

UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN
Facultad de Ingeniería
Ingeniería Civil Informática



Aplicación Móvil para Realizar Talleres de Metaplan

Claudio Andrés Bello Ortega

Informe de Proyecto de Título para optar al Título de
Ingeniero Civil Informático

Profesor Guía:

Pedro Sebastián Gómez Meneses

Concepción, Octubre de 2015

Resumen

Desde la creación del primer celular hace más de 30 años, el avance de la tecnología y de los dispositivos móviles, ha ayudado cada vez más al ser humano en las labores del día a día. Antiguamente eran dispositivos que no podían realizar tareas tan complejas y aparte de utilizarse para realizar llamadas, también eran capaces de agendar contactos, realizar cálculos como una calculadora y programar una alarma sonora.

Hoy es un dispositivo de uso masivo que se ha vuelto casi imprescindible. También se conocen como teléfonos inteligentes, ya que son capaces de realizar actividades similares a las de un computador pequeño pero con conectividad telefónica. Esto se debe a que al igual que un computador se le pueden instalar aplicaciones, o mejor dicho, aplicaciones móviles. La gran cantidad de aplicaciones móviles que existen en estos dispositivos, han reemplazado una gran cantidad de herramientas tradicionales y su alcance cada día se extiende más.

Con el fin de reemplazar la gran variedad de herramientas físicas que son utilizadas para realizar talleres de Metaplan, el objetivo del proyecto es implementar una aplicación Android para realizar talleres de Metaplan.

El desarrollo de este proyecto está registrado en el siguiente documento, y la lectura de éste le permitirá a estudiantes u otras personas interesadas iniciarse en la programación en Android.

Abstract

Since the creation of the first cellphone, over 30 years ago; the advancement of technology and mobile devices has progressively helped humanity with everyday tasks. In the past this devices could not be used to do complex tasks and apart to use for call, they were able to store contacts information, use as a calculator and set as an alarm clock.

Nowadays, it is a device that almost everybody has and its use has become indispensable. Moreover, they are also known as smartphones, as they are able to perform similar tasks to the ones a small computer performs, apart from having telephone connectivity. This is due to the fact that you can install applications, more specifically mobile applications, similarly the way you do with computers. The large number of mobile applications that exist on these devices, have replaced a lot of traditional tools and its extent grows every day.

Given the great numbers of physical tools which are used to conduct Metaplan workshops, the goal of the project is to replace that by implementing an Android application for Metaplan workshops.

The development of this project is discussed and presented in the following document, and the reading of it, will allow students or other people interested in the topic to get started in the Android programming.

Agradecimientos

Le quiero dar gracias a mi hijo Lucas y a su madre Estefania por estar a mi lado, por todo ese amor y fuerza que me brindan en el día a día, a mis padres que nunca dejaron de creer en mí y siempre me brindan su apoyo, a mi hermano que siempre me ayudo cuando solicite de su ayuda sin importar la distancia, sin ellos no hubiera sido capaz de terminar la carrera y les estoy eternamente agradecidos, también a todos los profesores que tuve durante la carrera y a la selección chilena que ganó la Copa América y generó una gran alegría a todos los chilenos.

Índice general

1. Introducción	1
1.1. Presentación del Problema	1
1.2. Justificación del Problema	2
1.3. Delimitación del Problema	2
1.4. Objetivos	2
1.4.1. Objetivo General	2
1.4.2. Objetivos Específicos	3
1.5. Metodología	3
2. Marco Teórico	5
2.1. Metaplan	5
2.1.1. Instrumentos	6
2.1.2. Montaje	6
2.1.3. Técnicas	7
2.1.3.1. Técnica del Collage	7
2.1.3.2. Técnica de la Visualización Permanente	7
2.1.3.3. Técnica de las Preguntas	7
2.2. Estrategias de Moderación Grupal	8
2.3. Computación Móvil	8
2.3.1. El Usuario Móvil.	9
2.3.2. Dispositivos Móviles.	10
2.3.3. Las Redes Inalámbricas.	10
2.3.4. Aplicación Móvil.	11
2.4. Redes Ad Hoc	11
2.5. Android	12

2.5.1.	Sus Origenes	12
2.5.2.	Versiones de Android	12
2.5.3.	Android Studio	14
3.	Estado del arte	15
3.1.	Metaplan Virtual	15
3.2.	Trello	16
4.	Especificación de Requerimientos	17
4.1.	Introducción	17
4.2.	Definiciones, Acrónimos y Abreviaturas	17
4.3.	Ámbito del Sistema	18
4.4.	Descripción General	18
4.4.1.	Característica de los Usuarios	18
4.4.2.	Restricciones	18
4.5.	Requisitos Específicos	19
4.5.1.	Funcionalidades	19
5.	Diseño	25
5.1.	Diagrama de Actividad	25
5.1.1.	Diagrama de Actividad - Ingresar Moderador	28
5.1.2.	Diagrama de Actividad - Ingresar Participante	28
5.1.3.	Diagrama de Actividad - Ingresar Taller	29
5.1.4.	Diagrama de Actividad - Consultar Talleres	29
5.1.5.	Diagrama de Actividad - Modificar Taller	30
5.1.6.	Diagrama de Actividad - Ingresar Pregunta	30
5.1.7.	Diagrama de Actividad - Consultar Preguntas	31
5.1.8.	Diagrama de Actividad - Modificar Pregunta	31
5.1.9.	Diagrama de Actividad - Ingresar Respuesta	32
5.1.10.	Diagrama de Actividad - Liberar Respuestas	32
5.1.11.	Diagrama de Actividad - Consultar Respuestas	33
5.1.12.	Diagrama de Actividad - Modificar Respuesta	33
5.1.13.	Diagrama de Actividad - Eliminar Respuesta	34
5.1.14.	Diagrama de Actividad - Consultar Participantes	34

5.1.15. Diagrama de Actividad - Finalizar Taller	35
5.2. Diagrama de Componentes	36
5.3. Interfaces de Usuario	37
5.4. Diccionario de Datos	43
5.5. Modelo Relacional	45
5.6. Modelo Entidad-Relación	46
6. Implementación	47
6.1. Estructura	47
6.2. Android Manifest	48
6.3. Iniciando la Aplicación	49
6.3.1. MainActivity.java	49
6.3.2. activity_main.xml	51
6.4. Pantalla Selecciona tu Perfil	52
6.4.1. MainOpciones.java	52
6.4.2. activity_main_opciones.xml	53
6.5. Pantalla Iniciando como Moderador	53
6.5.1. TrackModerador.java	53
6.5.2. activity_track_moderador.xml	57
6.6. Pantalla Iniciando como Participante	58
6.7. Pantalla Talleres	59
6.7.1. Talleres.java	59
6.7.2. activity_talleres.xml	63
6.8. Pantalla Preguntas	64
6.8.1. activity_preguntas.xml	64
6.9. Pantalla Respuestas	64
6.9.1. Respuestas.java	65
6.10. Bases de Datos	66
7. Pruebas	69
7.1. Pruebas de Usabilidad	69
7.2. Resultados Pruebas de Usabilidad	70
7.3. Pruebas de Funcionamiento	71
7.4. Pruebas de Requerimientos	71

7.5. Resultado Pruebas de Requerimientos	72
8. Conclusiones y Recomendaciones	73
Bibliografía	76
Anexo	77
Manual de Usuario	77
Manual del Moderador	79
Manual del Participante	88
Pruebas de Usabilidad	93

Índice de figuras

2.1. Panel en el que se pega el papel kraft y respuestas.	6
2.2. Clasificación de las redes inalámbricas.	11
3.1. Prototipo de Metaplan Virtual, obtenido de unlp.edu.ar.	15
3.2. Metaplan en Trello.	16
5.1. Parte 1 del Diagrama de Actividad del sistema.	26
5.2. Parte 2 del Diagrama de Actividad del sistema.	27
5.3. Diagrama de actividad del requisito funcional Ingresar Moderador.	28
5.4. Diagrama de actividad del requisito funcional Ingresar Participante.	28
5.5. Diagrama de actividad del requisito funcional Ingresar Taller.	29
5.6. Diagrama de actividad del requisito funcional Consultar Talleres.	29
5.7. Diagrama de actividad del requisito funcional Modificar Taller.	30
5.8. Diagrama de actividad del requisito funcional Ingresar Pregunta.	30
5.9. Diagrama de actividad del requisito funcional Consultar Preguntas.	31
5.10. Diagrama de actividad del requisito funcional Modificar Pregunta.	31
5.11. Diagrama de actividad del requisito funcional Ingresar Respuesta.	32
5.12. Diagrama de actividad del requisito funcional Liberar Respuesta.	32
5.13. Diagrama de actividad del requisito funcional Consultar Respuesta.	33
5.14. Diagrama de actividad del requisito funcional Modificar Respuesta.	33
5.15. Diagrama de actividad del requisito funcional Eliminar Respuesta.	34
5.16. Diagrama de actividad del requisito funcional Consultar Participantes.	34
5.17. Diagrama de actividad del requisito funcional Finalizar Taller.	35
5.18. Diagrama de Componentes del sistema.	36
5.19. Wireframe <i>Selecciona tu perfil</i>	37
5.20. Selecciona tu perfil.	37

5.21. Wireframe <i>Iniciando como Moderador</i>	38
5.22. <i>Iniciando como Moderador</i>	38
5.23. Wireframe <i>Iniciando como Participante</i>	39
5.24. <i>Iniciando como Participante</i>	39
5.25. Wireframe <i>Talleres</i>	40
5.26. <i>Talleres</i>	40
5.27. Wireframe <i>Preguntas</i>	41
5.28. <i>Preguntas</i>	41
5.29. Wireframe <i>Respuestas</i>	42
5.30. <i>Respuestas</i>	42
5.31. Modelo Relacional “Metaplandroid”.	45
5.32. Modelo Entidad-Relación “Metaplandroid”.	46
6.1. Pantalla de carga de la aplicación.	49
1. <i>Iniciando Aplicación</i>	77
2. <i>Selecciona tu Perfil</i>	78
3. <i>Error al seleccionar tu Perfil</i>	78
4. <i>Iniciando como Moderador</i>	79
5. <i>Indicaciones para iniciar como Moderador</i>	79
6. <i>Error al iniciar como Moderador</i>	80
7. <i>Sin Talleres creados</i>	80
8. <i>Creación de un Taller</i>	81
9. <i>Desplegando el Taller creado</i>	81
10. <i>Sin Preguntas creadas</i>	82
11. <i>Creación de una Pregunta</i>	82
12. <i>Desplegando Pregunta creada</i>	83
13. <i>Esperando Respuestas</i>	83
14. <i>Recibiendo Respuestas</i>	84
15. <i>Liberando Respuestas</i>	84
16. <i>Editando o Eliminando una Respuesta</i>	85
17. <i>Respuestas Listas</i>	85
18. <i>Pasando a otra Pregunta</i>	86
19. <i>Finalizando Taller</i>	86

20.	Enviando Taller.	87
21.	Iniciando como Participante.	88
22.	Indicaciones para iniciar como Participante	88
23.	Errores al iniciar como Participante.	89
24.	Esperando el Taller.	89
25.	Desplegando Taller creado.	90
26.	Esperando la Pregunta.	90
27.	Desplegando pregunta creada.	91
28.	Sin Respuestas.	91
29.	Ingresando una Respuesta.	92
30.	Desplegando Respuesta creada.	92

Índice de tablas

2.1. Listado de versiones Android.	13
5.1. Diccionario de Datos de la tabla Taller.	43
5.2. Diccionario de Datos de la tabla Pregunta.	43
5.3. Diccionario de Datos de la tabla Respuesta.	44
5.4. Diccionario de Datos de la tabla Participante.	44
7.1. Cuestionario de usabilidad System Usability Scale (SUS).	69
7.2. Promedio de los cuestionarios de usabilidad System Usability Scale (SUS).	70
7.3. Prueba de requerimientos de la aplicación móvil Metaplandroid.	71
7.4. Resultado de las prueba de requerimientos de la aplicación parte 1.	72
7.5. Resultado de las prueba de requerimientos de la aplicación parte 2.	72
1. Resultado 1 del cuestionario de usabilidad System Usability Scale (SUS).	93
2. Resultado 2 del cuestionario de usabilidad System Usability Scale (SUS).	93
3. Resultado 3 del cuestionario de usabilidad System Usability Scale (SUS).	94
4. Resultado 4 del cuestionario de usabilidad System Usability Scale (SUS).	94
5. Resultado 5 del cuestionario de usabilidad System Usability Scale (SUS).	95
6. Resultado 6 del cuestionario de usabilidad System Usability Scale (SUS).	95

Capítulo 1

Introducción

1.1. Presentación del Problema

Metaplan es un taller para recopilar información, en la que interactúa un moderador y un grupo de personas que son los participantes del taller. El moderador es un facilitador del proceso, que no interfiere en los contenidos. El taller consiste en responder preguntas por medio de tarjetas, evitando el cruce de diálogos y las respuestas influenciadas de los participantes. Las respuestas se van pegando en un panel, que está a la vista de los participantes en todo momento, luego el moderador agrupa las respuestas por sinónimos o similitud de la idea y se procede a la siguiente pregunta hasta finalizar el taller. Estos talleres necesitan de una gran cantidad de materiales físicos, entre los materiales podemos encontrar: plumones, tarjetas de cartulina, chinchas o cinta adhesiva, papel kraft, paneles y una sala de trabajo amplia.

Se pretende crear un sistema que permita realizar estos talleres de Metaplan desde dispositivos móviles Android, en que la comunicación entre dispositivos será a través de una red wifi. El sistema se llamará “Metaplandroid”. Ésta aplicación móvil tendrá dos interfaces, una para el moderador y otra para los participantes del taller. El moderador creará preguntas desde la aplicación de su dispositivo y serán visualizados desde los dispositivos de los participantes. Las respuestas son ingresadas por los participantes del taller y serán visualizadas por todos los demás, una vez el moderador confirme que todos han respondido a la pregunta, deberá presionar un botón para que todos visualicen las respuestas existentes. A continuación el moderador podrá interactuar con las respuestas, ya sea editando la ortografía de éstas o agregándole un número con la cantidad de gente que respondió de igual manera a la pre-

gunta y posteriormente eliminando las respuestas repetidas que ya han sido contabilizadas. Finalmente los resultados del taller serán guardados con la posibilidad de ser enviados por correo electrónico.

1.2. Justificación del Problema

- La necesidad que tienen los moderadores de Metaplan de poder realizar estos talleres, sin depender de materiales físicos y tiempo para armar una sala con estos materiales montados. De ésta manera se dará más libertad y comodidad a los moderadores.
- La inexistencia de una aplicación similar, que sea capaz de realizar talleres de Metaplan.
- Incorporar herramientas que ayuden al moderador al desarrollo más fluido del taller, integrando funciones que reducirán el tiempo de desarrollo del taller.

1.3. Delimitación del Problema

El sistema estará disponible en dispositivos móviles como teléfonos inteligentes y tabletas con sistema operativo Android, posiblemente podría existir restricciones de la versión de Android. El tipo de lenguaje de programación utilizado será Java.

En caso de que se necesite pagar por una herramienta o publicación de la aplicación, tendrá que quedar pendiente, es decir, sólo se empleará el uso de herramientas libres.

La comunicación entre dispositivos será a través de wifi.

1.4. Objetivos

A continuación se define el objetivo general y los objetivos específicos del sistema.

1.4.1. Objetivo General

Implementar una aplicación Android para realizar talleres de Metaplan.

1.4.2. Objetivos Específicos

En particular para la realización de este proyecto se definieron 4 objetivos específicos, indicados a continuación:

1. Analizar las funcionalidades del sistema.
2. Diseñar y modelar la aplicación.
3. Construir la aplicación.
4. Testear la aplicación creada.

1.5. Metodología

Para la metodología se seleccionó el *Modelo de Entrega Incremental*, con el que se realizaron tres entregas de la aplicación, cada versión más completa que la anterior y con nuevas funcionalidades.

1. Para el objetivo específico 1: “Analizar las funcionalidades del sistema”, se definieron una serie de actividades, que se clasifican en dos categorías:
 - Aprender sobre Metaplan.
 - Investigar en documentos oficiales.
 - Buscar sobre Metaplan en bases de datos como “Google Scholar”.
 - Comunicarse con un moderador de Metaplan.
 - Ver un taller de Metaplan.
 - Aprender sobre las herramientas y lenguaje Java.
 - Investigar en bases de datos como “SiBUCSC” o “Google Scholar”.
 - Investigar en proyectos de títulos similares.
 - Investigar aplicaciones con redes wifi.
 - Seleccionar la herramienta de desarrollo más adecuada.
2. Para el objetivo específico 2: “Diseñar y modelar la aplicación”, se realizaron las siguientes actividades:

- Estudiar los modelos más adecuados.
 - Seleccionar modelos.
 - Confeccionar los diferentes modelos.
3. Para el objetivo específico 3: “Construir la aplicación”, se realizaron las siguientes actividades:
- Estudiar y aplicar los conocimientos de Java.
 - Programar la aplicación en la herramienta de desarrollo seleccionada.
 - Entregar versiones de la aplicación para ser evaluadas.
4. Para el objetivo específico 4: “Testear la aplicación creada”, se realizaron las siguientes actividades:
- Seleccionar los test más adecuados.
 - Aplicar los test seleccionados.

Capítulo 2

Marco Teórico

2.1. Metaplan

Metaplan es un método de moderación grupal, en la que interactúa un moderador y un grupo de personas que son los participantes del taller, cuyo objetivo es solucionar problemas y crear, almacenar, estructurar y/o visualizar ideas (Hughes et al., 2009). El moderador es un facilitador del proceso, que no interfiere en los contenidos, y los participantes son personas involucradas con el problema. Metaplan además es una marca comercial alemana cuya compañía fue fundada por uno de los pioneros de esta metodología, Thomas Schnelle.

En éste taller se responden las preguntas por medio de tarjetas, con lo que se evita el cruce de diálogos y las respuestas influenciadas entre los participantes. Estas respuestas se van pegando en un panel, que está a la vista de los participantes en todo momento, luego el moderador puede agrupar las respuestas por similitud (cluster) y se procede a la siguiente pregunta hasta finalizar el taller.

Metaplan utiliza un conjunto de *instrumentos* y *técnicas* como de visualización y preguntas que permiten realizar este taller de la mejor forma posible (Hughes et al., 2009).

2.1.1. Instrumentos

La realización de un análisis utilizando Metaplan requiere varias herramientas:

- Panel.
- Grandes pliegos de papel (papel kraft).
- Tarjetas de cartulina.
- Chinchas o cinta adhesiva.
- Plumones (rotuladores).

Se debe considerar una cantidad de *plumones* y *tarjetas de cartulina* proporcional a la cantidad de participantes del taller de Metaplan, por ejemplo, si se desea buscar solución a cuatro problemas, se calculan dos tarjetas por pregunta para cada participante, por lo tanto, se necesitarán ocho tarjetas de cartulina por persona, y para ahorrar tiempo y evitar conversaciones mientras se responde las preguntas, es importante que cada participante posea un *plumón* o *rotulador* (Metaplan, a).

2.1.2. Montaje

Las grandes hojas de *papel kraft* están puestas en los *paneles* (ver Figura 2.1). Todas las contribuciones hechas durante el debate se escriben en las *tarjetas de cartulina* utilizando *plumones* y se fijan al *panel* utilizando *chinchas* o *cinta adhesiva*.

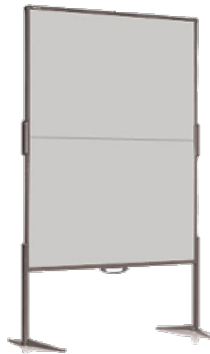


Figura 2.1: Panel en el que se pega el papel kraft y respuestas.

2.1.3. Técnicas

2.1.3.1. Técnica del Collage

El moderador le indica a los participantes que tienen un tiempo X para responder la pregunta en las tarjetas y que utilicen frases cortas. Por lo general el tiempo acordado es muy breve, ya que es un recurso muy importante de los participantes, y también se busca que respondan sobre la primera idea que se le viene a la cabeza.

Las tarjetas con respuestas similares son agrupadas, a esta acción se le conoce como *cluster*, y tiene que ser advertida por parte del moderador a los participantes. Además las respuestas pueden ser ordenadas posicionando más arriba las que tienen más apoyo, es decir, cuando ya se ha usado *cluster* (Metaplan, b).

2.1.3.2. Técnica de la Visualización Permanente

Todas las respuestas son escritas en *tarjetas de cartulina*, las cuales son ubicadas en el *panel* que estará a la vista de todos y en todo momento, esto permite que los participantes lleven la discusión más fácilmente (Metaplan, b).

