

UNIVERSIDAD DE SONORA

DIVISIÓN DE INGENIERÍA

Departamento de Ingeniería Industrial

**“Desarrollo e implementación de plataforma backend
GIS de código abierto y aplicación GeoMining Touch”**

“Memoria de Prácticas Profesionales”

Que como requisito parcial para la obtención del título

de:

INGENIERO EN SISTEMAS DE INFORMACIÓN

PRESENTA:

JOSUE ALFONZO MARQUEZ NAVARRO

Director: Dr. Mario Barceló Valenzuela

HERMOSILLO, SONORA.

ENERO 2015

Índice General

1. INTRODUCCIÓN	4
2. DESCRIPCIÓN DEL CONTEXTO	6
3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS	7
4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES REALIZADAS	14
5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA EN LA REALIZACION DE LAS PRACTICAS PROFESIONALES	24
6. CONCLUSIONES Y RECOMENDACIONES	26
7. REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES	27

1. INTRODUCCIÓN

1.1 Breve explicación del proyecto o actividad: La empresa Teknol ofrece un software como parte de su servicio llamado GeoMining Touch, actualmente esta aplicación está desarrollada en .NET. GeoMining Touch incorpora herramientas de búsqueda y visualización para información geográfica en México. Este proyecto nació por la necesidad de realizar una nueva aplicación basada en herramientas de código libre para ahorrar el pago de licencias, por ejemplo para la alimentación de los mapas actualmente se usa el servidor GIS MapDotNet de Microsoft y su equivalente en código abierto sería GeoServer. Por parte del desarrollo del GeoMining se utilizarían las herramientas de código abierto como el framework Ionic basado en Apache Cordova para el desarrollo de aplicaciones usando HTML, CSS y JavaScript.

1.2 Objetivos generales que se quieren lograr: Además de que se evitaría pagar licencias con la nueva implementación del servidor GeoServer y el desarrollo de la nueva aplicación GeoMining Touch, gracias al proyecto desarrollado por Apache llamado Cordova y al esfuerzo que han realizado por parte del equipo del framework Ionic, se lograría hacer de GeoMining Touch un software multiplataforma que sería compatible con Android, iOS, Mac, Windows y Linux. Como si todo lo mencionado anteriormente no fuera suficiente, con la nueva implementación de la aplicación se obtendrá un mejor rendimiento y una descarga de información más rápida por parte de GeoServer lo que reducirá notablemente los tiempos de espera en la búsqueda y visualización de información.

1.3 Línea metodológica: Al inicio del proyecto se tuvo que aprender cómo funcionaba el software que llegaría a remplazar con el nuevo desarrollo, se tenía que suplir las mismas necesidades y mejorar el funcionamiento con el que estas operaban. La metodología que se siguió fue muy parecida a la de Lean Software Development ya que lo que se intentaba lograr era hacer un software más óptimo y rápido eliminando

funciones innecesarias. Primeramente se conoció el software que sustituiría, después se aprendió a usar GeoServer que sería el servidor que alimentaría al programa. Una vez familiarizado con el entorno de trabajo y se inició el desarrollo con las herramientas más simples de GeoMining Touch, en el transcurso salían dudas de cómo deberían ser implementados ciertos módulos y para eso siempre estaba disponible un supervisor que además de verificar que se estuviera haciendo el trabajo, el mismo resolvía cualquier duda que se tuviera respecto a cómo se debía de implementar las herramientas. Conforme se avanzaba con el trabajo nuevas ideas eran plasmadas para el nuevo software las cuales tenían que estar concluidas antes de continuar con la línea de desarrollo planeado.

El desarrollo en la metodología Lean puede resumirse los siguientes principios.

Principios de Lean	Metodología usada en el proyecto
<ul style="list-style-type: none"> • Eliminar los desperdicios 	<ul style="list-style-type: none"> • Se eliminó código y funcionalidades innecesarias. • Se evitó retraso en el proceso de desarrollo de software, comunicando los avances cada día con el supervisor.
<ul style="list-style-type: none"> • Ampliar el aprendizaje 	<ul style="list-style-type: none"> • Probando el código una vez estando escrito y documentado después para de esta forma se evitan acumulación de errores.
<ul style="list-style-type: none"> • Decidir lo más tarde posible 	<ul style="list-style-type: none"> • Se realizaban cambios cada vez que se hacía una junta con la directora por nuevas ideas que se aportaban.
<ul style="list-style-type: none"> • Reaccionar tan rápido como sea posible 	<ul style="list-style-type: none"> • Era de vital importancia un desarrollo ágil debido al corto tiempo con el que se contaba. Se tenía que aprender rápido las nuevas tecnologías que se estarían usando en la nueva aplicación.

Tabla 1.1. Representación de la metodología Lean Software Development.

