

Universidad de las Ciencias Informáticas

Facultad 2



Título: Sistema de Información Geográfica UCI en su versión portable para dispositivos con Sistema Operativo Android

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autores

Yilian Martínez González

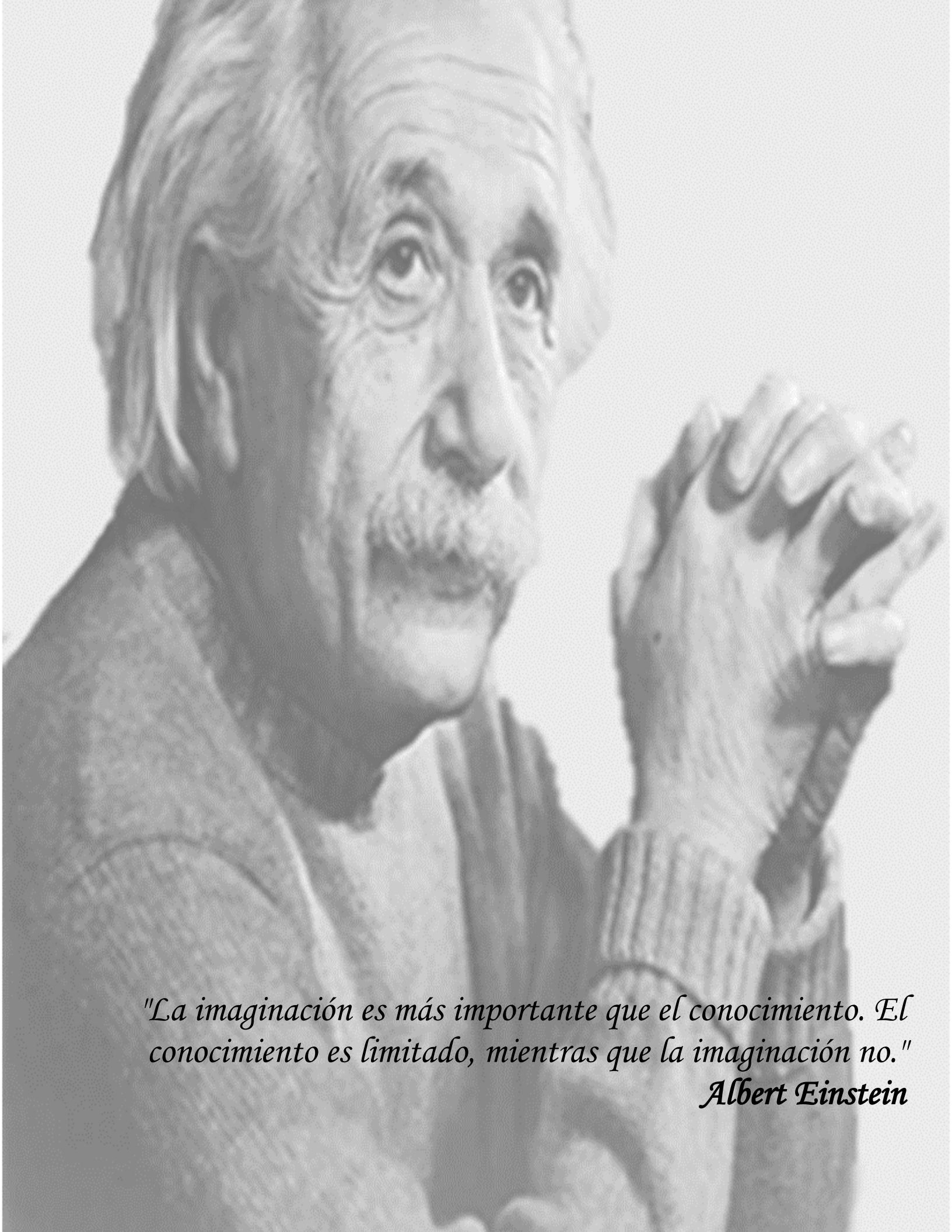
Jesús Magdiel Alonso García

Tutores

Ing. Katia Ramírez Bruzón

Ing. Yaniel Alfredo Velázquez Bruceta

“La Habana, junio de 2016”



"La imaginación es más importante que el conocimiento. El conocimiento es limitado, mientras que la imaginación no."

Albert Einstein

AGRADECIMIENTOS

Yilian

Primero que todo quisiera agradecerles a mis padres Onidia y Elexis por tanto esfuerzo durante años y por su amor incondicional. Gracias por su apoyo en todo momento de mi vida, por estar presentes cuando los necesité. Gracias a ambos por convertirme en la mujer que hoy soy. Los amo mucho.

A mi abuela Gladis, por estar estos años a mi lado. Por cuidarme, quererme y siempre estar pendiente de mí.

A todos mis tíos, en especial a mi tía Nexys por quererme como su hija y creer en mí en todo momento.

A mi hermanita Érica y a mis primos por su amor y cariño.

A mi novio Alberto que a pesar de llevar poco tiempo juntos me ha apoyado mucho. Te quiero mi amor.

A Jessica y Javier que son partes importantes de mi vida, más que amigos son como mis hermanos, por cuidarme y acompañarme siempre en los buenos y malos momentos. Gracias por estar ahí siempre que los necesito, por ayudarme, apoyarme y comprenderme. También agradecer a sus familias que me acogieron tantas veces en sus casas y me trataron como su hija. Gracias por brindarme esta bonita amistad que espero nunca termine.

A todos mis amigos Manuel, Mayara, Elaine, Alexey, Yisel, Lionys, Hermes, en fin, a todos los que me han ayudado en estos cinco años de universidad. Los quiero mucho a todos.

A mi compañero de tesis por su esfuerzo en la realización de este trabajo. Gracias Magdí.

A mis tutores por ayudarnos y guiarnos en este camino de tantos meses, por estar siempre disponibles y dispuestos a ayudarnos.

A todos los profesores que intervinieron en mi formación durante estos años.

A mis compañeros de grupo por ser parte de mi historia en esta universidad, por los buenos momentos vividos. A todos ustedes gracias.

Jesús Magdiel

En este momento tan especial para mí quisiera agradecerles a todas aquellas personas que de una forma u otra han sido partícipes de este triunfo, a los que confiaron en mí y nunca me dieron la espalda.

Quiero primeramente agradecerles a mis padres:

Manuel Alonso y Regla García, por estar siempre a mi lado en todos los momentos de mi vida, por sus regaños por sus caricias, porque siempre voy a estar en deuda con ellos, por la educación que me han dado, por confiar en mí en todo momento, por levantarme el ánimo cuando lo he necesitado, por enseñarme que no hay nada en el mundo que no pueda lograr si me lo propongo, a mi papa en especial por ser mi guía y aunque en los últimos años no lo he tenido ni he contado con su presencia, me ha hecho ver que no hay distancia en el mundo k pueda apagar el amor que siento por él y siempre ha estado pendiente a mi felicidad y mi vida. En el caso de mi mama quiero que sepa que no existe forma en el mundo en la que le pueda pagar todo lo que ha hecho por mí pero quiero aprovechar esta oportunidad para decirle que si dios me diera la oportunidad de casarme con ella lo haría, porque quiero ser prisionero de por vida de sus regaños, de sus caricias, de sus consejos, de sus besos, de sus castigos, porque a pesar de que discutimos en muchas ocasiones sé que nos amamos y nos queremos mucho, porque ha sido madre y padre a la vez y nunca ha tenido miedo de salir y comerse el mundo por ver a sus hijos felices...

Quiero agradecerle también a la persona que más quiero y anhelo en el mundo y a pesar de ser menor que yo es mi ídolo, esa persona es mi Hermana:

Primeramente quiero que sepas que fuiste esa personita que quizás sin saberlo me dio la alegría más grande de mi vida, hoy recuerdo cuando les dije a mis

padres que quería tener una hermanita hembra a la cual pudiera amar, querer y cuidar toda la vida y llegaste tú y de veras no quiero a nadie más porque tú tienes todo lo que me hace falta para vivir, me enorgullece decir que eres mi fuerza de voluntad porque quizás no lo sabes pero en muchas ocasiones sentí que no podía seguir adelante con esta carrera y lo único que me motivaba era que yo tenía que darte el ejemplo, para que tu siguieras mis pasos. Quiero aprovechar este momento para decirte que nada ni nadie en el mundo podrá ser capaz de alejarme de tí, ni de borrar o cambiar todos los sentimientos lindos que siento por tí...te quiero mucho mi ñaña....

Quedaría mal conmigo mismo si no le agradeciera a la primera persona que corrió conmigo que se fajo conmigo, comíamos en el mismo plato, la vida nos puso como primos pero nunca nos hemos querido como eso porque somos MAS QUE HERMANOS:

Reinier sinceramente no sé cómo catalogar que somos, porque es imposible decirte primo o hermano porque cuando aún no teníamos uso de razón nos estábamos abrazando, besándonos...etc. luego fuimos creciendo y nos era imposible dejar de discutir y fajarnos pero sabíamos que nos queríamos demasiado, ahora nos es imposible dejar de contarnos las cosas o de darnos un consejo o de darnos un abrazo o de sentarnos a tomarnos una botella, sabes que hay cosas de mi vida que solo tengo confianza en tí para decírlas, hemos pasado más que malos momentos juntos y nunca hemos tenido miedo a enfrentarlos, como cuando nos hicimos nuestro primer tatuaje juntos y después nos castigaron porque éramos unos niños, para tí es este título porque quisiera que la vida me siguiera dando la oportunidad de seguir compartiendo mis éxitos y mis fracasos contigo, porque estoy más que seguro que por muy fuertes que sean los tiempos nunca nos vamos a dar la espalda o nos vamos a separar.

Ahora quiero agradecerle a una persona que es súper importante para mí, que ha sido mi padre, mi amigo, mi guía, mi ejemplo, mi consejero...etc. esa persona es mi TIO YURI:

TIO gracias por tus consejos, por enseñarme a vivir, por tenerme como un hijo más, por ayudarme cuando lo he necesitado, por abrirme los ojos cuando no he querido ver las cosas, por los regaños, por tus castigos, por hacerme entender que la vida es más de lo que podemos ver, que nunca podemos quedarnos conformes que debemos ser optimistas. Quiero pedirte disculpas si en algún momento de la vida te he hecho sentir mal con alguna acción mía y quiero que sepas que nunca ha sido ni será esa mi intención porque sería incapaz de hacerle daño a las personas que amo y quiero en mi vida...Hoy recuerdo un día que nos contaste a mí y a mi primo que siempre quisiste ser como mi abuelo y sinceramente quiero que sepas que yo quiero ser como TÚ.

Agradecerle también a mi tía Sílvia y a mi primo Rainer, que son dos personitas súper importantes para mí, a mi tía muchísimas gracias por todo lo que has hecho por mí, por escucharme, por verme reír, llorar, por regalarme tus consejos, por regañarme cuando estoy equivocado y sobre todo por tenerme como tu hijo también. A mi primo Rainer bueno quiero que sepas que eres mi vida que te quiero como mi hermanito, que este título es tu meta también y sé que vas a ser mejor ingeniero que yo, la UCI lo va a decir, quiero que sepas que 100pre voy a estar ahí para ayudarte en lo que pueda para escucharte apoyarte en tus problemas y claro para disfrutar contigo cuando te hagas profesional.

No quisiera pasar por alto para agradecerle también a dos personas que hoy no se encuentran entre nosotros porque la vida lo decidió así, ellos son mi tía Teresa y mi Tío Alejo, a los dos muchas gracias por todas las cosas que hicieron en vida por mí, solo quiero que donde quiera que estén compartan este

momento conmigo y a mi tío quiero que sepa que el primer trago de ron me lo voy a dar a su nombre.

Agradecerle también a mi primo Wilson a Stalone, a mi prima Regla Caridad, Arlette, a mi tía Juana, muchas gracias por confiar en mí apoyarme, por aconsejarme, muchas gracias por todo y quiero que sepan que siempre los tengo presente.

Ahora quiero agradecerle a un grupo de personas que la vida me dio la oportunidad de conocerlos, luego se volvieron amistades, siguieron y fueron amigos y ahora son mis hermanos:

A Diego:

Muchas gracias por todo mi hermano, no sé qué decirte ni como agradecerte todas las noches que me soportaste llorando, riéndome, por estar siempre a mi lado en todos los momentos malos y buenos que tuve en la uci, nunca se me va a olvidar el día que tuve un problema y fuiste el único que me dijo no me importa que nos voten vamos a resolver esto, pero sería incapaz de involucrarte en algo y hoy estamos los dos aquí de ingenieros, tu antes y yo después pero logramos el objetivo ambos. Nunca dudes de que eres de las personas que más aprecio en el mundo y de veras que quisiera volver a empezar la universidad solo para contar por 5 años más con tu amistad y tu presencia. Te quiero mucho brother.

Ramón:

Tú me has enseñado que el hecho de que dos personas que tengan problemas, o que discutan a diario, no significa que no se quieran con la vida. Juntos hemos pasado muchas batallas y aunque la distancia nos separe mañana quiero que sepas que seguiremos enfrentando juntos las batallas, nunca se me va a olvidar el día en que discutimos en las canchas y tú te fuiste para la casa y cuando yo llegue a hablarte de lo que había pasado me dijiste que el problema lo habías dejado en las canchas que ahora estábamos en la casa y no había ningún tipo

de lío, como también me acuerdo como lllore el día que me escribiste el correo porque estabas envuelto en un proceso de que te querían botar y nunca dude en llamar a mi mama, para que hablara también con la gente, porque aun con nuestros problemas de carácter te quiero siempre a mi lado.

Odelis:

Quiero agradecerle grandemente a esta muchacha que fue más k mi amiga mi hermana en esta universidad, por darme su confianza, por apoyarme a estudiar en mis primeros años, por nunca darme la espalda, por enseñarme a creer que un hombre y una mujer si pueden ser amigos. Hoy deseo que la vida te regale lo mejor porque si una persona se lo merece eres tu Ode.

Yunior:

Quiero darle el más grande de los agradecimientos a Yunior por todos los momentos que me regalo en esta universidad, por las madrugadas estudiando él y yo solos, por su amistad sincera, por confiar en mí en todos los momentos y por sobre todas las cosas por ayudarme con mi tesis hasta el último momento antes de tomar su decisión, tu eres una de las personas a las que nunca tendré como pagarle gracias por todo mi hermano.

Carlos:

Hay personas que no importan si llevan poco tiempo en tu vida y tú eres uno de ellos, gracias por tu sinceridad, por cuidarme, por ayudarme con tus consejos, por evitarme problemas, por acogerme en tu casa cuando sabias que lo necesitaba. En fin, gracias por todo.

Anniel:

Gracias por estar a mi lado, escucharme, aconsejarme, por no dudar de mí, por estar siempre disponible para mis problemas, por cuidarme, por guiarme en sus momentos, sabes que nunca se me va a olvidar algo que me dijiste un día que veníamos de un plenitud y es que tú eras mi amigo para elogiar las cosas

cuando estaban bien hechas pero para llamarme la atención cuando estuvieran mal, muchas gracias por todo carnal.

Purís:

Mí hermano gracias mil veces por sacarme de los momentos malos por los que he pasado, gracias por echar para adelante conmigo, por darme lo mejor de tí, por ser incondicional, por regalarme tu amistad y sobre todo por confiar en mí.

Roberto Daniel:

No se cómo explicar qué nivel tiene nuestra amistad, sé que llevamos muchos años distanciados pero para mí sigues siendo el mismo que cuando jugábamos pelota juntos, gracias a tí hoy soy ingeniero ya que fuiste tú quien me incentivo a coger esta carrera...gracias por todos estos años de amistad.

Dayan Andrade:

Hoy quiero darte las gracias por ser parte de mi vida, nunca voy a olvidar como corríamos por las calles de mi cuadra juntos o como lo hacíamos en santo domingo, o como fuimos juntos solos por primera vez a un carnaval y nos emborrachamos siendo unos niños, o como hacíamos maldades en mi casa y en la tuya, nuestros secretos de la infancia, sabes que confió en tí como en nadie aunque hoy nos separen poco más de 90 millas, algún día estaremos juntos de nuevo lo sé. Te quiero hermano.

Agradecerle también a una persona que formo parte de mi vida y me ayudó en un momento determinado en todo lo que pudo, a Suinny mis agradecimientos de corazón, por todo los buenos momentos, por ayudarme a ser mejor persona, por aceptarme en su familia como uno más por el apoyo de todos sus familiares, gracias a Niovis, Richards, Consuelo, El Duque, Liodnys y Liordys, a todos de corazón muchas gracias por todo el apoyo que me brindaron, y quiero que sepan que aquí estaré siempre para lo que les haga falta.

