



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

Sistema para la automatización del proceso de creación, despliegue y gestión de sondeos en línea

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por el
Br. Ignacio A. Córdoba P. y el Br. Lino T. Hernández Z.

Para optar al título de Licenciado en Computación

Tutor: Prof. Franklin J. Sandoval S.

Mayo 2015

Acta de Veredicto

Quienes suscriben, miembros del jurado designado por el Consejo de Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela para examinar el Trabajo Especial de Grado de los Bachilleres Ignacio Córdoba, titular de la cédula de identidad No. 17.928.657, y el bachiller Lino Hernández, titular de la cédula de identidad No. 16.522.659 bajo el título: **“Sistema para la automatización en el proceso de creación, despliegue y gestión de sondeos”**, a fines de cumplir con el requisito legal para optar al grado de Licenciado en Computación, dejan constancia de lo siguiente:

Una vez suficientemente leído este trabajo por cada uno de los miembros del jurado, se fijó el día _____ fecha para la defensa de manera pública. Mediante una exposición oral de este Trabajo Especial de Grado, realizada en la Escuela de Computación, en la fecha acordada, luego de lo cual respondió satisfactoriamente a las preguntas que le fueron formuladas por el jurado, todo ello conforme dispuesto a la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela.

El jurado decidió APROBARLO con la nota _____ puntos. En fe de lo cual se levanta la presente acta a los ____ días del mes de Mayo de 2015.

Prof. Franklin Sandoval (Tutor)

Prof. Mercy Ospina (Jurado)

Prof. Marizé Mijares H (Jurado)

Agradecimientos

Doy mis más profundos agradecimientos a todo el equipo docente de la Escuela de Computación, por su vocación, compromiso y constante lucha para seguir formando venezolanos de bien en tiempos cada vez más difíciles.

A mi compañero Nacho, por conseguir esta inigualable oportunidad de trabajo.

También agradezco a nuestro tutor Franklin Sandoval, sin su enorme generosidad y apoyo incondicional alcanzar este sueño no hubiera sido posible.

Doy gracias a mis padres, Ana y Lino, por su amor incondicional y por ser mi sustento durante gran parte de esta jornada. Agradezco también a mi hermana Claudia, por enseñarme el valor del trabajo duro y honesto, lección que me llevó a conseguir este gran logro personal. Finalmente agradezco a mi novia Yineska, por todo el cariño, el apoyo y la inspiración a dar siempre lo mejor de mí.

Lino.

Mi gran agradecimiento a un grupo de personas dedicadas a la evolución científica, tecnológica y profesional de este país, profesionales admirables y de ejemplo a seguir, gracias equipo docente y administrativo de la Escuela de Computación.

Gracias Profesor Franklin Sandoval, formador fundamental en este capítulo de mi vida. Una persona que deja muy claro el significado de ayudar al prójimo incondicionalmente. A pesar de que nuestros caminos se cruzaron al final de este recorrido, queda grabado como un gran profesional y docente; será un honor ser su colega.

A mi compañero Lino, por tener la paciencia de aguantarme y ayudarme no por una, sino dos Tesis.

A mi familia, la heredada y la elegida, los Papatzikos y los Popovich, han sido un apoyo incondicional y necesario para finalizar esa etapa de mi vida, por siempre agradecido por el gran empujón.

A mi Padre, conocedor del trabajo duro y de la perseverancia. Dos elementos muy importantes para obtener resultados positivos en la vida.

A mi Hermano, mi gran amigo, mi compañero de andanzas e inventos. Gracias por darme otro punto de vista de la palabra sacrificio, y enseñarme a creer en mi mismo por encima de todo.

A mi Madre, mi ejemplo a seguir, mi primera y por siempre tutora; gracias por tener la paciencia para ayudarme a lograr dar este gran paso.

A mi novia, la Licenciada Jacqueline Popovich, gracias por estar, por entender, por ayudar, sobre todo por no tirar la toalla. Hoy inicia nuestro camino como compañeros profesionales, como colegas. Y así como cierro un capítulo, es el momento de abrir uno nuevo en nuestras vidas...

Ignacio.



**UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN**

**Sistema para la automatización en el proceso de creación,
despliegue y gestión de sondeos online**

Autores: Br. Ignacio A. Córdoba P.
Br. Lino T. Hernández Z.

Resumen

El presente Trabajo Especial de Grado consiste en el desarrollo e implementación de un sistema capaz de automatizar el proceso de creación, despliegue y gestión de sondeos en línea. Este sistema, que lleva por nombre Taima, está dividido en dos módulos: un módulo administrativo web que permite la creación y publicación de los sondeos, y un módulo de opinador móvil que permite el despliegue y consumo de los sondeos publicados. Para el desarrollo de esta solución se aplicó la metodología de desarrollo SCRUM. Las tecnologías principales utilizadas son: NodeJS y AngularJS (desarrollo web), Java para Android (desarrollo móvil) y MongoDB (sistema manejador de bases de datos). Como resultado se elaboró una propuesta para la gestión de sondeos en línea, que da solución a los problemas presentados y proporciona seguridad, control y respaldo a la data administrada en los sondeos, reduciendo los tiempos y optimización del proceso de gestión de las encuestas en todas sus etapas.

Palabras clave: Sondeo, Opinador, Computación móvil, Aplicación web.

Índice General

CAPÍTULO I	1
PROBLEMA DE LA INVESTIGACIÓN	1
1.1 Planteamiento del problema	1
1.2 Objetivo general	2
1.3 Objetivos específicos	2
1.4 Justificación	2
1.5 Solución propuesta	3
1.6 Alcance	4
CAPÍTULO II	5
MARCO CONCEPTUAL	5
2.1 Sondeos	5
2.1.1 Fortalezas de los Sondeos	6
2.1.2 Limitaciones de los Sondeos	6
2.1.3 Objetivos a plantearse para generar Sondeos	7
2.1.4 Sondeos de Opinión	8
2.2 Sistema de información	10
2.3 Desarrollo de aplicaciones web	11
2.3.1 Herramientas para el desarrollo (lado del cliente)	12
a. HTML (Hyper Text Markup Language)	12
b. CSS (Cascading Style Sheets)	13
c. JavaScript	15
d. AngularJS	16
2.3.2 Herramientas para el desarrollo (lado del servidor)	18
a. Nginx	19
b. Node.js	20
2.3.3 Desarrollo de servicios Web	21
a. REST	22
2.4 Base de datos no relacionales (NoSQL)	23
2.4.1 Escalabilidad horizontal	24

2.4.2 MongoDB	25
2.5 Desarrollo de aplicaciones móviles.....	26
2.5.1 Android	27
CAPITULO III	31
MARCO METODOLOGICO	31
3.1 Metodología AGILE.....	31
3.1.1 SCRUM	32
a. Roles de SCRUM	34
b. Eventos del Scrum.....	35
c. Artefactos del Scrum.....	36
CAPÍTULO IV.....	38
MARCO APLICATIVO	38
4.1 Modelo de desarrollo de software.....	38
4.2 Análisis y diseño general	39
4.2.1 Requerimientos funcionales (casos de uso)	39
4.2.2 Requerimientos no funcionales	54
4.2.3 Diseño general de la Solución	54
4.3 Construcción de la solución	55
4.3 Pruebas y aceptación	86
4.3.1 Pruebas de usuario final.....	86
CAPÍTULO V.....	97
CONCLUSIONES Y RECOMENDACIONES.....	97
REFERENCIAS BIBLIOGRÁFICAS.....	99

Índice de Figuras

Figura 1	Arquitectura de la solución.....	3
Figura 2	Ejemplo de código HTML.....	13
Figura 3.	Ejemplo de CSS.....	14
Figura 4.	Ejemplo de estilos CSS e identificadores.	15
Figura 5.	Ejemplo de código JavaScript.	16
Figura 6.	Importando AngularJS.	17
Figura 7.	Trabajando con AngularJS	18
Figura 8.	Código JavaScript para crear un servidor en NodeJS.....	21
Figura 9.	Proceso de SCRUM.....	37
Figura 10.	Casos de uso módulo administrativo	40
Figura 11.	Casos de uso módulo opinador.	47
Figura 12.	Arquitectura de la solución.	55
Figura 13.	Modelo de la base de datos.....	58
Figura 14.	Pantalla de inicio de sesión (administrador).....	59
Figura 15.	Vista de pantalla administrativa de sondeos.	61
Figura 16.	Creación de sondeo.....	62
Figura 17.	Formulario de creación de sondeo, parte 2.	63
Figura 18.	Código QR.	65
Figura 19.	Corte de imagen en creación de sondeo.	66
Figura 20.	Creación de pregunta.	67
Figura 20.	Segmentación de sondeo.	68
Figura 22.	Botones de resultados y exportar resultados.	70
Figura 23.	Resultados de un sondeo.	71
Figura 24.	Formato de exportar resultados.....	72
Figura 25.	Android Studio. Editor de código Android + Compilador.	73
Figura 26.	Pantalla inicial de la aplicación.	74
Figura 27.	Pantalla de registro de opinador.	75
Figura 28.	Pantalla de inicio de sesión.	76
Figura 29.	Menú de navegación.	77

Figura 30. Pantalla de sondeos (opinador).....	78
Figura 31. Pantalla de rechazo de pregunta.....	79
Figura 32. Pantalla de información de sondeo.	80
Figura 33. Pantalla de pregunta (móvil).....	81
Figura 34. Notificación Toast en Android.....	82
Figura 35. Lectura de QR.....	84
Figura 36. Perfil de opinador.....	85
Figura 37. Editar perfil opinador.....	86
Figura 38. La interfaz es fácil de entender.....	90
Figura 39. Fácil de aprender.....	90
Figura 40. Fácil de usar.	91
Figura 41. Los colores son agradables.....	91
Figura 42. La experiencia con la interfaz fue positiva.....	92
Figura 43. Es comprensible.	92
Figura 44. Es útil para lograr llevar a cabo un objetivo.....	93
Figura 45. Las funciones del menú son claras.	93
Figura 46. Los mensajes aportan significado.	94
Figura 47. Facilita las búsquedas.	94
Figura 48. Responder sondeos se hace fácilmente.....	95
Figura 49. El tiempo para la realización de las respuestas es óptimo.....	96

Índice de Tablas

Tabla 1. Login de usuario.....	41
Tabla 2. Consultar sondeos.....	42
Tabla 3. Crear sondeos.....	43
Tabla 4. Segmentación de sondeo.....	44
Tabla 5. Resultados de sondeo.....	45
Tabla 6. Exportar resultados.....	46
Tabla 7. Registro de opinador.....	48
Tabla 8. Inicio de sesión de opinador.....	49
Tabla 9. Consultar perfil.....	50
Tabla 10. Editar perfil.....	51
Tabla 11. Consultar sondeos (opinador).....	52
Tabla 12. Responder sondeo.....	53
Tabla 13. Preguntas sobre interfaz gráfica.....	87
Tabla 14. Preguntas sobre información de interfaz.....	88
Tabla 15. Preguntas opciones de usuario.....	89

Introducción

La opinión pública representa una de las principales fuentes de crecimiento de una organización orientada a prestar algún servicio. Entender el comportamiento, los intereses y las preocupaciones de un grupo de personas puede brindar información clave a una organización que pretenda atender las necesidades de dicho grupo, así como identificar debilidades y fortalezas en los procesos actuales de la empresa. Esta información permitiría a dicha organización refinar su proceso de toma de decisiones y como consecuencia, mejorar sus resultados (Rea & Parker, 2014).

A pesar de ser información valiosa, es común que estas instituciones o empresas fracasen en la obtención de dicho conocimiento o simplemente no incurran en la búsqueda del mismo. Las razones varían entre: el alto costo económico y humano que estas iniciativas puedan representar, la baja participación por parte de los opinadores que son el objetivo principal, la predisposición de los opinadores debido al factor humano, la sensación de poca privacidad por parte del opinador y la más importante, la inversión de tiempo tanto del opinador como de la organización interesada.

La tecnología de la información genera un punto evolutivo en la forma en la que se realizan los sondeos. El manejo de sondeos en línea ha surtido un efecto beneficioso para el opinador ya que resuelve varios factores de resistencia según Cowles (2015):

- Se eliminan intermediarios y como consecuencia se elimina su sesgo, reduce los costos de despliegue y minimiza los errores generados en la recopilación de resultados;
- El opinador decide cuándo y dónde realiza el sondeo lo que maximiza su privacidad y anonimato;
- Permite un mayor despliegue lo que aumenta el alcance de la población que participa

Así, el presente Trabajo Especial de Grado (T.E.G) utiliza el enfoque online para proponer una solución web y móvil que permita la creación, despliegue y gestión de sondeos.

Con el propósito de facilitar el entendimiento de este T.E.G se ha estructurado el documento de la siguiente manera:

Capítulo I - Problema de investigación: Se desarrolla una breve explicación de la problemática que pretende resolver el presente trabajo. Además contiene la declaración de los objetivos generales y específicos, así como el alcance, la justificación y la importancia de la investigación.

Capítulo II - Marco conceptual: Se exponen de forma breve todos los conceptos fundamentales involucrados en el desarrollo del presente T.E.G. Partiendo con la definición de los sondeos y sus características, continuando con nociones básicas sobre sistemas de información y posteriormente conceptualizando y desarrollando brevemente las diferentes tecnologías utilizadas para la construcción de este proyecto.

Capítulo III - Marco Metodológico: En este capítulo se profundiza sobre la metodología de desarrollo de software empleada en la elaboración del presente trabajo.

Capítulo IV - Marco Aplicativo: Este capítulo detalla el proceso de desarrollo de software enmarcado dentro de la metodología descrita en el Capítulo III y aplicando las tecnologías estudiadas en el Capítulo II.

Capítulo V - Conclusiones y Recomendaciones: Presenta un resumen de las metas alcanzadas por el desarrollo de la solución tecnológica propuesta, un análisis concluyente sobre el presente T.E.G. y recomendaciones para posibles trabajos futuros.

CAPÍTULO I

PROBLEMA DE LA INVESTIGACIÓN

En este capítulo se detalla el problema que justifica la realización de este T.E.G, describiendo la situación actual que enfrentan los servicios y empresas de opinión. Así mismo, se detallan el objetivo general y los objetivos específicos, también el alcance, la importancia y la justificación de la solución planteada.

1.1 Planteamiento del problema

Los sondeos son herramientas de mucho poder para el saber de la opinión pública, y no tienen implicaciones únicamente en relación a sus resultados, sino a la interpretación de su sentido. Varían según su ejecución y dependen de cómo se conduce la recolección de información, la forma de preguntas y el orden.

Las empresas o entes interesados en conocer la opinión pública, tienen la gran tarea de recopilar y analizar los resultados, que dependiendo del tamaño de la muestra necesarias, un gran número de participantes (opinadores) para lograr construir datos representativos de la población, puede ser muy costoso tanto financieramente como en tiempos, ya que debe disponerse de un amplio despliegue humano para alcanzar tal objetivo.

Otro factor importante es la locación donde se realiza, ya que esto determina los implementos y costos de movilización para realizar dicho despliegue. Por último, el mismo factor humano está muy ligado a los resultados durante la ejecución, lo que puede predisponer al opinador a la hora de realizar las preguntas y obtener las respuestas.

Por lo anteriormente expuesto nuestra principal interrogante es: ¿Cómo preguntar a la mayor cantidad de personas, en el menor tiempo posible y con el menor costo?

1.2 Objetivo general

Desarrollar un sistema web y móvil para la automatización del proceso de creación, despliegue y gestión de sondeos para la empresa Taima.

1.3 Objetivos específicos

- Conceptualizar los requerimientos para el desarrollo de una solución web y móvil para la automatización del proceso de creación, despliegue y gestión de sondeos.
- Desarrollo de la aplicación web para la gestión y despliegue de sondeos.
- Desarrollo de la aplicación móvil para el despliegue y consumo de sondeos.
- Realizar pruebas a los distintos módulos de software desarrollados.

1.4 Justificación

El sistema propuesto permite mejorar los tiempos de ejecución y disminuir los costos al sustituir el factor humano por interfaces web y móviles. Permite también hacer un extenso despliegue ya que sólo depende de que el opinador posea un dispositivo móvil para interactuar y la mínima interacción web del lado administrativo. Reduce al mínimo la interacción humana lo que garantiza la objetividad en la forma en que se realiza el sondeo, evitando así influencias no deseadas. Y por último recopila los resultados obtenidos en una base de datos centralizada lo que simplifica su proceso de extracción para análisis.

1.5 Solución propuesta

De acuerdo a lo expuesto en los apartados anteriores, en el presente T.E.G se propone el desarrollo de un sistema que permita la automatización del proceso de creación y consumo de sondeos. La siguiente Figura muestra la arquitectura de la solución propuesta.



Figura 1 Arquitectura de la solución.

Como se observa en la Figura, el sistema posee dos frentes a través de los cuales pueden interactuar los usuarios. El primero es una aplicación web mediante la cual es posible acceder a las funcionalidades necesarias para la creación y despliegue de sondeos. El segundo frente es una aplicación nativa para teléfonos inteligentes con sistema operativo Android, a través de este

frente los usuarios (opinadores) podrán registrarse y responder a los sondeos.

Estos frentes se comunican con un servidor de proxy inverso, provisto por la herramienta Nginx, el cual redirige sus peticiones a diferentes instancias de un servidor web implementado sobre el entorno de ejecución NodeJS. Estas instancias implementan la lógica central de la aplicación y la exponen a través de servicios web REST para su posterior consumo por parte de ambos clientes.

Además, para el caso del módulo de administración, el servidor web provee los recursos necesarios para el despliegue de una aplicación web en un navegador, como por ejemplo, archivos HTML, CSS y JavaScript.

Finalmente, la lógica implementada por el servidor web implica realizar operaciones de inserción, consulta, actualización y eliminación sobre un servidor de base de datos MongoDB.

En el Capítulo II se describen con mayor detalle las tecnologías utilizadas con la finalidad de justificar su elección.

1.6 Alcance

El sistema a desarrollar debe garantizar la creación de sondeos lineales, en donde se pueda seleccionar los tipos de respuesta: selección simple, selección múltiple y texto abierto. Que exista un proceso de publicación de los sondeos a un universo filtrado por una segmentación utilizando los datos básicos de identidad. Poder consultar los resultados del sondeo para su futuro análisis. Debe garantizar además la implementación de tecnologías móviles para el registro y validación de datos de usuarios, y consumo de los sondeos por los mismos.

Todo esto utilizando la red del internet como canal de comunicación para garantizar un despliegue extenso.

CAPÍTULO II

MARCO CONCEPTUAL

En este capítulo se estudian los conceptos sobre los cuales está fundamentado el presente T.E.G. Primero se desarrolla el concepto de sondeo; se mencionan sus características, ventajas y desventajas como herramienta para obtener información relevante de un grupo de personas. Posteriormente se estudia el concepto de Sistema de Información, el cual es el tipo de sistema a desarrollar en el presente trabajo.

Seguidamente, se visitan temas relacionados a los aspectos técnicos del desarrollo del presente trabajo, primero elaborando sobre el desarrollo de aplicaciones web, diferenciando el desarrollo del lado del cliente web del desarrollo del lado del servidor y haciendo hincapié en las tecnologías utilizadas. Posteriormente se da una breve reseña al proceso de desarrollo de servicios web REST, ya que constituyen el punto de acceso a las funcionalidades básicas del sistema propuesto.

A continuación, se estudia brevemente el concepto de base de datos NoSQL, haciendo énfasis en MongoDB, y las razones por las cuales se eligió esta tecnología.

Por último, se exploran conceptos fundamentales del desarrollo de aplicaciones para el sistema operativo Android y se describen algunas bondades del sistema operativo móvil seleccionado para el presente T.E.G.

2.1 Sondeos

El sondeo es una técnica o método de aprendizaje sobre las situaciones, experiencias, problemas y perspectivas de las personas. Es un acercamiento rápido e informal, que es simple de aplicar. Ofrece un gran potencial para

construir una alta participación de una comunidad para la recopilación de información, su posterior análisis y las distintas aplicaciones. En este sentido, es diferente a la encuesta que se enfoca más en extraer data de objetivos de investigación. El sondeo puede proveer una visión más profunda de una situación así como estimular el interés de la comunidad y responsabilidad sobre los resultados (Butler, 1995).