2. DESCRIPCIÓN DEL CONTEXTO

2.1 Equipamiento e instalaciones donde se desarrollaron las actividades que integra el programa de prácticas profesionales: En Teknol se tuvo disposición el equipo necesario para poder concluir satisfactoriamente con el desarrollo del software en su tiempo acordado. Se contaba con un servidor Dell para desarrollo temporal donde se instaló Apache Tomcat y en este GeoServer, también PostgreSQL/PostGIS. Las características de hardware del servidor eran suficientes para que se llevara a cabo el desarrollo del proyecto. Se contaba con una conexión a internet de 10megabits y con una red VPN la cual era necesaria para acceder al servidor en producción de la empresa donde se encontraba la información que alimentaría a GeoServer. Se contaba con una tablet iPad para las pruebas en iOS y una tablet Nexus para las pruebas en Android. Además para el área de trabajo se proporciono un escritorio, silla y una computadora con conexión a internet y buena capacidad de procesamiento y memoria.

2.2 Descripción de la normatividad o reglas de operación del programa o unidad receptora: Platicado con el director de sistemas de Teknol se acordó la presencia para realizar el trabajo de lunes a viernes 6 horas diarias pudiendo ser irregular en algunos días con el compromiso de recuperar las horas perdidas. Para el desarrollo se encomendó aprender cómo funcionaba el software actual y la lectura un par de libros sobre la tecnología que se usaría en la nueva plataforma, se autorizó acceso al repositorio del nuevo y antiguo proyecto en el cual se debía subir por lo menos una actualización diaria del desarrollo del software. En el programa de planificación trello se tenía la lista de cosas por hacer, haciendo y hechas las cuales había que administrar.

2.3 Entorno donde se ubica la unidad receptora: La empresa teknol está compuesta por alrededor de 15 empleados incluidos practicantes, cuenta con secretarias, personal de limpieza, director de sistemas, director de ventas, programadores, geólogo, ingenieros mecatronicos, licenciadas

en administración de empresas y recepcionista, todo este personal coordinado por un director general.

México



Dirección

Pueblo Nuevo 31 Col. Santa Fé; C.P. 83249
Hermosillo, Sonora, México

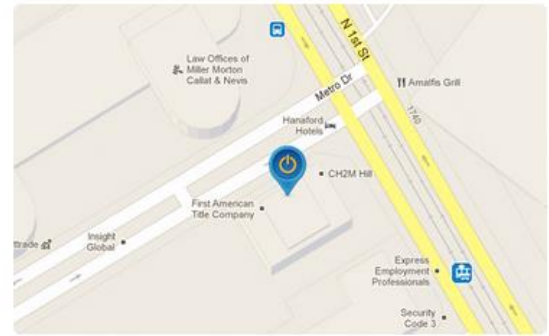
Ventas

ventas@teknol.net
Tel: +52 (662) 218 2900
Línea gratis: 01 800 836 5900

Soporte

En México, contactáctenos en soporte@teknol.net.

Estados Unidos



Dirección

1737 North First Street San Jose, CA 95112

Ventas

sales@teknol.net
(520) 395 9332

Soporte

En los Estados Unidos, contactáctenos
en support@teknol.net.

3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS

En esta sección conoceremos las diferentes herramientas que fueron necesaria para el desarrollo del proyecto viendo una breve introducción a cada una de ellas.

Apache Cordova

Apache Cordova es un framework de licencia libre que cuenta con muchas Apis de diversos dispositivos móviles para desarrollar aplicaciones nativas dentro de un Smartphone. Cada vez está tomando más énfasis en el mundo de los programadores y es que para el desarrollo de las aplicaciones se utilizan las tecnologías web HTML, CSS y JavaScript.

Una de las grandes peculiaridades de este entorno de trabajo es la posibilidad de desarrollar para iOS, Android y demás sistemas operativos sin la necesidad de programar en sus lenguajes nativos (Java, Objective-C, etc.)

Debido a que estas API de JavaScript son consistentes a través de múltiples plataformas de dispositivos y construido en estándares web, la aplicación debe ser portable a otras plataformas de dispositivos con un mínimo o ningún cambio.

Aplicaciones que utilizan Cordova se empaquetan como aplicaciones que utilizan los SDK de la plataforma, y pueden estar disponibles para la instalación desde la tienda de aplicaciones de cada dispositivo.

Ionic Framework

Ionic es un framework de aplicaciones móviles HTML5 dirigida a la creación de aplicaciones móviles híbridas. Las aplicaciones híbridas son esencialmente pequeños sitios web que se ejecutan en un navegador web dentro de una aplicación que tienen acceso a la capa de plataforma nativa. Las aplicaciones híbridas tienen muchas ventajas sobre las aplicaciones nativas puras, específicamente en términos de soporte de la plataforma, la velocidad de desarrollo, y el acceso a código externo.

Ionic se centra principalmente en la apariencia, y la interacción UI de la aplicación. Eso significa que no es un reemplazo para PhoneGap o cualquier otro framework JavaScript.

Ionic actualmente requiere AngularJS para trabajar en su pleno potencial. Si bien todavía se puede utilizar la parte CSS del framework, se perderá en poderosas interacciones de IU, gestos, animaciones y otras cosas.

AngularJS

AngularJS es un framework de JavaScript de código abierto, mantenido por Google, que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticas o dinámicas.

AngularJS está construido en torno a la creencia de que la programación declarativa es la que debe utilizarse para generar interfaces de usuario y enlazar componentes de software, mientras que la programación imperativa es excelente para expresar la lógica de negocio. Este framework adapta y amplía el HTML tradicional para servir mejor contenido dinámico a través de un data-binding bidireccional que permite la sincronización automática de modelos y vistas. Como resultado, AngularJS pone menos énfasis en la manipulación del DOM y mejora la testeabilidad y el rendimiento.

