidas", [En línea]. Disponible en: http://alertux.com/, 2017.

[7] Mi policía, “En defensa de la sociedad", [En línea]. Disponible en: https://play.google.com/store/apps, 2017.

[8] Mediatelecom, “Rezagada, adopción de smartphones en México”, [En línea]. Disponible en: http://mediatelecom.com.mx, 2017.

[9] El Economista, "Alphabet, la más valiosa del mercado y desplaza a Apple", Eleconomista.com.mx. [En línea]. Disponible: http://eleconomista.com.mx/mercados-estadisticas/2016/02/01/alphabet-mas-valiosa-mercado-desplaza-apple, 2017.

[10] Capital, "Tesla supera a Ford en valor de mercado", Revista Capital. [En línea]. Disponible: http://www.capital.cl/negocios/2017/04/04/138299/tesla-supera-a-ford-en-valor-de-mercado, 2017.

[11] Jordisan.net, "¿Qué es un 'framework'? | jordisan.net", jordisan.net, 2017. [En línea]. Disponible: http://jordisan.net/blog/2006/que-es-un-framework/, 2017.

[12] MSDN Microsoft, "Visual Studio IDE", Msdn.microsoft.com. [En línea]. Disponible: https://msdn.microsoft.com/library/dn762121(v=vs.140).aspx, 2017.

[13] J. Durán, "Visual Studio y la telemetría - Somos Binarios", Somos Binarios. [En línea]. Disponible: https://www.somosbinarios.es/visual-studio-y-la-telemetria/, 2017.

[14] MySQL, "MySQL :: Download Connector/Net", Dev.mysql.com. [En línea]. Disponible: https://dev.mysql.com/downloads/connector/net/6.0.html, 2017.

[15] .Net, ”Net”. [En línea]. Disponible: https://opbuildstorageprod.blob.core.windows.net/output-pdf-files/en-us/VS.core-docs/live.pdf, 2017.

[16] thoughtco, "Are JavaScript and Java the Same Language?", ThoughtCo. [En línea]. Disponible: https://www.thoughtco.com/what-is-javascript-2037921, 2017.

[17] R. Shannon, "What is HTML? | HyperText Markup Language explained", Yourhtmlsource.com. [En línea]. Disponible: http://www.yourhtmlsource.com/starthere/whatishtml.html, 2017.

[18] "Session Cookies, What is a Session Cookie Used for? - All about Cookies", Allaboutcookies.org. [En línea]. Disponible: http://www.allaboutcookies.org/cookies/session-cookies-used-for.html, 2017.

[19] "Cookies - What and Why?", Boston.co.uk. [En línea]. Disponible: https://www.boston.co.uk/info/cookies.aspx, 2017.

[20] "Leaflet — an open-source JavaScript library for interactive maps", Leafletjs.com. [En línea]. Disponible: http://Leafletjs.com/, 2017.

[21] "Documentation - Leaflet - a JavaScript library for interactive maps", Leafletjs.com. [En línea]. Disponible: http://Leafletjs.com/reference-1.1.0.html, 2017.

# Apéndice A. Modelo de clases

## **Apéndice A.1**: Clase Conexión con la base de batos

```
class DB
    {
        private static string connStr =
"SERVER=localhost;DATABASE=proyecto_integracion_api;UID=proyecto_integracion_user;PASSWORD=qwer12345
678";


        public static DataSet GetDataSet(MySqlCommand command)
        {
            var ds = new DataSet();
            using (var conn = new MySqlConnection(connStr))
            {
                conn.Open();
                command.Connection = conn;
                var sqlda = new MySqlDataAdapter(command);
                sqlda.Fill(ds);
                conn.Close();
            }
            return ds;
        }

        public static int QueryCommand(MySqlCommand command)
        {
            int success;
            using (var conn = new MySqlConnection(connStr))
            {
                conn.Open();
                command.Connection = conn;
                success = command.ExecuteNonQuery();
                conn.Close();
            }

            return success;
        }
    }
```

## **Apéndice A.2**: Enum Tipos de Incidentes

```
    public enum TipoIncidente
    {
        [Display(Name = "Homicidio")]
        Homicidio = 1,

        [Display(Name = "Suicidio")]
        Suicidio = 2,

        [Display(Name = "Robo o Asalto")]
        RoboAsalto = 3,

        [Display(Name = "Violación")]
        Violacion = 4,

        [Display(Name = "Explotación Sexual")]
```

```
        ExplotacionSexual = 5
    }
```

## **Apéndice A.3**: Clase Cuenta

```
public class Cuenta
    {
        public string Email { get; set; }
        public string Contrasena { get; set; }

        public string CreatePassword(int length)
        {
            const string valid = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";
            StringBuilder res = new StringBuilder();
            Random rnd = new Random();
            while (0 < length--)
            {
                res.Append(valid[rnd.Next(valid.Length)]);
            }
            return res.ToString();
        }

        public bool IniciarSesion()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_cuenta_inicio_sesion",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inEmail", Direction =
System.Data.ParameterDirection.Input, Value = this.Email });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inContrasena",
Direction = System.Data.ParameterDirection.Input, Value = this.Contrasena });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    this.SetDesde(datos.Tables[0].Rows[0]);
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
        public bool CerrarSesion()
        {
            throw new System.NotImplementedException();
        }

        public bool AgregarPerfil(Perfil Perfil)
        {
            try
            {
                Perfil.Cuenta = this;
                Perfil.Crear();
                return true;
            }
            catch (Exception ex)
            {
```

```
            }
            finally
            {

            }
            return false;
        }
        public bool RemoverPerfil(Perfil Perfil)
        {
            try
            {
                Perfil.Eliminar();
                return false;
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }

        //  CRUD
        public bool Crear()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_cuenta_crear", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inEmail", Direction =
System.Data.ParameterDirection.Input, Value = this.Email });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inContrasena",
Direction = System.Data.ParameterDirection.Input, Value = this.Contrasena });
                var datos = DB.QueryCommand(command);

                if (datos == 1)
                {
                    return true;
                }
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return false;
        }

        public bool Seleccionar(string Email)
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_cuenta_seleccionar",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inEmail", Direction =
System.Data.ParameterDirection.Input, Value = Email });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    this.SetDesde(datos.Tables[0].Rows[0]);
                    return true;
                }
            }
            catch (Exception ex)
```

```
            {
            }
            finally
            {
            }
            return false;
        }

        public bool CambiarContrasena(string nuevaContrasena)
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_cuenta_modificar", CommandType
= System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inEmail", Direction =
System.Data.ParameterDirection.Input, Value = this.Email });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inContrasena",
Direction = System.Data.ParameterDirection.Input, Value = nuevaContrasena });
                var datos = DB.QueryCommand(command);

                if (datos > 0)
                {
                    this.Contrasena = nuevaContrasena;
                    return true;
                }

            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return false;
        }

        public bool Eliminar()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_cuenta_eliminar", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inEmail", Direction =
System.Data.ParameterDirection.Input, Value = this.Email });
                var temp = DB.QueryCommand(command);

                if (temp == 1)
                {
                    return true;
                }
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return false;
        }


        private void SetDesde(DataRow dr)
        {
            this.Email = dr["Email"].ToString();
            this.Contrasena = dr["Contrasena"].ToString();
        }
    }
```

# **Apéndice A.4**: Clase Perfil

```
public class Perfil
    {
        public Int64 Id { get; set; }
        public string UrlImagen { get; set; }
        public string Nombre { get; set; }
        public Cuenta Cuenta { get; set; }
        public Ubicacion Ubicacion { get; set; }

        public List<Reporte> MisReportes()
        {
            List<Reporte> reportes = new List<Reporte>();
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_reporte_seleccionar_perfil",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId_perfil",
Direction = System.Data.ParameterDirection.Input, Value = this.Id });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    for (int i = 0; i < datos.Tables[0].Rows.Count; i++)
                    {
                        Reporte reporte = new Reporte();
                        reporte.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Id"]));
                        reporte.Descripcion = (datos.Tables[0].Rows[i]["Descripcion"].ToString());
                        reporte.FechaExpedicion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["Fecha"]);
                        reporte.Incidente =
(TipoIncidente)(Convert.ToInt16(datos.Tables[0].Rows[i]["Incidente"]));
                        reporte.Perfil = this;
                        var id_u = Convert.ToInt64(datos.Tables[0].Rows[i]["Ubicacion_Id"]);
                        Ubicacion u = new Ubicacion();
                        u.Seleccionar(id_u);
                        reporte.Ubicacion = u;
                        reportes.Add(reporte);
                    }
                    return reportes;
                }
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return reportes;
        }

        public bool ModificarImagen()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_perfil_modificar", CommandType
= System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = this.Id });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inNombre", Direction
= System.Data.ParameterDirection.Input, Value = this.Nombre });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inUrl", Direction =
System.Data.ParameterDirection.Input, Value = this.UrlImagen });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inEmail", Direction =
System.Data.ParameterDirection.Input, Value = this.Cuenta.Email });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inUbicacion",
Direction = System.Data.ParameterDirection.Input, Value = this.Ubicacion.Id });
                var temp = DB.QueryCommand(command);
```

```
                if (temp == 1)
                {
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }

        public bool AgregarReporte(Reporte Reporte)
        {
            Reporte.Perfil = this;
            try
            {
                return Reporte.Crear();
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
        public bool RemoverReporte(Reporte Reporte)
        {
            try
            {
                return Reporte.Eliminar();
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }

        //CRUD
        public bool Crear()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_perfil_crear", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inNombre", Direction
= System.Data.ParameterDirection.Input, Value = this.Nombre });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inUrlImagen",
Direction = System.Data.ParameterDirection.Input, Value = this.UrlImagen });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inEmail", Direction =
System.Data.ParameterDirection.Input, Value = this.Cuenta.Email });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inUbicacion",
Direction = System.Data.ParameterDirection.Input, Value = this.Ubicacion.Id });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "outId", Direction =
System.Data.ParameterDirection.Output });
                var datos = DB.QueryCommand(command);
                if (datos == 1)
```

```
                {
                    this.Id = Convert.ToInt64(command.Parameters["outId"].Value);
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
        public bool Seleccionar(Int64 Perfil_Id)
        {

            try
            {
                var command = new MySqlCommand() { CommandText = "sp_perfil_seleccionar",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = Perfil_Id });
                var datos = DB.GetDataSet(command);
                if (datos.Tables[0].Rows.Count > 0)
                {
                    this.SetDesde(datos.Tables[0].Rows[0]);
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }

        public bool Seleccionar(string email)
        {

            try
            {
                var command = new MySqlCommand() { CommandText = "sp_perfil_seleccionar_email",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inEmail", Direction =
System.Data.ParameterDirection.Input, Value = email });
                var datos = DB.GetDataSet(command);
                if (datos.Tables[0].Rows.Count > 0)
                {
                    this.SetDesde(datos.Tables[0].Rows[0]);
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
        public bool Modificar()
```

```
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_perfil_modificar", CommandType
= System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = this.Id });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inNombre", Direction
= System.Data.ParameterDirection.Input, Value = this.Nombre });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inUrl", Direction =
System.Data.ParameterDirection.Input, Value = this.UrlImagen });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inEmail", Direction =
System.Data.ParameterDirection.Input, Value = this.Cuenta.Email });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inUbicacion",
Direction = System.Data.ParameterDirection.Input, Value = this.Ubicacion.Id });
                var temp = DB.QueryCommand(command);
                if (temp == 1)
                {
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
        public bool Eliminar()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_perfil_eliminar", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = this.Id });
                var temp = DB.QueryCommand(command);
                if (temp == 1)
                {
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
        private void SetDesde(DataRow dr)
        {
            this.Id = Convert.ToInt64(dr["Id"]);
            this.Nombre = dr["Nombre"].ToString();
            this.UrlImagen = dr["UrlImagen"].ToString();
            this.Cuenta = new Cuenta();
            this.Cuenta.Seleccionar(dr["Cuenta_Email"].ToString());
            this.Ubicacion = new Ubicacion();
            this.Ubicacion.Seleccionar(Convert.ToInt64(dr["Ubicacion_Id"]));

        }

    }
```

# **Apéndice A.5**: Clase Reporte

```
    public class Reporte
    {
        public Int64 Id { get; set; }
        public Perfil Perfil { get; set; }
        public DateTime FechaExpedicion { get; set; }
        public string Descripcion { get; set; }
        public Ubicacion Ubicacion { get; set; }
        public TipoIncidente Incidente { get; set; }
        public DateTime FechaUltimaModificacion { get; set; }

        //CRUD
        public bool Crear()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_reporte_crear", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPerfil_Id",
Direction = System.Data.ParameterDirection.Input, Value = this.Perfil.Id });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inFecha", Direction =
System.Data.ParameterDirection.Input, Value = this.FechaExpedicion });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inIncidente",
Direction = System.Data.ParameterDirection.Input, Value = this.Incidente });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inDescripcion",
Direction = System.Data.ParameterDirection.Input, Value = this.Descripcion });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inUbicacion",
Direction = System.Data.ParameterDirection.Input, Value = this.Ubicacion.Id });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "outId", Direction =
System.Data.ParameterDirection.Output });
                var datos = DB.QueryCommand(command);
                if (datos == 1)
                {
                    this.Id = Convert.ToInt64(command.Parameters["outId"].Value);
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
        public bool Seleccionar(Int64 ID)
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_reporte_seleccionar",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = ID });
                var datos = DB.GetDataSet(command);
                if (datos.Tables[0].Rows.Count == 1)
                {
                    this.SetDesde(datos.Tables[0].Rows[0]);
                    return true;
                }
            }
            catch (Exception ex)
            {
```

```
            }
            finally
            {

            }
            return false;
        }
        public bool Modificar()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_reporte_modificar", CommandType
= System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = this.Id });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPerfil_Id",
Direction = System.Data.ParameterDirection.Input, Value = this.Perfil.Id });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inFecha", Direction =
System.Data.ParameterDirection.Input, Value = this.FechaExpedicion });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inIncidente",
Direction = System.Data.ParameterDirection.Input, Value = this.Incidente });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inDescripcion",
Direction = System.Data.ParameterDirection.Input, Value = this.Descripcion });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inUbicacion",
Direction = System.Data.ParameterDirection.Input, Value = this.Ubicacion.Id });
                command.Parameters.Add(new MySqlParameter() { ParameterName =
"outUltimaModificacion", Direction = System.Data.ParameterDirection.Output });
                var datos = DB.QueryCommand(command);
                if (datos == 1)
                {
                    this.FechaUltimaModificacion =
Convert.ToDateTime(command.Parameters["outUltimaModificacion"].Value);
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
        public bool Eliminar()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_reporte_eliminar", CommandType
= System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = this.Id });
                var temp = DB.QueryCommand(command);
                if (temp == 1)
                {
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
```

```
        private void SetDesde(DataRow dr)
        {
            this.Id = Convert.ToInt64(dr["Id"]);
            this.FechaExpedicion = Convert.ToDateTime(dr["Fecha"]);
            this.Descripcion = dr["Descripcion"].ToString();
            this.Ubicacion = new Ubicacion();
            this.Ubicacion.Seleccionar(Convert.ToInt64(dr["Ubicacion_Id"]));
            this.Perfil = new Perfil();
            this.Perfil.Seleccionar(Convert.ToInt64(dr["Perfil_Id"]));
            this.Incidente = (TipoIncidente)Convert.ToInt64(dr["Incidente"]);
            this.FechaUltimaModificacion = Convert.ToDateTime(dr["UltimaModificacion"]);
        }
    }
```

## **Apéndice A.6**: Clase Ubicación

```
public class Ubicacion
    {
        public Int64 Id { get; set; }
        public float Latitud { get; set; }
        public float Longitud { get; set; }
        public string Direccion { get; set; }
        public Int32 CodigoPostal { get; set; }

        public List<Ubicacion> Seleciconar(string direccion)
        {
            throw new NotImplementedException();
        }

        public List<Ubicacion> Seleciconar(Int32 cp)
        {
            throw new NotImplementedException();
        }

        //CRUD
        public bool Crear()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_ubicacion_crear", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inLatitude",
Direction = System.Data.ParameterDirection.Input, Value = this.Latitud });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inLongitude",
Direction = System.Data.ParameterDirection.Input, Value = this.Longitud });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "outId", Direction =
System.Data.ParameterDirection.Output });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inDelegacion",
Direction = System.Data.ParameterDirection.Input, Value = this.Direccion });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCodigoPostal",
Direction = System.Data.ParameterDirection.Input, Value = this.CodigoPostal });
                var temp = DB.QueryCommand(command);
                if (temp == 1)
                {
                    this.Id = Convert.ToInt64(command.Parameters["outId"].Value);
                    return true;
                }
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return false;
        }
        public bool Seleccionar(Int64 Id)
```

```
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_ubicacion_seleccionar",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = Id });
                var datos = DB.GetDataSet(command);
                if (datos.Tables[0].Rows.Count > 0)
                {
                    this.SetDesde(datos.Tables[0].Rows[0]);
                    return true;
                }
            }
            catch (Exception ex)
            {

            }
            finally
            {

            }
            return false;
        }
        public bool Modificar()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_ubicacion_modificar",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inLatitude",
Direction = System.Data.ParameterDirection.Input, Value = this.Latitud });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inLongitude",
Direction = System.Data.ParameterDirection.Input, Value = this.Longitud });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = this.Id });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inDelegacion",
Direction = System.Data.ParameterDirection.Input, Value = this.Direccion });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCodigoPostal",
Direction = System.Data.ParameterDirection.Input, Value = this.CodigoPostal });

                var temp = DB.QueryCommand(command);
                if (temp == 1)
                {
                    return true;
                }
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return false;
        }
        public bool Eliminar()
        {
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_ubicacion_eliminar",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inId", Direction =
System.Data.ParameterDirection.Input, Value = this.Id });
                var temp = DB.QueryCommand(command);
                if (temp == 1)
                {
                    return true;
                }
            }
```

```
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return false;
        }
        private void SetDesde(DataRow dr)
        {
            this.Id = Convert.ToInt64(dr["Id"]);
            var latitud =  Convert.ToDouble( dr[ "Latitude" ] ).ToString(
System.Globalization.CultureInfo.InvariantCulture );
            var longitud = Convert.ToDouble( dr[ "Longitude" ] ).ToString(
System.Globalization.CultureInfo.InvariantCulture );
            this.Latitud = float.Parse( latitud , System.Globalization.CultureInfo.InvariantCulture
);
            this.Longitud = float.Parse( longitud ,
System.Globalization.CultureInfo.InvariantCulture );
            this.Direccion = dr["Delegacion"].ToString();
            this.CodigoPostal = Convert.ToInt32(dr["CodigoPostal"]);
        }
    }
```