Según Butler, existen fortalezas y limitaciones sobre los sondeos.

2.1.1 Fortalezas de los Sondeos

- Ofrece un entendimiento profundo de una situación o problema específico.
- Disminuye la influencia (cultural, étnica, social, etc.)
- Rápida canalización de la data.
- Aplica para distintos grupos de población.
- Es efectivo en costos
- Facilita la participación y la comunicación amplia.
- Es cualitativo. Riqueza de la data.
- Construye una relación sólida entre profesionales y población local.
- Refuerza los lazos entre la organización y el cliente.
- Análisis y recopilación de la data no es asignada a un individuo
- Sondeos focalizados son utilizados para hacer seguimiento.

2.1.2 Limitaciones de los Sondeos

- No es adecuado para hacer análisis estadístico.
- No es posible generar todas las respuestas
- Limita la posibilidad de identificar algunos descubrimientos con grupos o participantes específicos.
- Si no posee un foco claro, puede no generar las especificaciones necesarias

- No asegura que todas las perspectivas y situaciones son reflejadas
- No garantiza el trato igualitario a todos los opinadores.
- Los procesos de recopilación y análisis requieren más tiempo.
- Frustran análisis cuantitativos de data por la falta de precisión y predicción.
- Si se aplica sin cuidado genera el efecto contrario al *rapport*, el opinador podría sentirse usado.

En su estudio, Butler define unos objetivos a plantearse al momento de generar un sondeo.

2.1.3 Objetivos a plantearse para generar Sondeos

- Determinar objetivos de reconocimiento rápido:
 - Conocer la situación actual
 - Identificar soluciones localmente aceptables a problemas
 - Construir interés y compromiso público
 - Generar ideas para productos o programas
 - Aclarar resultados de una encuesta cuantitativa.
 - Explorar actitudes y percepciones (ideas, productos y soluciones)
- Establecer prioridades en el acercamiento:
 - ¿Quiénes son afectados?
 - ¿Quiénes estarán interesados en los descubrimientos?
 - ¿Quiénes deben estar involucrados?
 - ¿Qué información está disponible?
 - ¿Cómo serán utilizados los descubrimientos?
 - ¿Cuándo serán necesitados los descubrimientos?
- Determinar el uso de los descubrimientos:
 - ¿Para quién son los descubrimientos?
 - ¿Cómo debe ser presentado?

- ¿Deben ser complementados con otra información?
- Explorar la necesidad de métodos complementarios para el reconocimiento rápido:
 - Data existente
 - Casos de estudio
 - Observaciones
 - Foros

2.1.4 Sondeos de Opinión

Uno de los principales usos de los sondeos es para conocer la opinión pública en diferentes temas o ámbitos. Son especialmente efectivos al dar a conocer la opinión del público con respecto a una marca o el marketing de la misma, así como la opinión política. (CEO, 2001)

CEO habla del sondeo como una investigación, que en la mayoría de los casos es una investigación social. Debe recordarse que es posible hacer una encuesta por muestreo sobre, por ejemplo, características de ciertos minerales, lo que no sería un caso de investigación social.

Existen diferentes formas de llevar a cabo un sondeo:

Sondeos de opinión personales: Al ser un tipo de encuesta cara a cara, con la encuesta personal siempre se puede obtener información adicional relevante para el estudio de que se trate, por ejemplo sobre el grado de colaboración del encuestado –correlacionado con la veracidad de sus respuestas- y además se puede aclarar cualquier duda que no influya en las respuestas. En investigación de mercado, siempre se puede mostrar un producto para evaluar su recordación, para que sea evaluado, etc. (CEO, 2001)

La principal desventaja de este tipo de encuesta cara a cara es su alto costo. En el caso de encuestas preelectorales y encuestas relacionadas con

campañas políticas, es conveniente hacer el esfuerzo económico para ganar confiabilidad en los resultados.

Otra desventaja se presenta en el sesgo que se pueda aplicar a los opinadores, por ejemplo, para indagar la opinión de sectores que no tienen teléfono o de difícil acceso, como ciertos jóvenes, etc. Estas encuestas tienen una alta tasa de respuesta y su empleo está científicamente poco fundamentado, a pesar de que se emplea con relativa frecuencia. Los sesgos son más que evidentes, pues la muestra se conforma con quienes aciertan a pasar por allí, y a responder. Además, el lugar donde se realice el abordaje condiciona la tasa de respuesta y la colaboración, para no mencionar el resto del universo que no pasa por allí en esas horas. (CEO, 2001)

Sondeos de opinión vía correo electrónico o internet: como su nombre lo sugiere, se utilizan canales electrónicos para el despliegue de los sondeos a los opinadores.

Un buen sondeo de opinión por correo postal o electrónico es la que parte de una lista completa de las personas a encuestar, o de las direcciones electrónicas de contacto de empresas, etc. que forman parte de la población o universo a sondear. Adicionalmente, se tratará de realizar un buen abordaje a los futuros encuestados a fin de garantizar su colaboración con auténticas opiniones. Una de las grandes ventajas de las encuestas que se envían por mail es, además de su bajo costo, como se ha señalado ya varias veces, es que permite llegar de manera inmediata a una población o grupo de personas bajo estudio aún cuando están muy dispersas geográficamente. (CEO, 2001).

Una de las principales ventajas de las encuestas “online” que presentan los sitios web, por ejemplo, sitios de noticias o diarios, es que su costo es muy bajo, y la obtención de resultados sobre la opinión acerca de algún tema siempre es muy rápida también.

Sin embargo, estas encuestas suelen presentar sesgos, determinados porque la muestra se autoselecciona, o sea, responde quien quiere participar de la encuesta. Si a esto se le suma que los seguidores de un determinado sitio web tienen un target o perfil particular, la confiabilidad de los resultados de la encuesta son dudosos, y las opiniones tienden a coincidir con las del sitio web. (CEO, 2001)

Tomando el lugar de publicación como principal desventaja, podríamos asumir que si se publica el sondeo en un espacio neutral, sin opiniones, es posible aliviar el sesgo y obtener mejores resultados.

2.2 Sistema de información

Para entender que es un sistema de información, se tomarán en cuenta algunas de las siguientes definiciones:

También se puede definir sistema como un “conjunto de cosas que relacionadas entre sí ordenadamente, contribuyen a determinar un objeto.” (RAE, 2010). Ajustando el concepto de sistema a la computación, un sistema son un conjunto de componentes o elementos que se relacionan entre sí para llegar a un fin común.

Según Davenport y Prusak (2000) definen información como: “un conjunto de datos que hacen diferencia”. Existe una diferencia importante entre datos e información, principalmente dada porque la información depende de los datos ya que la información es la relevancia de los datos. La información también es capaz de generar o reforzar el conocimiento.

Según Laudon y Laudon (1998) un sistema de información se define como un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control en una institución. Además, pueden también ayudar a los

administradores y al personal a analizar problemas, visualizar cuestiones complejas y crear nuevos productos.

Con base a la definiciones anteriores podemos decir que un sistema de información es un conjunto de elementos o componentes que se relacionan entre sí, cuyo fin es la adquisición u obtención de información a través de un conjunto de datos.

Para Retallick (2006) existen tres tipos de sistemas de información, sistemas que están diferenciados a través de distintas características, tales como su función y a qué tipo de roles empresariales van dirigidos o quienes manejan y utilizan el sistema.

2.3 Desarrollo de aplicaciones web

Como se mencionó anteriormente, a partir de este punto y en lo que resta de Capítulo se estudiarán brevemente las diferentes tecnologías que componen a la solución propuesta en el apartado 1.5. El primer punto a tocar será el desarrollo de aplicaciones web, el cual abarca las tecnologías utilizadas para el desarrollo del módulo administrativo de sondeos del presente T.E.G.

Una aplicación web es todo software que se ejecuta en un navegador web. Su gran ventaja es que elimina la necesidad de tener que distribuir e instalar la aplicación en las máquinas de sus usuarios (Luján, 2002).

Las aplicaciones web pueden ser vistas como una implementación de la arquitectura cliente-servidor, en donde el cliente es descargado y ejecutado por el navegador web y envía o recibe información hacia o desde el servidor web mediante HTTP.

En general, las aplicaciones web están compuestas de una combinación de código HTML, JavaScript y CSS, los cuales pueden ser ejecutados por los navegadores web. Además, es común que el servidor web requiera generar dinámicamente este contenido, así como ejecutar lógica adicional, por

ejemplo el manejo de las sesiones de los usuarios o la comunicación con la base de datos de la aplicación.

2.3.1 Herramientas para el desarrollo (lado del cliente)

En el contexto de las aplicaciones web, cuando se habla de desarrollo del lado del cliente normalmente es para referirse a la construcción de las interfaces gráficas con las que interactuará el usuario final a través de un navegador web.

Para el desarrollo del este T.E.G. se utilizaron HTML, JavaScript y CSS como tales tecnologías en conjunto con el *framework* AngularJS.

a. HTML (Hyper Text Markup Language)

Es un lenguaje de marcado especializado en la realización de páginas web. Un documento HTML define la estructura y el contenido de una página web. Los documentos HTML están compuestos por etiquetas HTML y cada etiqueta HTML describe los distintos elementos de esta página web. (Refsnes Data, 2015)

Las etiquetas HTML están compuestas por palabras claves rodeadas por corchetes angulares (< y >). En general, las etiquetas HTML vienen en pares compuestos por una etiqueta de apertura e.g. <p> y una etiqueta de cierre </p>. Nótese que la etiqueta de cierre se diferencia por la presencia del carácter “/” después del primer corchete. A continuación se muestra un ejemplo de código HTML:


```
<html>
<head>
  <title>Título de la página</title>
</head>
<body>
  <h1>Esto es un encabezado</h1>
  <p>Esto es un párrafo</p>
</body>
</html>
```

Figura 2 Ejemplo de código HTML.

Explicación del ejemplo:

- El texto entre **<html>** y **</html>** describe el documento HTML.
- El texto entre **<head>** y **</head>** provee información general sobre el documento.
- El texto entre **<title>** y **</title>** contiene el título del documento.
- El texto entre **<body>** y **</body>** describe el contenido visible dentro de la página web.
- El texto entre **<h1>** y **</h1>** describe un encabezado.
- El texto entre **<p>** y **</p>** describe un párrafo.

Utilizando esta descripción, un navegador web puede mostrar un documento con un encabezado y un párrafo.

b. CSS (Cascading Style Sheets)

CSS es una tecnología que permite definir el “cómo” se representan los elementos visuales de un documento HTML. La representación de dichos elementos en un explorador web viene definida por su atributo “estilo (*style*)” y CSS provee una manera eficiente de configurar el estilo de un elemento o un conjunto de elementos con alguna característica en común. Esta característica es particularmente útil al momento de mantener aplicaciones

web grandes que están compuestas de muchos documentos HTML. (Refsnes Data, 2015)

La definición de los estilos normalmente se encuentra dentro de un archivo “.css”. Este archivo es referenciado dentro de los documentos HTML en donde se requiera aplicar los estilos definidos.

Una regla de estilo en CSS consiste de un selector y de un bloque de declaración de estilo. El selector apunta a él o los elementos HTML a los que se aplicará el estilo y el bloque de declaración está compuesto por un conjunto de pares atributo-valor. A continuación se muestra un ejemplo de la sintáxis de una regla CSS:

```
p {  
    color:red;  
    text-align:center;  
}
```

Figura 3. Ejemplo de CSS.

El selector “p” implica que el estilo definido será aplicado a todos los párrafos del documento HTML (definidos por etiquetas <p> y </p>). El bloque declarativo siempre estará rodeado por llaves ({, }) y cada par atributo-valor estará seguido de un caracter punto y coma (;). Para este ejemplo en particular, los caracteres definidos dentro de un párrafo tendrán una fuente de color rojo (color:red) y estarán centrados (text-align:center).

El selector, además de identificar a los elementos de un documento HTML por su tipo, también lo puede hacer por su atributo “class” (clase) e “id” (identificador). También puede hacer composiciones de selecciones y agrupar diferentes criterios de selección para minimizar el código escrito. Por ejemplo:

```
p.centrados, #id1,  
.clase1, h1 {  
    color:red;  
    text-align:center;  
}
```

Figura 4. Ejemplo de estilos CSS e identificadores.

Aplicará el estilo a los párrafos con class="centrados" (p.centrados), a los elementos con id="id1" (#id1), a todos los elementos con class="clase1" y a los elementos del tipo h1 del documento HTML.

c. JavaScript

Es un lenguaje de programación que puede ser interpretado y ejecutado por los navegadores web con la finalidad de, a grosso modo, alterar los atributos y estilos de los elementos dentro de un documento HTML. (Refsnes Data, 2015)

El código, generalmente, se encuentra dentro de un archivo ".js" referenciado por un documento HTML, aunque también puede encontrarse dentro del cuerpo del HTML, específicamente dentro de las etiquetas <script> y </script>.

JavaScript define funciones que son llamadas en respuesta a ciertos eventos de la vista web definidos dentro del mismo HTML. Ejemplos de estos eventos son:

- La carga de una página web por primera vez.
- Hacer clic sobre un botón.
- Posicionar el cursor sobre cierto elemento de la página.

A continuación se muestra un ejemplo de uso de código JavaScript:

```
<html>
<head>
<script>
    function miFuncion() {
        document.getElementById("demo").i
        nnerHTML = "Párrafo Cambiado";
    }
</script>
</head>
<body>
<h1>Mi página web</h1>
<p id="demo">Un párrafo</p>
<button type="button"
    onclick="miFuncion()">Cambiar</button>
</body>
</html>
```

Figura 5. Ejemplo de código JavaScript.

El ejemplo inicialmente muestra un párrafo cuyo texto es “Un párrafo” y un botón. Al hacer clic sobre el botón, se dispara el evento “onclick” el cual llamará a la función de JavaScript llamada “miFuncion”. Esta función busca al elemento cuyo identificador sea igual a “demo”, en este caso el párrafo, y modifica su HTML interno a “Párrafo Cambiado”, resultando en una modificación dinámica del contenido de la página web sin la necesidad de establecer comunicación con el servidor web.

d. AngularJS

AngularJS es un framework desarrollado por Google. Está escrito en JavaScript, y basado en una versión compacta de la biblioteca para JavaScript jQuery llamada jqLite. La intención de AngularJS es proveer un marco de trabajo que permita implementar de manera sencilla páginas web bien diseñadas y estructuradas, utilizando el patrón MVC (Modelo Vista Controlador) (Dayley, 2014).

AngularJS provee funcionalidades que permiten manejar la entrada de un usuario a través del navegador web, manipular los datos del lado del cliente y manejar cómo los elementos son desplegados en pantalla.

A continuación se presenta un ejemplo de uso básico de AngularJS:

```
<html ng-app>
<head>
  <title>Título</title>
  <script
src="js/angular.min.js"></script>
</head>
<body>
  <p>Hola {{'Mun' + 'do'}}</p>
</body>
</html>
```

Figura 6. Importando AngularJS.

Dentro de la etiqueta de apertura del código de HTML se encuentra lo que se conoce como una directiva de AngularJS. La directiva ng-app dentro de la etiqueta <html> indica al framework que todo el documento se encuentra bajo el control de AngularJS.

También se puede apreciar que se incluye una referencia a la biblioteca de JavaScript de AngularJS mediante la etiqueta <script>. Sin esto, no sería posible la ejecución de ninguna funcionalidad de AngularJS.

Finalmente, se observa entre las cadenas “{{” y “}}” lo que se denomina una expresión de AngularJS. En este caso en particular el resultado de esta expresión es la concatenación de la cadena “Mun” con la cadena “do”. Así, al ser interpretado el documento HTML por un navegador web, este desplegará la frase “Hola Mundo”.

Otra directiva importante es ng-model. El siguiente ejemplo muestra su uso:

```
<html ng-app>
<head>
  <title>Título</title>
  <script
src="js/angular.min.js"></script>
</head>
<body>
  <label>Ciudad: </label>
  <input ng-model="ciudad"
type="text"/>
  <p>Usted ingresó: {{ciudad}}</p>
</body>
</html>
```

Figura 7. Trabajando con AngularJS

En este caso, la directiva ng-model es un atributo dentro de un campo de texto de entrada al que se le asigna el valor ciudad. De manera seguida se muestra un párrafo que contiene la expresión {{ciudad}}. El efecto de este código es que en cuanto el usuario empieza a llenar el campo de texto el valor del párrafo es actualizado de manera automática y en tiempo real.

2.3.2 Herramientas para el desarrollo (lado del servidor)

En una aplicación web, el software del lado del servidor es aquel con el que se comunican las aplicaciones cliente. Fundamentalmente, este software deberá procesar una trama HTTP, ejecutar una función de negocio como consecuencia, y por último, enviar una respuesta HTTP al cliente (W3C, 2004). Cuando el cliente se trata de un navegador web y de haber un procesamiento exitoso, esta respuesta contendrá una combinación de código HTML, CSS y JavaScript que el navegador podrá interpretar y mostrar al usuario final.

En el presente T.E.G., en conformidad con lo indicado por la arquitectura propuesta en la sección 1.5, la aplicación del lado del servidor está

construida sobre el entorno de ejecución Node.js. Adicionalmente, esta arquitectura presenta una capa entre el entorno Node.js y los clientes web y móvil; esta capa cumple con la función de servidor proxy inverso, funcionalidad que cubre la herramienta Nginx. Por lo tanto, en el presente apartado se estudiarán ambas tecnologías.

a. Nginx

Nginx es un servidor HTTP y a la vez un servidor de proxy inverso (Nginx, 2015). Un servidor de proxy inverso se encarga de redirigir las peticiones HTTP que recibe a uno o más servidores de acuerdo a un criterio determinado, alguna de las motivaciones detrás de esta elección de arquitectura son:

- Agrega una capa adicional de seguridad.
- Balanceo de carga. La información o la lógica de negocio puede estar físicamente distribuida en varios nodos de cómputo pero son expuestos al usuario final de manera centralizada.
- Permite generar caché de contenido estático. Lo que acelera las respuestas de las peticiones HTTP.

A continuación se listan algunas de las características adicionales de Nginx:

- Servidor de archivos estáticos, índices y generación automática de índices.
- Proxy inverso acelerado con caché.
- Balanceo de carga y tolerancia a fallos.
- Soporte de HTTP sobre SSL.
- Servidores virtuales basados en nombre y/o en dirección IP.
- Soporte para protocolo SPDY, en complemento a HTTP como protocolo en la capa de aplicación. Este protocolo tiene la capacidad de acelerar hasta en un 64% la comunicación entre un servidor y un cliente.

- Habilitado para soportar más de 10.000 conexiones simultáneas.

b. Node.js

Es una plataforma de desarrollo construida sobre el motor de ejecución de JavaScript v8. Este es un motor de código abierto escrito en lenguaje C++ empleado también en el navegador web Google Chrome. Node.js está orientado al desarrollo de aplicaciones de red (e.g. servidores web) de manera sencilla y escalable (Joyent, 2015).

Además, Node.js está basado en una arquitectura orientada a eventos. En este tipo de arquitectura la actividad primaria es la reacción a recibir una señal semánticamente significativa, por ejemplo un evento. Las señales pueden provenir de cualquier fuente, como sensores, actividad humana, temporizadores, observación de un estado compartido o señales emitidas por algún otro sistema (Cunningham, 2015).

La principal característica de Node.js es su modelo de entrada/salida no bloqueante. Los servidores de red tradicionales requieren un nuevo hilo de ejecución en el CPU por cada conexión por parte de un cliente, lo que eventualmente consume los recursos del sistema, Node.js utiliza un único hilo para todas las peticiones por lo que no bloquea la entrada/salida del CPU. Esto permite a Node.js tener decenas de miles de conexiones concurrentes y es una de las razones por las cuales es una elección popular para el desarrollo de aplicaciones web de alto tráfico (Krol, 2014).

A continuación se muestra un ejemplo de cómo construir una aplicación que inicia un servidor HTTP y emite una respuesta básica a un navegador web:


```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hola Mundo\n');
}).listen(8080, 'localhost');
console.log('Servidor ejecutándose en
http://localhost:8080');
```

Figura 8. Código JavaScript para crear un servidor en NodeJS

Al ejecutar este ejemplo en el entorno de ejecución de Node.js y posteriormente consultar a la dirección `http://localhost:8080` en un navegador, este desplegaría la frase “Hola Mundo” como respuesta.