Las ventajas de ésta técnica son:

- Ayuda a tener ópticamente presente el contenido del taller, esto facilita e incrementa la memorización de la información (Torres, 2007).
- Permite la documentación del taller, ya que al finalizar, se fijan las respuestas utilizando *cinta adhesiva* sobre el *papel kraft*, y se procede a fotografiarlo, doblarlo o enrollarlo.

2.1.3.3. Técnica de las Preguntas

Es el medio de intervención más eficiente de cualquier proceso de moderación grupal, ya que estimula la reflexión, el pensamiento y la búsqueda de opciones y facilitan e impulsan el trabajo grupal (Torres, 2007).

2.2. Estrategias de Moderación Grupal

El trabajo colaborativo se basa en una fuerte relación de interdependencia entre los diferentes miembros que lo conforman, donde es fundamental la participación activa de todos los miembros de forma cooperativa y abierta hacia el intercambio de ideas dentro del grupo (Gonzalez et al., 2011).

Algunas de las principales técnicas que favorecen el trabajo colaborativo son: trabajo en parejas, lluvia de ideas, rueda de ideas, votación, valoración de decisiones, debate y foro, subgrupos de discusión, controversia estructurada, grupos de investigación, juegos de rol, estudio de casos y trabajo por proyectos (Gonzalez et al., 2011).

2.3. Computación Móvil

La computación Móvil es el uso de cualquier dispositivo portátil que haga uso de la computación. Hoy en día, los dispositivos que más se utilizan son las tablets, smartphones y notebooks, pero también existen relojes, lentes inteligentes, entre otros.

Los smartphones y tablets, son dispositivos móviles que realizan actividades similares a las de un computador pequeño. Su característica principal es su pantalla táctil con la que se interactúa sin necesidad de un mouse o teclado físico, ya que cuenta con un teclado virtual que aparece en pantalla. La diferencia radica en que los smartphone o teléfonos inteligentes cuentan con conectividad telefónica y caben en un bolsillo o cartera, a diferencia de las tablets o tabletas.

En ambos casos, en los smartphones y tablets, los Sistemas Operativos (SO) están dominados por los dispositivos con Android, iOS (iPhone SO) y Windows Phone en menor medida (Dávalos, 2012).

La tecnología ha dado grandes avances en los últimos años, en especial en la microtecnología. Cada vez que se crea algo, el siguiente paso es hacerlo más pequeño y menos costoso, sin evitar que cumpla su función original. Esto ha abierto el camino de la computación en muchos ámbitos y ha hecho que la computación móvil esté presente en todas partes. Con el paso del tiempo, la computación portátil está resolviendo tareas en distintos ámbitos, que

antiguamente su solución era muy compleja y onerosa dado a que se necesitaban dispositivos adicionales, que hoy ya vienen inmersos en estos dispositivos móviles (Dávalos, 2012).

La computación móvil se soporta en tres pilares básicos:

1. Las redes inalámbricas.
2. La portabilidad (dispositivos móviles).
3. La movilidad (usuarios).

Existen 4 componentes o elementos a considerar:

1. El usuario móvil.
2. El dispositivo móvil.
3. La red móvil (red inalámbrica).
4. El Software (aplicación móvil).

A continuación se explicarán estos 4 componentes.

2.3.1. El Usuario Móvil.

El Usuario Móvil se diferencia del usuario estático o fijo, el cual hace uso de computadores de escritorio y estaciones de trabajo. Sus características son:

- Está en movimiento.
- Es multi-tarea, ya que generalmente no está enfocado únicamente en una única aplicación.
- Demanda tiempos cortos de respuesta.
- Cambia de tarea o de actividad muy rápidamente.
- Demanda acceso a la información en cualquier momento y en cualquier lugar, *anytime anywhere*.

2.3.2. Dispositivos Móviles.

Los dispositivos móviles como las tablets y smartphones, se caracterizan por ser de un pequeño tamaño y por no necesitar de accesorios como teclado y mouse, ya que poseen pantalla táctil que trae incorporado un teclado virtual. Algunas de las principales características que poseen los dispositivos móviles hoy en día son:

- Mayor duración de las baterías.
- Pantalla de mayor resolución y táctil a la vez.
- Ser de bajo costo y de gran poder.
- Teclados virtuales, interfaces *touch screen* y comandos de voz mejorados.
- Conectividad a través de redes como Bluetooth, GSM/GPRS y wifi.
- Aditamentos como cámara (resolución mejorada), GPS y giroscopio (sensores).
- Múltiples servicios/aplicaciones pre-instaladas.

2.3.3. Las Redes Inalámbricas.

Según Dávalos (2012) la clasificación más usada es la que considera el alcance de las redes inalámbricas:

1. Redes PAN (Personal Area Networks) o WPAN.
2. Redes de Área Local (LAN) o WLAN.
3. Redes de Área metropolitana (MAN) o WMAN.
4. Redes de Área Extensa (WAN) o WWAN.

Como se aprecia en la Figura 2.2, para el caso de dispositivos móviles con Bluetooth y wifi, la conectividad está en un rango de hasta 10 y 30 mts. respectivamente.

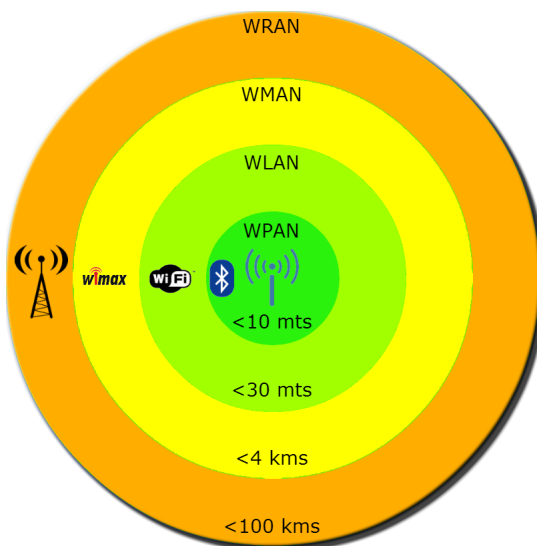


Figura 2.2: Clasificación de las redes inalámbricas.

2.3.4. Aplicación Móvil.

Es un programa gratuito o pagado que se ejecuta en los dispositivos móviles como smartphones o tablets y por lo general se encuentran disponibles a través de plataformas de distribución, asociada a cada sistema operativo móvil como Android, iOS, Windows Phone, entre otros.

El avance en los dispositivos móviles y las redes inalámbricas propició el surgimiento de nuevas de aplicaciones, que pudieran suplir las necesidades de los usuarios en movimiento.

2.4. Redes Ad Hoc

Es una red inalámbrica formada sin ninguna administración central, es decir, que no depende de una infraestructura pre-existente como routers (Frodigh et al., 2000).

En este tipo de red los nodos, ya sean smartphone o tablet, pueden servir de encaminadores y de anfitriones, con lo que pueden enviar paquetes en nombre de otros nodos (Frodigh et al., 2000).

Los nodos son libres de asociarse con cualquier otro dispositivo de red ad hoc en el rango de enlace (Frodigh et al., 2000).

2.5. Android

Android es un sistema operativo para dispositivos móviles como smartphones y tablets, pero también para dispositivos como relojes inteligentes, lentes inteligentes y dispositivos para televisores.

2.5.1. Sus Origenes

Fue desarrollado inicialmente por Android Inc., una pequeña compañía, que acababa de ser creada, orientada a la producción de aplicaciones para terminales móviles y que fue comprada por Google en 2005 (Gironés, 2012). En el año 2007 se crea el consorcio Handset Alliance, formada por Google y otras 77 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. El año 2008 aparece el primer móvil con Android (T-Mobile G1). Además se abre Android Market, para la descarga de aplicaciones, el cual es reemplazado por Play Store el 2012, que es un portal que unifica tanto la descarga de aplicaciones como de contenidos.

2.5.2. Versiones de Android

Las diferentes versiones de Android están asociadas a su vez con el nivel de la API y el nombre comercial. El nivel de API corresponde a números enteros comenzando desde 1, y es utilizada por los programadores para definir la versión para la que estará disponible una aplicación Android. Para los nombres comerciales se han elegido postres en orden alfabético, comenzado por la letra C desde la versión 1.5 (véase Tabla 2.1).

NOMBRE COMERCIAL	FECHA ORIGEN	VERSIÓN	NIVEL DE API	CARACTERÍSTICA
—	09/2008	1.0	1	Nunca se utilizó comercialmente
—	02/2009	1.1	2	Solo se corrigieron errores
Cupcake	04/2009	1.5	3	Teclado por pantalla, grabación de audio y video
Donut	09/2009	1.6	4	Capacidades de búsqueda avanzada en todo el dispositivo
Éclair	10/2009	2.0	5	Bluetooth 2.1
Éclair	01/2010	2.1	7	Reconocimiento de voz
Froyo	05/2010	2.2	8	Mejora la velocidad de ejecución de las aplicaciones
Gingerbread	12/2010	2.3	9	Soporta mayores tamaños de pantalla y resoluciones, debido a las tablets
Honeycomb	02/2011	3.0	11	Versión optimizada para tablets.
Honeycomb	05/2011	3.1	12	Permite manejar dispositivos conectados por USB
Honeycomb	07/2011	3.2	13	Zoom compatible para aplicaciones de tamaño fijo
Ice Cream Sandwich	10/2011	4.0	14	Se unifican las dos versiones anteriores (2.x para teléfonos y 3.x para tablets) en una sola compatible con cualquier tipo de dispositivo.
Ice Cream Sandwich	12/2011	4.0.3	15	Se introducen mejoras incluyendo el de redes sociales, calendario, revisor ortográfico, texto a voz y bases de datos
Jelly Bean	07/2012	4.1	16	Mejora la fluidez del interfaz de usuario. Se incorporan técnicas, como: sincronismo vertical, triple búfer y aumentar la velocidad del procesador al tocar la pantalla
Jelly Bean	11/2012	4.2	17	Varias cuentas de usuario en el mismo dispositivo
Jelly Bean	07/2013	4.3	18	Se mejora la seguridad para gestionar y ocultar las claves y credenciales.
KitKat	10/2013	4.4	19	Soporta más dispositivos, incluyendo aquellos con tamaños de memoria RAM de solo 512 MB
Lollipop	11/2014	5.0	21	Soporta nuevas plataformas, incluyendo Google Wear, Google TV y Google Card

Tabla 2.1: Listado de versiones Android.

2.5.3. Android Studio

Android Studio es un entorno de desarrollo que ofrece características y mejoras con respecto a Eclipse ADT y es el IDE oficial de Android, actualmente se encuentra disponible en la versión 1.3.0. A continuación se encuentran algunas de las características que éste ofrece:

- Sistema de construcción a base de Gradle flexible.
- Construir variantes y múltiples en APK.
- Plantilla de apoyo ampliado para los servicios de Google y varios tipos de dispositivos.
- Un gran editor de diseño, con soporte para edición.
- Herramientas que facilitan su uso y la compatibilidad de versiones.
- ProGuard y aplicación de firma de capacidades.
- El soporte integrado para Google Cloud Platform, lo que facilita la integración de Google Cloud Mensajería y App Engine.

Capítulo 3

Estado del arte

3.1. Metaplan Virtual

Es un proyecto en desarrollo por la Universidad Nacional de La Plata, Argentina, que busca lograr una aplicación web para Metaplan, como se aprecia en la Figura 3.1.

Ésta aplicación será de manejo intuitivo, sin necesidad de instalación de software adicional, con la funcionalidad necesaria para poder efectuar las etapas de creación de un curso, generar la división de participantes en subgrupos, construir el debate de cada subgrupo, incorporar la creación de las nubes de idea y el diseño de la lista de recomendaciones. El prototipo presenta una interfaz de administración, una de modo de diseño para el moderador y una en modo alumno.

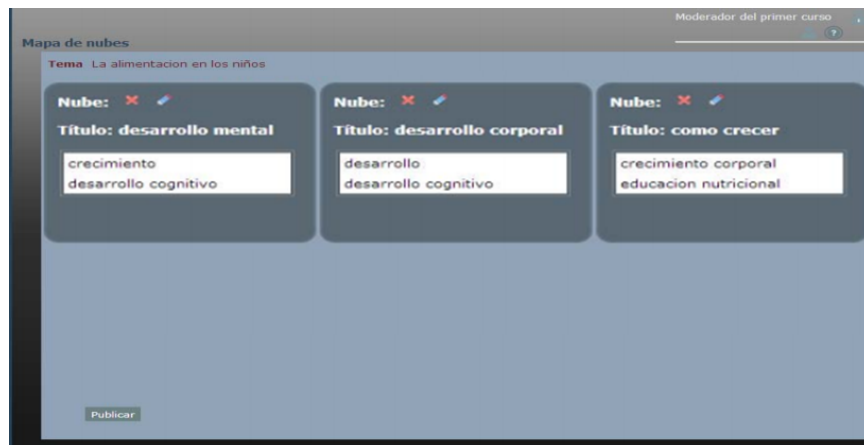


Figura 3.1: Prototipo de Metaplan Virtual, obtenido de unlp.edu.ar.

3.2. Trello

Trello es una herramienta colaborativa de gestión, que ayuda a organizar proyectos en tableros, está disponible como aplicación web y aplicación móvil. Su interfaz permite visualizar su contenido de una manera ordenada, además se puede interactuar con los tableros y tarjetas, moviéndolos y organizándolos a gusto (ver Figura 3.2).

A pesar de no estar diseñada para realizar estrategias de moderación grupal, su interfaz se asemeja mucho a la esperada para realizar Metaplan.



Figura 3.2: Metaplan en Trello.

Capítulo 4

Especificación de Requerimientos

4.1. Introducción

En el presente capítulo se dará a conocer los servicios, restricciones y funcionalidades de la aplicación móvil “Metaplandroid”, siguiendo el estándar IEEE-STD-830-1998. Con el fin de generar un documento para la realización de futuras actualizaciones y/o para una mayor comprensión del sistema.

4.2. Definiciones, Acrónimos y Abreviaturas

- **“Play Store”**: plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, así como una tienda en línea desarrollada y operada por Google que permite a los usuarios navegar y descargar aplicaciones.
- **“Red wifi”**: red inalámbrica que permite la conexión de dispositivos en una área determinada.
- **“Dirección IP”**: etiqueta numérica que identifica un dispositivo dentro de una red (ejemplo: 192.168.1.10).
- **“Cluster”**: acción realizada por el moderador, para agrupar respuestas similares.

4.3. **Ámbito del Sistema**

El propósito de la aplicación móvil Metaplandroid, cuyas funcionalidades se describen en el presente documento, es realizar talleres de Metaplan, sin la necesidad de materiales físicos. Actualmente no existen aplicaciones móviles capaces de llevar a cabo esta tarea. La implementación de Metaplandroid permitirá realizar talleres en una misma red wifi, en la que el moderador deberá dar su dirección IP, con el fin de que los participantes se conecten con él, y así poder realizar el taller. Además permitirá enviar mediante un correo electrónico los resultados obtenidos al correo que indique el moderador.

La aplicación no estará en “Play Store”, ya que para poder hacer esto, hay que pagar, lo que está fuera de los límites de este trabajo.

4.4. **Descripción General**

4.4.1. **Característica de los Usuarios**

Existen dos tipos de usuarios:

- **Moderador:** Persona encargada del taller y de los participantes, es quien crea el taller y las preguntas. Inicialmente su primera labor es informar a los participantes que se conecten a su misma “red wifi”, una vez iniciada la aplicación debe decirles su dirección IP, para que ellos la digiten y logren conectarse con él. En el transcurso del taller, debe *comunicarles* a los participantes cuando haya generado una nueva pregunta o haya liberado las respuestas para la visualización. Por último finaliza el taller y envía un correo electrónico con los resultados.
- **Participante:** Personas que asisten al taller, son aquellos que responden las preguntas y que tienen relación directa o indirecta sobre los temas que se tratan en el taller de Metaplan.

4.4.2. **Restricciones**

Las restricciones del sistema son las siguientes:

- La aplicación está diseñada para ser utilizada en dispositivos móviles con sistema operativo Android a partir de la versión 4.0.3. hasta la versión 4.4.4.

- Para la comunicación de los dispositivos es necesario estar en una misma “red wifi”.
- Para lograr una mayor concentración de los participantes y mejor comunicación de los dispositivos, es recomendable que todos los integrantes se encuentren en una misma habitación, en la que no exista ruidos o distracciones que afecten en la participación de los asistentes.

4.5. Requisitos Específicos

4.5.1. Funcionalidades

- **R1.- Seleccionar perfil moderador.**

Descripción: Este requerimiento permite conectarse a la aplicación como moderador.

Entrada: Nombre del moderador.

Proceso: Se inicia la aplicación que lleva a la pantalla *Seleccionar Perfil* en la que se pulsa sobre el botón MODERADOR. La aplicación verifica que el dispositivo esté conectado a una red wifi, si no lo está despliega un mensaje de error. Luego, el moderador debe ingresar su nombre y pulsar clic en el botón COMENZAR.... Posteriormente el moderador podrá crear talleres y preguntas.

Salida 1: Id moderador, nombre del moderador. Luego se pasará a la pantalla siguiente, donde podrá generar talleres.

Salida 2: Si el moderador no está conectado a una red wifi, se desplegará el mensaje de error: “*Compruebe su conexión wifi*”.

Salida 3: Si el moderador está conectado a una red wifi, pero no posee una dirección IP, se desplegará el mensaje de error: “*Reinicie wifi (No tiene IP)*”.

Salida 4: Si el moderador no ingresa un nombre se desplegará el mensaje de error: “*Ingrese su Nombre*”.

- **R2.- Seleccionar perfil participante**

Descripción: Este requerimiento permite conectarse a la aplicación como participante. Es necesario estar conectado a la misma red wifi del moderador.

Entrada: Nombre del participante, dirección IP del moderador.

Proceso: Se inicia la aplicación que lleva a la pantalla *Seleccionar Perfil* en la que se

pulsa sobre el botón PARTICIPANTE. La aplicación verifica que el dispositivo esté conectado a una red wifi, si no lo está despliega un mensaje de error. Luego, el participante debe ingresar su nombre, ingresar la dirección IP del moderador y pulsa el botón CONECTAR..., si no sabe la dirección IP del moderador, se la debe solicitar verbalmente. Posteriormente el participante podrá ingresar al taller, a las preguntas y responderlas.

Salida 1: Id participante, nombre participante. Luego se pasará a la pantalla siguiente, donde podrá ingresar a un taller.

Salida 2: Si el participante no está conectado a una red wifi, se desplegará el mensaje de error: *“Compruebe su conexión wifi”*.

Salida 3: Si el participante está conectado a una red wifi, pero no posee una IP, se desplegará el mensaje de error: *“Reinicie wifi (No tiene IP)”*.

Salida 4: Si el participante no ingresa un nombre y pulsa en conectar, se desplegará el mensaje de error: *“Ingrese su Nombre”*.

Salida 5: Si el participante no ingresa la dirección IP del moderador y pulsa en conectar, se desplegará el mensaje de error: *“Ingrese IP del moderador”*.

■ R3.- Ingresar taller.

Descripción: Este requerimiento le permite al moderador crear un taller al sistema.

Entrada: Nombre del taller, descripción del taller.

Proceso: Conectado como moderador, se debe dar clic en el botón CREAR TALLER, se generará una ventana emergente en la que se debe ingresar el nombre del taller, descripción del taller si se desea y pulsar en el botón GUARDAR, los datos serán almacenados en la memoria del dispositivo. Luego de crearlo se desplegará el taller en la pantalla.

Salida 1: Id taller, id creador, nombre del taller, descripción del taller, fecha creación taller.

Salida 2: Si el moderador no ingresa el nombre del taller, se desplegará el mensaje de error: *“Ingrese el Nombre del Taller”*.

■ R4.- Consultar talleres.

Descripción: Este requerimiento le permite al moderador y participante desplegar los talleres creados, en caso de no verlos.

Entrada: Id taller.

Proceso: Una vez conectado y si no se visualiza el taller, se presiona el botón ACTUALIZAR y se despliegan todos los talleres que ha generado el moderador.

Salida: Nombre del taller, descripción del taller.

■ **R5.- Modificar datos del taller.**

Descripción: Este requerimiento le permite al moderador la modificación de un taller.

Entrada: Id taller, nombre del taller modificado, descripción del taller modificado.

Proceso: Conectado como moderador, se debe mantener el clic sobre el taller, luego se desplegará una ventana emergente que permitirá modificar el nombre y la descripción del taller, finalmente sólo se debe dar clic en el botón GUARDAR.

Salida: Id taller, nombre del taller modificado, descripción del taller modificado.