Le estoy muy agradecido al Flaco Ariel, por regalarme su amistad, por ser más que un amigo, por apoyarme, en fin por todo, Ariel Parra porque a pesar de estar distanciado estuvo en muchos momentos de mi carrera apoyándome mientras pudo, para tí mis agradecimientos, a Raciél gracias por todo hermano, a Javier a Ignacio, a Julio, a Mayrelis, Marinelsa, Elizabeth Centeno Triana, a Jany mi chiquitica, Eddy, Glauver, a Fernando, al Poyo y a una persona que viene conmigo desde muy pequeños y nos hemos fajado, distanciado, pero tenemos mucho que ver porque nos queremos a pesar de todo, nos ayudamos, para tí LISVET mis agradecimientos, sabes que nadie me conoce mejor que tú en esta escuela y gracias por caerme arriba todos los días para que terminara con mi tesis, gracias por apoyarme nene, y muchas gracias sobre todo por comprenderme y aceptarme como soy te quiero mucho mi enana.

Le agradezco también a todos mis amigos, que han estado a mi lado a Rodiel, con sus pesadeces, a linares, a Javier, Batista, Yordi que lo quiero con la vida, al yoyo, al Rafa, Yanelis, a Claudia, Midalis, Milagros, Lisandra, a mis amigos de la vieja escuela Jorge Carlos, Osvaldo, Yosvani, Tony, Sevilla, Abelito, Rosnel, Elaine, Mayara, Yisell, a Pochet por ser la persona que me aconsejo en mis primeros años y lo tengo muy presente, a Elizabeth, a Liliana, a Saray, a la Flaca, a Eliani espero que algún día estén todas aquí, a mis amigos del Aula a Ilsen que gracias a, el hoy estoy aquí, gracias por tu apoyo, a David, Carlos, Andy y Abraham gracias por sus consejos y por apoyarme siempre, a Sheila mi compañera de asiento y amiga te quiero, a Cubertier, a Camaguey, a Erick, al Boza, a Leonis, Pablo, a Sosa, el chino (Manuel), a Yaritza mi hermanita que siempre la voy a querer mucho, y quiero que sepas que siempre voy a estar disponible para lo que te haga falta, a Sidielim a Dianelis por sus pesadeces te quiero negrita y a una persona que es muy especial para mí y aunque nos fajamos, discutimos, nos ofendemos, ella sabe que significa mucho para mí, esa persona eres tu ANGELA, si algún día te sientes sola no dudes en llamarme que

yo nunca voy a dudar en dejar lo que esté haciendo para correr a ayudarte mi hermana, te quiero con la vida.

*Hay personas que llegan a tu vida y en muy poco tiempo, tienen para marcarte, con su cariño, con sus atenciones, con su amor, con su paciencia, de veras hoy quiero darle un agradecimiento especial a esa personita que me dejó entrar por segunda vez en su vida y a la cual estoy dispuesto a no defraudarla de nuevo gracias a mi novia Jessica, por aguantarme cada una de estas noches, por ayudarme a estudiar, por relajarme, por aceptarme tal y como soy, por confiar en mí, quiero que sepas que no te miento cuando te digo que te quiero mucho, y le doy gracias a dios por haberte puesto nuevamente en mi camino. **ILY** mi **PELUCHIN**.*

Y por último y no por esto es menos importante, quisiera agradecer a mi compañera de tesis, por haber confiado en mí, que sinceramente de no haber sido por ella este sueño no se me hubiese hecho realidad, te quiero mucho y de veras te deseo todo lo mejor en tu vida profesional Yíli.

DEDICATORIA

Yilian

Quiero dedicar este trabajo de diploma a toda mi familia, especialmente a mi mamá y mi papá que son las personas que más me han ayudado en mi vida, sin ellos este sueño no hubiese podido hacerse realidad.

Jesús Magdiel

Dedico esta tesis a dos personas a las cuales les debo cada paso que doy en la vida, por todo el cariño, el amor, por sus regaños, por nunca darme la espalda a pesar de que me han regañado o me regañaron en su momento, y aunque lamentablemente no cuento con la presencia física de uno de ellos quiero que sepa que donde quiera que este sé que me está mirando, y cuidando, dedico esta tesis a mis abuelos Sixta Rodríguez Guerra y Jesús García Cruz (que Dios lo tenga siempre en la gloria).

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores de este trabajo y autorizamos al Centro de Telemática de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Yilian Martínez González

Firma Autor

Jesús Magdiel Alonso García

Firma Autor

Katia Ramírez Bruzón

Firma Tutor

Yaniel Alfredo Velázquez Bruceta

Firma Tutor

DATOS DE CONTACTO

Tutor

Nombre y Apellidos: Katia Ramírez Bruzón.

Sexo: F.

Institución: UCI.

Dirección de la institución: carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

Correo electrónico: kramirez@uci.cu.

Área: Facultad 2, Centro TLM.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación:

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Tutor

Nombre y Apellidos: Yaniel Alfredo Velázquez Bruceta.

Sexo: M.

Institución: UCI.

Dirección de la institución: carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

Correo electrónico: ybruceta@uci.cu

Área: Facultad 2, Centro TLM.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2014

Institución donde se graduó: Universidad de las Ciencias Informáticas.

RESUMEN

Con los disímiles avances tecnológicos que se han obtenido con el transcurso de los años se dio lugar al surgimiento de los Sistemas de Información Geográfica (SIG), constituyendo herramientas tecnológicas útiles para resolver problemas complejos de análisis, manejo, localización, visualización y consulta de datos geográficos. Con el inicio de esta tecnología, la sociedad procuró un paso de avance en todos los sentidos, tanto económico, como político y social.

Actualmente, Cuba se encuentra en el camino de digitalizar y automatizar toda información y proceso que sea posible. La Universidad de las Ciencias Informáticas (UCI) ocupa un lugar importante en la producción de software. En la UCI existen diversos Centros de Desarrollo de Software, tal es el caso del Centro de Desarrollo Geoinformática y Señales Digitales (GEYSED), el cual creó el Sistema de Información Geográfica UCI (SIGUCI) web en el que se hace una representación geoespacial del entorno universitario. El sistema funciona correctamente desde las computadoras, pero se dificulta el trabajo con él desde los teléfonos móviles con Sistema Operativo (SO) Android, así como el acceso desde el navegador de estos dispositivos, debido a que es necesario conectarse a una red inalámbrica (WIFI).

En el documento se encuentran las principales características de otros SIG que se consultaron en la investigación. Se seleccionan las herramientas y tecnologías idóneas y se guía el proceso de desarrollo conforme a lo establecido por la metodología XP. Como resultado de la investigación se obtiene un SIGUCI para dispositivos móviles con Sistema Operativo (SO) Android.

Palabras claves: Android, sistema, mapa, Sistema de Información Geográfica, dispositivos móviles.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPITULO I: FUNDAMENTACIÓN TEÓRICA	6
1.1 Introducción.....	6
1.2 Conceptos fundamentales	6
1.2.1 Mapa	6
1.2.2 Sistema	6
1.2.3 Sistema de Información Geográfica	6
1.2.4 Servicios.....	7
1.2.5 Servicios web	7
1.3 Análisis de las soluciones existentes	8
1.3.1 Internacional.....	8
1.3.2 Nacional	9
Resultado del análisis.....	9
1.4 Herramientas y Tecnologías	10
1.4.1 Sistema operativo.....	10
1.4.2 Lenguaje de programación.....	11
1.4.3 Entorno Integrado de Desarrollo.....	12
1.4.4 Herramientas de Desarrollo de Software.....	12
1.4.5 Sistema gestor de base de datos (SGBD).....	13
1.5 Servicio web.....	14
1.6 Metodología de desarrollo.....	14
Conclusiones del capítulo	17
CAPITULO II: CARACTERÍSTICAS DEL SISTEMA	18
2.1 Introducción	18
2.2 Propuesta de solución.....	18

2.3 Funcionalidades del sistema	18
2.4 Características no funcionales	18
2.5 Planificación	19
2.5.1 Historias de usuarios	19
2.5.2 Estimación de esfuerzo por HU	27
2.5.3 Iteraciones	28
2.5.4 Plan de entregas	28
Conclusiones del capítulo	29
CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	30
3.1 Introducción	30
3.2 Arquitectura de software	30
3.3 Patrón arquitectónico.....	32
3.3.1 Modelo-Vista-Controlador (MVC)	32
3.4 Patrones de diseño.....	34
3.4.1 Patrones para asignar responsabilidades (GRASP).....	34
3.5 Tarjetas Clases – Responsabilidad – Colaborador (CRC)	35
3.6 Tareas de ingeniería.....	36
CAPÍTULO IV: PRUEBA	40
4.1 Introducción	40
4.2 Validación de la propuesta.....	40
4.2.1 Pruebas unitarias.....	40
4.2.2 Pruebas de aceptación.....	41
Conclusiones del capítulo	45
CONCLUSIONES GENERALES	46
RECOMENDACIONES	47
REFERENCIAS	48
BIBLIOGRAFÍA.....	51

ANEXO 1	55
ANEXO 2	57
ANEXO 3	59

ÍNDICE DE TABLAS

Tabla 1. Diferencias entre Metodologías Ágiles y Tradicionales	15
Tabla 2. Características del equipo de desarrollo.....	16
Tabla 3. HU # 1 Buscar estructura	21
Tabla 4. HU # 2 Ubicar persona	22
Tabla 5. HU # 3 Actualizar base de datos	23
Tabla 6. HU # 4 Trazar camino.....	24
Tabla 7. HU # 5 Dibujar ruta	25
Tabla 8. HU # 6 Medir distancia	26
Tabla 9. HU # 7 Limpiar.....	27
Tabla 10. Estimación de esfuerzo por HU	27
Tabla 11. Plan de iteraciones	28
Tabla 12. Plan de entregas.....	28
Tabla 13. Tarjeta CRC Activity_map	35
Tabla 14. Tarjeta CRC Content_map	35
Tabla 15. Tarjeta CRC Bonuspack_bubble	36
Tabla 16. Tarjeta CRC Person_list	36
Tabla 17. TI # 1 Buscar estructura	37
Tabla 18. TI # 2 Ubicar persona	37
Tabla 19. TI # 3 Actualizar base de datos	38
Tabla 20. TI # 4 Trazar camino.....	38
Tabla 21. TI # 5 Dibujar ruta	38
Tabla 22. TI # 6 Medir distancia	39
Tabla 23. TI # 7 Limpiar.....	39
Tabla 24. CP # 1 Buscar estructura.....	42
Tabla 25. CP # 2 Ubicar persona	42
Tabla 26. CP # 3 Actualizar base de datos.....	43
Tabla 27. CP # 4 Trazar camino	43

Tabla 28. CP # 5 Dibujar ruta	44
Tabla 29. CP # 6 Medir distancia.....	44
Tabla 30. CP # 7 Limpiar.....	44
Tabla 31. Tarjeta CRC Detail_place	55
Tabla 32. Tarjeta CRC Person_list_item	55
Tabla 33. Tarjeta CRC DBHelper	55
Tabla 34. Tarjeta CRC Person	55
Tabla 35. Tarjeta CRC Place.....	56
Tabla 36. Tarjeta CRC PersonAdapter.....	56
Tabla 37. Tarjeta CRC Auxiliar	56
Tabla 38. Tarjeta CRC CustomInfomWindows.....	56
Tabla 39. Tarjeta CRC MapActivity	56

ÍNDICE DE FIGURAS

Figura 1. Arquitectura en tres capas.....	31
Figura 2. Patrón Modelo – Vista –Controlador.....	32
Figura 3. Estructura del proyecto utilizando el patrón MVC.....	33
Figura 4. Fragmento de código de la utilización de patrón GRASP Experto	34
Figura 5. Prueba unitaria Buscar estructura	41
Figura 6. Resultados de las pruebas de aceptación en cada iteración.....	45
Figura 7. Prueba unitaria Ubicar persona.....	57
Figura 8. Prueba unitaria Trazar camino	57
Figura 9. Prueba unitaria Medir distancia	58

INTRODUCCIÓN

Los mapas han sido utilizados durante siglos por los seres humanos para orientarse geográficamente. Un mapa es la representación de la superficie terrestre, o una parte de ella en un plano, lo que constituye el medio indispensable para la localización y la orientación (1). En la actualidad se utilizan en casi todas las esferas de la ciencia, convirtiéndose en un instrumento necesario en varios aspectos profesionales y académicos.

Con los disímiles avances tecnológicos que se han obtenido con el transcurso de los años en el mundo, se ha logrado integrar el uso de los mapas con las nuevas tecnologías, dando lugar al surgimiento de los Sistemas de Información Geográfica (SIG), estos son el resultado de la aplicación de las llamadas Tecnologías de la Información y las Comunicaciones (TIC) a la gestión de la Información Geográfica (IG). Los SIG son una integración organizada de hardware, software y datos geográficos, diseñados 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 (2).

El uso de estos sistemas se ha incrementado exponencialmente en la actualidad, pues cada vez son más las empresas que se suman al desarrollo de este tipo de software. Los SIG desarrollados para dispositivos móviles son un claro ejemplo de lo antes mencionado, por lo que son de gran importancia para la sociedad por la aplicación que tienen en la toma de decisiones en las empresas que día a día se sumergen más en el mundo de la telefonía digital.

El uso de SIG para dispositivos móviles ha proliferado debido a las ventajas que tienen estos con respecto a los SIG de escritorio. Con ellos no solo es posible localizar lugares de interés, o determinar la menor distancia entre dos puntos, sino que además se puede hacer uso de la aplicación a cualquier hora del día, independientemente del lugar en el que se encuentre el usuario. Estas características marcan una notable diferencia, evidenciando las ventajas de portabilidad y fácil acceso de estos sistemas con respecto a los SIG de escritorio.

Actualmente, Cuba se encuentra en el camino de digitalizar y automatizar toda la información y los procesos de la sociedad. El desarrollo de sistemas que capturen, procesen y representen la información de una forma más práctica y entendible es uno de los mayores aportes de las TIC, lo que contribuye al ahorro de recursos y tiempo del país, influyendo así positivamente en el desarrollo socio-económico del mismo.

Los SIG han sido utilizados en diferentes áreas del territorio cubano, como en la salud, donde se han realizado importantes aplicaciones entre las que se destaca el SIG para la ejecución de estudios

epidemiológicos. También, han sido desarrollados para proyectos relacionados con el planeamiento urbano, la gestión del medio ambiente, y además para elaborar una base de datos de la cual se alimenten las líneas directrices del Programa de Desarrollo Humano Local (HuPDHL).

Como entidad puntera en el proceso de informatización de nuestro país se encuentra la Universidad de las Ciencias Informáticas (UCI). En la UCI existen diversos Centros de Desarrollo de Software en los cuales se despliegan diferentes Proyectos Productivos como es el caso del Centro de Geoinformática y Señales Digitales (GEYSED), entidad que tiene como misión la creación de productos, servicios y soluciones informáticas para el procesamiento de Señales Digitales y la Geoinformática.

El centro GEYSED es el protagonista de la creación del sistema SIGUCI (Sistema de Información Geográfica de la UCI). Esta aplicación web hace una representación geoespacial del entorno universitario mostrando todos los puntos de interés con información socio-económica asociada. También brinda servicios de localización de personas, estructuras, lugares de interés y edificios de residencia. Otras de las principales funcionalidades que presenta están: medición de distancias y el cálculo de áreas.

A pesar de que se puede acceder al sistema correctamente desde las computadoras, se dificulta el trabajo con SIGUCI desde los dispositivos móviles con Sistema Operativo (SO) Android, debido a que para el uso de este sistema desde estos terminales se hace necesario tener red inalámbrica (WIFI), lo que obstaculiza en ocasiones el acceso al sistema ya que los usuarios no pueden consultar el SIG desde cualquier lugar de la universidad, pues en la UCI no se cuenta con WIFI en toda su extensión territorial.

SIGUCI web no posee una versión para el trabajo desde estos teléfonos móviles. Como consecuencia, existen dificultades para acceder y usar las funcionalidades de la aplicación utilizando el navegador de estos dispositivos, existen problemas de visualización y configuración adaptable. Otro inconveniente que presenta es que los usuarios pueden mostrar en algún momento inconformidad ya que se ven obligados a consultar la información de SIGUCI desde una computadora y puede ser que en un momento determinado el usuario no cuente con una PC y sí con un teléfono móvil.

Con el análisis de lo antes expuesto se desarrolla este trabajo de diploma sobre la base de la solución al siguiente **problema de la investigación**: ¿Cómo desarrollar una aplicación para dispositivos móviles con SO Android, que permita la visualización y ubicación de las estructuras y personas de la UCI sin depender del acceso a una red WIFI?

Se define como **objeto de estudio**: El proceso de consumo de información en los SIG desde los dispositivos móviles a través de servicios web, enmarcado en el **campo de acción**: Sistema de Información Geográfica UCI para dispositivos móviles con SO Android.

