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

Ángeles 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.

Ángeles 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 desarrollar 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 y se pretende que el usuario, el ciudadano, se registre y comience a realizar publicaciones con ayuda de la geolocalización sobre los incidentes delictivos los cuales ha sido víctima o testigo. De igual forma *ReportIt* permite, en caso de no tener cuenta en el sistema, usuario no autentificado, puedas realizar consultas de incidentes delictivos y te puedas mantener informado para tu propia seguridad y la de tus 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 te facilita la creación de tu aplicación.

Existen desventajas a la hora de utilizar este *framework*, esto debido a que ASP .NET no te realiza el mapeo de tu base de datos como bien lo realiza Hibernate[[1]](#footnote-2), pero de igual forma es una herramienta que se adaptó y que facilitó el desarrollo de *ReportIt*.

La IDE[[2]](#footnote-3) 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.

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 detalla 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 estos 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.

1. Gestión de tu perfil:

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

1. Gestión de tus 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, como lo es: modificar su ubicación, la fecha del incidente, la descripción del incidente y/o el tipo de incidente suscitado.

Índice

Capítulo 1: Acercamiento teórico.

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]](#footnote-4), 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 de nombre *ReportIt* para gestionar incidentes delictivos[[4]](#footnote-5) 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.

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.

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.

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]](#footnote-6).

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.

* 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.

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*.

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.

Objetivos

Objetivo general

Desarrollar una aplicación web para gestionar incidentes delictivos en México.

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 esta misma, los lenguajes de programación, librerías usadas, manejos de sesiones con cookies, el uso de la geolocalización en mapas y más.

Conector de MySQL .NET

Es un controlador ADO .NET [[6]](#footnote-7)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 le explicaremos la manera en la que se realizó la conexión con la aplicación web. En la liga [14] encontrarás el conector presentado.

Framework

### ¿Qué es un framework?

Es un patrón o esqueleto para el desarrollo y/o la implementación de una aplicación. 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 div 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.

### Ventajas de usar un 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 te proporciona un esqueleto el cual debes llenar, así, si estas programando en conjunto, los demás programadores podrán entender más rápido lo que hiciste.**
* **Es más fácil encontrar herramientas como librerías adaptadas al framework para facilitar el desarrollo.**

### 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 framework 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.**

**Es una tarea difícil el comenzar por escoger de los tantos frameworks que hay y en los diferentes lenguajes, cuáles son los mejores para nuestra aplicación, pero no te preocupen mucho por esto ya que al final solamente permanecerán los que estén mejor definidos, incluso si no permanece ningún framework puedes comenzar a crear tu propio framework con las necesidades que requiere tu aplicación.**

Visual Studio 2015

### ¿Qué es Microsoft Visual Studio 2015?

Es un conjunto de herramientas para crear software, desde la fase de diseño pasando por la fase de diseño 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]](#footnote-8)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].

### ¿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.

### Servicios de Visual Studio

Visual Studio te 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 diversos 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.

### Visual Studio 2015 como un IDE

Visual Studio como lo hemos estado mencionando con anterioridad, es una herramienta fuerte para el desarrollo de aplicaciones, por todo lo que esta nos puede dar, a continuación, veremos muchos de los servicios que nos proporciona la IDE para un desarrollo de aplicaciones cómodo y rápido, la posibilidad de mantenerte siempre actualizado en las versiones de software que usas, los servicios personalizados con el inicio de sesión, la posibilidad de trabajo en equipo con versiones, etc.

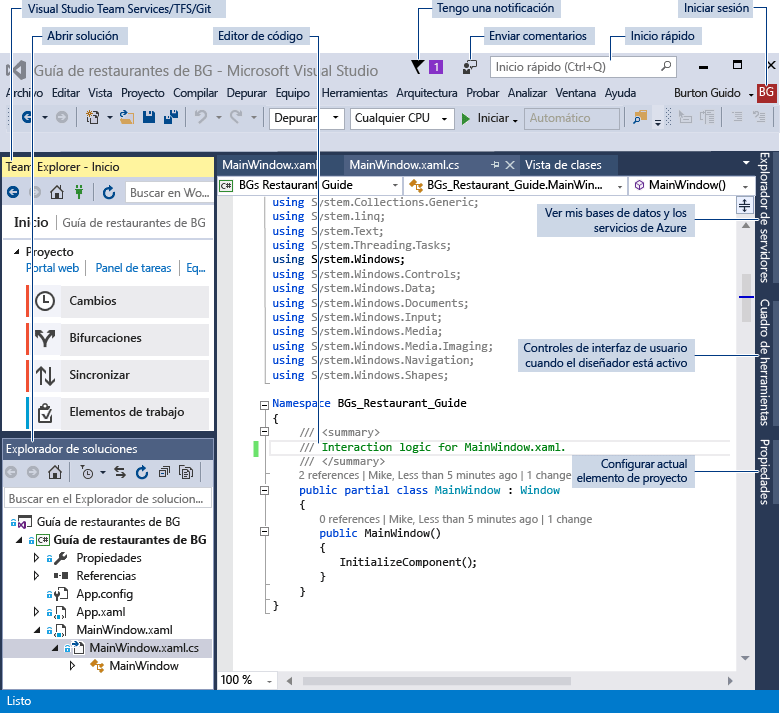


Figura 1. Captura de pantalla de Visual Studio con las secciones resaltadas. Imagen tomada de la referencia [12]

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.

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.

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 te llevará a la página de ayuda de MSDN de la ventana activa. De igual manera si colocas el símbolo de intercalación en algún lugar de tu código y presionas **F1** te proporcionara documentación sobre esa clase, método en el que estas posicionado.

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.

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. 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 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.

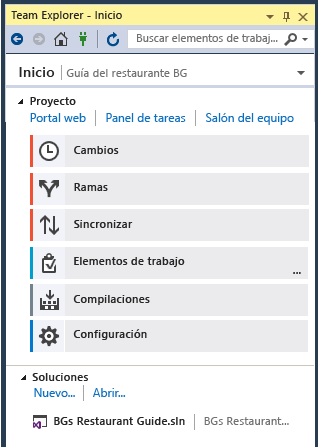


Figura 2. Panel desplegado de Team Services. Imagen obtenida de [12].

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. 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.

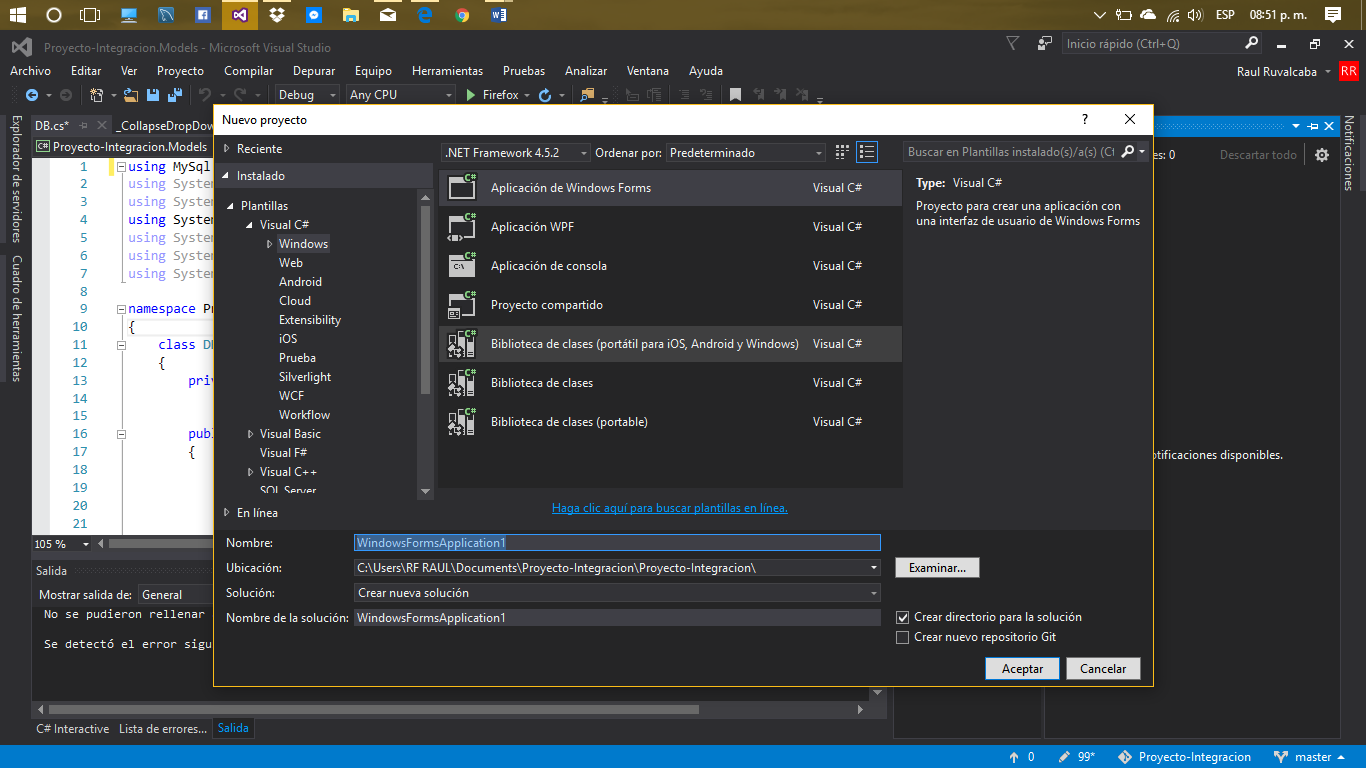


Figura 3. Cuadro desplegado a la hora de crear un proyecto nuevo, la biblioteca de clases es usada para el manejo de clases.

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 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 escritura sobre el código directamente en el editor 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. 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.

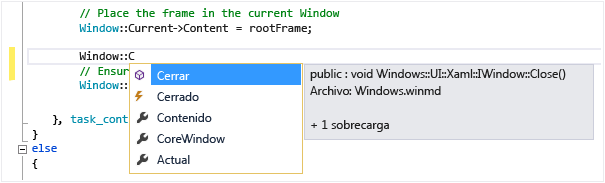


Figura 4 Captura de pantalla que muestra el uso de la configuración IntelliSense. Imagen tomada de [12]

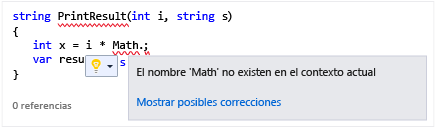


Figura 5 Captura de pantalla que muestra el uso de la configuración, subrayado ondulado, con la función de sugerencia activada.

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.

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”.

### Lenguajes de programación

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

Cree 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].

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].

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, con los fundamentos que son accesibles a la mayoría de la gente en una sentada; 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].

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.

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);

¿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].

Leaflet

Es la biblioteca de JavaScript de código abierto líder para mapas interactivos para móviles. Con un peso aproximado de 38 KB de JS, tiene todas las características de mapeo que la mayoría de los desarrolladores necesitan.

Folleto 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 hermosa, fácil de usar y 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 en diferencia hay muchas más herramientas que facilitan el desarrollo con leaflet por ser código abierto.