■ **R6.- Ingresar pregunta a un taller.**

Descripción: Este requerimiento le permite al moderador ingresar una pregunta en el taller.

Entrada: Id taller, la pregunta, descripción de la pregunta.

Proceso: Conectado como moderador y ya adentro de un taller, se debe dar clic en el botón PREGUNTAR, se desplegará una ventana emergente en la que se debe ingresar la pregunta, descripción de la pregunta si se desea y pulsar en el botón GUARDAR, los datos serán almacenados en la memoria del dispositivo. Luego de crearla se desplegará la pregunta en la pantalla.

Salida 1: Id pregunta, id taller, la pregunta, descripción de la pregunta, fecha creación pregunta.

Salida 2: Si el moderador no ingresa la pregunta, se desplegará el mensaje de error: *“Ingrese la Pregunta”*.

■ **R7.- Consultar preguntas.**

Descripción: Este requerimiento le permite al moderador y participante desplegar las preguntas de un taller, en caso de no verlos.

Entrada: Id taller, id pregunta.

Proceso: Una vez dentro del taller y si no se visualiza la pregunta, se presiona el botón ACTUALIZAR y se despliegan todas las preguntas del taller.

Salida: Las preguntas, descripciones de las preguntas.

■ **R8.- Modificar pregunta.**

Descripción: Este requerimiento le permite al moderador la modificación de una pregunta.

Entrada: Id pregunta, la pregunta modificada, descripción de la pregunta modificada.

Proceso: Conectado como moderador, se debe mantener el clic sobre la pregunta, luego se desplegará una ventana emergente que permitirá modificar la pregunta y la descripción, finalmente sólo se debe dar clic en el botón GUARDAR.

Salida: Id pregunta, La pregunta modificada, descripción de la pregunta modificada.

■ **R9.- Ingresar respuesta a una pregunta.**

Descripción: Este requerimiento le permite a los participantes ingresar respuestas a las preguntas.

Entrada: Id pregunta, La respuesta.

Proceso: Conectado como participante y ya adentro de la pregunta, se debe dar clic en el botón RESPONDER, se desplegará una ventana emergente en la que debe ingresar la respuesta y pulsar en el botón GUARDAR, los datos serán almacenados en la memoria del dispositivo. Luego de crearla se desplegará la respuesta en la pantalla.

Salida 1: Id respuesta, id pregunta, id participante, la respuesta, fecha creación respuesta.

Salida 2: Si el participante no ingresa la respuesta, se desplegará el mensaje de error: *“Ingrese la Respuesta”*.

■ **R10.- Liberar respuestas.**

Descripción: Este requerimiento permite enviar las respuestas de una pregunta a todos participantes.

Entrada: Id pregunta, id respuesta.

Proceso: Conectado como moderador y ya adentro de la pregunta, se debe esperar a que todos los participantes hayan respondido a la pregunta, una vez ocurrido esto, se debe dar clic en el botón LIBERAR. De esta forma se comparte las respuestas de todos los participantes.

Salida: Id respuesta, id pregunta, id participante, las respuestas.

■ **R11.- Consultar respuestas.**

Descripción: Este requerimiento le permite al moderador y participantes desplegar las respuestas de una pregunta, en caso de no verlas.

Entrada: Id pregunta, id respuesta.

Proceso: Una vez dentro de la pregunta y luego de que el moderador haya liberado las respuestas (R10.- Liberar respuestas), se presiona el botón ACTUALIZAR y se despliegan las respuestas de todos los participantes.

Salida: Las respuestas.

■ **R12.- Modificar respuesta.**

Descripción: Este requerimiento le permite al moderador y participante la modificación de una respuesta.

Entrada: Id pregunta, id respuesta, la respuesta modificada.

Proceso: Se debe mantener el clic sobre la respuesta, luego se desplegará una ventana emergente que permitirá modificar la respuesta, finalmente sólo se debe dar clic en el botón GUARDAR.

Salida: Id respuesta, la respuesta modificada.

■ **R13.- Eliminar respuesta.**

Descripción: Este requerimiento le permite al moderador la eliminación de una respuesta.

Entrada: Id respuesta

Proceso: Se debe mantener el clic sobre la respuesta, luego se desplegará una ventana emergente en la que se debe dar clic en el botón ELIMINAR.

Salida: —

■ **R14.- Consultar Participantes.**

Descripción: Este requerimiento permite desplegar los participantes conectados.

Entrada: Id taller

Proceso: Conectado como moderador, se presiona el botón ¿QUIÉN ESTÁ?, y se desplegará una lista con los participantes que han ingresado.

Salida: Una ventana emergente con los nombres de los participantes.

■ **R15.- Finalizar taller.**

Descripción: Este requerimiento permite finalizar el taller y a la vez enviar por correo electrónico toda la información.

Entrada: Id taller, estado taller

Proceso: Conectado como moderador, se debe mantener el clic sobre el taller, se desplegará una ventana emergente en la que se debe dar clic en el botón FINALIZAR.

Salida: Id taller, estado taller. Aparece una ventana en la que se solicita seleccionar método de envío para generar el correo electrónico. Luego sólo se debe ingresar el correo de origen y de destinatario; el cuerpo del correo electrónico ya está generado.

Capítulo 5

Diseño

En este capítulo, se presenta de manera gráfica el funcionamiento de la aplicación “Metaplandroid”.

5.1. Diagrama de Actividad

El siguiente diagrama de actividad (ver Figuras 5.1 y 5.2), muestra desde el punto de inicio hasta el punto final todas las actividades por las que se puede pasar, lo que permite ver más detalladamente los procesos que se pueden realizar en los distintas etapas del sistema.

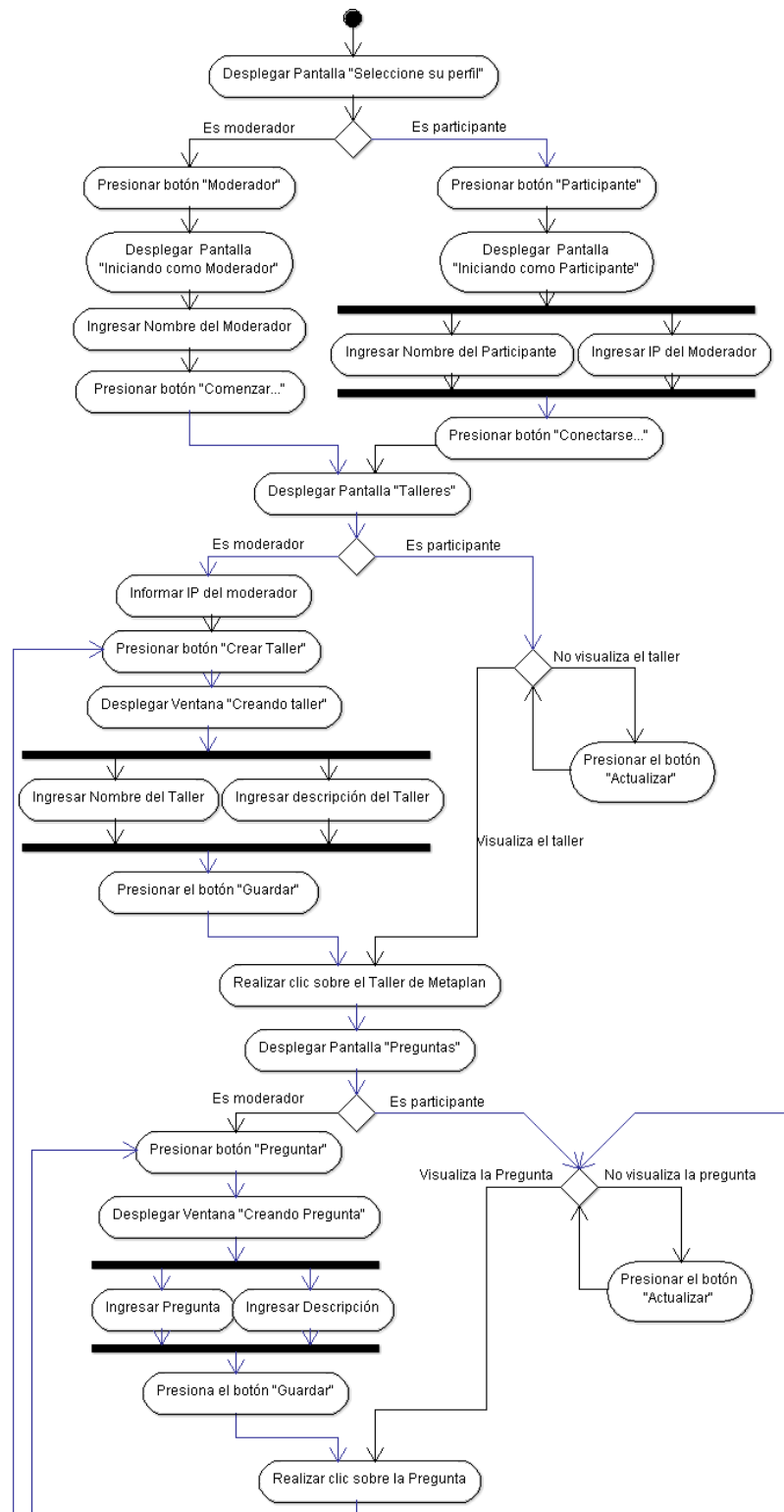


Figura 5.1: Parte 1 del Diagrama de Actividad del sistema.

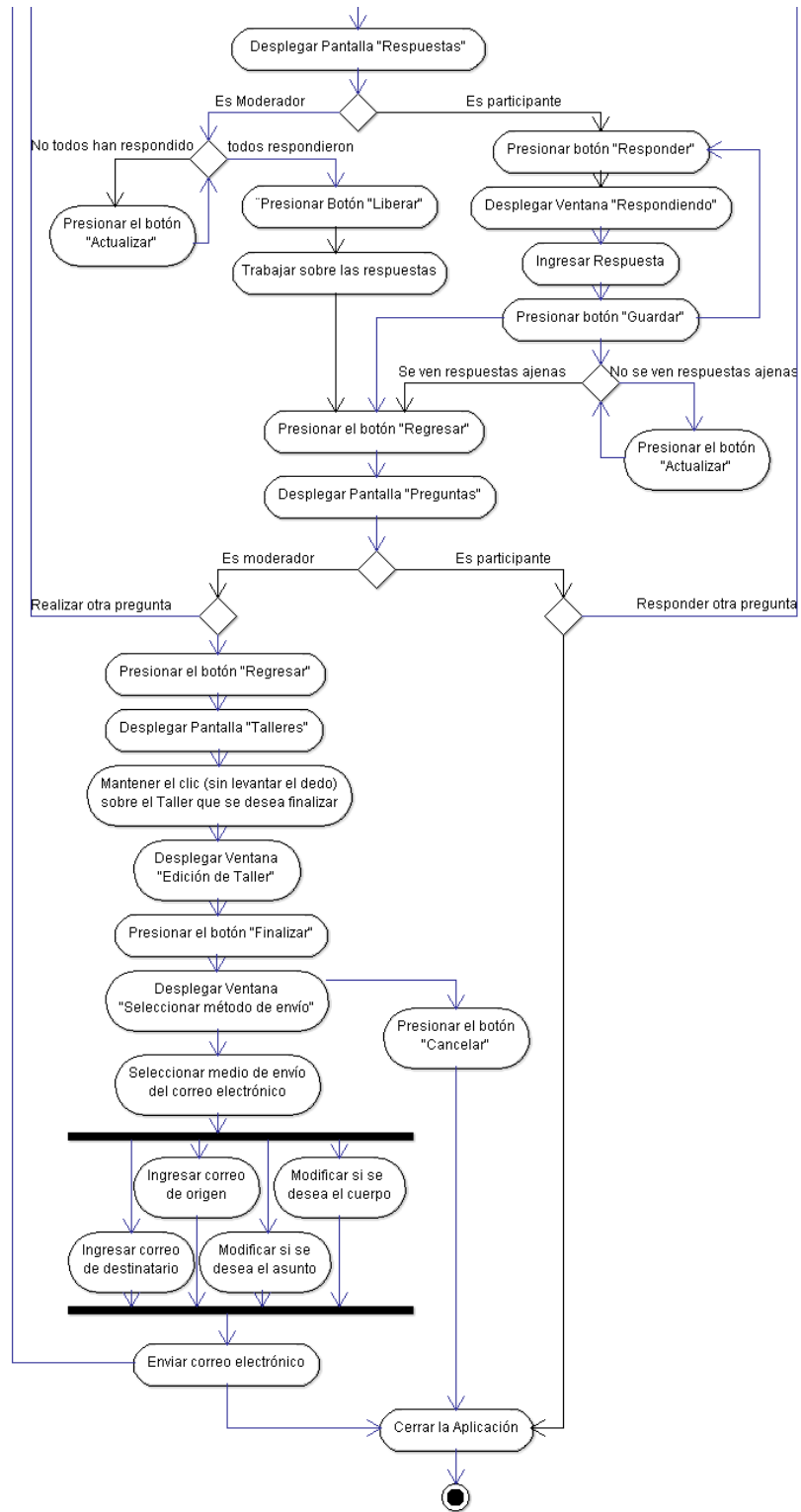


Figura 5.2: Parte 2 del Diagrama de Actividad del sistema.

5.1.1. Diagrama de Actividad - Ingresar Moderador

La Figura 5.3, muestra en orden las actividades del requisito funcional *Ingresar Moderador*.

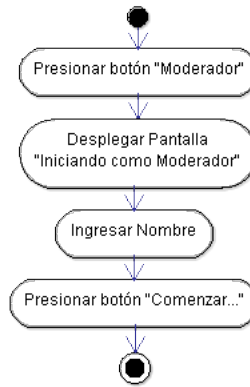


Figura 5.3: Diagrama de actividad del requisito funcional Ingresar Moderador.

5.1.2. Diagrama de Actividad - Ingresar Participante

La Figura 5.4, muestra en orden las actividades del requisito funcional *Ingresar Participante*.

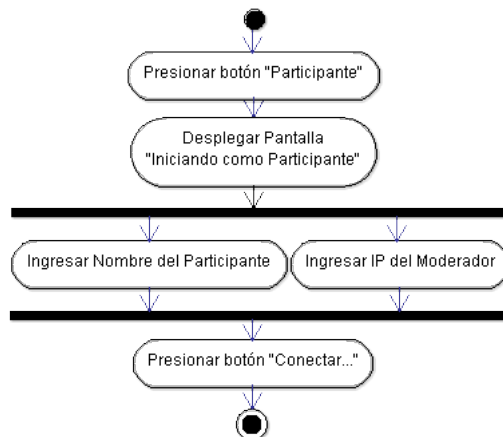


Figura 5.4: Diagrama de actividad del requisito funcional Ingresar Participante.

5.1.3. Diagrama de Actividad - Ingresar Taller

La Figura 5.5, muestra en orden las actividades del requisito funcional *Ingresar Taller*.

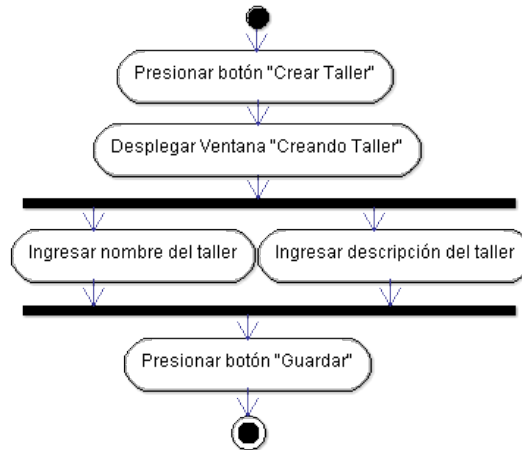


Figura 5.5: Diagrama de actividad del requisito funcional Ingresar Taller.

5.1.4. Diagrama de Actividad - Consultar Talleres

La Figura 5.6, muestra en orden las actividades del requisito funcional *Consultar Talleres*.

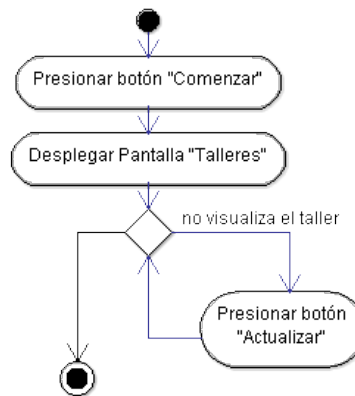


Figura 5.6: Diagrama de actividad del requisito funcional Consultar Talleres.

5.1.5. Diagrama de Actividad - Modificar Taller

La Figura 5.7, muestra en orden las actividades del requisito funcional *Modificar Taller*.

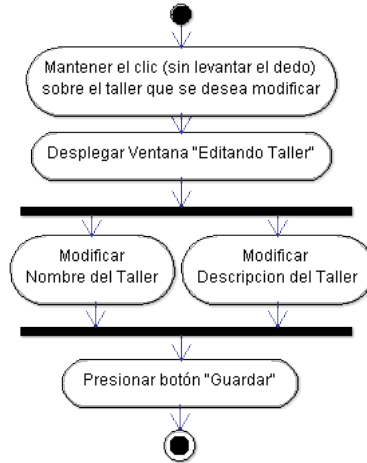


Figura 5.7: Diagrama de actividad del requisito funcional Modificar Taller.

5.1.6. Diagrama de Actividad - Ingresar Pregunta

La Figura 5.8, muestra en orden las actividades del requisito funcional *Modificar Taller*.

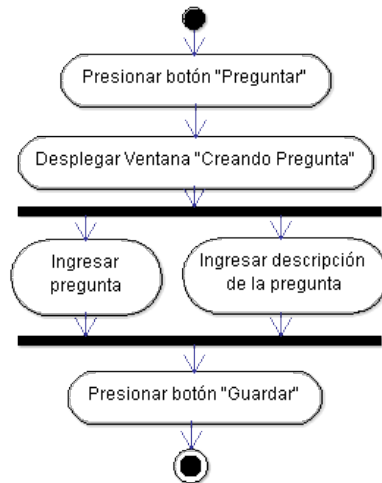


Figura 5.8: Diagrama de actividad del requisito funcional Ingresar Pregunta.

5.1.7. Diagrama de Actividad - Consultar Preguntas

La Figura 5.9, muestra en orden las actividades del requisito funcional *Consultar Preguntas*.

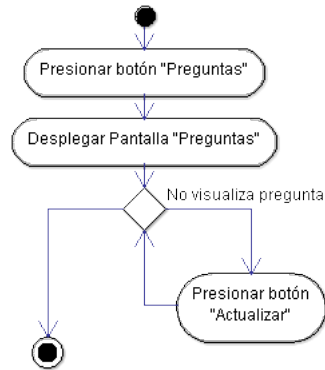


Figura 5.9: Diagrama de actividad del requisito funcional Consultar Preguntas.

5.1.8. Diagrama de Actividad - Modificar Pregunta

La Figura 5.10, muestra en orden las actividades del requisito funcional *Modificar Pregunta*.

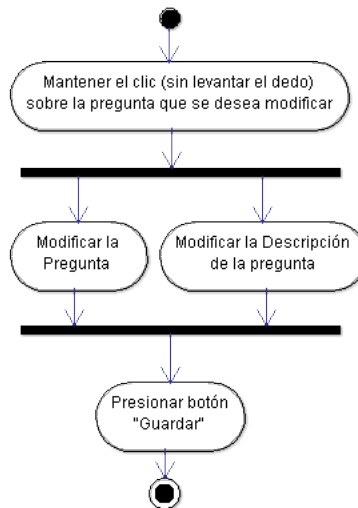


Figura 5.10: Diagrama de actividad del requisito funcional Modificar Pregunta.

5.1.9. Diagrama de Actividad - Ingresar Respuesta

La Figura 5.11, muestra en orden las actividades del requisito funcional *Ingresar Respuesta*.

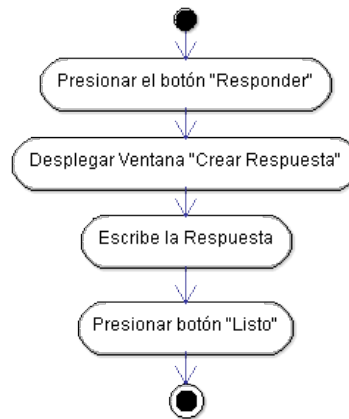


Figura 5.11: Diagrama de actividad del requisito funcional Ingresar Respuesta.

5.1.10. Diagrama de Actividad - Liberar Respuestas

La Figura 5.12, muestra en orden las actividades del requisito funcional *Liberar Respuestas*.