Otra característica notable de Node.js es su facilidad para comportarse de manera asíncrona.

Node.js en sí mismo es de muy bajo nivel. Consiste de un pequeño conjunto de módulos que realizan tareas muy específicas, como por ejemplo, herramientas para operar con el sistema de archivos, funciones de red sobre TCP y HTTP, seguridad y *streaming*.

2.3.3 Desarrollo de servicios Web

En muchas ocasiones es ventajoso exponer recursos o funciones de negocio en la forma de un servicio web, siendo la razón principal la facilidad de integración con cualquier otro sistema cliente al minimizar las dependencias tecnológicas entre las partes (bajo acoplamiento), idealmente siendo el conocimiento sobre el funcionamiento del servicio la única restricción para efectuar la integración (Erl, Carlyle, Pautasso, & Balasubramanian, 2012). Debido a esta ventaja, en el presente T.E.G. las funcionalidades centrales de la solución propuesta están expuestas como servicios web REST.

a. REST

La Transferencia de Estado Representacional (*Representational State Transfer*) o REST es un estilo de arquitectura de software que permite usar a la World Wide Web como plataforma para interconectar servicios a una escala global. El término se originó en el año 2000, en una tesis doctoral sobre la Web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP (Hypertext Transfer Protocol) (Fielding, 2000).

A grandes rasgos, REST permite disponibilizar un recurso o un servicio a través de la Web utilizando un identificador uniforme de recursos URI (*Uniform Resource Identifier*) y el protocolo de transporte de hipertexto HTTP para su interacción con el mismo.

Un URI tiene la forma *<esquema>:<estructura-propia-esquema>*. El esquema define cómo el resto del URI será interpretado. Por ejemplo, la parte *http* del URI *http://ejemplo.org/archivos/texto.txt* nos dice que el URI deberá ser interpretado de acuerdo al esquema HTTP. Bajo este esquema, el URI referencia a una máquina cuyo nombre de host, identificado por el sistema DNS (Domain Name System), es *ejemplo.org*. Finalmente, es responsabilidad de la máquina “escuchando” en la dirección *ejemplo.org* el asociar el remanente del URI, */archivos/texto.txt*, al recurso en concreto. Cualquier agente de software que comprenda el esquema HTTP podrá interactuar con este recurso si sigue las reglas establecidas por la especificación del protocolo (RFC 2616) (Webber, Parastatidis, & Robinson, 2010).

REST no presenta ninguna restricción respecto al formato del recurso que disponibiliza, este viene dado por la necesidad que atiende este recurso. Por ejemplo, de tratarse de una página web un formato adecuado suele ser un documento HTML y en el caso de un servicio web se prefieren formatos

estructurados como XML o JSON. En ambos casos, el formato del recurso viene declarado en la cabecera de la trama HTTP que se envía como respuesta a la invocación del recurso.

La manera de interactuar con un recurso o servicio está definida por los métodos de HTTP —GET, POST, PUT, DELETE, OPTIONS, HEAD, TRACE, CONNECT, y PATCH—. Adicionalmente, HTTP define una colección de códigos de respuesta, tales como 200 OK, 201 Creado y 404 No Encontrado, los que coordinan las interacciones instigadas por el uso de los métodos. Así, en conjunto, los métodos y los códigos de respuesta proveen de un framework general para operar con recursos en la red (Fielding, 2000).

2.4 Base de datos no relacionales (NoSQL)

La arquitectura de la solución propuesta en este T.E.G. requiere de un repositorio central de datos en donde los diferentes procesos puedan almacenar, actualizar, consultar y eliminar datos de manera coherente.

La alternativa tecnológica seleccionada para servir como repositorio de datos a la solución fue la de una base de datos NoSQL, específicamente el producto MongoDB, una base de datos orientada a documentos.

Con la finalidad de justificar esta decisión tecnológica, en el presente apartado se estudiarán las bases de datos, particularmente las conceptualizadas como NoSQL. Finalmente se analizarán las características del sistema de base de datos MongoDB.

“Una base de datos es una colección de datos interrelacionados que son manejados como una sola unidad” (Oppel, 2009). Esto implica que además de un repositorio físico, una base de datos está constituida por una organización lógica de datos donde se definen relaciones entre los mismos.

Tradicionalmente las bases de datos están esquematizadas bajo el modelo relacional propuesto por Edgar Codd en 1969, en el que los datos están

organizados a manera de tuplas (filas) y agrupados de acuerdo a sus campos o atributos (columnas) en grupos relacionados (tablas), pudiendo también definirse relaciones entre registros en diferentes tablas a través de atributos referenciales o claves foráneas (Codd, 1969).

En la última década ha tomado fuerza un paradigma alternativo a las bases de datos relacionales, este paradigma se conoce como NoSQL.

NoSQL (*Not only SQL*, No solamente SQL) es un término que se utiliza para denotar a aquellos sistemas de bases de datos que proveen mecanismos para el almacenamiento y consulta de datos que difieren de los tabulares empleados en los sistemas relacionales. Entre las estructuras de datos más populares utilizadas por estos sistemas están los grafos, los pares atributo-valor y los documentos (Vaish, 2013). El motivo principal detrás de la adopción de estos mecanismos en detrimento de los sistemas relacionales, es su capacidad de escalabilidad horizontal, concepto que se desarrollará en el siguiente punto.

2.4.1 Escalabilidad horizontal

Es común que los requerimientos de hardware de una aplicación, específicamente los relativos a las capacidades de sus bases de datos, sean complejos de predecir. Una aplicación puede incrementar o disminuir dramáticamente y de manera volátil su cantidad de usuarios y proporcionalmente su necesidad de recursos (Dayley, 2014).

De acuerdo a Dayley (2014), existen dos maneras fundamentales en las que se pueden adaptar (escalar) los sistemas de bases de datos a sus cambiantes necesidades de hardware: verticalmente u horizontalmente. Cuando se habla de escalamiento vertical, se refiere a incrementar la cantidad de recursos de un nodo de procesamiento/almacenamiento, por ejemplo, incrementar la capacidad o cantidad de CPU's, memoria y discos

duros. Por otro lado, el escalamiento horizontal implica agregar nodos adicionales a una red de procesamiento distribuido.

A pesar de la evidente complejidad en el manejo del escalamiento horizontal, este ofrece una mayor capacidad de dinamismo y mayores cantidades de recursos disponibles a un sistema que las que un escalamiento totalmente vertical podría ofrecer (Bondi, 2000).

Las implementaciones tradicionales de los sistemas de bases de datos relacionales, por limitaciones del mismo modelo relacional y las restricciones que el mismo requiere, presentan dificultades para poder escalar horizontalmente y al mismo tiempo ofrecer un rendimiento adecuado. Por otro lado, los sistemas NoSQL fueron construidos con este tipo de escalamiento en mente y lo aprovechan mejor y más fácilmente (McCreary & Kelly, 2013).

2.4.2 MongoDB

Es el sistema de base de datos no relacional más utilizado a nivel mundial, y el cuarto entre los sistemas de bases de datos en general (db-engines.com/en/ranking).

MongoDB almacena los datos en una estructura basada en documentos JSON con un esquema dinámico. Algunas ventajas de almacenar los datos en esta manera son (www.mongodb.org/document-databases):

- Facilita escribir la lógica de la aplicación. No es necesario traducir un modelo de objetos a sentencias SQL. El modelo de objetos se traduce directamente a un documento.
- Los datos no estructurados pueden ser almacenados fácilmente, ya que un documento puede contener las claves y valores que la lógica de la aplicación requiera en el momento que lo precise. La base de datos no necesita conocimiento previo de la estructura del documento.

Entre otras características específicas de MongoDB se puede destacar (docs.mongodb.org/manual/contents/):

- Consultas *Ad-hoc*. Es posible hacer búsquedas por campos, consultas por rangos de valores y expresiones regulares. Las consultas pueden retornar campos específicos de documentos y también incluir funciones JavaScript definidas por el usuario.
- Índices. Todo campo en un documento puede ser indexado de manera análoga a los campos de un sistema manejador de base de datos relacional.
- Alta disponibilidad. Es posible configurar MongoDB para que almacene conjuntos replicados de datos que se mantienen actualizados automáticamente. En caso de fallar el acceso a la copia original se dispone de una de las réplicas de manera totalmente transparente.
- Balanceo de carga. La escalabilidad horizontal es provista a través de un mecanismo llamado *sharding* (fragmentación). El usuario elige una clave de fragmentación, la cual determina cómo serán distribuidos los datos en los diferentes nodos del sistema distribuido.
- Almacenamiento de archivos. MongoDB puede ser utilizado como un sistema de archivos. Pudiendo hacer uso de las ventajas de balanceo de carga y replicación de datos para poder almacenar archivos en un ambiente distribuido.

2.5 Desarrollo de aplicaciones móviles

La solución propuesta en el presente T.E.G. incluye a una aplicación móvil para el consumo de sondeos ejecutable en un sistema operativo Android. Como consecuencia, este apartado se enfocará en las características y el proceso de desarrollo de aplicaciones sobre dicha plataforma.

2.5.1 Android

Android es un sistema operativo basado en el kernel de Linux, diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y también para relojes inteligentes, televisores y automóviles, inicialmente desarrollado por Android Inc.

Según Topolski (2007) las características y especificaciones actuales de este sistema operativo son:

- Soporte de las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC, WiMAX, GPRS, UMTS y HSDPA+.
- Almacenamiento local en motor de base de datos SQLite.
- Interfaz gráfica adaptable a diferentes resoluciones de pantallas. Compatibilidad con bibliotecas de gráficos 2D y bibliotecas de gráficos 3D basadas en las especificaciones de la OpenGL ES 2.0.
- Manejo nativo de mensajería SMS y MMS. Además, permite el uso de *Android Cloud to Device Messaging Framework* para funcionalidades extendidas de mensajería.
- Incluye un navegador web basado en el motor de renderizado de código abierto WebKit, en conjunto con el motor de ejecución de JavaScript v8.
- Desarrollo de aplicaciones en lenguaje Java. Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.

- Soporte multimedia Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
- Soporte para streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
- Soporte para hardware adicional. Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
- Mercado de aplicaciones Google Play. Google Play es un catálogo de aplicaciones gratuitas o pagas en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
- Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil.
- Reconocimiento de voz. La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
- *Tethering*. Android soporta *tethering*, tecnología que permite al dispositivo ser usado como un punto de acceso a la red de manera alámbrica o inalámbrica.

Android proporciona un conjunto de herramientas de desarrollo o SDK (*Software Development Kit*) para la construcción de sus aplicaciones. Con este, además de desarrollar aplicaciones, se puede ejecutar un emulador del sistema Android de la versión que sea necesaria. Todas las aplicaciones

Android se desarrollan en lenguaje Java utilizando esta familia de herramientas.

La plataforma Android está compuesta de los siguientes componentes:

Aplicaciones: Tales como el directorio de contactos, la aplicación de comunicación telefónica, los mapas, el navegador web, entre otras.

Interfaz de Programación de Aplicaciones (API): Es un conjunto de herramientas y subrutinas necesarias para el desarrollo de aplicaciones Android. Algunas de estas son: *Activity Manager* (Manejador de Actividades), *Fragment Manager* (Manejador de Fragmentos), *Content Provider* (Proveedor de Contenido) y Manejador de Notificaciones (*Notification Manager*).

Bibliotecas: Existe un conjunto de bibliotecas escritas en C/C++ que están expuestas al desarrollador a través de la API. Por ejemplo: El manejador de SQLite, OpenGL para generación de gráficas, SSL para comunicación a través de un canal seguro, etc.

Entorno de ejecución (Android Runtime): Consiste de un conjunto de bibliotecas que implementan la mayoría de las bibliotecas básicas del lenguaje de programación Java.

Kernel: Es el estrato más profundo de la plataforma e implementa operaciones relacionadas con el manejo del hardware del dispositivo y algunas otras operaciones especiales. Por ejemplo, controladores de cámara, de pantalla táctil, manejo de memoria y algunas rutinas de encriptación.

La estructura de una aplicación Android está definida por un conjunto de interfaces gráficas, también llamadas Actividades (*Activities*), que a su vez pueden estar compuestas por Fragmentos (*Fragments*). Las Actividades o Fragmentos poseen un flujo de vida controlado por una entidad denominada

Intent capaz de levantar estos elementos gráficos y mostrarlos en pantalla de acuerdo a la lógica del negocio.

2.6 Código QR

El código QR (*quick response code*) es una representación gráfica de información ideada por la empresa japonesa Denso Wave Inc. Un código QR es una matriz de puntos bidimensional con capacidad de almacenar hasta 7089 caracteres (Denso Wave Incorporated, 2015).

Con el crecimiento de la popularidad de los teléfonos inteligentes, estos han adquirido la posibilidad de leer los códigos QR haciendo uso de la cámara fotográfica de los dispositivos. Además, en la mayoría de los casos, el software encargado de la lectura del código QR en el dispositivo, también es capaz de consumir la información contenida en el mismo y ejercer una acción en respuesta. El ejemplo más común es la carga de una página web al capturar un código QR cuya información subyacente era la URL del mencionado sitio.

El sistema operativo Android posee las funcionalidades básicas necesarias para permitir la captura y procesamiento de códigos QR, por lo tanto esta tecnología es utilizada en el presente T.E.G. como agente disparador del consumo de sondeos. En el Capítulo IV se explica más detalladamente el aporte de los códigos QR a la solución propuesta.

CAPITULO III

MARCO METODOLOGICO

Planificar y ejecutar un proyecto que beneficie a una empresa u organización requiere de una metodología que dote de los mecanismos adecuados para que el sistema satisfaga las necesidades tanto de los usuarios como de los clientes que contratan dicho desarrollo. De nada sirven buenas notaciones y herramientas si no se proveen directivas para su aplicación.

Así, en estos últimos años se ha comenzado con un creciente interés en metodologías de desarrollo, que permitan obtener aplicaciones, mediante un proceso de tratamiento en capas, suficientemente usables y fáciles de mantener. Desde el punto de vista de la ingeniería del software no basta el conocimiento de lenguajes y técnicas de programación, de entornos de desarrollo o de editores de recurso, es imprescindible conocer la manera en la que se debe realizar un software.

La experiencia ha comprobado que un enfoque metódico para el desarrollo de software arroja como resultados menos defectos, reduciendo los tiempos de entrega y agregando valor.

Una metodología plantea un proceso disciplinado con el objetivo de hacer el proceso de desarrollo de software más predecible y eficiente. Alrededor de cómo hacer software hay un gran número de teorías, propuestas y disciplinas de desarrollo, sin embargo, el presente Capítulo se enfocará en el estudio de SCRUM, metodología empleada el presente Trabajo Especial de Grado.

3.1 Metodología AGILE

Según Rea & Parker (2014) Es un grupo de métodos de desarrollo de software en las que los requerimientos y soluciones evolucionan a través de la colaboración entre grupos multi-disciplinarios y auto-organizados. AGILE (proveniente de ágil) promueve el planteamiento adaptativo, el desarrollo

evolutivo, entregas rápidas, mejoras continuas y promueve una rápida y flexible respuesta a los cambios.

Existen distintas implementaciones de la metodología AGILE, siendo una de las más populares la metodología SCRUM, objeto de estudio en el siguiente apartado.

3.1.1 SCRUM

SCRUM es un framework para el desarrollo de productos basado en la metodología AGILE que tiene como característica fundamental el avance iterativo e incremental del producto (Schwaber & Beedle, 2002).

Se define como una estrategia flexible y holística de desarrollo de un producto, donde un equipo de desarrollo trabaja como una unidad para conseguir un objetivo común; tiene una marcada diferencia con el esquema tradicional secuencial de desarrollo de productos, y permite a los equipos a auto-organizarse, al promover la colaboración del equipo, ya sea física o de forma online, así como la comunicación diaria “cara a cara” entre los distintos miembros y disciplinas del proyecto.

Uno de los principales principios de SCRUM es reconocer que durante el desarrollo del producto, el cliente puede cambiar sus decisiones sobre lo que necesitan o quieren, algo que puede ser costoso en la metodología tradicional que es más predictiva y planeada. De esta forma, SCRUM adopta un enfoque empírico, aceptando que el problema no puede ser completamente entendido o definido, enfocándose en maximizar la habilidad del equipo en hacer entregas rápidas y responder a requerimientos emergentes.

SCRUM es un marco de trabajo en donde las personas pueden definir problemas adaptativos complejos, mientras se entrega, de forma productiva y creativa, producto de la mayor calidad posible. (Scrumguide.com)

SCRUM es:

- Ligerio
- Simple de entender
- Difícil de dominar.

Su figura principal es la de equipo, que consta de diferentes roles, eventos, artefactos y reglas. Cada componente en el marco de trabajo sirve un propósito específico y es esencial para el éxito y uso del SCRUM.

Las reglas del SCRUM unen los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos.

SCRUM está fundado bajo el empirismo, aseverando que el conocimiento proviene de la experiencia y que las decisiones se deben tomar basándose en lo que se sabe. Aplica un acercamiento iterativo e incremental para optimizar la predicción y el control de riesgo.

La implementación de control de proceso empírico se basa en tres pilares: transparencia, inspección y adaptación.

Transparencia: Aspectos importantes del proceso deben ser visibles a aquellos responsables del objetivo esperado. La transparencia requiere que esos aspectos sean definidos en un estándar común para que los distintos observadores compartan un entendimiento en lo que se está viendo.

Por ejemplo:

Un lenguaje común debe utilizarse al referirse al proceso.

Aquellos realizando el trabajo y aquellos que lo aceptan deben tener un mismo concepto de completo.

Inspección: Los usuarios de SCRUM deben inspeccionar frecuentemente los artefactos y el progreso hacia la meta de cada Sprint para detectar varianzas no deseables. La inspección, sin embargo, no debe ser demasiado frecuente para que no afecte el flujo de trabajo.

Adaptación: Si en la inspección se determina que uno o más aspectos de un proceso se desvían fuera de los límites aceptables, y el producto resultante no será adecuado, el proceso o el material procesado debe ser ajustado. Un ajuste debe ser ejecutado lo más rápido posible para minimizar la desviación.

a. Roles de SCRUM

El Equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. Los Equipos Scrum son auto-organizados y multifuncionales. Los equipos auto-organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad. Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto terminado aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

Product Owner: El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (*Product Backlog*).

Development Team: El Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo de entregar un Incremento de producto “Terminado”, que potencialmente se pueda poner en producción, al final de

cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

Scrum Master: El Scrum Master es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.

b. Eventos del Scrum

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum.

Sprint: El corazón de Scrum es el Sprint. Es un bloque de tiempo (*time-box*) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Cada Sprint puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, los Sprints se usan para lograr algo. Cada Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción, el trabajo y el producto resultante.

Sprint Planning meeting: El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación de Sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo.

Daily Scrum: El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente.

Sprint Review: Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran

acerca de lo que se hizo durante el Sprint. Basándose en esto, y en cualquier cambio a la Lista de Producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.

Sprint Retrospective: La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

Sprint Goal: El Objetivo del Sprint es una meta establecida para el Sprint que puede ser alcanzada mediante la implementación de la Lista de Producto. Proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento. Es creado durante la reunión de Planificación del Sprint.

c. Artefactos del Scrum

Los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

Product Backlog: La Lista de Productos es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Dueño de Producto (*Product Owner*) es el responsable de la Lista de Producto, incluyendo su contenido, disponibilidad y ordenación.

La Lista de Productos enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la Lista de Producto tienen como atributos la descripción, la ordenación, la estimación y el valor.

Sprint Backlog: La Lista de Pendientes del Sprint es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Lista de Pendientes del Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento "Terminado".