Los objetivos de diseño:

- Disociar la manipulación del DOM de la lógica de la aplicación. Esto mejora la capacidad de prueba del código.
- Considerar a las pruebas de la aplicación como iguales en importancia a la escritura de la aplicación. La dificultad de las pruebas se ve reducida dramáticamente por la forma en que el código está estructurado.
- Disociar el lado del cliente de una aplicación del lado del servidor. Esto permite que el trabajo de desarrollo avance en paralelo, y permite la reutilización de ambos lados.

- Guiar a los desarrolladores a través de todo el camino de la construcción de una aplicación: desde el diseño de la interfaz de usuario, a través de la escritura de la lógica del negocio, hasta las pruebas.

Angular sigue el patrón MVC de ingeniería de software y alienta la articulación flexible entre la presentación, datos y componentes lógicos. Con el uso de la inyección de dependencias, Angular lleva servicios tradicionales del lado del servidor, tales como controladores dependientes de la vista, a las aplicaciones web del lado del cliente. En consecuencia, gran parte de la carga en el backend se reduce, lo que conlleva a aplicaciones web mucho más ligeras.

jQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

OpenLayers

OpenLayers es una biblioteca de JavaScript de código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores web. OpenLayers ofrece un API para acceder a diferentes fuentes de información cartográfica en la red: Web Map Services, Mapas comerciales

(tipo Google Maps, Bing, Yahoo), Web Features Services, distintos formatos vectoriales, mapas de OpenStreetMap, etc.

Inicialmente fue desarrollado por MetaCarta en Junio del 2006. Desde el noviembre del 2007 este proyecto forma parte de los proyectos de Open Source Geospatial Foundation. Actualmente el desarrollo y el soporte corre a cargo de la comunidad de colaboradores.

GeoServer

GeoServer es un servidor de código abierto escrito en Java, permite a los usuarios compartir y editar datos geoespaciales. Diseñado para la interoperabilidad, publica datos de las principales fuentes de datos espaciales usando estándares abiertos. GeoServer ha evolucionado hasta llegar a ser un método sencillo de conectar información existente a globos virtuales tales como Google Earth y NASA World Wind (véase así como mapas basados en web como OpenLayers, Google Maps y Bing Maps). GeoServer sirve de implementación de referencia del estándar Open Geospatial Consortium Web Feature Service, y también implementa las especificaciones de Web Map Service y Web Coverage Service.

GeoServer pretende operar como un nodo a través de una Infraestructura de Datos Espaciales libre y abierta para ofrecer datos geoespaciales, tal y como ha hecho Apache HTTP Server ofreciendo un servidor web abierto y libre para publicar HTML.

Entre las principales características de Geoserver se pueden citar algunas como:

- Enteramente compatible con las especificaciones WMS, WCS y WFS, testados por el test de conformidad CITE de la OGC.
- Fácil utilización a través de la herramienta de administración vía web -no es necesario entrar en archivos de configuración grandes y complicados.

- Soporte amplio de formatos de entrada PostGIS, Shapefile, ArcSDE y Oracle. VFP, MySQL, MapInfo y WFS en cascada también están entre los formatos de entrada soportados (véase más abajo).
- Soporte de formatos de salida tales como JPEG, GIF, PNG, SVG y GML.
- Imágenes con antialiasing.
- Soporte completo de SLD, como definiciones del usuario (POST y GET), y como uso de configuración de estilos.
- Soporte para edición de datos de banco de datos individuales a través del protocolo WFS transaccional profile (WFS-T), disponible para todos los formatos de datos.
- Basado en servlets Java (JEE), puede funcionar en cualquier servlet contenedor.
- Diseñado para ser compatible con extensiones.
- Facilidad de escritura de nuevos formatos de datos con la interfaz de almacenamiento de datos GeoTools y clases de ayuda.

Además, GeoServer incluye un cliente integrado OpenLayers para previsualizar capas de datos. GeoServer también soporta la publicación de datos geoespaciales para Google Earth a través del uso de enlaces de red, utilizando KML para ello. Funciones avanzadas disponibles para output de Google Earth incluyen plantillas para pop-ups personalizados, visualizado de altitud y longitud, y "super-overlays".

GeoServer se basa en GeoTools, una biblioteca de sistemas de información geográfica.

QGIS

QGIS (anteriormente llamado también Quantum GIS) es un Sistema de Información Geográfica (SIG) de código libre para plataformas GNU/Linux, Unix, Mac OS y Microsoft Windows. Era uno de los primeros ocho proyectos de la Fundación OSGeo y en 2008 oficialmente graduó de la fase de incubación. Permite manejar formatos raster y vectoriales a través de las bibliotecas GDAL y OGR, así como bases de datos. Algunas de sus características son:

- Soporte para la extensión espacial de PostgreSQL, PostGIS.
- Manejo de archivos vectoriales Shapefile, ArcInfo coverages, Mapinfo, GRASS GIS, etc.
- Soporte para un importante número de tipos de archivos raster (GRASS GIS, GeoTIFF, TIFF, JPG, etc.)