# **Apéndice A.7**: Clase Estantería

```
public class Estanteria
    {
        public List<Reporte> BusquedaAvanzada(int pagina, int cantResult, String palabra, String
fecha, TipoIncidente incidente, String direccion, String nombrePerfil, Ubicacion ubicacion, int
radio)
        {
            int index = pagina;
            if (pagina > 0)
                index = cantResult * (pagina - 1);

            var reportes = new List<Reporte>();
            try
            {
                var command = new MySqlCommand() { CommandText =
"sp_Estanteria_Busqueda_Avanzada_Ubicacion", CommandType = System.Data.CommandType.StoredProcedure
};
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPalabra", Direction
= System.Data.ParameterDirection.Input, Value = palabra });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inFecha", Direction =
System.Data.ParameterDirection.Input, Value = fecha });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inIncidente",
Direction = System.Data.ParameterDirection.Input, Value = (int)incidente });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inDireccion",
Direction = System.Data.ParameterDirection.Input, Value = direccion });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPerfil", Direction
= System.Data.ParameterDirection.Input, Value = nombrePerfil });
                command.Parameters.Add(new MySqlParameter() { ParameterName =
"inUbicacion_Latitude", Direction = System.Data.ParameterDirection.Input, Value = ubicacion.Latitud
});
                command.Parameters.Add(new MySqlParameter() { ParameterName =
"inUbicacion_Longitude", Direction = System.Data.ParameterDirection.Input, Value =
ubicacion.Longitud });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inRadio", Direction =
System.Data.ParameterDirection.Input, Value = radio });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPage", Direction =
System.Data.ParameterDirection.Input, Value = index });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCantResult",
Direction = System.Data.ParameterDirection.Input, Value = cantResult });
                var datos = DB.GetDataSet(command);
```

```
                if (datos.Tables[0].Rows.Count > 0)
                {
                    for (int i = 0; i < datos.Tables[0].Rows.Count; i++)
                    {
                        Reporte reporte = new Reporte();
                        reporte.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Reporte_Id"]));
                        reporte.Descripcion =
(datos.Tables[0].Rows[i]["Reporte_Descripcion"].ToString());
                        reporte.FechaExpedicion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["Reporte_Fecha"]);
                        reporte.Incidente =
(TipoIncidente)(datos.Tables[0].Rows[i]["Reporte_Incidente"]);
                        reporte.Perfil = new Perfil();
                        reporte.Perfil.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Perfil_Id"]));
                        reporte.Perfil.Nombre =
(datos.Tables[0].Rows[i]["Perfil_Nombre"].ToString());
                        reporte.Perfil.UrlImagen =
(datos.Tables[0].Rows[i]["Perfil_UrlImagen"].ToString());
                        reporte.Perfil.Cuenta = new Cuenta();
                        reporte.Perfil.Cuenta.Email =
(datos.Tables[0].Rows[i]["Cuenta_Email"].ToString());
                        reporte.Perfil.Cuenta.Contrasena =
(datos.Tables[0].Rows[i]["Cuenta_Contrasena"].ToString());
                        reporte.Ubicacion = new Ubicacion();
                        var latitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Latitud"]).ToString(System.Globalization.Culture
Info.InvariantCulture);
                        var longitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Longitud"]).ToString(System.Globalization.Cultur
eInfo.InvariantCulture);
                        reporte.Ubicacion.Latitud = float.Parse(latitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Longitud = float.Parse(longitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Direccion =
datos.Tables[0].Rows[i]["Ubicacion_Direccion"].ToString();
                        reportes.Add(reporte);

                    }
                    return reportes;
                }
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return reportes;
        }

        public List<Reporte> BusquedaAvanzada(int pagina, int cantResult, String palabra, String
fecha, TipoIncidente incidente, String direccion, String nombrePerfil)
        {
            int index = pagina;
            if (pagina > 0)
                index = cantResult * (pagina - 1);

            var reportes = new List<Reporte>();
            try
            {
                var command = new MySqlCommand() { CommandText = "sp_Estanteria_Busqueda_Avanzada",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPalabra", Direction
= System.Data.ParameterDirection.Input, Value = palabra });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inFecha", Direction =
System.Data.ParameterDirection.Input, Value = fecha});
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inIncidente",
Direction = System.Data.ParameterDirection.Input, Value = (int)incidente });
```

```
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inDireccion",
Direction = System.Data.ParameterDirection.Input, Value = direccion });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPerfil", Direction
= System.Data.ParameterDirection.Input, Value = nombrePerfil });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPage", Direction =
System.Data.ParameterDirection.Input, Value = index });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCantResult",
Direction = System.Data.ParameterDirection.Input, Value = cantResult });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    for (int i = 0; i < datos.Tables[0].Rows.Count; i++)
                    {
                        Reporte reporte = new Reporte();
                        reporte.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Reporte_Id"]));
                        reporte.Descripcion =
(datos.Tables[0].Rows[i]["Reporte_Descripcion"].ToString());
                        reporte.FechaExpedicion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["Reporte_Fecha"]);
                        reporte.Incidente =
(TipoIncidente)(datos.Tables[0].Rows[i]["Reporte_Incidente"]);
                        reporte.Perfil = new Perfil();
                        reporte.Perfil.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Perfil_Id"]));
                        reporte.Perfil.Nombre =
(datos.Tables[0].Rows[i]["Perfil_Nombre"].ToString());
                        reporte.Perfil.UrlImagen =
(datos.Tables[0].Rows[i]["Perfil_UrlImagen"].ToString());
                        reporte.Perfil.Cuenta = new Cuenta();
                        reporte.Perfil.Cuenta.Email =
(datos.Tables[0].Rows[i]["Cuenta_Email"].ToString());
                        reporte.Perfil.Cuenta.Contrasena =
(datos.Tables[0].Rows[i]["Cuenta_Contrasena"].ToString());
                        reporte.Ubicacion = new Ubicacion();
                        var latitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Latitud"]).ToString(System.Globalization.Culture
Info.InvariantCulture);
                        var longitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Longitud"]).ToString(System.Globalization.Cultur
eInfo.InvariantCulture);
                        reporte.Ubicacion.Latitud = float.Parse(latitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Longitud = float.Parse(longitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Direccion =
datos.Tables[0].Rows[i]["Ubicacion_Direccion"].ToString();
                        reportes.Add(reporte);

                    }
                    return reportes;
                }
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return reportes;
        }

        public List<Reporte> Buscar(string termino, Ubicacion ubicacion, int pagina, int cantResult,
int radio)
        {
            int index = pagina;
            if (pagina > 0)
                index = cantResult * (pagina - 1);
            if (String.IsNullOrEmpty(termino))
            {
```

```
                termino = "";
            }

            var reportes = new List<Reporte>();
            try
            {
                var command = new MySqlCommand() { CommandText =
"sp_Reporte_Seleccionar_Reportes_Termino_Ubicacion", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inTermino", Direction
= System.Data.ParameterDirection.Input, Value = termino });
                command.Parameters.Add(new MySqlParameter() { ParameterName =
"inUbicacion_Latitude", Direction = System.Data.ParameterDirection.Input, Value = ubicacion.Latitud
});
                command.Parameters.Add(new MySqlParameter() { ParameterName =
"inUbicacion_Longitude", Direction = System.Data.ParameterDirection.Input, Value =
ubicacion.Longitud });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inRadio", Direction =
System.Data.ParameterDirection.Input, Value = radio });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPage", Direction =
System.Data.ParameterDirection.Input, Value = index });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCantResult",
Direction = System.Data.ParameterDirection.Input, Value = cantResult });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    for (int i = 0; i < datos.Tables[0].Rows.Count; i++)
                    {
                        Reporte reporte = new Reporte();
                        reporte.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Reporte_Id"]));
                        reporte.Descripcion =
(datos.Tables[0].Rows[i]["Reporte_Descripcion"].ToString());
                        reporte.FechaExpedicion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["Reporte_Fecha"]);
                        reporte.Incidente =
(TipoIncidente)(datos.Tables[0].Rows[i]["Reporte_Incidente"]);
                        reporte.Perfil = new Perfil();
                        reporte.Perfil.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Perfil_Id"]));
                        reporte.Perfil.Nombre =
(datos.Tables[0].Rows[i]["Perfil_Nombre"].ToString());
                        reporte.Perfil.UrlImagen =
(datos.Tables[0].Rows[i]["Perfil_UrlImagen"].ToString());
                        reporte.Perfil.Cuenta = new Cuenta();
                        reporte.Perfil.Cuenta.Email =
(datos.Tables[0].Rows[i]["Cuenta_Email"].ToString());
                        reporte.Perfil.Cuenta.Contrasena =
(datos.Tables[0].Rows[i]["Cuenta_Contrasena"].ToString());
                        reporte.Ubicacion = new Ubicacion();
                        var latitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Latitud"]).ToString(System.Globalization.Culture
Info.InvariantCulture);
                        var longitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Longitud"]).ToString(System.Globalization.Cultur
eInfo.InvariantCulture);
                        reporte.Ubicacion.Latitud = float.Parse(latitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Longitud = float.Parse(longitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Direccion =
datos.Tables[0].Rows[i]["Ubicacion_Direccion"].ToString();
                        reportes.Add(reporte);

                    }
                    return reportes;
                }
            }
            catch (Exception ex)
            {
```

```
            }
            finally
            {
            }
            return reportes;
        }

        public List<Reporte> Buscar(TipoIncidente incidente, int pagina, int cantResult)
        {
            int index = pagina;
            if (pagina > 0)
                index = cantResult * (pagina - 1);
            List<Reporte> reportes = new List<Reporte>();
            try
            {
                var command = new MySqlCommand() { CommandText =
"sp_Reporte_Seleccionar_Reportes_Termino_Incidente", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inIncidente",
Direction = System.Data.ParameterDirection.Input, Value = (int)incidente });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPage", Direction =
System.Data.ParameterDirection.Input, Value = index });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCantResult",
Direction = System.Data.ParameterDirection.Input, Value = cantResult });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    for (int i = 0; i < datos.Tables[0].Rows.Count; i++)
                    {
                        Reporte reporte = new Reporte();
                        reporte.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Reporte_Id"]));
                        reporte.Descripcion =
(datos.Tables[0].Rows[i]["Reporte_Descripcion"].ToString());
                        reporte.FechaExpedicion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["Reporte_Fecha"]);
                        reporte.Incidente =
(TipoIncidente)(datos.Tables[0].Rows[i]["Reporte_Incidente"]);
                        reporte.Perfil = new Perfil();
                        reporte.Perfil.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Perfil_Id"]));
                        reporte.Perfil.Nombre =
(datos.Tables[0].Rows[i]["Perfil_Nombre"].ToString());
                        reporte.Perfil.UrlImagen =
(datos.Tables[0].Rows[i]["Perfil_UrlImagen"].ToString());
                        reporte.Perfil.Cuenta = new Cuenta();
                        reporte.Perfil.Cuenta.Email =
(datos.Tables[0].Rows[i]["Cuenta_Email"].ToString());
                        reporte.Perfil.Cuenta.Contrasena =
(datos.Tables[0].Rows[i]["Cuenta_Contrasena"].ToString());
                        reporte.Ubicacion = new Ubicacion();
                        var latitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Latitud"]).ToString(System.Globalization.Culture
Info.InvariantCulture);
                        var longitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Longitud"]).ToString(System.Globalization.Cultur
eInfo.InvariantCulture);
                        reporte.Ubicacion.Latitud = float.Parse(latitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Longitud = float.Parse(longitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Direccion =
datos.Tables[0].Rows[i]["Ubicacion_Direccion"].ToString();
                        reportes.Add(reporte);

                    }
                    return reportes;
                }
            }
            catch (Exception ex)
```

```
            {
            }
            finally
            {
            }
            return reportes;
        }

        public List<Reporte> BuscarPorFecha(String fecha, int pagina, int cantResult)
        {
            int index = pagina;
            if (pagina > 0)
                index = cantResult * (pagina - 1);
            List<Reporte> reportes = new List<Reporte>();
            try
            {
                var command = new MySqlCommand() { CommandText =
"sp_Reporte_Seleccionar_Reportes_Termino_Fecha", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inFecha", Direction =
System.Data.ParameterDirection.Input, Value = fecha });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPage", Direction =
System.Data.ParameterDirection.Input, Value = index });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCantResult",
Direction = System.Data.ParameterDirection.Input, Value = cantResult });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    for (int i = 0; i < datos.Tables[0].Rows.Count; i++)
                    {
                        Reporte reporte = new Reporte();
                        reporte.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Reporte_Id"]));
                        reporte.Descripcion =
(datos.Tables[0].Rows[i]["Reporte_Descripcion"].ToString());
                        reporte.FechaExpedicion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["Reporte_Fecha"]);
                        reporte.Incidente =
(TipoIncidente)(datos.Tables[0].Rows[i]["Reporte_Incidente"]);
                        reporte.Perfil = new Perfil();
                        reporte.Perfil.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Perfil_Id"]));
                        reporte.Perfil.Nombre =
(datos.Tables[0].Rows[i]["Perfil_Nombre"].ToString());
                        reporte.Perfil.UrlImagen =
(datos.Tables[0].Rows[i]["Perfil_UrlImagen"].ToString());
                        reporte.Perfil.Cuenta = new Cuenta();
                        reporte.Perfil.Cuenta.Email =
(datos.Tables[0].Rows[i]["Cuenta_Email"].ToString());
                        reporte.Perfil.Cuenta.Contrasena =
(datos.Tables[0].Rows[i]["Cuenta_Contrasena"].ToString());
                        reporte.Ubicacion = new Ubicacion();
                        var latitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Latitud"]).ToString(System.Globalization.Culture
Info.InvariantCulture);
                        var longitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Longitud"]).ToString(System.Globalization.Cultur
eInfo.InvariantCulture);
                        reporte.Ubicacion.Latitud = float.Parse(latitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Longitud = float.Parse(longitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Direccion =
datos.Tables[0].Rows[i]["Ubicacion_Direccion"].ToString();
                        reportes.Add(reporte);

                    }
                    return reportes;
                }
            }
```

```
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return reportes;
        }

        public List<Reporte> BuscarPorPalabra(String palabra, int pagina, int cantResult)
        {
            int index = pagina;
            if (pagina > 0)
                index = cantResult * (pagina - 1);
            List<Reporte> reportes = new List<Reporte>();
            try
            {
                var command = new MySqlCommand() { CommandText =
"sp_Reporte_Seleccionar_Reportes_Termino_Descripcion", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPalabra", Direction
= System.Data.ParameterDirection.Input, Value = palabra });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPage", Direction =
System.Data.ParameterDirection.Input, Value = index });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCantResult",
Direction = System.Data.ParameterDirection.Input, Value = cantResult });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    for (int i = 0; i < datos.Tables[0].Rows.Count; i++)
                    {
                        Reporte reporte = new Reporte();
                        reporte.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Reporte_Id"]));
                        reporte.Descripcion =
(datos.Tables[0].Rows[i]["Reporte_Descripcion"].ToString());
                        reporte.FechaExpedicion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["Reporte_Fecha"]);
                        reporte.Incidente =
(TipoIncidente)(datos.Tables[0].Rows[i]["Reporte_Incidente"]);
                        reporte.Perfil = new Perfil();
                        reporte.Perfil.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Perfil_Id"]));
                        reporte.Perfil.Nombre =
(datos.Tables[0].Rows[i]["Perfil_Nombre"].ToString());
                        reporte.Perfil.UrlImagen =
(datos.Tables[0].Rows[i]["Perfil_UrlImagen"].ToString());
                        reporte.Perfil.Cuenta = new Cuenta();
                        reporte.Perfil.Cuenta.Email =
(datos.Tables[0].Rows[i]["Cuenta_Email"].ToString());
                        reporte.Perfil.Cuenta.Contrasena =
(datos.Tables[0].Rows[i]["Cuenta_Contrasena"].ToString());
                        reporte.Ubicacion = new Ubicacion();
                        var latitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Latitud"]).ToString(System.Globalization.Culture
Info.InvariantCulture);
                        var longitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Longitud"]).ToString(System.Globalization.Cultur
eInfo.InvariantCulture);
                        reporte.Ubicacion.Latitud = float.Parse(latitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Longitud = float.Parse(longitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Direccion =
datos.Tables[0].Rows[i]["Ubicacion_Direccion"].ToString();
                        reportes.Add(reporte);

                    }
                    return reportes;
                }
```

```
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return reportes;
        }

        public List<Reporte> BuscarPorDireccion(String direccion, int pagina, int cantResult)
        {
            int index = pagina;
            if (pagina > 0)
                index = cantResult * (pagina - 1);
            List<Reporte> reportes = new List<Reporte>();
            try
            {
                var command = new MySqlCommand() { CommandText =
"sp_Reporte_Seleccionar_Reportes_Termino_Direccion", CommandType =
System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inDireccion",
Direction = System.Data.ParameterDirection.Input, Value = direccion});
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPage", Direction =
System.Data.ParameterDirection.Input, Value = index });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCantResult",
Direction = System.Data.ParameterDirection.Input, Value = cantResult });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    for (int i = 0; i < datos.Tables[0].Rows.Count; i++)
                    {
                        Reporte reporte = new Reporte();
                        reporte.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Reporte_Id"]));
                        reporte.Descripcion =
(datos.Tables[0].Rows[i]["Reporte_Descripcion"].ToString());
                        reporte.FechaExpedicion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["Reporte_Fecha"]);
                        reporte.Incidente =
(TipoIncidente)(datos.Tables[0].Rows[i]["Reporte_Incidente"]);
                        reporte.Perfil = new Perfil();
                        reporte.Perfil.Id = (Convert.ToInt64(datos.Tables[0].Rows[i]["Perfil_Id"]));
                        reporte.Perfil.Nombre =
(datos.Tables[0].Rows[i]["Perfil_Nombre"].ToString());
                        reporte.Perfil.UrlImagen =
(datos.Tables[0].Rows[i]["Perfil_UrlImagen"].ToString());
                        reporte.Perfil.Cuenta = new Cuenta();
                        reporte.Perfil.Cuenta.Email =
(datos.Tables[0].Rows[i]["Cuenta_Email"].ToString());
                        reporte.Perfil.Cuenta.Contrasena =
(datos.Tables[0].Rows[i]["Cuenta_Contrasena"].ToString());
                        reporte.Ubicacion = new Ubicacion();
                        var latitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Latitud"]).ToString(System.Globalization.Culture
Info.InvariantCulture);
                        var longitud =
Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion_Longitud"]).ToString(System.Globalization.Cultur
eInfo.InvariantCulture);
                        reporte.Ubicacion.Latitud = float.Parse(latitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Longitud = float.Parse(longitud,
System.Globalization.CultureInfo.InvariantCulture);
                        reporte.Ubicacion.Direccion =
datos.Tables[0].Rows[i]["Ubicacion_Direccion"].ToString();
                        reportes.Add(reporte);

                    }
                    return reportes;
```

```
                }
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return reportes;
        }

        public List<Reporte> BuscarPorAll(string termino, int pagina, int cantResult)
        {
            List<Reporte> reportes = new List<Reporte>();
            int index = pagina;
            if (pagina > 0)
                index = cantResult * (pagina - 1);

            try
            {
                var command = new MySqlCommand() { CommandText = "sp_Estanteria_Buscar_All",
CommandType = System.Data.CommandType.StoredProcedure };
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inTermino", Direction
= System.Data.ParameterDirection.Input, Value = termino });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inPage", Direction =
System.Data.ParameterDirection.Input, Value = index });
                command.Parameters.Add(new MySqlParameter() { ParameterName = "inCantResult",
Direction = System.Data.ParameterDirection.Input, Value = cantResult });
                var datos = DB.GetDataSet(command);

                if (datos.Tables[0].Rows.Count > 0)
                {
                    for (int i = 0; i < datos.Tables[0].Rows.Count; i++)
                    {

                        Reporte reporte = new Reporte();
                        Perfil perfil = new Perfil();
                        reporte.Id = Convert.ToInt64(datos.Tables[0].Rows[i]["Reporte_Id"]);
                        reporte.FechaExpedicion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["FechaExpedicion"]);
                        reporte.Descripcion = datos.Tables[0].Rows[i]["Descripcion"].ToString();
                        reporte.Ubicacion = new Ubicacion();

reporte.Ubicacion.Seleccionar(Convert.ToInt64(datos.Tables[0].Rows[i]["Ubicacion_Id"]));
                        reporte.Perfil = new Perfil();

reporte.Perfil.Seleccionar(Convert.ToInt64(datos.Tables[0].Rows[i]["Perfil_Id"]));
                        reporte.Incidente =
(TipoIncidente)Convert.ToInt64(datos.Tables[0].Rows[i]["Incidente"]);
                        reporte.FechaUltimaModificacion =
Convert.ToDateTime(datos.Tables[0].Rows[i]["UltimaModificacion"]);

                        reportes.Add(reporte);
                    }
                }
            }
            catch (Exception ex)
            {
            }
            finally
            {
            }
            return reportes;

        }

        //Traer las utlimas publicaciones
        public List<Reporte> UltimosPublicados(int cantidad)
        {
```

```
            throw new System.NotImplementedException();
        }
    }
```

# Apéndice B. *Base de datos*

## **Apéndice B.1**: Creacion de tablas

### Apéndice B.1.1: Cuenta

```
CREATE TABLE IF NOT EXISTS Cuenta(
    Email       VARCHAR(256) NOT NULL PRIMARY KEY,
    Contrasena VARCHAR(256)NOT NULL
);
```