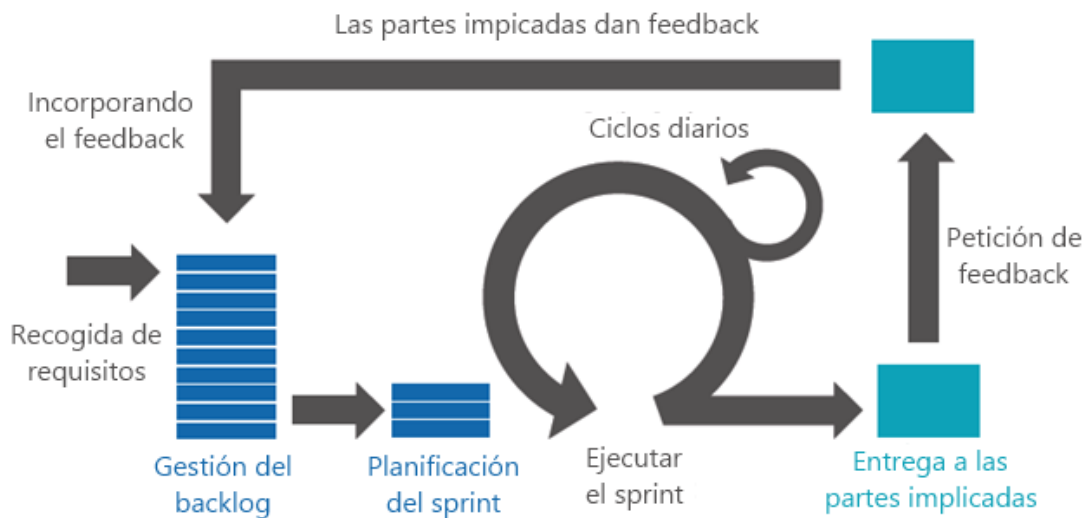


Figura 9. Proceso de SCRUM

CAPÍTULO IV

MARCO APLICATIVO

En este capítulo se describen los pasos y métodos seguidos para el desarrollo de la solución propuesta, que como se ha descrito anteriormente, es un sistema para la automatización en el proceso de creación, despliegue y gestión de sondeos.

Se explica en este capítulo el método de desarrollo de software, y las fases que fueron necesarias para la realización del mismo.

4.1 Modelo de desarrollo de software

Para la realización del sistema, se utiliza el método de desarrollo SCRUM, el cual tiene como finalidad minimizar el proceso de planificación del proyecto para favorecer y agilizar el desarrollo de los módulos necesarios bajo varios ciclos iterativos o *Sprints*, ajustándose SCRUM a los cambios del negocio.

Para efectos de este T.E.G se definieron los roles de Scrum de la siguiente forma:

Scrum Master: Franklin Sandoval

Development Team: Ignacio Córdoba, Lino Hernández

Product Owner: Isaac Casado (En representación de la empresa Taima)

Como la metodología sugiere, se hizo el levantamiento del Product Backlog, el cual recopila los requerimientos, funcionales y no funcionales, generales del sistema.

4.2 Análisis y diseño general

4.2.1 Requerimientos funcionales (casos de uso)

A continuación se definen los requerimientos funcionales que concretarán el alcance de la solución:

- Deben existir dos perfiles de usuario: Administrador y Opinador.
- El administrador debe poder:
 - Validar sus credenciales al iniciar sesión
 - Crear sondeos
 - Definir segmentación del sondeo
 - Publicar sondeo
 - Consultar resultados
 - Exportar resultados en formato universal
- El opinador debe poder:
 - Registrarse en el sistema
 - Validar sus credenciales al iniciar sesión
 - Consultar sondeos disponibles
 - Responder sondeos
 - Consultar y editar su perfil

A continuación se definen los casos de uso separados por los dos perfiles de usuario requeridos.

Casos de uso del módulo administrativo (web)

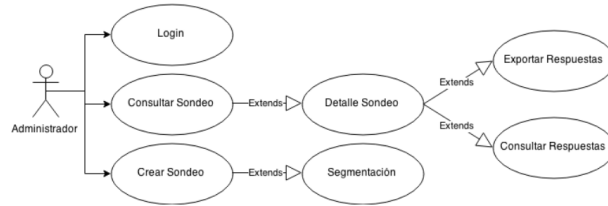


Figura 10. Casos de uso módulo administrativo

A continuación se encuentra la especificación de los casos de uso del módulo administrativo.

Tabla 1. Login de usuario

Nombre	Login de Usuario
Descripción	Funcionalidad que permite la entrada al sistema, para el módulo de administrador
Actores	Administrador
Pre Condición	El Usuario debe de estar cargado en la base de datos del sistema
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación web de Taima 2. El sistema muestra la pantalla de “Inicio de Sesión” y solicita los siguientes campos: Correo Electrónico y Contraseña. 3. El usuario introduce los datos correspondientes y hace clic en el botón “Iniciar sesión”. 4. El sistema valida los datos ingresados y muestra la pantalla principal administrativa.
Flujo Alternativo	<ol style="list-style-type: none"> 1. 3.1 Si hay un error en los datos se le notifica que los datos son erróneos y que hay que corregirlos.
Post condición	Ingreso al sistema

Tabla 2. Consultar sondeos.

Nombre	Consultar sondeos.
Descripción	Funcionalidad que permite consultar los sondeos creados por el administrador.
Actores	Administrador
Pre Condición	El usuario administrador debe haber iniciado sesión en el sistema administrativo.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario ingresa a la web administrativa. 2. El sistema muestra la pantalla principal administrativa donde se encuentra los sondeos creados.
Flujo Alternativo	
Post condición	<p>Se muestran los sondeos existentes</p> <p>Muestra las opciones de Ver Resultados y la opción de Exportar Resultados</p>

Tabla 3. Crear sondeos.

Nombre	Crear Sondeo
Descripción	El usuario con privilegio de Administrador podrá crear Sondeos, editarlos y publicarlos.
Actores	Administrador
Pre Condición	El usuario administrador debe haber iniciado sesión en el sistema administrativo.
Flujo Normal	<ol style="list-style-type: none"> 1. El Administrador hace clic en la opción “Nuevo Sondeo”. 2. El sistema presenta el formulario para crear sondeo. 3. El Administrador completa la información solicitada y hace clic en el botón “Guardar”. 4. El sistema valida que los datos estén correctos. 5. El sistema guarda los datos y notifica al usuario. 6. El sistema habilita la opción de agregar preguntas. 7. El Administrador hace clic en el botón de agregar preguntas. 8. El sistema muestra el formulario para crear la pregunta nueva
Flujo Alternativo	<ol style="list-style-type: none"> 1. 4.1 Si hay un error en los datos se le notifica que los datos son erróneos y que hay que corregirlos.
Post condición	Una vez registrada la totalidad de las preguntas en el sondeo, el sistema permitirá la publicación del sondeo. No se pueden hacer modificaciones, actualizaciones o eliminar los datos del sondeo después de su publicación.

Tabla 4. Segmentación de sondeo.

Nombre	Segmentación de sondeo
Descripción	El usuario Administrador podrá definir una segmentación para especificar los Opinadores que recibirán el sondeo.
Actores	Administrador
Pre Condición	El usuario administrador debe haber iniciado sesión en el sistema administrativo. Debe existir sondeo creado para aplicarle una segmentación
Flujo Normal	<ol style="list-style-type: none"> 1. Una vez creado el sondeo el Usuario administrador decide aplicar segmentación al público objetivo del sondeo. 2. El sistema muestra un formulario separado por cada elemento medible del perfil de usuario: edad, País, estado, género. 3. El Administrador, elige los valores para los que la segmentación ocurrirá y guarda. 4. El sistema actualiza en tiempo real el tamaño de la muestra de usuarios a que aplicará la segmentación. Y guarda la segmentación definida. 5. Una vez publicado el sondeo, este estará disponible solo para los usuarios opinadores que cumplan con las características.
Flujo Alternativo	
Post condición	

Tabla 5. Resultados de sondeo.

Nombre	Resultados de sondeo
Descripción	Funcionalidad que permite al Administrador consultar los resultados de un sondeo.
Actores	Administrador
Pre Condición	El usuario administrador debe haber iniciado sesión en el sistema administrativo. Debe existir sondeo creado El administrador debe haber seleccionado un sondeo
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra la información principal del sondeo, así como el botón de consultar resultados. 2. El Administrador hace click en el botón de consultar resultados. 3. El sistema actualiza la vista con la información de los resultados.
Flujo Alternativo	
Post condición	Se muestran los resultados del sondeo en la pantalla.

Tabla 6. Exportar resultados.

Nombre	Exportar resultados
Descripción	Funcionalidad que permite al Administrador exportar los resultados de un sondeo.
Actores	Administrador
Pre Condición	El usuario administrador debe haber iniciado sesión en el sistema administrativo. Debe existir sondeo creado El administrador debe haber seleccionado un sondeo
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema muestra la información principal del sondeo, así como el botón de exportar resultados. 2. El Administrador hace click en el botón de exportar resultados. 3. El sistema muestra una previsualización de los datos a exportar en forma de tabla. 4. El administrador selecciona la opción de exportar. 5. El sistema genera el archivo e inicia la descarga del mismo
Flujo Alternativo	
Post condición	El sistema genera un archivo con los resultados del sondeo.

Casos de uso del módulo opinador (móvil)

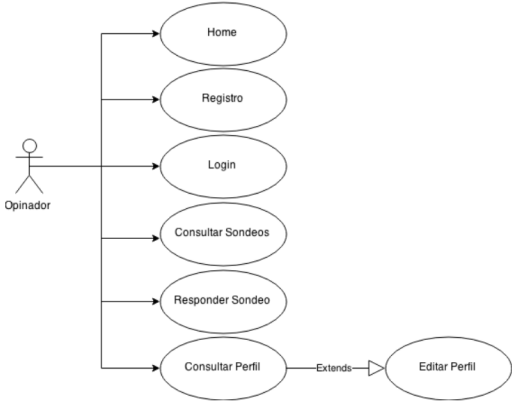


Figura 11. Casos de uso módulo opinador.

A continuación se encuentra la especificación de los casos de uso del módulo opinador.

Tabla 7. Registro de opinador.

Nombre	Registro de Opinador
Descripción	Lograr que el Opinador cree su cuenta, de ingreso al sistema para que pueda responder los sondeos.
Actores	Opinador
Pre Condición	El usuario no debe tener su correo electrónico registrado en el sistema.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario inicia desde su dispositivo móvil la aplicación de Taima previamente descargada. 2. El sistema muestra la pantalla de Home con los botones “Regístrate” e “Iniciar sesión”. 3. El usuario hace click en la opción “Regístrate”. 4. El sistema muestra el formulario para el registro de Opinador. 5. Una vez que ingresa los datos requeridos en el formulario, procede hacer clic en el botón “Regístrate”. 6. EL sistema valida los datos introducidos en el formulario. 7. El sistema guarda los datos de registro en la base de datos.
Flujo Alternativo	<ol style="list-style-type: none"> 1. 6.1 Si hay un error en los datos se le notifica que los datos son erróneos y que hay que corregirlos.
Post condición	Se guardan los datos de registro del opinador.

Tabla 8. Inicio de sesión de opinador.

Nombre	Inicio de sesión de Opinador
Descripción	Funcionalidad que permite la entrada al sistema, para el módulo de Opinador, donde el usuario según su perfil podrá disfrutar de las funcionalidades existentes
Actores	Opinador
Pre Condición	El usuario debe estar previamente registrado en el sistema.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación Android de Taima. 2. El sistema muestra la pantalla de “Inicio de Sesión” y solicita los siguientes campos: Correo Electrónico y Contraseña. 3. El usuario introduce los datos correspondientes y hace clic en el botón “Iniciar sesión”. 4. El sistema valida los datos ingresados. 5. El sistema muestra la pantalla principal administrativa.
Flujo Alternativo	<ol style="list-style-type: none"> 1. 5.1 Si hay un error en los datos se le notifica que los datos son erróneos y que hay que corregirlos.
Post condición	Se muestra la pantalla principal Dashboard para el Opinador.

Tabla 9. Consultar perfil.

Nombre	Consultar Perfil
Descripción	Funcionalidad que permite al opinador consultar la información de su perfil.
Actores	Opinador
Pre Condición	El usuario debe haber iniciado sesión en la aplicación móvil de Taima.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla de mis sondeos. 2. El usuario despliega el menú de navegación. 3. El sistema muestra la pantalla de perfil de usuario.
Flujo Alternativo	
Post condición	Se muestra la pantalla del perfil de usuario.

Tabla 10. Editar perfil.

Nombre	Editar Perfil
Descripción	Funcionalidad que permite al opinador editar la información de su perfil.
Actores	Opinador
Pre Condición	El usuario debe haber iniciado sesión en la aplicación móvil de Taima. El usuario debe encontrarse en la pantalla de perfil.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario elige el elemento del menú que desea editar. 2. El sistema muestra un campo de edición para que el usuario actualice, luego selecciona guardar. 3. El sistema valida el dato introducido. 4. El sistema guarda el cambio realizado.
Flujo Alternativo	<ol style="list-style-type: none"> 1. 3.1 Si los datos no son correctos, el sistema notifica que se debe arreglar.
Post condición	Se guardan los nuevos datos en la base de datos.

Tabla 11. Consultar sondeos (opinador)

Nombre	Consultar Sondeos (Opinador)
Descripción	Funcionalidad que permite consultar los sondeos de un Usuario que se encuentra registrado en la aplicación.
Actores	Opinador
Pre Condición	El usuario debe haber iniciado sesión en la aplicación móvil de Taima. El opinador debe estar en la pantalla principal de la aplicación.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema muestra los sondeos disponibles para el opinador. 2. El opinador hace click en refrescar 3. El sistema actualiza la lista de sondeos disponibles
Flujo Alternativo	
Post condición	El sistema muestra los sondeos disponibles.

Tabla 12. Responder sondeo.

Nombre	Responder Sondeos
Descripción	Funcionalidad que permite al Opinador responder los Sondeos disponibles.
Actores	Opinador
Pre Condición	El usuario debe haber iniciado sesión en la aplicación móvil de Taima. El opinador debe estar en la pantalla principal.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema muestra los sondeos disponibles para el opinador. 2. El opinador selecciona el sondeo que desea responder. 3. El sistema despliega la pantalla de pregunta de sondeo. 4. El opinador responde la pregunta y hace click en avanzar 5. El sistema verifica si la pregunta es requerida 6. El sistema muestra mensaje de finalización de sondeo. 7. El opinador hace click en finalizar sondeo.
Flujo Alternativo	<ol style="list-style-type: none"> 1. 4.1 El opinador no responde la pregunta y hace click en avanzar. 5.1 El sistema verifica si la pregunta es requerida. 5.2 Si la pregunta es requerida, el sistema notifica al usuario que debe responder para avanzar.
Post condición	Se guardan las respuestas de las preguntas en la base de datos.

3.2.2 Requerimientos no funcionales

- Capacidad de ingreso a módulo de administración de sondeos desde cualquier parte.

El módulo de administración de sondeos es un módulo Web, por lo cual pueden estar disponibles desde cualquier parte del mundo siempre que el servidor esté publicado.

- Capacidad de lectura de sondeo promocional en locación física.

Para impulsar la promoción de la plataforma se solicita que exista una interacción física con la aplicación.

3.2.3 Diseño general de la Solución

La solución cuenta con dos módulos principales enfocados en cada perfil de usuario respectivamente. El primero es el módulo administrativo que contiene el acceso con mayores privilegios del sistema, y es el único lugar donde se puede crear el elemento fundamental, el sondeo. Este módulo representa el punto central de interacción del sistema, estando muy cerca lógicamente de la base de datos y el API que ofrece los servicios públicos y privados del sistema.

El segundo es el módulo móvil, y es el que corresponde a la interacción del Opinador con el sistema.

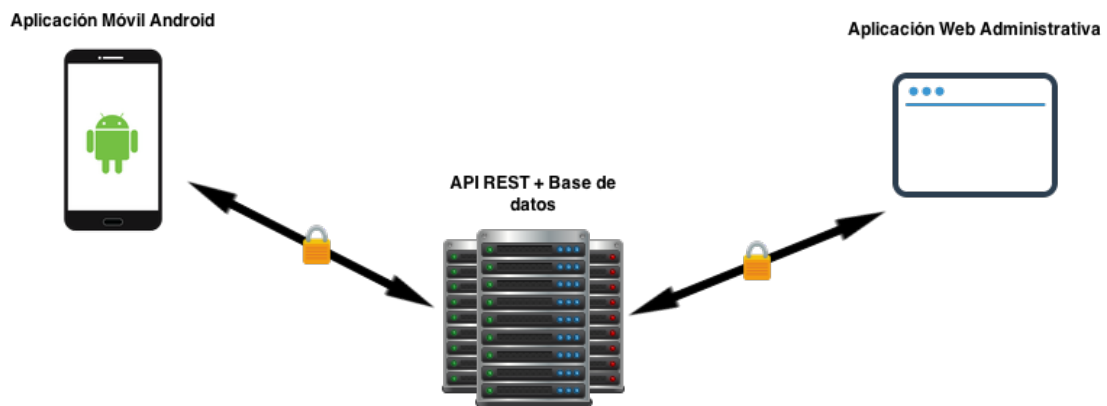


Figura 12. Arquitectura de la solución.

En la figura 12 se puede observar como los dos módulos interactúan directamente con el núcleo del sistema, que corresponde al API y a la base de datos.

3.3 Construcción de la solución

En este punto, se expone como se construye la solución de tal manera que satisfaga los requerimientos planteados. La construcción está dividida en cada uno de los requerimientos, y algunas funcionalidades esenciales para el funcionamiento de la solución.

Se define el desarrollo de las funcionalidades dentro de los Sprints que define la Metodología SCRUM. El artefacto principal de la metodología es el Product Backlog, definido en este T.E.G como la lista de casos de uso y especificaciones de los mismos, ya que estos representan los requerimientos de desarrollo del alcance de la solución. Adicionalmente se definieron Sprints de un mes de duración, necesitando así 5 para concretar las actividades que se derivan de estos requerimientos.

- Sprint 1
 - Configuración del sistema web.
 - Diseño y construcción de modelo de Base de datos.
 - Implementación Inicio de sesión (web)
- Sprint 2
 - Construcción del módulo administrativo.
 - Panel principal de sondeos
 - Creación de sondeos
- Sprint 3
 - Implementación de la segmentación
 - Implementación de gestión de resultados de sondeo.
 - Configuración del sistema móvil
- Sprint 4
 - Implementación de Inicio de sesión/Registro (móvil)
 - Pantalla principal de sondeos (móvil)
 - Interacción con sondeos (móvil)
- Sprint 5
 - Implementación de lector QR (móvil)
 - Implementación de perfil (móvil)
 - Pruebas de usuario final.

Configuración del sistema web:

Para construir una aplicación Web, es necesario contar con un servidor que procese la interacción entre las aplicaciones del lado del cliente y las aplicaciones del lado del servidor. En este caso se configuró en servidor Nginx.

Nginx permite el correcto redireccionamiento entre los diferentes puertos que el sistema estará utilizando. El puerto más utilizado en aplicaciones web es el

80, sin embargo en el caso de las tecnologías seleccionadas varios puertos serán utilizados como canales de comunicación.

Sobre este servidor se instala Node.js como plataforma de desarrollo. El desarrollo de la parte Web de la solución será realizado en el lenguaje JavaScript. No es necesario un IDE específico para trabajar con esta tecnología por lo que se usará VIM y Sublime text como herramientas de desarrollo.

Diseño y construcción de modelo de base de datos:

La base de datos fue diseñada bajo el esquema No-SQL específicamente con la tecnología Mongo-DB, ya que esta tecnología presenta un manejo rápido y sencillo de los datos.

