



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS

DISEÑO Y DESARROLLO DE UNA APLICACIÓN MÓVIL HÍBRIDA PARA
EL CONTROL DE AGENDA DEL ÁREA DE INGENIERÍA
CASO DE ESTUDIO: EMPRESA ANDEANTRADE S.A

MIELES MARTÍNEZ ANA CAROLINA

DIRECTOR: ING. RODRÍGUEZ FRANCISCO

“Trabajo de titulación previo a la obtención del Título de Ingeniero en Sistemas y
Computación”

Quito, septiembre de 2019

*A mis padres,
Anita y Patricio.*

*A mi hermana,
Stephany.*

*A mi abuelito,
Celio.*

AGRADECIMIENTO

Quisiera empezar dando un especial agradecimiento a Dios y a mis padres, que me han permitido ser parte de esta universidad, me han dado la oportunidad de formarme profesionalmente y me han acompañado en cada paso y en cada tropiezo; han creído en mí incluso cuando yo no lo hacía, por ello les dedico este logro que es más de ellos que mío porque sin sus consejos, cariño, apoyo y hasta sus regaños no hubiese podido llegar a donde estoy. Amarlos y agradecerlos infinitamente.

A mi hermana, le agradezco por estar todos estos años a mi lado apoyándome e impulsándome a ser cada día mejor tanto personal como profesionalmente. Gracias por cada consejo y por estar conmigo tanto en buenos como malos momentos; por enseñarme a ver la vida como un reto y también darme la fuerza para vencerla.

Agradezco particularmente a mi director de tesis, Ing. Francisco Rodríguez por estar a lo largo de este proceso, corregirme y guiarme con paciencia para poder llegar a realizarla. Agradezco también a mis revisores de tesis y a la universidad por permitirme cumplir este propósito.

Agradezco a la empresa AndeanTrade S.A por brindarme la oportunidad y la confianza de crear este proyecto orientado a ellos.

A mi familia: abuelitos, tíos y primos, por ser los pilares que forjan la persona que hoy en día soy y por permitirme ser parte de su orgullo. A mi abuelito Celio, por ser la persona más bondadosa y pura que he conocido e inspirarme a ser una persona con ideales y con sueños.

A mis amigos: Steven, Fernando, Pablo, Diego por manifestar su sincera amistad y ayudarme a no darme nunca por vencida, por aportar con su conocimiento y ser un soporte cuando más lo necesite. Gracias por su desinteresada ayuda y por los buenos momentos compartidos a lo largo de esta etapa.

Gracias a todos los que me ayudaron ya sea directa o indirectamente para permitirme consumir este logro.

Contenido

CAPÍTULO I: AndeanTrade S.A	1
1.1 Historia	1
1.2 Misión	2
1.3 Visión	2
1.4 Valores	2
1.5 Mapa de procesos	4
1.6 Diagrama Jerárquico	5
1.7 Análisis de la situación actual	14
CAPÍTULO II: Fundamentos teóricos.....	16
2.1 Metodologías ágiles	16
2.1.1 Scrum	19
2.1.2 XP	24
2.1.3 Mobile-D	26
2.2 Aplicaciones móviles.....	30
2.2.1 Aplicaciones nativas	30
2.2.2 Aplicaciones híbridas	32
2.2.3 Aplicaciones web.....	32
2.3 Herramientas de desarrollo	33

2.3.1 Ionic	33
2.3.2 React Native	35
2.3.3 PhoneGap	36
2.4 Base de datos	38
2.4.1 MySQL.....	38
2.4.2 PostgreSql.....	39
2.4.3 SqlServer	41
2.5 Selección de metodología y herramienta	43
CAPÍTULO III: DESARROLLO DE APLICATIVO UTILIZANDO METODOLOGÍA ÁGIL MOBILE-	
D.....	
3.1 Exploración	45
3.1.1 Primera etapa:	45
3.1.1.1 Identificación de actores.....	45
3.1.1.2 Definición de alcance	46
3.1.2 Segunda etapa	46
3.1.2.1 Recolección de requerimientos.....	46
3.1.3 Tercera etapa.....	47
3.1.3.1 Identificación de Procesos.....	47
3.2 Inicialización	49
3.1.1 Configuración de ambiente de desarrollo.....	49

3.2.2	Requerimientos iniciales	50
3.2.2.1	Requerimientos funcionales	50
3.2.2.1	Requerimientos no funcionales.....	50
3.2.3	Planificación inicial	50
3.2.3.1	Análisis de requerimientos iniciales	50
3.2.3.1.1	Módulo coordinador.....	51
3.2.3.1.2	Módulo ingeniero	52
3.2.4	Planificación de fases	53
3.3	Producción.....	55
3.3.1	Arquitectura de la aplicación.....	55
3.3.2	Modelo físico de la Base de datos	56
3.3.3	Descripción de módulos	57
3.3.3.1	Módulo de coordinador	57
3.3.3.2	Módulo de ingeniero	65
3.4	Estabilización	67
3.5	Pruebas	68
3.5.1.	Pruebas de carga	68
3.5.2.	Pruebas unitarias	69
CAPÍTULO IV: Entregables		74

4.1	Manual de usuario	74
4.2	Manual técnico	97
CAPÍTULO V: Recomendación		138
5.1	Conclusiones	138
5.2	Recomendaciones.....	139
Referencias bibliográficas.....		140

Contenido de Ilustraciones

Ilustración 1. Mapa de Procesos.	4
Ilustración 2. Diagrama jerárquico 1.	5
Ilustración 3. Diagrama jerárquico 2.	6
Ilustración 4. Diagrama jerárquico 3.	7
Ilustración 5. Diagrama jerárquico 4.	8
Ilustración 6. Diagrama jerárquico 5.	9
Ilustración 7. Diagrama jerárquico 6.	10
Ilustración 8. Diagrama jerárquico 7.	11
Ilustración 9. Diagrama jerárquico 8.	12
Ilustración 10. Diagrama jerárquico 9.	13
Ilustración 11. Gráfica de ciclo de vida de Mobile-D.	27
Ilustración 12. Proceso de mantenimiento.	47
Ilustración 13. Proceso de gestión de incidentes.	48
Ilustración 14. Proceso de soporte.	49
Ilustración 15. Arquitectura de la aplicación.	55
Ilustración 16. Modelo físico de la base de datos.	56
Ilustración 17. Pantalla de Login.	57
Ilustración 18. Pantalla Menú de coordinador.	58
Ilustración 19. Pantalla Gestión de servicios.	59
Ilustración 20. Pantalla Ingreso de servicios.	60
Ilustración 21. Pantalla Editar servicio.	61

Ilustración 22. Pantalla Eliminar servicio.....	62
Ilustración 23. Pantalla Ingreso de actividades.....	63
Ilustración 24. Pantalla Lista de actividades por fecha.	64
Ilustración 25. Pantalla Sobre App.....	65
Ilustración 26. Pantalla Lista de actividades.....	66
Ilustración 27. Pantalla Editar actividad.....	67
Ilustración 28. Prueba de carga.....	68
Ilustración 29. Prueba de carga 2.....	69

Contenido de tablas

Tabla 1. Diferencias entre frameworks	43
Tabla 2. Requerimientos funcionales.	50
Tabla 3. Requerimientos no funcionales.	50
Tabla 4. Requerimientos iniciales.	51
Tabla 5. Planificación de fases.....	53
Tabla 6. Prueba unitaria no.1	70
Tabla 7. Prueba unitaria no.2	71
Tabla 8. Prueba unitaria no.3	71
Tabla 9. Prueba unitaria no.4	72
Tabla 10. Prueba unitaria no.5	73
Tabla 11. Prueba unitaria no.6	73

CAPÍTULO I: AndeanTrade S.A

1.1 Historia

AndeanTrade SA es una empresa orgullosamente ecuatoriana que actualmente cuenta con sedes en Colombia, Bolivia y Estados Unidos. Formado por un equipo de Talento Humano altamente capacitado capaz de integrar servicios tecnológicos especializados, proveyendo equipamiento, servicios y soluciones de infraestructura en el sector público y privado.

Los primeros pasos de AndeanTrade SA fueron en 2000 cuando un grupo de emprendedores y personas de negocios ecuatorianos, bajo el liderazgo de OSCAR TRUJILLO crearon una empresa para generar negocios de exportación desde los países Andinos hacia el resto del mundo, por ello el nombre de la empresa. Posteriormente en el año 2001 se decide enfocar la empresa a Tecnología, justamente el área de mayor conocimiento de su principal socio. En el año 2004 se le unieron GERARDO TRUJILLO, ALVARO PRADO y FREDDY ECHEVERRIA, que empujaron a la empresa a especializarse y tomar un rumbo por verticales de acuerdo con las necesidades del cliente con todas las certificaciones y avales de la marca que se convertiría en poco tiempo en su principal fabricante como es CISCO Systems.

Con el pasar de los años rotaron algunos socios y surgieron otros nuevos los cuales aportaron a un crecimiento mayor de la compañía y ocupar un lugar importante en el mercado ecuatoriano con reconocimientos importantes por su gestión. De esta manera se empieza a internacionalizar a otros países como EE. UU., Colombia y Bolivia.

Siendo hoy una empresa sólida, en constante cambio además de estar consciente de que las personas cada vez buscan respuestas instantáneas, velocidad y seguridad en sus comunicaciones y con ella mejorar la rentabilidad y márgenes de la compañía, mejorando su productividad, AndeanTrade SA se encuentra en el proceso de fortalecer y acoplarse a las necesidades actuales

del mercado ofreciendo nuevas soluciones como Desarrollo de Software y Servicios Administrados para sus clientes.

La innovación es el elemento clave que explica la competitividad. Porter (1990) afirmó: “La competitividad de una nación depende de la capacidad de su industria para innovar y mejorar. La empresa consigue ventaja competitiva mediante innovaciones”

Con base a este modelo, AndeanTrade SA con su experiencia de más de 15 años como integradora de servicios tecnológicos especializados, ayuda a identificar, asegurar el control sobre la inversión y entrega de resultados medibles para que las Tecnologías de la Información (TI) o los nuevos modelos innovadores constituyen un factor efectivo en la cadena de valor que sostiene la gestión total de las organizaciones.

1.2 Misión

“Proveer a nuestros clientes soluciones tecnológicas de última generación, adaptadas a sus necesidades, que contribuyan al crecimiento de su negocio, generando valor para nuestros socios, colaboradores y comunidad.”

1.3 Visión

“Ser una compañía exitosa y reconocida por la implementación de soluciones tecnológicas de clase mundial hasta el año 2025.”

1.4 Valores

- **Integridad.** - Somos una empresa integral, fundamentada en Principios y Valores, por lo cual actuamos dentro de los más rigurosos principios éticos y legales.

- **Compromiso.** - Asumimos nuestro trabajo con dedicación y firmeza de ánimo, y aceptamos los retos que se presentan con la voluntad personal de llevarlos a cabo exitosamente.
- **Trabajo en equipo.** - Valoramos y fomentamos los aportes de las personas para el logro de los objetivos comunes.
- **Aprendizaje continuo.** - Es nuestra voluntad de aprendizaje permanente, con actitud positiva frente al cambio, para el óptimo desempeño individual y empresarial.
- **Adaptabilidad al cambio.** - Nuestra empresa está dispuesta a tomar medidas y hacer cambios para alcanzar las metas y objetivos.
- **Creatividad e ingenio.** - Poner en práctica la creatividad del personal en cuanto a redefinición y / o reinención de nuevos productos, estrategias, actividades y funciones con proyección de mejora.
- **Efectividad.** - Basamos nuestra gestión en el manejo adecuado de recursos y damos soluciones adecuadas a nuestros clientes.

1.5 Mapa de procesos

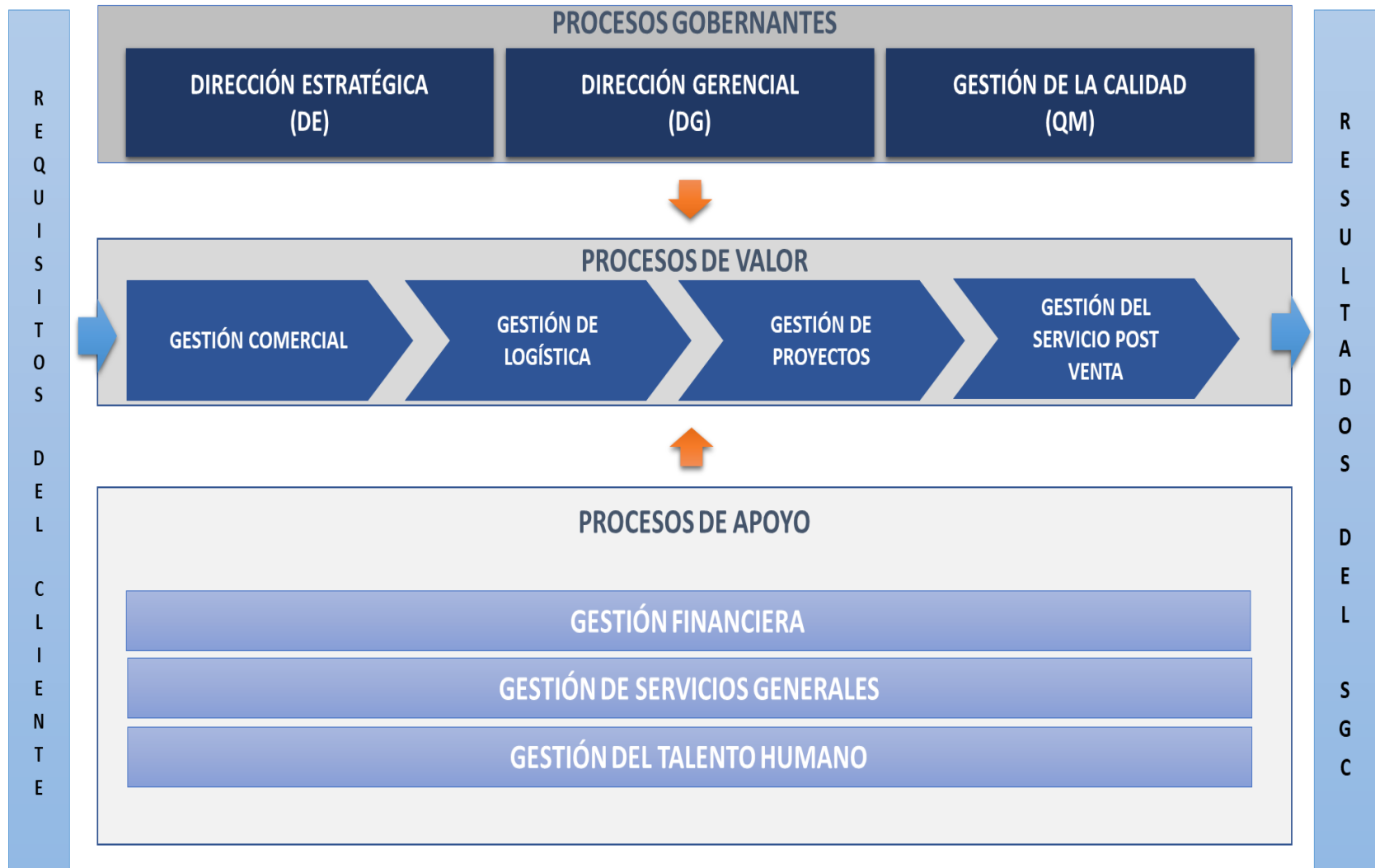


Ilustración 1. Mapa de Procesos. QM-D004 AndeanTrade S.A

1.6 Diagrama Jerárquico

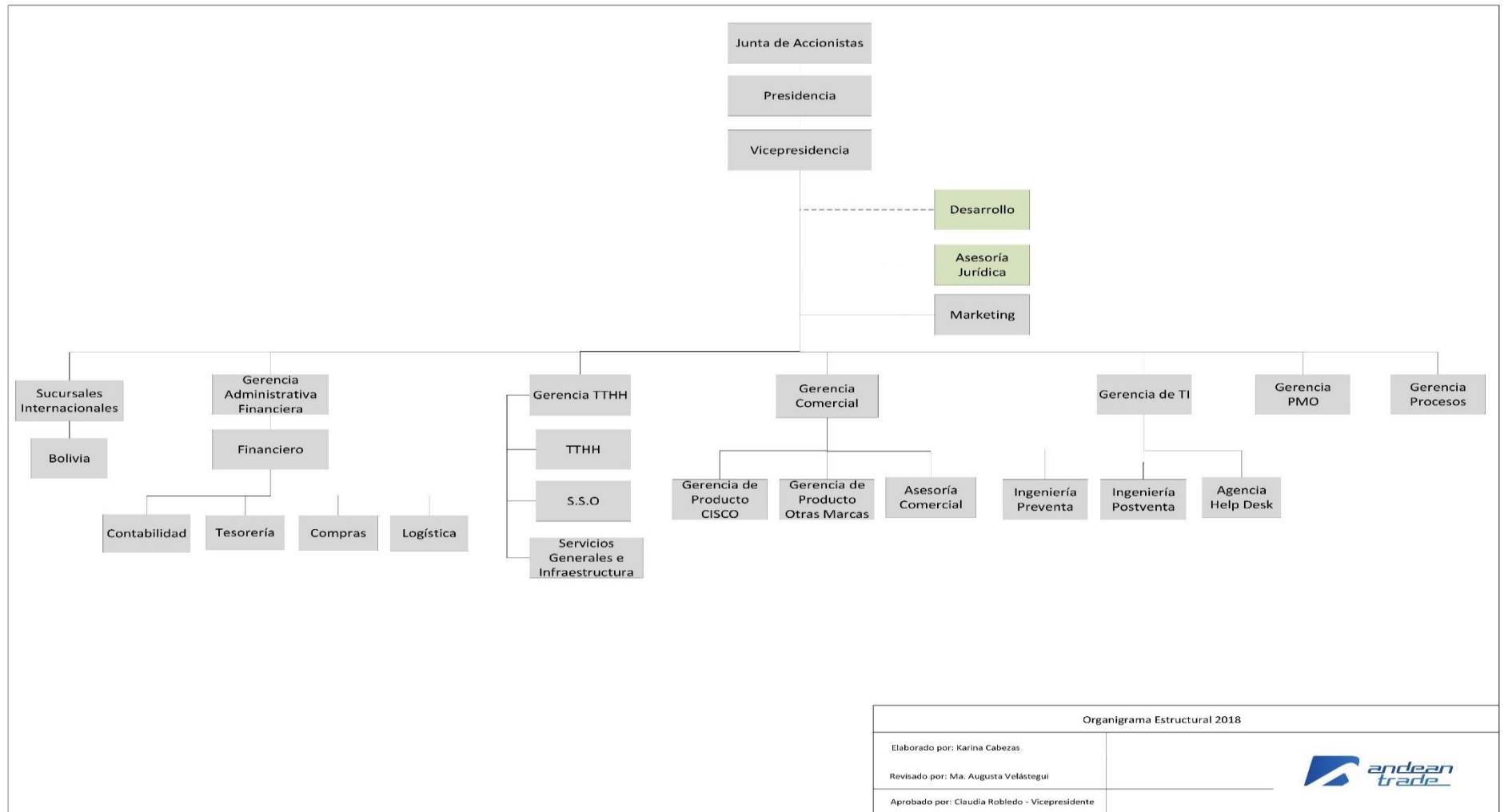


Ilustración 2. Diagrama jerárquico 1. OrganigramaV18_estructura_funcional AndeanTrade S.A

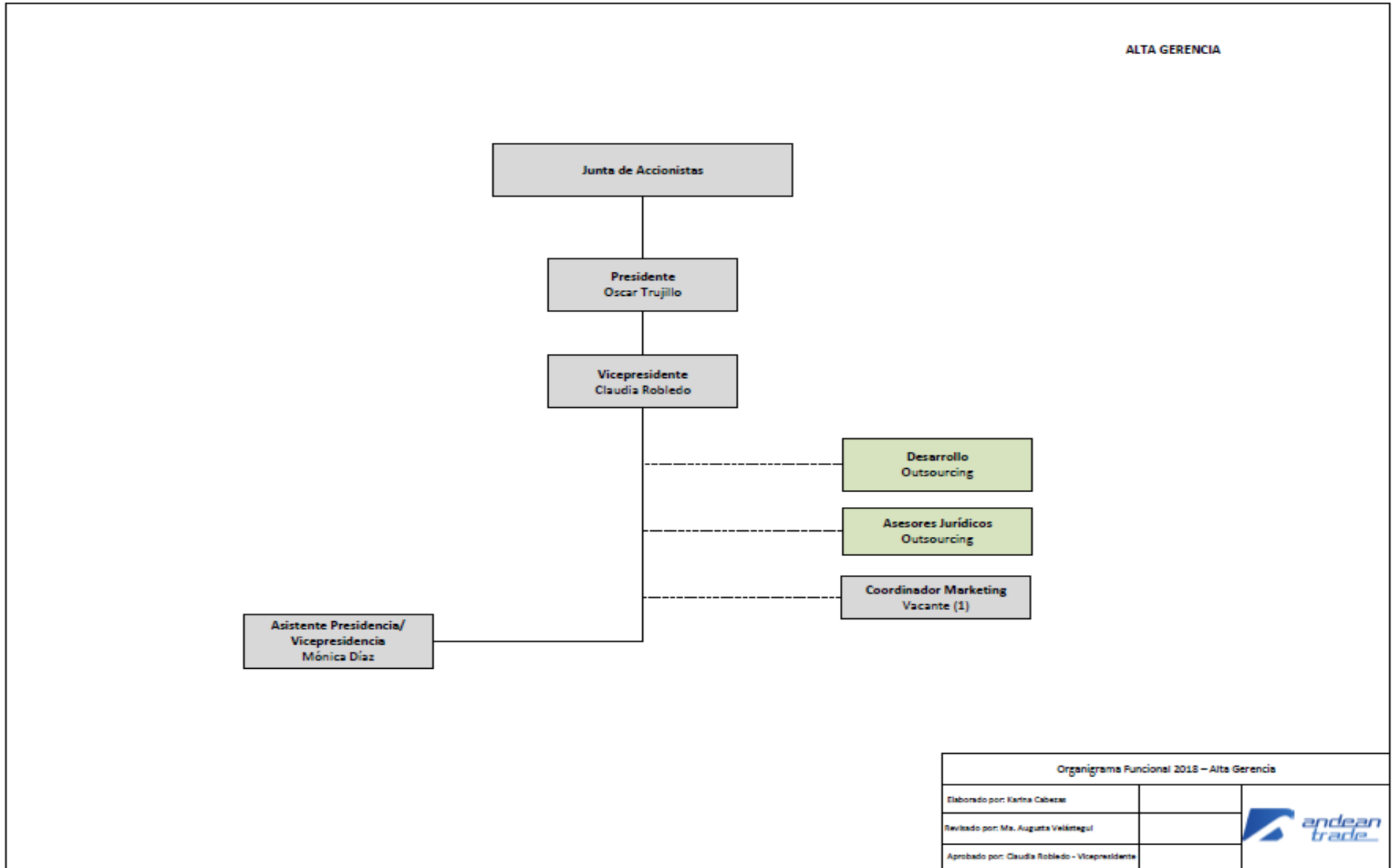


Ilustración 3. Diagrama jerárquico 2. OrganigramaV18_estructura_funcional AndeanTrade S.A

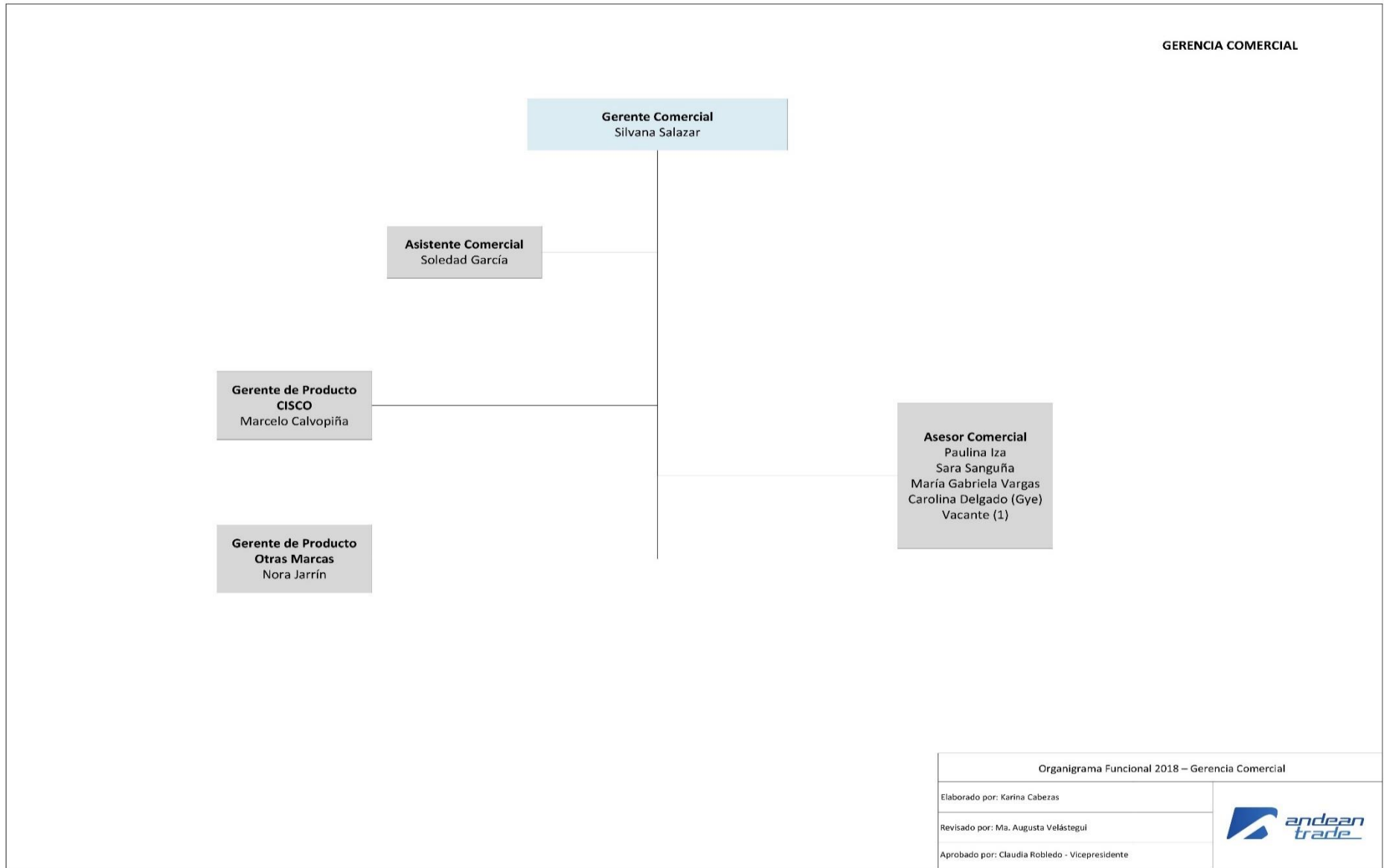


Ilustración 4. Diagrama jerárquico 3. OrganigramaV18_estructura_funcional AndeanTrade S.A

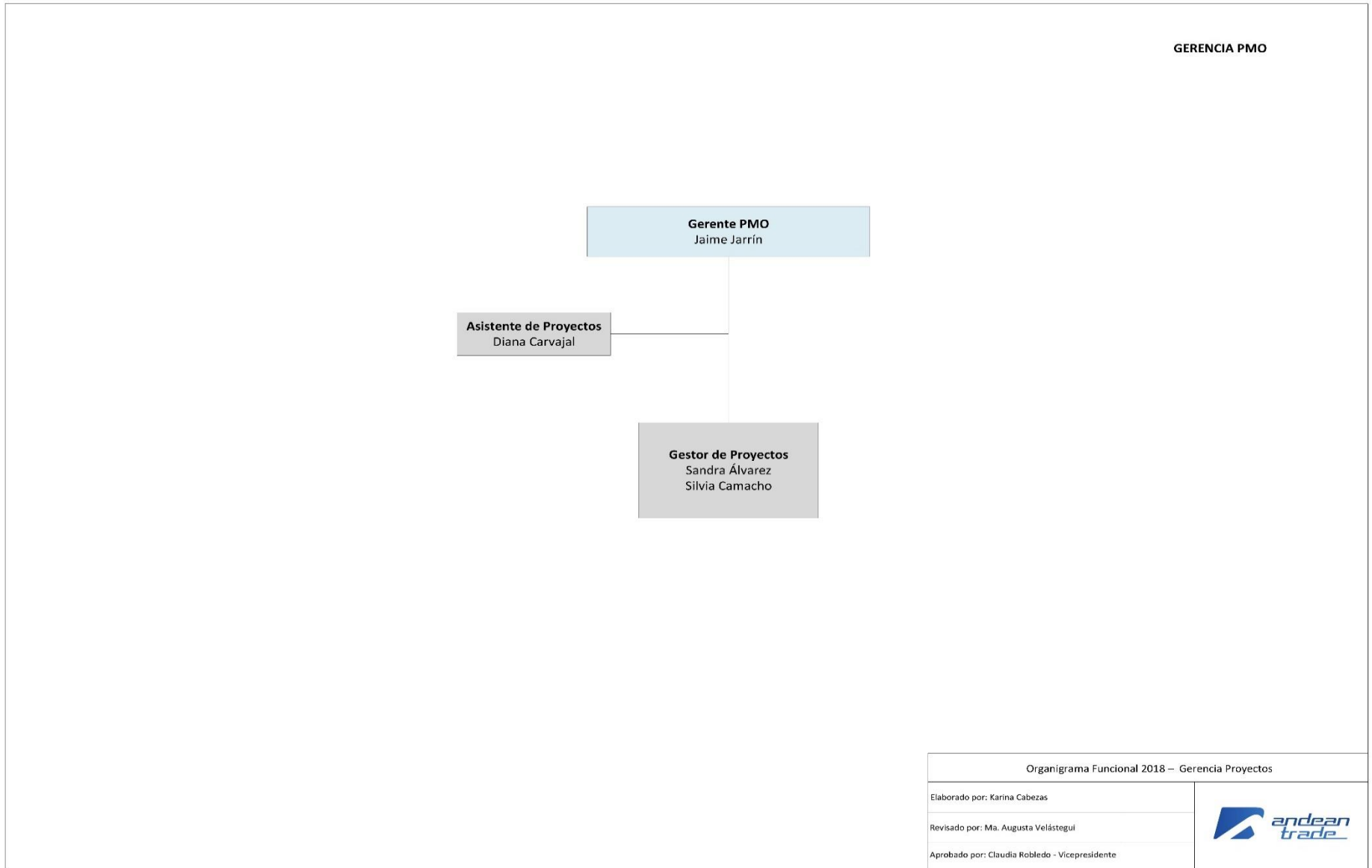


Ilustración 5. Diagrama jerárquico 4. OrganigramaV18_estructura_funcional AndeanTrade S.A