Para dar respuesta al problema antes mencionado se traza como **objetivo general**: Desarrollar una aplicación para dispositivos móviles con SO Android, que permita la visualización y ubicación de las estructuras y personas de la UCI sin depender del acceso a una red WIFI.

Como **tareas de la investigación** se proponen las siguientes:

1. Elaboración del marco teórico de la investigación para identificar conceptos fundamentales en la comprensión de la investigación.
2. Realización del estudio del estado del arte de los Sistemas de Información Geográfica para establecer semejanzas con las soluciones existentes e identificar características o funcionalidades similares a las del sistema a implementar.
3. Análisis de la metodología de desarrollo a utilizar para la elaboración de los artefactos que se generan durante el desarrollo de la aplicación SIGUCI para dispositivos con Sistema Operativo Android.
4. Selección de las tecnologías y herramientas necesarias para el desarrollo de la aplicación SIGUCI para dispositivos móviles que operan con SO Android.
5. Implementación de la aplicación SIGUCI para dispositivos móviles con SO Android
6. Realización de pruebas para la validación del sistema a partir de pruebas unitarias y de aceptación

Para el desarrollo de la investigación se emplearon los siguientes métodos:

Métodos teóricos:

- ✓ **Histórico-Lógico**: Posibilitó el estudio analítico de la trayectoria histórica de la evolución que han tenido los SIG y facilitó conocer de forma general el funcionamiento y desarrollo de los mismos.
- ✓ **Análisis-Síntesis**: Se realiza un análisis de toda la información obtenida referente a los SIG para poder organizarla y sintetizarla.

Métodos empíricos:

- ✓ **Observación:** Se diagnostica el problema que se investiga, a fin de obtener los elementos necesarios para que sea solucionado.
- ✓ **Entrevista:** Posibilitó realizar entrevistas de forma informal. Estas se realizaron en las etapas previas de la investigación, a los trabajadores del centro GEYSED, para así conocer las fortalezas y debilidades del sistema ya implementado, así como las tecnologías que utiliza.

El trabajo de diploma se divide en cuatro capítulos, los cuales estarán estructurados de la siguiente forma:

Capítulo 1: Fundamentación Teórica, se describen de forma general los aspectos teóricos de la investigación. Se realiza un estudio del estado del arte y se vincula con la aplicación. Se analizan las herramientas, metodología de desarrollo y lenguajes de programación favorables para la implementación de la solución.

Capítulo 2: Características del Sistema, se describen las características de la aplicación a desarrollar, además de reflejar las funcionalidades del sistema a través de las Historias de Usuario, y se definen las características no funcionales.

Capítulo 3: Análisis y diseño de la Aplicación, se define la arquitectura de la solución y los patrones de diseño. Se elaboraron las tarjetas Clase-Responsabilidad-Colaboradores (CRC) según plantea la metodología seleccionada para organizar y agilizar el proceso en la implementación y prueba del sistema.

Capítulo 4: Prueba, se presentan las principales características del proceso de implementación. Se desarrollan las tareas de ingenierías correspondientes a las Historias de Usuario y se realizan las pruebas que validan que el producto final cumpla con requerimientos definidos previamente.

CAPITULO I: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se realiza un estudio de los aspectos teóricos necesarios para la concepción del trabajo de diploma, mediante temas relacionados con los Sistemas de Información Geográfica (SIG). Se abordan diferentes conceptos que permiten la comprensión del problema, se selecciona el lenguaje de programación, herramientas y tecnologías necesarias, así como la metodología a utilizar.

1.2 Conceptos fundamentales

1.2.1 Mapa

Un mapa es una imagen o representación donde se simboliza gráficamente a partir de medidas longitudinales un territorio determinado en una superficie bidimensional. Se identifican rutas para establecer destinos de un punto a otro, se ubican localidades y se pueden observar los distintos tipos de terreno que pueda presentar dicha superficie. (3)

1.2.2 Sistema

Un sistema es un conjunto de elementos o partes organizadas y relacionadas que interactúan entre sí para lograr un objetivo. Los sistemas reciben datos, energía o materia del ambiente y proveen información, energía o materia. (4)

1.2.3 Sistema de Información Geográfica

SIG o GIS (Geographic Information System, en su acrónimo inglés), se define 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. En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. Genéricamente los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

Un Sistema de Información Geográfica (SIG) queda definido entonces como una integración de hardware, software, datos georeferenciados y personal, con el objetivo de capturar, almacenar, manipular, analizar y mostrar la información geográficamente referenciada en pos de resolver problemas complejos de planificación y gestión. (2)

Para lograr el correcto funcionamiento de un SIG se debe contar con ciertos componentes que son imprescindibles. Ellos son: (5)

Hardware: El hardware a emplear varía desde servidores para guardar la información así como computadoras conectadas a la red o individuales para capturar los datos. Para lograr un funcionamiento correcto dentro de una organización se debe utilizar hardware específico que solvente las necesidades de la aplicación (velocidad, costo, seguridad, etc.).

Software: Los principales componentes que los SIG proporcionan son los Sistemas de Manejo de Base de Datos (SMBD), la Interfaz Gráfica de Usuarios (GUI en sus siglas en inglés) para facilitar el acceso a las herramientas, instrumentos de captura y manejo de información geográfica y los de soporte de consultas, análisis y visualización de datos geográficos.

Datos: Son el componente más importante de un SIG. Los datos geográficos y alfanuméricos pueden obtenerse de varias formas, una de ellas es a través de los propios recursos y otra mediante otros proveedores.

Personal: Se hace necesario contar con personal experto que garantice la actualización de la información, de esta forma, el SIG no pierde valor y no se cometen errores que pueden afectar a otros individuos.

Métodos: Para lograr la exitosa implementación de un SIG hay que tener en cuenta un buen diseño y reglas bien definidas.

Algunas de las ventajas que brindan los SIG son:

- La sustitución del mapa impreso en papel al utilizar los mapas digitales.
- La manipulación de elementos de un mapa, relacionando cada objeto con una información más amplia.
- El análisis matemático y salidas gráficas visualizando resultados.
- La generación de muchos mapas con los mismos datos.

1.2.4 Servicios

Acción y efecto de servir. Es un conjunto de actividades que buscan responder a las necesidades de un cliente. (6)

1.2.5 Servicios web

Conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la web. Permite la comunicación y el intercambio de datos entre aplicaciones y sistemas heterogéneos en entornos distribuidos que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. (7)

1.3 Análisis de las soluciones existentes

Actualmente, a nivel mundial, los SIG son muy utilizados. Hay varias aplicaciones basadas en los SIG, desarrolladas en distintas tecnologías, donde se pueden encontrar los SIG de escritorio, SIG móviles, SIG de desarrollo, SIG de servicios web, entre otros. La investigación estará centrada en el análisis de diferentes soluciones que existen hoy a nivel internacional y nacional.

1.3.1 Internacional

Google Maps

Es un servidor de aplicaciones de mapas que pertenece a Google. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes. Ofrece la funcionalidad de realizar acercamientos y alejamientos para mostrar el mapa aunque no todas las áreas están cubiertas con la misma resolución en las imágenes de satélite, las zonas menos pobladas suelen mostrarse con menos detalle. Para utilizar la aplicación es necesario estar conectados a Internet, de manera que se pueda descargar mapas en tiempo real. También brinda la posibilidad de descargar secciones de mapas para poder consultarlos si no se dispone de conexión a Internet. (8)

GvSIG Móvil

GvSIG Mobile es un Sistema de Información Geográfica (SIG) orientado a dispositivos móviles, ideal para proyectos de captura y actualización de datos en campo (9). Se caracteriza por disponer de una interfaz amigable, siendo capaz de acceder a los formatos más comunes y cuenta con un amplio número de herramientas SIG y GPS ideales para trabajar con información de naturaleza geográfica. Este sistema tiene como objetivo ampliar las plataformas de ejecución de GvSIG Desktop a una gama de dispositivos móviles. Es el primer cliente de estas características licenciado como software libre.

OffMaps 2

OffMaps 2 da la posibilidad al usuario de obtener acceso a mapas escogiendo un lugar, esto lo puede hacer de manera online u offline. Con cobertura mundial, OffMaps ofrece mapas de cualquier región, pero centrándose en las principales ciudades del mundo. Esta aplicación no es gratis. Una de las cosas más destacadas de OffMaps es la posibilidad de hacer búsquedas en los mapas sin conexión (10).

OsmAnd

Es una aplicación para visualizar mapas que ofrece una funcionalidad de navegación, dando acceso a los mapas de alta calidad alojados en la base de datos de OpenStreetMap (OSM). Permite almacenar en la tarjeta de memoria del dispositivo los datos de los mapas para su uso sin conexión. Accediendo a la antena GPS del dispositivo, OsmAnd ofrece sistema de navegación guiado por voz, para rutas en

coche, bicicleta o sendas peatonales. Todas las funcionalidades que ofrece funcionan con conexión y sin conexión a Internet. (11)

Principales características (12)

- Realiza búsquedas de lugares de dirección, según el tipo (restaurante, hotel, gasolinera, museo), o mediante coordenadas geográficas.
- Permite visualizar la posición y orientación en el mapa.
- Visualiza opcionalmente el límite de velocidad, en caso de que se excedan notifica de esto al usuario.
- Los mapas incluyen senderismo y rutas en bicicleta, ideal para actividades al aire libre.
- Encaminamiento y modos de visualización especiales para ciclistas y peatones.
- Muestra de forma opcional las paradas de transporte público (autobús, tranvía, tren), incluyendo los nombres de las líneas.
- Calcula la distancia y el tiempo aproximado existente entre puntos determinados por el usuario.
- Tiene cobertura de mapas muy aproximadas y con buena calidad en muchos países del mundo principalmente en Europa occidental.

1.3.2 Nacional

SIGUCI web

Esta aplicación hace una representación geoespacial del entorno universitario mostrando todos los puntos de interés con información socio-económica asociada. Está diseñado y desarrollado para que funcione sobre una plataforma web. Entre las funcionalidades que tiene se pueden señalar el cálculo de área, medir distancia, buscar estructuras, edificios de residencia y personas. Se puede mencionar además que posee servicio de navegación (acercar, alejar y mover).

SIG-Rutas

SIG-Rutas es un SIG sobre las rutas que brindan servicio de transportación a los trabajadores de la comunidad universitaria de la UCI. Este sistema está diseñado y desarrollado para que funcione sobre una plataforma web. Entre las funcionalidades que tiene se pueden mencionar la navegación (acercar, alejar, mover, recentrar, entre otras), cálculo de áreas, medir distancias, visualización de capas y personalización de rutas. Además, cuenta con un sistema de alerta y permite conocer la localización de direcciones, municipios, paradas y rutas. Este sistema brinda a los usuarios la posibilidad de conocer las paradas más cercanas a partir de diferentes criterios introducidos en el sistema. (13)

Resultado del análisis

A partir del análisis realizado de las soluciones existentes se concluye que, de manera general, no podrán ser utilizadas para la representación de la información geográfica de la UCI. A pesar de que

todas ellas permiten consultar datos de forma offline, no se garantiza que sus mapas sean obtenidos de forma gratuita en todos los casos, como ocurre con OffMaps 2. De igual forma estos sistemas muestran información muy general que no responde a la necesidad de representar las estructuras y edificaciones de la UCI, y sin posibilitar la edición de los mapas que utilizan, independientemente de la forma en que estos se adquieran. Teniendo en cuenta además que frecuentemente utilizan servicios como Google Maps y Google Earth, su utilización dependería de la disponibilidad de acceso a una red WIFI y adicionalmente en Cuba depende en gran medida de la infraestructura tecnológica y del acceso que se tenga a dichos servicios. Por otra parte el SIGUCI existente posee problemas de visualización y configuración adaptable desde los dispositivos móviles, por lo que hay dificultades para acceder y usar las funcionalidades de esta aplicación utilizando el navegador de estos terminales. Este SIG obliga a los usuarios a conectarse al sistema mediante una interfaz web.

1.4 Herramientas y Tecnologías

1.4.1 Sistema operativo

Un sistema operativo (SO) es un conjunto de programas o software, destinado a permitir la comunicación entre el usuario y la máquina de forma cómoda y eficiente; se encarga de gestionar los recursos del ordenador, esto incluye la gestión del hardware desde los niveles más básicos. (14)

Android

Android es un sistema operativo orientado a dispositivos móviles. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma. El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. Proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, Wi-Fi, Bluetooth) de una forma sencilla en un lenguaje de programación muy conocido como es Java, haciendo uso del SDK de Android. (15)

Principales elementos:

- Framework de aplicaciones: permite el reemplazo y la reutilización de los componentes, muchos de estos integrados de forma nativa al SO.
- Máquina virtual Dalvik: máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados.
- Bluetooth, EDGE, 3g y Wi-Fi: orientado a la conexión, Android tiene soporte para tecnologías de conectividad dependiendo del terminal.
- Pantalla Táctil: las aplicaciones están diseñadas para el uso sobre dispositivos con pantallas táctiles.

Luego de analizar las características que posee el SO Android, se decide utilizar para el desarrollo del trabajo ya que actualmente es uno de los SO, para dispositivos móviles, más usado por parte de los usuarios, en particular los de la UCI. El lenguaje de programación que utiliza permite que el avance de la aplicación sea más ágil ya que el equipo de desarrollo posee experiencia en el trabajo con este lenguaje.

1.4.2 Lenguaje de programación

Es un lenguaje artificial diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar, es decir, es el encargado de controlar el comportamiento de una computadora. Se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas. Como algunos de los ejemplos de lenguajes de programación podemos encontrar a: C, C++, Pascal, PHP, Inteligencia artificial, Python y Java. (16)

Actualmente existen diversos ambientes para programar rápidamente en Android. Dentro de estas plataformas de programación se encuentra Basic4Android, cuyo lenguaje base es VisualBasic. Está orientado a personas que comienzan en el mundo de la programación de manera gráfica y no tan abstracta. No es el mismo lenguaje de Microsoft, pero su sintaxis es la misma, esta plataforma no es gratuita. (17)

Otro de los escenarios que se utilizan es LiveCode, el lenguaje de programación que usa se llama Programación Orientada a Eventos, y se basa en arrastrar elementos a un área de trabajo y programar los eventos que estén vinculados a este elemento. Este programa también es de pago y depende del alcance que se quiere tener en el desarrollo. (17)

Java 1.8

Para el desarrollo de aplicaciones Android se ha escogido el lenguaje Java para dar soporte a las aplicaciones de forma nativa. Java es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humana. Presenta características importantes como: (18)

- Es un lenguaje multiplataforma: el mismo código escrito en Java que funciona en un SO, funcionará en cualquier otro que tenga instalada la máquina virtual de Java.
- Gracias al API de Java se puede ampliar el lenguaje para que sea capaz de, por ejemplo, comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML dinámicas, crear aplicaciones visuales.

Partiendo del análisis realizado de las características antes expuestas, se decidió utilizar el lenguaje de programación Java, ya que este es recomendado para el desarrollo de aplicaciones en Android y se integra al Android Studio. El equipo de desarrollo tiene conocimientos sobre este lenguaje y posee

experiencias, capacidades y habilidades con el desarrollo de aplicaciones en Java. Además por ser este el lenguaje nativo en el desarrollo de aplicaciones Android.

1.4.3 Entorno Integrado de Desarrollo

Un Entorno de Desarrollo Integrado, *IDE* (Integrated Development Environment) por sus siglas en inglés, es una herramienta informática que conquista un entorno de programación amigable para uno o varios lenguajes, brindando facilidades al desarrollador, tales como: un editor de código fuente, herramientas de construcción automáticas, un depurador, un compilador y opcionalmente un constructor de interfaz gráfica. Los *IDE* están pensados y diseñados para maximizar la productividad del programador proporcionando componentes muy unidos con interfaces de usuario similares. Presentan un único programa en el que se lleva a cabo todo el desarrollo. Generalmente, este programa suele ofrecer muchas características para la creación, modificación, compilación, implementación y depuración de software. (19)

Android Studio 1.3.2

Android Studio es el Entorno de Desarrollo Integrado (*IDE*, por sus siglas en inglés) de Android; el cual brinda acceso a un conjunto de funciones que permiten el flujo de trabajo de desarrollo (20). Algunas de las características por lo que se decidió utilizar Android Studio son:

- Código y edición de recursos, diseño de interfaz de usuario:
 - ✓ Edición de código: Android Estudio aprovecha todas las capacidades de edición de código inteligente de IntelliJ IDEA (IDE para el desarrollo de programas informáticos desarrollado por JetBrains) como el autocompletado avanzado de código, refactorización, y análisis de código.
- Análisis de rendimiento:
 - ✓ Monitor de memoria: permite monitorear el uso de memoria de la aplicación en tiempo real para ayudar a encontrar maneras de mejorar el rendimiento de la aplicación.

