



**Decanato de Ingeniería e Informática
Escuela de Informática**

**DISEÑO DE SISTEMA DE APOYO A LA TOMA DE
DECISIONES CON RESPECTO A LA COMPRA, VENTA
Y RENTA DE INMUEBLES ABC POR WEB Y MÓVIL
UTILIZANDO SISTEMAS DE INFORMACIÓN GEOGRÁFICA**

Sustentantes:

Andrés Estepan	Mat. 2008-0116
David Díaz	Mat. 2008-0136
Carmen P. Ramírez	Mat. 2008-0822

Asesor:

Mitsuteru Nishio

Monografía para Optar por los Título de
Ingeniería en Sistemas de Información
Ingeniería en Sistemas de Computación

Distrito Nacional, República Dominicana
Agosto, 2014

**DISEÑO DE SISTEMA DE APOYO A LA TOMA DE
DECISIONES CON RESPECTO A LA COMPRA, VENTA
Y RENTA DE INMUEBLES ABC POR WEB Y MÓVIL
UTILIZANDO SISTEMAS DE INFORMACIÓN GEOGRÁFICA**

Dedicatoria

Es mi deseo dedicarle mi Trabajo de Grado en el presente Informe, en primera instancia a mi familia quienes permanentemente me apoyaron, contribuyendo incondicionalmente a lograr las metas y objetivos propuestos.

A los docentes que me han acompañado durante el largo camino, brindándome siempre su orientación con profesionalismo en el transcurso de mis estudios y afianzando mi formación como estudiante universitario.

Andres Estepan Ulloa

Dedicatoria

Le dedico este trabajo a mi padre **Esteban Rafael Díaz Jáquez** por apoyarme incondicionalmente y por priorizar mi educación y ayudarme a lograr mis metas.

Le dedico este trabajo también a mi madre **Ana Lissette Del Pozo Roquel**, por su énfasis en la educación y por que sin el esfuerzo de su vida yo no tendría hoy las oportunidades que tengo.

David Octavio Díaz Del Pozo

Dedicatoria

Primeramente a Dios quien supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban, enseñándome a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento.

A mi familia: Con todo mi cariño y amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se agotaba, en especial a mis padres: **Sandra Peña y Santiago Ramírez** por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles y por ayudarme con los recursos necesarios Gracias a ustedes he logrado llegar hasta aquí, Los amo!

A mis Amigos: A lo largo de mi camino universitario me demostraron que su apoyo hacia mí era sincero y sin buscar nada a cambio, lo cual estoy muy agradecida porque siempre hay malos momentos, ahí es donde se necesitan compañeros de buena fe, y yo le doy gracias a Dios por haberme dado a esos compañeros, porque sin su apoyo y su comprensión nada de esto hubiera sido posible, ahora terminamos todos una etapa más de la vida, pero iniciamos la siguiente con una sonrisa y muchas expectativas de que nos traiga mucha felicidad y espero que todos logremos nuestros objetivos profesionales, así como logramos los objetivos universitarios.

Gracias por formar parte importante en mi vida, por estar siempre presente y brindarme todo su apoyo **Jhacdira Liriano, Leiko Acosta y Daisbel Leon,**

Edgar Polanco y a todos aquellos que aun no estén plasmado en este papel siguen formando parte de todo esto.

A mis Maestros: Que en este andar por la vida, influyeron con sus lecciones y experiencias en formarme como un a persona de bien y preparada para los retos que vendrán en la vida.

Carmen Patricia Ramírez

Agradecimientos

Con todo mi cariño para la persona que hizo todo en la vida para que yo pudiera lograr mis sueños a mi madre **Magalys Pichardo** , por motivarme y darme la mano, a usted por siempre mi corazón y mi agradecimiento.

A mi tía **Epifania Guzman** quien sin su apoyo este sueño no hubiera sido posible.

A mis amigos quienes me apoyaron en todo el trayecto de mis estudios.

Andres Estepan Ulloa

Agradecimientos

Quiero agradecer a mi padre **Esteban Rafael Díaz Jáquez** por su apoyo y por siempre estar ahí.

Quiero darles las gracias a mi familia y a mis amigos.

Le agradezco a la Universidad APEC por la educación que me ha provisto y a toda la gente que la hacen posible y a los profesores, especialmente los fuertes a los que los estudiantes le ponen apodo, porque al empujarme he aprendido más de lo que hubiera aprendido sin la pelea.

David O. Díaz Del Pozo

Tabla de Contenidos

1	CICLO DE VIDA DE LOS SISTEMAS DE INFORMACIÓN	16
1.1	DESCRIPCIÓN	17
1.2	CARACTERÍSTICAS.....	18
1.3	LAS FUNCIONES PRINCIPALES DE UN CICLO DE VIDA SOFTWARE SON:	19
1.4	ANTECEDENTES.....	20
1.5	CLASIFICACIÓN	23
1.6	MODELO DE CASCADA	23
1.7	MODELO EN V.....	26
1.8	PROTOTIPADO	28
1.9	DESARROLLO RÁPIDO DE APLICACIONES	32
1.10	MODELO EN ESPIRAL	34
1.11	VENTAJAS Y DESVENTAJAS.....	37
1.12	MODELO “CODIFICAR Y CORREGIR”	37
	<i>Ventajas:</i>	37
	<i>Desventajas:</i>	38
1.13	MODELO EN “CASCADA”	38
	<i>Ventajas:</i>	38
	<i>Desventajas:</i>	39
1.14	MODELO EN “V”	39
	<i>Ventajas:</i>	39
	<i>Desventajas:</i>	40
1.15	“PROTOTIPOS”	40
	<i>Ventajas:</i>	40
	<i>Desventajas:</i>	41
1.16	“DESARROLLO RÁPIDO DE APLICACIONES”	41
	<i>Ventajas:</i>	41
	<i>Desventajas:</i>	41
1.17	MODELO EN “ESPIRAL”	42
	<i>Ventajas:</i>	42
	<i>Desventajas:</i>	42
1.18	¿COMO ELEGIR?	43
2	BASE DE DATOS	47
2.1	CONCEPTOS	47
2.1.1	<i>Datos</i>	47
2.1.2	<i>Base de Datos</i>	47
2.1.3	<i>Base de datos relacionales</i>	48
2.1.4	<i>Lenguaje de consulta estructurado (SQL)</i>	49
2.1.5	<i>Base de datos espaciales</i>	50
2.1.6	<i>Sistema de Gestión de Base de Datos (SGBD):</i>	50
2.1.7	<i>Sistema de Información geográfica</i>	51
2.1.8	<i>Business Inteligente</i>	52
2.2	CARACTERÍSTICAS.....	53
2.3	ORACLE VS MYSQL VS SQL SERVER: COMPARACIÓN DE SGBDR MÁS POPULARES.....	54
2.3.1	<i>Sumario Comparativo de Características</i>	54
2.3.2	<i>Oracle</i>	55
2.3.3	<i>SQL Server</i>	57
2.3.4	<i>MySQL</i>	58
2.4	BIGTABLE	60

2.5	GOOGLE EARTH	61
3	DESARROLLO WEB	63
3.1	HTML5, CSS3 Y JAVASCRIPT	64
3.2	CSS3 STYLING	68
4	DESARROLLO MÓVIL	70
4.1	TECNOLOGÍA MÓVIL EN LA REPUBLICA DOMINICANA	73
5	SISTEMAS DE INFORMACIÓN GEOGRÁFICA	75
5.1	CONCEPTOS	75
5.1.1	<i>Cartografía</i>	75
5.1.2	<i>Mapa</i>	75
5.1.3	<i>Sistemas de información geográfica</i>	75
5.1.4	<i>Servicios basados en Localización</i>	76
5.1.5	<i>Servicios de Orientación o Navegación</i>	76
5.1.6	<i>Tecnologías de Localización</i>	78
5.1.7	<i>GPS</i>	78
5.1.8	<i>Telefonía celular</i>	79
5.1.9	<i>WiFi</i>	80
5.1.10	<i>RFID</i>	81
5.1.11	<i>Beneficios de los Sistemas de Información Geográfica</i>	81
5.1.12	<i>Los beneficios de SIG generalmente se dividen en cinco categorías básicas:</i> 82	
5.1.13	<i>¿Qué se puede hacer con los SIG?</i>	83
5.2	SISTEMA DE INFORMACIÓN GEOGRÁFICA ESCOGIDA: GOOGLE MAPS	85
5.2.1	<i>Google Maps</i>	86
5.2.2	<i>Entre las cosas que se pueden hacer con Google Maps están</i>	87
5.2.3	<i>Diferentes APIs</i>	89
5.2.4	<i>Cobertura</i>	89
5.2.5	<i>Uso Comercial</i>	90
5.2.6	<i>Límites de uso</i>	91
5.2.7	<i>Costos</i>	92
5.3	GOOGLE MAPS FOR BUSINESS	93
5.3.1	<i>Determinación del uso de la aplicación</i>	95
5.3.2	<i>Google Analytics para el API de Google Maps for Business</i>	95
5.3.3	<i>Asistencia</i>	96
5.3.4	<i>Límites de uso</i>	98
5.4	GOOGLE MAPS ENGINE	101
5.4.1	<i>Visualiza tus datos geoespaciales en la nube</i>	101
5.4.2	<i>Cree y comparta mapas de Google Maps personalizados a escala</i>	102
5.4.3	<i>Capacidades de la plataforma de Google Maps Engine</i>	102
5.5	GOOGLE MAPS ENGINE PRO	103
5.6	GOOGLE MAPS PLACES	104
5.6.1	<i>Búsquedas</i>	105
5.6.2	<i>Eventos</i>	105
5.6.3	<i>API de autocompletado de Google Places</i>	106
5.6.4	<i>API de Google Places</i>	106
5.6.5	<i>Descripción general</i>	106
5.6.6	<i>Tipos de sitios admitidos</i>	106
5.6.7	<i>Tipos admitidos para buscar y añadir sitios</i>	107
5.6.8	<i>Tipos adicionales utilizados por el servicio de Google Places</i>	108
5.6.9	<i>Conecta personas y lugares</i>	108

5.6.10	<i>Trabaja con una potente base de datos</i>	109
5.6.11	<i>A continuación se indican los tipos de solicitudes de sitios disponibles.</i>	109
5.6.12	<i>Políticas para desarrolladores</i>	110
5.6.13	<i>Requisitos del logotipo</i>	110
5.6.14	<i>Límites de uso</i>	111
5.6.15	<i>Requisitos de las Condiciones de uso y la Política de privacidad</i>	112
5.6.16	<i>Aplicaciones para móviles</i>	112
5.6.17	<i>Aplicaciones web</i>	113
6	PROPUESTA DE DISEÑO DE LA APLICACIÓN	114
6.1	CONSIDERACIONES	114
7	DISEÑO DE LA APLICACIÓN WEB Y MÓVIL	117
7.1	BÚSQUEDA EN WEB	117
7.2	AGREGAR UN INMUEBLE EN WEB	119
7.3	BÚSQUEDA EN MÓVIL	121
7.4	AGREGAR UN INMUEBLE EN MÓVIL	123
7.5	ALERTA	125

Tablas

Tabla 2-I.....	54
Tabla 5-I.....	93
Tabla 5-II	98
Tabla 5-III	99
Tabla 5-IV	100
Tabla 5-V	101
Tabla 5-VI.....	111
Tabla 6-I.....	114

Ilustraciones

Figura 1-1.....	24
Figura 1-2.....	25
Figura 1-3.....	27
Figura 1-4.....	34
Figura 1-5.....	37
Figura 4-1	71
Figura 4-2.....	72
Figura 4-3.....	73
Figura 5-1	96
Figura 7-1.....	117
Figura 7-2.....	119
Figura 7-3.....	121
Figura 7-4.....	123
Figura 7-5.....	125

Resumen

Nuestro objetivo principal en este trabajo es proponer una solución para un sistema de soporte de decisiones para la compra, venta y renta de inmuebles para los clientes de Inmuebles ABC, una página de clasificados de casas, apartamentos y terrenos, por medio web y móvil apoyándonos de las tecnologías de sistemas de información geográfica.

En esta monografía se exploran diferentes tecnologías que se utilizan hoy en día en desarrollo de software, base de datos, Sistemas de Información Geográfica, etc. y luego se sugieren cual de todas las tecnologías es la más recomendada en base a lo que ha hecho y potencial de lo que hará en futuro.

Al final de este trabajo se habrá mostrado todos los elementos necesarios para poder armar una solución web y móvil, que sea eficaz, fácil de mantener, extensible, moderna y que se adaptara a las nuevas tendencias.

Introducción

La empresa Inmuebles ABC quiere mejorar sus servicios de clasificados de bienes raíces en la que se compran, venden y rentan inmuebles y en este trabajo se propone una solución a esta querrela. Hasta ahora los usuarios solo describen las características del inmueble y no saben del valor agregado dada las utilidades cercanas a este. Otro problema es que el usuario tiene que entrar al sitio web de vez en cuando si un inmueble con las características que quiere no está disponible, no existe la manera en que Inmueble ABC le avise al usuario que un inmueble que le gustaría esta o no disponible.

Con la solución que proponemos el usuario final será consiente que no solo compra, vende o alquila un inmueble, sino que realmente negocia en accesibilidad a otras utilidades a parte del inmueble como los son hospitales, escuelas, departamentos de policía, supermercados, gasolineras, etc.

Llegamos a esta solución gracias al análisis de la situación y a las tecnologías existentes de base de datos, sistemas de información geográfica, desarrollo web y desarrollo móvil. Por ello los primeros capítulos se expondrán dichos tópicos.

Hoy en día con mas personas en nuestro país conectadas al internet y con el auge en las tecnologías móviles y Android estando en 53.73% de todos los equipos en la patria, de acuerdo a StatCounter en Junio de 2014, la solución

propuesta ayudara a Inmuebles ABC aprovecharse del nuevo nicho de las Apps.

Descripción General

Empresa

Inmuebles ABC es una empresa relativamente nueva de clasificados de bienes raíces que funciona como medio para poner en contactos, compradores y vendedores de casas, apartamentos, terrenos, etc.

Es una empresa con líderes jóvenes que quieren estar a la vanguardia del mercado en que compiten y esperan que una nueva web y una aplicación móvil le den ese empuje y originalidad que buscan

Misión

Proporcionar un entorno de intercambio para los compradores, vendedores y alquiladores de inmuebles en la República Dominicana con información actualizada y amplia acerca del estado de bienes raíces del país.

Visión

Ser líderes como medio de intercambio en mercado de bienes raíces de la República Dominicana y ser la primera opción para todos en clasificados de inmuebles.

Valores

- Confianza
- Calidad
- Efectividad
- Confidencialidad
- Compromiso

Alcance

El alcance de este trabajo es el diseño de la aplicación web y móvil de esta solución de software sin comprometerse a ninguna implementación en particular sino describir todos los elementos necesarios para su implementación y tecnologías alternativas que se pueden utilizar y luego litigar haciendo recomendaciones de que es lo mejor para el producto final como.

Se expondrá acerca de ciclos de vida de sistemas de información, base de datos, diseño web, diseño móvil, sistemas de información geográfica y muchos temas relacionados.

Al final se muestran pantallas prototipo de como se vería una aplicación final realizadas en una aplicación de wireframing.

1 Ciclo de vida de los sistemas de información

El ciclo de vida de los sistemas es el proceso evolutivo que se sigue al implementar un sistema o subsistema de información basado en computadora. El cual consiste en una serie de tareas que siguen de cerca los pasos del enfoque de sistema. Puesto que las tareas siguen un patrón ordenado y se llevan a cabo en forma descendente (McLeod, 2000). (Conde, Nava, & Rojas, 2008)

Las primera cuatro fases son las de planificación, análisis, diseño e implementación. En conjunto esas fases constituyen el ciclo de vida del desarrollo de sistemas. Y por último la quinta fase es la fase de uso, que dura hasta que llega el momento de rediseñar el sistema. El rediseño requiere la repetición del ciclo completo.

En la planificación es donde se definen los objetivos generales y específicos, y la mejor forma de alcanzarlos.

Mientras que el análisis conlleva el estudio del sistema actual en caso que existiera, las necesidades que se desea cubrir y priorizarla.

En cuanto al diseño, como su nombre lo dice es donde se diseña la solución para cubrir las necesidades anteriormente encontradas y satisfacer los requisitos de los mismos.

Finalmente la implementación es el proceso en donde se pone en marcha la solución o software realizado. (Conde, Nava, & Rojas, 2008)

1.1 Descripción

(Conde, Nava, & Rojas, 2008) Nos continúa diciendo que ciclo de vida se refiere al período de tiempo que comienza cuando se concibe la idea de generar el programa hasta que finalmente se retira.

- Se denomina al conjunto de fases como ser: Todo proyecto de ingeniería tiene unos fines ligados a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto.
- Facilita el control sobre los tiempos en que es necesario aplicar recursos de todo tipo (personal, equipos, suministros, etc.) al proyecto. Si el proyecto incluye subcontratación de partes a otras organizaciones, el control del trabajo subcontratado se facilita en la medida en que esas partes encajen bien en la estructura de las fases. El control de calidad también se ve facilitado si la separación entre fases se hace corresponder con puntos en los que ésta deba verificarse (mediante comprobaciones sobre los productos parciales obtenidos).
- Es un conjunto ordenado de tareas como un proceso: una serie de pasos que incluye actividades, restricciones y recursos que producen un resultado de un determinado tipo.

1.2 Características

El proceso describe todas las actividades principales.

- El proceso utiliza recursos que están sujetos a una serie de restricciones y genera productos intermedios y finales.
- El proceso puede descomponerse en subprocesos que están enlazados de una forma determinada e, incluso, se puede definir como una jerarquía de procesos, organizados de forma que cada subproceso tenga su propio modelo de proceso.
- Cada actividad del proceso tiene definidos criterios de entrada y salida, de forma que queda claramente especificado cuándo empieza y termina cada actividad.
- Las actividades de un proceso están organizadas en forma de secuencia, de forma que puede saberse cuándo se realiza una actividad en relación con el resto.
- Todo proceso incorpora un conjunto de principios básicos que explican los objetivos de cada actividad.
- Pueden existir restricciones o controles para las actividades, recursos o productos. Así, por ejemplo, el presupuesto o la planificación pueden restringir la cantidad de tiempo que se puede dedicar a una actividad o, por otro lado, una herramienta puede limitar la forma en la que se utiliza un recurso.

El ciclo de vida software da respuesta a las siguientes preguntas de la gestión de un proyecto de software:

¿Qué haré a continuación?

¿Cuánto tiempo continuaré haciéndolo?

El ciclo de vida que se seleccione en un proyecto influirá en el éxito del proyecto, y puede ayudar a asegurar que cada paso que se dé acorte más la consecución del objetivo.

Dependiendo del ciclo de vida que se seleccione, se puede aumentar la velocidad de desarrollo, mejorar la calidad, el control y el seguimiento del proyecto, minimizar gastos y riesgos, o mejorar las relaciones con los clientes. Una selección ineficaz puede ser una fuente constante de ralentización del trabajo, trabajo repetitivo, innecesario y frustrante.

1.3 Las funciones principales de un ciclo de vida software

son:

- Determinar el orden de las fases y procesos involucrados en el desarrollo del software y su evolución, teniendo en cuenta el modelo de procesos que se utilice como referencia.
- Establecer los criterios de transición para pasar de una fase a la siguiente. Todo ello, incluye los criterios para la terminación de la fase actual y los criterios para seleccionar e iniciar la fase siguiente.

1.4 Antecedentes

Tradicionalmente el desarrollo de aplicaciones informáticas se llevaba a cabo de forma individualizada, nos dicen(Conde, Nava, & Rojas, 2008), a base de codificar (generar líneas de código) y probar lo realizado cuanto antes.

La misma persona escribía el código, lo ejecutaba y, si fallaba, lo depuraba. El proceso se realizaba sin ninguna planificación previa y sin que soliese existir documentación alguna.

Debido a que la movilidad en el trabajo era baja, los ejecutivos estaban seguros de que esa persona estaría allí cuando se produjese algún fallo.

En principio, el hecho de que desde un primer momento se vaya generando código, podría considerarse como un síntoma de enorme progreso, pero puede suponer posteriormente un gran retroceso e incluso la necesidad de desechar una gran parte de lo realizado en el caso de que existan errores y no se puedan llevar a cabo las modificaciones necesarias para subsanarlos (por ejemplo, si al 90% del código se descubre que el diseño de la base de datos es incorrecto, puede suponer desechar el trabajo y tener que comenzar de nuevo). Con este enfoque, cualquier cosa que no sea codificación pura y dura no se realiza (como, por ejemplo, actividades de planificación, de documentación, de aseguramiento de la calidad).

Esta forma de desarrollar aplicaciones es muy común en muchas organizaciones y, generalmente, se utiliza cuando no se elige o sigue un enfoque de desarrollo (ciclo de vida) concreto y/o apenas se realiza la actividad de planificación.

Además, otro factor que juega a favor de este enfoque de codificar y probar es que requiere poca experiencia y cualquier persona podrá fácilmente familiarizarse con él.

Esta forma de desarrollar software puede ser eficaz en programas pequeños. Para otro tipo de proyectos, puede resultar peligrosa su utilización, ya que no se puede conocer el progreso del proyecto, ni tampoco su calidad, simplemente se está codificando y probando hasta que finaliza el proyecto. Otras maneras de realizar el desarrollo software, como se verán en los siguientes apartados, permitirán, por ejemplo, conocer el progreso, detectar un error lo antes posible, etc.

Por lo tanto, es probable que las aplicaciones realizadas según este enfoque de codificar y probar:

- Sean poco flexibles, y ante posibles modificaciones se incrementa el coste de los proyectos e, incluso, en ocasiones, resulten virtualmente irrealizables debido a la naturaleza personalizada de los programas y a la falta de documentación (lo que provocará problemas de mantenimiento).

- Sean incompletas o no reflejen bien las necesidades del cliente, es decir, que no realicen todas las funciones requeridas y, además, lo hagan con una escasa fiabilidad.
- Provoquen el descontento de los clientes, pues se producen retrasos en la entrega (no se conoce el momento exacto en el que se entregarán), aparecen errores una vez que la aplicación ha sido entregada (lógico al no haberse realizado de forma sistemática actividades de verificación y validación en el proyecto).

Por tanto, es necesario que todo esfuerzo en el desarrollo del software con lleve un enfoque lógico para su realización. Dicho enfoque debe abarcar toda la vida del sistema, comenzando con su concepción y finalizando cuando ya no se utiliza o se retira.

El ciclo de vida software es la descripción de las distintas formas de desarrollo

de un proyecto o aplicación informática, es decir, la orientación que debe seguirse para obtener, a partir de los requerimientos del cliente, sistemas que puedan ser utilizados por dicho cliente. También puede definirse como el conjunto de fases o etapas, procesos y actividades requeridas para ofertar, desarrollar, probar, integrar, explotar y mantener un producto software.