Apache Tomcat

Tomcat es un contenedor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

PostgreSQL

PostgreSQL es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostGIS

PostGIS es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en Sistema de Información Geográfica. Se publica bajo la Licencia Pública General de GNU.

4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES REALIZADAS

El desarrollo de la aplicación GeoMining Touch fue sobre el framework Ionic que se ayuda con la herramienta web AngularJS, esta última puede funcionar sin la intervención de la librería jQuery pero funciona muy bien en conjunto con ella. Se eligió utilizar jQuery dado al conocimiento con el que se cuenta acerca de la librería ya que esto haría más ágil el desarrollo. La mayoría de las implementaciones que veremos en esta sección se realizaron con el trabajo en conjunto de AngularJS y jQuery.

Instalación y configuración de backend GeoServer / PostGIS

29/Septiembre - 3/Octubre

En un servidor Windows 2012 se instalaron y configuraron cada uno de los paquetes listados a continuación, vitales para el desarrollo del proyecto ya que estos serán los que alimenten de información a la aplicación final.

Pasos:

1. Instalación y configuración de Tomcat 7

Para poder obtener funcionando Tomcat 7 fue necesario tener instalado Java en el servidor, la instalación de este software fue trivial. Tomcat es una aplicación web que se ejecuta sobre la plataforma Java y existe para precisamente correr aplicaciones web Java. Hubo algunas configuraciones de optimización que se realizaron necesarias para obtener el máximo rendimiento de Tomcat 7 en el servidor.

2. Instalación y configuración de la aplicación web GeoServer

Una vez instalado y corriendo Tomcat se instaló la aplicación web GeoServer que se ejecuta sobre este contenedor web. La configuración de GeoServer la veremos después.

3. Instalación y configuración del servidor PostgreSQL

Pero de qué sirve las aplicaciones que instalemos si no le brindamos algo de información, para esto se utiliza el motor de base de datos PostgreSQL. Siguiendo le estándar de código abierto y que además es el motor que cuenta con más funciones cartográficas en la actualidad PostgreSQL fue el elegido esta vez para realizar el importante trabajo de servir la aplicación con datos.

4. Instalación de la extensión PostGIS en PostgreSQL

PostGIS es la extensión para el manejo de datos geográficos por excelencia para PostgreSQL. Con esta extensión fue posible cargar Shapefiles en la base de datos, estos Shapefiles contienen información geográfica. Una vez importados los archivos estos son conectados desde GeoServer y son leídos a GeoMining.

Capacitación en el uso de las herramientas GeoServer / PostGIS

6/Octubre - 17/Octubre

Para poder manejar la herramienta GeoServer así como PostGIS se recurrió a la ayuda del libro: **GeoServer, Stefano Lacovella & Brian Youngblood, Feb 2013.**

En este libro pudimos aprender como instalar Tomcat 7 en los diferentes sistemas operativos que existen, también nos enseñó a instalar GeoServer y configurarlo así como PostgreSQL, PostGIS, QGIS y entre otras herramientas útiles en los sistemas de información geográfica.

Entender los conceptos de cartografía es importante ya que estamos manejando formatos de datos espaciales. En GeoServer aprendí a manejar sus diferentes tipos de almacenamiento de datos como lo son:

- Shapefile: es un formato muy común en GIS y es el que utilizamos para exportar en PostgreSQL.
- PostGIS: es una famosa fuente de datos espacial.

Cada capa está configurada con un tipo de proyección, por ejemplo la más común es la EPSG:4326 usada en los mapas de google. Las capas vienen acompañadas también con una hoja de estilo la cual especifica como deberá visualizarse en el mapa.

Después que tenemos nuestros datos en PostgreSQL y los Layers están listos con sus respectivos estilos es hora de servirlos al cliente, esto se hace con 3 diferentes tipos de servicios que ofrece GeoServer: WCS, WMS, WFS.

Web Map Server (WMS): cuenta con un método llamado GetMap, este deja al cliente hacer peticiones por mapas como imágenes, por ejemplo, png y jpg.

Web Feature Service (WFS): permite compartir datos geoespaciales en el formato estándar, por ejemplo Shapefile, JSON y CSV.

Web Coverage Format (WCS): publica capas raster-based. Geo ArcGrid son un par de ejemplos cobertura geoespaciales. Es como tener WMS y WFS en un solo servicio.

GeoServer cuenta con diferentes maneras de agregar los datos de mapas como habíamos mencionado antes. Se puede agregar Shapefiles directamente con dos fuentes de datos. Con el primero se configura una carpeta conteniendo un grupo de Shapefiles con la opción de agregar nuevos después que la fuente de datos esta creada. La otra fuente de datos funciona de la misma manera como el directorio de almacenamiento del Shapefile, excepto que se provee un directorio para un solo Shapefile.

La otra forma de agregar información a nuestro GeoServer para que este comience a servir capas de mapas es a través de PostGIS. Esta forma es la más popular y la que hemos elegido para este proyecto ya que los datos son leídos directamente de la base de datos PostgreSQL lo que significa que tenemos más libertad al momento de manejar las consultas de datos.

