

INTRODUCCIÓN

1. INTRODUCCIÓN

Hoy en día la tecnología informática juega un papel importante en nuestra sociedad, en consecuencia al aporte que brinda para facilitar la comunicación y el trabajo en diferentes sectores, es por esto que el desarrollo de las aplicaciones móviles se ha expandido en los últimos años debido, por un lado, a la masificación de los teléfonos móviles con mejores capacidades y funciones, mejorando significativamente el acceso a servicios de datos, y por otro, a la aparición de herramientas de software que facilitan su desarrollo y aplicaciones.

La Liga Boliviana de Robótica, es el resultado de varias iniciativas en Bolivia, que buscan realizar actividades de competencia, en favor de la educación en el ámbito de la robótica. Diferentes grupos académicos y desde varias ciudades del país, vienen realizando competencias de robótica desde hace muchos años, motivando la participación de estudiantes de nivel escolar y universitario. El crecimiento de actividades competitivas ha creado escenarios de investigación y el incremento notable de participantes en cada año, logrando conformar equipos y delegaciones que salen a competir internacionalmente, obteniendo cada vez resultados importantes y muy satisfactorios para Bolivia.

El presente trabajo permitirá tener una aplicación móvil, que constará de 3 partes: módulo administrativo, módulo tutor y módulo participante. El módulo administrativo permitirá registrar: organizadores, tutores, información del evento, actividades y obtener reportes de los inscritos. El módulo tutor será para realizar los registros de inscripción de los participantes. El módulo participante permitirá visualizar la información del evento, haciendo conocer el lugar de la competencia y las diferentes actividades que se realizarán, también permitirá que el participante previamente ya inscrito pueda acceder con su nombre de usuario y clave, para realizar su registro de pago por inscripción de la competencia.

Para el desarrollo de este trabajo se utilizarán los lenguajes y herramientas más apropiadas, que permitan un trabajo óptimo y brinden resultados de calidad. El almacenamiento de la información se lo implementará con un gestor de base de datos en la red de Internet, con el propósito de que la información pueda ser actualizada de forma instantánea y en tiempo real.

El desarrollo de la aplicación, será trabajado utilizando la metodología Scrum y para el modelado se aplicará UML “El Lenguaje Unificado de Modelado”.

1.1. Descripción del Problema

Muchos estudiantes que se encuentran en etapa escolar y universitaria, reciben convocatorias mediante sus unidades educativas y también universidades, para participar en concursos departamentales, nacionales e internacionales de robótica; que fomentan la educación y desarrollo de conocimiento tecnológico para niños y jóvenes.

La inscripción de los participantes en una competencia se realiza de forma manual o con el uso de un formulario digital que no presenta información suficientemente integrada a la mayoría de las actividades involucradas. Es muy frecuente que la información se la proporcione a los participantes mediante grupos de WhatsApp, esto con lleva a que los interesados reciban alguna información que tiene el evento, pero de manera informal y desorganizada, debido a esta situación, es que la información se acumula y se cruzan las temáticas de consultas o anuncios. Por otro lado, la mayoría de organizadores y eventos no cuentan con una página específica, que aporte a la integración de datos e interactividad de los participantes, con información organizada y directa para cada interesado (organizadores, tutores y participantes en general).

1.2. Planteamiento del Problema

La liga boliviana de robótica de la ciudad de Tarija utiliza formularios físicos, formularios de Google, también páginas de Facebook y envío de información por medio del WhatsApp para la organización del proceso de inscripción de los participantes. Esta forma de recepcionar y enviar información, dificulta tanto a los organizadores y participantes para mantener su información organizada y actualizada sobre todo lo que se involucra, en las actividades de inscripción y organización de una competencia.

1.3. Objetivo General

Desarrollar una Aplicación móvil, mediante el modelado vista controlador, para la inscripción de los participantes en la competencia de la Liga boliviana de robótica en el departamento de Tarija.

1.4. Objetivos Específicos

- Diseñar un interfaz amigable e interactiva para el módulo administrador y módulo tutor y módulo participante.
- Diseñar el registro de persona, registro de evento, registro de auspicios, registro de áreas de competencia, registro de anuncios y registro de ubicación del evento, reposte de participantes y reporte de cobros de participantes en el módulo administrador.
- Diseñar el registro de cobro, registro de participante y credencial en el módulo tutor.
- Diseñar el interfaz para el módulo participante con la visualización de las noticias, evento, áreas, auspicios, participantes inscritos, dirección y galería de fotos.
- Diseñar una base de datos no relacional específica para el almacenamiento de la información.
- Utilizar la metodología scrum para el desarrollo de la aplicación móvil.

1.5. Justificación

El desarrollo de la aplicación móvil es una herramienta tecnológica de gran utilidad y gran alcance, las empresas las utilizan como un canal de comunicación rápida, de esta manera los usuarios acceden a información desde cualquier lugar de manera efectiva con una mínima conexión a internet.

1.6. Alcances y Limitaciones

1.6.1. Alcances

- La aplicación móvil tendrá tres módulos, administrativo, tutor y participante el en Módulo participante visualizará la información del evento, el módulo del tutor registrará la inscripción del participante y los cobros y el Módulo Administrativo permitirá el registro de personas, evento, auspicios, áreas, reporte de participantes y cobros.
- La aplicación móvil contará con una base de datos para el almacenamiento de la información específica.
- El sistema móvil brindara reportes de la inscripción del participante y el registro del comprobante del cobro del participante.

- Se ingresará al sistema con su usuario y clave previamente registrado por el Administrador y el tutor de la Competencia de la liga boliviana de robótica de Tarija.

1.6.2. Limitaciones

- La aplicación móvil estará disponible para celulares que tengan Android 7.0 en adelante.
- La aplicación móvil será implementada solo en el idioma español.
- La Aplicación funcionará si el celular tiene acceso a internet.
- La aplicación contara con módulo administrativo y módulo tutor y módulo participante.
- La aplicación del módulo participante contará con acceso libre.

MARCO TEÓRICO

2. MARCO TEÓRICO

2.1. Aplicaciones Móviles

Las aplicaciones móviles son programas diseñados para ser ejecutados en teléfonos, tablets y otros dispositivos móviles, que permiten al usuario realizar actividades profesionales, acceder a servicios, mantenerse informado, entre otro universo de posibilidades.

Actualmente vivimos en la era tecnológica, específicamente con la aparición de dispositivos móviles como tablets, celulares y especialmente con los Smartphone, que con costos cada vez más accesibles, han permitido que estén al alcance de la mayoría de los usuarios, con ello las aplicaciones móviles también han sufrido un auge en cuanto a sus ofertas hacia sus usuarios, dándoles la libertad de tener una gran variedad de aplicaciones de todo ámbito y permitiendo el acceso a la información en cualquier momento y desde cualquier lugar del mundo. Hoy en día un gran número de compañías, sobre todo las del sector de ventas se ven impulsadas a tener o desarrollar su propia aplicación móvil para sacar ventaja ante la competencia y así llegar a muchos otros mercados y agilizar sus procesos. Los usuarios necesitan obtener información de su empresa en tiempo real y en cualquier parte. (nextu, 2015).

Una aplicación promueve el fortalecimiento de marca, tu empresa se destacará de la competencia al tener una aplicación móvil mediante la cual tus usuarios puedan interactuar contigo de una manera que no puedes conseguir mediante otros canales.

Mayor visibilidad: Una aplicación móvil, al estar hospedada en las principales tiendas de aplicaciones como Google Play Store y App Store de Apple, estará disponible para miles de usuarios interesados en tu producto¹.

Velocidad: una aplicación móvil funciona en tiempo real.

Usabilidad: Tu aplicación móvil podrá ser como tú quieras, con un diseño orientado a tener una gran experiencia de usuario y obtener los máximos resultados de venta para tu negocio.

¹ <https://appdesignbook.com/es/contenidos/las-aplicaciones/>

Disponibilidad offline: No todas las personas tienen acceso a Internet durante todo el día. Una app móvil permitirá que tus usuarios puedan navegar por el catálogo de tus productos.

2.2. Tipos de Aplicaciones

A nivel de programación, existen varias formas de desarrollar una aplicación. Cada una de ellas tiene diferentes características y limitaciones, especialmente desde el punto de vista técnico. Existen tres tipos de Apps la cual condicionara el diseño visual y la interacción. (appdesignbook, 2017)

2.2.1. Aplicaciones Nativas

Las aplicaciones nativas son aquellas que han sido desarrolladas con el software que ofrece cada sistema operativo, llamado genéricamente Software Development Kit o SDK. Android, iOS y Windows Phone tienen un sistema diferente y las aplicaciones nativas se diseñan y programan específicamente para cada plataforma, en el lenguaje utilizado por el SDK, cuando hablamos de desarrollo móvil casi siempre nos estamos refiriendo a aplicaciones nativas. Por ejemplo:

- Las apps para iOS se desarrollan con lenguaje Objective-C.
- Las apps para Android se desarrollan con lenguaje Java.
- Las apps en Windows Phone se desarrollan en .Net.

Las aplicaciones nativas se actualizan continuamente y el usuario debe volver a descargarlas para obtener la última versión, que a veces corrige errores o añade mejoras. Una característica generalmente menospreciada es que pueden hacer uso de las notificaciones del S.O. para mostrar avisos importantes al usuario, aun cuando no se esté usando la aplicación, como los mensajes de WhatsApp².

² <https://www.solbyte.com/blog/2014/07/21/tipos-de-aplicaciones-moviles-nativas-webs-hibridas/>

Tabla 2. 1

Comparación entre ventajas e inconvenientes de las aplicaciones nativas.

Ventajas	Inconvenientes
<ul style="list-style-type: none">• Acceso completo al dispositivo.• Mejor experiencia del usuario.• Visibilidad en APP Store.• Envío de notificaciones o avisos a los usuarios.• La actualización de las apps es constante.	<ul style="list-style-type: none">• Diferentes habilidades/ idiomas/ herramientas para cada plataforma de destino.• Tienden a ser más caras de desarrollar.• El código del cliente no es reutilizable entre las diferentes plataformas.

2.2.2 Aplicaciones Web App

Esta desarrollada con lenguajes muy conocidos por los programadores, como es el HTML, Javascript y CSS. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones.

Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo a través de una URL. Por ejemplo, en Safari, si se trata de la plataforma iOS. El contenido se adapta a la pantalla adquiriendo un aspecto de navegación APP. (eumed.net, 2016).

¿Puede considerarse esto una APP? En realidad, la gran diferencia con una aplicación nativa (además de los inconvenientes que se muestran en la tabla) es que no necesita instalación por lo que no pueden estar visibles en app store y la promoción y comercialización debe realizarse de forma independiente. De todas formas, se puede crear un acceso directo que sería como “instalar” la aplicación en el dispositivo.

Las apps web móviles son siempre una buena opción si nuestro objetivo es adaptar la web a formato móvil.

Tabla 2. 2

Comparación entre ventajas e inconvenientes de las aplicaciones web.

Ventajas	Inconvenientes
<ul style="list-style-type: none">• El mismo código base reutilizable en múltiples plataformas.• Proceso de desarrollo más sencillo y económico.• No necesitan ninguna aprobación externa para publicarse.• El usuario siempre dispone de la última versión.	<ul style="list-style-type: none">• Requiere conexión a internet.• Acceso muy limitado a los elementos y características del hardware del dispositivo.• La experiencia del usuario (navegación, interacción) y el tiempo de respuesta es menor que una app nativa.• Requiere de mayor esfuerzo en promoción y visibilidad.

2.2.3. Aplicaciones Híbridas

Este tipo de aplicaciones es una combinación entre las dos anteriores. La forma de desarrollarlas es parecida a la de una aplicación web, usando HTML, CSS y JavaScript, y una vez que la aplicación está terminada, se compila o empaqueta de tal forma, que el resultado final es como si se tratara de una aplicación nativa. Por aquello también son denominadas Web App nativa. Esto permite agrupar los códigos, obtener diferentes aplicaciones y permite su uso en cada plataforma y distribuirlas en cada una de sus tiendas.

A diferencia de las aplicaciones web, éstas permiten acceder, usando librerías, a las capacidades del teléfono, tal como lo haría una app nativa.

Tabla 2. 3

Comparación entre ventajas e inconvenientes de las aplicaciones híbridas.

Ventajas	Inconvenientes
<ul style="list-style-type: none">• Es posible distribuirla en las tiendas de IOS y Android.• Instalación nativa pero construida con JavaScript, HTML y CSS.• El mismo código base para múltiples plataformas.• Acceso a parte del hardware del dispositivo.	<ul style="list-style-type: none">• Experiencia del usuario más propia de la aplicación web que de la app nativa.• Diseño visual no siempre relacionado con el sistema operativo en el que se muestre.

2.3. Sistemas Operativos para Móviles

Un sistema operativo (SO) móvil controla un dispositivo móvil al igual que los PC utilizan Windows o Linux entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos El Sistema Operativo (SO) móvil de un teléfono o tableta realiza la interacción real con lo que podemos hacer a partir de las capacidades del hardware que conforman un equipo. A manera de traductor, esta plataforma interpreta lo que el usuario quiere que la terminal realice y cada vez, lo ejecuta con mayor inteligencia. (www.lancetalent.com, 2016) Una de las cualidades más atractivas de un sistema operativo móvil es la rapidez con la que en general se desempeña los sistemas operativos con los que funcionan adquieren mayor importancia. La cuota de mercado de sistemas operativos móviles en el segundo trimestre de 2017 era el siguiente:

- Android 84,7%.
- iOS 11,7%.
- Windows Phone 2,5%.
- BlackBerry OS 0,5%.
- Otros 0,6%.

2.4. Android

Es un sistema operativo móvil desarrollado por Google, basado en el Kernel de Linux y otros softwares de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles y televisores.³

Es así como nace Android. Android es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, netbooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, widgets, o incluso, modificar el propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser muy fácil comenzar a programar en esta plataforma. (Báez, 2016).

Cada una de las versiones de Android incorpora cambios, mejoras y novedades, haciendo que a cada nuevo postre que llega al mercado se incorpore alguna particularidad destinada a mejorar las posibilidades de los usuarios. Los dispositivos con Android se centran en el ahorro de energía, algo para lo cual la plataforma está diseñada siempre teniendo el objetivo de hacer que la memoria RAM y el consumo de energía se encuentren al mínimo. Esto provoca que la inteligencia del propio sistema se tome por su mano la decisión de suspender las aplicaciones que están abiertas y que no han sido manualmente cerradas por el usuario.

Por otro lado, al ser un sistema abierto esto ha ayudado mucho a su expansión entre las distintas empresas, tanto famosas como secundarias, que ven una opción ideal para poder llegar a todo tipo de público partiendo de una plataforma que se sabe que funciona de forma más que correcta.

2.5. Node.js

Es un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para construir aplicaciones en red y que sean escalables.

³ <https://www.xatakandroid.com/sistema-operativo/que-es-android>

Node.js es un entorno Javascript del lado del servidor, basado en eventos. Node.js ejecuta JavaScript utilizando el motor V8, desarrollado por Google para uso de su navegador Chrome. Aprovechando el motor V8 permite a Node.js proporciona un entorno de ejecución del lado del servidor que compila y ejecuta JavaScript a velocidades increíbles. El aumento de velocidad es importante debido a que V8 compila JavaScript en código de máquina nativo, en lugar de interpretarlo o ejecutarlo como bytecode (netconsulting, 2019).

Node.js funciona con un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas asíncronas las cuales pueden ejecutarse concurrentemente en un número de hasta cientos de miles sin incurrir en costos asociados al cambio de contexto. Este diseño de compartir un único hilo de ejecución entre todas las solicitudes atiende a necesidades de aplicaciones altamente concurrentes, en el que toda operación que realice entradas y salidas debe tener una función callback. Un inconveniente de este enfoque de único hilo de ejecución, es que Node.js requiere de módulos adicionales como cluster⁹ para escalar la aplicación con el número de núcleos de procesamiento de la máquina en la que se ejecuta.

2.6. Ionic

Ionic es un SDK completo de código abierto para el desarrollo de aplicaciones móviles híbridas creado por Max Lynch, Ben Sperry y Adam Bradley de Drifty Co.

Es un framework que se está haciendo muy popular últimamente. Es una herramienta que los programadores pueden utilizar totalmente gratis, para desarrollar apps basadas en HTML5, CSS y JavaScript. Está construido con Sass y optimizado para AngularJS. Además, es libre y de código abierto, open source (udemy, 2019).

2.7. Angular

AngularJS (comúnmente llamado Angular.js o AngularJS 1), es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles. (Azaustre, s.f.).

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada

o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o ser recuperados de los recursos JSON estáticos o dinámicos.

AngularJS se puede combinar con el entorno en tiempo de ejecución Node.js, el framework para servidor Express.js y la base de datos MongoDB para formar el conjunto MEAN. AngularJS está construido en torno a la creencia de que la programación declarativa es la que debe utilizarse para generar interfaces de usuario y enlazar componentes de software. Este framework adapta y amplía el HTML tradicional para servir mejor contenido dinámico a través de un data binding bidireccional que permite la sincronización automática de modelos y vistas. Como resultado. (Azaustre, s.f.).

Angular sigue el patrón MVVM (Model View View-Model) de ingeniería de software y alienta la articulación flexible entre la presentación, datos y componentes lógicos. Con el uso de la inyección de dependencias, Angular lleva servicios tradicionales del lado del servidor, tales como controladores dependientes de la vista, a las aplicaciones web del lado del cliente. En consecuencia, gran parte de la carga en el backend se reduce, lo que conlleva a aplicaciones web mucho más ligeras. (Azaustre, s.f.).

2.8. TypeScript

Es un lenguaje de programación de código abierto (de uso libre y gratuito) desarrollado por Microsoft. Se dice que es un superconjunto de JavaScript, lo que viene a significar que es un lenguaje “montado” sobre otro lenguaje.

Digamos que el propósito de TypeScript es cubrir las carencias o deficiencias del lenguaje JavaScript, haciéndolo un lenguaje más completo, escalable, seguro e incluso más fácil de utilizar. Tiene un gran parecido al nuevo estándar de JavaScript, ECMAScript6, el cual en la actualidad aún no está completamente implementado, así que de cierta manera TypeScript nos acerca al futuro de JavaScript.

Sus principales características frente al JavaScript es el tipado estático (o sea, declaración del tipo de dato que debe alojar cada variable) y la programación orientada a objetos basado en clases, lo cual, aunque de alguna manera era posible con JavaScript, resultaba complicado ya que el lenguaje no está pensado para ello.

Aunque Angular no te obliga a usar TypeScript, el equipo del core de Angular sí que lo ha adoptado y en la documentación sugiere usar TypeScript por defecto. Esto implica que los ejemplos relacionados y los proyectos de código abierto parezcan más familiares y consistentes. Angular ya ofrece ejemplos claros que enseñan cómo usar el compilador TypeScript (tutorialesenpdf, 2016).

2.9. Cordova

Apache Cordova es un marco de desarrollo móvil de código abierto. Permite utilizar las tecnologías estándar web como HTML5, CSS3 y JavaScript para desarrollo multiplataforma, evitando el lenguaje de desarrollo nativo cada plataforma móvil. Aplicaciones ejecutan dentro de envolturas para cada plataforma y dependen de enlaces estándares API para acceder a los componentes de cada dispositivo, sensores, datos y estado de la red (pdfmanuales, 2017).

La tecnología PhoneGap o Apache Cordova permite compilar un código Web (HTML, CSS, JavaScript etc.) como si fuera una aplicación nativa para móvil. Es decir, con un solo desarrollo web es posible obtener aplicaciones nativas para diferentes plataformas destino como Android, iOS, Windows Phone, etc.

2.10. Chrome

Es un navegador web rápido, seguro y gratuito, diseñado para la Web actual.

2.11. Proceso de Diseño y Desarrollo de una App

Según (Cuello & Vittone, 2013) El proceso de diseño y desarrollo de apps, abarca desde la concepción de la idea hasta el análisis posterior a su publicación en las tiendas. Durante las diferentes etapas, diseñadores y desarrolladores trabajan la mayor parte del tiempo de manera simultánea y coordinada. (nextu, 2015).

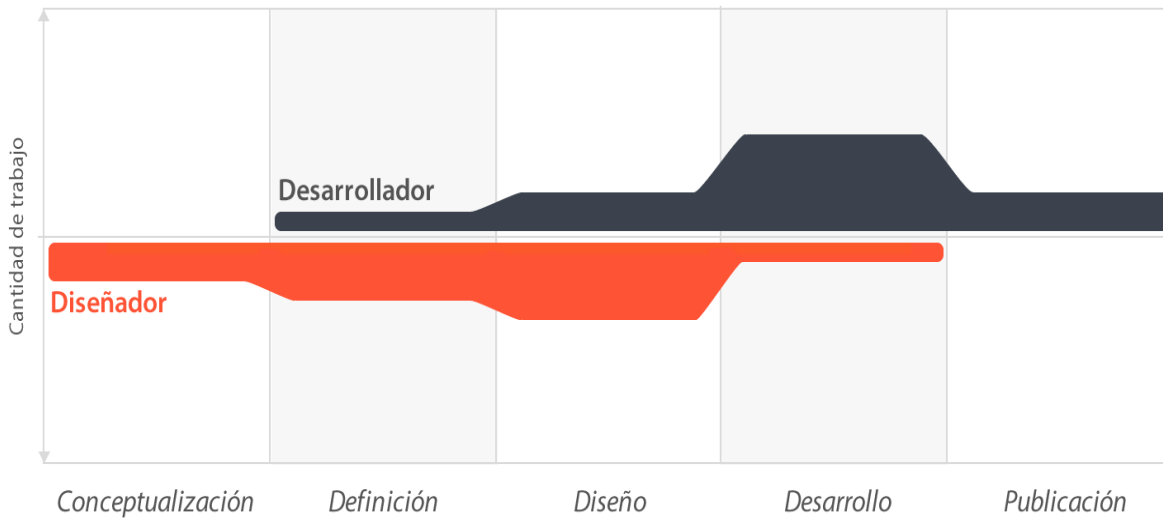


Figura 2.1. Proceso de Diseño y Desarrollo de una Aplicación.

2.11.1. Conceptualización

El resultado de esta etapa es una idea de aplicación, que tiene en cuenta las necesidades y problemas de los usuarios. La idea responde a una investigación preliminar y a la posterior comprobación de la viabilidad del concepto. En este proceso se da paso a la idea, investigación y formalización de la misma.

2.11.2. Definición

En este paso del proceso se describe con detalle a los usuarios para quienes se diseñará la aplicación, usando metodologías como «Personas» y «Viaje del usuario». También aquí se sientan las bases de la funcionalidad, lo cual determinará el alcance del proyecto y la complejidad de diseño y programación de la app. En esta etapa se define usuarios y requerimiento funcional.

2.11.3. Diseño

En la etapa de diseño se llevan a un plano tangible los conceptos y definiciones anteriores, primero en forma de wireframes, que permiten crear los primeros prototipos para ser probados con usuarios, y posteriormente, en un diseño visual acabado que será provisto al

desarrollador, en forma de archivos separados y pantallas modelo, para la programación del código. En síntesis, se crean los prototipos de pantalla, se crea un diseño visual para realizar un test con los usuarios.

2.11.4. Desarrollo

El programador se encarga de dar vida a los diseños y crear la estructura sobre la cual se apoyará el funcionamiento de la aplicación. Una vez que existe la versión inicial, dedica gran parte del tiempo a corregir errores funcionales para asegurar el correcto desempeño de la app y la prepara para su aprobación en las tiendas.

2.11.5. Publicación

La aplicación finalmente es puesta a disposición de los usuarios en las tiendas. Luego de este paso importante se realiza un seguimiento a través de analíticas, estadísticas y comentarios de usuarios, para evaluar el comportamiento y desempeño de la app, corregir errores, realizar mejoras y actualizarla en futuras versiones. (appdesignbook, 2017).

2.12. Firebase

Firebase es un conjunto de herramientas orientadas a la creación de aplicaciones de alta calidad, es la nueva y mejorada plataforma de desarrollo móvil en la nube de Google. Se trata de una plataforma disponible para diferentes plataformas (IOS, Android, aplicaciones web, Unity y C++). Con la base en tiempo real de Firebase podrás guardar todos los datos que requiera tu aplicación. Se lleva muy bien con React (Una biblioteca de JavaScript) y su patrón reactivo que permite actualizar los datos en los componentes automáticamente. (Zamora, 2016).

En conclusión, es una plataforma en la nube disponible para todos.

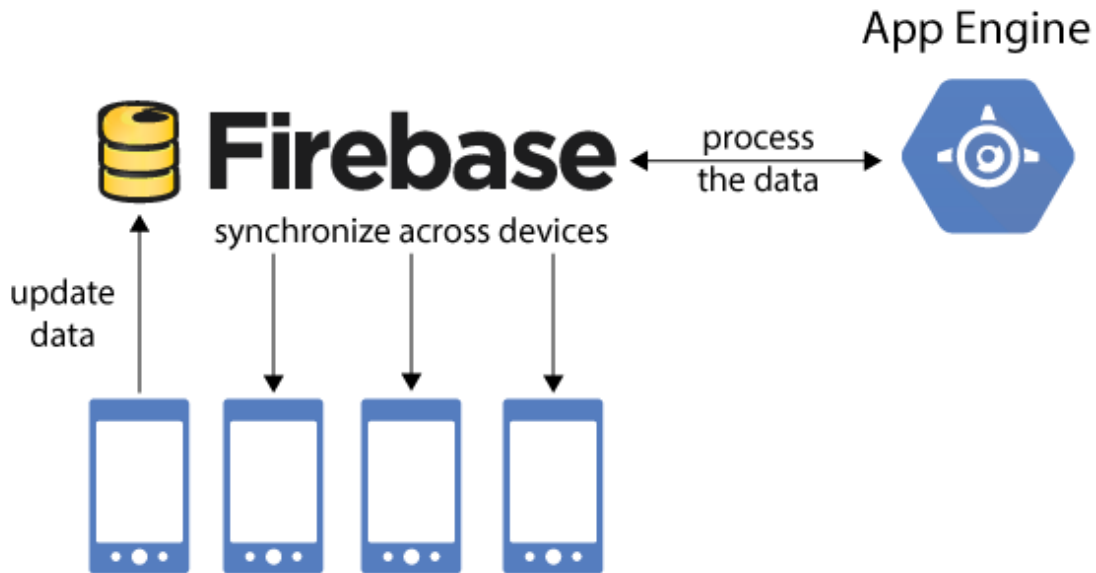


Figura 2.2. Diagrama del Flujo de Información Mediante Firebase.

Sus características fundamentales están divididas en varios grupos, las cuales podemos agrupar en:

- a) **Analíticas:** Provee una solución gratuita para tener todo tipo de medidas (hasta 500 tipos de eventos), para gestionarlo todo desde un único panel.
- b) **Desarrollo:** Permite construir mejores apps, permitiendo delegar determinadas operaciones en Firebase, para poder ahorrar tiempo, evitar bugs y obtener un aceptable nivel de calidad. Entre sus características destacan el almacenamiento, testeo, configuración remota, mensajería en la nube o autenticación, entre otras.
- c) **Crecimiento:** Permite gestionar los usuarios de las aplicaciones, pudiendo además captar nuevos. Para ello dispondremos de funcionalidades como las de invitaciones, indexación o notificaciones.

2.13. Metodología Ágil

Así como existen métodos de gestión de proyectos tradicionales, como el propuesto por el Project Management Institute, más conocido como PMI podemos encontrarnos con una rama diferente en la gestión de proyectos, conocida como Agile. El desarrollo ágil de software, no es más que una metodología de gestión de proyectos adaptativa, que permite llevar a cabo,

proyectos de desarrollo de software, adaptándose a los cambios y evolucionando en forma conjunta con el software. (bahit, 2014).

Las metodologías ágiles, solapan las etapas de trabajo, permitiendo ahorrar tiempo, evitando la dependencia (cada etapa es independiente de la otra) y haciendo del ciclo de vida, un proceso iterativo (se inicia con el relevamiento, se finaliza con la implementación y se vuelve a comenzar para abordar nuevas funcionalidades).



Figura 2.3 Procesos de la Metodología Ágil de Desarrollo.

2.14. Metodología Scrum

“Scrum es un marco de trabajo interactivo para el desarrollo de proyectos, productos y aplicaciones. Mediante ciclos de trabajo llamados Sprints. Son iteraciones de 1 a 4 semanas, y se van sucediendo una detrás de otra. Los Sprints son de duración fija, terminan en una fecha específica, aunque no se haya terminado el trabajo, y nunca se alargan. Se limitan en tiempo. Al comienzo de cada Sprint, un equipo multifuncional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos. (bahit, 2014).

El objetivo principal del sprint es transformar un conjunto de ítems requeridos por el cliente en un incremento de funcionalidad 100% operativa para el software.

En un sentido amplio, el marco de trabajo de Scrum se compone de una serie de reglas, que definen roles que integran los equipos, artefactos necesarios para los procesos, bloques de tiempo preestablecidos y ceremonias que deben respetarse⁴.

Los roles que se caracterizan en la metodología Scrum, son:

- El Dueño de Producto (Product Owner), responsable de maximizar el valor del producto.
- El Scrum Master, responsable de asegurar los procesos.
- El Equipo, responsable de realizar el trabajo.

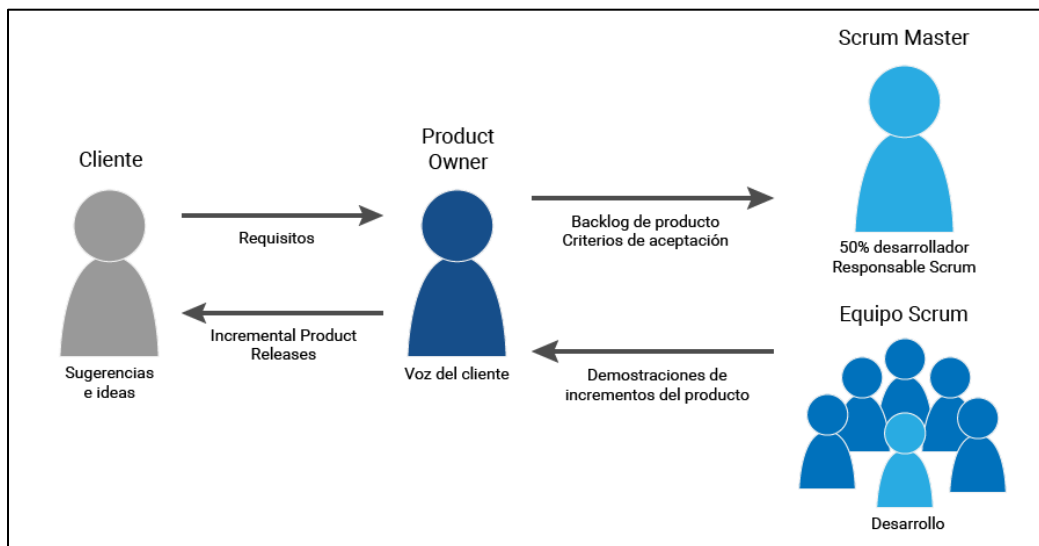


Figura 2.4. Roles de la Metodología Scrum.

2.14.1. Eventos

En el marco metodológico de Scrum, los eventos son las etapas caracterizadas por reuniones, que permiten definir y orientar oportunamente los objetivos propuestos para cada Sprint.

En Scrum existen 5 eventos claramente identificados y aplicados, estos son: el Sprint, reunión de planificación (Sprint Planning), Scrum diario (Daily Scrum), reunión de revisión del Sprint (Sprint Review) y reunión de retrospectiva del Sprint (Sprint Retrospective).

⁴ <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum#:~:text=Entre%20las%20diferentes%20metodologías%20de,donde%20la%20incertidumbre%20sea%20elevada.>

En la actualidad se encuentran en varias publicaciones un sexto evento denominado reunión de refinamiento del Sprint, que aporta a solucionar detalles pendientes y muy específicos del producto, pero que se lo menciona como opcional dependiendo de la complejidad del proyecto.

2.14.1.1. Sprint

El Sprint se entiende como un evento determinado por un intervalo de tiempo para generar una serie de iteraciones con tiempo acotado, que garantice poder realizar las entregas periódicas y rápidas de un producto. Es un rango de tiempo definido (no mayor a 1 mes) en el cual se desarrollará un incremento del entregable funcional. Los Sprints trabajan de manera secuencial es decir para empezar el siguiente sprint hay que terminar el anterior.

2.14.1.2 Reunión de Planificación del Sprint

Se trata de una reunión de trabajo, que se realiza de manera previa al inicio de un Sprint. En esta reunión se determina cual será el objetivo del Sprint y todas las tareas necesarias para conseguirlo. La reunión de planificación del Sprint se sugiere limitarla a 8 horas de trabajo.

2.14.1.3. Scrum Diario

Es una reunión matutina (no mayor a 15 minutos) que se da en equipo y su objetivo es establecer las metas diarias de desarrollo y mostrando los avances logrados con respecto a las metas del Scrum diario anterior.

Los ítems a tratar en cada Scrum diario suelen ser:

- ¿Qué hicimos ayer para lograr el objetivo del sprint?
- ¿Qué haremos hoy para lograr el objetivo del sprint?
- ¿Cuáles son los problemas que impiden que logremos el objetivo del sprint?

El equipo utiliza el Scrum diario para evaluar el avance hacia el objetivo del sprint. Permite optimizar las posibilidades de cumplimiento del sprint. El Scrum master es el encargado de asegurar la realización de los Scrum diarios, pero son los miembros del equipo quienes dirigen la reunión.

2.14.1.4. Revisión del Sprint

Se trata de una reunión para realizar un análisis e inspección del incremento generado y una adaptación de las características del producto en caso de ser necesario. Para esta actividad se recomienda un tiempo de 4 horas de trabajo, aplicadas a un Sprint de un mes de duración.

2.14.1.5. Retrospectiva del Sprint

La retrospectiva del Sprint consiste en una reunión en la que el equipo, analiza todos los aspectos operativos de la forma de trabajo, con el propósito de crear un plan de mejoras para ser aplicadas en el próximo Sprint. Se sugiere para esta actividad aplicar un margen de 3 horas de trabajo, aplicadas a un Sprint de un mes de duración.

2.14.1.6. Refinamiento del Sprint

La reunión para refinamiento del Sprint, no está específicamente comprendida entre los eventos obligatorios del marco de trabajo de Scrum, sin embargo, su utilidad es necesaria en muchas ocasiones. El refinamiento del Sprint se trata de una reunión, que permita aclarar y afinar ciertas historias de usuario que por diversas razones pudieron quedar pendientes, durante la reunión de planificación del Sprint.

2.14.2. Artefactos

Los artefactos son todos los elementos que garantizan la transparencia y el registro de la información fundamental del proceso Scrum.

2.14.2.1. Product Backlog

Es un listado de todo lo necesario para el desarrollo del proyecto, incluye los requerimientos, características, funcionalidades, cambios, etc. Es responsabilidad del Product Owner su mantenimiento y actualización.

2.14.2.2. Sprint Backlog

Son un conjunto de pendientes tomados del product backlog, incluyendo un plan para el desarrollo y el objetivo del sprint. Es un estimado elaborado por el equipo de desarrollo que indica que funcionalidad estará aplicada para el siguiente incremento y que será lo necesario para poder terminarlo.

2.14.2.3. Incremento

El incremento es el resultado obtenido del Sprint, se trata de un producto de software entregable, utilizable y potencialmente desplegable. Es importante que el incremento debe estar terminado de acuerdo al Sprint.

2.14.3. Roles

2.14.3.1. Scrum Master

El Scrum Master representa la persona encargada de liderar el equipo de desarrollo, haciendo cumplir las reglas y procesos de la metodología, actúa también como entrenador (coach) para los nuevos miembros del equipo.

2.14.3.2. Equipo de Desarrollo (Team)

El equipo de desarrollo es un grupo multidisciplinario de profesionales que trabajaran en pro del desarrollo de un proyecto de software, cumpliendo con las historias que se han comprometido a desarrollar al comenzar cada sprint.

2.14.3.3. Product Owner

Es el Stakeholder designado a trabajar con el equipo el cual proporcionara los requerimientos, aprueba los objetivos del sprint y permite maximizar el valor del producto final.

2.15. Faces de Scrum

Cada fase describe cada proceso en detalle, incluyendo sus entradas, herramientas y salidas asociadas. En cada proceso, algunas entradas, herramientas y salidas son obligatorias, y existen otras que son opcionales, cuyo uso dependerá de la naturaleza del proyecto.

2.15.1. Iniciación

En esta fase se crea la Visión del Proyecto que sirve de enfoque y dirección del mismo. Se crean e identifican roles claves del proyecto como el Scrum Master, Product Owner, interesados, equipo del proyecto. Así mismo, se define la lista de prioridades o el Product Backlog la cual sirve de base para la elaboración del plan de lanzamiento y tamaño de cada Sprint.

2.15.2. Planificación y Estimación

Aquí se definen y aterrizan en los Sprints las historias de usuarios, se alinean a todo lo que genera valor a la organización y se hacen las estimaciones de tiempo y esfuerzo para cumplirlas, los cuales se traducen en listas de tareas cuyos tiempos de desarrollo se definen en reuniones de equipo correspondientes, así como el proceso de definición del Sprint Backlog que contiene todas las tareas que deben completarse en el Sprint.

2.15.3. Implementación

En esta fase se trabaja en las tareas del Sprint Backlog para crear Sprint Delivery, para ello se utiliza a menudo un Scrumboard para realizar el seguimiento del trabajo y de actividades que se llevan a cabo. También, los inconvenientes o problemas que enfrenta el Equipo Scrum se actualizan en un Impediment Log. Igualmente se actualiza o revisa la lista de prioridades de pendientes del producto.

2.15.4. Revisión y Retrospectiva

La etapa de revisión y retrospectiva está orientada principalmente para proyectos grandes que involucran varios equipos de Scrum. Lo que se busca en esta etapa es la realización de reuniones que permitan juntar a los equipos para discutir y revisar avances, dependencias e impedimentos en el desarrollo del proyecto. Es en esta etapa en la que el Equipo Scrum le demuestra el Sprint Delivery al propietario del producto y socios relevantes. La reunión permite discutir las lecciones aprendidas en el Sprint y que pueden ser aplicadas a futuro.

2.15.5. Lanzamiento

El lanzamiento se trata de la fase más esperada por los interesados o socios del proyecto, así como del Scrum Master y Equipo Scrum. En esta fase se desarrolla el proceso donde los Accept Delivery se les entregan a los Socios relevantes. Un acuerdo formal llamado Working Delivery Agreement documenta la finalización con éxito del Sprint. Del mismo modo, se realizan actividades de retrospectiva que permite identificar mejoras y lecciones aprendidas del proyecto.

2.16. UML

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común, con una semántica muy práctica y comprensible para la arquitectura, el diseño

y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, por ejemplo, en el flujo de procesos en la fabricación.⁵

El Lenguaje Unificado de Modelado, es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

A continuación, se describirán los diagramas más comunes del UML y los conceptos que representan:

- Diagrama de Casos de Uso.
- Diagrama de Actividades.
- Diagrama de Secuencias.

2.16.1. Diagrama de Casos de Uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

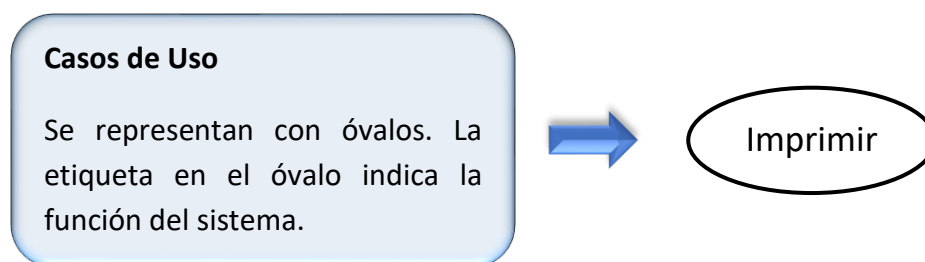


Figura 2.5. Representación de Casos de Uso.