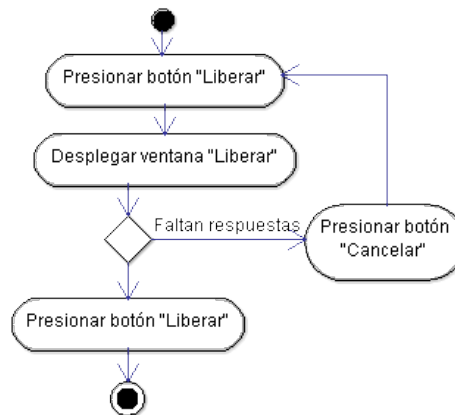


Figura 5.12: Diagrama de actividad del requisito funcional Liberar Respuesta.

5.1.11. Diagrama de Actividad - Consultar Respuestas

La Figura 5.13, muestra en orden las actividades del requisito funcional *Consultar Respuesta*.

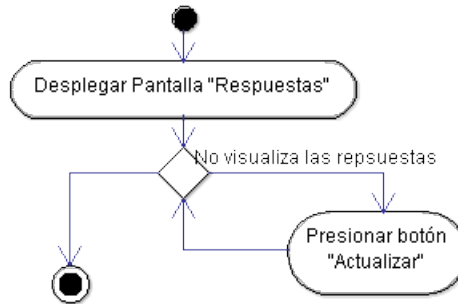


Figura 5.13: Diagrama de actividad del requisito funcional Consultar Respuesta.

5.1.12. Diagrama de Actividad - Modificar Respuesta

La Figura 5.14, muestra en orden las actividades del requisito funcional *Modificar Respuesta*.

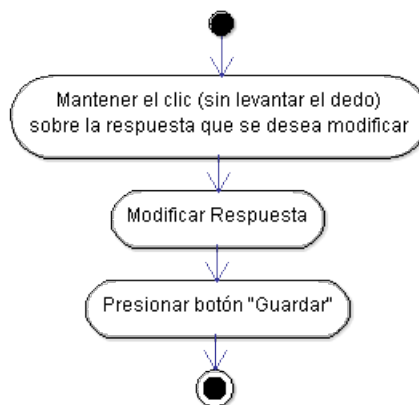


Figura 5.14: Diagrama de actividad del requisito funcional Modificar Respuesta.

5.1.13. Diagrama de Actividad - Eliminar Respuesta

La Figura 5.15, muestra en orden las actividades del requisito funcional *Eliminar Respuesta*.

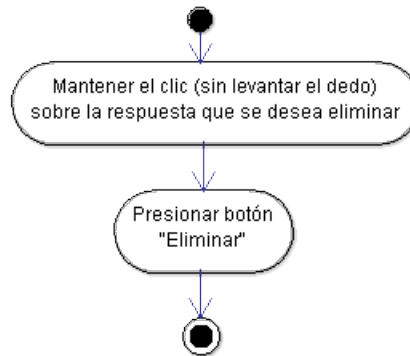


Figura 5.15: Diagrama de actividad del requisito funcional Eliminar Respuesta.

5.1.14. Diagrama de Actividad - Consultar Participantes

La Figura 5.16, muestra en orden las actividades del requisito funcional *Consultar Participantes*.



Figura 5.16: Diagrama de actividad del requisito funcional Consultar Participantes.

5.1.15. Diagrama de Actividad - Finalizar Taller

La Figura 5.17, muestra en orden las actividades del requisito funcional *Finalizar Taller*.

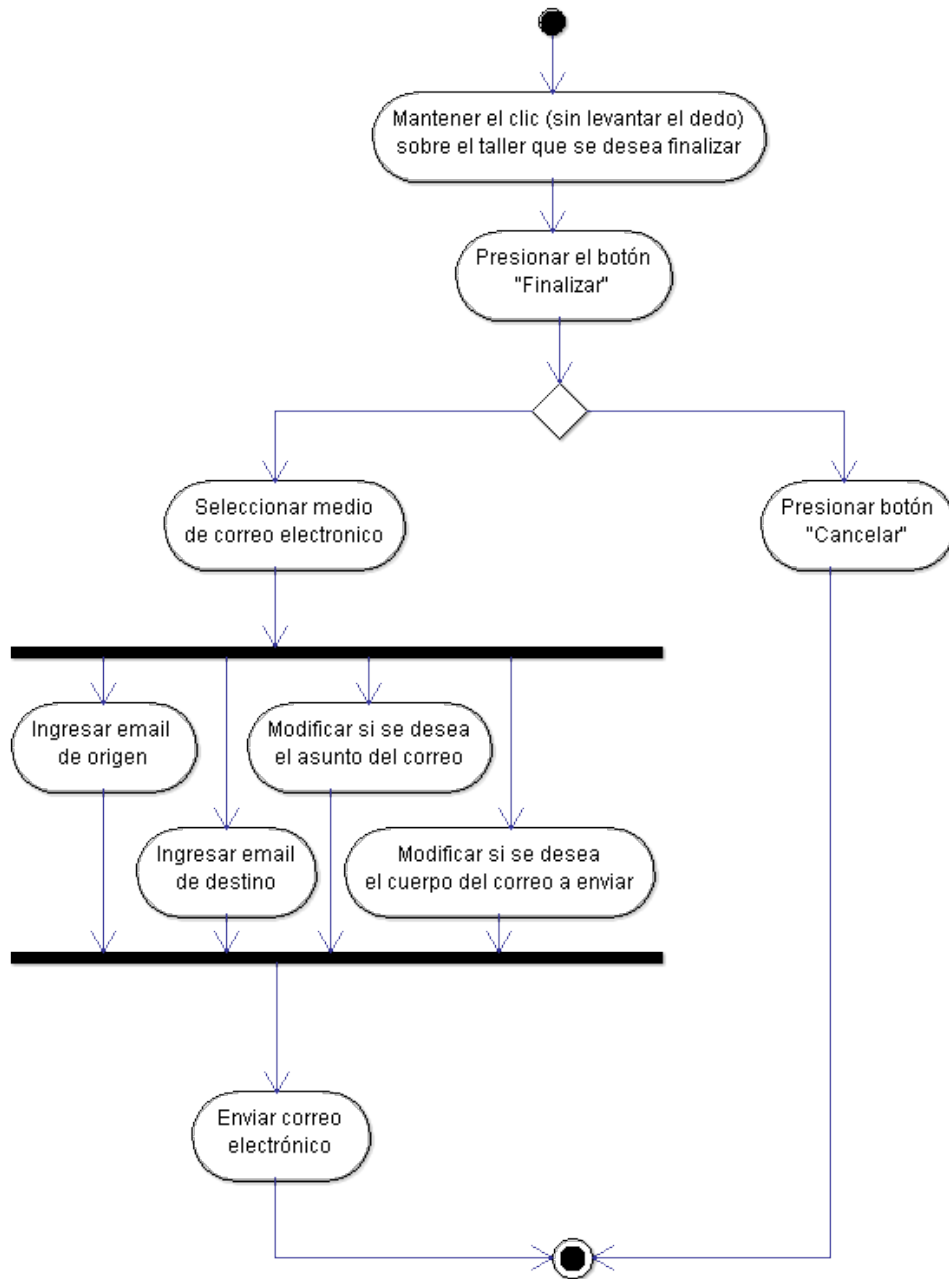


Figura 5.17: Diagrama de actividad del requisito funcional Finalizar Taller.

5.2. Diagrama de Componentes

El diagrama de componentes en UML (ver Figura 5.18) representa cómo el sistema “Metaplandroid” está dividido en componentes y muestra las dependencias entre estos componentes.

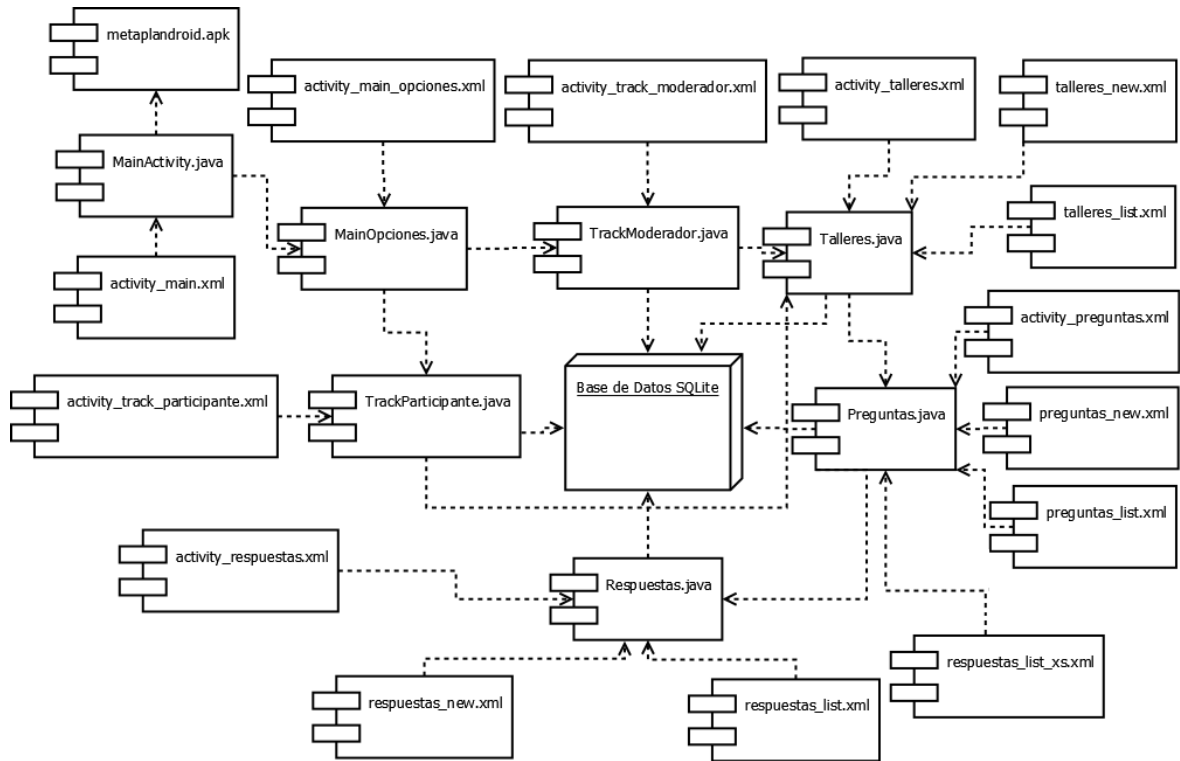


Figura 5.18: Diagrama de Componentes del sistema.

5.3. Interfaces de Usuario

En la Figura 5.19 se puede visualizar la pantalla inicial de la aplicación, en donde el moderador debe dar clic en el botón MODERADOR y el participante debe dar clic en el botón PARTICIPANTE.

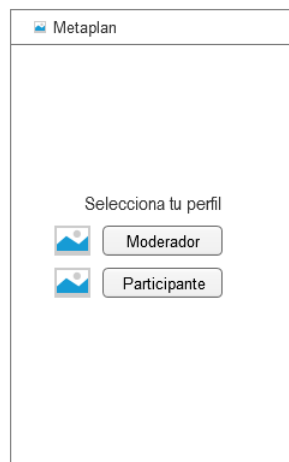


Figura 5.19: Wireframe *Selecciona tu perfil*.

La pantalla a utilizar será: (Véase Figura 5.20)



Figura 5.20: Selecciona tu perfil.

En la Figura 5.21 se puede visualizar la pantalla *Iniciando como Moderador* que aparece luego de dar clic en el botón MODERADOR de la Figura 5.20. El moderador debe asegurarse de que todos los participantes estén conectados a su misma red wifi, luego debe ingresar su nombre en el primer campo, y presionar clic en el botón CONECTAR.... En la misma pantalla se visualizan las INSTRUCCIONES que inicialmente están ocultas, y se vuelven visibles dando clic en el icono de ayuda (?).

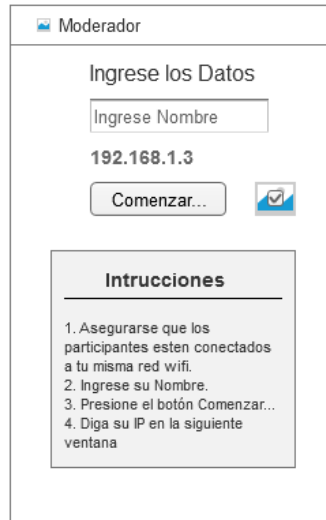


Figura 5.21: Wireframe *Iniciando como Moderador*.

La pantalla a utilizar será: (Véase Figura 5.22)

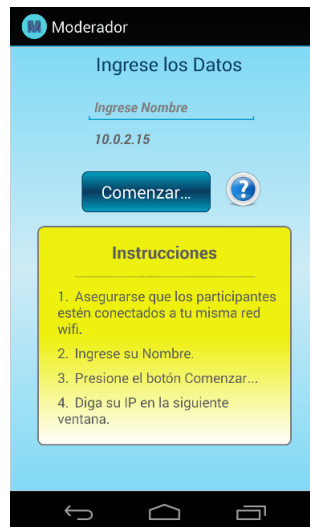
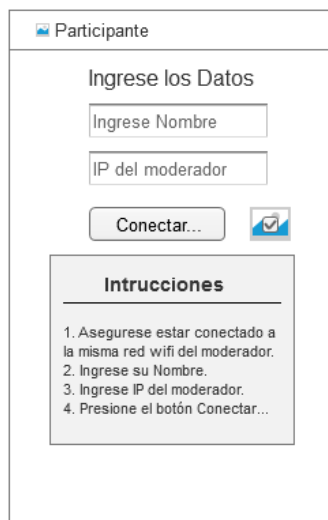


Figura 5.22: *Iniciando como Moderador*.

En la Figura 5.23 se puede visualizar la pantalla *Iniciando como Participante*, que aparece luego de dar clic en el botón PARTICIPANTE de la Figura 5.20. El participante debe ingresar su nombre en el primer campo, ingresar la dirección IP del moderador en el segundo campo y presionar clic en el botón CONECTAR.... En la misma pantalla se visualizan las INSTRUCCIONES que inicialmente están ocultas, y se vuelven visibles dando clic en el icono de ayuda (?).



Participante

Ingrese los Datos

Ingrese Nombre

IP del moderador

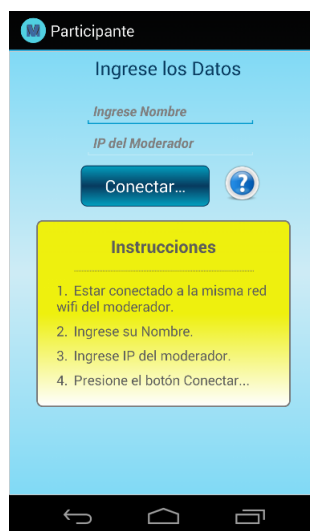
Conectar...

Instrucciones

1. Asegurese estar conectado a la misma red wifi del moderador.
2. Ingrese su Nombre.
3. Ingrese IP del moderador.
4. Presione el botón Conectar...

Figura 5.23: Wireframe *Iniciando como Participante*.

La pantalla a utilizar será: (Véase Figura 5.24)



Participante

Ingrese los Datos

Ingrese Nombre

IP del Moderador

Conectar...

Instrucciones

1. Estar conectado a la misma red wifi del moderador.
2. Ingrese su Nombre.
3. Ingrese IP del moderador.
4. Presione el botón Conectar...

Figura 5.24: Iniciando como Participante.

En la Figura 5.25 se puede visualizar la pantalla *Talleres*, que aparece luego de dar clic en el botón COMENZAR... de la Figura 5.22 y en el botón CONECTAR... de la Figura 5.24, para el moderador y participante respectivamente. Solo en el caso del moderador aparece el botón CREAR TALLER. En el caso en que existan más talleres de los que puedan aparecer en la pantalla, se descenderá deslizando de manera vertical el dedo sobre la pantalla.



Figura 5.25: Wireframe *Talleres*.

La pantalla a utilizar será: (Véase Figura 5.26)

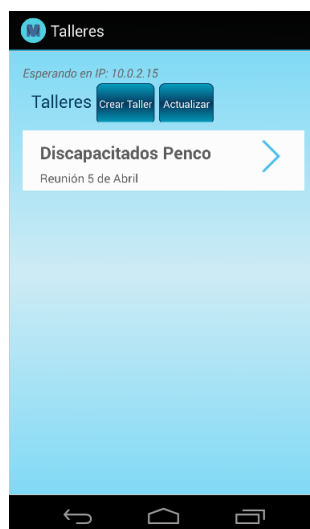


Figura 5.26: Talleres.

En la Figura 5.27 se puede visualizar la pantalla *Preguntas*, que aparece luego de dar clic en el recuadro donde aparece el nombre y descripción del taller en la Figura 5.26. Solo en el caso del moderador aparece el botón PREGUNTAR. En el caso en que existan más preguntas, se accederá a ella(s) deslizando de manera horizontal el dedo sobre la pantalla.

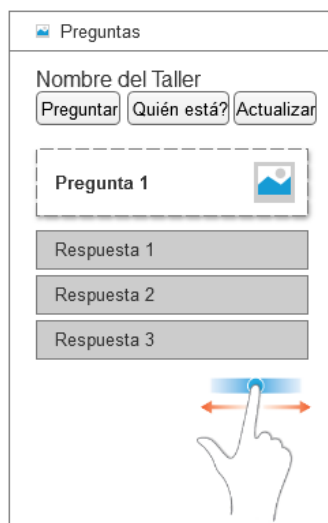


Figura 5.27: Wireframe *Preguntas*.

La pantalla a utilizar será: (Véase Figura 5.28)



Figura 5.28: Preguntas.

En la Figura 5.29 se puede visualizar la pantalla *Respuestas*, que aparece luego de dar clic en el recuadro donde aparece el nombre y descripción de la Pregunta en la Figura 5.28. Solo en el caso del moderador aparece el botón LIBERAR y para el participante el botón RESPONDER. En el caso en que existan más respuestas de las que puedan aparecer en la pantalla, se descenderá deslizando de manera vertical el dedo sobre la pantalla.

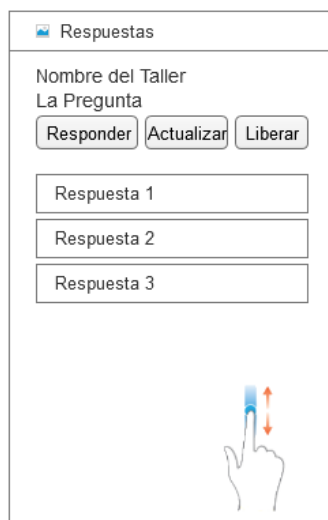


Figura 5.29: Wireframe *Respuestas*.

La pantalla a utilizar será: (Véase Figura 5.30)

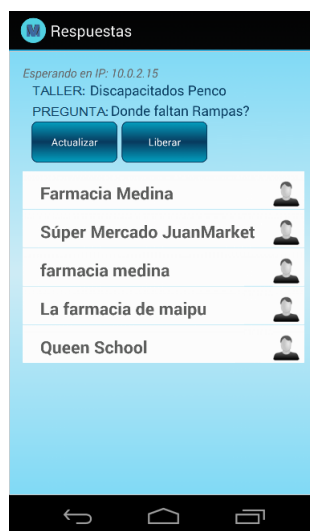


Figura 5.30: Respuestas.

5.4. Diccionario de Datos

Un diccionarios de datos ayuda a identificar las bases de datos que posee un sistema y las características de sus atributos. El sistema “Metaplandroid” cuenta con cuatro tablas, las cuales son: *Taller*, *Pregunta*, *Respuesta* y *Participante* respectivamente.

En la Tabla 5.1, se aprecia la tabla *Taller* con sus atributos, clave, tipo de dato, valor nulo, dominio y referencia respectivamente.

TALLER					
ATRIBUTO	CLAVE	TIPO DATO	VALOR NULO	DOMINIO	REFERENCIA
IdTaller	PK	Interger	NOT NULL		
IdCreador	FK	Interger	NOT NULL		PARTICIPANTE
NombreTaller		Text	NOT NULL		
DescripcionTaller		Text	NULL		
EstadoTaller		Interger	NOT NULL		
FechaIniTaller		Date	NOT NULL		
FechaFinTaller		Date	NOT NULL		

Tabla 5.1: Diccionario de Datos de la tabla Taller.

En la Tabla 5.2, se aprecia la tabla *Pregunta* con sus atributos, clave, tipo de dato, valor nulo, dominio y referencia respectivamente.

PREGUNTA					
ATRIBUTO	CLAVE	TIPO DATO	VALOR NULO	DOMINIO	REFERENCIA
IdPregunta	PK	Interger	NOT NULL		
IdTaller	FK	Interger	NOT NULL		TALLER
NombrePregunta		Text	NOT NULL		
DescripcionPregunta		Text	NULL		
EestadoPregunta		Interger	NOT NULL		
FechaIniPregunta		Date	NOT NULL		
FechaFinPregunta		Date	NOT NULL		

Tabla 5.2: Diccionario de Datos de la tabla Pregunta.