### Apéndice B.1.2: Perfil

```
CREATE TABLE IF NOT EXISTS Perfil(
    Id BIGINT UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
    Nombre      VARCHAR(256),
    UrlImagen VARCHAR(256),
    Cuenta_Email VARCHAR(256) NOT NULL,
    Ubicacion_Id BIGINT UNSIGNED NOT NULL,
    FOREIGN KEY(Cuenta_Email) REFERENCES Cuenta(Email) ON DELETE CASCADE,
    FOREIGN KEY(Ubicacion_Id) REFERENCES Ubicacion(Id) ON DELETE RESTRICT
);
```

### Apéndice B.1.3: Ubicación

```
CREATE TABLE IF NOT EXISTS Ubicacion(
    Id BIGINT UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
    Latitude FLOAT( 10, 6 ) NOT NULL ,
    Longitude FLOAT( 10, 6 ) NOT NULL,
    Delegacion VARCHAR(256),
    CodigoPostal INT
);
```

### Apéndice B.1.4: Reporte

```
CREATE TABLE IF NOT EXISTS Reporte(
```

```
    Id BIGINT UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
    Perfil_Id BIGINT UNSIGNED NOT NULL,
    Fecha DATETIME,
    Incidente INT NOT NULL,
    Descripcion TEXT,
    Ubicacion_Id BIGINT UNSIGNED NOT NULL,
    UltimaModificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (Perfil_Id) REFERENCES Perfil(Id) ON DELETE CASCADE,
    FOREIGN KEY (Ubicacion_Id) REFERENCES Ubicacion(Id) ON DELETE RESTRICT
);
```

# **Apéndice B.2**: Procedimientos almacenados

## **Apéndice B.2.1**: Cuenta – Crear

```
--Delimiter $$
DROP PROCEDURE IF EXISTS sp_cuenta_crear $$

CREATE PROCEDURE sp_cuenta_crear(
        inEmail VARCHAR(256),
        inContrasena VARCHAR(256)
)

BEGIN
        INSERT INTO Cuenta
                (Email, Contrasena)
        VALUES
                (inEmail,inContrasena);
END
$$
```

## **Apéndice B.2.2**: Cuenta – Seleccionar

```
--Delimiter $$
DROP PROCEDURE IF EXISTS sp_cuenta_seleccionar $$

CREATE PROCEDURE sp_cuenta_seleccionar(
        inEmail VARCHAR(256)
)

BEGIN
        SELECT Email, Contrasena
    FROM Cuenta
    WHERE inEmail = Email;
END
$$
```

## **Apéndice B.2.3**: Cuenta – Modificar

```
--Delimiter $$
DROP PROCEDURE IF EXISTS sp_cuenta_modificar $$

CREATE PROCEDURE sp_cuenta_modificar(
        inEmail VARCHAR(256),
        inContrasena VARCHAR(256)
)
```

```
BEGIN
        UPDATE Cuenta
        SET
                Contrasena = inContrasena
        WHERE
                inEmail = Email;
END
$$
```

## **Apéndice B.2.4**: Cuenta – Eliminar

```
--DELIMITER $$
DROP PROCEDURE IF EXISTS sp_cuenta_eliminar $$

CREATE PROCEDURE sp_cuenta_eliminar(
        inEmail VARCHAR(256)
)
BEGIN
        DELETE FROM Cuenta
                WHERE
                        Email = inEmail;

END
$$
```

## **Apéndice B.2.5**: Estanteria - Busqueda por todo

```
--Delimiter $$
DROP PROCEDURE IF EXISTS sp_Estanteria_Buscar_All $$

CREATE PROCEDURE sp_Estanteria_Buscar_All (
        inTermino varchar(255),
    inPage int,
        inCantResult int
)
BEGIN
        SELECT DISTINCT
                r.Id as Reporte_Id,
        r.Perfil_Id AS Perfil_Id,
        r.fecha as FechaExpedicion,
        r.Incidente as Incidente,
        r.Descripcion as Descripcion,
        r.Ubicacion_Id as Ubicacion_Id,
        p.Nombre as Perfil_Nombre,
        r.UltimaModificacion as UltimaModificacion

        FROM
                ubicacion AS u,
                perfil AS p,
                reporte AS r
        WHERE

                u.Id = r.Ubicacion_Id
                AND
                r.Perfil_Id = p.Id
                AND
                (
                        u.Delegacion LIKE CONCAT('%', SUBSTRING_INDEX(SUBSTRING_INDEX( inTermino , '
', 2 ),' ',1) , '%')
                        OR
```

```
                    r.Descripcion LIKE CONCAT('%', SUBSTRING_INDEX(SUBSTRING_INDEX( inTermino, ' ',
-1 ),' ',2) , '%')
                    OR
            p.Nombre = inTermino
            OR
                    CAST(r.Fecha AS DATE)= inTermino
                    OR
                    r.Incidente = CAST( inTermino AS decimal)
                )
        ORDER BY r.Fecha DESC
        LIMIT inPage, inCantResult;
END $$
```

## **Apéndice B.2.6**: Estanteria - Busqueda por todo y ubicación de un perfil

```
--Delimiter $$
DROP PROCEDURE IF EXISTS sp_Reporte_Seleccionar_Reportes_Termino_Ubicacion $$
CREATE PROCEDURE sp_Reporte_Seleccionar_Reportes_Termino_Ubicacion (
        inTermino VARCHAR(256),
    inUbicacion_Latitude float(10,6),
    inUbicacion_Longitude float(10,6),
    inPage int,
        inCantResult int,
    inRadio     INT
)
BEGIN
SELECT
        r.Id                    AS      Reporte_Id,
    r.Descripcion       AS      Reporte_Descripcion,
    r.Fecha                     AS      Reporte_Fecha,
    r.Incidente         AS      Reporte_Incidente,
    r.Perfil_Id         AS      Perfil_Id,
    r.Ubicacion_Id  AS  Ubicacion_Id,
    p.Nombre            AS      Perfil_Nombre,
    p.UrlImagen         AS      Perfil_UrlImagen,
    p.Cuenta_Email      AS      Cuenta_Email,
    c.ContrasenaAS      Cuenta_Contrasena,
    u.Latitude          AS      Ubicacion_Latitud,
    u.Longitude         AS      Ubicacion_Longitud,
    u.DelegacionAS      Ubicacion_Direccion


FROM
    reporte             AS r
INNER JOIN perfil AS p
        ON p.Id = r.Perfil_Id
INNER JOIN ubicacion AS u
        ON u.Id = r.Ubicacion_Id
INNER JOIN cuenta AS c
        ON p.Cuenta_Email =  c.Email
    WHERE
    ( 6371 * acos( cos( radians(inUbicacion_Latitude) ) * cos( radians( u.Latitude ) ) * cos(
radians( u.Longitude ) - radians(inUbicacion_Longitude) ) + sin( radians(inUbicacion_Latitude) ) *
sin( radians( u.Latitude ) ) ) ) <= inRadio AND
                (u.Delegacion LIKE CONCAT('%', SUBSTRING_INDEX(SUBSTRING_INDEX( inTermino , ' ', 2
),' ',1) , '%')
                OR
                r.Descripcion LIKE CONCAT('%', SUBSTRING_INDEX(SUBSTRING_INDEX( inTermino, ' ', -1
),' ',2) , '%')
                OR
                        p.Nombre = inTermino
                    OR
                CAST(r.Fecha AS DATE)= inTermino)
        ORDER BY r.Fecha desc
    LIMIT inPage, inCantResult;
```

```
END $$
```

## **Apéndice B.2.7**: Estanteria - Busqueda por palabras en descripción

```
--Delimiter $$
DROP PROCEDURE IF EXISTS sp_Reporte_Seleccionar_Reportes_Termino_Descripcion $$
CREATE PROCEDURE sp_Reporte_Seleccionar_Reportes_Termino_Descripcion (
        inPalabra VARCHAR(255),
    inPage int,
        inCantResult int
)
BEGIN
SELECT
        r.Id                    AS      Reporte_Id,
    r.Descripcion       AS      Reporte_Descripcion,
    r.Fecha                     AS      Reporte_Fecha,
    r.Incidente         AS      Reporte_Incidente,
    r.Perfil_Id         AS      Perfil_Id,
    r.Ubicacion_Id  AS  Ubicacion_Id,
    p.Nombre            AS      Perfil_Nombre,
    p.UrlImagen         AS      Perfil_UrlImagen,
    p.Cuenta_Email      AS      Cuenta_Email,
    c.ContrasenaAS      Cuenta_Contrasena,
    u.Latitude          AS      Ubicacion_Latitud,
    u.Longitude         AS      Ubicacion_Longitud,
    u.DelegacionAS      Ubicacion_Direccion


FROM
    reporte             AS r
INNER JOIN perfil AS p
        ON p.Id = r.Perfil_Id
INNER JOIN ubicacion AS u
        ON u.Id = r.Ubicacion_Id
INNER JOIN cuenta AS c
        ON p.Cuenta_Email =  c.Email
    WHERE
    r.Descripcion LIKE CONCAT('%', SUBSTRING_INDEX(SUBSTRING_INDEX( inPalabra , ' ', 2 ),' ',1) ,
'%')
        ORDER BY r.Fecha desc
    LIMIT inPage, inCantResult;


END $$
```

## **Apéndice B.2.8**: Estanteria - Busqueda avanzada

```
--Delimiter $$
DROP PROCEDURE IF EXISTS sp_Estanteria_Busqueda_Avanzada $$

CREATE PROCEDURE sp_Estanteria_Busqueda_Avanzada (
        inPalabra               varchar(255),
    inFecha                     varchar(10),
    inIncidente         int,
    inDireccion         varchar(255),
        inPerfil        varchar(20),
    inPage                      int,
        inCantResult    int
)
BEGIN
        SELECT DISTINCT
        r.Id                    AS      Reporte_Id,
    r.Descripcion       AS      Reporte_Descripcion,
    r.Fecha                     AS      Reporte_Fecha,
    r.Incidente         AS      Reporte_Incidente,
```

```
    r.Perfil_Id         AS      Perfil_Id,
    r.Ubicacion_Id  AS  Ubicacion_Id,
    p.Nombre            AS      Perfil_Nombre,
    p.UrlImagen         AS      Perfil_UrlImagen,
    p.Cuenta_Email      AS      Cuenta_Email,
    c.ContrasenaAS      Cuenta_Contrasena,
    u.Latitude          AS      Ubicacion_Latitud,
    u.Longitude         AS      Ubicacion_Longitud,
    u.DelegacionAS      Ubicacion_Direccion

        FROM
                ubicacion AS u,
                perfil AS p,
                reporte AS r,
        cuenta AS c
        WHERE
                c.Email = p.Cuenta_Email
        AND
                u.Id = r.Ubicacion_Id
                AND
                r.Perfil_Id = p.Id
                AND
                (
                        u.Delegacion LIKE CONCAT('%', SUBSTRING_INDEX(SUBSTRING_INDEX( inDireccion ,
' ', 2 ),' ',1) , '%')
                        Or
                    r.Descripcion LIKE CONCAT('%', SUBSTRING_INDEX(SUBSTRING_INDEX( inPalabra, ' ',
-1 ),' ',2) , '%')
                    Or
            p.Nombre = inPerfil
            Or
                    CAST(r.Fecha AS DATE)= inFecha
                    Or
                    r.Incidente = inIncidente
                )
        ORDER BY r.Fecha DESC
        LIMIT inPage, inCantResult;
END
$$
```

## **Apéndice B.2.9**: Estanteria - Busqueda avanzada con ubicación de un perfil

```
--Delimiter $$
DROP PROCEDURE IF EXISTS sp_Estanteria_Busqueda_Avanzada_Ubicacion $$

CREATE PROCEDURE sp_Estanteria_Busqueda_Avanzada_Ubicacion (
        inPalabra               varchar(255),
    inFecha                     varchar(10),
    inIncidente         int,
    inDireccion         varchar(255),
        inPerfil        varchar(20),
    inUbicacion_Latitude float(10,6),
    inUbicacion_Longitude float(10,6),
    inRadio     int,
    inPage                      int,
        inCantResult    int
)
BEGIN
        SELECT DISTINCT
        r.Id                    AS      Reporte_Id,
    r.Descripcion       AS      Reporte_Descripcion,
    r.Fecha                     AS      Reporte_Fecha,
    r.Incidente         AS      Reporte_Incidente,
    r.Perfil_Id         AS      Perfil_Id,
    r.Ubicacion_Id  AS  Ubicacion_Id,
    p.Nombre            AS      Perfil_Nombre,
```

```
    p.UrlImagen         AS      Perfil_UrlImagen,
    p.Cuenta_Email      AS      Cuenta_Email,
    c.ContrasenaAS      Cuenta_Contrasena,
    u.Latitude          AS      Ubicacion_Latitud,
    u.Longitude         AS      Ubicacion_Longitud,
    u.DelegacionAS      Ubicacion_Direccion

        FROM
                ubicacion AS u,
                perfil AS p,
                reporte AS r,
        cuenta AS c
        WHERE
                c.Email = p.Cuenta_Email
        AND
                u.Id = r.Ubicacion_Id
                AND
                r.Perfil_Id = p.Id
                AND
                (
                        ( 6371 * acos( cos( radians(inUbicacion_Latitude) ) * cos( radians(
u.Latitude ) ) * cos( radians( u.Longitude ) - radians(inUbicacion_Longitude) ) + sin(
radians(inUbicacion_Latitude) ) * sin( radians( u.Latitude ) ) ) ) <= inRadio AND
                        u.Delegacion LIKE CONCAT('%', SUBSTRING_INDEX(SUBSTRING_INDEX( inDireccion ,
' ', 2 ),' ',1) , '%')
                        Or
                    r.Descripcion LIKE CONCAT('%', SUBSTRING_INDEX(SUBSTRING_INDEX( inPalabra, ' ',
-1 ),' ',2) , '%')
                    Or
            p.Nombre = inPerfil
            Or
                    CAST(r.Fecha AS DATE)= inFecha
                    Or
                    r.Incidente = inIncidente
                )
        ORDER BY r.Fecha DESC
        LIMIT inPage, inCantResult;
END
$$
```

## **Apéndice B.2.10**: Uso de Apache ant para caragr la base de datos.

```
<project name="Database creation" default="dist" basedir=".">
        <property name="sql.driver"     value="com.mysql.jdbc.Driver"/>
        <property name="sql.url"
        value="jdbc:mysql://localhost:3306/proyecto_integracion_api" />
        <property name="sql.db"         value="proyecto_integracion_api"/>
        <property name="sql.user"       value="proyecto_integracion_user"/>
        <property name="sql.pass"               value="qwer12345678"/>
        <property name="sql.encode"     value="UTF-8"/>
        <target name="Init">
      <input message="Do you really want to delete this Database (y/n)?" validargs="y,n"
addproperty="do.delete" />
      <condition property="do.abort">
            <equals arg1="n" arg2="${do.delete}"/>
      </condition>
      <fail if="do.abort">Build aborted by user.</fail>
      <sql driver="${sql.driver}"
                url="${sql.url}"
                userid="${sql.user}"
                password="${sql.pass}"
                classpath = "../Lib/mysql-connector-java/mysql-connector-java-5.1.39-bin.jar"
                        delimiter = "$$"
                        encoding = "${sql.encode}">
                   SET FOREIGN_KEY_CHECKS=0;$$
            DROP TABLE IF EXISTS ${sql.db}.`perfil_reporte`;$$
            DROP TABLE IF EXISTS ${sql.db}.`ubicacion`;$$
            DROP TABLE IF EXISTS ${sql.db}.`reporte`;$$
```

```
          DROP TABLE IF EXISTS ${sql.db}.`perfil`;$$
          DROP TABLE IF EXISTS ${sql.db}.`cuenta`;$$
          SET FOREIGN_KEY_CHECKS=1;$$
      </sql>
   </target>

        <target name="CreateTables">
                <sql
                        driver="${sql.driver}"
                        url="${sql.url}"
                        userid="${sql.user}"
                        password="${sql.pass}"
                        autocommit="true"
                        classpath = "../Lib/mysql-connector-java/mysql-connector-java-5.1.39-
bin.jar"
                        delimiter = "$$"
                        encoding = "${sql.encode}">
                        <path>
                                <fileset dir=".">
                                        <include name= "00*.sql"/>
                                </fileset>
                        </path>

                </sql>
        </target>

        <target name="CreateStores">
                <sql
                        driver="${sql.driver}"
                        url="${sql.url}"
                        userid="${sql.user}"
                        password="${sql.pass}"
                        autocommit="true"
                        classpath = "../Lib/mysql-connector-java/mysql-connector-java-5.1.39-
bin.jar"
                        delimiter = "$$"
                        encoding = "${sql.encode}">
                        <path>
                                <fileset dir=".">
                                        <include name= "4*.sql"/>
                                </fileset>
                        </path>

                </sql>
        </target>

        <target name="InsertData">
                <sql
                        driver="${sql.driver}"
                        url="${sql.url}"
                        userid="${sql.user}"
                        password="${sql.pass}"
                        autocommit="true"
                        classpath = "../Lib/mysql-connector-java/mysql-connector-java-5.1.39-
bin.jar"
                        delimiter = "$$"
                        encoding = "${sql.encode}">
                        <path>
                                <fileset dir=".">
                                        <include name= "9*.sql"/>
                                </fileset>
                        </path>

                </sql>
        </target>

        <target name ="CreateDataBase">
                <antcall target = "CreateTables"/>
                <antcall target = "CreateStores"/>
```

```
                <antcall target = "InsertData"/>
        </target>
</project>
```

# Apéndice C. *Gestión de un perfil*

## **Apéndice C.1**: Utilidades

### Apéndice C.1.1: Email

```
      public class Email
    {
        public static void SendEmail(string Subject, string To, string Message)
        {
            using (MailMessage mm = new MailMessage())
            {
                mm.From = new MailAddress("noreplyreportit@gmail.com");
                mm.Subject = Subject;
                mm.Body = Message;
                mm.IsBodyHtml = true;
                mm.To.Add(new MailAddress(To));

                SmtpClient smtp = new SmtpClient();
                smtp.Host = "smtp.gmail.com";
                smtp.EnableSsl = true;
                var credential = new NetworkCredential { UserName = "noreplyreportit@gmail.com",
Password = "ReportIt-2017" };
                smtp.UseDefaultCredentials = true;
                smtp.Credentials = credential;
                smtp.Port = 587;
                smtp.Send(mm);
            }
        }
    }
```

### Apéndice C.1.2: Validador

```
public class Validator
{
public static Boolean isNullOrEmptyOrWhiteSpace(List<String> list)
{
foreach(String input in list)
{
if (String.IsNullOrEmpty(input) || string.IsNullOrWhiteSpace(input))
{
return true;
}
}
return false;
}

public static Boolean esValido(string email)
{
String expresion;
expresion = "\\w+([-+.']\\w+)*@\\w+([-.]\\w+)*\\.\\w+([-.]\\w+)*";
if (Regex.IsMatch(email, expresion))
```

```
{
if (Regex.Replace(email, expresion, String.Empty).Length == 0)
{
return true;
}
else
{
return false;
}
}
else
{
return false;
}
}

public static bool verificarExtension(String ext)
{
if (ext.ToLower().Contains("gif") || ext.ToLower().Contains("jpg") || ext.ToLower().Contains("jpeg")
|| ext.ToLower().Contains("png"))
{
return true;
}
return false;
}
}
```