⁵ <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

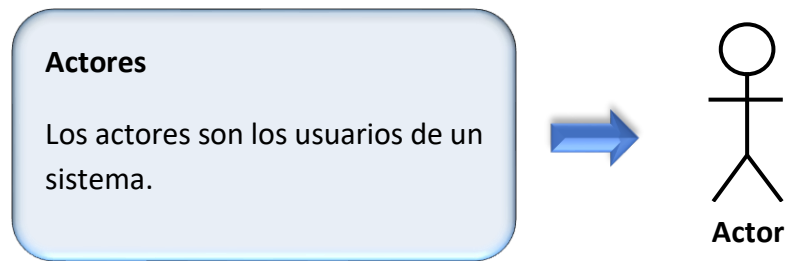


Figura 2.6. Representación de los Actores o Usuarios.

2.16.2. Diagrama de Actividad

Un diagrama de actividades ilustra la naturaleza dinámica de un sistema mediante el modelado del flujo ocurrente de actividad en actividad. Una actividad representa una operación en alguna clase del sistema y que resulta en un cambio en el estado del sistema. Típicamente, los diagramas de actividad son utilizados para modelar el flujo de trabajo interno de una operación.

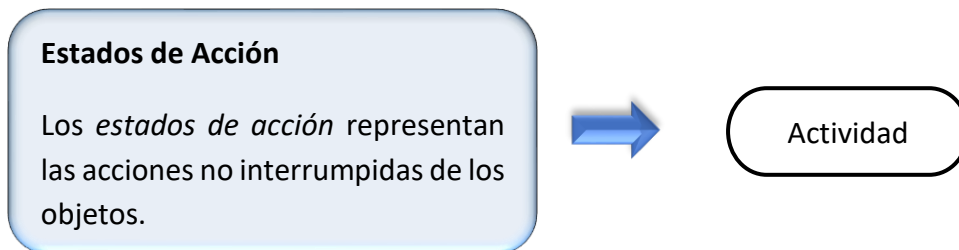


Figura 2.7. Representación de Estados de Acción

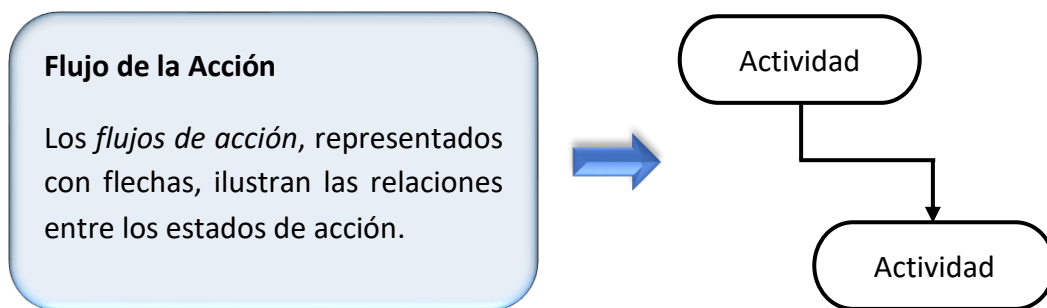


Figura 2.8. Representación de Flujo de la Acción

Elementos principales en los diagramas de actividad son las representaciones de: estado inicial, estado final, estado de acción, flujo de acción y bifurcación. A partir de la complejidad del sistema que se quiera modelar, existen otros recursos gráficos para la especificación requerida.

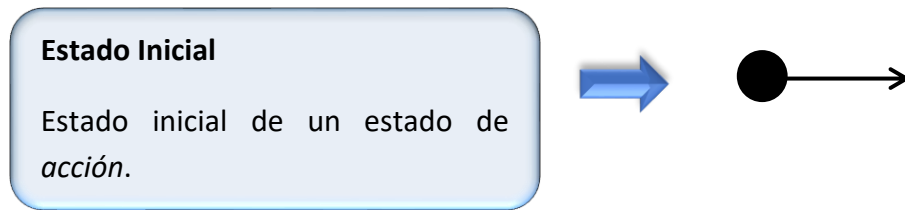


Figura 2.9. Representación de Estado Inicial

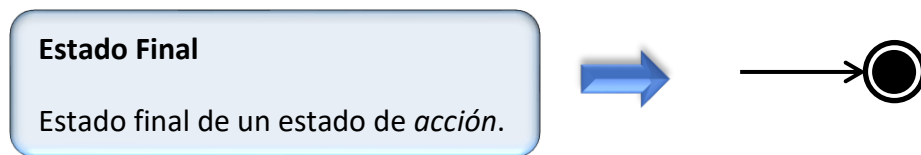


Figura 2.10. Representación de Estado Final.

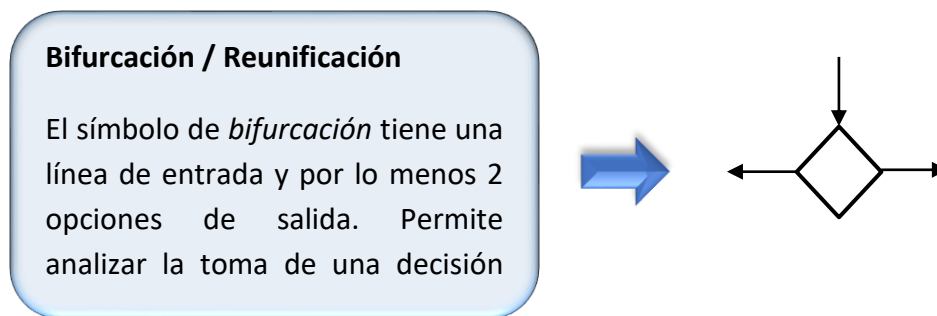


Figura 2.11. Representación de Bifurcación.

2.16.3. Diagrama de Secuencia

El diagrama de secuencia es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos. Existe una variedad de recursos gráficos para el modelado del diagrama de secuencia, los más frecuentes son: clases, activación, interfaz de selección, objeto de control, entidad, mensaje sincrónico, mensaje de flujo de rutina y nota.

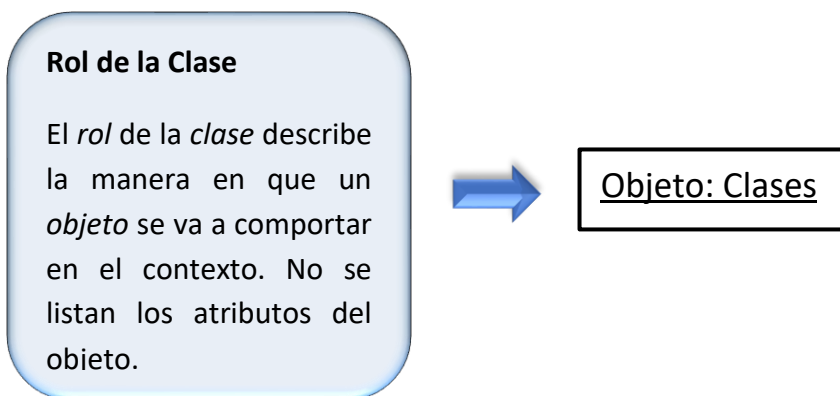


Figura 2.12. Representación Rol de la Clase.

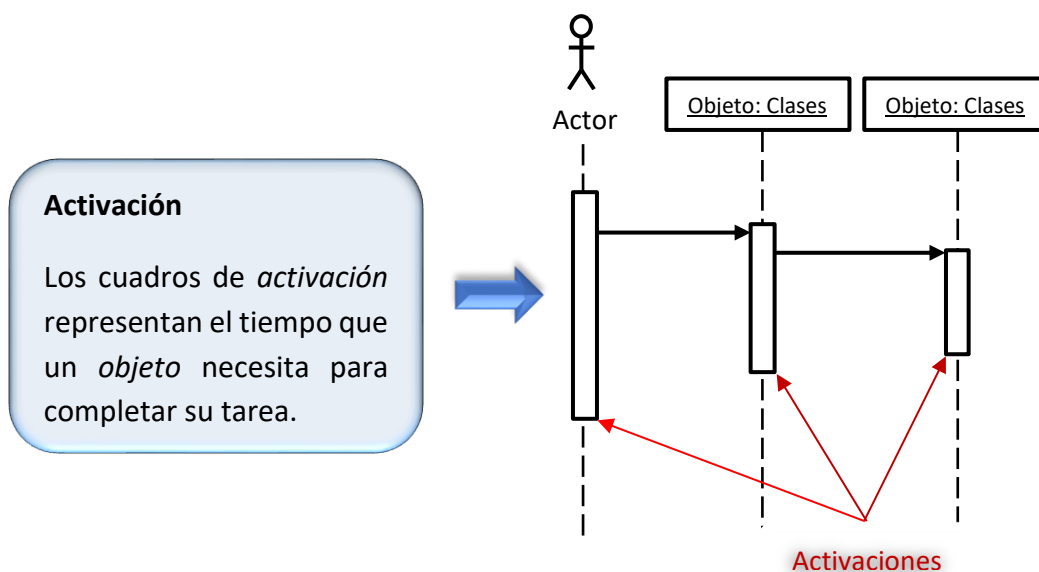


Figura 2.13. Representación de Activación.

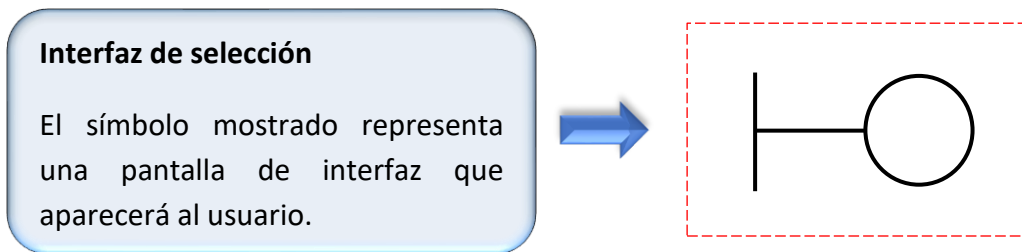


Figura 2.14. Representación Interfaz de Selección.

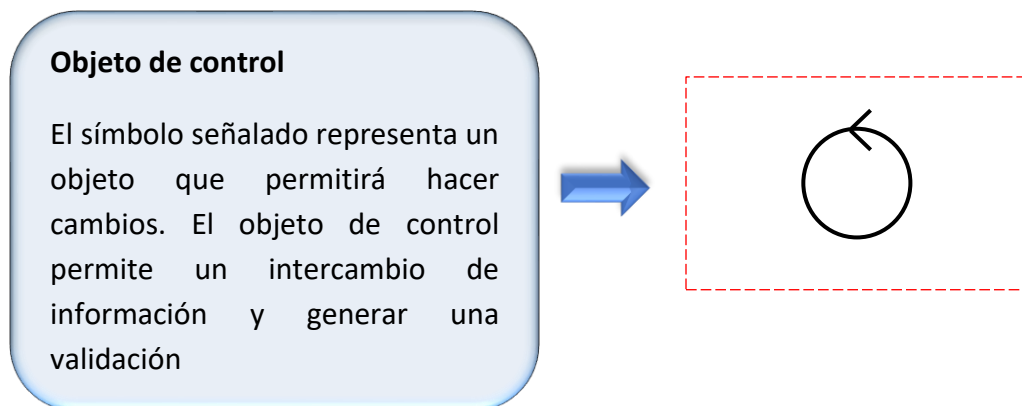


Figura 2.15. Representación Objeto de Control.

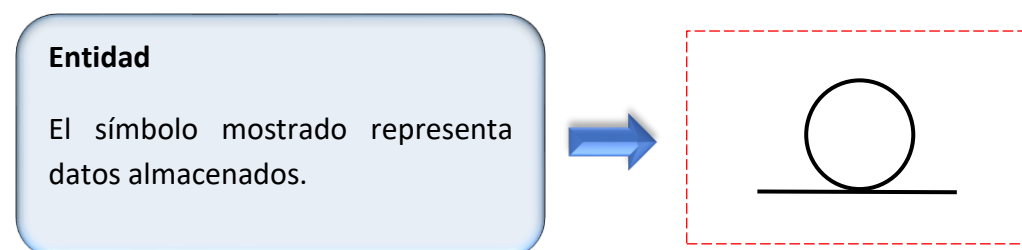


Figura 2.16. Representación de Entidad.

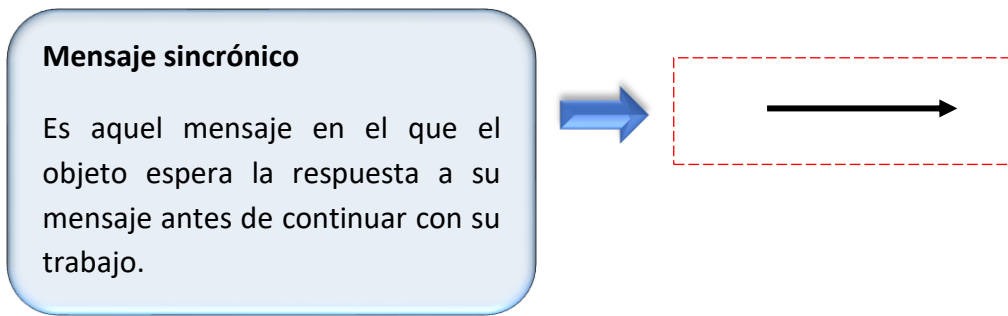


Figura 2.17. Representación de Mensaje Sincrónico.

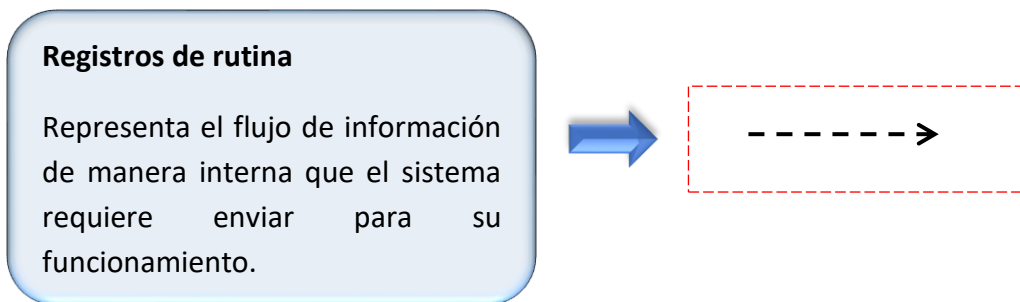


Figura 2.18. Representación Registros de Rutina.

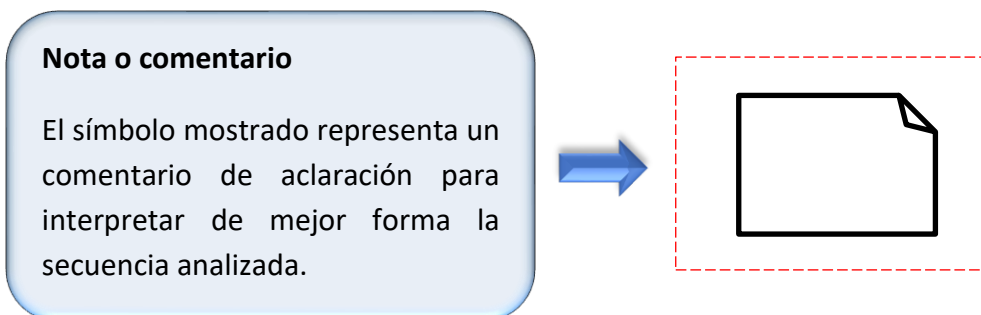


Figura 2.19. Representación de Nota.

ANÁLISIS

3. ANÁLISIS

A continuación, se presentan los casos de uso analizados para el desarrollo de la aplicación que permitirá facilitar el proceso de inscripción para las competencias de robótica de la Liga Boliviana de Robótica de la ciudad de Tarija.

3.1. Caso de Uso Rol Administrador

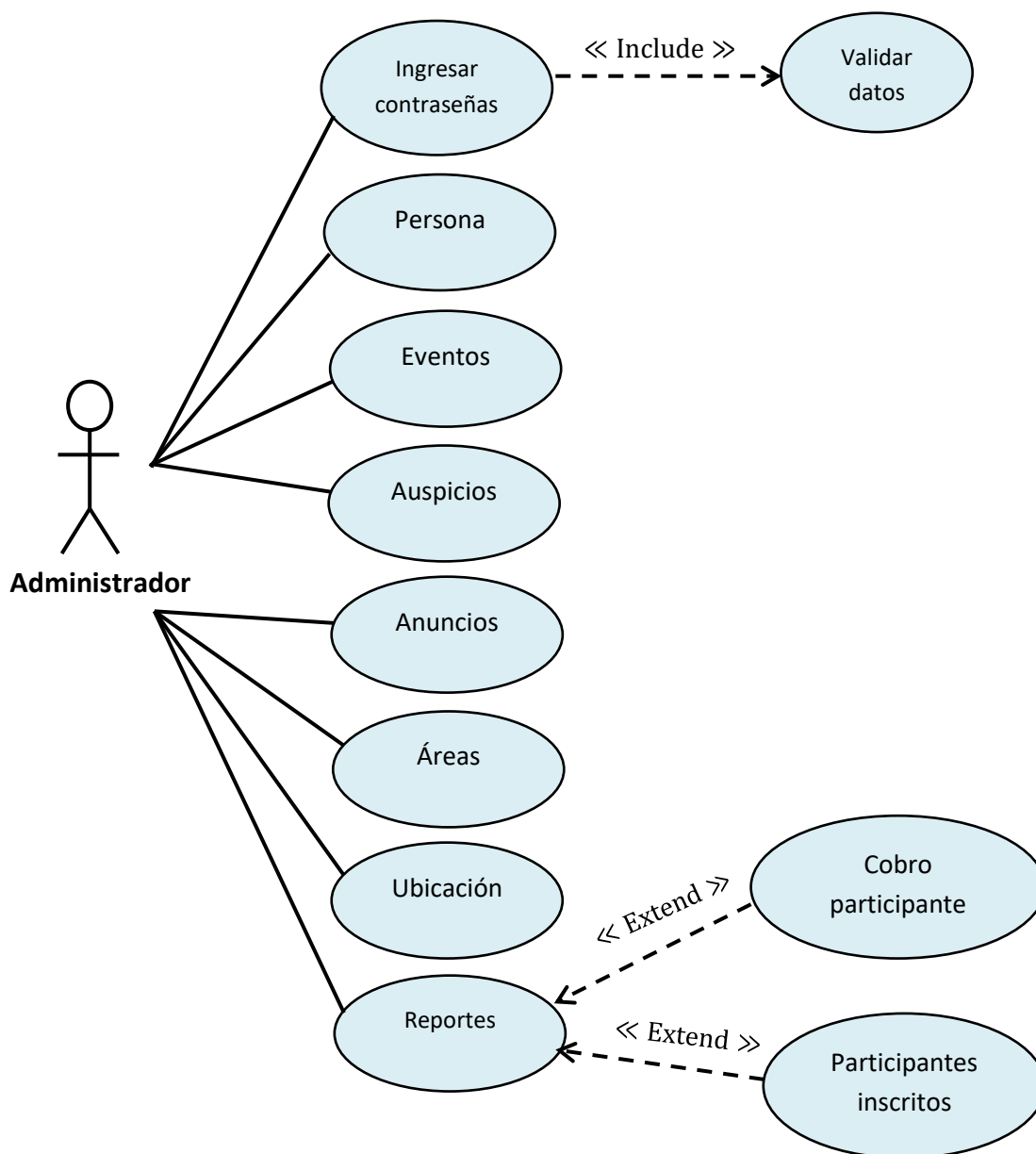


Figura 3.1. Caso de Uso Rol Administrador.

3.2. Caso de Uso Rol Tutor

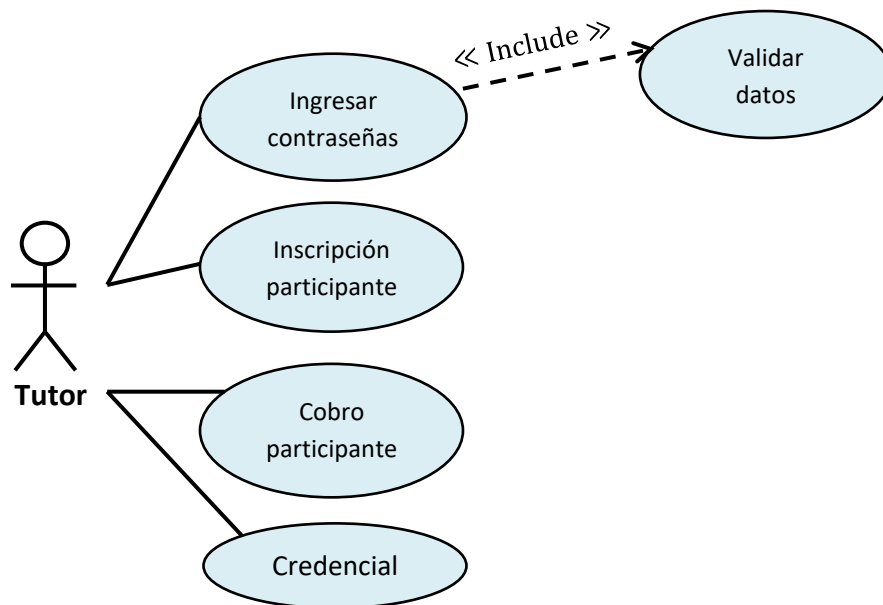


Figura 3.2. Caso de Uso Rol Tutor.

3.3. Caso de Uso Rol Participante

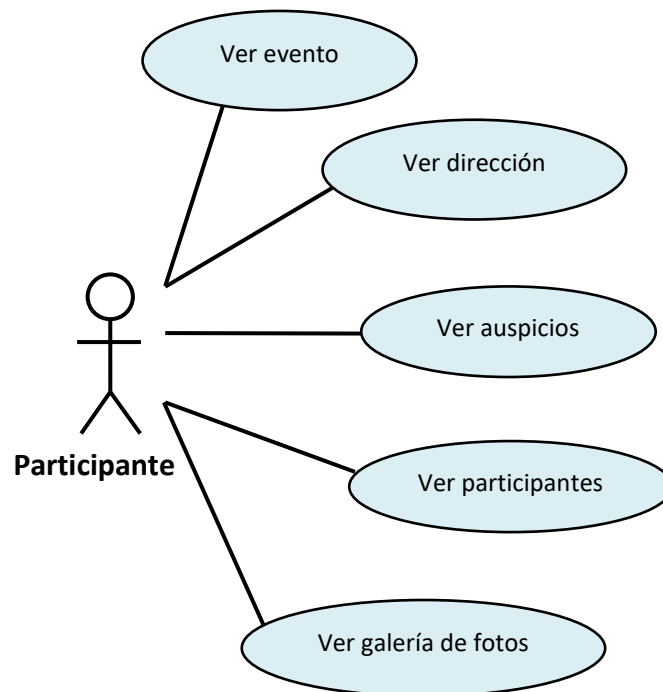


Figura 3.3. Caso de Uso Rol Participante.

3.4. Sprints

La creación de los Sprints permite planificar el margen de tiempo y las actividades detalladas de acuerdo a los procesos distinguidos en los casos de uso. Para el presente proyecto se elaboraron los siguientes Sprints para cada uno de los módulos que componen la aplicación propuesta.

3.4.1. Sprint para el Rol Administrador

Tabla 3. 1

Sprint para el Rol Administrador

N°	Casos de Uso	Tareas	Días	Horas
Sprint 1	Ingresar contraseña	Diseño de interfaz para el logueo. Programar la acción de logueo.	5	40
	Validar datos	Programar la validación de datos.	2	16
Sprint 2	Persona	Diseñar Pantallas de listado de persona, adicionar modificar y borrar personas. Programar las acciones de listado, nuevo, modificado y borrado.	10	80
Sprint 3	Eventos	Diseñar Pantallas de listado de eventos, adicionar modificar y borrar eventos. Programar las acciones de listado, nuevo, modificado y borrado.	7	56
Sprint 4	Auspicio	Diseñar Pantallas de listado de auspicios, adicionar modificar y borrar auspicios.	8	64

		Programar las acciones de listado, nuevo, modificado y borrado.		
Sprint 5	Anuncios	Diseñar Pantallas de listado de anuncio, adicionar modificar y borrar anuncios. Programar las acciones de listado, nuevo, modificado y borrado.	10	80
Sprint 6	Áreas	Diseñar Pantallas de listado de áreas, adicionar modificar y borrar áreas. Programar las acciones de listado, nuevo, modificado y borrado.	7	56
Sprint 7	Ubicación	Diseñar pantallas de listado de ubicación, adicionar modificar y borrar ubicación. Programar las acciones de listado, nuevo, modificado y borrado.	15	120
Sprint 8	Reporte de cobro participante	Diseñar pantalla de reporte, programar el listado de cobros participantes.	4	32
Sprint 9	Reporte de participantes inscritos	Diseñar pantalla de reporte, programar el listado de participantes inscritos.	4	32
Total, de días y horas			72	576

3.4.2 Sprint Rol Tutor

Tabla 3.2

Sprint Rol Tutor

N°	Casos de Uso	Tareas	Días	Horas
Sprint 10	Ingresar contraseñas	Diseño de interfaz para el logueo. Programar la acción de logueo.	5	40
	Validar datos	Programar la validación de datos.	2	16
Sprint 11	Inscripción del participante	Diseñar Pantallas de inscripciones, adicionar modificar y borrar inscripción. Programar las acciones de listado, nuevo, modificado y borrado.	10	80
Sprint 12	Cobro de participante	Diseñar Pantallas de cobros de participantes, adicionar modificar y borrar cobros. Programar las acciones de listado, nuevo, modificado y borrado.	10	80
Total, de días y horas			27	216

3.4.3. Sprint Rol Participante

Tabla 3.3

Sprint Rol Participante

N°	Casos de Uso	Tareas	Días	Horas
Sprint 13	Ver noticia	Diseñar la pantalla de salida de información noticia. Programar la salida.	7	56
Sprint 14	Ver evento	Diseñar la pantalla de salida de información del evento. Programar la salida.	8	64
Sprint 15	Ver auspicio	Diseñar la pantalla de salida de los auspiciadores. Programar la salida.	9	72
Sprint 16	Ver participantes	Diseñar la pantalla de salida de listado de los participantes. Programar la salida.	7	56
Sprint 17	Ver dirección	Diseñar la pantalla de salida de la dirección Programar la salida	10	80
Sprint 18	Ver galería de fotos	Diseñar la pantalla de salida de la galería de fotos Programar la salida	10	80
Total, de días y horas			51	408

3.5. Diagrama de Actividad Rol Administrador

3.5.1. Diagrama de Actividad Caso Persona

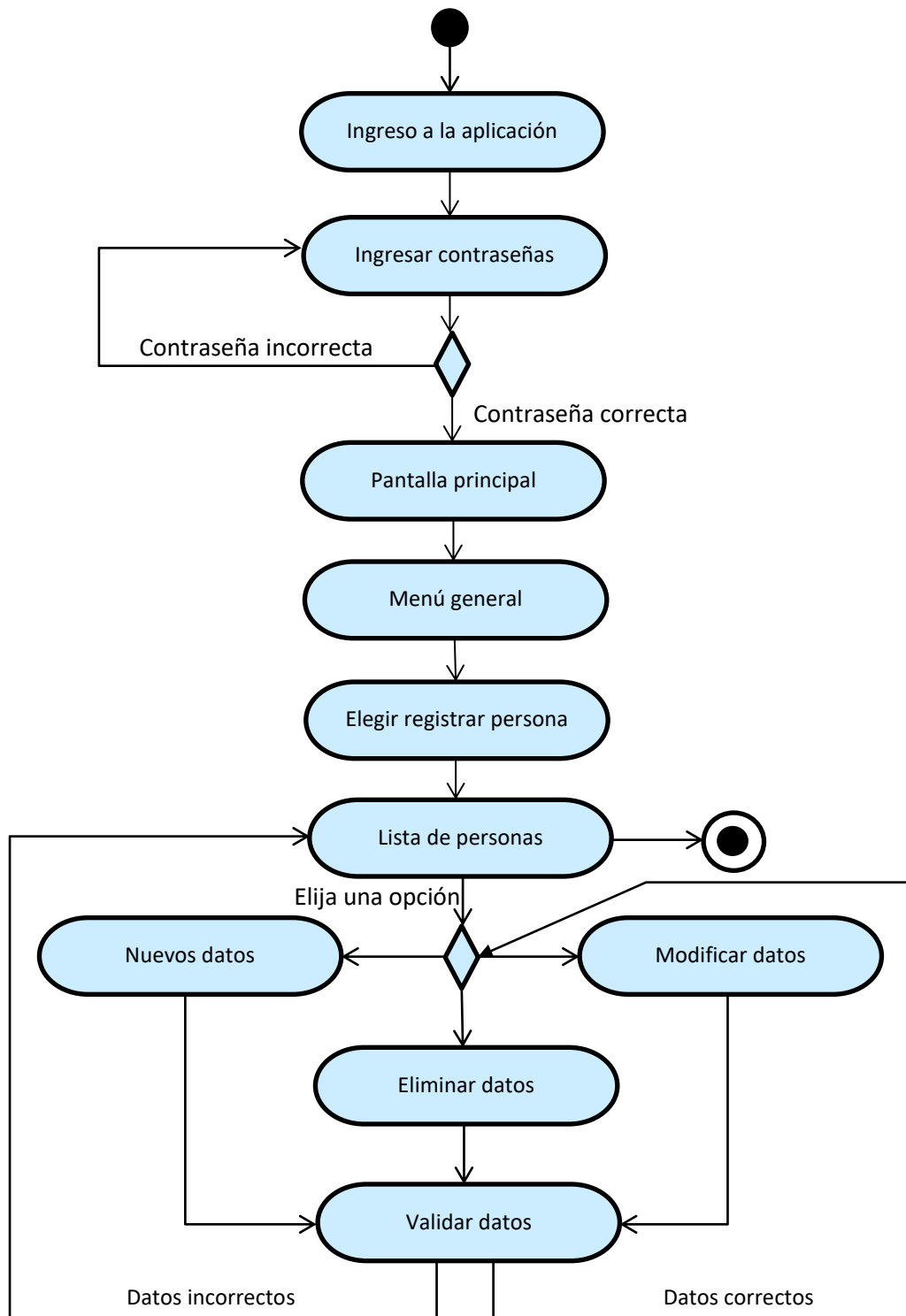


Figura 3.4. Diagrama de Actividad Caso Persona.

3.5.2. Diagrama de Actividad Caso Evento

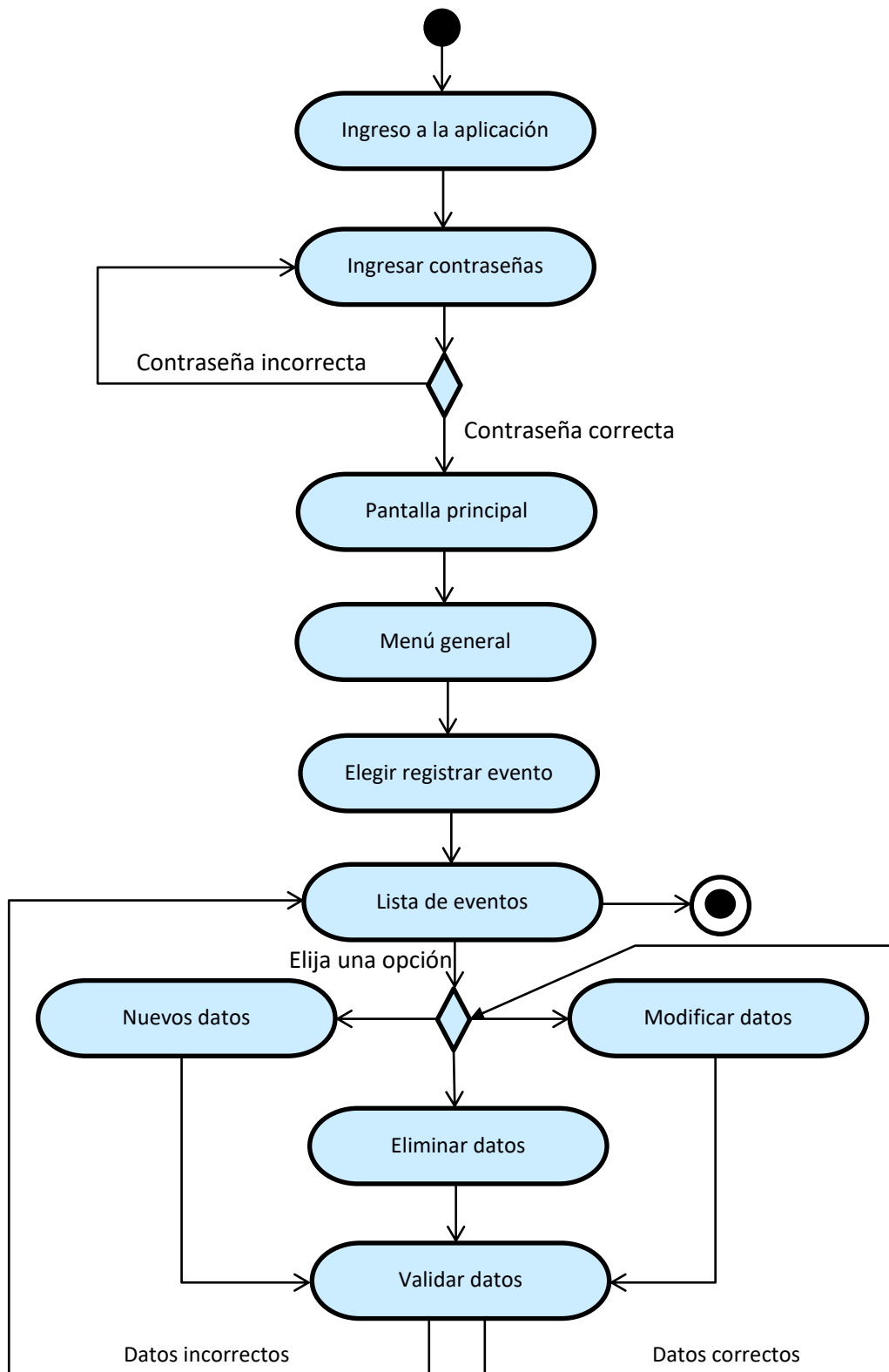


Figura 3.5. Diagrama de Actividad Caso Evento.

3.5.3. Diagrama de Actividad Auspicio

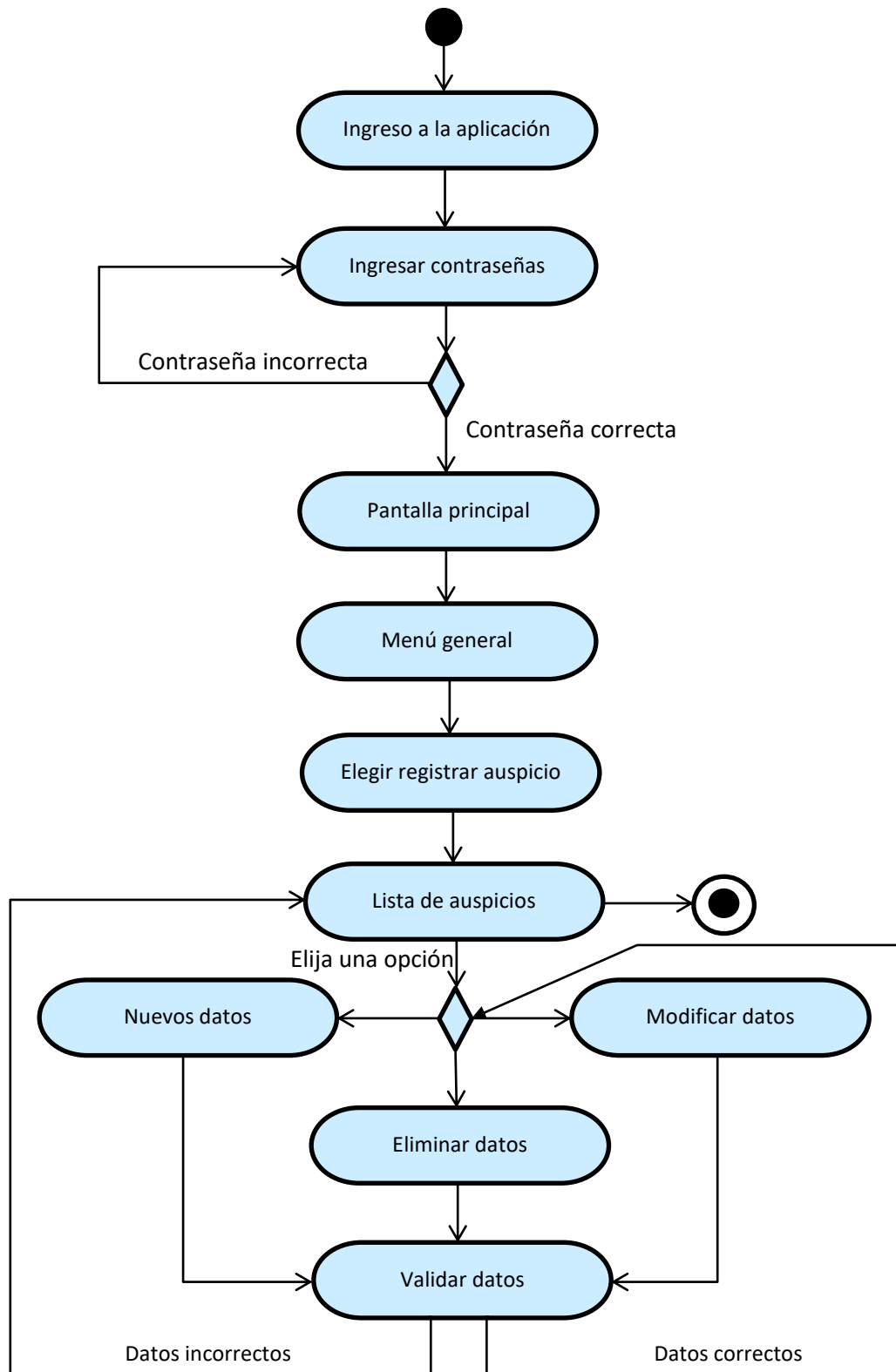


Figura 3.6. Diagrama de Actividad Auspicio.