1.5 Clasificación

En la literatura existe una multitud de modelos de ciclo de vida. Algunos son prescripciones sobre la forma en la que debería realizarse el desarrollo de software, mientras que otros son descripciones de la forma en la que se desarrollan programas actualmente. En teoría ambos tipos de modelos deberían ser iguales o similares, pero en la práctica no lo son. La construcción un modelo de ciclo de vida y la discusión sobre sus subprocesos ayuda a comprender este salto entre lo que debería ser y lo que realmente es. (Conde, Nava, & Rojas, 2008)

1.6 Modelo de Cascada

En los años 70 se impuso un nuevo enfoque de desarrollo del software, introducido por Royce en 1970, a través de un ciclo de vida en “cascada” (así denominado por la disposición de las distintas fases de desarrollo, en las que los resultados de una fase parecen caer en cascada hacia la siguiente fase, tal como se muestra en la Figura 1). El método ideado por Royce constituye uno de los primeros modelos de ciclo de vida publicados, por lo que también recibe el nombre de modelo de ciclo de vida clásico. Este método modela el ciclo convencional de la Ingeniería del Software, aplicando un enfoque sistemático y secuencial de desarrollo que comienza con la ingeniería del sistema y progresa a través del análisis, diseño, codificación, pruebas y mantenimiento. (Conde, Nava, & Rojas, 2008)

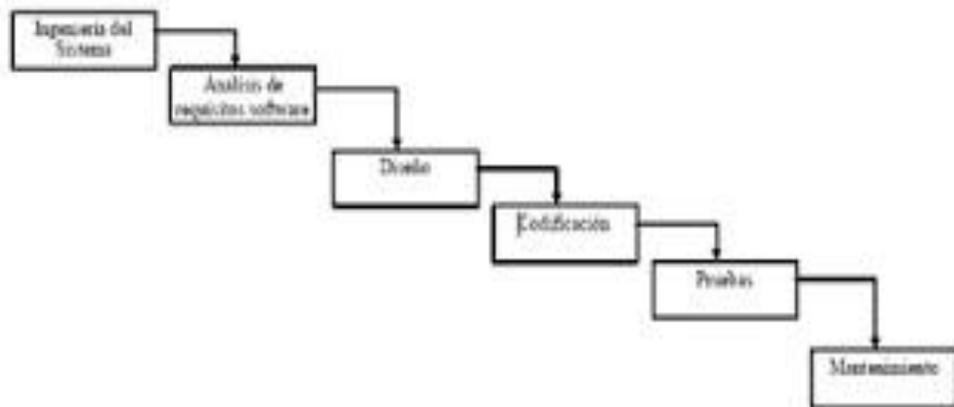


Figura 1-1 Ciclo de vida en cascada o clásico

Como sugiere el esquema del modelo en cascada, antes de poder avanzar a la siguiente etapa, es necesario haber finalizado completamente la etapa anterior. Asociada con cada etapa del proceso existen hitos y documentos, de tal forma que se puede utilizar el modelo para comprobar los avances del proyecto y para estimar cuánto falta para su finalización.

Este modelo es muy útil pues ayuda a los desarrolladores a comprender qué es lo que tienen que hacer en cada momento. Su simplicidad hace que resulte sencillo explicárselo a los clientes que no están familiarizados el proceso software. Además, se muestran de forma explícita qué productos intermedios se tienen que obtener antes de abordar las siguientes tareas.

Una modificación sobre este modelo consiste en la introducción de una revisión y vuelta atrás, con el fin de corregir las deficiencias detectadas durante las distintas etapas, o para completar o aumentar las funcionalidades del sistema en desarrollo, resultando un diagrama de fases y etapas tal como

el que se muestra en la Figura 2. De esta manera, durante cualquiera de las fases se puede retroceder momentáneamente a una fase previa para solucionar los problemas que se pudieran haber encontrado.

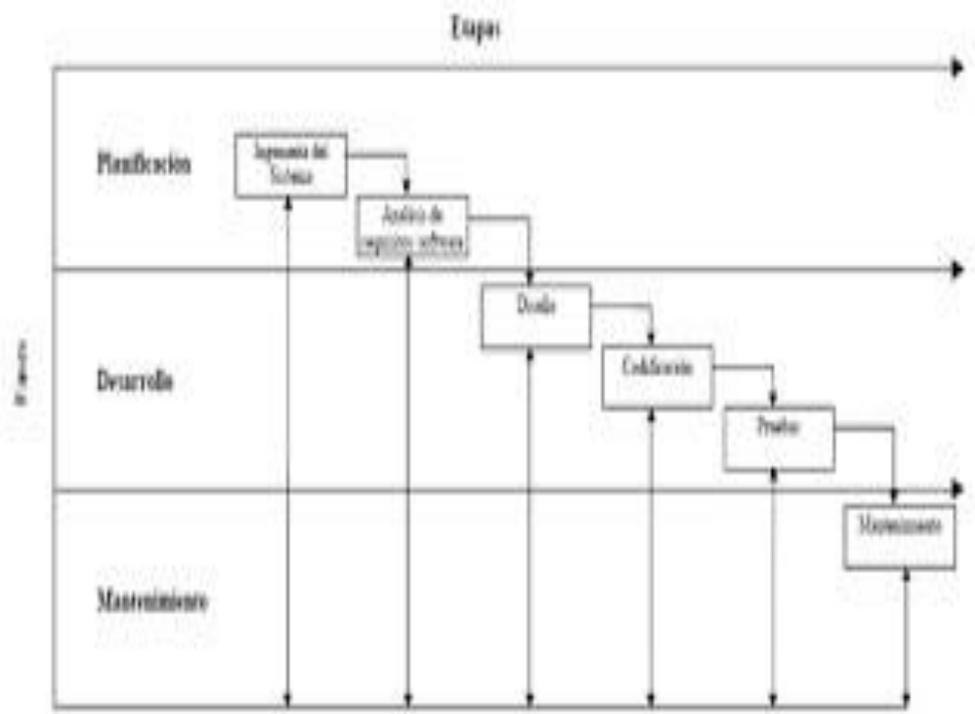


Figura 1-2 Ciclo de vida en cascada o clásico completo

Normalmente, el ciclo de vida del software se suele dividir en tres fases: una de Planificación, otra de Desarrollo y una tercera de Mantenimiento, que engloban a las seis etapas (Ingeniería del Sistema, Análisis de los Requisitos, Diseño, Codificación, Pruebas y Mantenimiento) tradicionales del ciclo de vida.

La fase de Planificación del software comprende las etapas de Ingeniería del Sistema o Análisis del Sistema (en concreto el establecimiento de los Requisitos del Software o “Plan Software”) y el Análisis de los Requisitos del Software (que se traduce en una “Especificación de Requisitos”). La

fase de Desarrollo comprende las etapas de Diseño, Codificación y Pruebas. Por último, la fase de Mantenimiento incorpora solamente la etapa propia de Mantenimiento.

A pesar de su antigüedad, el ciclo de vida clásico se ha hecho con un lugar importante en el área de la Ingeniería del Software. Proporciona una guía de trabajo en la que se encuentran métodos para el análisis, diseño, codificación, pruebas y mantenimiento. El ciclo de vida en cascada sigue siendo el modelo de proceso más extensamente utilizado por los ingenieros del software, principalmente por su sencillez y facilidad de llevar a cabo. Pese a tener debilidades, es significativamente mejor que un enfoque arbitrario (como el de codificar y corregir) para el desarrollo del software. Muchos de los posteriores modelos de ciclo de vida son, en realidad, modificaciones sobre el modelo clásico, al que se le incorporan iteraciones o nuevas actividades.(Conde, Nava, & Rojas, 2008)

1.7 Modelo en V

El modelo en V es una variación del modelo en cascada que muestra cómo se relacionan las actividades de prueba con el análisis y el diseño. Como se muestra en la Figura 3, la codificación forma el vértice de la V, con el análisis y el diseño a la izquierda y las pruebas y el mantenimiento a la derecha.

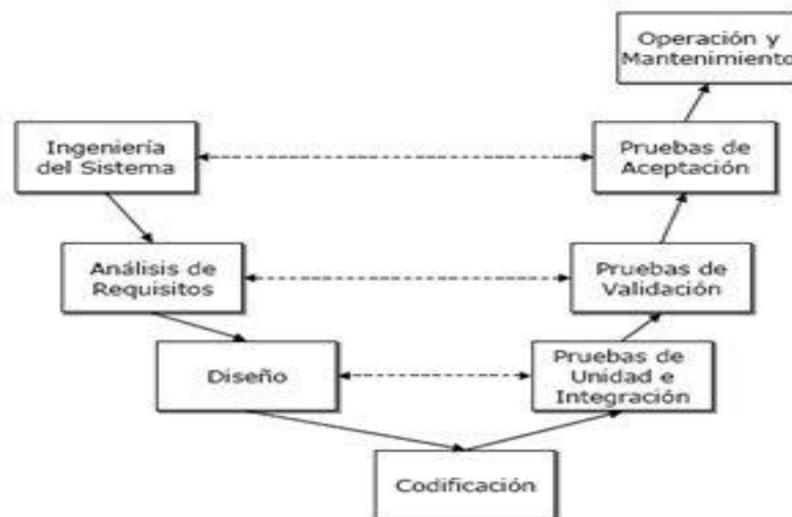


Figura 1-3 Modelo de Ciclo de Vida en V

La unión mediante líneas discontinuas entre las fases de la parte izquierda y las pruebas de la derecha representa una doble información. Por un lado sirve para indicar en qué fase de desarrollo se deben definir las pruebas correspondientes. Por otro sirve para saber a qué fase de desarrollo hay que volver si se encuentran fallos en las pruebas correspondientes.

Por lo tanto el modelo en V hace más explícita parte de las iteraciones y repeticiones de trabajo que están ocultas en el modelo en cascada. Mientras el foco del modelo en cascada se sitúa en los documentos y productos desarrollados, el modelo en V se centra en las actividades y la corrección.

1.8 Prototipado

En contraste con la Ingeniería de Software de la década de los 70, que dio respuesta a proyectos grandes pero con requisitos estables, la Ingeniería de Software de los 80 reaccionó a las complicaciones resultantes de encontrarse con requisitos poco claros y dinámicos, dando lugar a la “construcción de prototipos”. El modelo de ciclo de vida de prototipos fue propuesto por Gooma en 1984. (Conde, Nava, & Rojas, 2008)

Un prototipo es un mecanismo para identificar los requisitos del software. La construcción de prototipos es un proceso que facilita al ingeniero de software el desarrollo de la aplicación. El prototipo suele tomar una de las tres formas siguientes:

- Un modelo en papel o en computadora que describe la interacción hombre-máquina, de forma que facilite al usuario la comprensión de su funcionamiento. Por ejemplo, si el sistema a construir es un cajero automático, se puede hacer un programa que simule la interacción del usuario con el cajero sin que el programa esté conectado a ninguna base de datos real ni se despache dinero. De esta manera el cliente puede hacerse a la idea de cómo va a funcionar el sistema final sin tener que construirlo, y así discutirlo con el ingeniero de software. Naturalmente, en un prototipo no se simularán todas las funcionalidades del sistema pero, si es necesario, se

podrán construir otros a medida que la aplicación se vaya desarrollando (ver más abajo cuáles son las etapas para su utilización)

- Un modelo que implementa una función requerida importante. Es el mismo caso que anteriormente pero sin centrarse en la interacción hombre-máquina. Por ejemplo, el modelo podría simular todos los pasos a seguir internamente en el sistema en el acceso a la base de datos de clientes cuando se quiere obtener dinero del cajero, pero sin que realmente se trate de una base de datos real ni de un cliente del banco.

- Un programa real que se adecue en parte al software que se desea desarrollar. Por ejemplo, se puede disponer de una aplicación relacionada con un “cajero automático”, que al presentarla al cliente, permita al analista identificar las necesidades del cliente y por lo tanto los requisitos del software a construir.

Normalmente, el prototipo sirve como mecanismo para identificar los requisitos del software, y su construcción suele llevar las siguientes etapas:

- 1) Recolección de requisitos. El ingeniero de software y el cliente definen los objetivos globales del software, y aquéllos más específicos que se desean destacar con el prototipo.

- 2) Diseño rápido. Centrado en los aspectos del software visible al usuario (por ejemplo, interfaz de usuario, entradas y salidas...).

- 3) Construcción del prototipo.
- 4) Evaluación del prototipo. Se realiza por el cliente y usuarios, lo que permitirá concretar y refinar los requisitos del software a desarrollar.
- 5) Refinamiento del prototipo. Se produce un proceso iterativo en el que el prototipo es refinado para que satisfaga las necesidades del cliente, al tiempo que facilita al ingeniero de software un mejor conocimiento del sistema.
- 6) Producto. En la mayoría de los casos este sistema refinado (piloto) hay que desecharlo y hacer uno nuevo. Por ello, el desarrollo de un prototipo se debe planificar con el acuerdo expreso del cliente.

Algunos ingenieros del software abogan por desarrollar rápidamente un prototipo que les permita especificar completamente el sistema y obtener más consistentemente el producto final. Sobre el desarrollo rápido de prototipos, pueden realizarse las siguientes observaciones:

- Un prototipo rápido es básicamente una técnica de análisis que permite completar el conjunto de requisitos funcionales de un sistema software.
- Lo deseable es evolucionar el prototipo hasta obtener el producto final, en lugar de deshacerlo y construir un producto final nuevo. Este deseo es válido si del prototipo se puede obtener dicho producto (lo que no suele ser fácil), y su coste es inferior a su reconstrucción. Incluso, se podría

recomendar utilizar aquellas técnicas que permitan evolucionar un prototipo hasta el producto final.

- Cualquier aplicación nueva que el ingeniero de software sospeche que su funcionalidad puede presentar el riesgo de no ser aceptable para el usuario o si la interfaz de usuario es importante para el éxito de la aplicación, es una aplicación fuertemente candidata para que se desarrolle un rápido prototipo.
- En un proyecto de Prototipado bien planificado, aproximadamente el 50% del esfuerzo de desarrollo, desde su inicio hasta la aprobación final de su funcionalidad, es la contribución del usuario. Los equipos de Prototipado están compuestos típicamente por la mitad de usuarios y la otra mitad de desarrolladores software.
- Es habitual tener que tirar la primera versión de cualquier sistema que se desarrolle por primera vez. Por ello, es aconsejable que la primera demostración de un prototipo rápido sea intencionalmente imperfecta, de forma que sea barato de producir y muy fácil de modificar, para que se pueda garantizar que el sistema final que se suministra se ajuste mejor a los requisitos del usuario.
- El Prototipado rápido es una solución que “evita el riesgo” en lugar de una solución de riesgo. Así, el Prototipado rápido no introduce nuevos riesgos políticos o económicos al proceso de desarrollo de software, sino que reduce

significativamente varios factores de riesgo asociados con su desarrollo, como los que se han señalado anteriormente.

- El Prototipado rápido es un método normal para el desarrollo de nuevas aplicaciones y llegará a ser más y más evidente que el Prototipado rápido produce mejores sistemas y con costes más bajos.

1.9 Desarrollo Rápido de Aplicaciones

El Desarrollo Rápido de Aplicaciones, abreviado como RAD (del inglés Rapid Application Development) es un modelo de ciclo de vida que enfatiza un desarrollo extremadamente corto. Se trata de una adaptación del modelo tradicional en cascada en el que se logra el desarrollo rápido utilizando una construcción basada en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso RAD permite crear un sistema completamente funcional dentro de periodos cortos de tiempo (entre 60 y 90 días).(Conde, Nava, & Rojas, 2008)

Cuando se utiliza para aplicaciones de sistemas de información, el enfoque RAD tiene las siguientes fases:

1. Modelado de Gestión: se modela el flujo de información entre las funciones de gestión.

2. Modelado de Datos: se refina el flujo de información como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características de cada uno de los objetos y sus relaciones.
3. Modelado del Proceso: se definen las transformaciones (añadir, modificar, suprimir o recuperar) sobre los objetos del modelo de datos para lograr los flujos de información de cada función de gestión.
4. Generación de Aplicaciones: codificación de una función de gestión.
5. Pruebas y entrega: prueba de los componentes y entrega del programa que realiza una función de gestión.

La clave del modelo RAD está en los cambios en las etapas de codificación y pruebas:

- Codificación. El modelo RAD asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, se trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas para facilitar la construcción de software.
- Pruebas. Como se enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce en muchos casos el tiempo de pruebas, aunque se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

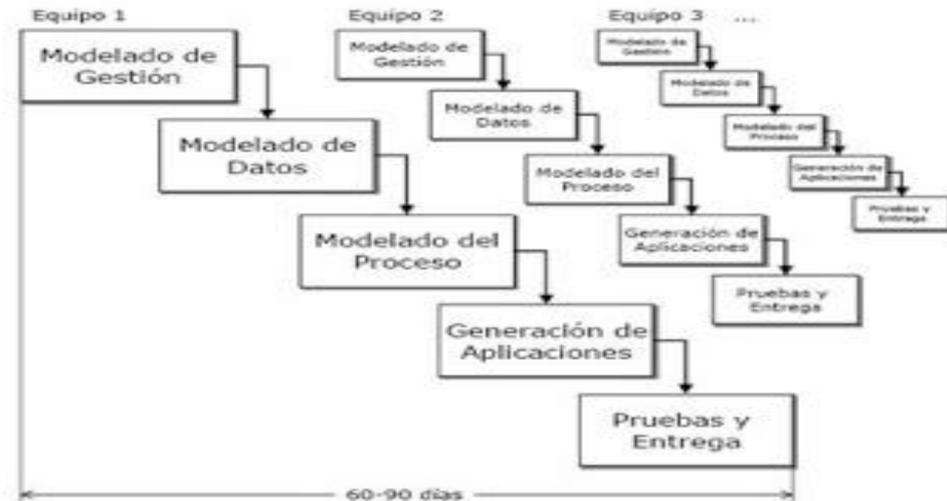


Figura 1-4 El modelo de Ciclo de Vida RAD

El modelo de ciclo de vida RAD se muestra en la Figura 4. Las limitaciones de tiempo impuestas en un proyecto RAD exigen que la aplicación cumpla con el requisito de “ámbito en escalas”, que indique que una aplicación pueda modularse de forma que cada una de las funciones principales pueda completarse en menos de tres meses. Además, cada una de las funciones puede ser afrontada por un equipo RAD separado e integrarse en un único conjunto.

1.10 Modelo en Espiral

Boehm cerró la década de los 80 publicando en 1988 un modelo de ciclo de vida en espiral (Figura 5) que sustituye a la solución en fases del “modelo en cascada” con ciclos de experimentación y aprendizaje. El modelo incorpora un nuevo elemento en el desarrollo de software como es el

“análisis de riesgos” y define cuatro actividades principales representadas por los cuatro cuadrantes de la figura:

- Planificación -> Determina objetivos, alternativas y restricciones
- Análisis de riesgo -> Evalúa alternativas, identifica y resuelve riesgos.
- Ingeniería -> Desarrollo y verificación del producto del siguiente nivel.
- Evaluación del cliente -> Valoración de los resultados y planificación de la siguiente fase.

Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se van construyendo sucesivas versiones del software, cada vez más completas. Durante la primera vuelta de la espiral, en el primer cuadrante (superior izquierdo) se determinan objetivos, alternativas y restricciones; y en el segundo cuadrante (superior derecho) se analizan e identifican los riesgos (¿se dispone de personal?, ¿está preparado?, ¿existe mercado para el producto?, etc.). Si el análisis de los riesgos indica que existe incertidumbre en los requisitos se puede desarrollar un prototipo para su valoración, y también se pueden usar simulaciones y otros modelos para definir más el problema y refinar los requisitos. (Conde, Nava, & Rojas, 2008)

En el cuadrante tercero (inferior derecho) se incorporan incrementalmente las etapas del ciclo de vida tradicional en cada ciclo de la espiral.

En el cuarto cuadrante (inferior izquierdo) el cliente evalúa el trabajo de ingeniería de esa espiral y sugiere modificaciones. Basándose en los comentarios del cliente, se produce la siguiente fase de planificación y de análisis de riesgos. En cada bucle alrededor de la espiral, al finalizar el análisis de riesgo, se debe tomar la decisión de seguir adelante o no con el proyecto. Si se sigue avanzando, cada vuelta alrededor de la espiral conduce más hacia fuera, hacia un modelo más completo del sistema, y al final al propio sistema operacional. Cada vuelta requiere más desarrollo de ingeniería de software, y el número de actividades del tercer cuadrante aumenta al alejarse del centro de la espiral.

El paradigma del modelo en espiral es actualmente el enfoque más realista en la ingeniería del software tradicional para sistemas grandes, ya que utiliza un enfoque evolutivo que permite al ingeniero y al cliente entender y reaccionar a los riesgos que se detectan en cada espiral. Utiliza la creación de prototipos como un mecanismo de reducción del riesgo y mantiene el enfoque del ciclo de vida clásico, pero incorporándolo dentro de un proceso iterativo que refleja de forma más realista el mundo real. No se disponen de cifras comparativas de su bondad con respecto a otros ciclos de vida, pero indudablemente sus resultados parecen superiores ya que engloba a los ciclos clásicos y de prototipos.