## **Apéndice C.1.3**: Manejador de sesiones

```
public class SessionManager
 {
     public static void Ingresar(String Email)
     {
         HttpCookie _sessionCookie = new HttpCookie("_sessionCookie");
         _sessionCookie.Values["Email"] = Email;
         _sessionCookie.Expires = DateTime.Now.AddDays(1);
         HttpContext.Current.Response.Cookies.Add(_sessionCookie);
     }

     public static void SalirPerfil()
     {
         HttpCookie _sessionCookie = HttpContext.Current.Request.Cookies["_sessionCookie"];
         if (!String.IsNullOrEmpty(_sessionCookie.Values["Perfil"]))
             _sessionCookie.Values["Perfil"] = string.Empty;
         HttpContext.Current.Response.Cookies.Add(_sessionCookie);
     }

     public static void Salir()
     {
         HttpCookie _sessionCookie = HttpContext.Current.Request.Cookies["_sessionCookie"];
         _sessionCookie.Expires = DateTime.Now.AddDays(-1);
         _sessionCookie.Values["Email"] = String.Empty;
         _sessionCookie.Values["Perfil"] = string.Empty;
         HttpContext.Current.Response.Cookies.Add(_sessionCookie);
     }

     public static Cuenta CuentaActiva()
     {
         Cuenta cuenta = null;
         HttpCookie _session = HttpContext.Current.Request.Cookies["_sessionCookie"];

         if (_session != null &&
             !string.IsNullOrEmpty(_session.Values["Email"]))
         {
             cuenta = new Cuenta() { Email = _session.Values["Email"] };
             return cuenta;
         }
```

```
        return cuenta;
    }

    public static Perfil PerfilActivo()
    {
        Perfil perfil = null;
        HttpCookie _session = HttpContext.Current.Request.Cookies["_sessionCookie"];

        if (_session != null &&
            !string.IsNullOrEmpty(_session.Values["Email"]) &&
            !string.IsNullOrEmpty(_session.Values["Perfil"]))
        {
            perfil = new Perfil();
            perfil.Seleccionar(Convert.ToInt64(_session.Values["Perfil"]));
            return perfil;
        }

        return perfil;
    }

    public static void RegistarPerfil(long Id)
    {
        HttpCookie _session = HttpContext.Current.Request.Cookies["_sessionCookie"];

        if (_session != null && !string.IsNullOrEmpty(_session.Values["Email"]))
        {
            _session.Values["Perfil"] = "" + Id;
            HttpContext.Current.Response.Cookies.Add(_session);
        }
    }
}
```

## **Apéndice C.1.4**: Geolocalizacón

```
 public static class GeoLocation
{
    /// <summary>
    /// Metodo que muestra la direccion completa dado una ubicacion
    /// </summary>
    /// (rruvalcaba)
    /// <param>Ubicacion</param>
    /// <returns>String</returns>
    public static string direccion( Ubicacion u )
    {
        DataRow row = getDataRow( u );
        return row[ "formatted_address" ].ToString( );
    }

    /// <summary>
    /// Metodo que muestra la ciudad dado una ubicacion
    /// </summary>
    /// (rruvalcaba)
    /// <param>Ubicacion</param>
    /// <returns>String</returns>
    public static string ciudad( Ubicacion u )
    {
        DataRow row = getDataRow( u );
        try
        {
            var value = row[ "formatted_address" ].ToString( ).Split( ',' );
            var count = value.Length;

            return value[ count - 3 ];
        }
        catch ( Exception ex )
        {
```

```
                return "";
            }
        }

        /// <summary>
        /// Metodo que muestra la region dado una ubicacion
        /// </summary>
        /// (rruvalcaba)
        /// <param>Ubicacion</param>
        /// <returns>String</returns>
        public static string region( Ubicacion u )
        {
            DataRow row = getDataRow( u );
            try
            {
                var value = row[ "formatted_address" ].ToString( ).Split( ',' );
                var count = value.Length;

                return value[ count - 2 ];
            }
            catch ( Exception ex )
            {
                return "";
            }
        }

        /// <summary>
        /// Metodo que obtiene una ubicacion usando la IP.
        /// </summary>
        /// (rruvalcaba)
        /// <param></param>
        /// <returns>Ubiczcion</returns>
        public static Ubicacion ubicacion( )
        {
            var ip = getUserIP();
            var url = "http://freegeoip.net/xml/" + ip;
            WebRequest request = WebRequest.Create(url);
            using (WebResponse response = (HttpWebResponse)request.GetResponse())
            {
                using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
                {
                    DataSet dsResult = new DataSet();
                    dsResult.ReadXml(reader);

                    DataRow row = dsResult.Tables["response"].Select()[0];
                    Ubicacion u = new Ubicacion();
                    u.Latitud = float.Parse(row["Latitude"].ToString(),
System.Globalization.CultureInfo.InvariantCulture);
                    u.Longitud = float.Parse(row["Longitude"].ToString(),
System.Globalization.CultureInfo.InvariantCulture);
                    return u;
                }
            }
        }

        /// <summary>
        /// Metodo que obtiene una ubicacion dado una ciudad
        /// </summary>
        /// (rruvalcaba)
        /// <param>string</param>
        /// <returns>Ubicacion</returns>
        public static Ubicacion buscar(string ciudad)
        {
            try
            {
                var url = "http://maps.googleapis.com/maps/api/geocode/xml?address=" + ciudad +
"&sensor=false";
                XmlDocument doc = new XmlDocument( );
```

```
                doc.Load( url );
                XmlNode element = doc.SelectSingleNode( "//GeocodeResponse/status" );
                if ( element.InnerText != "ZERO_RESULTS" )
                {
                    element = doc.SelectSingleNode( "//GeocodeResponse/result/geometry/location" );
                    var lat = Convert.ToSingle( element[ "lat"
].Value.ToString(System.Globalization.CultureInfo.InvariantCulture));
                    var lon = Convert.ToSingle( element[ "lng"
].Value.ToString(System.Globalization.CultureInfo.InvariantCulture));
                    return new Ubicacion( ) { Latitud = lat , Longitud = lon };
                }
            }
            catch ( Exception ex )
            {

            }
            return ubicacion( );


        }

        /// <summary>
        /// Metodo que obtiene datos desde la API de Google
        /// </summary>
        /// (rruvalcaba)
        /// <param>Ubicacion</param>
        /// <returns>DataRow</returns>
        private static DataRow getDataRow( Ubicacion u )
        {
            var url = "http://maps.googleapis.com/maps/api/geocode/xml?latlng=" +
u.Latitud.ToString(System.Globalization.CultureInfo.InvariantCulture) + "," +
u.Longitud.ToString(System.Globalization.CultureInfo.InvariantCulture) + "&sensor=false";
            WebRequest request = WebRequest.Create( url );
            using ( WebResponse response = ( HttpWebResponse ) request.GetResponse( ) )
            {
                using ( StreamReader reader = new StreamReader( response.GetResponseStream( ) ,
Encoding.UTF8 ) )
                {
                    DataSet dsResult = new DataSet( );
                    dsResult.ReadXml( reader );

                    return dsResult.Tables[ "result" ].Select( )[ 0 ];

                }
            }
        }

        /// <summary>
        /// Metodo que la IP del Usuario
        /// </summary>
        /// (rruvalcaba)
        /// <param>Ubicacion</param>
        /// <returns>DataRow</returns>
        private static string getUserIP( )
        {
            System.Web.HttpContext context = System.Web.HttpContext.Current;
            string ipAddress = context.Request.ServerVariables[ "HTTP_X_FORWARDED_FOR" ];

            if ( !string.IsNullOrEmpty( ipAddress ) )
            {
                string[ ] addresses = ipAddress.Split( ',' );
                if ( addresses.Length != 0 )
                {
                    return addresses[ 0 ];
                }
            }

            if( context.Request.ServerVariables[ "REMOTE_ADDR" ] != "::1")
            {
```

```
                return context.Request.ServerVariables[ "REMOTE_ADDR" ];
            }
            return "" ;
        }
    }
```

# Apéndice C.2: Layout

## Apéndice C.2.1: Layout

```
<!DOCTYPE html>

<html lang="es">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="initial-scale=1.0">
    <title>@ViewBag.Title</title>

    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Cedarville+Cursive|Text+Me+One"
rel="stylesheet">

    <!-- font-awesome -->
    <link href="~/Contents/css/Font-awesome/font-awesome.min.css" rel="stylesheet" />

    <!-- bootstrap -->
    <link href="~/Contents/css/bootstrap/bootstrap.min.css" rel="stylesheet">

    <!-- Layout stylesheet -->
    <link rel="stylesheet" href="~/Contents/css/layout.css" />

    <!--Leaflet Stylesheet-->
    <link rel="stylesheet" href="~/Contents/css/Leaflet/Leaflet.css" />

    <!--Datepicker-->
    <!--<link rel="stylesheet" href="~/Contents/css/bootstrap/datepicker.css" />-->

    <!--jquery-ui-->
    <link rel="stylesheet" href="~/Contents/css/bootstrap/jquery-ui.css" />

    <!--bootstrap-datepicker-->
    <link rel="stylesheet" href="~/Contents/css/bootstrap/bootstrap-datepicker.css" />

    <!--Pluggin Leaflet.awesome-markers-->
    <link rel="stylesheet" href="~/Contents/css/Leaflet/Leaflet.awesome-markers.css" />

    <!--Leaflet.draw-->
    <link rel="stylesheet" href="~/Contents/css/Leaflet/Leaflet.draw.css" />



    @RenderSection("Styles", false)

</head>
<body>

    <div id="wrapper">
        <div id="header" class="container">
            @Html.Partial("~/Views/Shared/_Header.cshtml")
        </div>
        <div id="content">
            @RenderBody()
        </div>
    </div>
    <div id="footer">
```

```
        @Html.Partial("~/Views/Shared/_Footer.cshtml")
    </div>

    <!-- jquery -->
    <script src="~/Contents/js/jquery-3.1.1.min.js"></script>

    <!-- bootstrap -->
    <script src="~/Contents/js/bootstrap.min.js"></script>

    <!--Leaflet-->
    <script src="~/Contents/js/Leaflet.js"></script>

    <!--bootstrap-datepicker.es.min-->
    <script src="~/Contents/js/bootstrap-datepicker.es.min.js"></script>

    <!--bootstrap-datepicker-->
    <script src="~/Contents/js/bootstrap-datepicker.js"></script>

    <!--bootstrap-datepicker-->
    <script src="~/Contents/js/bootstrap-datepicker.min.js"></script>

    <!--jquery-ui.min-->
    <script src="~/Contents/js/jquery-ui.js"></script>

    <!--Leaflet.awesome-marker-->
    <script src="~/Contents/js/Leaflet.awesome-markers.js"></script>

    <!--Leaflet.awesome-markers.min-->
    <script src="~/Contents/js/Leaflet.awesome-markers.min.js"></script>

    <!--Marker.Text-->
    <script src="~/Contents/js/Marker.Text.js"></script>

    <!--Icon.Cavas-->
    <script src="~/Contents/js/Icon.Canvas.js"></script>

    <!--Icon.draw-->
    <script src="~/Contents/js/Leaflet.draw.js"></script>

    <!--Collapse-->
    <script type="text/javascript" src="~/Contents/js/collapse.js"></script>

    <!--Ir atras-->

    <script type="text/javascript">
        $(document).ready(function () {
            $("#date").datepicker({
                dateFormat: 'yy-mm-dd'
            });
        });

    </script>

    <script type="text/javascript">
        function irAtras() {
            window.history.back();
        }
    </script>

    @RenderSection("Scripts", false)
</body>
</html>
```

## Apéndice C.2.2: Header

```
@{
    var cuenta_activa = Proyecto_Integracion.WebApp.Utils.SessionManager.CuentaActiva();
    var Perfil_activo = Proyecto_Integracion.WebApp.Utils.SessionManager.PerfilActivo();
```

```
}

<nav class="navbar navbar-default navbar-lm-white">
    <div class="container-fluid">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand texto-naranja letra-prompt" href="@Url.Action("Index", "Home")">
                <strong>ReportIt</strong>
            </a>
        </div>
        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav">
                <li class="">
                    <a href="@Url.Action("Index", "Home")">
                        <i class="fa fa-home" aria-hidden="true"></i>  Inicio
                    </a>
                </li>
                <li>
                    <a href="@Url.Action("Faq", "Home")">
                        <i class="fa fa-question-circle" aria-hidden="true"></i> FAQ
                    </a>
                </li>
                <li class="busqueda_avanzada">
                    <a href="@Url.Action("BusquedaAvanzada", "Home")">
                        <i class="fa fa-search-plus" aria-hidden="true"></i> Busqueda Avanzada
                    </a>
                </li>
            </ul>
            <ul class="nav navbar-nav navbar-right">
                @{
                    if (cuenta_activa != null)
                    {
                        if (Perfil_activo != null)
                        {
                            <li><a class="texto-naranja letra-prompt nuevo_reporte"
href="@Url.Action("Crear","Reporte")">Nuevo Reporte</a></li>
                                <li>
                                    <a class="texto-naranja letra-prompt dropdown-toggle"
id="dropdownMenu1" data-toggle="dropdown" aria-haspopup="true" aria-expanded="true" href="#">
                                        <span class="fa fa-user"></span>
                                        <strong> @Perfil_activo.Nombre</strong>
                                        <span class="caret texto-naranja"></span>
                                        <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">
                                            <li><a href="@Url.Action("Detalles","Perfil",new { Id =
Perfil_activo.Id})">Mi Perfil</a></li>
                                            <li><a href="@Url.Action("MisReportes","Perfil", new {
page = 1 ,Id = Perfil_activo.Id})">Mis Reportes</a></li>
                                            <li role="separator" class="divider"></li>
                                            <li><a
href="@Url.Action("CambiarPassword","Cuenta")">Cambiar Contraseña</a></li>
                                            <li><a
href="@Url.Action("Salir","Cuenta")">Salir</a></li>
                                        </ul>
                                    </a>
                                </li>
                        }
                    }
                    else
                    {
                        <li><a class="texto-naranja btn_ingresar"
href="@Url.Action("Ingresar","Cuenta")">Ingresar</a></li>
```

```
                }
            }


            </ul>
        </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
</nav>


<div class="page-header letra-prompt text-justify header_1">
    @if (Perfil_activo != null && cuenta_activa != null)
    {
        <h1>ReportIt es una nueva forma de publicar incidentes delictivos.</h1>
        <h1><a href="@Url.Action("Detalles", "Perfil", new { Id = Perfil_activo.Id})" class="texto-
naranja">@Perfil_activo.Nombre</a> aporta un granito de arena! <small>Si ves algo di
algo</small></h1>
    }
    else
    {
    <h1>ReportIt es una nueva forma de publicar incidentes delictivos.</h1>
    <h1>¡<a class="texto-naranja" href="@Url.Action("Crear", "Cuenta")">Registrate</a> y comienza a
reportar todo! <small>Si ves algo di algo</small></h1>
    }
</div>
<div class="jumbotron text-center header_2" style="margin-top:5px">
    <h1 class="letra-prompt label_busca">Busca algun incidente</h1>
    <div class="btn-group btn-group-lg filtros" role="group" aria-label="...">
        <button type="button" class="btn btn-default" id="btn_all" data-toggle="collapse"
href="#collapseAll" aria-expanded="true" aria-controls="collapseAll">Todo</button>
        <button type="button" class="btn btn-default" id="btn_direccion" data-toggle="collapse"
href="#collapseDireccion" aria-expanded="true" aria-controls="collapseDireccion">Dirección</button>
        <button type="button" class="btn btn-default" id="btn_fecha" data-toggle="collapse"
href="#collapseFecha" aria-expanded="true" aria-controls="collapseFecha">Fecha</button>
        <button type="button" class="btn btn-default" id="btn_palabrac" data-toggle="collapse"
href="#collapsePalabra" aria-expanded="true" aria-controls="collapsePalabra">Descripción</button>
        <button type="button" class="btn btn-default" id="btn_incidente" data-toggle="collapse"
href="#collapseIncidente" aria-expanded="true" aria-controls="collapseIncidente">Tipo de
delito</button>
    </div>
    <br />
    <br />
    <br />

    <div class="collapse" id="collapseFecha">
        @Html.Partial("~/Views/Home/_CollapseDropDown_Fecha.cshtml")
    </div>

    <div class="collapse" id="collapseMiUbicacion">
        @Html.Partial("~/Views/Home/_CollapseDropDown_MiUbicacion.cshtml")
    </div>


    <div class="collapse" id="collapseAll">
        @Html.Partial("~/Views/Home/_CollapseDropDown_All.cshtml")
    </div>

    <div class="collapse" id="collapseDireccion">
        @Html.Partial("~/Views/Home/_CollapseDropDown_Direccion.cshtml")
    </div>



    <div class="collapse" id="collapsePalabra">
        @Html.Partial("~/Views/Home/_CollapseDropDown_Palabra.cshtml")
    </div>


    <div class="collapse" id="collapseIncidente">
```

```
        @Html.Partial("~/Views/Home/_CollapseDropDown_TipoIncidente.cshtml", new
Proyecto_Integracion.Models.Reporte())
    </div>
    <br />
</div>
```

### *Apéndice C.2.2.1: _CollapseDropDown_All - Parcial*

```
@using (Html.BeginForm("BusquedaGeneral", "Home", new { page = 1, cantResult = 10, filtro = 5 }, FormMethod.Post))
{//Busqueda por ALL
  <fieldset>
    <div class="col-lg-offset-1 col-lg-10">
      <h3 class="text-left">Todo</h3>
      <div class="input-group">
        <input type="text" name="buscar" id="buscar" class="form-control input-lg" placeholder="Busca un incidente por cualquier
dato..." />
        <span class="input-group-btn">
          <button type="submit" class="btn btn-default btn-lg"><i class="fa fa-search texto-naranja"></i></button>
        </span>
      </div>
    </div>
  </fieldset>
}
```

### *Apéndice C.2.2.2: _CollapseDropDown_Direccion - Parcial*

```
@using (Html.BeginForm("BusquedaGeneral", "Home", new { page = 1, cantResult = 10, filtro = 1 }, FormMethod.Post))
{//Busqueda por Direccion
  <fieldset>
    <div class="col-lg-offset-1 col-lg-10">
      <h3 class="text-left">Dirección</h3>
      <div class="input-group">
        <input type="text" name="buscar" id="buscar" class="form-control input-lg" placeholder="Azcapotzalco, Calle Viveros de la
floresta, San pablo... " />
        <span class="input-group-btn">
          <button type="submit" class="btn btn-default btn-lg"><i class="fa fa-search texto-naranja"></i></button>
        </span>
      </div>
    </div>
  </fieldset>
}
```

### *Apéndice C.2.2.3: _CollapseDropDown_Fecha - Parcial*

```
@using (Html.BeginForm("BusquedaGeneral", "Home", new { page = 1, cantResult = 10, filtro = 2 },
FormMethod.Post))
{//Busqueda por ALL
    <fieldset>
        <div class="col-lg-offset-1 col-lg-10">
            <h3 class="text-left">Fecha</h3>
            <div class="input-group">
                <input type="text" name="date" id="date" class="form-control input-lg"
placeholder="Busca un incidente por cualquier dato..." data-provide="datepicker"/>
                <span class="input-group-btn">
                    <button type="submit" class="btn btn-default btn-lg"><i class="fa fa-search
texto-naranja"></i></button>
                </span>
            </div>
        </div>
    </fieldset>
}
```

### ***Apéndice C.2.2.4**: _CollapseDropDown_Palabra – Parcial*

```
@using (Html.BeginForm("BusquedaGeneral", "Home", new { page = 1, cantResult = 10, filtro = 3},
FormMethod.Post))
{//Busqueda por ALL
    <fieldset>
        <div class="col-lg-offset-1 col-lg-10">
            <h3 class="text-left">Palabra clave</h3>
            <div class="input-group">
                <input type="text" name="buscar" id="buscar" class="form-control input-lg"
placeholder="Busca un incidente por cualquier dato..." />
                <span class="input-group-btn">
                    <button type="submit" class="btn btn-default btn-lg"><i class="fa fa-search
texto-naranja"></i></button>
                </span>
            </div>
        </div>
    </fieldset>
}
```

### ***Apéndice C.2.2.5**: _CollapseDropDown_TipoIncidente – Parcial*

```
@model Proyecto_Integracion.Models.Reporte

@using (Html.BeginForm("BusquedaGeneral", "Home", new { page = 1, cantResult = 10, filtro = 6},
FormMethod.Post))
{
    <fieldset>
        <div class="col-lg-offset-1 col-lg-10">
            <h3 class="text-left">Tipo de Incidente</h3>
            <div class="input-group">
                @Html.EnumDropDownListFor(x => x.Incidente, "--Selecciona--", new { @class = "form-
control input-lg tipo_incidentes_dropdown" })
                <span class="input-group-btn">
                    <button type="submit" class="btn btn-default btn-lg"><i class="fa fa-search
texto-naranja"></i></button>
                </span>
            </div>
        </div>
    </fieldset>
}
```