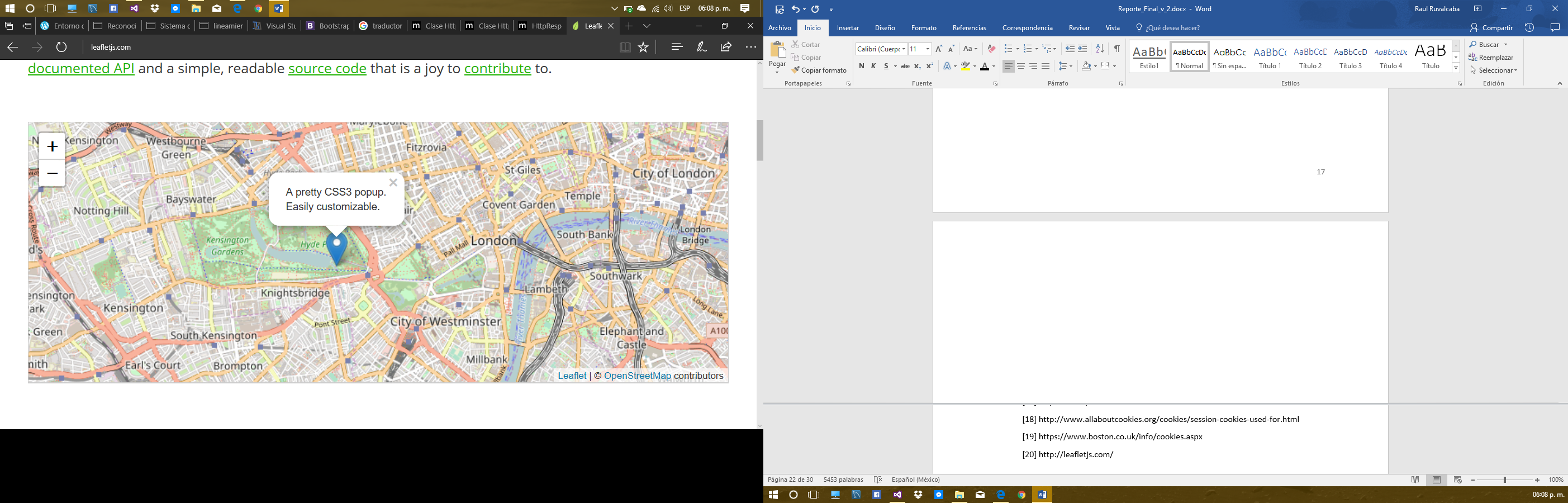


Figura 6. Ejemplo de mapa con marcador en OpenSteetMap usado y manejado por medio de Leaflet.

Como comenzar

Para comenzar a utilizar leaflet dentro de una aplicación web lo primero que debemos realizar es descargar leaflet, este se descargara como un zip el cual debemos descomprimir. Nos aparecerá una carpeta con contenido como se muestra en la ilustración 7, ahora únicamente resta por incorporar a estas librerías a nuestro proyecto.

Para importar el JavaScript que aparece en la carpeta se tiene que agregar en la sección de JavaScript de nuestra aplicación, posteriormente, agregaremos el 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.

Mapa Sencillo

Para incorporar un mapa leaflet 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 <div>, centrado en cierta posición, con un marcador y una etiqueta con texto dentro que de una descripción del marcador.

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 el incorpora el siguiente código:

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

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();

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. Plugins más populares se muestran en la Figura 8.

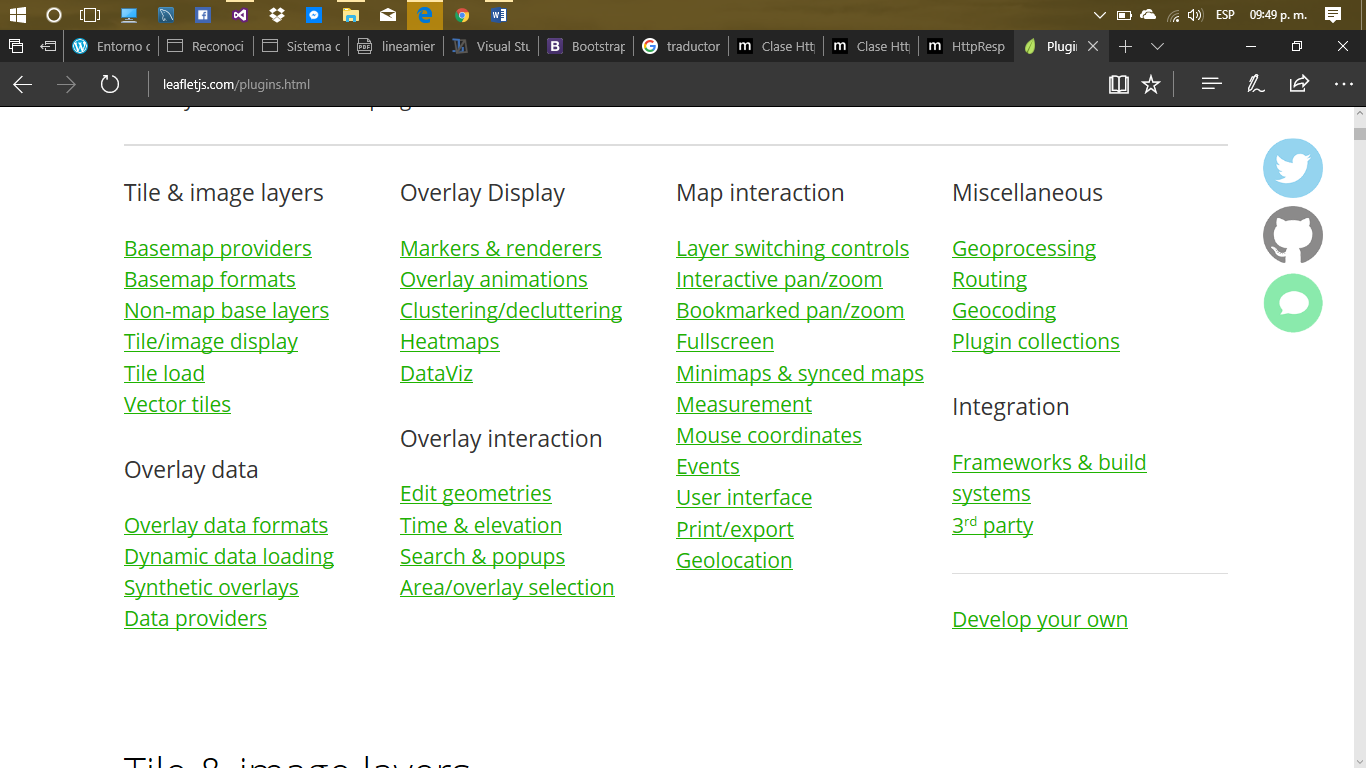


Figura 8 Plugins obtenidos de la página principal de Leaflet.

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. Esto pude ver en la documentación de leaflet [21].

Capítulo 3: Desarrollo del proyecto

El objetivo de este capítulo es dar la metodología ocupara para llegar a desarrollar una apliacion como la nuestra. Este capitulo esta dividido en cuatro modulos cada uno con diferentes submodulos internos. El primer modulo consta del modelo de clases, en este modulo tiremos por qué consideramos que es por lo que se debe comenzar y en que te ayudará en los siguientes modulos. Segundo modulo, la base de datos, en este modulo se pretende que consigas desarrollar toda la parte del servicio de base de datos incluyendo la conexión con la aplicación y los procedimientos almacenados. Tercer modulo, en este modulo nos encargaremos de mostrarte los pasos que se siguieron para realizar toda la gestión de un perfil de un usuario. Cuarto modulo, Gestion de pubicaciones de usuarios, de igual modo que en el modulo tres te explicaremos la metodológia utilizada para conseguir los objetivos.

**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 mas profunda de lo que se puede realizar en la aplicación, te da una imagen de como puedes manejar las vistas, te da estructura en tu proyecto junto con las relaciones entre los mismos objetos y claro, lo mas importante, te da las herramientas suficientes para modelar tu base de datos y saber como se llevara acabo el manejo de los datos. En la Figura 9**,** se encuentra el modelo de clases utilizado para desarrollar esta aplicación.

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 publicacion, etc. En nuestra aplicación manejamos seis objetos los cuales detallaremos mas a continuación.

* **Cuenta**:

Este objeto cuenta con varias propiedades y métodos como se puede ver en el código del Apéndice A.3. Una cuenta consta de dos propiedades, un email y una contraseña y 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 perfil consta de muchas mas propiedades debido a que con este tiene mucha mas interacción dentro de la aplicación. Las propiedades de este objeto son: Id, una UrlImagen, Nombre, una Cuenta, como lo había mensionado 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 ultima de las propiedades es una Ubicación, la Ubicación es importante debido a que con esta se realizara 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. código 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 realizo la publicación), una fecha de expedición, una descripción, una Ubicación (usada para la geolocalización de publicaciones dentro de un radio), un incidente (de este 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 mas importantes debido a que con este se realiza las geolocalizaciones de los incidentes y 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, este ultimo 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.

* **Estanteria:**

Este objeto fue creado para únicamente realizar busquedas dentro de la base de datos, en el Apéndice A.7, se encuentra la codificación de este objeto. Como es de parecer, este objeto no cuenta con propiedades debido a que no las requere, pero si con muchas acciones, que son nada mas que menos todas las busquedas que la aplicación realiza.

* **Base de datos:**

Este objeto tiene como función hacer la conexión de la base de datos y ejecutar querys o realizar consultas 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.

Este 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. El segundo 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 Query, este método es ocupado en todos los casos cuando se inserta en una tabla o cuando se actualizan valores esta misma.

* **Tipo de delito :**

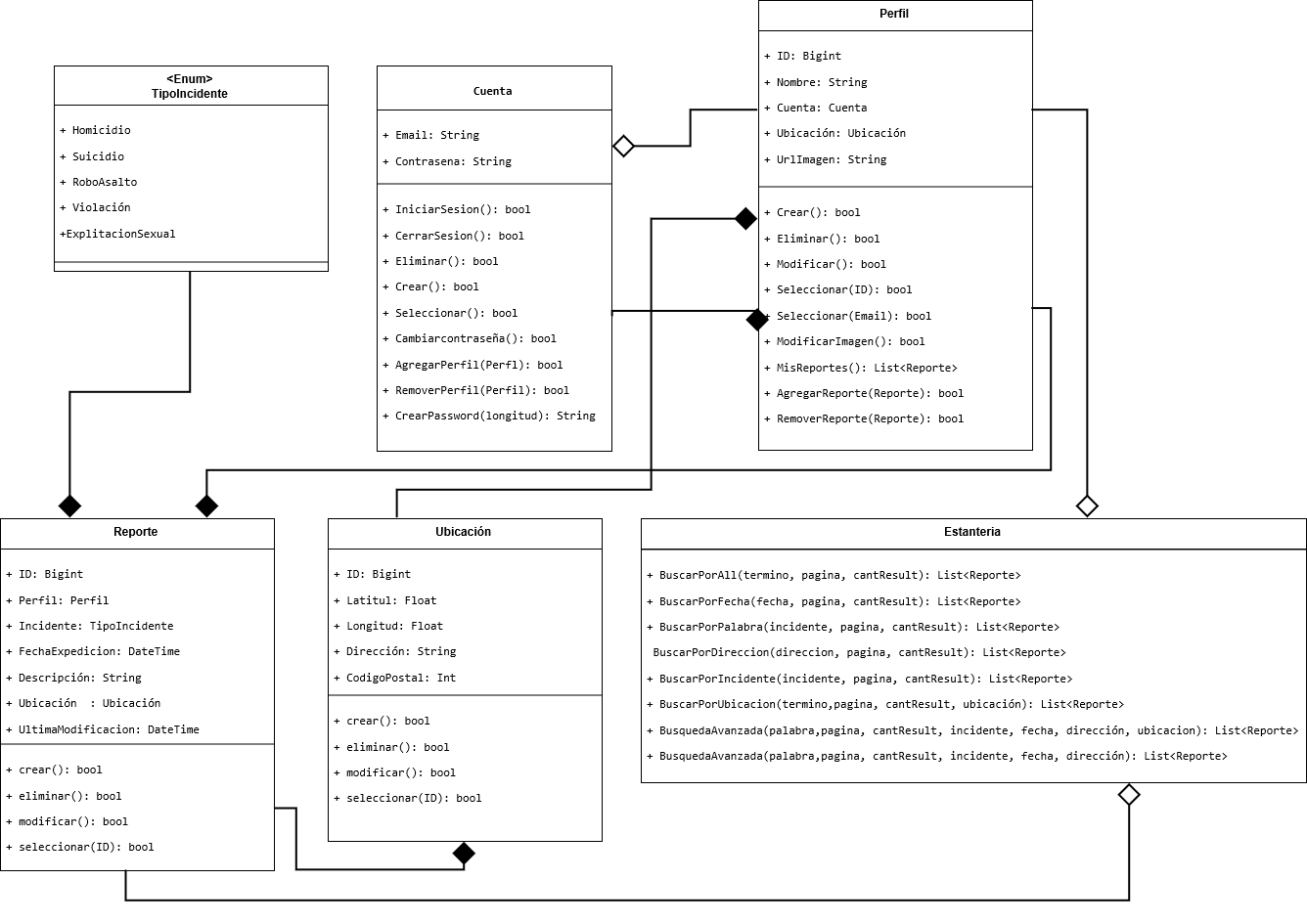
En este caso, los tipos de delitos son manejados desde la capa de aplicación y no desde la base de datos, es por ellos que los tipos de delitos no son una tabla y mucho menos una clase, sino mas bien es un Enum que contiene tipos de incidentes, para esta aplicación se manejaron cinco tipos de incidentes: 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 aun no podemos estar tan seguros de que ese modelo será el final, aun asi, debemos comenzar la tarea de codificar cada una de estas clases dentro de nuestro proyecto, para ellos seleccionamos **Archivo** -> **Nuevo** -> **Poryecto**, nos desplegará una ventana donde debemos irnos a las opciones de C# y buscar la **biblioteca de clases**, no olvides darle un nombre y un directorio antes de guardar.