PostGIS es la herramienta más capaz de todas las base de datos relacionales de código abierto con capacidades espaciales. A continuación listamos los pasos que seguimos para importar datos a GeoServer a través de PostGIS.

1. Descargamos e instalamos la extensión de PostGIS.
2. Ejecutamos la herramienta de importación/exportación de PostGIS. Hacemos clic en el botón "View connections details" y después insertamos los parámetros necesarios para conectarnos a la base de datos.
3. Hicimos clic en el botón de agregar archivo y navegamos hasta el directorio donde se encuentra el Shapefile que queremos agregar. Esta herramienta no reconoce SRS (Spatial Reference System) contenido en el archivo prj por lo tanto insertamos 4326.
4. Después hicimos clic en el botón de importar y establecimos la codificación del archivo LATIN1 como DBF.
5. Esperamos hasta que PostGIS convierta los datos y los inserte en la nueva tabla, una vez hecho esto verificamos en PostgreSQL y existía

una tabla con el nombre del Shapefile conteniendo toda la información geográfica del archivo lista para ser consultada.

Implementar coberturas de QuickMining Library en GeoServer / PostGIS

20/Octubre - 31/Octubre

Pasos:

1. Cargar Shapefiles en PostgreSQL mediante PostGIS
2. Cargar capas en GeoServer de PostgreSQL a través de PostGIS
3. Crear los estilos para cada una de las capas mediante SLD (Styled Layer Descriptor)

QuickMining Library es un servicio de consulta de información que ofrece la empresa teknol, está constituido por varias secciones como por ejemplo: Lotes en Proceso, Mapa Metalogenetico, Libertad de Terrenos, etc. Nos estaremos refiriendo a estas secciones como “Capas”.

La información de estas capas estaba almacenada en una base de datos SQL Server y la primera pregunta que surgió fue: ¿Cómo obtengo esta información y la importo en PostgreSQL? En verdad era un problema ya que la estructura de las tablas en SQL Server no era compatibles con PostgreSQL y aquí fue donde QGIS entro a ayudar. Con QGIS me conecte directo a la base de datos SQL SERVER, navegue por las tablas geográficas y las exporte a modo Shapefile, una vez obtenido este archivo lo siguiente que tenía que hacer era importar estos Shapefiles en PostgreSQL con la ayuda de la extensión PostGIS. Al final tenía la misma información geográfica que tenía en SQL SERVER ahora en PostgreSQL.

GeoServer tiene una herramienta para poder cargar capas mediante PostGIS, lo siguiente fue conectarme a PostgreSQL y realizar una conexión a todas las tablas geográficas para su manejo y visualización. Una vez hecho esto era necesario crear estilos para estas capas, por ejemplo poder diferenciar elementos químicos en la capa del mapa Metalogenetico mediante colores. QGIS cuenta con una herramienta para poder lograr esto, el Shapefile

previamente adquirido con esta misma herramienta se cargaba en QGIS y se daba estilo a mediante filtros gracias a que podíamos ver la información de cada uno de los atributos del Shapefile. Finalmente se generaba un archivo con extensión .sld (Styled Layer Descriptor) que contenía el estilo para una capa específica. Lo siguiente era cargar estos .sld's en GeoServer para poder visualizarlos en cada una de las capas.

Implementación de QuickMining Library en OpenLayers

3 /Noviembre - 7/Noviembre

Para empezar con el desarrollo de esta actividad tuve que tener cargado en GeoServer todas las capas de QuickMining configuradas y con sus estilos correspondientes ya listas para ser servidas a la aplicación. Se utilizó la librería OpenLayers para la visualización de los mapas de QuickMining Library. OpenLayers está escrito en JavaScript y tiene un amplia documentación además que se encuentra en su versión 3, lo que da la confianza de estabilidad para varias plataformas. Utilizando WMS (Web Map Service) que ofrece GeoServer se implementó una herramienta de gestión en JavaScript con la ayuda de OpenLayers para poder visualizarla y manipularla en el mapa.

Implementación de Búsqueda (Poligonal, normal e identificación)

10/Noviembre - 28/Noviembre

Cada una de las búsquedas que explicaremos a continuación se utilizó un método que ofrece OpenLayers denominado "getFeatureInfo" el cual interactúa con otro servicio de GeoServer llamado WFS (Web Feature Service) en el cual podemos acceder a datos geoespaciales y brindar información al usuario respecto a alguna capa específica.

Búsqueda Poligonal

Para el uso de la búsqueda poligonal OpenLayers ofrece una herramienta de dibujo de polígonos, esta herramienta consiste en dibujar un polígono en el mapa para después buscar datos de interés dentro de esta área enviando las

coordenadas del polígono a GeoServer que después las aplicara dentro de un filtro CQL (Contextual Query Language) en la capa. También se implementó una opción de ingresar los puntos de cada polígono manualmente, esto quiere decir por coordenadas ya sea en formato UTM o Geográficas. Una última opción de búsqueda para esta herramienta es la avanzada en la cual el usuario puede elegir sobre que capas realizar la búsqueda. Esta herramienta tiene los controles necesarios para agregar varios polígonos en el mapa así como eliminarlos

Búsqueda Normal

La implementación de la búsqueda normal consiste en un texto simple que se manda a GeoServer buscando sobre las capas visibles el texto facilitado. Se implementó la opción de búsqueda avanzada en donde el usuario puede elegir sobre que capa desea realizar la búsqueda y también sobre que atributos de esa capa. El texto es enviado a GeoServer y este busca en la capa especificada mediante un filtro CQL (Contextual Query Language) y se devuelve la respuesta a la aplicación mediante JSON para mostrarla al usuario de manera apropiada.