En la Tabla 5.3, se aprecia la tabla *Respuesta* con sus atributos, clave, tipo de dato, valor nulo, dominio y referencia respectivamente.

RESPUESTA					
ATRIBUTO	CLAVE	TIPO DATO	VALOR NULO	DOMINIO	REFERENCIA
IdRespuesta	PK	Integer	NOT NULL		
IdPregunta	FK	Integer	NOT NULL		PREGUNTA
IdParticipante	FK	Integer	NOT NULL		PARTICIPANTE
NombreRespuesta		Text	NOT NULL		
EstadoRespuesta		Integer	NOT NULL		
FechaIniRespuesta		Date	NOT NULL		
FechaFinRespuesta		Date	NOT NULL		

Tabla 5.3: Diccionario de Datos de la tabla Respuesta.

En la Tabla 5.4, se aprecia la tabla *Participante* con sus atributos, clave, tipo de dato, valor nulo, dominio y referencia respectivamente.

PARTICIPANTE					
ATRIBUTO	CLAVE	TIPO DATO	VALOR NULO	DOMINIO	REFERENCIA
IdParticipante	PK	Integer	NOT NULL		
NombreParticipante		Text	NOT NULL		

Tabla 5.4: Diccionario de Datos de la tabla Participante.

5.5. Modelo Relacional

El modelo relacional (ver Figura 5.31) es una representación de la base de datos del sistema, enfatizando la relación que existe entre las diferentes tablas de la base de datos, en la que algunos elementos de la tabla apuntan a su tabla de procedencia. En el sistema “Metaplandroid”, hay 4 relaciones que entrelazan las 4 tablas.

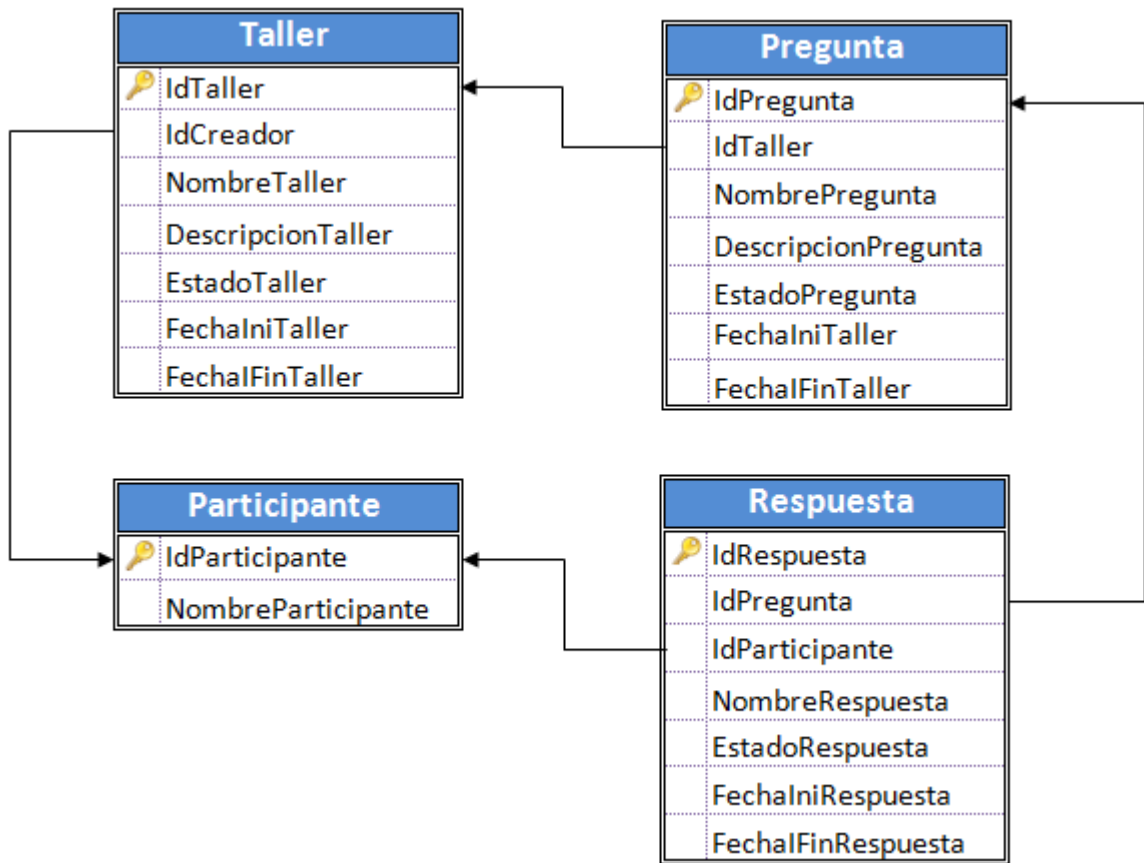


Figura 5.31: Modelo Relacional “Metaplandroid”.

5.6. Modelo Entidad-Relación

En el modelo entidad relacional o MER (ver Figura 5.32) se muestra la relación entre las entidades del sistema, en la que también se indica los elementos o atributos que posee cada entidad y la cardinalidad existente entre las entidades.

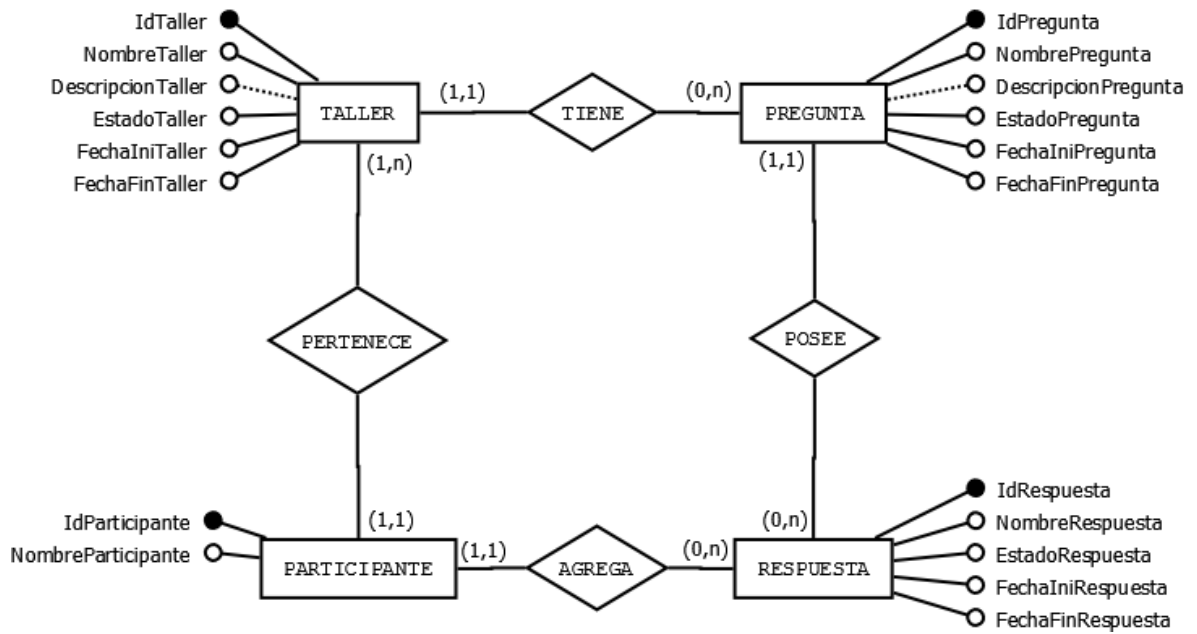


Figura 5.32: Modelo Entidad-Relación "Metaplandroid".

Capítulo 6

Implementación

En este capítulo se presenta la descripción de como se implemento la aplicación. A continuación recordaremos la estructura de la aplicación y luego se explicará la manera en que fue ésta implementada.

6.1. Estructura

Como se describió en el capítulo anterior (ver Figura 5.18), la aplicación está conformada por archivos `.java` y otros `.xml`, estos últimos son la parte visual de la aplicación, ya sea como pantallas o ventana emergentes. El sistema consta de 7 clases o ficheros `.java`, a los que llamaremos *actividades*, 14 archivos `.xml`, de los cuales 7 son la representación visual de cada actividad y los otros 7 son ventanas emergentes o tablas que se despliegan si hay datos en las base de datos.

Para una mayor comprensión de los ficheros `*.java`, o *actividades*, comenzaremos explicando que están divididos en 2 métodos: `onCreate()`, `onStart()` (aunque existen más métodos).

1. `onCreate()`: Se llama cuando se crea por primera vez la actividad. Aquí es donde se deben crear las vistas (llamar al `*.xml`), enlazar datos de listas, definir variables, etc. Este método también permite vincular variables guardadas de una actividad anterior.
2. `onStart()`: Se llama cuando la actividad se está convirtiendo en visible para el usuario. Es en donde se define que hará cada elemento de la vista, como que hará cada botón, que se cargará en los textos que no son estáticos o con que variable asociaremos

lo que se escribe en una campo de texto. En este método también se puede definir a la actividad a la que se pasará, si es que existe otra.

6.2. Android Manifest

El Android Manifest está situado en la raíz de cada aplicación, es un archivo llamado de configuración, llamado `AndroidManifest.xml`, donde podemos aplicar las configuraciones básicas de nuestra aplicación, además de definir los *permisos* y *actividades* que poseerá Metaplandroid. Éste sistema solicitan 4 permisos a la hora de instalar la aplicación:

1. INTERNET: Le permite a la aplicación crear conexiones vía socket.
2. ACCESS_WIFI_STATE: Permite a la aplicación acceder a la información de las redes wifi.
3. ACCESS_NETWORK_STATE: Permite a la aplicación acceder a la información de las redes.
4. READ_PHONE_STATE: Da acceso a información interna del teléfono.

A continuación se encuentra el código `AndroidManifest.xml` de la aplicación Metaplandroid:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     package="com.example.kfmaguz.metaplan" >
5     <!-- PERMISOS -->
6     <uses-permission android:name="android.permission.INTERNET" />
7     <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
8     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
9     <uses-permission android:name="android.permission.READ_PHONE_STATE" />
10
11     <application
12         android:allowBackup="true"
13         android:icon="@drawable/ic_launcher"
14         android:theme="@style/AppTheme" >
15         <activity
16             android:name="com.example.kfmaguz.metaplan.MainActivity"
17             android:label="Metaplandroid" >
18             <intent-filter>
19                 <action android:name="android.intent.action.MAIN" />
20                 <category android:name="android.intent.category.LAUNCHER" />
21             </intent-filter>
22         </activity>
23         <activity
```

```

24         android:name="com.example.kfmaguz.metaplan.MainOpciones"
25         android:label="Selecciona tu perfil" >
26     </activity>
27     <activity
28         android:name="com.example.kfmaguz.metaplan.TrackModerador"
29         android:label="Moderador" >
30     </activity>
31     <!-- Lo mismo para cada actividad -->
32 </application>
33 </manifest>

```

6.3. Iniciando la Aplicación

Corresponde a la primera actividad, la que ofrece una imagen representativa del sistema (ver Figura 6.1), mientras se crea o carga la base de datos del sistema y finalmente redirige a la siguiente actividad.



Figura 6.1: Pantalla de carga de la aplicación.

6.3.1. MainActivity.java

A continuación se muestra parte del código del fichero, cuya función principal se llama `Tiempo` la que se encarga de crear o cargar la base de datos, mientras ofrece una imagen del sistema y finalmente carga la siguiente pantalla `MainOpciones.java`.

```

1 public class MainActivity extends Activity {
2     ViewPager mViewPager;
3     protected Context context;

```

```
4     Tiempo time;
5
6     @Override
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.activity_main); // carga la vista .xml asociada
10
11         // Para cargar la base de datos (BDD) y pasar a la otra actividad
12         time = new Tiempo(this, 2000, 1000);
13         time.start();
14         context = getApplicationContext();
15     }
16
17     public class Tiempo extends CountdownTimer {
18         Activity actTemp;
19         public Tiempo(Activity act, long millisInFuture, long countDownInterval) {
20             super(millisInFuture, countDownInterval);
21             actTemp = act;
22         }
23
24         @Override
25         public void onTick(long millisUntilFinished) {
26             if(millisUntilFinished ==0){
27                 onFinish();
28             }
29         }
30
31         @Override
32         public void onFinish() {
33
34             //Crea la tabla en caso de no existir
35             TallerBD dbt = new TallerBD(getApplicationContext());
36             dbt.open();
37             if(dbt.isTableExists()==0){
38                 dbt.onDrop();
39                 dbt.onCreate();
40             }
41             dbt.close();
42
43             //ÍDEM para todas las tablas (PreguntasBD, RespuestasBD, ParticipantesBD)
44
45             //Creamos un Intent para pasar a la segunda actividad
46             Intent opcionesView = new Intent(getApplicationContext(), MainOpciones.class);
47             startActivity(opcionesView);
48             finish();
49         }
50     }
51 }
```

6.3.2. activity_main.xml

A continuación el código completo del .xml con comentarios de lo que hace, en las siguientes actividades se omitirán todos las propiedades de los elementos y solo serán mencionados por encima.

```

1 <?xml version="1.0" encoding="utf-8"?> <!--Para usar tildes-->
2 <RelativeLayout <!--layout que permite poner varios layout dentro-->
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/pager"
6     android:layout_width="match_parent" <!--que use todo el ANCHO del dispositivo-->
7     android:layout_height="match_parent" <!--que use todo el ALTO del dispositivo-->
8     tools:context=".MainActivity"
9     android:background="@drawable/fondob"> <!--otro .xml con el color del fondo-->
10
11 <TableLayout <!--layout con forma de tabla-->
12     android:gravity="center" <!--layout centrado-->
13     android:layout_width="match_parent" <!--el mismo ancho del RelativeLayout-->
14     android:layout_height="match_parent"> <!--el mismo alto del RelativeLayout-->
15
16     <TableRow> <!--layout con forma fila-->
17         <ImageView <!--Se usa para poner imágenes-->
18             android:layout_width="0dp"
19             android:layout_height="120dp"
20             android:layout_weight="1.5"
21             android:id="@+id/imageLogo"
22             android:src="@drawable/metaplan"
23             <!--Ubicación de la imagen-->
24             android:contentDescription="@string/app_logo" />
25     </TableRow>
26
27     <TableRow android:gravity="center">
28         <TextView <!--se usa para poner textos-->
29             <!-- existen 2 formas de poner el texto -->
30             android:text="@string/app_logo" <!--llamando el Texto desde otro archivo -->
31             <!--ó se se escribe directamente entre las comillas-->
32             android:id="@+id/textView"
33             android:layout_width="match_parent"
34             android:layout_height="wrap_content"
35             style="@style/AppLogo" />
36     </TableRow>
37 </TableLayout>
38 </RelativeLayout>

```

6.4. Pantalla Selecciona tu Perfil

Corresponde a la segunda actividad, que ofrece 2 opciones (ver Figura 5.20). La primera opción, el botón MODERADOR, es para entrar como moderador y la segunda opción, el botón PARTICIPANTE, es para entrar como participante.

6.4.1. MainOpciones.java

El siguiente código es en donde se definen los botones para ingresar como moderador o participante. Esta actividad se divide en otras 2, la primera es la actividad TrackModerador y la otra es TrackParticipante.

```

1  @Override
2  protected void onCreate(Bundle savedInstanceState) {
3      super.onCreate(savedInstanceState);
4      setContentView(R.layout.activity_main_opciones); // Carga la vista .xml
5  }
6
7  @Override
8  public void onStart() {
9      super.onStart();
10
11     //Declaramos "bundle" con la variable [nuevo=0]
12     final Bundle bundle = new Bundle();
13     bundle.putString("nuevo", "0");
14
15     //Inicializamos instancia de Botón MODERADOR y sus tareas
16     Button btnModerador= (Button)findViewById(R.id.btnModerador);
17     btnModerador.setOnClickListener(new OnClickListener(){
18         //Si damos clic y soltamos
19         @Override
20         public void onClick(View view){
21             //Nos lleva a la actividad del moderador
22             Intent myInSecciones= new Intent(getApplicationContext(), TrackModerador.class);
23             //Método para pasar "bundle" a la siguiente actividad
24             myInSecciones.putExtras(bundle);
25             startActivity(myInSecciones);
26         }
27     });
28     //Hacer lo mismo con el Botón PARTICIPANTE
29 }

```

6.4.2. activity_main_opciones.xml

A continuación el código reducido del `.xml`, en la que sólo se enfatiza que los botones deben llevar el parametro `android:id="@+id/..."`, para poder llamarlo desde la actividad.

```

1 <RelativeLayout>
2     <TableLayout>
3         <TableRow>
4             <TextView
5                 android:text="Seleccione la Opción">
6         </TableRow>
7
8         <TableRow>
9             <ImageView android:src="@drawable/moderator"/>
10            <Button
11                android:text="Moderador"
12                <!-- ponemos id para llamarlo desde la actividad-->
13                android:id="@+id/btnModerador"/>
14        </TableRow>
15
16        <TableRow>
17            <ImageView android:src="@drawable/guest"/>
18            <Button
19                android:text="Participante"
20                <!-- ponemos id para llamarlo desde la actividad-->
21                android:id="@+id/btnParticipante"/>
22        </TableRow>
23    </TableLayout>
24 </RelativeLayout>

```

6.5. Pantalla Iniciando como Moderador

Actividad que solicita el nombre del moderador y muestra su dirección IP (ver Figura 5.22). El moderador debe ingresar lo solicitado y apretar el botón COMENZAR... o dar clic en el icono de ayuda (?) en caso de dudar.

6.5.1. TrackModerador.java

El siguiente código muestra lo que ocurre al crear la actividad, es decir, en el método `onCreate()`. Primero se define y relacionan los elementos de la vista, se carga la dirección IP del dispositivo, se carga la variable `nuevo` de la actividad anterior y finalmente se crea la

función que impide iniciar la aplicación si no se está conectado a una red wifi o si el dispositivo no posee una dirección IP , con su mensaje de error correspondiente.

```

1  ServerSocket serverSocket;           // Para la conexión
2  String imei, cadenatalleres = "", estado, nuevo; // variables tipo string
3  Bundle i;           // para cargar el "bundle" (con variables de la actividad anterior)
4  TextView txtIP, txtMsg;           // textview de la vista
5  EditText txtPuerto, txtNombre; // EditText de la vista
6
7  @Override
8  protected void onCreate(Bundle savedInstanceState) {
9      super.onCreate(savedInstanceState);
10     setContentView(R.layout.activity_track_moderador); // cargar vista .xml
11
12     // Relacionamos los elementos de la vista
13     txtIP=           (TextView) findViewById(R.id.txtIP);
14     txtNombre=      (EditText) findViewById(R.id.txtNombre);
15     txtMsg=         (TextView) findViewById(R.id.textMsg);
16
17     //Cargamos en "bundle" de la actividad anterior
18     i = getIntent().getExtras();
19     nuevo = i.getString("nuevo");           // Si hay que cargar más variables se repite ésta
        linea
20
21     //Retornamos IP del dispositivo y lo escribimos en txtIP
22     txtIP.setText(Util.getIpAddress());
23
24     //Cuando no esta conectado el wifi
25     if (!Util.verificaConexion(this)) {
26         // Mensajes de ERROR 1
27         Toast.makeText(getApplicationContext(), getApplicationContext().getString(R.string.
                error_wifi1), Toast.LENGTH_SHORT).show();
28         // Termina la actividad. (vuelve actividad anterior)
29         this.finish();
30     }
31
32     //Cuando está conectado, pero el router no le a asignado una dirección IP
33     if(txtIP.getText().toString().trim().equals("")){
34         // Mensajes de ERROR 2
35         Toast.makeText(getApplicationContext(), getApplicationContext().getString(R.string.
                error_wifi2), Toast.LENGTH_SHORT).show();
36         // Termina la actividad. (vuelve actividad anterior)
37         this.finish();
38     }
39 }

```

Sobrescribimos método para destruir conexión vía socket, si se detiene la aplicación, con el método `onDestroy()`.

```

1  @Override
2  protected void onDestroy() {
3      super.onDestroy();
4
5      if(serverSocket != null) {
6          try {
7              serverSocket.close();
8          } catch (IOException e) {
9              e.printStackTrace();
10         }
11     }
12 }

```

Se crea un método que valida si los datos solicitados han sido ingresados correctamente a la hora de dar clic en el botón COMENZAR..., de lo contrario se retornará un mensaje de error. Este método también es utilizado en la actividad `TrackParticipante.java`.