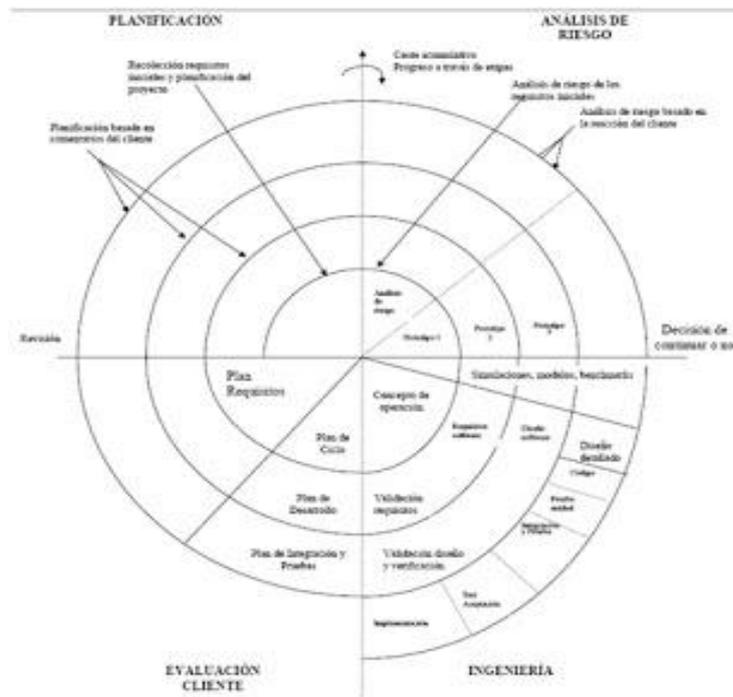


Figura 1-5 Modelo en espiral

1.11 Ventajas y Desventajas

Cada modelo de los existentes presenta sus ventajas e inconvenientes según el tipo de desarrollo que se quiera realizar y dependiendo de multitud de factores, puede resultar más adecuado utilizar uno y otro. Seguidamente, se van a resumir las principales características de los modelos de ciclo de vida más importantes:

1.12 Modelo “codificar y corregir”

Ventajas:

- Permite una construcción rápida del sistema

- Es útil para sistemas de un tamaño muy reducido, que no requiera más de dos o tres programadores y que no requiera un mantenimiento posterior
- No pierde tiempo en las etapas de planificación, documentación, control de calidad...
- Cualquiera, sin preparación técnica, lo puede utilizar

Desventajas:

- Carece de cualquier control y gestión del proceso
- No dispone de las fases necesarias en todo proyecto de software: especificaciones, diseño...
- Se dificulta la corrección de errores y el mantenimiento al carecer de una documentación del proceso adecuada
- No proporciona medios de evaluación ni de prevención de riesgos
- Resulta peligroso para proyectos grandes

1.13 Modelo en “cascada”

Ventajas:

- Es un modelo sencillo y disciplinado
- Es fácil aprender a utilizarlo y comprender su funcionamiento
- Está dirigido por los tipos de documentos y resultados que deben obtenerse al final de cada etapa
- Ha sido muy usado y, por tanto, está ampliamente contrastado
- Ayuda a detectar errores en las primeras etapas a bajo costo

- Ayuda a minimizar los gastos de planificación, pues se realiza sin problemas

Desventajas:

- Los proyectos raramente siguen el proceso lineal tal como se definía originalmente el ciclo de vida
- Es difícil que el cliente exponga explícitamente todos los requisitos al principio
- El cliente debe tener paciencia pues obtendrá el producto al final del ciclo de vida
- No refleja exactamente cómo se programa realmente el sistema, en el que suele haber un gran componente iterativo
- Puede resultar complicado regresar a etapas anteriores (ya acabadas) para realizar correcciones
- El producto final obtenido puede que no refleje todos los requisitos del usuario

1.14 Modelo en “V”

Ventajas:

- La relación entre las etapas de desarrollo y los distintos tipos de pruebas facilitan la localización de fallos
- Es un modelo sencillo y de fácil aprendizaje
- Hace explícito parte de la iteración y trabajo que hay que revisar
- Especifica bien los roles de los distintos tipos de pruebas a realizar

- Involucra al usuario en las pruebas

Desventajas:

- Es difícil que el cliente exponga explícitamente todos los requisitos
- El cliente debe tener paciencia pues obtendrá el producto al final del ciclo de vida
- Las pruebas pueden ser caras y, a veces, no lo suficientemente efectivas
- El producto final obtenido puede que no refleje todos los requisitos del usuario

1.15 “Prototipos”

Ventajas:

- Permite la construcción del sistema con requisitos poco claros o cambiantes
- El cliente recibe una versión del sistema en muy poco tiempo, por lo que lo puede evaluar, probar e, incluso, empezar a utilizarlo
- Se pueden introducir cambios en las funcionalidades del sistema en cualquier momento
- Involucra al usuario en la evaluación de la interfaz de usuario
- Se reduce el riesgo y la incertidumbre sobre el desarrollo
- Genera signos visibles de progreso, que se utilizan cuando existe una demanda en la velocidad del desarrollo
- Permite entender bien el problema antes de la implementación final

Desventajas:

- El cliente puede quedar convencido con las primeras versiones y, quizás, no vea la necesidad de completar el sistema o rediseñarlo con la calidad necesaria
- Requiere trabajo del cliente para evaluar los distintos prototipos y traducirlo en nuevos requisitos
- Requiere un tiempo adicional para definir adecuadamente el sistema
- No se sabe exactamente cuánto será el tiempo de desarrollo ni cuantos prototipos se tienen que desarrollar
- Si un prototipo fracasa, el coste del proyecto puede resultar muy caro

1.16 “Desarrollo Rápido de Aplicaciones”

Ventajas:

- Enfatiza ciclos de desarrollo extremadamente cortos
- Tiene las ventajas del modelo clásico
- Se asegura de que el producto entregado cumple las necesidades del cliente

Desventajas:

- Solo se puede aplicar si el sistema se puede modularizar de forma que permita completarse cada una de las funciones principales en menos de tres meses
- Para proyectos grandes puede requerir muchos equipos de trabajo distintos

- Requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias
- No resulta adecuado cuando los riesgos técnicos son elevados
- Se pueden tener problemas con la aceptación del prototipo

1.17 Modelo en “espiral”

Ventajas:

- Incorpora muchas de las ventajas de los otros ciclos de vida
- Conjuga la naturaleza iterativa de los prototipos con los aspectos controlados y sistemáticos del modelo clásico
- Proporciona el potencial para el desarrollo rápido de versiones incrementales
- Puede adaptarse y aplicarse a lo largo de la vida del software
- Es un enfoque realista del desarrollo del software
- Permite aplicar el enfoque de construcción de prototipos en cualquier momento para reducir riesgos
- Reduce los riesgos antes de que se conviertan en problemáticos
- Controla muy bien los riesgos y mientras más iteraciones se realicen, menos riesgos habrá
- Monitoriza y controla los riesgos continuamente

Desventajas:

- Puede resultar difícil convencer a algunos clientes de que el enfoque evolutivo es controlable

- Solo resulta aplicable para proyectos de gran tamaño
- Supone una carga de trabajo adicional, no presente en otros ciclos de vida
- Requiere una considerable habilidad para la evaluación y resolución del riesgo, y se basa en esta habilidad para el éxito
- Si un riesgo importante no es descubierto y gestionado, indudablemente surgirán problemas
- Es bastante complicado de realizar y su complejidad puede incrementarse hasta hacerlo impracticable
- El modelo no se ha utilizado tanto como otros, por lo que tendrán que pasar años antes de que determine con certeza la eficacia de este modelo

1.18 ¿Como elegir?

Luego de ver algunos de los modelos de ciclo de vida más utilizados surge la pregunta con la respuesta más codiciada: ¿qué modelo de ciclo de vida elegir? Sabemos que ninguno predomina sobre los otros. Por ello, debemos elegir el modelo que mejor se adapte al proyecto que desarrollaremos. Podemos analizar, para guiarnos en nuestra elección, la complejidad del problema, el tiempo que disponemos para hacer la entrega final, o si el usuario o cliente desea entregas parciales, la comunicación que existe entre el equipo de desarrollo y el usuario y, por último, qué certeza (o incertidumbre) tenemos de que los requerimientos dados por el usuario son correctos y completos

Tal y como se ha visto en el apartado anterior cualquier modelo tiene ventajas e inconvenientes, con lo que, al comenzar un proyecto, habrá que examinar la situación actual para comprobar cuál es el modelo más adecuado al caso.

Una primera pauta para elegir el modelo de ciclo de vida es que, cuanto más lineal sea el modelo, más rápido será su desarrollo. Sin embargo, y en contrapartida, cuanto más lineal sea los modelos más completos deberán ser los requisitos antes del comienzo del proyecto. Algunas cuestiones que pueden plantearse para elegir entre los distintos modelos de ciclo de vida son las siguientes:

- ¿Existe la posibilidad de que el entendimiento entre las partes cambie significativamente a medida que avance el proyecto? Si puede cambiar ese entendimiento, es decir, si los requisitos pueden cambiar, es recomendable optar por modelos que incluyan la realización de prototipos o el desarrollo evolutivo, de forma que se puedan acometer de forma cómoda los cambios del programa debidos a nuevos requisitos.
- ¿Se comprende bien toda la arquitectura del sistema al comenzar el proyecto? Si la arquitectura es estable desde el principio, puede optarse por modelos de ciclo de vida lineales en las fases de diseño y codificación. En caso contrario deberá optarse por modelos que realicen varias iteraciones en esas fases.

- ¿Cuánta fiabilidad se necesita? Si se necesita un alto grado de fiabilidad entonces será necesario acudir a modelos de ciclo de vida basados en la especificación formal de requisitos. En estos modelos se dedica una gran cantidad de tiempo a especificar de forma correcta el programa, que luego se construye derivando una implementación a partir de esa especificación mediante métodos que garantizan la corrección de lo que se produce.
- ¿Cuánto tiempo extra se necesita para planificar y diseñar durante el proyecto para posibles versiones futuras? Cuando se desea crear componentes reutilizables para futuras versiones del mismo programa, o bien para futuros programas que funcionen en el mismo entorno, es conveniente elegir modelos de ciclo de vida que pongan énfasis en la reutilización, como el modelo RAD.
- ¿Se está sometido a una planificación predefinida? En estos casos deberá elegirse el modelo que más se ajuste a esa planificación y que permita un desarrollo más controlado en riesgos, como el modelo en espiral.
- ¿Puede ser necesario realizar modificaciones a medio camino? Si es así deberá optarse por modelos de ciclo de vida basados en prototipos evolutivos o incrementales.

- ¿Debe proporcionarse a los clientes signos visibles de progreso durante el proyecto? En estos casos también se debe optar por modelos que permitan la construcción de prototipos incrementales.
- ¿Cuánta preparación y formación son necesarias para utilizar el modelo de ciclo de vida con éxito? En muchas ocasiones la selección del ciclo de vida viene impuesta por los conocimientos del personal del equipo de desarrollo. Si el elegir un modelo exige un largo periodo de entrenamiento, puede ser recomendable elegir otro modelo de ciclo de vida menos eficiente pero que ya conozcan los desarrolladores.

2 Base de Datos

2.1 Conceptos

2.1.1 Datos

“Es el componente fundamental de la base de datos que están relacionados entre sí formando un conjunto con mínimas redundancias. (Ángel Cobo.2008).”

2.1.2 Base de Datos

(Laudon & Laudon, 2012)Definen en su libro Sistemas de Información Gerencial (2012) una base de datos como una colección de datos organizados para dar servicios a muchas aplicaciones de manera eficiente, al centralizar los datos y controlar los que son redundantes. En vez de guardar los datos en archivos separados para cada aplicación, se almacenan de modo que los usuarios creen que están en una sola ubicación. Una sola base de datos de servicio a varias aplicaciones. Por ejemplo, en vez de que una corporación almacene los datos de los empleados en sistemas de información y archivos separados para personal, nómina y beneficios, podría crear una sola base de datos común de recursos humanos.

“Una base de datos es un conjunto de datos almacenados sin redundancias innecesarias en un soporte informático y accesible simultáneamente por

distintos usuarios y aplicaciones. Los datos deben estar estructurados y almacenados de forma totalmente independiente de las aplicaciones que la utilizan. (Ángel Cobo.2008).”

2.1.3 Base de datos relacionales

Una base de datos es una colección ordenada de datos, los cuales normalmente son almacenados en uno o más archivos asociados. La data se estructura en tablas, en las cuales la referencia cruzada es posible. La existencia de dicha relación entre tablas es que lo da como resultado lo que ha sido llamado bases de datos relacionales.(Kofler, 2005)

El modelo relacional surge en los años 70s y este modelo se caracteriza porque la información se guarda en tablas cuyos datos se relacionan. Las bases de datos relacionales a pesar de los grandes avances en la computación son una de la más utilizadas debido a su solidez y la forma íntegra como se almacena la información cuando se ha realizado un diseño adecuado de la estructura de datos,(Cabello, 2005).

El modelo relacional implicó un cambio mayor no tanto en la vista conceptual de los datos sino en su forma de implementar las relaciones a través de correspondencias entre claves primarias y foráneas y a través de la implementación de joins (*búsqueda de datos que se encuentran en distintas tablas*). Una ventaja era la aproximación, todavía más cercana, a la realidad

de la información, implicando mucho menos necesidades de navegación que el modelo jerárquico y más facilidad de mantenimiento que el modelo de redes, ya que muchos cambios de la estructura de datos se podía realizar mientras se utilizaba en ambiente de datos. Ha habido innumerables implementaciones exitosas del modelo relacional que continúan evolucionando hasta nuestros días y siguientes (Oracle, Infomix, Sybase, SQLServer, DB2,...).(Cardoso M., 2006)

El modelo de bases de datos relacional es uno de lo más usado en la actualidad debido a la gran flexibilidad a la hora de realizar reportes empresariales y la rapidez y eficiencia como los gestores de bases de datos relacionales manejan la información. Entre los principales gestores de bases de datos relacionales están: Microsoft SQL Server, Oracle, MySQL, etc.

2.1.4 Lenguaje de consulta estructurado (SQL)

Según (Kofler, 2005)(2005) el lenguaje de consulta estructurado es el mecanismo usado para manipular bases de datos. SQL es usado para formular instrucciones y procesos a sistemas de bases de datos, incluyendo consultas y comandos para cambiar o eliminar objetos de la base de datos.

2.1.5 Base de datos espaciales

Para (Dave, 2007)(2007), una base de datos espacial es una base de datos optimizada para almacenar y consultar data relacionada con objetos en el espacio, incluyendo puntos, líneas y polígonos. Mientras las bases de datos típicas pueden procesar datos numéricos y datos tipos caracteres, es necesario funciones adicionales para lograr el manejo de data de tipo espacial.

2.1.6 Sistema de Gestión de Base de Datos (SGBD):

Es un software con conjunto de programas que permite crear y mantener una base de datos. El SGBD actúa como interfaz entre los programas de aplicación (Usuarios) y el sistema operativo. El objetivo principal de un SGBD es proporcionar un entorno eficiente a la hora de almacenar y recuperar la información de las bases de datos. (Cobo, 2008)

(Laudon & Laudon, 2012)Definen un SGBD como un software que permite a una organización centralizar los datos, administrarlos en forma eficiente y proveer acceso a los datos almacenados mediante programas de aplicación y los archivos de datos físicos. Cuando el programa de aplicación solicita un elemento de datos, como el sueldo bruto, el DBMS lo busca en la base de datos y lo presenta al programa de aplicación. Si utilizara archivo de datos tradicionales, el programador tendría que especificar el tamaño y formato de

cada elemento de datos utilizado en el programa y después decir a la computadora en donde están ubicados.

(Laudon & Laudon, 2012) Continúan explicando que el DBMS libera al programador o al usuario final de la tarea de comprender en dónde y cómo están almacenados los datos en realidad, al separar las vistas lógicas y física de los datos. La vista lógica presenta los datos según la manera en que los perciben los usuarios finales o los especialistas de negocios, mientras que la vista física muestra la verdadera forma en que están organizados y estructurados los datos en los medios de almacenamientos físicos.

El DBMS hace que los datos almacenados estén disponibles para los usuarios, digamos que es la herramienta para entrar a la base de dato y obtener los datos almacenados en ella.

2.1.7 Sistema de Información geográfica

(SIG o GIS, en su acrónimo inglés Geographic Information System) es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información.

2.1.8 Business Inteligente

Es el conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar datos de los sistemas transaccionales e información desestructurada en información estructurada, para su explotación directa o para su análisis y conversión en conocimiento, dando así soporte a la toma de decisiones sobre el negocio.

“Las aplicaciones de Business Intelligence (BI) son herramientas de soporte de decisiones que permiten en tiempo real, acceso interactivo, análisis y manipulación de información crítica para la empresa. Estas aplicaciones proporcionan a los usuarios un mayor entendimiento que les permite identificar las oportunidades y los problemas de los negocios. Los usuarios son capaces de acceder y apalancar una vasta cantidad de información y analizar sus relaciones y entender las tendencias que últimamente están apoyando las decisiones de los negocios. Estas herramientas previenen una potencial pérdida de conocimiento dentro de la empresa que resulta de una acumulación masiva de información que no es fácil de leer o de usar.”(CherryTree, 2000)

El exceso de información no es poder, pero el conocimiento si lo es, dicen (Vitt, Luckevich, & Misner, 2003). Con demasiada frecuencia, la transformación y el análisis de toda la información y los datos que las propias compañías generan se convierte en un verdadero problema y, por lo tanto, la toma de decisiones se vuelve desesperadamente lenta.

Las tecnologías de BI intentan ayudar a las personas a entender los datos más rápidamente a fin de que puedan tomar mejores y más rápidas decisiones y, finalmente, mejorar sus movimientos hacia la consecución de objetivos de negocios. Los impulsores claves detrás de los objetivos de BI son incrementar la eficiencia organizacional y la efectividad. Algunas de las tecnologías de BI apuntan a crear un flujo de datos dentro de la organización más rápido y accesible. Por otro lado, novedosas tecnologías de BI toman un enfoque más agresivo redefiniendo los procesos existentes con otros nuevos, mucho más estilizados que eliminan gran cantidad de pasos o crean nuevas capacidades.(SiGIS, 2014)(Vitt, Luckevich, & Misner, 2003)

2.2 Características

De conformidad con FISMAT(2000), las principales características de una base de datos relacional son:

- Una base de datos relacional se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.

- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones.

2.3 Oracle vs MySQL vs SQL Server: Comparación de SGBDR más Populares

2.3.1 Sumario Comparativo de Características

La siguiente tabla muestra información acerca de Oracle, MySQL y Bases de Datos de Servidores SQL, y como ellas se comparan.

Tabla 2-1

Feature	Oracle	MySQL	SQL Server
Interfaz	GUI, SQL	SQL	GUI, SQL, Various
Lenguaje Soportado	Many, including C, C#, C++, Java, Ruby, and Objective C	Many, including C, C#, C++, D, Java, Ruby, and Objective C	Java, Ruby, Python, VB, .Net, and PHP
Sistema Operativo	Windows, Linux, Solaris, HP-UX, OS X, z/OS, AIX	Windows, Linux, OS X, FreeBSD, Solaris	Windows
Licencia	Propietario	Código Libre	Propietario

2.3.2 Oracle

2.3.2.1 Historia

IBM fue la primera empresa en desarrollar un sistema de gestión de bases de datos relacionales, sin embargo, Oracle Corporation hizo historia en 1980 por la liberación para uso comercial de su RDBMS, Oracle. Sólo unos pocos años después, la compañía lanzaría una versión de su sistema de computadoras de IBM. Desde su exposición al mercado de RDBMS, Oracle ha llevado el camino constantemente. De acuerdo con Gartner, Oracle poseía casi el 50 % del mercado de RDBMS en 2011. Además de la apertura del mercado comercial para RDBMS, Oracle Corporation también fue la primera empresa en desarrollar una versión de nivel comercial de SQL, que fue diseñado para manipular datos en un RDBMS utilizando (en ese momento) consultas y conexiones. (Dubretic, 2014)

2.3.2.2 Características

La primera versión “real” del sistema de gestión de bases de datos relacionales Oracle fue Oracle 2. Este sistema admitía sólo las características básicas de SQL, y estaba escrito en un lenguaje ensamblador. Al año siguiente, y durante los próximos 10 años más o menos, Oracle Corporation lanzó actualizaciones a su base de datos buque insignia. Probablemente una de las razones por las que el sistema de gestión de bases de datos relacionales de Oracle ha logrado permanecer en la cima sea gracias a sus actualizaciones de productos que están estrechamente

vinculados a los cambios en el mercado. Palabras de moda de bases de datos tales como “escalable”, “programable”, “distribuida”, y “portátil” también están vinculadas a la liberación de Oracle. Por ejemplo, en 1985 se añadió soporte para un modelo cliente-servidor a la espera de una aceptación cada vez mayor de la comunicación por red. A medida que Internet allanó el camino para la era digital, el sistema de gestión de bases de datos relacionales de Oracle se ha actualizado para incluir una máquina virtual Java nativa (JVM) . (Dubretic, 2014)

Oracle Database 12c es la más reciente liberación, e incluye las siguientes características:

- Nueva redacción de datos para mejorar la seguridad de datos sensibles
- La introducción de la plataforma de Oracle Advanced Analytics
- Nuevo manejo de base de datos para los archivos Flash Data Archive (FDA)
- El apoyo a la integración con los grupos de procesadores de sistema operativo
- Apoyo al bombeo de los datos para la consolidación de las base de datos
- Varias mejoras en Oracle Application Express, una herramienta de desarrollo rápido que permite a los usuarios desarrollar aplicaciones web utilizando SQL y / o PL /SQL.
- Compresión avanzada de la red para mejorar el rendimiento

2.3.3 SQL Server

2.3.3.1 Historia

Microsoft SQL Server entró en el mercado de los RDBMS como un competidor serio a mediados de 1990, cuando Microsoft compró a Sybase, y luego lanzó la versión 7.0. Las empresas originalmente trabajaron juntas para desarrollar la plataforma y hacerla funcionar en el sistema operativo de IBM OS/2. No obstante, Microsoft finalmente desarrolló su propio sistema operativo (Windows NT), y quería trabajar solo para crear una gestión de base de datos única para su nuevo sistema operativo. Se necesitarían varios años para que Microsoft y Sybase cortaran completamente sus lazos. Sybase finalmente cambió el nombre de su producto de modo que fuera totalmente diferente al producto vendido a Microsoft. Microsoft SQL Server versión 4.2 fue la versión inicial. (Dubretic, 2014)

2.3.3.2 Características

En 2000, Microsoft lanzó SQL Server 2000. El lanzamiento fue un hito importante para la empresa, ya que fue la primera versión del producto, donde se reemplazó el código original de Sybase. Trabajando en la misma línea que Oracle Corporation, Microsoft ha tratado de mejorar SQL Server para seguir el ritmo de los cambios tecnológicos. SQL Server 2005 es un ejemplo. El lenguaje de marcado extensible (XML) recibió el sello de aprobación del W3C y comenzó a ganar terreno a finales de 1990. Una de las principales novedades de SQL Server 2005 fue el apoyo a los datos

XML. Otras características notables del producto insignia incluyen la introducción de SQL Server Always On (tecnología de gestión de datos para disminuir el tiempo de inactividad del usuario a raíz de fallos en el sistema), soporte para datos estructurados y semi-estructurados, una mayor compresión, y varios complementos para apoyar a otros productos en el mercado. SQL Server 2012 se proclamó como la última versión que incluye soporte nativo para OLE. (Dubretic, 2014)

SQL Server 2014 es la última versión de SQL Server e incluye las siguientes características:

- Introducción de In-Memory Online Traction Processing (OLTP), una característica incorporada que permite una sofisticada gestión de base de datos para mejorar el rendimiento.
- Nuevas soluciones para manejar la recuperación de desastres
- Versión actualizada de la Herramientas de Datos de SQL Server para Inteligencia de Negocios (BI SSDT)

2.3.4 MySQL

2.3.4.1 Historia

Hay dos aspectos únicos de MySQL en comparación con Oracle y SQL Server: no se desarrolló originalmente para uso comercial y es una base de datos de código abierto. El surgimiento de esta plataforma de base de datos fue una casualidad sucedida a las personas que comenzaron a desarrollarla

mientras trataban de usar MSQL para conectar sus tablas en la base de datos, y decidieron que necesitaban una interfaz mucho más potente. La fase inicial de MySQL utilizó una API heredada de mSQL, mejoras que aumentan considerablemente la velocidad, y otras características que incluían el motor de almacenamiento InnoDB, búsqueda de texto, la portabilidad y la internacionalización. (Dubretic, 2014)

(Dubretic, 2014)Continúa diciendo que otra diferencia de la plataforma de MySQL en comparación con las otros dos es que es de código abierto. La era digital dio lugar a un movimiento de colaboración para el desarrollo de software que se ha convertido en un mercado competitivo para las bases de datos y otro software. De acuerdo con informes de mercado, hay alrededor de 10 millones de instalaciones de MySQL, lo que indica que la plataforma se está moviendo rápidamente en el espacio empresarial.