Búsqueda por Identificación

La búsqueda por identificación consiste en realizar un clic sobre la capa que este visible en el mapa, los datos sobre el área donde se realizó el clic serán enviados a GeoServer devolviendo la información de dicha capa o capas en caso de que existan más de una capa visible a la vez y estas estén encimadas.

Implementación de Herramientas (Regla, Go-to Point, Paint y Add Note)

17/Noviembre - 28/Noviembre

Regla

Esta herramienta fue desarrollada con el trazo de líneas que ofrece OpenLayers, la línea dibujada muestra su longitud ya sea en metros o

kilómetros con la opción de poder eliminarla o agregar cuantas el usuario desee. Una vez dibujada la línea se le agregaba el evento de selección de OpenLayers para que cuando esta fuera seleccionada pudiera ser editada o eliminada, también se le agrego una etiqueta por medio de jQuery donde se mostraba la distancia ya sea en kilómetros o metros.

Go-to Point

Go-to Point está basado en la implementación del ingreso de coordenadas manual de la búsqueda poligonal, consiste en ingresar las coordenadas deseadas y la aplicación navegara hasta dicho punto. Se utilizó la herramienta modal de Ionic framework en la cual ingresábamos las coordenadas, después estas coordenadas fueron pasadas al método de OpenLayers setCenter el cual posiciona el mapa en los argumentos dados.

Paint

Al principio del desarrollo de esta herramienta se intentó realizarla con Canvas HTML5 propio pero por razones técnicas se prefirió hacerlo con interacción de trazos que ofrece la librería OpenLayers en el mapa, OpenLayers cuenta con la posibilidad de poder dibujar puntos, líneas y polígonos, con la ayuda del trazo de líneas se pudo implementar una simulación de pincel sobre el mapa. Con opciones de elegir colores y eliminar todos los trazos.

Add Note

Add-note consiste en hacer clic en cualquier parte del mapa y a partir de este evento se abrirá una ventana para ingresar la nota deseada. Una vez terminada la nota se hace otro clic fuera del cuadro de texto y la nota estará en el mapa como un marcador/pin exactamente donde se hizo el clic. Cuando esta herramienta se utiliza en un dispositivo móvil se implementó la acción que muestra el teclado para poder ingresar el texto en el cuadro gracias al plugin de teclado que ofrece Apache Cordova. Cada uno de las notas agregadas al mapa son manejadas con jQuery y la extensión de Cordova para el teclado. Cuando

se hace clic sobre el campo de texto la extensión de Cordova se encarga de mostrar el teclado en caso que la aplicación se ejecute en un dispositivo móvil. El papel de jQuery aquí es la del manejo de eventos para la eliminación de estas notas haciendo clic sobre el botón de cerrar del cuadro de nota.

Implementación del control de resultados de búsquedas

1 /Diciembre - 5/Diciembre

Cuando se realiza una búsqueda (identificación, normal, poligonal) los resultados que arroje dicha búsqueda son presentados en una tabla. Se implementó una tabla para mostrar los resultados, si los resultados provienen más de una capa, una lista de selección estará disponible para navegar entre los resultados de diferentes capas. Al hacer clic sobre una fila de resultados en el mapa se resalta el área que abarca el elemento, también se cuenta con la opción de que la aplicación navegue en el mapa hasta el elemento seleccionado, esta última opción gracias a OpenLayers en la cual podemos posicionar la pantalla con cierto zoom y encaje en un polígono proporcionado.

Para la realización de esta actividad fue necesario el uso de dibujo que ofrece OpenLayers en la cual se puede dibujar en el mapa desde polígonos hasta simples líneas, cada elemento dibujado es seleccionable y por medio de este se obtiene sus coordenadas para después enviarlas a GeoServer y procesarlas para el regreso de información. Para la búsqueda normal de texto se utilizó una simple petición a GeoServer por medio del método ofrecido por OpenLayers llamado GetFeatureInfo.

Implementación de la simbología de cada una de las capas de QuickMining Library

8/Diciembre - 12/Diciembre

Cuando se muestra una capa de en el mapa automáticamente se muestra un cuadro en la parte superior derecha de la aplicación, este cuadro muestra información de simbología respecto a la capa visible. En el cuadro de simbología solo aparecen listado las capas visibles con un dropdown en caso

de que la capa visible contenga subcategorías. Si ninguna de las capas esta visible entonces el cuadro de simbología tampoco es visible, la herramienta que se utilizó para esta actividad fue las directiva ng-hide y ng-show de AngularJS.