Por defecto te agregará una clase llamada program.cs, puedes eliminarla, nose 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 asi 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 desplegara una nueva ventana y ahí buscaras 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 modulo 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 modificaciones, asi como realizar procedimientos almacenados que realicen busquedas por localización.

A continuación, se presenta el modelo entidad – relación ocupado para dar estructura a nuestra base de datos.

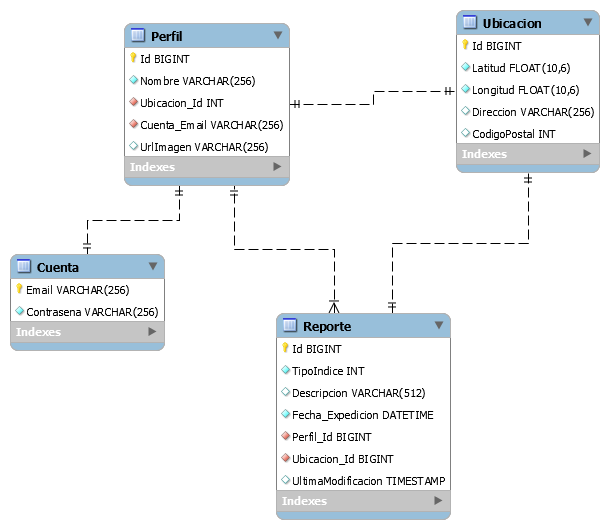


Figura 10 Modelo entidad – relación de la base de datos de nuestra aplicación.

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 basaras 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 ultima modificación de un reporte esto 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 menciono en el modulo 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 limite, esto debido a que en el Apéndice B.2.9 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 primer, elimina todo de la base de datos, incializa la base de datos, y la segunda únicamente ejecuta todos los scripts, es por ellos que en mi proyecto tengo 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 ejmplo: 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.

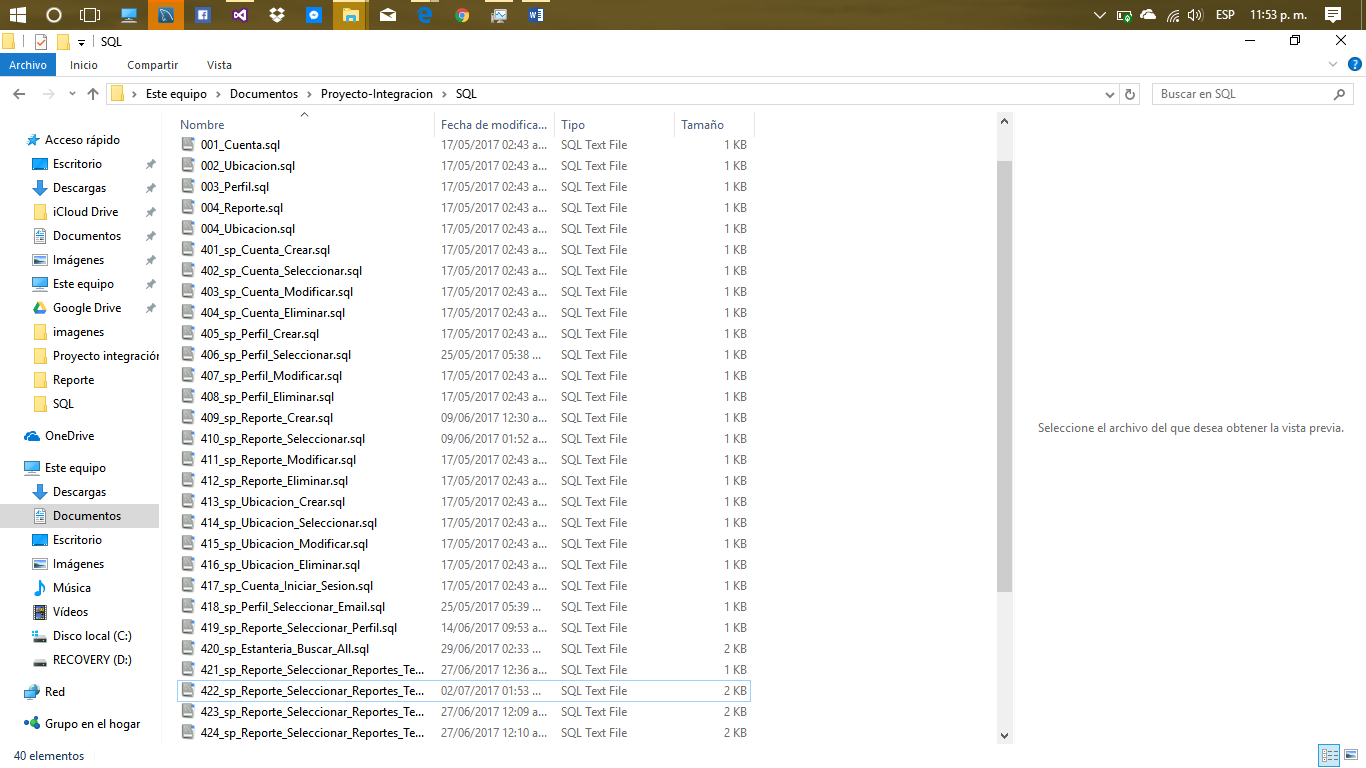


Figura 11 Captura de pantalla que muestra la numeración que se sigue para la ejecución de todos los scripts por medio de apache ant.

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, estos scripts se puede ver Apéndice B.2.1 al B.2.4. Unicamente se muestran los cuatro procedimientos almacenados para el manejo de datos de la tabla cuenta debido al gran espacio que ocuparía agregar los demas. La sisntaxis que se utiliza para realizar estos procedimientos almacenados son de conocimientos básicos para un programador, es por ello que no se entrara mucho en detalle sobre estos.

Para los procedimientos almacenados que realizan busquedas de la clase Estanteria, nos adentraremos un poco mas, debido a que tienen un poco mas de complejidad.

Procedimientos almacenados que realizan las busquedas

* **En todos los atributos:**

Este procedimiento almacenado lo que realiza es una búsqueda de palabras dentro de la descripción, y la dirección del reporte, 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 este, pero en diferencia se tiene que pasar como argumento la latitud y longitud del perfil desde donde se realizará la búsqueda y es entonces que realiza un calculo con cada uno de las ubicacones de un reporte para determinar si este esta o no dentro de un determinado radio, para nuestra aplicación se uso un radio máximo de 40 Km. La formula que realiza el calculo para determinar si esta 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 diferentes, pero mas específicos. 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 buesqueda en uno de ellos podrás repetirlo en las demas búsquedas. 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 asi poder entregar resultados mas genéricos.

De la misma manera como se hace la búsqueda en descripción, se realiza con la direccion. En el caso de la fecha, se recibe un varchar con la fecha, esta 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 únicamente se retorna todos los incidentes que concuerden con el entero que se busca, correspondiente a un tipo de incidente dentro del enum en nuestro modelo de clases.

* **Busqueda avanzada:**

La búsqueda avanzada es una búsqueda mas especifica a las anteriores debido a que se puede realizar búsqueda por diferentes campos e incluso por la ubicación de un perfil. El objetivo de esta búsqueda es que se use para buscar reportes de manera mas especial, puedes realizar la combinación de parámetros de búsqueda que tu quieras. Esta búsqueda se encuentra en el Apéndice B.2.7 y B.2.8 con un parámetro mas, que es la ubicación del tu perfil.

¿Como 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 agregara a resultados. Para los campos que no son llenados en la búsqueda avanzada, realiza la búsqueda por nulo, claro esta que debemos asegurarnos de no tener tuplas con campos nulos, es por ellos que estas validaciones deben hacerse desde la capa de presentación.

* 1. Gestión de un perfil

# Email

Email es una clase que ocupamos como un servicio para nuestra aplicación, si bien, este servicio consta de envíos de correo electrónico a nuestros usuarios únicamente cuendo ellos se registran o desean recuperar su contrasea. 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 valida nada de eso. Esta clase Email podemos verla en el Apéndice C.1.1.

En el caso de recuperación de contraseña, se genera un nueva contrasea 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, y en si consiste en seleccionar caracteres de un alfabeto alfanumérico de manera alleatoria, 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 envio 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 pude llenar sus 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 asi como la contraseña y servicios de seguridad si asi lo queremos.

# Validador

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

Como se explico 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 contra de tres tipos de validaciones, la primera validación es ejemplificada en el primer párrafo, la segunda es para validar si el email que se esta ingresando ya sea para autentificarse en nuestra aplicación o para registrarse no son validos, 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 este cumple con el formato el método retornara un valor booleano de true. Con el tercer método se valida las extenciones 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 extención del archivo a descargar y lo compara con extenciones de imagnes (.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.

# 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 capitulo 2, una cookie puede ser vista como un Dictionary, que en si consiste en una key y un value.

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

* Ingresos: se crea una cookie con un valor del email de usuario que quiere ingresar o se haya registrado.
* Resistrar perfil: agrega el Id de un perfil a una cookie con key Perfil.
* Salir: consiste en modificar los valores de email y Id de un perfil por vacios.
* Cuenta activa, retorna un objeto de tipo Cuenta con el email que esta 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 mas personalizado, mostrándole que conocemos su nombre de usuario y con ello puede accesar a todo lo relacionado a su cuenta.