3.5.4. Diagrama de Actividad Anuncio

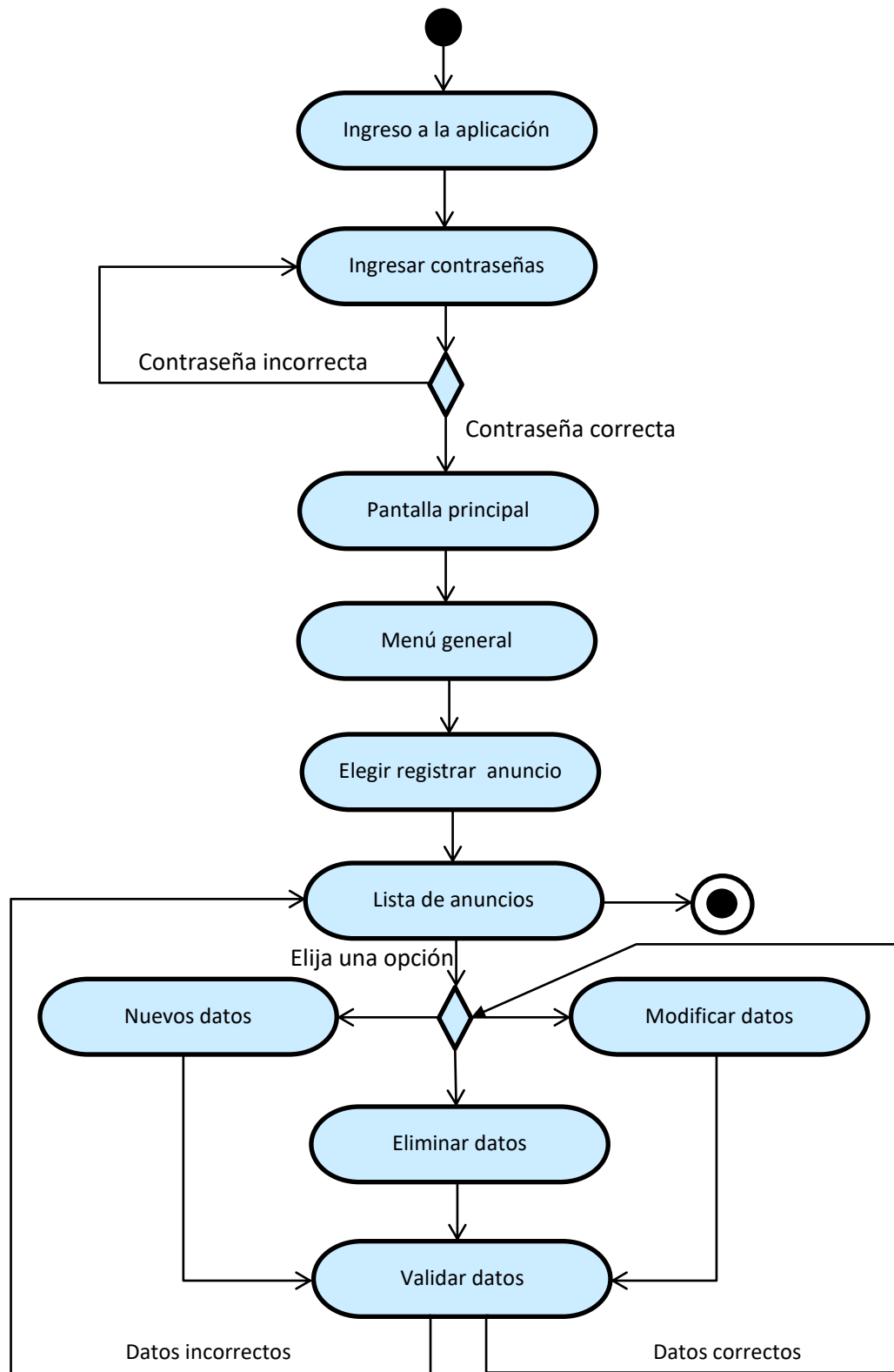


Figura 3.7. Diagrama de Actividad Anuncio.

3.5.5. Diagrama de Actividad Área

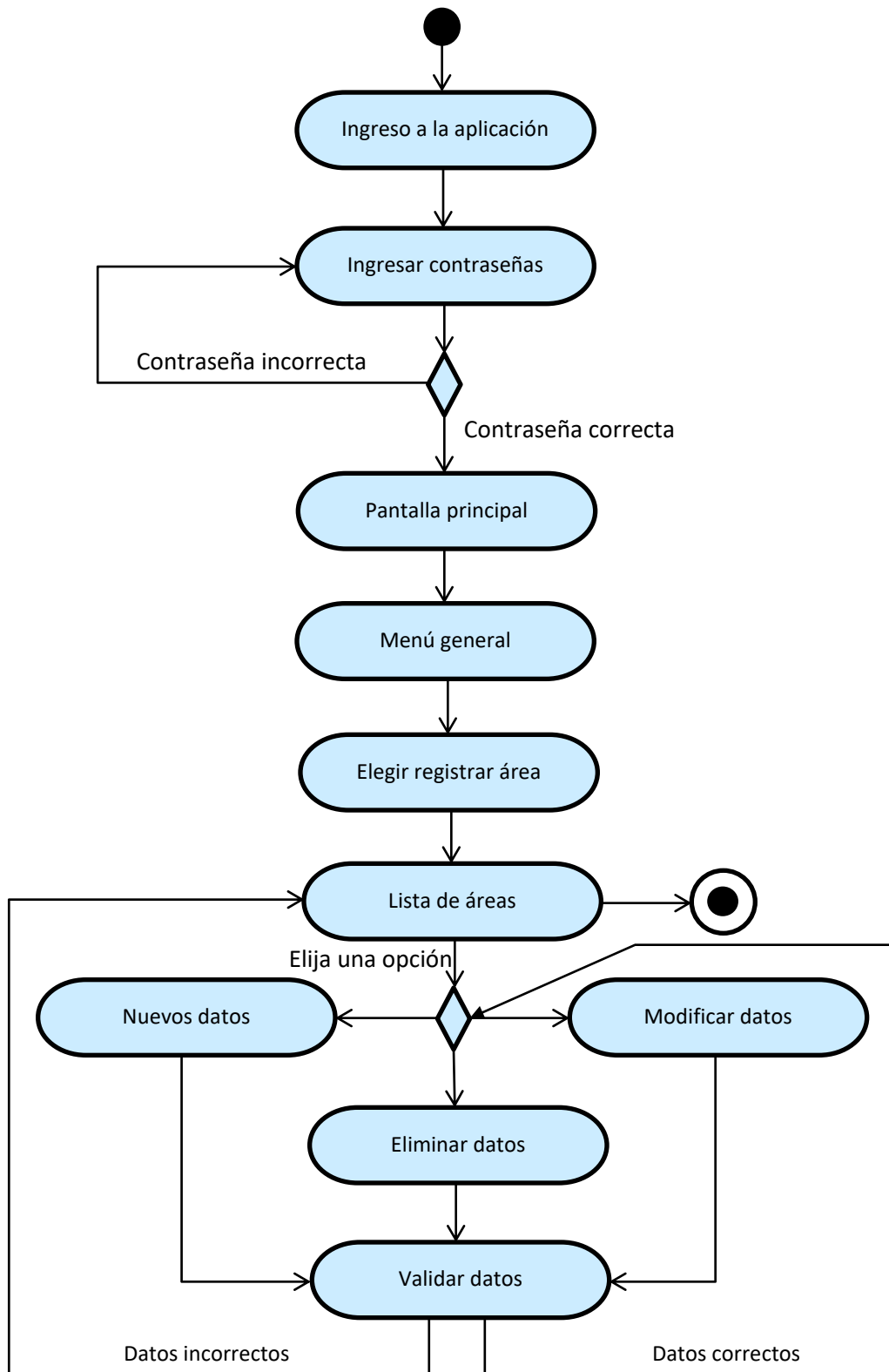


Figura 3.8. Diagrama de Actividad Área.

3.5.6. Diagrama de Actividad Reporte de Participante

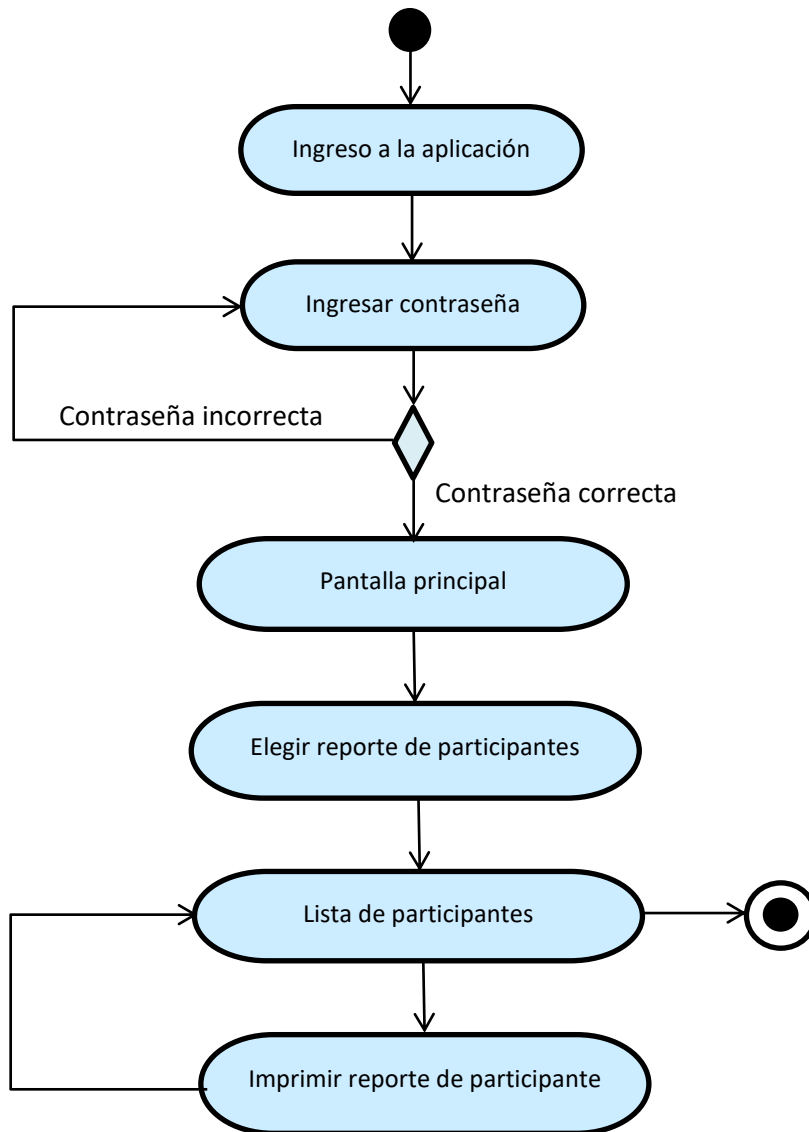


Figura 3.9. Diagrama de Actividad Reporte de Participante.

3.5.7. Diagrama de Actividad Reporte Cobro de Participante

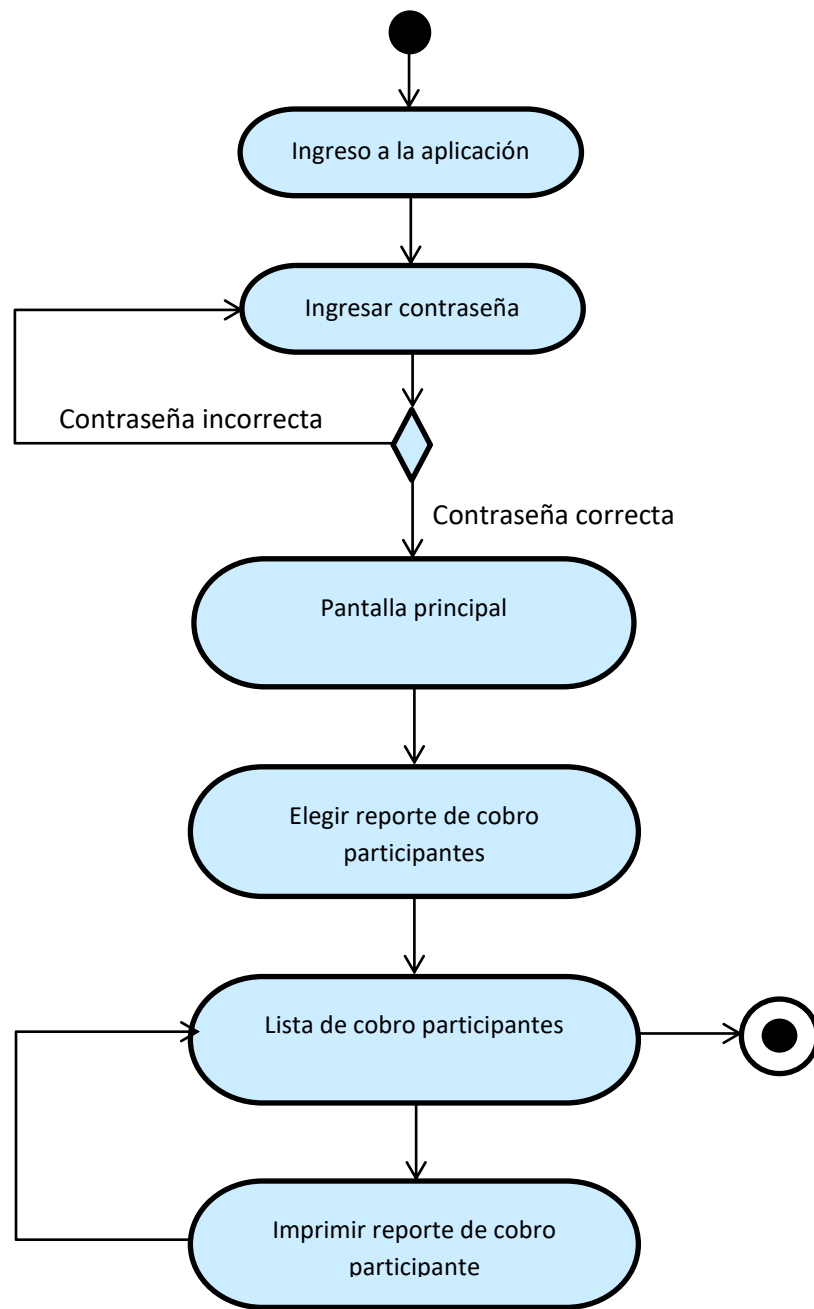


Figura 3.10. Diagrama de Actividad Reporte Cobro de Participante.

3.6. Diagramas de Actividad Rol Tutor

3.6.1. Diagrama de Actividad Registrar Participante

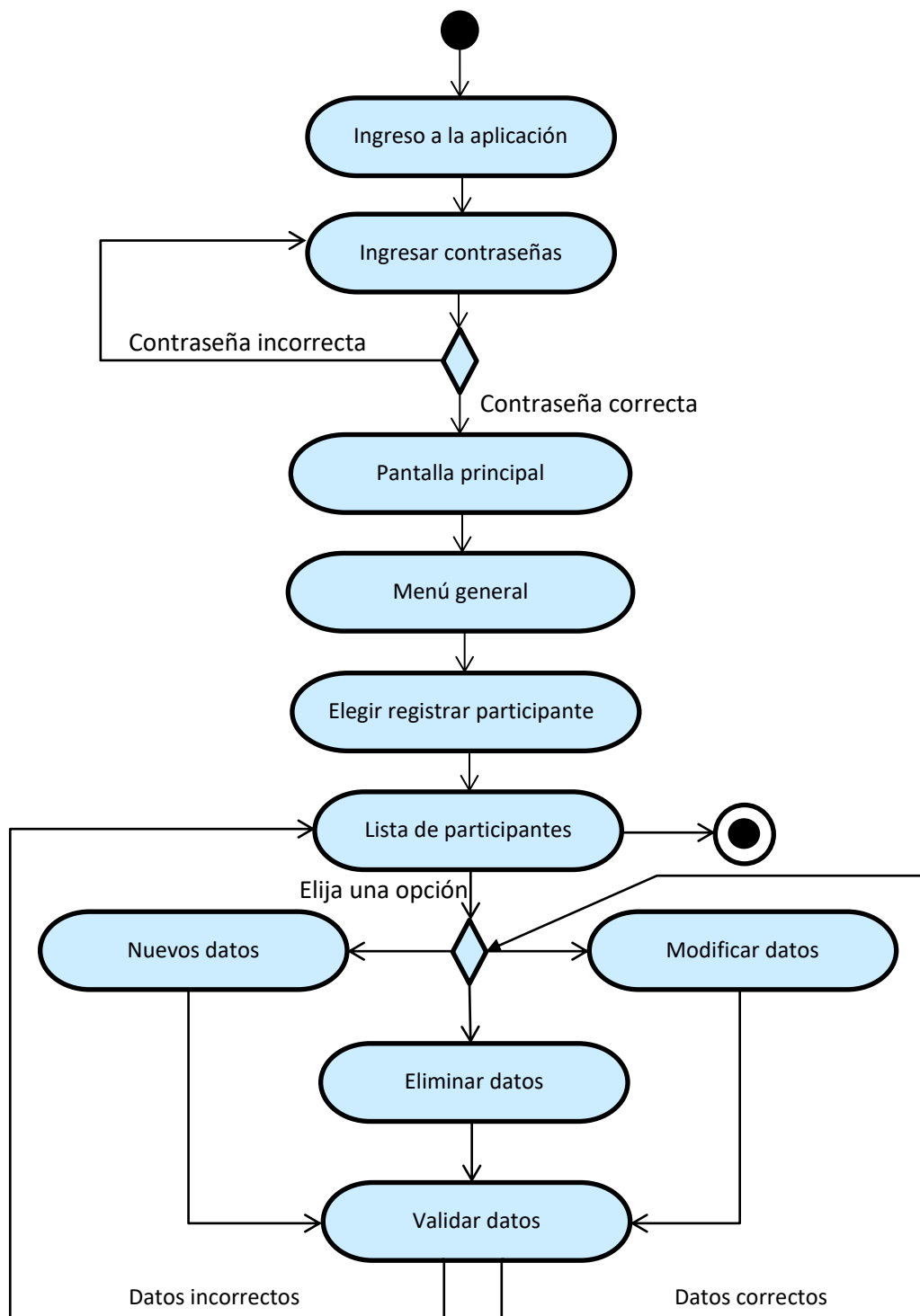


Figura 3.11. Diagrama de Actividad Registrar Participante.

3.6.2. Diagrama de Actividad Registrar Cobro Participante

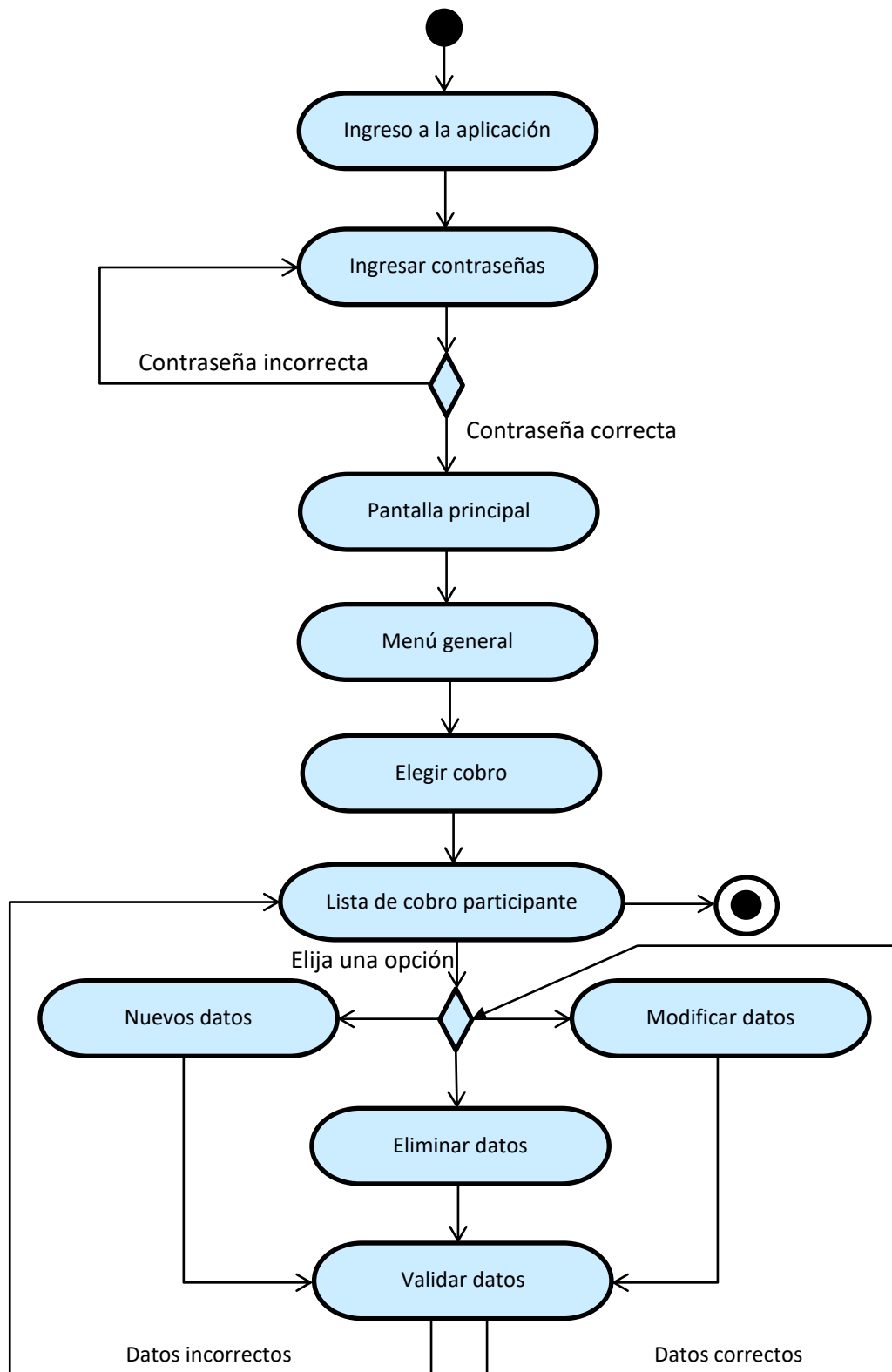


Figura 3.12. Diagrama de Actividad Registrar Cobro Participante.

3.6.3. Diagrama de Actividad Credencial

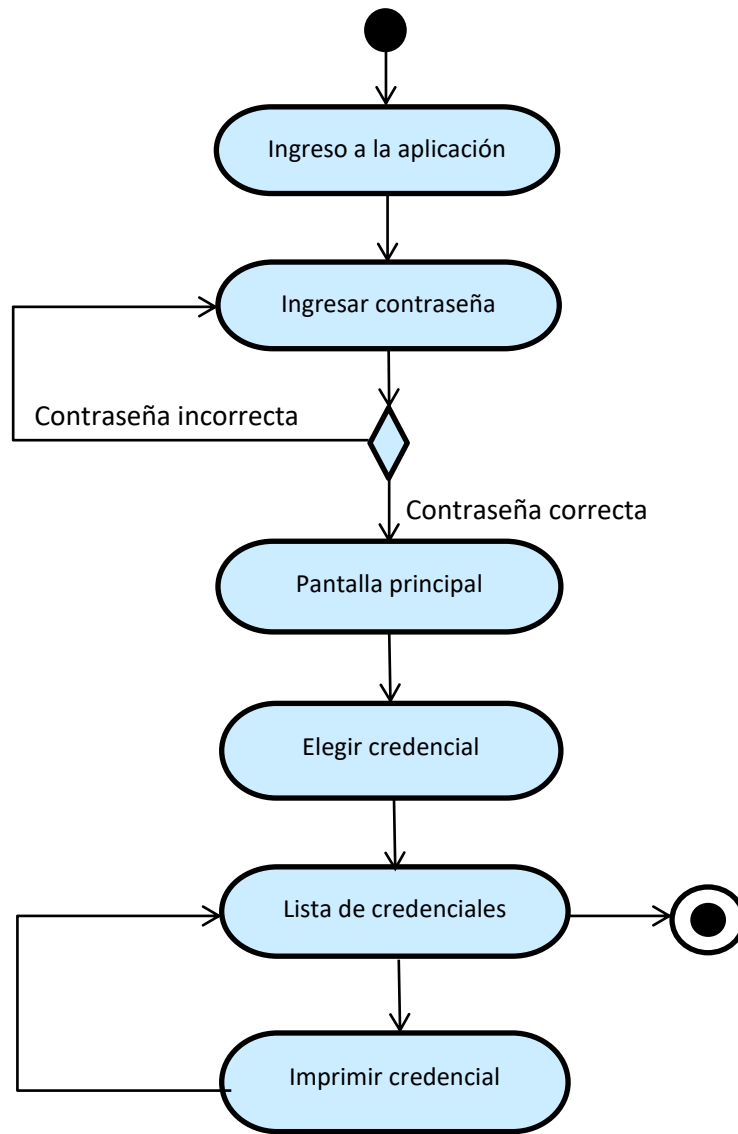


Figura 3.13. Diagrama de Actividad Credencial.

3.7. Diagrama de Actividades Rol Participante

3.7.1. Diagrama de Actividad Noticia - Evento - Área

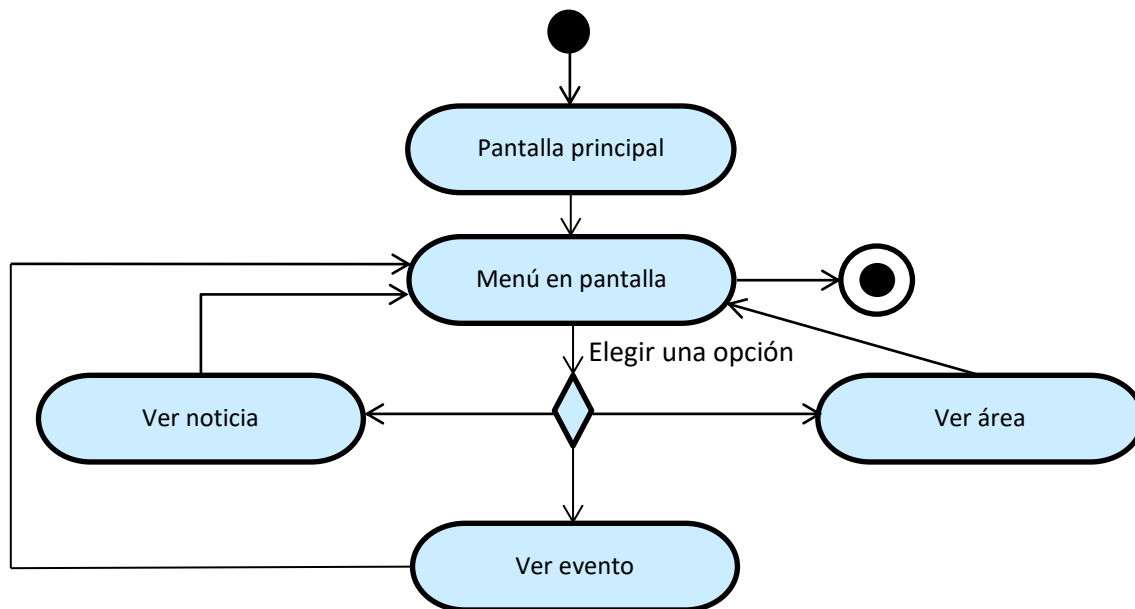


Figura 3.14. Diagrama de Actividad Noticia - Evento - Área.

3.7.2. Diagrama de Actividad Auspicio - Lista Participante - Dirección - Galería Fotos.

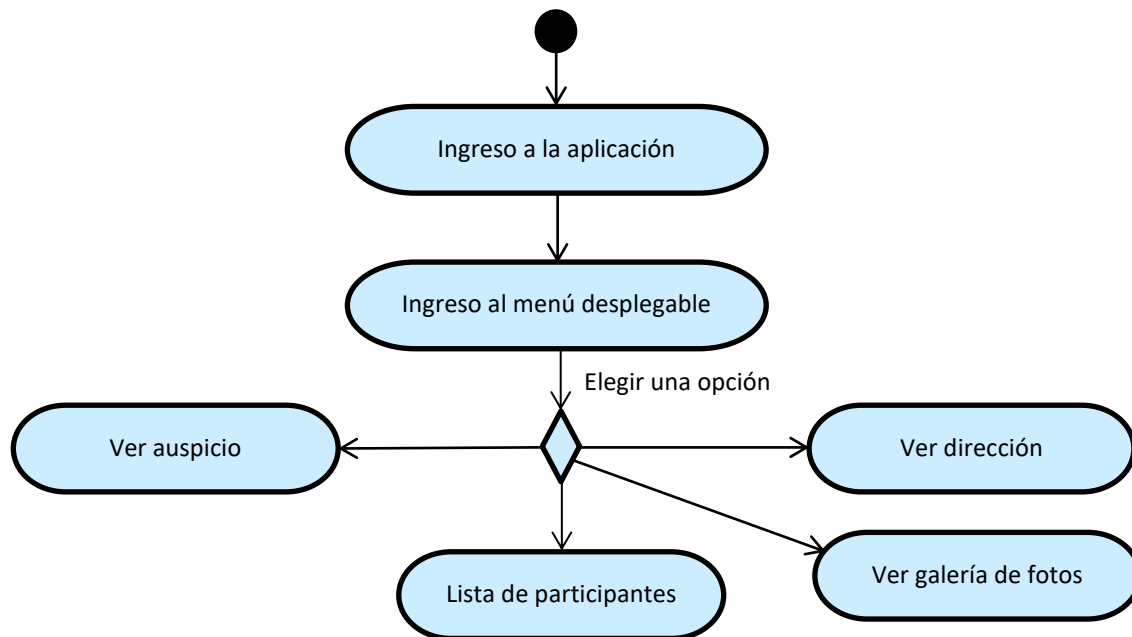


Figura 3.15. Diagrama de Actividad Auspicio - Lista Participante - Dirección - Galería Fotos.

3.8. Diagramas de Secuencia Rol Administrador

3.8.1 Diagrama de Secuencia Registro de Persona

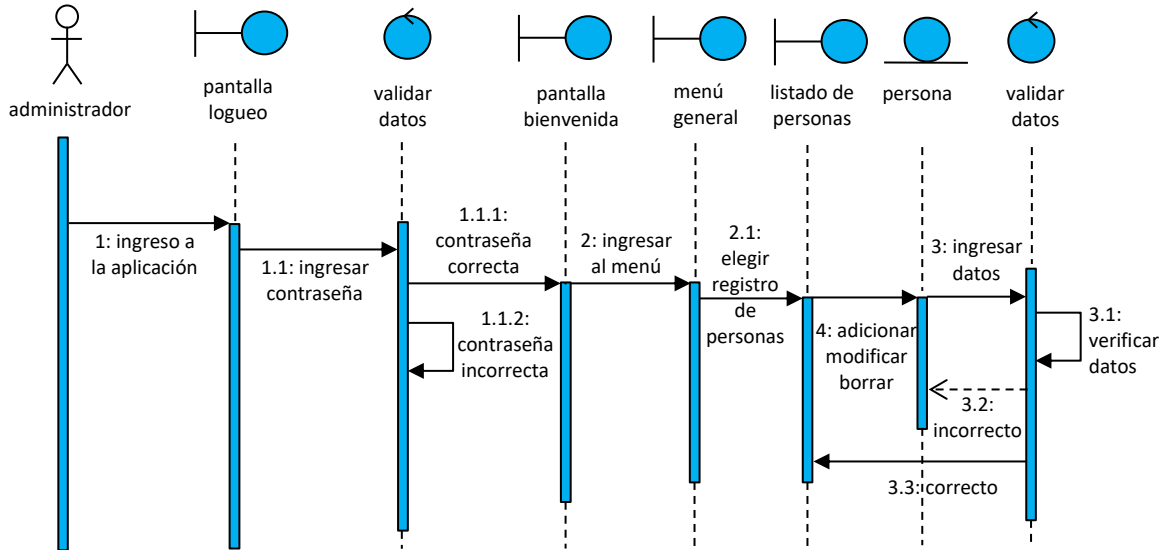


Figura 3.16. Diagrama de Secuencia Registro de Persona.

3.8.2. Diagrama de Secuencia Registro del Evento

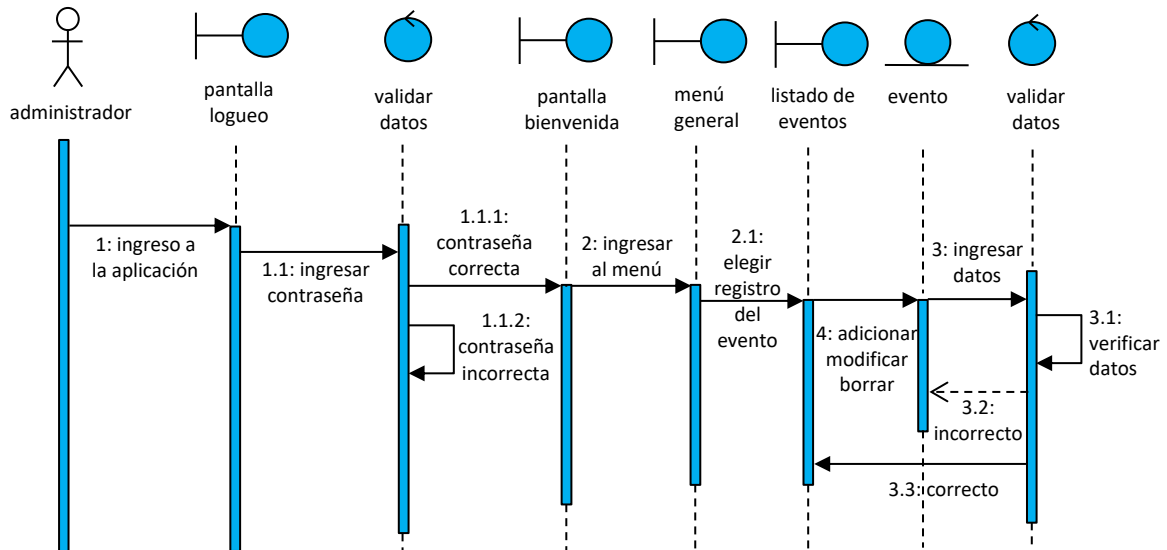


Figura 3.17. Diagrama de Secuencia Registro del Evento.

3.8.3. Diagrama de Secuencia Registro de Anuncio

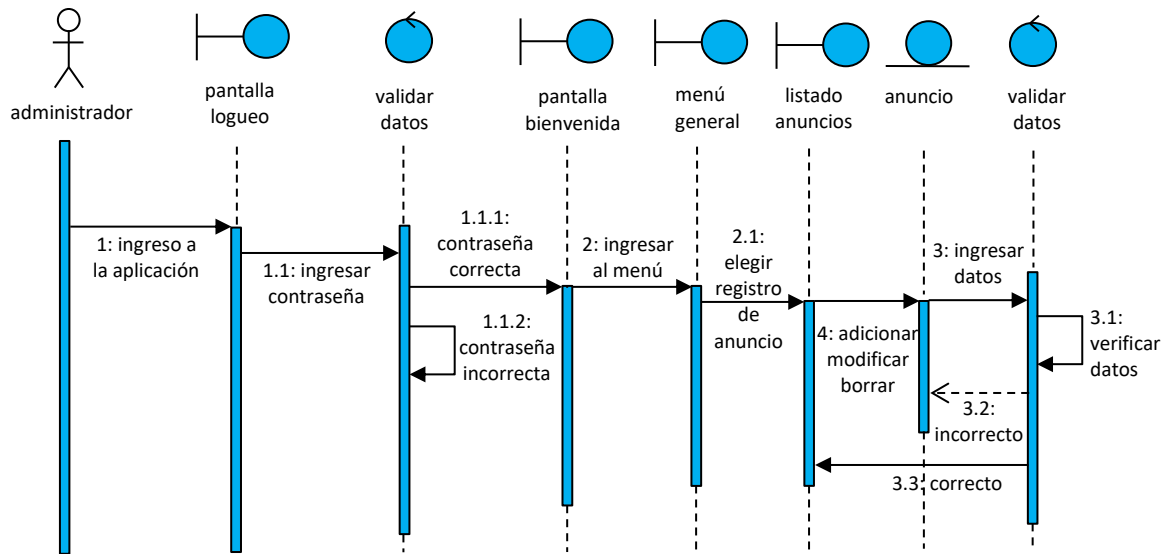


Figura 3.18. Diagrama de Secuencia Registro de Anuncio.

3.8.4. Diagrama de Secuencia Registro Área de Competencia

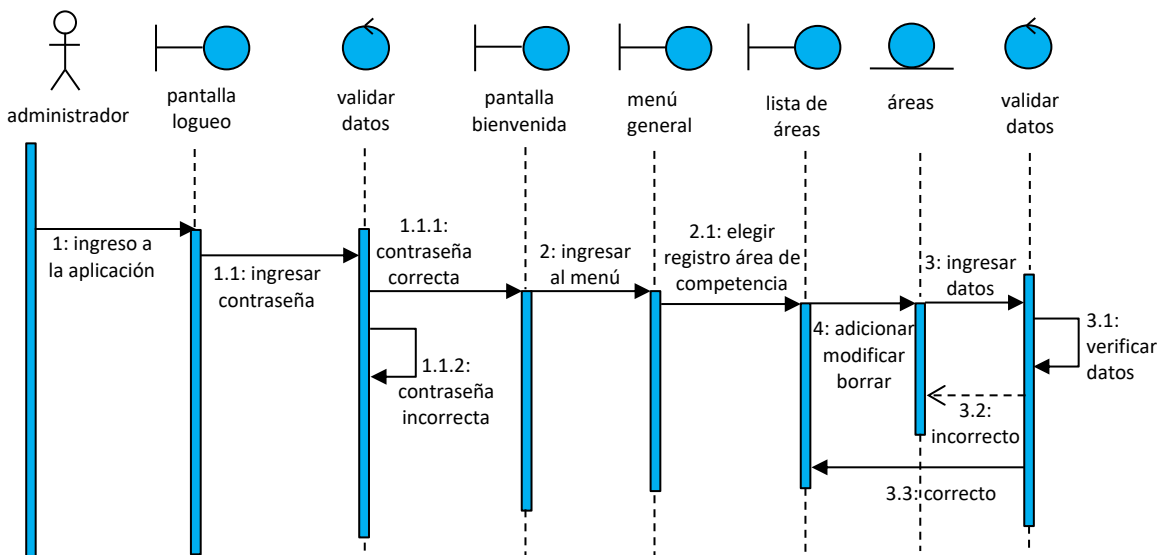


Figura 3.19. Diagrama de Secuencia Registro Área de Competencia.

3.8.5. Diagrama de Secuencia Registro de Auspiciadores

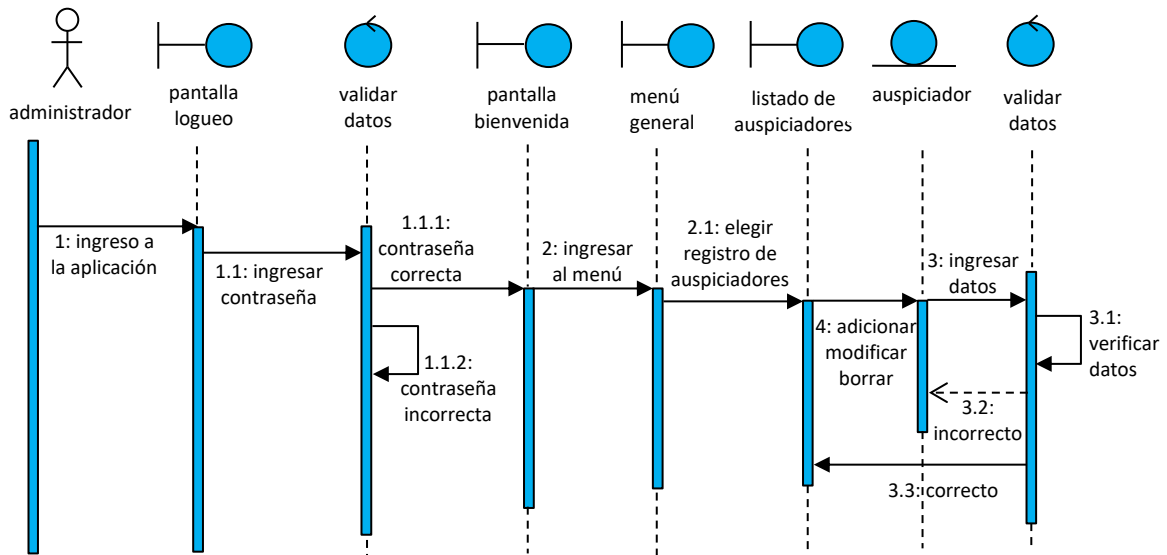


Figura 3.20. Diagrama de Secuencia Registro de Auspiciadores.