La propiedad de MySQL ha hecho la transición desde los humildes inicios del producto. Las dos adquisiciones más notables son: (1) en 2008, cuando Sun Microsystems adquirió MySQL AB, la compañía que creó MySQL, y (2) en 2010, cuando Oracle compró Sun Microsystems.

2.3.4.2 Características

Oracle y SQL Server se consideran herramientas que favorecen a los usuarios con los sistemas empresariales de gran tamaño, mientras que MySQL se considera una herramienta que apela más a menudo a las personas interesadas en la gestión de las bases de datos asociadas a sus sitios

web. Al igual que con Oracle y SQL Server, MySQL ha liberado actualizaciones para su software casi todos los años. La versión original fue desarrollada a mediados de la década de 1990. Los cambios más notables a MySQL fueron en 2010, el momento de la última adquisición en 2010. Las mejoras en esta versión (GA release 5.5) incluyen replicación semisincrónica, el particionamiento personalizado, soporte mejorado para SMP y las actualizaciones del subsistema InnoDB E/S. (Dubretic, 2014)

2.4 BigTable

Bigtable es un sistema de almacenamiento distribuido para gestionar datos estructurados que está diseñado para escalar a un gran tamaño: petabytes de datos a través de miles de productos básicos servidores. Muchos proyectos de datos de la tienda de Google guardan sus datos en Bigtable, incluida la indexación web, Google Earth y Google Finance. Estas aplicaciones colocan muy diferentes demandas en Bigtable, tanto en términos de tamaño de los datos (de URLs a páginas web a las imágenes de satélite) y requisitos de latencia (de procesamiento a granel backend de datos en tiempo real que sirve). A pesar de estas distintas condiciones Bigtable tiene éxito proporcionado una solución flexible y de alto rendimiento para todos estos productos de Google. (Google, Inc., 2006)

2.5 Google Earth

Google opera un conjunto de servicios que proporcionan los usuarios con acceso a la alta resolución de imágenes de satélite de la superficie mundial, tanto a través de la Google basado en la Web Mapas de interfaz (maps.google.com) y a través de la Google Earth (earth.google.com) cliente de software-ware personalizado. Estos productos permiten a los usuarios navegar a través de la superficie del mundo: pueden desplazarse, ver y realizar anotaciones en las imágenes-satelitales en muchos diferentes niveles de resolución. Este sistema utiliza una tabla con los datos de pre proceso, y un diferente conjunto de mesas para servir los datos del cliente. (Google, Inc., 2006)

(Google, Inc., 2006) Continúa diciendo que el pipeline de procesamiento utiliza una tabla para almacenar imágenes crudas. Durante el pre procesamiento, las imágenes se limpian y consolidan en los datos finales que se sirven. En esta tabla se contiene aproximadamente 70 terabytes de datos y por lo tanto es servido desde el disco. Las imágenes están eficientemente comprimidas así que la compresión de BigTable está deshabilitada.

Cada fila de la tabla de las imágenes corresponde a un segmento geográfico singular. Las filas se nombran para asegurarse de que segmentos geográficos adyacentes se almacenan cerca unos de otros. La tabla contiene una familia de columnas para hacer un seguimiento de la fuente de datos para cada segmento. Esta familia de columnas tiene un gran número de

columnas: una para cada imagen de datos en bruto. Dado que cada segmento sólo se construye a partir de unas pocas imágenes, esta familia de columnas es muy escasa.

El pipeline de pre procesamiento se apoya fuertemente en MapReduce sobre BigTable para transformar los datos. El sistema global procesa más de 1 MB / seg de datos por servidor comprimido durante algunos de sus ejecuciones.

El sistema de servicio utiliza una tabla con los datos de índice almacenados en GFS. Esta tabla es relativamente pequeño (500 GB), pero debe servir a decenas de miles de peticiones por segundo a cada centro de datos con baja latencia. Como resultado, esta tabla es organizada a través de cientos de servidores de tabletas y contiene en memoria familias de columna.

3 Desarrollo Web

(Wikipedia, 2014) Nos dice Desarrollo web es un título algo arbitrario para el conjunto de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos de base de datos con el uso de un navegador en Internet a fin de realizar determinadas tareas o mostrar información. Wikipedia, por ejemplo, ha sido realizada por desarrolladores web.

(Wikipedia, 2014) continua diciendo que tradicionalmente un software departamental o incluso un ambicioso proyecto corporativo de gran envergadura es desarrollado en forma standalone, es decir, usando lenguajes ya sea compilados (C, C++, Delphi), semicompilados (.NET, Mono, Java), o interpretados (Python) para crear tanto la funcionalidad como toda la interfaz de los usuarios, pero cabe perfectamente un desarrollo orientado a web para dichos propósitos, siendo más homogéneo y multiplataforma, y dependiendo de las tecnologías utilizadas, más rápido y robusto tanto para diseñar, implementar y probar, como para su uso una vez terminado.,

Funcionalmente, el desarrollador web, que es quien realiza esta labor, normalmente sólo se preocupa por el funcionamiento del software, es tarea del diseñador web preocuparse del aspecto final (layout) de la página y del web máster el integrar ambas partes. En ocasiones el web máster también se encarga de actualizar los contenidos de la página.

Los lenguajes de programación más usados en desarrollo web son principalmente: ASP.NET, PHP y JSP, aunque aún hay quienes usan ASP, Macromedia ColdFusion y Perl. Algunos desarrolladores web hablan muy bien de Ruby, y el framework Ruby on Rails, pero no está muy difundido todavía. También hay un proyecto para usar Pascal en web a través del lenguaje Pascal Server Pages - PSP pero aún no se han concretado desarrollos, hasta donde se conoce.

El sistema de gestión de base de datos más popular en desarrollo web es MySQL, seguida por Oracle, SQL Server y PostgreSQL, también puede usarse perfectamente Firebird o HSQL.

Idealmente un desarrollador web debería conocer, además de al menos un lenguaje de programación y el manejo de al menos una base de datos, HTML, CSS y JavaScript, tres tecnologías que bien trabajadas logran DHTML. También, si el diseñador desea usar Adobe Flex (Flash) debe ser capaz de integrarlo a su solución, o puede dejar los efectos en el mismo DHTML.

3.1 HTML5, CSS3 y JavaScript

(Mozilla Developer Network, 2013) Expone que HTML5 es la última evolución de la norma que define HTML. El término representa dos conceptos diferentes

- Se trata de una nueva versión del *lenguaje* HTML, con nuevos elementos, atributos y comportamientos,

- y un conjunto más amplio de tecnologías que permite a los sitios Web y las aplicaciones más diversas y de gran alcance. Este conjunto se le llama *HTML5 y amigos* y, a menudo reducido a sólo *HTML5*.
Diseñado para ser utilizable por todos los desarrolladores de Open Web, esta referencia página enlaza numerosos recursos sobre las tecnologías de HTML5, que se clasifican en varios grupos según su función.
- *Semántica*: lo que le permite describir con mayor precisión cuál es su contenido.
- *Conectividad*: lo que le permite comunicarse con el servidor de formas nuevas e innovadoras.
- *Desconectado y almacenamiento*: permite a páginas web almacenar datos, localmente, en el lado del cliente y operar fuera de línea de manera más eficiente.
- *Multimedia*: permite hacer vídeo y audio de ciudadanos de primera clase en la Web abierta.
- *Gráficos y efectos 2D/3D*: permite una gama mucho más amplia de opciones de presentación.
- *Rendimiento e Integración*: proporcionar una mayor optimización de la velocidad y un mejor uso del hardware del equipo.
- *Dispositivo de Acceso*: admite el uso de varios dispositivos de entrada y salida.
- *Stirling*: deja a los autores escribir temas más sofisticados.

Analizador de HTML5 compatible

(Mozilla Developer Network, 2013) nos continua diciendo que el programa de análisis, lo que convierte los bytes de un documento HTML en un DOM, se ha ampliado y ahora define con precisión el comportamiento de utilizar en todos los casos, incluso cuando se enfrentan a HTML inválido. Esto conduce a una mucho mayor previsibilidad e interoperabilidad entre los navegadores compatibles con HTML5.

WebRTC

Esta tecnología, donde RTC es sinónimo de comunicación en tiempo real, permite conectar con otras personas y servicio de videoconferencia de control directamente en el navegador, sin necesidad de un plugin o una aplicación externa.

IndexedDB

Es un estándar de Internet para el almacenamiento de grandes cantidades de datos estructurados en el navegador y para las búsquedas de alto rendimiento de estos datos mediante índices.

Uso de archivos desde aplicaciones web

Soporte para la nueva API de archivos de HTML5 ha sido añadido a Gecko, por lo que es posible que las aplicaciones de Internet accedan a los archivos locales seleccionados por el usuario. Esto incluye el apoyo para la selección de varios archivos utilizando el <input> de tipo *file* del elemento HTMLmúltiples atributos. También esta FileReader.

Usando la API de la cámara

Permite utilizar, manipular y almacenar una imagen de la cámara del ordenador.

Motores JIT compilación tarde JavaScript

La nueva generación de motores de JavaScript es mucho más poderosa, lo que lleva a un mayor rendimiento.

Historia API

Permite la manipulación del historial del navegador. Esto es especialmente útil para la carga interactivamente nueva información en las páginas.

Fullscreen API

Controla el uso de la pantalla completa de una página Web o aplicación, sin la interfaz de usuario del explorador mostrada.

API Pointer Lock

Permite bloquear el puntero al contenido, por lo que los juegos y aplicaciones similares no pierden el foco cuando el puntero alcanza el límite de la ventana.

Eventos táctiles

Manipuladores para reaccionar a los eventos creados por un usuario cuando pulsa pantallas táctiles.

El uso de geo localización

Permite a los navegadores localizar la posición del usuario mediante geo localización.

Detección de la orientación del dispositivo

Obtiene la información cuando el dispositivo, en el que se ejecuta el navegador, cambia de orientación. Esto puede ser utilizado como un dispositivo de entrada (por ejemplo, para hacer que los juegos que reaccionan a la posición del dispositivo) o adaptar el diseño de una página a la orientación de la pantalla (vertical u horizontal).

3.2 CSS3 STYLING

Según (Mozilla Developer Network, 2013)CSS se ha ampliado para ser capaz de elementos de estilo de una manera mucho más compleja. Esto se refiere a menudo como CSS3, aunque CSS no es una especificación monolítica más y los diferentes módulos no están todos en el nivel 3: algunos están en el nivel 1 y un poco en el nivel 4, con todos los niveles intermedios cubiertos.

Animación de su estilo

Utilizando Transiciones CSS para animar entre los diferentes estados o utilizando animaciones CSS para animar partes de la página, sin que evento triggering, ahora pueda controlar los elementos móviles en su página.

Tipografía mejorada

Los autores tienen un mejor control para llegar mejor a la tipografía. Se puede controlar el texto de desbordamiento y guiones, pero también puede una sombra de ella o controlar con mayor precisión sus decoraciones. Tipos de letra personalizados se pueden descargar y aplicar gracias a la nueva regla `@font-face` .

4 Desarrollo Móvil

(Pastor, 2014) Nos dice que la empresa VisionMobile realiza desde hace tiempo su informe Developer Economics, y en la sexta edición, que estudia el primer trimestre de 2014, se analizan las tendencias del mercado en cuanto al interés de los desarrolladores por el segmento de la movilidad.

THE PLATFORM SHOOTOUT

	 ANDROID	 iOS	 HTML5 MOBILE	 WINDOWS PHONE	 BLACK BERRY 10
Sales market share (smartphones, Q3 2013)	81%	13%	-	4%	2%
Mindshare	71%	55%	52%	26%	14%
Priority	37%	32%	14%	6%	5%
Loyalty	52%	59%	26%	24%	35%
Most popular in	Asia	North America	South America	Asia	South America
Median revenues	\$150	\$750	\$150	\$25	\$75
Differentiating selection criterion	Open Source	Revenue potential	Ease of porting	Choice of development environment	Documentation/ Access to hardware APIs
3rd party tools index	2,8	3,1	2,5	2,5	2,3
Top revenue model	Advertising	Contract development	Contract development	Advertising	Pay per download
Segments with a strong preference to the platform	Hobbyists, Gold Seekers	Digital Media Publishers, Hunters, Guns for Hire	Product Extenders, Enterprise IT	Hobbyists, Explorers	-



Licensed under CC BY ND | Copyright VisionMobile

Source: Developer Economics Q1 2014 | www.DeveloperEconomics.com/go

Figura 4-1 Competencia de plataformas

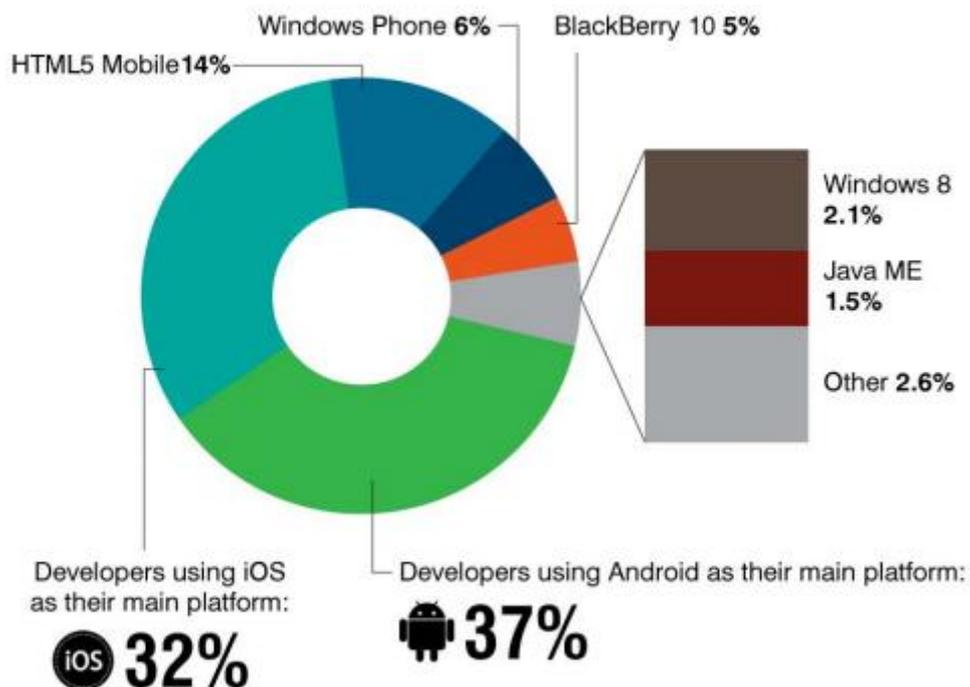
La base del estudio es sólida, continúa(Pastor, 2014): tanto Intel como Mozilla son los principales patrocinadores —hay muchos más en la lista—, y en él participan más de 7.000 desarrolladores de 127 países distintos. Una buena representación para un total estimado de 2,3 millones de desarrolladores en 2013, y para una economía que generó 68.000 millones de dólares en 2013 y que se estima ingresará 143.000 millones en 2016.

El estado del desarrollo de aplicaciones móviles parece bastante claro: Android e iOS se repartieron el 94% de las ventas de software en smartphones en el cuarto trimestre de 2013 según ese estudio. De ese porcentaje el 71% se dedicó a Android, mientras que el 55% desarrolla en iOS. Como se puede comprobar, parte de los desarrolladores trabaja de forma paralela en ambas plataformas móviles.

Lo interesante no es tanto lo que se da hoy en día como las tendencias que deja entrever ese estudio: por ejemplo, que iOS es la plataforma preferida en países desarrollados, y los que solo se dedican a dicha plataforma (el 59% de los desarrolladores encuestados) son más en porcentaje que los que solo se dedican a Android (el 52%).

DEVELOPER LOYALTY SPLIT BETWEEN IOS & ANDROID

% of developers using each platform as their main platform (n=6,311)



Licensed under CC BY ND | Copyright VisionMobile

Source: Developer Economics Q1 2014 | www.DeveloperEconomics.com/go

Figura 4-2 Lealtad de desarrolladores

Lo cierto es que la evolución parece ir determinando que cuantos más usuarios, mejor para los desarrolladores, a pesar de los mayores ingresos que tradicionalmente se consiguen en iOS. En la última encuesta el 37% de los desarrolladores priorizan Android, por un 34% en la anterior edición, mientras que iOS se ha mantenido estable con un 32% de interés en priorizar para la plataforma de Apple. En el caso de HTML5, la prioridad se sitúa en un 14%, mientras que en el tercer trimestre de 2013 era del 17%: la

falta de herramientas de desarrollo maduras parece haber sido la causa de ese descenso.(Pastor, 2014)

4.1 Tecnología Móvil en la Republica Dominicana

La siguiente imagen nos muestra el crecimiento de los OS Móviles en la Republica Dominicana a partir de www.statcounter.com

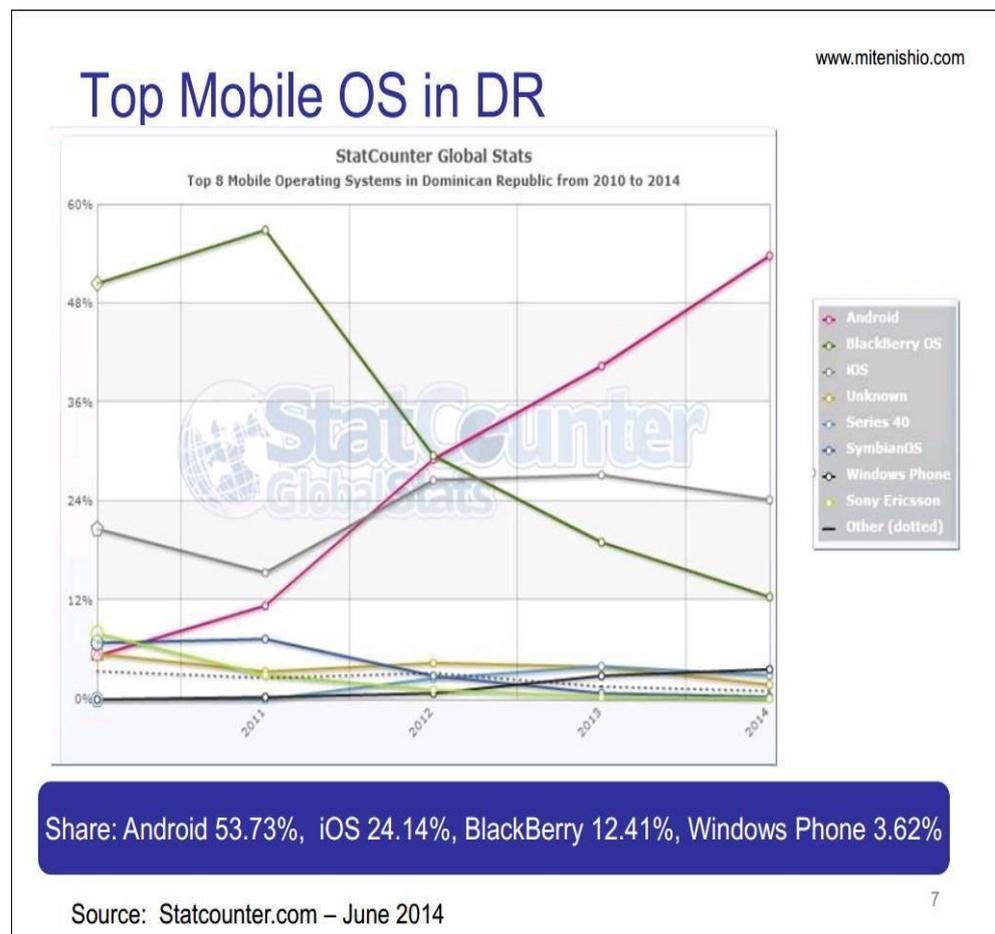


Figura 4-3 Mejores Sistemas Operativos Móviles

Dada estas estadísticas para este proyecto se sugiere una implementación para android de la aplicación.

5 Sistemas de Información Geográfica

5.1 Conceptos

5.1.1 Cartografía

Este quizá es uno de los elementos más importantes a considerar en el desarrollo de un proyecto SIG. Gran parte de los datos y en general la mayoría de los resultados son de tipo cartográfico. Por ello, esta disciplina aporta elementos de gran valor no solo para la confección, sino para el uso de mapas topográficos, y temáticos (Iturbe, S. Nchez, & Ch as, 2009).

5.1.2 Mapa

Es el lenguaje gráfico de la Geografía, que se utiliza para representar espacialmente los análisis y resultados de los estudios que combinan hechos y fenómenos ambientales, sociales y económicos (E. Rhoades).

Un mapa contiene la información geográfica, resalta las relaciones geográficas importantes y presenta resultados de análisis. Debido a que la mayoría de los usuarios de los SIG, tiene que presentar sus datos espaciales gráficamente a una gran variedad de lectores, ha hecho de ellos también diseñadores de mapa o cartógrafos (Peña Llopis, 2006).

5.1.3 Sistemas de información geográfica

Un Sistema de Información Geográfica (SIG) es un conjunto organizado de hardware, software y datos geográficos, diseñados específicamente para capturar, almacenar, actualizar, manipular, analizar y mostrar todo tipo de información referenciada geográficamente (ESRI, 1990).

5.1.4 Servicios basados en Localización

Un Servicio basado en localización (LBS) es un conjunto de herramientas que proporcionan servicios personalizados con ayuda de la localización geográfica del usuario o alguno otro objeto (González, 2007). Los Servicios Basados en Localización proporcionan información precisa sobre la localización, mediante dispositivos móviles como teléfonos celulares, GPS o RFID.

Un Servicio Basado en Localización no necesariamente es usado para localizar a una persona. Existen muchos sistemas que usan estos servicios para rastrear mercancías o vehículos incluso algunas compañías aseguradoras utilizan estos servicios para conocer el lugar donde ocurren los accidentes sin tener que preguntar a la persona afectada donde se encuentra.

5.1.5 Servicios de Orientación o Navegación

Los Servicios de Orientación o Navegación son una aplicación específica de los Servicios Basados en Localización. Estos servicios explotan la localización del usuario para permitirle construir rutas sobre un mapa,

llevarlos paso a paso por la ruta, determinar la distancia a recorrer y mostrar sitios que le podrían interesar al usuario, como restaurantes, museos o sitios turísticos.

Los ejemplos más importantes de sistemas que usan Servicios de Orientación o Navegación son Google Maps y más recientemente Ovi Maps de Nokia:

- 1) Google Maps (Google Inc., 2013). La herramienta cartográfica de Google ofrece mapas en 2D de calles y carreteras, mapas satelitales, vista dual y vista con el relieve del terreno. Google Maps está disponible vía Web, en aplicaciones de escritorio como Google Earth y también para dispositivos. Ofrece varios servicios como:
 - Mi ubicación: Detecta de la localización del usuario por medio de su navegador web usando Google Gears o con el API de Geo localización de la W3C. Esta opción también está disponible para dispositivos móviles por medio de GPS, Cell-ID y localización por WiFi.
 - Encontrar empresas junto con su información de contacto y como llegar a ellas dando una dirección o su nombre.
 - Street View: Esta opción permite ver imágenes de las calles de una ciudad y poder navegar como si se estuviera ahí.
 - Calcula rutas para caminar, trasladarse en auto, transporte público o bicicleta determinado distancia y el tiempo del recorrido, además de mostrar sitios de interés en la ruta y tráfico. Esta opción solo está

disponible en EU. Incorporar otras búsquedas relacionadas con la ubicación actual del mapa como videos, fotografía y artículos de Wikipedia.