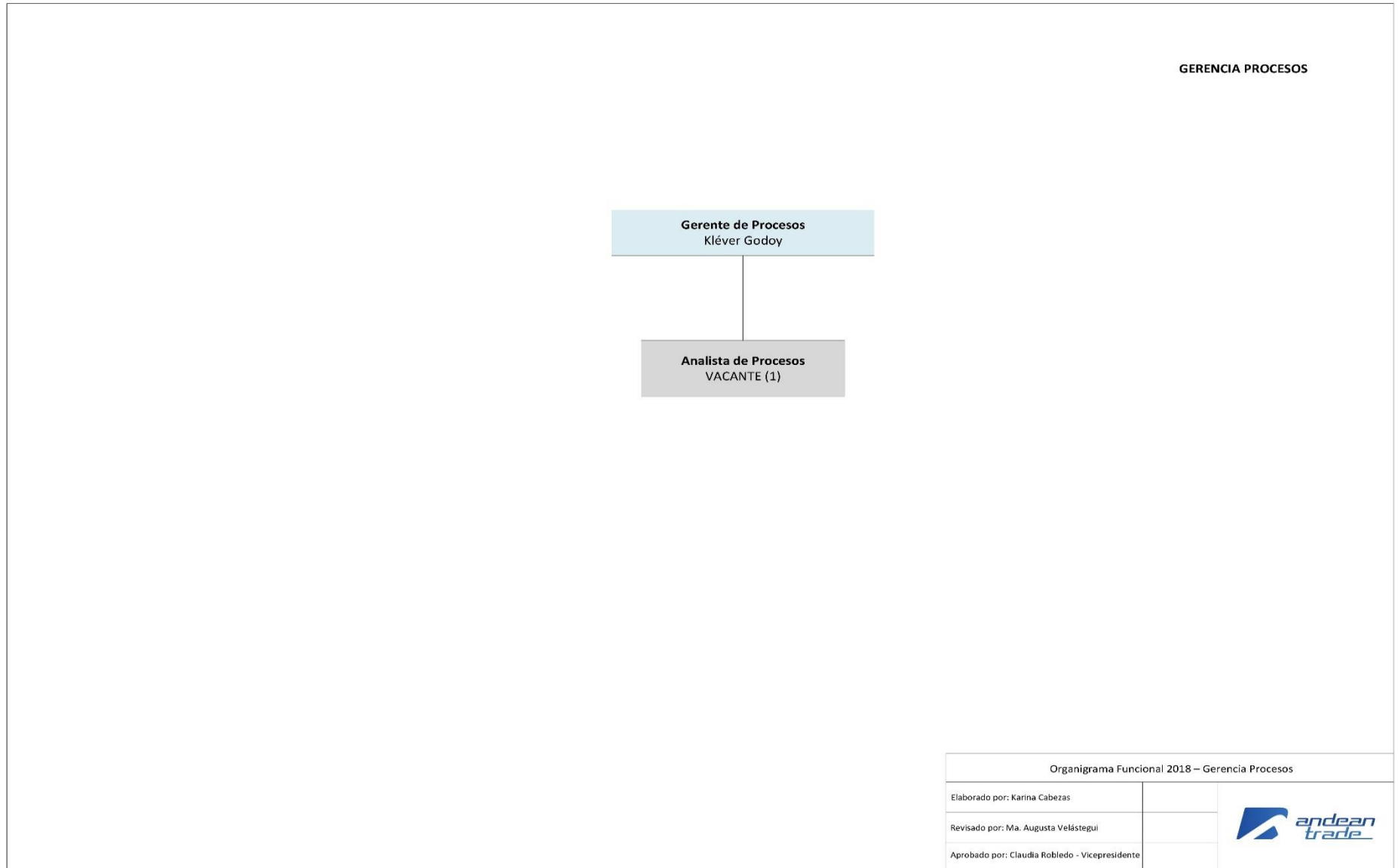


Ilustración 6. Diagrama jerárquico 5. OrganigramaV18_estructura_funcional AndeanTrade S.A

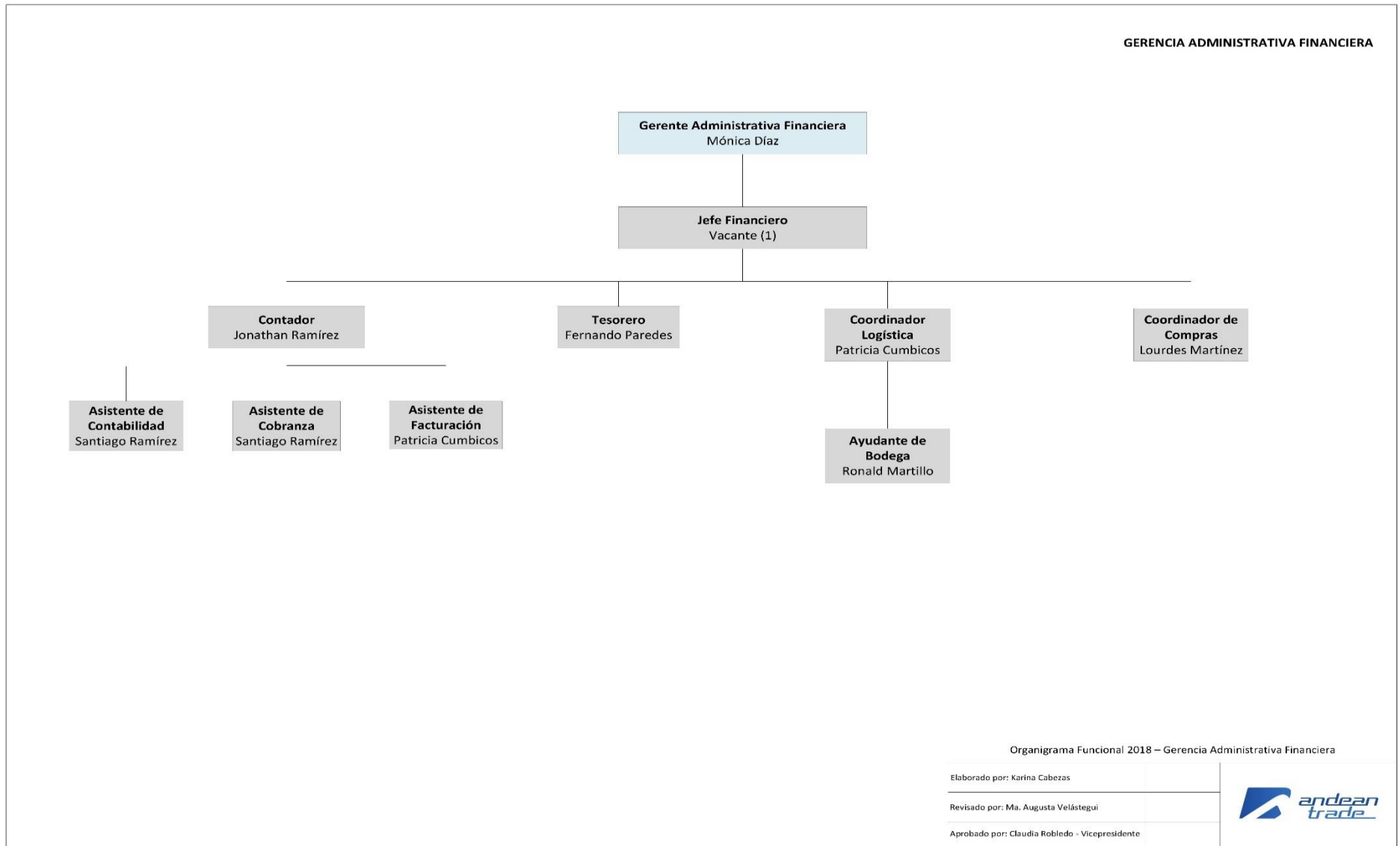


Ilustración 7. Diagrama jerárquico 6. OrganigramaV18_estructura_funcional AndeanTrade S.A

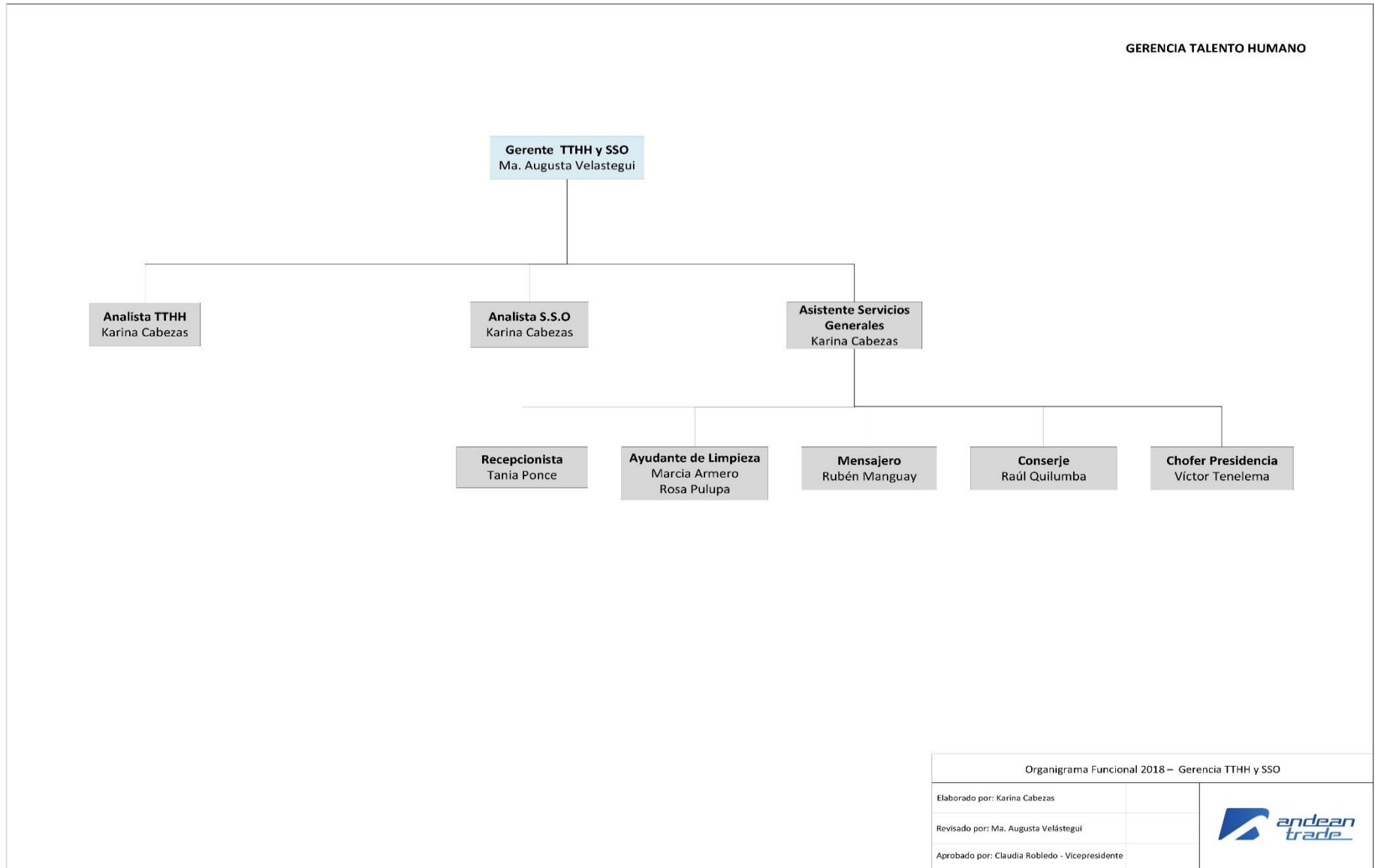


Ilustración 8. Diagrama jerárquico 7. OrganigramaV18_estructura_funcional AndeanTrade S.A

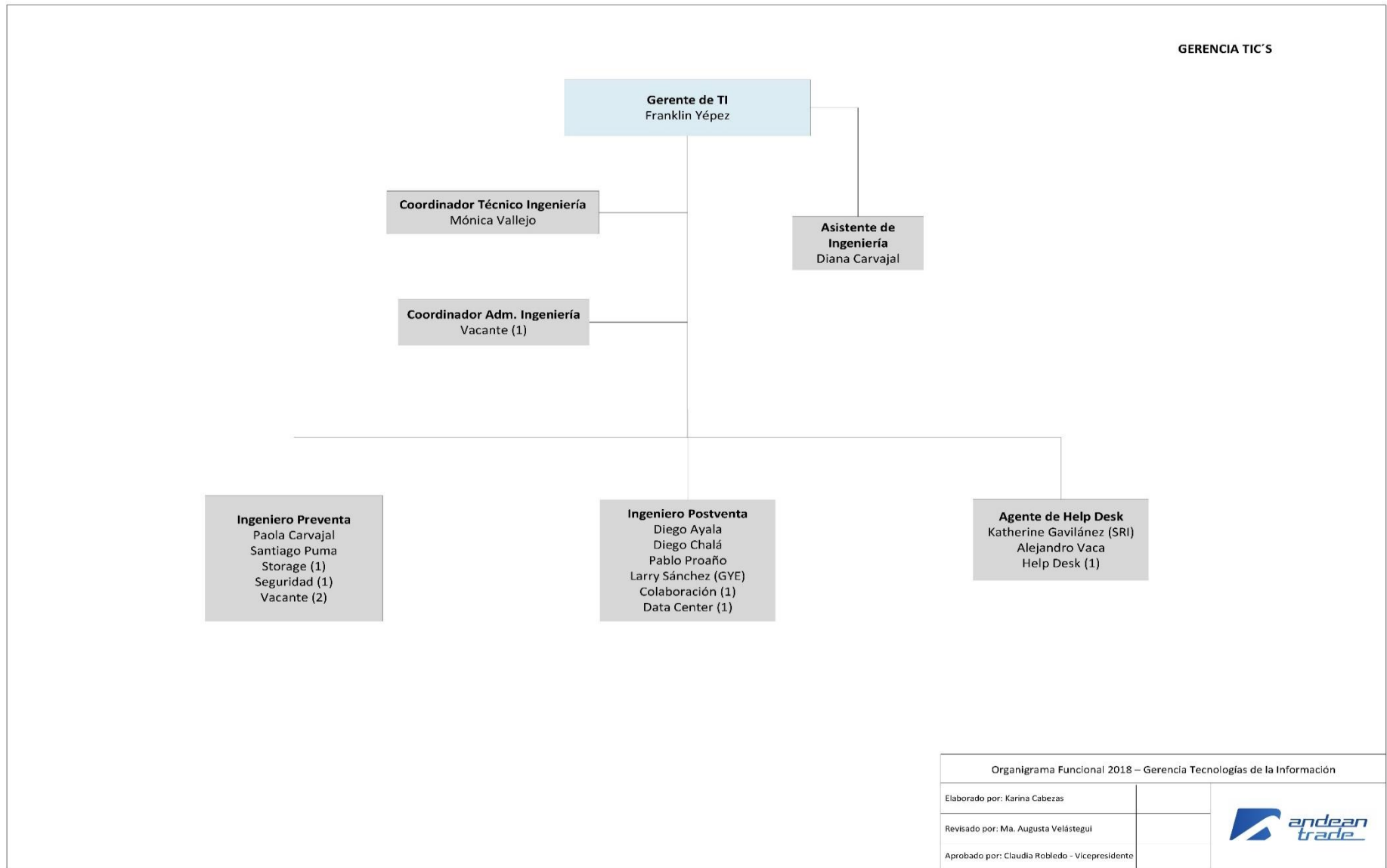


Ilustración 9. Diagrama jerárquico 8. OrganigramaV18_estructura_funcional AndeanTrade S.A

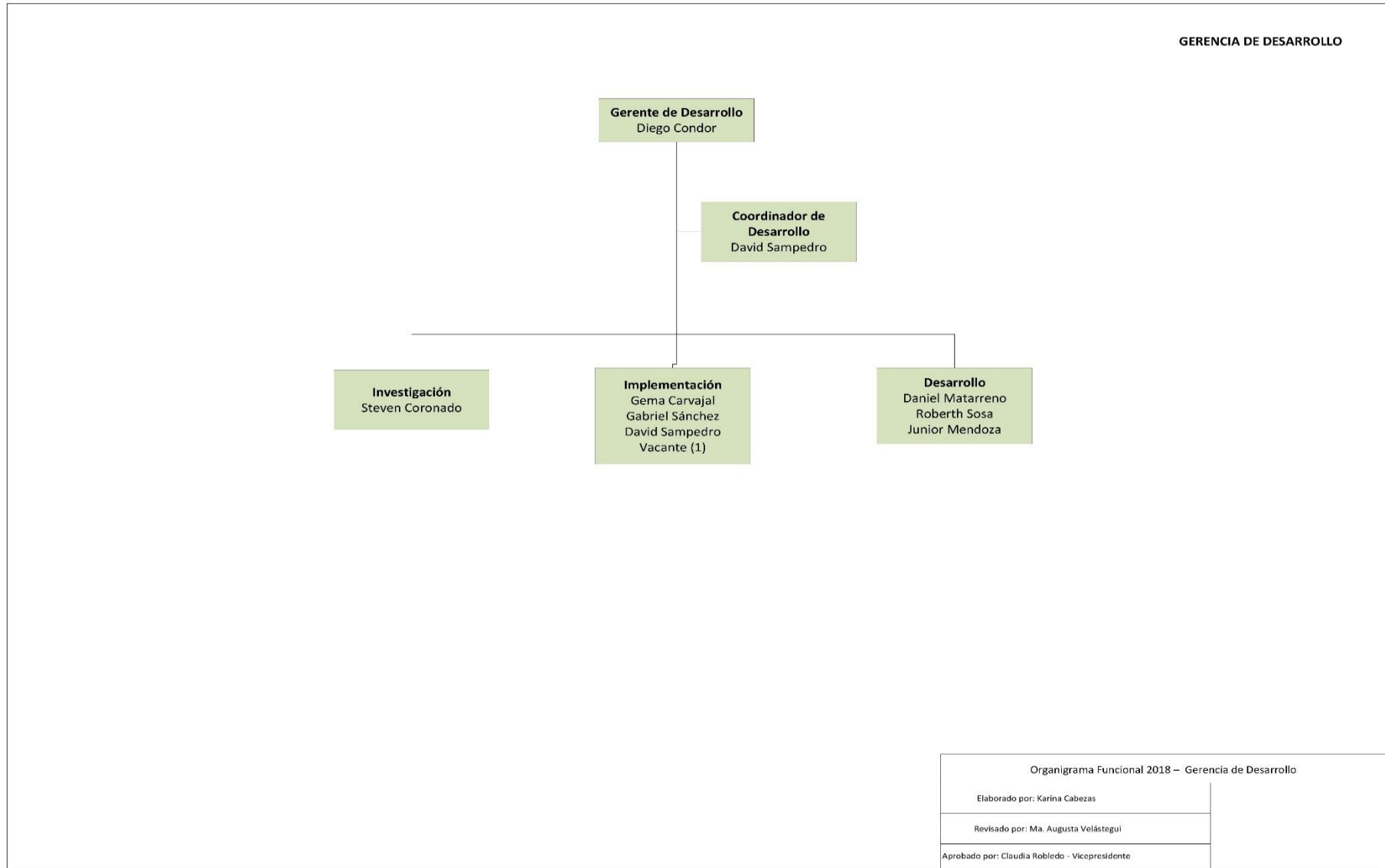


Ilustración 10. Diagrama jerárquico 9. OrganigramaV18_estructura_funcional AndeanTrade S.A

1.7 Análisis de la situación actual

Se realizó una investigación a la empresa AndeanTrade, empresa latinoamericana creada en el año 2000, que brinda a los usuarios soluciones tecnológicas en cuanto a desarrollo de software, redes corporativas, seguridad, esta empresa es líder en el mercado en cuanto a la integración de TI, servicios de infraestructura, consultoría, capacitaciones y marketing digital. Actualmente, la empresa maneja un agendamiento del área de ingenieros por medio de envíos de documentos al mail institucional de cada ingeniero en el que se detallan las actividades, soportes o procesos diarios asignados a cada uno. Sin embargo, lo que se busca es automatizar este proceso tanto para el administrador como para los ingenieros del área mediante una aplicación móvil que estará enviando y recibiendo datos de la base de datos interna de la empresa para el ingreso de empleados y para el almacenamiento diario de actividades.

Considerando estos antecedentes se ha decidido crear un aplicativo móvil enfocado al ámbito empresarial no solo porque su rendimiento y agilidad sea mayor al de un sitio web, sino que además cuenta con varias ventajas como otorgar una facilidad en cuanto a su acceso, distribución, personalización y utilización de diversas funcionalidades del dispositivo. “Para las empresas poseer una aplicación móvil propia se han convertido en un gran soporte y en una herramienta indispensable, tanto para la relación y contacto con empleados y clientes, como para su desarrollo comercial y de negocio. Cualquier empresa apuesta e invierte en el desarrollo de una aplicación móvil personalizada, puesto que es un valor añadido y podría marcar la diferencia con respecto a la competencia.” (WebMaster, 2017) La ventaja de crear aplicativos híbridos es que se basan en un solo código y se adaptan a múltiples plataformas, al igual que las aplicaciones nativas una aplicación híbrida puede acceder a los recursos de hardware, la creación de estas es más sencilla y económica. Se tiene como posibles soluciones dos aplicativos que se encuentran

actualmente en el mercado y estos son: “B2E Apps”, para empleados que buscan mejorar la comunicación de la empresa y una empresa logra una gran ventaja mediante ellas ya que reducen costes y optimizar procesos internos de la organización, disponible para la plataforma Android; y las “Nubelist” y “Nubelist Time” que son herramientas orientadas al control de fichada y mejorar la productividad de la empresa por ello es fundamental que una empresa de cualquier tamaño cuente con aplicaciones para optimizar procesos internos, disponible para la plataforma Android.

Es importante que la aplicación sea flexible ya que debido a la diferencia de necesidades de un cliente se debe adaptar hacia un beneficio común y de esta manera mantener la permanencia y fidelidad del usuario lo que generará un impulso y posicionamiento para la empresa. Es fundamental que el desarrollo de software se maneje mediante una metodología ya sea ágil o tradicional. “El uso de Metodologías Ágiles en la gestión empresarial de proyectos está irrumpiendo con enorme fuerza en el panorama corporativo de los últimos años. Según un estudio del Project Management Institute (PMI), el 71% de las organizaciones a nivel mundial ya usa Metodologías Ágiles. Además, más del 75 % de las organizaciones españolas coinciden en que las metodologías ágiles son cruciales para el éxito de la transformación digital en una empresa, según un estudio llevado a cabo por CA Technologies.” (Cuartero, 2018)

Tomando en cuenta las soluciones posibles mencionadas anteriormente, la solución que planteo es desarrollar una aplicación móvil que permita gestionar las actividades diarias del área de ingeniería en la empresa Andean Trade S.A, que implemente bases de datos para almacenar la información personal y corroborar la integridad del usuario mediante una seguridad adecuada utilizando el motor de SQL Server, infraestructura de la empresa; de esta manera se busca brindar a la empresa la automatización del proceso de agendamiento de ingeniería, monitoreando

el cumplimiento de los objetivos a nivel institucional y hacer un seguimiento que permitirá tomar acciones inmediatas.

El valor agregado de esta aplicación es que, a diferencia de las existentes en el mercado, es una aplicación multiplataforma, lo que permite tener un mayor rendimiento, ya que utiliza el mismo código para compilar en ambas plataformas; además reduce el tiempo para el coordinador de ingeniería al asignar las actividades diarias. Además, la aplicación es personalizada, lo que permite acceder a la base de datos con la información de los ingenieros, clientes y servicios de la empresa.

CAPÍTULO II: Fundamentos teóricos

2.1 Metodologías ágiles

Los métodos ágiles se han consolidado y se aplican en muchos campos, más allá de su nacimiento en el mundo del desarrollo de software. Algunas de ellas se centran en la forma más productiva de desarrollar aplicaciones informáticas, mientras que otras buscan manifestar específicamente procesos de la empresa. La influencia de los métodos ágiles es tal que está afectando a la forma de contratar proyectos, a través de los contratos ágiles, que reflejan la flexibilidad y capacidad de conciliación aplicada a las relaciones formales entre empresas. De entre todos los métodos ágiles enfatiza Scrum por su propagación y aprobación. (Álvarez, de las Heras, & Lasa, 2012, p.34)

Todos los métodos ágiles tienen características que los distinguen, así como también características en común, todos ellos consideran la **colaboración** un elemento clave. Tanto las personas que están construyendo el producto como el cliente deben trabajar en constante comunicación y sentirse miembros de un gran equipo. Un método es ágil si permite erigir un

producto de forma **incremental**, es decir crear algo muy sencillo inicialmente y que vaya siendo enriquecido y completado de forma progresiva. (Álvarez, de las Heras, & Lasa, 2012, p.34)

Según Letelier & Penadés(2006) la filosofía ágil se resume en el documento del manifiesto ágil, creado el 12 de febrero de 2001, el cual expone los principales valores que inspiran los doce principios del manifiesto, los mismos son:

- **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.** La gente es el primordial factor de éxito de un proyecto software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado; sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. No se necesitan desarrolladores brillantes, sino desarrolladores que se adapten bien al trabajo en equipo. Así mismo, las herramientas (compiladores, depuradores, control de versiones, etc.) son importantes para mejorar el rendimiento del equipo, pero el disponer más recursos que los estrictamente necesarios también puede afectar negativamente. En resumen, es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades
- **Desarrollar software que funciona más que conseguir una buena documentación.** Aunque se parte de la base de que el software sin documentación es un desastre, la regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se

incorpora al equipo de desarrollo, se considera que los dos elementos que más le van a servir para ponerse al día son: el propio código y la interacción con el equipo

- **La colaboración con el cliente más que la negociación de un contrato.** Las características particulares del desarrollo de software hacen que muchos proyectos hayan fracasado por intentar cumplir unos plazos y unos costes preestablecidos al inicio de este, según los requisitos que el cliente manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito
- **Responder a los cambios más que seguir estrictamente un plan.** La habilidad de responder a los cambios que puedan surgir al largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso de este. Por lo tanto, la planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses.

El desarrollo ágil se basa en los principios del manifiesto ágil y sus valores éticos, que tratan de dar más valor a algunos conceptos, pero sin dejar de lado los demás. Según Vique(2013, p.41) los principios son los siguientes:

1. Dar más valor a los individuos y a sus interacciones que a los procesos y herramientas
2. Dar más valor al software que funciona que a la documentación exhaustiva
3. Dar más valor a la colaboración con el cliente que a la negociación contractual
4. Dar más valor a la respuesta al cambio que al seguimiento de un plan

Existen métodos ágiles de proceso o gestión como son **Scrum** o **Kanban**. En el caso de que nuestro proyecto sea un desarrollo de software, una buena gestión no es suficiente y necesitamos seguir unas buenas prácticas de programación, de forma que podamos gestionarnos de forma optimizada, a la vez que se crea un software de calidad. Es aquí donde entran en juego los métodos ágiles de programación como por ejemplo la **Programación extrema (XP)**. En definitiva, para crear un buen producto software, necesitamos una combinación de mejores prácticas de gestión con mejores prácticas de programación, ambas compartiendo los valores y principios ágiles. (Álvarez, de las Heras, & Lasa, 2012, p.34)

2.1.1 Scrum

Scrum se ha utilizado para gestionar el trabajo en productos complejos desde principios de los años noventa. Scrum se encuentra dentro de "Ágil", que es el término general para varios tipos de enfoques para lograr cualquier alcance complejo e innovador del trabajo realizado. El concepto es dividir los proyectos grandes en etapas más pequeñas, revisando y adaptando a lo largo del camino. (Scrum Alliance, 2019)

El término "Scrum" proviene de un artículo de Harvard Business Review publicado en el año de 1986, en el que los autores Takeuchi y Nonaka hicieron una analogía al comparar equipos multifuncionales de valioso rendimiento con la alineación de scrum utilizada por los equipos de rugby.

“Scrum propone un marco de trabajo que puede dar soporte a la innovación, basándose en equipos autogestionados. Con Scrum se pueden obtener resultados con calidad, en iteraciones cortas llamadas Sprint.” (Álvarez, de las Heras, & Lasa, 2012, p.39)

Con la metodología Scrum el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Asimismo, le permite en cualquier momento realinear el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema (Sof Teng, 2018).

Esta metódica de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades (Sof Teng, 2018).

Según Álvarez, de las Heras, & Lasa (2012, p.39) Scrum se basa en los siguientes principios:

- **Inspección y adaptación.** En Scrum se trabaja en iteraciones llamadas *Sprint*, que tienen una duración de entre 1 y 4 semanas. Cada iteración termina con un producto entregable
- Al finalizar cada iteración, este producto se muestra al cliente para que opine sobre él. A continuación, el equipo se reunirá para analizar la manera en que está trabajando. Uniendo los dos puntos de vista, “el qué” se ha hecho y “el cómo” se está construyendo, se aprenderá con la experiencia y podremos mejorar iteración tras iteración
- **Autoorganización y colaboración.** El equipo se gestiona y organiza a sí mismo. Este nivel de libertad implica asumir una responsabilidad y un gran nivel de compromiso por parte de todos. Esta Autoorganización funcionará siempre que exista una alta colaboración y espíritu de equipo. Los líderes y clientes colaborarán igualmente con el equipo de desarrollo en todo momento facilitando su trabajo, resolviendo dudas y eliminando posibles impedimentos

- **Priorización.** Como en el resto de los métodos ágiles, es crucial no perder tiempo y dinero en algo que no interesa inmediatamente para el producto. Para ello es necesario tener unos requisitos perfectamente priorizados reflejando el valor del negocio
- **Mantener un latido.** Es tremendamente valioso mantener un ritmo que dirija el desarrollo. Este latido marcará la pauta del trabajo y ayudará a los equipos a optimizar su trabajo. El tener un ritmo fijo de trabajo, tanto a nivel del día a día como a nivel de *Sprint*, permite que el equipo sea predecible ya que este aprenderá a estimar la cantidad de trabajo a la que puede comprometerse.