# Geolocalización

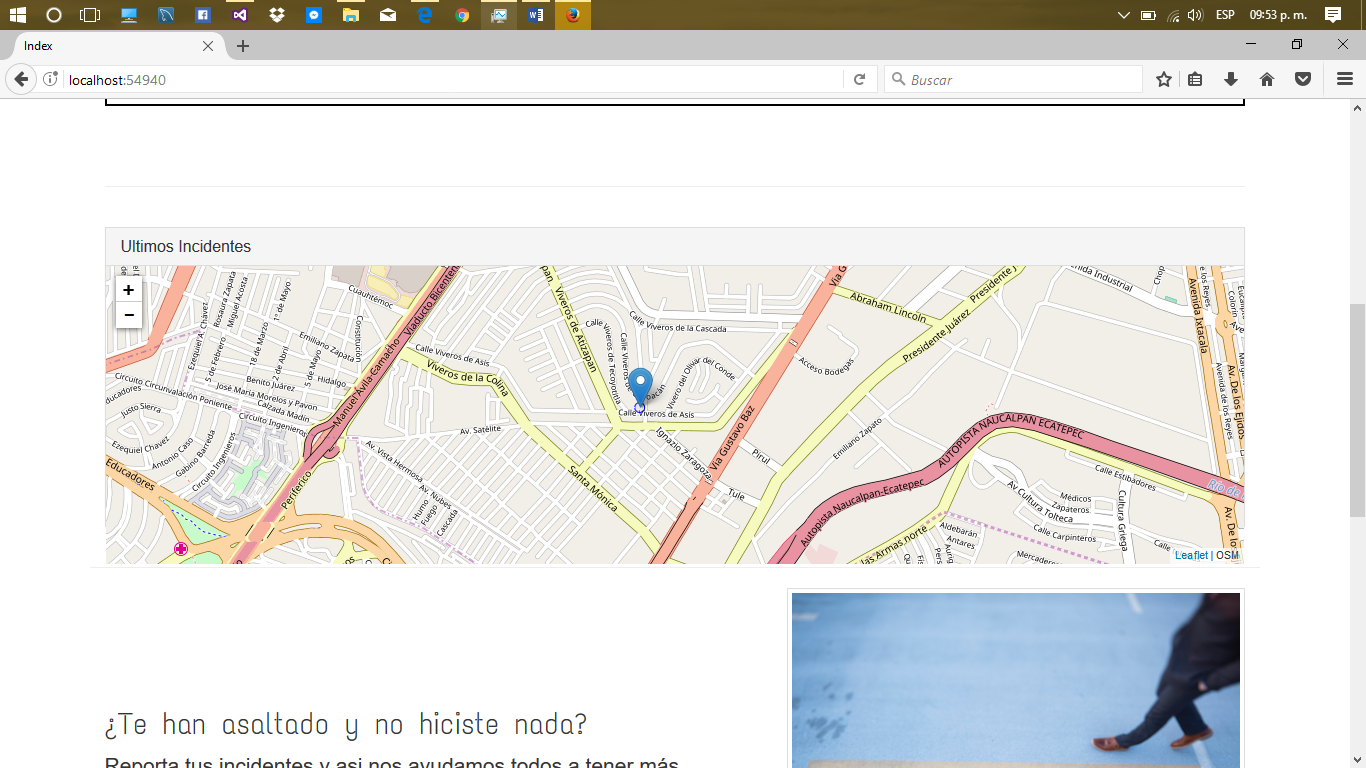


Figura 12 Mapa de últimos reportes de la vista inicio con marcador geolocalizado a la posición de un dispositivo.

La primera vista que se mostrará cuando igreses a nuestra aplicación, será la vista de incio, esta vista da una introducción al servicio y objetivos que tenemos dentro de nuestra aplicación. En esta vista se mostrará el mapa de ultimas modificaciones de la Figura 12, este es un mapa de leaflet y esta geolocalizado pero esta vez es por parte del servicio de leaflet.

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

Todos los métodos implemetados dependen del método etDataRow ya que este le pasas una ubicación con lalitud y longitud y reliza 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, y finalmente se retorna el renglón cero que es el que contiene toda la informacion en un objeto tipo DataRow.

Cuando realizamos la selección de una ubicación ya sea de un perfil o un reporte y damos el clic en el botón de confirmar, se hace una llamada a un método Post dentro de un controlador y un método especifico el cual realiza la búsqueda de la direccion por medio de la clase Geolocation y la latitud y 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 asi como se conocen las direcciones de los marcades en los mapas.

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

# Layout

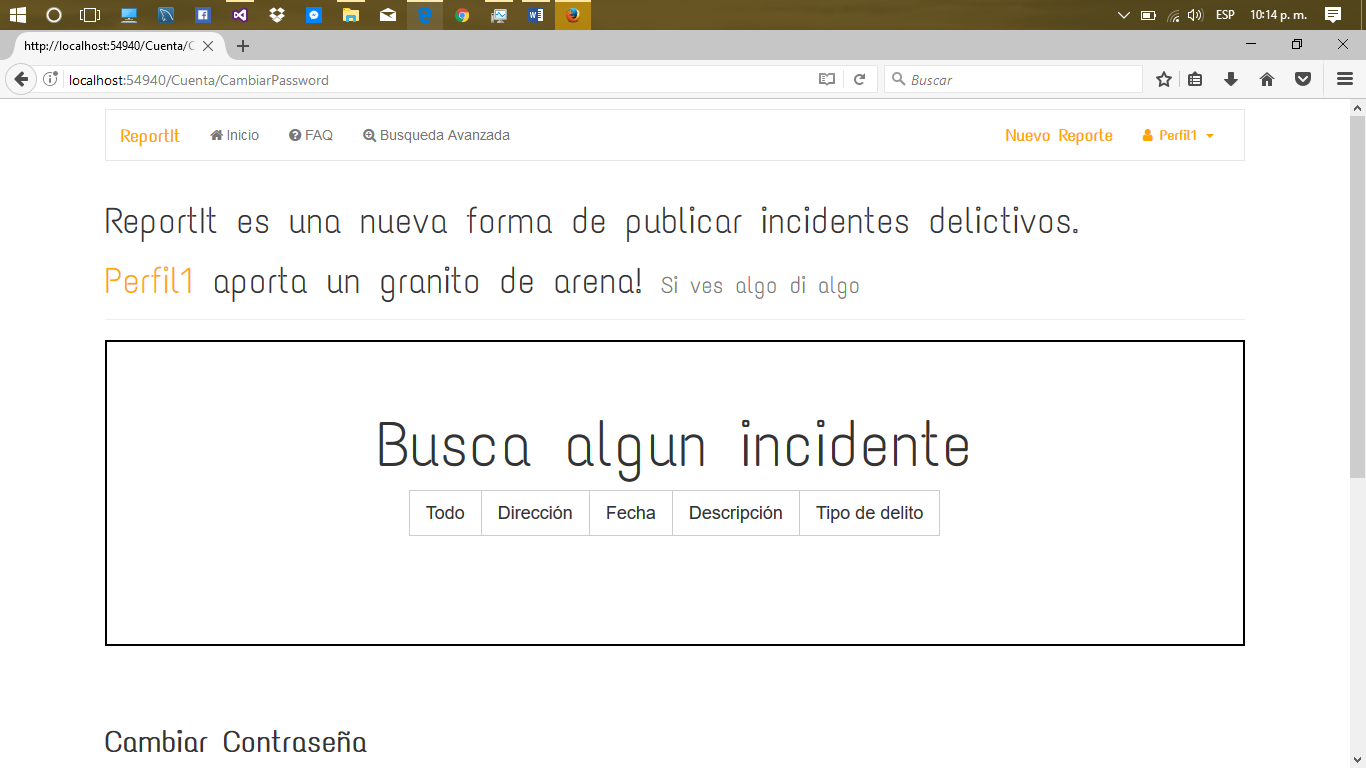


Figura 13 Header de nuestra aplicación: Barra superior (Menú), una introducción y un buscador filtrado.

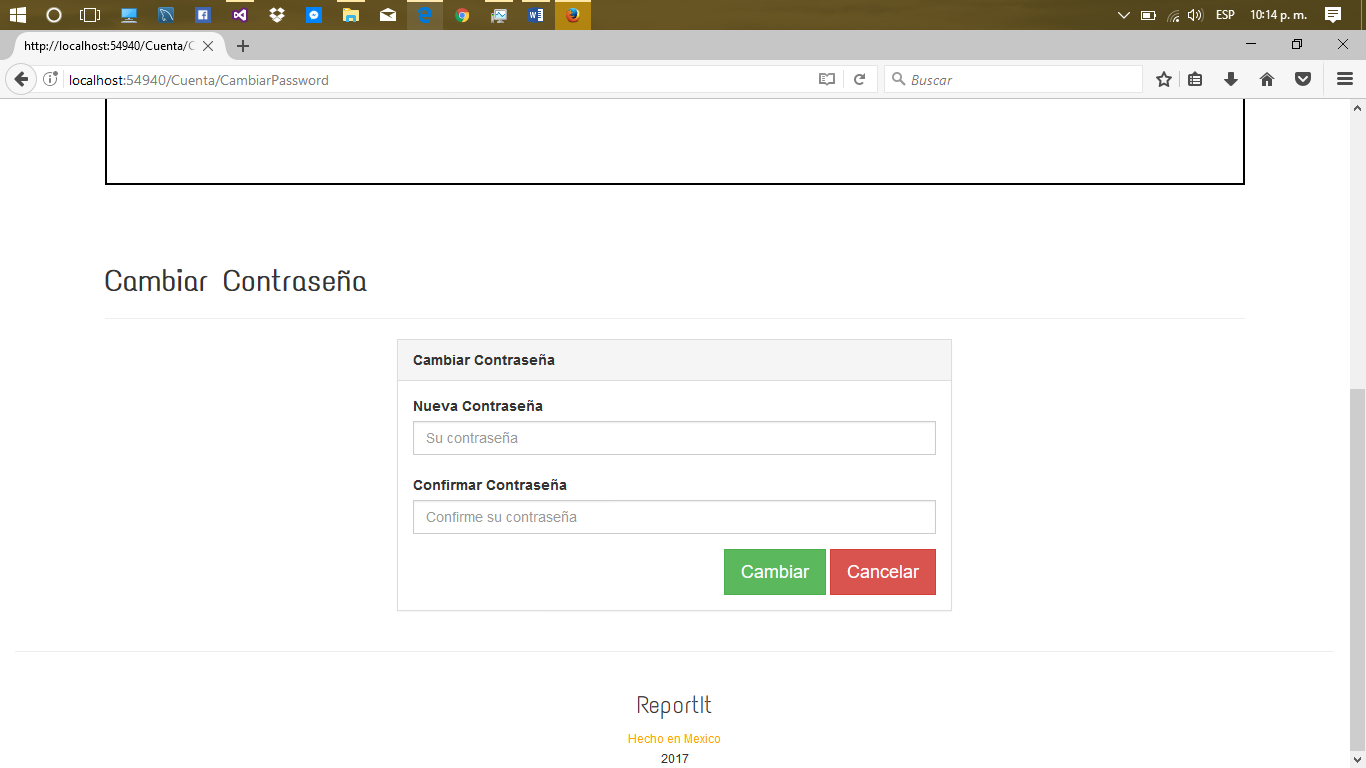


Figura 14 Footer de nuestra aplicación: nombre de nuestra aplicación, año de desarrollo, país origen y este con una liga hacia un acerca del desarrollador de nuestra aplicación.

Desde cualquier vista de nuestra aplicación puedes visualizar que el contenido de estas dos 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 mas demas vistas.

Layout es una vista que encontraras en el apéndice C.2.1 y no es una vista parcial como el Header o el Footer, 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.

