

Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

**APLICACIÓN MÓVIL DE GEOLOCALIZACIÓN Y
MONITOREO DE GUARDIAS**

ESTEBAN FABIÁN GÓMEZ MONTERO

CRISTIAN OMAR HENRÍQUEZ URIBE

Profesor Guía:

Pamela Hermosilla Monckton

Profesor Co-referente:

Daniela Quiñones Otey

Carrera:

Ingeniería de Ejecución en Informática

Diciembre 2018

Dedicatoria

Este proyecto necesitó de mucho tiempo y dedicación, pero no habría sido posible sin la ayuda de mi compañero de proyecto, que sin su colaboración no podría haberlo realizado. También me gustaría agradecer a todos los que estuvieron cerca de mí en este tiempo, en especial a mi papá cuya presencia fue muy importante en todo este proceso. Además me gustaría dar las gracias a mi mamá y a mi hermana que a pesar de la distancia mantiene su amor incondicional y siempre me ha dado su apoyo y ha estado pendiente de mí sin importar las circunstancias. Finalmente quiero agradecer a mi tío Jaime y mi tío Fernando quienes me dieron su apoyo y preocupación en mis momentos difíciles.

Esteban Fabián Gómez Montero

Agradezco a mi familia y a seres cercanos, por siempre apoyarme en todos los procesos difíciles a lo largo de mi vida, especialmente a mis padres que siempre me entregaron valores, salud, educación y amor, sin ellos no hubiese sido el mismo camino, quizás no tendría los logros que he tenido en mi vida. Además, gracias a todas las personas que aportaron con un granito de arena desde pequeño, en mi formación académica y como ser humano. Finalmente, a Dios por no haber tenido grandes tristezas en mi vida, por darme las fuerzas y ánimo para seguir adelante y siempre guiarme por el buen camino, para terminar este primer y gran paso en mi vida.

Cristian Omar Henríquez Uribe

Índice

Dedicatoria	ii
Índice	iii
Resumen	v
Abstract	v
Lista de Figuras	vi
Lista de Tablas	vii
1 Introducción	1
2 Descripción general de la situación en estudio.	1
2.1 Definición de objetivos	1
2.1.1 Objetivo general	1
2.1.2 Objetivo específicos	2
3 Situación en Estudio	2
3.1 contexto organizacional	2
3.2 Descripción de la situación actual	2
3.3 Identificación de los problemas detectados	3
3.4 Estado del arte del tema	3
4 Solución propuesta	4
4.1 Descripción general de la solución	4
4.2 Modelo de procesos de desarrollo	4
4.3 Arquitectura lógica y física de la solución	5
5. Gestión del proyecto	6
5.1 Recursos requeridos para la solución	6
5.1.1 Recursos humanos	6
5.1.2 Recursos tecnológicos	7
5.1.3 Recursos Físicos	7
5.2 Estudio de factibilidad	7
5.2.1 Factibilidad Económica	7
5.2.1.1 Costos de software y tecnologías a utilizar	7
5.2.1.2 Horas hombres	8
5.2.1.3 Costos fijos	8
5.2.1.4 Total de costos	9
5.2.2 Factibilidad Legal	9
5.2.3 Factibilidad Operacional	9
5.2.4 Factibilidad Técnica	10
5.3 Carta Gantt	10
5.3.1 Carta Gantt general	11
5.4. Gestión de riesgos	11
6 Requerimientos	12
6.1 Requerimientos funcionales	12
6.2 Requerimientos no funcionales	16
	iii

6.3	Análisis de requerimiento	17
6.3.1	Diagrama de Flujo de Datos	18
6.3.2	Modelo de Procesos BPMN	18
6.4	Diseño de aplicación	20
6.5	Prototipos	20
6.6	Motor de base de datos	25
6.7	Modelo Relacional	26
7	Conclusión	27
7.1	Sobre el trabajo realizado	27
7.2	Sobre el trabajo futuro	27
8	Bibliografía	28

Resumen

Este documento contiene todas las etapas del software “Compani”, el análisis, diseño, programación y testing con la metodología Scrum, además se explican los resultados que se obtuvieron con sus respectivas conclusiones. El software consiste en una aplicación móvil Android en donde un administrador puede llevar control y visualización de las rondas que realizan los guardias que utilizan la aplicación dentro de una ubicación determinada. En este informe se especifica todo el contexto de la aplicación, como es el estudio previo que se realizó en torno a la aplicación a desarrollar, la solución según el modelo de negocio que requiere el cliente, a través de diagramas de modelamiento, posteriormente se muestra la planificación a través de una carta Gantt con sus tareas respectivas. Finalmente se presentan prototipos, herramientas y lenguajes a utilizar en la confección del software.

Palabras-claves: seguridad, guardias, geolocalización, monitoreo y aplicación móvil.

Abstract

This document contains every field of the development of "Compani" software, analysis, design, coding, and testing with Scrum methodology, also it explains the results and respective conclusions obtained. This software is an Android mobile application, where an administrator can control and visualize guard routes in determined places, where every guard has the app installed. This report specifies all the application context and how was the needed study to start coding. Making the solution to be in line with the client desired business model, with according diagrams. The development process is shown in a Gantt chart with the tasks and respective times. Finally, this report shows the prototypes, tools, and languages used to code this software.

Keywords: Security, Guard management, Geolocalization, Monitoring, Mobile App.

Lista de Figuras

Figura 4.1 Arquitectura Lógica.....	5
Figura 4.2 Arquitectura Física.....	5
Figura 5.1 Carta Gantt General.....	11
Figura 6.1 DFD de Contexto.....	18
Figura 6.2 BPMN Iniciar Sesión.....	19
Figura 6.3 BPMN Agregar Administradores.....	19
Figura 6.4 Pantalla Inicio de Sesión.....	20
Figura 6.5 Pantalla Home Guardia.....	21
Figura 6.6 Pantalla Ubicación Actual Guardia.....	21
Figura 6.7 Pantalla iniciar ronda.....	22
Figura 6.8 Home admin.....	23
Figura 6.9 Ver y asignar ronda.....	23
Figura 6.10 Gestionar comunidad.....	24
Figura 6.11 Gestionar administradores de comunidad.....	25
Figura 6.12 Modelo Relacional.....	26

Lista de Tablas

Tabla 5.1 Costos de software y tecnologías a utilizar.....	7
Tabla 5.2 Horas hombres.....	8
Tabla 5.3 Costos fijos.....	8
Tabla 5.4 Total de costos.....	9
Tabla 5.5 Gestión Riesgo 1.....	11
Tabla 5.6 Gestión Riesgo 2.....	12
Tabla 6.1 Requerimientos Funcionales Login.....	13
Tabla 6.2 Requerimientos Funcionales Administrador Sistema.....	13
Tabla 6.3 Requerimientos Funcionales Administrador.....	14
Tabla 6.4 Requerimientos Funcionales Guardia.....	15
Tabla 6.5 Requerimientos No Funcionales.....	16

1 Introducción

Hoy en día a lo largo de nuestro país la seguridad, es un problema trascendental el cual ha sido tema para todos chilenos, por lo que un cliente nos contactó porque tenía la necesidad de controlar sus guardias, ya que ellos nos estaban realizando las rondas correspondientes, aumentando la cantidad de robos en las instalaciones en donde se está construyendo.

Se aceptó la propuesta del cliente, por lo que en este presente informe se mostrará todo lo correspondiente al proyecto, sobre el estudio, análisis, desarrollo e implementación y resultados con sus respectivas conclusiones de la aplicación móvil “Compani”. Es decir este documento tiene como finalidad informar sobre todo el contexto de la aplicación, las herramientas que se utilizaron, los prototipos, dispositivos, APIs, base de datos, servicios etc. El software “Compani” intenta dar solución al problema de control y administración de rondas de guardias dentro de un recinto, a través de una aplicación que utiliza geolocalización y otras características que serán detalladas en el transcurso de este informe.

2 Descripción general de la situación en estudio.

Esta idea nació para solucionar el problema de una empresa de seguridad, llamada “Compani” la cual desea monitorear y gestionar sus guardias. A pedido del dueño de la compañía, se solicitó diseñar y desarrollar una plataforma móvil que permita supervisar, a través de un dispositivo móvil y su sistema de geolocalización, el desempeño de los guardias en cuanto al cumplimiento de las rondas y las labores asignadas considerando la fecha, hora y tiempos en que estas deben ser realizadas. El sistema, a su vez, debe entregar estadísticas y reportes que permitan al administrador de la compañía visualizar con mayor claridad que el guardia cumple de mejor manera sus tareas a través de información como puntualidad, rapidez en que realiza sus labores, nivel de respuesta ante alguna eventualidad, etc.

2.1 Definición de objetivos

A continuación se presenta el objetivo general y los objetivos específicos del proyecto.

2.1.1 Objetivo general

Desarrollar una aplicación para dispositivos móviles de plataforma Android, que permita automatizar la gestión y monitoreo de guardias de seguridad según sus rondas de vigilancia.

2.1.2 Objetivo específicos

Los objetivos específicos son los siguientes:

- Investigar sobre herramientas útiles para el desarrollo de aplicaciones móviles que utilicen geolocalización.
- Establecer una solución práctica sobre la problemática definida acorde al modelo del negocio.
- Definir metodología a utilizar en el proceso de desarrollo.
- Planificar tareas a realizar durante la duración del proyecto.
- Utilizar servicio de Firebase Cloud Messaging para enviar notificaciones push.
- Integrar API de geolocalización de google maps para obtener posiciones.
- Desarrollar pruebas en tiempo real con geolocalización en dispositivos móviles.

3 Situación en Estudio

A continuación, se presenta el contexto organizacional, descripción de la situación actual, identificación de los problemas detectados y estado del arte.

3.1 contexto organizacional

El contexto en el cual será utilizada la aplicación móvil por parte del guardia será en su horario de trabajo, especialmente durante la realización de sus rondas dentro del recinto al cual pertenece. En cambio, por parte de los administradores y super administrador, la aplicación móvil puede ser utilizada durante todo el día, ya sea dentro de su horario de trabajo o fuera de éste.

3.2 Descripción de la situación actual

Compani es una entidad de seguridad la cual está constituida [1], en el área gerencial por un Staff de Profesionales Infantes de Marina en retiro. Se dedica al servicio de evaluar riesgos por medio de análisis de vulnerabilidades de instalaciones, con el objetivo de determinar las deficiencias de seguridad, después de todo el análisis entregan una propuesta, con una solución de cada agresión y amenaza que tienen los clientes. La empresa tiene guardias de seguridad a los cuales se le otorgan diferentes, comunidades y turnos para trabajar.

El problema actual que tiene la empresa, es que no tiene control sobre sus guardias en las rondas de vigilancia y en la gestión de ellos en donde deben trabajar. Por lo que el cliente ha solicitado realizar una aplicación móvil para Android, en donde se pueda automatizar la gestión y monitoreo de sus guardias, con el fin de facilitar su trabajo diariamente. Actualmente este proceso lo llevan a cabo de manera manual, lo cual es un proceso engorroso, si consideramos que se quiere tener un control de muchos guardias, trabajando en demasiadas comunidades simultáneamente.

A continuación, se detallan los procesos involucrados. La empresa administra sus comunidades y guardias en carpetas, además tienen un archivo Excel, en donde registran los turnos en donde trabaja cada guardia, en una comunidad determinada. La planilla Excel tiene las siguientes columnas; nombre, tipo de turno, comunidad y fecha. Esta planilla es llenada a mano diariamente, los días de trabajo. Por lo tanto, el administrador de la empresa, consulta la planilla, para tener control sobre las comunidades y los guardias.