1.4.4 Herramientas de Desarrollo de Software

Estas herramientas ayudan al desarrollo de sistemas de información, surgieron para intentar dar solución a los problemas inherentes a los proyectos de generación de aplicaciones informáticas. Existen al menos 20 herramientas libres para diseñar software. Estas permiten la generación de código en Java, C++, Python y PHP. (21)

Android SDK 24.0.1

Se utilizará Android SDK en su versión 24.0.1 ya que incluye un conjunto de herramientas de desarrollo y comprende un depurador de código, bibliotecas, un simulador de dispositivos, documentación, ejemplos de código y tutoriales. (22)

Esta herramienta se integra perfectamente con el IDE Android Studio, por su soporte extendido por parte de los desarrolladores del SO Android y por disponer de las librerías necesarias para desarrollar aplicaciones en la plataforma Android. Además, es libre, gratuito y de código abierto. (22)

1.4.5 Sistema gestor de base de datos (SGBD)

Un SGBD es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Algunos ejemplos de SGBD son Oracle, PostgreSQL, SQLite y MySQL. (23)

SQLite 3.0

SQLite es un manejador de código abierto de bases de datos que combina una interfaz muy limpia de SQL. Permite trabajar con poca memoria y con una velocidad bastante rápida, características que son necesarias cuando se habla de entornos móviles. SQLite soporta las características estándar de las bases de datos relacionales como la sintaxis que se basa en SQL, transacciones y la elaboración de consultas. Debido a esto, cualquier desarrollador que haya trabajado con bases de datos sin importar el entorno, no encontrará una dificultad especial en trabajar con bases de datos locales en Android.

Las principales características de este gestor de bases de datos son las siguientes: (24)

- ✓ Tamaño: tiene pequeña memoria y necesita una biblioteca única para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- ✓ Portabilidad: se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- ✓ Costo: es de dominio público, por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.
- ✓ No posee configuración: de la forma en que fue creado y diseñado, no necesita ser instalado. No es necesario prender, reiniciar o apagar un servidor, e incluso configurarlo; esta cualidad permite que no haya un administrador de base de datos para crear las tablas, vistas, asignar permisos o bien la adopción de medidas de recuperación de servidor por cada caída del sistema.

Se decide utilizar SQLite como SGBD pues la base de datos completa se encuentra en un solo archivo permitiendo que pueda ser exportada a cualquier otra plataforma. Este sistema gestor puede funcionar enteramente en memoria, lo que la hace muy rápida. Además, su código fuente es de dominio público y se encuentra muy bien documentado.

1.5 Servicio web

Un servicio web está compuesto por un conjunto de aplicaciones o tecnologías con capacidad para interoperar en la Web. Estos sistemas intercambian datos entre sí con el objetivo de ofrecer servicios que proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar. (25)

WSDL

El Lenguaje de Descripción de Servicios Web (Web Service Description Language, WSDL por sus siglas en inglés) es un dialecto basado en XML. Un documento WSDL proporciona la información necesaria al cliente para interactuar con el servicio web. WSDL es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de red. Define una gramática XML que describe los servicios de red como colecciones de puntos finales de comunicación capaces de intercambiar mensajes. Las definiciones de servicio de WSDL proporcionan documentación para sistemas distribuidos y sirven como fórmula para automatizar los detalles que toman parte en la comunicación entre aplicaciones. (26)

Un documento WSDL utiliza los siguientes elementos en la definición de servicios de red: (26)

- **Types:** contenedor de definiciones del tipo de datos que utiliza algún sistema de tipos (por ejemplo XSD).
- **Message:** definición abstracta y escrita de los datos que se están comunicando.
- **Operation:** descripción abstracta de una acción admitida por el servicio.
- **Port Type:** conjunto abstracto de operaciones admitidas por uno o más puntos finales.
- **Binding:** especificación del protocolo y del formato de datos para un tipo de puerto determinado.
- **Port:** punto final único que se define como la combinación de un enlace y una dirección de red.
- **Service:** colección de puntos finales relacionados.

1.6 Metodología de desarrollo

Para realizar un excelente trabajo durante el desarrollo de software hacen falta más que notaciones de modelado y herramientas, haciéndose imprescindible para esto la metodología de desarrollo, que se cataloga en tradicional o ágil, encargada de proveer una dirección a seguir para la correcta aplicación de los demás elementos.

Las metodologías tradicionales son usadas en proyectos de gran tamaño (respecto a tiempo y recursos), donde se exige un alto grado de formalidad en el proceso. En el desarrollo del presente trabajo, no se consideró su utilización, ya que es un proyecto pequeño donde el entorno del sistema es

muy cambiante, presentando este método tradicional altos costos al implementar un cambio, además no ofrecer una buena solución para proyectos donde el entorno es volátil.

Partiendo de lo antes expuesto es que se decide hacer uso de las metodologías ágiles, por estar orientadas a proyectos pequeños, donde se considera más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan. Por lo que las mismas constituyen una solución juiciosa para ese contexto, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto.

A continuación, se pueden observar las principales diferencias de las metodologías ágiles con respecto a las tradicionales (“no ágiles”).

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparadas para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1. Diferencias entre Metodologías Ágiles y Tradicionales

XP (Extreme Programming)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, propiciando un buen clima de trabajo. La metodología XP se distingue, fundamentalmente, por su pronta, concreta y continua retroalimentación en ciclos cortos de tiempo, así

como por su enfoque de planificación incremental, el cual permite definir un plan general que se espera evolucione durante la vida del proyecto. (27)

La metodología XP tiene como objetivos establecer las mejores prácticas de Ingeniería de Software en el desarrollo de proyectos, mejorar la productividad de estos y garantizar la calidad del software desarrollado, haciendo que este supere las expectativas del cliente. Esta metodología se aplica en un contexto bien definido por, un cliente, requisitos cambiantes, grupo pequeño y muy integrado, y un equipo con formación elevada y capacidad de aprender. XP posee también facilidad para flexibilizar la implementación de funcionalidades, respondiendo a los cambios que el negocio necesita. Se encuentra bien documentada con innumerables recursos en línea disponibles, comunidades libres y grupos de noticias, encontrándose una gran cantidad de proyectos desarrollados con esta metodología. (27)

Características del equipo de desarrollo

En la siguiente tabla se muestran las principales aptitudes del equipo de desarrollo del Sistema de Información Geográfica UCI en su versión portable para dispositivos con Sistema Operativo Android.

Características	Valoración
Composición del equipo de trabajo	4 Personas ✓ 2 clientes ✓ 2 desarrolladores
Experiencia productiva	✓ Media
Conocimientos de metodologías ágiles	✓ cliente (medio)
Conocimientos de metodologías tradicionales	✓ cliente (medio) ✓ desarrolladores(bajo)
Conocimientos del Lenguaje de Programación Java	✓ cliente (medio) ✓ desarrolladores(medio)

Tabla 2. Características del equipo de desarrollo

Teniendo en consideración lo antes mencionado y las características de XP, se selecciona como la metodología a emplear, porque combina las mejores prácticas para desarrollar software, y las lleva al extremo. Los clientes y los desarrolladores forman parte del equipo de trabajo de forma tal que se logra una integración cliente - equipo de desarrollo, permitiendo la retroalimentación, corrección de errores y finalmente la realización de un producto capaz de satisfacer todas las necesidades del mismo. Además, se reduce el proyecto en todas las etapas del ciclo de vida del sistema, puesto que se desarrollan

pequeñas partes de este y son probadas permitiendo terminar todas las funcionalidades e integrar todo el producto final. No se genera gran cantidad de artefactos y roles ya que el proyecto es pequeño y está diseñado para ser realizado en el menor tiempo posible.

Conclusiones del capítulo

En este capítulo se realizó un estudio referente a las tecnologías que se utilizarán para desarrollar la aplicación y se logró organizar y guiar el trabajo hacia el objetivo trazado en el inicio de la investigación. Se profundizó en los conceptos fundamentales para el desarrollo de una aplicación, que permita, entre otras funcionalidades, la búsqueda de estructuras y personas pertenecientes a la UCI, desde los dispositivos móviles.

Se definieron las herramientas y tecnologías mayormente adecuadas para dar solución al problema planteado, teniendo en cuenta las características de estas y las necesidades descritas con anterioridad a las que da respuesta la propuesta de solución. Como resultado de la investigación se seleccionó Android Studio 1.3.2 como IDE de desarrollo, Java 8.1 como lenguaje de programación y SQLite 3.0 como sistema gestor de base de datos. Además, se investigó sobre las metodologías de desarrollo de software, escogiéndose XP como la metodología de desarrollo a utilizar.

CAPITULO II: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el presente capítulo se hace el análisis y diseño de la propuesta del sistema. La investigación se centra en las características de la aplicación, dirigida por la metodología de desarrollo. Se realiza una descripción de las historias de usuario que se proponen para dar solución al problema originado por la situación problemática. Se especifican las principales funcionalidades que debe cumplir la aplicación, para satisfacer las necesidades del cliente.

2.2 Propuesta de solución

Para dar solución a las deficiencias identificadas se propone desarrollar una aplicación para dispositivos móviles con SO Android que muestre la mapificación de la Universidad de las Ciencias Informáticas (UCI).

El usuario poseerá la aplicación SIGUCI móvil en el teléfono y además tendrá la base de datos (BD) de las direcciones de las estructuras de la universidad. Para consultar la información referente a la ubicación de las personas tendrá que conectarse la primera vez a la WIFI y podrá descargar para su teléfono la información contenida en esta base de datos, permitiéndole hacer uso de esta de forma offline y en cualquier lugar donde se encuentre. Para actualizar esta información el usuario deberá conectarse a la WIFI y en la aplicación seleccionará la opción Actualizar base de datos, donde la aplicación se conectará con la base de datos de las personas y consumirá a través de servicios web esta información, descargándose para el teléfono del usuario. La aplicación permitirá realizar las funcionalidades: Ubicar persona, Buscar estructura, Trazar camino, Dibujar ruta y Medir distancia.

2.3 Funcionalidades del sistema

- Buscar estructura.
- Ubicar persona.
- Actualizar base de datos.
- Trazar camino.
- Dibujar ruta.
- Medir distancia.

2.4 Características no funcionales

Restricciones de implementación

- ✓ El sistema deberá ser implementado en el lenguaje de programación Java 1.8.
- ✓ Se utilizará el marco de trabajo de desarrollo Android en su versión 5.0.1.
- ✓ Se empleará la herramienta de desarrollo Android Studio 1.3.2.

Software

Requisitos mínimos para el dispositivo móvil:

- ✓ Soporte para conexiones WIFI.
- ✓ Sistema operativo Android 4.2.

Hardware

Requisitos para abrir la aplicación:

- ✓ La base de datos debe estar guardada en la raíz del teléfono.
- ✓ La base de datos debe tener capacidad mínima de 20 MB.

Interfaz de Usuario

- ✓ El sistema debe presentar un diseño adaptable a las dimensiones del contenido y muestre los elementos de una forma ordenada y optimizada independientemente de la resolución del dispositivo.

2.5 Planificación

El ciclo de vida de un proyecto realizado con la metodología XP se inicia con la Planificación. En esta fase, el cliente plantea a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas y tecnologías que se utilizarán en el proyecto. Además se establece la prioridad de cada Historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la entrega y se determina un cronograma en conjunto con el cliente. (28)

2.5.1 Historias de usuarios

Las Historias de usuario (HU) corresponden a la técnica utilizada para especificar los requisitos del software. Se trata de formatos en los cuales el cliente describe brevemente las características que el sistema debe poseer. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. (28)

Para hacer más comprensible las mismas, a continuación se describe su leyenda:

- **Número:** Número de la historia de usuario incremental en el tiempo.
- **Nombre de la Historia de usuario:** Nombre de la historia de usuario especificado por el programador.

- **Usuario:** Personas involucradas en el desarrollo de las HU.
- **Iteración asignada:** Número de la iteración.
- **Prioridad en negocio (Baja, Media, Alta):**
 - Baja: Se le otorga a las HU que son de funcionalidades auxiliares y que son independientes del sistema.
 - Media: Se le otorga a las HU que son de funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
 - Alta: Se le otorga a las HU que son de funcionalidades fundamentales en el desarrollo del sistema.
- **Riesgo en desarrollo (Bajo, Medio, Alto):**
 - Bajo: Cuando en la implementación de las HU puedan existir errores, pero éstos son tratados fácilmente y no afectan el desarrollo del sistema.
 - Medio: Cuando en la implementación de las HU puedan existir errores y retrasen la entrega del producto.
 - Alto: Cuando en la implementación de las HU pueda existir algún error y afecte la disponibilidad del sistema.
- **Puntos estimados:** Tiempo estimado que se demorará el desarrollo de la HU.
- **Descripción:** Breve descripción de la HU.
- **Observaciones:** Señalamiento o advertencia del sistema.
- **Prototipo de interfaz:** Prototipo de interfaz solo si aplica.

A continuación se describen las HU:

Historia de usuario	
Número: 1	Usuario: Usuarios
Nombre de Historia de usuario: Buscar estructura	
Prioridad en negocio: Alta	Riesgo de Desarrollo: Media
Puntos estimados: 3	Iteración asignada: 1
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario abre la aplicación, selecciona la opción Buscar estructura. Escoge en el menú el tipo de estructura que desea buscar, especifica el nombre de esa estructura y elige el botón Buscar. Luego de esto, la aplicación resalta en el mapa el lugar donde se encuentra localizada dicha estructura.	
Observaciones: Se brinda información general de la estructura.	

Prototipo de interfaz:



Tabla 3. HU # 1 Buscar estructura

Historia de usuario	
Número: 2	Usuario: Usuarios
Nombre de Historia de usuario: Ubicar persona	
Prioridad en negocio: Alta	Riesgo de Desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
<p>Descripción: El usuario abre la aplicación, selecciona la opción Ubicar persona. Entra en el menú el nombre o usuario de la persona que se desea ubicar y elige el botón Buscar persona. Luego de esto, la aplicación muestra un cartel con datos específicos de este individuo y además especifica en el mapa el lugar donde radica dicha persona.</p>	
<p>Observaciones: Se brinda información de la persona, como por ejemplo el edificio y apartamento donde reside y el número de teléfono.</p>	

Prototipo de interfaz:



Tabla 4. HU # 2 Ubicar persona

Historia de usuario	
Número: 3	Usuario: Usuarios
Nombre de Historia de usuario: Actualizar base de datos	
Prioridad en negocio: Alta	Riesgo de Desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario se conecta a la WIFI desde el teléfono y selecciona la opción Actualizar base de datos. Inmediatamente la aplicación se conecta a la base de datos de la ubicación de las personas y esta información se descarga para el teléfono, actualizándose la que se encontraba anteriormente.	
Observaciones: NA	
Prototipo de interfaz:	



Tabla 5. HU # 3 Actualizar base de datos

Historia de usuario	
Número: 4	Usuario: Usuarios
Nombre de Historia de usuario: Trazar camino	
Prioridad en negocio: Media	Riesgo de Desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario abre la aplicación, selecciona la opción Trazar camino. Luego especifica en el mapa el camino que desea trazar, haciendo puntos sobre este.	
Observaciones: NA	

Prototipo de interfaz:

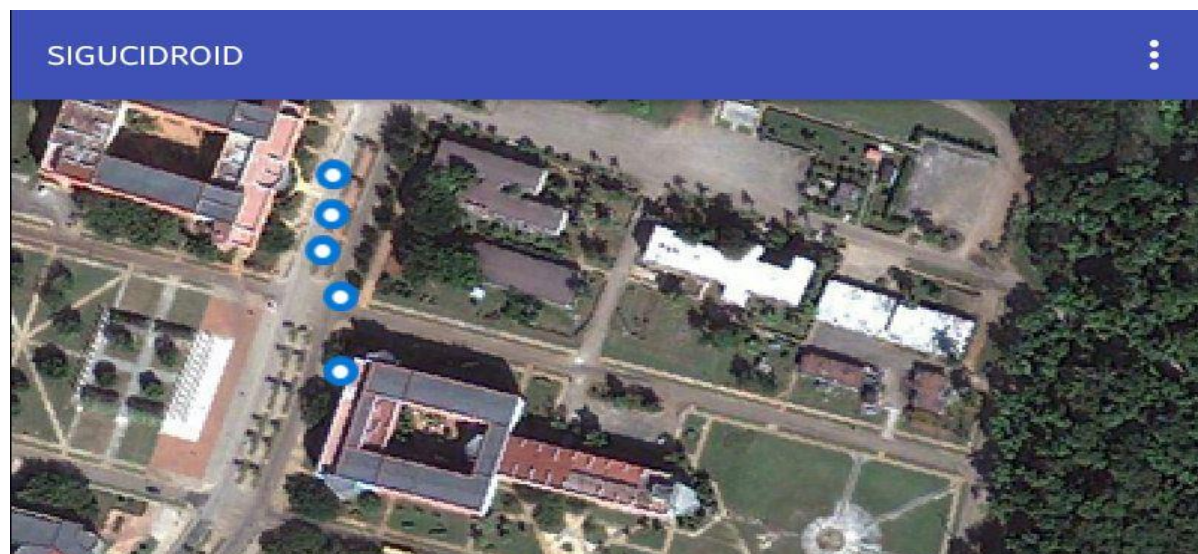


Tabla 6. HU # 4 Trazar camino

Historia de usuario	
Número: 5	Usuario: Usuarios
Nombre de Historia de usuario: Dibujar ruta	
Prioridad en negocio: Media	Riesgo de Desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario abre la aplicación, selecciona la opción Trazar camino y realiza las operaciones de esta funcionalidad, luego selecciona la opción Dibujar ruta, esta muestra en el mapa una ruta, surgiendo de los puntos antes señalados en la funcionalidad Trazar camino.	
Observaciones: Se obtiene una ruta en el mapa uniéndose todos los puntos realizados en la funcionalidad Trazar camino.	