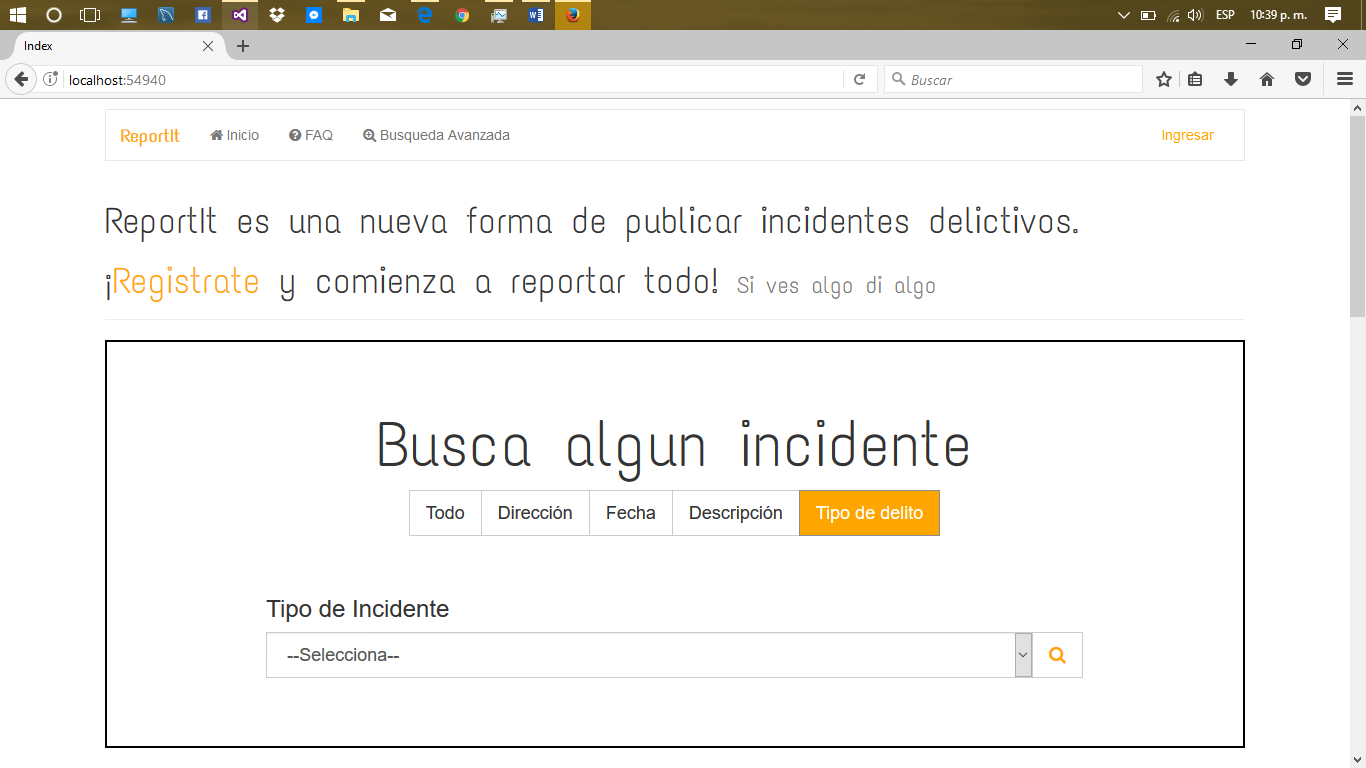


Figura 15 Header dde nuestra aplicación con barra superior sin un usuario autentificado.

El header consta de tres secciones:

* Una barra superior (navbar), nuestro menú, en donde se pueden pueden realizar diferentes acciones dependiendo si estas 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 desi esta o no autentificado el usuario ya que se personaliza. Si se compara la introducción del header de la Figura 13 y 15 se nota que en la 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 pagina y en que les puede beneficiar.

* Un buscador filtrado, este buscador se quedo como buscador principal y tiene cinco filtros, como se explico en la sección 3.2.1 de este capitulo, 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 perso. El segundo filtro, por direccion. 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 ultimo despliega una dropdown como se muestra en la Figura 15.

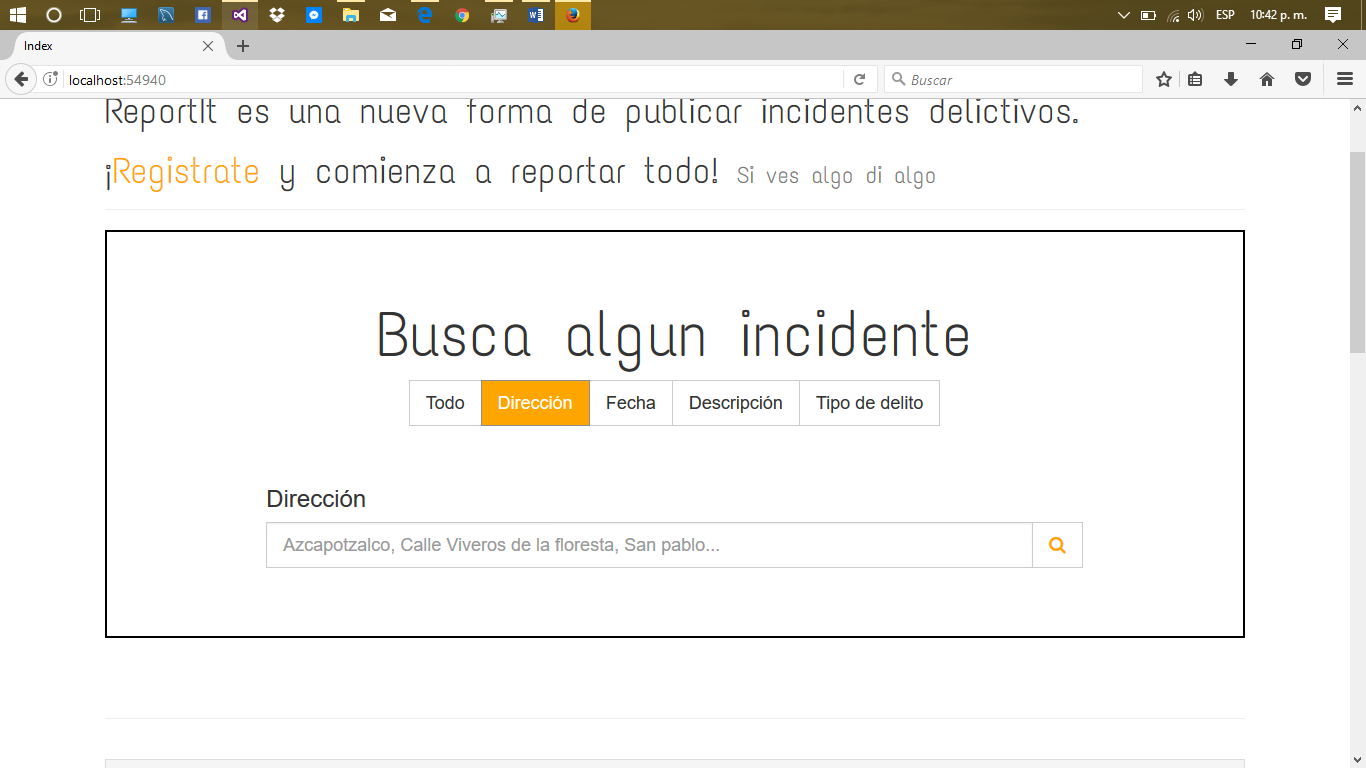


Figura 16 Buscador principar con la opción de filtro por dirección activada.

Para realizar que únicamente se muestre 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 los demas se escondían, este efecto se realizón en un JavaScript mostrado en el Apéndice C.2.4.

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

# Controlador básico (HomeController)

Todos los controladores son una clase como las del modelo de clases, pero en diferencia, los controladores incorporan la interfaz controller para el manejo las vistas. Existen dos tipos de métodos que se pueden imlemenar 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 returnan una vista y posiblemente un modelo también. Mas adelante se explicará que es un modelo y como es que funciona en una vista. En general un método Get se ejecura 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, bueno, esa acción es gracias a estos métodos tipo Get.

Metodos 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 nuev 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 Cuenta el cual hace que los datos ingresado desde la vista sean guardados en la base de datos haciéndolos persistentes en el sistema.

Los modelos de las vistas, son estructuras que se manejaran dentro de las, por ejemplo dentro de detalles de un perfil le damos clic a modificar, este clic ejecuta un método Get el cual retorna una vista (*return View(model)* ) y como argumento se manda el objeto que fungirá como modelo de la vista, en este caso el reporte que se desea modificar y es asi como los datos se mostraran dentro de la vista modificar reporte.

En la esta clase HomeController, se ejecuta el método Get para visualizar las vistas: Inicio, Faq, About, búsqueda avanzada. Y métodos Post como: Busqueda avanzada, busqueda 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 ligara el atributo “Nombre” con el objeto Perfil, asi de fácil es desarrollar con .Net, la otra manera es agregar un objeto tipo *FormCollection,* que abstractamente puede verse como un Dictionary.

# Acerca de (“About”)

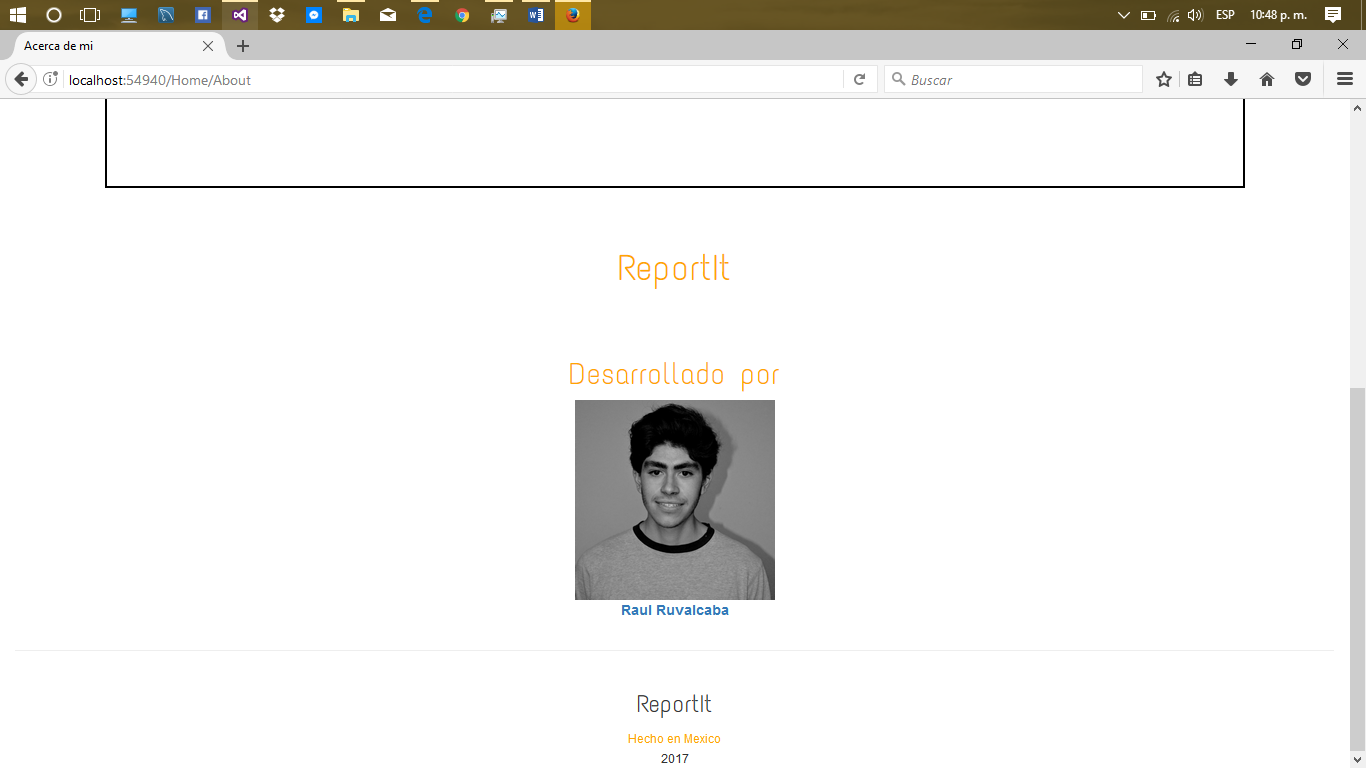


Figura 17 Vista Acerca de, que muestra el desarrollador de esta aplicación y una liga hacia su linkedin.