3.3 Identificación de los problemas detectados

Los problemas detectados son los siguientes:

1. La información de los guardias y comunidades se encuentran almacenadas en carpetas físicas y no en una base de datos virtual, donde sería más fácil agrupar y consultar la información.
2. Al contener la información de los registros de las comunidades, turnos y guardias en una planilla Excel, se hace difícil y toma demasiado tiempo tomar decisiones laborales, especialmente al momento de solicitar reportes y estadísticas.
3. El monitoreo a los guardias se realiza a través de llamados o mensajes de Whatsapp, esto es muy lento porque no es un proceso automatizado, además teniendo en cuenta que son bastantes guardias y comunidades.
4. El cliente pierde mucho tiempo llenando los campos de la planilla Excel diariamente, además, al hacerlo manualmente se corre el riesgo de cometer errores al momento de ingresar la información.

3.4 Estado del arte del tema

Si bien ya existe una investigación directa de una aplicación móvil para Android de geolocalización y monitoreo de guardias, pero se concluyó que es un concepto más general, es decir, que abarca a todas las aplicaciones web o móviles que utilizan geolocalización junto con monitoreo, por lo cual investigamos el estado del arte de las aplicaciones más utilizadas de dicho tema.

¿Por qué son necesarios estos tipos de software? Durante las últimas décadas, la industria ha sido atacada por numerosos problemas de seguridad y gestión tales como robos, miedo, inseguridad, mala gestión, monitoreo del personal, etc. Como resultado se produce un descontrol en la empresa por lo que se generan grandes pérdidas económicas. Para dar solución a esta problemática existe el software de geolocalización y monitoreo del personal.

La industria de desarrollo de software ha recibido mucha ayuda, con una gran cantidad de herramientas muy útiles como, nuevos lenguajes de programación, framework, apis, nuevas tecnologías y especialidades, máquinas más avanzadas y sofisticadas, etc. Una cuestión clave es ¿Cómo pueden las herramientas, tecnologías y personas trabajar juntos con el fin de lograr estos objetivos cada vez más desafiantes? Posibles respuestas a esta pregunta implican cambios en el proceso de desarrollo de software, los cuales, a su vez, requerirán una cantidad significativa de recursos para implementar. Actualmente el software de geolocalización y monitoreo se está utilizando en diversas áreas como la seguridad, delivery de comida, tráfico en las ciudades etc.

Softwares existentes más utilizados de geolocalización y monitoreo:

- Guarnic: Es una plataforma web y móvil para IOS y Android [2], que le permite gestionar equipos de guardias de cualquier tamaño. No importa si es una compañía que ofrece o contrata guardias de seguridad, Guarnic intenta reducir sus costos y aumentar la eficacia de sus operaciones. El problema de esta aplicación, se basa principalmente en que su diseño es poco atractivo para los usuarios, esto provoca que los potenciales usuarios no sientan motivación de adquirir o utilizar la plataforma.

- Uber: Esta es una aplicación móvil para Android e Ios [3], que conecta los pasajeros con los conductores de vehículos registrados en su servicio, los cuales ofrecen un servicio

de transporte a particulares. Esta aplicación muestra en pantalla la ubicación en tiempo real del vehículo, lo cual permite un monitoreo de la ubicación de este por parte del pasajero.

4 Solución propuesta

A continuación se presenta la información general de la solución propuesta, desde una descripción general de ésta, hasta las herramientas que se utilizaron, y la arquitectura lógica y física correspondiente.

4.1 Descripción general de la solución

La solución que se propuso como equipo, fue desarrollar una aplicación móvil para ANDROID, la cual tiene una base de datos en “Firebase” llamada “Realtime Database” la cual es eficiente de baja latencia para las aplicaciones móviles, que necesitan estados sincronizados entre los clientes en tiempo real [4]. Esta base de datos almacena toda la información necesaria, para cumplir con la necesidad del cliente, teniendo tres perfiles de usuario: super administrador, administrador y guardia. El súper administrador tiene permisos para gestionar comunidades y administradores, el administrador tiene permisos para gestionar sus guardias, rondas y comunidades, finalmente el guardia tiene permisos para ver las comunidades en donde ha sido asignado, con sus respectivas rondas determinadas, estas rondas las puede realizar. Finalmente la aplicación permite comunicar al guardia con el administrador través de notificaciones push.

Las funcionalidades más importantes las tendrá el administrador, éste podrá obtener reportes y estadísticas de sus guardias según las rondas, además podrá revisar un ranking con los guardias más atentos según los tiempos de respuesta de ellos al enviar su ubicación actual. Estas funcionalidades no se encuentran desarrolladas, posteriormente se explica detalladamente.

4.2 Modelo de procesos de desarrollo

Para llevar a cabo lo anteriormente mencionado, se hizo uso de la metodología ágil de desarrollo de software Scrum [5], la cual fue seleccionada por una investigación realizada y se concluyó que era la más adecuada ya que cada iteración tiene su revisión del trabajo realizado, ya sea internamente como equipo y externamente con el cliente. Esta metodología consiste en realizar sprint (Incrementos) con el fin de que los equipos, funcionan como unidades integradas y ajustadas con cada miembro del equipo jugando un papel bien definido y todo el equipo se centra en un solo rol. En el equipo de desarrollo, cada miembro del equipo debe entender su trabajo y las tareas para cada incremento, todo el equipo debe tener un solo foco, en este caso realizar la aplicación móvil de geolocalización y monitoreo de guardias. Además las prioridades deben ser claras, como describimos recientemente, el desarrollo de Scrum es un proceso que facilita el enfoque de equipo.

4.3 Arquitectura lógica y física de la solución

A continuación, se presenta la arquitectura lógica de la solución de la aplicación móvil Compani.

ESTRUCTURA LÓGICA

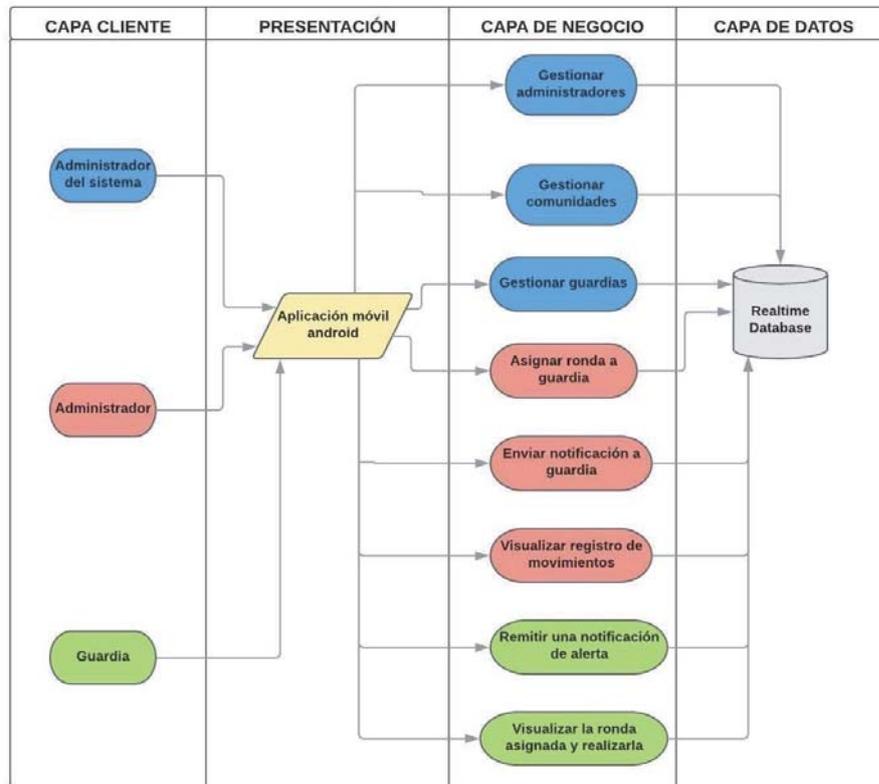


Figura 4.1 Arquitectura Lógica

Seguidamente, se presenta la arquitectura física de la solución de la aplicación móvil Compani.

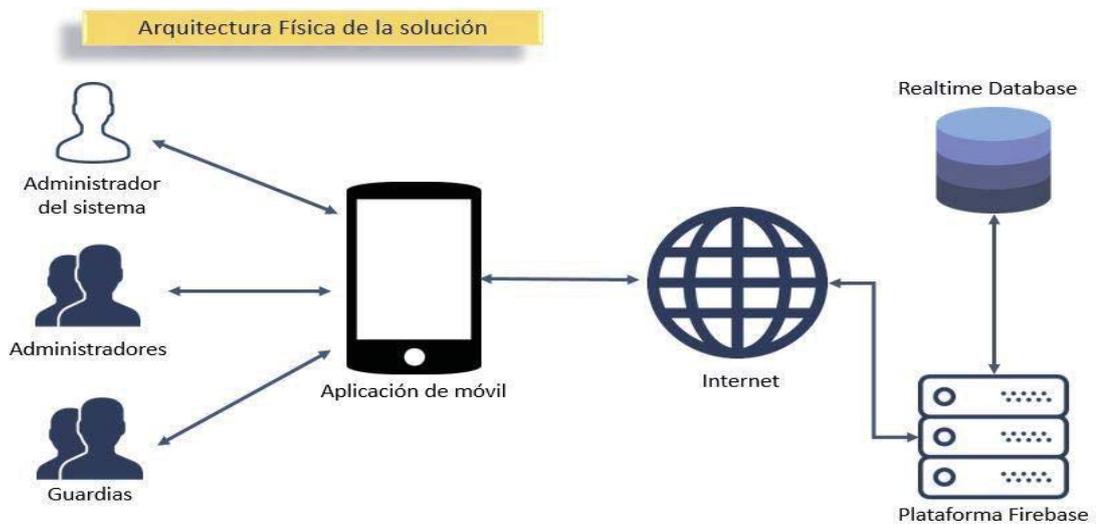


Figura 4.2 Arquitectura Física

4.4 Herramientas de desarrollo

Las herramientas de desarrollo de software, son programas, frameworks, entornos, APIs, plugins, plataformas, etc. Estos son de gran utilidad a la hora de desarrollar la aplicación móvil de manera más segura, ágil, sencilla, eficiente, eficaz, confiable, etc. A continuación se detallan las herramientas que utilizamos para el desarrollo de la aplicación, con sus correspondientes razones.

El entorno de tiempo de ejecución multiplataforma que estamos usando es Node.js [6], ya que este fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables. El framework que estamos utilizando es “Ionic”, ya que es un kit de desarrollo de software completo (SDK) [7] para realizar funcionalidades avanzadas con sus respectivos plugins, este está basado en angular por lo que sirve para desarrollar aplicaciones híbridas con tecnologías web como CSS, HTML5 Y Sass. Además las aplicaciones pueden compilarse con estas tecnologías web y luego distribuirse a través de tiendas de aplicaciones nativas, para ser instalado en dispositivos aprovechando Cordova. Por último también estamos empleando “Firebase” ya que es una plataforma para el desarrollo de aplicaciones web y móviles, la cual contiene servicios muy útiles, seguros y confiables. Como “Firebase Cloud Messaging” [8] que es una plataforma para mensajes y notificaciones para Android, IOS y aplicaciones web que actualmente puede ser usada de forma gratuita, también está “Firebase Auth” el cual es un servicio que puede autenticar los usuarios utilizando únicamente código del lado del cliente. Incluye la autenticación mediante Facebook, GitHub, Twitter y Google. Además, tiene un sistema de administración del usuario por el cual los desarrolladores pueden habilitar la autenticación de usuarios con email y contraseña que se almacenarán en Firebase. Existen otros servicios muy completos como “Realtime Database”, “Firebase Storage”, “Firebase Firestore”, etc.