Prototipo de interfaz:

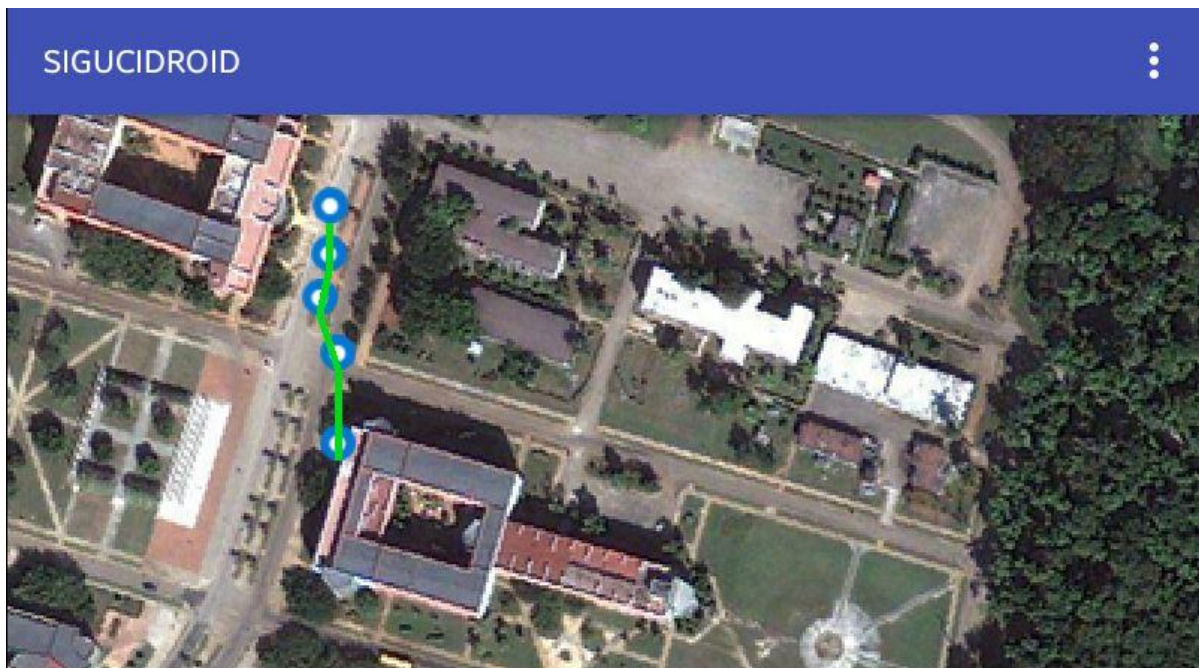


Tabla 7. HU # 5 Dibujar ruta

Historia de usuario	
Número: 6	Usuario: Usuarios
Nombre de Historia de usuario: Medir distancia	
Prioridad en negocio: Media	Riesgo de Desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario abre la aplicación, selecciona la opción Trazar camino y Dibujar ruta, realizando en ambos casos las operaciones que producen dichas funcionalidades, luego selecciona la opción Medir distancia y automáticamente se muestra la distancia de la ruta especificada.	
Observaciones: NA	

Prototipo de interfaz:



Tabla 8. HU # 6 Medir distancia

Historia de usuario	
Número: 7	Usuario: Usuarios
Nombre de Historia de usuario: Limpiar	
Prioridad en negocio: Media	Riesgo de Desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario, después de haber realizado una búsqueda de estructura o una ubicación de persona, elige la opción Limpiar, esta borra la operación antes realizada y muestra el mapa con todas las estructuras incorporadas nuevamente.	
Observaciones: Se muestra el mapa con todas las estructuras.	
Prototipo de interfaz:	



Tabla 9. HU # 7 Limpiar

2.5.2 Estimación de esfuerzo por HU

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación, donde los miembros del equipo de desarrollo trabajan sin interrupción.

Número	Historia de usuario	Puntos estimados
1	Buscar estructura	3
2	Ubicar persona	2
3	Actualizar base de datos	1
4	Trazar camino	0.5
5	Dibujar ruta	0.5
6	Medir distancia	1
7	Limpiar	1

Tabla 10. Estimación de esfuerzo por HU

2.5.3 Iteraciones

Luego de ser identificadas y descritas las HU, el próximo paso será la planeación de la fase de implementación, donde se realizarán dos iteraciones, las cuales se describen a continuación:

Iteración 1

Serán desarrolladas las HU 1, 2 y 3 que son las de prioridad Alta. El cliente podrá probar las funcionalidades ya implementadas.

Iteración 2

Serán desarrolladas las HU 4, 5, 6 y 7 que son las de prioridad Media, una vez implementadas estas funcionalidades, facilitarán un funcionamiento más completo de la aplicación.

Iteración	Orden de las HU a implementar	Duración Total
1	Buscar estructura	6 semanas
	Ubicar persona	
	Actualizar base de datos	
2	Trazar camino	3 semanas
	Dibujar ruta	
	Medir distancia	
	Limpiar	

Tabla 11. Plan de iteraciones

2.5.4 Plan de entregas

El plan de entrega da un aproximado de las versiones, tomando como fecha de inicio de la implementación el 4 de abril del 2016:

Iteración	Fecha de Entrega
1	13 de mayo del 2016
2	3 de junio del 2016

Tabla 12. Plan de entregas

Conclusiones del capítulo

Con la realización de este capítulo se identificaron las funcionalidades y la lista de reserva que debe cumplir la aplicación. Se realizó una estimación de esfuerzo dedicado al desarrollo para cada funcionalidad, identificando el orden en que serán desarrolladas, facilitando así una mejor organización del trabajo. Además se definió un plan de entregas que permite establecer fechas de culminación para cada iteración.

CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

3.1 Introducción

Luego de haber desarrollado la fase de Planificación en el capítulo anterior, se continúa con otra de las etapas de la metodología XP, el Diseño de la Aplicación. En este capítulo se define el patrón de arquitectura a seguir, además de los patrones de diseños que serán utilizados en la implementación de la aplicación. Serán definidas las tarjetas CRC como técnicas de diseño.

3.2 Arquitectura de software

La Arquitectura de Software es la forma en la que se organizan los componentes de un sistema, interactúan y se relacionan entre sí y con el contexto, aplicando normas y principios de diseño y calidad, que fortalezcan y fomenten la usabilidad a la vez que dejan preparado el sistema, para su propia evolución. La calidad del software está dada por el cumplimiento de un conjunto de atributos, dentro los que se destacan: (29)

- Integridad de los módulos independientes del sistema.
- Capacidad de interoperar con otros sistemas (interoperabilidad).
- Capacidad de permitir ser modificable a futuro (modificabilidad).
- Ser fácilmente mantenible (mantenibilidad).
- Tener una estructura que facilite la reusabilidad de la misma en futuros sistemas.
- Mantener un diseño arquitectónico escalable que permita su ampliación (escalabilidad).
- Facilidad de ser sometido a pruebas que aseguren que el sistema falla cuando es lo que se espera (testeabilidad).

3.2.1 Arquitectura en tres capas

Para el desarrollo del sistema se empleará la Arquitectura en tres capas. Consiste en dividir el sistema en varias capas lográndose de esta forma reducir el grado de complejidad. Las aplicaciones contienen códigos para la presentación, procesamiento y almacenamiento de datos, la diferencia se encuentra en la organización del mismo. Ofrece ventajas de tipo organizativo debido a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La utilización de esta arquitectura permite desarrollar capas en paralelo, además posibilita un mantenimiento más sencillo ya que se puede modificar un componente de ser necesario en lugar de toda la aplicación y permite agregarle nuevos módulos aumentando su flexibilidad.

El sistema se divide en las siguientes capas lógicas: (30)

Presentación: Esta capa es la que ve el usuario (también se le denomina capa de usuario). Es la encargada de presentarle el sistema al usuario, además comunica y captura la información. También

es conocida como interfaz gráfica y debe tener la característica de ser amigable, entendible y fácil de usar. Esta capa se comunica únicamente con la capa de Negocio.

Negocio: En esta capa es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. También se le conoce como capa de lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Se comunica con la capa de Presentación para recibir las solicitudes y presentar los resultados. También se relaciona con la capa de Datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Datos: En esta capa es donde residen los datos y es la encargada de acceder a los mismos. Está formada por un gestor de base de datos que realiza toda la acumulación de datos, recibe solicitudes de almacenamiento o recuperación de información desde la capa de Negocio.

A continuación se muestra una imagen donde se puede observar las relaciones de las capas por la que está conformada esta Arquitectura.

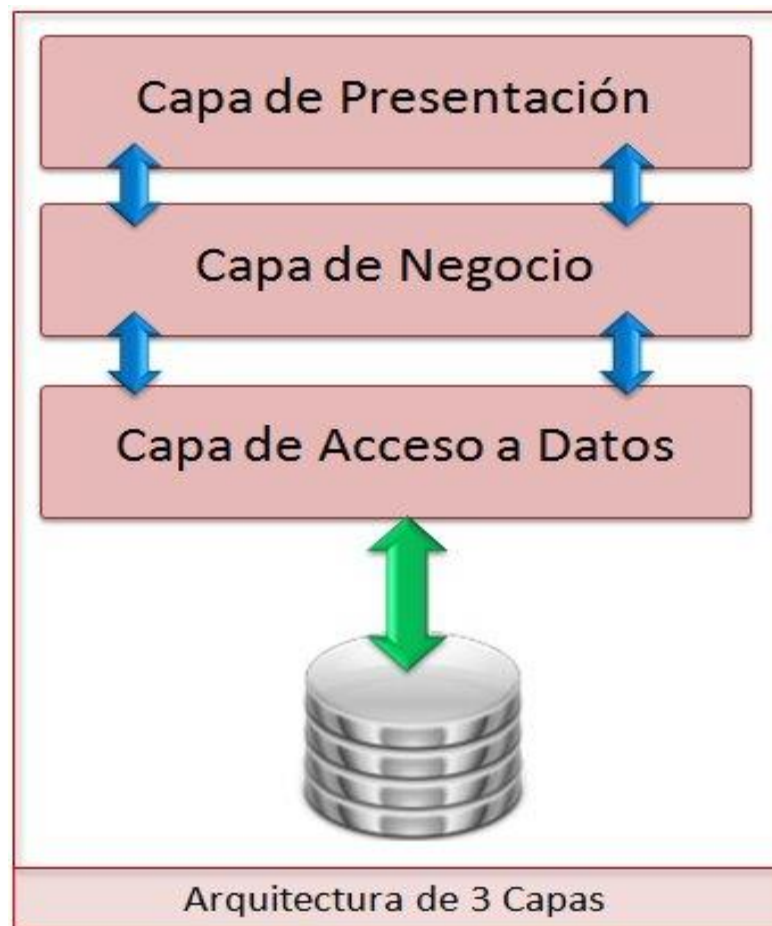


Figura 1. Arquitectura en tres capas

3.3 Patrón arquitectónico

Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. (31)

3.3.1 Modelo-Vista-Controlador (MVC)

Existen muchos estilos arquitectónicos entre los que se encuentra el estilo de llamada y retorno, dentro del cual se encuentra el patrón de arquitectura **Modelo-Vista-Controlador** (MVC). Éste separa la lógica de negocio, la interfaz de usuario y los datos de una aplicación en tres componentes distintos. Se utiliza puesto que se tiene una capa de acceso a datos donde van a estar todas las clases que de una forma u otra piden información a una base de datos.

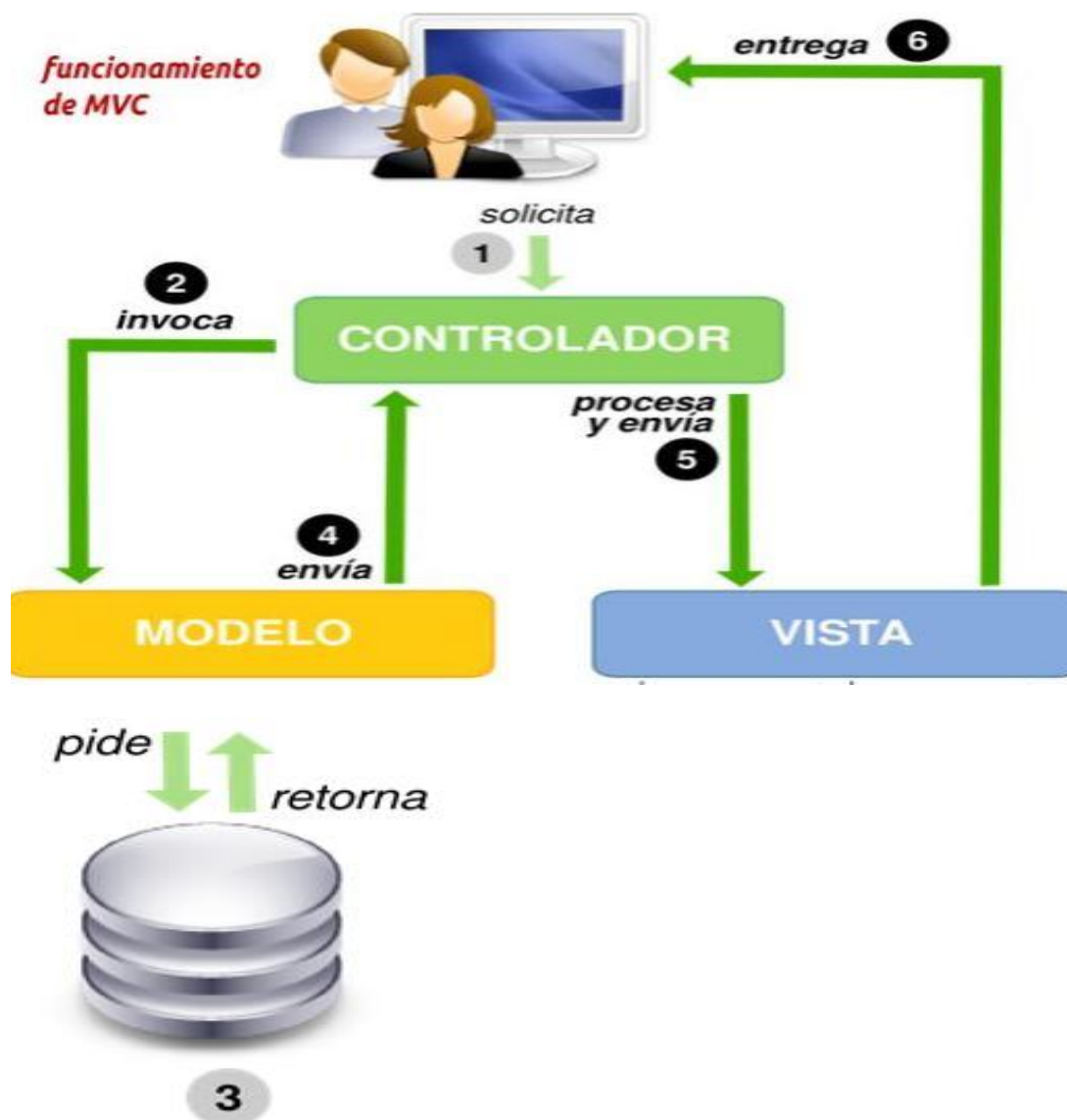


Figura 2. Patrón Modelo – Vista –Controlador

La capa **Modelo** compuesta por el paquete **Model** contiene todas las clases que tienen el código relacionado con el acceso a datos, para que este sea lo más genérico posible y se pueda reutilizar en otras situaciones y proyectos. Se incluirán consultas a las bases de datos y validaciones de entrada de datos. Se evidencia dentro del modelo la clase **DBHelper**, la cual contiene varios métodos que envían a ejecutar consultas SQL al gestor obteniendo y devolviendo los resultados de las mismas para luego ser procesados.