Esta metodología aporta diversos beneficios:

- **Cumplimiento de expectativas.** El cliente establece sus expectativas indicando el valor que le aporta cada requisito / historia del proyecto, el equipo los estima y con esta información el Product Owner establece su prioridad. De manera regular, en las demos de Sprint el Product Owner comprueba que efectivamente los requisitos se han cumplido y transmite se feedback al equipo
- **Flexibilidad a cambios.** Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado. La metodología está diseñada para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos
- **Reducción del Time to Market.** El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- **Mayor calidad del software.** La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior

- **Mayor productividad.** Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse
- **Maximiza el retorno de la inversión (ROI).** Producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión
- **Predicciones de tiempos.** Mediante esta metodología se conoce la velocidad media del equipo por sprint (los llamados puntos historia), con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está en el Backlog
- **Reducción de riesgos.** El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada (Sof Teng, 2018).

Una de las principales características de Scrum es que en cada iteración todas las etapas de la creación de un producto se solapan, es decir, en cada *Sprint* se realiza la planificación, análisis, creación y comprobación de lo que se va a entregar al final del mismo (Álvarez, de las Heras, & Lasa, 2012, p. 40).

El marco de trabajo general de Scrum según Álvarez, de las Heras, & Lasa (2012, pp. 40-41) está compuesto por una serie de roles, reuniones y de paneles de información o artefactos que se indican a continuación:

Los roles en el equipo Scrum:

- El *Product Owner* o dueño del producto. Es el responsable desde el punto de vista del negocio
- El *Scrum Master* es el responsable de que el equipo sea productivo ayudándole en todo momento a conseguir el objetivo acordado y de asegurar que los principios se estén respetando.
- El equipo. Es el responsable de la construcción del producto
- Los artefactos de Scrum. Los *Backlog* o repositorios son artefactos en los que el *Product Owner*, equipo y *Scrum Master* escriben los requisitos y tareas.
- El *Product Backlog*. Es el lugar que contiene los requisitos del cliente priorizados y estimados
- El *Sprint Backlog*. Es la selección de requisitos del *Product Backlog* negociados para el *Sprint* y que se ha descompuesto en tareas por el equipo para expresar los requisitos del cliente en un lenguaje técnico
- El *Burndown Chart*. Es la gráfica en la que se representa el avance del proyecto.

Según Álvarez, de las Heras, & Lasa (2012, p. 42) de forma muy simplificada se podría resumir el flujo del trabajo con Scrum de la siguiente manera:

1. El *Product Owner* escribe en el *Product Backlog* todas las funcionalidades y requisitos que quiera que su producto contemple. Eso sí, debe priorizarla indicando el orden en que quiere que se vaya construyendo su producto. Los ítems más prioritarios deben estar más detallados que los que no son tan urgentes
2. El equipo estimará cada uno de estos requisitos en función de su complejidad. Teniendo en cuenta la prioridad marcada por el *Product Owner* y la estimación realizada por el equipo, se acordará la cantidad de trabajo que se vaya a abordar en el siguiente *Sprint*

3. Empieza el *Sprint* y el equipo se sincronizará diariamente con la *Daily Meeting*
4. Al finalizar el *Sprint*, el equipo muestra al *Product Owner* el trabajo realizado que debe ser un producto potencialmente entregable. Con opinión y sugerencias del *Product Owner* y la información obtenida en retrospectiva posterior que realizará el equipo, se preparará la siguiente iteración.

2.1.2 XP

Extreme Programming (XP) es una de las metodologías más conocidas y utilizadas actualmente para el desarrollo de productos de software, “es muy útil a la hora de abordar proyectos con requisitos vagos o cambiantes. XP es especialmente útil si se aplica a equipos de desarrollo pequeños o medianos” (Álvarez, de las Heras, & Lasa, 2012, p. 49).

Es un método adaptativo, es decir, se ajusta muy bien a los cambios. Propone desarrollar el código de forma que su diseño, arquitectura y codificación permitan incorporar modificaciones y añadir una funcionalidad nueva sin demasiado impacto en la calidad de este (Álvarez, de las Heras, & Lasa, 2012, p. 40).

Desarrollando como propone XP, se obtienen rápidamente resultados. Al trabajar con pequeñas iteraciones, se puede obtener con frecuencia comentarios del cliente, lo que tiene como resultado que el producto final cubra ampliamente sus expectativas y necesidades. [...] Para XP las pruebas son la base de la construcción y propone que sean los desarrolladores los que escriban las pruebas a medida que van construyendo el código y se realice una integración continua, de forma que el software creado tenga una gran estabilidad (Álvarez, de las Heras, & Lasa, 2012, p. 40).

Según Álvarez, de las Heras, & Lasa (2012, p. 269), el ciclo de vida de XP se basa en:

- **Planificación.** En cualquier equipo que vaya a desarrollar un proyecto debe haber personas responsables de tomar decisiones de negocio y que tengan clara cuál es la visión del producto, el plan de entregas, establezcan las necesidades que debe cubrir el sistema y gestionen los riesgos. El resto del equipo hará sugerencias y estimaciones para matizar este plan inicial de entregas. Las historias de usuario deberán estar priorizadas para reflejar el orden en que se debe construir el producto, desde el punto de vista más técnico. Naturalmente, en las etapas iniciales del proyecto, es necesario dedicar un mayor esfuerzo a la planificación

Las historias más prioritarias se implementarán en primer lugar. A su vez el equipo, al comienzo de cada semana, planificará de forma detallada la manera de abordar la siguiente iteración. Además, cada día, el equipo organizará el trabajo para esa jornada. En definitiva, se planifica al inicio del proyecto, al inicio de cada iteración y aún más en detalle, todos los días

- **Análisis.** Para que el análisis se mantenga actualizado durante el proyecto, los clientes deben comunicarse constantemente y cercanos a las personas que se encuentran construyendo el producto. Cuando un desarrollador tenga dudas sobre cómo implementar un requisito, debe poder preguntar al cliente. De igual manera, se debe trabajar en estrecha colaboración con los responsables tanto de pruebas como de diseño gráfico de manera que no quepa ambigüedad en la definición de requisitos
- **Diseño y codificación.** XP propone trabajar de manera que, tanto el diseño como la arquitectura se creen de forma incremental. De este modo, se mejora el diseño y la arquitectura poco a poco y de forma constante

- Pruebas. Uno de los pilares sobre los que se fundamenta XP son las pruebas. Las pruebas deben llevarse a cabo a todos los niveles y todos los implicados en un proyecto deben contribuir a realizarlas. Los desarrolladores deben realizar las pruebas a la par con la construcción del código
- Despliegue. La forma de construir el producto con XP hace posible que, al finalizar cada semana, el software obtenido pueda ser puesto en producción ya que la funcionalidad comprometida está asegurada. Esto no significa que se realicen entregas al cliente final con esta frecuencia. Las entregas se realizarán siguiendo el plan de entregas establecido previamente con el cliente.

2.1.3 Mobile-D

El método Mobile-D se desarrolló junto con un proyecto finlandés en el 2004. Fue realizado, principalmente, por investigadores de la VTT (Instituto de Investigación Finlandés) y, a pesar de que es un método antiguo, sigue en vigor (se está utilizando en proyectos de éxito y está basado en técnicas que funcionan) (Vique, 2013, p. 42).

El objetivo es conseguir ciclos de desarrollos muy rápidos en equipos muy pequeños (de no más de diez desarrolladores) trabajando en un mismo espacio físico. Según este método, trabajando de esa manera se deben conseguir productos totalmente funcionales en menos de diez semanas (Vique, 2013, p. 42).

Se trata de método basado en soluciones conocidas y consolidadas: Extreme Programming (XP), Crystal Methodologies y Rational Unified Process (RUP), XP para las prácticas de



desarrollo, Crystal para escalar los métodos y RUP como base en el diseño del ciclo de vida (Vique, 2013, p. 42).

Ilustración 11. Gráfica de ciclo de vida de Mobile-D (Letelier & Penadés, 2006).

Cada fase (excepto la inicial) tiene siempre un día de planificación y otro de entrega.

Exploración. Se dedica a la planificación y a los conceptos básicos del proyecto. Es diferente del resto de fases.

(Agile, 2008) Se centra la atención a la planificación y a los conceptos básicos del proyecto. Se realizan los alcances del proyecto y su establecimiento con las funcionalidades donde se va a llegar. Tipo de patrón: Patrón de fase El propósito de esta fase es la planificación y establecimiento de una buena planificación “*A well planned is half done*”, esta fase es muy importante para establecer las bases para una implementación bien controlada de software, la arquitectura del producto, el proceso de desarrollo y la selección del medio ambiente.

Según Wiliams (s.f.), los objetivos de la fase de exploración son:

- Establecer los grupos de actores necesarios en la planificación y el seguimiento del proyecto de desarrollo de software.
- Definir los alcances y límites del proyecto de desarrollo de software de desarrollo.
- Planificar el proyecto respecto al entorno, el personal y los problemas del proceso.

Inicialización. Se preparan e identifican todos los recursos necesarios. Se establece el entorno técnico (Vique, 2013, p. 42).

“El propósito de esta fase es permitir el éxito de las siguientes fases del proyecto mediante la preparación y verificación de todas las cuestiones fundamentales del desarrollo a fin de que todos están en plena disposición de la aplicación de los requisitos seleccionados por el cliente” (wiliams, s.f.).

Según Wiliams(s.f.), los objetivos de la fase de exploración son:

- Establecer los grupos de actores necesarios en la planificación y el seguimiento del proyecto de desarrollo de software.
- Definir los alcances y límites del proyecto de desarrollo de software de desarrollo.
- Planificar el proyecto respecto al entorno, el personal y los problemas del proceso.

Productización o fase de producto. Se repiten iterativamente las subfases, con un día de planificación, uno de trabajo y uno de entrega. Aquí se intentan utilizar técnicas como la del *test driven development* para obtener la mayor calidad.

El propósito en la fase de producción es implementar la funcionalidad requerida en el producto mediante la aplicación del ciclo de desarrollo iterativo e incremental.

(wiliams, s.f.)

Según Wiliams(s.f.), los objetivos de esta fase son:

- Implementar la funcionalidad del producto priorizando los requerimientos del cliente.
- Centrarse en la funcionalidad básica fundamental para permitir múltiples ciclos de mejora.

Estabilización. Se llevan a cabo las acciones de integración para asegurar que el sistema completo funciona correctamente. El propósito de la fase de estabilización es asegurar la calidad de la implementación del proyecto. (wiliams, s.f.)

Según Wiliams(s.f.), los objetivos de la fase de estabilización son:

- Finalizar la implementación del producto.
- Mejorar y garantizar la calidad del producto.
- Finalizar la documentación del proyecto.

Fase de pruebas y reparación. Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema según los requisitos del cliente.

(Agile, 2008) El propósito de la fase de pruebas es ver si el sistema productora implementa la funcionalidad definida del cliente correctamente, proporcionar la retroalimentación al equipo de desarrollo de los defectos y errores encontrados en la funcionalidad del software para ser corregidos estos defectos encontrados (wiliams, s.f.).

Según Wiliams(s.f.), los objetivos de la fase de pruebas son:

- Probar el sistema basado en la documentación producida en el proyecto.
- Proporcionar información de defectos encontrados.
- Planificar la solución a los defectos encontrados.
- Fijar los errores hallados.
- Producir un sistema libre de errores como sea posible.

2.2 Aplicaciones móviles

2.2.1 Aplicaciones nativas

“Las aplicaciones nativas son las aplicaciones propias de cada plataforma. Deben ser desarrolladas pensando en la plataforma concreta. No existe ningún tipo de estandarización, ni en las capacidades ni en los entornos de desarrollo, por lo que los desarrollos que pretenden soportar plataformas diferentes suelen necesitar un esfuerzo extra” (Ramírez Vique, s.f.).

Estas aplicaciones son las que mayor potencial tienen, pues aprovechan al máximo los dispositivos y consiguen, de esa manera, una mejor experiencia de usuario. Existen muchas plataformas, una gran parte de ellas ligadas al tipo de dispositivo, aunque también hay plataformas, como Android, que existen para diferentes tipos de dispositivos. Algunas de las más conocidas son iOS, Android, Blackberry, bada, Java Me, Windows Phone (antes Windows Mobile o Windows Ce), Symbian, Web OS, Brew, etc. Todas ellas tienen diferentes tipos de dispositivos con una base común entre ellos (Ramírez Vique, s.f.).

Según Ramírez Vique (s.f.), las ventajas de las aplicaciones nativas son las siguientes:

- Acceso total al contexto, con todas las posibilidades que eso conlleva.
Consigue las mejores experiencias de usuario.

- Posibilidad de gestión de interrupciones en la aplicación o en las capacidades del dispositivo. Desde saber si tenemos conexión de datos o conexión de localización hasta tener información sobre la batería.
- Son relativamente fáciles de desarrollar si solo se contempla una plataforma.
- Se pueden distribuir por los canales conocidos de aplicaciones que permita la plataforma, con lo que se pueden vender más fácilmente.
- Todas las novedades llegan primero a este tipo de aplicaciones, pues es en este tipo de aplicaciones donde se prueban.

En cambio, según Ramírez Vique (s.f.), las aplicaciones nativas cuentan con los siguientes inconvenientes:

- Portar aplicaciones es costoso. En el caso de querer realizar una aplicación para más de una plataforma, se complica el desarrollo, debido a los problemas de la fragmentación.
- Dependiendo de la plataforma elegida, puede haber fragmentación dentro de cada plataforma, debido a los diferentes tipos de dispositivos o versiones de la plataforma.
- No existe un estándar, por lo que cada plataforma ofrecerá sus peculiaridades.
- Normalmente, para desarrollar, distribuir o probar estas aplicaciones en dispositivos reales, es necesario tener una licencia de pago, dependiendo de la plataforma.
- Las ganancias por estas aplicaciones suelen repartirse entre el creador de la aplicación y la plataforma de distribución.

2.2.2 Aplicaciones híbridas

Se conoce a una aplicación híbrida como la composición de aplicación web y aplicación nativa. “La forma de desarrollarlas es parecida a la de una aplicación web -usando HTML, CSS y JavaScript-, y una vez que la aplicación está terminada, se compila o empaqueta de forma tal, que el resultado final es como si se tratara de una aplicación nativa” (Cuello & Vittone, 2013, p.23).

“Esto permite casi con un mismo código obtener diferentes aplicaciones, por ejemplo, para Android y iOS, y distribuir las en cada una de sus tiendas” (Cuello & Vittone, 2013, p.23).

“A diferencia de las aplicaciones web, estas permiten acceder, usando librerías, a las capacidades del teléfono, tal como lo haría una app nativa” (Cuello & Vittone, 2013, p.23).

Las aplicaciones híbridas, también tienen un diseño visual que no se identifica en gran medida con el sistema operativo. Sin embargo, hay formas de usar controles y botones nativos de cada plataforma para apegarse más a la estética propia de cada una (Cuello & Vittone, 2013, p.24).

“Existen algunas herramientas para desarrollar este tipo de aplicaciones. Entre ellas, Apache Cordova, Ionic, Icenium, React Native” (Cuello & Vittone, 2013, p.24).

2.2.3 Aplicaciones web

“Existe un tipo de aplicaciones, llamadas aplicaciones web móviles nativas, que no son aplicaciones web propiamente ni tampoco nativas. Se ejecutan con un navegador o, mejor dicho, con un componente nativo que delega en un navegador, y tienen algunas de las ventajas de las aplicaciones nativas” (Ramírez Vique, s.f.).

“Este tipo de aplicaciones pueden ser instaladas en el dispositivo, con lo que pueden utilizar los canales estándares de distribución de aplicaciones nativas, o bien incorporarse como accesos

directos (como el resto de las aplicaciones). Sin embargo, estas aplicaciones no tienen la potencia de las aplicaciones nativas, sino que simplemente ejecutan código en un navegador embebido (generalmente con HTML 5)” (Ramírez Vique, s.f.).

Según Ramírez Vique (s.f.), las ventajas de las aplicaciones web móviles nativas son las siguientes:

- Todos los puntos a favor de las aplicaciones web móviles.
- Se pueden considerar, en lo que respecta a la instalación y la distribución, como aplicaciones nativas.

Según Ramírez Vique (s.f.), los inconvenientes de aplicaciones web son:

- La mayoría de los inconvenientes de las aplicaciones web móviles, a excepción de la instalación en el cliente.
- La experiencia del usuario es, en ocasiones, contradictoria, pues a pesar de tratarse de una aplicación nativa, requiere de conexión a Internet para poder trabajar y funciona según los tiempos de respuesta del navegador.

2.3 Herramientas de desarrollo

2.3.1 Ionic

Es un marco de desarrollo híbrido típico desarrollado en 2015 como personalización de Apache Cordova por el proveedor de software Drifty Co, utiliza tecnologías para escribir y representar la aplicación y requiere de PhoneGap y Cordova para acceder a características nativas, está en línea con el patrón MVC, donde la vista y lógica se encuentran separadas.

Ionic es capaz de ejecutar el código independientemente de la plataforma por lo que adapta ciertos comportamientos de acuerdo con las recomendaciones de las distintas plataformas. Ionic

es un framework basado en Angular, de JavaScript; utiliza plantillas HTML5, CSS, Sass. (Masnou, 2016) El ecosistema de Ionic es caracterizado por una búsqueda constante de la optimización e integración con dispositivos móviles para conseguir el mayor rendimiento y la mejor experiencia de usuario posible (Perez, 2015).

“Ionic es de código abierto publicado bajo una licencia MIT. Esto significa que se puede utilizar en nuestros proyectos personales o comerciales, de forma gratuita. MIT es la misma licencia que utiliza otros proyectos populares como jQuery y Ruby on Rails” (Perez, 2015).

Según Perez (2015), sus principales características son:

Alto rendimiento

La velocidad es importante. Tan importante que sólo se nota cuando no está en tu app. Ionic está construido para ser rápido gracias a la mínima manipulación del DOM, con cero jQuery y con aceleraciones de transiciones por hardware.

AngularJS & Ionic

Ionic utiliza AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Ionic no sólo se ve bien, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones. Trabaja perfectamente con AngularJS.

Centro nativo

Ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para iOS o Android. Lo interesante, como sabéis, es que desarrollas una vez, y compilas para varios.

Bonito diseño

Limpio, sencillo y funcional. Ionic ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos, etc.

Un potente CLI

Con un sólo comando podrás crear, construir, probar y compilar tus aplicaciones en cualquier plataforma.

2.3.2 React Native

React

“Es una biblioteca escrita en JavaScript, desarrollada por Facebook otorga la facilidad para crear interfaces de usuario interactivas, posee vistas declarativas que permiten que su código sea más predecible y fácil de depurar. Debido a que los componentes están escritos en JavaScript mas no en plantillas, se puede pasar datos enriquecidos a través de la aplicación y mantener su estado fuera del DOM. React se puede mostrar en el servidor mediante el uso de Node y aplicaciones móviles de potencia mediante React Native” (Grados, 2016).

“React es flexible y presenta la característica de interactuar con otros marco y bibliotecas, asimismo uno de sus puntos más destacados es que no solo se utiliza en el lado del cliente, sino que también se puede representar en el servidor y trabajar conjuntamente; utiliza DOM Virtual, concepto que permite mantener componentes actualizados, crea sub-arboles con los nodos sobre los cuales se presentan los cambios de estado” (Grados, 2016).

React Native

“React Native te permite crear aplicaciones móviles usando solo JavaScript. Utiliza el mismo diseño que React, lo que le permite componer una interfaz de usuario móvil rica utilizando componentes declarativos” (Facebook Inc., s.f.).

“Las aplicaciones que está creando con React Native no son aplicaciones web para dispositivos móviles porque React Native utiliza los mismos bloques de construcción fundamentales de UI que las aplicaciones normales de iOS y Android. En lugar de usar Swift, Kotlin o Java, estás poniendo esos bloques de construcción juntos usando JavaScript y React” (Facebook Inc., s.f.).

React Native te permite construir tu aplicación más rápido. En lugar de recompilar, puedes recargar tu aplicación al instante. Con Hot Reloading, incluso puede ejecutar un nuevo código manteniendo el estado de su aplicación. Pruébalo, es una experiencia mágica. Se combina sin problemas con componentes escritos en Swift, Java o Objective-C. Es fácil desplegar el código nativo si necesita optimizar algunos aspectos de su aplicación. También es fácil crear una parte de su aplicación en React Native, y otra parte de su aplicación utilizando directamente el código nativo, así es como funciona la aplicación de Facebook (Facebook Inc., s.f.).

2.3.3 PhoneGap

Surgido de un iPhone Dev Camp organizado por Apple en 2008 de la mano de la empresa Nitobi Software. En 2011 Nitobi fue adquirida por el gigante Adobe. Después de esta adquisición el código de PhoneGap fue donado a la Fundación Apache pasando a llamarse Apache Cordova, pudiendo ser utilizado para construir otras herramientas. La última gran

evolución de PhoneGap se produjo en 2012 con el nacimiento de PhoneGap Build, que permitía acabar con la engorrosa duplicidad de entornos de trabajo (Siles, 2013).

PhoneGap es un framework gratuito y open source que permite crear aplicaciones móviles multiplataforma a partir de tecnologías web como HTML5, CSS3 y Javascript. PhoneGap está realizado sobre Apache Cordova, fue creado por Nitobi, actualmente es propiedad de Adobe y crea de las llamadas aplicaciones híbridas, a mitad de camino entre las nativas y las puramente webs, esto lo consigue a partir de una serie de APIs que permiten controlar características de los dispositivos como el acelerómetro, la cámara, las notificaciones o la geolocalización (Siles, 2013).

Las aplicaciones que podemos desarrollar con PhoneGap se pueden publicar en las conocidas tiendas de aplicaciones (Google Play, Windows Store o App Store de Apple) y, al igual que las aplicaciones nativas, también son capaces de acceder a los periféricos de los dispositivos como la cámara, acelerómetro, etc (Baquero, 2015).

Lo mejor de PhoneGap es que permite escribir una única vez el código de la aplicación, con tecnologías HTML5, y publicarlo en cualquier plataforma móvil dentro de las más conocidas. Así que, o bien desarrollas tu aplicación en nativo para cada uno de los sistemas operativos, con el consiguiente trabajo que ello conlleva, o la programas una única vez usando alguna plataforma como PhoneGap (Baquero, 2015).

[...] PhoneGap es un excelente camino para resolver necesidades de creación de aplicaciones de una manera única y compatible con todos los dispositivos. Las ventajas saltan a la vista en este sentido, pero como todo en la vida también tiene su lado malo, ya que en rendimiento y posibilidades nunca va a poder llegar a la altura del desarrollo nativo. Por tanto, escoger o no

PhoneGap para el desarrollo de un nuevo producto es una decisión que hay que tomar con cuidado (Baquero, 2015).

2.4 Base de datos

2.4.1 MySql

Es un sistema de base de datos relacional, es más rápido, más confiable y más barato según la mayoría. El hecho es que hay un número cada vez mayor de usuarios de MySQL, y la gran mayoría de ellos están bastante satisfechos con MySQL. Por lo tanto, para estos usuarios podemos decir que MySQL es lo suficientemente bueno (Kofler, 2001).

Según Kofler (s.f., pp. 7,8) las características de MySQL son:

- Sistema de base de datos relacional. Al igual que casi todos los demás sistemas de bases de datos en el mercado, MySQL es un sistema de base de datos relacional.
- Arquitectura cliente/servidor. Hay un servidor de base de datos (MySQL) y arbitrariamente muchos clientes (programas de aplicación), que se comunican con el servidor; Es decir, las consultas de datos, guardar cambios, etc.
- SQL. MySQL soporta como su lenguaje de base de datos SQL (*Structured Query Language*). SQL es un lenguaje estandarizado de consulta y actualización de datos y para la administración de una base de datos.
- Lenguajes de programación. Hay una gran cantidad de API (interfaces de programación de aplicaciones) y bibliotecas para el desarrollo de aplicaciones MySQL. Para la programación del cliente puede utilizar, entre otros, los lenguajes C, C ++, Java, Perl, PHP, Python y Tcl.

- ODBC. Hay una interfaz ODBC para MySQL. Con esto, MySQL puede ser abordado por todos los lenguajes de programación habituales que se ejecutan bajo Microsoft Windows (Delphi, Visual Basic, etc.). MyODBC se encuentra actualmente en la versión 2.5, nivel 0. La interfaz ODBC también se puede instalar bajo Unix, lo cual es raramente necesario.
- Independencia de la plataforma. No solo las aplicaciones cliente pueden ejecutarse en varios sistemas operativos. MySQL también puede ejecutarse bajo una variedad de sistemas operativos. Los más importantes son Apple Macintosh OS X, IBM OS / 2, Linux, Microsoft Windows, así como innumerables bocetos de Unix.
- Velocidad. MySQL es considerado un sistema de base de datos rápido. Esta evaluación ha sido apoyada por innumerables pruebas de referencia. En parte, la ventaja de la velocidad de MySQL es el resultado de la ausencia de ciertas características.

2.4.2 PostgreSQL

“Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado”(Zea Ordóñez & Molina Ríos, Jimmy Rolando Redrován Castillo, 2017).

“PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando” (Zea Ordóñez & Molina Ríos, Jimmy Rolando Redrován Castillo, 2017, p.12).

Según Zea Ordóñez & Molina Ríos, Jimmy Rolando Redrován Castillo (2017, p.12), son varias las características que definen a esta base de datos:

- Es una base de datos 100% ACID
- Integridad referencial
- *Tablespaces*
- *Nested transactions (savepoints)*
- Replicación asincrónica/sincrónica/*Streaming replication*
- *Two-phase commit*
- *Point in time recovery*
- Copias de seguridad
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- *Multi-Version Concurrency Control (MVCC)*
- Múltiples métodos de autenticación
- Acceso encriptado vía SSL
- Actualización in-situ integrada (*pg_upgrade*)
- SE-postgres
- Documentación completa
- Licencia BSD
- Disponible para Linux y UNIX en todas sus variantes.

2.4.3 SqlServer

SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial. SQL Server se ejecuta en T-SQL (Transact -SQL), un conjunto de extensiones de programación de Sybase y Microsoft que añaden varias características a SQL estándar, incluyendo control de transacciones, excepción y manejo de errores, procesamiento fila, así como variables declaradas (Margaret Rouse, 2017).

Bajo el nombre código Yukon en su etapa de desarrollo, SQL Server 2005 fue lanzado en noviembre de 2005. Se dice que el producto 2005 proporcionó una mayor flexibilidad, escalabilidad, confiabilidad y seguridad a las aplicaciones de base de datos, y permitió que fueran más fáciles de crear y desplegar, lo que reduce la complejidad y el tedio involucrado en la gestión de bases de datos. SQL Server 2005 también incluía más soporte administrativo (Margaret Rouse, 2017).

El código original de SQL Server ha sido desarrollado por Sybase; a finales de 1980, Microsoft, Sybase y Ashton-Tate colaboraron para producir la primera versión del producto, SQL Server 4.2 para OS/2. Posteriormente, tanto Sybase como Microsoft ofrecieron productos de SQL Server. Sybase cambió después el nombre de su producto a Adaptive Server Enterprise (Margaret Rouse, 2017).

Según Microsoft (2017), SQL Server Management Studio incluye las siguientes características generales:

- Compatibilidad con la mayoría de las tareas administrativas de SQL Server.
- Un entorno único integrado para la administración del Motor de base de datos de SQL Server y la creación.

- Cuadros de diálogo para administrar objetos de Motor de base de datos de SQL Server, Analysis Services y Reporting Services, lo que permite ejecutar las acciones inmediatamente, enviarlas a un editor de código o escribirlas en script para ejecutarlas posteriormente.
- Cuadros de diálogo no modales y de tamaño variable que permiten obtener acceso a varias herramientas mientras un cuadro de diálogo está abierto.
- Un cuadro de diálogo común de programación que permite realizar acciones de los cuadros de diálogo de administración en otro momento.
- Exportación e importación del registro de servidor de SQL Server Management Studio desde un entorno de Management Studio a otro.
- Guardado o impresión de archivos de plan de presentación XML o de interbloqueo generados por SQL Server Profiler, revisión posterior o envío a los administradores para su análisis.
- Un nuevo cuadro de mensaje de error e informativo que presenta mucha más información permite enviar a Microsoft un comentario sobre los mensajes, copiar mensajes en el Portapapeles y enviar fácilmente los mensajes por correo electrónico al equipo de soporte.
- Un explorador web integrado para una rápida exploración de MSDN o la Ayuda en pantalla.
- Integración de la Ayuda de comunidades en línea.
- Un tutorial sobre SQL Server Management Studio para ayudarle a aprovechar las ventajas de las numerosas características nuevas y a que sea más productivo de forma inmediata.

- Un nuevo monitor de actividad con filtro y actualización automática.
- Interfaces de Correo electrónico de base de datos integradas.

2.5 Selección de metodología y herramienta

He decidido crear un prototipo funcional de una aplicación híbrida debido a que la misma es una combinación de aplicación nativa y web; cuenta con varias ventajas como su bajo coste de diseño y desarrollo, su adaptabilidad a cualquier plataforma permite acceder a la funcionalidad y el hardware del teléfono mediante el uso de librerías, el código en este tipo de aplicaciones se escribe una sola vez y se adapta según sea la plataforma, además permite un prototipado rápido y sencillo.

A continuación, un cuadro que muestra las diferencias entre los frameworks Ionic, React y PhoneGap:

Tabla 1. Diferencias entre frameworks

<i>Ionic</i>	<i>React</i>	<i>PhoneGap</i>
<i>Basado en Angular respaldado por Google</i>	Basado en React respaldado por Facebook	Basado sobre Apache Cordova respaldado por Adobe
<i>Codificado en Typescript</i>	Codificado en JSX	Codificado en Javascript.
<i>No utiliza widgets nativos, solo representa todo en HTML.</i>	Utiliza plantillas para crear la GUI, crea widgets nativos	Utiliza plugins. Se puede ampliar con complementos nativos.
<i>Fácil de aprender con muchos componentes prefabricados y prediseñados</i>	Curva de aprendizaje con pocos componentes prefabricados	Fácil de aprender.
<i>Mediano rendimiento ya que utiliza WebView</i>	Muy buen rendimiento	Mediano rendimiento ya que utiliza WebView

<i>Documentación de primera clase, muy clara y consistente</i>	Documentación muy básica en comparación con Ionic	Documentación consistente
<i>Al correr el código en iOS se utiliza un interpretador JIT con plugin</i>	Al correr el código en iOS únicamente se utiliza el interpretador	Interpretado por el navegador.
<i>La lógica y el diseño están separados, se usa MVC.</i>	La lógica de visualización (diseño) se combina con la lógica de negocios.	La lógica y el diseño están separados
<i>Diseño clásico: HTML y CSS.</i>	Se crean y diseñan interfaces utilizando componentes propios nuevos similares a HTML.	Diseño clásico: HTML y CSS.
<i>“Write once, run everywhere”.</i>	“Learn once, write anywhere”.	“Write once, run everywhere”.
<i>Apache Cordova se usa para acceder a la funcionalidad del hardware del teléfono</i>	La aplicación React Native se compila en código nativo y puede acceder directamente a las características del teléfono nativo	Apache Cordova se usa para acceder a la funcionalidad del hardware del teléfono
<i>Ciclo de prueba de desarrollo muy rápido.</i>	Las pruebas se pueden realizar en el navegador, sin necesidad de cargar un emulador pesado	Las pruebas se pueden realizar en el navegador, sin necesidad de cargar un emulador pesado

Autoría propia.