Todas estas herramientas fueron elegidas porque son de código abierto, es decir tienen muy buena documentación que se encuentra en internet, en donde podemos tener soporte en caso de tener problema y dudas significantes.

5. Gestión del proyecto

A continuación, se presenta toda la información necesaria para la gestión del proyecto, señalando aspectos como recursos, estudio de factibilidad, carta gantt y gestión de riesgo.

5.1 Recursos requeridos para la solución

Seguidamente, se presentan los recursos humanos, tecnológicos y físicos para desarrollar la solución

5.1.1 Recursos humanos

Para llevar a cabo la solución del problema contamos con 2 personas que trabajan en todas las tareas del desarrollo del software, como analista encargado de capturar, analizar, especificar, validar y gestionar los requerimientos del cliente, de diseñador encargado de modelar una solución, como testing encargado de realizar pruebas de software y finalmente de jefe de proyecto el cual se encarga de contactar con el cliente y asegurar la calidad de la aplicación móvil.

5.1.2 Recursos tecnológicos

Los recursos tecnológicos del proyecto fueron 2 notebook en donde se investigó, específico, analizó, documento, codificó y se probó todo esto fue necesario para crear la aplicación móvil, además en los notebook se ejecutaron los distintos programas (node.js, ionic, firebase, bootstrap) que fueron útiles para llevar a cabo la aplicación. También se tuvo que contar con un dispositivo móvil Android, para poder correr la aplicación y ver si existen problemas mientras se está ejecutando. Finalmente se contó con la plataforma Firebase la cual permite alojar la base de datos de la aplicación, en un servicio llamado “Realtime Database” [9].

5.1.3 Recursos Físicos

Para lograr crear la aplicación móvil se necesita que los 2 integrantes del proyecto, tuvieran una conexión de internet estable así obtuvieron una comunicación fluida entre ellos y pudieron investigar lo necesario para llevar a cabo el proyecto.

5.2 Estudio de factibilidad

Para todo proyecto informático, hay que tener en cuenta el alcance que el sistema pueda conllevar, es necesario realizar un estudio de factibilidad para determinar la capacidad técnica, operacional, económica y legal. Este estudio fue realizado para este sistema de control y monitoreo de guardias llamado “Compani”, permitirá determinar todos los costos involucrados al proyecto.

5.2.1 Factibilidad Económica

Para realizar el estudio de factibilidad económica del presente proyecto, es necesario detallar todas las herramientas y tecnologías ocupadas en este proyecto, tanto en lo que se refiere a la programación neta, la elaboración de informes, planificación e incluso hardware utilizado.

5.2.1.1 Costos de software y tecnologías a utilizar

Para el desarrollo del sistema y el cumplimiento de las definiciones formales del sistema de gestión y monitoreo de guardias, las tecnologías utilizadas fueron los sistemas de ofimática, tales como: Microsoft Word, las de diseño y planificación como: Microsoft Visio y las netamente de diseño como Lucidchart [10] y Whimsical Wireframes, entre otras que serán descritas más adelante en la factibilidad técnica.

Finalmente, con toda la información presentada anteriormente los costos de software son los que se muestran a continuación. Cabe destacar que el valor del dólar se encuentra en \$661 [11] pesos chilenos.

Tabla 5.1 Costos de software y tecnologías a utilizar

Software y tecnologías	Monto
Microsoft Office Hogar 2016	\$17.097
Microsoft Visio 2016	\$419.999
PhpStorm	\$39.432
Total	\$476.528

5.2.1.2 Horas hombres

El desarrollo del sistema será por parte de dos estudiantes de la escuela de ingeniería en informática de la Pontificia Universidad Católica de Valparaíso, por lo que el costo de los desarrolladores fue de \$528.000 mensuales, trabajando jornada completa. En la siguiente tabla se muestra un desglose más específico del valor por cada uno de los desarrolladores del presente proyecto:

Tabla 5.2 Horas hombres

Nombre de trabajador	Descripción	Monto
Esteban Gómez Montero	8x20x3.000=\$480.000 mensualmente. Valor Hora Hombre: \$3.000 Horas Diarias Trabajadas: 8hr. Meses de trabajo: 3.	\$1.140.000
Cristian Henríquez Uribe	8x20x3.000=\$480.000 mensualmente. Valor Hora Hombre: \$3.000 Horas Diarias Trabajadas: 8hr. Meses de trabajo: 3.	\$1.140.000

Total \$2.240.000

5.2.1.3 Costos fijos

Durante el desarrollo del presente proyecto de gestión y monitoreo de guardias, se calcularon los gastos en diferentes actividades, tales como:

- Luz de los artefactos utilizados (Notebooks, etc.)
- Internet

En la siguiente tabla que se mostrará a continuación se detallaran los costos fijos más considerables para el desarrollo de este proyecto.

Tabla 5.3 Costos fijos

Nombre de costo	Descripción	Monto
Pack Mega 200 de VTR	Velocidad máxima de bajada de 200 Mbps Velocidad máxima de subida de 7 Mbps Con Nuevo Nextgen WiFi \$25.990 X 3 meses	\$77.970
Luz mensual	Se considera para este costo el uso de los computadores de los alumnos para el desarrollo del sistema.	\$8.000 (aprox)

Total \$85.970

5.2.1.4 Total de costos

Algunos de los costos mencionados anteriormente son contados solamente 1 vez durante el proyecto (por ejemplo Microsoft Visio) y otros se cubrieron durante los 3 meses de duración del proyecto. A continuación se calcula el costo total del proyecto realizando la suma de los 3 costos.

Tabla 5.4 Total de costos

Costos involucrados	Monto
Costo software y tecnología.	\$476.528
Horas hombre	\$2.240.000
Costos fijos	\$85.970
Total	\$2.802.498

Con el estudio anterior se ha podido deducir el precio final del proyecto, corresponde a \$2.802.498 pesos chilenos, considerando tanto los materiales de trabajo necesarios, como las Horas Hombre de cada desarrollador invertidas en este.

5.2.2 Factibilidad Legal

La factibilidad legal es muy importante en el desarrollo y mantención del software, es por esto que la ley N°19.223 [\[12\]](#)(TIPIFICA FIGURAS PENALES RELATIVAS A LA INFORMÁTICA) y la ley N°17.336 (PROPIEDAD INTELECTUAL; DERECHOS DE AUTOR) son pilares fundamentales en la construcción del software y para la protección de este.

Se han observado diversos artículos que componen a diversas leyes que cuentan con una gran relevancia en este software. La ley N°19.223 por ejemplo, cuenta con el artículo II el cual especifica el no uso indebido de la información y el artículo IV especifica la no difusión de la información, es por esto que toda información recolectada por el software, será almacenada y sólo presentada al cliente (Encargado del establecimiento). Además se cuenta con los conocimientos respecto a la ley N°17.336 y su artículo VI sección 9, que deja protegido el software con el respaldo del proyecto y la sección 16 que nos habla del respaldo por programas computacionales.

Se ha decidido implementar la norma ISO 27001 (protección de la confidencialidad, integridad y disponibilidad de la información), con un estudio de riesgos y se verá implementado un futuro plan de mitigación para estos para disminuir los riesgos del mal uso de la información personal de las personas que hagan uso del sistema.

5.2.3 Factibilidad Operacional

Desde el punto de vista operativo, el impacto del presente sistema de monitoreo y gestión de guardias será innegable y sin grandes inconvenientes debido a los siguientes ítems:

- La problemática es presentada por un cliente real que ya tiene un mercado establecido en el tema de seguridad y que desea ampliar el alcance de su empresa permitiendo el desarrollo de la presente aplicación móvil. Además, el cliente pretende mejorar su actual manera de manejar la información de su empresa la cual, como se mencionó anteriormente, carece de agilidad y rapidez al momento de ingresar datos y realizar consultar. Por lo tanto, el sistema a exponer resuelve esta problemática y fija un punto de partida de la necesidad planteada por el cliente.

- El sistema será de fácil acceso, solo requerirá que los guardias y los administradores tengan un dispositivo móvil con la aplicación instalada, descargada desde Play store. Además, al estar en constante contacto con el cliente mientras se desarrolló el software gracias al modelo de desarrollo seleccionado por el equipo de trabajo, permitió que el sistema se desarrolle a la medida y gustos del cliente, lo cual permite un menor tiempo de adaptación y aprendizaje sobre el funcionamiento del sistema.

5.2.4 Factibilidad Técnica

Para la realización del presente proyecto se necesitó la disposición de diferentes tecnologías, que son necesarias para implementar las variadas funcionalidades.

En las primeras etapas del proyecto corresponden a estudio, análisis y diseño del sistema a desarrollar por lo que ocuparan diferentes herramientas provenientes de Microsoft, tales como Office y Visio [13]. Además, para modelar prototipos se utilizó una aplicación web llamada “Whimsical Wireframes” la cual permite diseñar prototipos para aplicaciones móviles de manera gratuita. Una última herramienta necesaria para modelar, principalmente para realizar diagramas, es la aplicación web llamada “Lucidchart” que permite, de manera gratuita, el desarrollo de distintos tipos de diagramas.

Para la etapa de implementación de la solución se utilizarán tecnologías y herramientas tales como PHPStorm [14] que brinda una interfaz amigable para la edición de código, IONIC que es un framework para el desarrollo de aplicaciones móviles que tiene la característica de otorgar la posibilidad de programar tanto para IOS como Android, Firebase la cual es una plataforma para diseño web y móvil la cual permite el uso de una base de datos no SQL y de tiempo real llamada “Realtime Database”, la cual presenta una gran ventaja al momento de trabajar con posiciones y movimientos del GPS.

Para el desarrollo de la aplicación se utilizará un servidor local proveído por IONIC y una vez finalizado el software se procederá a subirla a una cuenta de Play Store con una cuenta brindada por el cliente mismo, por lo cual no necesita mayor análisis.

5.3 Carta Gantt

A continuación se mostrarán las Cartas Gantt con metodología Scrum, es decir con iteraciones cada iteración puede tener tareas, actividades, hitos y responsables correspondientes.

5.3.1 Carta Gantt general

Seguidamente se mostrará la Carta Gantt general, con sus tareas desde agosto a diciembre del 2018. Las demás Cartas Gantt se encuentran en el Anexo A.

	Nombre de tarea	Duración	Comienzo	Fin	Hito
1	Confirmación de tema proyecto	3 días	mié 15-08-18	vie 17-08-18	No
2	Entrega de formulario de inscripción	1 día	lun 20-08-18	lun 20-08-18	Sí
3	Sprint 1 Análisis	10 días	mar 21-08-18	lun 03-09-18	No
4	Sprint 2 Diseño	8 días	mar 04-09-18	jue 13-09-18	No
5	Sprint 3 Programación	55 días	vie 14-09-18	jue 29-11-18	No
6	Entrega informe de avance	1 día	vie 28-09-18	vie 28-09-18	Sí
7	Presentación de avance	1 día	vie 05-10-18	vie 05-10-18	Sí
8	Sprint 4 Testing	6 días	mié 21-11-18	mié 28-11-18	No
9	Entrega de software terminado	1 día	vie 30-11-18	vie 30-11-18	No
10	Revisión del software	1 día	lun 03-12-18	lun 03-12-18	Sí
11	Entrega informe final	1 día	vie 07-12-18	vie 07-12-18	Sí
12	Exposición final	10 días	lun 10-12-18	vie 21-12-18	Sí

Figura 5.1 Carta Gantt General

5.4. Gestión de riesgos

En el presente ítem se define la gestión de dos posibles riesgos importantes que pudieron ocurrir durante el desarrollo del proyecto, indicando en cómo afectaría en el proyecto si es que ocurrieran.