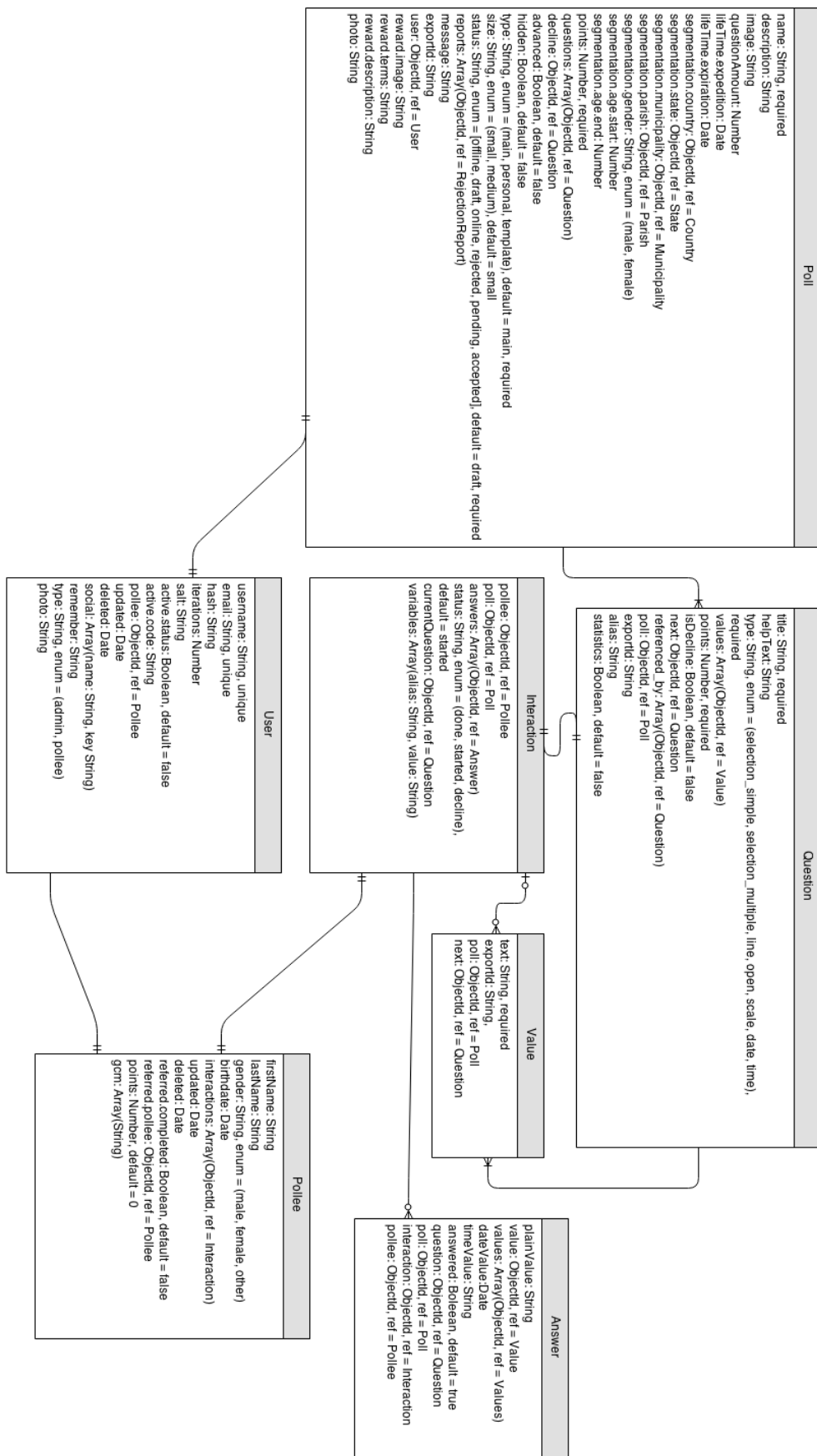


Figura 13. Modelo de la base de datos.

En la figura 13 se puede observar el esquema de la base de datos. Es importante destacar el objeto Sondeo como parte central del esquema y el objeto Interactions que representa el almacenamiento de las respuestas a los sondeos.

El proceso de construcción de una base de datos NoSQL con MongoDB se realiza implementando los objetos de la estructura de datos en formato JSON.

Implementación de inicio de sesión (Web):

Una vez configurada la base de datos se pasa a la creación del punto de seguridad principal del sistema, el cual es el ingreso o inicio de sesión. La pantalla de inicio de sesión fue creada con tecnologías web básicas como HTML y CSS, adicionalmente se trabajó con el framework AngularJS como plataforma de front-end.

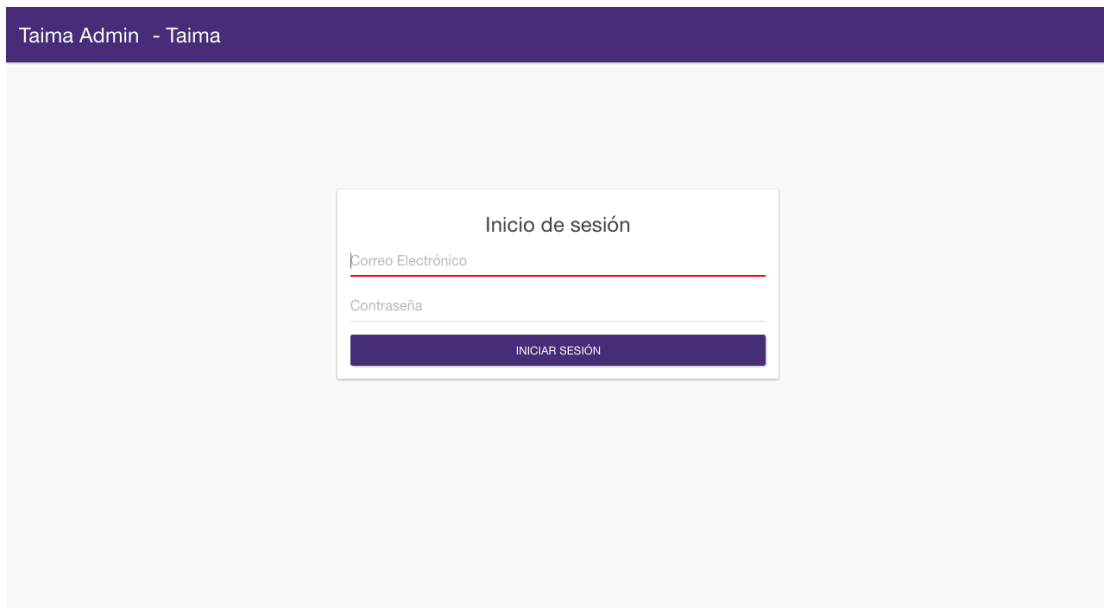


Figura 14. Pantalla de inicio de sesión (administrador)

En la figura 14 se puede observar la pantalla de inicio de sesión. El punto principal de esta vista es el formulario donde se solicitan los datos: correo electrónico y contraseña; propios de cada usuario administrador.

Una vez lleno el formulario y al hacer clic en el botón Iniciar Sesión, el sistema pasa a validar el formato del correo así como que exista al menos un caracter en el campo de contraseña. Luego de validar correctamente, el sistema utiliza la librería http de AngularJS para hacer la conexión de forma asíncrona con el API, donde se validan los datos contra los salvados en la base de datos. Para el inicio de sesión se utiliza el endpoint [GET] *api/users/:user* que recibe como parámetro esta estructura de datos:

En el backend la información recibida es validada y comparada con la base de datos. En caso de ser exitoso el backend se encargará de generar un Token único para el usuario, este token será el único elemento que podrá identificar al usuario durante el uso del sistema, y debe ser enviado en todas las solicitudes posteriores. El backend responde con un código exitoso y con el token como dato en la respuesta.

En el momento de recibir la respuesta el sistema verifica si es exitoso 200, o no. En caso de no serlo el sistema muestra una notificación en la parte inferior derecha con la información del fallo. En caso exitoso el sistema almacena el token único enviado y pasa a mostrar la pantalla principal.

Módulo administrativo del sistema y panel principal de sondeos:

Una vez autenticado el usuario es llevado a la pantalla principal del sistema administrativo, donde se encuentra la información más relevante de los sondeos: lista de sondeos en edición y publicados; así como el panel de creación de sondeo, como se puede ver en la figura 15.

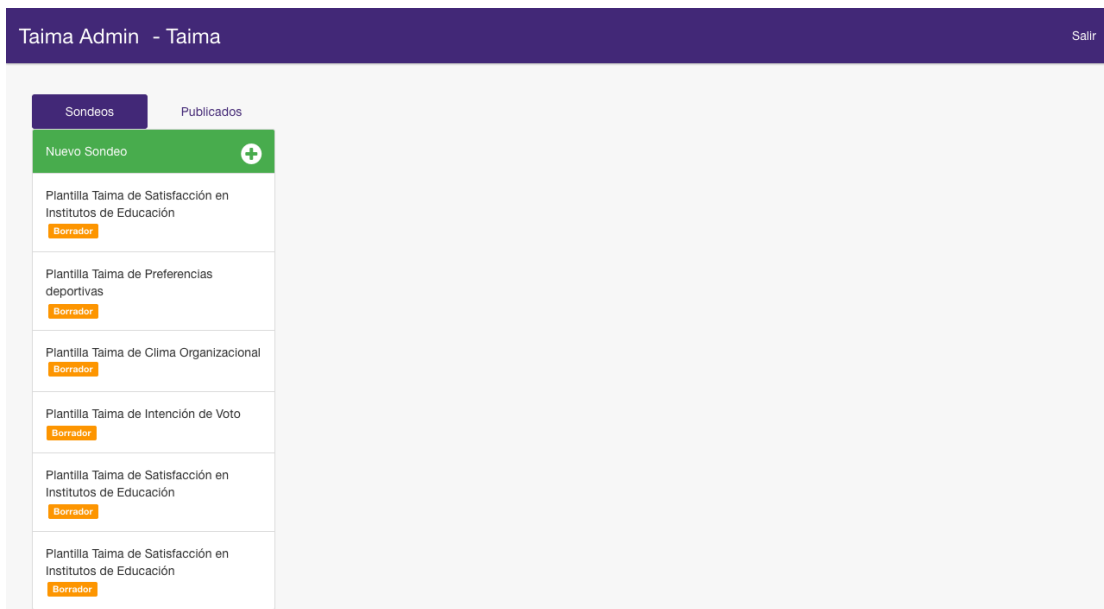


Figura 15. Vista de pantalla administrativa de sondeos.

Del lado izquierdo de la vista se observa la lista de sondeos creados por el administrador, separados en dos tabs, la primera **Sondeos** es donde se agrupan los sondeos creados pero sin publicar; la segunda **Publicados** se encuentran los sondeos ya publicados por el administrador.

El sistema administrativo de Taima contempla 2 estados para los sondeos:

1. Borrador: el sondeo fue creado, aún no ha sido publicado (se pueden realizar cambios)
2. Publicado: el sondeo fue creado, ya fue publicado (no se puede realizar cambios)

Esta vista se comunica con el endpoint de consulta de sondeos *api/polls/new* donde se genera una lista de todos los sondeos que el administrador ha creado.

Creación de sondeos:

Al hacer click en el botón de crear sondeo se actualiza el panel derecho de la vista en donde se ubica la primera parte del formulario de creación de sondeo.

The screenshot shows the 'Taima Admin - Taima' interface. On the left, there is a sidebar with a 'Sondeos' tab and a 'Publicados' tab. Under 'Sondeos', there is a 'Nuevo Sondeo' button with a plus icon. Below it, there are several template cards, each with a title, a 'Taima' button, and either a 'Plantilla' or 'Avanzado' button. The main content area is titled 'Creación de Sondeos' and contains a form with the following fields: 'Nombre del sondeo', 'Descripción del sondeo', 'Puntaje', 'Número de preguntas', 'Tamaño' (with a dropdown arrow), and 'Mensaje de finalización' (with a text area icon). At the bottom of the form is a green 'GUARDAR' button and a purple button with a plus icon.

Figura 16. Creación de sondeo.

Este primer bloque de información contempla los datos básicos y principales del sondeo como el nombre, descripción, puntaje, número de preguntas, tamaño y mensaje de finalización del sondeo. Se utiliza la estructura de formulario para poder mostrar los campos de solicitud de información, así como poder manejarlos de una forma adecuada. Uno de los principales factores positivos del uso de form es la validación de los datos de forma automatizada.

Una vez completa esta información el administrador hace click en el botón guardar, en este punto el sistema valida que la información de los campos tiene el formato correcto, en caso de que no lo tenga el usuario es notificado en el campo respectivo; si es exitoso, el sistema genera una petición CREATE al endpoint de creación: [POST] *api/polls/* donde es validada la

información y se crea el nuevo objeto de sondeo en la base de datos, en este objeto se realiza el guardado de la primera parte del sondeo y se envía una notificación 200 al frontend con el sondeo creado.

Al recibir la notificación de guardado exitoso el frontend pasa a habilitar las demás opciones del sondeo.

The image shows a web form for creating a survey. At the top, it says 'Sondeo prueba' with a red 'BORRAR' button. The form fields are as follows:

- Nombre del sondeo:** Sondeo prueba
- Descripción del sondeo:** Es un sondeo de prueba
- Puntaje:** 10
- Número de preguntas:** 5
- Tamaño:** Muy Corto (dropdown menu)
- Fecha de Expedición:** 13 / Abril / 2015 (date pickers)
- Fecha de Expiración:** Día / Mes / Año (date pickers)
- Mensaje de finalización:** Includes checkboxes for 'Generar Código QR' and 'Pregunta de rechazo', and a '+ IMAGEN DE SONDEO' button.
- Premios:** Includes an 'IMAGEN' button, a 'Descripción' text area, and 'Términos y Condiciones' text area.

At the bottom, there are two buttons: 'ACTUALIZAR' (purple) and 'ENVIAR' (green), followed by a dark purple bar with a white '+' icon.

Figura 17. Formulario de creación de sondeo, parte 2.

Como se puede observar en la Figura 17 los nuevos campos son: Fecha de expedición, Fecha de expiración, Generar código QR, Pregunta de rechazo, Imagen del sondeo, Premio; así como el botón de agregar pregunta (+) en la parte inferior de la vista.

Los elementos que se pueden agregar al sondeo a partir de este punto son opcionales, esto con el propósito de poder dejar a libertad del Administrador el nivel de complejidad que requiere cada sondeo. A continuación se describen los elementos y su funcionalidad:

- Generar código QR: este campo permite activar la opción de generar un código QR que sirva como referencia para cargar el sondeo. El objetivo detrás de esta funcionalidad es permitir vías promocionales extras para el consumo de los sondeos por parte de los opinadores, así como poder restringir a quienes podrán recibir el sondeo en cuestión. La idea principal es que el código sea utilizado en establecimientos físicos en forma de publicidad y promoción.

Para la creación del código QR se utilizó la librería angular-qr la cual se encarga de transformar el identificador único del sondeo en un código QR como imagen.

En la figura 18 se observa la imagen de código QR generado para un sondeo.



Figura 18. Código QR.

- Pregunta de rechazo: permite al administrador activar la opción de agregar una pregunta de rechazo al sondeo, esta funcionalidad nace de la necesidad de entender la posible razón por la cual un opinador expresa que no desea responder un sondeo en específico; esta información es muy valiosa para el feedback y el análisis de los sondeos publicados.
- Imagen del sondeo: este campo permite la subida o carga de una imagen que funcionará como imagen promocional del sondeo. Este elemento podrá ser de gran ayuda a la hora de promocionar un sondeo en la lista de sondeos disponibles para un opinador, así como una descripción gráfica de lo que se preguntará en cada sondeo.

La carga viene acompañada de una herramienta de corte de imagen, que permite al administrador un pequeño proceso de edición para garantizar que la imagen tenga las dimensiones adecuadas para el espacio que dispondrá en la vista.

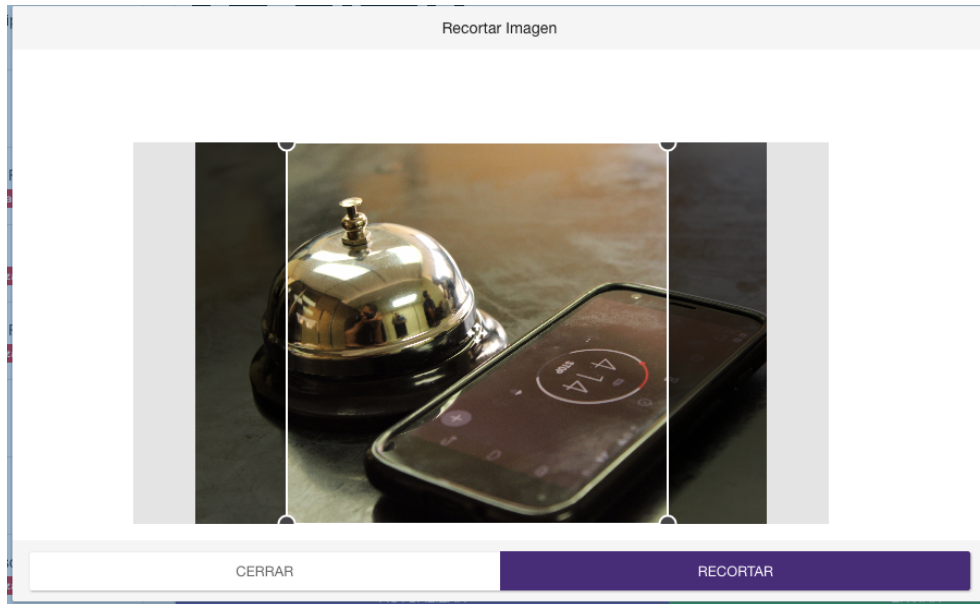
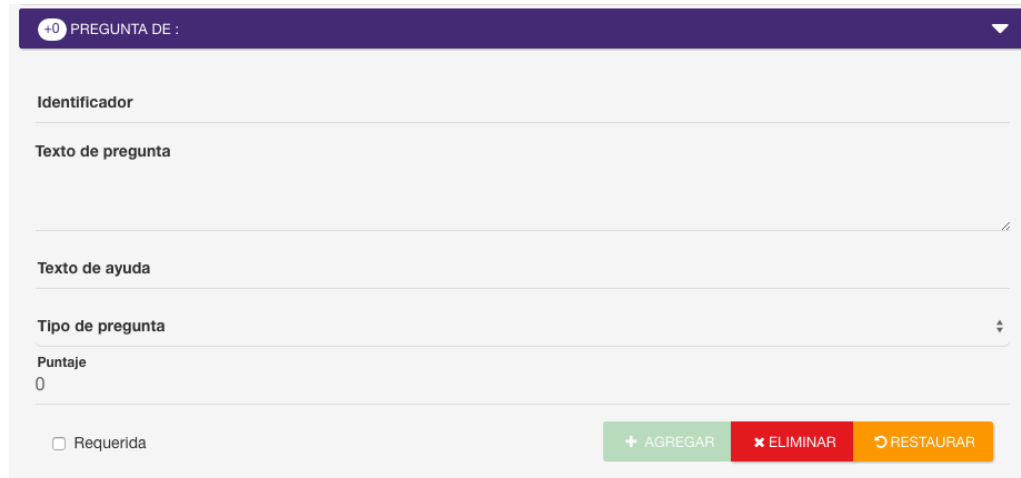


Figura 19. Corte de imagen en creación de sondeo.

- Agregar pregunta (+): elemento de gran importancia para la plataforma, en este punto se le da forma al sondeo a enviar, agregando preguntas por cada click en el botón (+) y hasta que se llegue al número de preguntas especificado en la parte inicial de la creación del sondeo.

El objeto pregunta contiene un grupo de características propios que el usuario administrador podrá elegir para concretar el sondeo de la forma deseada. Los campos son: Texto de pregunta, Texto de ayuda, Tipo de pregunta y requerida; siendo no requerido sólo el campo de texto de ayuda.



+0 PREGUNTA DE :

Identificador

Texto de pregunta

Texto de ayuda

Tipo de pregunta

Puntaje
0

Requerida

+ AGREGAR x ELIMINAR ↻ RESTAURAR

Figura 20. Creación de pregunta.

Como se observa en la figura 19, el objeto pregunta es tratado como un formulario dentro de un formulario ya que posee una estructura propia dentro del esquema de la base de datos. Un punto importante es la selección del tipo de respuesta que va a manejarse para esta pregunta, los posibles tipos son:

- Texto: cadena de caracteres, se espera que sea corta.
- Texto tipo párrafo: cadena de caracteres, se espera que sea larga.
- Selección simple: se definen opciones de respuesta de donde el opinador podrá escoger una.
- Selección múltiple: se definen opciones de respuesta donde el opinador podrá escoger una o varias.

Si el usuario administrador considera que la pregunta en creación representa un punto de avance obligatorio dentro del sondeo puede activar la opción de Requerida. Esto configurará una regla de no avanzar con respuesta en blanco, del lado de la aplicación del opinador.