- Además de mapas de la Tierra, se están incorporando Google Moon y Google Mars que son mapas de la Luna y Marte.

5.1.6 Tecnologías de Localización

Lo primero que debe hacer un Servicio Basado en Localización es encontrar la posición del usuario. En respuesta al problema de localización se han desarrollado tecnologías como las que usan los GPS (Global Positioning System), los sistemas basados en RFID (Radio Frequency Identification) o técnicas de localización basadas telefonía celular con información provista por proveedores de servicios.

5.1.7 GPS

GPS o Global Positioning System por sus siglas en inglés es un sistema de 24 satélites que orbitan la Tierra usados para determinar la localización del dispositivo GPS, ésta se obtiene con al menos 4 satélites para triangular la posición mediante la intersección de esferas que trazan los satélites, aunque en la práctica se pueden usar más satélites dado el tiempo respuesta de los satélites con el dispositivo GPS y refinar la posición obtenida.(Martínez Gens, 2006)

Su más grande ventaja es darnos nuestra posición con gran precisión y estar disponible en todo el planeta, pero tiene serias limitaciones en ambientes urbanos debido al fenómeno llamado “urban canyon” que impide a los satélites ubicar al dispositivo GPS(Rashid, 2008).

5.1.8 Telefonía celular

Los teléfonos celulares pueden ayudar en el problema de la localización gracias a su portabilidad, sencillez para manipularlos y la información provista por los proveedores del servicio de telefonía.

Las técnicas de localización que pueden usarse para determinar la ubicación de un teléfono celular sin necesidad de un GPS u otro dispositivo extra, que para (Castañeda, 2006) haciendo uso de los datos del proveedor de servicios de telefonía son:

- COO (Cell of Origin): Consiste en encontrar el ID de la celda donde se encuentra conectado un dispositivo móvil, cuya información nos puede orientar sobre la zona donde se encuentra el usuario, la cual varía según la zona urbana (entre 2 km y 20 km).
- ToA (Time of Arrival): Es el tiempo que tarda en llegar la señal de una torre al dispositivo móvil. Para obtener su posición se necesitan al menos 3 torres para hacer una triangulación.

- AoA (Angle of Arrival): Se mide el ángulo de arribo de 2 estaciones para determinar el cruce y así saber la localización del dispositivo móvil.

Estas técnicas no son muy eficaces en cuanto precisión y dependen de la información provista por las estaciones base. Esta información no está disponible para el público por cuestiones de seguridad, aunque existen bases de datos libres como OpenCellID que tiene la información de la mayoría de estaciones base del mundo y provee un API para obtener la longitud y latitud del teléfono por su Cell-ID con una precisión de 1000 metros (8Motions, 2009).

5.1.9 WiFi

La localización por medio de redes inalámbricas como WiFi intenta obtener la ubicación del usuario por medio de triangulación o búsqueda en bases de datos de las direcciones IP de la red para intentar una aproximación.

Se han intentado varias formas de localización como en la presentada por el Centro para el Desarrollo de las Telecomunicaciones de Castilla y León, Cedetel en España (G. Pedraz, 2009), en donde se intenta obtener la ubicación de un usuario en base a tres puntos de acceso y trazar circunferencias con tiempo en que llega la señal o la potencia recibida del punto al usuario.

Esto produce los puntos de cruce necesarios para determinar la ubicación del usuario, similar al caso del GPS. En la figura 1.7 se muestra un ejemplo de cómo se obtiene la ubicación de un usuario.

Otro ejemplo es el Geo localizador de Google Gears y la especificación de Geo localización de la W3C. Estos tratan de ubicar al usuario no solo por WiFi, sino también por GPS o Cell-ID (Google Inc., 2013). Ambos funcionan con exploradores web mediante código en JavaScript invocando, en el caso de Google Gears, al objeto `google.gears.factory.create('beta.geolocation')` o, en el caso del W3C, a la función de localización:

5.1.10 RFID

Los sistemas llamados RFID (Radio Frequency Identification), utilizados en sistemas de localización en tiempo real. Estos tiene una precisión de alrededor de 10 metros en interiores pero menor en exteriores por la poca densidad de puntos de acceso o el uso de antenas externas. Esta precisión en general es buena y se han desarrollado sistemas para encontrar niños en parques de diversiones con una precisión 2 metros (Moen, 2009).

5.1.11 Beneficios de los Sistemas de Información Geográfica

Un sistema de información geográfica (SIG) integra hardware, software y datos para capturar, gestionar, analizar y mostrar todas las formas de información geográficamente referenciada.

SIG nos permite ver, comprender, cuestionar, interpretar y visualizar datos de muchas maneras que revelan las relaciones, patrones y tendencias en forma de mapas, globos terráqueos, informes y gráficos.

Un SIG ayuda a responder preguntas y resolver problemas al examinar sus datos de una manera que se entienda de manera rápida y fácil de compartir. La tecnología SIG se puede integrar en cualquier marco de sistema de información empresarial.(ESRI, 2014)

5.1.12 Los beneficios de SIG generalmente se dividen en cinco categorías básicas:

5.1.12.1 Ahorro de costes y una mayor eficiencia:

SIG es ampliamente utilizado para optimizar los programas de mantenimiento y los movimientos de las flotas diarias. Implementaciones típicas pueden resultar en un ahorro de 10 a 30 por ciento en los gastos operativos mediante la reducción del uso de combustible y tiempo del personal, la mejora del servicio al cliente, y la programación más eficiente.

5.1.12.2 Mejor Toma de Decisiones:

GIS es la tecnología de go-to para la toma de mejores decisiones sobre ubicación. Los ejemplos más comunes incluyen la selección de bienes raíces sitio, ruta / selección de corredores, planes de evacuación, la conservación, la extracción de recursos naturales, etc. Tomar decisiones correctas respecto a la ubicación es fundamental para el éxito de una organización.

5.1.12.3 Mejora de la comunicación

Mapas y visualizaciones basadas en SIG de gran ayuda en la comprensión de las situaciones y en la narración. Son un tipo de lenguaje que mejora la comunicación entre los diferentes equipos, departamentos, disciplinas, sectores profesionales, las organizaciones y el público.

5.1.12.4 Mejor de registros

Muchas organizaciones tienen la responsabilidad primordial de mantener los registros de autoridad sobre la situación y el cambio de la geografía. SIG proporciona un marco sólido para la gestión de este tipo de registros con soporte de transacciones completa y herramientas de reporte.

5.1.12.5 Gestión Geográficamente

SIG se está convirtiendo en esencial para entender lo que está sucediendo y lo que sucederá en el espacio geográfico. Una vez que entendemos, podemos prescribir la acción. Este nuevo enfoque de la gestión-gestión geográficamente está transformando la forma en que operan las organizaciones.

5.1.13 ¿Qué se puede hacer con los SIG?

5.1.13.1 Localizar en un mapa.

Cartografía de dónde están las cosas te permite encontrar lugares que tienen las características que usted está buscando.

5.1.13.2 Mapa de Cantidades.

La gente se asigna cantidades para encontrar lugares que cumplen con sus criterios y tomar medidas. Empresa de ropa para niños puede ser que desee para encontrar los códigos postales con muchas familias jóvenes con ingresos relativamente altos. Funcionarios de salud pública puede ser que desee asignar el número de médicos por cada 1.000 habitantes en cada sección censal para identificar cuáles son servidos adecuadamente las zonas, y que no lo son.

5.1.13.3 Mapa de densidad.

Un mapa de densidad le permite medir el número de funciones utilizando una unidad de superficie uniforme, de modo que pueda ver claramente la distribución. Esto es especialmente útil cuando se asignan áreas, tales como extensiones o condados del censo, que varían mucho en tamaño. En los mapas que muestran el número de personas por sección censal, las extensiones más grandes pueden tener más personas que las pequeñas. Pero algunos tramos más pequeños pueden tener más personas por milla cuadrada, una densidad más alta.

5.1.13.4 Encuentre lo que está dentro.

Utilice GIS para controlar lo que está pasando y tomar medidas específicas por asociación de lo que hay dentro de un área específica. Por ejemplo, un fiscal de distrito supervisaría arrestos relacionados con las drogas para

determinar si una detención es a 1.000 pies de una escuela-en tal caso, se aplican penas más severas.

5.1.13.5 Encuentra qué hay cerca.

GIS puede ayudar a encontrar lo que está ocurriendo dentro de una distancia establecida de una característica mediante mapeo lo que hay cerca.

5.1.13.6 Cambio de Mapa.

Asigne el cambio en un área de anticipar las condiciones futuras, decidir sobre un curso de acción, o para evaluar los resultados de una acción o política. Mediante la cartografía de dónde y cómo se mueven las cosas en un período de tiempo, se puede obtener una idea de cómo se comportan. Por ejemplo, un meteorólogo podría estudiar las trayectorias de los huracanes para predecir dónde y cuándo podrían ocurrir en el futuro.

5.2 Sistema de Información Geográfica Escogida: Google Maps

Google Maps es un servicio de mapas web gratuito basado en navegador proporcionado por Google con muchas características de gran alcance y fácil de usar. No sólo que el mapa geográfico básico de un área de muestra, sino que también proporciona información local de negocio, información de

contacto, direcciones de conducción, planificador de rutas e imágenes de satélite de la zona. En algunas ciudades, el servicio de tránsito también está disponible en las rutas de transporte público se muestran.(Varinder Taprial, 2011)

5.2.1 Google Maps

El API de Google Maps(API de Google Maps, 2013) te permite añadir mapas interactivos fáciles de utilizar a tu sitio web público, con lo que se mejora la experiencia de tus clientes permitiéndoles encontrar exactamente lo que necesitan de tu organización. Por eso es el API de mapas más popular del mundo. En los casos en los que Google Maps desempeña un papel importante en los sitios web públicos, o se incorpora como parte de una aplicación interna para los empleados, las organizaciones requieren una aplicación que esté a la altura, que sea capaz de gestionar grandes volúmenes y que ofrezca la asistencia necesaria.

El API de Google Maps es un servicio gratuito que te permite insertar Google Maps en tus aplicaciones para móviles o en tus páginas web de acceso gratuito.

Google Maps dispone de un amplio conjunto de APIs que te permiten trasladar la gran funcionalidad y la utilidad diaria de Google Maps a tu propio sitio web y a tus propias aplicaciones, así como superponer tus datos.

La versión 3 de esta API está especialmente diseñada para proporcionar una mayor velocidad y que se pueda aplicar más fácilmente tanto a móviles como a las aplicaciones de navegador de escritorio tradicionales.

El API proporciona diversas utilidades para manipular mapas (como la de la página <http://maps.google.com>) y para añadir contenido al mapa mediante diversos servicios, permitiéndote crear sólidas aplicaciones de mapas en tu sitio web.

La versión 3 del API de JavaScript de Google Maps es un servicio gratuito disponible para cualquier sitio web que sea gratuito para el consumidor.

5.2.2 Entre las cosas que se pueden hacer con Google Maps están

- Insertar una imagen de Google Maps o una imagen panorámica de Street View en tu página web.
- Hacer que los usuarios lleguen del punto A al punto B. Hay muchas formas de insertar rutas de Google Maps en tus aplicaciones. Gracias al API de matriz de distancia, los usuarios pueden encontrar las mejores rutas en coche y conocer el tiempo necesario para llegar a su destino (haz una prueba ahora con la demostración interactiva de la izquierda). Utiliza Street View en el API de Google Maps para que los usuarios tengan una visión global de su destino.
- Incorporar servicios personalizados. Envía alertas, habilita servicios, crea aplicaciones de delimitación geográfica y mucho más con el API de Google Latitude, que te permite crear experiencias personalizadas para usuarios que compartan su ubicación contigo.

- Crea aplicaciones basadas en la ubicación
- Crea mapas para aplicaciones móviles. Crea aplicaciones de alto rendimiento que funcionen en distintos dispositivos móviles.
- Visualiza datos geoespaciales. Crea imágenes en 3D con el API de Google Earth, mapas de calor en Google Fusion Tables y mucho más.
- Personaliza tus mapas. Crea mapas personalizados que destaquen tus datos, tus imágenes y tu marca.
- Servicios web. Utiliza solicitudes de URL para acceder a información de lugares, de elevación, de direcciones y de codificación geográfica.
- API de Google Earth. Lleva a los usuarios a cualquier lugar del mundo sin salir de la página.
- Visualizar datos geoespaciales. Crea visualizaciones interactivas de tus datos geoespaciales.
- Más allá de los marcadores de mapas. Crea sólidas aplicaciones de mapas con la versión 3 del API de JavaScript de Google Maps, un API que proporciona el framework necesario para añadir una serie de recursos y de contenidos únicos a la interfaz de Google Maps.
- Añade contexto de Google Places a tu aplicación. Potencia tu aplicación basada en la ubicación con el API de Google Places, que se puede utilizar para encontrar lugares cercanos en un gran número de categorías. El API de autocompletado de Google Places ayuda a los usuarios a encontrar lo que buscan más rápido sugiriéndoles lugares cercanos mientras escriben.

5.2.3 Diferentes APIs

Las API de Google Maps proporcionan a los desarrolladores diversas formas de insertar Google Maps en páginas web. Además, permiten un uso sencillo y una amplia personalización. Actualmente se ofrecen varias API:

- API de JavaScript de Google Maps
- API de Google Maps para Flash
- API de Google Static Maps

En función de tus necesidades, puedes utilizar únicamente una de estas API o una combinación de varias de ellas.

5.2.4 Cobertura

El equipo de Google Maps publica constantemente nuevos datos de mapas y mejora la cobertura internacional.

La versión 3 del API de JavaScript de Google Maps se ha desarrollado para ofrecer servicios para dispositivos móviles y es apropiada para aplicaciones de navegador destinadas tanto a ordenadores de sobremesa como a dispositivos que incluyan un navegador web con implementación de JavaScript completa como, por ejemplo, el iPhone de Apple.

Para incorporar Google Maps a una aplicación de Android, utiliza la biblioteca externa de Google Maps para Android.

Para incorporar Google Maps a una aplicación nativa de iPhone, utiliza el framework MapKit para iPhone de Apple.

5.2.5 Uso Comercial

Puedes utilizar el API de Google Maps siempre que tu sitio sea accesible desde el punto de vista general a los clientes de forma gratuita. Por ejemplo, si tu sitio web recibe ingresos a través de la publicidad, es probable que esté dentro de las Condiciones de servicio del API de Google Maps. Si cobras a aquellas personas que desean incluir información en tu mapa (por ejemplo, anunciar una casa en venta), pero muestras esa información mediante el API de Google Maps en una sección de acceso gratuito de tu sitio, también cumples las Condiciones de servicio del API de Google Maps.

Sin embargo, hay usos comerciales que no están permitidos. Por ejemplo, si tu sitio cumple alguna de las siguientes condiciones, tendrás que adquirir la licencia del API de Google Maps for Business correspondiente:

- El sitio solo está disponible para clientes de pago.
- Solo se puede acceder al sitio a través de una red intranet o desde dentro de la empresa.
- Tu aplicación está relacionada con envíos comerciales, gestión de vehículos, seguimiento de activos de la empresa u otras aplicaciones similares.

Recuerda que Google se reserva el derecho de suspender o finalizar el uso del API de Google Maps en cualquier momento, por lo que te recomendamos que leas detenidamente las Condiciones de servicio.

5.2.6 Límites de uso

Los sitios web y las aplicaciones que utilizan el API de Google Maps pueden generar de forma gratuita hasta 25.000 cargas de mapas al día por cada servicio.

Si tu aplicación sobrepasa estos límites de uso, deberás realizar una de las siguientes acciones para poder seguir utilizando el API de Google Maps en tu aplicación:

- modificar la aplicación del API de Google Maps para que el número de cargas de mapas diarias sea inferior al límite de uso correspondiente a cada uno de los servicios que utilice la aplicación,
- habilitar la facturación automática de las sobrecargas de mapas en la consola de las API de Google,
- adquirir una licencia del API de Google Maps for Business.

Para poder ofrecer también el servicio a los sitios que experimentan aumentos de tráfico a corto plazo, los límites de uso solo se aplican a un sitio determinado una vez que ese sitio ha excedido el límite durante más de 90 días consecutivos.

Las aplicaciones sin ánimo de lucro y aquellas que Google, según su propio criterio, considere de interés general no estarán sujetas a estos límites de uso. Por ejemplo, no se aplicarán límites de uso a un mapa de ayuda humanitaria aunque una entidad comercial se haya encargado de su desarrollo o alojamiento. Por otro lado, recomendamos a las organizaciones sin ánimo de lucro que cumplan los requisitos que soliciten una licencia del

API de Google Maps for Business a través del programa Google Earth Solidario, que ofrece numerosas ventajas.

Para consultar una definición precisa de las "cargas de mapas", accede a esta página. Para obtener información detallada sobre los precios, accede a esta página.

Los límites de uso del API de Google Maps solo se aplican a los siguientes servicios del API de Google Maps:

- Versión 2 del API de JavaScript de Google Maps
- Versión 3 del API de JavaScript de Google Maps
- API de Google Maps para Flash
- API de imágenes de Google Maps
 - API de Google Static Maps
 - API de imágenes de Google Street View

El API de Google Earth y las API nativas de Google Maps para plataformas móviles como iOS y Android no se ven afectadas. Estos límites tampoco afectan al uso de las funciones integradas de Google Maps ni de otros productos de Google que cuenten con una función integrada que incluya un mapa.

5.2.7 Costos

Las aplicaciones que generen un volumen de cargas de mapas inferior a los límites de uso pueden utilizar el API de Google Maps de forma gratuita siempre que cumplan los requisitos de las Condiciones de servicio del API de Google Maps.

Si superas los límites de uso de un servicio del API de Google Maps durante 90 días consecutivos y decides adquirir sobrecargas de mapas online mediante la consola de las API de Google, el uso excesivo del servicio correspondiente se cobrará en base a las tarifas que aparecen a continuación. Ten en cuenta que si tu sitio genera continuamente un gran volumen de cargas de mapas diarias, es posible que sea más rentable adquirir una licencia del API de Google Maps for Business.

Tabla 5-1

Servicio	Límite de uso (diario)	1.000 sobrecargas diarias (en dólares estadounidenses)
Versión 2 del API de JavaScript de Google Maps	25.000	1,00 USD
Versión 3 del API de JavaScript de Google Maps	25.000	0,50 USD
API de Google Static Maps	25.000	0,50 USD
API de imágenes de Street View	25.000	0,50 USD

5.3 Google Maps for Business

El API de Google Maps for Business ofrece funciones mejoradas y una asistencia más completa a organizaciones que añaden mapas a sus

aplicaciones para móviles, a sus sitios web de pago o a sus sitios web internos.(API de Google Maps for Business, 2013)

El API de Google Maps for Business ofrece a tu empresa asistencia para las necesidades relativas a las aplicaciones de creación de mapas. El API de Google Maps for Business utiliza la misma base de código que la versión estándar del API de Google Maps pero, además, ofrece las siguientes funciones y ventajas adicionales:

Mayor capacidad de solicitudes de servicio, como geo codificación,

Condiciones y términos de uso adaptados al sector,

Opciones de asistencia y servicio con un acuerdo de servicios (SLA) completo,

Compatibilidad con aplicaciones de intranet dentro de la empresa,

Control sobre los anuncios en los mapas.

Para poder acceder a estas funciones adicionales del API de Google Maps for Business, es necesario que registres tu sitio web.

Ciente ID

Para acceder a las funciones especiales del API de Google Maps for Business, deberás proporcionar una *identificación de cliente* cuando accedas a cualquiera de los servicios o de las bibliotecas del API. Cuando te registres en el API de Google Maps for Business, recibirás esta *identificación de cliente* del servicio de asistencia para empresas de Google. Todas las ID de cliente comienzan por un prefijo gme-.

Esta identificación de cliente no es una clave. Solo funcionara desde las URL que autorices, por lo que no tienes que preocuparte por mantenerla en secreto.

5.3.1 Determinación del uso de la aplicación

El API de Google Maps for Business te permite determinar y analizar la actividad de tu aplicación mediante la identificación de cliente. Para obtener información sobre cómo interactúan los clientes con tu aplicación de mapas, puedes activar la opción de Google Analytics para el API de Google Maps for Business.

5.3.2 Google Analytics para el API de Google Maps for Business

Gracias a Google Analytics para el API de Google Maps for Business puedes aprender cómo interactúan los usuarios con tus mapas para que puedas mejorar tu aplicación en función de estadísticas basadas en un uso real. Google Analytics para el API de Google Maps for Business responde preguntas como las siguientes:

1. ¿Cuáles son el nivel de zoom y el tipo de mapa más comunes?
2. ¿Para qué países solicitan los usuarios resultados de codificación geográfica?
3. ¿Con qué frecuencia se utilizan los servicios de JavaScript?

Google Analytics se puede habilitar de forma sencilla sin tener que realizar cambios en tu código actual. Todos los datos se acumulan y son anónimos

para proteger la privacidad de los usuarios. Actualmente, se dispone de informes para la versión 3 del API de JavaScript (versión 3.7 y versiones superiores) y para servicios relacionados.

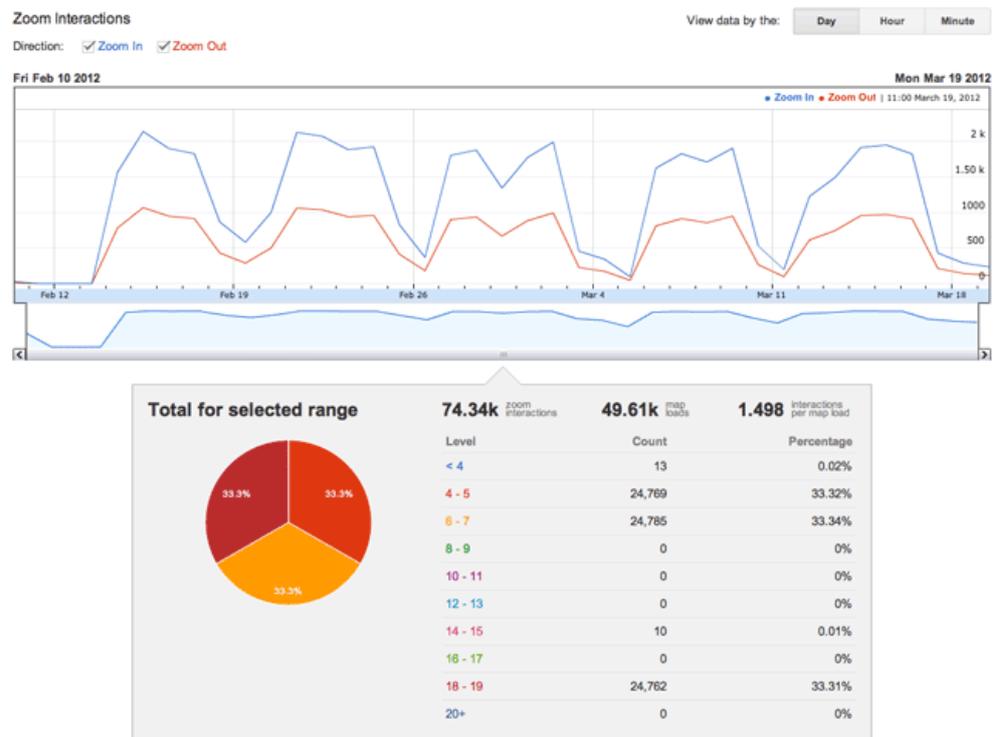


Figura 5-1 El informe sobre interacciones de zoom muestra que la mayoría de los usuarios prefieren utilizar niveles de zoom entre 4 y 7.