El primer riesgo por considerar es que el cliente proponga realizar un cambio en los requerimientos del sistema.

Tabla 5.5 Gestión Riesgo 1

Impacto en empresa	Puede provocar un retraso en la entrega estimada del proyecto, debido a la posible eliminación, agregación o modificación de los requerimientos establecidos.
Impacto para los desarrolladores	Gran impacto en el desarrollo del sistema, debido a que alguna modificación, agregación o eliminación puede causar reconstruir gran parte del diseño en el sistema.
Trabajo para producir	Se deberá realizar un rediseño en el sistema a crear dependiendo del tamaño de los cambios.

El segundo riesgo por considerar, es realizar una mala elección de la herramienta para el desarrollo del sistema por parte de los desarrolladores puede provocar distintos impactos.

Tabla 5.6 Gestión Riesgo 2

Impacto en empresa	Existiría un retraso en la entrega final del proyecto, debido a que se deberían elegir otras herramientas de desarrollo o aprendizaje de otras nuevas, por lo que generaría un retraso en el cierre del proyecto.
Impacto para los desarrolladores	Destinar horas hombres extras por parte de los desarrolladores para poder realizar un aprendizaje de las nuevas herramientas o solo que trabajen con herramientas ya aprendidas, ambas difieren con el acuerdo realizado con el cliente, por lo que se debe postergar la entrega final del proyecto.
Trabajo para producir	Existen tres opciones: -La primera es realizar capacitación para continuar con las mismas herramientas de desarrollo seleccionadas y así el proyecto continuaría con estas mismas. - Lo segundo es seleccionar otras herramientas de desarrollo ya aprendidas por los desarrolladores, para reconstruir el sistema. - La tercera sería aprender nuevas herramientas completamente, lo que llevaría a empezar el proyecto desde 0.

6 Requerimientos

En el siguiente ítem se especifican los requerimientos funcionales y no funcionales.

6.1 Requerimientos funcionales

A continuación se presentan los requerimientos funcionales por módulo de la App y se va analizar si se cumplieron, con lo especificado en el informe de avance o si se tuvieron que modificar algunos requerimientos.

- El primer módulo se cumplió a cabalidad.

Tabla 6.1 Requerimientos Funcionales Login

Módulo de Login		
ID Requerimiento	Requerimiento	Prioridad
RF-1.1	El sistema deberá aceptar el inicio de sesión, mediante el correo y la contraseña.	Alta
RF-1.2	El sistema podrá, recuperar la clave, ingresando su correo, en el que llegará un mensaje con un link, para poder cambiar la clave.	Alta
RF-1.3	Los usuarios del sistema podrán, cerrar su sesión.	Alta

- El segundo módulo casi se realizó completamente, solamente faltó un buscador específico en los administradores y comunidades, para así facilitar las tareas del super administrador o administrador del sistema.

Tabla 6.2 Requerimientos Funcionales Administrador Sistema

Módulo de Administrador del sistema		
ID Requerimiento	Requerimiento	Prioridad
RF-2.1	Gestionar (Agregar, modificar, eliminar y buscar) cuenta de los administradores. (Falto Buscador)	Alta
RF-2.2	Gestionar (Agregar, modificar, eliminar y buscar) las comunidades que pueden tener los administradores. (Falto Buscador)	Alta

- En el tercer módulo faltaron varias funcionalidades las cuales se acordó terminar más adelante:
 - Un buscador para comunidades y guardias.
 - En las Notificaciones Push hubo un problema de versiones, el cual fue solucionado en un mes estableciendo el servicio notificaciones, el cual no fue ocupado en las funcionalidades que lo requerían, porque se prefirió hacer otras funcionalidades esenciales de la aplicación móvil.
 - El requerimiento “Revisar un ranking, con los guardias más atentos, según las respuestas de ellos, al enviar su ubicación actual”, no se realizó.
 - El requerimiento “Visualizar un registro, de los movimientos (latitud,longitud) de los guardias, según ronda fecha y horario”, no se llevó a cabo totalmente porque el requerimiento fue modificado, a que solo el guardia marcará correcto el punto según un radio que se estableció a partir de un estudio pertinente, marcando los puntos correctos de color verde y los incorrectos de color rojo.
 - La alerta no se implementó.

Tabla 6.3 Requerimientos Funcionales Administrador

Módulo de Administrador		
ID Requerimiento	Requerimiento	Prioridad
RF-3.1	Gestionar (Agregar, modificar, eliminar y buscar) sus guardias, mediante el rut. (Falto Buscador)	Alta
RF-3.2	Asignar ronda a sus guardias, enviando una notificación Push. (Problema notificación)	Alta
RF-3.3	Gestionar (Agregar, modificar, eliminar y buscar) sus comunidades. (Falto Buscador)	Alta
RF-3.4	Enviar una notificación a un guardia, para saber su posición en tiempo real. (No se implementó)	Media
RF-3.5	Revisar un ranking, con los guardias más atentos, según las respuestas de ellos, al enviar su ubicación actual. (No se implementó)	Media

RF-3.6	Asignar guardias a las comunidades.	Alta
RF-3.7	Visualizar un registro, de los movimientos (latitud, longitud) de los guardias, según ronda fecha y horario. (Se modificó)	Alta
RF-3.8	Le llegarán notificaciones, sobres alertas y rondas terminadas de guardia. (No se implementó)	Media
RF-3.9	Observar un listado de guardias, con los resultados de las rondas asignadas. (No se implementó)	Alta

En el cuarto módulo faltaron varias funcionalidades importantes:

- Las Notificaciones Push no fueron implementadas a totalidad, anteriormente en el módulo administrador se explicó detalladamente, sin embargo en el módulo de guardia existe la funcionalidad capaz de enviar un mensaje a su administrador.
- El requerimiento “Compartir y visualizar su ubicación en tiempo real”, no fue desarrollado.
- El requerimiento “Remitir una notificación de alerta al administrador”, no fue realizado.
- El requisito “Revisar su ranking, sobre los guardias más atentos, según las respuestas de ellos, al compartir su ubicación en tiempo real”, no fue implementado.

Tabla 6.4 Requerimientos Funcionales Guardia

Módulo de Guardia		
ID Requerimiento	Requerimiento	Prioridad
RF-4.1	Recibirá una notificación, indicando que debe compartir su ubicación en tiempo real, por un tiempo estimado por el administrador. (No fue desarrollado)	Media
RF-4.2	Compartir y visualizar su ubicación en tiempo real. (No fue implementado)	Media
RF-4.3	Le llegara una notificación, avisando que tiene asignada una ronda. (Se implementó servicio de notificación)	Media

RF-4.4	Visualizar la ronda asignada y realizarla, viendo en tiempo real su camino.	Alta
RF-4.5	Enviar una notificación al administrador, cuando ya ha realizado la ronda.(Problema Notificación)	Media
RF-4.6	Remitir una notificación de alerta al administrador. (No se desarrolló)	Media
RF-4.7	Mandar una notificación al administrador, indicando que está compartiendo la ubicación en tiempo real con él. (No fue desarrollado)	Media
RF-4.8	Revisar su ranking, sobre los guardias más atentos, según las respuestas de ellos, al compartir su ubicación en tiempo real. (No fue implementado)	Media

Debido a los problemas presentados y al poco tiempo destinado para la ejecución del proyecto, se llegó a un acuerdo con el cliente de modificar las prioridades de algunas funcionalidades, para así realizar los requerimientos indispensables de la aplicación.

6.2 Requerimientos no funcionales

En seguida se muestra los requerimientos no funcionales del sistema.

Tabla 6.5 Requerimientos No Funcionales

Atributo	Descripción
Plataforma	Aplicación móvil Android desde la versión 4.1 (lollipop) en adelante.
Requerimientos del producto	Portabilidad: Debe ser exclusiva para teléfonos móvil y tablets.
Requerimientos externos	Privacidad: Información solo debe ser visible de acuerdo a los permisos asignados a su perfil.
Requerimientos específicos	Debe utilizar la API de google maps.

Gestor de base de datos	Firestore con servicio Realtime Database, utilizando plan spark el cual tiene límites generosos para aficionados, este se detalla más abajo.
Conexiones simultáneas	100
GB almacenados	1 GB
GB descargados	10GB/mes
Notificaciones Push	Firestore con servicio Cloud Functions, se detalla a continuación.
Invocaciones	125.000/mes
GB-segundo	40.000/mes
CPU-Segundo	40.000/mes
Hosting	Firestore tiene su propio hosting, se detalla enseguida.
GB almacenados	1 GB
GB transferidos	10 GB/ mes
Diseño	La interfaz de usuario, tendrá los colores institucionales de la empresa, que son los siguientes, estarán detallados con su respectivo valor en html: Dorado: #d59b00 Dorado claro: #f2ba25 Blanco: #f9f2e1 Negro: #080703 Gris: #87847b Dorado más claro: #f9e169

6.3 Análisis de requerimiento

A continuación se presentan los diagramas realizados para el análisis de requerimientos del sistema según lo conversado con el cliente.

6.3.1 Diagrama de Flujo de Datos

El análisis de requerimiento se hizo por medio de un diagrama de flujos de datos, ya que este traza el flujo de la información para cualquier proceso o sistema, en él se podrá apreciar los diferentes módulos del sistema, a continuación, se mostrará el diagrama de contexto donde se muestran los datos que entran y salen de la aplicación móvil.



Figura 6.1 DFD de Contexto

6.3.2 Modelo de Procesos BPMN

Uno de los diagramas seleccionados para explicar de mejor manera, cómo funcionan ciertas funcionalidades del sistema es el BPMN. A continuación se van a presentar las principales funcionalidades, que influyen en el correcto funcionamiento del requerimiento funcional “Asignar ronda a sus guardias” en el módulo del Administrador:

- La figura 6.2 demuestra cómo funciona el proceso de iniciar sesión, para los usuarios del sistema.