3.8.6. Diagrama de Secuencia Ubicación del Evento

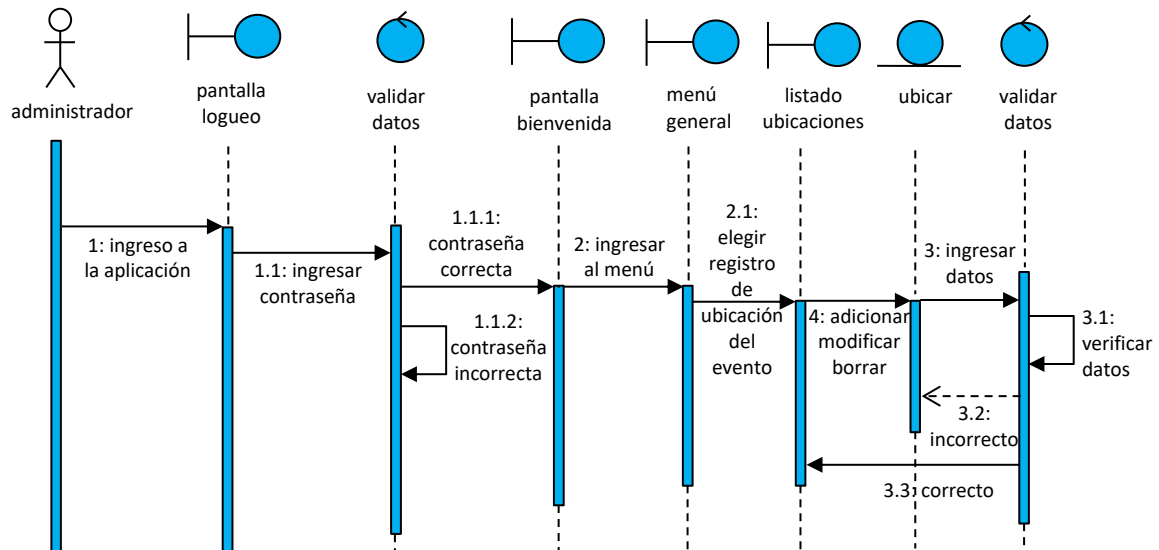


Figura 3.21. Diagrama de Secuencia Ubicación del Evento.

3.8.7. Diagrama de Secuencia Reporte de Participantes

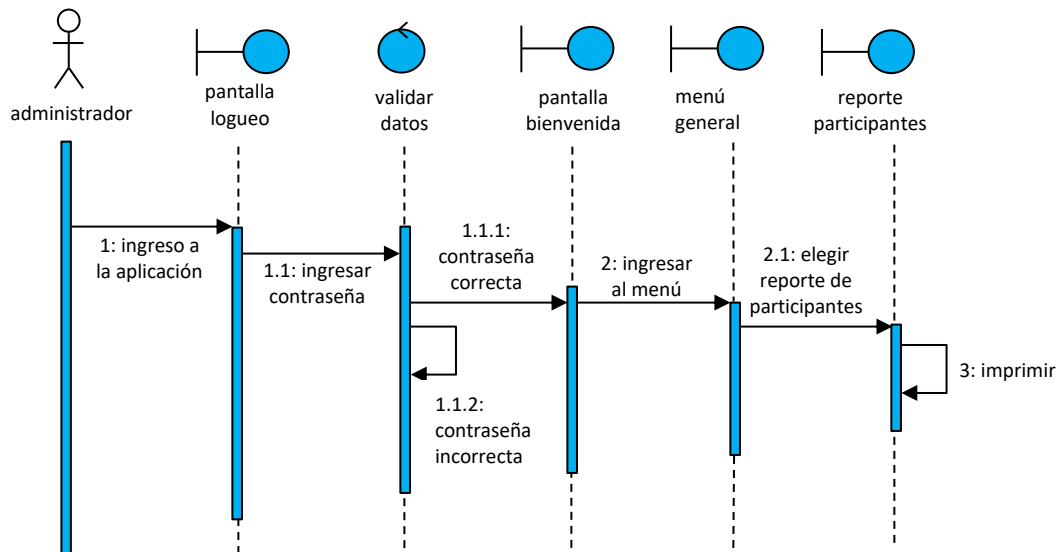


Figura 3.22. Diagrama de Secuencia Reporte de Participantes.

3.8.8. Diagrama de Secuencia Reporte de Cobro Participantes

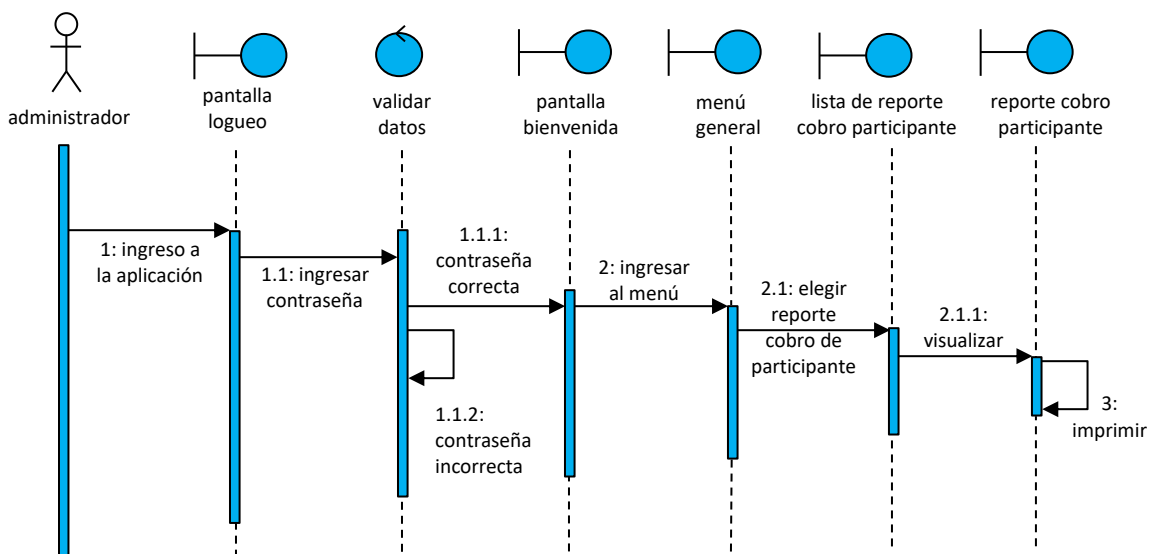


Figura 3.23. Diagrama de Secuencia Reporte de Cobro Participantes.

3.9. Diagrama de Secuencia Rol Tutor

3.9.1. Diagrama de Secuencia Registro de Participantes

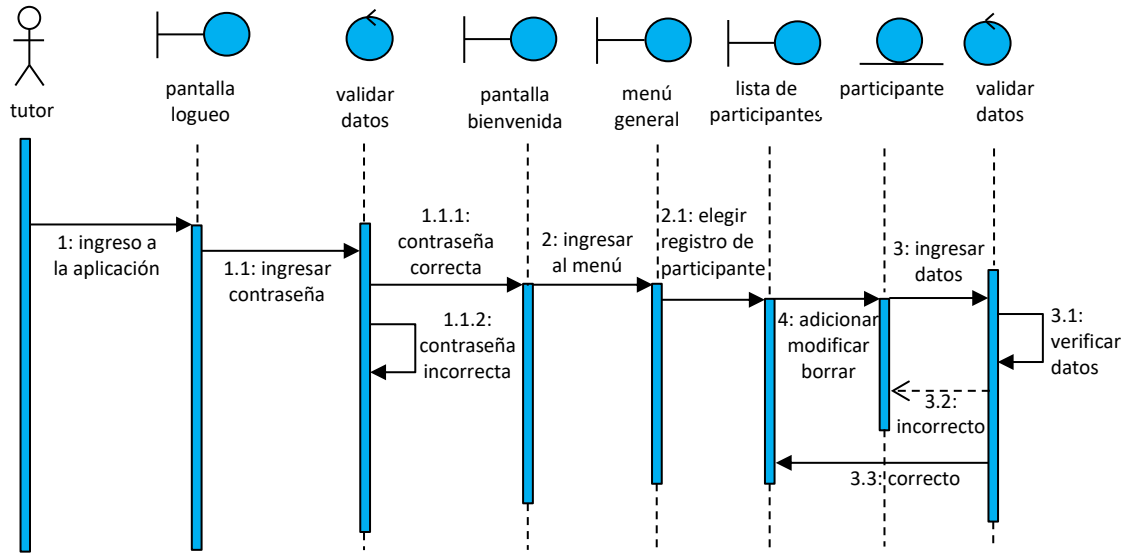


Figura 3.24. Diagrama de Secuencia Registro de Participantes.

3.9.2. Diagrama de Secuencia Registro de Cobro

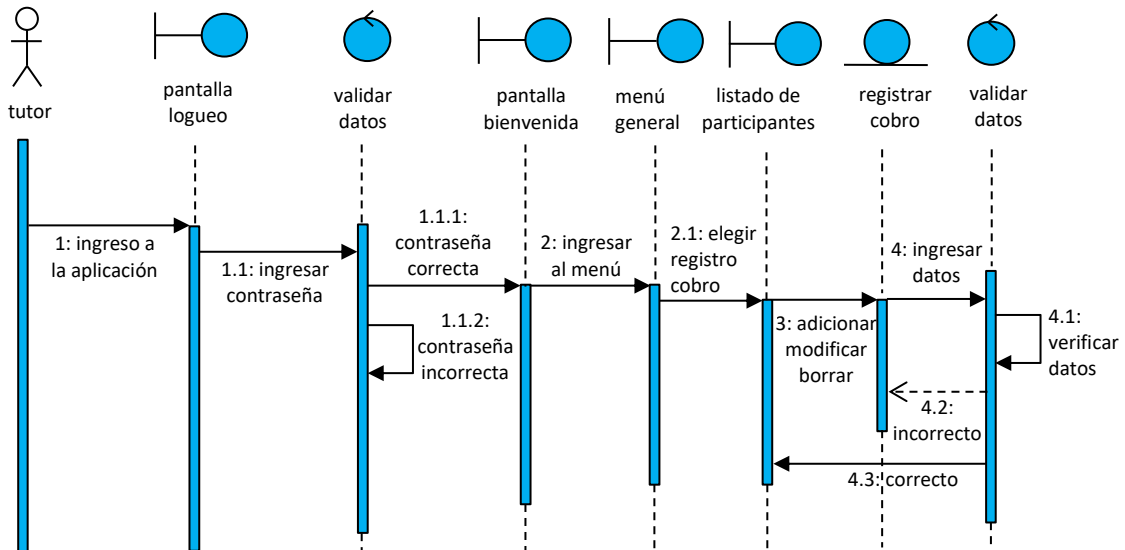


Figura 3.25. Diagrama de Secuencia Registro de Cobro.

3.9.3. Diagrama de Secuencia Reporte de Cobro

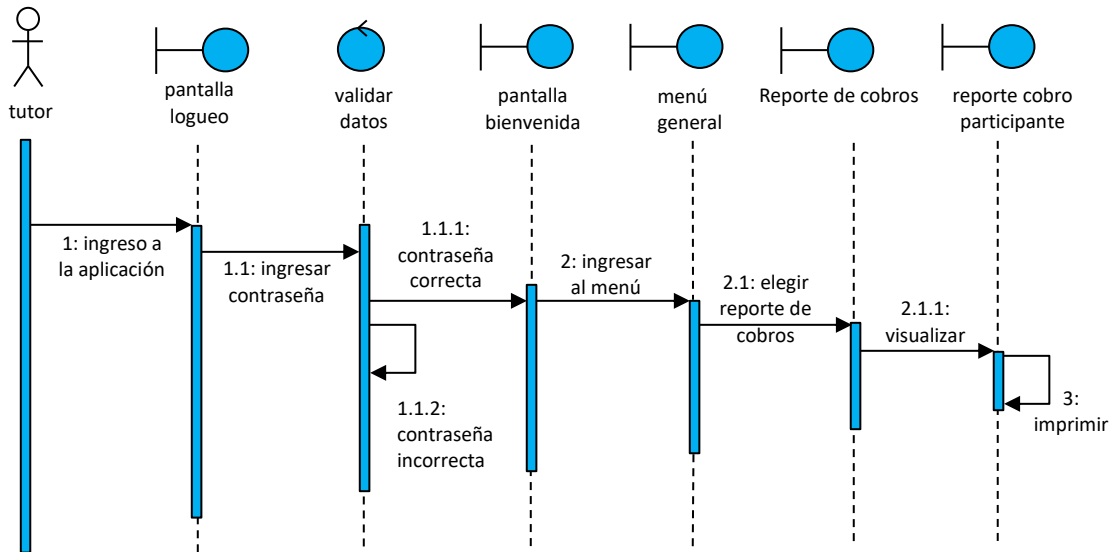


Figura 3.26. Diagrama de Secuencia Reporte de Cobro.

3.9.4. Diagrama de Secuencia Reporte Credencial de Participantes

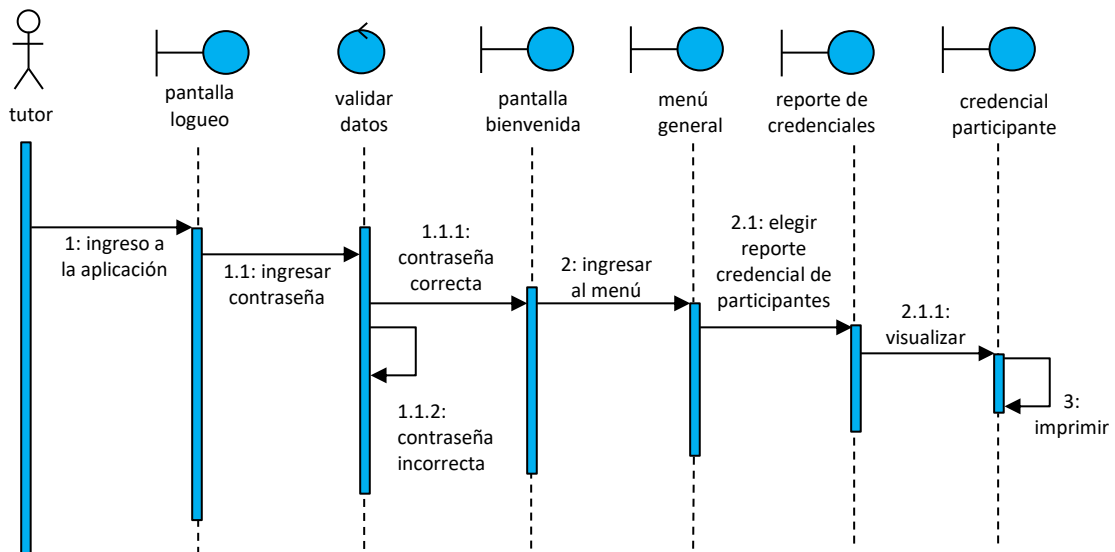


Figura 3.27. Diagrama de Secuencia Reporte Credencial de Participantes.

3.10. Diagrama de Secuencia Rol Participante

3.10.1. Diagrama de Secuencia Noticia

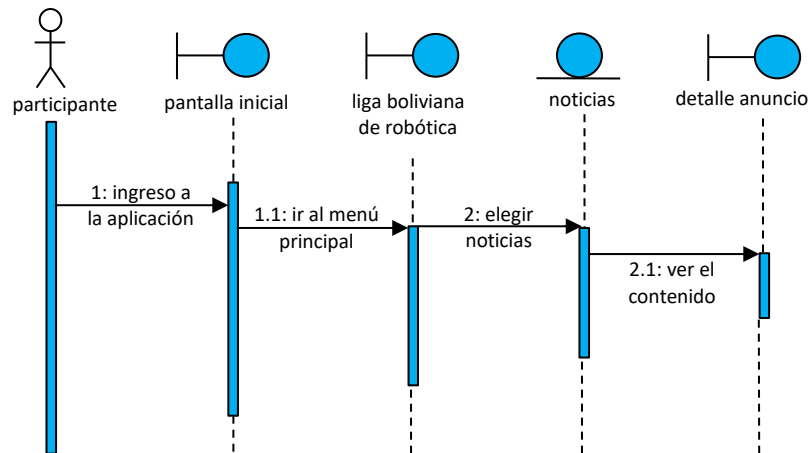


Figura 3.28. Diagrama de Secuencia Noticia.

3.10.2. Diagrama de Secuencia Evento

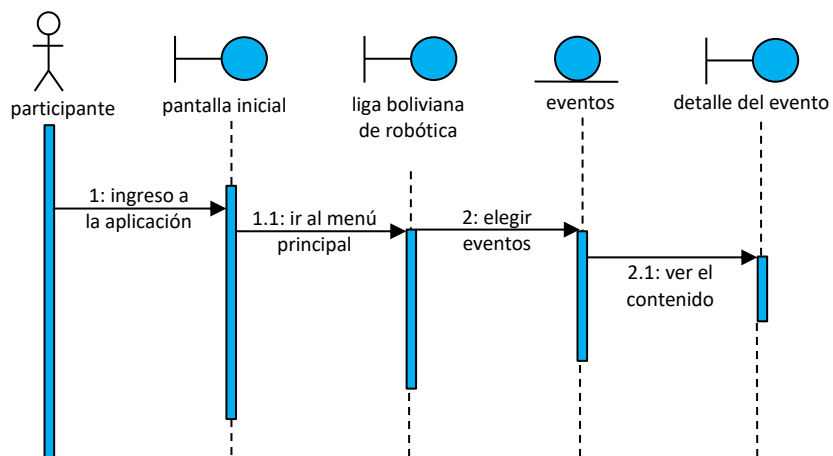


Figura 3.29. Diagrama de Secuencia Evento.

3.10.3. Diagrama de Secuencia Áreas

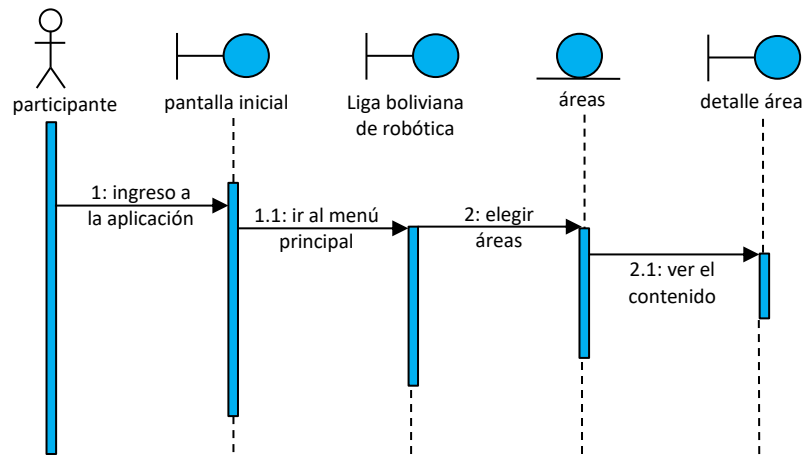


Figura 3.30. Diagrama de Secuencia Áreas.

3.10.4. Diagrama de Secuencia Auspicios

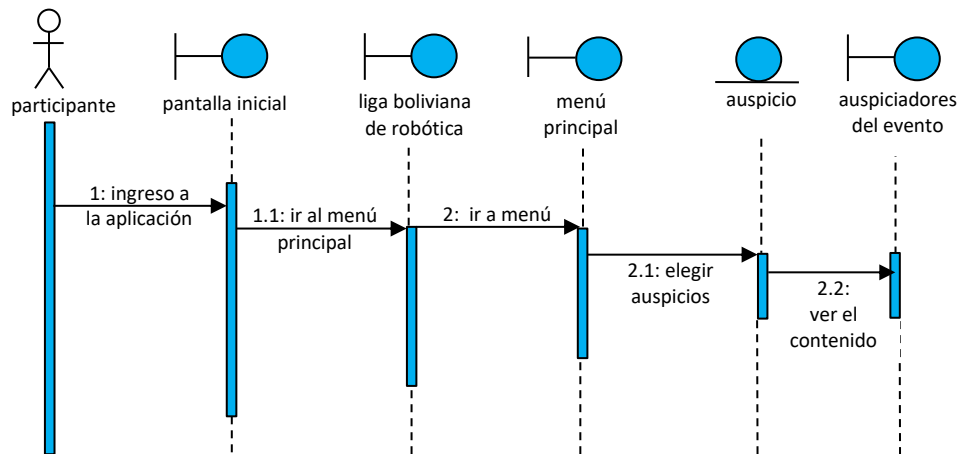


Figura 3.31. Diagrama de Secuencia Auspicios.

3.10.5. Diagrama de Secuencia Participantes Inscritos

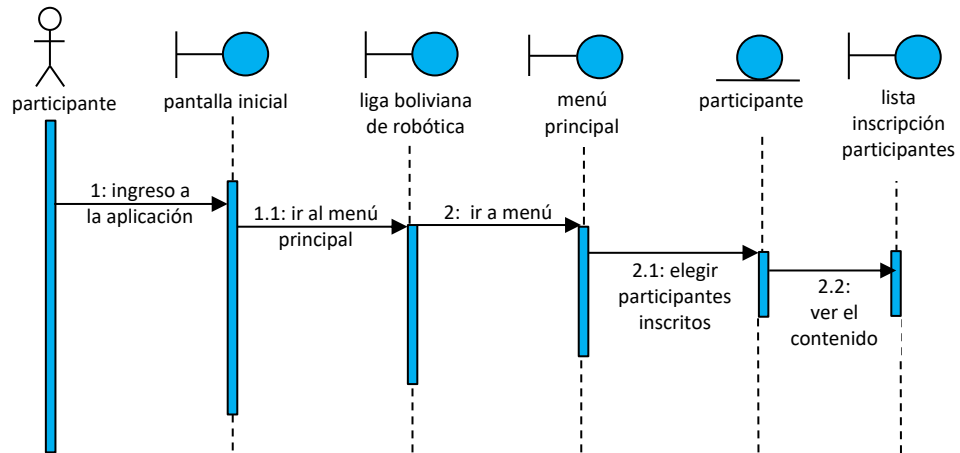


Figura 3.32. Diagrama de Secuencia Participantes Inscritos.

3.10.6. Diagrama de Secuencia Dirección

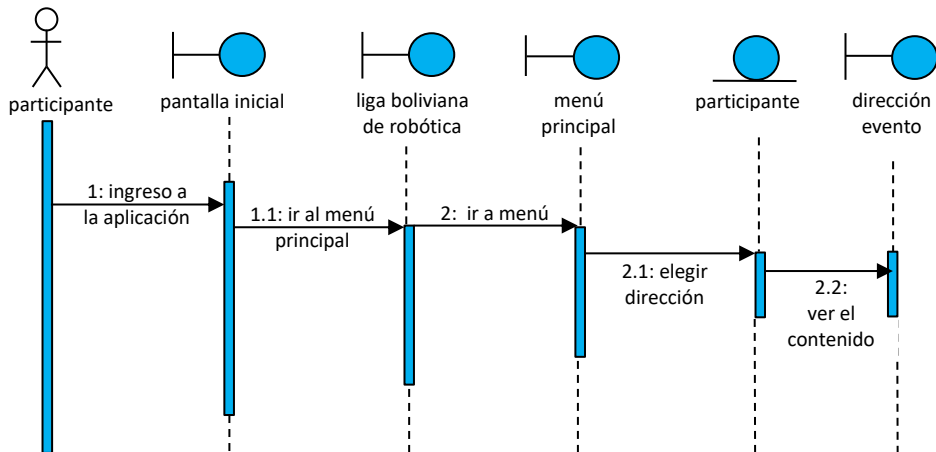


Figura 3.33. Diagrama de Secuencia Dirección.

3.10.7. Diagrama de Secuencia Galería de Fotos

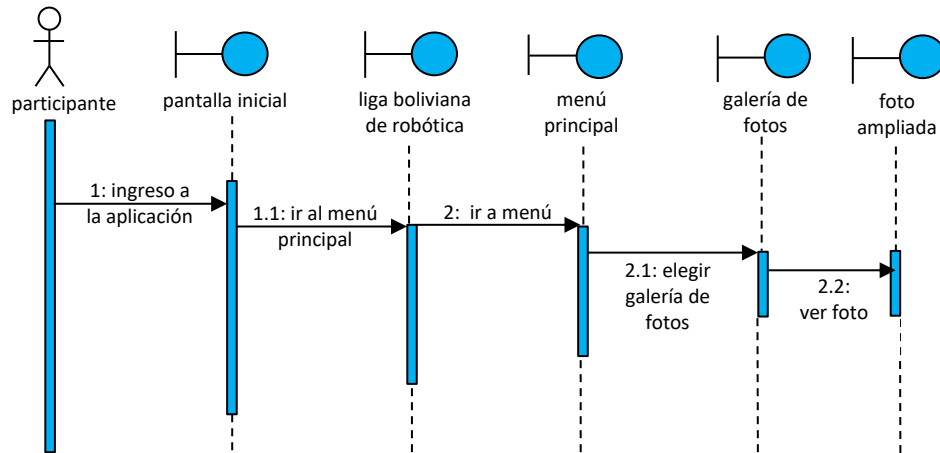


Figura 3.33. Diagrama de Secuencia Galería de Fotos.

3.11. Estructuras de Pantallas

3.11.1. Estructuras del Rol Administrador

3.11.1.1. Estructura 1 - Logueo

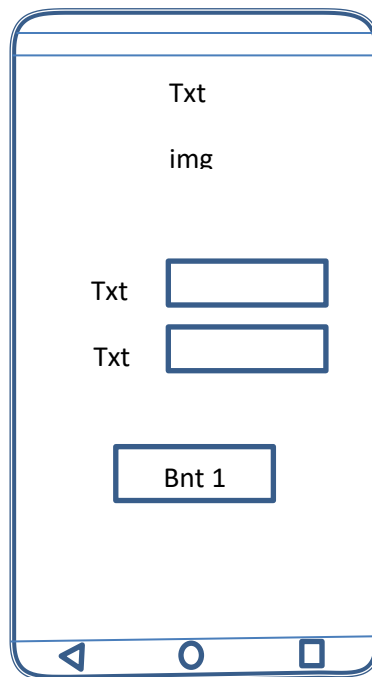


Figura 3.34. Diseño Estructura 1 – Logueo.

3.11.1.2. Estructura 2 - Bienvenida

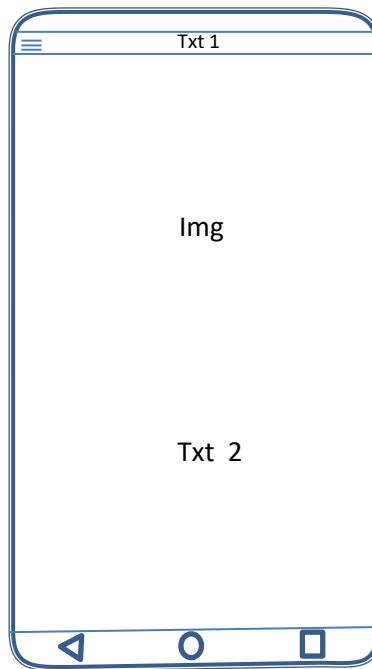


Figura 3.35. Diseño Estructura 2 - Bienvenida.

3.11.1.3. Estructura 3 - Menú General

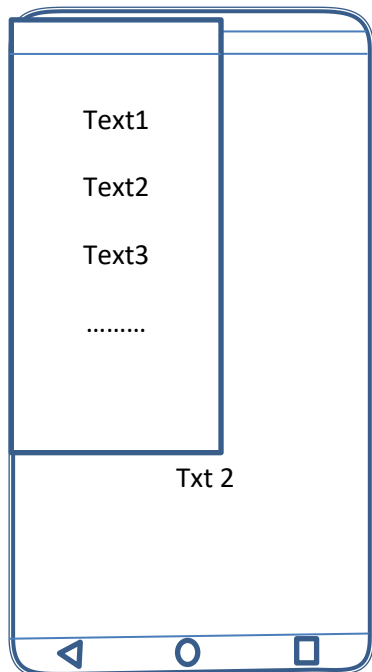


Figura 3.36. Diseño Estructura 3 – Menú General.

3.11.1.4. Estructura 4 - Listado de Personas



Figura 3.37. Diseño Estructura 4 - Listado de Personas.

3.11.1.5. Estructura 5 - Nueva Persona

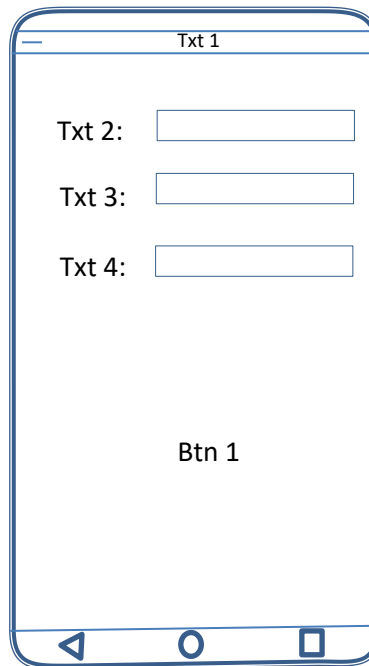


Figura 3.38. Diseño Estructura 5 - Nueva Persona.

3.11.1.6. Estructura 6 - Modificar Persona

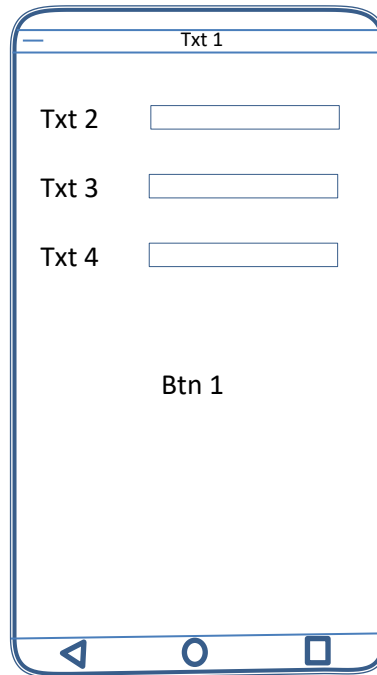


Figura 3.39. Diseño Estructura 6 - Modificar Persona.

3.11.1.7. Estructura 7 - Borrar Persona

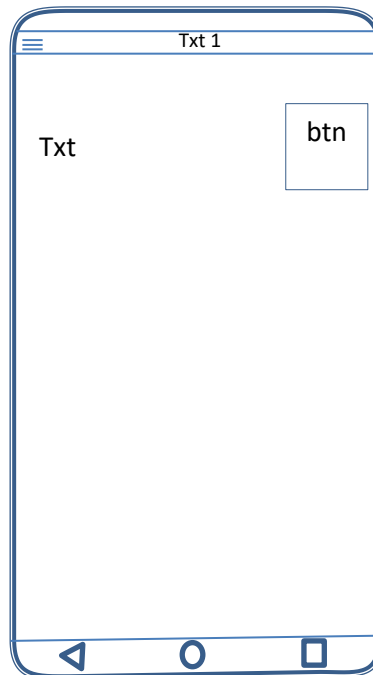


Figura 3.40. Diseño Estructura 7 - Borrar Persona.

3.11.1.8. Estructura 8 - Listado de Participantes

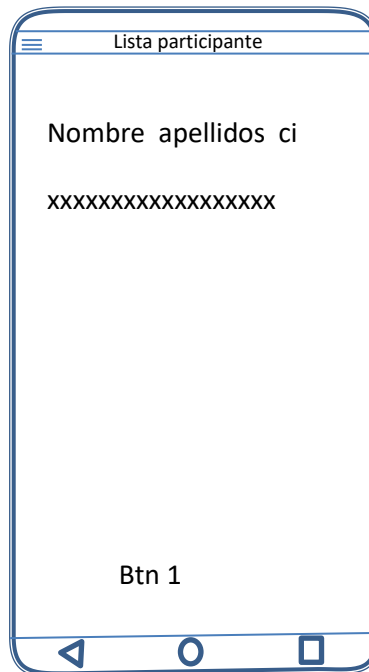


Figura 3.41. Diseño Estructura 8 - Listado de Participantes.

3.11.1.9 Estructura 9 - Nuevo Participante

0

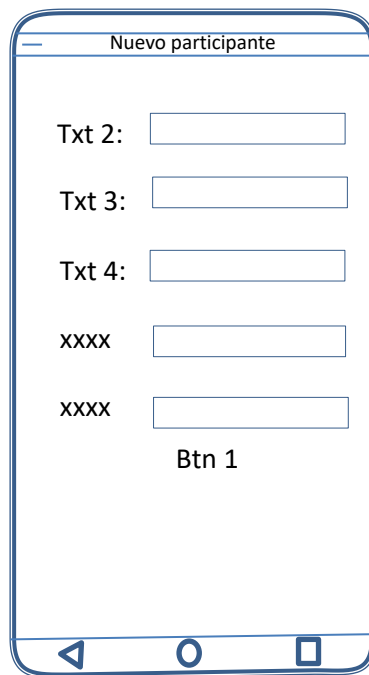


Figura 3.42. Diseño Estructura 9 - Nuevo Participante.

3.11.1.10. Estructura 10 - Modificar Participante

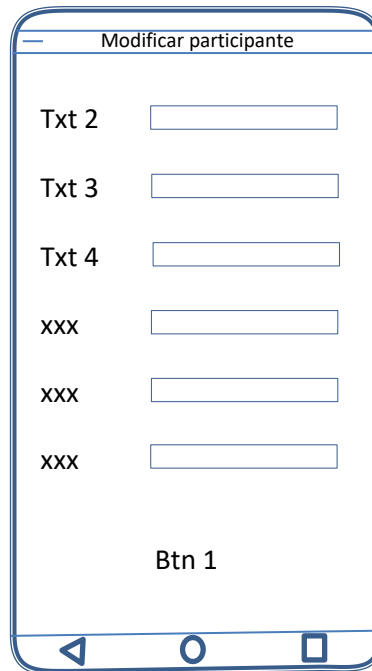


Figura 3.43. Diseño Estructura 10 - Modificar Participante.

3.11.1.11. Estructura 11 - Eliminar Participante

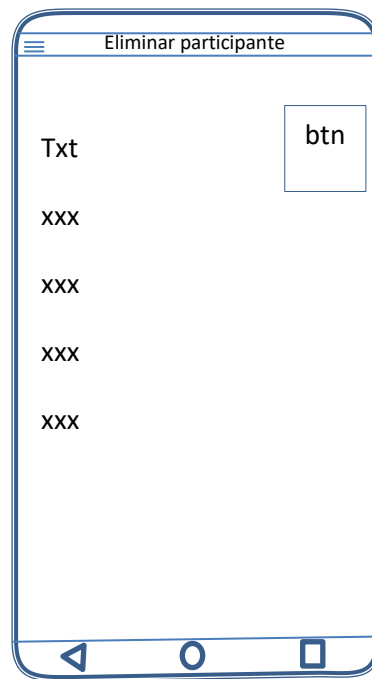


Figura 3.44. Diseño Estructura 11 – Eliminar Participante.

3.11.1.12. Estructura 12 - Listado de Eventos

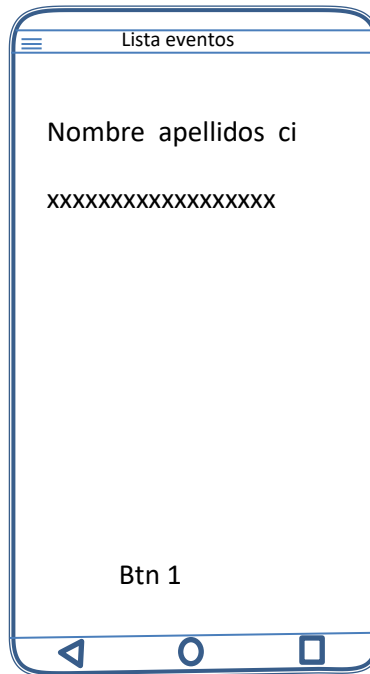


Figura 3.45. Diseño Estructura 12 – Listado de Eventos.

3.11.1.13. Estructura 13 - Nuevos Eventos

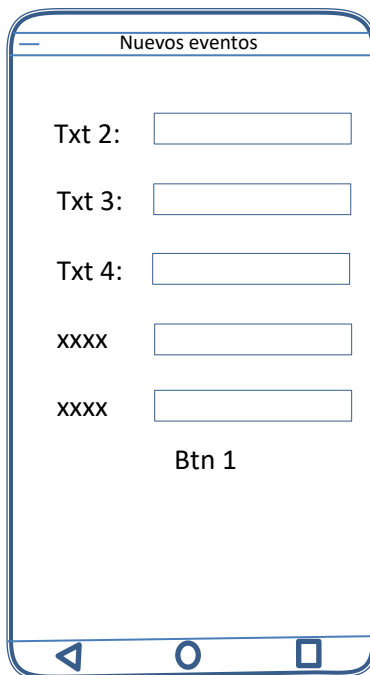


Figura 3.46. Diseño Estructura 13 – Nuevos Eventos.

3.11.1.14 Estructura 14 - Modificar Eventos

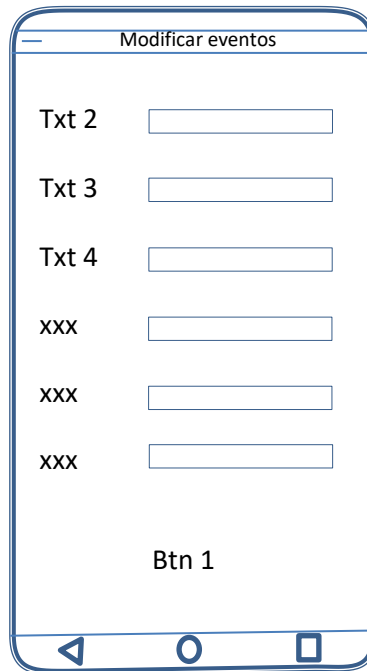


Figura 3.47. Diseño Estructura 14 - Modificar Eventos.

3.11.1.15. Estructura 15 - Eliminar Eventos

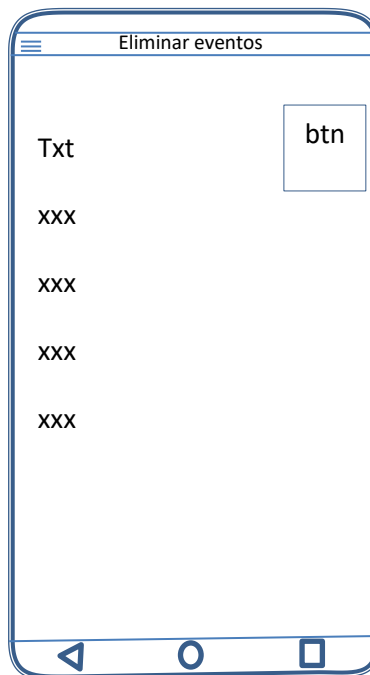


Figura 3.48. Diseño Estructura 15 – Eliminar Eventos.

3.11.1.16. Estructura 16 - Listado de Auspicios

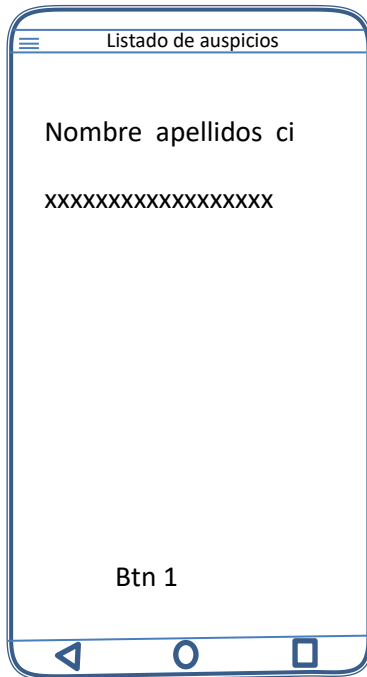


Figura 3.49. Diseño Estructura 16 - Listado de Auspicios.