Implementación del control de visualización de las capas de QuickMining Library

8/Diciembre - 12/Diciembre

Se implementó un cuadro de navegación izquierdo donde están listadas todas las capas de QuickMining Library, en este listado se tiene control para activar y hacer visibles en el mapa cada una de las capas, también se implementó un control de transparencia donde el usuario puede seleccionar el porcentaje de transparencia que desea para la capa. Para las capas que cuentan con filtros/subsecciones se programó un dropdown para poder elegir cual filtro/subsección se desea hacer visible. Estas implementaciones se realizaron gracias a las diferentes herramientas que nos presenta Ionic framework y AngularJS la cual permite cargar una lista de elementos en un arreglo y ordenarlos con las funciones que proporciona, este arreglo esta enlazado con la vista así que cualquier cambio que se haga ya sea en el arreglo se ve reflejado inmediatamente en el HTML.

Una última herramienta de control que se implemento fue la ordenación de capas, consiste en poder ordenar las capas para que estas se muestren en el mapa en el orden que se listen en el cuadro de navegación izquierdo todo esto gracias a las diferentes opciones con las que cuenta la herramienta “ion-list” de Ionic.

Implementar autenticación de TEKNOLID para el consumo de QuickMining Library

15/Diciembre - 19/Diciembre

Para obtener el acceso del usuario con cuenta en Teknol se hizo mediante **Resource Owner Password Credentials Grant**. Para obtener los permisos con los que cuenta cada usuario fue necesario interactuar con la biblioteca

SignalR, esta librería establece una conexión persistente con la aplicación. Una vez iniciada la sesión de cada usuario, se verifican las áreas (Layers) que están permitidas para el usuario y se autoriza/deniega su lectura. Cada vez es más requerido que se mejoren las técnicas con las que se autentica un usuario esto con el fin de mejorar la seguridad y proteger la información del usuario y la empresa. Por eso el uso de tokens acompañado con el protocolo HTTPS es una de las opciones más segura actuales para la autenticación. Los recursos del sistema se encuentran en un servidor diferente que donde se realiza la autenticación. Esta separación es recomendable con este tipo de acceso. Como los recursos están en un servidor distinto surge el problema del cross-domain y con la ayuda de cookies evitamos ataques que sufren este tipo de aplicaciones cross-domain.

Dado el tipo de sistemas que es GeoMining, cuando el usuario inicie sesión ninguna otra persona podrá iniciar sesión con la misma cuenta mientras esta esté en línea. Esto con la finalidad del mal uso de las cuentas y obligar al cliente a adquirir una cuenta nueva si así lo desea.

5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA EN LA REALIZACION DE LAS PRACTICAS PROFESIONALES

5.1 Análisis general del programa, su diseño, desarrollo y organización: Durante mi estancia en Teknol me pude dar cuenta cómo es que se (Teknol, Teknol, n.d.) (Teknol, Teknol, n.d.)trabaja en el ámbito laboral, si bien es cierto que no es una empresa grande, es lo más cerca que he estado hasta hoy del desarrollo de software profesional, ya que lo que realice era para consumo de clientes importantes que pagan una buena suma de dinero por el servicio ofrecido por la aplicación. La empresa tiene una buena organización, cuentan por el personal capacitado para lleva a cabo sus tareas y hacer que progrese el desarrollo de la misma. Claro que como toda cosa puede mejorar teniendo una mejor capacitación, tienen sus altas y bajas pero en general fue una muy buena empresa para realizar mis prácticas profesionales porque además de todo lo mencionado anteriormente el software que implemente no era

el típico punto de ventas, inventario, etc. Si no que era algo más atractivo donde tenía que usar diferentes herramientas que no se usan en los típicos programas requeridos en el mercado.

5.2 Análisis de los objetivos del programa:

Cuando fui a la empresa en busca de un lugar para poder realizar mis prácticas profesionales se me dieron instrucciones precisas de lo que debía hacer y para cuando estaba contemplado estar listo. Acordamos el horario y las herramientas que usaría para el desarrollo del proyecto, tenía a mi disposición de personal capacitado que me ayudaría en cualquiera de mis dudas. Todos los días debería de actualizar el repositorio de los avances del día o al menos comentar a las personas encargadas del trabajo realizado en el día. Hubo días que fueron muy productivos y otros no tantos pero todos los días aprendía algo nuevo. Para la última semana que se tenía contemplado se terminaría el proyecto nos dimos cuenta que tendríamos que alargarlo un par de semanas más debido a algunos cambios que se ajustaron al sistema. Para la semana final mi tarea en teknoI estaba concluida exitosamente.

5.3 Análisis de las actividades realizadas:

Cada una de las actividades que realice en teknoI fueron un reto para mí, si bien tenía un poco de experiencia por trabajos de freelance que había hecho durante mi carrera como estudiante había bastedad de cosas que no sabía. Me sentí muy complacido con mi trabajo y sé que los directivos de la empresa también ya que me lo hicieron saber a tal grado de ofrecerme un puesto definido para poder seguir trabajando con el proyecto de desarrollo en el que estuve a cargo. Este proyecto fue algo nuevo para mí, conocía algunas de las herramientas que utilice, otras solo de nombre y otras definitivamente era la primera vez que sabía de ellas pero no por eso eran malas o famosas lo que pasaba era que no me había involucrado en el entorno adecuado para poder aprender de todas estas tecnologías. Como había mencionado al principio de este documento la empresa buscaba migrar su plataforma al código abierto y para esto tenía que aprender a usar estas nuevas herramientas (al