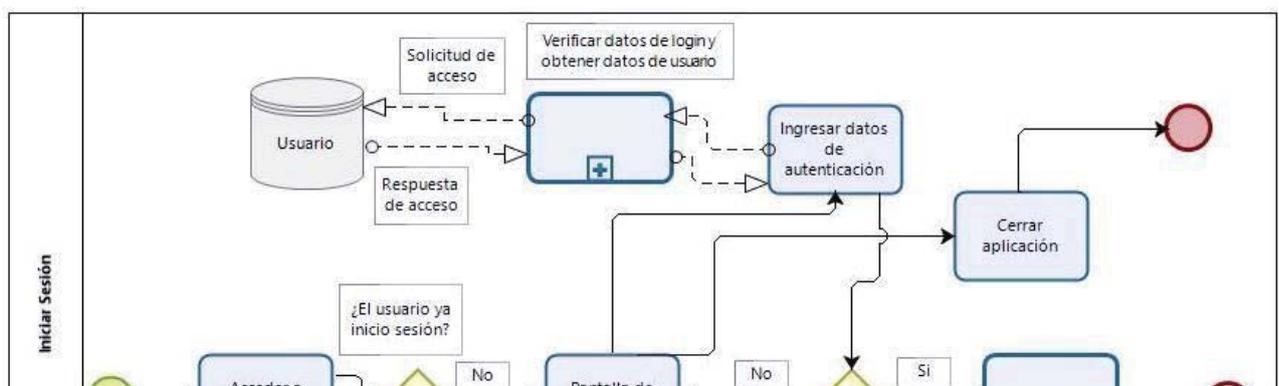
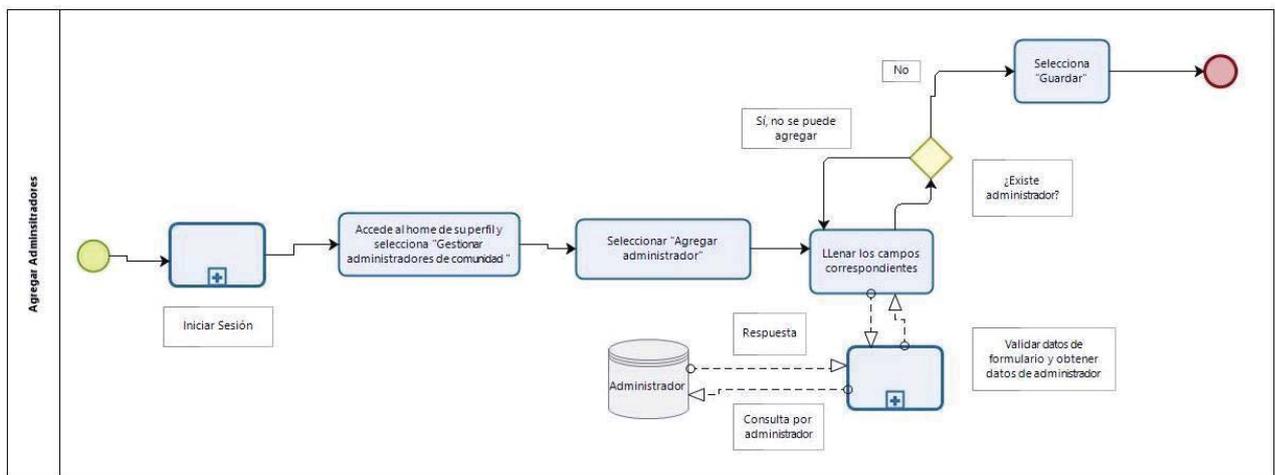


Figura 6.2 BPMN Iniciar Sesión

- La figura 6.2 enseña cómo funciona el proceso de agregar administradores por parte del súper administrador o administrador del sistema, además cabe mencionar que el administrador debe ser asignado a una comunidad en otro módulo, para poder continuar con la demostración del requerimiento “Asignar ronda a sus guardias”.



Powered by
bizagi
Modeler

Figura 6.3 BPMN Agregar Administradores

Los BPMN de “agregar un guardia a una comunidad”, “crear ronda por el administrador” y “asignar ronda a un guardia” se encuentran en el Anexo B en el orden mencionado.

6.4 Diseño de aplicación

Para realizar el diseño de la App utilizaremos Ionic UI, el cual es un framework que utiliza CSS y HTML5, este contiene una multitud de elementos de front-end reutilizables y personalizables, que permiten desarrollar una interfaz de usuario relativamente rápido.

6.5 Prototipos

A continuación se presentan algunas pantallas principalmente del sistema, tanto en el perfil del guardia como en los de administradores. Se detallan pantallas del guardia tales como: el “inicio de sesión”, pantalla “bienvenida del guardia”, pantalla “ver ubicación actual” e “iniciar ronda”; las cuales se muestran a continuación.



El prototipo muestra una pantalla con un encabezado amarillo que contiene el texto "COMPANI". Debajo del encabezado hay un formulario con dos campos de entrada: "Correo" y "Contraseña". Debajo del formulario hay un botón que dice "INICIAR SESIÓN" y un enlace que dice "¿Olvidó su contraseña?".

Figura 6.4 Pantalla Inicio de Sesión

En la figura 6.4 se muestra la pantalla de inicio al iniciar la aplicación en el teléfono móvil, en esta pantalla se debe ingresar correo y contraseña para iniciar sesión o bien, en caso que haya olvidado su contraseña puede clicar en el botón “¿Olvidó su contraseña?” para recuperarla.

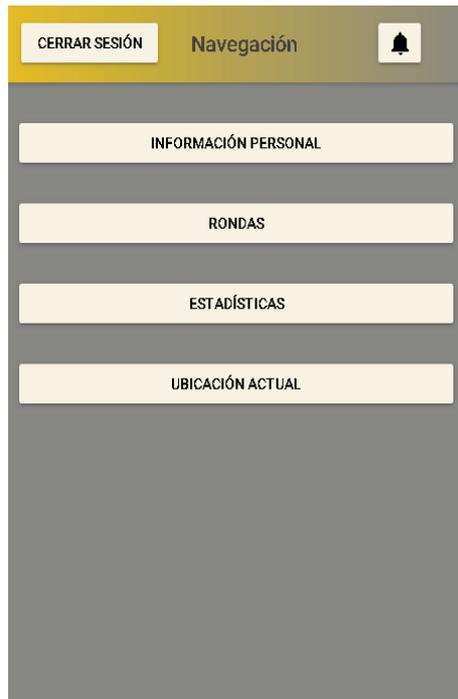


Figura 6.5 Pantalla Home Guardia

En la figura 6.5 se muestra la pantalla inicial del guardia una vez que inicia sesión. Se le mostrará 4 botones con las distintas funcionalidades que se le permite a este perfil de usuario. Puede seleccionar alguna de ellas, puede hacer click en el botón superior izquierdo para cerrar sesión o también, puede clickear en el botón con la campana para revisar si tiene alguna notificación.

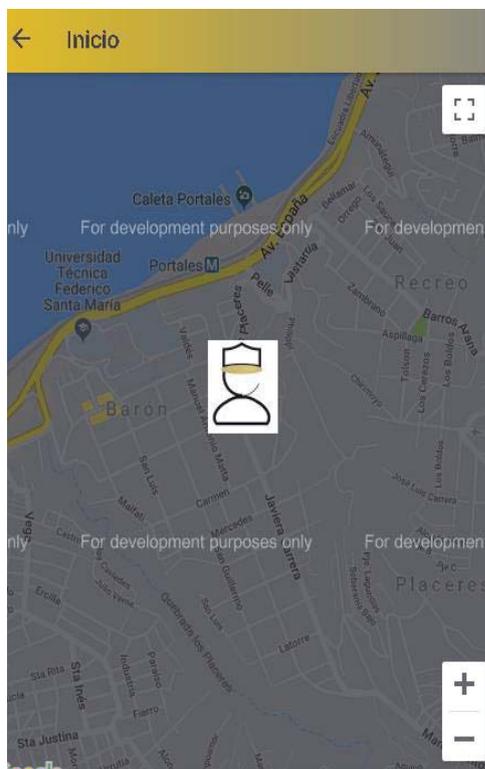


Figura 6.6 Pantalla Ubicación Actual Guardia

En la figura 6.6 se muestra la ubicación actual del guardia. Se llega a esta pantalla una vez que el guardia oprimió el botón de ubicación actual. En esta pantalla el guardia puede ver su ubicación actual o presionar la flecha en la esquina superior izquierda para volver atrás.

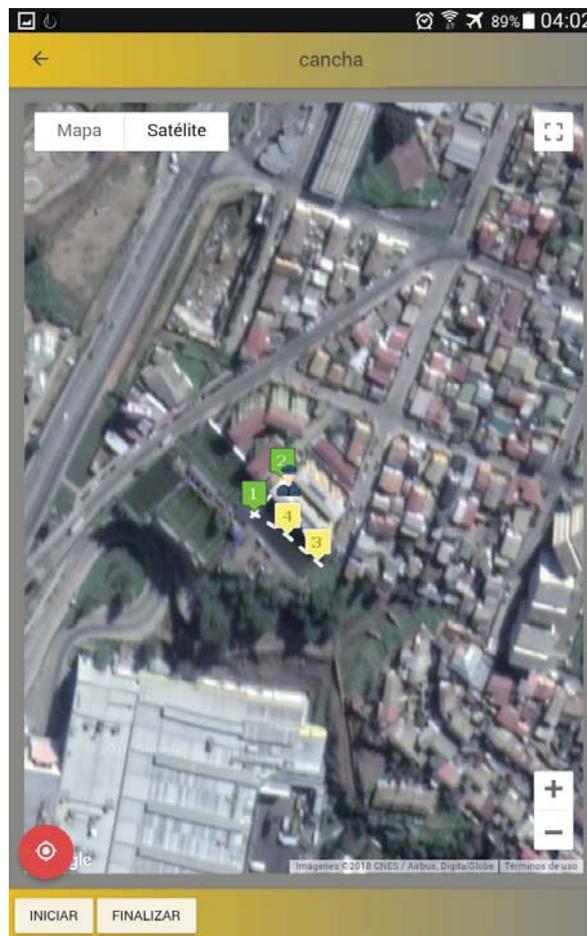


Figura 6.7 Pantalla iniciar ronda

En la figura 6.7 se enseña la pantalla del guardia en donde puede iniciar la ronda. En la pantalla se muestra la figura del guardia y va marcando los puntos recorridos con el color verde y los puntos que aún no se marcan con color amarillo. El guardia debe presionar iniciar para empezar el contador de tiempo de la ronda y el finalizar para detenerlo.

Por parte del administrador del sistema se enseñan las pantallas de “Menú administrador” y “Ver y asignar ronda”

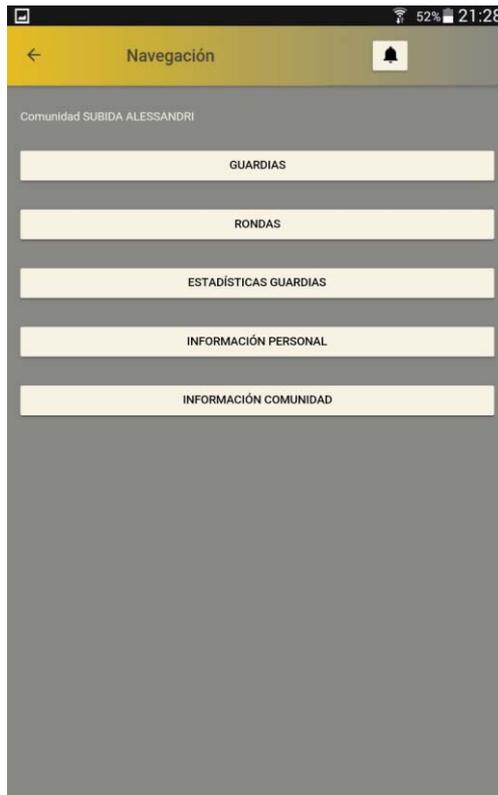


Figura 6.8 Home administrador

En la figura 6.8 se muestra la pantalla de “Home administrador”, en la cual se pueden apreciar 6 opciones: “GUARDIAS”, “RONDAS”, “ESTADÍSTICAS GUARDIAS”, “INFORMACIÓN PERSONAL”, “INFORMACIÓN COMUNIDAD” y el botón para notificaciones en la parte superior.



Figura 6.9 Ver y asignar ronda

La figura 6.9 detalla la pantalla de “ver y asignar ronda”, en donde el administrador puede ver a ronda creada, asignarle un guardia y guardar o cancelar los cambios. Al presionar la lista desplegable en la parte superior de la pantalla, se mostrará el listado completo de guardias que pertenecen a la comunidad que administra el administrador.