La capa **Vista** compuesta por el paquete **Res** contiene todas las clases que poseen el código representando o lo que es igual a la parte que será visualizada en pantalla por el usuario. Se evidencian dentro de este paquete las clases **Menu** y **Layout**, interfaces visuales que son mostradas al usuario para que el mismo interactúe con la aplicación.

La capa **Controladora** compuesta por el paquete **Utils** que contiene las clases que ejecutan la lógica de la aplicación, realizan llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. Se evidencian en este paquete varias clases; CustomInfoWindow, DatabaseFileArchive, entre otras, las cuales se encargan de recopilar toda la información y enviar los datos a las clases de la capa Vista.

A continuación, se muestra como queda estructurado el proyecto haciendo uso del patrón MVC:

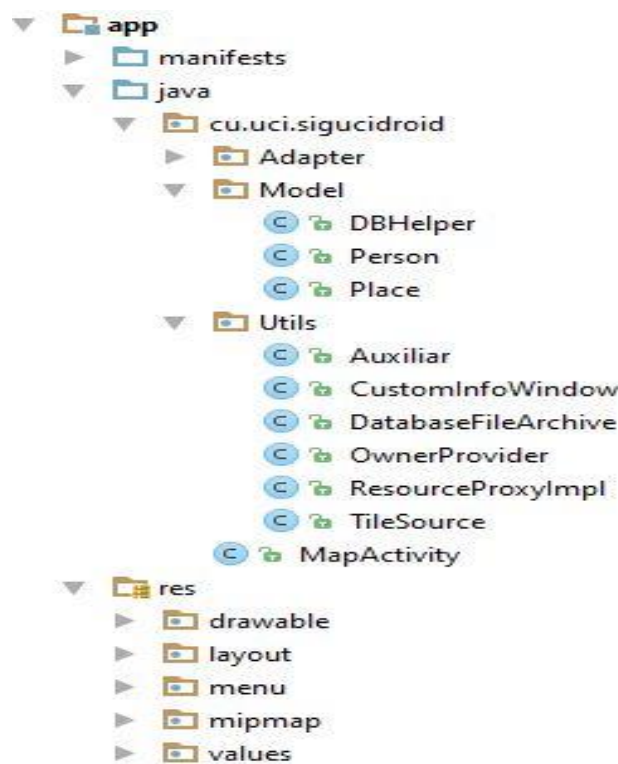


Figura 3. Estructura del proyecto utilizando el patrón MVC

3.4 Patrones de diseño

Un patrón de diseño identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades, además de que ayuda a construir clases y a estructurar sistemas de clases. Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. (32)

3.4.1 Patrones para asignar responsabilidades (GRASP)

Los patrones de principios generales para asignar responsabilidades (GRASP) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. (33)

- Experto: este patrón es el encargado de asignar una responsabilidad al experto en información, es decir, la clase que posee la información necesaria y suficiente para ejecutar la responsabilidad asignada. En la aplicación, este patrón se evidencia en la clase MapActivity.java.

A continuación se muestra un fragmento de código donde este patrón es utilizado:

```
public class MapActivity extends AppCompatActivity {

    private MapView mapView;
    private ResourceProxy mResourceProxy;
    private Double latitude;
    private Double longitude;
    private Integer[] dms;
    private DBHelper mDBHelper;
    private Dao dao;
    ArrayList<Marker> routeNodes;
    ArrayList<PointXY> routeNodesXY;
    private AlertDialog dialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        routeNodes = new ArrayList<Marker>();
        routeNodesXY = new ArrayList<PointXY>();
        loadMap();
        initSearch();
        try {
            loadPlaces();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        //addPlaceToMap();
    }
}
```

Figura 4. Fragmento de código de la utilización de patrón GRASP Experto

- Bajo Acoplamiento: este patrón consiste en mantener las clases lo menos relacionadas posible, de forma tal que, al producirse un cambio en alguna clase, se tenga el mínimo de repercusión

en las otras. En la aplicación, la utilización de este patrón se evidencia en las clases Person.java, Place.java y Auxiliar.java porque tienen la menor dependencia posible de otras clases.

- Alta cohesión: cada elemento del diseño debe realizar una labor única dentro del sistema, lo cual expresa que la información que almacena una clase debe ser coherente y estar en la mayor medida de lo posible relacionada con ella. Este patrón es el encargado de asignar responsabilidades de manera que la información que se almacena en una clase, sea la necesaria y esté bien delimitada. Este patrón se refleja en las clases CustomInfoWindow.java, OwnerProvider.java, ResourceProxyImpl.java y TileSource.java.

3.5 Tarjetas Clases – Responsabilidad – Colaborador (CRC)

La técnica CRC propone una forma de trabajo en equipo, para encontrar los objetos del dominio de la aplicación, sus responsabilidades y su colaboración con otros para realizar tareas. Las tarjetas CRC registran el nombre de las clases, sus responsabilidades y las clases con la que colaboran. Las principales características de las tarjetas CRC son: (34)

- Identificación de clases y asociaciones que participan del diseño del sistema.
- Obtención de las responsabilidades que debe cumplir cada clase.
- Establecimiento de cómo una clase colabora con otras clases para cumplir con sus responsabilidades.

A continuación, se muestran algunas de estas tarjetas CRC:

Activity_map	
Responsabilidad:	Colaborador:
Muestra una interfaz de inicio de la aplicación.	Content_map

Tabla 13. Tarjeta CRC Activity_map

Content_map	
Responsabilidad:	Colaborador:
Muestra la tabla donde se realizan los diferentes tipos de búsquedas en la aplicación.	Content_map

Tabla 14. Tarjeta CRC Content_map

Bonuspack_bubble	
Responsabilidad:	Colaborador:

Muestra la burbuja en el mapa donde se visualiza el tipo de estructura y su nombre.	
---	--

Tabla 15. Tarjeta CRC Bonuspack_bubble

Person_list	
Responsabilidad:	Colaborador:
Es un List_view encargado de listar los usuarios.	Person_list_item

Tabla 16. Tarjeta CRC Person_list

Para consultar el resto de las Tarjetas CRC ver **Anexo 1**.

3.6 Tareas de ingeniería

Las tareas de ingeniería (TI) son escritas por los desarrolladores, a partir de las Historias de usuario creadas por el cliente, las cuales brindan una mayor información para la implementación. Las tareas de ingeniería se representan mediante tablas, divididas por las siguientes secciones.

- **Número tarea:** los números deben ser consecutivos.
- **Número Historia de usuario:** número de la historia de usuario a la que pertenece la tarea.
- **Nombre tarea:** nombre que identifica a la tarea.
- **Tipo de tarea:** las tareas pueden ser de: Desarrollo, Corrección, Mejora, Otra (Especificar).
- **Puntos estimados:** tiempo estimado en semanas que se le asignará a su desarrollo.
- **Fecha de inicio:** fecha en que inicia el desarrollo de la tarea.
- **Fecha fin:** fecha en que finaliza el desarrollo de la tarea.
- **Programador responsable:** nombre y apellidos del programador.
- **Descripción:** breve descripción de la tarea.

A continuación, se muestran las TI realizadas a cada Historia de usuario:

Tarea de ingeniería	
Número de la tarea: 1	Historia de usuario: 1

Nombre de la Tarea: Buscar estructura	
Tipo: Desarrollo.	Puntos Estimados: 3
Fecha Inicio: 4/4/2016	Fecha fin: 22/4/2016
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario abre la aplicación, selecciona la opción Buscar estructura. Escoge en el menú el tipo de estructura que desea buscar, especifica el nombre de esa estructura y elige el botón Buscar. Luego de esto, la aplicación resalta en el mapa el lugar donde se encuentra localizada dicha estructura.	

Tabla 17. TI # 1 Buscar estructura

Tarea de ingeniería	
Número de la tarea: 2	Historia de usuario: 2
Nombre de la Tarea: Ubicar persona	
Tipo: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 25/4/2016	Fecha fin: 6/5/2016
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario abre la aplicación, selecciona la opción Ubicar persona. Entra en el menú el nombre o usuario de la persona que se desea ubicar y elige el botón Buscar persona. Luego de esto, la aplicación muestra un cartel con datos específicos de este individuo y además especifica en el mapa el lugar donde radica dicha persona.	

Tabla 18. TI # 2 Ubicar persona

Tarea de ingeniería	
Número de la tarea: 3	Historia de usuario: 3
Nombre de la Tarea: Actualizar base de datos	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 9/5/16	Fecha fin: 13/5/16
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	

Descripción: El usuario se conecta a la WIFI desde el teléfono y selecciona la opción Actualizar base de datos. Inmediatamente la aplicación se conecta a la base de datos de la ubicación de las personas y esta información se descarga para el teléfono, actualizándose la que se encontraba anteriormente.

Tabla 19. TI # 3 Actualizar base de datos

Tarea de ingeniería	
Número de la tarea: 4	Historia de usuario: 4
Nombre de la Tarea: Trazar camino	
Tipo: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 16/5/16	Fecha fin: 18/5/16
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario abre la aplicación, selecciona la opción Trazar camino. Luego especifica en el mapa el camino que desea trazar, haciendo puntos sobre este.	

Tabla 20. TI # 4 Trazar camino

Tarea de ingeniería	
Número de la tarea: 5	Historia de usuario: 5
Nombre de la Tarea: Dibujar ruta	
Tipo: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 19/5/16	Fecha fin: 21/5/16
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario abre la aplicación, selecciona la opción Trazar camino y realiza las operaciones de esta funcionalidad, luego selecciona la opción Dibujar ruta, esta muestra en el mapa una ruta, surgiendo de los puntos antes señalados en la funcionalidad Trazar camino.	

Tabla 21. TI # 5 Dibujar ruta

Tarea de ingeniería	
Número de la tarea: 6	Historia de usuario: 6
Nombre de la Tarea: Medir distancia	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23/5/16	Fecha fin: 27/5/16
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario abre la aplicación, selecciona la opción Trazar camino y Dibujar ruta, realizando en ambos casos las operaciones que producen dichas funcionalidades, luego selecciona la opción Medir distancia y automáticamente se muestra la distancia de la ruta especificada.	

Tabla 22. TI # 6 Medir distancia

Tarea de ingeniería	
Número de la tarea: 7	Historia de usuario: 7
Nombre de la Tarea: Limpiar	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 30/5/16	Fecha fin: 3/6/16
Programador(es) responsable(s): Jesús Magdiel Alonso García, Yilian Martínez González	
Descripción: El usuario, después de haber realizado una búsqueda de estructura o una ubicación de persona, elige la opción Limpiar, esta borra la operación antes realizada y muestra el mapa con todas las estructuras incorporadas nuevamente.	

Tabla 23. TI # 7 Limpiar

Conclusiones del capítulo

En este capítulo se llevaron a cabo las fases de diseño e implementación de la aplicación, propuestas por la metodología XP. Se definieron los patrones arquitectónicos y de diseño, usados con el fin de lograr una mejor organización de los elementos que dan forma a la aplicación. A su vez se especificaron las tarjetas CRC permitiendo identificar y organizar las clases orientadas a objetos. Se describieron las tareas de ingeniería para cada historia de usuario, necesarias para la implementación del sistema.

CAPÍTULO IV: PRUEBA

4.1 Introducción

En el presente capítulo, según dicta la metodología de desarrollo XP, se documenta la fase de Prueba. Se muestran los resultados de las pruebas unitarias y de aceptación realizadas a la aplicación.

4.2 Validación de la propuesta

Uno de los pilares de la metodología utilizada es el proceso de pruebas. XP anima a realizar pruebas constantemente, tanto como sea posible. Esto permite aumentar la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. (35)

XP divide las pruebas del sistema en dos grupos: pruebas unitarias (encargadas de verificar el código y diseñada por los programadores) y pruebas de aceptación (destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente). (35)

4.2.1 Pruebas unitarias

La base de este método es hacer pruebas en pequeños fragmentos del código de la aplicación, estos fragmentos deben ser unidades estructurales del programa, encargados de una tarea específica. En programación orientada a objetos se puede afirmar que estas unidades son los métodos o las funciones que se tienen definidos. El objetivo de estas pruebas es el aislamiento de partes del código y la demostración de que no contienen errores. Estas no generan artefactos y no son directamente palpables para el cliente. (35)

Resultados de las pruebas unitarias

Las pruebas unitarias fueron ejecutadas sistemáticamente, cada vez que se terminaba de implementar cada una de las iteraciones. Dichas pruebas fueron desarrolladas utilizando la herramienta Android JUnit. Esta herramienta permite probar componentes específicos mediante clases de casos de prueba. Estas clases proporcionan métodos auxiliares para la creación de objetos de imitación y métodos que ayudan a controlar el ciclo de vida de una aplicación.

A continuación, se muestran los resultados de las pruebas realizadas a las funcionalidades:

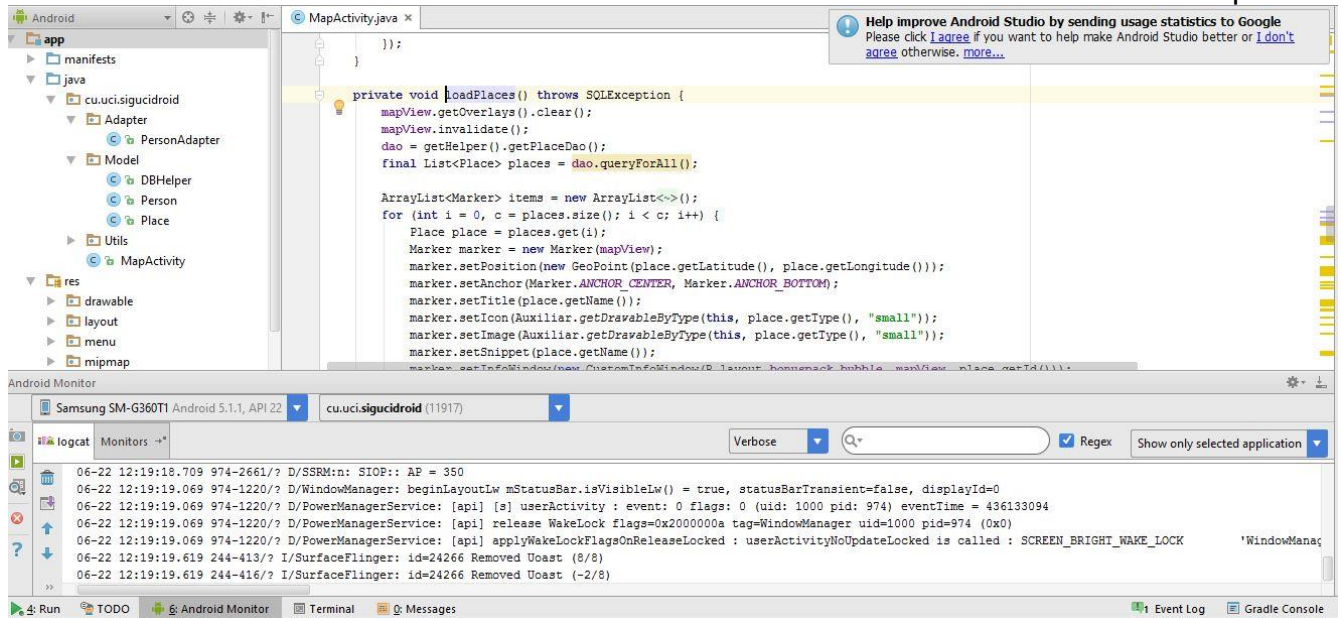


Figura 5. Prueba unitaria Buscar estructura

Para consultar las pruebas unitarias realizadas a las demás funcionalidades consultar el **Anexo 2**.

4.2.2 Pruebas de aceptación

Las pruebas de aceptación son un tipo de pruebas de caja negra. Son especificadas por el cliente y son creadas en base a las Historias de usuario. Son las encargadas de comprobar que las funcionalidades desarrolladas cumplan con las expectativas del cliente. Estas pruebas indican el camino a seguir en cada iteración, mostrándole al equipo de desarrollo qué camino seguir y en que funcionalidades debe poner mayor atención.

El objetivo de estas pruebas es verificar los requisitos, por este motivo, los propios requisitos del sistema son la principal fuente de información a la hora de construir las pruebas de aceptación. Estos casos de pruebas son realizados por el cliente.

A continuación, se muestran los casos de prueba de aceptación correspondientes a las Historias de usuario de cada iteración.