## Apéndice C.2.3: Footer

```
<footer>
    <div class="container-fluid">
        <hr />
        <div class="row">
            <div class="col-md-4"></div>
            <div class="col-md-4">
                <h3 class="text-center texto-verde letra-prompt">
                    ReportIt
                </h3>
                <p class="text-center sin-margen"><small><a href="@Url.Action("About","Home")"
class="texto-naranja">Hecho en Mexico</a></small></p>
                <p class="text-center sin-margen"><small>@DateTime.Now.Year</small></p>
            </div>
            <div class="col-md-4"></div>
        </div>
    </div>
</footer>
```

## **Apéndice C.2.4**: Collapse Buscador

```
//Mostrar solo Buscador Todo
$('#btn_all').on('click', function () {
    $('#collapseFecha').collapse('hide');
    $('#collapseDireccion').collapse('hide');
    $('#collapsePalabra').collapse('hide');
    $('#collapseIncidente').collapse('hide');
    $('#collapseMiUbicacion').collapse('hide');
})

//Mostrar solo Buscador Direcciòn
$('#btn_direccion').on('click', function () {
    $('#collapseFecha').collapse('hide');
    $('#collapseAll').collapse('hide');
    $('#collapsePalabra').collapse('hide');
    $('#collapseIncidente').collapse('hide');
    $('#collapseMiUbicacion').collapse('hide');
})


//Mostrar solo Buscador Fecha
$('#btn_fecha').on('click', function () {
    $('#collapseAll').collapse('hide');
    $('#collapseDireccion').collapse('hide');
    $('#collapsePalabra').collapse('hide');
    $('#collapseIncidente').collapse('hide');
    $('#collapseMiUbicacion').collapse('hide');
})


//Mostrar solo Buscador Palabra Clave
$('#btn_palabrac').on('click', function () {
    $('#collapseFecha').collapse('hide');
    $('#collapseDireccion').collapse('hide');
    $('#collapseAll').collapse('hide');
    $('#collapseIncidente').collapse('hide');
    $('#collapseMiUbicacion').collapse('hide');
})



//Mostrar solo Buscador Incidente
$('#btn_incidente').on('click', function () {
    $('#collapseFecha').collapse('hide');
    $('#collapseDireccion').collapse('hide');
    $('#collapsePalabra').collapse('hide');
    $('#collapseAll').collapse('hide');
    $('#collapseMiUbicacion').collapse('hide');
})

//Mostrar solo Buscador Incidente
$('#btn_mi_ubicacion').on('click', function () {
    $('#collapseFecha').collapse('hide');
    $('#collapseDireccion').collapse('hide');
    $('#collapsePalabra').collapse('hide');
    $('#collapseAll').collapse('hide');
    $('#collapseIncidente').collapse('hide');
})
```

## **Apéndice C.2.5**: Acerca de (About)

```
@{
    Layout = "~/Views/_Layout.cshtml";
    ViewBag.Title = "Acerca de mi";
}
```

```
<div class="container">
    <h1 class="text-center letra-prompt texto-naranja">ReportIt</h1>
    <br />
    <br />
    <h2 class="text-center letra-prompt texto-naranja">Desarrollado por</h2>
    <div class="row">
        <div class="col-md-12 col-xs-12">
            <a href="https://mx.linkedin.com/in/raul-alberto-ruvalcaba-flores-360064134">
                <img src="~/Contents/img/about/ruben.jpg" class="img-responsive center-block"
style="max-height:200px">
                <p class="text-center"><strong>Raul Ruvalcaba</strong></p>
            </a>
        </div>
    </div>
</div>
```

## **Apéndice C.3**: Controlador básico (HomeController)

```
public class HomeController : Controller
    {
        // GET: Home
        public ActionResult Index()
        {
            Estanteria e = new Estanteria();
            return View(e);
        }

        // GET: Home
        public ActionResult About()
        {
            return View();
        }


        public ActionResult BusquedaGeneral(FormCollection collection, String buscar, int page = 1,
int cantResult = 10, TipoIncidente incidente = 0, int filtro = 5)
        {
            Estanteria e = new Estanteria();
            if (String.IsNullOrEmpty(buscar))
            {
                buscar = "";
            }
            var result = new List<Reporte>();
            switch (filtro)
            {
                case 1: //Dirección
                    result = e.BuscarPorDireccion(buscar, page, cantResult);
                    break;
                case 2: //Fecha
                    if (buscar.Equals(""))
                        buscar = Request.Form["date"];
                    result = e.BuscarPorFecha(buscar, page, cantResult);
                    break;
                case 3: //Palabra Clave
                    result = e.BuscarPorPalabra(buscar, page, cantResult);
                    break;
                case 4: //Ubicacion usamos radio de 40. ALL
                    try
                    {
                        var p = Utils.SessionManager.PerfilActivo();
                        Ubicacion u = p.Ubicacion;
                        var radio = 1;
                        result = e.Buscar(buscar, u, page, cantResult, radio);
                    }
                    catch (Exception ex)
                    { }
                    break;
```

```
                case 5: //All
                    result = e.BuscarPorAll(buscar, page, cantResult);
                    break;
                case 6: //Tipo de Incidente
                    result = e.Buscar(incidente, page, cantResult);
                    break;
            }
            this.ViewBag.Page = page;
            this.ViewBag.Results = cantResult;
            if (incidente != 0 && filtro == 6)
                this.ViewBag.Termino = Utils.EnumHelper<TipoIncidente>.GetDisplayValue(incidente);
            else
            this.ViewBag.Termino = buscar;
            this.ViewBag.Filtro = filtro;

            if (result.Count == cantResult)
            {
                this.ViewBag.More = 1;
            }
            else
            {
                this.ViewBag.More = 0;
            }

            return View(result);
        }

        public ActionResult BusquedaAvanzada()
        {
            return View();
        }

        public ActionResult ResultadosAvanzados(String check_miUbicacion, String palabra, String
fecha, String nombrePerfil, String direccion, TipoIncidente incidente, String Latitud = "0", String
Longitud = "0", int page = 1, int cantResult = 10)
        {
            Estanteria e = new Estanteria();
            var result = new List<Reporte>();

            //var latitud =
Convert.ToDouble(Latitud.ToString(System.Globalization.CultureInfo.InvariantCulture));
            //var longitud =
Convert.ToDouble(Longitud.ToString(System.Globalization.CultureInfo.InvariantCulture));
            var u = new Ubicacion();
            u.Latitud = float.Parse(Latitud.ToString(),
System.Globalization.CultureInfo.InvariantCulture);
            u.Longitud = float.Parse(Longitud.ToString(),
System.Globalization.CultureInfo.InvariantCulture);

            if (palabra == "") palabra = null;
            if (fecha == "") fecha = null;
            if (direccion == "") direccion = null;
            if (nombrePerfil == "") nombrePerfil = null;
            if (check_miUbicacion == null) check_miUbicacion = "";
            if (check_miUbicacion.Equals("True"))
            {
                result = e.BusquedaAvanzada(page, cantResult, palabra, fecha, incidente, direccion,
nombrePerfil, u, 40);
            }
            else
            {
                result = e.BusquedaAvanzada(page, cantResult, palabra, fecha, incidente, direccion,
nombrePerfil);
            }
            this.ViewBag.Page = page;
            this.ViewBag.Results = cantResult;
            this.ViewBag.Palabra = palabra;
            this.ViewBag.Fecha = fecha;
            this.ViewBag.Direccion = direccion;
```

```
            this.ViewBag.NombrePerfil = nombrePerfil;
            this.ViewBag.Check_Ubicacion = check_miUbicacion;
            if (incidente != 0)
                this.ViewBag.Incidente = Utils.EnumHelper<TipoIncidente>.GetDisplayValue(incidente);
            else
                this.ViewBag.Incidente = "0";
            if (u != null)
            {
                this.ViewBag.Latitud = u.Latitud;
                this.ViewBag.Longitud = u.Longitud;
            }
            else
            {
                this.ViewBag.Latitud = 0;
                this.ViewBag.Longitud = 0;
            }

            if (result.Count == cantResult)
            {
                this.ViewBag.More = 1;
            }
            else
            {
                this.ViewBag.More = 0;
            }
            return View(result);
        }

        public ActionResult Faq()
        {
            return View();
        }
    }
}
```

## **Apéndice C.4**: Inicio (Index)

```
@model Proyecto_Integracion.Models.Estanteria
@{
    Layout = "~/Views/_Layout.cshtml";
    ViewBag.Title = "Index";
    var con = 0;
    var perfil_activo = Proyecto_Integracion.WebApp.Utils.SessionManager.PerfilActivo();
}
@section Styles{
    <link href="~/Contents/css/Maps.css" rel="stylesheet" />
    <link href="~/Contents/css/index.css" rel="stylesheet" />
}
@section Scripts{
    <!--<script src="~/Contents/js/Google-Maps-Night.js"></script>-->
    <script src="~/Contents/js/maps-buscador.js"></script>
}


<div class="container">
    <br />
    <hr />
    <br />
    <div class="panel panel-default panel-centro-ayuda">
        <div class="panel-heading">
            <h3 class="panel-title">Ultimos Incidentes</h3>
        </div>
        <div class="panel-body" id="map">
        </div>
    </div>
```

```
    @{
        //var ubicacion = Proyecto_Integracion.WebApp.Utils.GeoLocation.ubicacion();
        var reportes = Model.BuscarPorAll("", 0, 10);
        <input id="report_count" value="@reportes.Count" hidden />
        var i = 0;
        foreach (var reporte in @reportes)
        {


            var incidente =
Proyecto_Integracion.WebApp.Utils.EnumHelper<Proyecto_Integracion.Models.TipoIncidente>.GetDisplayVa
lue(reporte.Incidente);
            <input id="incidente+@i" value="@reporte.Incidente" hidden />
            <input id="Latitud+@i" name="Latitud+@i" value="@reporte.Ubicacion.Latitud" hidden />
                <input id="Longitud+@i" value="@reporte.Ubicacion.Longitud" hidden />
            i++;


        }
    }



    <div class="row">
        <hr />
        <div class="col-md-7">
            <br />
            <br />
            <br />
            <br />
            <br />
            <h2 class="featurette-heading letra-prompt">¿Te  han asaltado y no hiciste nada?</h2>
            <p class="lead">Reporta tus incidentes y asi nos ayudamos todos a tener más información
y más seguridad.</p>
        </div>
        <div class="col-md-5">
            <div style="background-color: white">
                <img class="img-responsive center-block img-thumbnail"
src="~/Contents/img/Index/stop.jpg" alt="Generic placeholder image">
            </div>
        </div>
    </div>
    <hr>
    <div class="row">
        <div class="col-md-5">
            <img class="img-responsive center-block img-thumbnail"
src="~/Contents/img/Index/manos_sucias.jpg" alt="Generic placeholder image">
        </div>
        <div class="col-md-7">
            <h2 class="featurette-heading letra-prompt">Somos una aplicación que se interesa por tu
bienestar, por ello ponemos a tu alcance información de incidentes delictivos por tu CDMX.</h2>
            <p class="lead">Nuestro objetivo es ayudarte a tener mejor seguridad conociendo la
incidencia de delitos.</p>
            <p class="lead"><big><a class="texto-naranja" href="@Url.Action("Crear",
"Cuenta")">Regístrate</a></big>, busca un delitoy comienza a cuidar de ti y tu familia.</p>
        </div>
    </div>
    <br />
    <br />
    <br />
</div>
```

## Apéndice C.5: Faq

```
@{
    ViewBag.Title = "Preguntas Frecuentes";
    Layout = "~/Views/_Layout.cshtml";
```

```
}

@section Scripts{
}

@section Styles{
    <link rel="stylesheet" href="~/Contents/css/faq.css" />
    <link rel="stylesheet" href="~/Contents/css/dotText.css" />
}

<div class="container">
    <div class="col-md-12 col-sm-12 col-xs-12 col-md-offset-1">
        <br />
        <h2 class="letra-prompt"><strong>Preguntas Frecuentes</strong></h2>
        <hr />
    </div>
    <div class="col-md-10 col-sm-12 col-xs-12 col-md-offset-1">
        <div class="tab-content">
            <div class="tab-pane active in fade" id="cuenta-tab">
                <div class="panel-group">
                    <div class="panel panel-default panel-faq">
                        <div class="panel-heading">
                            <a data-toggle="collapse" data-parent="#accordion-cat-1"
href="#libromatico-faq">
                                <h4 class="panel-title">
                                    ¿Qué es ReportIt?
                                    <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                                </h4>
                            </a>
                        </div>
                        <div id="libromatico-faq" class="panel-collapse collapse">
                            <div class="panel-body">
                                ReportIt es una Aplicación con fines de ayuda a la sociedad
aportando información de publicaciones de incidentes delictivos con localización.
                            </div>
                        </div>
                    </div>
                    <div class="panel panel-default panel-faq">
                        <div class="panel-heading">
                            <a data-toggle="collapse" data-parent="#accordion-cat-1" href="#cuenta-
crear-faq">
                                <h4 class="panel-title">
                                    ¿Cómo crear una cuenta?
                                    <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                                </h4>
                            </a>
                        </div>
                        <div id="cuenta-crear-faq" class="panel-collapse collapse">
                            <div class="panel-body">
                                Para crear una cuenta accede <a href="@Url.Action("Crear",
"Cuenta")">aquí</a>.
                            </div>
                        </div>
                    </div>
                    <div class="panel panel-default panel-faq">
                        <div class="panel-heading">
                            <a data-toggle="collapse" data-parent="#accordion-cat-1" href="#cuenta-
gestionar-perfiles-faq">
                                <h4 class="panel-title ">
                                    ¿Por qué debo tener una cuenta?
                                    <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                                </h4>
                            </a>
                        </div>
                        <div id="cuenta-gestionar-perfiles-faq" class="panel-collapse collapse">
                            <div class="panel-body">
```

```
                            Tener una cuenta te da la posibilidad de realizar publicaciones y
asi aportar a nuestros demas usuarios como tu a tener mas información sobre la seguridad en la CDMX.
                        </div>
                    </div>
                </div>
                <div class="panel panel-default panel-faq">
                    <div class="panel-heading">
                        <a data-toggle="collapse" data-parent="#accordion-cat-1" href="#item-
poseer-faq">
                            <h4 class="panel-title">
                                ¿Cuántos Reportes de incidentes puedo tener?
                                <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                            </h4>
                        </a>
                    </div>
                    <div id="item-poseer-faq" class="panel-collapse collapse">
                        <div class="panel-body">
                            Todos los que tu quieras!.
                        </div>
                    </div>
                </div>
                <div class="panel panel-default panel-faq">
                    <div class="panel-heading">
                        <a data-toggle="collapse" data-parent="#accordion-cat-1" href="#perfil-
agregar-item-faq">
                            <h4 class="panel-title">
                                ¿Cómo agregar un nuevo Reporte?
                                <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                            </h4>
                        </a>
                    </div>
                    <div id="perfil-agregar-item-faq" class="panel-collapse collapse">
                        <div class="panel-body">
                            Para agregar un nuevo Reporte necesitas estar autenticado, si lo
estás, debes dar click en <strong><a href="@Url.Action("Crear", "Reporte")">Nuevo</a></strong>, o
desde tu perfil en <strong><a href="@Url.Action("Crear", "Reporte")">Nuevo Reporte</a></strong>
colocale una fecha, una descripción y un tipo de incidente y coloca en el mapa la ubicación del
incidente que viste o fuiste victima.
                        </div>
                    </div>
                </div>
                <div class="panel panel-default panel-faq">
                    <div class="panel-heading">
                        <a data-toggle="collapse" data-parent="#accordion-cat-1" href="#perfil-
eliminar-item-faq">
                            <h4 class="panel-title">
                                ¿Cómo eliminar uno de mis Reportes?
                                <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                            </h4>
                        </a>
                    </div>
                    <div id="perfil-eliminar-item-faq" class="panel-collapse collapse">
                        <div class="panel-body">
                            Puedes eliminar un Reporte cuando quieras, debes entrar a visualizar
tu reporte desde <strong>Mis Reportes</strong>,
                            dar click en el y entrar en detalles, y finalmente ahí encontrarás
el ícono eliminar en color rojo en su esquina superior derecha.
                        </div>
                    </div>
                </div>
                <div class="panel panel-default panel-faq">
                    <div class="panel-heading">
                        <a data-toggle="collapse" data-parent="#accordion-cat-1" href="#item-
editar-faq">
                            <h4 class="panel-title">
                                ¿Cómo editar uno de mis Reportes?
```

```
                                    <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                                </h4>
                            </a>
                        </div>
                        <div id="item-editar-faq" class="panel-collapse collapse">
                            <div class="panel-body">
                                Si estás autenticado y tienes Reportes que deseas editar, debes ir a
<strong>Mis Reportes</strong> y hacer click en alguno de ellos, entonces ahí encontrarás la opción
para <strong>Módificar</strong>
                            </div>
                        </div>
                    </div>
                    <div class="panel panel-default panel-faq">
                        <div class="panel-heading">
                            <a data-toggle="collapse" data-parent="#accordion-cat-1" href="#item-
agregar-prop-faq">
                                <h4 class="panel-title">
                                    ¿Cómo realizar una busqueda de un Reporte?
                                    <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                                </h4>
                            </a>
                        </div>
                        <div id="item-agregar-prop-faq" class="panel-collapse collapse">
                            <div class="panel-body">
                                <p class="text-justify">!Para realizar las busquedas de reportes no
debes estar autentificado!.</p>
                                <p class="text-justify">
                                    Nuestros <strong>Filtros</strong> con los que podrás realizar
las busquedas son los siguientes:
                                    <ul>
                                        <li>
                                            <strong>Por Dirección</strong>
                                        </li>
                                        <li>
                                            <strong>Por Fecha</strong>
                                        </li>
                                        <li>
                                            <strong>Por Palabra Clave</strong>
                                        </li>
                                        <li>
                                            <strong>Por Incidente</strong>
                                        </li>
                                        <li>
                                            <strong>Por Todo</strong>
                                        </li>
                                        <li>
                                            <strong>Por Mi Ubicación**</strong>
                                        </li>
                                    </ul>
                                </p>
                                <p class="text-left"><strong>Buscador en la página de
inicio:</strong></p>
                                <p class="text-justify">Desde nuestra página de inicio encontraras
nuestras opciones para realizar busquedas por diferentes filtros.</p>
                                <p class="text-left"><strong>Buscador desde cualquier otra
página</strong></p>
                                <p class="text-justify">
                                    Desde cuaquier otra vista de nuestra aplicación encontraras un
buscador en la parte superior debajo de nuestro menu,
                                    ahí podrás realizar busquedas por cualquier tipo de filtro. Por
default tenemos el filtro de buscar<strong>Por Todo</strong> que realiza un busqueda en todos los
campos de un Reporte.
                                </p>
                                <br />
                                <p class="text-justify">
```

```
                                    Recuerda que una vez que selecciones el filtro que desees en el
buscador, este se permanecera guardado para que realices todas las busquedas que quieras hasta que
decidas cambiar el filtro de busqueda.
                                </p>
                                <p class="text-justify">
                                    **El filtro de <strong>Mi Ubicación</strong> únicamente está
disponible para usuarios de la aplicación autentificados.
                                </p>
                                <p class="text-justify">
                                    Para realizar <strong>Busquedas por Fecha</strong> deberas
seleccionar el filtro <strong> Por Fecha</strong> y en nuestro buscador escribir la fecha con el
formato <strong>yyyy-mm-dd.</strong>
                                </p>
                                <p class="text-justify">
                                    Para realizar <strong>Busquedas por Tipo de Incidente</strong>
deberas seleccionar el filtro <strong> Por Incidente</strong> y en nuestro buscador escribir alguna
de las siquentes cadenas:
                                    <ul>
                                        <li>
                                            <strong>#1</strong> : para busqueda por Homicidio
                                        </li>
                                        <li>
                                            <strong>#2</strong> : para busqueda por Suicidio
                                        </li>
                                        <li>
                                            <strong>#3</strong> : para busqueda por Robo o Asalto
                                        </li>
                                        <li>
                                            <strong>#4</strong> : para busqueda por Violación
                                        </li>
                                        <li>
                                            <strong>#5</strong> : para busqueda por Extorción Sexual
                                        </li>
                                    </ul>
                                </p>
                                <p>
                                    Para los demas filtros:
                                    <ul>
                                        <li>
                                            <strong>Por Dirección</strong> : Realiza busquedas por
direcciones específicas de los reportes.
                                        </li>
                                        <li>
                                            <strong>Por Palabra Clave</strong> : Realiza busquedas
por palabras claves contenidas en la descripción de un reporte.
                                        </li>
                                    </ul>
                                </p>

                            </div>
                        </div>
                    </div>
                    <div class="panel panel-default panel-faq">
                        <div class="panel-heading">
                            <a data-toggle="collapse" data-parent="#accordion-cat-1"
href="#prestamo-ver-faq">
                                <h4 class="panel-title">
                                    ¿Cómo ver un Reporte?
                                    <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                                </h4>
                            </a>
                        </div>
                        <div id="prestamo-ver-faq" class="panel-collapse collapse">
                            <div class="panel-body">
                                Si quieres ver tus Reportes, debes hacer click en el botón del
extremo superior derecho y se desplegara la opción:
                                <br>
                                <ul>
```

```
                                    <li>
                                        <strong>Mis Reportes:</strong> Todo los Reportes que has
realizado
                                    </li>
                                </ul>
                            </div>
                        </div>
                    </div>
                    <div class="panel panel-default panel-faq">
                        <div class="panel-heading">
                            <a data-toggle="collapse" data-parent="#accordion-cat-1"
href="#prestamo-cancelar-faq">
                                <h4 class="panel-title">
                                    ¿Cómo módificar o eliminar mi Perfil?
                                    <span class="pull-right span-icono"><i class="glyphicon
glyphicon-chevron-down icono-despliegue"></i></span>
                                </h4>
                            </a>
                        </div>
                        <div id="prestamo-cancelar-faq" class="panel-collapse collapse">
                            <div class="panel-body">
                                Si quieres <strong>Módificar</strong> o <strong> Eliminar</strong>
tu perfil debes hacer click en el botón del extremo superior derecho y se desplegara la opción:
                                <br>
                                <ul>
                                    <li>
                                        <strong>Mi Perfil</strong>
                                    </li>
                                </ul>
                                Te redireccionará a ver tu perfil y en la esquina superior derecha
encontrarás las dos opciones para realizar la Módificación o Eliminación de tu Perfil, pero te
sugerimos que no lo hagas;
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

# **Apéndice C.6**: CuentaController