Para el tercer módulo, creado para el perfil de administrador de sistema, se enseñarán las pantallas de “Gestionar comunidades” y “Gestionar administradores de comunidad”

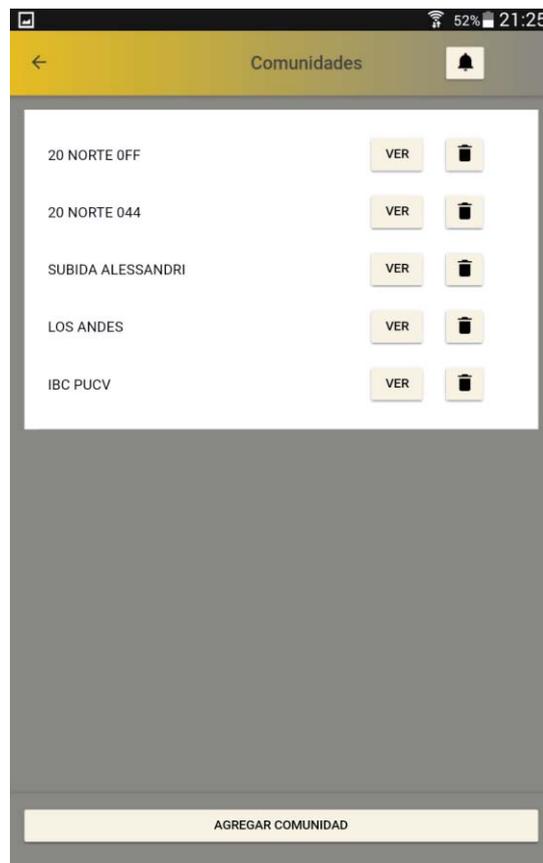


Figura 6.10 Gestionar comunidades

En la figura 6.10 se muestra la pantalla “Gestionar comunidades” en donde se puede evidenciar el listado de comunidades en el sistema y los botones “VER”, un botón para eliminar ubicado a la derecha de “VER” y el botón “AGREGAR COMUNIDAD” en la parte inferior.

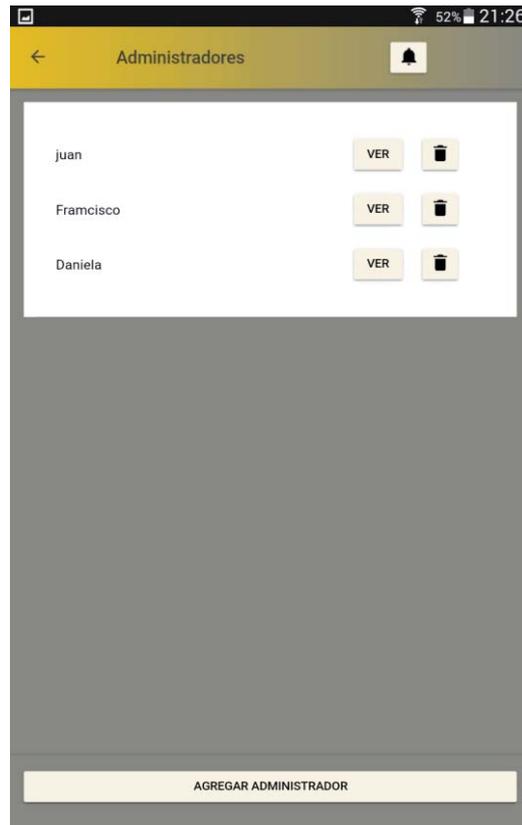


Figura 6.11 Gestionar administrador de comunidad

En la figura 6.11 se detalla la pantalla “Gestionar administrador de comunidad” en donde se puede evidenciar el listado de administradores de comunidades en el sistema y los botones “VER”, un botón para eliminar ubicado a la derecha de “VER” y el botón “AGREGAR ADMINISTRADOR” en la parte inferior.

Además de las pantallas recién mencionadas, en el Anexo C se muestran algunas de las pantallas más importantes desarrolladas según prioridades, separadas por administrador del sistema, administrador y guardias.

6.6 Motor de base de datos

Como motor de base de datos se decidió utilizar “Firebase”, en particular su servicio llamado “Realtime Database”, el cual almacena y sincroniza datos con la base de datos NoSQL alojada en la nube. Los datos se sincronizan con todos los clientes en tiempo real y se mantienen disponibles cuando la app no tiene conexión.

Se optó por utilizar una base de datos NoSql porque esta es más flexible que la Sql, no hace falta realizar grandes consultas para obtener información de la base de datos, y a su vez, garantiza la integridad y la seguridad como una base Sql.

6.7 Modelo Relacional

A continuación se presenta el modelo relacional de la solución propuesta que será implementado a través de un modelo no relacional. Este modelo presentado a continuación mostrará cómo se relacionan las tablas entre sí, sin precisamente trabajar con base de datos relacional, ya que se utilizará un motor de BD no relacional llamado Firebase. Cabe destacar que la mayoría de las tablas, además de contener los atributos que se muestran en la Figura 6.17, contienen un campo llamado “activo”, en donde indica si el registro se encuentra actualmente activo y por lo tanto, visible para los usuarios del sistema, es decir, se elimina de manera lógica pero no de manera física, por lo cual el registro continuará en la base de datos solo que quedará inhabilitado para los usuarios.

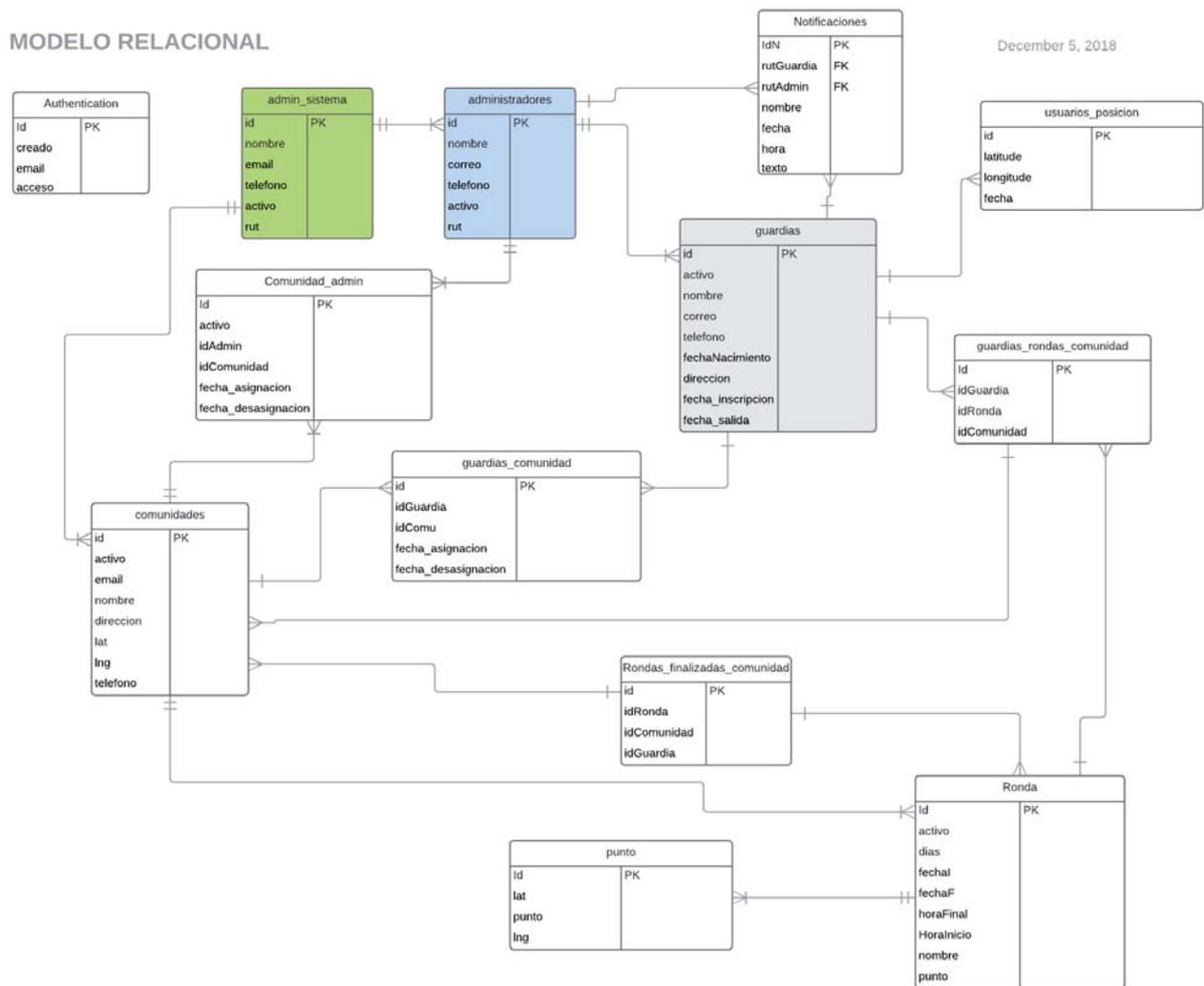


Figura 6.12 Modelo Relacional

7 Conclusión

A continuación, se detallan las conclusiones con referencia al proyecto de aplicación móvil de geolocalización y monitoreo de guardias. En este ítem se exponen las conclusiones tanto sobre el trabajo realizado, resultados y el trabajo a futuro.

7.1 Sobre el trabajo realizado

Luego del planteamiento del problema por la empresa Compani, se realizaron las etapas del desarrollo de software, análisis, diseño y codificación. El análisis y diseño fue un trabajo en equipo con el cliente, ya que él tiene bastante claro la necesidad que requiere para su empresa, después de eso hubo que investigar las herramientas que debemos utilizar para llevar a cabo el proyecto y llevar a cabo estudios de otras aplicaciones similares. Entonces, al ya tener documentada la información necesaria se comenzó a realizar prototipos para exhibirlos al cliente, así pudiendo capturar y validar requerimientos en las respectivas reuniones. Una vez teniendo los requerimientos validados por el cliente, se empezó a codificar haciendo uso de las siguientes herramientas: node.js, firebase, ionic y angular. Todo este trabajo se llevó a cabo mediante la metodología scrum, la cual nos permitió avanzar con cierta rapidez y detectar ciertos problemas presentados durante el desarrollo que retrasaron un poco el avance de éste, acortando los tiempos destinados al desarrollo de otras tareas para poder solucionar dichos percances. Toda la documentación del proceso de análisis y diseño fue dada a conocer a nuestro cliente con anterioridad, además de la presentación de pantallas y entregables desarrollados y pruebas reales de funcionalidades indispensables.

Cabe destacar que durante el desarrollo del proyecto nos encontramos con varias dificultades que retrasaron bastante el avance, con problemas relacionados con el uso de Angular, Firebase y su actualización en tiempo real, versiones con node.js y sus dependencias. Si bien el proyecto contemplaba cierto margen para estos casos, las dificultades fueron tan reiterativas y prolongadas que terminaron por empañar el trabajo finalmente realizado, considerando que en cualquier proyecto se pueden encontrar con dificultades como esta cuando se trabaja con herramientas nuevas y novedosas.

7.2 Sobre el trabajo futuro

Este proyecto sobre geolocalización, gestión y monitoreo de guardias lleva gran avance en comparación con los objetivos planteados en un principio, teniendo en cuenta que este proyecto está pensado para continuar luego de la entrega y aún faltan funciones por lograr el objetivo principal de este proyecto, que es desarrollar una aplicación totalmente funcional con geolocalización según como decidió el cliente.

Para el futuro se espera poder completar la totalidad de las funcionalidades, esperando llegar a las expectativas del cliente el cual tiene conocimientos de las dificultades presentadas por el proyecto considerando la envergadura y dificultad de éste.