Una vez definida la información de la pregunta se utiliza el botón de guardar (dentro de la pregunta) para validar los campos y guardar la pregunta como parte del sondeo.

Este proceso puede ser realizado tantas veces como lo permita el valor definido en el campo de número de preguntas.

Implementación de la segmentación:

Una vez alcanzado el contenido informativo ideal del sondeo, el usuario administrador puede decidir realizar la publicación del mismo, definiendo las características de la población objetivo.

Segmentación

Venezuela

Estado de residencia

Edad inicial Edad final

Género

Demográfico: 39 de 39 personas

Figura 21. Segmentación de sondeo.

En la Figura 20 se observa el panel desplegable con un formulario para definir: El país de residencia, el estado de residencia, el rango de edad y el género de la población objetivo. Se identifica también el número de opinadores que representa la población objetivo en comparación con la población total de opinadores registrados, este valor cambia automáticamente al hacer click en actualizar. Esta información es guardada como atributo del sondeo.

Una vez que todos los datos del sondeo y la segmentación estén definidos, el administrador puede decidir publicar el sondeo al hacer click en el botón **publicar**. Un punto importante es que si el administrador decide no publicar, y cierra el explorador de internet o cierra sesión, todos los campos actualizados serán guardados en la base de datos.

El sondeo está publicado en el momento que el administrador hace click en **publicar**. En este punto, el sondeo aparecerá en las búsquedas de sondeo que se realicen.

Implementación de gestión de resultados de sondeos:

El consumo de los resultados de los sondeos fue manejado lo más práctico posible, teniendo en cuenta dos escenarios de interacción, el primero que el administrador pueda ver de forma rápida los resultados de los sondeos creados, a través de la misma interfaz web; el segundo escenario contempla una interacción un poco más a fondo con los resultados, requiriendo de un proceso de exportación de resultados a un archivo universal de manejo de datos tabulados. Ambos escenarios se implementaron en la vista principal del sondeo con un mínimo de navegación para el usuario administrador.

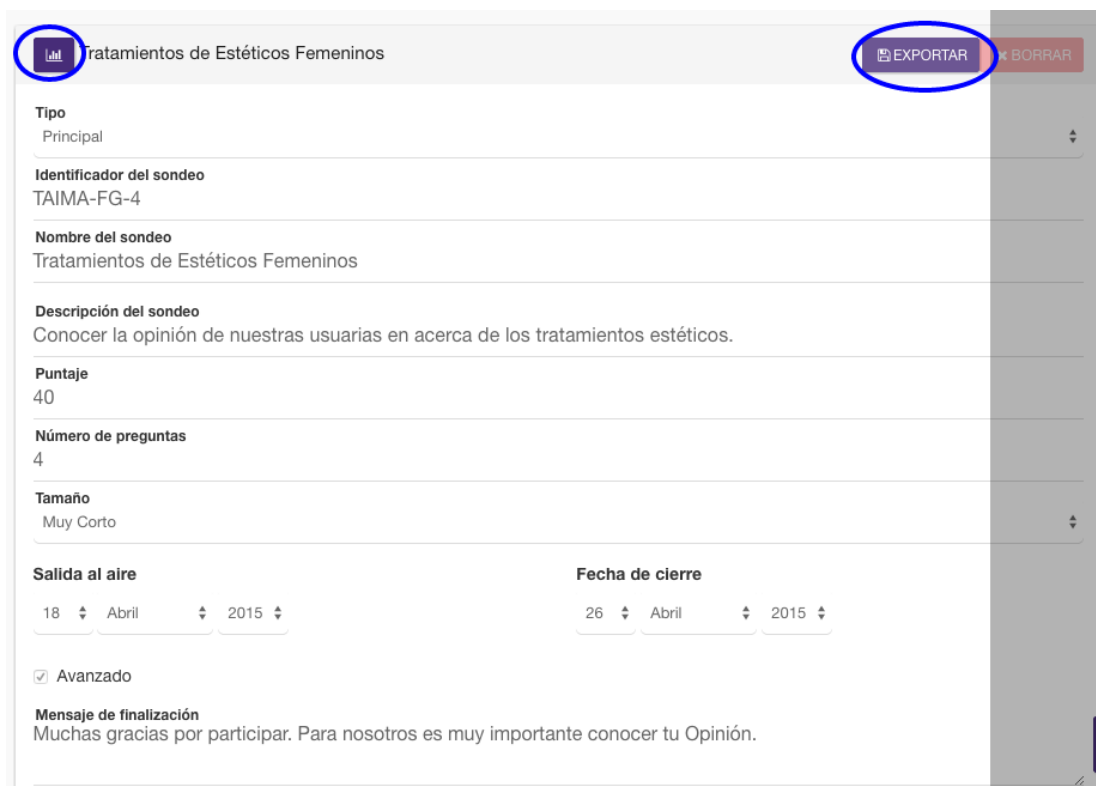


Figura 22. Botones de resultados y exportar resultados.

Como se observa en la figura 22 existen dos botones para consultar resultados de un sondeo, el de la izquierda para consultar resultados en la vista web y el botón exportar a la derecha para generar el archivo.

1. Resultados web: La vista se comunica con el API específicamente con el endpoint [GET] *api/polls/:poll/questions* en donde el servidor se encarga de recopilar todas las respuestas en la estructura de *interactions*. El backend se encarga de realizar los cálculos y enviar sólo la información necesaria para que el frontend pueda generar los gráficos de las respuestas.

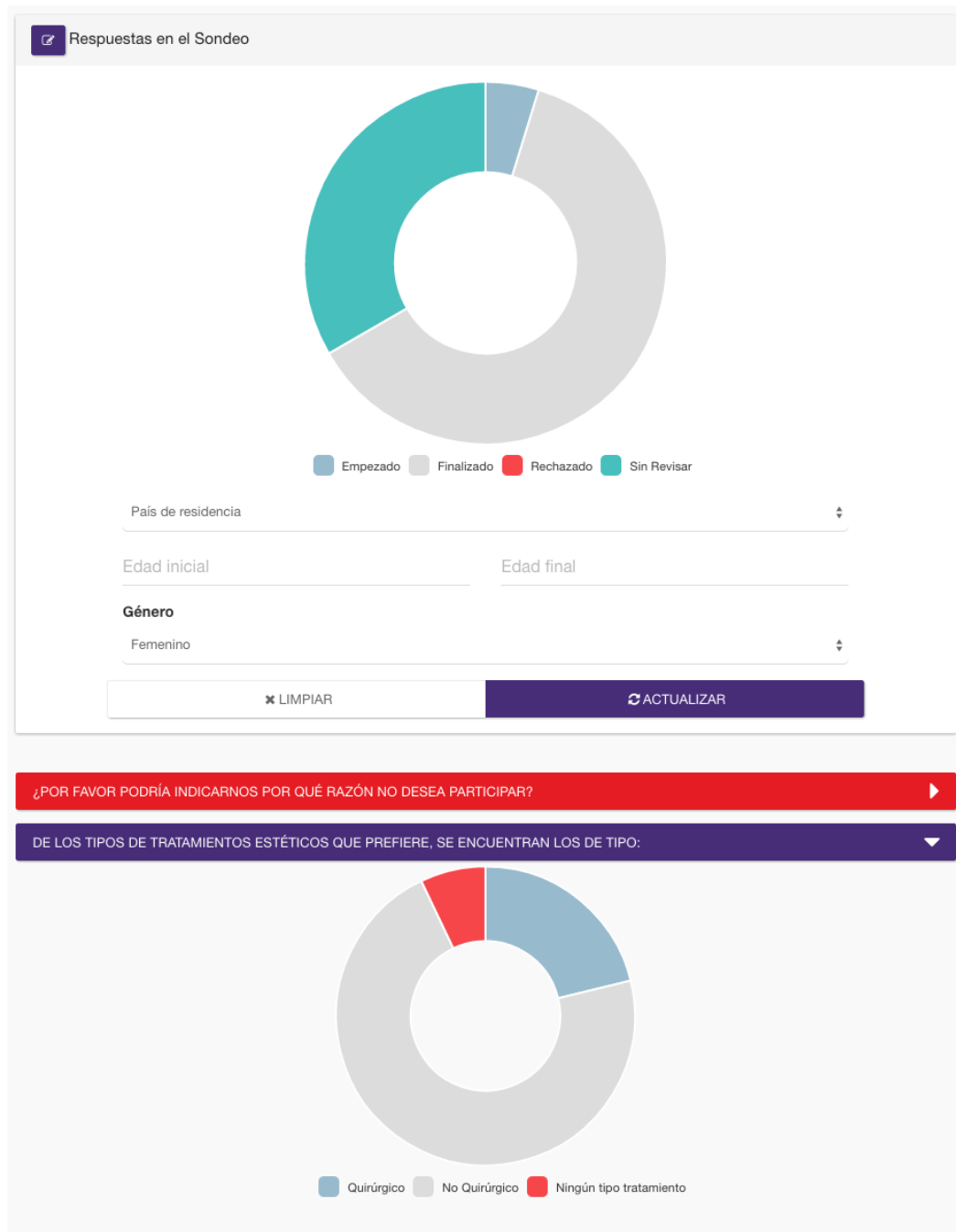


Figura 23. Resultados de un sondeo.

Como se observa en la figura 23 la interfaz muestra un resumen general de cuántos opinadores han iniciado, rechazado, finalizado y

cuales aún no han interactuado. Luego continúa con las gráficas de resultados para cada pregunta del sondeo que sea de tipo Selección simple o múltiple. Las preguntas con respuestas de tipo texto abierto se dejaron para el formato de exportación, ya que carece de sentido funcional porque ser muchas respuestas y muy largas, lo que las hace difíciles de mostrar y más aún de analizar.

2. Exportar resultados: pensando en un estudio más detallado se llegó a la conclusión que se debía generar un archivo de formato csv que contuviera todas las preguntas y respuestas generadas en el sondeo, de esta forma se abría la posibilidad de interpretar los resultados utilizando una herramienta experta de análisis de datos estadísticos. Con base en esto se definió una estructura tipo tabla para el archivo:

Exportar			
TAIMA-FG-4-1	TAIMA-FG-4-2	TAIMA-FG-4-3	TAIMA-FG-4-4
No Quirúrgico		Alguno pero ninguno de los anteriores	
No Quirúrgico		La depilación láser Peeling Químico	
Quirúrgico	Aumento de senos		
No Quirúrgico		Alguno pero ninguno de los anteriores	
Ningún tipo tratamiento			Prefiero estar completamente al natural
No Quirúrgico		La toxina botulínica tipo A	
Quirúrgico			
No Quirúrgico		La depilación láser	
No Quirúrgico		La depilación láser	
Quirúrgico	Liposucción		

SIGUIENTE

Figura 24. Formato de exportar resultados.

Configuración Módulo Aplicación móvil:

Como se comenta a lo largo del documento, la aplicación móvil es construida en el lenguaje Java, para dispositivos Android. Se configuró el ambiente de desarrollo utilizando Android Studio, con una versión objetivo de Android desde la 2.5 en adelante.

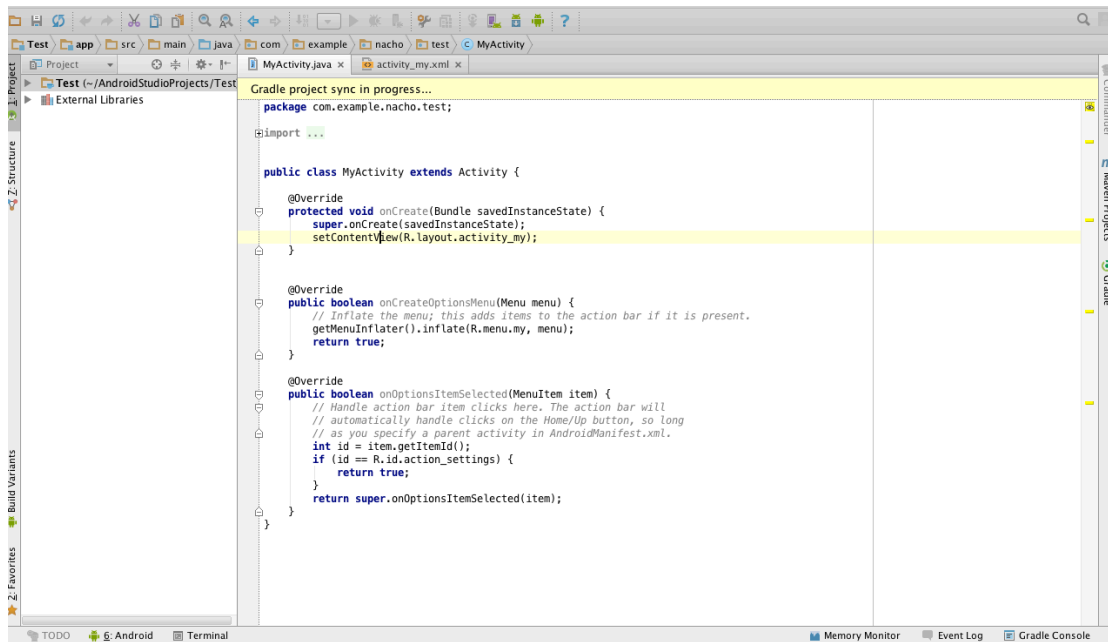


Figura 25. Android Studio. Editor de código Android + Compilador.

Implementación de Registro/Inicio de Sesión:

Esta aplicación tiene como punto inicial una pantalla de inicio que agrupa las dos primeras interacciones con el sistema para un usuario nuevo o ya registrado.

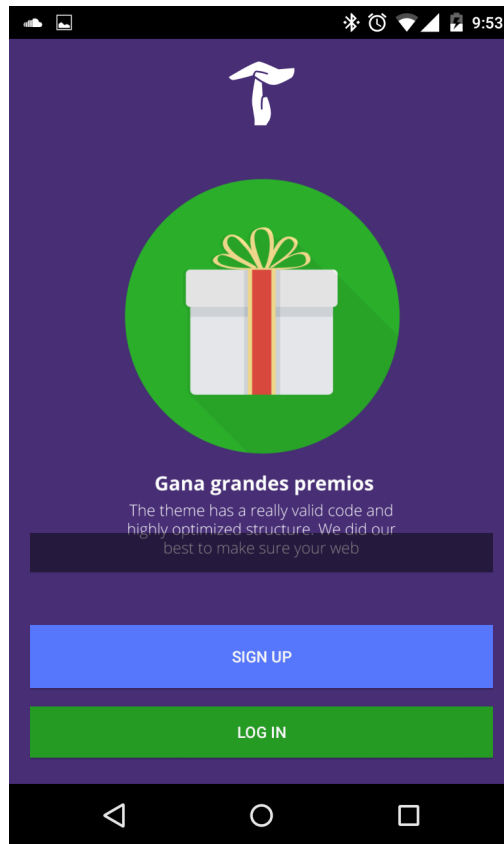


Figura 26. Pantalla inicial de la aplicación.

En la figura 26 se pueden observar los botones de inicio de sesión y registro en la vista de inicio de la aplicación.

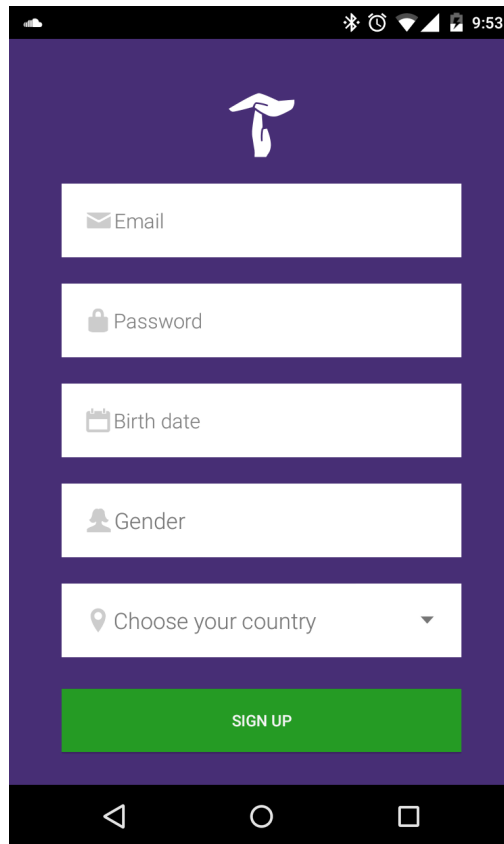


Figura 27. Pantalla de registro de opinador.

La vista de registro representa el punto de inicio para un nuevo usuario, como se observa en la figura 27 en esta vista el usuario podrá ingresar los datos solicitados y posterior a una validación la vista iniciará una comunicación HTTP con el API de taima.

Luego de un registro exitoso, el usuario podrá iniciar sesión en la aplicación. En este caso la interacción se da con la vista de Iniciar Sesión. En esta vista el sistema muestra los dos campos mínimos obligatorios para la validación del usuario: correo electrónico y contraseña como se muestra en la figura 27.

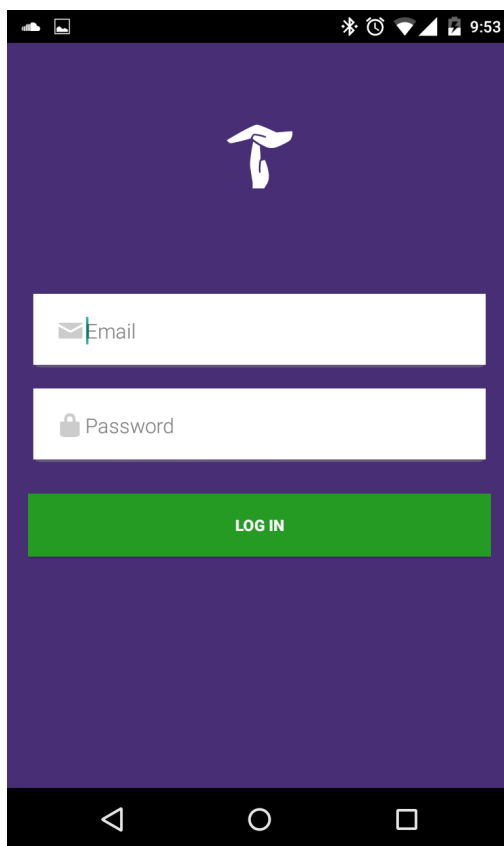


Figura 28. Pantalla de inicio de sesión.

Al igual que en la vista de registro, esta se comunica con el API el cual se encarga de validar los datos para permitir al usuario ingresar. Una vez validado, el backend genera un token único que funcionará como la llave de seguridad entre la aplicación y el backend para todas las futuras transacciones.

Pantalla principal de sondeos:

Una vez validadas las credenciales del usuario el sistema muestra la pantalla principal de la aplicación. En esta pantalla se definen dos puntos importantes, el menú de navegación y la lista de sondeos disponibles.

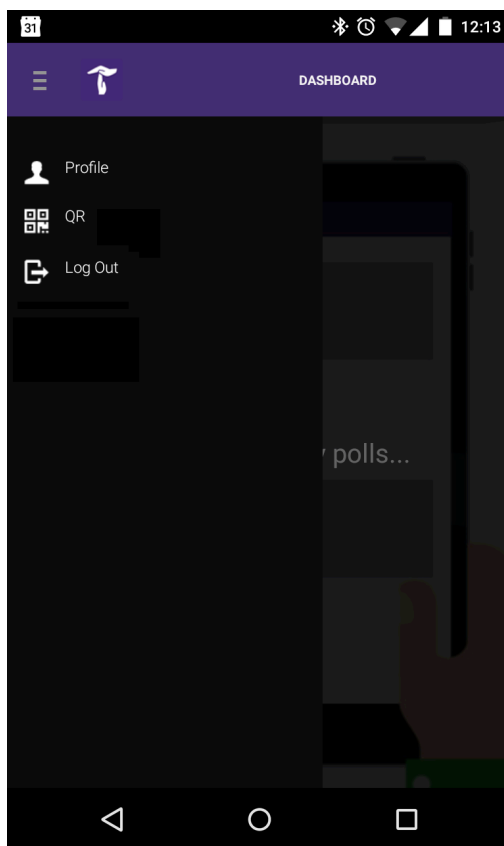


Figura 29. Menú de navegación.