Ionic tiene una gran ventaja de alcanzar más plataformas por la reutilización de código, asimismo las aplicaciones creadas se pueden visualizar en sitios web no solo Android e iOS, tiene la facilidad de delegar la lógica del diseño al propio framework y centrarse únicamente en la lógica de negocio, por otro lado, React Native no cuenta con esta facilidad y no permite en todos los casos utilizar el mismo código para todas las plataformas. Sin embargo, si se desea crear interfaces se puede utilizar Ionic Creator, herramienta de Ionic, que permite crear la interfaz de la aplicación sin necesidad de escribir el código de esta.

Se presenta el uso de Angular JS lo cual lo hace perfecto para el desarrollo de aplicaciones interactivas, de fácil mantenimiento y escalables que a su vez cuentan con una gama de herramientas y servicios. Además de tener un diseño cómodo, al crear aplicaciones híbridas para todos los dispositivos nos permite ahorrar dinero, tiempo y esfuerzo. En caso de que deseemos crear aplicaciones para Android, Ionic cuenta con una gran ventaja debido a que existe una fuerte unión entre los equipos de desarrollo de este con Google, lo cual le otorga una gran compatibilidad en estos dispositivos. Además, cuenta con la posibilidad de utilizar plugin de Phonegap (La conexión entre el cliente JS y el código nativo). Con Ionic tenemos lo mejor de PhoneGap, sus plugin; por estas razones considero que la mejor opción al momento de utilizar un framework de aplicaciones híbridas es Ionic.

La base de datos a utilizar es SQL Server debido a que esta es la utilizada por la empresa AndeanTrade S.A.

CAPÍTULO III: DESARROLLO DE APLICATIVO UTILIZANDO METODOLOGÍA ÁGIL MOBILE-D

3.1 Exploración

3.1.1 Primera etapa:

3.1.1.1 Identificación de actores

Desarrollador. La persona que lleva a cabo el análisis, diseño y desarrollo de la aplicación móvil, el tesista.

Coordinadores. Los coordinadores del área de ingeniería que son aquellos que utilizarán el aplicativo para asignar actividades y gestionar servicios.

Ingenieros. Son las personas que conforman al área de Ingeniería en la empresa y para los cuales está pensado el aplicativo para que visualicen las actividades designadas a cada uno.

3.1.1.2 Definición de alcance

Se va a implementar una aplicación híbrida que se basará en una investigación exploratoria y descriptiva que generará un entregable destinado a la asignación y control de actividades, tareas y soportes diarios del departamento de ingeniería en la empresa AndeanTrade S.A.

Finalmente, el proyecto terminará con la entrega del prototipo funcional a la empresa, en base a los requerimientos levantados en la fase de exploración.

3.1.2 Segunda etapa

3.1.2.1 Recolección de requerimientos

Se busca realizar un prototipo funcional para la empresa AndeanTrade S.A mediante la cual se automatice el proceso de agendamiento de actividades en el área de ingeniería, para ello se ha hablado con el personal correspondiente de la empresa y de esta manera se han fijado los requerimientos necesarios.

3.1.3 Tercera etapa

3.1.3.1 Identificación de Procesos

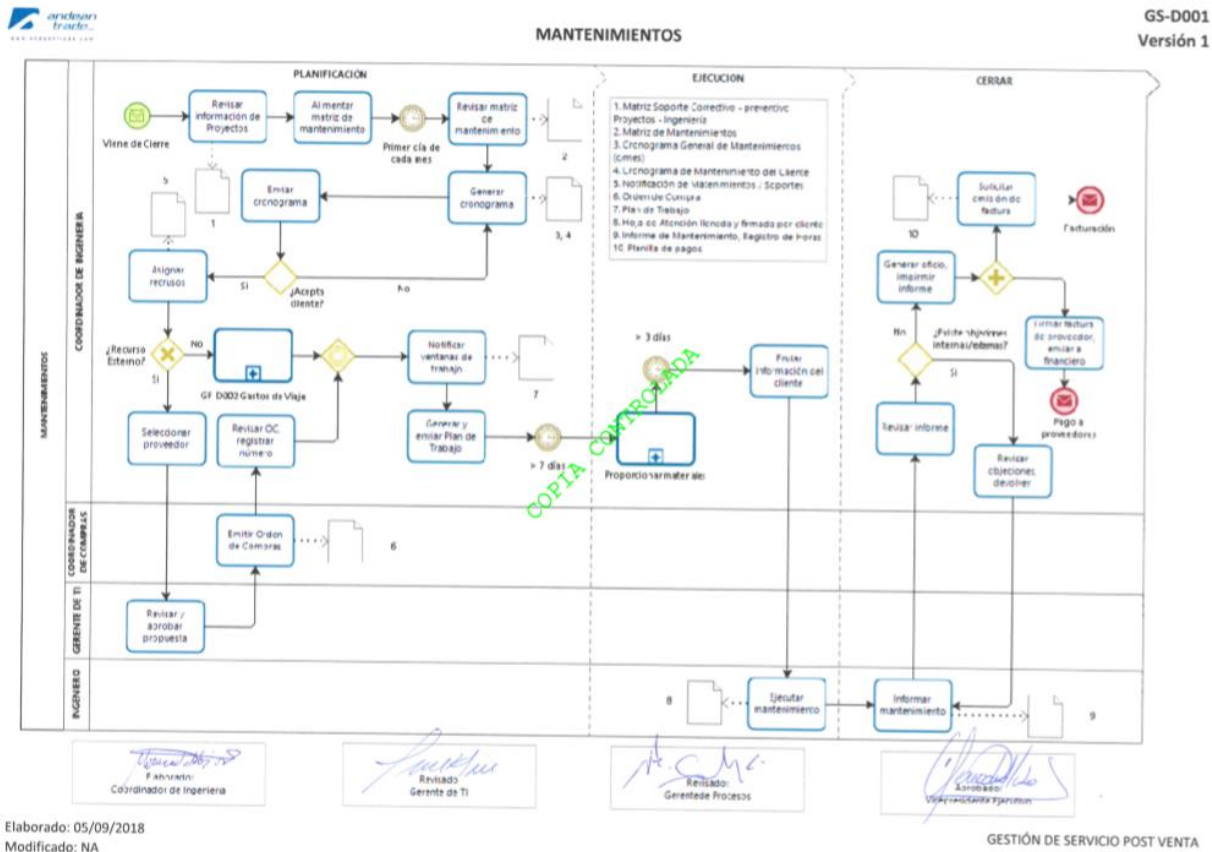
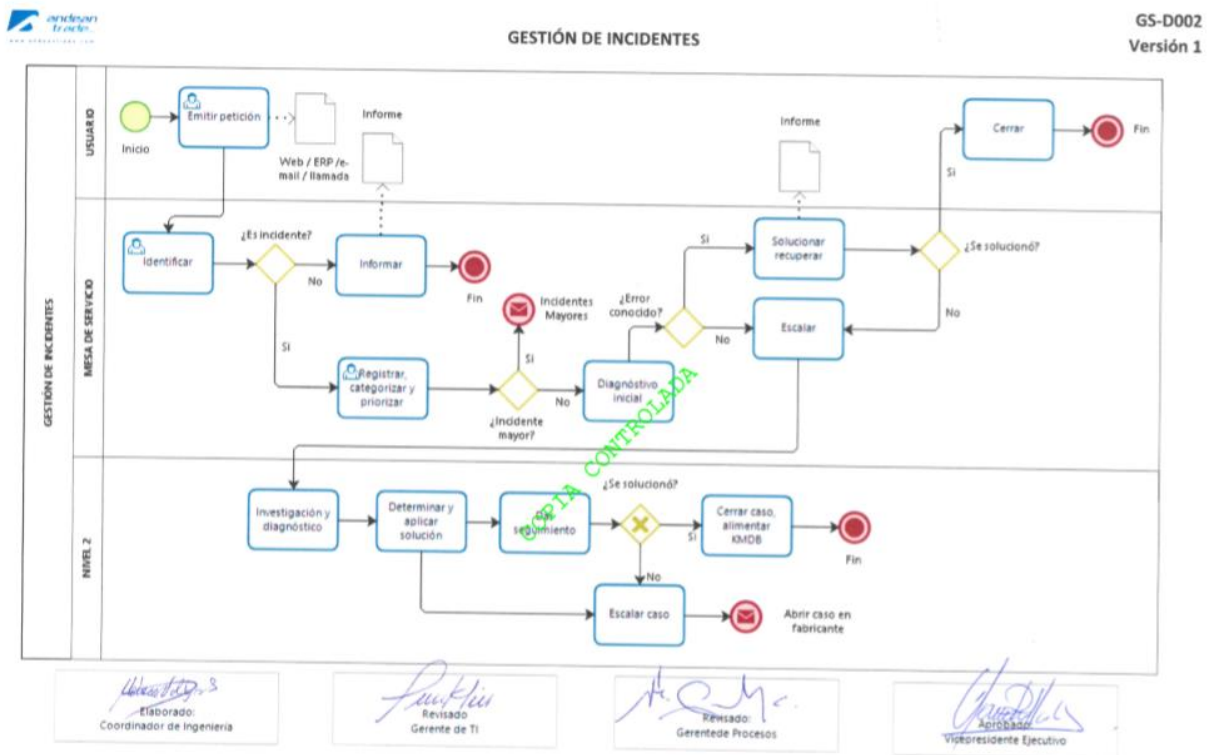


Ilustración 12. Proceso de mantenimiento. GS-D001 Versión 1 AndeanTrade S.A



Elaborado: 03/09/2018
Modificado: NA

GESTIÓN DE SERVICIO POST VENTA

Ilustración 13. Proceso de gestión de incidentes. GS-D002 Versión 1 AndeanTrade S.A

3.2.2 Requerimientos iniciales

3.2.2.1 Requerimientos funcionales

Tabla 2. Requerimientos funcionales.

<i>Código</i>	<i>Descripción</i>
RF001	Verificar información de usuario
RF002	Visualizar información general de los servicios
RF003	Gestionar los servicios
RF004	Filtrar las actividades por fecha
RF005	Gestionar las actividades
RF006	Visualizar información general de las actividades por usuario
RF007	Visualizar información sobre aplicativo

Autoría propia.

3.2.2.1 Requerimientos no funcionales

Tabla 3. Requerimientos no funcionales.

<i>Código</i>	<i>Descripción</i>
RNF001	El aplicativo es desarrollado en un framework.
RNF002	El aplicativo cuenta con una interfaz amigable.
RNF003	El aplicativo tiene una arquitectura cliente-servidor.
RNF004	El aplicativo es multiusuario.
RNF005	El aplicativo muestra diferentes opciones según el rol del usuario.
RNF006	El aplicativo es móvil e híbrido.

Autoría propia.

3.2.3 Planificación inicial

3.2.3.1 Análisis de requerimientos iniciales

En base a los requerimientos señalados anteriormente, se ha podido definir su importancia para identificar los procesos y establecer los módulos del aplicativo.

Tabla 4. Requerimientos iniciales.

Módulo	Código	Descripción de Proceso	Requerimientos
Coordinador	P001	Iniciar sesión	RF001
	P002	Listar todos los servicios	RF002
	P003	Ingresar un servicio	RF003
	P004	Editar un servicio	RF002, RF003
	P005	Eliminar un servicio	RF002, RF003
	P006	Crear actividades	RF005
	P007	Consultar actividades por fecha	RF004
	P008	Visualizar información sobre el aplicativo	RF007
Ingeniero	P001	Iniciar sesión	RF001
	P009	Listar las actividades designadas	RF006
	P010	Editar actividades designadas	RF005, RF006

Autoría propia.

3.2.3.1.1 Módulo coordinador

P001: Iniciar sesión:

- Verifica los datos ingresados
- Identifica el rol del usuario
- Muestra el menú de coordinador

P002: Listar todos los servicios:

- Consulta de servicios existentes en la base de datos de AndeanTrade S.A

P003: Ingresar un servicio:

- Registro de servicio
- Actualiza la lista de servicios existentes

P004: Editar un servicio:

- Consulta de servicios existentes en la base de datos de AndeanTrade S.A
- Indica los datos del servicio especificado

- Se editan los campos deseados
- Actualiza la lista de servicios existentes

P005: Eliminar un servicio:

- Consulta de servicios existentes en la base de datos de AndeanTrade S.A
- Indica una confirmación del proceso de eliminación
- Elimina el servicio especificado
- Actualiza la lista de servicios existentes

P006: Crear actividades:

- Consulta de clientes y usuarios existentes en la base de datos de AndeanTrade S.A
- Registro de actividad
- Actualiza la lista de actividades existentes

P007: Consulta actividades por fecha:

- Selección de fecha
- Consulta de actividades existentes en la base de datos de AndeanTrade S.A
- Se filtran los resultados en función de la fecha seleccionada

P008: Visualizar información sobre el aplicativo:

- Se muestra la información personal del desarrollador del aplicativo

3.2.3.1.2 Módulo ingeniero

P001: Iniciar sesión:

- Verifica los datos ingresados

- Identifica el rol del usuario
- Muestra el menú de ingeniero

P009: Listar las actividades designadas:

- Se identifica el usuario
- Consulta de actividades existentes en la base de datos de AndeanTrade S.A
- Se filtran los resultados en función del usuario

P010: Editar actividades designadas:

- Consulta de actividades existentes en la base de datos de AndeanTrade S.A
- Se filtran los resultados en función del usuario
- Se editan los campos deseados
- Actualiza la lista de actividades existentes

3.2.4 Planificación de fases

Tabla 5. Planificación de fases

Fase	Iteración	Descripción
Exploración	Iteración 0	Identificación de actores, definición de alcance, recolección de requerimientos, identificación de procesos
Inicialización	Iteración 0	Configuración de ambiente de desarrollo, requerimientos iniciales, planificación inicial, planificación de fases

Producción	Iteración 1: Módulo coordinador	Implementación del módulo coordinador. Presentar servicios y actividades desde la base de datos de AndeanTrade, gestionar los servicios existentes.
	Iteración 2: Módulo ingeniero	Implementación del módulo ingeniero. Presentar actividades desde la base de datos de AndeanTrade.
Estabilización	Iteración 1: Módulo coordinador	Ajustes del módulo coordinador, modificación en la interfaz de usuario.
	Iteración 2: Módulo ingeniero	Ajustes del módulo ingeniero, modificación en la interfaz de usuario.
Pruebas	Iteración 1 y 2	Evaluación de Pruebas. Análisis de resultados

Autoría propia.

3.3 Producción

3.3.1 Arquitectura de la aplicación



Ilustración 15. Arquitectura de la aplicación. Autoría propia.

3.3.2 Modelo físico de la Base de datos

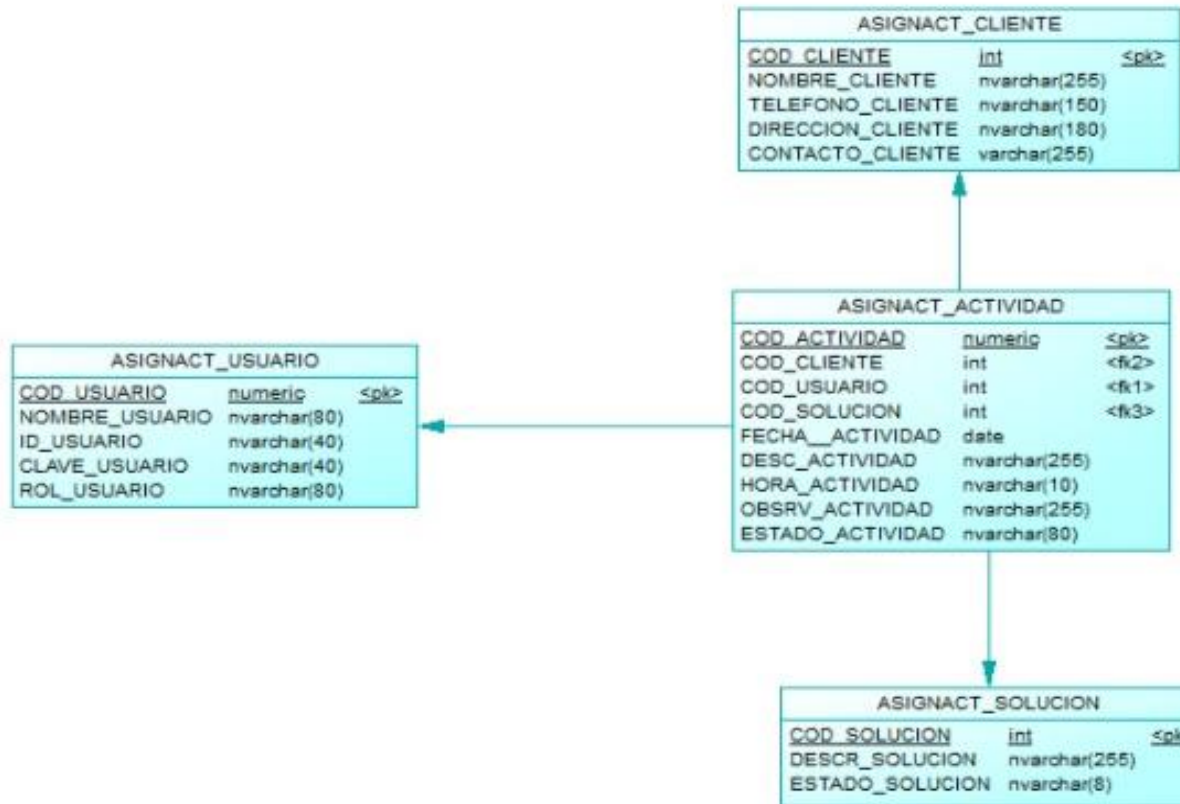


Ilustración 16. Modelo físico de la base de datos. Autoría propia.

3.3.3 Descripción de módulos

3.3.3.1 Módulo de coordinador

P001: Iniciar sesión

Se presenta la pantalla principal de la aplicación (Login) la cual permite verificar el usuario y la clave ingresada mediante *web service* que las compara con las existentes en la base de datos de la empresa. Además, verifica el rol del usuario y con respecto a este se mostrará la interfaz correspondiente.

Esta pantalla muestra el formulario de ingreso al sistema el cuál cuenta con un botón que convierte la contraseña de tipo *password* a tipo texto en caso de que el usuario requiera revisarla, además valida un máximo de número de caracteres y exige que ambos campos estén llenos para habilitar el botón de ingreso.

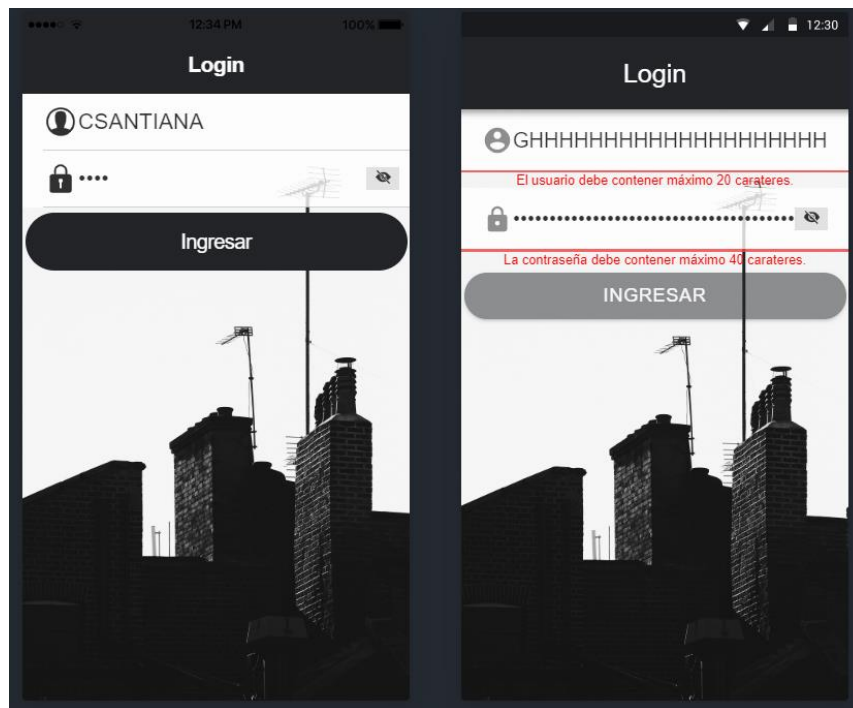


Ilustración 17. Pantalla de Login. Autoría propia.

Una vez que los datos han sido confirmados y se ha detectado el rol del usuario como coordinador se pasa a la pantalla de menú de coordinador el mismo que cuenta con 4 opciones: Gestión de servicios, agenda, sobre app y cerrar sesión.

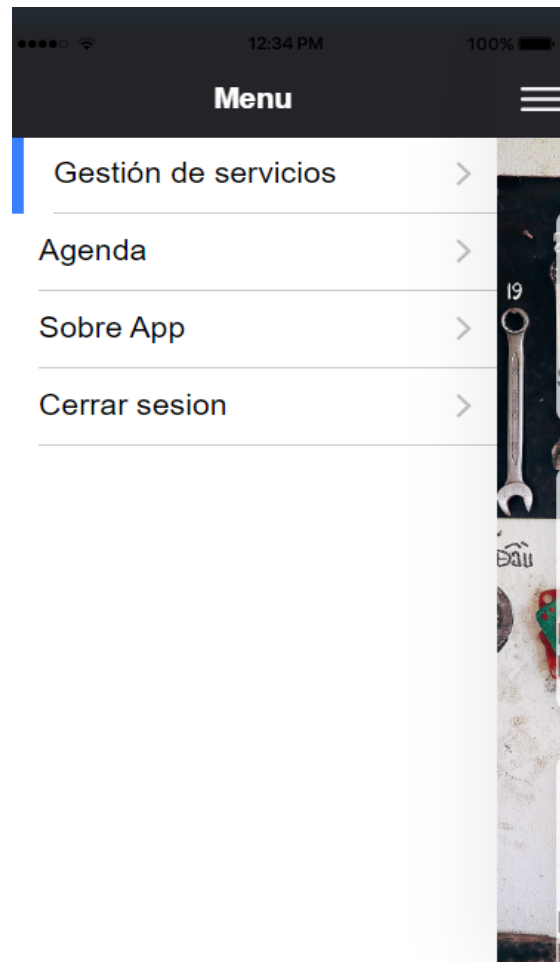


Ilustración 18. Pantalla Menú de coordinador. Autoría propia.

P002: Listar todos los servicios:

Cuando el usuario se encuentra en la opción de gestión de servicios, la pantalla mostrará el resultado de la consulta de servicios existentes en la base de datos de AndeanTrade S.A.



*Ilustración 19. Pantalla Gestión de servicios.
Autoría propia.*

P003: Ingresar un servicio:

Si el coordinador desea ingresar un servicio ingresará a la pantalla Ingreso de servicios la cuál mostrará un formulario con los campos descripción y estado que son necesarios para habilitar el botón de ingreso. Después de ingresar los

datos del formulario el aplicativo se comunicara mediante *web service* con el servidor para el ingreso del nuevo servicio en la base de datos de la empresa.

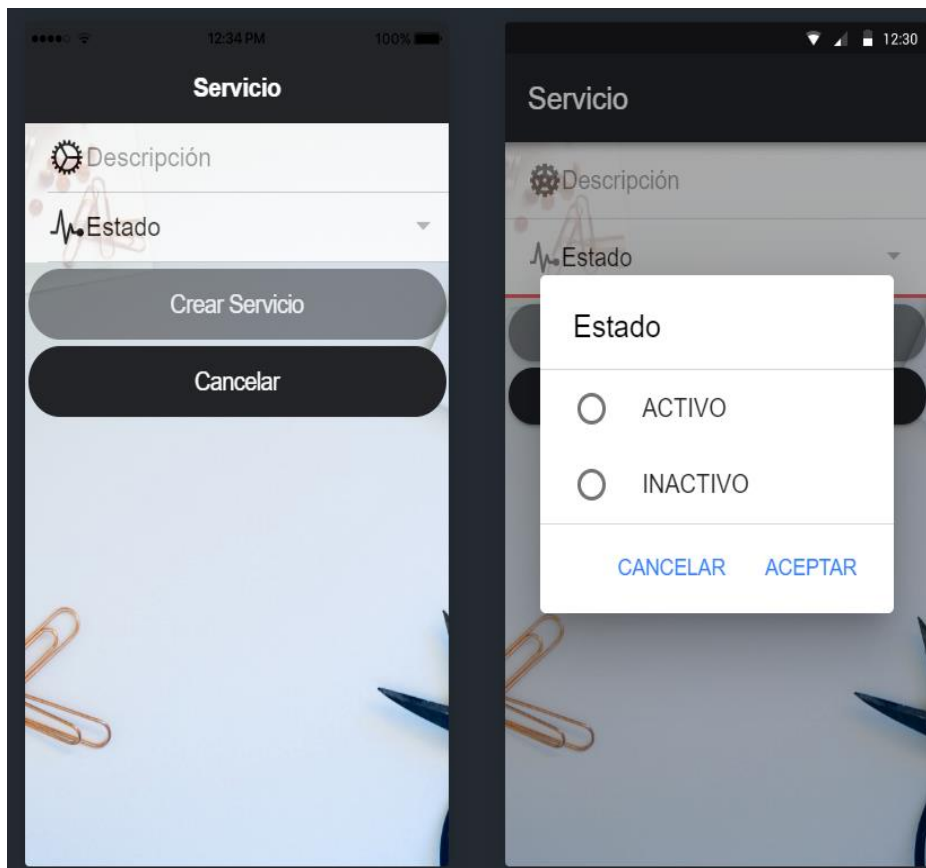


Ilustración 20. Pantalla Ingreso de servicios. Autoría propia.

P004: Editar un servicio:

Tras la consulta de servicios existentes en la base de datos de AndeanTrade S.A el coordinador en caso de ser necesario podrá modificar un servicio, se mostrarán los mismos campos de ingreso de formulario y el usuario deberá editar los campos que desee, mediante *web service* se enviarán los datos del servicio especificado y se modificarán en la base de datos de la empresa únicamente los campos establecidos por el usuario.

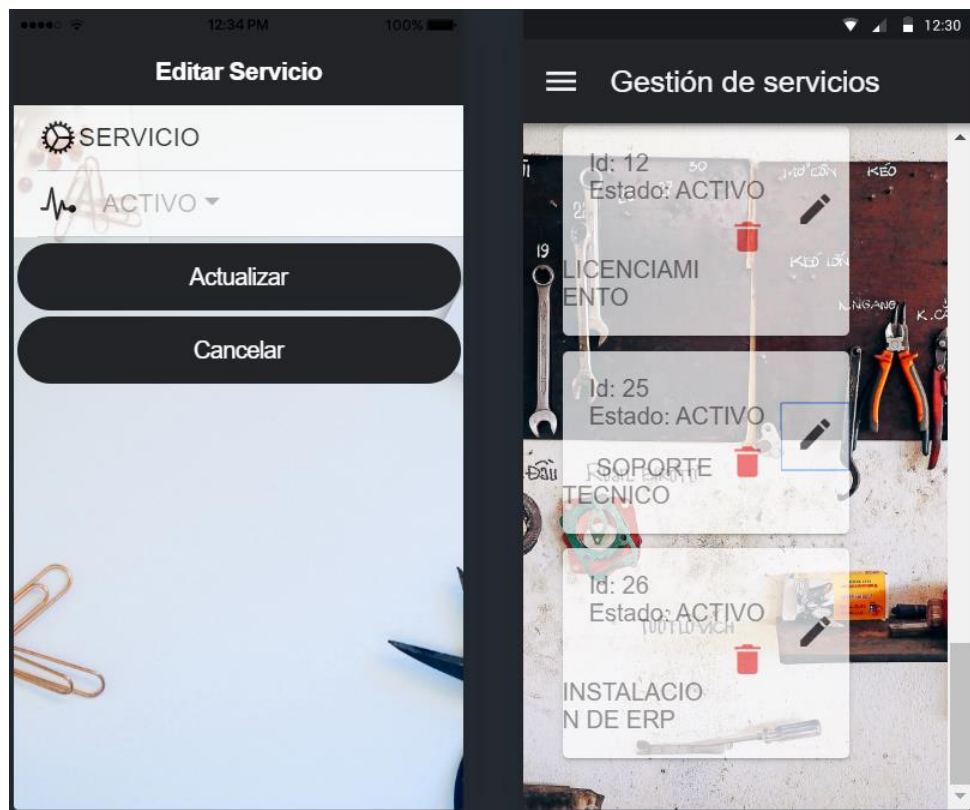


Ilustración 21. Pantalla Editar servicio. Autoría propia.

P005: Eliminar un servicio:

Si el coordinador desea eliminar un servicio se mostrará un *action sheet* que indicará la opción de confirmar el proceso de eliminación del servicio seleccionado. Después de aceptar la confirmación el aplicativo se comunicará mediante *web service* con el servidor enviándole a este un identificador del servicio y permitirá eliminar el servicio de la base de datos de la empresa.