5.3.3 Asistencia

El API de Google Maps for Business incluye asistencia especial dedicada a empresas. Fundamentalmente, se ofrece asistencia a través del portal de asistencia para empresas de Google. Una vez que obtengas una licencia para el API de Google Maps for Business, recibirás información de contacto de

un representante del servicio de asistencia para empresas del API de Google Maps.

<https://developers.google.com/maps/documentation/business/faq?hl=es>

En el API de Google Maps for Business se incluyen los productos de Google que se indican a continuación. Las aplicaciones que utilicen estos productos estarán cubiertas por el acuerdo de servicios del API de Google Maps for Business y podrán disponer de asistencia técnica:

- Versión 3 del API de JavaScript de Google Maps
- Versión 2 del API de JavaScript de Google Maps
- API de Google Maps para Flash
- API de Google Static Maps
- API de imágenes de Google Street View
- API de Google Earth
- API de codificación geográfica de Google
- API de elevación de Google
- API de rutas de Google
- API de matriz de distancia de Google
- API de Google Places (si se habilita de forma específica para servir de soporte al API de Google Maps for Business). La biblioteca de Google Places de la versión 3 del API de JavaScript de Google Maps se debe habilitar de forma específica para que estén cubiertas.

5.3.4 Límites de uso

Tabla 5-II

Servicio web	Límites
Rutas	100.000 solicitudes diarias y un máximo de 23 hitos por solicitud
Elevación	100.000 solicitudes diarias, un máximo de 512 puntos en cada solicitud y un máximo de 100.000 puntos diarios
Codificación geográfica	100.000 solicitudes diarias
Matriz de distancia	100.000 elementos cada 24 horas, un máximo de 625 elementos en cada consulta y un máximo de 1.000 elementos cada diez segundos
<i>Todos los servicios web</i>	Existe un límite de diez solicitudes por segundo y por servicio web. El API de Google Places cuenta con los límites de uso independientes.

Estos límites se aplican a todos los ID de cliente del API de Google Maps for Business y se comparten en todas las aplicaciones y en las direcciones IP que utilizan un ID de cliente determinado.

Si necesitas límites de uso más elevados, ponte en contacto con el administrador de cuentas del API de Google Maps for Business para obtener información de compra.

Si has adquirido el API de Google Places como parte de tu contrato del API de Google Maps for Business, los límites del API de Google Places

aparecerán en la sección de límites de la consola de las API. Las solicitudes del API de Google Places no están limitadas.

Los clientes del API de Google Maps for Business disponen de los siguientes límites de uso para las solicitudes de codificación geográfica realizadas desde las API de JavaScript o para Flash:

Tabla 5-III

Servicio	API utilizada	Límites
Codificación geográfica	Versión 3 del API de JavaScript Versión 2 del API de JavaScript	La codificación geográfica tiene una asignación diaria y un límite que se aplica por sesión de usuario, independientemente del número de usuarios que compartan una dirección IP determinada. Este servicio no se debe utilizar para operaciones de codificación geográfica de lotes. Para ello, puedes utilizar el API de codificación geográfica.
	API de Google Maps para Flash	15.000 solicitudes y direcciones IP diarias

Los clientes del API de Google Maps for Business disponen de los siguientes límites de uso para las API de imágenes de Google Maps:

Tabla 5-IV

Servicio	Límites
API de Google Static Maps	<p>Las solicitudes se aplican al número total de visitas a la página adquiridas con la licencia del API de Google Maps for Business.</p> <p>El tamaño máximo de la imagen es de 2.048x2.048 píxeles. Esto se calcula multiplicando el valor size por el valor scale, como se explica más adelante.</p>
API de imágenes de Street View	<p>Las solicitudes se aplican al número total de visitas a la página adquiridas con la licencia del API de Google Maps for Business.</p> <p>El tamaño máximo de la imagen es de 2.048x2.048 píxeles.</p>

Las imágenes del API de Google Static Maps pueden tener un tamaño de hasta 2.048x2.048 píxeles. Debido a que el valor `scale` en una solicitud de mapa estático afecta al número de píxeles que se devuelven, el valor máximo permitido para `size` varía según el nivel de escala:

Tabla 5-V

Escala	Tamaño máximo	Devuelve
1	2048x2048	2.048x2.048
2	1024x1024	2.048x2.048
4	512x512	2.048x2.048

5.4 Google Maps Engine

5.4.1 Visualiza tus datos geoespaciales en la nube

Google Maps Engine es una solución geoespacial revolucionaria que te permite publicar tus datos de mapas en la plataforma de mapas segura basada en la nube de Google. Compártelos de una forma fácil y rápida a través de dispositivos Android, Google Earth, Google Maps y las API de Google Maps. Se reducirán los gastos de TI de forma significativa y se necesitará menos tiempo para el mantenimiento, los ajustes y las actualizaciones del software y de los servidores. La plataforma de nube de Google permite gestionar fácilmente los aumentos del tráfico de usuarios (por ejemplo, el incremento de tráfico que se produce después de un desastre natural). (Google Maps Engine API, 2014)

5.4.2 Cree y comparta mapas de Google Maps personalizados a escala

Use Google Maps Engine para llevar la potencia de Google Maps a su organización. Agregue una capa con sus datos a un mapa de base de Google y cree sus propios mapas y aplicaciones geoespaciales con nuestra infraestructura fiable y de comprobada eficiencia a nivel mundial.

Google Maps Engine y ofrece una solución de plataforma y una aplicación profesional y permite un amplio abanico de creación de mapas. Obtenga más información y decida cuál es la edición que mejor se adapta a sus demandas.

5.4.3 Capacidades de la plataforma de Google Maps Engine

5.4.3.1 Cartografía

Cree mapas de forma intuitiva gracias a la tecnología fácil de usar y popular de Google Maps.

Diseñe mapas integrales con varias capas para optimizar el contenido preciso y detallado de los mapas de Google.

5.4.3.2 Uso compartido

Comparta mapas personalizados de Google Maps en cualquier momento y lugar y publíquelos a escala.

Gracias a la compatibilidad entre los distintos productos de Google, como la Búsqueda de Google, Google Earth y más de 1 millón de mapas que usan la tecnología de la API de Google Maps, el mundo podrá descubrir sus mapas.

5.4.3.3 Diseño

Diseñe aplicaciones con capacidad geoespacial sobre la nube rápida y fiable de Google.

Desarrolle en todo tipo de plataforma (web, Android, iOS, servidores, software GIS) para crear innumerables aplicaciones, como buscadores de tiendas, mapas de la comunidad o aplicaciones para administrar activos.

5.4.3.4 Administre sus mapas, pero no los servidores

Las capacidades de la plataforma central de Google Maps Engine (crear mapas, compartir y diseñar) son posibles gracias a la infraestructura en la nube rápida y confiable de Google. Ofrezca un acceso seguro a mapas entre distintas plataformas para programadores y un número ilimitado de usuarios. Administre sus mapas con la infraestructura en la nube de Google y permita que los usuarios accedan a datos de mapas fiables que siempre están actualizados.

5.5 Google Maps Engine Pro

Con Google Maps Engine Pro se puede liberar la información que contienen sus datos y examínala de formas más reveladoras y con más posibilidades de acción. Se puede importar datos desde una hoja de cálculo

y crear mapas sofisticados con varias capas. Con Maps Engine Pro, se puede usar mapas para:

Colaborar: tomar mejores decisiones con un mayor nivel de colaboración.

Compartir: acceder a mapas y compartirlos desde cualquier dispositivo móvil o computadora.

Personalizar: crear mapas adaptados a las necesidades de su empresa.

Se puede integrar Google Maps en sus aplicaciones personalizadas y habilite los servicios web y la visualización mediante las API de Maps Engine.

También es posible transferir sus datos de GIS a la nube y ofrezca herramientas fiables de creación de mapas a sus empleados sin la necesidad de una inversión considerable en GIS. Colabore con los equipos y tome decisiones fundamentadas observando los datos en una interfaz de mapas que ya conoce. Se puede aplicar diferentes estilos y edite mapas con varias capas gracias a herramientas cartográficas muy simples.

Finalmente se puede distribuir el contenido de sus mapas a todo el mundo con Google Maps Engine y permitir que los participantes clave accedan fácilmente a sus mapas en línea a través de Google.

5.6 Google Maps Places

Potencia tu aplicación basada en la ubicación con la API de Google Places, que se puede utilizar para encontrar información detallada sobre sitios

pertenecientes a un gran número de categorías. La API de Google Places, que utiliza la misma base de datos que Google Maps y Google+ Local, incluye más de 95 millones de empresas y sitios de interés que se actualizan con frecuencia mediante fichas de empresa verificadas por propietarios y contribuciones sometidas a un proceso de moderación por parte de los usuarios. La API de Google Places cuenta con una comunidad de desarrolladores muy activa. (Google Places API, 2014)

5.6.1 Búsquedas

La API de Google Places permite buscar información detallada sobre sitios en diversas categorías (por ejemplo, en establecimientos, sitios de interés destacados, ubicaciones geográficas, etc.). La API de Google Places también está integrada en la API de Google Maps como una biblioteca JavaScript.

5.6.2 Eventos

La API de Google Places es compatible con los eventos (que se definen como cualquier tipo de promoción, acto o reunión de carácter público o privado) y permite buscar eventos que se desarrollen en ubicaciones incluidas en el servicio Places y registrar visitas a esos eventos.

5.6.3 API de autocompletado de Google Places

La API de autocompletado de Google Places muestra información sobre sitios basada en términos de búsqueda de texto y se puede utilizar para proporcionar funciones de autocompletado para búsquedas geográficas basadas en texto.

5.6.4 API de Google Places

Accede a información sobre establecimientos, ubicaciones geográficas o puntos de interés importantes.

5.6.5 Descripción general

Las funciones de la biblioteca de JavaScript de Google Places permiten que una aplicación busque lugares (definidos en esta API como establecimientos, ubicaciones geográficas o puntos de interés destacados) incluidos en un área definida como, por ejemplo, los límites de un mapa o los alrededores de un punto fijo.

5.6.6 Tipos de sitios admitidos

En esta página se incluye una lista de los valores admitidos por la propiedad `types` de la API de Google Places. La primera lista incluye los tipos que se pueden utilizar para hacer búsquedas y para añadir sitios. La segunda lista

incluye tipos adicionales que pueden obtenerse en los resultados de las búsquedas de sitios.

5.6.7 Tipos admitidos para buscar y añadir sitios

Puedes utilizar los siguientes valores en el filtro types para realizar búsquedas de sitios y para añadir sitios.

Accounting	establishment	moving_company
Airport	finance	museum
amusement_park	fire_station	night_club
aquarium	florist	painter
art_gallery	food	park
atm	funeral_home	parking
bakery	furniture_store	pet_store
bank	gas_station	pharmacy
bar	general_contractor	physiotherapist
beauty_salon	grocery_or_superm	place_of_worship
bicycle_store	arket	plumber
book_store	gym	police
bowling_alley	hair_care	post_office
bus_station	hardware_store	real_estate_agency
cafe	health	restaurant
campground	hindu_temple	roofing_contractor
car_dealer	home_goods_store	rv_park
car_rental	hospital	school
car_repair	insurance_agency	shoe_store
car_wash	jewelry_store	shopping_mall
casino	laundry	spa
cemetery	lawyer	stadium
church	library	storage
city_hall	liquor_store	store
clothing_store	local_government_o	subway_station
convenience_store	ffice	synagogue
courthouse	locksmith	taxi_stand
dentist	lodging	train_station
department_store	meal_delivery	travel_agency
doctor	meal_takeaway	university
electrician	mosque	veterinary_care
electronics_store	movie_rental	zoo
embassy	movie_theater	

5.6.8 Tipos adicionales utilizados por el servicio de Google Places

Los tipos indicados a continuación pueden aparecer en los resultados de las búsquedas de sitios, además de los tipos anteriores.

Nota: Estos tipos no se pueden utilizar en el filtro types al hacer una búsqueda de sitios ni en la propiedad types al añadir un sitio.

administrative_area_ level_1	locality	route
administrative_area_ level_2	natural_feature	street_address
administrative_area_ level_3	neighborhood	street_number
administrative_area_ level_4	political	sublocality
administrative_area_ level_5	point_of_interest	sublocality_level_4
colloquial_area	post_box	sublocality_level_5
country	postal_code	sublocality_level_3
floor	postal_code_prefix	sublocality_level_2
geocode	postal_town	sublocality_level_1
intersection	premise	subpremise
	room	transit_station

5.6.9 Conecta personas y lugares

Potencia tu aplicación basada en la ubicación con el API de Google Places, que se puede utilizar para encontrar lugares cercanos en un gran número de categorías. Ayuda a los usuarios a encontrar más rápidamente lo que buscan con el API de autocompletado de Google Places, que sugiere lugares cercanos a medida que escriben los usuarios.

5.6.10 Trabaja con una potente base de datos

Al utilizar la misma base de datos de lugares que usa Google Maps, el API de Google Places muestra más de 50 millones de empresas y puntos de interés. Esta base de datos se actualiza diariamente con contribuciones moderadas por los usuarios y verificadas por los propietarios.

La API de Google Places es un servicio que devuelve información sobre sitios, definidos en la API como establecimientos, ubicaciones geográficas o sitios de interés importantes, mediante solicitudes HTTP.

En las solicitudes de sitios, las ubicaciones se especifican en forma de coordenadas de latitud/longitud.

5.6.11 A continuación se indican los tipos de solicitudes de sitios disponibles.

- Búsquedas de sitios: al realizar estas solicitudes, se obtiene una lista de sitios basada en una cadena de búsqueda o en la ubicación del usuario.
- Detalles de sitios: al realizar estas solicitudes, se obtiene información más detallada sobre un sitio específico, incluidas las opiniones de los usuarios.
- Acciones de sitios: te permiten complementar los datos de la base de datos de Google Places con datos de tu aplicación. Puedes programar eventos, añadir o eliminar sitios o ponderar la clasificación de los

sitios en función de la actividad de los usuarios con los registros de visitas.

- Fotos de sitios: permiten acceder a millones de fotos relacionadas con los sitios almacenados en la base de datos de Google.
- Autocompletado de sitios: se puede utilizar para completar automáticamente el nombre y la dirección de un sitio a medida que se escribe.
- Autocompletado de consultas: se puede utilizar para proporcionar un servicio de predicción de consultas para búsquedas geográficas basadas en texto al ofrecer consultas sugeridas a medida que se escribe.

5.6.12 Políticas para desarrolladores

En esta sección se indican los requisitos específicos de todas las aplicaciones desarrolladas con la API de Google Places o con la API de autocompletado de Google Places.

5.6.13 Requisitos del logotipo

Si tu aplicación muestra datos de la API de Google Places en un mapa, ese mapa debe proceder de Google.

Si tu aplicación muestra datos de la API de Google Places en una página o en una vista en la que no aparezca un mapa de Google, tendrás que incluir el

logotipo "Con la tecnología de Google" junto con esos datos. Por ejemplo, si tu aplicación muestra una lista de sitios en una pestaña y un mapa de Google con esos sitios en otra, en la primera pestaña deberá aparecer el logotipo "Con la tecnología de Google".(Google Places API, 2014)

Tabla 5-VI

Uso sobre un fondo blanco	Uso sobre un fondo que no sea blanco
	

En la página oficial se puede encontrar el archivo ZIP que contiene el logotipo "Con la tecnología de Google" con los tamaños adecuados para aplicaciones de escritorio, Android e iOS. Estos logotipos no se pueden someter a ningún tipo de ajuste de tamaño ni modificación.

5.6.14 Límites de uso

La API de Google Places tiene los siguientes límites de consultas:

- Los usuarios que tengan una clave de API pueden realizar 1.000 solicitudes cada 24 horas.
- Los usuarios que hayan verificado su identidad a través de la consola de las API pueden realizar 100.000 solicitudes cada 24 horas. Se necesita una tarjeta de crédito para realizar la verificación. Para ello, se debe habilitar la facturación en la consola.

5.6.15 Requisitos de las Condiciones de uso y la Política de privacidad

Si desarrollas una aplicación con la API de Google Places, debes incluir unas *Condiciones de uso* y una *Política de privacidad* en tu aplicación conformes con las Condiciones de servicio de las API de Google Maps y Google Earth (sección 9.3):

- Tus *Condiciones de uso* y tu *Política de privacidad* deben estar disponibles públicamente.
- Debes indicar explícitamente en las *Condiciones de uso* de tus aplicaciones que al utilizar tu aplicación, los usuarios se comprometen a cumplir las Condiciones de servicio de Google.
- Debes notificar a los usuarios en tu *Política de privacidad* que estás utilizando las API de Google Maps e incorporar como referencia la Política de privacidad de Google.

El lugar recomendado para colocar tus *Condiciones de uso* y tu *Política de privacidad* dependerá de tu plataforma de aplicaciones.

5.6.16 Aplicaciones para móviles

Si desarrollas una aplicación para móviles, es recomendable proporcionar un enlace a tus *Condiciones de uso* y tu *Política de privacidad* en la página de descarga de tu aplicación en la tienda de aplicaciones correspondiente y en un menú de configuración de la aplicación.

5.6.17 Aplicaciones web

Si desarrollas una aplicación web, es recomendable proporcionar un enlace a tus *Condiciones de uso* y tu *Política de privacidad* en el pie de página del sitio web.

6 Propuesta de Diseño de la Aplicación

6.1 Consideraciones

La entidad principal que nos concierne es el inmueble que puede ser tanto, casa, apartamento o terreno y puede estar en venta o alquiler. La entidad tendrá tanto precio de venta y de alquiler por que es posible que el usuario quiera tener ambas opciones disponibles.

La siguiente tabla muestra la similitud y diferencias entre como son representadas casa, apartamento y terreno

Tabla 6-1

Casa	Apartamento	Terreno
Id	Id	Id
LatLng	LatLng	LatLng
Habitaciones	Habitaciones	
Baños	Baños	
Metros cuadrados	Metros cuadrados	
Precio de venta	Precio de venta	Precio de venta
Renta de alquiler(mensual)	Renta de alquiler(mensual)	
Metros cuadrados de Terreno		Metros cuadrados de Terreno

Como en visionamos Inmuebles ABC esta va a funcionar con Google Maps en la nube y no una base de datos privada pero esto lleva a las siguientes consideraciones.

El sitio web y móvil pueden generar gratuitamente 25,000 cargas de mapas al día por cada servicio si sobre pasa de esto durante 90 días seguidos habría que habilitar facturación automática o adquirir una licencia del API de Google Maps for Business.

La facturación automática facturaría 0.50 dólares por cada 1,000 sobre cargas lo que puede ser rentable al corto plazo si uno no cree que el número de usuario va a aumentar mucho más de ahí

Se usará Google Maps Engine que permite compartir la información con Android y las APIs de Google Maps. Se reducirán los gastos de TI de forma significativa y se necesitará menos tiempo para el mantenimiento, los ajustes y las actualizaciones del software y de los servidores.

Se pueden crear mapas y ser guardados en la nube lo que se utilizará para los layers de utilidades que serán creados con Google Maps Engine y guardados en la nube. El engine nos permite administrar nuestros mapas sin preocuparnos por los servidores. Todo esto es gracias a BigTable el sistema administrador de base de datos privado de Google.

Google Maps Places tiene más de 95 millones de empresas y sitios de interés que se actualizan con frecuencia mediante fichas de empresa verificadas por propietarios y contribuciones sometidas a un proceso de moderación por parte de los usuarios.

La API de Google Places permite buscar información detallada sobre sitios en diversas categorías.

A continuación están las categorías de Google Places que creemos les concierne a cualquier persona buscando comprar, vender o rentar un inmueble:

ATM	cuidado del pelo	Café
Banco	Estación de autobús	Panadería
Bar	Estación de tren	Car Wash
Tienda de licor	Estación de	Iglesia
Salón de belleza	subterráneo	Lugar de adoración

Hospital	Supermercado	Comida a domicilio
Doctor	Colmado	Cine
Salud	Centro comercial	Club nocturno
Policía	Tienda de bienes del hogar	Parque
Embajada	Comida, Restaurante	Estacionamiento
Estación de bomberos	Gasolinera	Farmacia
Tiendas	Gimnasio	Escuela
Tienda de ropa	Lavandería	Universidad
Tienda de departamentos	Comida para llevar	Estadio

Al buscar en Google se puede ver información de los sitios hasta opinión de los usuarios que han estado en él y también se puede acceder a millones de fotos relacionadas con los sitios almacenadas en la base de datos de Google. Los desarrolladores deben recordar buscar por el logotipo de Google Maps Places API que por legalidad debe ser colocado en cualquier aplicación que la utiliza.

7 Diseño de la Aplicación Web y Móvil

7.1 Búsqueda en Web

The screenshot displays the 'Inmuebles ABC' website interface. At the top, there is a search bar with the URL 'http://www.inmueblesabc.com' and a search button. Below the search bar, there are filters for 'Venta/Renta', 'Nuevo', 'Barato', 'Metros cds', and 'Mas'. The main content area is divided into two sections: a list of property cards on the left and a map view on the right. The property cards list details such as price, location, and features. The map view shows a grid of streets with several red location pins indicating the positions of the properties listed. A yellow highlighted area on the map suggests a specific region of interest.

Property Type	Price	Location	Features
Casa en venta	RD\$ 1,500,000	Av. No. 7 No. 70	4 hdb, 2 baños, 1200 mc
Casa en venta	RD\$ 1,500,000	Av. No. 7 No. 75	4 hdb, 2 baños, 1200 mc
Apartamento en renta	RD\$ 20,000 c/m	Av. 5 No. 15	2 hdb, 15 baños, 800 mc
Apartamento en venta	RD\$ 1,100,000	Av. 4 No. 14	3 hdb, 2 baños, 1000 mc
Casa en venta	RD\$ 2,000,000	Av. 4 No. 20	5 hdb, 3 baños, 1500 mc
Casa en venta	RD\$ 1,300,000	Av. 17 No. 80	3 hdb, 2 baños, 1000 mc
Casa en venta	RD\$ 1,200,000	Av. 17 No. 84	3 hdb, 2 baños, 900 mc

Figura 7-1 Fuente propia.