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

# Inicio

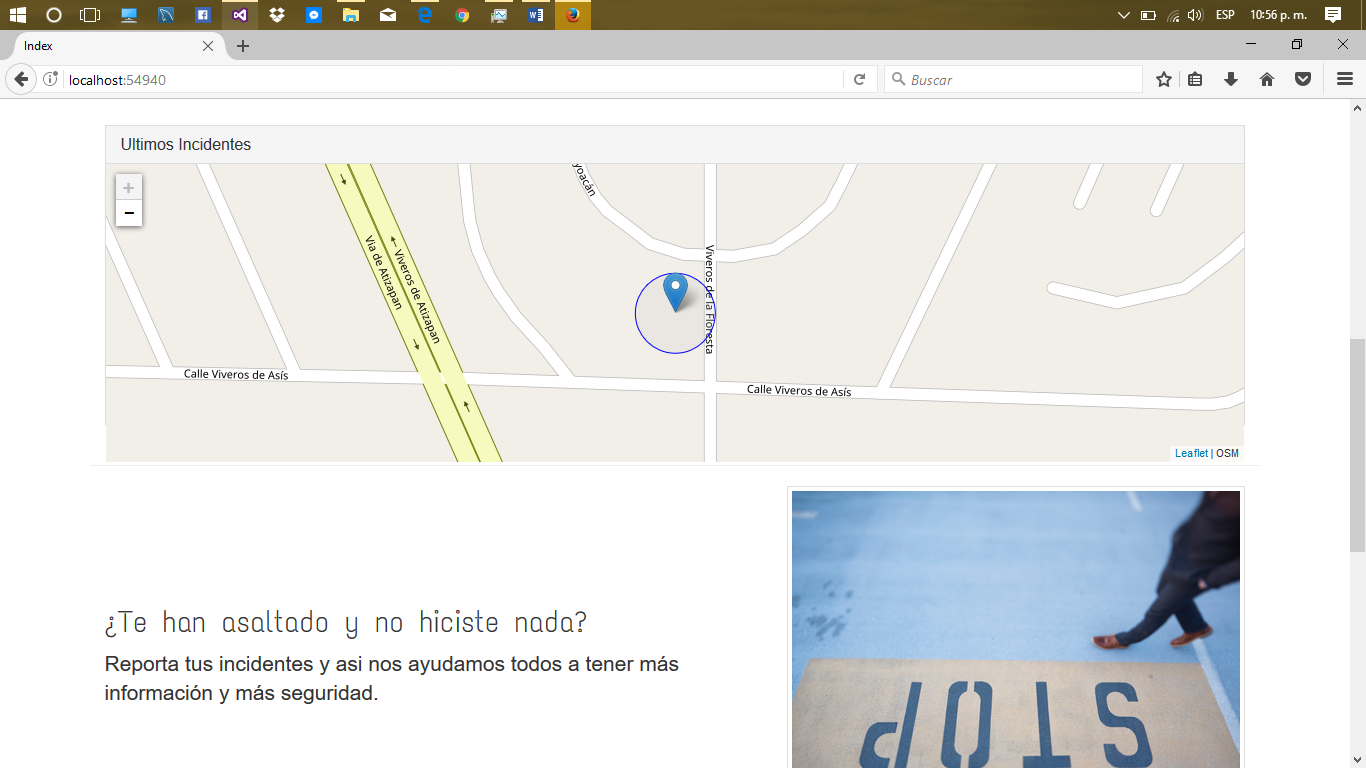


Figura 18 Vista inicio (Primera parte).



Figura 19 Vista inicio (Segunda parte).

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

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

# Preguntas frecuentes (Frequently Asked Questions - FAQ)

Cuando un usuario es nuevo dentro de nuestra aplicación y no sabe como es que funciona correctamente nuestra aplicación para explotarla al cien porciento, se desarrolló esta vista que al igual que la vista de incio, no se plantearon en los objetivos específicos.

El objetivo de esta vista es ayudar a los usuarios con respuestas de las preguntas mas frecuentes que se harían dentro de nuestra aplicación, esto para saber por ejempo: cómo realizar una búsqueda avanzada, cuantos reportes puedo realizar, dónde realizar un reporte, etc.

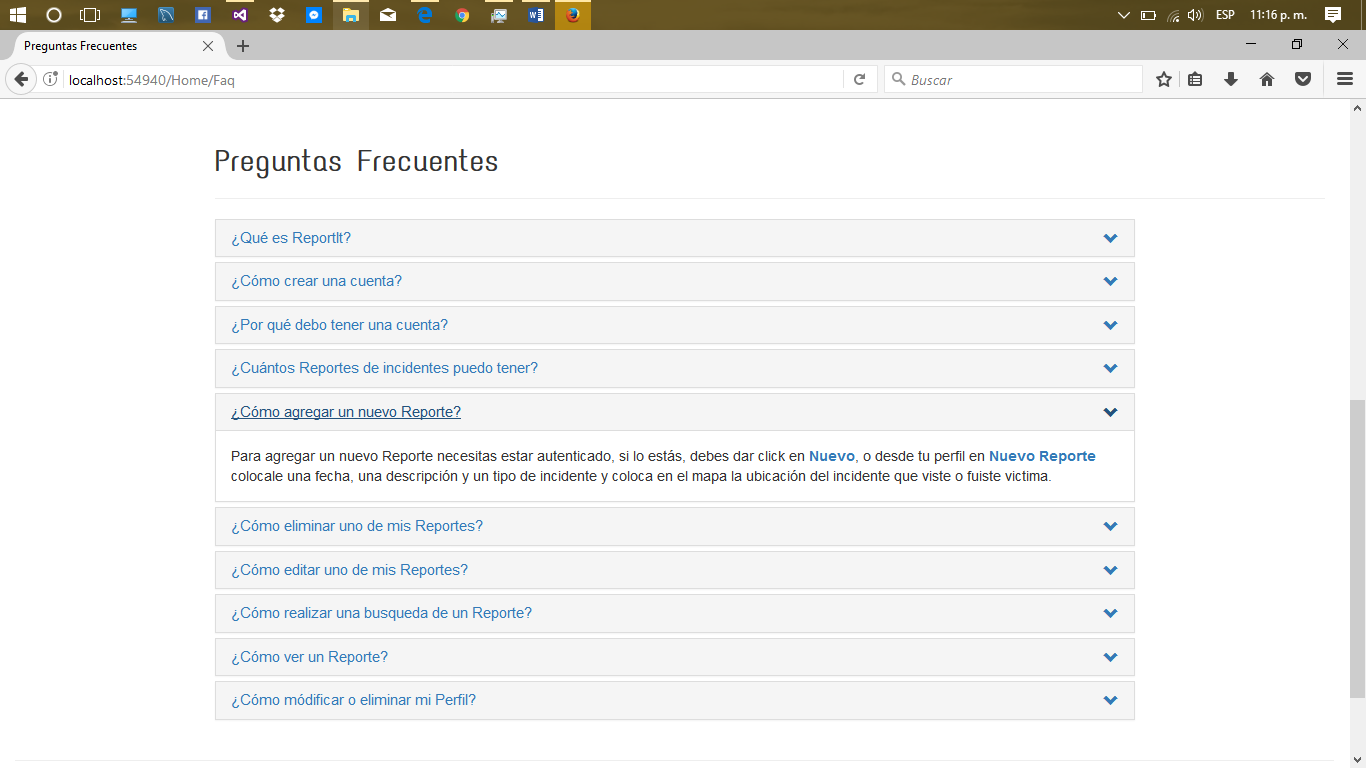


Figura 20 Vista de preguntas frecuentes.

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.

# Controlador de cuenta (CuentaController)

Esta clase controladora se encarga de registrar cookies cuendo un usuario ingresa y limpiar estas 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 asi esta 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 ejecuto 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, 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.

# Controlador de perfil (PerfilController)

Clase controladora de un perfil complementa la clase controladora de cuentas ya que como están fuertemente ligadas una a la otra desde el controlador de cuenta puedes eliminar una cuenta, o crear una cuenta y a la vez el perfil será creado y eliminado, es por ello que en esta clase se implementará métodos como modificar perfil o el visualizar perfil.

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 realiza la paginación de mis reportes

**3.4**: Gestión de un reporte

Resultados

Análisis y Discusión de resultados

//expectativas de un principio a las obtenidas

//mejoras

//observaciones de un usuario

Conclusiones

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] http://eleconomista.com.mx/mercados-estadisticas/2016/02/01/alphabet-mas-valiosa-mercado-desplaza-apple

[10] http://www.capital.cl/negocios/2017/04/04/138299/tesla-supera-a-ford-en-valor-de-mercado

[11] http://jordisan.net/blog/2006/que-es-un-framework/

[12] https://msdn.microsoft.com/library/dn762121(v=vs.140).aspx

[13] https://www.somosbinarios.es/visual-studio-y-la-telemetria/

[14] https://dev.mysql.com/downloads/connector/net/6.0.html

[15] https://opbuildstorageprod.blob.core.windows.net/output-pdf-files/en-us/VS.core-docs/live.pdf

[16] https://www.thoughtco.com/what-is-javascript-2037921

[17] http://www.yourhtmlsource.com/starthere/whatishtml.html

[18] http://www.allaboutcookies.org/cookies/session-cookies-used-for.html

[19] https://www.boston.co.uk/info/cookies.aspx

[20] <http://leafletjs.com/>

[21] http://leafletjs.com/reference-1.1.0.html

**ReportIt**

* **Modelo de clases**
* **Modelo entidad relación**
* **Biblioteca de clases**
* **Consola para testeo**
* **Proyecto web : ASP .NET MVC**
* **Referencias agregadas**
* **Uso de Home controller (GET :: POST)**
* **Uso de un metodo Get para una vista**
* **Uso del Layout en las vistas**
* **Manejador de Sesiones (Http cookies)**
* **Uso de Cuuenta controller y sus vistas**
* **Send Email de bienvenida y para recuperación de contraseña**
* **Como cambiar contraseña**
* **Uso de la librería LEAFLET para mapas (Qué es, Hoja de estilos)**
* **Utils. Geolocation**

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=qwer12345678";