Como equipo estamos conformes con el trabajo realizado, ya que la planificación inicial se acercó mucho a la planificación finalmente realizada y que a pesar de los problemas presentados, pudimos sobrepasarlos y salir adelante con lo que teníamos pensado. Si bien, se tuvo que acortar la cantidad de funcionalidades a desarrollar, siempre nos mantuvimos avanzando o intentando solucionar problemas considerando que en todo proyecto se encuentran conflictos y dificultades que pueden retrasar o afectar en las entregas, especialmente cuando el tiempo de desarrollo es tan limitado.

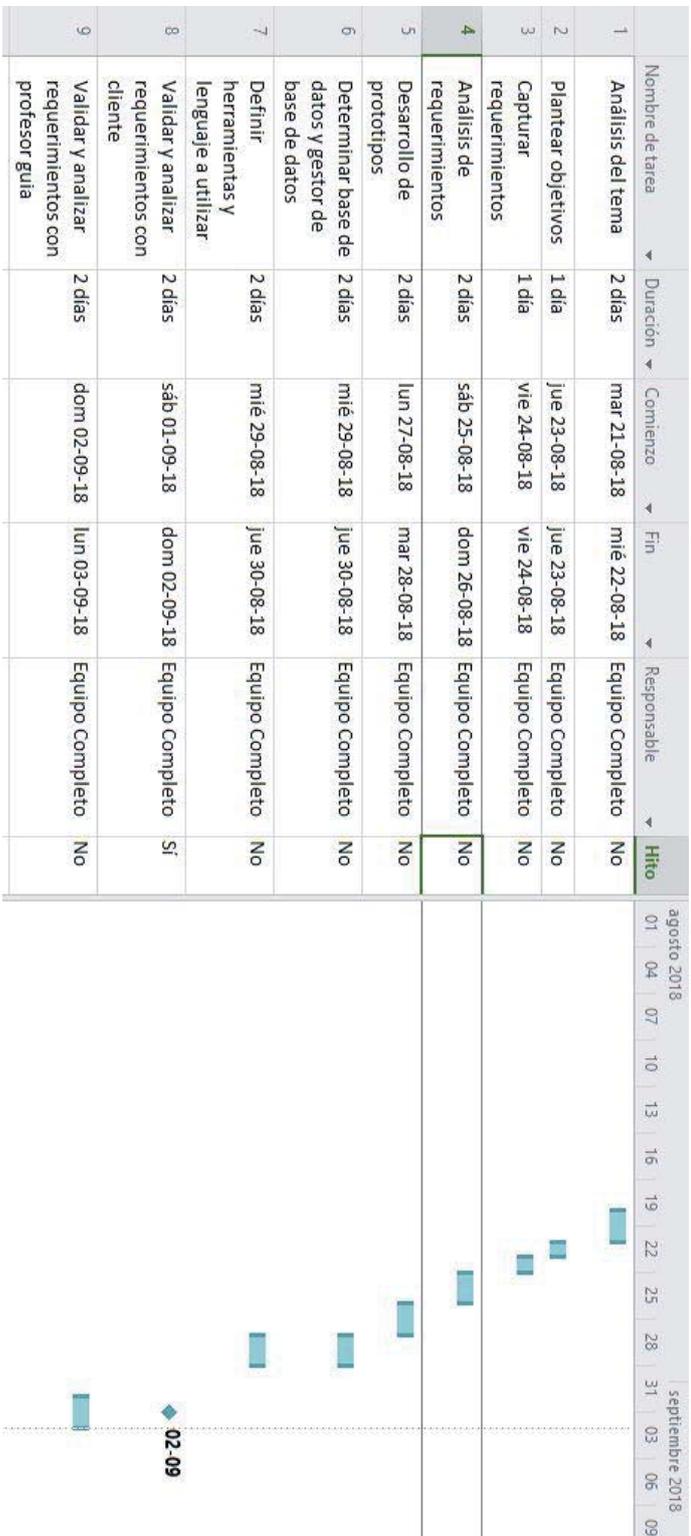
8 Bibliografía

- [1] Página web de la empresa que requiere la aplicación móvil. Disponible vía web en <https://www.compani.cl/> . Revisado por última vez el 20/09/2018.
- [2] Sitio web de Guarnic, aplicación web y móvil de guardias .Disponible vía web en <https://www.guarnic.com/es/control-de-guardias/>. Revisado por última vez el 15/09/2018.
- [3] Página web de Uber, aplicación móvil de transporte .Disponible vía web en <https://www.uber.com/es-CL/cities/santiago/>. Revisado por última vez el 15/09/2018.
- [4] Sitio web oficial de Firebase, Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles. Disponible vía web <https://firebase.google.com/?hl=es-419>. Revisado por última vez el 28/11/2018.
- [5] Se explica como funciona la metodología Scrum. Disponible vía internet <https://proyectosagiles.org/como-funciona-scrum/>. Revisada por última vez el 04/09/2018
- [6] Sitio web oficial de Nodejs. Disponible vía internet <https://nodejs.org/es/>. Revisado por última vez el 20/08/2018.
- [7] Documentación oficial de Ionic. Disponible vía internet <https://ionicframework.com/docs/> .Revisado por última vez el 25/11/2018.
- [8] Documentación oficial de Firebase Cloud Messaging. Disponible vía web <https://firebase.google.com/docs/cloud-messaging/> . Revisada por última vez el 25/10/2018.
- [9]. Documentación oficial de Realtime Database. Disponible vía web <https://firebase.google.com/docs/database/web/read-and-write?hl=es-419>. Revisado por última vez el 28/11/2018.
- [10]__ Modelamiento de la herramienta Lucidchart. Disponible vía internet <https://www.lucidchart.com/pages> . Revisada por última vez el 28/11/2018.
- [11] Valor de dolar. Disponible vía web <https://www.valor-dolar.cl/>. Revisada por última vez el 20/11/2018.
- [12] ley N°19.223 (TIPIFICA FIGURAS PENALES RELATIVAS A LA INFORMÁTICA). Disponible vía internet <https://www.leychile.cl/Navegar?idNorma=30590> fue revisado el 16/09/2018.
- [13] ley N°17.336 (PROPIEDAD INTELECTUAL; DERECHOS DE AUTOR). Disponible vía internet <https://www.leychile.cl/Navegar?idNorma=30590> fue revisado el 16/09/2018.
- [15] Herramienta de desarrollo, Editor y Compilador PHP STORM Disponible vía internet <https://www.jetbrains.com/phpstorm/>, fue revisado el 10/09/2018 .