3.11.1.17. Estructura 17 - Nuevo Auspicio

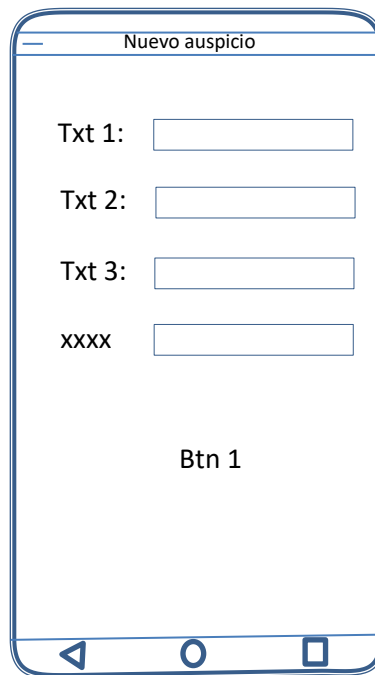


Figura 3.50. Diseño Estructura 17 – Nuevo Auspicio.

3.11.1.18. Estructura 18 - Modificar Auspicio

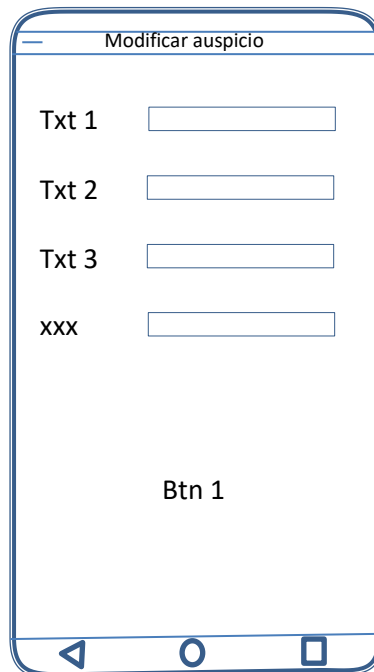


Figura 3.51. Diseño Estructura 18 – Modificar Auspicio.

3.11.1.19. Estructura 19 - Eliminar Auspicio

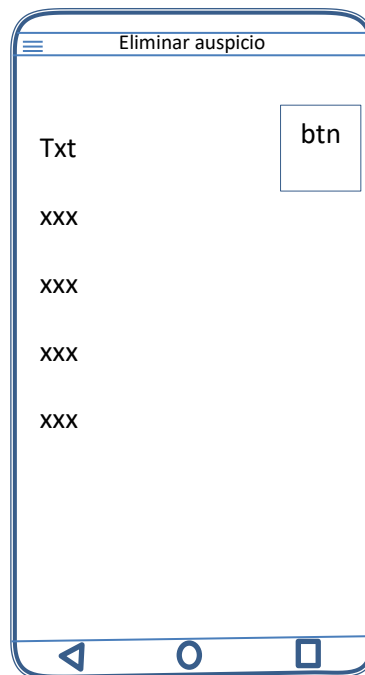


Figura 3.52. Diseño Estructura 19 - Eliminar Auspicio.

3.11.2. Estructuras del Rol Tutor

3.11.2.1. Estructura 1 - Pantalla de Bienvenida

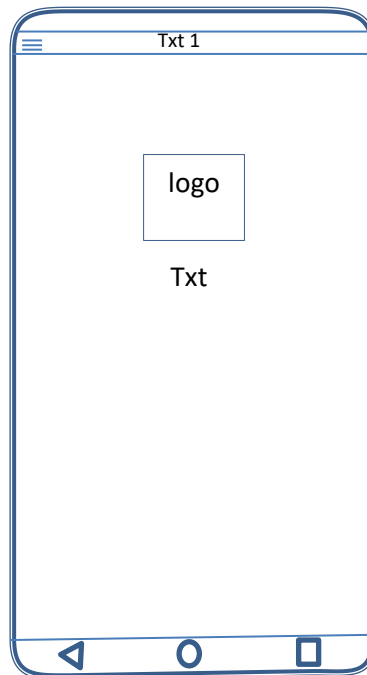


Figura 3.53. Diseño Estructura 1 – Pantalla de Bienvenida.

3.11.2.2. Estructura 2 - Pantalla de Menú

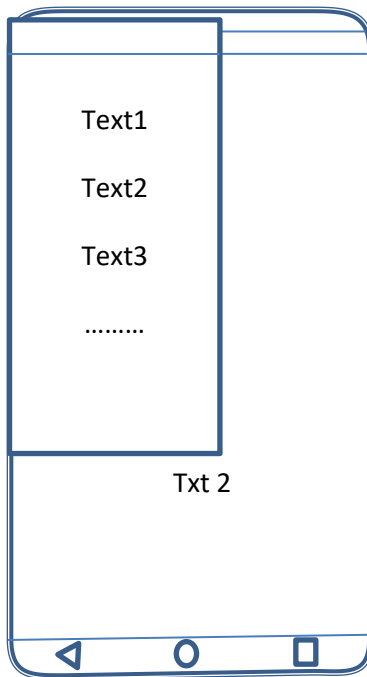


Figura 3.54: Diseño Estructura 2 – Pantalla de Menú.

3.11.2.3. Estructura 3 - Pantalla Listado de Participantes



Figura 3.55. Diseño Estructura 3 – Pantalla Listado de Participantes.

3.11.2.4. Estructura 4 - Pantalla de Registro del Participante

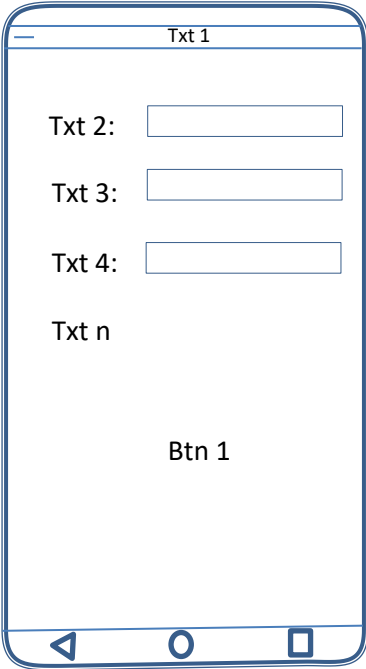


Figura 3.56. Diseño Estructura 4 – Pantalla de Registro del Participante.

3.11.2.5. Estructura 5 - Pantalla Modificar Datos del Participante

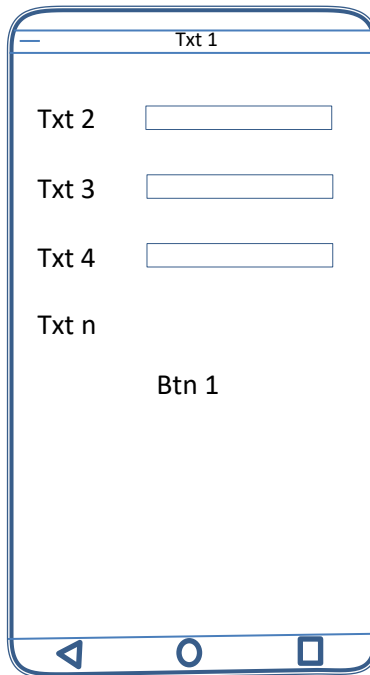


Figura 3.57. Diseño Estructura 5 – Pantalla Modificar Datos del Participante.

3.11.2.6. Estructura 6 - Pantalla Eliminar Datos del Participante

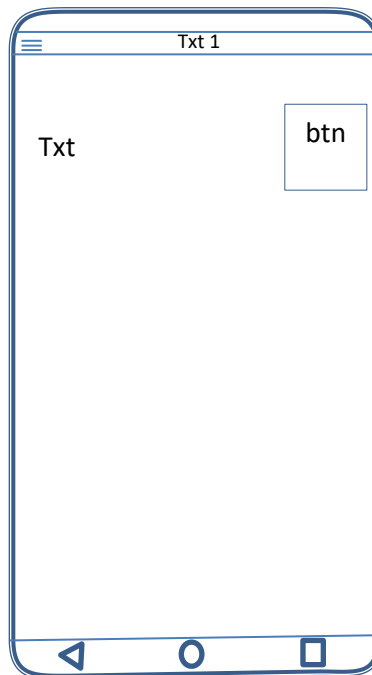


Figura 3.58. Diseño Estructura 6 - Pantalla Eliminar Datos del Participante.

3.11.2.7. Estructura 7 - Pantalla Listado para el Comprobante de Cobro

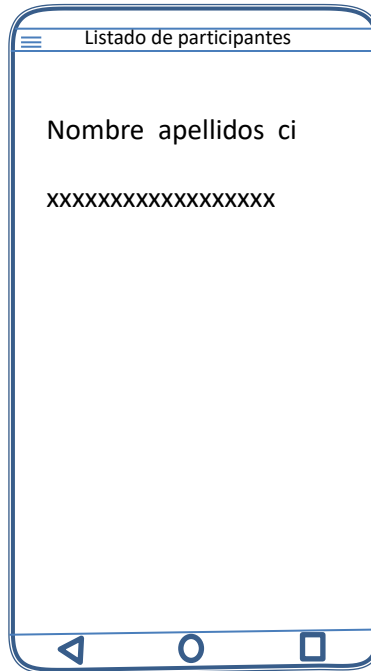


Figura 3.59. Diseño Estructura 7 – Pantalla Listado para el Comprobante de Cobro.

3.11.2.8 Estructura 8 - Pantalla Registro de Comprobante de Cobro

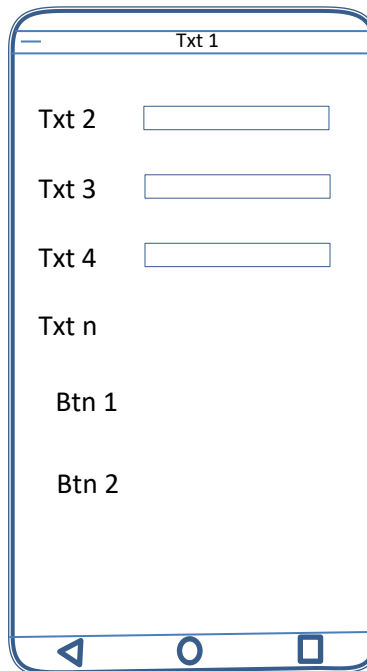


Figura 3.60. Diseño Estructura 8 - Pantalla Registro de Comprobante de Cobro.

3.11.3. Estructuras del Rol Participante

3.11.3.1. Estructura 1 - Bienvenida

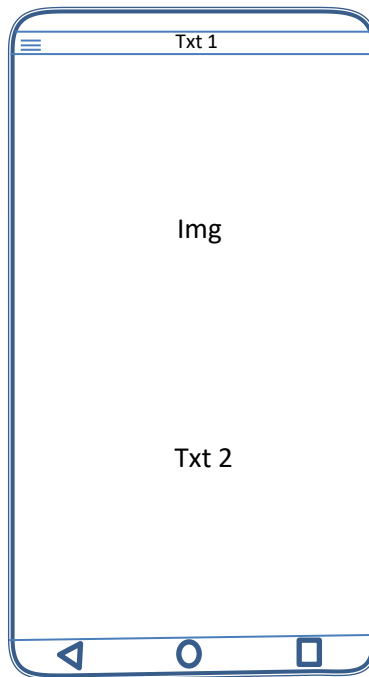


Figura 3.61. Diseño Estructura 1 - Bienvenida.

3.11.3.2. Estructura 2 - Menú Principal

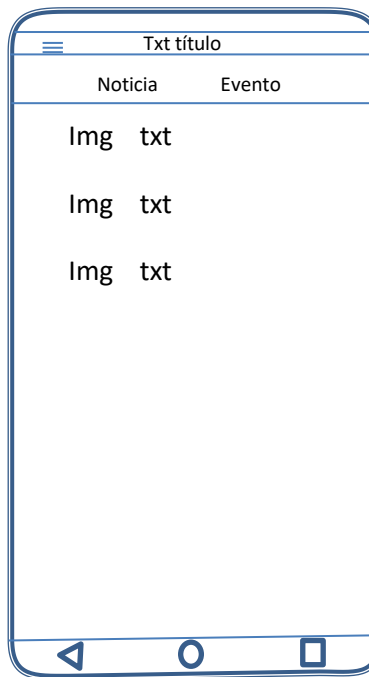


Figura 3.62. Diseño Estructura 2 - Menú Principal.

3.11.3.3. Estructura 3 - Detalle Eventos

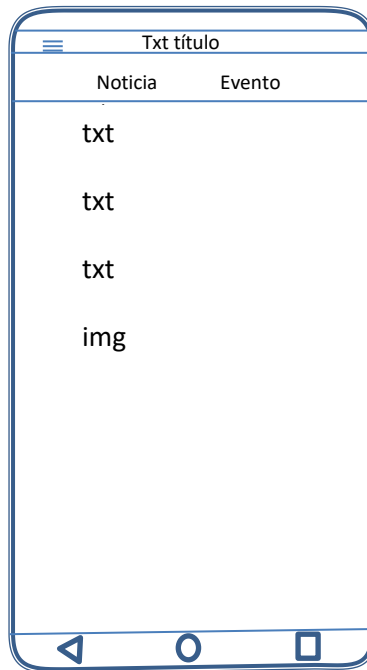


Figura 3.63. Diseño Estructura 3 - Detalle Eventos.

3.11.3.4. Estructura 4 - Áreas

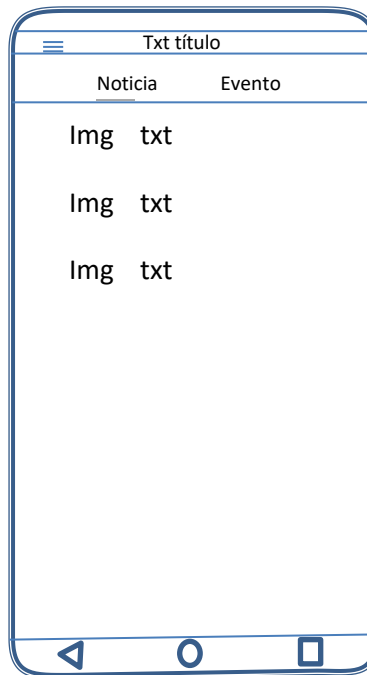


Figura 3.64. Diseño Estructura 4 – Áreas.

3.11.3.5. Estructura 5 - Detalle de Área

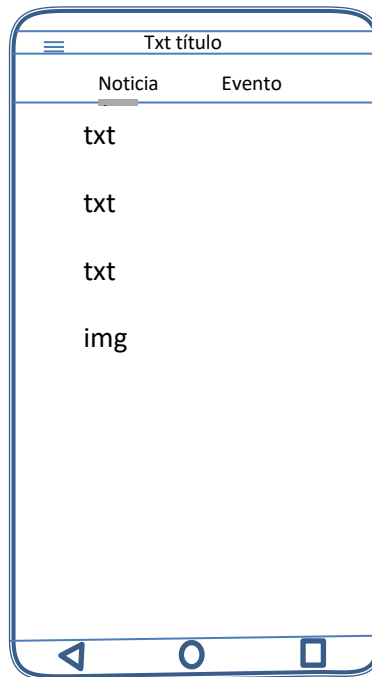


Figura 3.65. Diseño Estructura 5 – Detalle de Área.

3.11.3.6. Estructura 6 - Anuncio

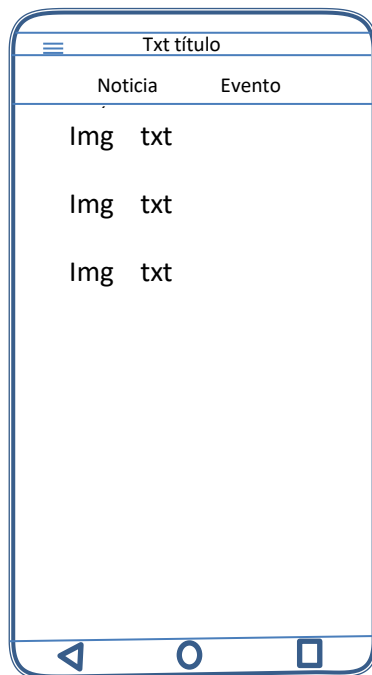


Figura 3.66. Diseño Estructura 6 - Anuncio.

3.11.3.7. Estructura 7 - Detalle Anuncio

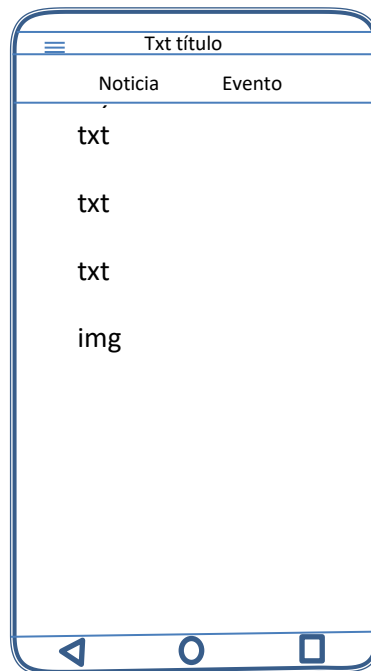


Figura 3.67. Diseño Estructura 7 – Detalle Anuncio.

3.11.3.8. Estructura 8 - Dirección

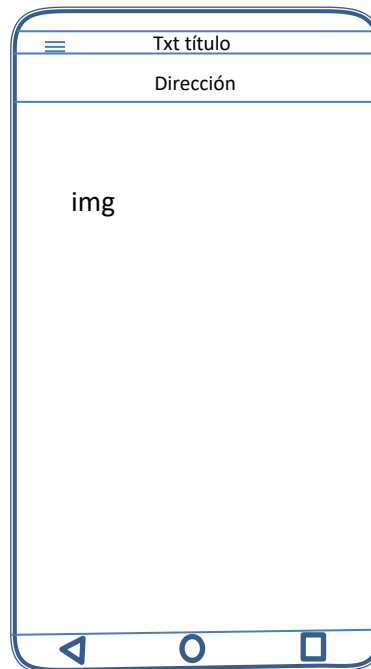


Figura 3.68. Diseño Estructura 8 – Dirección.

3.11.3.9. Estructura 9 - Menú desplegable

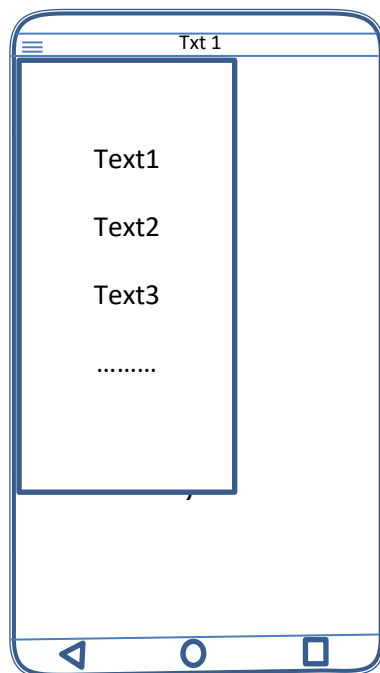


Figura 3.69. Diseño Estructura 9 - Menú Desplegable.

3.11.3.10. Estructura 10 - Galería de Fotos

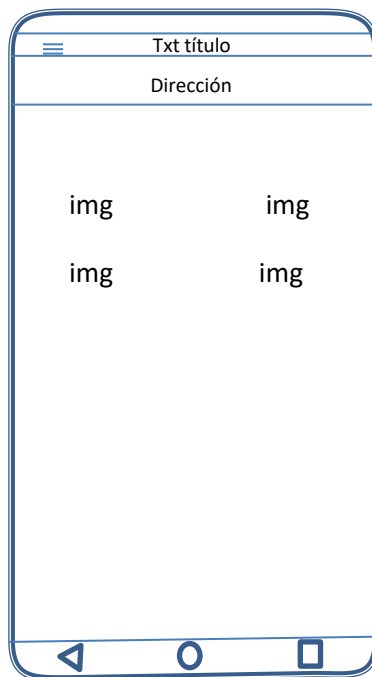


Figura 3.70. Diseño Estructura 10 – Galería de Fotos.

DISEÑO

4. DISEÑO

4.1. Base de Datos en Firebase

4.1.1. Colección Anuncio

eventosrobotica		anuncio		2RKqnbiQlbJnBGyLRBdB	
+ Iniciar colección		+ Añadir documento		+ Iniciar colección	
anuncio >		2RKqnbiQlbJnBGyLRBdB >		+ Añadir campo	
area		4VNQzs4WGceLqhKxJ0Q5		archivo: "saadfff"	
departamento		ef0EZUve0JIX0MsVazuU		descripcion: "Regla sorpresa: De la misma forma que en las competencias regionales, el día de la Final Nacional será anunciada una regla sorpresa para cada categoría, esta regla reemplazará una de las tareas existentes en cada uno de los retos, por lo que los equipos deberán resolverla para poder obtener el puntaje completo de su categoría.czxxcv 🚀"	
evento				fecha: "2020-09-02"	
organizador				imagen: "https://wro.org.mx/wp-content/uploads/2019/08/29.08.19-1.jpg"	
participante				nombre: "WRO Bolivia Regla sorpresa Nacional"	
tutor				userId: "bvBuS2XZohTKAtbqbpOgMV7TRx2"	
ubicar					
users					

Figura 4.1. Base de Datos – Colección Anuncio.

4.1.2. Colección Área

home > area > 3SItx0jm9GWm...		
eventosrobotica	area	3SItx0jm9GWmKpc9V6b8
+ Iniciar colección	+ Añadir documento	+ Iniciar colección
anuncio	3SItx0jm9GWmKpc9V6b8	+ Añadir campo
area	OAP15fMJucuSFe8fpI0j	archivo: "https://firebasestorage.googleapis.com/v0/b/eventosrobotica.app...alt=media&token=b3360ff8-0b47-498d-9bc2-3f8357c8d967"
departamento		detalle: "La robotica es importante en el desempeño de los circuitos"
evento		logo: "http://robomatrix.org/wp-content/uploads/2014/07/lineas-150x150.png"
organizador		nombre: "Segui Linea"
participante		userId: "bvBus2XZohTKAtbqppoQgMV7TRx2"
tutor		
ubicar		
users		

Figura 4.2. Base de Datos - Colección Área.

4.1.2. Colección Evento

home > evento > EWRbC4IdmYjs...		
eventosrobotica	evento	EWRbC4IdmYjsJ8psuWPG
+ Iniciar colección	+ Añadir documento	+ Iniciar colección
anuncio	EWRbC4IdmYjsJ8psuWPG	+ Añadir campo
area	tNoC4EWRbC4IdmYjsJ8psuWPG	actividad: "Realizar Video tutorial sobre el tema de configuracion"
departamento	vBqjneMqr0LEvd5GrRgI	archivo: "https://firebasestorage.googleapis.com/v0/b/eventosrobotica.e...alt=media&token=b3360ff8-0b47-498d-9bc2-3f8357c8d967"
evento		celular: "72943972"
organizador		direccion: "Av. Luis de Fuentes"
participante		fechafinal: "2020-03-16"
tutor		fechainicio: "2020-02-14"
ubicar		hora: "15:00"
users		informacion: "Campus Chiapas Carretera a Tapanatepec Km. 149 + 746 Col, Juan Crispín, 29020 Tuxtla Gutiérrez, Chis."
		logotipo: "https://wro.org.mx/wp-content/uploads/2019/05/27.05.19-4-2-732x415.jpg"
		nombre: "Competencia Regional Sede Tarija"
		organiza: "Bienestar y Consejería, LIFE, Tec de Monterrey"

Figura 4.3. Base de Datos - Colección Evento.

4.1.3. Colección Organizador (Persona)

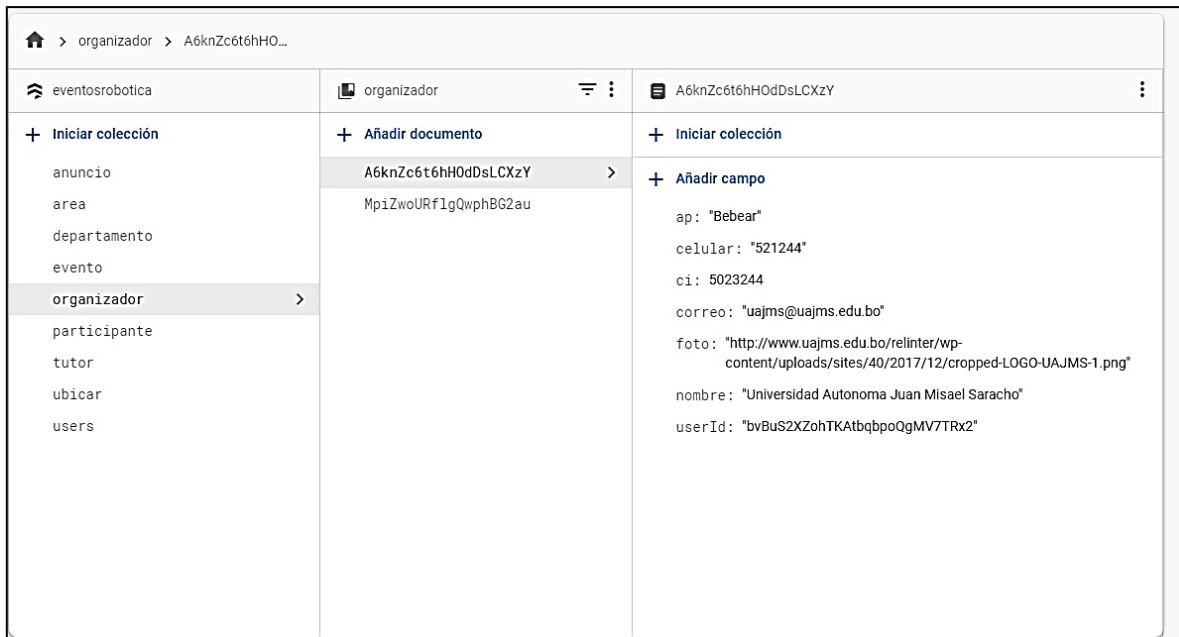


Figura 4.4. Base de Datos – Colección Organizador (Persona).

4.1.4. Colección Participante - Cobrar

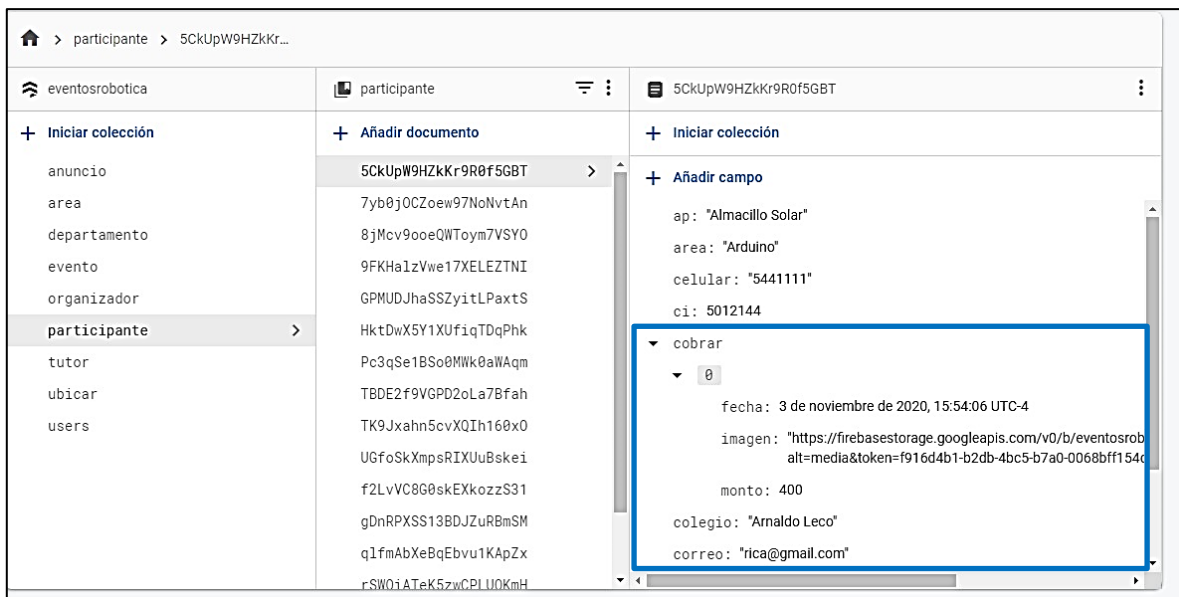


Figura 4.5. Base de Datos – Colección Participante - Cobrar.

4.1.5. Colección Tutor

61qPH486K0TV...		
eventosrobotica	tutor	61qPH486K0TV3sVQrcVMJ1uDy8G2
+ Iniciar colección	+ Añadir documento	+ Iniciar colección
anuncio	61qPH486K0TV3sVQrcVMJ1uDy8G2 >	+ Añadir campo
area	JYsfLGrp7uHTBYe1z7nb	ap: "Blacut"
departamento	KPzRPZ6YzQK4V9LNIKRN	celular: "123123"
evento	M3ymHVohgXY4Aw0LjuLkm7iBFBI3	ci: 5675757
organizador	TwJYHy1LPiUy9ZY9Vpi0	clave: "123456"
participante	nv5ZrW8G5ETDRyPB3MaiaiWqaZe2	correo: "rami@test.com"
tutor >		foto: "ewewrerw"
ubicar		nombre: "Ramiro"
users		tipo: "tutor"
		userId: "bvBuS2XZohTKAtqbpoQgMV7TRx2"

Figura 4.6. Base de Datos – Colección Tutor.

4.1.6. Colección Ubicar

Datos	Reglas	Índices	Uso
🏠 > ubicar > p5UpsQUS7pAB...			
🏠 eventosrobotica	📄 ubicar	☰ ⋮	📄 p5UpsQUS7pABEsShRLDh
+ Iniciar colección	+ Añadir documento		+ Iniciar colección
anuncio	p5UpsQUS7pABEsShRLDh >		+ Añadir campo
area			latitud: -21.541204
departamento			longitud: -64.714916
evento			mensaje: "evento luis"
organizador			posicion: false
participante			
tutor			
ubicar >			
users			

Figura 4.7. Base de Datos – Colección Ubicar.

4.2. Creación de Colecciones (Tablas en la Base de Datos)

Tabla 4.1

Anuncio - Área - Auspicio

```
export interface Anuncio {
  id?: string;
  descripcion?: string;
  imagen?: string;
  archivo?: string;
  fecha?:string;
  nombre?:string;
  userId?: string;

export interface Area {
  nombre?: string;
  detalle?: string;
  logo?: string;
  archivo?: string;
  userId?: string;
}

export interface Auspicio {
  nombre?: string;
  foto?:string;
  userId?: string;
}
```

Tabla 4.2

Evento - Persona

```
export interface Evento {
  id?: string;
  idper?:string;
  nombre?: string;
  fechainicio?: string;
  actividad?: string;
  logotipo?: string;
  direccion?: string;
  celular?: number;
  informacion?: string;
  fechafinal?: string;
  fechainicio?: string;
  hora?:string;
  userId?: string;
}
export interface persona {
  id?: string;
  nombre?: string;
  ap?: string;
  ci?: string;
  correo?:string;
  celular?:string;
  foto?:string;
  fecha?: number;
  userId?: string;
}
```

Tabla 4.3

Cobrar – Participante - Dirección

```
export interface Cobrar {
  mes?: string;
  gestion?: number;
  monto?: number;
  imagen?:string;
  fecha?: Date;
}

export interface Participante {
  id?: string;
  area?:string;
  depto?:string;
  nombre?: string;
  ap?: string;
  ci?: string;
  correo?:string;
  imgcomprobante?:string;
  userId?: string;
}

export interface direccion {
  mensaje?:string;
  latitud?:number;
  longitud?:number;
  posicion?:boolean;
}
```

4.3. Mapa Navegacional Módulo Administrador

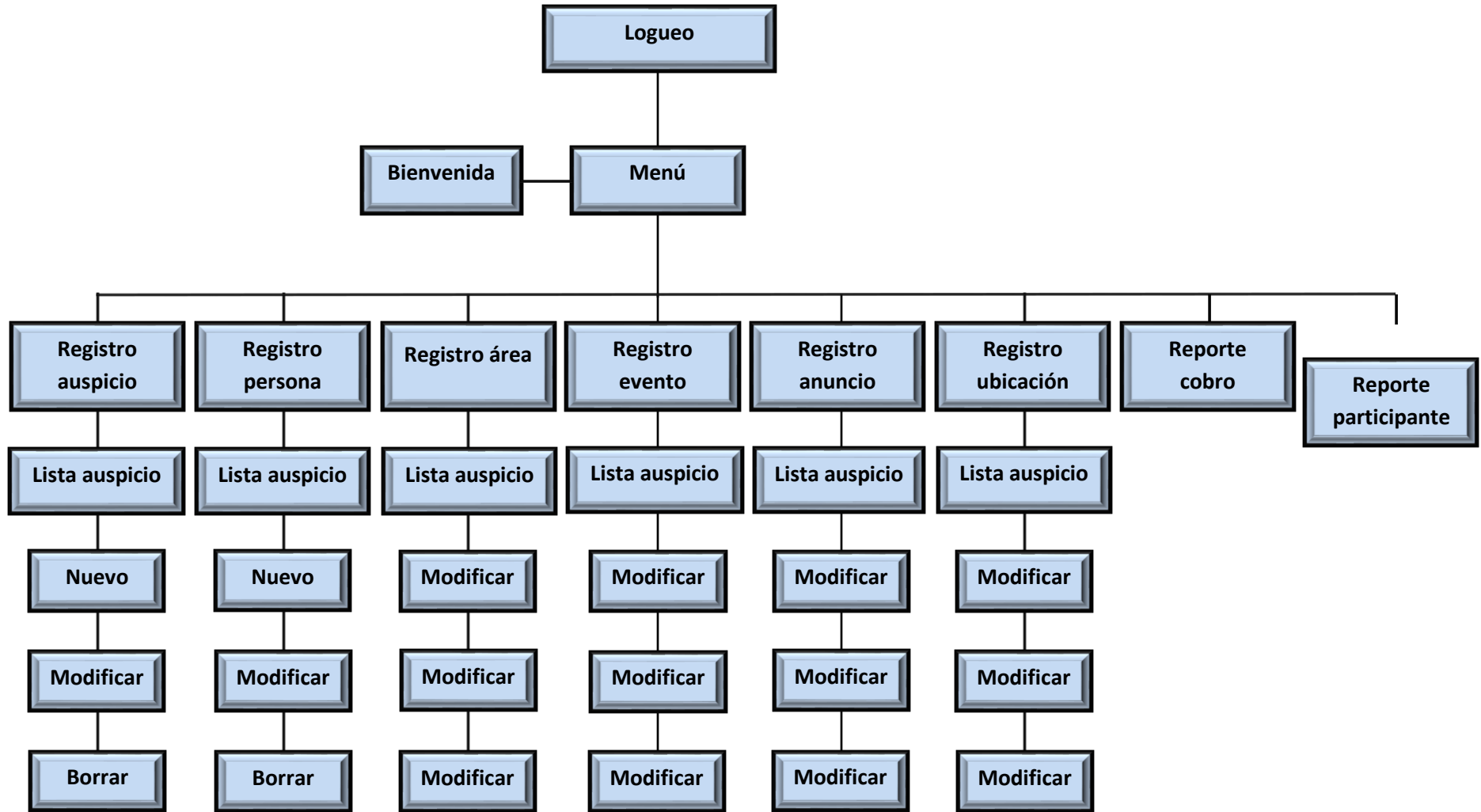


Figura 4.8. Mapa Navegacional Módulo Administrador.

4.4. Mapa Navegacional Módulo Tutor

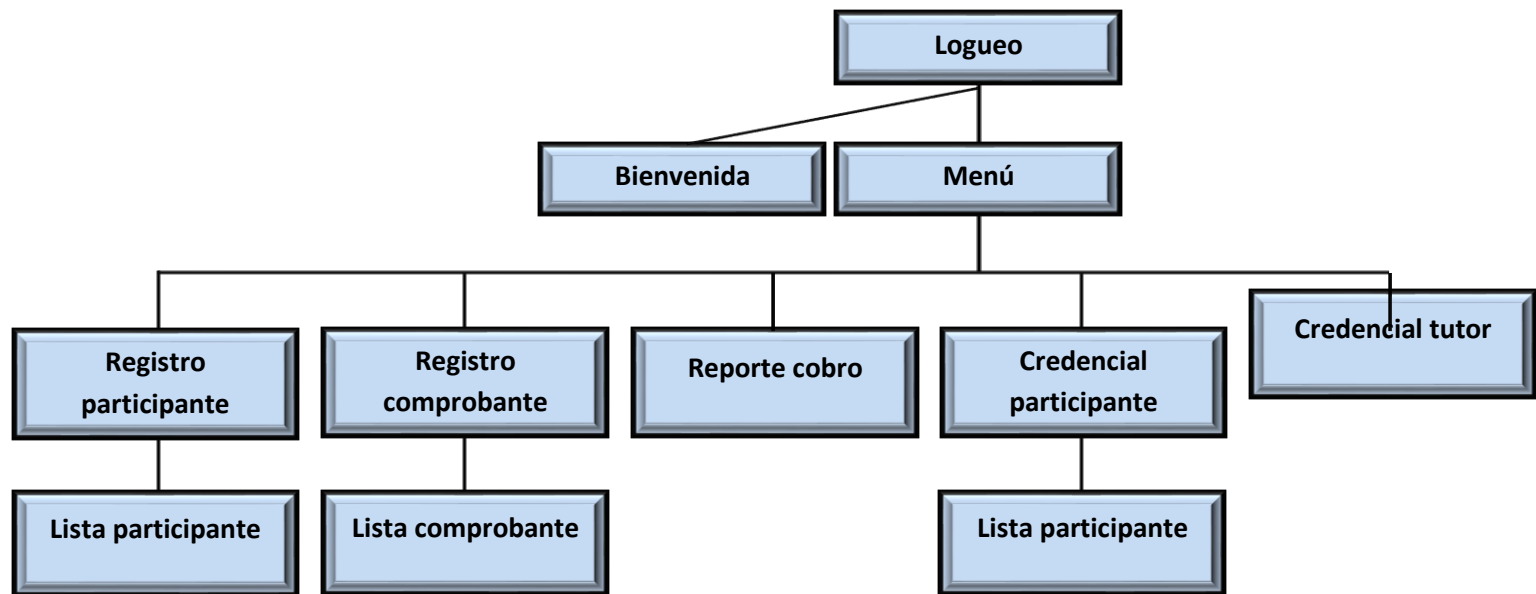


Figura 4.9. Mapa Navegacional Módulo Tutor.