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.CultureInfo.InvariantCulture);

var longitud = Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion\_Longitud"]).ToString(System.Globalization.CultureInfo.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.CultureInfo.InvariantCulture);

var longitud = Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion\_Longitud"]).ToString(System.Globalization.CultureInfo.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.CultureInfo.InvariantCulture);

var longitud = Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion\_Longitud"]).ToString(System.Globalization.CultureInfo.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.CultureInfo.InvariantCulture);

var longitud = Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion\_Longitud"]).ToString(System.Globalization.CultureInfo.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.CultureInfo.InvariantCulture);

var longitud = Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion\_Longitud"]).ToString(System.Globalization.CultureInfo.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.CultureInfo.InvariantCulture);

var longitud = Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion\_Longitud"]).ToString(System.Globalization.CultureInfo.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.CultureInfo.InvariantCulture);

var longitud = Convert.ToDouble(datos.Tables[0].Rows[i]["Ubicacion\_Longitud"]).ToString(System.Globalization.CultureInfo.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.Contrasena AS Cuenta\_Contrasena,

u.Latitude AS Ubicacion\_Latitud,

u.Longitude AS Ubicacion\_Longitud,

u.Delegacion AS 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.Contrasena AS Cuenta\_Contrasena,

u.Latitude AS Ubicacion\_Latitud,

u.Longitude AS Ubicacion\_Longitud,

u.Delegacion AS 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.3.8**: 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.Contrasena AS Cuenta\_Contrasena,

u.Latitude AS Ubicacion\_Latitud,

u.Longitude AS Ubicacion\_Longitud,

u.Delegacion AS 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.2.8**: 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.Contrasena AS Cuenta\_Contrasena,

u.Latitude AS Ubicacion\_Latitud,

u.Longitude AS Ubicacion\_Longitud,

u.Delegacion AS 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.2.9**: 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.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 B: Controladores

**Apéndice B.2**: CuentaController

using Proyecto\_Integracion.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Proyecto\_Integracion.WebApp.Controllers

{

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 B.3**: PerfilController

using Proyecto\_Integracion.Models;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using PagedList.Mvc;

using PagedList;

namespace Proyecto\_Integracion.WebApp.Controllers

{

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 B.4**: ReporteController

using Proyecto\_Integracion.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using Proyecto\_Integracion.WebApp.Utils;

using System.Globalization;

namespace Proyecto\_Integracion.WebApp.Controllers

{

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 C: Vistas

**Apéndice C - Home.1**: 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>.GetDisplayValue(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 - Home.2**: 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 - Home.3**: 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 - Home.4**: Busqueda Avanzada

@model Proyecto\_Integracion.Models.Reporte

@{

Layout = "~/Views/\_Layout.cshtml";

ViewBag.Title = "Búsqueda Avanzada";

var perfil\_activo = Proyecto\_Integracion.WebApp.Utils.SessionManager.PerfilActivo();

var cuenta\_activa = Proyecto\_Integracion.WebApp.Utils.SessionManager.CuentaActiva();

}

@section Scripts{

<script type="text/javascript">

$(".header\_2").hide();

$(".header\_1").hide();

$(".busqueda\_avanzada").hide();

var check\_clicked = function () {

if ($('#mi\_ubicacion').is(':checked')) {

var input\_check = document.getElementById('check\_miUbicacion');

input\_check.value = "True";

}

else {

var input\_check = document.getElementById('check\_miUbicacion');

input\_check.value = "";

}

}

$(document).ready(function () {

$("#fecha").datepicker({

dateFormat: 'yy-mm-dd'

});

});

</script>

}

<div class="container">

<br />

<h2 class="letra-prompt"><strong>Busqueda Avanzada</strong></h2>

<hr />

@using (Html.BeginForm("ResultadosAvanzados", "Home", FormMethod.Post, new { @class = "form-horizontal" }))

{

<fieldset>

<div class="form-group">

<label for="inputPassword3" class="col-sm-2 control-label letra-prompt">Palabra Clave</label>

<div class="col-sm-6">

<input type="text" class="form-control" id="palabra" name="palabra" placeholder="Robo a mano armada, Lesión con arma blanca, etc.">

</div>

<p class="col-sm-4">

Ingresa palabras de descripciones de reportes.

</p>

</div>

<div class="form-group">

<label for="inputPassword3" class="col-sm-2 control-label letra-prompt">Fecha</label>

<div class="col-sm-6">

<input type="text" class="form-control" id="fecha" name="fecha" placeholder="2017-02-21">

</div>

<p class="col-sm-4">

Ingresa fecha de publicaciones de reportes.

</p>

</div>

<div class="form-group">

<label for="inputPassword3" class="col-sm-2 control-label letra-prompt">Tipo de Incidente</label>

<div class="col-sm-6">

@Html.EnumDropDownListFor(x => x.Incidente, "--Selecciona--", new { @class = "form-control" })

</div>

<p class="col-sm-4">

Seleccióna el tipo de incidente de los reportes a consultar.

</p>

</div>

<div class="form-group">

<label for="inputPassword3" class="col-sm-2 control-label letra-prompt">Dirección</label>

<div class="col-sm-6">

<input type="text" class="form-control" id="direccion" name="direccion" placeholder="Azcapotzalco, Calle Viveros de la floresta, San pablo... ">

</div>

<p class="col-sm-4">

Ingresa la dirección de reportes a consultar.

</p>

</div>

<div class="form-group">

<label for="inputPassword3" class="col-sm-2 control-label letra-prompt">Nombre de un Perfil</label>

<div class="col-sm-6">

<input type="text" class="form-control" id="nombrePerfil" name="nombrePerfil" placeholder="Alejandro, Monik\_0223, RaulAB...">

</div>

<p class="col-sm-4">

Filtra por el nombre del <strong>Perfil</strong> que realizo los reportes que desas buscar.

</p>

</div>

@if (perfil\_activo != null && cuenta\_activa != null)

{

<div class="form-group">

<div class="col-sm-offset-2 col-sm-6">

<div class="checkbox">

<label>

<input id="mi\_ubicacion" onclick="check\_clicked()" type="checkbox"> Buscar por mi ubicación

</label>

</div>

</div>

</div>

<input id="Latitud" name="Latitud" type="text" value="@perfil\_activo.Ubicacion.Latitud" hidden />

<input id="Longitud" name="Longitud" type="text" value="@perfil\_activo.Ubicacion.Longitud" hidden />

<input id="Id" name="Id" type="text" value="@perfil\_activo.Ubicacion.Id" hidden />

}

<input id="check\_miUbicacion" name="check\_miUbicacion" type="text" hidden />

<div class="form-group">

<div class="col-sm-offset-2 col-sm-6 text-right">

<button type="submit" class="btn btn-primary"><i class="fa fa-search" aria-hidden="true"></i> Buscar</button>

</div>

</div>

</fieldset>

}

</div>

**Apéndice C - Home.5**: \_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 - Home.6**:

\_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 - Home.7**:

\_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 - Home.8**: \_CollapseDropDown\_MiUbicacion - Parcial

@using (Html.BeginForm("BusquedaGeneral", "Home", new { page = 1, cantResult = 10, filtro = 4 }, FormMethod.Post))

{//Busqueda por Mi Ubicación

<fieldset>

<div class="col-lg-offset-1 col-lg-10">

<h3 class="text-left">Mi Ubicación</h3>

<div class="input-group">

<input type="text" name="buscar" id="buscar" class="form-control input-lg" placeholder="Busca un incidente por cualquier dato en tus alrededores..." />

<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 - Home.9**:

\_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 - Home.10**:

\_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 - Home.11**: 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>.GetDisplayValue(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>

**Apéndice C - Home.11**: Resultados Avanzados

@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();

Proyecto\_Integracion.Models.TipoIncidente incidente\_buscado = 0;

if (ViewBag.Incidente != "0")

{

incidente\_buscado = Proyecto\_Integracion.WebApp.Utils.EnumHelper<Proyecto\_Integracion.Models.TipoIncidente>.Change(ViewBag.Incidente);

}

}

@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">

@{

String palabra = null;

String fecha = null;

String direccion = null;

String nombrePerfil =null;

String incidente\_des = null;

if (ViewBag.Palabra != null)

{

palabra = "Palabra: " + ViewBag.Palabra + ",";

}

if (ViewBag.Fecha != null)

{

fecha = " Fecha: " + ViewBag.Fecha + ",";

}

if (ViewBag.Direccion != null)

{

direccion = " Dirección: " + ViewBag.Direccion + ",";

}

if (ViewBag.NombrePerfil != null)

{

nombrePerfil = " Nombre de un Perfil: " + ViewBag.NombrePerfil + ",";

}

if (ViewBag.Incidente != "0")

{

incidente\_des = " Tipo de Incidente: " + ViewBag.Incidente + ".";

}

}

<h4 class="panel-title">

Resultados de Busqueda Avanzada: <strong>@palabra @fecha @direccion @nombrePerfil @incidente\_des</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>@palabra @fecha @direccion @nombrePerfil @incidente\_des</strong>"</p>

</div>

</div>

}

@{var i = 0; }

@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>.GetDisplayValue(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("ResultadosAvanzados", new { page = ViewBag.Page - 1, cantResult = ViewBag.Results, palabra = ViewBag.Palabra, ViewBag.Fecha, direccion = ViewBag.Direccion, nombrePerfil = ViewBag.NombrePerfil, incidente = incidente\_buscado, radio = 40, latitud = ViewBag.Latitud, longitud = ViewBag.Longitud, check\_miUbicacion = ViewBag.Check\_Ubicacion})"

class="btn btn-default">

&laquo; Atras

</a>

}

else

{

<a href="@Url.Action("ResultadosAvanzados", new { page = ViewBag.Page, cantResult = ViewBag.Results, palabra = ViewBag.Palabra, ViewBag.Fecha, direccion = ViewBag.Direccion, nombrePerfil = ViewBag.NombrePerfil, incidente = incidente\_buscado, radio = 40, latitud = ViewBag.Latitud, longitud = ViewBag.Longitud, check\_miUbicacion = ViewBag.Check\_Ubicacion } )" class="btn btn-default" style="opacity: 0.65; cursor: not-allowed;" disabled>

&laquo; Atras

</a>

}

@if (ViewBag.More > 0)

{

<a href="@Url.Action("ResultadosAvanzados", new { page = ViewBag.Page + 1, cantResult = ViewBag.Results, palabra = ViewBag.Palabra, ViewBag.Fecha, direccion = ViewBag.Direccion, nombrePerfil = ViewBag.NombrePerfil, incidente = incidente\_buscado, radio = 40, latitud = ViewBag.Latitud, longitud = ViewBag.Longitud, check\_miUbicacion = ViewBag.Check\_Ubicacion } )"

class="btn btn-default">

Siguiente &raquo;

</a>

}

else

{

<a href="@Url.Action("ResultadosAvanzados", new { page = ViewBag.Page, cantResult = ViewBag.Results, palabra = ViewBag.Palabra, ViewBag.Fecha, direccion = ViewBag.Direccion, nombrePerfil = ViewBag.NombrePerfil, incidente = incidente\_buscado, radio = 40, latitud = ViewBag.Latitud, longitud = ViewBag.Longitud, check\_miUbicacion = ViewBag.Check\_Ubicacion } )" class="btn btn-default" style="opacity: 0.65; cursor: not-allowed;" disabled>

Siguiente &raquo;

</a>

}

</div>

</div>

</div>

</div>

**Apéndice C - Cuenta.1**: Ingresar

@{

Layout = "~/Views/\_Layout.cshtml";

ViewBag.Title = "Ingresar";

}

@section Styles{

}

@section Scripts{

<script type="text/javascript">

$(".btn\_ingresar").hide();

//$(".header\_2").hide();

$(".header\_1").hide();

</script>

}

<div class="container">

<br />

<h2 class="letra-prompt"><strong>Ingresar</strong></h2>

<hr />

<div class="panel panel-default">

<div class="panel-heading">

<h3 class="panel-title letra-prompt">Ingresar</h3>

</div>

<div class="panel-body">

@using (Html.BeginForm("Ingresar", "Cuenta", FormMethod.Post))