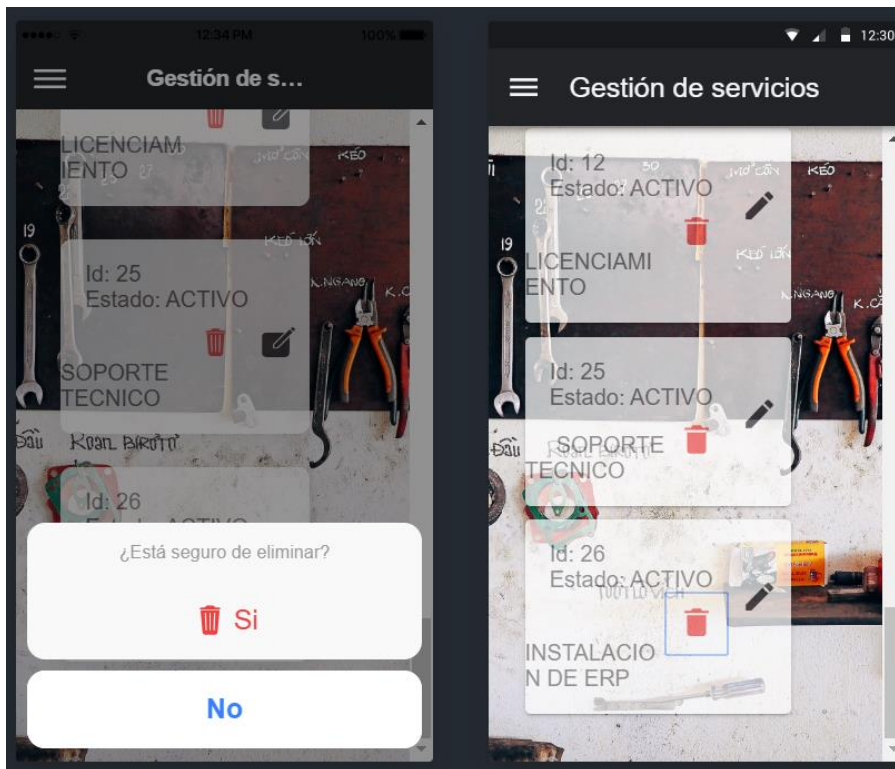


Ilustración 22. Pantalla Eliminar servicio. Autoría propia.

P006: Crear actividades:

Si el coordinador desea ingresar un servicio ingresará a la pantalla Ingreso de actividades la cuál mostrará un formulario con los campos: fecha, hora, ingeniero, solución, cliente y descripción que son necesarios para habilitar el botón de ingresar actividad. El usuario podrá hacer selección de ingeniero, solución y cliente puesto que serán los mismos datos que se encuentran almacenados en la base de datos de la empresa. Después de ingresar todos los datos del formulario el aplicativo se comunicara mediante *web service* con el servidor para el ingreso de la nueva actividad en la base de datos de la empresa.

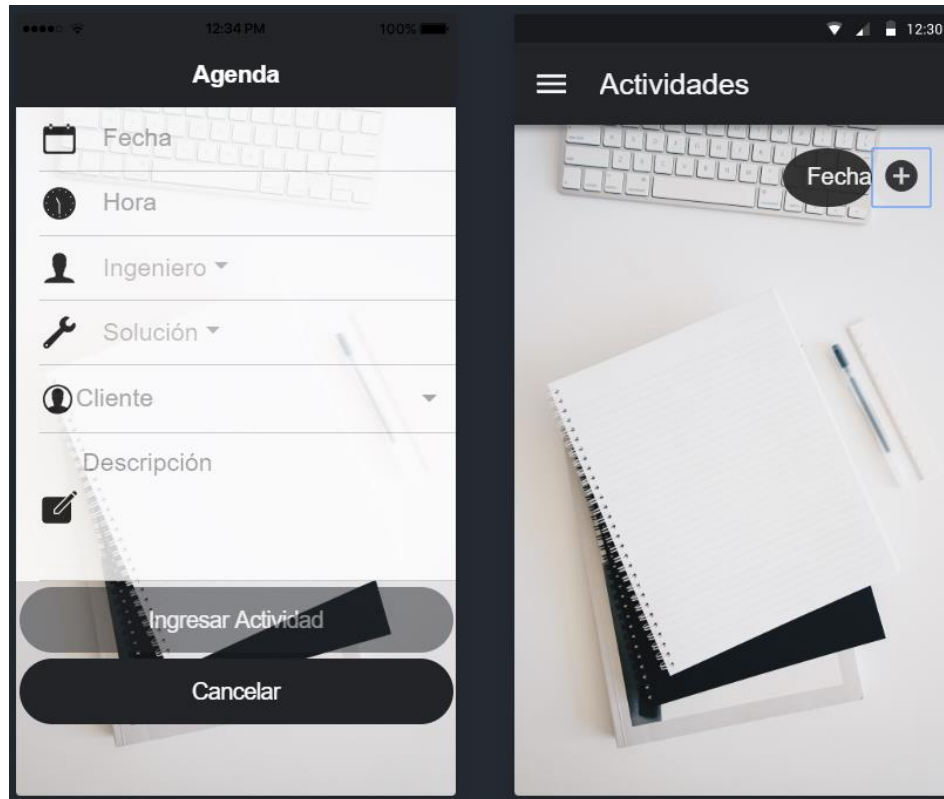


Ilustración 23. Pantalla Ingreso de actividades. Autoría propia.

P007: Consulta actividades por fecha:

Cuando el usuario se encuentra en la opción de agenda, la pantalla permitirá al usuario seleccionar una fecha, se enviará mediante *web service* una petición que consultará las actividades existentes en la base de datos de AndeanTrade S.A y las filtrará en función de la fecha seleccionada.

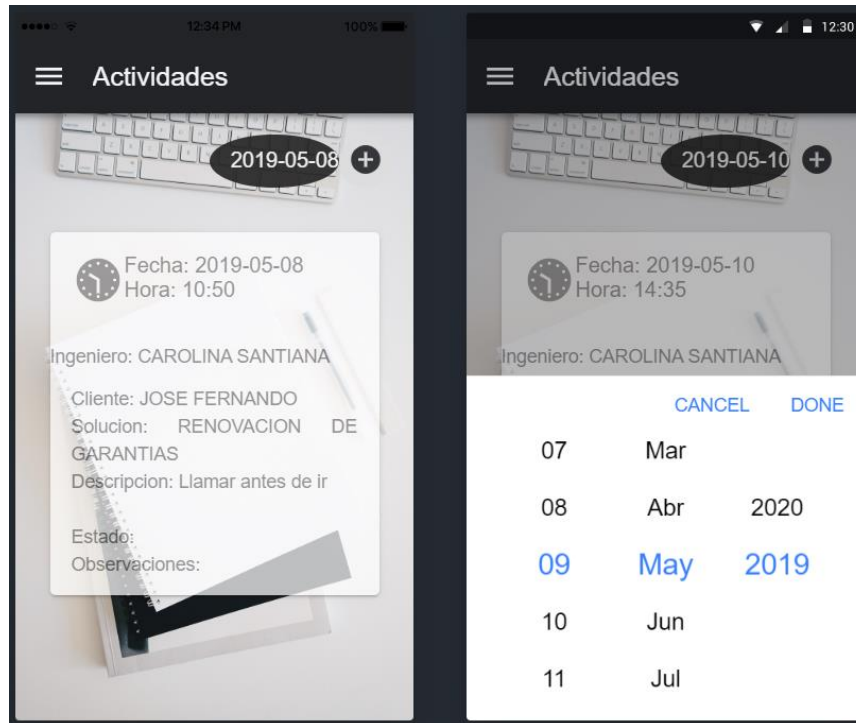


Ilustración 24. Pantalla Lista de actividades por fecha. Autoría propia.

P008: Visualizar información sobre el aplicativo:

Si el usuario desea visualizar la información del aplicativo al seleccionar la opción Sobre App en el menú y se desplegará una pantalla que contiene datos sobre el desarrollador, información personal sobre el mismo y el nombre del aplicativo.

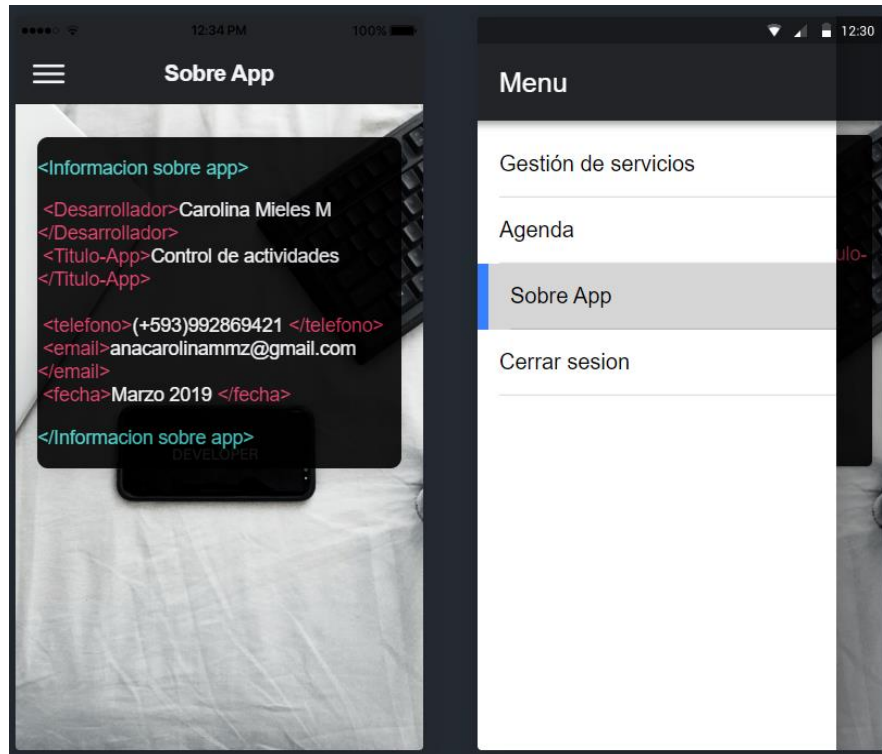


Ilustración 25. Pantalla Sobre App. Autoría propia.

3.3.3.2 Módulo de ingeniero

P009: Listar las actividades designadas:

Una vez que el usuario ha ingresado sesión y su rol ha sido identificado como ingeniero, se abrirá la pantalla Lista de Actividades. La pantalla mostrará el resultado de la consulta de las actividades existentes, designadas al usuario (ingeniero) correspondiente, en la base de datos de AndeanTrade S.A.

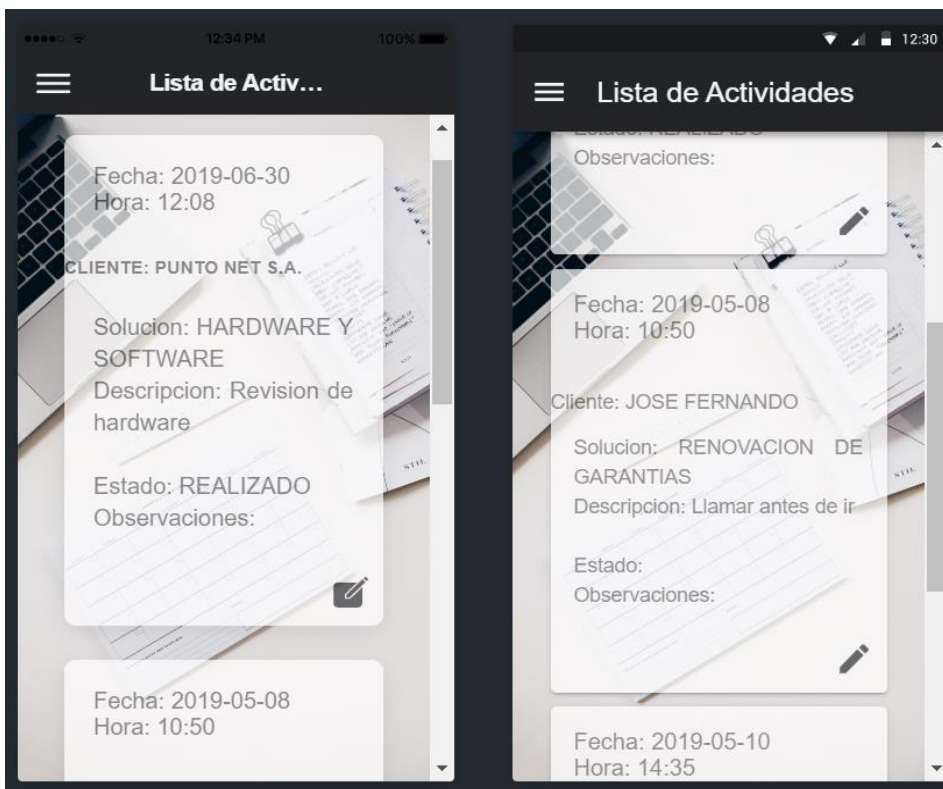


Ilustración 26. Pantalla Lista de actividades. Autoría propia.

P010: Editar actividades designadas:

Tras la consulta de actividades existentes filtradas por usuario (ingeniero) en la base de datos de AndeanTrade S.A el ingeniero en caso de ser necesario podrá modificar una actividad, se mostrarán únicamente los campos estado y observaciones; el usuario deberá editar los campos que desee, mediante *web service* se enviarán los datos de la actividad especificada y se modificarán en la base de datos de la empresa.

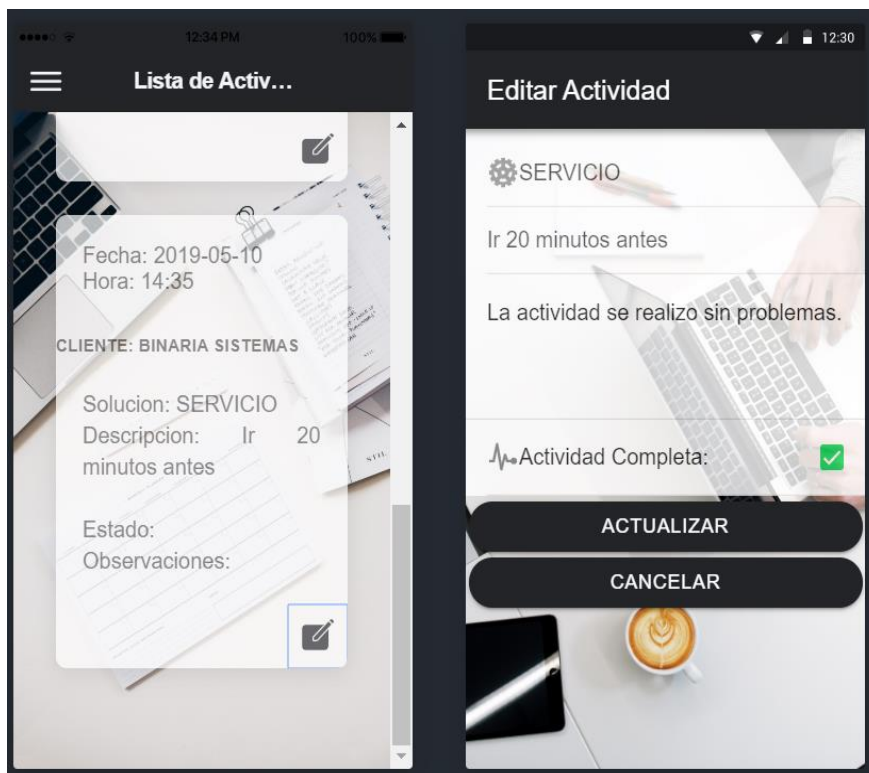


Ilustración 27. Pantalla Editar actividad. Autoría propia

3.4 Estabilización

“En esta fase se implementa la funcionalidad de la aplicación; además se asegura la calidad, y se comprueba que la aplicación implementada posea la funcionalidad correcta” (Tumipamba Borja, 2016).

Según Tumipamba Borja (2016), los objetivos de esta fase son:

- Implementar las funcionalidades
- Finalizar la implementación del producto
- Corregir defectos
- Producir un prototipo libre de errores dentro de lo posible

3.5 Pruebas

3.5.1. Pruebas de carga

Estas pruebas nos permiten verificar la capacidad que tiene el servidor una vez que se realizan peticiones simultáneamente, por ello he utilizado la herramienta en línea *Load Impact*, esta herramienta online “permite testear, y por lo tanto conocer, la capacidad que tiene tu sitio para soportar picos de demanda mucho más altos que lo habitual” (Klew, 2009).

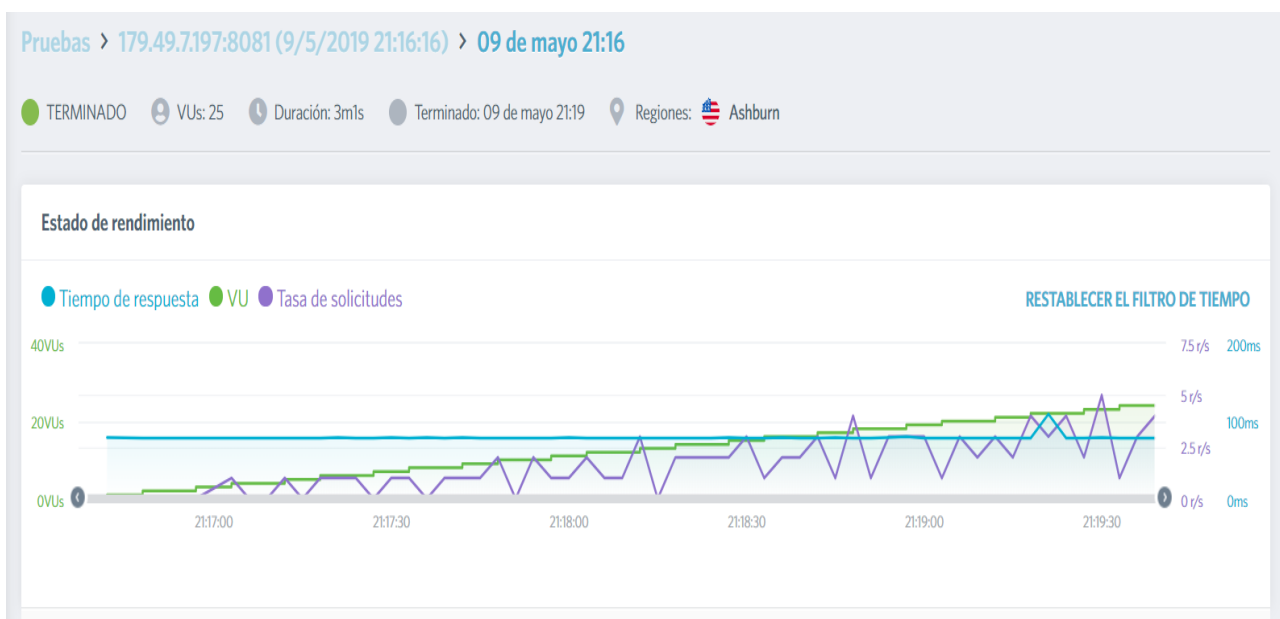


Ilustración 28. Prueba de carga. Load Impact.

Se puede visualizar en la figura 28 una simulación con 25 usuarios los cuáles aumentan de forma lineal, realizando solicitudes simultáneamente al servidor (la tasa promedio de transacciones es de 2,33 transacción/segundo); el tiempo de respuesta se mantiene constante siendo este aproximadamente de 100ms por usuario. Gracias a esta ilustración pudimos probar que el servidor tiene una gran capacidad y permitirá un gran número de usuarios a la vez por lo mismo se encuentra en un estado óptimo.

La herramienta nos permitió verificar el uso de memoria y CPU que únicamente fue del 1 y 4%, lo que nos indica que el servidor trabaja correctamente realizando transacciones entre usuarios a la misma vez.



Ilustración 29. Prueba de carga 2. Load Impact.

3.5.2. Pruebas unitarias

Una vez que se ha finalizado con un módulo a la par se han realizado pruebas unitarias para distintas funcionalidades del aplicativo móvil, en función de comprobar el correcto funcionamiento de estas. Las pruebas unitarias nos permiten evidenciar el funcionamiento adecuado de las partes de código por separado lo que nos brinda muchas ventajas.

Según Apiumhub (2017), algunas de estas ventajas son:

1. Como procedimiento ágil que es, te permite poder detectar los errores a tiempo, de forma que puedas reescribir el código o corregir errores sin necesidad de tener que volver al principio y rehacer el trabajo.
2. Al realizar pruebas continuamente y detectar los errores, cuando el código está terminado, es un código limpio y de calidad.
3. Detectar errores rápido.
4. Facilita los cambios y favorece la integración.
5. Gracias al continuo flujo de información y la superación de errores, se puede recopilar gran cantidad de información para evitar bugs venideros.
6. Ayudan en el proceso de *debugging*. Cuando se encuentra un error o bug en el código, solo es necesario desglosar el trozo de código testado. Esta es uno de los motivos principales por los que las pruebas unitarias se hacen en pequeños trozos de código, simplifica mucho la tarea de resolver problemas.
7. Reduce el coste.

Tabla 6. Prueba unitaria no.1

<i>Prueba unitaria 1</i>	
<i>Autenticar()</i>	
<i>Pruebas</i>	<i>Resultados</i>
<i>Se llenan los campos y se oprime Ingresar.</i>	<i>Se validan los datos e ingresa al sistema.</i>

<i>Se llenan los campos y excede el número de caracteres permitido.</i>	No se habilita el botón Ingresar.
<i>No se llena algún campo.</i>	No se habilita el botón Ingresar.

Autoría propia.

Tabla 7. Prueba unitaria no.2

<i>Prueba unitaria 2</i>	
<i>IngresarServicio()</i>	
<i>Pruebas</i>	<i>Resultados</i>
<i>Se llenan todos los campos y se oprime Ingresar.</i>	Se ingresan los datos.
<i>No se llenan los campos.</i>	No se habilita el botón Ingresar.
<i>Se oprime el botón Cancelar.</i>	No se ingresa y se regresa a la página anterior.

Autoría propia.

Tabla 8. Prueba unitaria no.3

<i>Prueba unitaria 3</i>	
<i>ActualizarServicio()</i>	
<i>Pruebas</i>	<i>Resultados</i>
<i>Se llenan todos los campos y se oprime Actualizar.</i>	Se modifican los datos.

<i>No se llenan los campos.</i>	No se habilita el botón Actualizar.
<i>Se oprime el botón Cancelar.</i>	No se actualiza y se regresa a la página anterior.

Autoría propia.

Tabla 9. Prueba unitaria no.4

Prueba unitaria 4	
EliminarActividad()	
Pruebas	Resultados
<i>Se confirma el proceso de eliminación.</i>	Se elimina el registro.
<i>No se confirma el proceso de eliminación.</i>	No se elimina.

Autoría propia.

Tabla 10. Prueba unitaria no.5

<i>Prueba unitaria 5</i>	
<i>IngresarActividad()</i>	
<i>Pruebas</i>	<i>Resultados</i>
<i>Se llenan todos los campos y se oprime Ingresar.</i>	Se ingresan los datos.
<i>No se llenan los campos.</i>	No se habilita el botón Ingresar.
<i>Se oprime el botón Cancelar.</i>	No se ingresa y se regresa a la página anterior.

Autoría propia.

Tabla 11. Prueba unitaria no.6

<i>Prueba unitaria 6</i>	
<i>ActualizarActividad()</i>	
<i>Pruebas</i>	<i>Resultados</i>
<i>Se llenan todos los campos y se oprime Actualizar.</i>	Se modifican los datos.
<i>No se llenan los campos.</i>	No se habilita el botón Actualizar.
<i>Se oprime el botón Cancelar.</i>	No se actualiza y se regresa a la página anterior.

Autoría propia.

CAPÍTULO IV: Entregables

4.1 Manual de usuario



ASIGNACT

MANUAL DE USUARIO

Diseño y desarrollo de una aplicación móvil
híbrida para el control de agenda del área
de ingeniería

Versión 1.0
Mayo de 2019

Contenido

1. Introducción	77
2. Requisitos del dispositivo.....	77
3. Navegación de la aplicación.....	78

Ilustraciones

Ilustración 1. Preferencias de USB.....	78
Ilustración 2. Archivo apk descargado	79
Ilustración 3. Instalación.....	79
Ilustración 4. Instalación 1.....	80
Ilustración 5. Instalación 2.....	80
Ilustración 6. Icono de aplicativo.....	81
Ilustración 7. Inicio de aplicación.....	81
Ilustración 8. Login.....	82
Ilustración 9. Menú Principal Coordinador.....	83
Ilustración 10. Gestión de servicios.....	84
Ilustración 11. Creación de servicio.....	85
Ilustración 12. Creación de servicio 1.....	86
Ilustración 13. Edición de servicio.....	87
Ilustración 14. Eliminación de servicio.....	88
Ilustración 15. Gestión de actividades.....	89
Ilustración 16. Selección de fecha.....	90
Ilustración 17. Resultado de búsqueda por fecha.....	91
Ilustración 18. Creación de actividad.....	92
Ilustración 19. Sobre App.....	93
Ilustración 20. Menú Principal - Ingeniero.....	94
Ilustración 21. Lista de Actividades.....	95
Ilustración 22. Edición de actividades.....	96

1. Introducción

El desarrollo de aplicaciones móviles ha adquirido una gran importancia principalmente en el ámbito empresarial no solo porque su rendimiento y agilidad sea mayor al de un sitio web, sino que además cuenta con varias ventajas como otorgar una facilidad en cuanto a su acceso, distribución, personalización y utilización de diversas funcionalidades del dispositivo.

Por ello, se desarrolló una aplicación móvil que permite gestionar las actividades diarias del área de ingeniería en la empresa Andean Trade S.A, se implementó bases de datos para almacenar la información personal y corroborar la integridad del usuario mediante una seguridad adecuada utilizando el motor de SQL Server, infraestructura de la empresa; de esta manera se podrá brindar a la empresa un seguimiento y automatización del proceso de agendamiento de ingeniería.

2. Requisitos del dispositivo

Las características mínimas que el dispositivo necesita para utilizar el aplicativo ASIGNAct son:

- Sistema Android 4.4 o superior
- 41 Mb de memoria en el dispositivo
- 1 Gb de memoria RAM
- Conectividad a internet (3G, 4G y/o Wifi)

3. Instalación de la aplicación

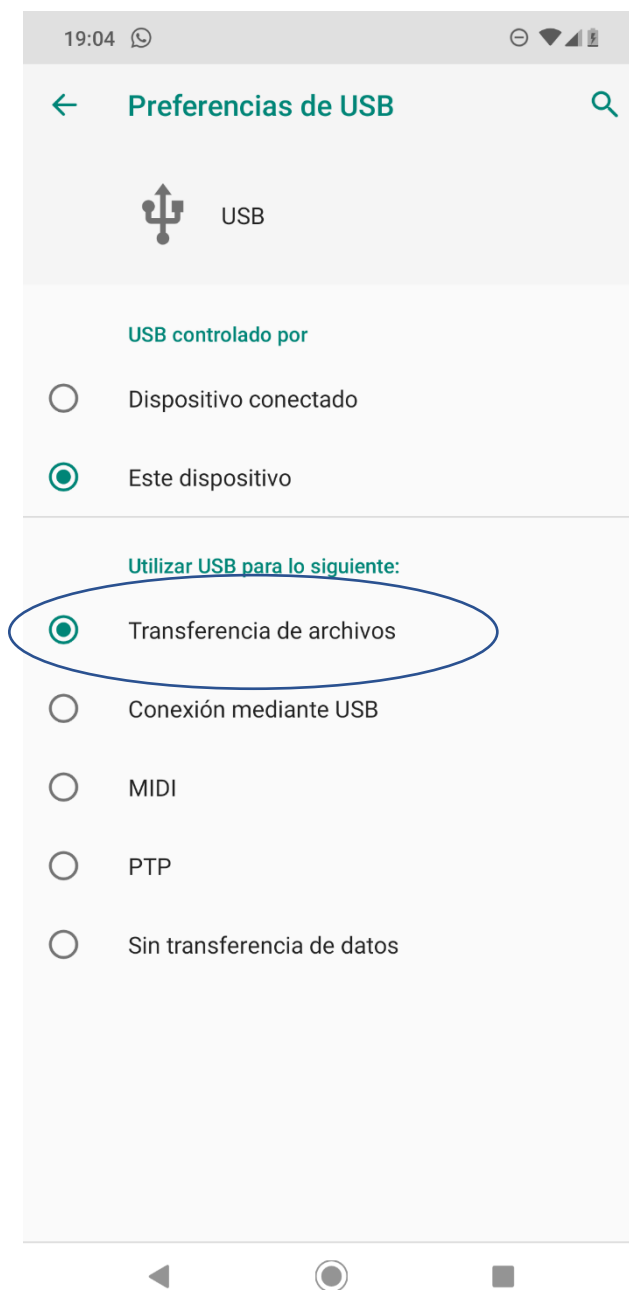


Ilustración 30. Preferencias de USB.

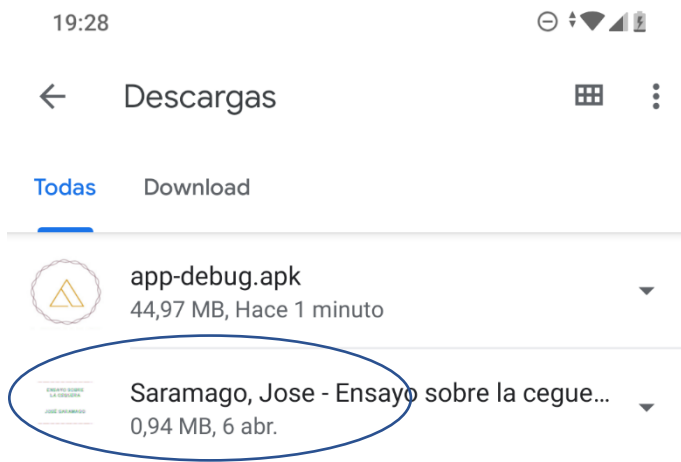


Ilustración 31. Archivo apk descargado

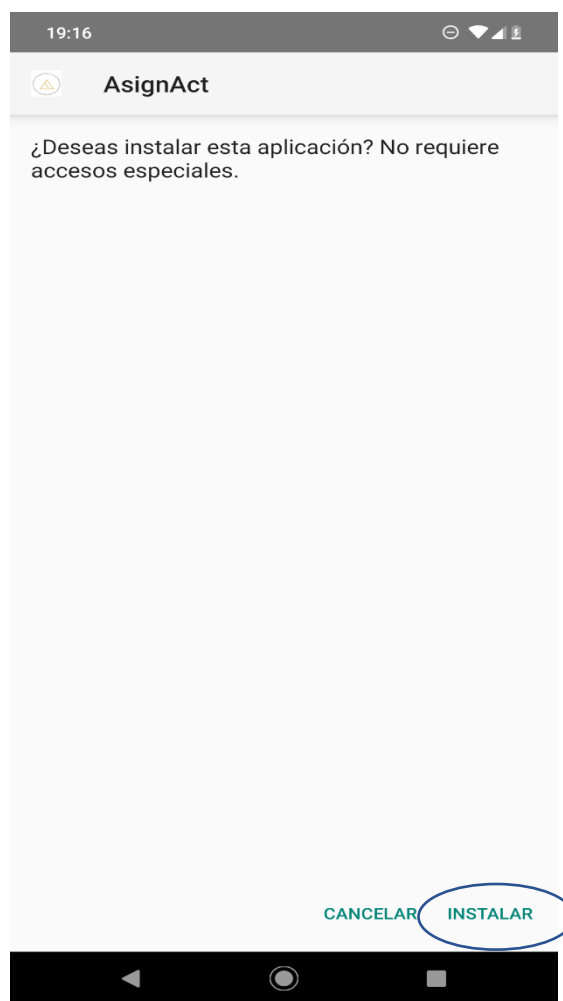


Ilustración 32. Instalación.