Como se observa en la figura 29 se muestra el menú de navegación de la aplicación que incluye el acceso a las pantallas de Perfil y Escaneo de código QR que serán explicadas posteriormente, y el botón de Cerrar sesión.

La lista de sondeos o dashboard, corresponde al lugar donde se agrupan todos los sondeos que el opinador tiene disponible para responder. Cada sondeo de la lista contiene información resumida del sondeo que representa. Se definieron los siguientes valores como elementos informativos de valor para una primera presentación del sondeo:

- Título del sondeo.
- Imagen del sondeo

- Cantidad de puntos a recolectar
- Fecha de expiración del sondeo
- Estimación de duración de sondeo
- Botón de información del sondeo



Figura 30. Pantalla de sondeos (opinador)

En la figura 30 se observa la distribución de los elementos en la ficha del sondeo.

Esta vista principal se comunica con el endpoint [GET] *api/polls/new* encargado de enviar la lista de sondeos filtrados por segmentación.

Existen dos interacciones secundarias con respecto al flujo natural de la aplicación en este contenedor de sondeo, el botón de rechazo de sondeo y el botón de información de sondeo.

El primero representa el procedimiento a llevar a cabo cuando un opinador decide no querer responder el sondeo, en este punto el opinador hace click en la parte superior derecha de la ficha. En ese punto la aplicación muestra una vista modal con una respuesta de selección simple con la cual se busca especificar cuál fue la razón de rechazo de sondeo.

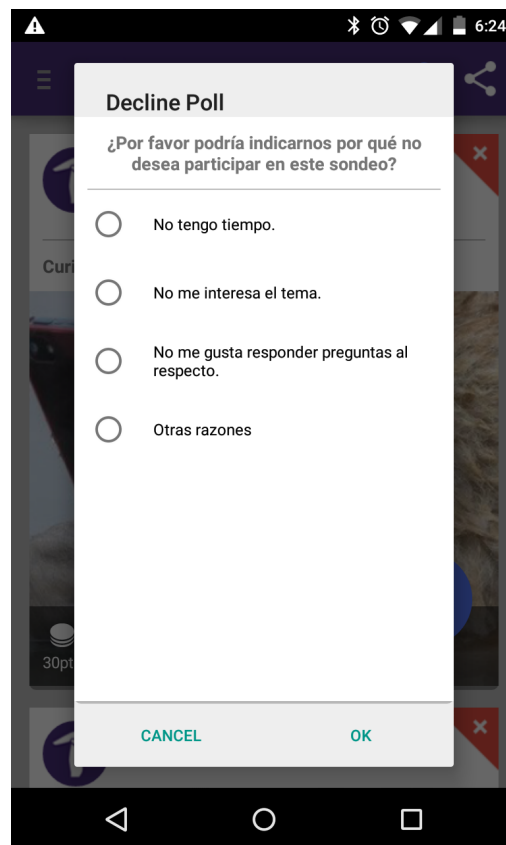


Figura 31. Pantalla de rechazo de pregunta.

El segundo representa la información adicional sobre el sondeo seleccionado. Al igual que el rechazo, la acción despliega una vista modal

que en este caso muestra la información detallada del sondeo seleccionado como se puede observar en la figura 32:

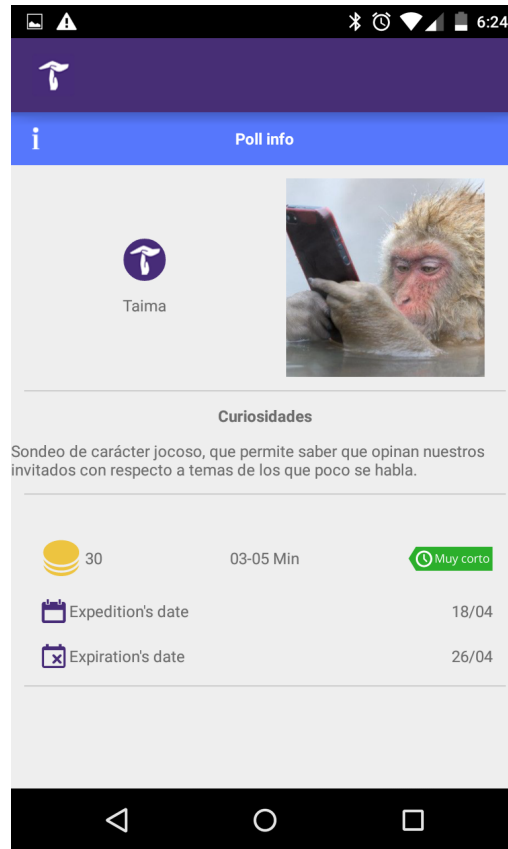


Figura 32. Pantalla de información de sondeo.

Interacción con sondeos:

Al seleccionar un sondeo la aplicación muestra la vista de sondeo en la que se define la interacción principal del opinador con la plataforma.

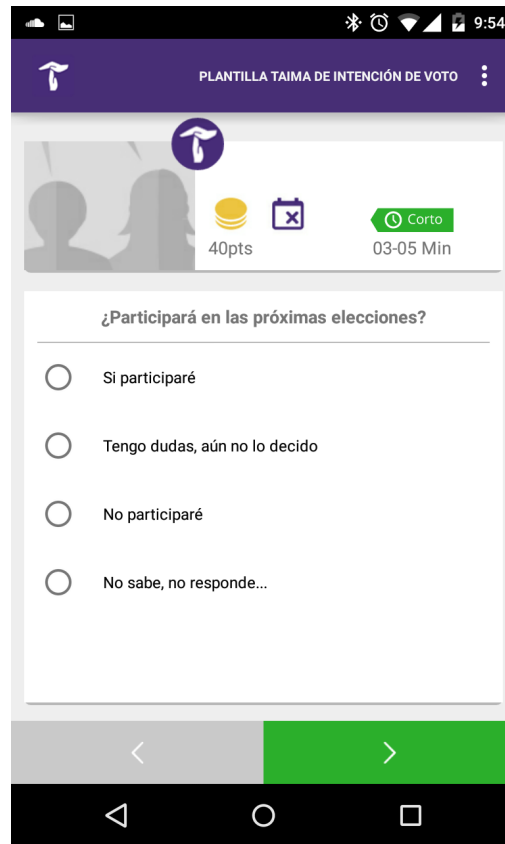


Figura 33. Pantalla de pregunta (móvil)

En la figura 33 se observa la estructura de la vista. En la parte superior se muestra el resumen del sondeo, con elementos resumidos. Luego de esto se presenta la pregunta a responder. La aplicación se configuró para mostrar una pregunta a la vez, esto para garantizar la atención del opinador en cada una. Como lo especifican los requerimientos se manejan tres formatos de respuesta:

1. Selección simple: es una lista de radio buttons que muestran las distintas opciones a responder. En este caso el opinador sólo puede seleccionar una opción.

2. Selección múltiple: es una lista de checkboxes que muestran las distintas opciones a responder. En este caso el opinador puede elegir una o más opciones como respuesta.
3. Texto abierto: este tipo de respuesta espera del usuario un texto abierto, donde el opinador puede expresarse libremente.

La aplicación maneja la posibilidad de permitir al usuario avanzar a una siguiente pregunta o retroceder a una anterior utilizando los botones de navegación inferiores. En el caso de que la respuesta de una pregunta sea requerida, el sistema notifica al usuario que debe responderla para poder avanzar. La notificación se maneja con la función Toast de Java Android, se puede observar en la figura 34 dicha notificación.

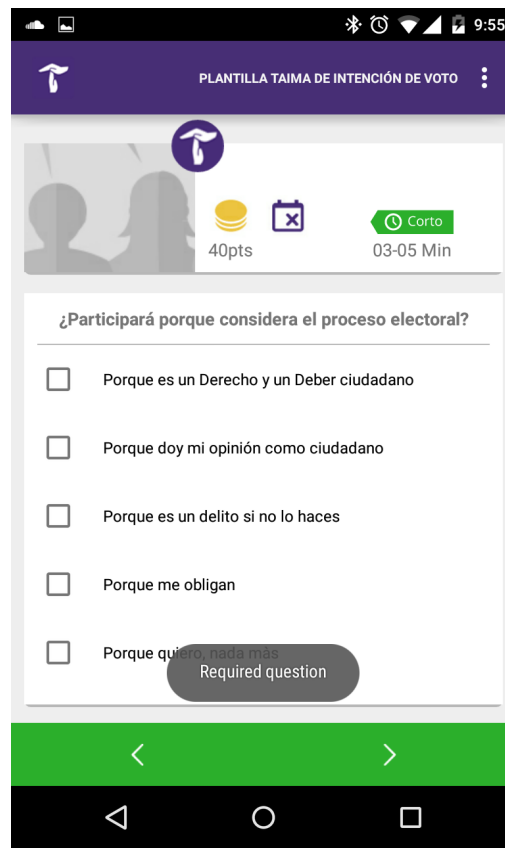


Figura 34. Notificación Toast en Android.

Los opinadores serán notificados en el momento que terminen las preguntas y se proceda a cerrar el sondeo. Una vez finalizado, el opinador es llevado a la pantalla principal de la aplicación.

Para un manejo efectivo de los distintos formatos de respuesta se utilizaron los Fragments de Android, este acercamiento permite diseñar varios formatos visuales que pueden variar su contenido de forma dinámica dependiendo de los valores que se asignen desde la actividad que los contiene.

Implementación de lector QR:

Como requerimiento no funcional se busca que el opinador pueda interactuar con un elemento físico que represente un canal de acceso a los sondeos, con base en esto se decidió utilizar el formato QR como medio interactivo, de esta forma el opinador puede hacer un escaneo con la aplicación de un código impreso que lo llevará directo al sondeo en cuestión.

Para realizar la implementación se apoyó en la librería Zxing, que utiliza la cámara del dispositivo móvil como scanner para leer el código, extraer la identificación del sondeo y que la aplicación lo interprete.

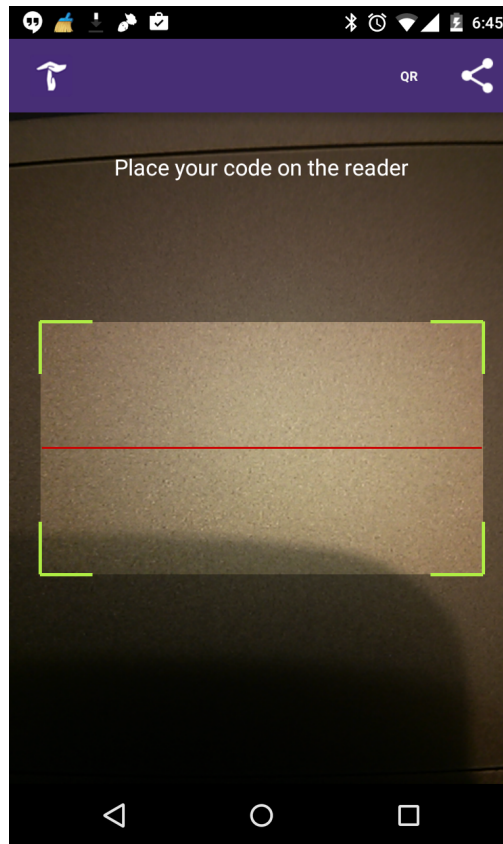


Figura 35. Lectura de QR.

En la figura 35 se observa la funcionalidad de escaneo QR.

Perfil de usuario:

Los datos de registro del opinador son de alta importancia para la plataforma ya que definen los valores por los que va a ser segmentado el opinador. Se definió la funcionalidad que permite al opinador actualizar sus datos de manera que se mantenga al día con cualquier cambio que aplique a su registro. Para lograr esto, se implementó una vista en la app que permitiera dicha acción.

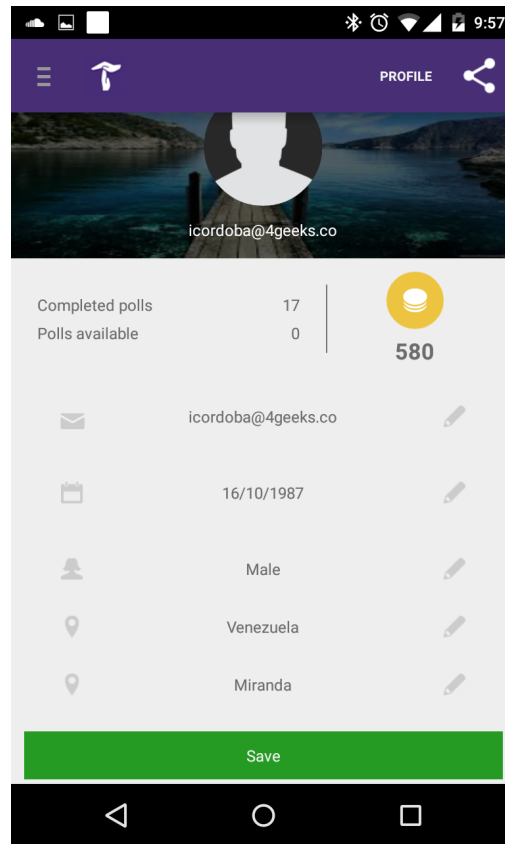


Figura 36. Perfil de opinador.

En la figura 36 se observa la vista de perfil, con los datos del usuario que ingresó en su registro. Se muestra también un resumen de los sondeos respondidos y disponibles así como los puntos acumulados por responder sondeos.

En la vista se incluye un icono de edición, representado por un lápiz, que acompaña cada elemento. La interacción se ejecuta al hacer click en el icono, lo que activa la edición de la información del campo.

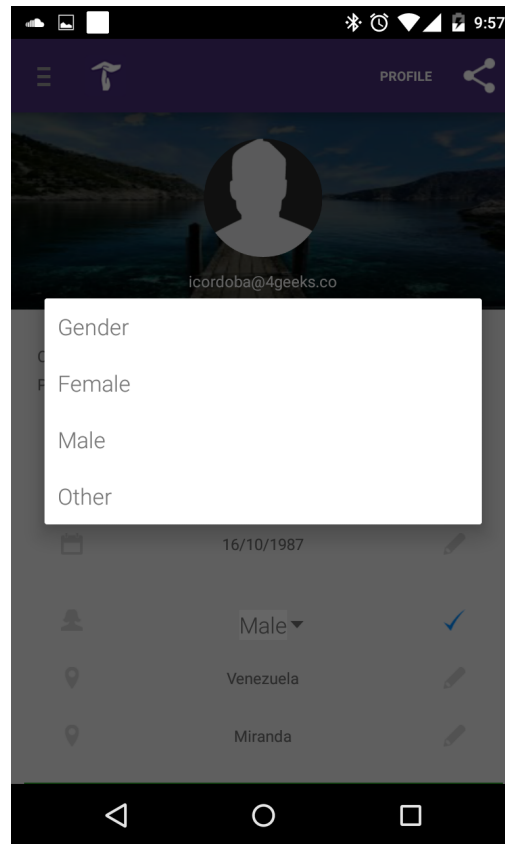


Figura 37. Editar perfil opinador.

4.3 Pruebas y aceptación

4.3.1 Pruebas de usuario final

Para culminar con el desarrollo de la aplicación móvil, fueron aplicadas pruebas de usabilidad con el objetivo de corroborar que el módulo desarrollado es un producto de software usable, tolerante a fallas y que cumple con el funcionamiento esperado. Las pruebas de usabilidad consistieron en un cuestionario donde se empleó la escala de Likert, en esta se plantean enunciados, ante los cuales el individuo debe mostrar su acuerdo o desacuerdo. (Nogales, 2004)

Se utilizaron cinco alternativas de respuestas para cada enunciado: totalmente de acuerdo, de acuerdo, ni de acuerdo ni en desacuerdo, en desacuerdo y totalmente en desacuerdo.

El cuestionario se realizó mediante un formulario físico y fue aplicado a 20 personas que variaban entre estudiantes y personas que no pertenecen al área de Computación.

A continuación el cuestionario utilizado:

Con respecto a la interfaz gráfica:

Tabla 13. Preguntas sobre interfaz gráfica.

Preguntas sobre Interfaz					
	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni endesacuerdo	De acuerdo	Totalmente de acuerdo
Es fácil comprender las acciones que se pueden realizar					
Es fácil de aprender (En poco tiempo se conoce las funciones)					
Es fácil de usar (Las acciones tienen bajo nivel de complejidad)					
Los colores son agradables					
La experiencia con la interfaz fue positiva					

Con respecto a la información que provee la interfaz:

Tabla 14. Preguntas sobre información de interfaz.

Preguntas sobre Interfaz					
	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
Es compatible					
Es útil para lograr llevar a cabo un objetivo					
Las funciones del menú son claras					
Los mensajes aportan significado					

Tabla 15. Preguntas opciones de usuario.

Preguntas opciones del Usuario					
	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
Considera que es fácil responder los sondeos					
El tiempo empleado en responder el sondeo es adecuado					

Resultados de las pruebas

Del cuestionario presentado anteriormente, se pudieron obtener los siguientes resultados:

Las primeras seis gráficas corresponden a la interacción del usuario con la Interfaz Gráfica de la aplicación.

La figura 38 muestra que las acciones que se pueden realizar mediante la aplicación son fáciles de comprender ya que el 95% de las personas encuestadas estuvieron entre “Totalmente de acuerdo” (70%) y “De acuerdo” (25%).

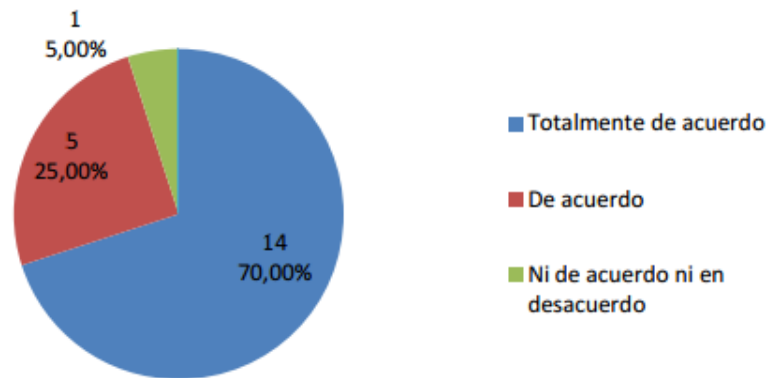


Figura 38. La interfaz es fácil de entender.

Las figuras 39 y 40 indican que la aplicación es tanto fácil de aprender como fácil de usar; puesto que en ambas el 95% de las respuestas oscilan entre “Totalmente de acuerdo”(70%) y “De acuerdo”(25%).

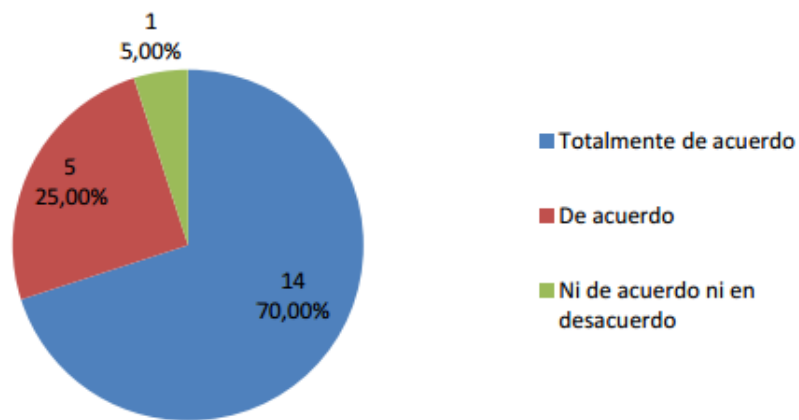


Figura 39. Fácil de aprender.