```
    public class CuentaController : Controller
    {
        // GET: Cuenta
        public ActionResult Ingresar()
        {
            Cuenta c = Utils.SessionManager.CuentaActiva();
            if (c != null)
            {
                return RedirectToAction("Index", "Home");
            }
            return View();

        }

        [HttpPost]
        public ActionResult Ingresar(Cuenta c)
        {
            if (!Utils.Validator.isNullOrEmptyOrWhiteSpace(new List<String>() { c.Email,
c.Contrasena }) && c.IniciarSesion())
            {
                Utils.SessionManager.Ingresar(c.Email);
                Perfil perfil = new Perfil();
                perfil.Seleccionar(c.Email);
                Utils.SessionManager.RegistarPerfil(perfil.Id);
```

```
                return RedirectToAction("Index", "Home");
            }
            Utils.UIWarnings.SetError("Su Email o Contraseña es incorrecto");
            return View();
        }

        public ActionResult Salir()
        {
            Utils.SessionManager.Salir();
            return RedirectToAction("Index", "Home");
        }

        //Get : Cuenta
        public ActionResult Crear()
        {
            Cuenta c = Utils.SessionManager.CuentaActiva();
            if (c != null)
            {
                return RedirectToAction("Index", "Home");
            }
            return View();
        }

        [HttpPost]
        public ActionResult Crear(Cuenta c, Perfil p, Ubicacion u)
        {
            if (!Utils.Validator.isNullOrEmptyOrWhiteSpace(new List<String>() { c.Email,
c.Contrasena, p.Nombre}) && Utils.Validator.esValido(c.Email))
            {
                if (c.Crear())
                {
                    Utils.SessionManager.Ingresar(c.Email);
                    p.Cuenta = c;
                    p.UrlImagen = "PICON_023.png"; //imagen por default
                    u.Direccion = Utils.GeoLocation.direccion(u);
                    if (u.Crear())
                    {
                        p.Ubicacion = u;
                        p.Crear();
                        Utils.SessionManager.RegistarPerfil(p.Id);

                        String body = "Hola,<br> Bienvenido a ReportIt,<br> Esperamos que nuestra
aplicación te sea de ayuda<br><br>Saludos";

                        Utils.Email.SendEmail("Bienvenido a ReportIt", c.Email, body);
                        return RedirectToAction("Index", "Home");
                    }
                }
                else
                {
                    Utils.UIWarnings.SetError("Ya existe una cuenta usando este Email");
                    return RedirectToAction("Ingresar", "Cuenta");
                }
            }
            Utils.UIWarnings.SetError("Campos invalidos");
            return RedirectToAction("Ingresar", "Cuenta");

        }
        [HttpPost]
        public ActionResult OlvidePassword(Cuenta p)
        {
            string nuevopassword = p.CreatePassword(8);
            if (p.CambiarContrasena(nuevopassword))
            {
                string subject = "Nueva contraseña ReportIt";
                string mensaje = string.Format("Hola, <br><br> Tu nueva contraseña es: {0}.
<br><br>Saludos<br><br>ReportIt", nuevopassword);

                Utils.Email.SendEmail(subject, p.Email, mensaje);
```

```
                Utils.UIWarnings.SetInfo("Su nueva contraseña ha sido enviada al correo: " +
p.Email);
                return RedirectToAction("Ingresar", "Cuenta");
            }
            Utils.UIWarnings.SetError("Lo sientimos, No se pudo recuperar la contraseña.");
            return RedirectToAction("Ingresar", "Cuenta");
        }

        [HttpPost]
        public ActionResult CambiarContrasena(Cuenta p, string confirm_password)
        {
            if (p != null)
            {
                if (p.Contrasena == confirm_password)
                {
                    if (p.CambiarContrasena(p.Contrasena))
                    {
                        Utils.UIWarnings.SetInfo("Se ha cambiado su contraseña Exitosamente");
                        return RedirectToAction("Index", "Home");
                    }
                }
                Utils.UIWarnings.SetError("Lo sientimos, No se pudo cambiar la contraseña.");
                return RedirectToAction("Index", "Home");
            }
            Utils.UIWarnings.SetError("Usted no tiene los permisos para cambiar una contraseña.");
            return RedirectToAction("Index", "Home");
        }

        public ActionResult CambiarPassword()
        {
            Cuenta c = Utils.SessionManager.CuentaActiva();
            if (c != null)
            {
                return View(c);
            }
            Utils.UIWarnings.SetError("Debe estar autenticado para cambiar su contraseña.");
            return RedirectToAction("Ingresar", "Cuenta");
        }

        public ActionResult Eliminar(Cuenta c)
        {
            if(Utils.SessionManager.CuentaActiva() != null && Utils
                .SessionManager.PerfilActivo() != null)
            {
                if (c.Eliminar())
                {
                    Utils.SessionManager.Salir();
                }
            }
            return RedirectToAction("Index","Home");

        }
    }
```

## **Apéndice C.7**: PerfilController

```
public class PerfilController : Controller
    {
        // GET: Perfil
        public ActionResult Detalles(long Id)
        {
            Cuenta cuenta = Utils.SessionManager.CuentaActiva();
            Perfil perfilActivo = Utils.SessionManager.PerfilActivo();
            Perfil perfil = new Perfil();
            perfil.Seleccionar(Id);
            if (cuenta != null && perfilActivo != null)
```

```
            {
                return View(perfil);
            }
            else
            {
                return RedirectToAction("Ingresar", "Cuenta");
            }
        }

        [HttpPost]
        public ActionResult CargarImagen(Int64 Id)
        {
            string directory = System.AppDomain.CurrentDomain.BaseDirectory +
@"Contents\img\perfiles";
            HttpPostedFileBase photo = Request.Files["photo"];
            var perfil = new Perfil();
            perfil.Seleccionar(Id);
            try
            {
                if (photo != null && photo.ContentLength > 0 &&
Utils.Validator.verificarExtension(Path.GetExtension(photo.FileName)))
                {
                    if (photo.FileName.Length > 30)
                    {
                        Utils.UIWarnings.SetError("Nombre de la imagen demasiado largo");
                        return RedirectToAction("Detalles", "Perfil", new { Id = Id });
                    }
                    var fileName = Guid.NewGuid().ToString() + "_" +
Path.GetFileName(photo.FileName);

                    photo.SaveAs(Path.Combine(directory, fileName));
                    perfil.UrlImagen = fileName;
                    if (!perfil.ModificarImagen())
                    {
                        Utils.UIWarnings.SetError("No se pudo agregar la imagen");
                        return RedirectToAction("Detalles", "Perfil", new { Id = Id });
                    }
                    else
                    {
                        Utils.UIWarnings.SetInfo("Modificación realizada");
                        return RedirectToAction("Detalles", "perfil", new { Id = Id });
                    }
                }
                else
                {
                    Utils.UIWarnings.SetError("El archivo no corresponde a una imagen");
                    return RedirectToAction("Detalles", "perfil", new { Id = Id });
                }
            }
            catch (Exception e)
            {
                return RedirectToAction("Index", "Home");
            }
        }

        //GET:Modificar
        public ActionResult Modificar(long Id)
        {

            if(Utils.SessionManager.CuentaActiva() !=null && Utils.SessionManager.PerfilActivo() !=
null && Utils.SessionManager.PerfilActivo().Id == Id)
            {
                Perfil p = new Perfil();
                p.Seleccionar(Id);
                return View(p);
            }
            return RedirectToAction("Index","Home");
        }
```

```
        [HttpPost]
        public ActionResult Modificar(Perfil p, Ubicacion u)
        {
            if (!Utils.Validator.isNullOrEmptyOrWhiteSpace(new List<String>() { p.Nombre,
Convert.ToString(u.Latitud), Convert.ToString(u.Longitud) }))
            {
                var nombre = p.Nombre;
                p.Seleccionar(p.Id);
                p.Nombre = nombre;
                p.Ubicacion.Latitud = u.Latitud;
                p.Ubicacion.Longitud = u.Longitud;
                p.Ubicacion.Direccion = Utils.GeoLocation.direccion(p.Ubicacion);
                if (p.Modificar() && p.Ubicacion.Modificar())
                {
                    Utils.UIWarnings.SetInfo("Módificacion dde perfil exitosa");
                    return RedirectToAction("Detalles", "Perfil", new { Id = p.Id });
                }
            }
            else
            {
                Utils.UIWarnings.SetError("No tiene los permisos para realizar modificaciones a este
perfil");
                return RedirectToAction("Index", "Home");
            }
            return View();
        }

        public ActionResult MisReportes(int? page, long Id)
        {
            Cuenta cuenta = Utils.SessionManager.CuentaActiva();
            Perfil perfilActivo = Utils.SessionManager.PerfilActivo();
            Perfil p = new Perfil();

            if (cuenta != null && perfilActivo != null && p.Seleccionar(Id))
            {
               List<Proyecto_Integracion.Models.Reporte> items = p.MisReportes();
                int pageSize = 10;
                int pageNumber = (page ?? 1);
                return View(items.ToPagedList(pageNumber, pageSize));
            }
            return RedirectToAction("Ingresar", "Cuenta");
        }
    }
}
```

## **Apéndice C.7.1**: Carga de imágenes (JavaScript)

```
$(document).ready(function () {
    $(document).on('change', ':file', function () {
        var input = $(this),
        numFiles = input.get(0).files ? input.get(0).files.length : 1,
        label = input.val().replace(/\\/g, '/').replace(/.*\//, '');
        input.trigger('fileselect', [numFiles, label]);
    });

    $(':file').on('fileselect', function (event, numFiles, label) {
        var input = $(this).parents('.input-group').find(':text'),
        log = numFiles > 1 ? numFiles + ' files selected' : label;
        if (input.length) {
            input.val(log);
        } else {
            if (log) alert(log);
        }
    });
    $('.filefield').on('click', function (event, numFiles, label) {
        var fileinput = $(this).parents('.input-group').find(':file')
        if (fileinput != null) { fileinput.focus().trigger('click'); }
```

```
    });
});
```

## **Apéndice C.8**: Registrar ubicación en mapa (JavaScript)

```
var loadMap = function (id) {

    var HELSINKI = [60.1708, 24.9375];
    var map = L.map(id);
    var marker;
    var circle;
    var tile_url = 'http://{s}.tile.osm.org/{z}/{x}/{y}.png';
    var layer = L.tileLayer(tile_url, {
        attribution: 'OSM'
    });
    map.addLayer(layer);
    map.setView(HELSINKI, 19);

    map.locate({ setView: true, watch: false}) /* This will return map so you can do chaining */
        .on('locationfound', function (e) {
            marker = L.marker([e.latitude, e.longitude]).bindPopup('Ubicacion actual');
            console.log(e.latitude);
            console.log(e.longitude);
            var long = document.getElementById('Longitud');
            long.value = e.longitude

            var lati = document.getElementById('Latitud');
            lati.value = e.latitude;
            circle = L.circle([e.latitude, e.longitude], e.accuracy / 2, {
                weight: 1,
                color: 'blue',
                fillColor: '#cacaca',
                fillOpacity: 0.2
            });
            map.addLayer(marker);
            map.addLayer(circle);
        })
       .on('locationerror', function (e) {
           console.log(e);
           alert("Location access denied.");
       });
    map.on('click', function (e) {
        var popLocation = e.latlng;
        var long = document.getElementById('Longitud');
        long.value = popLocation.lng;

        var lati = document.getElementById('Latitud');
        lati.value = popLocation.lat;
        console.log(marker);
        if (marker && circle) {
            marker.setLatLng([popLocation.lat, popLocation.lng]);
            circle.setLatLng([popLocation.lat, popLocation.lng]);
        } else {
            marker = L.marker([popLocation.lat, popLocation.lng]).addTo(map);
            circle = L.circle([popLocation.lat, popLocation.lng], e.accuracy / 2, {
                weight: 1,
                color: 'blue',
                fillColor: '#cacaca',
                fillOpacity: 0.2
            }).addTo(map);
        }
        var popup = L.popup()
        .setLatLng(popLocation)
        .setContent('<p>Ubicación nueva</p>')
        .openOn(map);
    });
};

loadMap('map');
```

# **Apéndice C.9**: Crear Cuenta (JavaScript)

```
@{
    Layout = "~/Views/_Layout.cshtml";
    ViewBag.Title = "Crear Cuenta";
}

@section Styles{
    <link href="~/Contents/css/Maps.css" rel="stylesheet" />
    <link href="~/Contents/css/Crear_Cuenta.css" rel="stylesheet" />
}

@section Scripts{
    <script src="~/Contents/js/maps-registrar.js"></script>
    <script type="text/javascript">
            $(".btn_ingresar").hide();
            $(".header_2").hide();
            $(".header_1").hide();
    </script>
}


<div class="container">
    <br />
    <h2 class="letra-prompt"><strong>Crear Cuenta</strong></h2>
    <hr />
    <div class="col-md-offset-2 col-md-8 ">

        <div class="panel panel-default">
            <div class="panel-heading">
                <h3 class="panel-title letra-prompt">Registrar</h3>
            </div>
            @using (Html.BeginForm("Crear", "Cuenta", FormMethod.Post))
            {
                <div class="panel-body">
                    <!--<form class="form-horizontal">-->
                    <div class="form-group">
                        <label for="Email" class="letra-prompt">Email</label>
                        <input type="email" name="Email" class="form-control input-lg"
placeholder="Email">
                    </div>
                    <div class="form-group">
                        <label for="Contrasena" class="letra-prompt">Contraseña</label>
                        <input type="password" name="Contrasena" class="form-control input-lg"
placeholder="Contraseña" maxlength="15" minlength="6">
                        <p><small>Introduce contraseña mayor 6 caracteres</small></p>
                    </div>
                    <div class="form-group">
                        <label for="Nombre" class="letra-prompt">Nombre</label>
                        <input type="text" name="Nombre" class="form-control input-lg"
placeholder="Juan Perez" maxlength="15" minlength="6">
                    </div>
                    <div class="form-group">
                        <!--<label for="Latitud" class="letra-prompt">Latitude</label>-->
                        <input type="text" name="Latitud" id="Latitud" hidden>
                    </div>
                    <div class="form-group">
                        <!--<label for="Longitud" class="letra-prompt">Longitud</label>-->
                        <input type="text" name="Longitud" id="Longitud" hidden>
                    </div>

                    <!--</form>-->
                    <div class="form-group">
                        <div class="panel panel-default mapa-mi-ubicacion">
                            <div class="panel-heading">
                                <h3 class="panel-title letra-prompt">Mi Ubicación</h3>
                            </div>
```

```
                        <div class="panel-body" id="map">
                        </div>
                    </div>
                </div>
                <br />
            </div>
            <div class="panel-footer text-right">
                <input class="btn btn-primary" value="Registrame" type="submit" />
                <a href="@Url.Action("Index", "Home")" type="button" class="btn btn-
danger">Cancelar</a>
            </div>

        }
    </div>
</div>
</div>
```

## **Apéndice C.10**: Visualizar en un mapa (JavaScript)

```
var loadMap = function (id) {
    var lati = document.getElementById('Latitud').value;
    var long = document.getElementById('Longitud').value;
    console.log(parseFloat(lati)+" "+parseFloat (long));
    var HELSINKI = [parseFloat(lati), parseFloat(long)];
    var map = L.map(id);
    var marker = L.marker([parseFloat(lati), parseFloat(long)]).bindPopup('Ubicación actual');
    var circle = L.circle([parseFloat (lati),parseFloat (long)],{
        weight: 1,
        color: 'blue',
        fillColor: '#cacaca',
        fillOpacity: 0.2
    });
    map.addLayer(marker);
    map.addLayer(circle);
    var tile_url = 'http://{s}.tile.osm.org/{z}/{x}/{y}.png';
    var layer = L.tileLayer(tile_url, {
        attribution: 'OSM'
    });
    map.addLayer(layer);
    map.setView(HELSINKI, 19);
};

loadMap('map');
```

## **Apéndice C.11**: Detalles de un Perfil

```
@model Proyecto_Integracion.Models.Perfil
@{
    Layout = "~/Views/_Layout.cshtml";
    ViewBag.Title = "Detalles Reporte";
    var Perfil_Activo = Proyecto_Integracion.WebApp.Utils.SessionManager.PerfilActivo();
    var Cuenta_Activa = Proyecto_Integracion.WebApp.Utils.SessionManager.CuentaActiva();
}
@section Styles{
    <link href="~/Contents/css/detalles-perfil.css" rel="stylesheet" />
    <link href="~/Contents/css/Mis-Reportes.css" rel="stylesheet" />
}
@section Scripts{
    <script src="~/Contents/js/map-visualizar.js" type="text/javascript"></script>
    <script src="~/Contents/js/imagen.js"></script>
}