4.5. Mapa Navegacional Módulo Participante

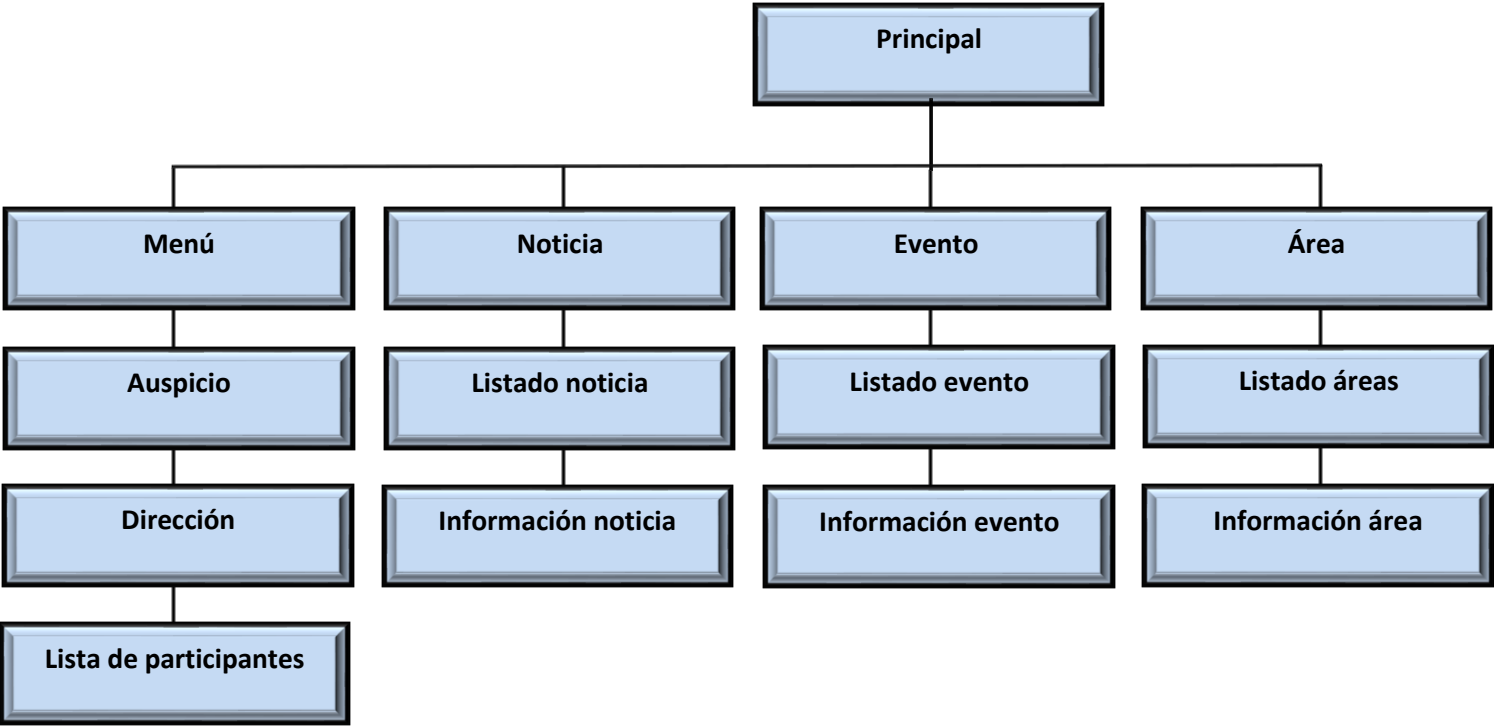


Figura 4.10. Mapa Navegacional Módulo Participante.

4.6. Pantallas Rol Administrador

4.6.1. Pantalla 1 - Logueo

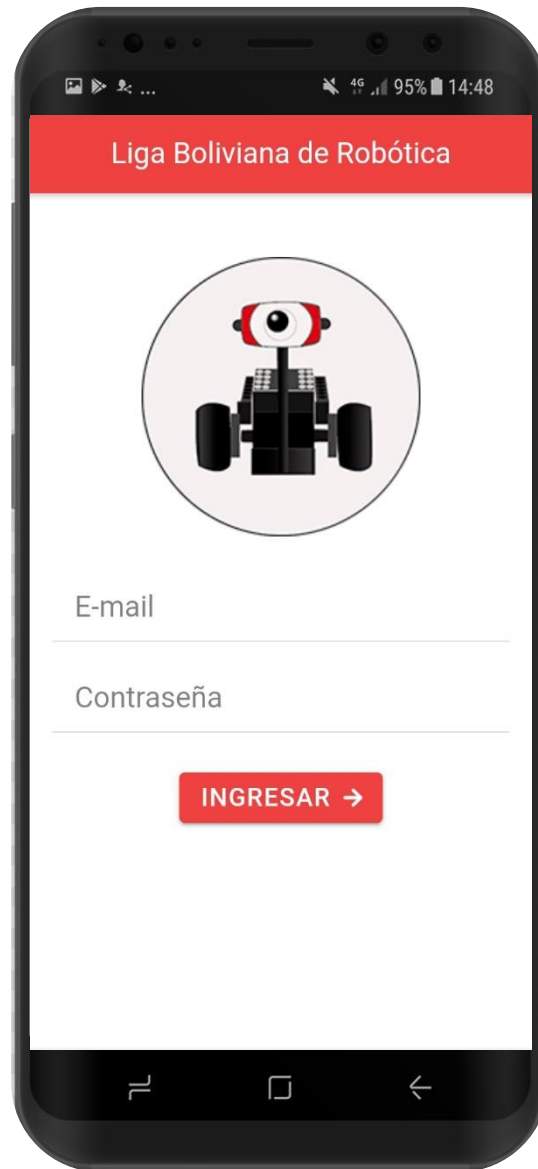


Figura 4.11. Rol Administrador - Pantalla 1- Logueo.

En esta pantalla el administrador ingresa los datos de usuario y clave.

4.6.2. Pantalla 2 - Bienvenida



Figura 4.12. Rol Administrador - Pantalla 2 - Bienvenida.

Esta es la pantalla de bienvenida una vez que se ingresó el usuario y contraseña.

En la parte superior a lado izquierdo se muestra un botón de menú.

4.6.3. Pantalla 3 - Menú General

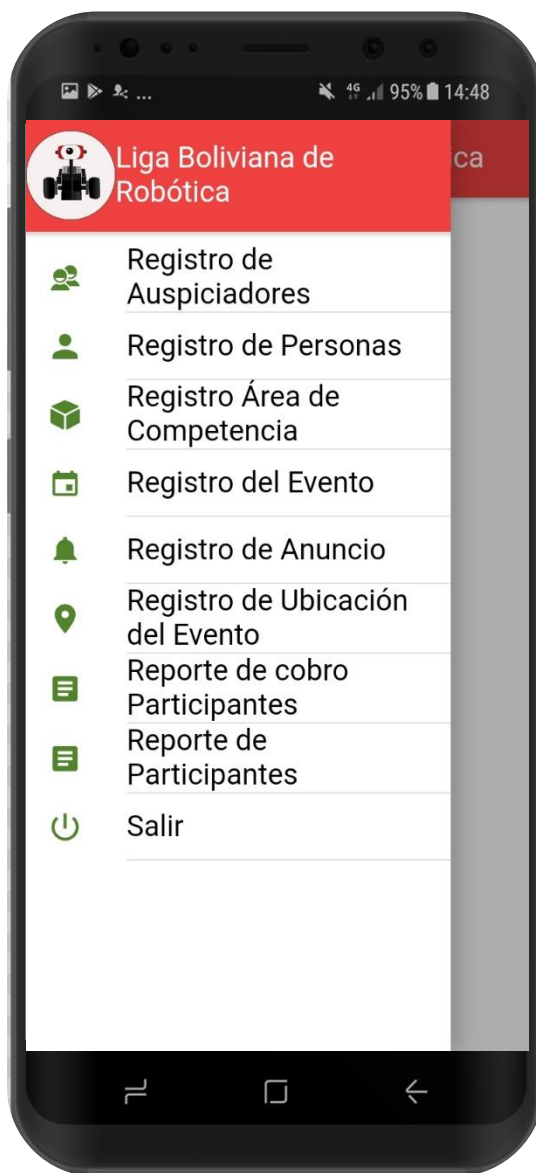


Figura 4.13. Rol Administrador – Pantalla 3 - Menú General.

En esta pantalla se muestra el menú que tiene la aplicación para acceder a cualquiera de las opciones que requiera el usuario.

4.6.4. Pantalla 4 - Listado de Personas

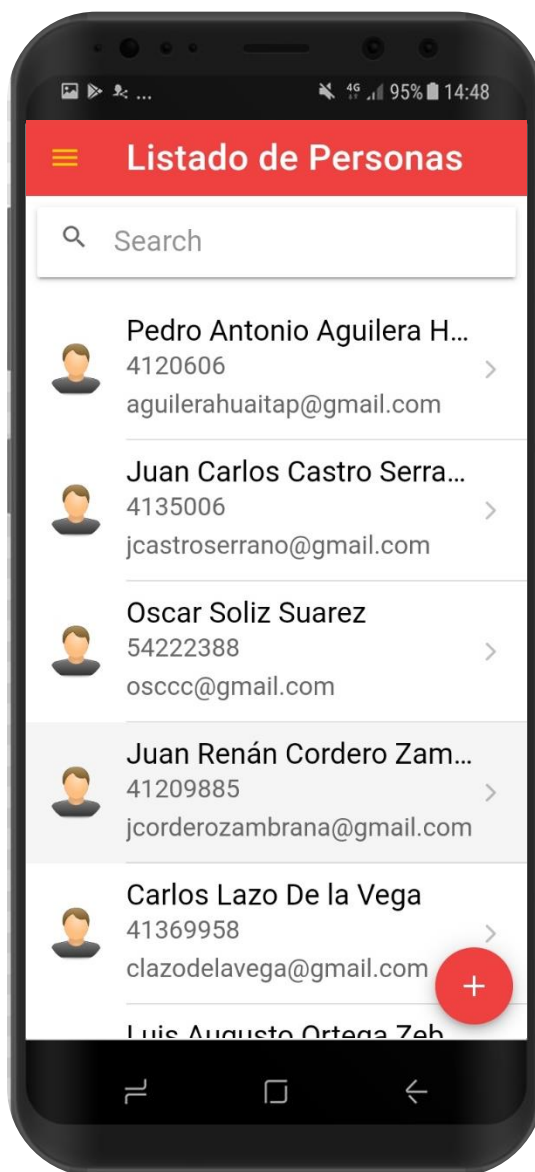


Figura 4.14. Rol Administrador - Pantalla 4 - Listado de Personas.

En esta pantalla se muestra el listado de las personas que tendrán los roles de administradores o tutores de equipos. En la parte superior se tiene la opción de un buscador, en la inferior un botón par nuevo registro, y haciendo click en la lista tenemos la opción de modificar y deslizando hacia la izquierda con click sostenido, tenemos la opción de borrar.

4.6.5. Pantalla 5 - Nueva Persona

The screenshot shows a mobile application interface for adding a new person. The screen is titled "Persona" and features a red header bar with a back arrow on the left. Below the header, there are several input fields for user information: "Cedula de Identidad:", "Nombres:", "Apellidos:", "Celular:", "Tipo de Usuario" (with a dropdown arrow), "Usuario(E-mail):", and "Clave:". At the bottom of the form, there is a red button with a plus sign and the text "ACEPTAR". The status bar at the top shows the time as 14:48, 95% battery, and 4G signal.

Figura 4.15. Rol Administrador – Pantalla 5 - Nueva Persona.

En esta pantalla registramos los nuevos datos de la persona con la opción si es administrador o tutor.

4.6.6. Pantalla 6 - Modificar Persona

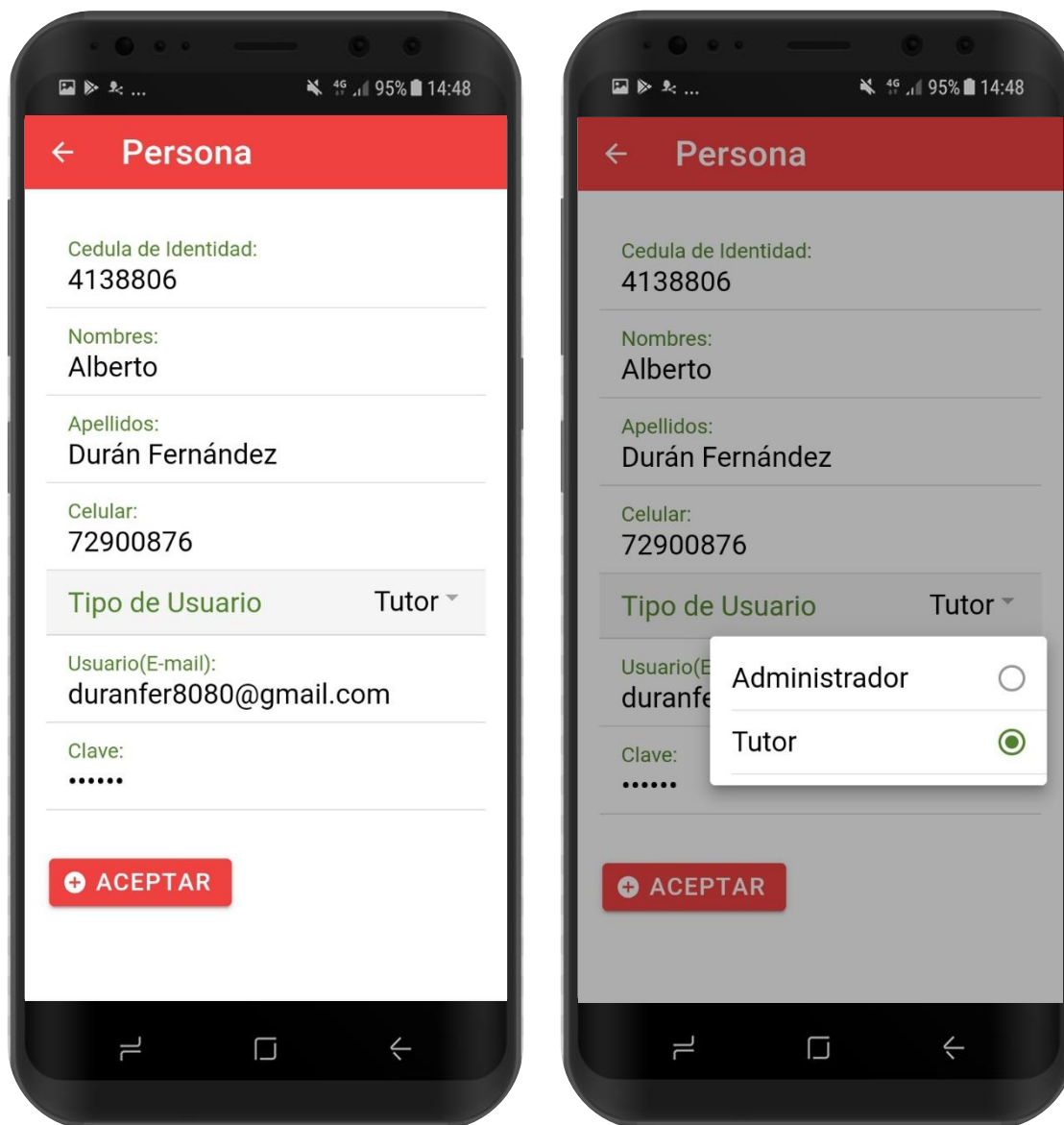


Figura 4.16. Rol Administrador - Pantalla 6 - Modificar Persona.

En esta pantalla podemos modificar los datos de las personas por el administrador.

4.6.7. Pantalla 7 - Eliminar Persona

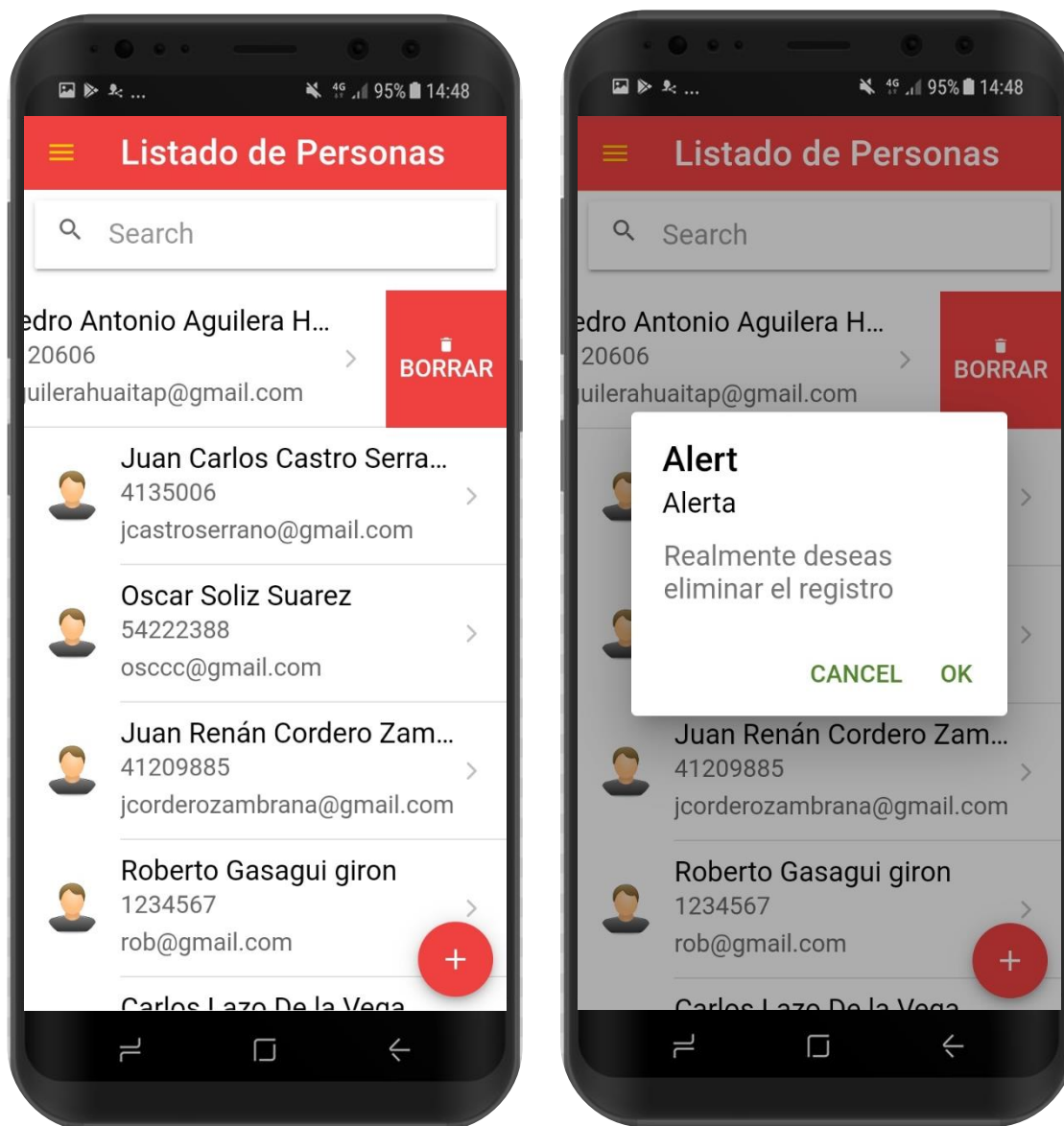


Figura 4.17. Rol Administrador - Pantalla 7 - Eliminar Persona.

En esta pantalla se permite el borrado de los datos de una persona al presionarse el texto y deslizar el cuadro hacia la izquierda. Al presionar el botón BORRAR aparece un mensaje de confirmación por seguridad de la información.

4.6.8. Pantalla 8 - Listado de Auspiciadores

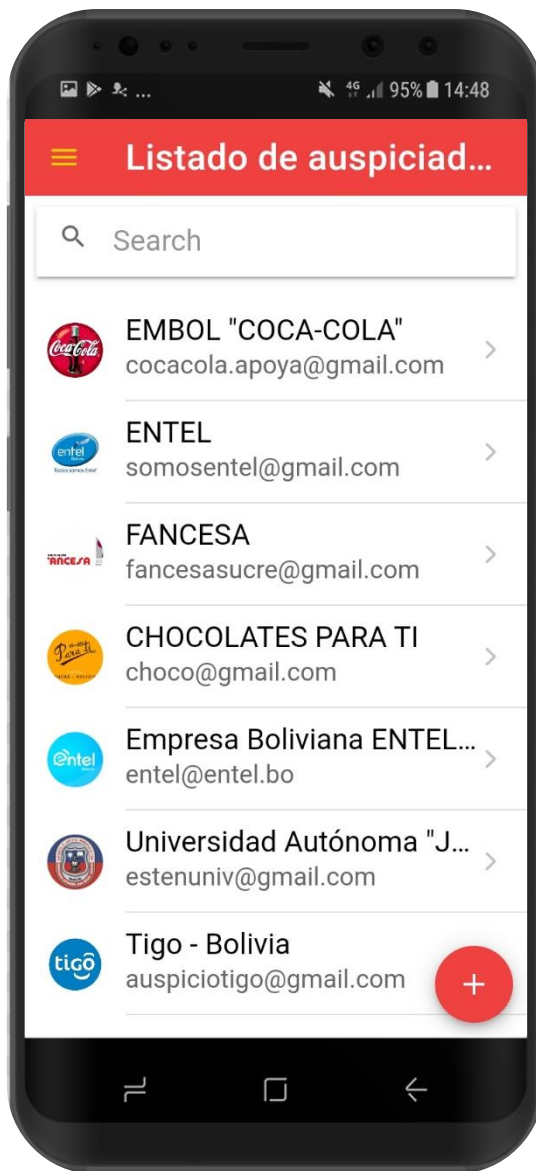


Figura 4.18. Rol Administrador - Pantalla 8 - Listado de Auspiciadores.

En esta pantalla se muestra el listado de auspiciadores y tiene la opción de un buscador. En la parte inferior existe un botón para realizar un nuevo registro y haciendo click en la lista tenemos la opción de modificar. Al deslizar hacia la izquierda con click sostenido, tenemos la opción de borrar.

4.6.9. Pantalla 9 - Nuevo Auspiciador



Figura 4.19. Rol Administrador - Pantalla 9 - Nuevo Auspiciador.

En esta pantalla se permite el registro de los nuevos datos de un auspiciador.

4.6.10. Pantalla 10 - Modificar Auspicio

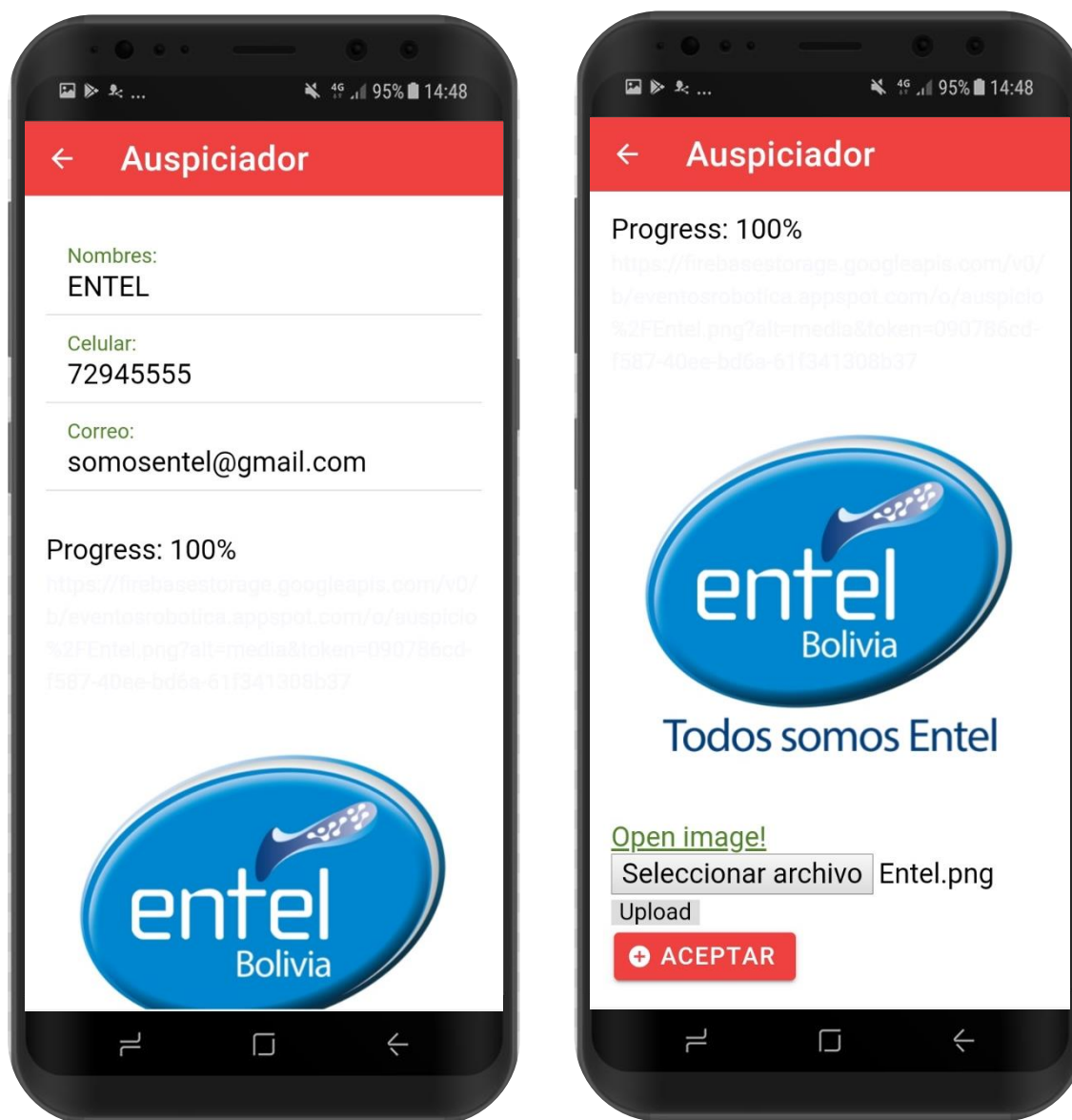


Figura 4.20. Rol Administrador - Pantalla 10 - Modificar Auspicio.

En esta pantalla se permite la modificación de los datos del auspiciador registrado.

4.6.11. Pantalla 11 - Borrar Auspicio

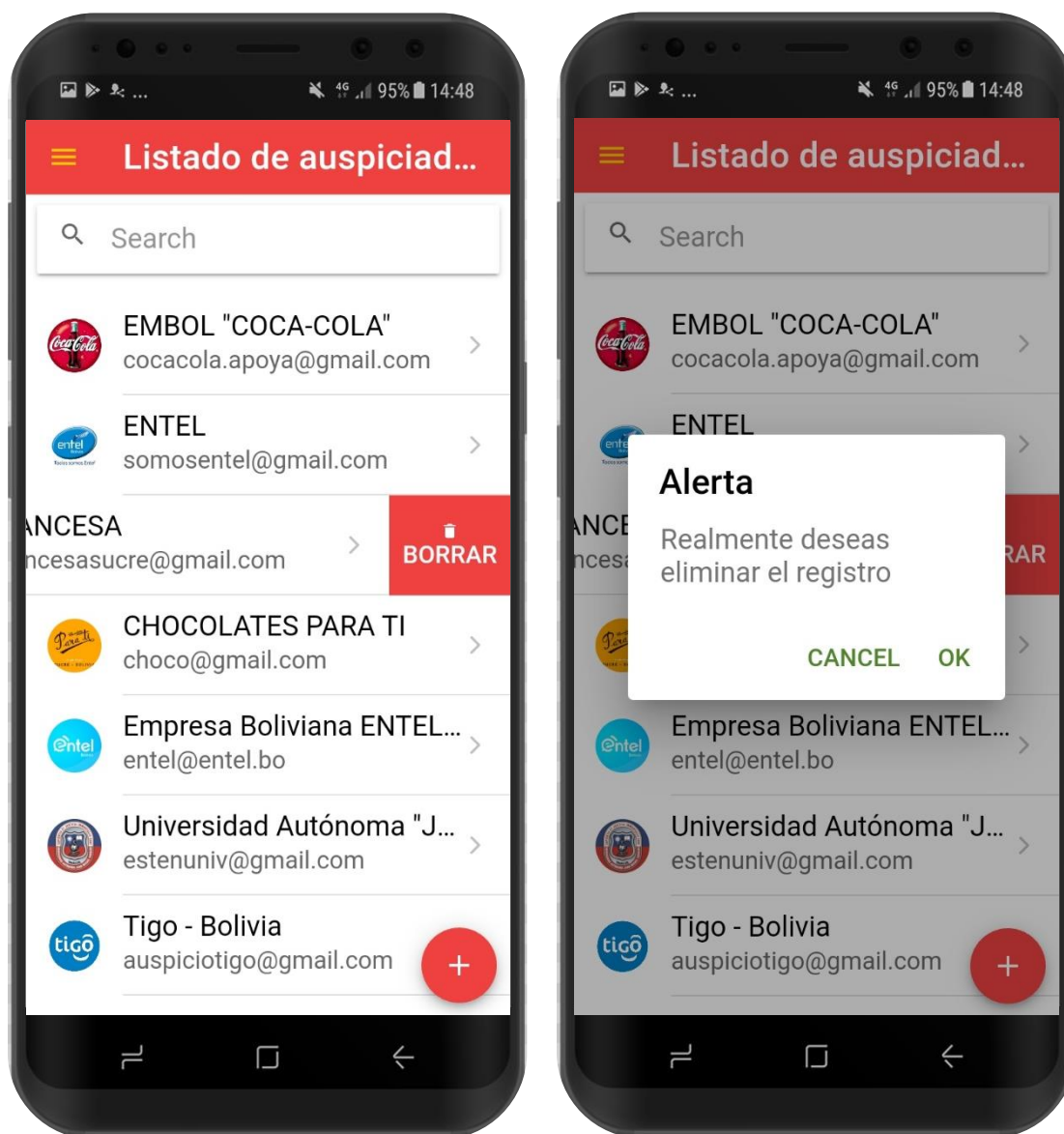


Figura 4.21. Rol Administrador - Pantalla11- Borrar Auspicio.

En esta pantalla se permite el borrado de los datos de un auspiciador al presionarse el texto y deslizar el cuadro hacia la izquierda. Al presionar el botón BORRAR aparece un mensaje de confirmación por seguridad de la información.

4.6.12. Pantalla 12 - Listado Área

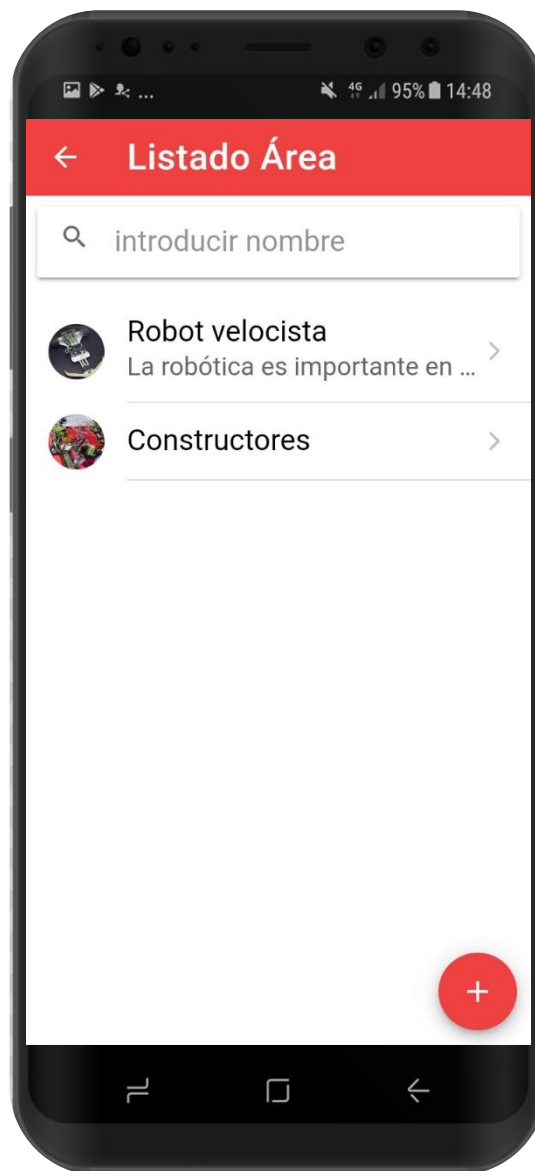


Figura 4.22. Rol Administrador - Pantalla 12 - Listado Área.

En esta pantalla se muestra el listado de áreas y tiene la opción de un buscador. En la parte inferior derecha se muestra un botón para un nuevo registro y haciendo click en la lista tenemos la opción de modificar. Al deslizar hacia la izquierda con click sostenido tenemos la opción de borrar.

4.6.13. Pantalla 13 - Nueva Área



Figura 4.23. Rol Administrador - Pantalla 13 - Nueva Área.

En esta pantalla se permite un nuevo registro de área, ingresando un afiche y documento de especificaciones para los participantes.

4.6.14. Pantalla 14 - Modificar Área

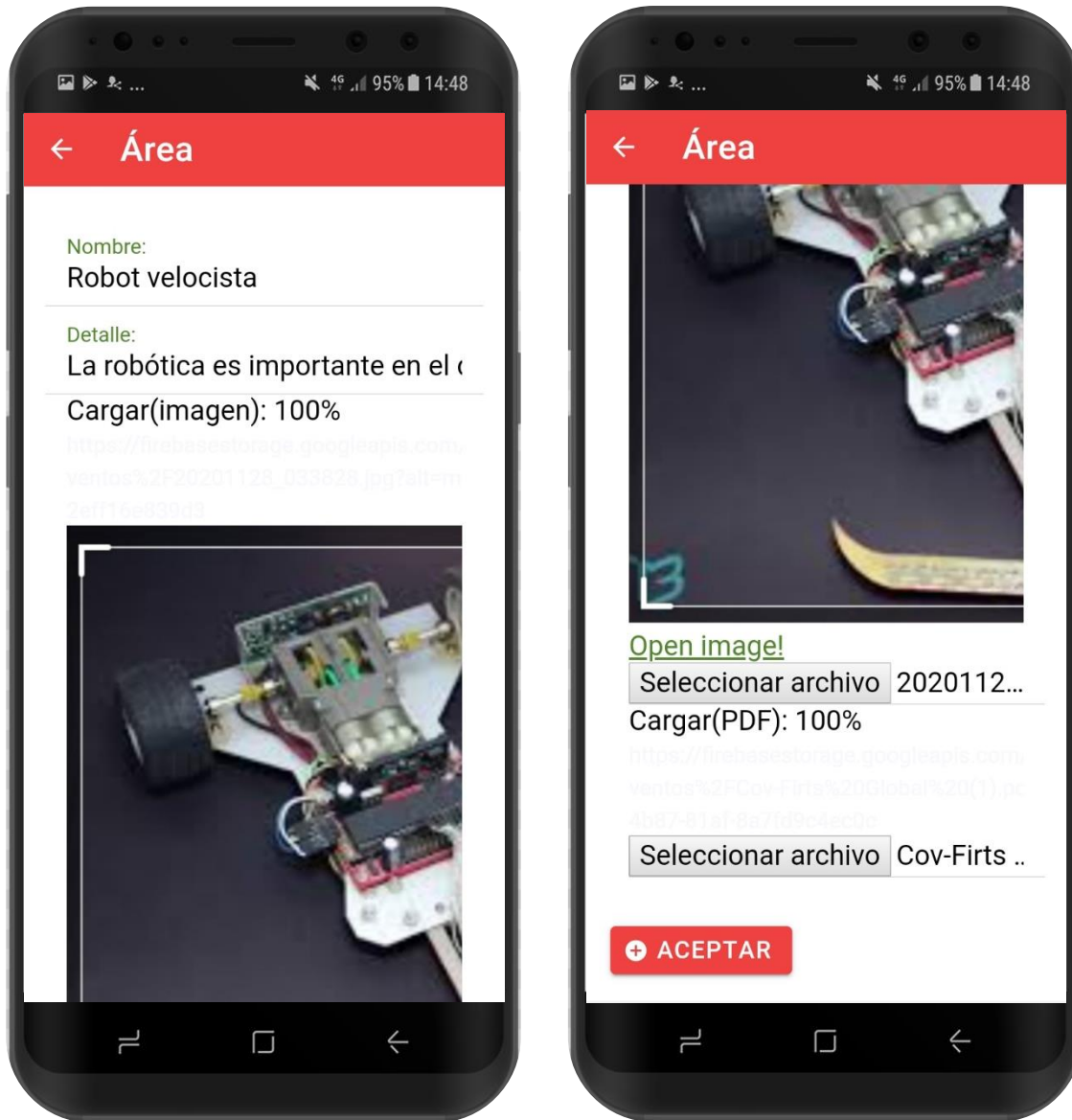


Figura 4.24. Rol Administrador - Pantalla 14 - Modificar Área.

En esta pantalla se permite la modificación de los datos del área, siendo necesario cargar el afiche de referencia y el documento de especificaciones para la competencia.

4.6.15. Pantalla 15 - Borrar Área

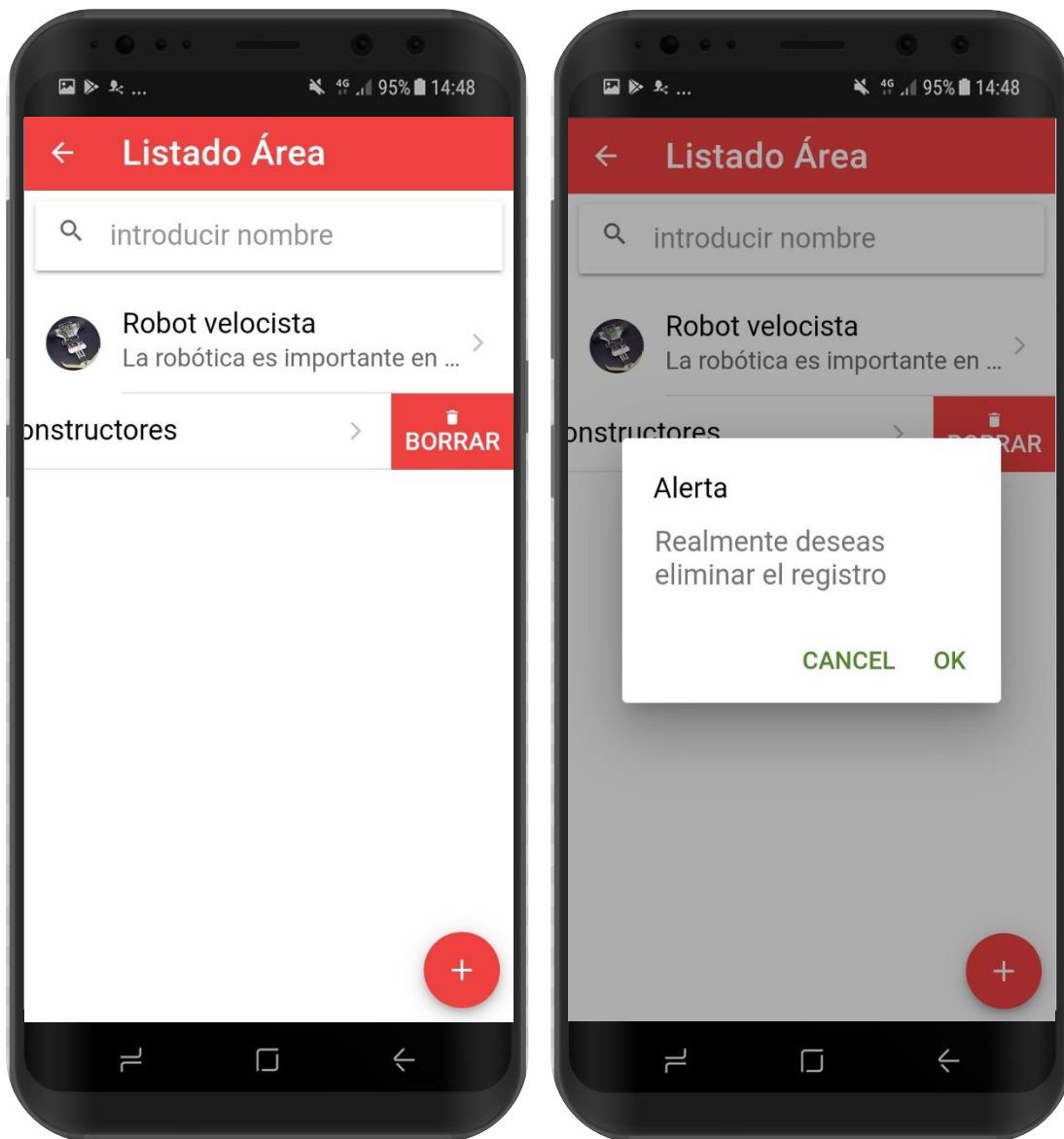


Figura 4.25. Rol Administrador - Pantalla 15 - Borrar Área.

En esta pantalla se permite el borrado de los datos de un área al presionarse el texto y deslizar el cuadro hacia la izquierda. Al presionar el botón BORRAR aparece un mensaje de confirmación por seguridad de la información.