Esta imagen es un mockup construido en Balsamiq Mockup una aplicación para hacer prototipos y wireframes. Aquí nos presenta como funciona el sistema de búsqueda propuesto.

El usuario puede buscar por provincia y ciudad, elegir cuantos baños deben tener las casas y o apartamentos y cuantos cuartos. A continuación se presentaran en lado izquierdo de la pantalla los inmuebles que cumplen con los criterios de la búsqueda y son señalados en el mapa.

Los detalles que se presentan son, si es una casa en venta, renta o ambas, la dirección por medio del mapa, el precio de venta y o renta, cuantas habitaciones y baños, el tamaño de la casa y el tamaño del terreno en el que la casa se encuentra.

En la propuesta sugerida todo es basado en Google Maps que es la base de datos gratis más amplia en información geográfica y funciona en la nube.

7.2 Agregar un Inmueble en Web

The screenshot shows a web browser window with the URL <http://www.inmueblesabc.com>. The page title is "Inmuebles ABC" and the location is set to "Provincia, Ciudad". A search button labeled "Buscar" is visible. The main form is titled "Inmueble" and contains the following fields:

- Tipo de Inmueble: Casa
- No Habitaciones: 3
- No Baños: 2
- Precio de Venta: 0.00
- Precio de Renta: 0.00
- Metros Cuadrados: 0
- Terreno m. cds: 0

The "Descripción" field contains the text: "Buena casa en un buen vecindario. Hospital de la RD a 50 mts. Departamento de la policía a 70 mts. Avenida Principal a 80 mts. Farmacia Milagros a 200 mts." Below this is a "Subir fotos..." button.

The "Utilidades" section lists the following amenities with checkboxes:

- Hospital de la RD
- Departamento de la policía
- Gasolinera Este
- Farmacia Milagros
- Super mercado Manati
- Avenida Principal
- Estacion de Metro
- Colegio "La Educacion"

An "Agregar" button is located at the bottom of the utility list. To the right of the form is a map showing a street grid with a green shaded area representing the property. A yellow line highlights the property's location on the map. The map includes navigation controls and a home icon.

Figura 7-2 Fuente propia.

En esta imagen mostramos un caso hipotético de registro. Por defecto la pagina se centra en la ubicación de la maquina donde se ha accedido a la

pagina esto es logrado por medio de Wi-Fi, RFID (Radio Frequency Identification) y GPS gracias a los servicios basados en locación. El mapa de google permite al usuario moverse libremente en la superficie del mapa y ver cualquier parte del país y también puede buscar una dirección, sector o provincia por la caja de texto de buscador.

Conforme hayan utilidades cercanas al área escogida esta aparecerán en un grupo de checkbox que permitirá al usuario agregar esta información a la descripción del inmueble que luego el usuario comprador vera y conocerá todas las utilidades que se obtienen al comprar o alquilar una propiedad. Al usuario se le presentaran estas en forma de links que mostrara los objetos InfoWindow que google tiene asociados a estos si los hay.

También presionando el botón Subir fotos se puede subir imágenes internas y externas de la propiedad.

7.3 Búsqueda en Móvil

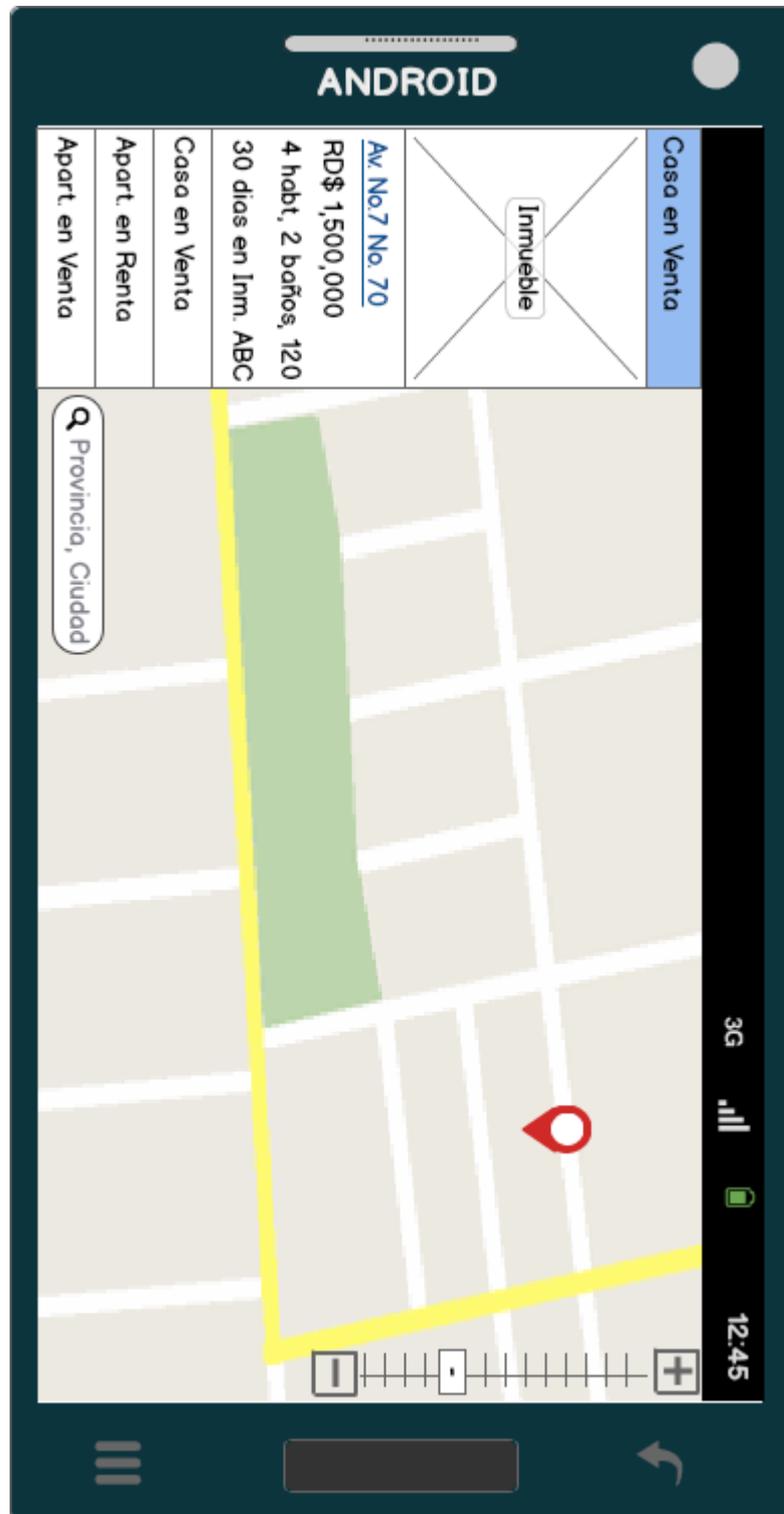


Figura 7-3 Fuente propia.

En esta pantalla vemos un Smartphone android girado a la izquierda que es como vemos que la aplicación funcionaria y no verticalmente y funciona de manera similar a la búsqueda por web. En este caso, probablemente mejor que en el caso web, se utilizara la tecnología de servicios basados en locación para encontrar el Smartphone esto funciona mejor cuando el GPS del celular esta encendido, también se puede utilizar el mapa y la caja de texto de buscar para encontrar los inmuebles que se quieran.

La información es mostrada de manera mas compacta.

7.4 Agregar un Inmueble en Móvil

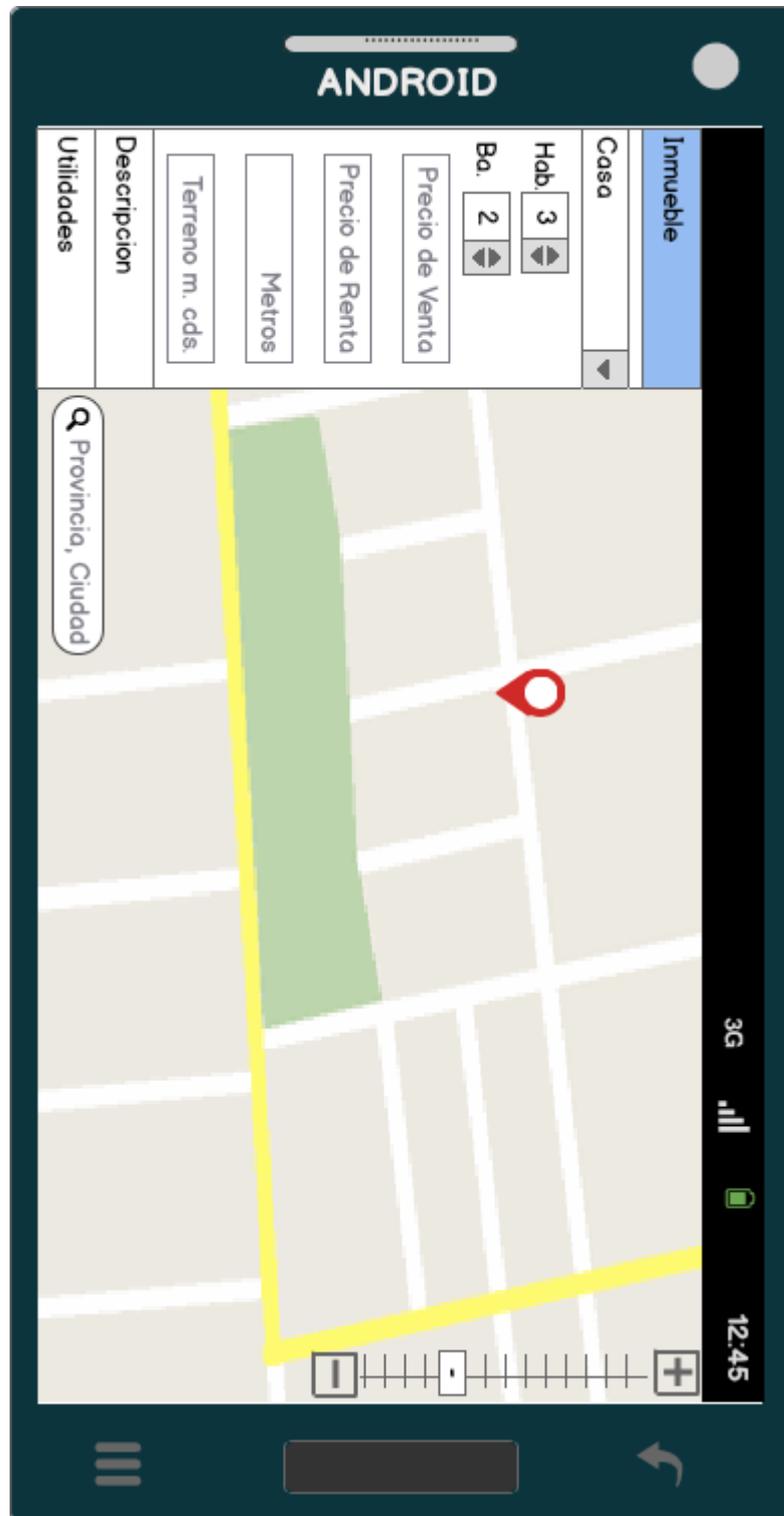


Figura 7-4 Fuente propia

Agregar un inmueble por móvil también es similar al web pero más compacta. Se utiliza por defecto la dirección GPS asumiendo que el inmueble que se va a vender es el en que se esta pero esta se cambia fácilmente navegando por el mapa y usando la búsqueda.

No es necesario escribir dirección solo elegir el punto en LatLng donde se encuentra el inmueble. Como en la versión web el sistema experto compuesto de Google Maps Engine, Google Maps Places, y Google Maps for Business utiliza su base de datos para buscar las utilidades mas cercanas y a partir de esta el usuario elige cuales quiere mostrarlo al posible comprador puede que haya sitios que no quiera decir que están cerca como por ejemplo cementerios o quiera ocultar que algunas utilidades están muy lejos, por ejemplo, el colegio mas cercano a 5 kilómetros.

7.5 Alerta



Figura 7-5 Fuente propia.

Este es un ejemplo de cómo funcionaría la alarma que alerta de que un inmueble como el que uno estaba buscando ha sido agregado al sistema.

Funciona como los Google Alert.

Conclusión

En conclusión el conocimiento es poder y armando a sus usuarios con el conocimiento de los alrededores de sus bienes raíces hasta kilómetros en los alrededores les permite promoverlos de mejor manera que si no lo supieran.

En un principio la única manera de vender era entrando los datos del inmueble y esperar a que alguien se interesara en las características del inmueble. Y del lado del comprador este puede tener una idea vaga de lo que quiere sin pensar tampoco en todas las utilidades cercanas a lo que quiere y luego tomar una decisión que lamenta. El problema en si es que el usuario no era consciente de que no solo compra, vende o alquila un inmueble. Lo que el usuario realmente negocia es accesibilidad en la ciudad.

También concluimos que Google es una presencia poderosa e influyente en los sistemas de información geográfica gracias a sus servicios como Google Maps y Google Earth y proponemos Google Maps y sus diferentes APIs como el camino correcto para implementar esta solución de software.

En el futuro solo podemos esperar que Google Maps mapeara todo el planeta incluyendo esta isla con StreetView y que adoptando ahora esta solución y aprovechando los millones de datos gratis acerca de lugares que

pone a nuestra disposición Inmuebles ABC estará encaminada para el futuro.

Bibliografía

- FISMAT*. (23 de Mayo de 2000). Obtenido de FISMAT:
<http://www.fismat.umich.mx/~elizalde/tesis/node15.html>
- ¿Que es IDE?* (2010). Obtenido de Agencia Federal Sociedad de la
Informacion: <http://www.afesif.gov.ar/IDE.html>
- API de Google Maps*. (31 de Enero de 2013). Recuperado el Agosto de
2014, de Google Developers: <https://developers.google.com/maps/>
- API de Google Maps for Business*. (31 de Enero de 2013). Obtenido de
Google Developers:
<https://developers.google.com/maps/documentation/business/?hl=es>
- Relaciones de BD*. (2013). Obtenido de Nelson Sistema:
<http://nelsonsisistema.webnode.com.ve/base-de-datos/relaciones-de-bd/>
- (2014). Obtenido de SiGIS:
<http://www.sigis.com.ve/index.php/nosotros/aprende-mas>
- Google Maps Engine API*. (4 de Febrero de 2014). Obtenido de Google
Developers:
<http://www.google.com/enterprise/mapsearch/products/mapsengine.html>
- Google Places API*. (10 de Enero de 2014). Obtenido de Google
Developers: <https://developers.google.com/places/policies?hl=es>
- Google Places API*. (10 de Enero de 2014). Obtenido de Google
Developers: <https://developers.google.com/places/policies?hl=es>

- Google Places API*. (10 de Enero de 2014). Obtenido de Google Developers: <https://developers.google.com/places/?hl=es>
- 8Motions. (2009). *OpenCellID API*. Documento en Línea <http://www.opencellid.org/api>.
- Cabello, V. N. (2005). *Introducción a la Bases de Datos Relacionales*. Madrid: Vision Libros.
- CANTONE, D. (2011). *ADMINISTRACION STORAGE Y BACKUPS*. Madrid: RA-MA EDITORIAL.
- Cardoso M., L. I. (2006). *Sistemas de base de datos II. Teoría aplicada para profesores y estudiantes*.
- Castañeda, H. A. (2006). *Proveedor de Servicios Basados en Localización para Dispositivos Móviles*. Documento en Línea http://pisis.unalmed.edu.co/avances/archivos/ediciones/2006/castaneda_etal06.pdf.
- CherryTree. (2000). *Business Intelligence-The Missing Link*.
- Cobo, A. (2008). *Diseño y Programacion de Base de Datos*.
- Conde, K., Nava, L., & Rojas, M. (Septiembre de 2008). Obtenido de Ciclos de Vida: <http://ciclosdevida1.blogspot.com>
- Cotos Yáñez , J. M., & Taboada Gonzál, J. Á. (2005). *Sistemas de información medioambiental*. Netbiblo.
- Daniel Jacobson, G. B. (2012). *APIs: A Strategy Guide*. California: O'Reilly.
- Date, C. (2001). *Introducción a los sistemas de bases de datos*. Mexico: Prentice Hall.

- Date, C. J. (2001). *INTRODUCCION A LOS SISTEMAS DE BASES DE DATOS (7ª ED.)*. Mexico: Pearson Educación.
- Dave, P. (9 de Junio de 2007). *Journey to SQL Authority with Pinal Dave*.
Obtenido de Journey to SQL Authority with Pinal Dave:
<http://blog.sqlauthority.com/2007/06/09/sql-server-spatial-database-definition-and-research-documents/>
- Dubretic, M. (26 de 4 de 2014). *Oracle vs MySQL vs SQL Server: una comparación entre los Sistemas Gestores de Bases de Datos Relacionales más Populares*. Obtenido de UdeMy:
<https://www.udemy.com/blog/es/oracle-vs-mysql-vs-sql-server-una-comparacion-entre-los-sistemas-gestores-de-bases-de-datos-relacionales-mas-populares/>
- E. Rhoades, R. (s.f.). *Desarrollo con identidad: comunidad, cultura, y sustentabilidad en los Andes*.
- El-Rabbany, A. (2002). *GPS The Global Positioning System*. Norwood: Artech House, Inc.
- ESRI. (1990). *Environmental Systems Research Institute, inc. Understanding GIS*.
- ESRI. (2014). *What is GIS?* Obtenido de Esri: <http://www.esri.com/what-is-gis>
- ESRI. (2014). *What is GIS?* Obtenido de Esri: <http://www.esri.com/what-is-gis/overview>

- Fernández Alarcón, V. (2006). *Desarrollo de sistemas de información: una metodología basada en el modelado*. Barcelona: Univ. Politèc. de Catalunya.
- G. Avola, J. R. (2012). *Smashing Mobile Web Development*. New Jersey, Estados Unidos: John Wiley & Sons.
- G. Pedraz, C. (2009). *Tecnología wifi como medio de localización en interiores*. Documento en Línea <http://www.dicyt.com/noticias/tecnologia-wifi-como-medio-de-localizacion-en-interiores>.
- GEOSAT. (2013). *Satelites*. Obtenido de GEOSAT Comunidad de Imágenes Satelitales: http://dgpcfadu.com.ar/2013/1_cuat/vm28/tp/satelites.html
- González, V. y. (2007). *Mercadotecnia basada en localización*. Dirección Estratégica ITAM. Documento en Línea <http://direccionestrategica.itam.mx/Administrador/Uploader/materi>.
- Google Inc. (2013). *Geolocation API*. Documento en Línea http://code.google.com/intl/es/apis/gears/api_geolocation.html.
- Google Inc. (2013). *Google Maps*. California: Documento en Línea <http://maps.google.com/>.
- Google, Inc. (2006). *Bigtable: A Distributed Storage System for Structured Data*.
- GPS. (2014). Obtenido de GPS: <http://www.gps.gov/spanish.php>
- Iturbe, A., S. Nchez, L., & Ch as, L. (2009). *Consideraciones Conceptuales Sobre Los Sistemas de Informaci N Geogr Fica*.

- Jardine, D. A. (1977). *The ANSI/SPARC DBMS Model*. Montreal: North-Holland Pub. Co.
- Kofler, M. (2005). *The Definitive Guide to MySQL 5*. Berkely: APRESS.
- Laudon, J. P., & Laudon, K. C. (2012). *Sistemas de Informacion Gerencial*.
- LeBlanc, C., Belbin, J., & MacNeil, J. (1990). *Cartografía de recursos marinos, un manual de introducción*. Food & Agriculture Org.
- Martínez Gens, L. E. (2006). *Tecnologías de Localizacion y Posicionamiento para Servicios Basados en Localizacion (LBS)*. Documento en Línea <http://dialnet.unirioja.es/servlet/articulo?codigo=1983412&orden=65872&info=link>.
- McLeod, R. (2000). *Sistemas de información gerencial*. Ciudad de México: Marisa de Anta.
- Mehta, N. (2008). *Mobile Web Development: Building Mobile Websites, SMS and MMS Messaging, Mobile Payments, and Automated Voice Call Systems with XHTML MP, WCSS, and Mobile AJAX*. Reino Unido: Packt Publishing Ltd.
- Moen, H. L. (2009). *The potential for location-based services with WI-FI RFID tags in citywide wireless networks*. Documento en Línea <http://dx.doi.org/10.1109/ISWCS.2007.4392319>.
- Mozilla Developer Network. (2013). *HTML5*. Obtenido de Mozilla Developer Network: <https://developer.mozilla.org/es/docs/HTML/HTML5>

- Pastor, J. (12 de 3 de 2014). *Desarrollo de aplicaciones móviles (I): así está el mercado*. Obtenido de Xataka Movil: <http://www.xatakamovil.com/mercado/desarrollo-de-aplicaciones-moviles-i-asi-esta-el-mercado>
- Peña Llopis, J. (2006). *Sistemas de información geográfica aplicados a la gestión del territorio*. Madrid: Club Universitario.
- Pérez Navarro, A. (2011). *Introducción a los sistemas de información geográfica y geotelemática*. Barcelona: Carrera edicio, S.L.
- Rafael Camps Paré, L. A. (2005). *Base de datos*. Barcelona: Eureka Media, SL.
- Rashid, O. C. (2008). *Providing location based information/advertising for existing mobile phone users*. Documento en Línea <http://dx.doi.org/10.1007/s00779-006-0121-4>.
- Rolf A. de By, R. A. (2000). *Principles of Geographic Information Systems*. Enschede: The International Institute for AeroSpace Survey and Earth Sciences(ITC).
- Shackles, G. (2012). *Mobile Development with C#: Building Native iOS, Android, and Windows Phone Applications*. Massachusetts, Estados Unidos: O'Reilly Media.
- Silberschatz, A., Korth, H. F., & S., S. (2002). *Fundamentos de bases de datos (4ta ed.)*. Madrid: McGRAW-HILL.
- Stair, R., & Reynolds, G. W. (2000). *Principios de sistemas de información: enfoque administrativo*. México, D. F.: Cengage Learning Editores.

STEPHENS, R. (2009). *DISEÑO DE BASE DE DATOS* . España: ANAYA
MULTIMEDIA.

Varinder Taprial, P. K. (2011). *Google Beyond google*. Bookboon.

Vitt, E., Luckevich, M., & Misner, S. (2003). *BUSINESS INTELLIGENCE:
TECNICAS DE ANALISIS PARA LA TOMA DE DECISIONES
IMPORTANTES*.

VV.AA. (2006). *TECNOLOGIA Y DISEÑO DE BASES DE DATOS*.
España: RA-MA.