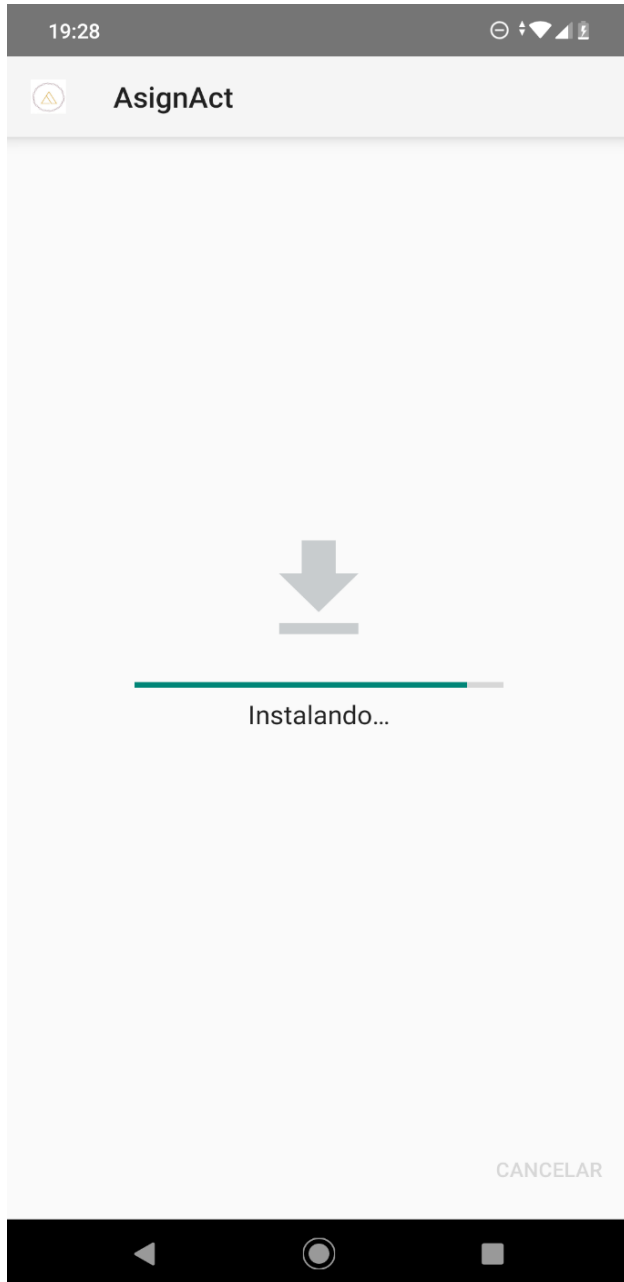


Ilustración 33. Instalación 1.

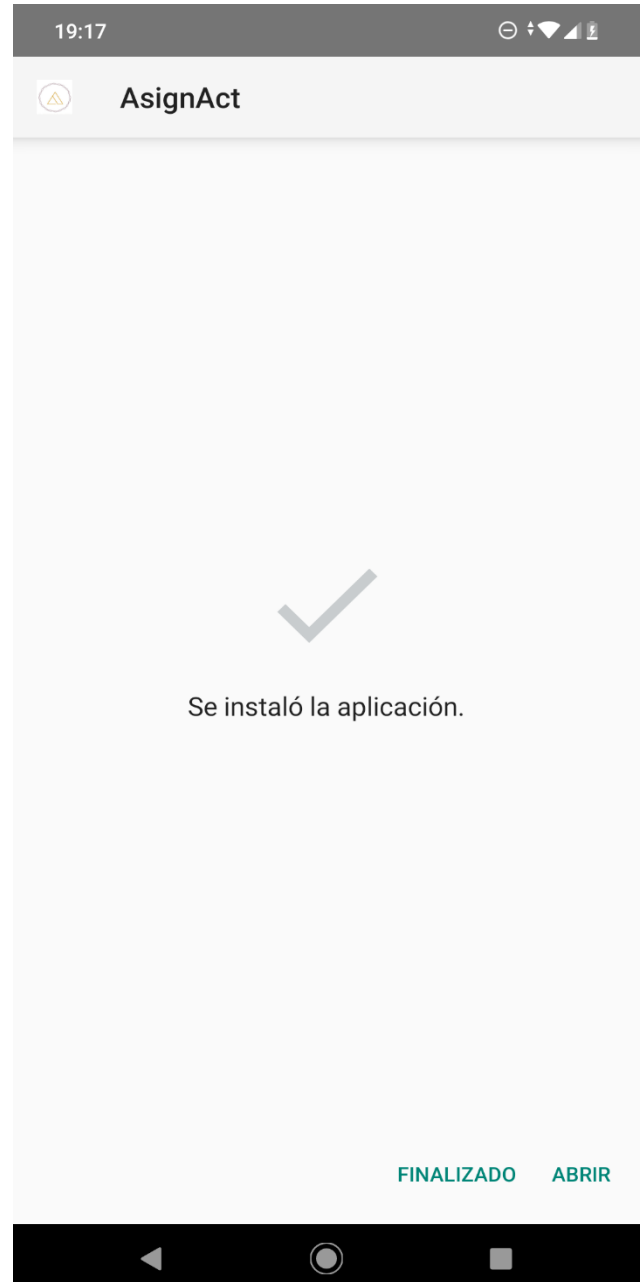


Ilustración 34. Instalación 2.

4. Navegación de la aplicación

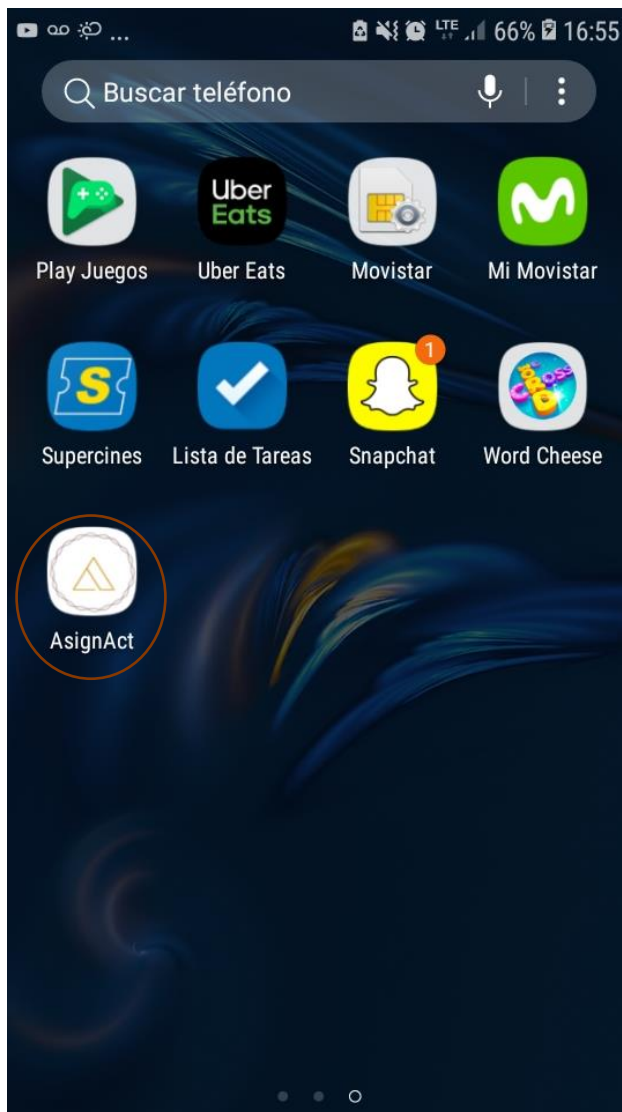


Ilustración 36. Icono de aplicativo.



Ilustración 35. Inicio de aplicación.

4.1 Login

La primera pantalla que se muestra tras el inicio es el Login que permitirá al usuario ingresar a la aplicación y gestionar las actividades. La primera opción de captura dentro de la pantalla de Login es usuario y la siguiente es clave. Tras validar los datos el aplicativo mostrará dependiendo el rol del usuario el menú correspondiente.



Ilustración 37. Login.

4.2 Menú Principal – Coordinador

Tras validar los datos de ingreso del usuario, si se identifica el rol de usuario como coordinador de ingeniería se mostrará la pantalla a continuación que muestra distintas opciones.

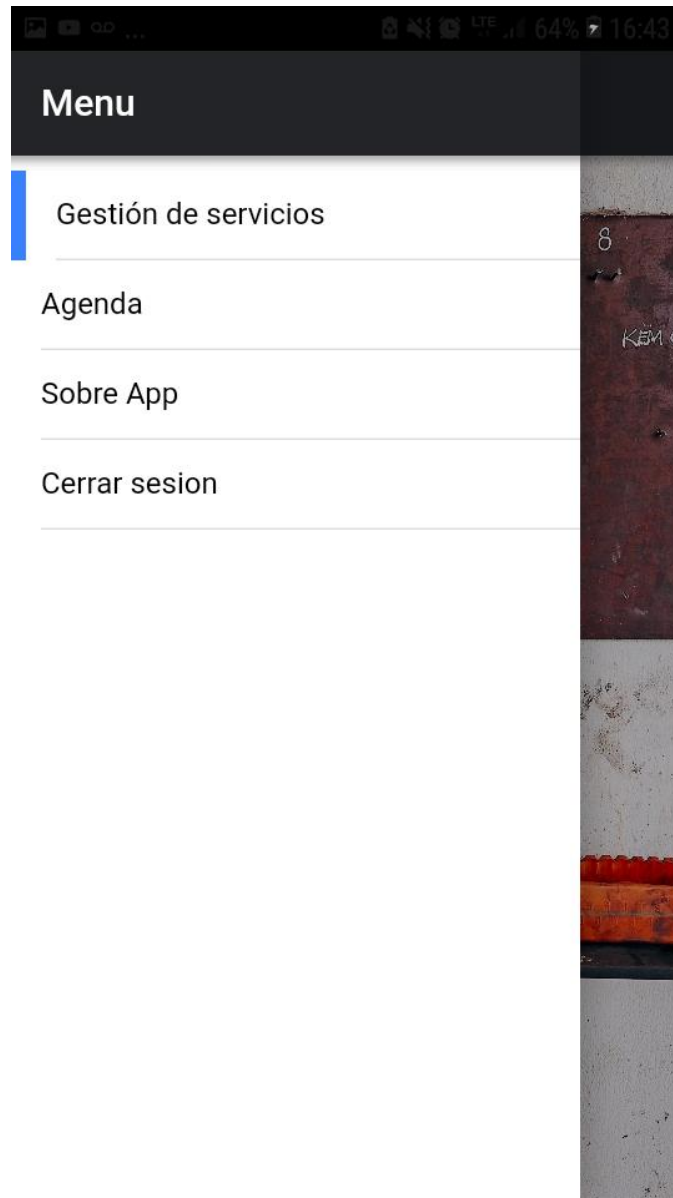


Ilustración 38. Menú Principal Coordinador.

4.3 Gestión de servicios

Cuando se muestra el menú principal de coordinador la primera opción siempre se mostrará, esta nos permite crear, editar y eliminar los servicios.

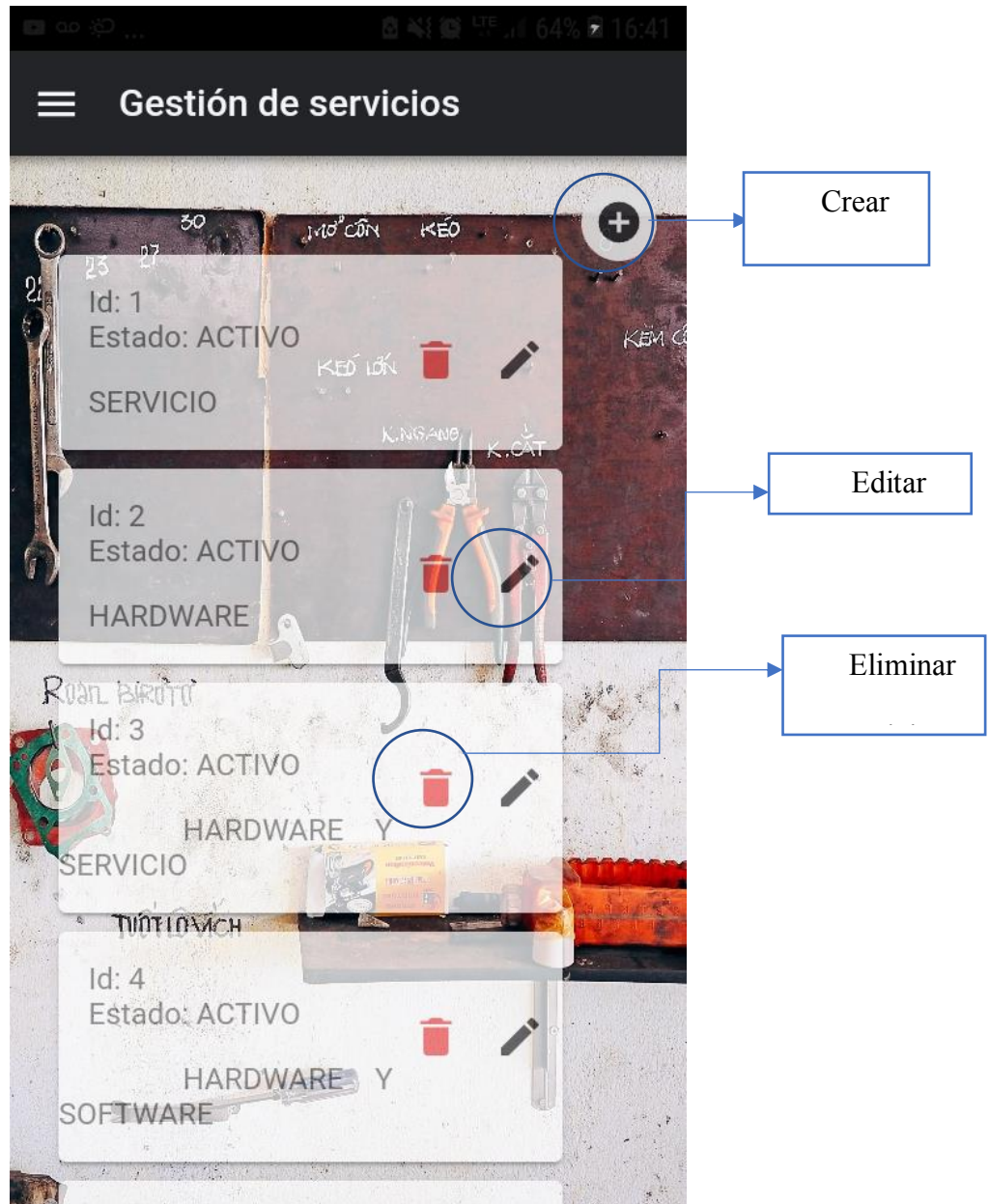


Ilustración 39. Gestión de servicios.

4.4 Agregar nuevo servicio

Tras apretar el botón de crear nuevo servicio se abrirá la siguiente pantalla solicitando los datos del nuevo servicio.

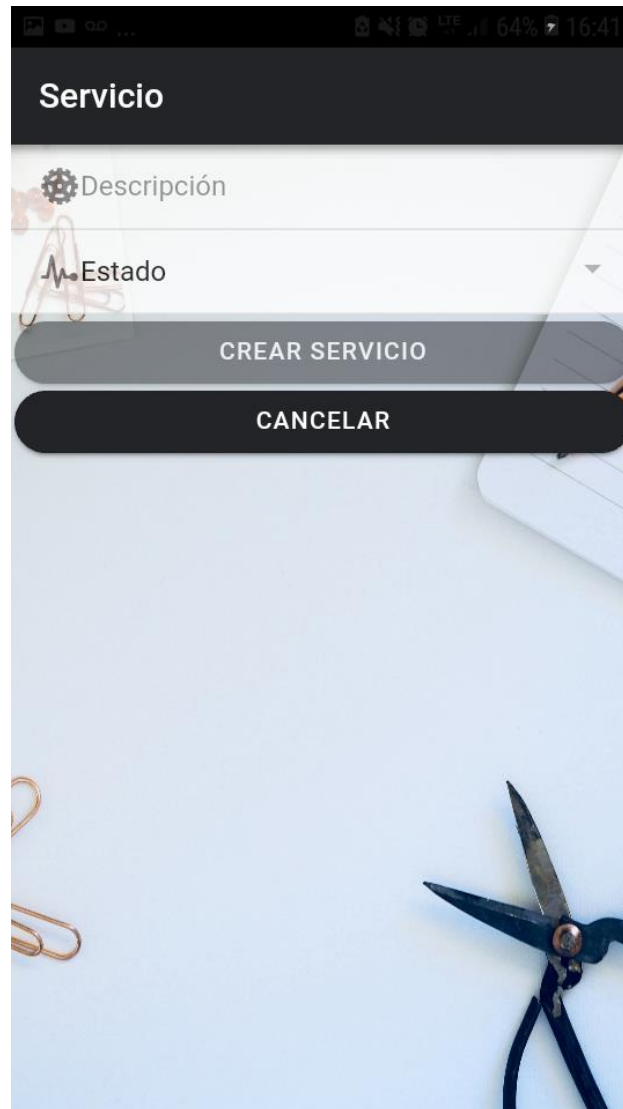


Ilustración 40. Creación de servicio.

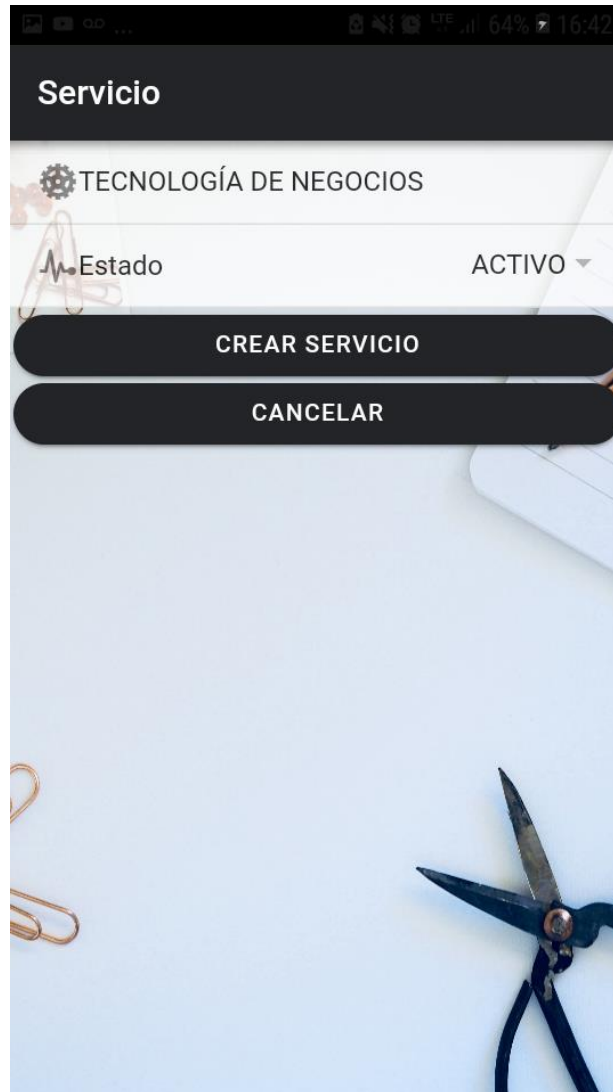


Ilustración 41. Creación de servicio 1.

4.5 Editar servicio

Tras apretar el botón de editar servicio se abrirá la siguiente pantalla mostrando los datos del servicio correspondiente y nos permitirá editarlos en caso de ser necesario y actualizarlo.

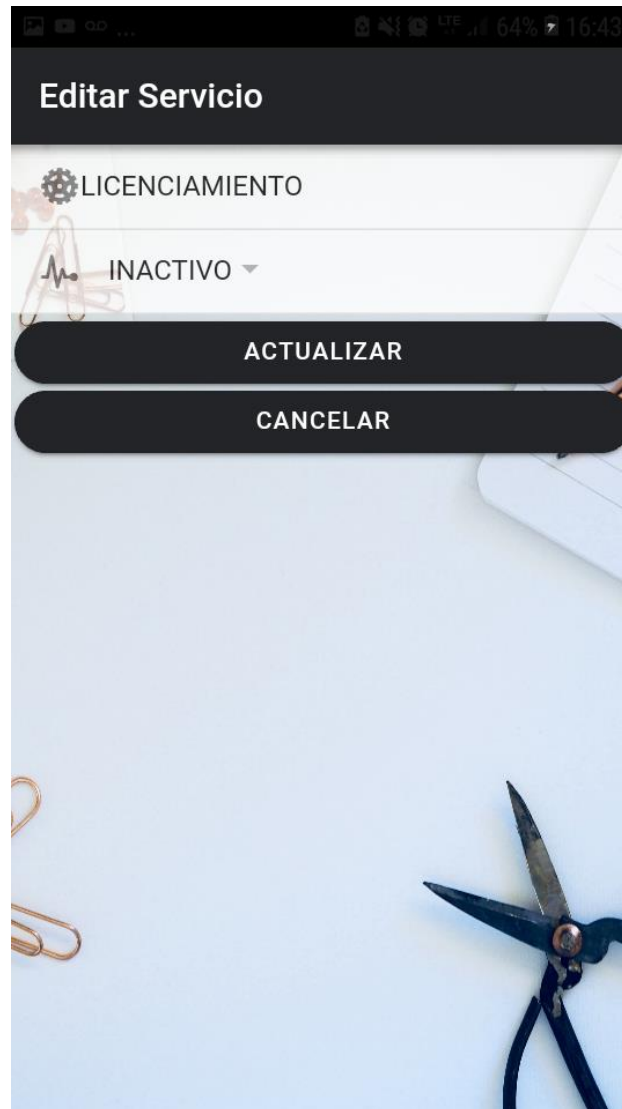
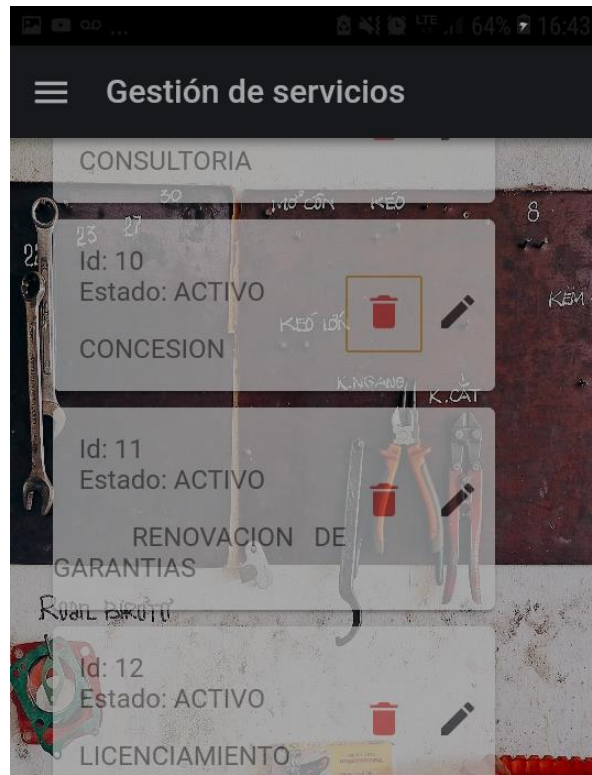


Ilustración 42. Edición de servicio.

4.6 Eliminar servicio

Tras apretar el botón de eliminar servicio se abrirá la siguiente pantalla mostrando una alerta de confirmación, una vez que hemos aceptado se eliminará el servicio correspondiente.



¿Está seguro de eliminar?

Si

No

Ilustración 43. Eliminación de servicio.

4.7 Agenda

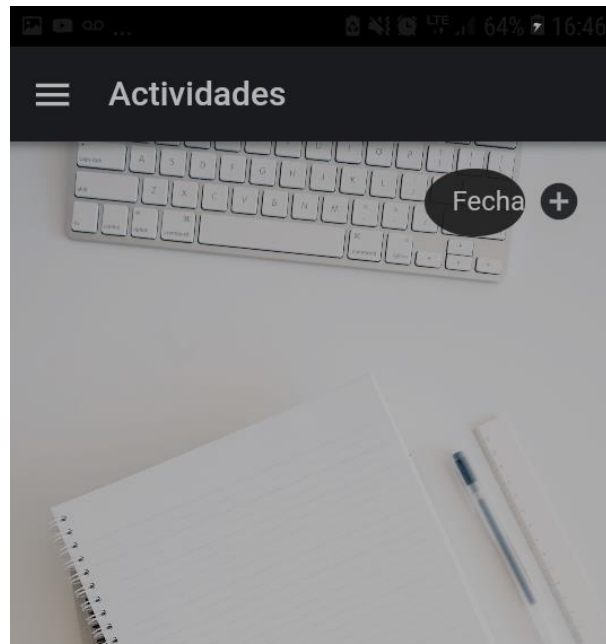
Al seleccionar en el menú la opción agenda, se presentará una pantalla que nos permitirá seleccionar una fecha para revisar las actividades que están asignadas a la fecha indicada.



Ilustración 44. Gestión de actividades.

4.8 Búsqueda de actividades por fecha

Una vez seleccionada la fecha, se buscarán las actividades que estén designadas a esta y se mostrarán en pantalla.



		CANCEL	DONE
20	Mar		
21	Abr	2020	
22	May	2019	
23	Jun		
24	Jul		

Ilustración 45. Selección de fecha.



Ilustración 46. Resultado de búsqueda por fecha.

4.9 Agregar nueva actividad

Tras apretar el botón de crear nueva actividad se abrirá la siguiente pantalla solicitando los datos necesarios para crear la actividad.



The screenshot shows a mobile application interface for creating a new activity. The title bar is dark with the word "Agenda" in white. Below the title bar, there are several input fields with icons on the left: a calendar icon for the date "2019-05-21", a clock icon for the time "13:45", a person icon for the name "PABLO SANT...", a wrench icon for the category "TECNOLOGÍA...", and a dropdown menu for "CORPORACION DE PROMOCION UNIVER...". Below these fields is a text input field containing "revision de negocios de TI". At the bottom of the screen, there are two large, dark buttons with white text: "INGRESAR ACTIVIDAD" and "CANCELAR". The background of the form is a light gray with a faint image of a keyboard and a notebook.

Ilustración 47. Creación de actividad.

4.10 Sobre App

Esta opción del menú nos permite visualizar la información sobre el aplicativo y sobre el desarrollador de este.



Ilustración 48. Sobre App.

4.11 Pantalla Principal – Ingeniero

Tras validar los datos de ingreso del usuario, si se identifica el rol de usuario ingeniero se mostrará la pantalla a continuación que muestra la pantalla de actividades asignadas y la opción de cerrar sesión.



Ilustración 49. Menú Principal – Ingeniero

4.12 Lista de actividades

Cuando se muestra el menú principal de ingeniero la primera pantalla que se mostrará es la lista de actividades, esta nos permite visualizar las actividades que tenemos asignadas y además permite editar las actividades indicando si está ha sido completada y colocando las observaciones necesarias.

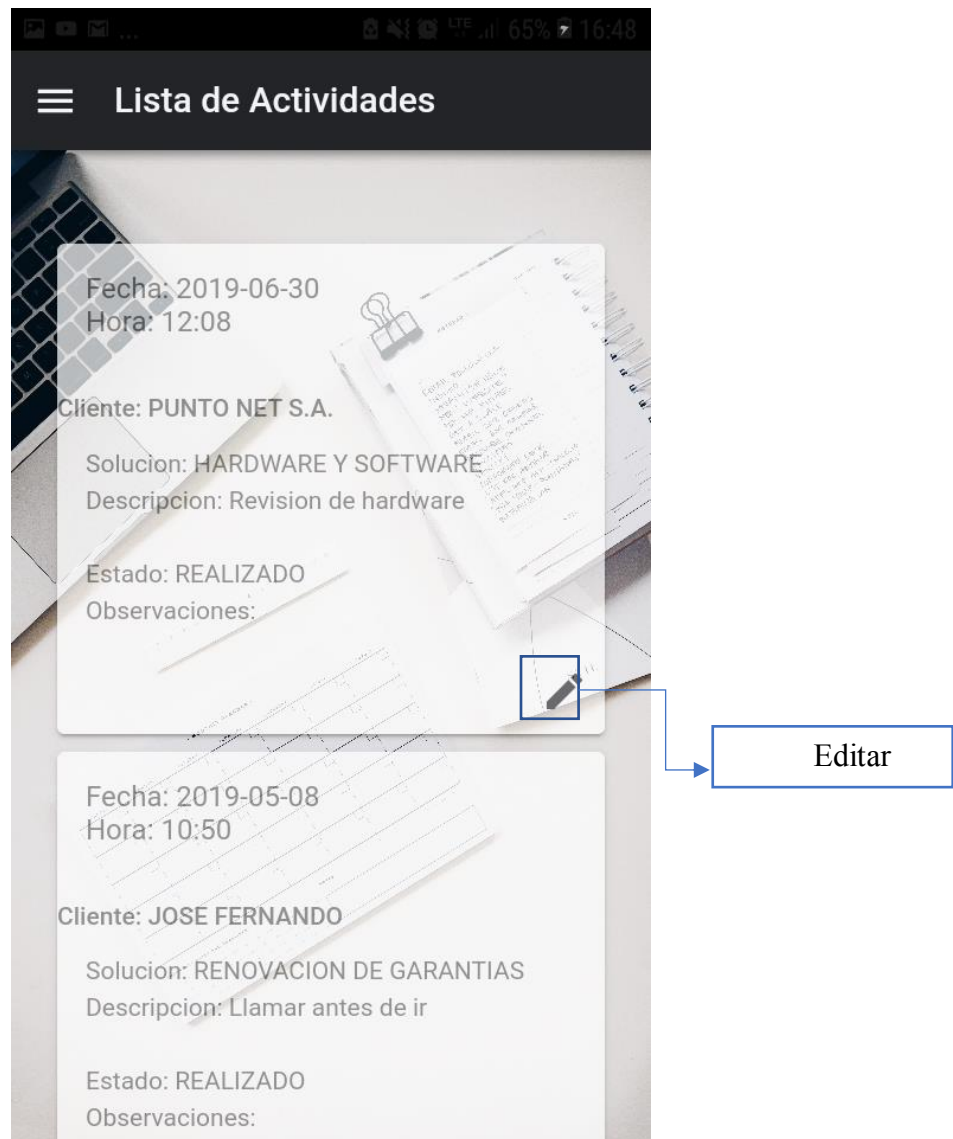


Ilustración 50. Lista de Actividades.