Anexos

A: Cartas Gantt

Carta Gantt Sprint Análisis



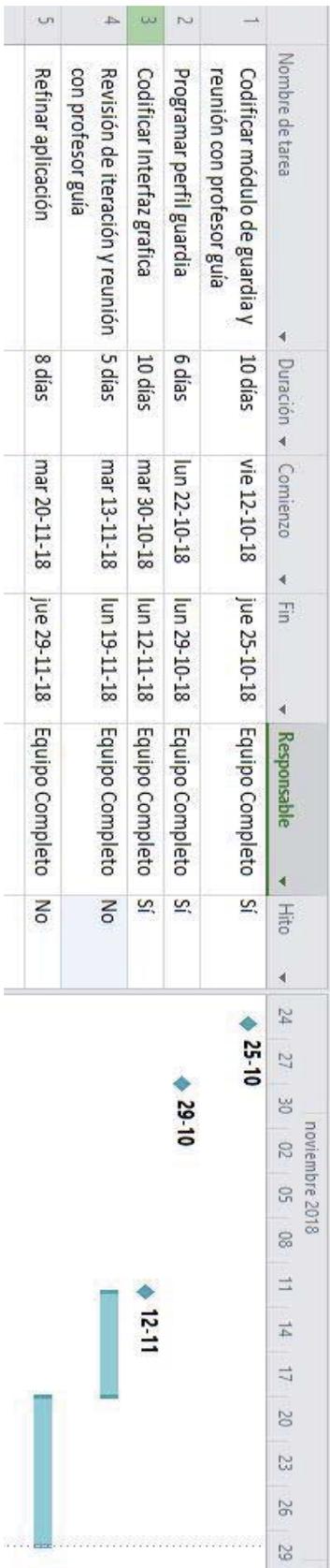
Carta Gantt Sprint Diseño



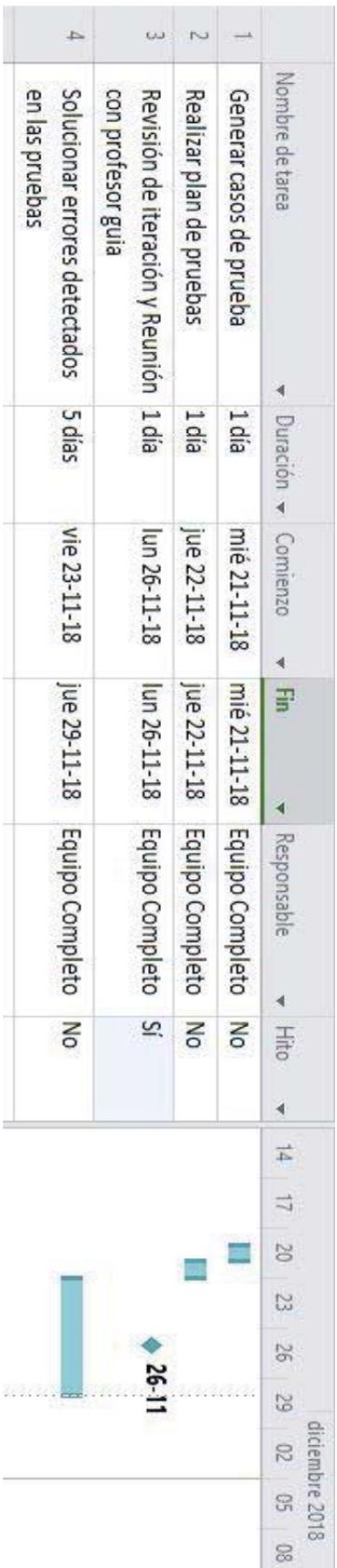
Carta Gantt Sprint Programación 1



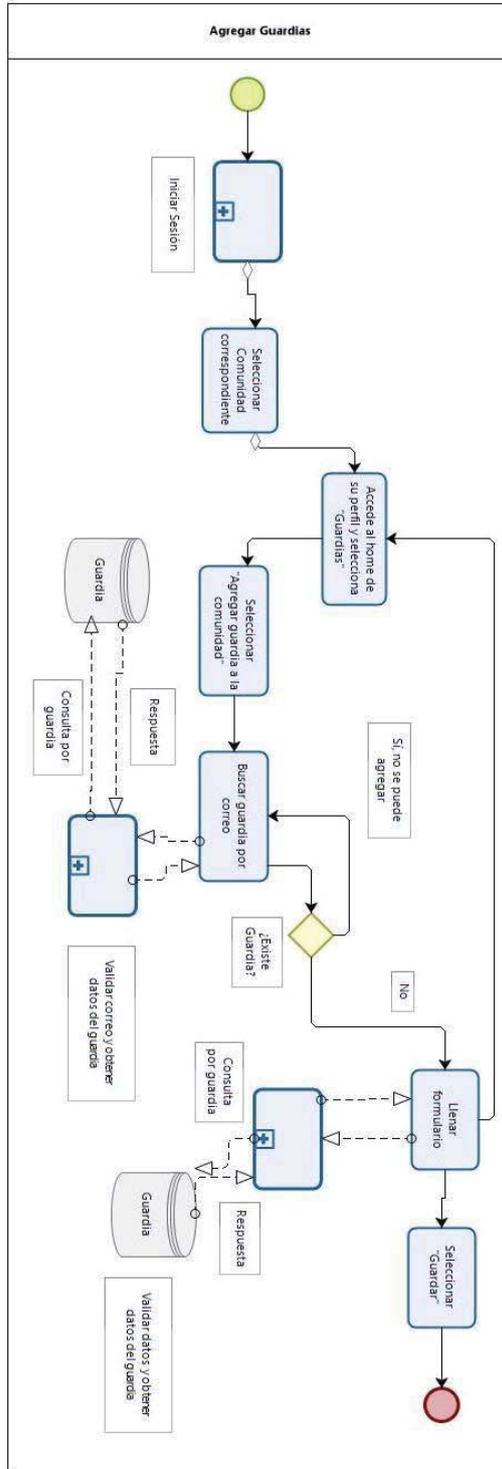
Carta Gantt Sprint Programación 2



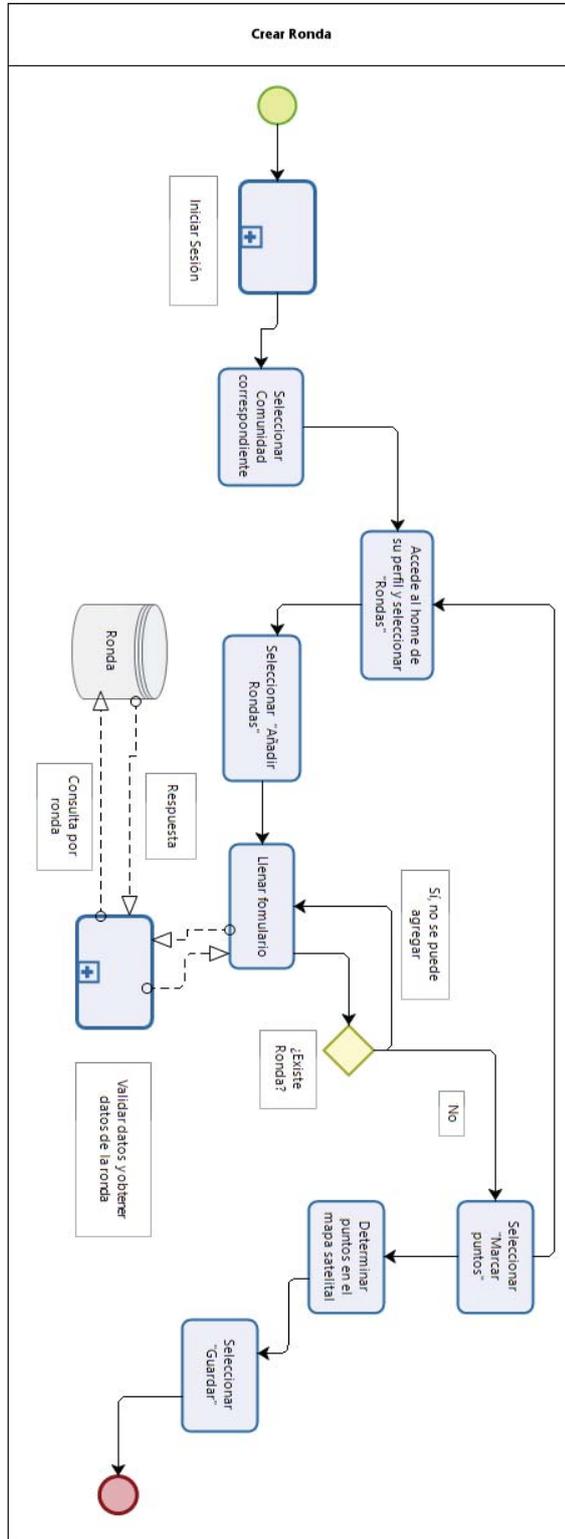
Carta Gantt Sprint Testing



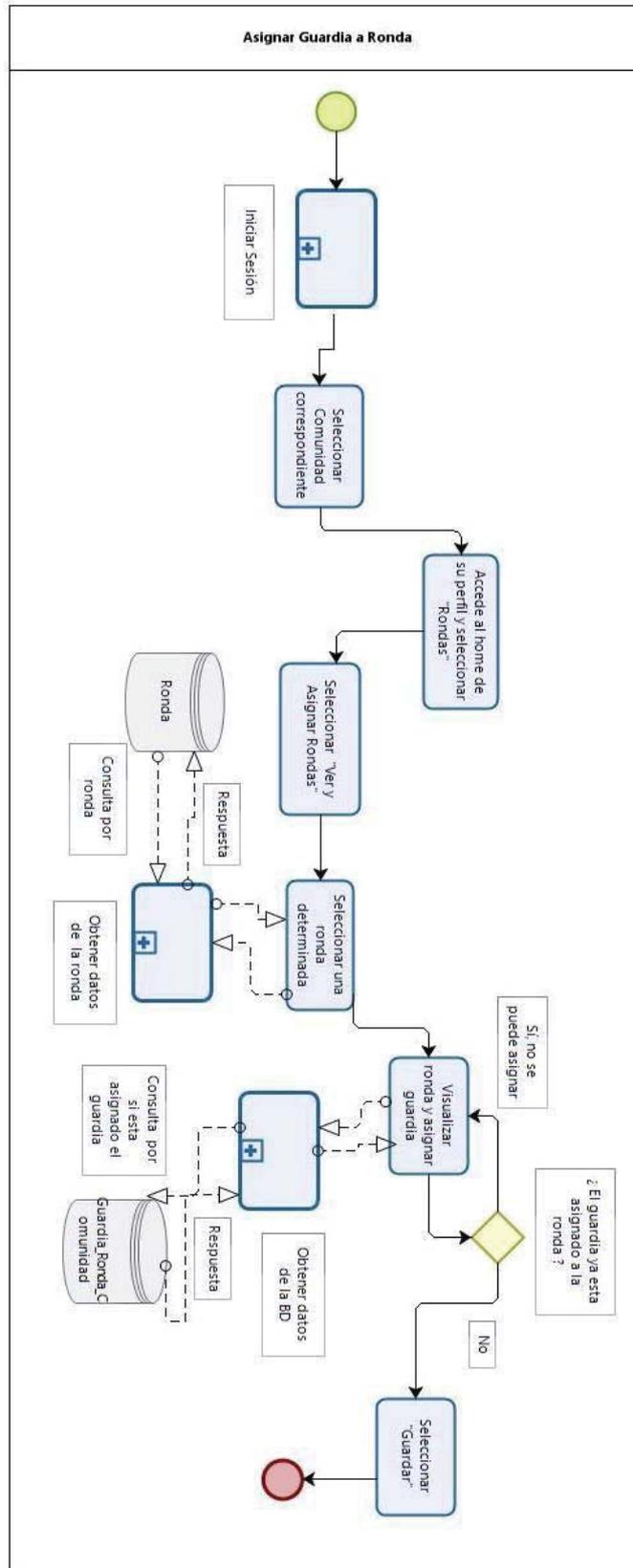
B: BPMN Agregar Guardia



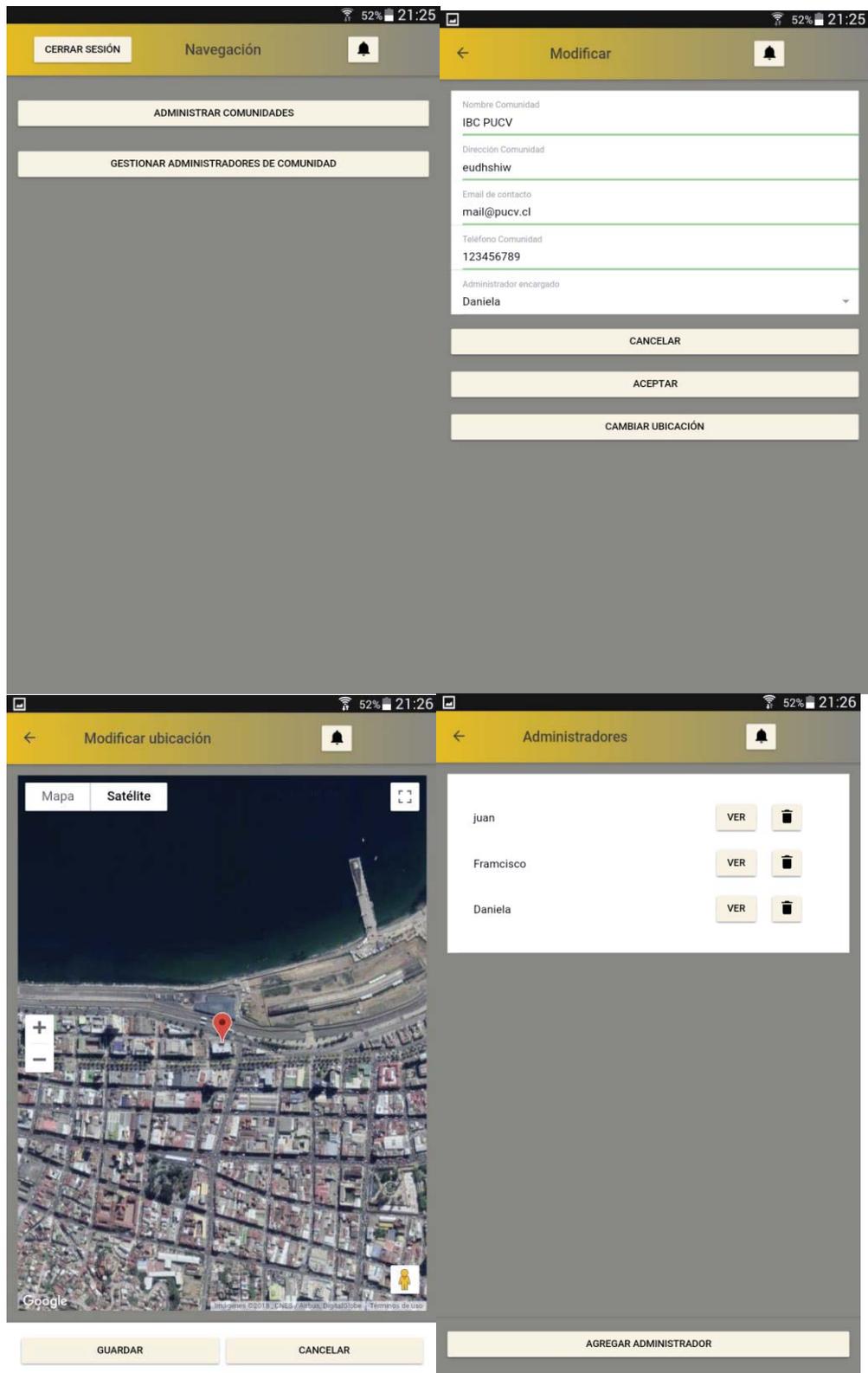
BPMN Crear Ronda



BPMN: Asignar guardia a ronda en una comunidad



C: Pantallas desarrolladas



Administradores

CERRAR SESIÓN Seleccione comu...

Nombre Administrador
Daniela

Rut (Ej: 12345678-9)
11111111-1

Email
daniela@tes.cl

Teléfono
965814573

ACTUALIZAR

SUBIDA ALESSANDRI

LOS ANDES

Ingresar guardia

Guardias

Nombre Guardia
 Nombre Guardia

Rut (Ej: 12345678-9)
 Ej: 12345678-9

Email
 Ej: email@ejemplo.com

Teléfono (9 dígitos)
 Ej: 923456789

Dirección
 Dirección

Fecha de Nacimiento (día/mes/año)

Contraseña (6 caracteres)
 Contraseña

Repetir contraseña
 Repetir Contraseña

AÑADIR GUARDIA

teoeo VER

manuel VER