Wikipedia. (2014). *Desarrollo web*. Obtenido de Wikipedia:
http://es.wikipedia.org/wiki/Desarrollo_web

ANEXOS



Decanato de Ingeniería e Informática
Escuela de Informática

**DISEÑO DE SISTEMA DE APOYO A LA TOMA
DE DECISIONES CON RESPECTO A LA
COMPRA, VENTA Y RENTA DE INMUEBLES
ABC POR WEB Y MÓVIL UTILIZANDO
SISTEMAS DE INFORMACIÓN GEOGRÁFICA**

**Anteproyecto de la monografía para optar por los títulos de
Ingeniería en Sistemas de Información y Computación**

Sustentantes:

Andrés Estepan Mat. 2008-0116
David Díaz Mat. 2008-0136
Carmen P. Ramírez Mat. 2008-0822

Asesor:

Mitsuteru Nishio

**Distrito Nacional, República Dominicana
Junio, 2014**

Índice de Contenido

Selección del título y definición del tema

Selección del Título

Definición del Tema

Organización del Contenidos

Planteamiento del problema de la investigación

Objetivos de la investigación

Objetivos Generales

Objetivos Específicos

Justificación de la investigación

Justificación Teórica

Justificación Metodológica

Justificación Práctica

Tipos de investigación

Marcos de referencia

Marco Teórico

Marco Conceptual

Marco Espacial

Marco Temporal

Métodos, procedimientos y técnicas usadas en la investigación

Métodos

Procedimientos

Técnicas

Tabla de contenidos

Bibliografía

Selección del Título y Definición del Tema

Selección del Título

Diseño de sistema de apoyo de toma de decisiones con respecto a la compra, venta y renta de inmuebles por web y móvil utilizando sistemas de información geográfica.

Definición del Tema

La propuesta presenta una solución de software para Inmuebles ABC, una empresa relativamente nueva de clasificados de bienes raíces, que funcionaría como un medio para poner en contactos compradores y vendedores de casas, apartamentos, terrenos, etc. Se proponen que el sitio web, y una nueva aplicación móvil utilicen información geográfica al introducir y buscar inmuebles.

Se explicarán los conceptos pertinentes acerca de temas como sistemas de información geográfica, servicios web, aplicaciones móviles, ciclo del desarrollo de software y otros.

Organización de contenidos

Esta investigación estará dividida en de la manera siguiente:

- Se presenta la empresa, la problemática y se da una visión global del propósito del trabajo.
- **Capítulo 1:** Se habla del ciclo de desarrollo de software y de las filosofías principales que se pueden emplear al implementar este plan de solución.
- **Capítulo 2:** Se habla de las bases de datos, sistemas de gestión de base de datos, motores comerciales y OpenSource y comparación de estos, y APIs.
- **Capítulo 3:** Se expone acerca del desarrollo web.
- **Capítulo 4:** Se expone acerca del desarrollo móvil.
- **Capítulo 5:** Se habla de sistemas de información geográfica enfocado al desarrollo web y móvil que nos compete.
- **Capítulo 6:** Se toman algunas consideraciones para el diseño del sistema
- **Capítulo 7:** Se hace el diseño del sistema del plan de solución integrado todo lo discutido en los capítulos anteriores.

Planteamiento del Problema

Hasta ahora cuando los usuarios querían vender o rentar un inmueble en Inmuebles ABC debían registrarlo en la pagina y esperar que otro usuario se interesara en este. Los usuarios generalmente solo describen las características del inmueble, por ejemplo cuántos baños, cuántos cuartos, etc. pero pocas veces describen las utilidades cercanas o solo una grande y obvia y desconocen todas el valor agregado que tiene el inmueble en relación a su locación

Del lado del comprador este puede tener una idea vaga de lo que quiere sin pensar tampoco en todas las utilidades cercanas a lo que quiere y luego tomar una decisión que lamenta. También hasta ahora a tenido que entrar al sitio web de vez en cuando si un inmueble con las características que quiere no esta disponible, no existe la manera en que Inmueble ABC le avise al usuario que un inmueble que le gustaría esta o no disponible.

El problema en si es que el usuario no es consiente de que no solo compra, vende o alquila un inmueble. Lo que el usuario realmente negocia es accesibilidad a otras utilidades a parte del inmueble como los son hospitales, escuelas, departamentos de policía, supermercados, gasolineras, etc.

Objetivos de la Investigación

Objetivo General

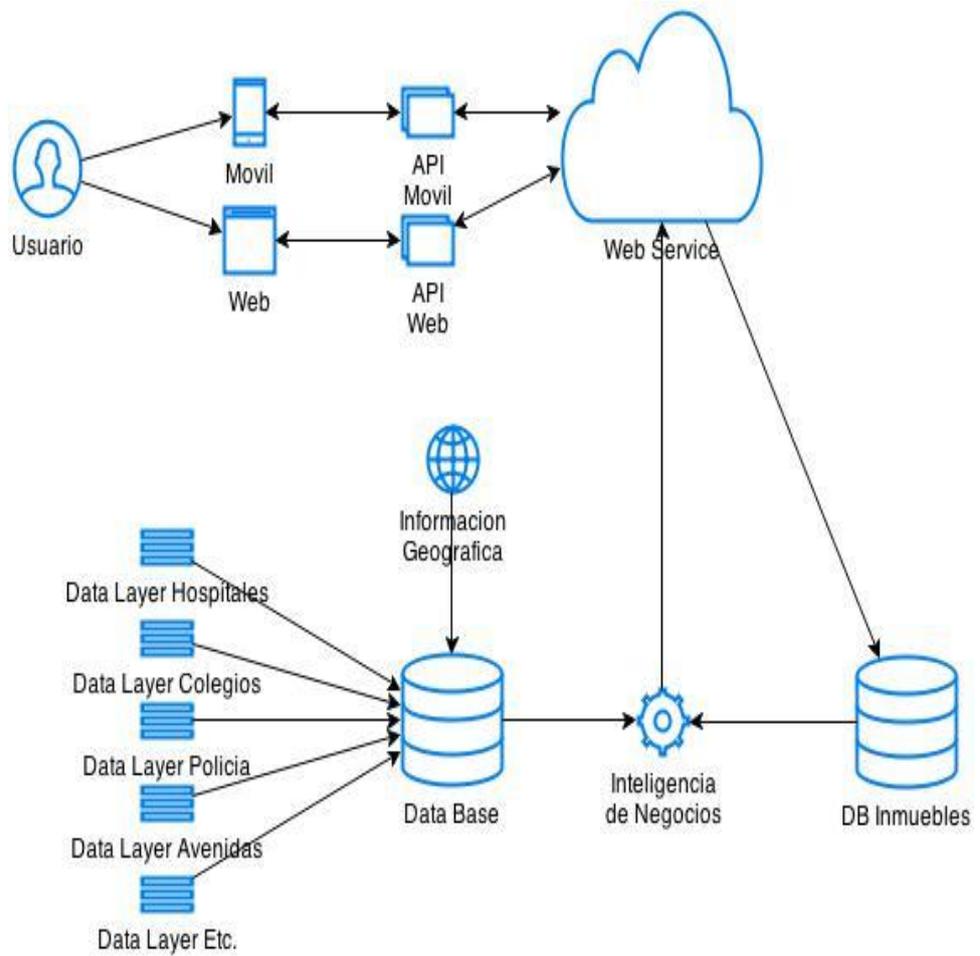
Diseñar un sistema de apoyo de decisiones de compra, venta y renta de inmuebles por medio de la web y móvil en base a las utilidades cercanas a estos.

Objetivos Específicos

Los objetivos específicos de este trabajo son los siguientes:

- Ilustrar el funcionamiento del sistema de apoyo al usuario en compra, venta y renta de inmuebles con un sistema de información geográfica por medio de sitio web y móvil
- Diseñar aplicación web y móvil
- Formular como representar visualmente de la información geográfica derivada de la relaciones de clústeres de utilidades.
- Explicar cómo se realiza descubrimiento de conocimientos relacionando capas de diferentes tipos de utilidades en clústeres
- Describir como se relacionan información geográfica similar en capas de datos.
- Identificar datos a recolectar de información geográfica de utilidades del país.

Diagrama General del Sistema



Fuente propia.

Justificación de la Investigación

Justificación Teórica

Es simple ver hay valor escondido en los bienes raíces de un territorio en relación a las utilidades cercanas a este y es evidente que traer esta información a la luz va a llevar a una mejor toma de decisiones y precios con respecto a la compra, venta y renta de estos inmuebles.

Se diseñará un sistema de información geografía y una aplicación móvil debido a que en el pasado reciente éstas han estado en auge en la agilización de los procesos de los negocios debido a su gran disponibilidad ya que los servicios GPS y Smartphone se han vuelto más baratos.

Justificación Metodológica

Se usará método analítico ya que es una propuesta de solución de software y es a partir del análisis de la situación existente de donde ha venido la idea de la propuesta. También se documentará en el trabajo todo el conocimiento necesario para entender la solución lo que incluye exponer sistemas de información geográfico, aplicaciones web y móvil.

Justificación Práctica

El apoyo en decisiones para la compra, venta y renta de inmuebles pondrá a Inmuebles ABC delante de toda la competencia. Las tecnologías GIS le darán una perspectiva a los compradores y vendedores que no es posible de ninguna otra manera.

La solución mostrará la cercanía de una propiedad a una utilidad o avenida principal y también puede decidirse elevar el valor de un inmueble debido a una nueva utilidad o infraestructura que este en medio de ser construida y usar esta información para ofertar el bien raíz.

Esta solución también propone la existencia de una alarma que le permitirá a un usuario introducir los criterios del inmueble que esta buscando y tan pronto una oferta que se conforme con estos criterios sea introducida la alarma avisara al usuario por su móvil. Esto salvará tiempo tanto al comprador como el vendedor ya que no hay que esperar a que las partes interesadas estén explorando el sitio web y será mas tiempo-real.

Tipo de Investigación

La investigación será de tipo descriptiva por que se deben describir cómo funcionan los módulos del sistema sus responsabilidades y cómo colaboran con otras partes del sistema. Se describirán también los conceptos

pertinentes al trabajo como los concepto de GIS, Mobile App, página web, ciclo de desarrollo de software, base de datos, etc.

La investigación será de tipo documental por que se utilizaran técnicas de análisis, resumen y comparación de diferente fuentes bibliográficas como libros, estándares o convenciones, e información en internet.

Marco Referencial

Marco Teórico

El marco teórico de la investigación es la programación orientada a objetos. Este paradigma caracteriza toda la solución y se hace uso de sus características pertinentes. Se definirán las clases que representaran las partes que actúan en el sistema y estas tendrán sus responsabilidades. Conformando con las practicas de la programación orientada a objetos se definirán interfaces claras(front-end interfaces) abstraídas de sus motores(back-end engines) de manera que provean de modularidad a la aplicación y la hará independiente de las APIs que se utilicen ya sean móvil, base de datos o de GIS en caso de que se quiera cambiar de motor o admitir motores diferentes.

Las bases de datos espaciales son otra parte del marco teorico, en estas las características son representadas con sus atributos (geométricos y no geométricos) y relaciones. Los paquetes de software GIS proveen soporte para datos espaciales y atributos, por ejemplo: ellos soportan el

almacenamiento de datos espaciales usando un enfoque de vectores, al igual que los datos de atributo son soportados con tablas. (Davis, 2001).

Marco Conceptual

API

Interfaz de programación de aplicaciones (del inglés Application Programming Interface), es en esencia un contrato que define un marco de referencia sobre cómo se puede manipular cierta data provista por una aplicación externa ... Con el gran auge del desarrollo de aplicaciones orientadas a servicios, los APIs constituyen unos de los pilares de la programación de aplicaciones orientadas a múltiples plataformas. Daniel et al.(2012).

Aplicación Móvil

Aplicación móvil es como se le denomina al software que es específico para equipos móviles como celulares Smartphone y tabletas.

Base de Datos

Una base de datos es una colección de datos interrelacionados almacenados dentro de un ambiente computarizado. En estos ambientes, los datos son persistentes, lo que quiere decir que este sobrevive problemas imprevistos de software y hardware (con excepción de si el disco se corrompe). El largo volumen de datos y la persistencia, ambas características de las bases de datos, están en contraste con información manipulada por los lenguajes de programación, el cual es suficientemente pequeño en volumen para residir en la memoria principal y el cual desaparece una vez el programa es terminado (Rigaux et al. 2002).

Business Intelligence

Es el conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar datos de los sistemas transaccionales e información desestructurada en información estructurada, para su explotación directa o para su análisis y conversión en conocimiento, dando así soporte a la toma de decisiones sobre el negocio.

“Las aplicaciones de Business Intelligence (BI) son herramientas de soporte de decisiones que permiten en tiempo real, acceso interactivo, análisis y manipulación de información crítica para la empresa. Estas aplicaciones proporcionan a los usuarios un mayor entendimiento que les permite identificar las oportunidades y los problemas de los negocios. Los usuarios son capaces de acceder y apalancar una vasta cantidad de información y

analizar sus relaciones y entender las tendencias que últimamente están apoyando las decisiones de los negocios.

Estas herramientas previenen una Potencial pérdida de conocimiento dentro de la empresa que resulta de una acumulación masiva de información que no es fácil de leer o de usar.”(CherryTree, 2000)

GIS

Un Sistema de información geográfica integra hardware, software y datos para capturar, gestionar, analizar, y mostrar toda forma de información geográfica referenciada.

GIS nos permite ver, entender, cuestionar, interpretar y visualizar datos de muchas maneras que revelen relaciones, patrones y modas en la manera de mapas, globos, reportes y gráficos.

Un GIS ayuda a responder preguntas y resolver problemas viendo los datos de una manera que es rápidamente entendida y fácilmente compartida.

La tecnología GIS puede integrarse en muchas enterprises de sistemas de información.(ESRI, 2014)

GPS

El Sistema de Posicionamiento Global (GPS) es un sistema de radionavegación de los Estados Unidos de América, basado en el espacio, que proporciona servicios fiables de posicionamiento, navegación, y cronometría gratuita e ininterrumpidamente a usuarios civiles en todo el mundo. A todo el que cuente con un receptor del GPS, el sistema le

proporcionará su localización y la hora exacta en cualesquiera condiciones atmosféricas, de día o de noche, en cualquier lugar del mundo y sin límite al número de usuarios simultáneos.

El GPS se compone de tres elementos: los satélites en órbita alrededor de la Tierra, las estaciones terrestres de seguimiento y control, y los receptores del GPS propiedad de los usuarios. Desde el espacio, los satélites del GPS transmiten señales que reciben e identifican los receptores del GPS; ellos, a su vez, proporcionan por separado sus coordenadas tridimensionales de latitud, longitud y altitud, así como la hora local precisa.

Hoy están al alcance de todos en el mercado los pequeños receptores del GPS portátiles. Con esos receptores, el usuario puede determinar con exactitud su ubicación y desplazarse fácilmente al lugar a donde desea trasladarse, ya sea andando, conduciendo, volando o navegando. El GPS es indispensable en todos los sistemas de transporte del mundo ya que sirve de apoyo a la navegación aérea, terrestre y marítima. Los servicios de emergencia y socorro en casos de desastre dependen del GPS para la localización y coordinación horaria de misiones para salvar vidas. Actividades cotidianas como operaciones bancarias, de telefonía móvil e incluso de las redes de distribución eléctrica, ganan en eficiencia gracias a de la exactitud cronométrica que proporciona el GPS. Agricultores, topógrafos, geólogos e innumerables usuarios trabajan de forma más eficiente, segura, económica y precisa gracias a las señales accesibles y gratuitas del GPS.(GPS, 2014)

Inmueble

La palabra inmueble proviene de inmóvil y refiere a bienes raíces como casas, apartamentos, terrenos, etc que son bienes que por su naturaleza están anclados al suelo y no es posible trasladarlos.

Utilidad

Para contexto de este trabajo la palabra utilidad se refiere a lugares útiles de una ciudad como hospitales, departamentos de policía y bomberos, gasolineras, colegios y también para calles, avenidas, elevado, estación del metro, etc.

Marco Espacial

El marco espacial es la Republica Dominicana pero al ser una solución de software general puede ser aplicado para cualquier territorio del planeta.

Marco Temporal

El marco temporal es el presente.

Métodos, procedimientos y técnicas utilizadas en la investigación

Métodos

Observación

Hacemos uso de la observación al encontrar un problema o situación que se puede mejorar con respecto a la compra, renta y venta de inmuebles utilizando sistemas de información geográfica e inteligencia de negocios.

Analítico

También vamos a utilizar el método analítico por que empezamos con una situación que hay descomponer en sus partes funcionales y las relaciones entre estas para así formar la propuesta de la solución.

Sintético

El otro método a utilizar es el sintético por que vamos a integrar los temas de GIS, con inteligencia de negocios, aplicación web, aplicación móvil, base de datos, etc

Procedimientos

Se definirán y explicaran los conceptos generales y específicos a través del trabajo en un orden coherente siguiendo la tabla de contenidos. En esto se dependerá de libros y artículos en internet que serán citados. Y por ultimo está la solución propuesta que integra todo los temas expuestos.

Técnicas

Al ser una investigación de tipo documental principalmente la principal técnica es la lectura, el resumen y la sintonización

Tabla de Contenidos

1 CICLO DE VIDA DE LOS SISTEMAS DE INFORMACIÓN

- 1.1 DESCRIPCIÓN
- 1.2 CARACTERÍSTICAS
- 1.3 LAS FUNCIONES PRINCIPALES DE UN CICLO DE VIDA SOFTWARE SON:
- 1.4 ANTECEDENTES
- 1.5 CLASIFICACIÓN
- 1.6 MODELO DE CASCADA
- 1.7 MODELO EN V
- 1.8 PROTOTIPADO
- 1.9 DESARROLLO RÁPIDO DE APLICACIONES
- 1.10 MODELO EN ESPIRAL
- 1.11 VENTAJAS Y DESVENTAJAS
- 1.12 MODELO “CODIFICAR Y CORREGIR”
 - Ventajas:*
 - Desventajas:*
- 1.13 MODELO EN “CASCADA”
 - Ventajas:*
 - Desventajas:*
- 1.14 MODELO EN “V”
 - Ventajas:*
 - Desventajas:*
- 1.15 “PROTOTIPOS”
 - Ventajas:*
 - Desventajas:*
- 1.16 “DESARROLLO RÁPIDO DE APLICACIONES”
 - Ventajas:*
 - Desventajas:*
- 1.17 MODELO EN “ESPIRAL”
 - Ventajas:*
 - Desventajas:*
- 1.18 ¿COMO ELEGIR?

2 BASE DE DATOS

- 2.1 CONCEPTOS
 - 2.1.1 *Datos*
 - 2.1.2 *Base de Datos*
 - 2.1.3 *Base de datos relacionales*
 - 2.1.4 *Lenguaje de consulta estructurado (SQL)*
 - 2.1.5 *Base de datos espaciales*
 - 2.1.6 *Sistema de Gestión de Base de Datos (SGBD):*
 - 2.1.7 *Sistema de Información geográfica*
 - 2.1.8 *Business Inteligente*
- 2.2 CARACTERÍSTICAS
- 2.3 ORACLE VS MYSQL VS SQL SERVER: COMPARACIÓN DE SGBDR MÁS POPULARES
 - 2.3.1 *Sumario Comparativo de Características*
 - 2.3.2 *Oracle*
 - 2.3.3 *SQL Server*
 - 2.3.4 *MySQL*
- 2.4 BIGTABLE

2.5 GOOGLE EARTH

3 DESARROLLO WEB

3.1 HTML5, CSS3 Y JAVASCRIPT

3.2 CSS3 STYLING

4 DESARROLLO MÓVIL

4.1 TECNOLOGÍA MÓVIL EN LA REPUBLICA DOMINICANA

5 SISTEMAS DE INFORMACIÓN GEOGRÁFICA

5.1 CONCEPTOS

5.1.1 *Cartografía*

5.1.2 *Mapa*

5.1.3 *Sistemas de información geográfica*

5.1.4 *Servicios basados en Localización*

5.1.5 *Servicios de Orientación o Navegación*

5.1.6 *Tecnologías de Localización*

5.1.7 *GPS*

5.1.8 *Telefonía celular*

5.1.9 *WiFi*

5.1.10 *RFID*

5.1.11 *Beneficios de los Sistemas de Información Geográfica*

5.1.12 *Los beneficios de SIG generalmente se dividen en cinco categorías básicas:*

5.1.13 *¿Qué se puede hacer con los SIG?*

5.2 SISTEMA DE INFORMACIÓN GEOGRÁFICA ESCOGIDA: GOOGLE MAPS

5.2.1 *Google Maps*

5.2.2 *Entre las cosas que se pueden hacer con Google Maps están*

5.2.3 *Diferentes APIs*

5.2.4 *Cobertura*

5.2.5 *Uso Comercial*

5.2.6 *Límites de uso*

5.2.7 *Costos*

5.3 GOOGLE MAPS FOR BUSINESS

5.3.1 *Determinación del uso de la aplicación*

5.3.2 *Google Analytics para el API de Google Maps for Business*

5.3.3 *Asistencia*

5.3.4 *Límites de uso*

5.4 GOOGLE MAPS ENGINE

5.4.1 *Visualiza tus datos geoespaciales en la nube*

5.4.2 *Cree y comparta mapas de Google Maps personalizados a escala*

5.4.3 *Capacidades de la plataforma de Google Maps Engine*

5.5 GOOGLE MAPS ENGINE PRO

5.6 GOOGLE MAPS PLACES

5.6.1 *Búsquedas*

5.6.2 *Eventos*

5.6.3 *API de autocompletado de Google Places*

5.6.4 *API de Google Places*

5.6.5 *Descripción general*

5.6.6 *Tipos de sitios admitidos*

5.6.7 *Tipos admitidos para buscar y añadir sitios*

5.6.8 *Tipos adicionales utilizados por el servicio de Google Places*

5.6.9 *Conecta personas y lugares*

- 5.6.10 *Trabaja con una potente base de datos*
- 5.6.11 *A continuación se indican los tipos de solicitudes de sitios disponibles.*
- 5.6.12 *Políticas para desarrolladores*
- 5.6.13 *Requisitos del logotipo*
- 5.6.14 *Límites de uso*
- 5.6.15 *Requisitos de las Condiciones de uso y la Política de privacidad*
- 5.6.16 *Aplicaciones para móviles*
- 5.6.17 *Aplicaciones web*

6 PROPUESTA DE DISEÑO DE LA APLICACIÓN

6.1 CONSIDERACIONES

7 DISEÑO DE LA APLICACIÓN WEB Y MÓVIL

- 7.1 BÚSQUEDA EN WEB
- 7.2 AGREGAR UN INMUEBLE EN WEB
- 7.3 BÚSQUEDA EN MÓVIL
- 7.4 AGREGAR UN INMUEBLE EN MÓVIL
- 7.5 ALERTA

Bibliografia

CherryTree. (2000). *Business Intelligence-The Missing Link*.

Daniel Jacobson, G. B. (2012). *APIs: A Strategy Guide*. California: O' Reilly.

El-Rabbany, A. (2002). *GPS The Global Positioning System*. Norwood: Artech House, Inc.

ESRI. (2014). *What is GIS?* Obtenido de Esri: <http://www.esri.com/what-is-gis/overview>

GPS. (2014). Obtenido de GPS: <http://www.gps.gov/spanish.php>

Rolf A. de By, R. A. (2000). *Principles of Geographic Information Systems*. Enschede: The International Institute for AeroSpace Survey and Earth Sciences(ITC).