```

1  public boolean RegistraDatosUsuario(int perfil){
2      boolean result;
3      String msg="";
4
5      //Verificamos si hay algo escrito
6      if(txtNombre.getText().toString().trim().equals("")){
7          msg+=getApplicationContext().getString(R.string.error_name);
8      }
9      //Imprime error si falta llenar texto
10     if(!msg.trim().equals("")){
11         result=false;
12         Toast.makeText(getApplicationContext(),msg,Toast.LENGTH_LONG).show();
13     } else {
14         result=true;
15     }
16     return result;
17 }

```

A continuación se inicializa el icono de ayuda (?) y el botón COMENZAR.

```

1  @Override
2  public void onStart() {
3      super.onStart();
4
5      //Iniciamos el botón de ayuda (?), de manera oculta
6      final TableLayout menuhelp = (TableLayout)findViewById(R.id.menuhelp);
7      menuhelp.setVisibility(View.INVISIBLE); //GONE ó INVISIBLE

```

```

8     estado = "0"; //invisible
9
10    //Inicializamos instancia de BTNHELP y sus tareas
11    ImageButton btnhelp= (ImageButton)findViewById(R.id.btnhelp);
12    btnhelp.setOnClickListener(new OnClickListener() {
13        @Override
14        public void onClick(View v) {
15            //Si se da clic y estado=0, se vuelve VISIBLE
16            if (estado.trim().equals("0")) {
17                menuhelp.setVisibility(View.VISIBLE);
18                estado = "1"; //visible
19            } else {
20                menuhelp.setVisibility(View.GONE); //GONE ó INVISIBLE
21                estado = "0"; //invisible
22            }
23        }
24    });
25
26    //Inicializamos instancia de Botón PARTICIPANTE
27    Button btnTalleres= (Button)findViewById(R.id.btnTalleres);
28    btnTalleres.setOnClickListener(new OnClickListener() {
29        @Override
30        public void onClick(View arg0) {
31            //Si esta ingresado el nombre, se crea Servidor
32            if (RegistraDatosUsuario(1)) { //Se pone 2 en el participante
33                ServerTh my = new ServerTh(TrackModerador.this, getApplicationContext(),
34                    cadenatalleres, txtIP.getText().toString().trim(), txtPuerto.getText().
35                    toString().trim(), 1, imei);
36                my.Ejecutar(); //CREAMOS UNA INSTANCIA DEL HILO SERVIDOR
37
38                // Guardamos Variables para la otra Actividad
39                Bundle bundle = new Bundle();
40                bundle.putString("perfil", "1");
41                bundle.putString("nuevo", "0");
42                bundle.putString("ipmod", txtIP.getText().toString().trim());
43                bundle.putString("username", txtNombre.getText().toString().trim());
44
45                //Cargamos la Actividad Siguiente
46                Intent myInSecciones = new Intent(getApplicationContext(), Talleres.class);
47                myInSecciones.putExtras(bundle);
48                startActivity(myInSecciones);
49            }
50        }
51    });
52 }

```

Finalmente la clase finaliza con el método `onKeyDown()` que define la actividad a la que debemos volver en caso de apretar el botón VOLVER del dispositivo. Éste método se agrega a en todas las actividades siguiente, siempre cambiando la actividad por la anterior.

```

1  @Override
2  public boolean onKeyDown(int keyCode, KeyEvent event) {
3      if ((keyCode == KeyEvent.KEYCODE_BACK)) {
4
5          Intent inst = new Intent(getApplicationContext(), MainOpciones.class);
6          startActivity(inst);
7          finish();
8      }
9      return super.onKeyDown(keyCode, event);
10 }
11 } //cierra toda la clase

```

6.5.2. activity_track_moderador.xml

La siguiente vista resumida, es para indicar con claridad los objetos que son llamados desde la actividad `TrackModerador.java`. Se debe asignar un ID al objeto que se quiere llamar usando `android:id="..."`.

```

1  <RelativeLayout>
2      <TableLayout>
3          <TableRow>
4              <TextView                                <!--TITULO, no cambia-->
5                  android:text="Ingrese los datos"/>
6          </TableRow>
7
8          <TableRow >
9              <!--ACÁ SE INGRESA EL NOMBRE-->
10             <EditText
11                 android:hint="Ingrese su nombre" <!--texto desaparece al dar clic-->
12                 android:id="@+id/txtNombre"          <!--Ingrese su nombre-->
13                 android:inputType="text" />          <!--definimos tipo de dato-->
14             </EditText>
15
16             <TableRow>
17                 <!-- MOSTRAR IP del dispositivo, si cambia -->
18                 <TextView
19                     android:id="@+id/txtIP"/>
20             </TableRow>
21
22             <LinearLayout>
23                 <TableRow>
24                     <Button
25                         android:text="Comenzar..."
26                         android:id="@+id/btnTalleres" />
27                     <ImageButton                    <!-- icono de ayuda (?) -->
28                         android:id="@+id/btnhelp"    <!-- su id para llamarlo en la act -->

```

```

29         android:src="@drawable/help" <!-- ubicación del icono -->
30         android:background="@android:color/transparent"/>
31     </TableRow>
32 </LinearLayout>
33
34 <LinearLayout>
35     <!-- MENU que aparece al dar clic en icono de ayuda (?) -->
36     <TableLayout>
37         android:id="@+id/menuhelp">
38         <TableLayout>
39             <TableRow>
40                 <TextView
41                     android:text="Instrucciones"/>
42             </TableRow>
43             <TableRow>
44                 <TextView
45                     android:text="1.\tAsegurarse que los participantes estén
46                         conectados a tu misma red wifi."/>
47             </TableRow>
48             <TableRow>
49                 <TextView
50                     android:text="2.\tIngrese su Nombre."/>
51             </TableRow>
52             <TableRow>
53                 <TextView
54                     android:text="3.\tPresione el botón Comenzar..."/>
55             </TableRow>
56             <TableRow>
57                 <TextView
58                     android:text="4.\tDiga la IP en la siguiente ventana."/>
59             </TableRow>
60         </TableLayout>
61     </TableLayout>
62 </LinearLayout>
63 </RelativeLayout>

```

6.6. Pantalla Iniciando como Participante

Actividad que solicita el nombre del participante, la dirección IP del moderador (ver Figura 5.24). El participante debe ingresar los datos solicitados y apretar el botón CONECTAR... o dar clic en el icono de ayuda (?) en caso de dudar.

Debido a que la actividad anterior es similar, pasaremos a la siguiente actividad.

6.7. Pantalla Talleres

Actividad usada tanto por moderador como por el participante, y dependiendo del perfil del usuario se ocultan textos o botones. En el caso del moderador se visualiza en la parte superior su IP, la que debe mencionar verbalmente a los participantes, también visualiza dos botones, el primero es CREAR TALLER y el segundo es el botón ACTUALIZAR. En el caso del participante ve el botón ACTUALIZAR. Una vez el moderador haya creado el taller de Metaplan, será visible por el moderador y participantes. Ver Figura 5.26.

6.7.1. Talleres.java

Antes de utilizar cualquier método y antes de declarar variables, es necesario crear un arreglo para registrar talleres en el dispositivo.

```

1 public class Talleres extends Activity{
2     ArrayList<Taller> datos = new ArrayList<>();
3
4     String perfil, nuevo, ipmod, username;
5     Bundle i;
6     TextView infoIP;

```

En el método `onCreate()` se vinculan los elementos de la vista y se carga el `bundle` de igual manera que la actividad anterior. En el método `onStart()` se vuelve a definir qué harán los botones.

Es importante tener claro que la variable `Perfil` es del tipo *integer* y toma el valor 1 si el usuario es moderador y el valor 2 si es participante.

El botón ACTUALIZAR está implementado de igual manera en las siguientes actividades y el botón CREAR TALLER está un poco alterado para una mayor comprensión del código.

```

1 @Override
2 public void onStart() {
3     super.onStart();
4
5     ShowGrid(); //Función que carga los talleres ya existentes en la bdd
6
7     //Inicializamos el Botón Actualizar
8     Button btnUpdate = (Button) findViewById(R.id.btnUpdate);
9     btnUpdate.setOnClickListener(new View.OnClickListener() {

```

```

10     @Override
11     public void onClick(View view) {
12         ShowGrid();          // Función que carga los talleres
13     }
14 });
15
16 //Inicializamos el Boton Nuevo Taller
17 Button btnNuevo = (Button)findViewById(R.id.btnNuevo);
18 //Ocultar boton a los PARTICIPANTES
19 if(perfil.trim().equals("2")) {
20     btnNuevo.setVisibility(View.GONE);
21 }
22 btnNuevo.setOnClickListener(new View.OnClickListener() {
23     @Override
24     public void onClick(View view) {
25         final View grabar = view;
26
27         //Creamos "Ventana Emergente" - CREAR NUEVO TALLER
28         LayoutInflater inflater = LayoutInflater.from(view.getContext());
29         //se llama a la VISTA "talleres_new.xml"
30         View promptView = inflater.inflate(R.layout.talleres_new, null);
31         AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(view.getContext
32             ());
33         alertDialogBuilder.setView(promptView);
34         alertDialogBuilder.setTitle("Nuevo Taller");
35
36         //INGRESE NOMBRE DEL TALLER
37         final TextView txtDescripcion = (TextView) promptView.findViewById(R.id.
38             txtDescripcion);
39         //INGRESE DESCRIPCION DEL TALLER
40         final TextView txtDescripcionTec = (TextView) promptView.findViewById(R.id.
41             txtDescripcionTec);
42
43         //Vinculamos BOTON FINALIZAR
44         final Button btnFinalizar = (Button) promptView.findViewById(R.id.btnFinalizar);
45         //Ponemos Invisible el boton la primera vez
46         btnFinalizar.setVisibility(View.GONE);
47
48         //AGREGAMOS BOTONES Listo-Cancelar (NO son del .xml)
49         alertDialogBuilder
50             .setCancelable(false)
51
52         //Al apretar boton listo se Agrega a TallerBD
53         .setPositiveButton("Listo", new DialogInterface.OnClickListener() {
54             public void onClick(DialogInterface dialog, int id) {
55                 //REGISTRAMOS EL TALLER
56                 Calendar c = Calendar.getInstance();
57                 SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss"
58                     , java.util.Locale.getDefault());
59                 Date fechas = new Date();

```

```

56         try {
57             fechas = dateFormat.parse(c.toString());
58         } catch (ParseException e) {
59             e.printStackTrace();
60         }
61
62         Taller my = new Taller(0, //la ID se cambia sola
63                               IdCreador,
64                               txtDescripcion.getText().toString().trim(),
65                               txtDescripcionTec.getText().toString().trim(),
66                               2, //ESTADO 2=Liberado
67                               fechas,
68                               fechas);
69         TallerBD db = new TallerBD(grabar.getContext());
70         db.open();
71         db.Insert(my);
72         db.close();
73
74         ShowGrid();
75     }
76     })
77     .setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
78         public void onClick(DialogInterface dialog, int id) {
79             dialog.cancel();
80         }
81     });
82     AlertDialog alertD = alertDialogBuilder.create();
83     alertD.show();
84 }
85
86 //Mostramos "Ventana Emergente"
87 ListView mylist =(ListView) findViewById(R.id.LVTalleres);
88 mylist.setAdapter(mAdapter);
89
90 //Mostramos el taller recién creado
91 ShowGrid();
92 }

```

La función ShowGrid() es la encargada de desplegar los talleres creados. En las siguientes actividades se usa de manera similar.

```

1 public void ShowGrid() {
2     try {
3         try {
4             datos.clear();
5         } catch (Exception exc) {
6             exc.printStackTrace(); //Para ver errores si se cae
7         }

```

```
8
9 //Declaramos una instancia de la tabla de TALLERES
10 TallerBD db = new TallerBD(this);
11 db.open();
12
13 //Declaramos un cursor que representara el resultado de la consulta a la tabla de
14 talleres
15 Cursor myCur = db.showList();
16
17 //declaramos variables de posición usando el ID
18 int indId= myCur.getColumnIndex(TallerBD.keyId);
19 //Nombre taller
20 int indDescripcion= myCur.getColumnIndex(TallerBD.keyDescripcion);
21 //Descripcion taller
22 int indDescripcionTec= myCur.getColumnIndex(TallerBD.keyDescripcionTec);
23 //Estado taller
24 int indEstado= myCur.getColumnIndex(TallerBD.keyEstado);
25
26 // Corro las filas de la base de datos
27 if(myCur.moveToFirst()){
28     do{
29         Calendar c = Calendar.getInstance();
30         SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss",
31             java.util.Locale.getDefault());
32         Date fechas = new Date();
33         try {
34             fechas = dateFormat.parse(c.toString());
35         } catch (ParseException e) {
36             e.printStackTrace();
37         }
38
39         //declaramos e inicializamos variables que representan los valores
40         informados en el cursor.
41         String myId= myCur.getString(indId);
42         String myDescripcion= myCur.getString(indDescripcion);
43         String myDescripcionTec= myCur.getString(indDescripcionTec);
44         int estado = myCur.getInt(indEstado);
45
46         //ingresamos cada taller del cursor en el Arreglo
47         datos.add(new Taller(Integer.parseInt(myId),
48             myCreator
49             myDescripcion,
50             myDescripcionTec,
51             estado,
52             fechas,
53             fechas)
54     );
55     }while (myCur.moveToNext());
56 }/*else{
```

```

55         //Toast.makeText(this, "No hay datos! ", Toast.LENGTH_LONG).show();
56     }*/
57     db.close();
58     mAdapter.notifyDataSetChanged(); //informamos a la Clase Adapter que escuche los
        cambios para cargar nuevamente los talleres
59
60     }catch(Exception ex) {
61         ShowGrid();
62     }
63 }

```

6.7.2. activity_talleres.xml

La siguiente vista resumida es para explicar donde se despliegan los talleres e indicar los `android:id="..."` llamados desde la actividad `Talleres.java`.

```

1 <LinearLayout>
2     <TableLayout>
3         <TableRow>
4             <TextView
5                 android:id="@+id/infoIP"/>
6         </TableRow>
7     </TableLayout>
8
9     <TableLayout>
10        <TableRow>
11            <TextView
12                android:text="@string/title_talleres" <!--botón Talleres-->
13                android:id="@+id/textView"/>
14            <Button
15                android:text="@string/boton_Taller" <!--botón Crear Taller-->
16                android:id="@+id/btnNuevo"/>
17            <Button
18                android:text="@string/boton_Update" <!--Actualizar-->
19                android:id="@+id/btnUpdate"/>
20        </TableRow>
21    </TableLayout>
22
23    <!-- ES ACA DONDE APARECEN LOS TALLERES -->
24    <ListView
25        android:id="@+id/LVTalleres">
26    </ListView>
27
28 </LinearLayout>

```

6.8. Pantalla Preguntas

Actividad usada tanto por moderador como por el participante, y dependiendo del perfil del usuario se ocultan textos o botones. En el caso del moderador se visualiza en la parte superior su IP, también visualiza tres botones, PREGUNTAR, QUIÉN ESTÁ? y ACTUALIZAR. En el caso del participante ve el botón ACTUALIZAR. Una vez el moderador haya creado la pregunta, será visible por el moderador y participantes. Ver Figura 5.28.

Esta actividad funciona de manera similar que la actividad `Talleres.java`, por lo que sólo se explicara como agregar el *layout* que permite desplazarse de manera horizontal.

6.8.1. activity_preguntas.xml

Para desplazarse de manera horizontal en la pantalla del dispositivo Android, es necesario agregar un *layout* que no trae AndroidStudio, de la siguiente manera:

```

1 <LinearLayout>
2     <!--Acá se omiten todos los botones y textos-->
3
4     <com.devsmart.android.ui.HorizontalListView
5         android:id="@+id/LVPreguntas"
6         android:layout_width="fill_parent"
7         android:layout_height="wrap_content"
8         android:layout_marginTop="10dp"
9         android:layout_marginLeft="10dp"
10        android:layout_marginRight="10dp">
11 </com.devsmart.android.ui.HorizontalListView>
12 </LinearLayout>

```

6.9. Pantalla Respuestas

Actividad usada tanto por moderador como por el participante, y dependiendo del perfil del usuario se ocultan textos o botones. En el caso del moderador se visualiza en la parte superior su dirección IP, también visualiza dos botones, ACTUALIZAR y LIBERAR. En el caso del participante ve los botones RESPONDER y ACTUALIZAR. Una vez todos los participantes hayan respondido la pregunta, el moderador debe apretar el botón LIBERAR y las respuestas serán visibles por todos los participantes. Ver Figura 5.30.

6.9.1. Respuestas.java

Dentro de las funciones que se pueden explicar, y que no se han explicado antes, se encuentra la del botón LIBERAR la que envía las respuesta entre todos los participantes. Hay que dejar en claro que cada vez que un participante responde, su respuesta es enviada de manera automática al Moderador.

```

1 //Inicializamos el Boton Liberar
2 Button btnLiberar= (Button)findViewById(R.id.btnLiberar);
3 //Ocultamos el boton de los Participantes
4 if(perfil.trim().equals("2")) {
5     btnLiberar.setVisibility(View.GONE);
6 }
7 btnLiberar.setOnClickListener(new View.OnClickListener() {
8     @Override
9     public void onClick(View view) {
10
11         //creando "Ventana Emergente"
12         AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(view.getContext());
13         alertDialogBuilder.setTitle("Liberar Respuesta");
14         alertDialogBuilder
15             .setCancelable(false)
16             //Creando botones sin un .xml
17             .setPositiveButton("Liberar", new DialogInterface.OnClickListener() {
18                 public void onClick(DialogInterface dialog, int id) {
19                     RespuestaBD db = new RespuestaBD(getApplicationContext());
20                     db.open();
21                     db.UpdateAll(Integer.parseInt(taller) , Integer.parseInt(pregunta));
22                     //Idtaller e IdPregunta
23                     db.close();
24                 }
25             })
26             .setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
27                 public void onClick(DialogInterface dialog, int id) {
28                     dialog.cancel();
29                 }
30             });
31         AlertDialog alertD = alertDialogBuilder.create();
32         alertD.show();
33     }
34 });

```

6.10. Bases de Datos

Gran parte de las funciones utilizadas han sido creada en otras clases, de manera que sean definidas una sola vez y después sean llamadas. Dentro de estas funciones se encuentran todas las que interactúan con las bases de datos, ya que éstas pueden ser llamadas desde distintas actividades y permite una mayor facilidad al desarrollador en caso de cambiar o agrandar la aplicación.

Las tablas se declaran de la siguiente manera:

```

1 public class TallerBD {
2     public static final String keyId ="IdTaller";
3     public static final String keyCreator="IdCreador";
4     public static final String keyDescripcion="NombreTaller";
5     public static final String keyDescripcionTec="DescripcionTaller";
6     public static final String keyEstado="EstadoTaller";
7     public static final String keyCreate="FechaIniTaller";
8     public static final String keyExpired="FechaFinTaller";
9
10    private static final String DATABASE_NAME = "Metaplan"; //Nombre DB
11    private static final String DATABASE_TABLE = "Taller"; //Nombre Tabla
12    private static final int DATABASE_VERSION = 1;
13    private static final String TAG = "DBAdapter";
14
15    private static final String DATABASE_DROP ="DROP TABLE IF EXISTS "+DATABASE_TABLE;
16    private static final String DATABASE_CREATE =" Create Table "+DATABASE_TABLE+" ( "+
17        " IdTaller integer Primary key Autoincrement, "+
18        " IdCreador integer not null, "+
19        " NombreTaller text not null, "+
20        " DescripcionTaller text null, "+
21        " EstadoTaller integer not null, "+
22        " FechaIniTaller Date not null, "+
23        " FechaFinTaller Date not null);" +
24        " Foreign Key (IdCreador) References Participante (IdParticipante)" +
25        " ON DELETE CASCADE);" +
26        " CREATE INDEX "+DATABASE_TABLE+"_idx ON "+DATABASE_TABLE+" (Id);COMMIT;";
27
28    private final Context context;
29    private DatabaseHelper DBHelper;
30    private SQLiteDatabase db;
31
32    private final Context context;
33    private DatabaseHelper DBHelper;
34    private SQLiteDatabase db;
35
36    public TallerBD(Context ctx){
37        this.context = ctx;
38        DBHelper = new DatabaseHelper(context);

```

```

39     }
40
41     public static class DatabaseHelper extends SQLiteOpenHelper{
42         DatabaseHelper(Context context){
43             super(context, DATABASE_NAME, null, DATABASE_VERSION);
44         }
45
46
47         public void onDrop(SQLiteDatabase db){
48             db.execSQL(DATABASE_DROP);
49         }
50
51         @Override
52         public void onCreate(SQLiteDatabase db){
53             db.execSQL(DATABASE_CREATE);
54         }
55
56         @Override
57         public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){
58             Log.w(TAG, "Upgrading database from version " + oldVersion
59                 + " to "
60                 + newVersion + ", which will destroy all old data");
61             db.execSQL("DROP TABLE IF EXISTS " + DATABASE_TABLE);
62             onCreate(db);
63         }
64     }
65
66     public void onDrop() throws SQLException{
67         DBHelper.onDrop(db);
68     }
69     public void onCreate() throws SQLException{
70         DBHelper.onCreate(db);
71     }

```

Y todas la consultas se hacen a continuación, y éstas pueden ser llamadas desde las actividades.