<div class="container">
    <br />
```

```
    @if(Perfil_Activo != null && Cuenta_Activa != null && Perfil_Activo.Id == Model.Id)
    {
    <h2 class="letra-prompt"><strong>Mi Perfil</strong></h2>
    }
    else
    {
        <h2 class="letra-prompt"><strong>Detalles de Perfil</strong></h2>
    }
    <hr />
    <div class="col-md-3">
        <div class="thumbnail text-center">
            <img src="~/Contents/img/perfiles/@Perfil_Activo.UrlImagen" alt="Imagen perfil">
            @if (Perfil_Activo.Id == Model.Id && Perfil_Activo != null && Cuenta_Activa != null)
            {
                <div class="caption">
                    <a class="btn btn-success" data-toggle="modal" data-target="#Imagen">
                        Módificar
                        <i class="glyphicon glyphicon-pencil"></i>
                    </a>
                </div>
            }
        </div>
        <div class="row text-center">
            <br />
            <br />
            <a class="btn btn-success" href="@Url.Action("Index","Home")">
                <i class="fa fa-chevron-left" aria-hidden="true"></i>
                 Regresar
            </a>
        </div>
    </div>
    <div class="col-md-7">
        <div class="panel panel-default panel-mis-datos">
            <div class="panel-body">
                <p><strong>Nombre:</strong> @Model.Nombre</p>
                <p><strong>Dirección:</strong> @Model.Ubicacion.Direccion</p>
                <input type="text" id="Latitud" value="@Model.Ubicacion.Latitud" hidden />
                <input type="text" id="Longitud" value="@Model.Ubicacion.Longitud" hidden />
            </div>
            <div class="panel-footer panel-ubicacion-dettallesP" id="map"></div>
        </div>
    </div>
    <div class="col-md-2 text-right">
        @if (Perfil_Activo != null && Cuenta_Activa != null && Perfil_Activo.Id == Model.Id)
        {

            <a class="btn btn-warning" href="@Url.Action("Modificar", "Perfil", new { Id = Model.Id
})">
                Módificar
                <i class="glyphicon glyphicon-pencil"></i>
            </a>
            <br />
            <br />
            <br />
            <a class="btn btn-danger" data-toggle="modal" data-target="#modal-eliminar">
                Eliminar  
                <i class="glyphicon glyphicon-remove"></i>
            </a>
        }
    </div>
@if (Perfil_Activo != null && Cuenta_Activa != null && Perfil_Activo.Id == Model.Id)
{
    <div class="col-md-12">
        <div class="text-center">
            <a href="@Url.Action("Crear", "Reporte")" type="button" class="btn btn-warning letra-
prompt btn-lg">Nuevo Reporte <i class="fa fa-plus-square" aria-hidden="true"></i></a>
            <br />
            <br />
        </div>
```

```
    </div>
}
    <!--Mis ultimos reportes-->
    <div class="col-md-12">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h4 class="panel-title"> Mis últimos reportes</h4>
            </div>
            <div class="panel-body">
                <div class="list-group">
                    <div class="row" style="display: flex; flex-wrap: wrap;">

                        @{
                            var mis_reportes = Model.MisReportes();
                            if (mis_reportes.Count() == 0)
                            {
                                <div class="col-xs-12 text-center">
                                    <h4><strong>Aún no cuentas con algún reporte</strong></h4>
                                </div>
                            }
                            if (mis_reportes.Count() > 4)
                            {
                                <div class="col-xs-12 text-right"> <a class="btn btn-xs"
href="@Url.Action("MisReportes", "Perfil", new { page = 1, id = Model.Id })"><i class="fa fa-angle-
down"></i> Ver todos</a> </div>
                            }
                            foreach (var reporte in mis_reportes.Take(4))
                            {

                                <div class="col-md-6 reporte">

                                    <a href="@Url.Action("Detalles", "Reporte", new { Id =
reporte.Id})" class="list-group-item">

                                        <div class="media">
                                            <div class="media-left media-middle">
                                                <i class="fa fa-map-marker fa-3x texto-naranja"
aria-hidden="true"></i>
                                            </div>
                                            <div class="media-body text-justify">
                                                @{
                                                    var incidente =
Proyecto_Integracion.WebApp.Utils.EnumHelper<Proyecto_Integracion.Models.TipoIncidente>.GetDisplayVa
lue(reporte.Incidente);
                                                    <h4 class="media-heading"><strong>Incidente:
</strong>@incidente</h4>
                                                }
                                                <p><strong>Descripción:
</strong>@reporte.Descripcion</p>
                                                <p><strong>Fecha: </strong>
@reporte.FechaExpedicion.ToString("d")</p>
                                                <p><strong>Dirección: </strong>
@reporte.Ubicacion.Direccion</p>
                                            </div>
                                        </div>

                                    </a>
                                </div>

                                                    }
                        }

                    </div>
                </div>

            </div>
        </div>
    </div>
```

```
</div>


<div id="Imagen" class="modal fade" role="dialog">
    <div class="modal-dialog">
        @using (Html.BeginForm("CargarImagen", "Perfil", FormMethod.Post, new { @enctype =
"multipart/form-data" }))
            {
            <fieldset>
                @Html.HiddenFor(Model => Model.Id)
                <div class="modal-content">
                    <div class="modal-header">
                        <button type="button" class="close" data-dismiss="modal">&times;</button>
                        <h3 class="modal-title text-center">Modificar Imagen</h3>
                    </div>
                    <div class="modal-body">
                        <form class="form-horizontal">
                            <div class="container-fluid">
                                <label for="photo" class="col-xs-2">Imagen:</label>
                                <div class="input-group  col-xs-10">
                                    @*<label class="input-group-btn">
                                            Abrir...
                                            <input type="file" multiple="" id="photo" name="photo">
                                        </label>*@
                                    <input type="text" class="form-control input-sm filefield"
placeholder="Url..." readonly="">
                                    <span class="input-group-btn">
                                        <label class="btn btn-warning btn-sm btn-file">
                                            <span class="glyphicon glyphicon-camera"></span> Abrir
                                            <input type="file" style="display: none;" id="photo"
name="photo">
                                        </label>
                                    </span>
                                </div>
                                <br />
                                <br />
                            </div>
                        </form>
                    </div>
                    <div class="modal-footer">
                        <input type="submit" value="Modificar" class="btn btn-success" />
                        <button type="reset" class="btn btn-danger" data-
dismiss="modal">Cancelar</button>
                    </div>
                </div>
            </fieldset>
        }
    </div>
</div>


<div id="modal-eliminar" class="modal fade" role="dialog">
    <div class="modal-dialog">

        <!-- Modal Registrar-->
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title letra-prompt">¿Seguro que desea eliminar su cuenta?</h4>
            </div>
            <div class="modal-footer">
                <a href="@Url.Action("Eliminar", "Cuenta", new { Email = Cuenta_Activa.Email})"
type="button" class="btn btn-primary">Eliminar</a>
                <button type="button" class="btn btn-default" data-dismiss="modal">Cancelar</button>
            </div>
        </div>
    </div>
</div>
```

## **Apéndice C.12**: modificar ubicación en mapa (JavaScript)

```
var loadMap = function (id) {

    var lati = document.getElementById('Latitud').value;
    var long = document.getElementById('Longitud').value;
    console.log(parseFloat(lati) + " " + parseFloat(long));
    var HELSINKI = [parseFloat(lati), parseFloat(long)];
    var map = L.map(id);
    var marker = L.marker([parseFloat(lati), parseFloat(long)]).bindPopup('Ubicación actual');
    var circle = L.circle([parseFloat(lati), parseFloat(long)], {
        weight: 1,
        color: 'blue',
        fillColor: '#cacaca',
        fillOpacity: 0.2
    });
    map.addLayer(marker);
    map.addLayer(circle);
    var tile_url = 'http://{s}.tile.osm.org/{z}/{x}/{y}.png';
    var layer = L.tileLayer(tile_url, {
        attribution: 'OSM'
    });
    map.addLayer(layer);
    map.setView(HELSINKI, 19);

    map.on('click', function (e) {
        var popLocation = e.latlng;
        var long = document.getElementById('Longitud');
        long.value = popLocation.lng;

        var lati = document.getElementById('Latitud');
        lati.value = popLocation.lat;
        if (marker && circle) {
            marker.setLatLng([popLocation.lat, popLocation.lng]);
            circle.setLatLng([popLocation.lat, popLocation.lng]);
        } else {
            marker = L.marker([popLocation.lat, popLocation.lng]).addTo(map);
            circle = L.circle([popLocation.lat, popLocation.lng], e.accuracy / 2, {
                weight: 1,
                color: 'blue',
                fillColor: '#cacaca',
                fillOpacity: 0.2
            }).addTo(map);
        }
        var popup = L.popup()
        .setLatLng(popLocation)
        .setContent('<p>Ubicación nueva</p>')
        .openOn(map);
    });
};

loadMap('map');
```

## **Apéndice C.13**: Módificar un Perfil

```
@model Proyecto_Integracion.Models.Perfil
@{
    Layout = "~/Views/_Layout.cshtml";
    ViewBag.Title = "Módificar Reporte";
    var Perfil_Activo = Proyecto_Integracion.WebApp.Utils.SessionManager.PerfilActivo();
    var Cuenta_Activa = Proyecto_Integracion.WebApp.Utils.SessionManager.CuentaActiva();
}
@section Styles{
    <link href="~/Contents/css/Maps.css" rel="stylesheet" />
    <link href="~/Contents/css/Crear_Cuenta.css" rel="stylesheet" />
}
```

```
@section Scripts{
    <script src="~/Contents/js/maps-modificar.js"></script>
}
<div class="container">
    <br />
    <h2 class="letra-prompt"><strong>Módificar Perfil</strong></h2>
    <hr />
    <div class="col-md-offset-2 col-md-8">
        @using (Html.BeginForm("Modificar", "Perfil", FormMethod.Post))
        {
            <div class="panel panel-default">
                <div class="panel-heading">
                    <h4 class="panel-title">Modificar Perfil</h4>
                </div>
                <div class="panel-body">
                    <!--<form class="form-horizontal">-->
                    <div class="form-group">
                        <label for="Nombre" class="letra-prompt">Nombre</label>
                        <input type="text" name="Nombre" value="@Model.Nombre" maxlength="15"
class="form-control input-lg">
                    </div>
                    <div class="form-group">
                        <!--<label for="Latitud" class="letra-prompt">Latitude</label>-->
                        <input type="text" name="Latitud" id="Latitud"
value="@Model.Ubicacion.Latitud" hidden>
                    </div>
                    <div class="form-group">
                        <!--<label for="Longitud" class="letra-prompt">Longitud</label>-->
                        <input type="text" name="Longitud" id="Longitud"
value="@Model.Ubicacion.Longitud" hidden>
                    </div>

                    <!--</form>-->
                    <div class="form-group">
                        <div class="panel panel-default mapa-mi-ubicacion">
                            <div class="panel-heading">
                                <h3 class="panel-title letra-prompt">Mi Ubicación</h3>
                            </div>
                            <div class="panel-body" id="map">
                            </div>
                        </div>
                    </div>
                    <br />
                </div>
                <div class="panel-footer text-right">
                    <input class="btn btn-primary" value="Guardar" type="submit" />
                    <a href="@Url.Action("Detalles", "Perfil", new {Id = Model.Id})" type="button"
class="btn btn-danger" >Cancelar</a>
                </div>
            </div>
        }

    </div></div>
```

# Apéndice D. *Gestión de reportes*

## **Apéndice D.1**: ReporteController

```
    public class ReporteController : Controller
    {
        // GET: Reporte
        public ActionResult Detalles(long Id)
        {
            Reporte r = new Reporte();
            r.Seleccionar(Id);
            return View(r);
        }

        public ActionResult Crear()
        {
            if (Utils.SessionManager.PerfilActivo() != null && Utils.SessionManager.CuentaActiva()
!= null)
            {
                return View();
            }
            return RedirectToAction("Index", "Home");
        }

        [HttpPost]
        public ActionResult Crear(Reporte r, Ubicacion u, FormCollection collection)
        {

            String date = Request.Form["fecha"];
            var perfil_Activo = Utils.SessionManager.PerfilActivo();
            if (perfil_Activo != null && Utils.SessionManager.CuentaActiva() != null && r.Incidente
!= 0)
            {
                u.Direccion = Utils.GeoLocation.direccion(u);
                if (u.Crear())
                {
                    r.Perfil = perfil_Activo;
                    DateTime dt;
                    DateTime.TryParse(date, out dt);
                                    //DateTime fecha;
                                    //DateTime.TryParse(date, out fecha);
                                    r.FechaExpedicion = dt;
                    r.Ubicacion = u;
                    if (r.Crear())
                    {
                        Utils.UIWarnings.SetError("Reporte creado exitosamente");
                        return RedirectToAction("Detalles", "Perfil", new { Id = perfil_Activo.Id
});
                    }
                    else
                    {
                        Utils.UIWarnings.SetError("El reporte no pudo ser creado");
                        return RedirectToAction("Index", "Home");
                    }

                }

            }
            return RedirectToAction("Index", "Home");
        }

        //GET: modifcicar/Id
        public ActionResult Modificar(long Id)
        {
            Reporte r = new Reporte();
            r.Seleccionar(Id);
            if (Utils.SessionManager.PerfilActivo() != null && Utils.SessionManager.CuentaActiva()
!= null)
            {
                if (r.Perfil.Id == Utils.SessionManager.PerfilActivo().Id)
                {
                    return View(r);
                }
```

```
            }
            else
            {
                Utils.UIWarnings.SetError("No tiene permisos para modificar este reporte");
                return RedirectToAction("Index", "Home");
            }

            return RedirectToAction("Index", "Home");

        }

        [HttpPost]
        public ActionResult Modificar(Reporte r, Ubicacion u, FormCollection collection)
        {
            Ubicacion ubicacion = new Ubicacion();
            ubicacion.Seleccionar(Convert.ToInt64(Request.Form["Id_Ubicacion"]));
            ubicacion.Latitud = u.Latitud;
            ubicacion.Longitud = u.Longitud;
            ubicacion.Direccion = Utils.GeoLocation.direccion(ubicacion);
            if (ubicacion.Modificar() && r.Incidente != 0)
            {
                r.Ubicacion = ubicacion;
                String date = Request.Form["fecha"];
                string pattern = "yyyy-dd-mm";
                DateTime dt;
                DateTime.TryParse(date, out dt);
                //DateTime fecha;
                //DateTime.TryParse(date, out fecha);
                r.FechaExpedicion = dt;
                if (r.Modificar())
                {
                    Utils.UIWarnings.SetInfo("Modificación de reporte exitosa");
                    return RedirectToAction("Detalles", "Reporte", new { Id = r.Id });
                }
            }
            Utils.UIWarnings.SetError("No se pudo realizar las modificaciones del reporte");
            return RedirectToAction("Index","Home");
        }

        public ActionResult Eliminar(long Id)
        {
            var Perfil_activo = Utils.SessionManager.PerfilActivo();
            Reporte r = new Reporte();
            r.Seleccionar(Id);
            if(Perfil_activo != null && Utils.SessionManager.CuentaActiva() !=  null &&
Perfil_activo.Id == r.Perfil.Id)
            {
                r.Eliminar();
                Utils.UIWarnings.SetInfo("Eliminacion exitosa del reporte");
                return RedirectToAction("Detalles", "Perfil", new { Id = Perfil_activo.Id });
            }
            else
            {
                Utils.UIWarnings.SetError("No tiene los permisos necesarios");
                return RedirectToAction("Index", "Home");
            }
        }
    }
```

## Apéndice D.2: Manejo de Enums

```
 public static class EnumHelper<T>
    {
        public static IList<T> GetValues(Enum value)
        {
            var enumValues = new List<T>();
```

```
            foreach (FieldInfo fi in value.GetType().GetFields(BindingFlags.Static |
BindingFlags.Public))
            {
                enumValues.Add((T)Enum.Parse(value.GetType(), fi.Name, false));
            }
            return enumValues;
        }

        public static T Parse(string value)
        {
            return (T)Enum.Parse(typeof(T), value, true);
        }

        public static IList<string> GetNames(Enum value)
        {
            return value.GetType().GetFields(BindingFlags.Static | BindingFlags.Public).Select(fi =>
fi.Name).ToList();
        }

        public static IList<string> GetDisplayValues(Enum value)
        {
            return GetNames(value).Select(obj => GetDisplayValue(Parse(obj))).ToList();
        }

        private static string lookupResource(Type resourceManagerProvider, string resourceKey)
        {
            foreach (PropertyInfo staticProperty in
resourceManagerProvider.GetProperties(BindingFlags.Static | BindingFlags.NonPublic |
BindingFlags.Public))
            {
                if (staticProperty.PropertyType == typeof(System.Resources.ResourceManager))
                {
                    System.Resources.ResourceManager resourceManager =
(System.Resources.ResourceManager)staticProperty.GetValue(null, null);
                    return resourceManager.GetString(resourceKey);
                }
            }

            return resourceKey; // Fallback with the key name
        }

        public static string GetDisplayValue(T value)
        {
            var fieldInfo = value.GetType().GetField(value.ToString());

            var descriptionAttributes = fieldInfo.GetCustomAttributes(
                typeof(DisplayAttribute), false) as DisplayAttribute[];

            if (descriptionAttributes[0].ResourceType != null)
                return lookupResource(descriptionAttributes[0].ResourceType,
descriptionAttributes[0].Name);

            if (descriptionAttributes == null) return string.Empty;
            return (descriptionAttributes.Length > 0) ? descriptionAttributes[0].Name :
value.ToString();
        }

        public static SelectList GetDirectionSelectList()
        {
            Array values = Enum.GetValues(typeof(Proyecto_Integracion.Models.TipoIncidente));
            List<ListItem> items = new List<ListItem>(values.Length);

            foreach (var i in values)
            {
                items.Add(new ListItem
                {
                    Text =
Proyecto_Integracion.WebApp.Utils.EnumHelper<Proyecto_Integracion.Models.TipoIncidente>.GetDisplayVa
lue((Proyecto_Integracion.Models.TipoIncidente)i),
```

```
                    Value = Enum.GetName(typeof(Proyecto_Integracion.Models.TipoIncidente), i)

                });
            }

            return new SelectList(items);
        }

        public static SelectList Seleccionar_SelectList(Proyecto_Integracion.Models.TipoIncidente
incidente)
        {
            Array values = Enum.GetValues(typeof(Proyecto_Integracion.Models.TipoIncidente));
            List<ListItem> items = new List<ListItem>(values.Length);

            foreach (var i in values)
            {
                if (Enum.GetName(typeof(Proyecto_Integracion.Models.TipoIncidente), i) ==
Proyecto_Integracion.WebApp.Utils.EnumHelper<Proyecto_Integracion.Models.TipoIncidente>.GetDisplayVa
lue(incidente))
                {
                    items.Add(new ListItem
                    {
                        Text =
Proyecto_Integracion.WebApp.Utils.EnumHelper<Proyecto_Integracion.Models.TipoIncidente>.GetDisplayVa
lue((Proyecto_Integracion.Models.TipoIncidente)i),
                        Value = Enum.GetName(typeof(Proyecto_Integracion.Models.TipoIncidente), i),
                        Selected = true
                    });
                }
                else
                {
                    items.Add(new ListItem
                    {
                        Text =
Proyecto_Integracion.WebApp.Utils.EnumHelper<Proyecto_Integracion.Models.TipoIncidente>.GetDisplayVa
lue((Proyecto_Integracion.Models.TipoIncidente)i),
                        Value = Enum.GetName(typeof(Proyecto_Integracion.Models.TipoIncidente), i)
                    });
                }
            }

            return new SelectList(items);
        }

       public static TipoIncidente Change(String value)
        {
            String inci = null;
            switch (value)
            {
                case "Homicidio":
                    inci = "Homicidio";
                    break;
                case "Suicidio":
                    inci = "Suicidio";
                    break;
                case "Robo o Asalto":
                    inci = "RoboAsalto";
                    break;
                case "Violación":
                    inci = "Violacion";
                    break;
                case "Explotación Sexual":
                    inci = "ExplotacionSexual";
                    break;
            }
            var salida = new TipoIncidente();
            Enum.TryParse(inci, out salida);
            return salida;
        }
```

```
        }
```

## **Apéndice D.3**: Resultados de búsqueda – pintar en mapa (JavaScript)