Caso de prueba	
Código: PA-1	Historia de usuario: 1
Nombre: Buscar estructura	
Descripción: Prueba de la funcionalidad Buscar estructura.	
Condiciones de ejecución: El usuario debe seleccionar la opción Buscar estructura.	

<p>Entrada/ Pasos de ejecución: El usuario selecciona la opción Buscar estructura, después selecciona el tipo de estructura que desea buscar y por último especifica la estructura en particular que quiere localizar.</p>
<p>Resultado esperado: Se indica en el mapa donde se localiza la estructura deseada por el usuario. Además se muestra información detallada de la estructura.</p>
<p>Evaluación: Satisfactoria.</p>

Tabla 24. CP # 1 Buscar estructura

Caso de prueba	
Código: PA-2	Historia de usuario: 2
Nombre: Ubicar persona	
Descripción: Prueba de la funcionalidad Ubicar persona.	
Condiciones de ejecución: El usuario debe seleccionar la opción Ubicar persona.	
Entrada/ Pasos de ejecución: El usuario selecciona la opción Ubicar persona, especifica por donde se quiere filtrar la búsqueda ya sea por nombre o usuario, luego de haber especificado esto ingresa el valor y presiona el botón Ubicar.	
Resultado esperado: Se indica en el mapa donde se encuentra la persona deseada por el usuario. Además, se muestra información de dicha persona.	
Evaluación: Satisfactoria.	

Tabla 25. CP # 2 Ubicar persona

Caso de prueba	
Código: PA-3	Historia de usuario: 3
Nombre: Actualizar base de datos	
Descripción: Prueba de la funcionalidad Actualizar base de datos.	
Condiciones de ejecución: El usuario debe conectarse a la WIFI y seleccionar la opción Actualizar base de datos.	

Entrada/ Pasos de ejecución: El usuario se conecta a la WIFI desde el teléfono y selecciona la opción Actualizar base de datos. Inmediatamente la aplicación se conecta a la base de datos de la ubicación de las personas y esta información se descarga para el teléfono.
Resultado esperado: La base de datos contenida en el teléfono se actualiza satisfactoriamente.
Evaluación: Satisfactoria.

Tabla 26. CP # 3 Actualizar base de datos

Caso de prueba	
Código: PA-4	Historia de usuario: 4
Nombre: Trazar camino.	
Descripción: Prueba de la funcionalidad Trazar camino.	
Condiciones de ejecución: El usuario debe seleccionar la opción Trazar camino.	
Entrada/ Pasos de ejecución: El usuario selecciona la opción Trazar camino y selecciona, haciendo puntos en el mapa, el camino que quiere trazar.	
Resultado esperado: Se muestran en el mapa los puntos del camino.	
Evaluación: Satisfactoria.	

Tabla 27. CP # 4 Trazar camino

Caso de prueba	
Código: PA-5	Historia de usuario: 5
Nombre: Dibujar ruta.	
Descripción: Prueba de la funcionalidad Dibujar ruta.	
Condiciones de ejecución: El usuario debe seleccionar primero la opción Trazar camino y después elegir la opción Dibujar ruta.	
Entrada/ Pasos de ejecución: El usuario escoge la opción Dibujar ruta.	

Resultado esperado: Se obtiene una ruta en el mapa uniéndose todos los puntos realizados en la funcionalidad Trazar camino.
Evaluación: Satisfactoria.

Tabla 28. CP # 5 Dibujar ruta

Caso de prueba	
Código: PA-6	Historia de usuario: 6
Nombre: Medir distancia.	
Descripción: Prueba de la funcionalidad Medir distancia.	
Condiciones de ejecución: El usuario debe seleccionar primero las opciones Trazar camino y Dibujar ruta. Después elige la opción Medir distancia.	
Entrada/ Pasos de ejecución: El usuario elige la opción Medir distancia.	
Resultado esperado: El usuario obtiene el resultado de la funcionalidad Medir distancia.	
Evaluación: Satisfactoria.	

Tabla 29. CP # 6 Medir distancia

Caso de prueba	
Código: PA-7	Historia de usuario: 7
Nombre: Limpiar.	
Descripción: Prueba de la funcionalidad Limpiar.	
Condiciones de ejecución: El usuario, debe de haber realizado una búsqueda de estructura o una ubicación de persona.	
Entrada/ Pasos de ejecución: El usuario elige la opción Limpiar.	
Resultado esperado: Se borra la operación antes realizada y muestra el mapa con todas las estructuras incorporadas nuevamente.	
Evaluación: Satisfactoria.	

Tabla 30. CP # 7 Limpiar

Resultados de las pruebas aceptación

En la etapa de pruebas se encontraron un número de tres no conformidades, las cuales fueron resueltas con éxito.

Algunas de las deficiencias identificadas en las iteraciones fueron las siguientes:

- ✓ No se hallaba a la persona que se quería ubicar, entrando su usuario.
- ✓ No se hallaba a la estructura que se quería buscar.
- ✓ No se mostraba el resultado de la medición de distancia.

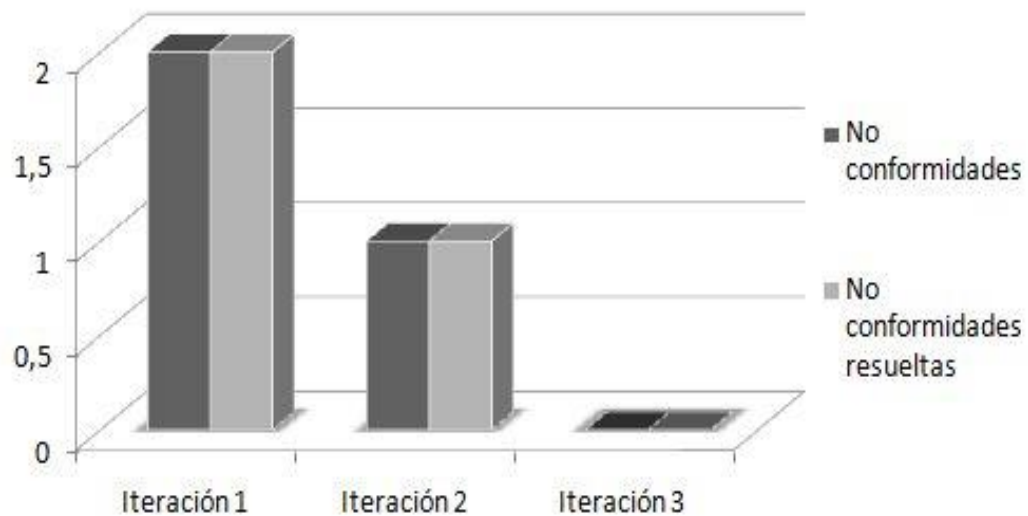


Figura 6. Resultados de las pruebas de aceptación en cada iteración

Conclusiones del capítulo

En este capítulo fueron generados y desarrollados los artefactos propuestos por la metodología XP en su fase de Prueba. La ejecución de las pruebas unitarias al código de la aplicación definidas por los desarrolladores, así como de las pruebas de aceptación conjuntamente con el cliente, permitieron presentar los resultados arrojados en cada iteración logrando finalmente una aplicación que responde al conjunto de funcionalidades identificadas.

CONCLUSIONES GENERALES

Con el propósito de dar cumplimiento al objetivo general y a la problemática planteada en el presente trabajo, fueron resueltas cada una de las tareas trazadas al inicio de la investigación de forma satisfactoria. Se profundizó en el conocimiento de los sistemas de información geográfica, así como los mecanismos necesarios para su implementación. Se obtuvo, como resultado del trabajo realizado, el Sistema de Información Geográfica UCI para dispositivos con Sistema Operativo Android.

Para guiar el proceso de desarrollo de la aplicación se aplicó la metodología XP. Mediante esta metodología de desarrollo fueron generados los artefactos necesarios para la implementación y validación de la solución. Por tanto, se puede concluir:

- ✓ La elaboración del marco teórico de la investigación a través de la definición de conceptos fundamentales para su comprensión y el estudio del estado del arte realizado, de las soluciones existentes dedicadas a los SIG para dispositivos que operan con Android, permitió al equipo de desarrollo establecer un punto de partida de la posición actual de estos sistemas.
- ✓ La implementación de las funcionalidades de la aplicación, acorde a las pautas de diseño, garantizó el cumplimiento de lo establecido en el objetivo general de la investigación.
- ✓ Las etapas de Exploración y Planificación definidas por la metodología de desarrollo seleccionada permitieron cumplir con el plan de entrega predefinido con el cliente, posibilitando el terminado de la aplicación por iteraciones y la integración de un producto final completamente funcional.
- ✓ Se realizaron las pruebas necesarias, comprobando el correcto funcionamiento de la aplicación.

RECOMENDACIONES

Luego de la experiencia adquirida durante la realización de la investigación, se recomienda:

- ✓ Desarrollar la aplicación SIGUCIDROID para teléfonos inteligentes con sistema operativo IOS, así como también para Windows Phone.
- ✓ Implementar la funcionalidad calcular área.

REFERENCIAS

1. Aguirrebeña, Lorena Flavia. Geografía Virtual. [En línea] [Citado el: 12 de noviembre de 2015.] <http://virtualgeografia.blogspot.com/2009/04/la-importancia-del-mapa.html>.
2. Langle Campos, Ruben. Sistema de Información Geográfica. [En línea] [Citado el: 20 de noviembre de 2015.] <https://langleruben.wordpress.com/%C2%BFque-es-un-sig/>.
3. Fernandez Nuñez, Hector Manuel. SIG-ESAC. Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba . La Habana : s.n, 2006.
4. ALEGSA. DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA. [En línea] [Citado el: 12 de junio de 2016.] <http://www.alegsa.com.ar/Dic/sistema.php>.
5. Geoenseñanza. saber.ula.ve. [En línea] 1 de Enero de 2006. [Citado el: 13 de Febrero de 2016.] <http://www.saber.ula.ve/bitstream/123456789/27748/1/articulo9.pdf>.
6. Real Academia Española. Diccionario de la Real Academia Española. [En línea] 2005. <http://buscon.rae.es/drae/>.
7. World Wide Web Consortium. World Wide Web Consortium. [En línea] 2004. <http://w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
8. Joubert, Alejandro. Eskola 2.0. España : s.n., 2011.
9. gvSIG. [En línea] [Citado el: 20 de noviembre de 2015.] <http://www.gvsig.com/es/productos/gvsig-mobile>.
10. Vessuri, Matias. iPodTota. [En línea] [Citado el: 3 de diciembre de 2015.] <http://www.ipodtotal.com/software-iphone/offmaps-2>.
11. Google Play. [En línea] [Citado el: 15 de febrero de 2016.] https://play.google.com/store/apps/details?id=net.osmand&hl=es_419.
12. Digital Millennium Copyright Act (DMCA). APPSAPK.Osmand. [En línea] [Citado el: 17 de Febrero de 2016.] <http://www.appsapk.com/osmand-maps-navigation/>.
13. SIG-Rutas. Ramirez Martín, Carlos Enrique, García Avila, Adrian y Pérez Vazquez, Maridalia. La Habana : s.n., 12 de diciembre de 2011.
14. EcuRed. Sistema Operativo. [En línea] [Citado el: 14 de noviembre de 2015.] http://www.ecured.cu/index.php/Sistema_operativo.
15. Que es Android. [En línea] [Citado el: 10 de diciembre de 2015.] <http://www.xatakandroid.com/sistema-operativo/que-es-android>.
16. CCM. Lenguajes de Programacion. [En línea] [Citado el: 20 de diciembre de 2015.] <http://es.ccm.net/contents/304-lenguajes-de-programacion>.

17. Androideity. [En línea] [Citado el: 12 de junio de 2016.] <http://androideity.com/2012/07/16/5-lenguajes-para-programar-en-android/>.
18. EXES. [En línea] [Citado el: 10 de diciembre de 2015.] http://www.mundojava.net/caracteristicas-del-lenguaje.html?Pg=java_inicial_4_1.html.
19. Programacion Java. [En línea] [Citado el: 15 de enero de 2016.] <http://programacion-laura.blogspot.com/2011/08/entorno-de-desarrollo-integrado-ide.html>.
20. Android Developers. [En línea] [Citado el: 12 de diciembre de 2015.] <http://developer.android.com/tools/studio/index.html>.
21. SlideShare. [En línea] 26 de marzo de 2014. [Citado el: 20 de diciembre de 2015.] <http://es.slideshare.net/yeniferbravo1/tecnicas-y-herramientas-para-el-desarrollo-de-software>.
22. SOFTONIC. [En línea] [Citado el: 16 de junio de 2016.] <http://android-sdk.softonic.com/>.
23. CAVSI. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] [Citado el: 22 de diciembre de 2015.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
24. Flores Cueto, Juan José y Bertolotti Zuñiga, Carmen. Diagrama de clases en UML. Diagrama de clases en UML. [En línea] [Citado el: 25 de febrero de 2016.] <http://es.scribd.com/doc/31096724/Diagrama-de-Clases-en-UML#scribd>.
25. W3C España. Guía Breve de Servicios Web. [En línea] [Citado el: 19 de junio de 2016.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
26. Gonzalez, Benjamín. DesarrolloWeb.com. WSDL para la documentación de Servicios Web. [En línea] 26 de julio de 2004. <http://www.desarrolloweb.com/articulos/1581.php>.
27. XP – Extreme Programming Ingeniería de Software. [En línea] [Citado el: 15 de diciembre de 2015.] http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html.
28. Gimson, Loraine. Metodologías ágiles y desarrollo basado en conocimientos. 2012.
29. BAHIT, E. Sunshine. El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC. [En línea] [Citado el: 13 de junio de 2016.] <http://sunshine.prod.uci.cu/book/4eb204eb0571745ff5000007/>.
30. Lopez, Eliazar. Academia. Arquitectura de n capas. [En línea] [Citado el: 14 de junio de 2016.] http://www.academia.edu/10102692/Arquitectura_de_n_capas.
31. Sánchez Juárez, Abraham. Diseño y arquitectura de software . Mexico : s.n. AL12526906 .
32. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Patrones de Diseño. Elementos de software orientados a objetos reutilizable. Madrid : Perason Educación, 2003.

33. Practicas de Software. Patrones GRASP. [En línea] [Citado el: 10 de abril de 2016.]
<http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
34. Extreme Programming. [En línea] [Citado el: 3 de mayo de 2016.]
<http://www.extremeprogramming.org/rules/crccards.html>.
35. Maite Rodríguez Corbea, Meylin Ordóñez Pérez. La Metodología XP Aplicable al Desarrollo del Software. Universidad de las Ciencias Informáticas, Cuba : s.n., 2007.

BIBLIOGRAFÍA

1. Aguirrebeña, Lorena Flavia. Geografía Virtual. [En línea] [Citado el: 12 de noviembre de 2015.] <http://virtualgeografia.blogspot.com/2009/04/la-importancia-del-mapa.html>.
2. Langle Campos, Ruben. Sistema de Información Geográfica. [En línea] [Citado el: 20 de noviembre de 2015.] <https://langleruben.wordpress.com/%C2%BFque-es-un-sig/>.
3. Fernandez Nuñez, Hector Manuel. SIG-ESAC. Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba . La Habana : s.n, 2006.
4. ALEGSA. DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA. [En línea] [Citado el: 12 de junio de 2016.] <http://www.alegsa.com.ar/Dic/sistema.php>.
5. Geoenseñanza. saber.ula.ve. [En línea] 1 de Enero de 2006. [Citado el: 13 de Febrero de 2016.] <http://www.saber.ula.ve/bitstream/123456789/27748/1/articulo9.pdf>.
6. Real Academia Española. Diccionario de la Real Academia Española. [En línea] 2005. <http://buscon.rae.es/drae/>.
7. World Wide Web Consortium. World Wide Web Consortium. [En línea] 2004. <http://w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
8. Joubert, Alejandro. Eskola 2.0. España : s.n., 2011.
9. gvSIG. [En línea] [Citado el: 20 de noviembre de 2015.] <http://www.gvsig.com/es/productos/gvsig-mobile>.
10. Vessuri, Matias. iPodTota. [En línea] [Citado el: 3 de diciembre de 2015.] <http://www.ipodtotal.com/software-iphone/offmaps-2>.
11. Google Play. [En línea] [Citado el: 15 de febrero de 2016.] https://play.google.com/store/apps/details?id=net.osmand&hl=es_419.
12. Digital Millennium Copyright Act (DMCA). APPSAPK.Osmand. [En línea] [Citado el: 17 de Febrero de 2016.] <http://www.appsapk.com/osmand-maps-navigation/>.
13. SIG-Rutas. Ramirez Martín, Carlos Enrique, García Avila, Adrian y Pérez Vazquez, Maridalia. La Habana : s.n., 12 de diciembre de 2011.
14. EcuRed. Sistema Operativo. [En línea] [Citado el: 14 de noviembre de 2015.] http://www.ecured.cu/index.php/Sistema_operativo.
15. Que es Android. [En línea] [Citado el: 10 de diciembre de 2015.] <http://www.xatakandroid.com/sistema-operativo/que-es-android>.
16. CCM. Lenguajes de Programacion. [En línea] [Citado el: 20 de diciembre de 2015.] <http://es.ccm.net/contents/304-lenguajes-de-programacion>.