```

1 //Abrir Tabla
2 public TallerBD open() throws SQLException{
3     db = DBHelper.getWritableDatabase();
4     return this;
5 }
6 //Cerrar Tabla
7 public void close(){
8     DBHelper.close();
9 }
10
11 //Insertar en tabla
12 public long Insert(Taller my){

```

```
13     long resultado=0;
14
15     try{
16         ContentValues initialValues = new ContentValues();
17
18         //initialValues.put(keyId,my.getId());
19         initialValues.put(keyCreator,my.getCreator());
20         initialValues.put(keyDescripcion,my.getName());
21         initialValues.put(keyDescripcionTec,my.getLastname());
22         initialValues.put(keyEstado,my.getStatus());
23         initialValues.put(keyCreate,my.getCreate().toString());
24         initialValues.put(keyExpired,my.getExpired().toString());
25         resultado= db.insert(DATABASE_TABLE, null, initialValues);
26     }catch(Exception ex)
27     {
28         String error = ex.getMessage();
29         error = error.trim();
30     }
31     return resultado;
32 }
33
34 //Eliminar tabla
35 public boolean DeleteAll(){
36     return db.delete(DATABASE_TABLE, null, null) > 0;
37 }
38
39 //Retornar elementos de la tabla
40 public Cursor showList(){
41
42     String clausula = " 1 = 1 "; //
43     try{
44         return db.query(DATABASE_TABLE, new String[]{
45             keyId,
46             keyCreator,
47             keyDescripcion,
48             keyDescripcionTec,
49             keyEstado
50             keyCreate
51             keyExpired},
52             clausula,
53             null,
54             null,
55             null,
56             keyId);
57     }catch(SQLiteException ex){
58         String g = ex.getMessage();
59         return null;
60     }
61 }
```

Capítulo 7

Pruebas

7.1. Pruebas de Usabilidad

Para la prueba de usabilidad se eligieron 3 grupos de a dos individuos entre los 18 y 55 años de edad, en uno de los grupos hay un individuo que ha realizado reuniones con grupos de personas en las que él realiza preguntas (posible moderador). Se les entregaron a cada uno un teléfono inteligente marca Motorola, modelo Moto G, con la versión de Android 4.4.4. El cuestionario utilizado (ver Tabla 7.1) fue obtenido de libro *Usability Evaluation in Industry* de John Brooke (Brooke, 1986).

Prueba de Usabilidad					
	EN DESACUERDO			DE ACUERDO	
Nivel de acuerdo	1	2	3	4	5
Utilizaría con frecuencia el programa					
Encontré el programa muy complejo					
Fue fácil utilizarlo					
Necesitaría de un experto para utilizarlo					
Las diversas funciones están bien integrados					
Hubo demasiada inconsistencia visual					
Lo encontré muy difícil de usar					
Las personas lo aprenderían a usar rápidamente					
Me sentí muy confiado en la navegación					
Necesitaría aprender más antes de utilizarlo					

Tabla 7.1: Cuestionario de usabilidad System Usability Scale (SUS).

Esta prueba se desarrollo de acuerdo al método *Protocolo del Pensamiento Manifestado*

o *Think Aloud Protocol* en inglés, en la que se solicita que el usuario exprese en voz alta sus pensamientos, sensaciones y opiniones mientras interactúa con el producto.

7.2. Resultados Pruebas de Usabilidad

La prueba de usabilidad consistió en 4 fases. En la primera, se le explica a los usuarios el propósito y actividades a realizar en el estudio de usabilidad. En la segunda fase, el usuario tuvo hasta 5 minutos para familiarizarse con el teléfono inteligente y el programa Metaplandroid. En la tercera fase, los usuarios interactuaron con los dispositivos. En la cuarta fase los individuos respondieron el cuestionario de usabilidad.

Los resultados se encuentran en el Anexo. Los datos fueron promediados siendo 5 el máximo para un total acuerdo y 1 el mínimo para un total desacuerdo. Los datos arrojaron la siguiente tabla (véase Tabla 7.2).

Promedio de Pruebas de Usabilidad	
Utilizaría con frecuencia el programa	2,8
Encontré el programa muy complejo	2,2
Fue fácil utilizarlo	4
Necesitaría de un experto para utilizarlo	1,2
Las diversas funciones están bien integrados	3,2
Hubo demasiada inconsistencia visual	1,8
Lo encontré muy difícil de usar	2,2
Las personas lo aprenderían a usar rápidamente	3,5
Me sentí muy confiado en la navegación	3,7
Necesitaría aprender más antes de utilizarlo	2,5

Tabla 7.2: Promedio de los cuestionarios de usabilidad System Usability Scale (SUS).

Los individuos dijeron durante el test que no es una aplicación que instalarían de forma independiente, pero que si la instalaran para participar de un taller, los resultados muestran que no es un programa complejo y que en general fue fácil usarlo. El apoyo de un experto no es una necesidad, mientras que el moderador tuviera claridad de lo que se debe hacer, en cuanto a que las personas lo aprenderán a usar rápidamente, varios dijeron que hacen falta indicar que hay que mantener el clic para modificar datos. La mayoría se sintió confiado en la navegación, pero algunos dijeron que no se nota cuando hay otras preguntas, si el moderador no les avisa. Finalmente la mayoría dijo que no es necesario aprender más antes de utilizarlo, y que se aprende en la práctica.

7.3. Pruebas de Funcionamiento

Para la prueba de funcionamiento se realizó un taller en una habitación con 8 dispositivos conectados, la aplicación funcionó correctamente, salvo que hubo usuarios que recibieron llamadas y se les cerró la aplicación pero al volver a conectar no tuvieron problemas, otro usuario experimentó pérdida de dirección IP, por lo que se caía constantemente, pero al parecer fue un caso aislado debido a su alcance del wifi. Se espera repetir la prueba con una mayor cantidad de dispositivos o realizar una prueba de estrés.

7.4. Pruebas de Requerimientos

Para llevar a cabo la prueba de requerimientos se seleccionó un individuo que tenía entre los 18 y 55 años de edad, que tuviese un dispositivo móvil con Android y que ha realizado reuniones con un grupo de personas en las que se realizan preguntas. A continuación se presenta la prueba que se realizó para verificar la validez de los requerimientos documentados en la especificación de requerimientos. (Véase Tabla 7.3)

Prueba de Requerimientos							
	C: Cumple					NC: No cumple	
	R1	R2	R3	R4	R5	R6	...
El requisito se encuentra debidamente documentado							
El requisito no tiene errores de sintaxis y morfológicos							
El requisito cumple con las expectativas del cliente							
El requisito no tiene palabras ambiguas							
El requisito tiene documentado todo lo solicitado							
Se realizó el proceso de reunión y se tiene la documentación correcta							

Tabla 7.3: Prueba de requerimientos de la aplicación móvil Metaplandroid.

7.5. Resultado Pruebas de Requerimientos

A continuación los resultado de la prueba de requerimientos realizadas para cada requerimiento de la aplicación móvil Metaplandroid (véase Tablas 7.4 y 7.5).

Prueba de Requerimientos								
	C: Cumple					NC: No cumple		
	R1	R2	R3	R4	R5	R6	R7	R8
El requisito se encuentra debidamente documentado	C	C	C	C	C	C	C	C
El requisito no tiene errores de sintaxis y morfológicos	C	C	C	C	C	C	C	C
El requisito cumple con las expectativas del cliente	C	C	C	C	C	C	C	C
El requisito no tiene palabras ambiguas	C	C	C	C	C	C	C	C
El requisito tiene documentado todo lo solicitado	C	C	C	C	C	C	C	C

Tabla 7.4: Resultado de las prueba de requerimientos de la aplicación parte 1.

Prueba de Requerimientos							
	C: Cumple					NC: No cumple	
	R9	R10	R11	R12	R13	R14	R15
El requisito se encuentra debidamente documentado	C	C	C	C	C	C	C
El requisito no tiene errores de sintaxis y morfológicos	C	C	C	C	C	C	C
El requisito cumple con las expectativas del cliente	C	C	C	C	C	C	C
El requisito no tiene palabras ambiguas	C	C	C	C	C	C	C
El requisito tiene documentado todo lo solicitado	C	C	C	C	C	C	C

Tabla 7.5: Resultado de las prueba de requerimientos de la aplicación parte 2.

En base a las pruebas realizadas para cada requerimiento se pudo comprobar el correcto funcionamiento de la aplicación móvil “Metaplandroid”.

Capítulo 8

Conclusiones y Recomendaciones

La tecnología ha dado grandes avances en los últimos años, en especial en la microtecnología, donde cada vez se crean dispositivos con mejores capacidades y los dispositivos móviles como celulares inteligentes, tablets y notebook están al alcance de casi todas las personas, y su uso comienza cada vez a más temprana edad. Cada vez que se crea un dispositivo, el siguiente paso es hacerlo más pequeño, sin evitar que cumpla su función original, y por la gran demanda y competencia se pueden adquirir fácilmente.

Esto ha abierto el camino de la computación en muchos ámbitos y ha hecho que la computación móvil esté presente en todas partes. Con el paso del tiempo, la computación portátil está resolviendo tareas en distintos ámbitos, que antiguamente no tenían solución o que su solución era muy compleja.

El principal objetivo de este proyecto fue realizar una aplicación en Android, para realizar talleres de Metaplan. De esta manera ofrecer una alternativa diferente a la que existe hoy en día. De acuerdo a los objetivos propuestos en la sección 1.4, se puede concluir lo siguiente:

- **Objetivo Especifico 1:** Se analizaron las funcionalidades del sistema. Primero se investigo y estudio sobre los talleres de Metaplan, se indago en documentos y se converso con una moderadora de Metaplan, la profesora *Claudia Carrasco Sagredo* de la *Universidad Católica de la Santísima Concepción*, a la que se le realizó todas las dudas que quedaban. Después se investigo sobre las diferentes alternativas que existen para desarrollar aplicaciones Android en la que se estableció casi desde un comienzo que la herramienta a utilizar sería Android Studio debido a que es un entorno de desarrollo desarrollada por Android y que está creciendo velozmente, además ofrece una gran can-

tividad de herramientas y su editor de diseño es bastante cómodo, todo esto acompañado de un aprendizaje del lenguaje Java. Con todo este conocimiento se pudo comprender el sistema, se empezó a definir que funcionalidades podría tener el sistema y cuales estaban fuera del alcance ya sea por su complejidad o porque incurrirán en gastos (véase capítulo 4).

- **Objetivo Especifico 2:** Se Diseño y modelo la aplicación. Primero se estudio que modelos eran los más adecuados para el sistema, luego se seleccionaron los más adecuados que fueron los diagrama de actividad, diagrama de despliegue, interfaces de usuario, diccionario de datos, modelo relacional y modelo entidad relación, finalmente se llevo a cabo su confección (véase Capítulo 5).
- **Objetivo Especifico 3:** Se construyo la aplicación. Primero se estudió el lenguaje Java y XML. Luego se comenzó a desarrollar la aplicación en Android Studio, la que causo varias dificultades debido a la falta de experiencia, sus constantes actualizaciones y su gran consumo de recursos del ordenador. El desarrollo de la aplicación esta explicado en esta tesis (véase capítulo 6).
- **Objetivo Especifico 4:** Finalmente se desarrollo la etapa de pruebas o testing en la que se realizaron 3 pruebas: la prueba de usabilidad, prueba de estrés y prueba de requerimiento, las que a pesar de no tener mal resultado dejaron a entender que la aplicación aun tiene que mejorar.

Las ventajas de este desarrollo permite obtener una nueva herramienta con la que se pueden desarrollar talleres de Metaplan sin la necesidad de los materiales que frecuentemente se usan, y por otro lado, seguir fomentando en el desarrollo de las aplicaciones móviles para que puedan seguir ayudando a las personas.

Para trabajos futuros en relación a la aplicación se busca mejorar la comunicación entre dispositivos y que ésta no solicite la dirección IP del moderador. Integran la comunicación ad hoc entre los dispositivos ya sea por wifi o Bluetooth. Que al finalizar el taller no se abra la ventana emergente para enviarlo por correo, y que ésta funcionalidad se implemente en un nuevo botón o en un icono en la toolbar. Mejorar el cuerpo de envío de la información y ofrecerle al usuario alternativas a la hora de enviar el taller como por ejemplo que sea un documento .pdf u otro formato similar. Permitir agarrar las respuestas y moverlas, de tal manera que se puedan subir, bajar e incluso agrupar. Agregar opciones como activar/desactivar *Ver*

nombre del usuario que respondió, Cronometrar respuestas o No recibir llamadas entrantes, cuando la aplicación este en primer plano.