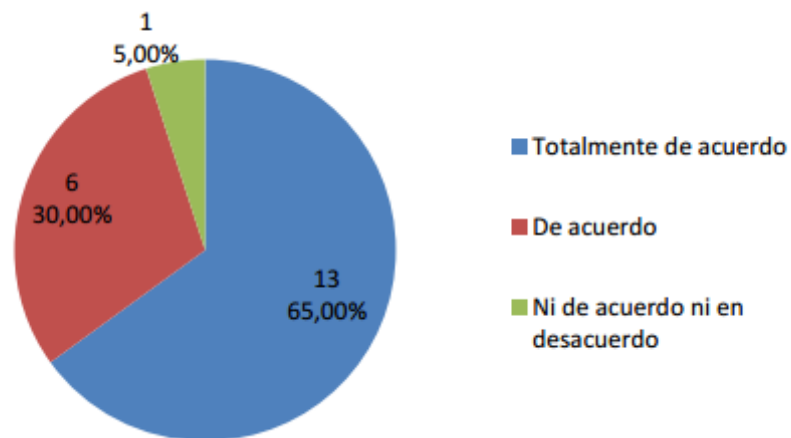


Figura 40. Fácil de usar.

La figura 41 muestra como la mayoría de las personas coincidieron en que los colores de la aplicación les resultan agradables.

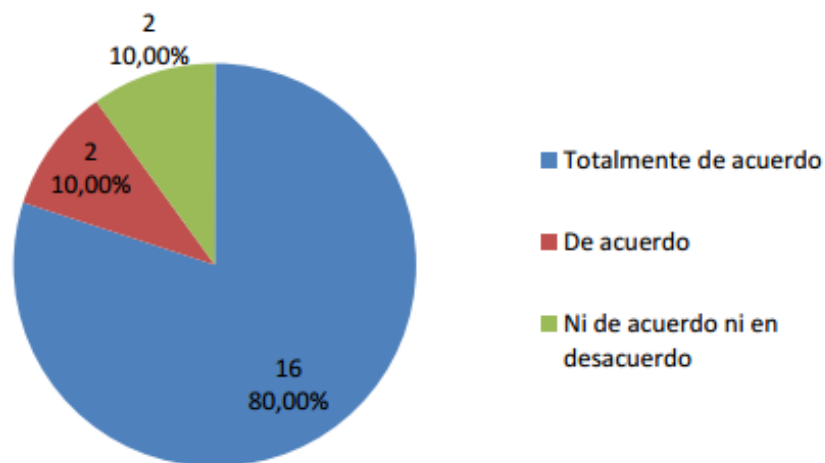


Figura 41. Los colores son agradables.

La figura 42 muestra que el 95% de los encuestados tuvieron una experiencia positiva con la interfaz de la aplicación; siendo este resultado consistente con los resultados previamente analizados referentes a la interacción del usuario con la aplicación y con la interfaz gráfica de la misma.

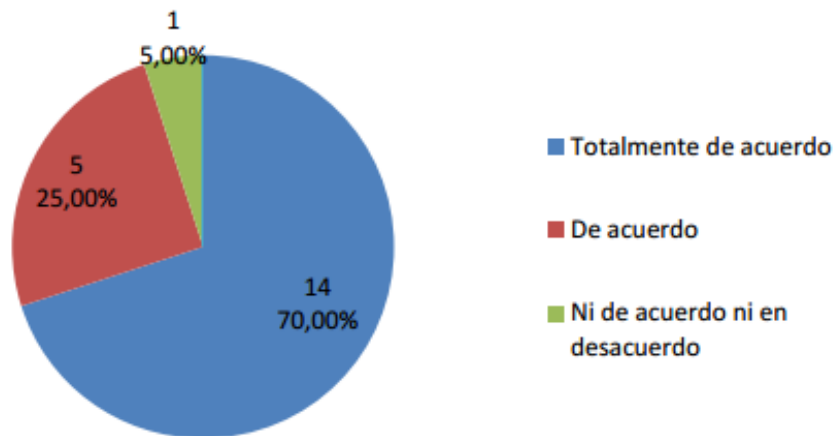


Figura 42. La experiencia con la interfaz fue positiva.

La figura 43 muestra que la información que provee la interfaz es comprensible para el usuario, puesto que el 55% de los encuestados están totalmente de acuerdo con esto, el 40% estuvo de acuerdo, y sólo un 5% representado por una persona estuvo en desacuerdo.

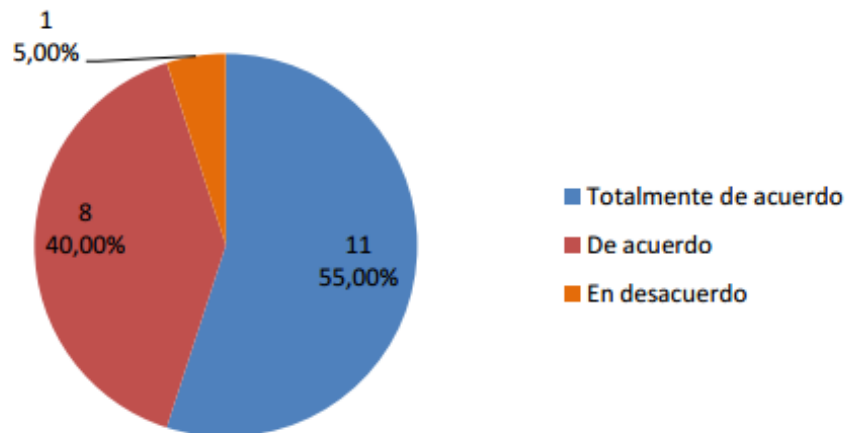


Figura 43. Es comprensible.

La figura 44 muestra que la información que provee la interfaz le resulta útil a los usuarios para llevar a cabo sus objetivos dentro de la aplicación, ya que

el 90% de los encuestados estuvo entre “Totalmente de acuerdo” (80%) y “De acuerdo” (10%) con esto.

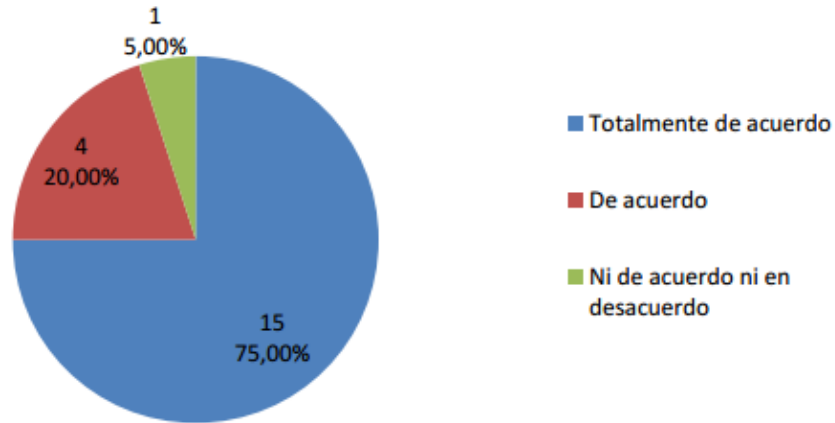


Figura 44. Es útil para lograr llevar a cabo un objetivo.

La figura 45 muestra que las funciones del menú son claras, ya que el 95% de los encuestados estuvo entre “Totalmente de acuerdo” (75%) y “De acuerdo” (20%) con esto.

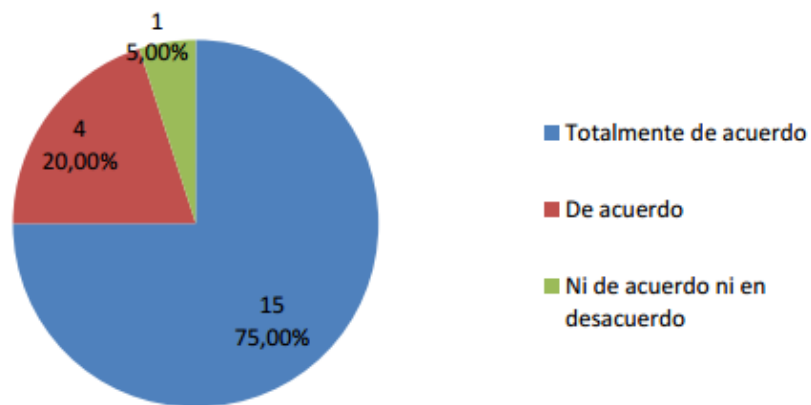


Figura 45. Las funciones del menú son claras.

La figura 46 muestra que los mensajes que provee la interfaz aportan significado a los usuarios, ya que el 95% de los encuestados estuvo entre “Totalmente de acuerdo” (65%) y “De acuerdo” (30%) con esto.

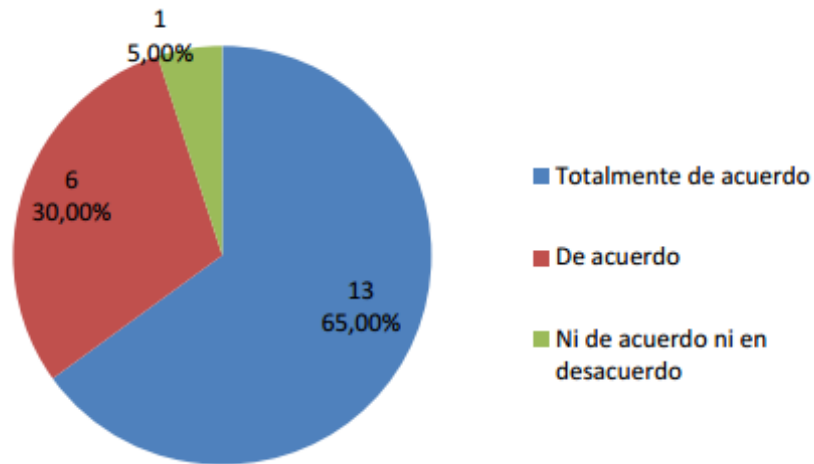


Figura 46. Los mensajes aportan significado.

Con respecto a la utilidad del Módulo de Historial de Búsqueda, se observa en la figura 47 que el 100% de los encuestados concuerda en que este módulo le facilita sus búsquedas, puesto que el 90% estuvo “Totalmente de acuerdo” con esto y el 10% estuvo “De acuerdo”.

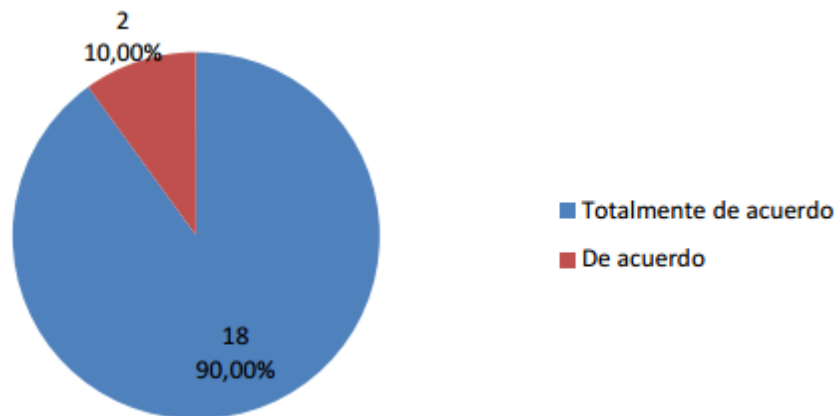


Figura 47. Facilita las búsquedas.

La figura 48 muestra que la evaluación es fácil de realizar, ya que el 95% de los encuestados estuvo entre “Totalmente de acuerdo” (65%) y “De acuerdo” (30%) con esto

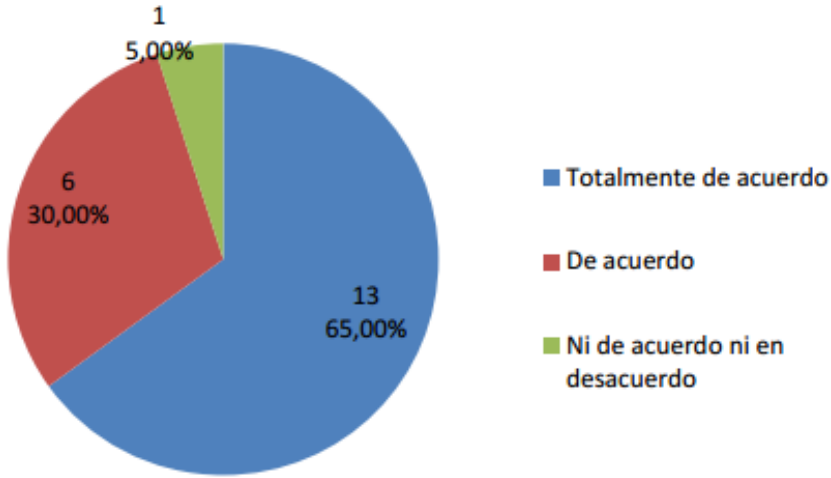


Figura 48. Responder sondeos se hace fácilmente.

La figura 49 muestra que el tiempo empleado en realizar la evaluación es óptimo, ya que el 95% de los encuestados estuvo entre “Totalmente de acuerdo” (65%) y “De acuerdo” (30%) con esto.

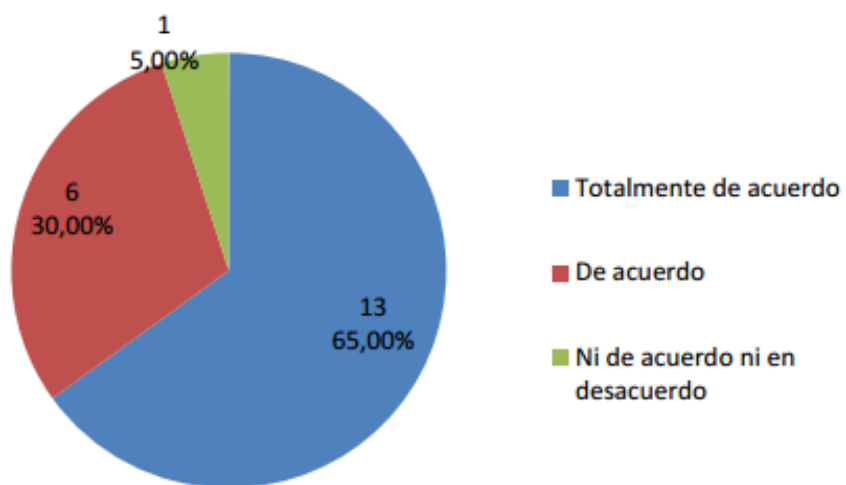


Figura 49. El tiempo para la realización de las respuestas es óptimo.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

Al culminar este trabajo se considera que se cumplieron los objetivos planteados ya que se realizó el desarrollo de un sistema para la automatización en el proceso de creación, despliegue y gestión de sondeos. Para cumplir el objetivo principal del trabajo de grado y los objetivos derivados del mismo, se utilizó SCRUM como metodología de desarrollo, Javascript con NodeJS y AngularJS como stack tecnológico para la web y Android como plataforma de desarrollo móvil, obteniendo resultados satisfactorios.

La herramienta permite:

- a) Crear sondeos fácilmente porque proporciona un diseño web sencillo y usable para usuarios.
- b) Ver y analizar resultados, ya que puede aplicar rápidamente funciones avanzadas a un sondeo.
- c) Utilizar un flujo de trabajo simple en un sondeo.
- d) Crear sondeos para ayudarle a responder rápidamente las necesidades del cliente.
- e) Garantizar que las respuestas obtenidas sean estadísticamente correctas ya que se puede segmentar el público objetivo.
- f) Acceso automatizado y validado para que puedan interactuar masas de opinadores.
- g) Responder sondeos fácilmente porque proporciona un diseño móvil sencillo y usable para los opinadores.
- h) Actualizar los perfiles de los opinadores directamente en la aplicación móvil.
- i) Interacción física con elemento de promoción código QR para sondeos promocionales.

Por medio de la encuesta realizada, se obtuvo opiniones, comentarios y sugerencias sobre la aplicación realizada. Los resultados obtenidos a través de las encuestas son alentadores para la continuación del proyecto, pues tuvo un alto grado de aceptación. Además de esto, la retroalimentación conseguida a través de las personas encuestadas brindó un valioso aporte para la mejora de la aplicación.

El sistema en este punto está en una etapa inicial, de donde podría derivar en una plataforma completa y avanzada para la gestión de sondeos. Con base en esto algunas recomendaciones serían:

- Desarrollo de otras funcionalidades sobre la creación y consumo de los sondeos, como:
 - Aplicar flujos de respuestas a los sondeos.
 - Aplicar una capa de gráficos para las respuestas.
- La plataforma ofrece una oportunidad de negocio importante que podría explotarse si se desarrolla un módulo para clientes, en donde puedan crear sus propios sondeos.
- Expandir el alcance de los opinadores desarrollando aplicaciones en otras plataformas como iOS y Web.
- Clasificaciones en los sondeos para que pudieran ser filtrados por temas de interés de los opinadores.

REFERENCIAS BIBLIOGRÁFICAS

Agile Alliance. (04 de 04 de 2015). *What is Agile Software Development?* From Agile Alliance: <http://www.agilealliance.org/the-alliance/what-is-agile/>

Bondi, A. (2000). Characteristics of scalability and their impact on performance. *Proceedings of the 2nd international workshop on software and performance* , 195-203.

Butler, L. M. (1995). *The "Sondeo", a rapid reconnaissance approach for situational assessment*. Estados Unidos de Norteamérica: United States Department of Agriculture.

CEO. (2001). *Las características de los Sondeos de Opinión Pública*. (U. d. Antioquia, Ed.) Antioquia, Colombia: Centro de Estudios de Opinión.

Codd, F. (1969). *Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks*. IBM Research Report.

Cowles, E., & Nelson, E. (2015). *An Introduction to Survey Research*. Business Expert Press.

Cunningham, C. &. (07 de 04 de 2015). *Event Driven Programming*. From Cunningham & Cunningham, Inc.: <http://c2.com/cgi/wiki?EventDrivenProgramming>

Davenport, T., & Prusak, L. (2000). *Working Knowledge: How organizations manage what they know*. Estados Unidos de Norteamérica: Working Knowledge: How organizations manage what they know.

Dayley, B. (2014). *Node.js, MongoDB, and AngularJS Web Development*. Addison-Wesley Professional.

Denso Wave Incorporated. (17 de 04 de 2015). *QRCode.com*. From QRCode.com: <http://www.qrcode.com/en/>

Erl, T., Carlyle, B., Pautasso, C., & Balasubramanian, R. (2012). *SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST*. Prentice Hall.

Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Irvine, Estados Unidos de Norteamérica: University of California.

Fisher, R. A. (1953). *Statistical Methods and Scientific Inference*. Edinburgh: Oliver & Boyd.

Joyent. (04 de 04 de 2015). *Node.js*. From <https://nodejs.org/>

Krol, J. (2014). *Web Development with MongoDB and Node.js*. Packt Publishing.

Laudon, & Laudon. (1998). *Administración de los sistemas de información: organización y tecnología*. México: Prentice Hall.

Likert, R. (1932). A Technique for the Measurement of Attitudes. *Archives of Psychology* (140), 1-55.

Luján Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos.

McCreary, D., & Kelly, A. (2013). *Making Sense of NoSQL: A guide for managers and the rest of us*. Manning Publications.

Nginx. (04 de 04 de 2015). *Nginx News*. From <http://nginx.org/>

Oppel, A. (2009). *Databases*. Estados Unidos de Norteamérica: Mc Graw Hill.

RAE. (10 de 04 de 2015). *Definición de sistema*. From Diccionario de la lengua española: <http://lema.rae.es/drae/?val=sistema>

Rea, L. M., & Parker, R. A. (2014). *Designing and Conducting Survey Research: A Comprehensive Guide, 4th Edition*. Jossey-Bass.

Refsnes Data. (01 de 04 de 2015). *W3Schools*. From <http://www.w3schools.com/>

Retallick, B. (16 de 05 de 2011). *Types of Information Systems*. From CSE1IS Systems Week 2 Lecture 2: <http://149.144.20.200/subjects/IS/sem22010/lectures/l022istypes.shtml>

Schwaber, K., & Beedle, M. (2002). *Agile Software Development with Scrum*. Prentice Hall.

Topolski, J. (2007). *Google's Android OS early look SDK now available*. From Engadget: <http://www.engadget.com/2007/11/12/googles-android-os-early-look-sdk-now-available/>

Vaish, G. (2013). *Getting Started with NoSQL*. Packt Publishing.

W3C. (2004). *Architecture of the World Wide Web, Volume One*. From The World Wide Web Consortium: <http://www.w3.org/TR/2004/REC-webarch-20041215/>

Webber, J., Parastatidis, S., & Robinson, I. (2010). *REST in Practice*. O'Reilly Media, Inc.