4.6.16 Pantalla 16 - Listado de Eventos

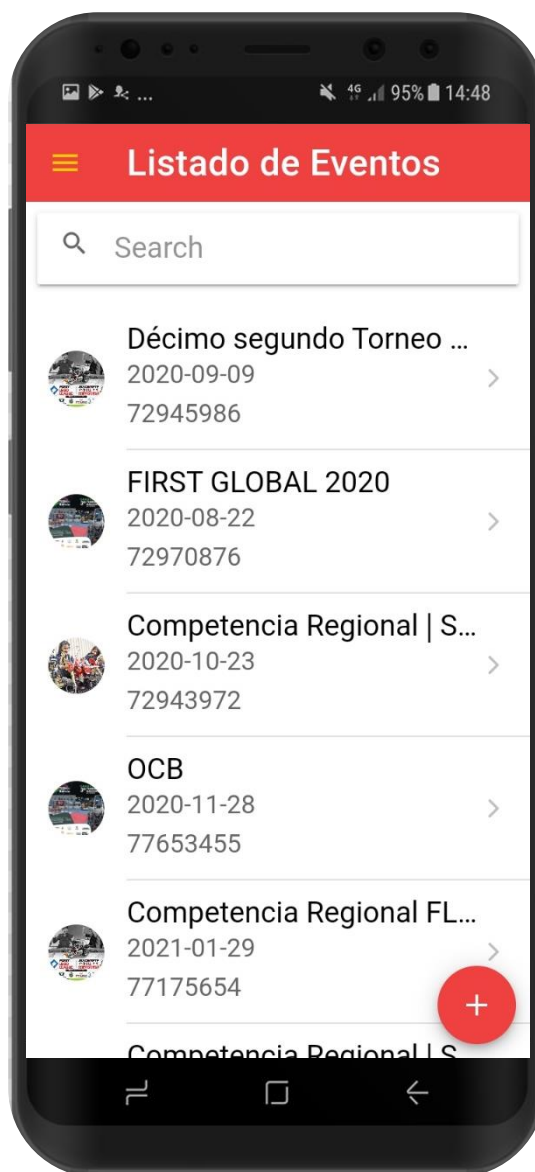


Figura 4.26. Rol Administrador - Pantalla 16 - Listado de Eventos.

En esta pantalla se muestra el listado de eventos y tiene la opción de un buscador. En la parte inferior existe un botón para hacer un nuevo registro y haciendo click en la lista tenemos la opción de modificar. Deslizando hacia la izquierda con click sostenido tenemos la opción de borrar.

4.6.17. Pantalla 17 - Nuevo Evento

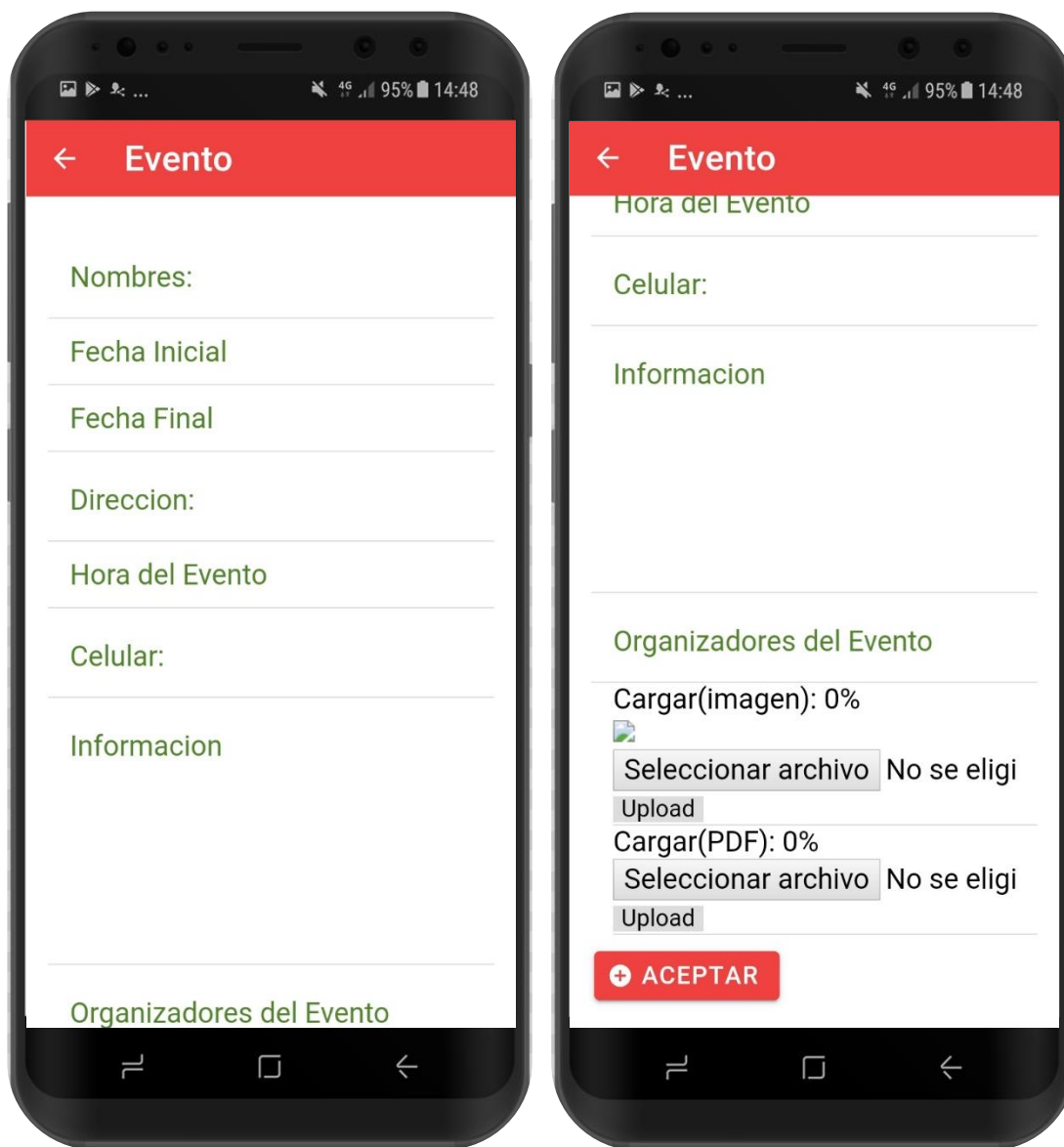


Figura 4.27. Rol Administrador - Pantalla 17 - Nuevo Evento.

En esta pantalla se permite el registro de nuevo de datos de un evento. En esta pantalla se podrá enviar el afiche de evento y el documento de la convocatoria.

4.6.18. Pantalla 18 - Modificar Evento

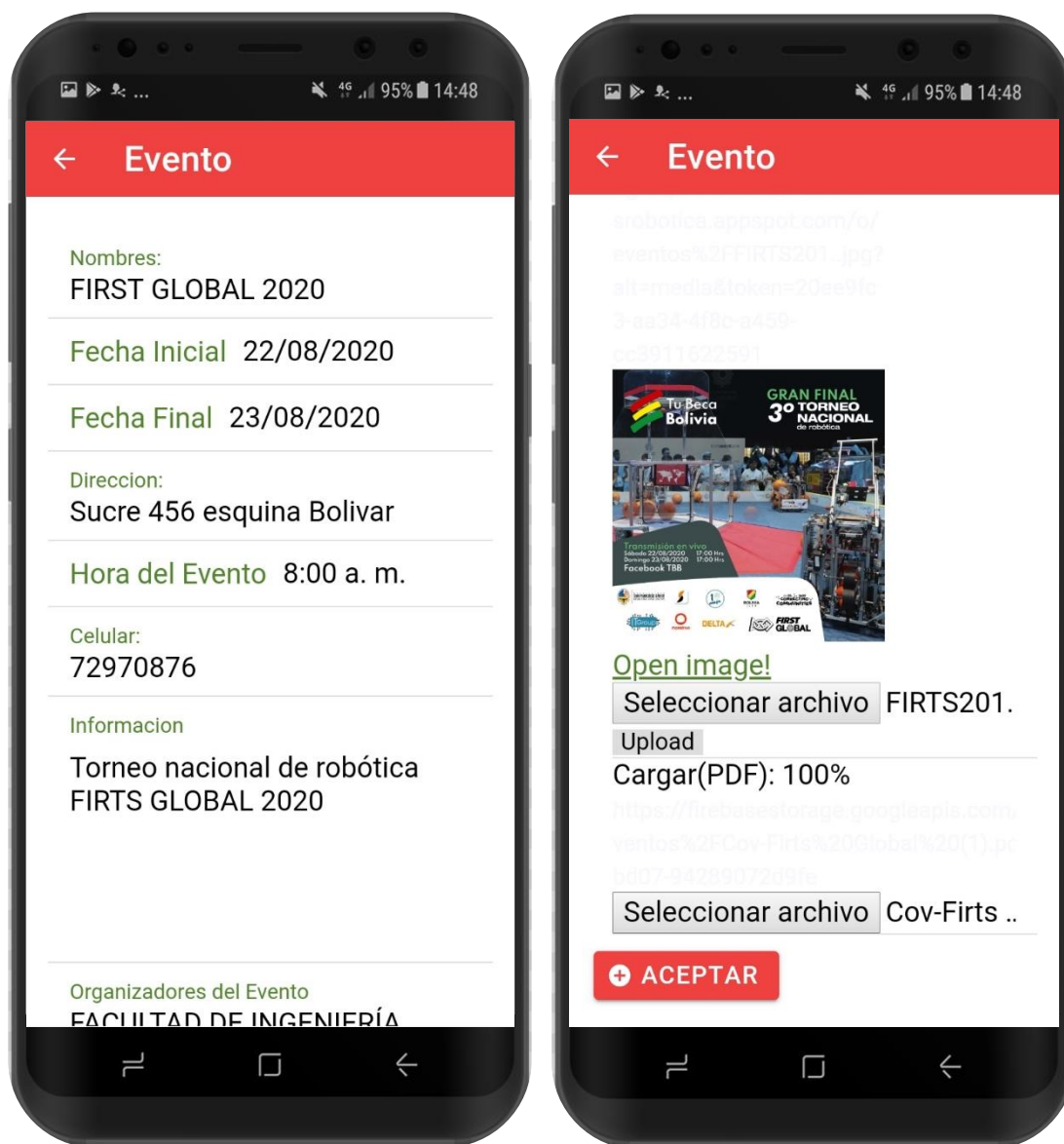


Figura 4.28. Rol Administrador - Pantalla 18 - Modificar Evento.

En esta pantalla se permite la modificación de los datos del evento. En esta pantalla se actualiza el afiche y documento de la convocatoria.

4.6.19. Pantalla 19 - Borrar Evento

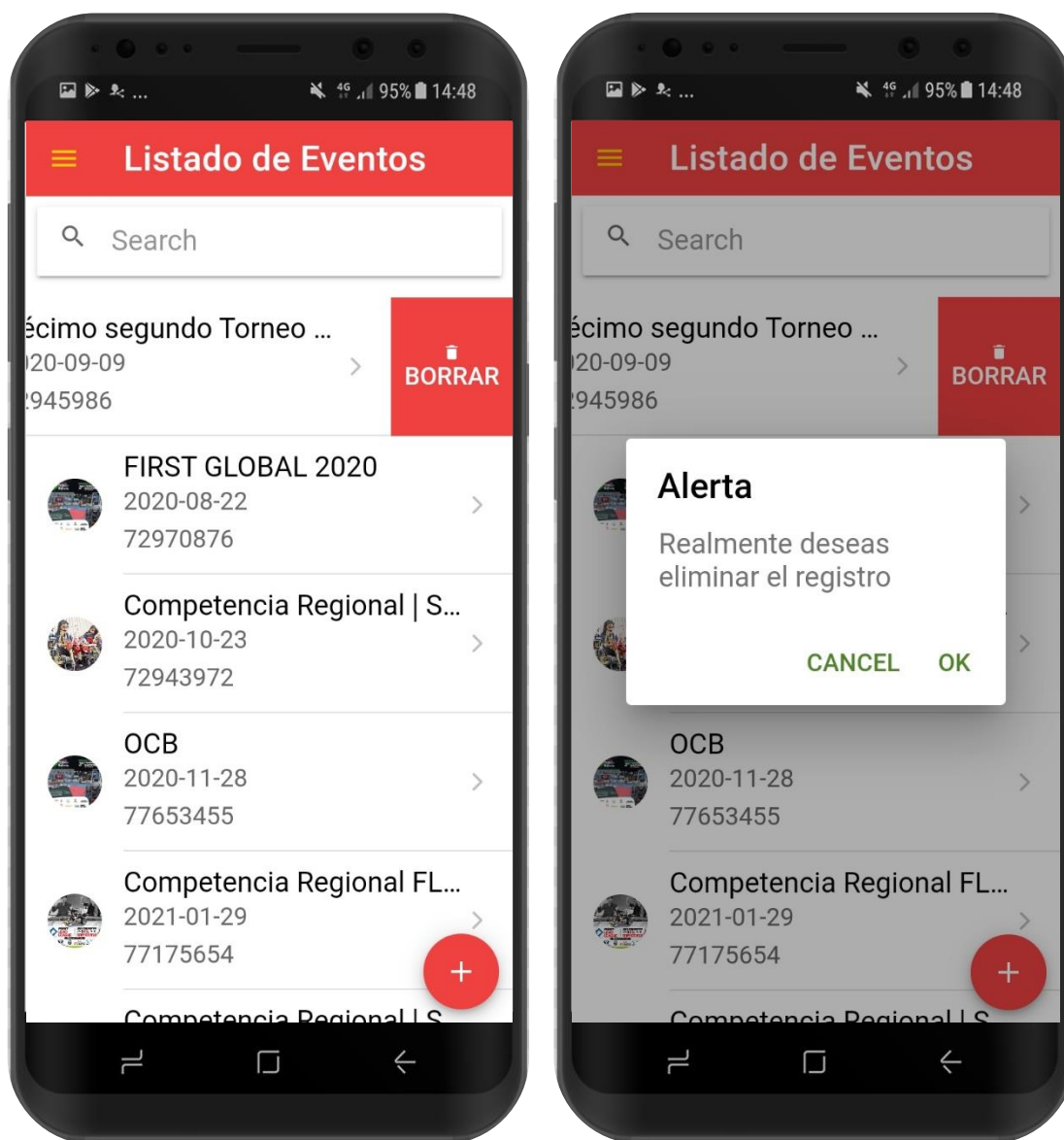


Figura 4.29. Rol Administrador - Pantalla 19 - Borrar Evento.

En esta pantalla se permite el borrado de los datos de un evento al presionarse el texto y deslizar el cuadro hacia la izquierda. Al presionar el botón BORRAR aparece un mensaje de confirmación por seguridad de la información.

4.6.20. Pantalla 20 - Listado de Anuncios



Figura 4.30. Rol Administrador - Pantalla 20 - Listado de Anuncios.

En esta pantalla se muestra el listado de anuncios y tiene la opción de un buscador. En la parte inferior se tiene un botón para agregar un nuevo registro y haciendo click en la lista tenemos la opción de modificar. Al deslizar hacia la izquierda con un click sostenido, tenemos la opción de borrar.

4.6.21. Pantalla 21 - Nuevo Anuncio

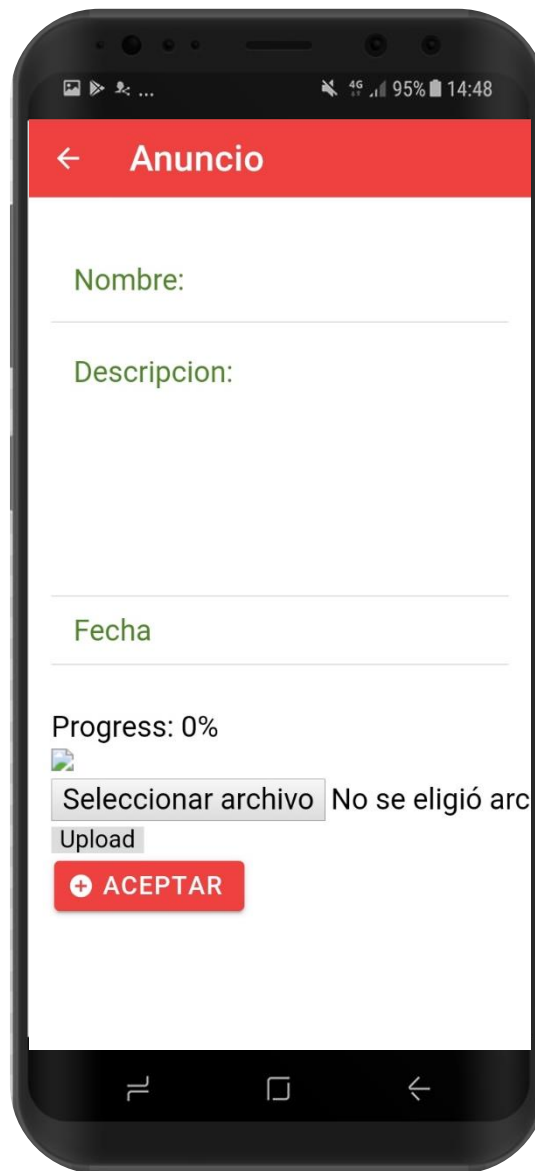


Figura 4.31. Rol Administrador - Pantalla 21 - Nuevo Anuncio.

En esta pantalla se permite realizar un nuevo registro de anuncio. Se cuenta con un buscador de archivos para enviar un afiche.

4.6.22. Pantalla 22 - Modificar Anuncio

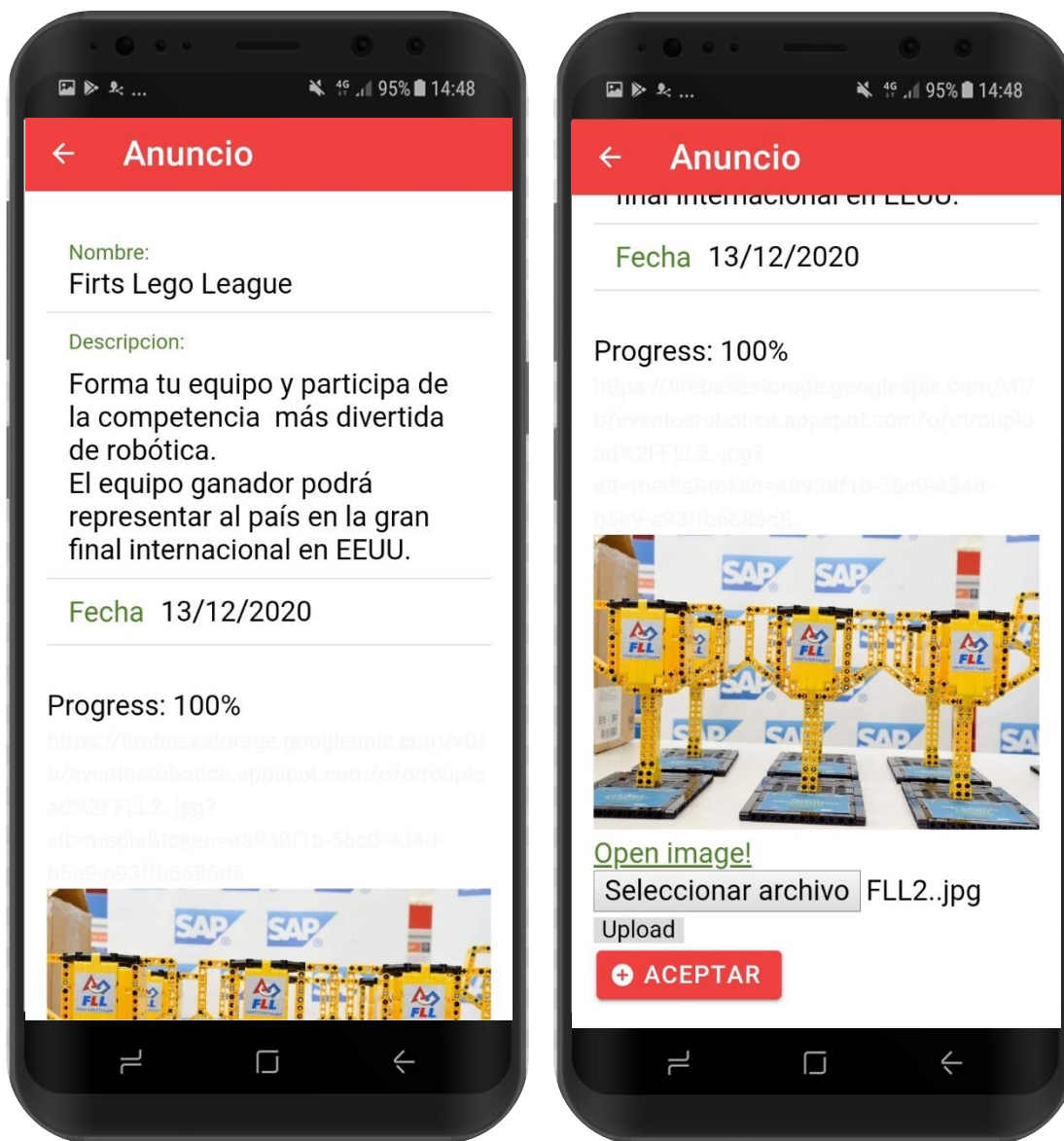


Figura 4.32. Rol Administrador - Pantalla 22 - Modificar Anuncio.

En esta pantalla se permite la modificación de los datos de anuncios.

4.6.23. Pantalla 23 - Borrar Anuncio

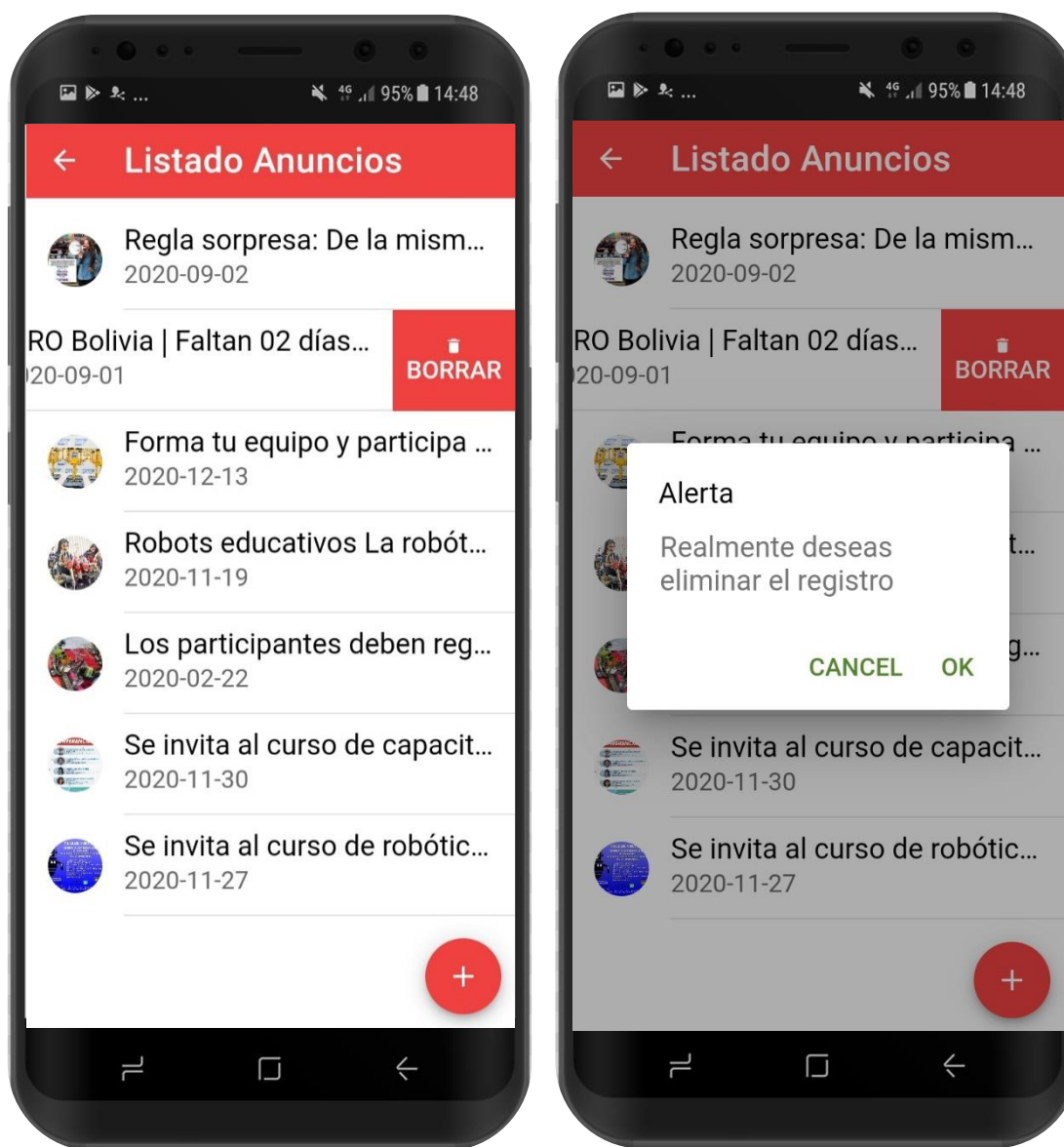


Figura 4.33. Rol Administrador - Pantalla 23 - Borrar Anuncio.

En esta pantalla se permite el borrado de los datos de un anuncio al presionarse el texto y deslizar el cuadro hacia la izquierda. Al presionar el botón BORRAR aparece un mensaje de confirmación por seguridad de la información.

4.6.24. Pantalla 24 - Listado Ubicaciones

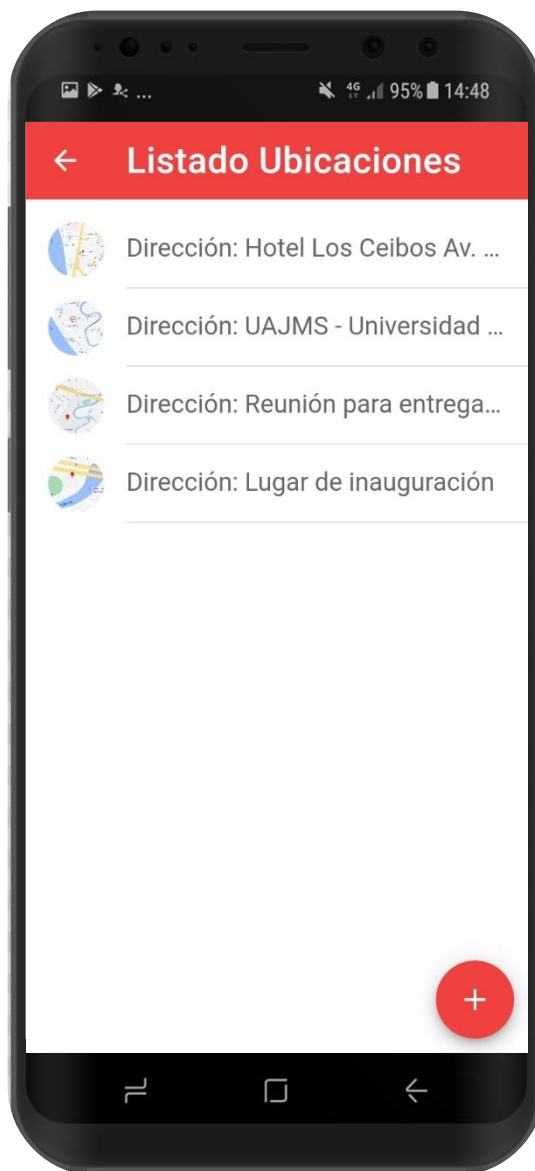


Figura 4.34. Rol Administrador - Pantalla 24 - Listado Ubicaciones.

En esta pantalla se muestra el listado de ubicación, en la inferior un botón par nuevo registro y haciendo click en la lista tenemos la opción de modificar. Al deslizar hacia la izquierda con click sostenido, tenemos la opción de borrar.

4.6.25. Pantalla 25 - Nueva Ubicación

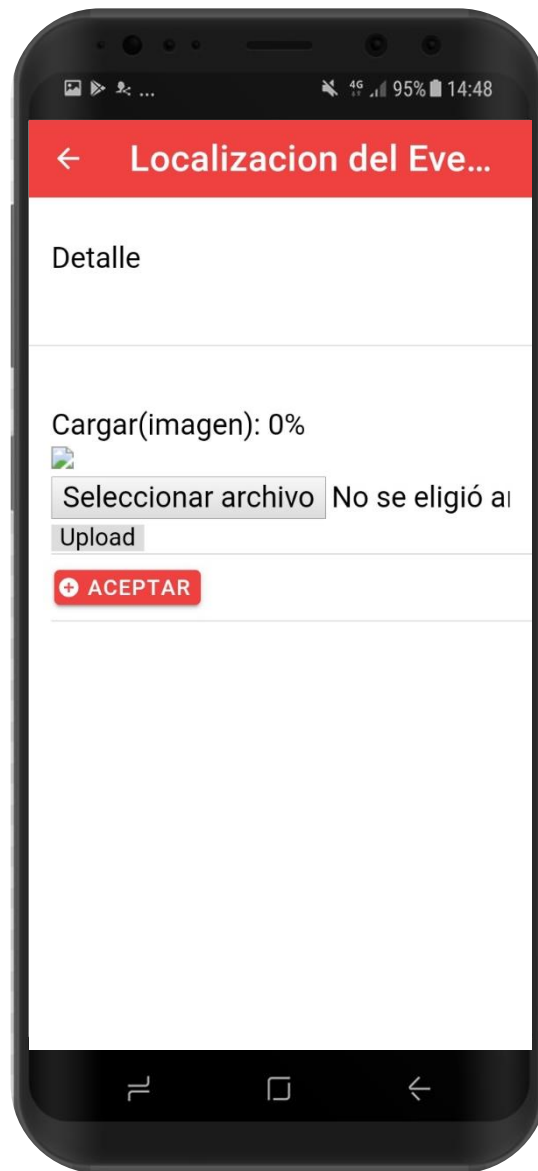


Figura 4.35. Rol Administrador - Pantalla 25 - Nueva Ubicación.

En esta pantalla se permite registrar y enviar una imagen con referencia a la nueva ubicación de una actividad del evento.

4.6.26. Pantalla 26 - Modificar Ubicación

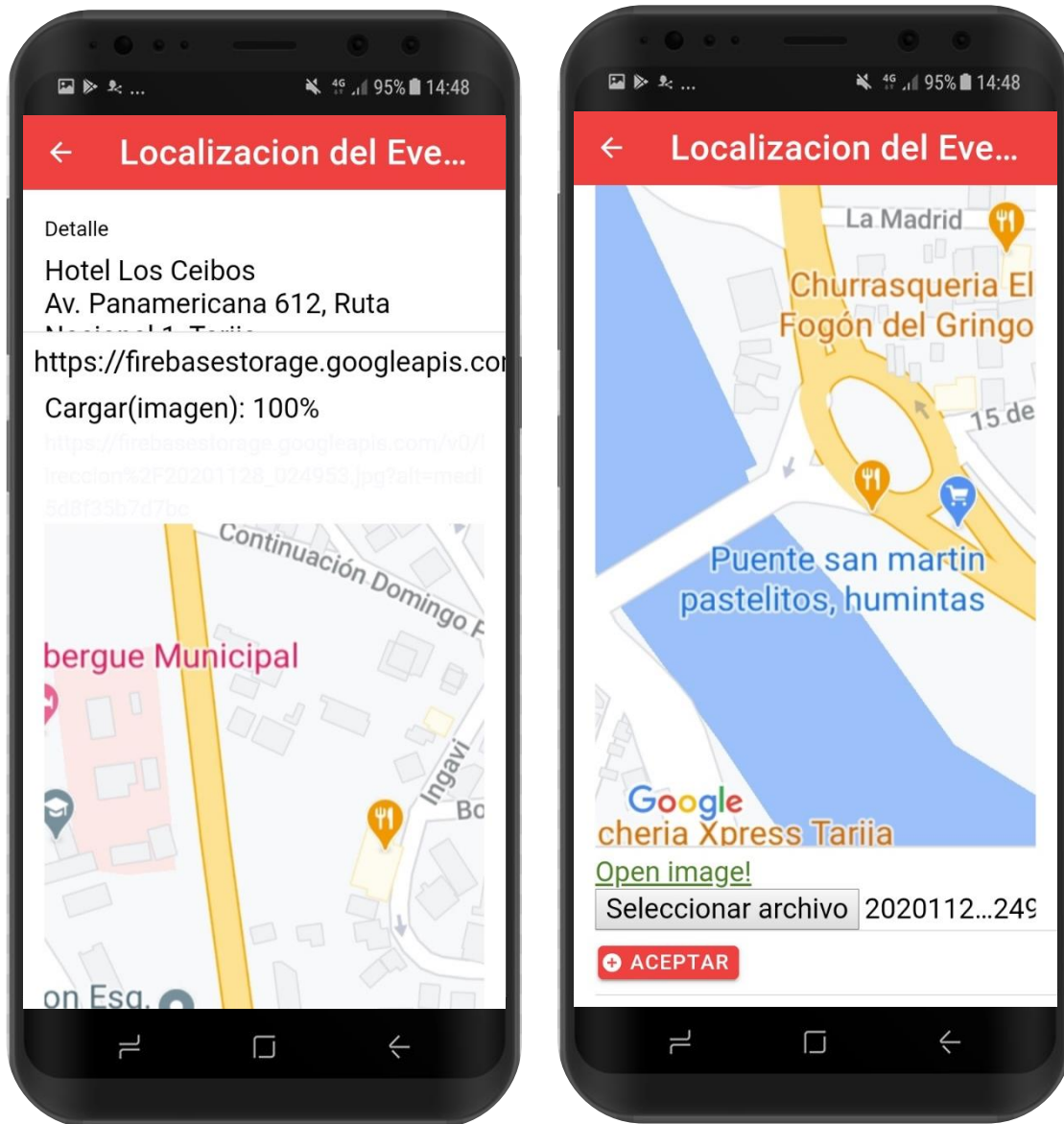


Figura 4.36. Rol Administrador – Pantalla 26 - Modificar Ubicación.

En esta pantalla se permite la modificación de los datos de ubicación.

4.6.27. Pantalla 27 - Borrar Ubicación

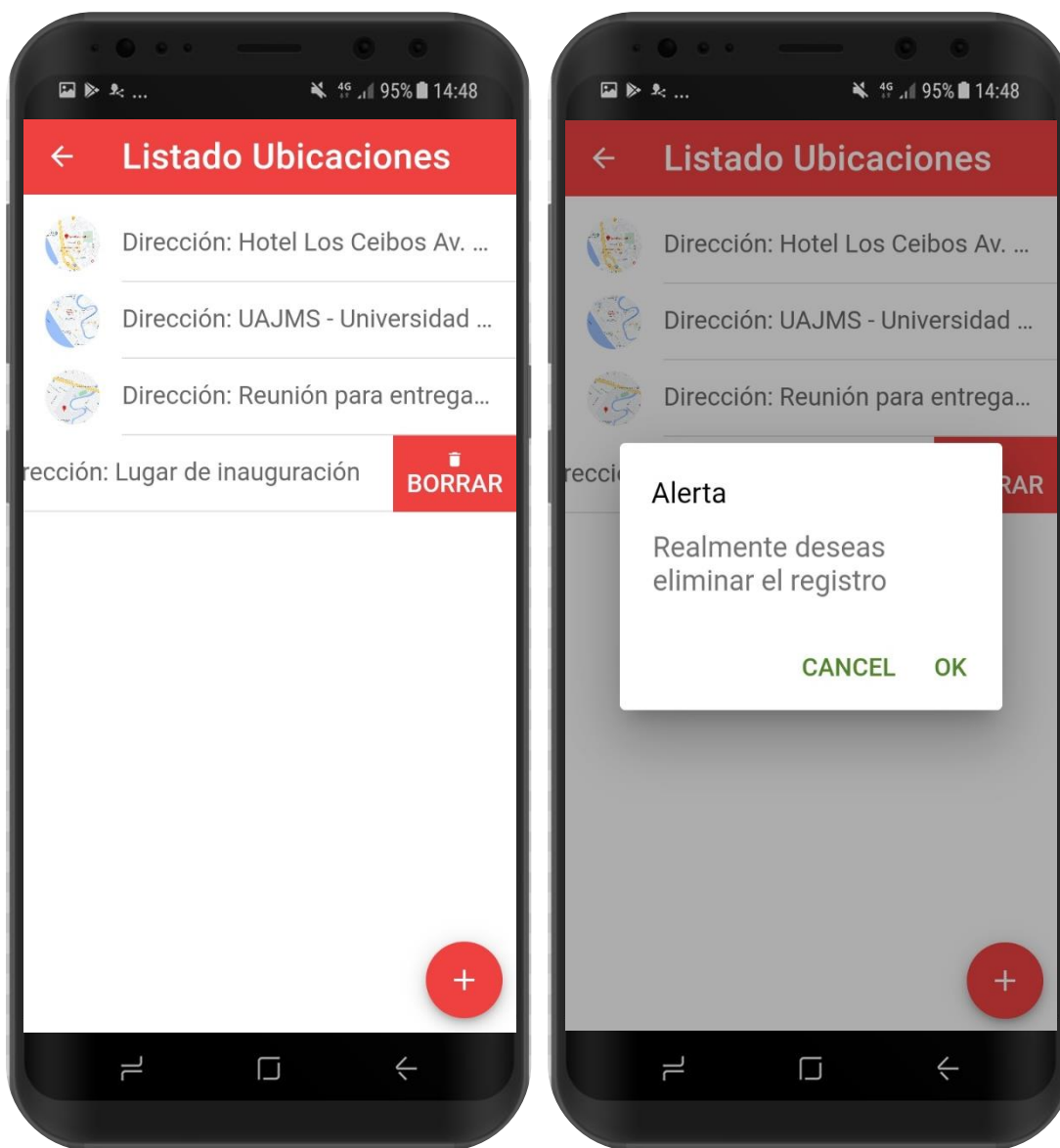


Figura 4.37. Rol Administrador - Pantalla 27 - Borrar Ubicación.

En esta pantalla se permite el borrado de los datos de una ubicación al presionarse el texto y deslizar el cuadro hacia la izquierda. Al presionar el botón BORRAR aparece un mensaje de confirmación por seguridad de la información.

4.6.28. Pantalla 28 - Reporte Cobros de Participante

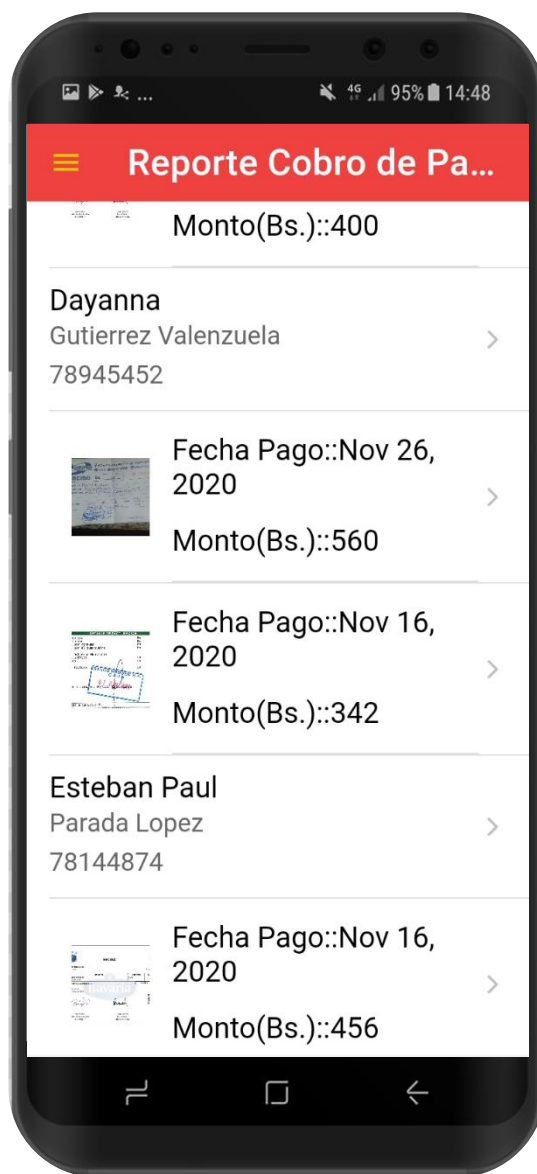


Figura 4.38. Rol Administrador - Pantalla 28 - Reporte Cobros de Participante.