menos para mí lo eran). Herramientas como AngularJS, Apache Cordova y GeoServer que fueron las principales involucradas en el desarrollo de la aplicación, había escuchado hablar de ellas mas no me había visto comprometido en algún proyecto donde las usara y para este proyecto lo hice, me comprometí a aprender a usarlas y las aprendí. Con mi trabajo en teknoI concluido puedo decir que estas herramientas fueron una maravilla, me ayudaron bastante, agilizaron notablemente el desarrollo ya que sin ellas probablemente habría tardado el triple de tiempo en hacer cada una de las actividades. Cada una de las herramientas usadas fueron una maravilla y sin duda alguna las recomiendo. Pienso que debería existir alguna clase especial donde se hable y practique solo del uso de nuevas tecnologías de desarrollo que se están usando actualmente y de esta manera tendremos al menos la mínima idea lo que nos espera en el mundo del desarrollo de software en las empresas.

5.4 Análisis de la metodología utilizada: Como mencionamos en la introducción la metodología utilizada si bien no fue en su totalidad una en específica nos orientamos mucho en Lean Software Development, Este método ofrece todo un marco teórico sólido y basado en la experiencia, para las prácticas ágiles de gestión. Dado que lo que se haría sería remplazar el sistema actual con uno nuevo, el cual debía de ser más rápido, eficiente y con nuevas funciones. Al principio no se pensó en que la metodología seria esta pero la manera en que surgieron las cosas nos obligaron a tomar este camino. Era necesario eliminar los desperdicios, excluir funciones innecesarias y como el tiempo era limitado necesitábamos de un desarrollo ágil y que respondiera con nuestras necesidades.

6. CONCLUSIONES Y RECOMENDACIONES

A lo largo del estudio de la carrera el estudiante va aprendiendo cada vez más y nunca dejamos de hacerlo, pero llega un momento determinado que con las

prácticas y ejercicios que realizas en el salón de clase no es suficiente y empiezas a sentirte curioso y te preguntas como sería aplicar los conocimientos adquiridos en el “mundo real”. Llega la duda de cómo lo harán los profesionales y algunos alumnos cuando todavía no se encuentran en un semestre avanzado se “lanzan” al mundo a experimentar ya que la escuela no es suficiente, quieren saber más, sienten ese deseo, esa curiosidad de aprender más y hacer las cosas como se supone se deben hacer. Al principio cometemos muchos errores pero así es como mejor se aprende. Las prácticas profesionales que son requisito en la universidad son de vital importancia y si uno se “lanza” como lo había mencionado antes, será mucho mejor porque ya llevaremos esa experiencia que te permitirá desenvolverte con más naturaleza y la entrada de aprendizaje será más amplia para el practicante. En lo personal en el desarrollo de la aplicación GeoMining Touch aprendí muchísimo, siempre he tenido en mente que nunca dejare de aprender, y cuando aprendo algo nuevo, eso nuevo que aprendí me abre dos caminos más que puedo tomar para descubrir nuevas cosas. Quede muy satisfecho con el framework AngularJS y la idea de que Ionic lo utilizara fue genial y una ventaja que atraerá a muchos desarrolladores a usarlo. Por otra parte tenía un poco de experiencia en el área de GIS (Geographical Information System) y la integración de GeoServer/OpenLayers/PostgreSQL-PostGIS fue magnífica y lo mejor en el área con mucha documentación y el mejor rendimiento. Aprendí mucho de estas herramientas mencionadas y sé que me servirán en un futuro para proyectos venideros. Creo que es muy importante realizar este tipo de trabajo en una empresa ya que te das cuenta como se trabaja en el mercado, creas relaciones con personas que piensan similar a ti y te ayuda a crecer profesionalmente.

7. REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES

Referencias

Apache. (s.f.). *Apache Cordova*. Obtenido de <http://tech.batanga.com/13241/que-es-apache-cordova>

Drifty. (s.f.). *Ionic framework*. Obtenido de <http://ionicframework.com/docs/guide/>

GeoServer. (s.f.). *GeoServer*. Obtenido de <http://docs.geoserver.org/>

Google. (s.f.). *AngularJS*. Obtenido de <https://angularjs.org/>

OpenLayers. (s.f.). *OpenLayers*. Obtenido de <http://openlayers.org/>

PostGIS. (s.f.). *PostGIS*. Obtenido de <http://postgis.net/>

PostgreSQL. (s.f.). *PostgreSQL*. Obtenido de <http://www.postgresql.org/>

QGIS. (s.f.). *QGIS*. Obtenido de <http://www.qgis.org/es/docs/index.html>

Stefano Lacovella, B. Y. (2013). *GeoServer*. PACKT Publishing.

Teknol. (s.f.). *Geomining Touch*. Obtenido de <http://teknol.net/soporte/install-geomining-touch/>

Teknol. (s.f.). *Quikmining Library*. Obtenido de <http://teknol.net/our-solutions/quickmining-library/>

Teknol. (s.f.). *Teknol*. Obtenido de <http://teknol.net>