La recomendación que puede dejar, es que mientras estén desarrollando una aplicación en Android Studio no actualicen la versión, ya que esto puede ocasionar que su aplicación deje de funcionar y gasten demasiado tiempo buscando la solución. En mi caso comencé con el desarrollo en la versión 0,8 y luego de actualizar (cada vez que me lo sugería) termine con la versión 1.2, pero varias veces no podía ejecutar la aplicación debido a que éstas actualizaciones volvían obsoleto algunos comandos o librerías.

Bibliografía

Brooke, J. (1986). Usability evaluation in industry.

Dávalos, A. G. (2012). Introducción a la computación móvil.

Frodigh, M., Johansson, P., and Larsson, P. (2000). Formación de redes inalámbricas ad hoc. el arte de la formación de redes sin red.

Gironés, J. T. (2012). *El gran libro de Android*. Marcombo, 2 edition.

Gonzalez, A., Madoz, C., Saadi, F., and Hughes, D. (2011). Estrategias de virtualización para la moderación grupal basados en la técnica de metaplan.

Hughes, D., Saadi, M. F., Madoz, C., and Gonzalez, A. (2009). Aplicación para la administración y desarrollo de cursos con la técnica de metaplan que aporta etapas virtuales mediante la web.

Metaplan (a). Metaplan basic techniques. moderating group discussions using the metaplan approach.

Metaplan (b). Primer for the metaplan. how to moderate group discussions using the metaplan technique.

Torres, X. C. (2007). Virtualización de la enseñanza-aprendizaje de una metodología de moderación grupal: Metaplan.

Anexo

Manual de Usuario

El presente documento corresponde al Manual de Usuario de la aplicación móvil Metaplandroid. Antes de comenzar hay que dejar en claro que la aplicación móvil funciona solo con dispositivos conectados a una misma red wifi. Es recomendable que el taller sea realizado en una misma sala, en la que el moderador tenga la atención de todos los participantes, el motivo es para que los participantes estén concentrados y las respuestas sean más fluidas.

- **Iniciando Aplicación:** Apenas el usuario abre la aplicación móvil, aparece esta pantalla (ver Figura 1) que apenas dura un par de segundos. El usuario no puede interactuar sobre ella, ya que solo corresponde a una pantalla de presentación y para cargar algunos recursos.



Figura 1: Iniciando Aplicación.

- Selecciona tu Perfil:** Para que el usuario ingrese como moderador debe dar clic en el botón MODERADOR, y si es participante debe dar clic en el botón PARTICIPANTE (ver Figura 2).



Figura 2: Selecciona tu Perfil.

En caso de que el usuario no esté conectado a una red wifi la aplicación arrojará un mensaje de error (Ver Figura 3).



Figura 3: Error al seleccionar tu Perfil.

Manual del Moderador

- **Iniciando como Moderador:** El moderador debe ingresar su nombre y presionar el botón COMENZAR... (ver Figura 4). En la siguiente pantalla el moderador debe decir verbalmente o anotar su dirección IP para que los participantes puedan conectarse.

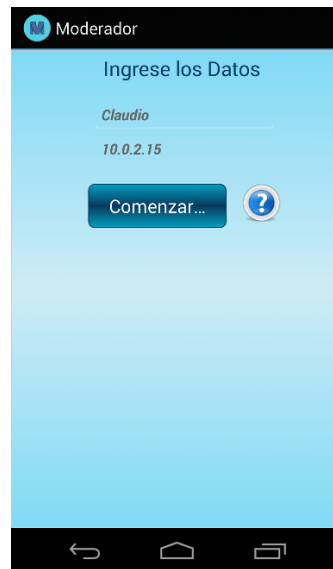


Figura 4: Iniciando como Moderador.

En caso de dudas el moderador puede pulsar el icono de ayuda (?) (ver Figura 5).

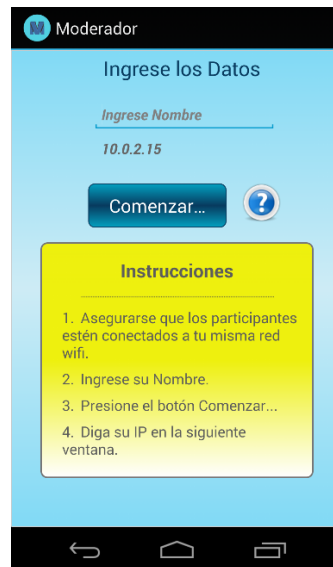


Figura 5: Indicaciones para iniciar como Moderador.

Si se pulsa el botón COMENZAR... sin haber ingresado un nombre, se despliega un mensaje de error (ver Figura 6).

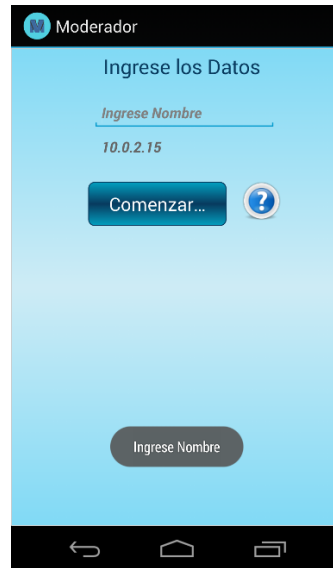


Figura 6: Error al iniciar como Moderador.

- **Creando Talleres:** Primero el moderador debe decir su IP a los participantes, para que comiencen a conectarse. Luego dar clic en el botón “Crear Taller” (ver Figura 7).

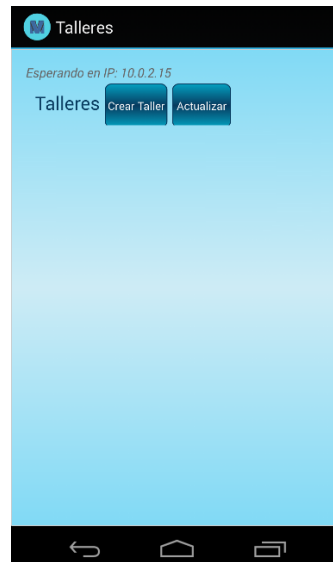


Figura 7: Sin Talleres creados.

A continuación el moderador debe ingresar el nombre del taller y si desea una descripción (ver Figura 8).

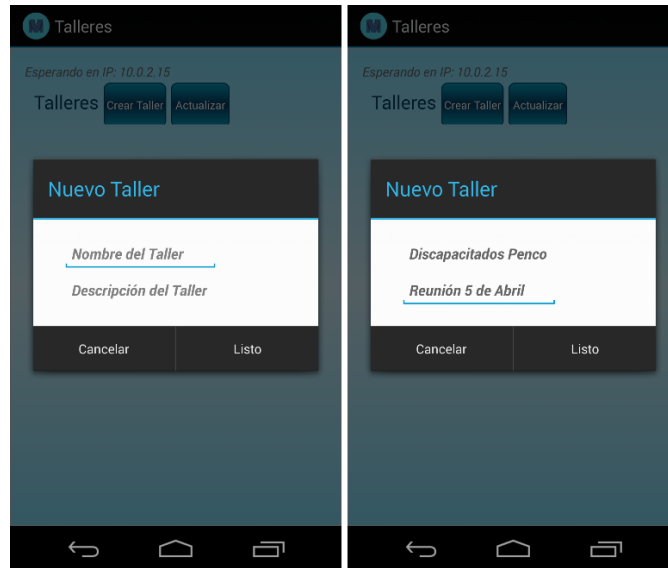


Figura 8: Creación de un Taller.

Finalmente el moderador debe ingresar y decirle a los participantes que ingresen en el taller, y a los participantes que no ven el taller, que presionen en el botón “Actualizar” (ver Figura 9).

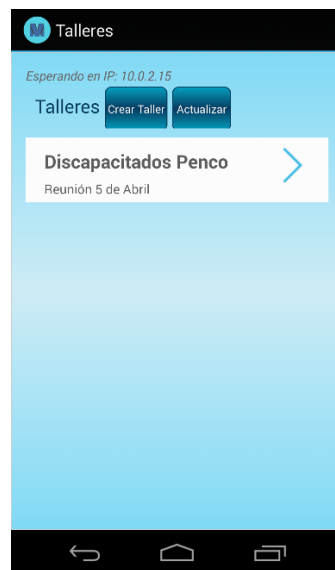


Figura 9: Desplegando el Taller creado.

- **Creando Preguntas:** El moderador debe crear la pregunta dando clic en el botón “Preguntar” (ver Figura 10). Si se crea más de una pregunta, éstas se van agregando hacia la derecha, por lo que hay que deslizar el dedo para encontrarla.

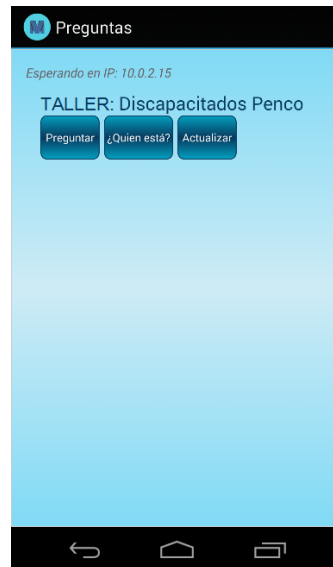


Figura 10: Sin Preguntas creadas.

Luego el moderador debe ingresar la pregunta y si desea una descripción de ésta (ver Figura 11).

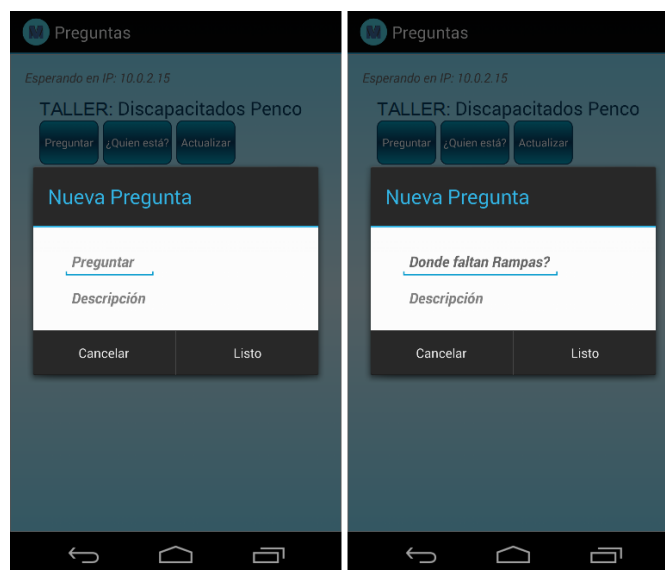


Figura 11: Creación de una Pregunta.

Finalmente el moderador debe ingresar y decirle a los participantes que ingresen en la pregunta, y a los participantes que no ven la pregunta, que presionen en el botón “Actualizar” (ver Figura 12).

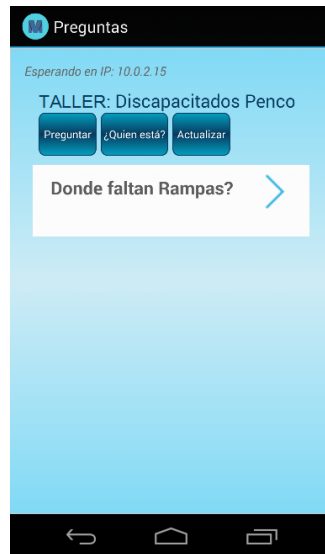


Figura 12: Desplegando Pregunta creada.

- **Esperando Respuestas:** El Moderador debe decirle a los participantes que den clic en “Responder” y esperar a que los Participantes terminen de responder. Si el moderador no ve las respuestas debe apretar el botón “Actualizar” (ver Figura 13).



Figura 13: Esperando Respuestas.

Una vez que los participantes hayan terminado de responder, se debe dar clic en el botón “Liberar” (ver Figura 14).

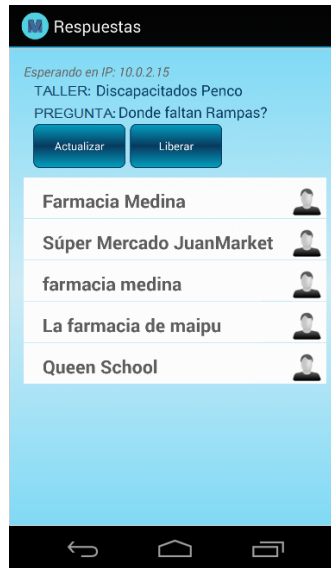


Figura 14: Recibiendo Respuestas.

Aparecerá una ventana emergente para confirmar, en la que hay que dar clic en “Liberar” (ver Figura 15).

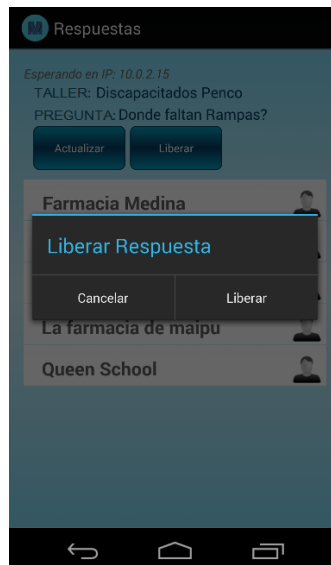


Figura 15: Liberando Respuestas.

Ahora el moderador puede empezar a trabajar sobre las respuestas, ya sea editándolas o eliminándolas, manteniendo un clic sobre la respuesta, con el fin de agrupar las respuestas similares e indicando cuanta gente la respondió (ver Figura 16).



Figura 16: Editando o Eliminando una Respuesta.

Una vez terminada la edición hay que presiona en el botón “Volver” del dispositivo (ver Figura 17). Y luego se puede finalizar la pregunta manteniendo el clic sobre ella y pulsando “Finalizar”.

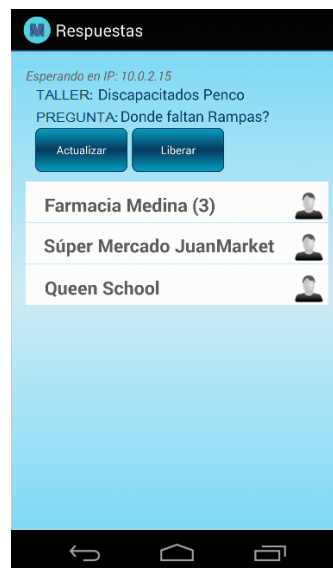


Figura 17: Respuestas Listas.

- **Continuar Preguntas:** Si el moderador crea otra pregunta, se debe deslizar hacia la derecha de la pantalla para encontrarla.

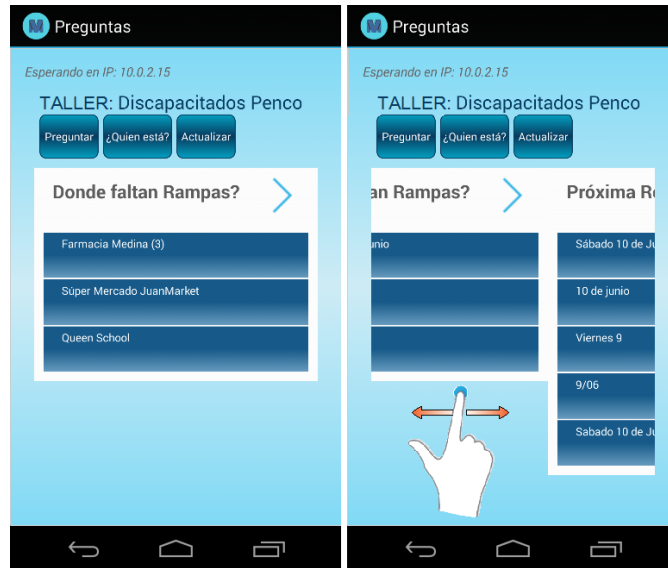


Figura 18: Pasando a otra Pregunta.

- **Finalizando Taller:** Una vez finalizadas las preguntas hay que ir a la ventana “Talleres” (ver Figura 9) y mantener clic sobre el taller, para que luego se despliegue una ventana (ver Figura 19).



Figura 19: Finalizando Taller.

Luego el moderador debe dar clic en el botón “Finalizar” y se abrirá un menú para seleccionar medio de envío (ver Figura 19), en la que debe seleccionar correo e ingresar correo de destino y origen.

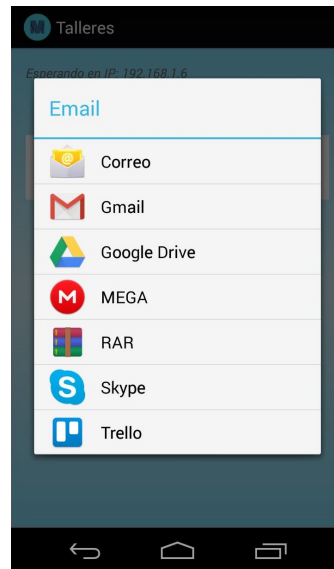


Figura 20: Enviando Taller.

Manual del Participante

- **Iniciando como Participante:** El participante debe ingresar su nombre e ingresar el IP del moderador (solicitar si no se tiene), finalmente debe presionar el botón “Conectar...” (ver Figura 21).

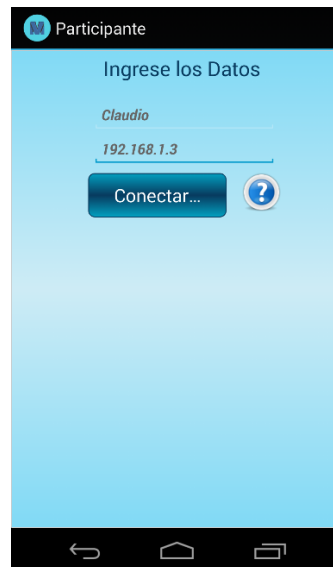


Figura 21: Iniciando como Participante.

En caso de dudas el participante puede pulsar el botón “?” (ver Figura 22).

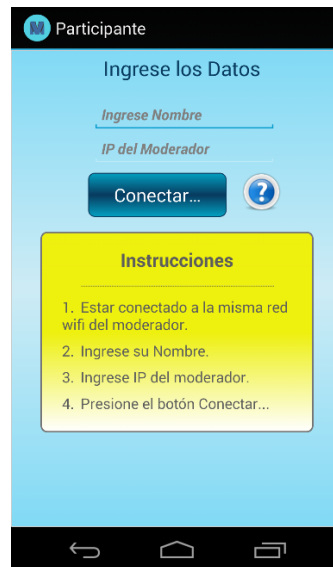


Figura 22: Indicaciones para iniciar como Participante

Si se pulsa a “Conectar...” sin haber ingresado un nombre y/o la IP del moderador se despliega un mensaje de error (ver Figura 23).

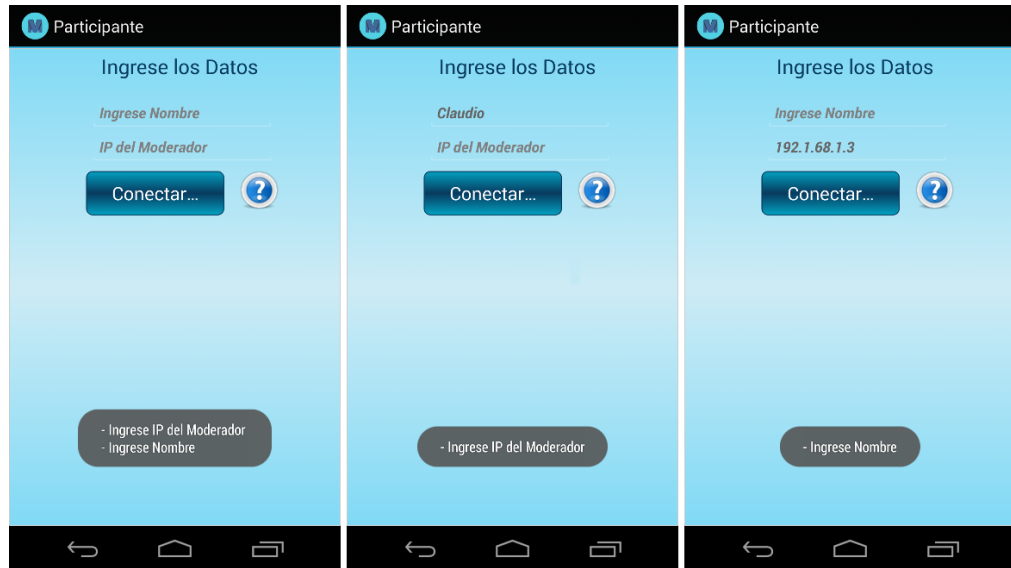


Figura 23: Errores al iniciar como Participante.

- **Esperando Taller:** Una vez conectado, el participante debe esperar a que el moderador cree el taller (ver Figura 24).

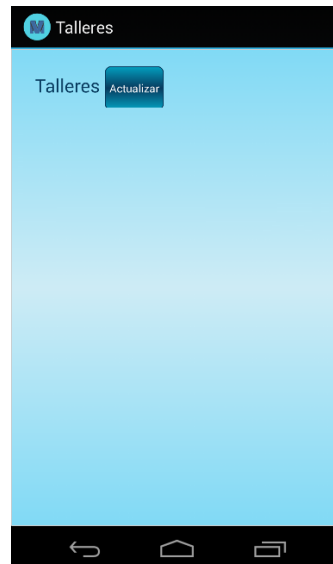


Figura 24: Esperando el Taller.

Una vez el moderador indique que el taller ha sido creado, se debe presionar el botón “Actualizar” y aparecerá el taller (ver Figura 25). Luego se debe ingresar al taller dando clic sobre él.

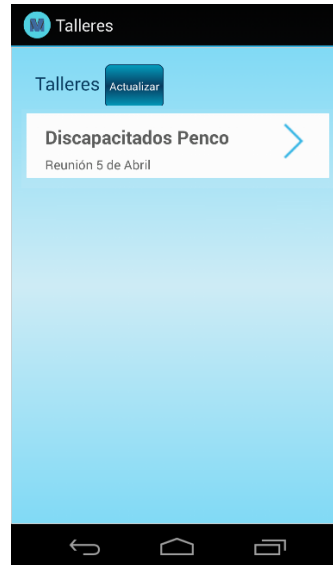


Figura 25: Desplegando Taller creado.

- **Esperando Preguntas:** Una vez que el participante ingrese a taller, debe esperar a que el moderador cree la pregunta (ver Figura 26).

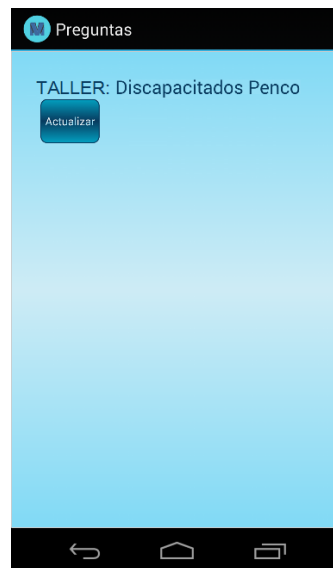


Figura 26: Esperando la Pregunta.

Una vez el moderador indique que la pregunta ha sido creada, se debe presionar el botón “Actualizar” y aparecerá el pregunta (ver Figura 27). Luego se debe ingresar a la pregunta dando clic sobre ella.

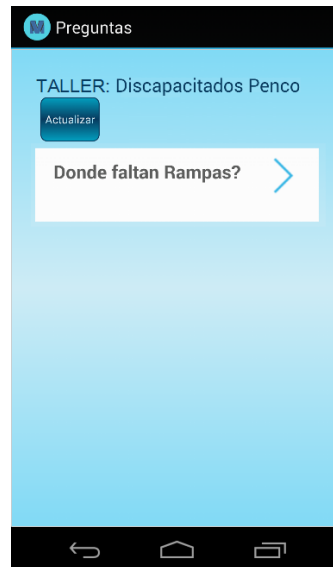


Figura 27: Desplegando pregunta creada.

- **Creando Respuestas:** El participante debe presionar el botón “Responder” (ver Figura 28).



Figura 28: Sin Respuestas.

El participante debe escribir la respuesta y presionar en el botón “Listo” (ver Figura 29).



Figura 29: Ingresando una Respuesta.

Ahora el participante puede visualizar la respuesta (ver Figura 30). Si desea puede responder nuevamente o editar su respuesta.



Figura 30: Desplegando Respuesta creada.

Pruebas de Usabilidad

Prueba de Usabilidad					
	En desacuerdo			De acuerdo	
Utilizaría con frecuencia el programa				X	
Encontré el programa muy complejo			X		
Fue fácil utilizarlo				X	
Necesitaría de un experto para utilizarlo	X				
Las diversas funciones están bien integrados			X		
Hubo demasiada inconsistencia visual	X				
Lo encontré muy difícil de usar			X		
Las personas lo aprenderían a usar rápidamente				X	
Me sentí muy confiado en la navegación			X		
Necesitaría aprender más antes de utilizarlo				X	

Tabla 1: Resultado 1 del cuestionario de usabilidad System Usability Scale (SUS).

Prueba de Usabilidad					
	En desacuerdo			De acuerdo	
Utilizaría con frecuencia el programa		X			
Encontré el programa muy complejo			X		
Fue fácil utilizarlo				X	
Necesitaría de un experto para utilizarlo		X			
Las diversas funciones están bien integrados		X			
Hubo demasiada inconsistencia visual			X		
Lo encontré muy difícil de usar			X		
Las personas lo aprenderían a usar rápidamente			X		
Me sentí muy confiado en la navegación				X	
Necesitaría aprender más antes de utilizarlo			X		

Tabla 2: Resultado 2 del cuestionario de usabilidad System Usability Scale (SUS).

Prueba de Usabilidad					
	En desacuerdo			De acuerdo	
Utilizaría con frecuencia el programa	X				
Encontré el programa muy complejo	X				
Fue fácil utilizarlo				X	
Necesitaría de un experto para utilizarlo	X				
Las diversas funciones están bien integrados				X	
Hubo demasiada inconsistencia visual		X			
Lo encontré muy difícil de usar		X			
Las personas lo aprenderían a usar rápidamente			X		
Me sentí muy confiado en la navegación			X		
Necesitaría aprender más antes de utilizarlo				X	

Tabla 3: Resultado 3 del cuestionario de usabilidad System Usability Scale (SUS).

Prueba de Usabilidad					
	En desacuerdo			De acuerdo	
Utilizaría con frecuencia el programa		X			
Encontré el programa muy complejo		X			
Fue fácil utilizarlo				X	
Necesitaría de un experto para utilizarlo	X				
Las diversas funciones están bien integrados			X		
Hubo demasiada inconsistencia visual	X				
Lo encontré muy difícil de usar			X		
Las personas lo aprenderían a usar rápidamente			X		
Me sentí muy confiado en la navegación				X	
Necesitaría aprender más antes de utilizarlo	X				

Tabla 4: Resultado 4 del cuestionario de usabilidad System Usability Scale (SUS).

Prueba de Usabilidad					
	En desacuerdo			De acuerdo	
Utilizaría con frecuencia el programa					X
Encontré el programa muy complejo	X				
Fue fácil utilizarlo				X	
Necesitaría de un experto para utilizarlo	X				
Las diversas funciones están bien integrados				X	
Hubo demasiada inconsistencia visual		X			
Lo encontré muy difícil de usar	X				
Las personas lo aprenderían a usar rápidamente				X	
Me sentí muy confiado en la navegación				X	
Necesitaría aprender más antes de utilizarlo	X				

Tabla 5: Resultado 5 del cuestionario de usabilidad System Usability Scale (SUS).

Prueba de Usabilidad					
	En desacuerdo			De acuerdo	
Utilizaría con frecuencia el programa			X		
Encontré el programa muy complejo			X		
Fue fácil utilizarlo				X	
Necesitaría de un experto para utilizarlo	X				
Las diversas funciones están bien integrados			X		
Hubo demasiada inconsistencia visual		X			
Lo encontré muy difícil de usar	X				
Las personas lo aprenderían a usar rápidamente				X	
Me sentí muy confiado en la navegación				X	
Necesitaría aprender más antes de utilizarlo		X			

Tabla 6: Resultado 6 del cuestionario de usabilidad System Usability Scale (SUS).