{

<!--<form class="form-horizontal">-->

<div class="row">

<div class="form-group text-center">

<label for="inputEmail3" class="col-sm-2 control-label input-lg letra-prompt">Email</label>

<div class="col-sm-8">

<input type="email" maxlength="80" class="form-control input-lg" name="Email" placeholder="Ingrese su email...">

</div>

</div>

</div>

<div class="row">

<div class="form-group text-center">

<label for="inputPassword3" class="col-sm-2 control-label input-lg letra-prompt">Contraseña</label>

<div class="col-sm-8">

<input type="password" class="form-control input-lg" name="Contrasena" placeholder="Contraseña">

</div>

</div>

</div>

<p class="form-group text-center"><a class="letra-prompt" data-toggle="modal" data-target="#modal-contraseña"><i class="fa fa-lock" arian-hidden="true"></i> ¿Olvidaste tu contraseña?</a></p>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" class="btn btn-primary btn-lg letra-prompt pull-right">Ingresar</button>

</div>

</div>

<!--</form>-->

}

</div>

<div class="panel-footer">

<p><a class="letra-prompt" href="@Url.Action("Crear", "Cuenta")">Registrarme</a></p>

</div>

</div>

</div>

<!-- Modal Contraseña -->

<div id="modal-contraseña" 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">Recuperar contraseña</h4>

</div>

@using (Html.BeginForm("OlvidePassword", "Cuenta", FormMethod.Post))

{

<div class="modal-body">

<p>Ingrese su correo de recuperación.</p>

<div class="form-group">

<label for="Email">Email:</label>

<input class="form-control" name="Email" placeholder="Su Email" maxlength="80" type="text" value="" required>

</div>

</div>

<div class="modal-footer">

<input class="btn btn-primary" value="Recuperar" type="submit" />

<button type="button" class="btn btn-default" data-dismiss="modal">Cerrar</button>

</div>

}

</div>

</div>

</div>

**Apéndice C - Cuenta.2**: Crear Cuenta

@{

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 - Cuenta.3**: Cambiar Password

@model Proyecto\_Integracion.Models.Cuenta

@{

Layout = "~/Views/\_Layout.cshtml";

}

@section Scripts

{

<script>

$(".navbar-search").hide();

</script>

<script>

function validatePassword() {

var password = document.getElementById("Password");

var confirm\_password = document.getElementById("confirm\_password");

if (password.value != confirm\_password.value) {

document.getElementById("mensaje").innerHTML = "Contraseñas no coinciden.";

document.getElementById("button").disabled = true;

} else {

if(password.value.length<6 || confirm\_password.value.length<6)

{

document.getElementById("mensaje").innerHTML = "La contraseña debe tener un largo minimo de 6 caracteres";

document.getElementById("button").disabled = true;

}

else {

document.getElementById("mensaje").innerHTML = "";

document.getElementById("button").disabled = false;

}

}

}

</script>

}

<div class="container">

<br />

<h2 class="letra-prompt"><strong>Cambiar Contraseña</strong></h2>

<hr />

<div class="lm-ingreso">

<div id="formulario" class="row">

<div class="col-md-6 col-md-offset-3 col-sm-8 col-sm-offset-2 col-xs-12">

<div class="panel panel-default">

<div class="panel-heading">

<b>Cambiar Contraseña</b>

</div>

@using (Html.BeginForm("CambiarContrasena", "Cuenta", FormMethod.Post))

{

<div class="panel-body">

<div class="form-group">

@Html.Hidden("Email", Model.Email, new { @class = "hidden" })

<label for="Password"> Nueva Contraseña </label>

<input name="Contrasena" id="Password" type="password" class="form-control" onchange="validatePassword();" placeholder="Su contraseña" value="" maxlength="15" minlength="6" required />

<br>

<label for="confirm\_password"> Confirmar Contraseña </label>

<input name="confirm\_password" id="confirm\_password" type="password" class="form-control" onkeyup="validatePassword();" placeholder="Confirme su contraseña" maxlength="15" minlength="6" value="" required />

<p id="mensaje"></p>

</div>

<div class="pull-right">

<button type="submit" id="button" class="btn btn-success btn-lg">Cambiar</button>

<button type="button" class="btn btn-danger btn-lg" onclick="irAtras()">Cancelar</button>

</div>

</div>

}

</div>

</div>

</div>

</div>

</div>

**Apéndice C - Perfil.1**: 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>

&nbsp;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>&nbsp;@Model.Nombre</p>

<p><strong>Dirección:</strong>&nbsp;@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&nbsp;&nbsp;

<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>.GetDisplayValue(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 - Perfil.2**: 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 C - Perfil.3**: Mis Reportes

@model PagedList.IPagedList<Proyecto\_Integracion.Models.Reporte>

@using PagedList.Mvc;

@{

Layout = "~/Views/\_Layout.cshtml";

ViewBag.Title = "Mis Reportes";

var Perfil\_Activo = Proyecto\_Integracion.WebApp.Utils.SessionManager.PerfilActivo();

var Cuenta\_Activa = Proyecto\_Integracion.WebApp.Utils.SessionManager.CuentaActiva();

}

@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>Mis Reportes</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"> Mis reportes</h4>

</div>

<div class="panel-body">

<div class="list-group text-right">

<label>Pagina @( Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) de @Model.PageCount</label>

@Html.PagedListPager(Model, page => Url.Action("MisReportes", new { page, sortOrder = ViewBag.CurrentSort, currentFilter = ViewBag.CurrentFilter }))

<div class="row" style="display: flex; flex-wrap: wrap;">

<input id="report\_count" value="@Model.Count()" hidden/>

@{

var i = 0;

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>.GetDisplayValue(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++;

}

}

</div>

</div>

</div>

</div>

</div>

</div>

**Apéndice C - Reporte.1**: 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 C - Reporte.2**: 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>

&nbsp;Regresar

</a>

}

else

{

<a class="btn btn-success" onclick="irAtras()">

<i class="fa fa-chevron-left" aria-hidden="true"></i>

&nbsp;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>.GetDisplayValue(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&nbsp;&nbsp;

<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 C - Reporte.3**: 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 C - Layout.0**: 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 - Layout.1**: 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 - Layout.2**: 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 D: Utilidades

**Apéndice D.1**: Envio de Email

using System;

using System.Net;

using System.Net.Mail;

using System.Threading.Tasks;

namespace Proyecto\_Integracion.WebApp.Utils

{

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 D.2**: Manejo de Enums

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Reflection;

using System.Web;

using System.Web.Mvc;

using System.Web.UI.WebControls;

using Proyecto\_Integracion.Models;

namespace Proyecto\_Integracion.WebApp.Utils

{

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>.GetDisplayValue((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>.GetDisplayValue(incidente))

{

items.Add(new ListItem

{

Text = Proyecto\_Integracion.WebApp.Utils.EnumHelper<Proyecto\_Integracion.Models.TipoIncidente>.GetDisplayValue((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>.GetDisplayValue((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**: Geolocalizacón

using Proyecto\_Integracion.Models;

using System;

using System.Data;

using System.IO;

using System.Net;

using System.Text;

using System.Xml;

using System.Device.Location;

namespace Proyecto\_Integracion.WebApp.Utils

{

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 D.4**: Manejador de sesiones

using Proyecto\_Integracion.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Proyecto\_Integracion.WebApp.Utils

{

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 D.5**: Validador

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Text.RegularExpressions;

namespace Proyecto\_Integracion.WebApp.Utils

{

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 E: JavaScript

**Apéndice E.1**: 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 E.2**: Visualizar en un mapa

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 E.3**: Resultados de búsqueda – pintar en mapa

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 E.4**: modificar ubicación en mapa

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 E.5**: Registrar ubicación en mapa

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 E.6**: Carga de imágenes

$(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 G: Imágenes

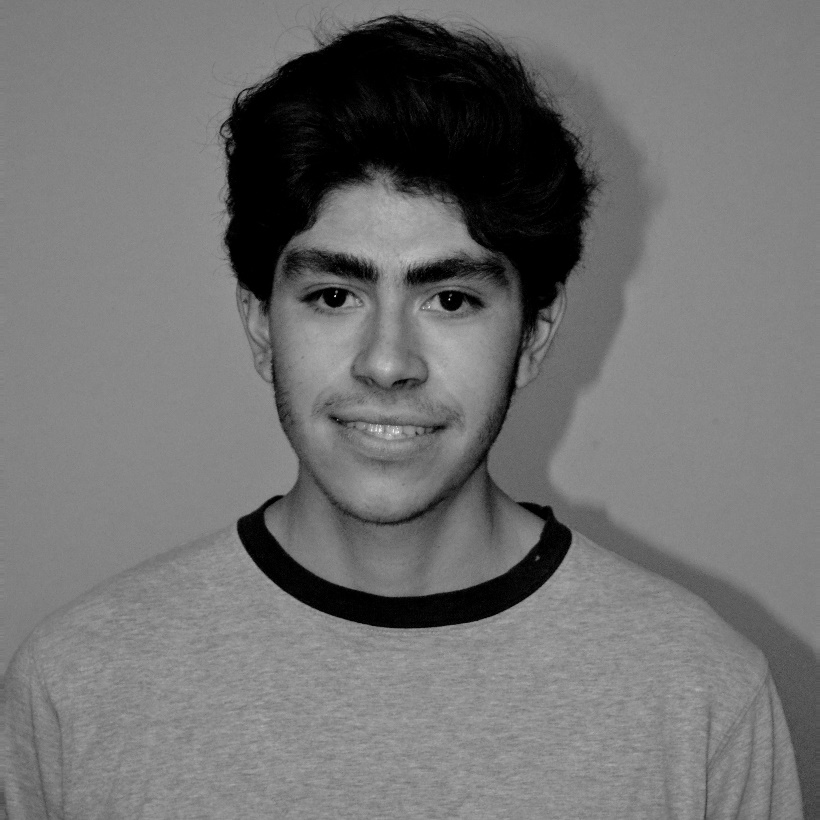
**Apéndice G.1**: Imágenes Index



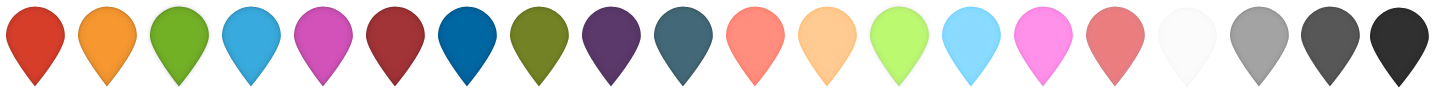
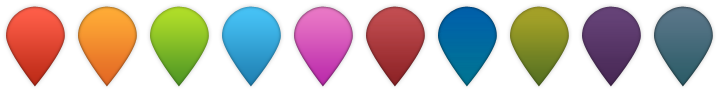
**Apéndice G.2**: Imagen por defecto en un perfil



**Apéndice G.3**: Imagen de acerca del desarrollador



**Apéndice G.4**: Imágenes usadas por leaflet

C:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\layers.pngC:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\markers-soft.pngC:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\markers-shadow@2x.pngC:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\markers-shadow.pngC:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\markers-plain.pngC:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\markers-matte.pngC:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\marker-shadow.pngC:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\marker-icon-2x.pngC:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\marker-icon.pngC:\Users\RF RAUL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\layers-2x.png

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). [↑](#footnote-ref-2)
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. [↑](#footnote-ref-3)
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]. [↑](#footnote-ref-4)
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). [↑](#footnote-ref-5)
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. [↑](#footnote-ref-6)
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. [↑](#footnote-ref-7)
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]. [↑](#footnote-ref-8)