17. Androideity. [En línea] [Citado el: 12 de junio de 2016.] <http://androideity.com/2012/07/16/5-lenguajes-para-programar-en-android/>.
18. EXES. [En línea] [Citado el: 10 de diciembre de 2015.] http://www.mundojava.net/caracteristicas-del-lenguaje.html?Pg=java_inicial_4_1.html.
19. Programacion Java. [En línea] [Citado el: 15 de enero de 2016.] <http://programacion-laura.blogspot.com/2011/08/entorno-de-desarrollo-integrado-ide.html>.
20. Android Developers. [En línea] [Citado el: 12 de diciembre de 2015.] <http://developer.android.com/tools/studio/index.html>.
21. SlideShare. [En línea] 26 de marzo de 2014. [Citado el: 20 de diciembre de 2015.] <http://es.slideshare.net/yeniferbravo1/tecnicas-y-herramientas-para-el-desarrollo-de-software>.
22. SOFTONIC. [En línea] [Citado el: 16 de junio de 2016.] <http://android-sdk.softonic.com/>.
23. CAVSI. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] [Citado el: 22 de diciembre de 2015.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
24. Flores Cueto, Juan José y Bertolotti Zuñiga, Carmen. Diagrama de clases en UML. Diagrama de clases en UML. [En línea] [Citado el: 25 de febrero de 2016.] <http://es.scribd.com/doc/31096724/Diagrama-de-Clases-en-UML#scribd>.
25. W3C España. Guía Breve de Servicios Web. [En línea] [Citado el: 19 de junio de 2016.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
26. Gonzalez, Benjamín. DesarrolloWeb.com. WSDL para la documentación de Servicios Web. [En línea] 26 de julio de 2004. <http://www.desarrolloweb.com/articulos/1581.php>.
27. XP – Extreme Programming Ingeniería de Software. [En línea] [Citado el: 15 de diciembre de 2015.] http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html.
28. Gimson, Loraine. Metodologías ágiles y desarrollo basado en conocimientos. 2012.
29. BAHIT, E. Sunshine. El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC. [En línea] [Citado el: 13 de junio de 2016.] <http://sunshine.prod.uci.cu/book/4eb204eb0571745ff5000007/>.
30. Lopez, Eliazar. Academia. Arquitectura de n capas. [En línea] [Citado el: 14 de junio de 2016.] http://www.academia.edu/10102692/Arquitectura_de_n_capas.
31. Sánchez Juárez, Abraham. Diseño y arquitectura de software . Mexico : s.n. AL12526906 .
32. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Patrones de Diseño. Elementos de software orientados a objetos reutilizable. Madrid : Perason Educación, 2003.

33. Practicas de Software. Patrones GRASP. [En línea] [Citado el: 10 de abril de 2016.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
34. Extreme Programming. [En línea] [Citado el: 3 de mayo de 2016.] <http://www.extremeprogramming.org/rules/crccards.html>.
35. Maite Rodríguez Corbea, Meylin Ordóñez Pérez. La Metodología XP Aplicable al Desarrollo del Software. Universidad de las Ciencias Informáticas, Cuba : s.n., 2007.
36. Configurar Equipos. Que es Android: Características y Aplicaciones. [En línea] [Citado el: 20 de enero de 2016.] <http://www.configurarequipos.com/doc1107.html>.
37. Ingenieria de Software. [En línea] [Citado el: 7 de diciembre de 2015.] http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html.
38. Ian. Ingeniería de Software. Sommerville. Madrid : Pearson Educación, 2005.
39. Vera Prado, Gabriel y González Vergara, Paz. Portal Educativo. [En línea] <http://www.portaleducativo.net/quinto-basico/537/Mapas-y-planos>.
40. Android Developers. [En línea] [Citado el: 15 de diciembre de 2015.] <http://developer.android.com/intl/es/sdk/index.html>.
41. Instituto Geografico Nacional. [En línea] [Citado el: 10 de enero de 2016.] <http://www.ign.es/ign/layoutIn/actividadesSistemaInfoGeografica.do>.
42. Orozco, David. ConceptoDefinicion.de. Definicion de Mapa. [En línea] <http://conceptodefinicion.de/mapa/>.
43. Significados. Sistema. [En línea] <http://www.significados.com/sistema/>.
44. SourceForge.net. Ciclo de vida de un proyecto XP. [En línea] [Citado el: 23 de febrero de 2016.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
45. L. Bass, P. Clements, R. Kazman. Software Architecture in Practice. s.l. : Addison Wesley, 2003. 2nd Edition.
46. C. de la Torre, U.Zorrilla, M.A. Ramos,J Calvarro. Guía de Arquitectura N-Capas orientada al Dominio con .Net 4.0. s.l. : Krasis Consulting , 2010.
47. —. Guía de Arquitectura N-Capas orientada al Dominio con .Net 4.0. s.l. : Krasis Consulting, 2010.
48. Lime Creative Labs. [En línea] [Citado el: 3 de abril de 2016.] <http://www.limecreativelabs.com/mvp-android/>.
49. Geoenseñanza . [En línea] 1 de enero de 2006. [Citado el: 12 de enero de 2016.] <http://www.saber.ula.ve/bitstream/123456789/27748/1/articulo9.pdf>.

50. Maite Rodríguez Corbea, Meylin Ordóñez Pérez. La Metodología XP Aplicable al Desarrollo del Software. Universidad de las Ciencias Informáticas, Cuba : s.n., 2007.

ANEXO 1

Detail_place	
Responsabilidad:	Colaborador:
Muestra los detalles del lugar que ha sido seleccionado.	

Tabla 31. Tarjeta CRC Detail_place

Person_list_item	
Responsabilidad:	Colaborador:
Es un elemento del List_view, Person_list que visualiza la descripción de los usuarios.	

Tabla 32. Tarjeta CRC Person_list_item

DBHelper	
Responsabilidad:	Colaborador:
Clase encargada de gestionar toda la información referente a la base de datos, así como sus tablas.	Omrlitesqlopenhelper Person_Place

Tabla 33. Tarjeta CRC DBHelper

Person	
Responsabilidad:	Colaborador:
Clase entidad encargada de manejar la información referente a las persona: nombre completo, usuario, provincia, teléfono, apartamento.	

Tabla 34. Tarjeta CRC Person

Place	
Responsabilidad:	Colaborador:

Clase entidad encargada de manejar la información de los lugares: nombre, descripción, latitud, longitud, tipo.	
---	--

Tabla 35. Tarjeta CRC Place

PersonAdapter	
Responsabilidad:	Colaborador:
Crear un adapter para la vista Person_list_item para visualizarlo en el List_view.	Person_list_item

Tabla 36. Tarjeta CRC PersonAdapter

Auxiliar	
Responsabilidad:	Colaborador:
Encargada de cargar las imágenes dependiendo del tipo de estructura, así como los nombres y sus propiedades.	

Tabla 37. Tarjeta CRC Auxiliar

CustomInfomWindows	
Responsabilidad:	Colaborador:
Clase encargada de mostrar en el mapa la burbuja con la ubicación de la estructura a través del uso de la vista Bonuspack_bubble.	DBHelper Bonuspack_bubble

Tabla 38. Tarjeta CRC CustomInfomWindows

MapActivity	
Responsabilidad:	Colaborador:
Encargada de inicializar los mapas y capturar todos los eventos de la vista, es la clase principal del proyecto.	DBHelper

Tabla 39. Tarjeta CRC MapActivity

ANEXO 2

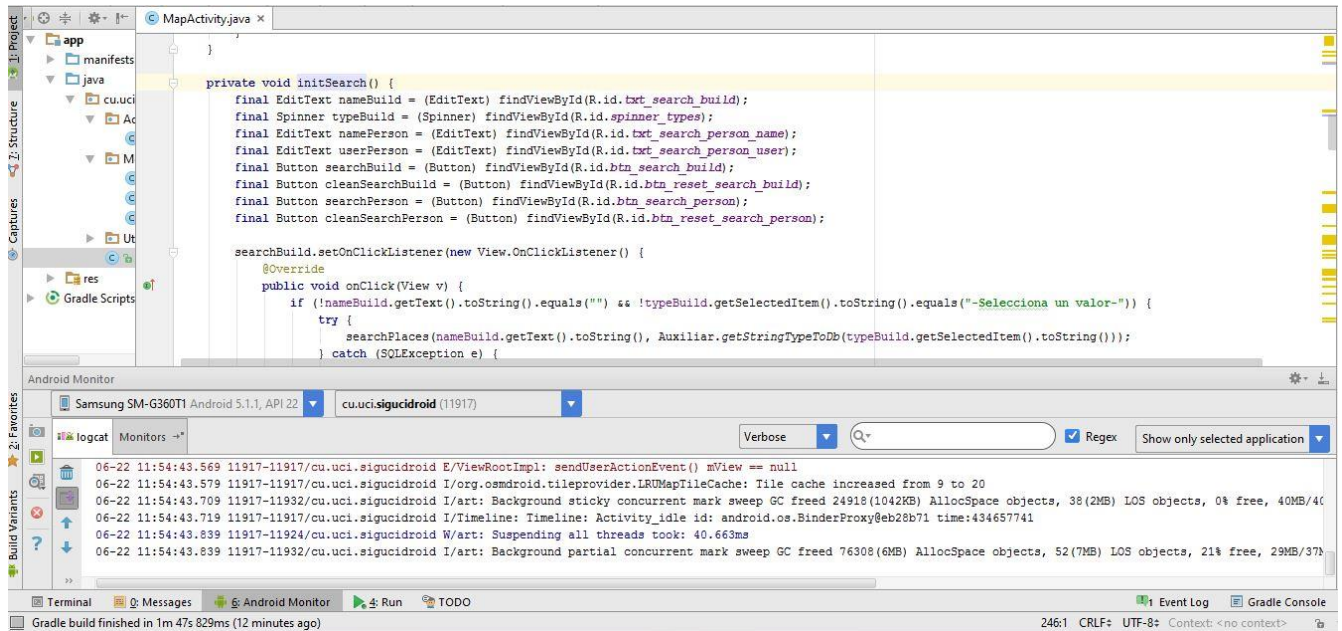


Figura 7. Prueba unitaria Ubicar persona

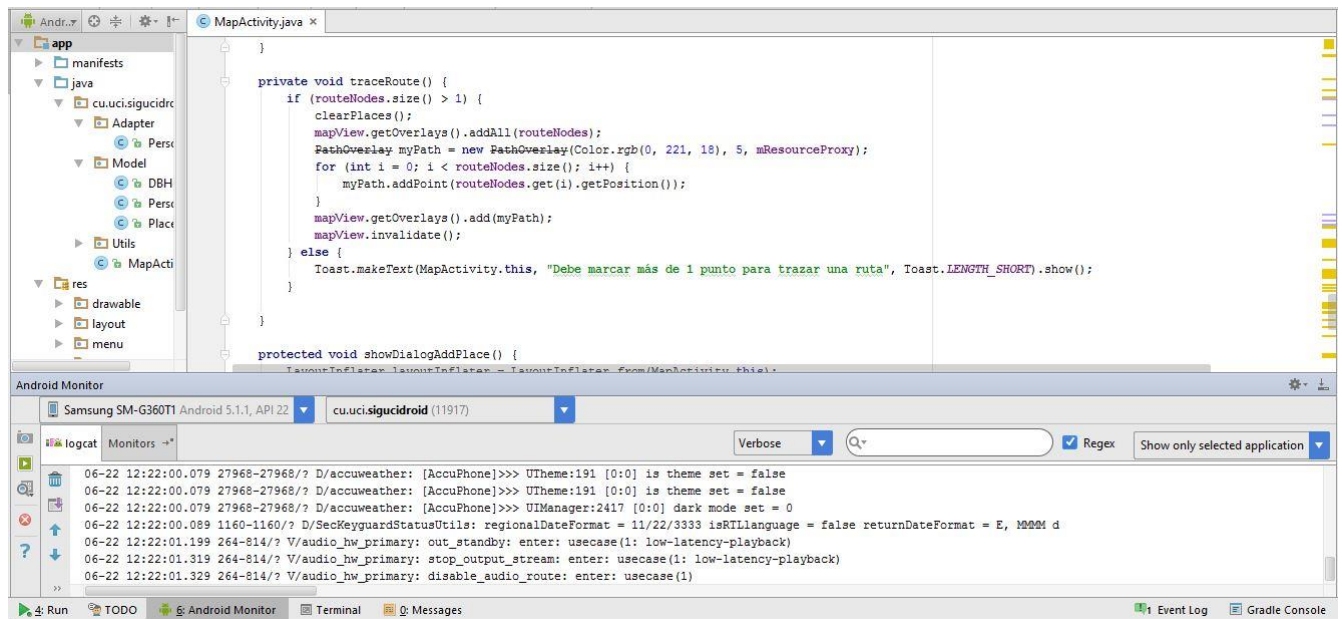


Figura 8. Prueba unitaria Trazar camino

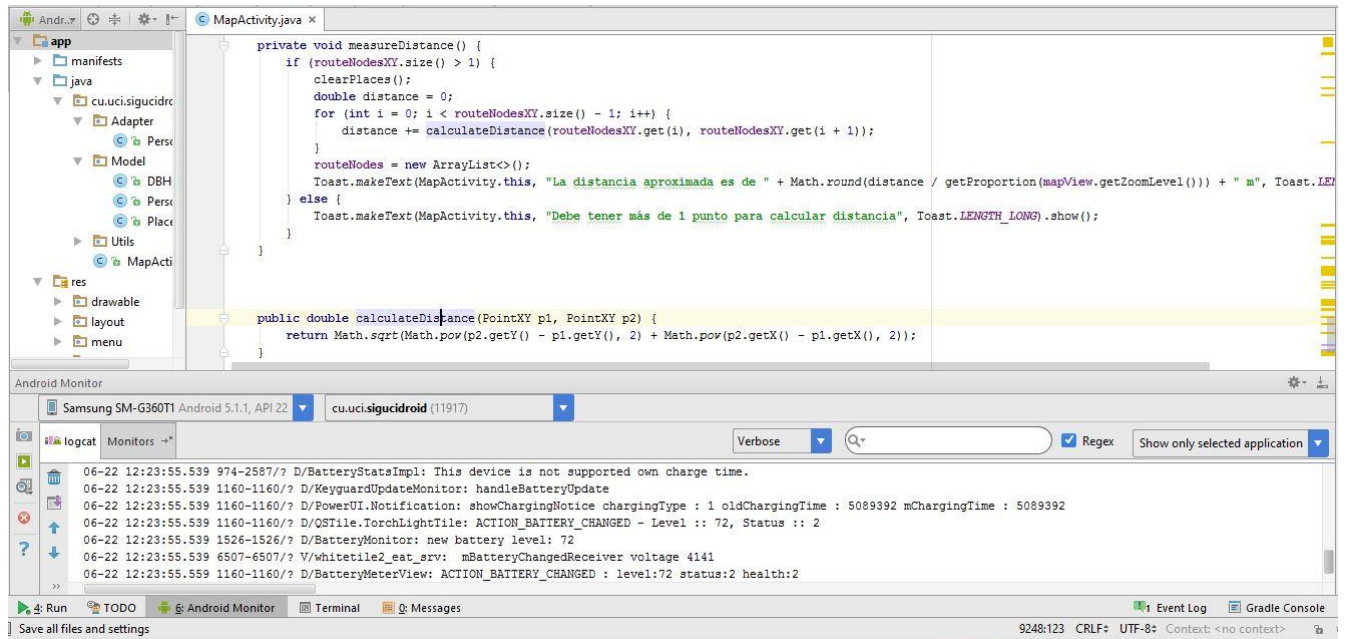


Figura 9. Prueba unitaria Medir distancia

ANEXO 3

Modelo de Entrevista

Entrevista realizada a los trabajadores del Centro de Desarrollo de Software GEYSED, para conocer las fortalezas y debilidades del sistema implementado.

1. ¿El SIGUCI web se encuentra disponible las 24 horas?
2. ¿Cómo se actualiza la información brindada por SIGUCI web?
3. ¿Con que frecuencia se actualiza la información que brinda este sistema?
4. Si en algún momento el usuario desea acceder al sistema desde un teléfono móvil, ¿se podrán realizar todas las funcionalidades sin que ocurra algún problema?
5. ¿Qué tecnologías se utilizaron para el desarrollo del SIGUCI web?