4.13 Edición de actividades



Ilustración 51. Edición de actividades.

4.2 Manual técnico



MANUAL DE USUARIO

Diseño y desarrollo de una aplicación móvil
híbrida para el control de agenda del área
de ingeniería

Versión 1.0
Mayo de 2019

Contenido

1. Introducción	100
2. Requerimientos técnicos	100
Requerimientos del computador.....	100
Requerimientos del dispositivo	100
3. Software utilizado	100
4. Instalación de herramientas de software	102
5. Código Fuente	108
5.1 Inicio de sesión	108
5.2 Gestión de servicios	113
5.3 Crear servicio.....	117
5.4 Editar servicio.....	119
5.5 Eliminar servicio.....	122
5.6 Agenda	124
5.7 Crear Actividades.....	129
5.8 Lista de Actividades.....	131
5.9 Editar Actividad	133
5.10 Sobre App.....	135

Ilustraciones

Ilustración 1. Instalación.....	102
Ilustración 2. Instalación 1.....	103
Ilustración 3. Instalación 2.....	103
Ilustración 4. Instalación 3.....	104
Ilustración 5. Instalación 4.....	104
Ilustración 6. Instalación 5.....	105
Ilustración 7. Instalación 6.....	105
Ilustración 8. Instalación 7.....	106
Ilustración 9. Instalación 8.....	106
Ilustración 10. Instalación 9.....	107
Ilustración 11. Inicio de sesión.....	108
Ilustración 12. Gestión de srevicios.....	113
Ilustración 13. Crear servicio.....	117
Ilustración 14. Editar servicio.....	119
Ilustración 15. Eliminar servicio.....	122
Ilustración 16. Agenda.....	124
Ilustración 17. Crear Actividades.....	129
Ilustración 18. Lista de Actividades.....	131
Ilustración 19. Editar Actividad.....	133
Ilustración 20. Sobre App.....	136

1. Introducción

Este manual busca brindar información acerca de la lógica diseñada para el aplicativo ASIGNAct. Para la creación de la base de datos se utilizó el motor de SqlServer, mismo que se encuentra en el servidor de la empresa, además en este servidor se crearon web services que permitieron el intercambio de datos entre el aplicativo y la base de datos. Se presentará a continuación el software necesario que se utilizó para la creación del aplicativo y los requerimientos mínimos para la instalación de los programas y del aplicativo.

2. Requerimientos técnicos

Requerimientos del computador

- Windows 7/8
- Core i3/5/7
- 1 GB de RAM
- Disco duro 500GB
- Privilegios de administrador
- Acceso a escritorio remoto para conexión al servidor

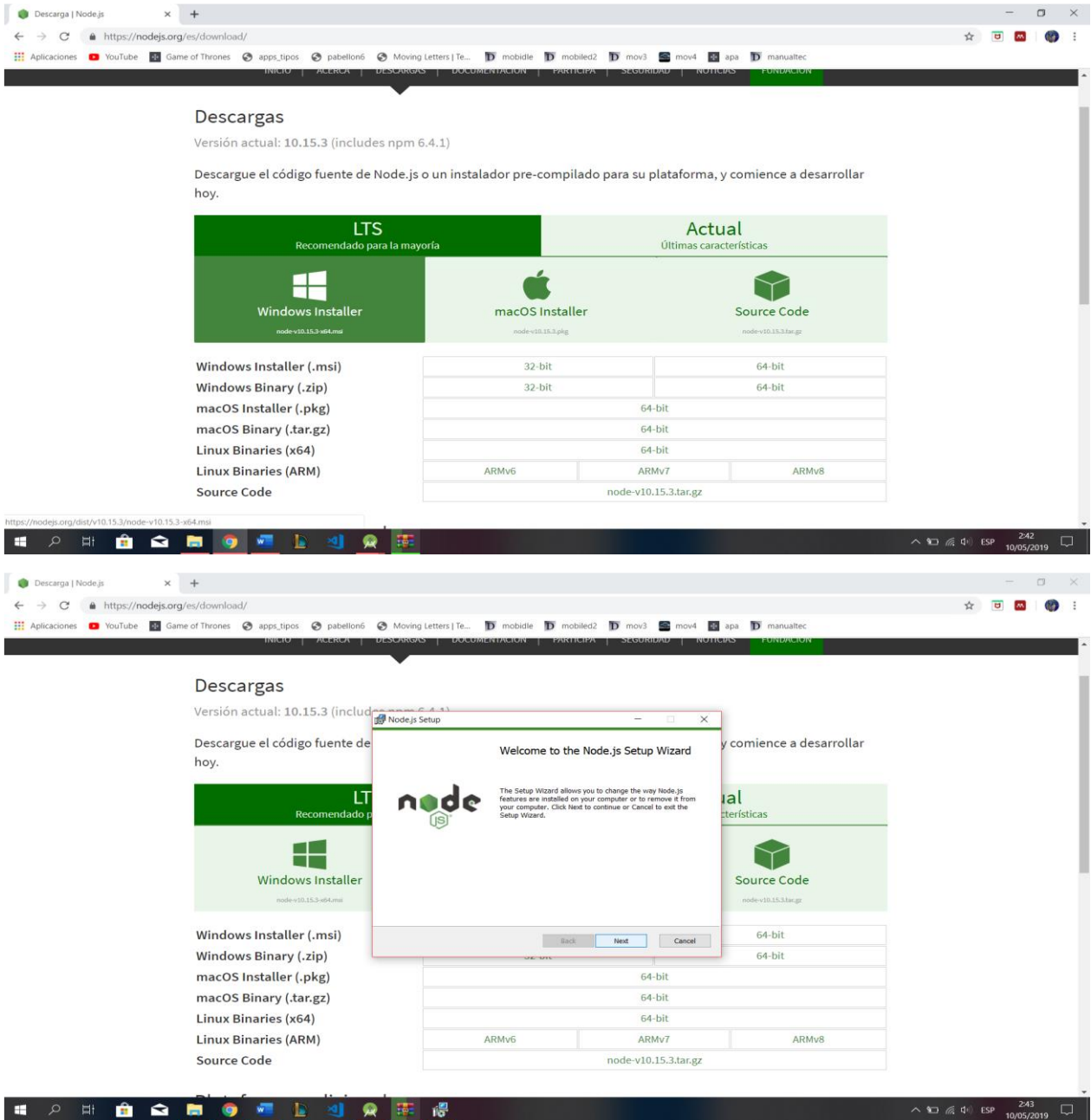
Requerimientos del dispositivo

- Sistema Android 4.4 o superior
- 41 Mb de memoria en el dispositivo
- 1 Gb de memoria RAM
- Conectividad a internet (3G, 4G y/o Wifi)

3. Software utilizado

- PHP y extensiones
- Apache
- SQLServer
- Node Js
- Ionic
- Visual Studio Code
- Android Virtual Device

4. Instalación de herramientas de software



1 Install Ionic

```
npm install -g ionic
```

Ilustración 52. Instalación.

2 Start an app

```
ionic start myApp tabs
```

Ilustración 53. Instalación 1.

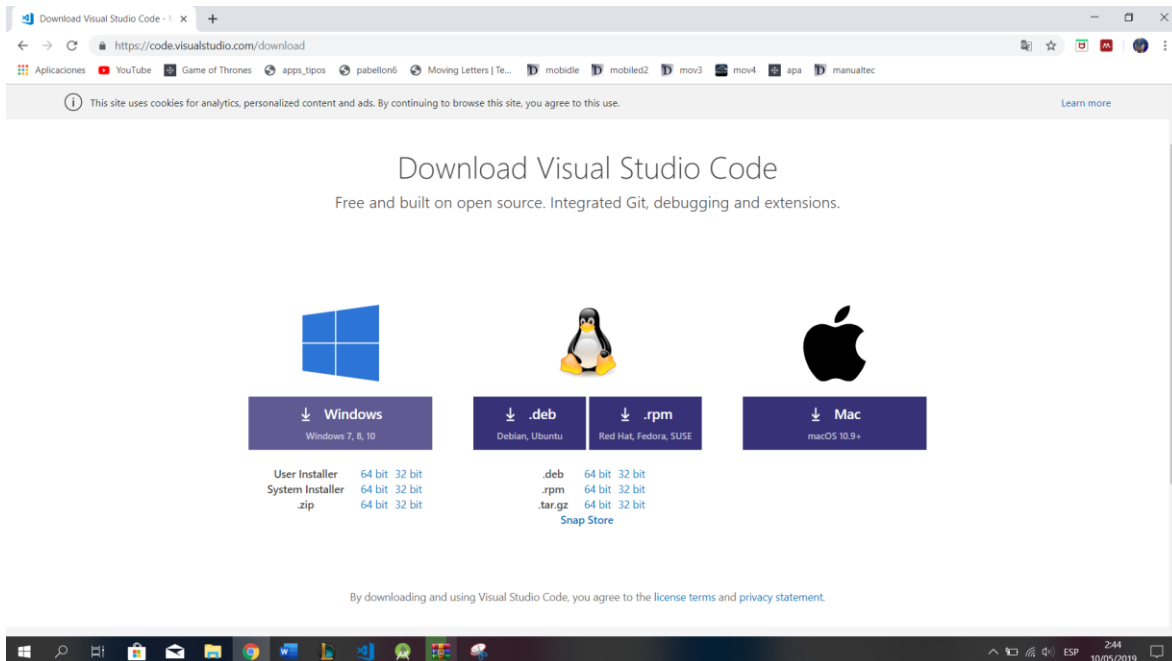


Ilustración 54. Instalación 2.

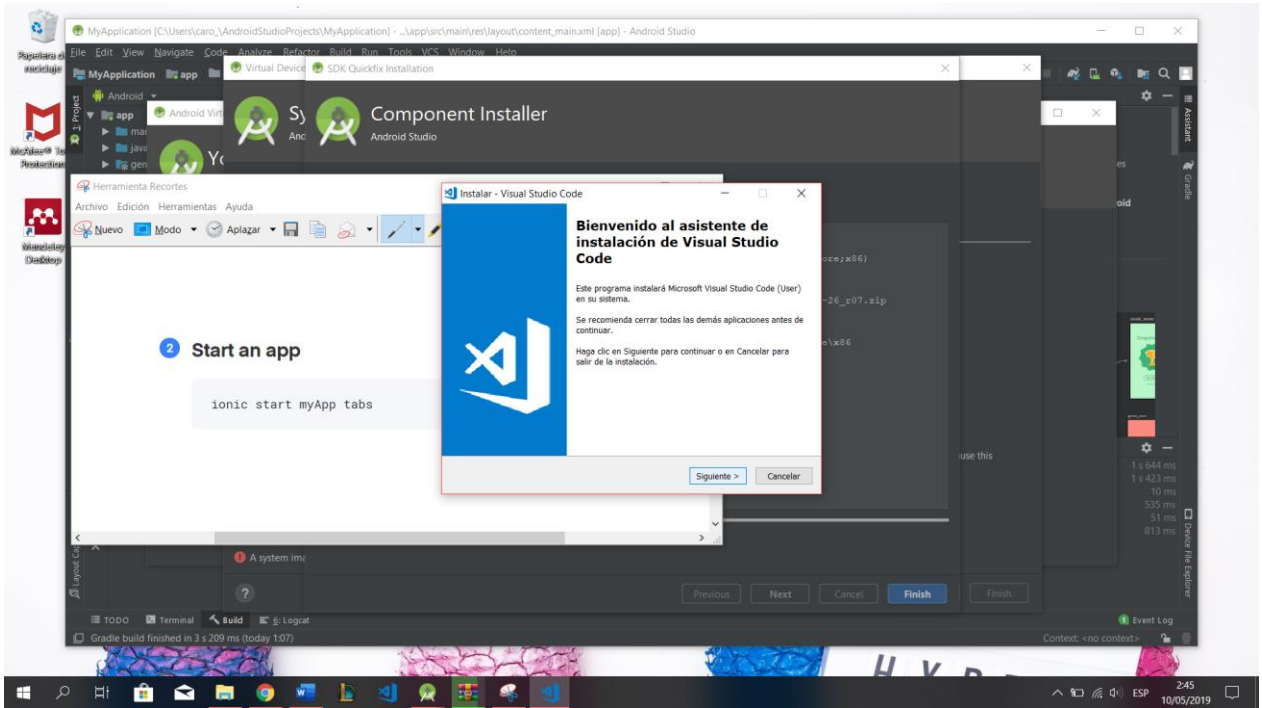


Ilustración 55. Instalación 3.

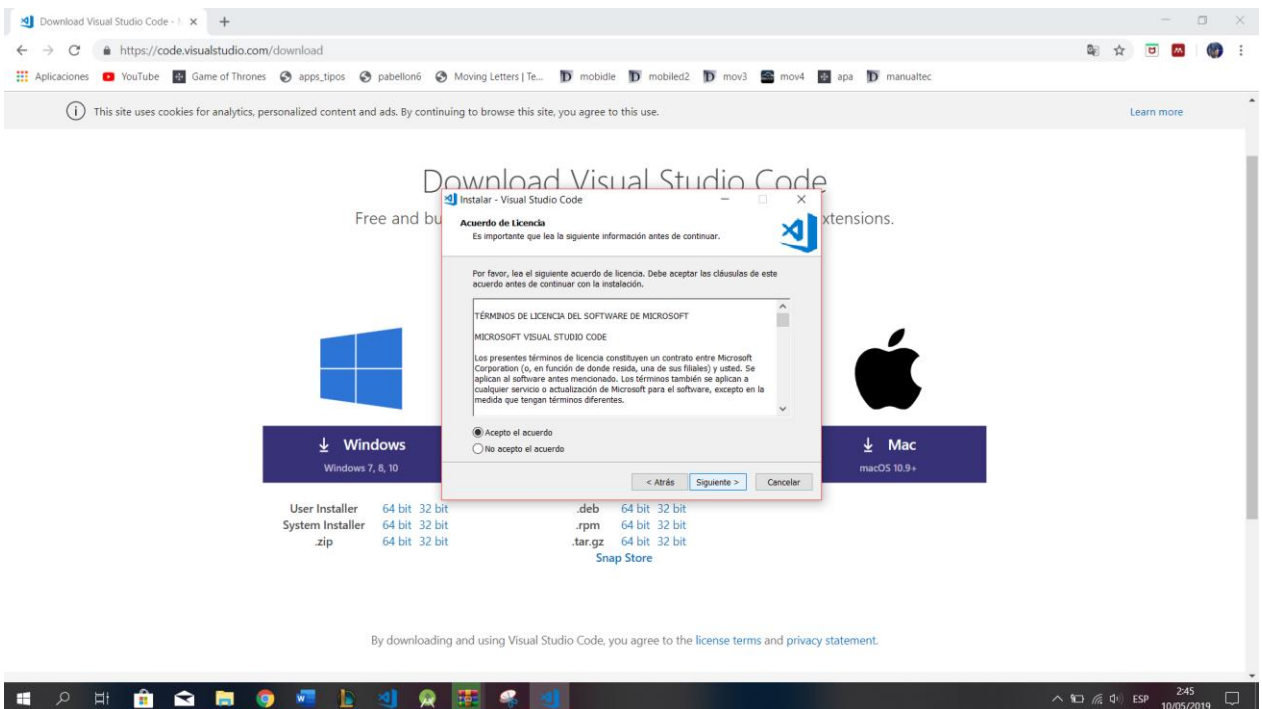


Ilustración 56. Instalación 4.

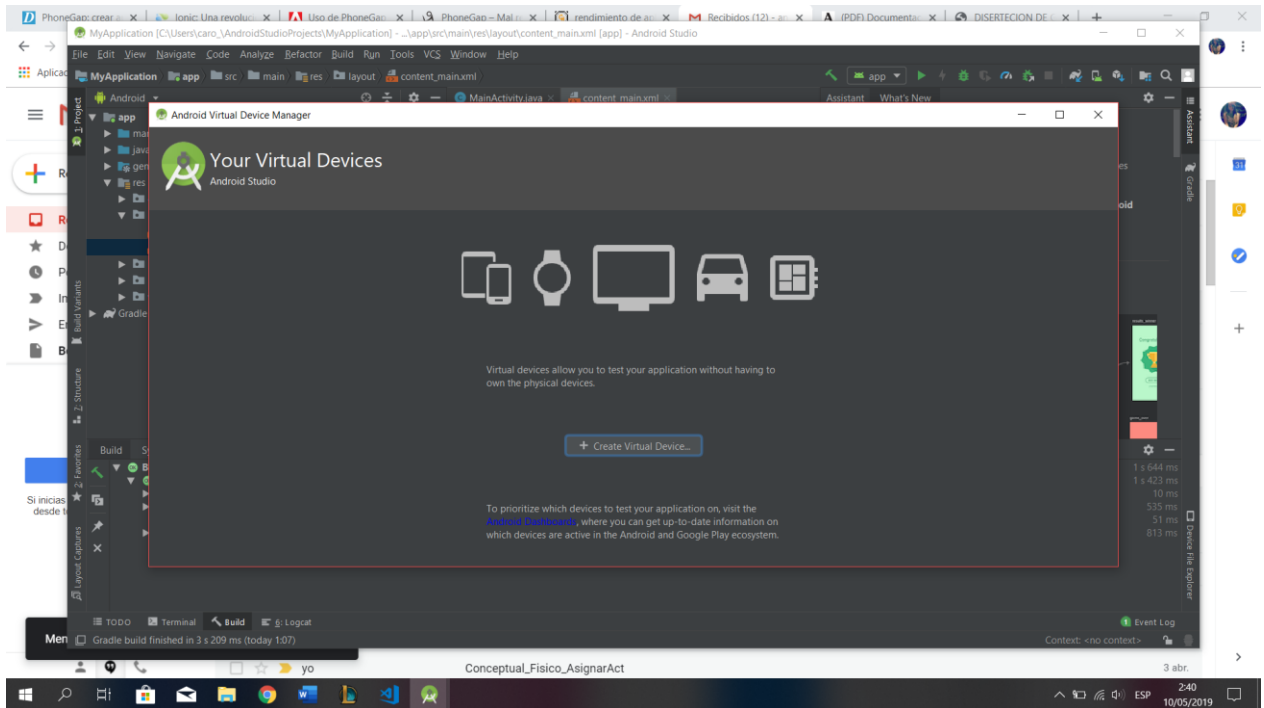


Ilustración 57. Instalación 5.

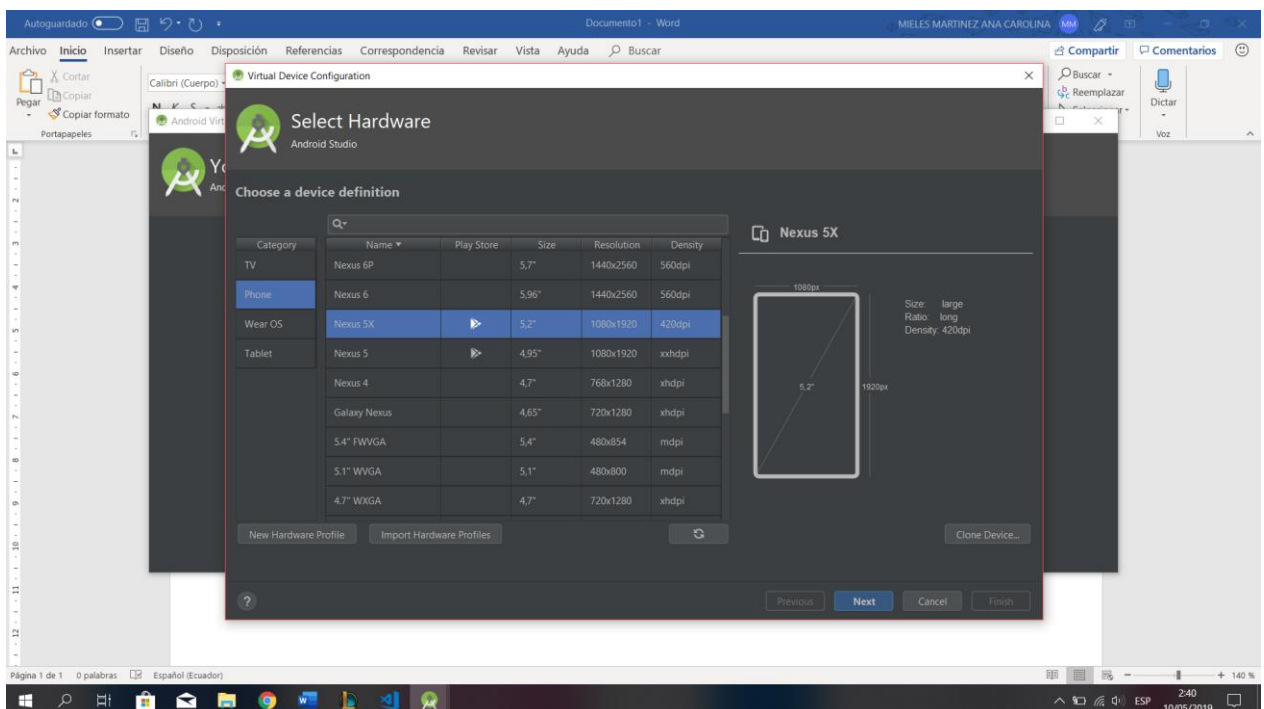


Ilustración 58. Instalación 6.

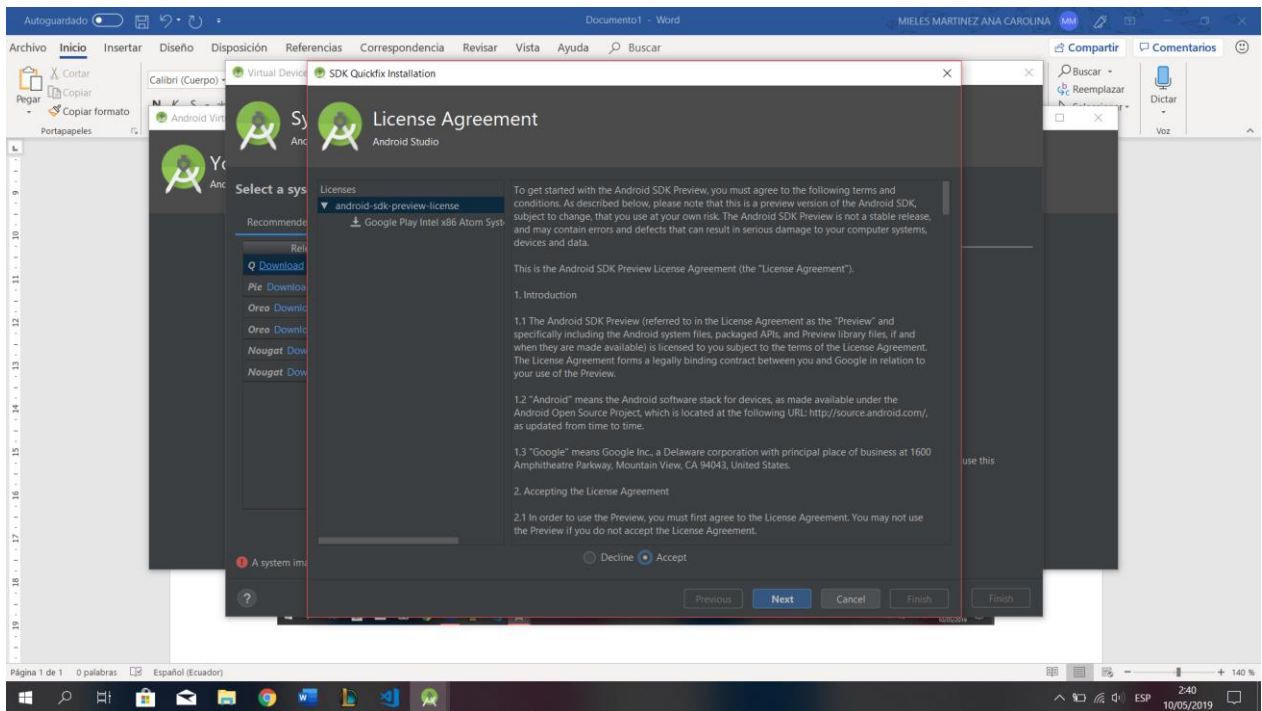


Ilustración 59. Instalación 7.

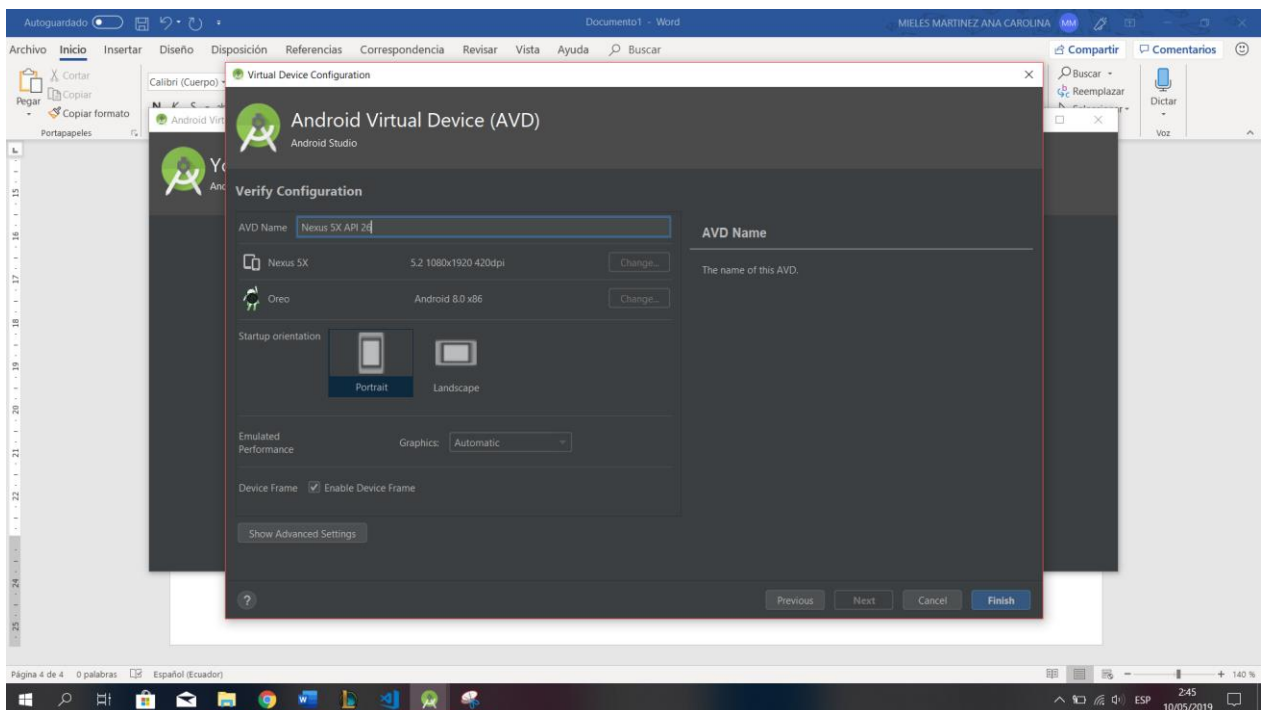


Ilustración 60. Instalación 8.

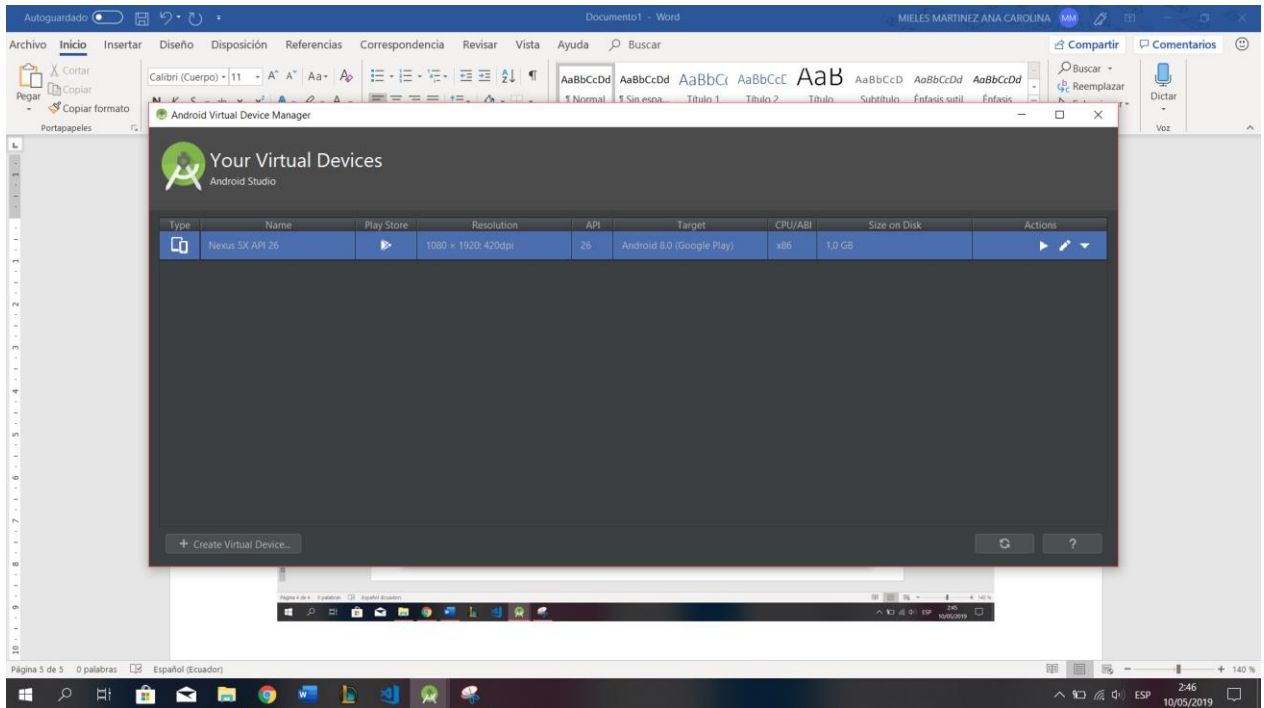


Ilustración 61. Instalación 9.