```
var loadMap = function (id) {

    var HELSINKI = [60.1708, 24.9375];
    var map = L.map(id);
    var marker;
    var circle;
    var tile_url = 'http://{s}.tile.osm.org/{z}/{x}/{y}.png';
    var layer = L.tileLayer(tile_url, {
        attribution: 'OSM'
    });
    //var drawnItems = new L.FeatureGroup();
    //map.addLayer(drawnItems);
    //var drawControl = new L.Control.Draw({
    //    edit: {
    //        featureGroup: drawnItems
    //    }
    //});
    //map.addControl(drawControl);
    map.addLayer(layer);
    map.setView(HELSINKI, 50);
    console.log(id);
    map.locate({ setView: true, watch: false }) /* This will return map so you can do chaining */
        .on('locationfound', function (e) {
            marker = L.marker([e.latitude, e.longitude]).bindPopup('Mi ubicacion actual');
            console.log(e.latitude);
            console.log(e.longitude);
            circle = L.circle([e.latitude, e.longitude], e.accuracy / 2, {
                weight: 1,
                color: 'blue',
                fillColor: '#cacaca',
                fillOpacity: 0.2
            });
            map.addLayer(marker);
            map.addLayer(circle);
        })
       .on('locationerror', function (e) {
           console.log(e);
           alert("Location access denied.");
       });


    var cont = document.getElementById("report_count").value;
    console.log(cont);
    for (i = 0; i < cont; i++) {
        var latitude = document.getElementById('Latitud+' + i).value;
        var longitude = document.getElementById("Longitud+" + i).value;
        var incidente = document.getElementById("incidente+" + i).value;
        switch (incidente) {
            case 'Homicidio':
                var marker = L.marker([latitude, longitude], { icon: L.AwesomeMarkers.icon({icon:
'exclamation-triangle', prefix: 'fa', markerColor: 'red', iconColor: 'white' }) }).addTo(map);
                marker.bindTooltip("Homicidio", { permanent: true, className: "homicidio", offset:
[0, 0] });
                marker.addTo(map);
                break;
            case "Suicidio":
                var marker = L.marker([latitude, longitude], { icon: L.AwesomeMarkers.icon({ icon:
'exclamation-triangle', prefix: 'fa', markerColor: 'darkred', iconColor: 'white' }) });
                //L.Marker.Text([latitude, longitude], 'Suicidio').addTo(map);
                marker.bindTooltip("Suicidio", { permanent: true, className: "suicidio", offset: [0,
0] });
```

```
                marker.addTo(map);
                break;
            case "RoboAsalto":
                var marker = L.marker([latitude, longitude], { icon: L.AwesomeMarkers.icon({ icon:
'exclamation-triangle', prefix: 'fa', markerColor: 'lightred', iconColor: 'white' }) });
                //L.Marker.Text([latitude, longitude], 'Robo o Asalto').addTo(map);
                marker.bindTooltip("Robo o Asalto", { permanent: true, className: "roboasalto",
offset: [0, 0] });
                marker.addTo(map);
                break;
            case "Violacion":
                var marker = L.marker([latitude, longitude], { icon: L.AwesomeMarkers.icon({ icon:
'exclamation-triangle', prefix: 'fa', markerColor: 'orange', iconColor: 'white' }) });
                //L.Marker.Text([latitude, longitude], 'Violación').addTo(map);
                marker.bindTooltip("Violación", { permanent: true, className: "violacion", offset:
[0, 0] });
                marker.addTo(map);
                break;
            case "ExplotacionSexual":
                var marker = L.marker([latitude, longitude], { icon: L.AwesomeMarkers.icon({ icon:
'exclamation-triangle', prefix: 'fa', markerColor: 'black', iconColor: 'white' }) });
                //L.Marker.Text([latitude, longitude], 'Violación').addTo(map);
                marker.bindTooltip("Explotación Sexual", { permanent: true, className:
"explotacionsexual", offset: [0, 0] });
                marker.addTo(map);
        }

        console.log(latitude + " " + longitude + " " + incidente);
    }
};

loadMap('map');
```

## **Apéndice D.4**: Crear Reporte

```
@model Proyecto_Integracion.Models.Reporte
@{
    Layout = "~/Views/_Layout.cshtml";
    ViewBag.Title = "Crear Reporte";
}

@section Styles{
    <link href="~/Contents/css/Maps.css" rel="stylesheet" />
    <link href="~/Contents/css/Crear_Cuenta.css" rel="stylesheet" />

}

@section Scripts{
    <script src="~/Contents/js/maps-registrar.js"></script>
 <script type="text/javascript">
    $(document).ready(function () {
        $("#fecha").datepicker({
            dateFormat: 'yy-mm-dd'

        });
        console.log(document.getElementById("fecha").value);
    });
</script>
}
```

```
<div class="container">
    <br />
    <h2 class="letra-prompt"><strong>Crear Reporte</strong></h2>
    <hr />
    <div class="panel panel-default">
        <div class="panel-heading">
            <h4 class="panel-title">Nuevo Reporte</h4>
        </div>
        <!--<form class="form-horizontal">-->
        @using (Html.BeginForm("Crear", "Reporte", FormMethod.Post))
            {
            <fieldset>
                @Html.HiddenFor(Model => Model.Id)
                @Html.HiddenFor(Model => Model.FechaExpedicion)
                @Html.HiddenFor(Model => Model.Perfil.Id)
                @Html.HiddenFor(Model => Model.Perfil.Nombre)
                @Html.HiddenFor(Model => Model.Perfil.UrlImagen)
                @Html.HiddenFor(Model => Model.Perfil.Ubicacion)
                @Html.HiddenFor(Model => Model.Perfil.Cuenta)
                <div class="panel-body">
                    <div class="form-group">
                            <label for="FechaExpedicion" class="letra-prompt">Fecha</label>
                            <input type="text" class="form-control" id="fecha" name="fecha" />
                    </div>

                    <div class="form-group">
                        <label for="Descripcion" class="letra-prompt">Descripción</label>
                        @Html.TextAreaFor(m => m.Descripcion, new { @class = "form-control
textarea", @id = "Resumen", maxlength = "255" })
                    </div>
                    <div class="form-group">
                        <label for="Incidente" class="letra-prompt">Tipo de incidente</label>
                        @{

                        @Html.EnumDropDownListFor(x => x.Incidente, "--Selecciona--", new { @class =
"form-control" })
                        }
                    </div>
                    <div class="form-group">
                        <input type="text" name="Latitud" id="Latitud" hidden>
                    </div>
                    <div class="form-group">
                        <input type="text" name="Longitud" id="Longitud" hidden>
                    </div>

                    <!--</form>-->
                    <div class="form-group">
                        <div class="panel panel-default mapa-mi-ubicacion">
                            <div class="panel-heading">
                                <h3 class="panel-title letra-prompt">
                                    Ubicación del incidente
                                </h3>
                            </div>
                            <div class="panel-body" id="map">
                            </div>
                        </div>
                    </div>
                    <br />
                </div>

                <div class="panel-footer text-right">
                    <input class="btn btn-primary" value="Crear" type="submit" />
                    <button type="button" class="btn btn-danger"
onclick="irAtras()">Cancelar</button>
                </div>
            </fieldset>
        }
```

```
    </div>
</div>
```

## **Apéndice D.5**: Detalles de un Reporte

```
@model Proyecto_Integracion.Models.Reporte
@{
    Layout = "~/Views/_Layout.cshtml";
    ViewBag.Title = "Detalles Reporte";
    var perfil_activo = Proyecto_Integracion.WebApp.Utils.SessionManager.PerfilActivo();
    var cuenta_activo = Proyecto_Integracion.WebApp.Utils.SessionManager.CuentaActiva();
}

@section Styles{
    <link href="~/Contents/css/Maps.css" rel="stylesheet" />
    <link href="~/Contents/css/detalles-reporte.css" rel="stylesheet" />
}

@section Scripts{
    <script src="~/Contents/js/map-visualizar.js"></script>
}

<div class="container">
    <br />
    <h2 class="letra-prompt"><strong>Detalles de Reporte</strong></h2>
    <hr />
    <div class="col-md-2 text-center">
        @if (perfil_activo != null && cuenta_activo != null && perfil_activo.Id == Model.Perfil.Id)
        {
            <a class="btn btn-primary" href="@Url.Action("MisReportes", "Perfil", new { Id =
Model.Perfil.Id})">
                <i class="fa fa-chevron-left" aria-hidden="true"></i>
                 Regresar
            </a>
        }
        else
        {
        <a class="btn btn-success" onclick="irAtras()">
            <i class="fa fa-chevron-left" aria-hidden="true"></i>
             Regresar
        </a>
        }
    </div>
    <div class="col-md-8">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h4 class="panel-title">Reporte</h4>
            </div>
            <!--<form class="form-horizontal">-->
            <div class="panel-body">
                <div class="form-group">
                    <p><strong>Fecha: </strong>@Model.FechaExpedicion.ToString("d")</p>
                </div>
                <div class="form-group">
                    <p><strong>Dirección: </strong>@Model.Ubicacion.Direccion</p>
                </div>
                <div class="form-group">
                    <label for="Descripcion" class="letra-prompt">Descripción</label>
                    <p class="text-justify">@Model.Descripcion</p>
                </div>
                <div class="form-group">
                    @{
                        var incidente =
Proyecto_Integracion.WebApp.Utils.EnumHelper<Proyecto_Integracion.Models.TipoIncidente>.GetDisplayVa
lue(Model.Incidente);
                        <p><strong>Tipo de Delito: </strong>@incidente</p>
                    }
```

```
                </div>
                <div class="form-group">
                    <input type="text" name="Latitud" id="Latitud" value="@Model.Ubicacion.Latitud"
hidden>
                </div>
                <div class="form-group">
                    <input type="text" name="Longitud" id="Longitud"
value="@Model.Ubicacion.Longitud" hidden>
                </div>

                <!--</form>-->
                <div class="form-group">
                    <div class="panel panel-default mapa-ubicacion_reporte">
                        <div class="panel-heading">
                            <h3 class="panel-title letra-prompt">
                                Ubicación del incidente
                            </h3>
                        </div>
                        <div class="panel-body" id="map">
                        </div>
                    </div>
                </div>
                <br />
            </div>
        </div>
    </div>
    <div class="col-md-2">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h4 class="panel-title">Perfil</h4>
            </div>
            <div class="panel-body">
                <p><strong>Nombre:</strong><a href="@Url.Action("Detalles", "Perfil", new { Id =
Model.Perfil.Id})"> @Model.Perfil.Nombre</a></p>
            </div>
        </div>
            @if (perfil_activo != null && cuenta_activo != null && perfil_activo.Id ==
Model.Perfil.Id)
            {
                <div class="text-center">
                    <br />
                    <br />
                    <a class="btn btn-warning" href="@Url.Action("Modificar", "Reporte", new { Id =
Model.Id })">
                        Módificar
                        <i class="glyphicon glyphicon-pencil"></i>
                    </a>
                    <br />
                    <br />
                    <br />
                    <a class="btn btn-danger" data-toggle="modal" data-target="#modal-eliminar">
                        Eliminar  
                        <i class="glyphicon glyphicon-remove"></i>
                    </a>
                </div>
            }
    </div>
</div>


<div id="modal-eliminar" class="modal fade" role="dialog">
    <div class="modal-dialog">

        <!-- Modal Registrar-->
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title letra-prompt">¿Seguro que desea eliminar este repote?</h4>
            </div>
```

```
            <div class="modal-footer">
                <a href="@Url.Action("Eliminar", "Reporte", new { Id = Model.Id })" type="button"
class="btn btn-primary">Eliminar</a>
                <button type="button" class="btn btn-default" data-dismiss="modal">Cancelar</button>
            </div>
        </div>
    </div>
</div>
```

## **Apéndice D.6**: Modificar un Reporte

```
@model Proyecto_Integracion.Models.Reporte
@{
    Layout = "~/Views/_Layout.cshtml";
    ViewBag.Title = "Módificar Reporte";
    var Perfil_Activo = Proyecto_Integracion.WebApp.Utils.SessionManager.PerfilActivo();
}

@section Styles{
    <link href="~/Contents/css/Maps.css" rel="stylesheet" />
    <link href="~/Contents/css/detalles-reporte.css" rel="stylesheet" />

}

@section Scripts{
    <script src="~/Contents/js/maps-modificar.js"></script>
    <script type="text/javascript">
    $(document).ready(function () {
       $("#fecha").datepicker({
            dateFormat: 'yy-mm-dd'

       });
    });
</script>
}

<div class="container">
    <br />
    <h2 class="letra-prompt"><strong>Módificar Reporte</strong></h2>
    <hr />
    <div class="col-md-offset-2 col-md-8">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h4 class="panel-title">Modificar Reporte</h4>
            </div>
            <!--<form class="form-horizontal">-->
            @using (Html.BeginForm("Modificar", "Reporte", FormMethod.Post))
            {
                <fieldset>
                    @Html.HiddenFor(Model => Model.Id)
                    @Html.HiddenFor(Model => Model.FechaExpedicion)
                    @Html.HiddenFor(Model => Model.Perfil.Id)
                    @Html.HiddenFor(Model => Model.Perfil.Nombre)
                    @Html.HiddenFor(Model => Model.Perfil.UrlImagen)
                    @Html.HiddenFor(Model => Model.Perfil.Ubicacion)
                    @Html.Hidden("Id_Ubicacion", Model.Ubicacion.Id)
                    @Html.HiddenFor(Model => Model.Perfil.Cuenta)
                    <div class="panel-body">
                        <div class="form-group">
                            <label for="FechaExpedicion" class="letra-prompt">Fecha</label>
                            <input name="fecha" id="fecha"
value="@Model.FechaExpedicion.ToString("d")" class="form-control" />
                        </div>
                        @{
                        }
                        <div class="form-group">
```

```
                            <label for="Descripcion" class="letra-prompt">Descripción</label>
                            @Html.TextAreaFor(m => m.Descripcion, new { @class = "form-control
textarea", @id = "Resumen", maxlength = "255", @value = Model.Descripcion })
                        </div>
                        <div class="form-group">
                            <label for="Incidente" class="letra-prompt">Tipo de incidente</label>
                            @{
                             @Html.EnumDropDownListFor(x => x.Incidente, "--Selecciona--",  new {
@class = "form-control" })
                            }
                        </div>
                        <div class="form-group">
                            <input type="text" name="Latitud" id="Latitud"
value="@Model.Ubicacion.Latitud" hidden>
                        </div>
                        <div class="form-group">
                            <input type="text" name="Longitud" id="Longitud"
value="@Model.Ubicacion.Longitud" hidden>
                        </div>

                        <!--</form>-->
                        <div class="form-group">
                            <div class="panel panel-default mapa-ubicacion_reporte">
                                <div class="panel-heading">
                                    <h3 class="panel-title letra-prompt">
                                        Ubicación del incidente
                                    </h3>
                                </div>
                                <div class="panel-body" id="map">
                                </div>
                            </div>
                        </div>
                        <br />
                    </div>

                    <div class="panel-footer text-right">
                        <input class="btn btn-primary" value="Modificar" type="submit" />
                        <a href="@Url.Action("Detalles", "Perfil", new { Id = Model.Perfil.Id})"
type="button" class="btn btn-danger" >Cancelar</a>
                    </div>
                </fieldset>
            }

        </div>
    </div>

</div>
```

## **Apéndice D.7**: Busqueda General

```
@model IEnumerable<Proyecto_Integracion.Models.Reporte>
@{
    Layout = "~/Views/_Layout.cshtml";
    ViewBag.Title = "Resultados";
    //var Perfil_Activo = Proyecto_Integracion.WebApp.Utils.SessionManager.PerfilActivo();
    //var Cuenta_Activa = Proyecto_Integracion.WebApp.Utils.SessionManager.CuentaActiva();
    var incidente_res = new Proyecto_Integracion.Models.TipoIncidente();
}

@section Styles{
    <link href="~/Contents/css/Mis-Reportes.css" rel="stylesheet" />
    <link href="~/Contents/css/Maps.css" rel="stylesheet" />
    <link href="~/Contents/css/Letra_Iconos.css" rel="stylesheet" />
}
```

```
@section Scripts{
    <script src="~/Contents/js/maps-buscador.js"></script>
}
<div class="container">
    <!--Mis ultimos reportes-->
    <div class="col-md-12">
        <br />
        <h2 class="letra-prompt"><strong>Resultados</strong></h2>
        <hr />
        <div class="panel panel-default ubicaciones_mis_reportes">
            <div class="panel-heading">
                <h4 class="panel-title">Ubicaciones</h4>
            </div>
            <div class="panel-body" id="map">
            </div>
        </div>
    </div>



    <div class="col-md-12">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h4 class="panel-title"> Resultados de Busqueda :
<strong>@ViewBag.Termino</strong></h4>
            </div>
            <div class="panel-body">
                <div class="list-group">
                    <div class="row" style="display: flex; flex-wrap: wrap;">
                        <input id="report_count" value="@Model.Count()" hidden />

                        @if (Model.Count() == 0)
                        {
                            <br />
                                <div class="col-xs-12 col-md-6">
                                    <div class="container">
                                        <p class="text-center" id="no-resultados">No se han
encontrado resultados para "<strong>@ViewBag.Termino</strong>"</p>
                                    </div>
                                </div>

                        }



                        @{var i = 0;
                            if (Model.Count() > 0) {
                                incidente_res = Model.First().Incidente;
                            }
                        }
                        @try
                        {
                            foreach (var reporte in Model)
                            {


                                <div class="col-md-6 reporte" style="display: flex; flex-wrap:
wrap;">
                                    <a href="@Url.Action("Detalles", "Reporte", new { Id =
reporte.Id })" class="list-group-item">

                                        <div class="media">
                                            <div class="media-left media-middle">
                                                <i class="fa fa-map-marker fa-3x texto-naranja"
aria-hidden="true"></i>
                                            </div>
                                            <div class="media-body text-justify">
                                                @{
```

```
                                                var incidente =
Proyecto_Integracion.WebApp.Utils.EnumHelper<Proyecto_Integracion.Models.TipoIncidente>.GetDisplayVa
lue(reporte.Incidente);
                                                <input id="incidente+@i"
value="@reporte.Incidente" hidden />
                                                    <h4 class="media-heading"><strong>Incidente:
</strong>@incidente</h4>
                                            }
                                            <p><strong>Descripción:
</strong>@reporte.Descripcion</p>
                                            <p><strong>Fecha: </strong>
@reporte.FechaExpedicion.ToString("d")</p>
                                            <p><strong>Dirección: </strong>
@reporte.Ubicacion.Direccion</p>
                                            <input id="Latitud+@i" name="Latitud+@i"
value="@reporte.Ubicacion.Latitud" hidden />
                                            <input id="Longitud+@i"
value="@reporte.Ubicacion.Longitud" hidden />
                                        </div>
                                    </div>

                                </a>
                            </div>
                                                    i++;
                                                }
                                            }
                                            catch (Exception ex)
                                            {
                                                <p class="text-center">... No hay resultados
de busqueda ...</p>
                                            }

                    </div>
                </div>

            </div>
            <div class="panel-footer">
                @if (ViewBag.Page > 1)
                {
                    <a href="@Url.Action("BusquedaGeneral", new { page = ViewBag.Page - 1 , buscar=
ViewBag.Termino, cantResult = ViewBag.Results, filtro = ViewBag.Filtro, incidente = incidente_res
})"
                       class="btn btn-default">
                        &laquo; Atras
                    </a>
                }
                else
                {
                    <a href="@Url.Action("BusquedaGeneral", new { page = ViewBag.Page , buscar=
ViewBag.Termino , cantResult = ViewBag.Results, filtro = ViewBag.Filtro, incidente = incidente_res}
)" class="btn btn-default" style="opacity: 0.65; cursor: not-allowed;" disabled>
                        &laquo; Atras
                    </a>
                }
                @if (ViewBag.More > 0)
                {
                    <a href="@Url.Action("BusquedaGeneral", new { page = ViewBag.Page + 1 , buscar=
ViewBag.Termino , cantResult = ViewBag.Results, filtro = ViewBag.Filtro, incidente = incidente_res }
)"
                       class="btn btn-default">
                        Siguiente &raquo;
                    </a>
                }
                else
                {
                    <a href="@Url.Action("BusquedaGeneral", new { page = ViewBag.Page , buscar=
ViewBag.Termino , cantResult = ViewBag.Results, filtro = ViewBag.Filtro, incidente = incidente_res }
)" class="btn btn-default" style="opacity: 0.65; cursor: not-allowed;" disabled>
                        Siguiente &raquo;
```

```
                    </a>
                }
            </div>

        </div>
    </div>
</div>
```