5. Código Fuente

5.1 Inicio de sesión



Ilustración 62. Inicio de sesión.

```
loginForm: FormGroup;  
  
data: any[]=[];  
  
lista: any[]=[];  
  
public type = 'password';  
  
public showPass = false;
```

```
mensajes_error = {  
  
  'username': [  
  
    {type: 'required', message: 'El usuario es requerido. Porfavor ingrese un usuario  
válido.'},  
  
    {type: 'maxlength', message: 'El usuario debe contener máximo 20 caracteres.'}  
  
  ],  
  
  'password': [  
  
    {type: 'required', message: 'La contraseña es requerida. Porfavor ingrese una  
contraseña válida.'},  
  
    {type: 'maxlength', message: 'La contraseña debe contener máximo 40 caracteres.'}  
  
  ]  
  
}  
  
constructor(  
  
  public FormBuilder: FormBuilder, private web: WebService, private router: Router )  
  
  {  
  
    this.loginForm = this.formBuilder.group({  
  
      username: new FormControl("", Validators.compose([  
  
        Validators.required,  
  
        Validators.maxLength(20)
```

```
   )),  
  
    password: new FormControl("", Validators.compose([  
  
        Validators.required,  
  
        Validators.maxLength(40)  
  
    ]))  
  
});  
  
}  
  
ngOnInit() {  
  
}  
  
autenticar(){  
  
    this.web.autenticar(this.loginForm.value['username'],  
this.loginForm.value['password']).then(data => {  
  
        if(JSON.stringify(data)=="0"){  
  
            console.log("No es usuario");  
  
        }  
  
        else {
```

```
var text = '{"nombre":'+JSON.stringify(data)+'}';

var obj = JSON.parse(text);

for(var i=0; i<(Object.keys(obj.nombre).length); i++) {

this.data.push( {nombre:obj.nombre[i].ID_USUARIO,rol:obj.nombre[i].ROL_USUARIO})

}

this.lista = this.data;

console.log(this.lista[0].nombre);

console.log(this.lista[0].rol);

if (this.lista[0].rol=="COORDINADOR_INGENIERIA")

{this.router.navigateByUrl(`menu/gestion-servicios`);}

if (this.lista[0].rol=="INGENIERIA")

{this.router.navigateByUrl(`menu-ing/lista-act-ing/${this.lista[0].nombre}`);}

}

});

}

pasar(){
```

```
this.autenticar();

//this.router.navigateByUrl('menu/gestion-servicios');

}

showPassword() {

    this.showPass = !this.showPass;

    if(this.showPass){

        this.type = 'text';

    } else {

        this.type = 'password';

    }

}

}
```

5.2 Gestión de servicios

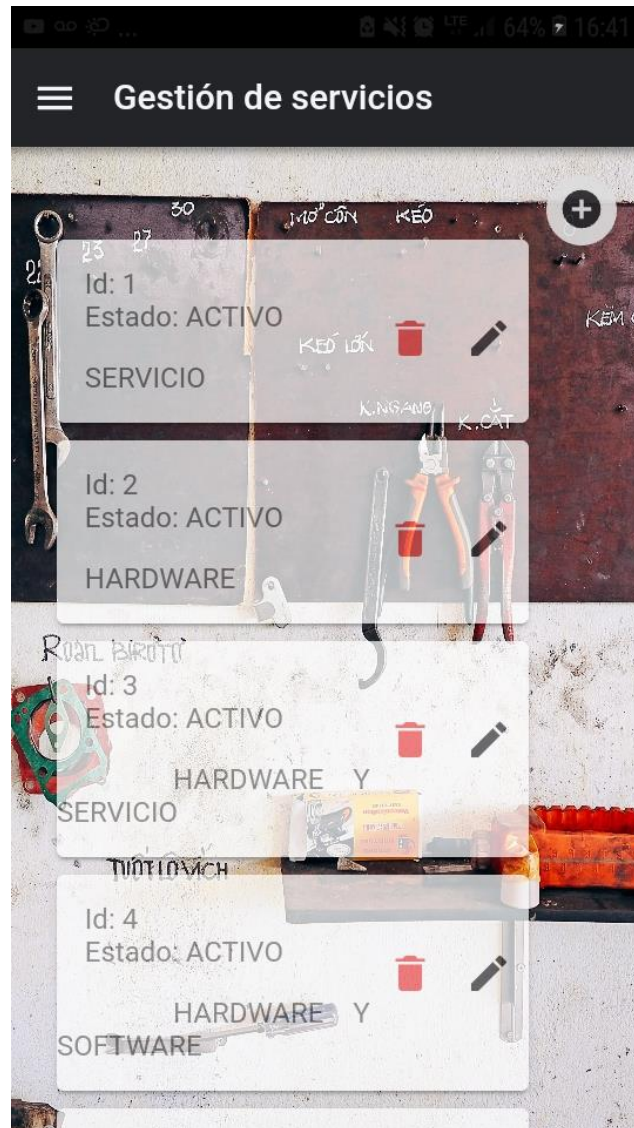


Ilustración 63. Gestión de srrvicios.

soluciones: any = [];

```

    constructor(private activatedRoute: ActivatedRoute, private actionSheetControl:
    ActionSheetController, private web: WebService,private router: Router) {

    }

```

argumento = ";

```
ngOnInit() {  
  
  this.verSoluciones();  
  
}  
  
verSoluciones(){  
  
  this.soluciones.length = 0;  
  
  this.web.verSoluciones().subscribe(res => {  
  
    var obj = JSON.parse(JSON.stringify(res));  
  
    for(var i=0; i<(Object.keys(obj).length); i++) {  
  
this.soluciones.push({id:obj[i].COD_SOLUCION,descr:obj[i].DESCR_SOLUCION,estado:obj[i  
].ESTADO_SOLUCION})  
  
    }  
  
    })  
  
  }  
  
  crearServicio(){  
  
    this.router.navigateByURL('form-servicio');  
  
  }  
  
  async presentarActionSheet(id){
```

```
const actionSheet = await this.actionSheetControl.create({

header: '¿Está seguro de eliminar?',

buttons: [

  {

    text: 'Si',

    role: 'destructive',

    icon: 'trash',

    handler: () => {

      console.log(this.web.eliminarSolucion(id));

      this.router.navigateByUrl(`menu/gestion-servicios`);

      this.ngOnInit();

    }

  },

  {

    text: 'No',

    role: 'cancel',

    handler: () => {

      this.router.navigateByUrl(`menu/gestion-servicios`);
```

```
    }  
  }  
  ] })  
  
  await actionSheet.present();  
  
  }  
  
  editarServicio(id) {  
  
    this.router.navigateByUrl(`editar-servicio/${id}`);  
  
  }  
  
}
```

5.3 Crear servicio

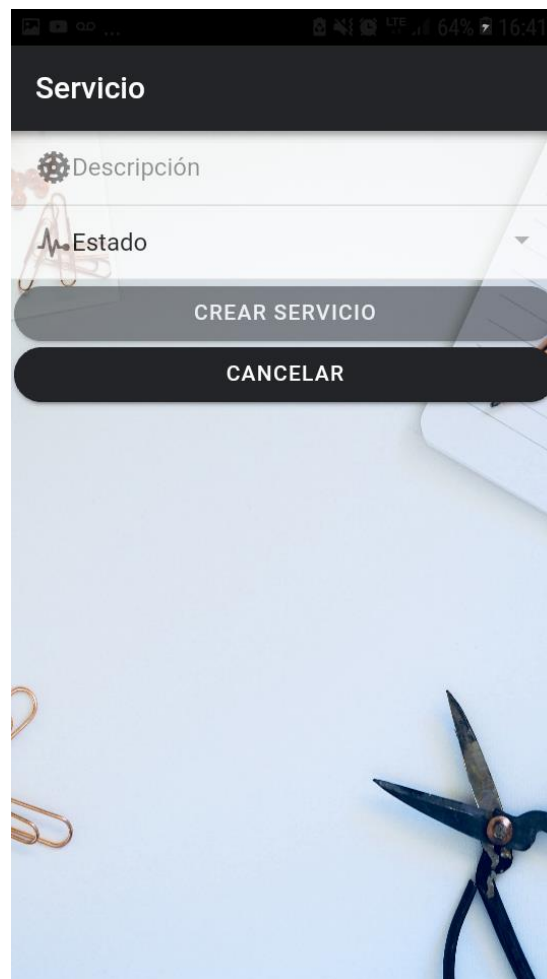


Ilustración 64. Crear servicio.

```
servicioForm: FormGroup;

constructor(public formBuilder: FormBuilder, private web: WebService, private router:
Router) {

    this.servicioForm = this.formBuilder.group({

        descripcion: new FormControl("", Validators.compose([

            Validators.required

        ])),
```

```
estado: new FormControl("", Validators.compose([

    Validators.required

])));

});

}

ngOnInit() {

}

crearServicio(){

this.web.crearSolucion(this.servicioForm.value.descripcion,this.servicioForm.value.estado);

    this.router.navigateByUrl('menu/gestion-servicios');

}

Cancelar(){

    this.router.navigateByUrl('menu/gestion-servicios');

}

}
```

5.4 Editar servicio

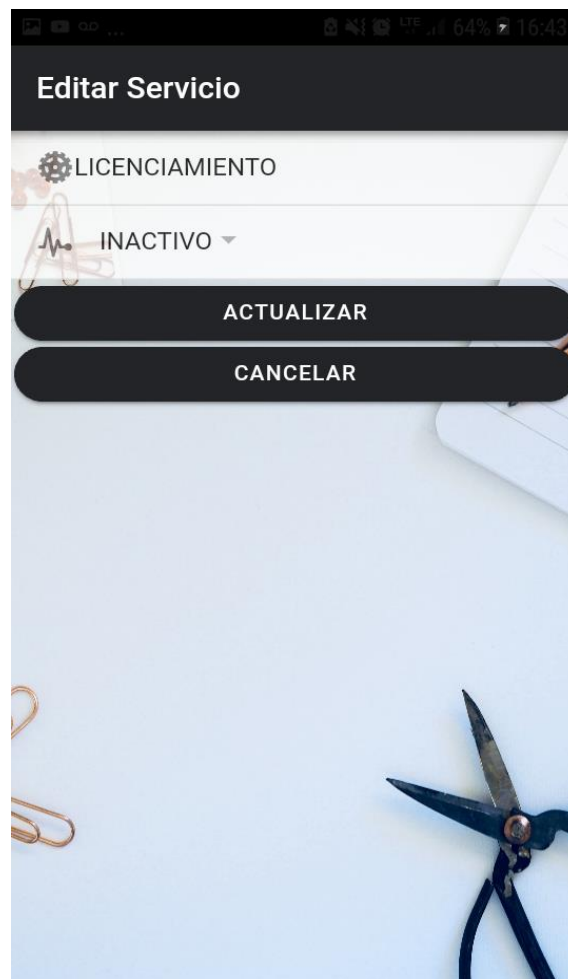


Ilustración 65. Editar servicio.

```
id: any;
```

```
soluciones: any = [];
```

```
constructor(private web: WebService, private activatedRoute: ActivatedRoute, private router:  
Router, private navCtrl: NavController) { }
```

```
ngOnInit() {
```

```
    this.id = this.activatedRoute.snapshot.paramMap.get('id');
```

```
    console.log(this.id);
```

```
    this.verSoluciones(this.id);

}

verSoluciones(id){

    this.web.verSolucionxId(id).subscribe(res => {

        var obj = JSON.parse(JSON.stringify(res));

        for(var i=0; i<(Object.keys(obj).length); i++) {

            this.soluciones.push( {id:obj[i].COD_SOLUCION,descr:obj[i].DESCR_SOLUCION,
estado:obj[i].ESTADO_SOLUCION})

        }

    })

}

Actualizar(d,e){

    if (d == undefined){

        d = this.soluciones[0].descr;

    }

    if (e == undefined){

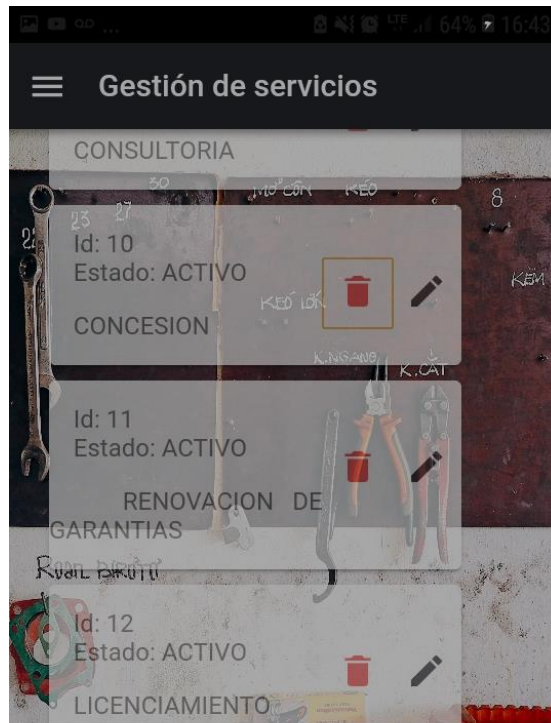
        e = this.soluciones[0].estado;

    }

}
```

```
this.web.editarSolucion(this.id,d,e).subscribe(res => {  
  
    console.log(res);  
  
} ) }  
  
Cancelar(){  
  
    this.router.navigateByUrl('menu/gestion-servicios');  
  
}  
  
}
```

5.5 Eliminar servicio



¿Está seguro de eliminar?

 Si

No

Ilustración 66. Eliminar servicio.

```

async presentarActionSheet(id) {

  const actionSheet = await this.actionSheetControl.create({

    header: '¿Está seguro de eliminar?',

    buttons: [

      {

        text: 'Si',

```

```
    role: 'destructive',

    icon: 'trash',

    handler: () => {

        console.log(this.web.eliminarSolucion(id));

        this.router.navigateByUrl(`menu/gestion-servicios`);

        this.ngOnInit();

    }

},

{

    text: 'No',

    role: 'cancel',

    handler: () => {

        this.router.navigateByUrl(`menu/gestion-servicios`);

    }

}

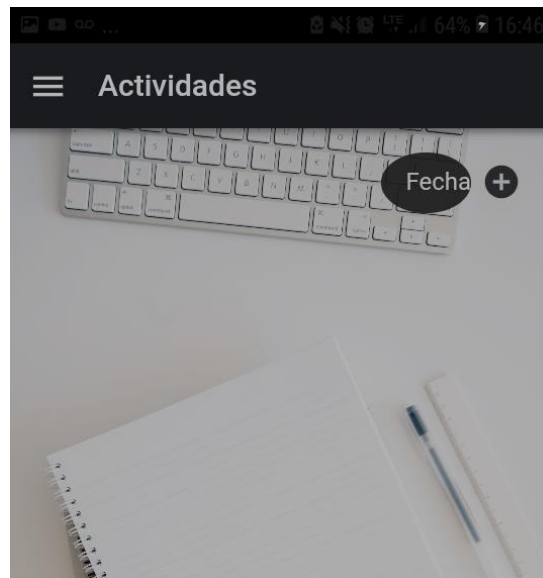
]

})

await actionSheet.present();
```

}

5.6 Agenda



CANCEL DONE

20	Mar	
21	Abr	2020
22	May	2019
23	Jun	
24	Jul	

Ilustración 67. Agenda.

actividadForm: FormGroup;

soluciones: any = [];

clientes: any = [];

usuarios: any = [];

```
offerSelected(selectedValue: any){

    var item = this.soluciones.find(item => item['id'] === selectedValue);

    var position = this.soluciones.findIndex(item => item['id'] === selectedValue);

}

portChange(event: {

    component: IonicSelectableComponent,

    value: any

}) {

    console.log('cliente:', event.value);

}

getData() {

    this.web.verSolucion().then(res => {

        var obj = JSON.parse(JSON.stringify(res));

        for(var i=0; i<(Object.keys(obj).length); i++) {

            this.soluciones.push({id:obj[i].COD_SOLUCION,descr:obj[i].DESCR_SOLUCION})

        }

    })

    this.web.verCliente().subscribe(res => {
```

```
// console.log(res);

var obj = JSON.parse(JSON.stringify(res));

for(var i=0; i<(Object.keys(obj).length); i++) {

this.clientes.push({id:obj[i].COD_CLIENTE,nombre:obj[i].NOMBRE_CLIENTE})

}

})

this.web.verUsuario().then(res => {

var obj = JSON.parse(JSON.stringify(res));

for(var i=0; i<(Object.keys(obj).length); i++) {

this.usuarios.push({id:obj[i].COD_USUARIO,nombre:obj[i].NOMBRE_USUARIO})

}

})

}

cancelar(){

this.router.navigateByUrl('menu/lista-act')

}
```

```
constructor(public formBuilder: FormBuilder,private web: WebService, private router:
Router) {

    this.getData();

    this.actividadForm = this.formBuilder.group( {

        fecha: new FormControl("", Validators.compose([

            Validators.required

        ])),

        hora: new FormControl("", Validators.compose([

            Validators.required

        ])),

        usuario: new FormControl("", Validators.compose([

            Validators.required,

            Validators.maxLength(40)

        ])),

        solucion: new FormControl("", Validators.compose([

            Validators.required,

            Validators.maxLength(20)
```

```
   )),  
  
    cliente: new FormControl("", Validators.compose([  
  
        Validators.required,  
  
        Validators.maxLength(20)  
  
    ])),  
  
    descripcion: new FormControl("", Validators.compose([  
  
        Validators.required,  
  
        Validators.maxLength(200)  
  
    ]))  
  
});  
  
}  
  
ingresarAct(){  
  
    console.log(this.actividadForm.value);  
  
    this.web.crearActividad(this.actividadForm.value.cliente.id,this.actividadForm.value.usuario,this  
    .actividadForm.value.solucion,
```

```
this.actividadForm.value.fecha,this.actividadForm.value.descripcion,this.actividadForm.value.hora);
```

```
    this.router.navigateByUrl(`menu/lista-act`);
```

```
  }
```

```
}
```

5.7 Crear Actividades



Ilustración 68. Crear Actividades.

```
ingresarAct(){
```

```
console.log(this.actividadForm.value);

this.web.crearActividad(this.actividadForm.value.cliente.id,this.actividadForm.value.usuario,this
.actividadForm.value.solucion,

this.actividadForm.value.fecha,this.actividadForm.value.descripcion,this.actividadForm.value.ho
ra);

this.router.navigateByUrl(`menu/lista-act`);

}
```

5.8 Lista de Actividades



Ilustración 69. Lista de Actividades.

```

constructor(private web: WebService,private router: Router) { }

actividades: any = [];

ngOnInit() {

}

fechaSeleccionada(fechaBuscada){

var fechaB = moment(fechaBuscada).format("YYYY-MM-DD");

console.log(fechaB);

```

```
this.actividades.length = 0;

this.verActividades(fechaB);

}

verActividades(fechaBuscada){

this.web.verActividades(fechaBuscada).subscribe(res => {

var obj = JSON.parse(JSON.stringify(res));

for(var i=0; i<(Object.keys(obj).length); i++) {

this.actividades.push({id:obj[i].COD_ACTIVIDAD,cliente:obj[i].COD_CLIENTE,
usuario:obj[i].COD_USUARIO,solucion:obj[i].COD_SOLUCION,

fecha:obj[i].FECHA_ACTIVIDAD,          hora:obj[i].HORA_ACTIVIDAD,
estado:obj[i].ESTADO_ACTIVIDAD,

descripcion:obj[i].DESC_ACTIVIDAD})

}

})

}

crearActividad(){

this.router.navigateByUrl(`/agenda`);

}
```

5.9 Editar Actividad

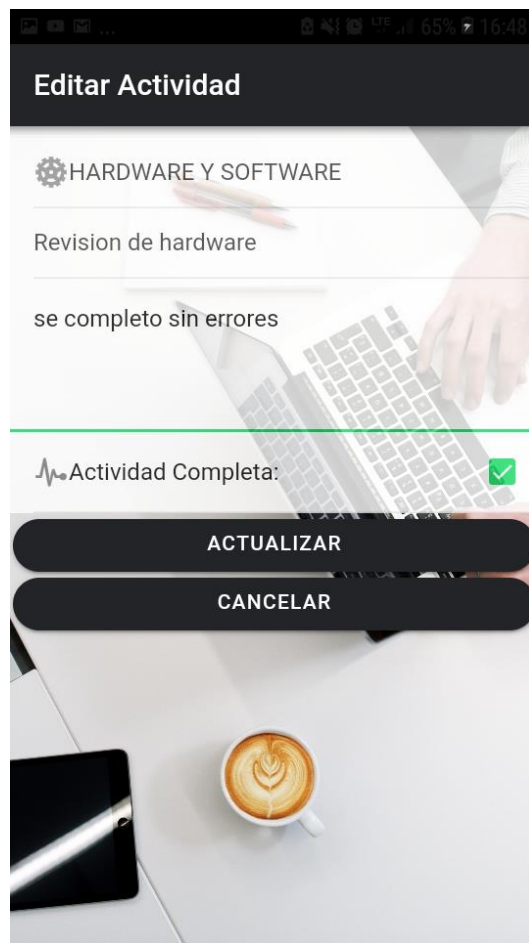


Ilustración 70. Editar Actividad.

```
id: any;
```

```
actividades: any = [];
```

```
constructor(private web: WebService, private activatedRoute: ActivatedRoute, private  
navCtr: NavController) { }
```

```
ngOnInit() {
```

```
    this.id = this.activatedRoute.snapshot.paramMap.get('id');
```

```
    console.log(this.id);
```

```
    this.verActividades(this.id);

}

verActividades(id){

    this.web.verActividadesxId(id).subscribe(res => {

        var obj = JSON.parse(JSON.stringify(res));

        for(var i=0; i<(Object.keys(obj).length); i++) {

this.actividades.push({id:obj[i].COD_ACTIVIDAD,solucion:obj[i].COD_SOLUCION,
descr:obj[i].DESC_ACTIVIDAD,

        obser:obj[i].OBSRV_ACTIVIDAD, est:obj[i].ESTADO_ACTIVIDAD})

        }

        console.log(obj);

    })

}

Actualizar(o,e){

    if(e == true){

        this.web.editarActividad(this.id,o,"REALIZADO").subscribe(res => {

            console.log(res);


```

```
    } ) }  
  
    else  
  
    {  
  
        this.web.editarActividad(this.id,o,"NO REALIZADO").subscribe(res => {  
  
            console.log(res);  
  
        } )  
  
    }  
  
    this.navCtr.pop();  
  
    }  
  
Cancelar(){  
  
    this.navCtr.pop();  
  
    }  
  
    }
```

5.10 Sobre App



Ilustración 71. Sobre App.

```
<ion-header>
```

```
<ion-toolbar color="dark">
```

```
<ion-buttons slot="start" >
```

```
<ion-menu-button color="light" slot="start" >
```

```
</ion-menu-button>
```

```
</ion-buttons>

<ion-title color="light">Sobre App</ion-title>

</ion-toolbar>

</ion-header>

<ion-content >

<ion-card class="item3">

<p class="item2"> &lt;Informacion sobre app&gt;</p>

<ion-label class="item1 ">&nbsp;&lt;Desarrollador&gt;</ion-label>

<ion-label color="light">Carolina Mieles M</ion-label>

<ion-label class="item1 "> &lt;/Desarrollador&gt;</ion-label>

<br>

<ion-label class="item1 ">&nbsp;&lt;Titulo-App&gt;</ion-label>

<ion-label color="light">Control de actividades</ion-label>

<ion-label class="item1 "> &lt;/Titulo-App&gt;</ion-label>

<br>

<br>

<ion-label class="item1 ">&nbsp;&lt;telefono&gt;</ion-label>

<ion-label color="light">(+593)992869421</ion-label>
```

```
<ion-label class="item1"> &lt;/telefono&gt;</ion-label>

<br>

<ion-label class="item1">&nbsp;&lt;email&gt;</ion-label>

<ion-label color="light">anacarolinammz@gmail.com</ion-label>

<ion-label class="item1"> &lt;/email&gt;</ion-label>

<br>

<ion-label class="item1">&nbsp;&lt;fecha&gt;</ion-label>

<ion-label color="light">Marzo 2019</ion-label>

<ion-label class="item1"> &lt;/fecha&gt;</ion-label>

<p class="item2">&lt;/Informacion sobre app&gt;</p>

</ion-card>

</ion-content>
```

CAPÍTULO V: Recomendación

5.1 Conclusiones

- La materia de bases de datos I y II me permitió conocer acerca de las distintas bases de datos y las sentencias por lo que me facilitó la creación y manejo de SQL Server en este aplicativo.
- Al utilizar un *framework* direccionado a aplicaciones híbridas podemos adaptar nuestro aplicativo a los diferentes sistemas operativos móviles basándonos en un solo código.

- La ventaja de crear aplicativos híbridos es que se basan en un solo código y se adaptan a múltiples plataformas, al igual que las aplicaciones nativas una aplicación híbrida puede acceder a los recursos de hardware, la creación de estas es más sencilla y económica.
- El aplicativo creado podrá brindar a la empresa un seguimiento y automatización del proceso de agendamiento en el área de ingeniería.
- Las pruebas unitarias nos ayudan a evaluar los componentes por separado otorgando varias ventajas y facilitando a la integración de módulos.
- Las pruebas son imprescindibles debido a que nos permiten garantizar el correcto funcionamiento de nuestro aplicativo, verificando sus defectos y permitiéndonos corregirlos logrando así asegurar la calidad de nuestro producto de software.

5.2 Recomendaciones

- Sería recomendable que en la carrera se revisen materias que nos enseñen a configurar y utilizar servidores, también materias que instruyan y fomenten el uso de aplicaciones móviles híbridas.
- Utilizar un framework que nos permita visualizar los cambios en tiempo real ya que esto nos brindará una reducción tanto de tiempo como de coste.
- Es importante que la aplicación sea flexible ya que debido a la diferencia de necesidades de un cliente se debe adaptar hacia un beneficio común y de esta manera mantener la permanencia y fidelidad del usuario lo que generará un impulso y posicionamiento para la empresa.
- Si se busca crear un aplicativo móvil para distintas plataformas lo mejor es optar por utilizar frameworks que se concentran en aplicaciones híbridas, que a diferencia de las

nativas no tienen una curva de aprendizaje tan alta y permiten crear aplicaciones multiplataforma.

- Si conocemos que nuestra aplicación estará sometida a cambios recurrentes es necesario optar por una metodología ágil.
- Sería importante que a futuro la empresa opte por utilizar una base de datos en la nube puesto que esto brindará mayor seguridad y portabilidad.

Referencias bibliográficas

Álvarez, A., de las Heras, R., & Lasa, C. (2012). *Métodos Ágiles y Scrum*. Madrid: Anaya.

Apihumhub. (2017). Pruebas unitarias: ventajas y características. Retrieved May 10, 2019, from <https://apiumhub.com/es/tech-blog-barcelona/beneficios-de-las-pruebas-unitarias/>

Baquero, J. (2015). ¿Qué es PhoneGap y para qué podemos utilizarlo? - Blog de arsys.es. Retrieved March 20, 2019, from <https://www.arsys.es/blog/programacion/disenio-web/que-es-phonegap/>

Cuartero, J. A. (2018). America Economía. Obtenido de Metodologías ágiles: <https://www.americaeconomia.com/analisis-opinion/metodologias-agiles-el-nuevo-paradigma-empresarial-que-llega-para-quequedarse>

Cuello, J., & Vittone, J. (2013). *Diseñando apps para móviles*. Argentina.

Facebook Inc. (n.d.). React Native · A framework for building native apps using React. Retrieved March 20, 2019, from <https://facebook.github.io/react-native/>

Grados, J. (2016). ¿Cómo funciona React.js? Retrieved March 20, 2019, from <https://devcode.la/blog/como-funciona-reactjs/>

Klew, W. (2009). Load Impact. Retrieved May 10, 2019, from <http://gizmos.republica.com/programas-y-aplicaciones/load-impact-comprueba-si-tu-sitio-web-puede-soportar-una-alta-demanda-de-trafico.html>

Kofler, M. (n.d.). *MySQL*.

Letelier, P., & Penadés, M. (26 de Junio de 2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Técnica Administrativa*. Obtenido de Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP): <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Margaret Rouse. (2017). ¿Qué es SQL Server? Retrieved March 22, 2019, from <https://searchdatacenter.techtarget.com/es/definicion/SQL-Server>

Masnou, A. (2016). Creando una aplicación con Ionic Creator. Retrieved March 20, 2019, from <https://blog.interactius.com/creando-una-aplicación-con-ionic-creator-7718310131cb>

Microsoft. (2017). Características de SQL Server Management Studio - SQL Server. Retrieved March 22, 2019, from <https://docs.microsoft.com/es-es/sql/database-engine/features-in-sql-server-management-studio?view=sql-server-2014>

Perez, J. (2015). Qué es y cómo empezar con Ionic Framework | PhoneGap Spain. Retrieved March 20, 2019, from <https://www.phonegapspain.com/que-es-y-como-empezar-con-ionic-framework/>

Ramírez Vique, R. (n.d.). *Métodos para el desarrollo de aplicaciones móviles*. Retrieved from [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_4\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf)

Siles, F. (2013). PhoneGap - Framework open source para desarrollo de apps móviles. Retrieved March 20, 2019, from <https://www.genbeta.com/desarrollo/phonegap>

Scrum Alliance. (2019). Obtenido de Learn about Scrum: <https://www.scrumalliance.org/learn-about-scrum>

Sof Teng. (2018). Obtenido de Metodología Scrum: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>

Tumipamba Borja, E. E. (2016). *Desarrollo de una aplicación móvil que permite a los docentes y estudiantes de la universidad central del ecuador acceder a las bases de datos científicas*. Universidad Central del Ecuador. Retrieved from <http://www.dspace.uce.edu.ec/bitstream/25000/5466/1/T-UCE-0011-244.pdf>

WebMaster. (2017). Unitel. Obtenido de ¿Por qué tu empresa necesita una aplicación móvil?: <https://unitel-tc.com/blog/empresa-necesita-una-aplicacion-movil/>

wiliams, jean. (n.d.). Mobile-D. Retrieved from <https://www.academia.edu/23746235/Mobile-D>

Zea Ordóñez, M. P., & Molina Ríos, Jimmy Rolando Redrován Castillo, F. F. (2017). *ADMINISTRACIÓN DE BASES DE DATOS CON POSTGRESQL*. 3Ciencias. Retrieved from https://books.google.com.ec/books?id=5-mkDgAAQBAJ&pg=PA12&dq=postgresql&hl=es&sa=X&ved=0ahUKEwi12Z3F_pThAhUGrIkKHe24CZ4Q6AEIOjAC#v=onepage&q=postgresql&f=false