En esta pantalla se permite ver el listado de los participantes que enviaron su comprobante de pago y también de los que no registraron su envío de comprobante.

4.6.29. Pantalla 29 - Registro Comprobante de Cobro

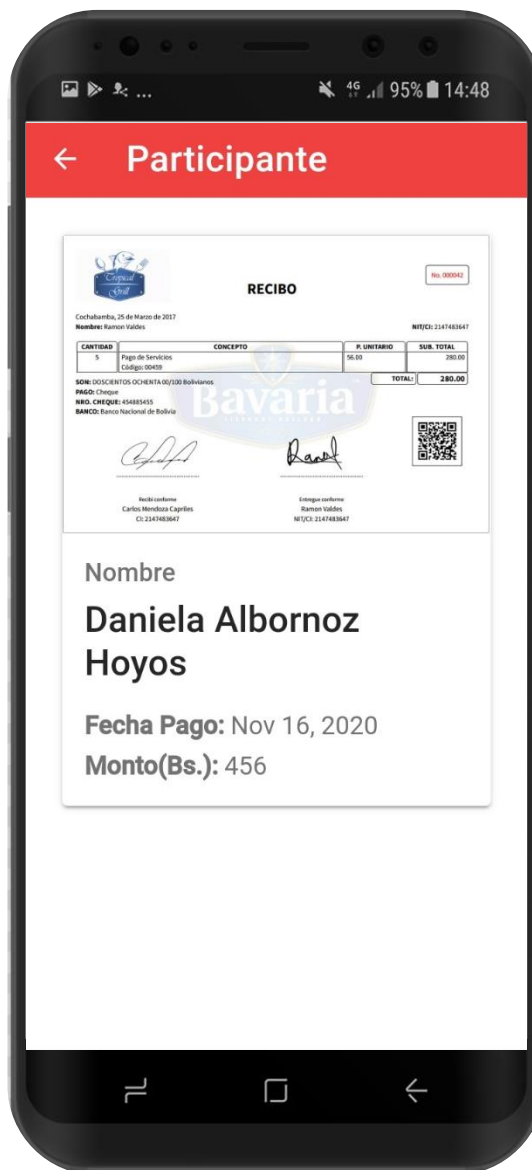


Figura 4.39: Rol Administrador - Pantalla 29 - Registro Comprobante de Cobro.
En esta pantalla se permite ver el comprobante de pago enviado por el participante.

4.6.30. Pantalla 30 - Borrar Cobro de Participante

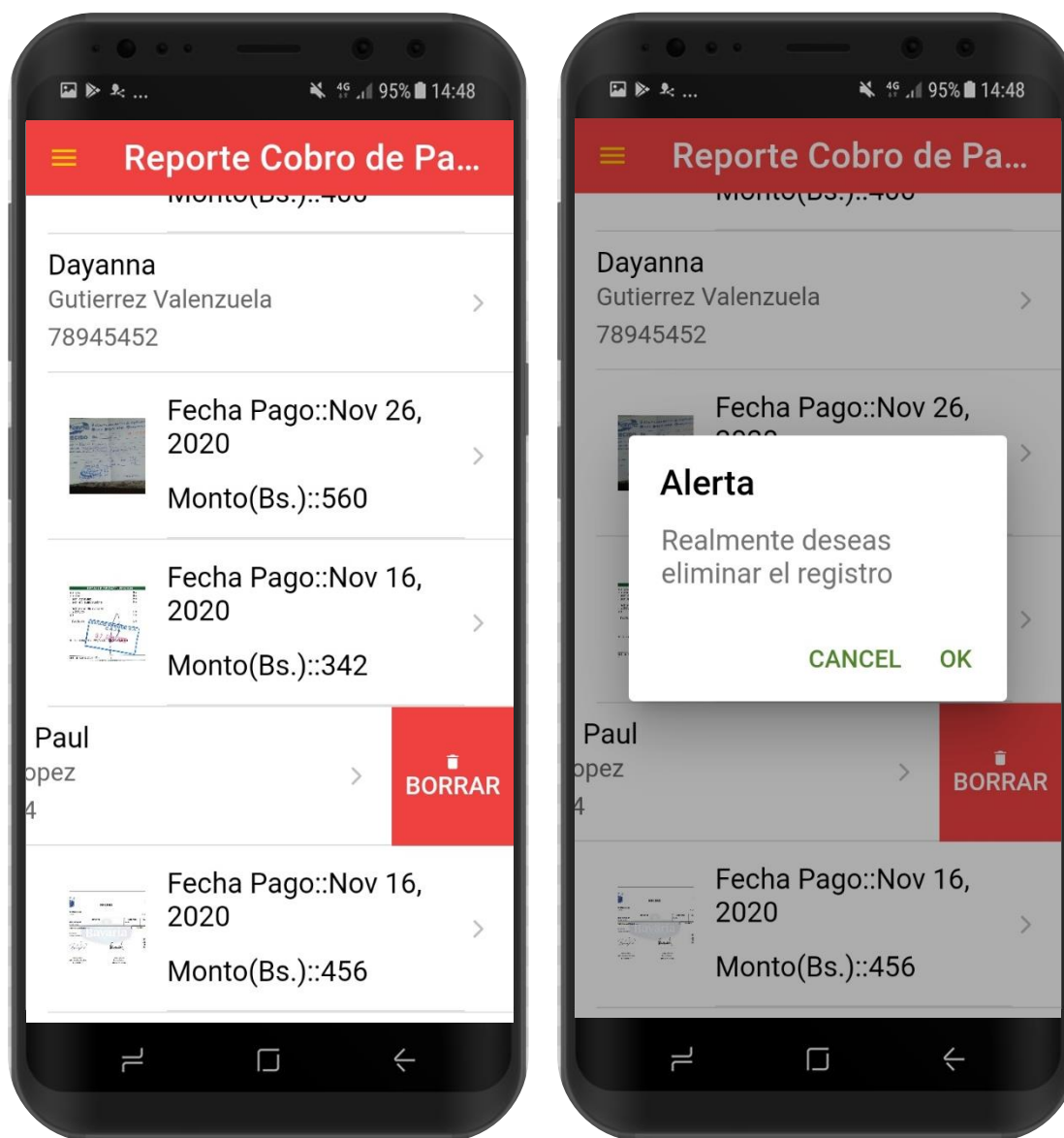


Figura 4.40. Rol Administrador - Pantalla 30 - Borrar Cobro de Participante.

En esta pantalla se permite el borrado de los datos de un cobro al presionarse el texto y deslizar el cuadro hacia la izquierda. Al presionar el botón BORRAR aparece un mensaje de confirmación por seguridad de la información.

4.6.31. Pantalla 31 - Reporte de Participantes

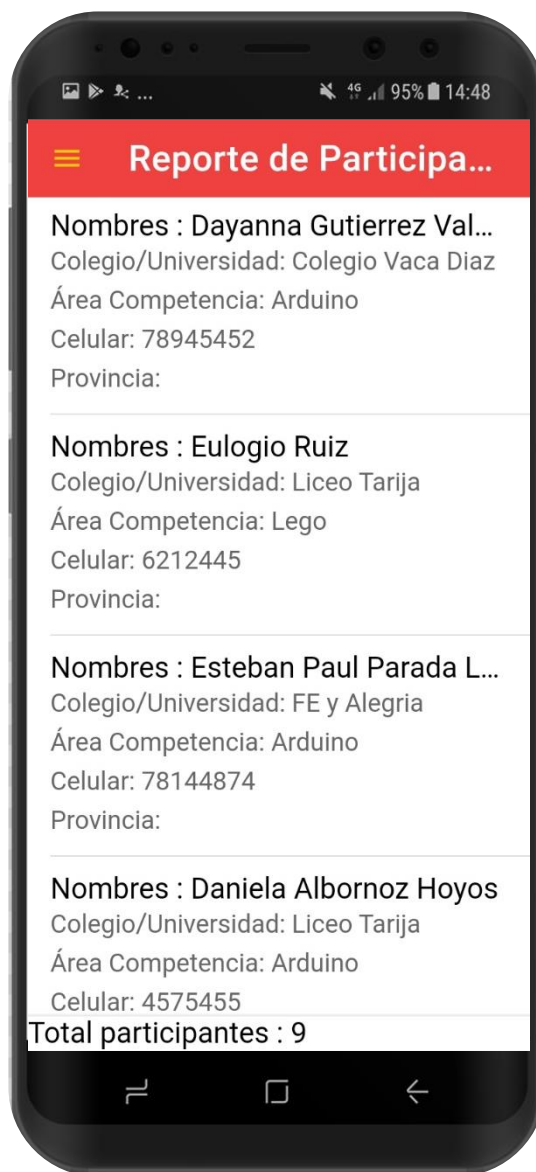


Figura 4.41. Rol Administrador - Pantalla 31- Reporte de Participantes.

En esta pantalla se permite ver el listado de los participantes inscritos. En la parte inferior se visualiza un contador para conocer la cantidad de participantes en competencia.

4.7. Pantallas Rol Tutor

4.7.1. Pantalla 1 - Logueo

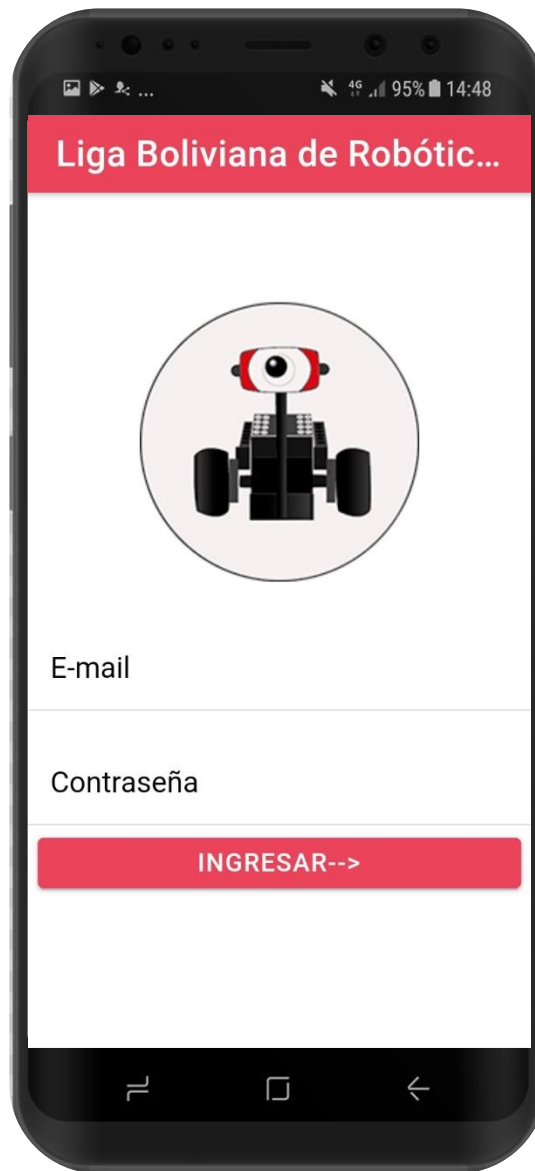


Figura 4.42. Rol Tutor - Pantalla 1 – Logueo.

En esta pantalla se permite el ingreso del usuario y clave. Los dos datos de acceso deben ser proporcionados por la persona que cumpla el rol de administrador.

4.7.2. Pantalla 2 - Bienvenida



Figura 4.43. Rol Tutor - Pantalla 2 - Bienvenida.

En esta pantalla se muestra una bienvenida a la aplicación, se identifica el módulo para el rol de tutor.

4.7.3. Pantalla 3 - Menú General

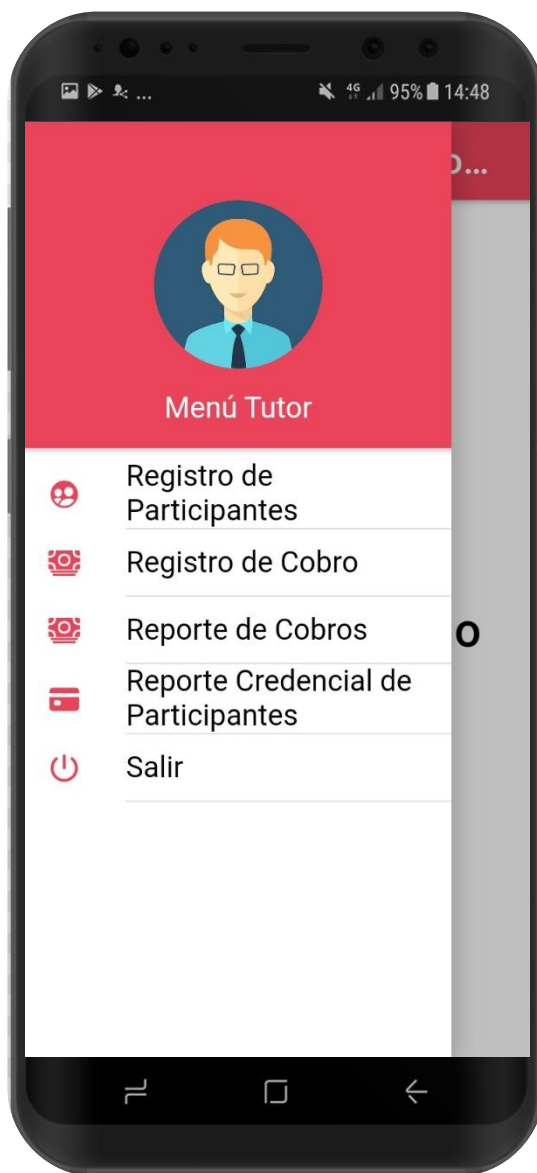


Figura 4.44. Rol Tutor - Pantalla 3 - Menú General.

En esta pantalla se permite ver el menú general de la aplicación módulo tutor.

4.7.4. Pantalla 4 - Listado de Participantes

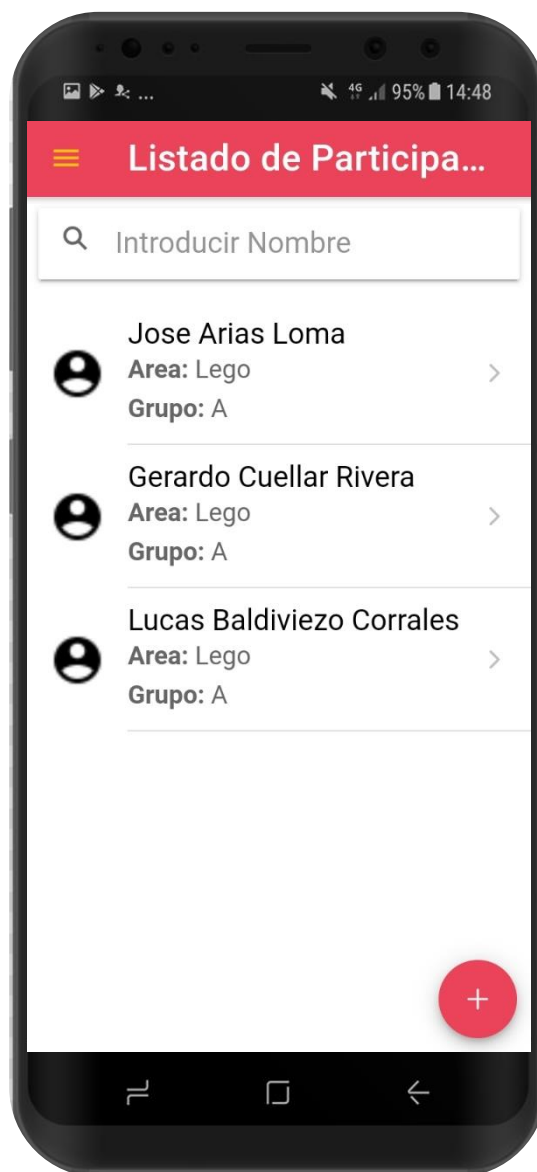


Figura 4.45. Rol Tutor - Pantalla 4 - Listado de Participantes.

En esta pantalla se muestra el listado de participantes y presenta tiene la opción de un buscador. En la parte inferior se tiene un botón para adicionar un nuevo registro y haciendo click en la lista tenemos la opción de modificar. Al deslizar hacia la izquierda con click sostenido, tenemos la opción de borrar.

4.7.5. Pantalla 5 - Registro Nuevo Participante

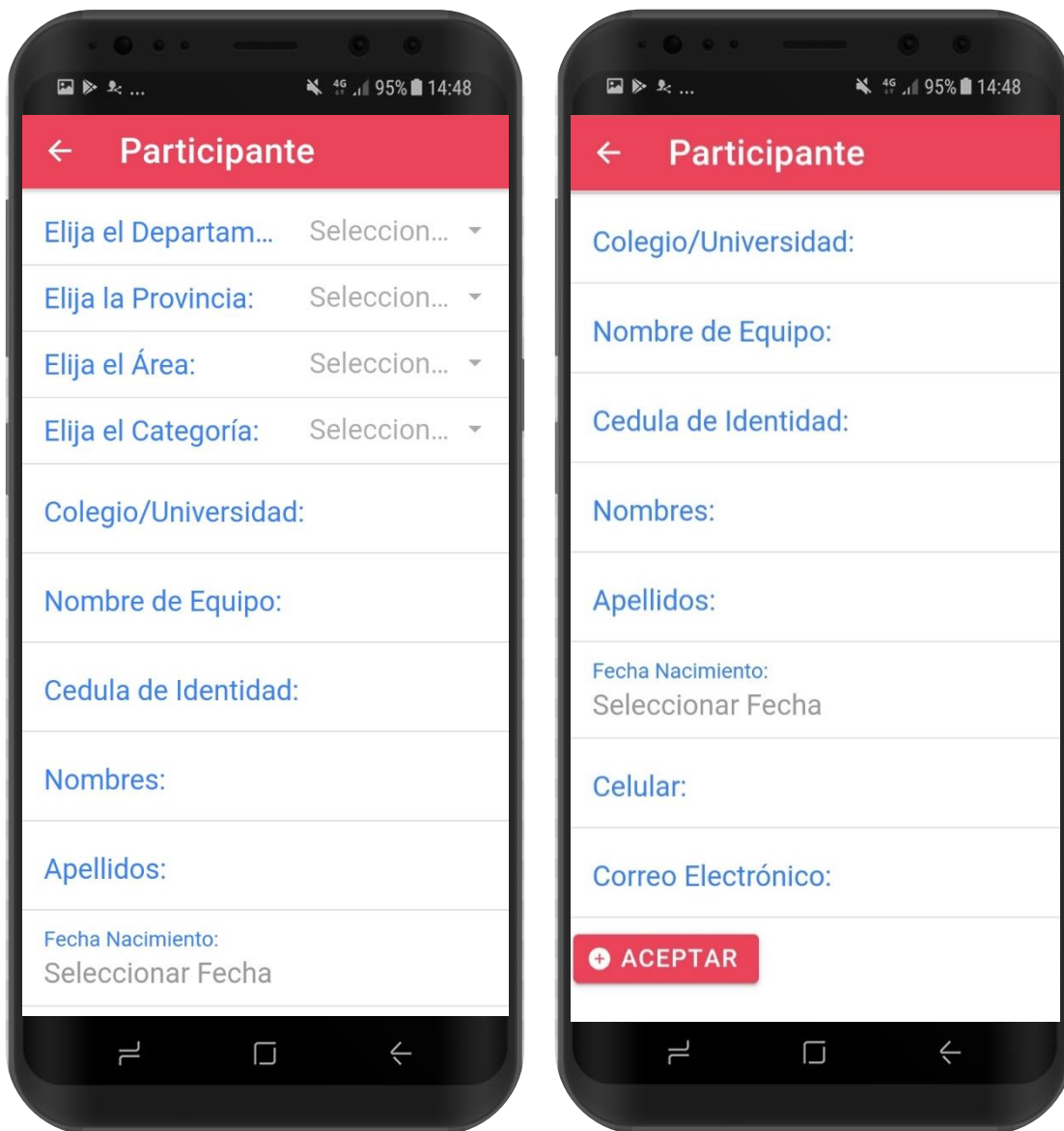


Figura 4.46. Rol Tutor - Pantalla 5 - Registro Nuevo Participante.

En esta pantalla se permite registrar nuevos datos del participante.

4.7.6. Pantalla 6 - Modificar Datos del Participante

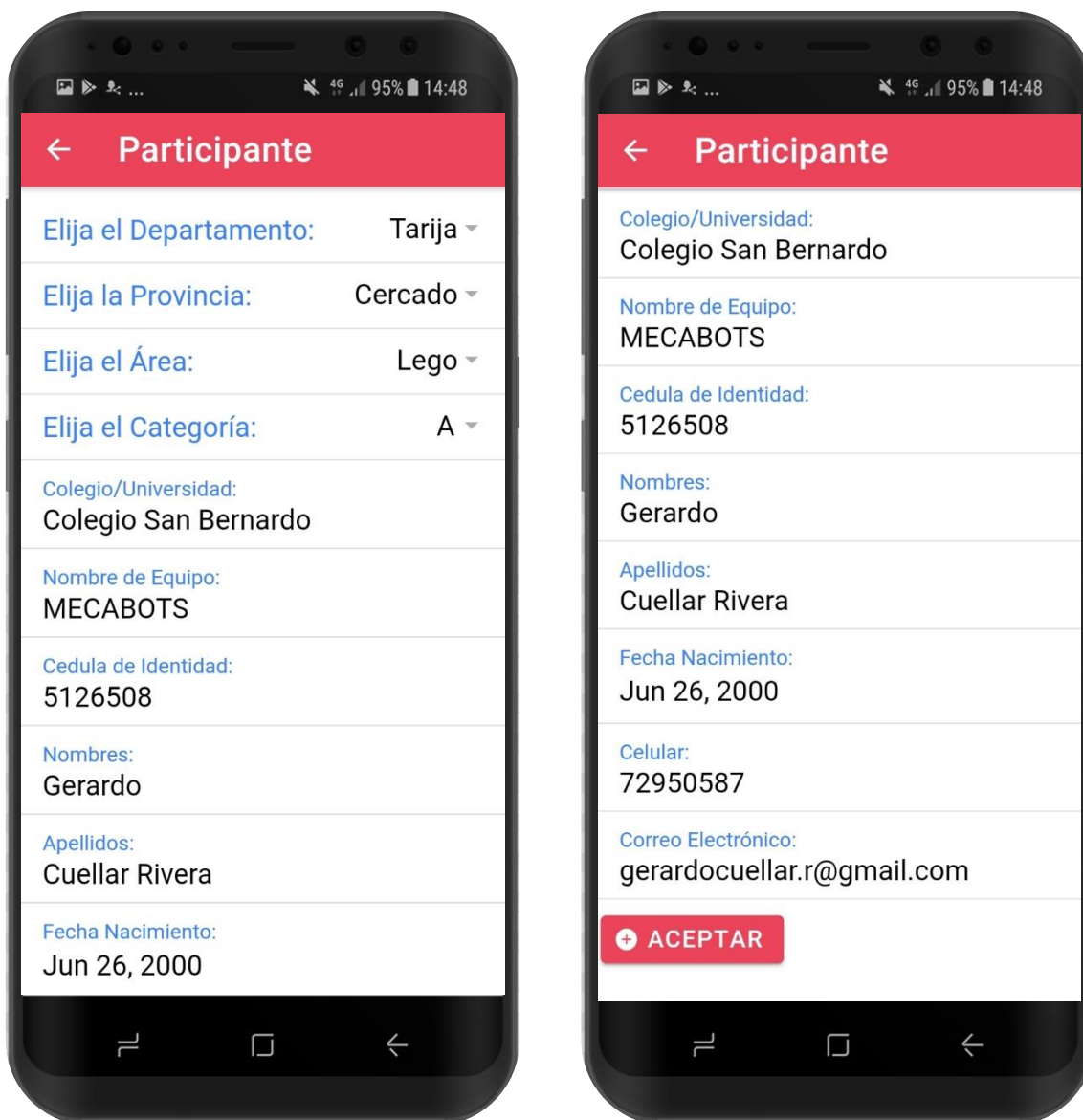


Figura 4.47. Rol Tutor - Pantalla 6 - Modificar Datos del Participante.

En esta pantalla se permite modificar los datos del participante.

4.7.7. Pantalla 7 - Eliminar Datos del Participante

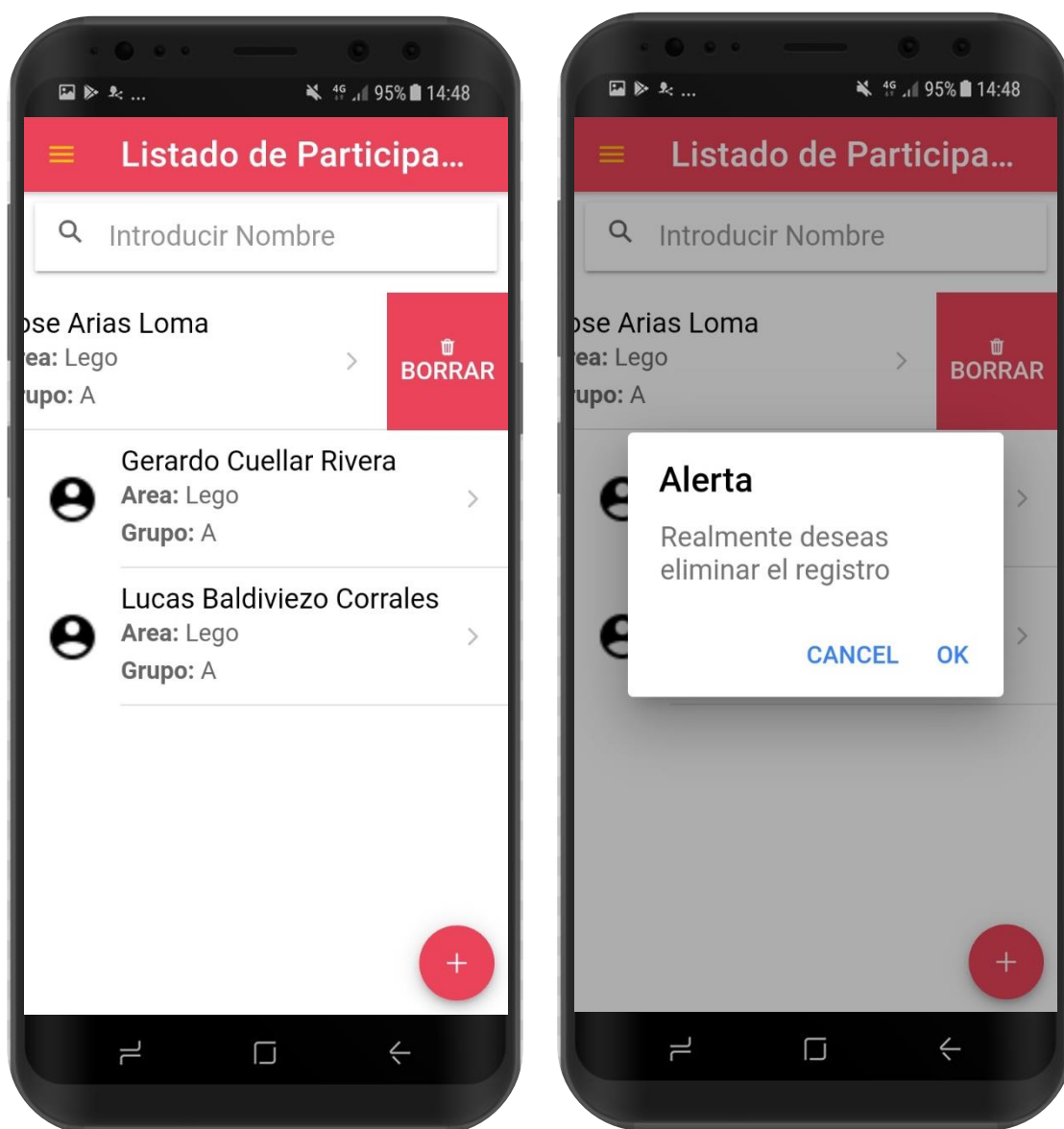


Figura 4.48. Rol Tutor – Pantalla 7- Eliminar Datos del Participante.

En esta pantalla se permite el borrado de los datos de un participante al presionarse el texto y deslizar el cuadro hacia la izquierda. Al presionar el botón BORRAR aparece un mensaje de confirmación por seguridad de la información.

4.7.9. Pantalla 8 - Registro Comprobante de Cobro

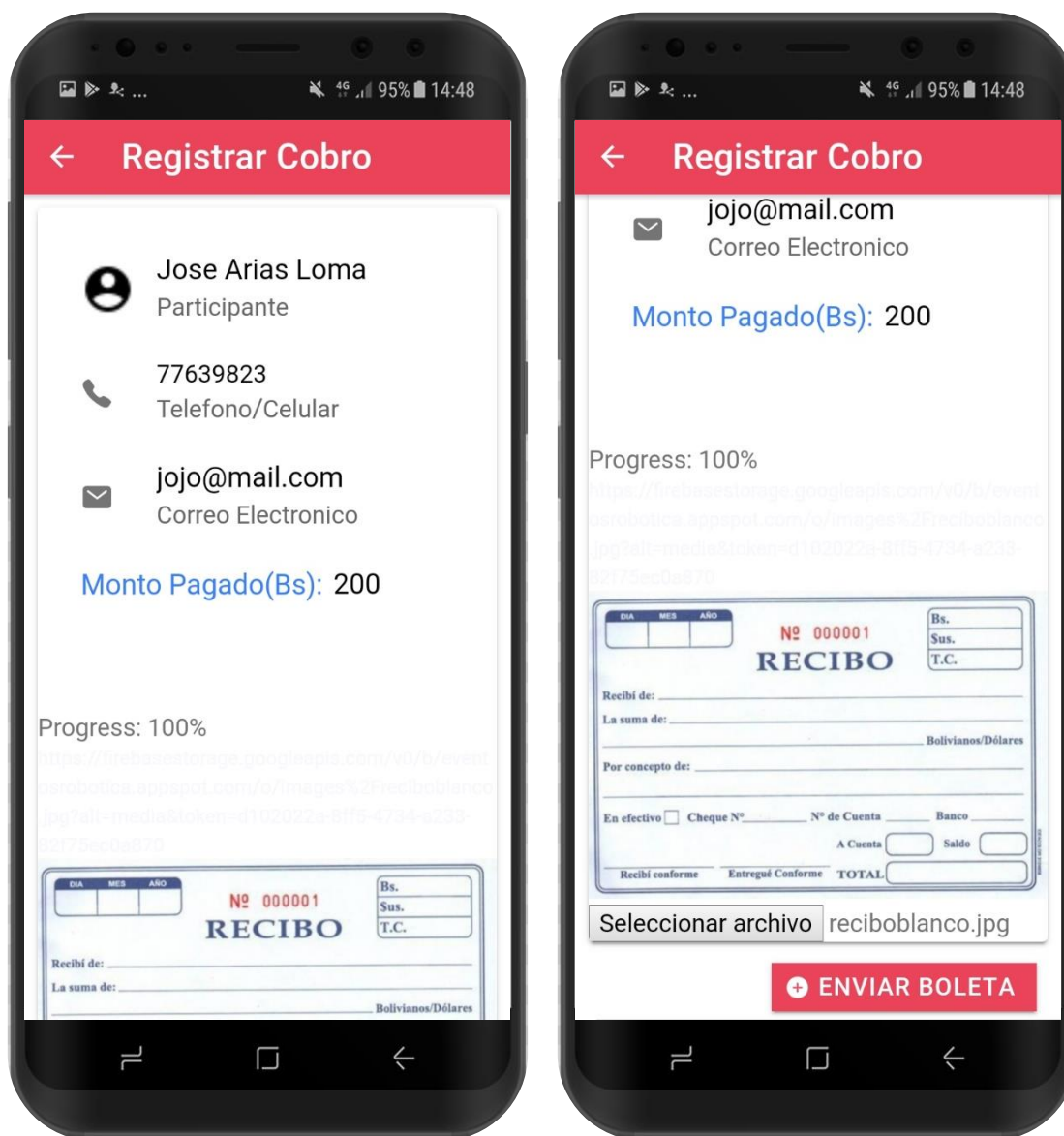


Figura 4.49. Rol Tutor – Pantalla 8 - Registro Comprobante de Cobro.

En esta pantalla se permite registrar el comprobante de pago del participante.

4.7.10. Pantalla 9 - Reporte de Cobros

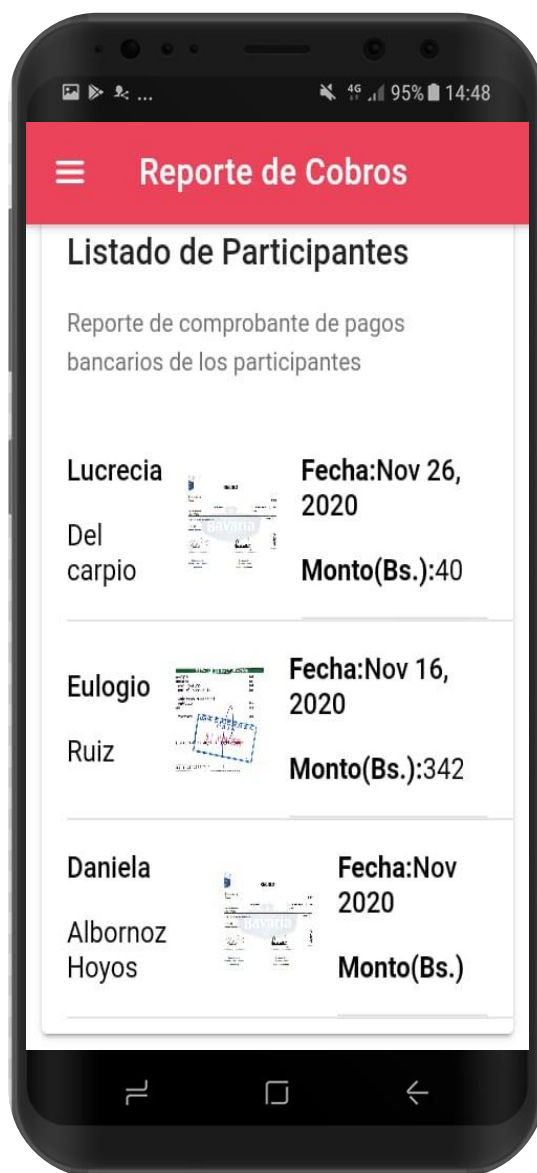


Figura 4.50. Rol Tutor - Pantalla 9 - Reporte de Cobros.

En esta pantalla se permite ver la lista de los pagos realizados por los participantes.

4.7.11. Pantalla 10 - Reporte de Credenciales

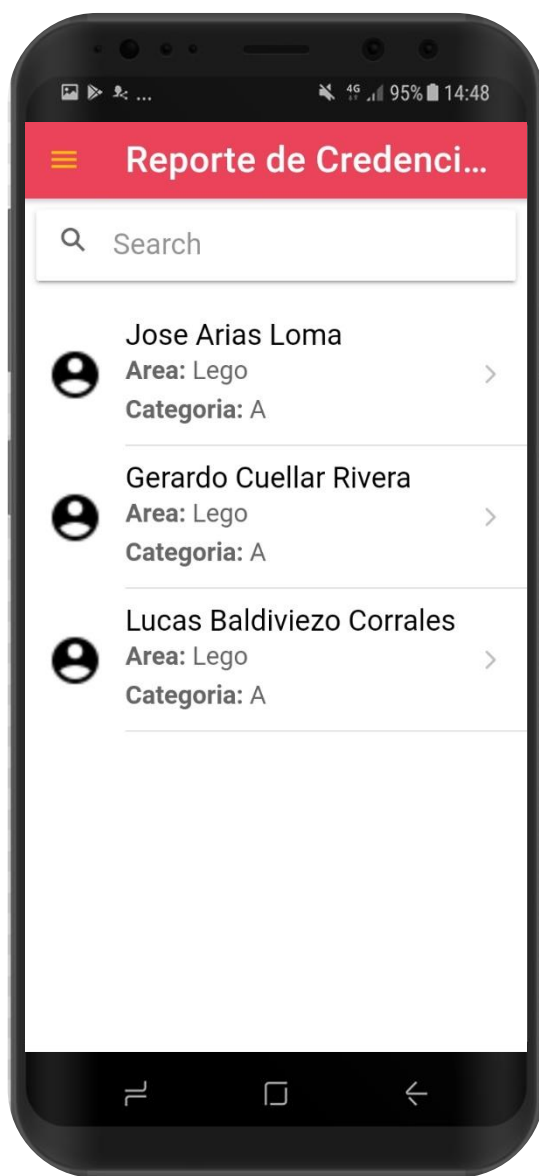


Figura 4.51. Rol Tutor - Pantalla 10 - Reporte de Credenciales.

En esta pantalla se permite ver la lista de los participantes inscritos y seleccionar una persona para ver su credencial.

4.7.12. Pantalla 11 - Credencial del Participante

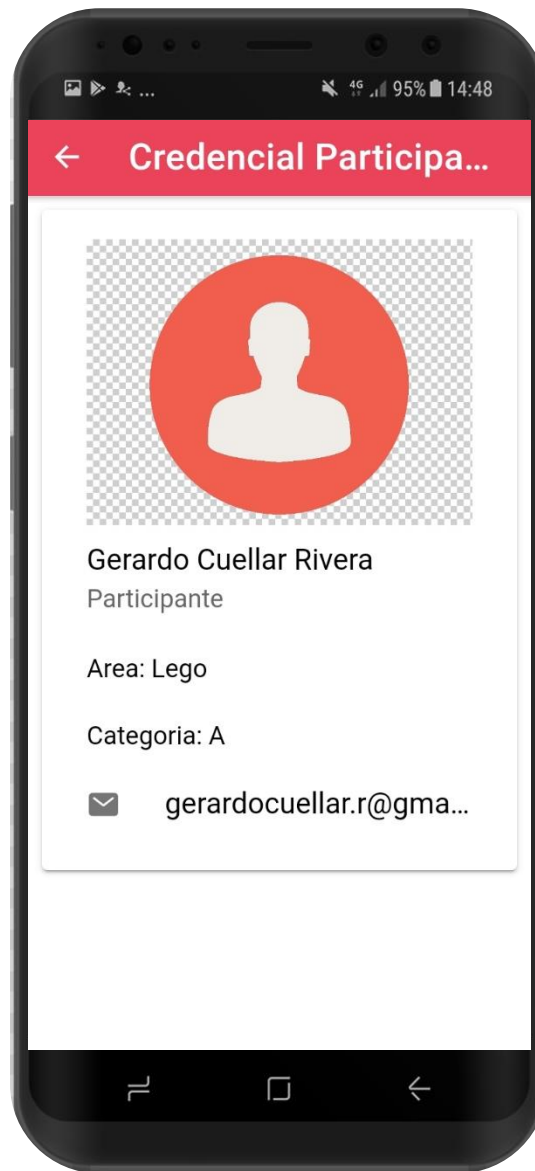


Figura 4.52. Rol Tutor - Pantalla 11 - Credencial del Participante.

En esta pantalla se permite visualizar la credencial del participante y verificar el nombre correcto con la categoría y área seleccionada.

4.8. Pantallas Rol Participante

4.8.1. Pantalla 1 - Pantalla Inicial (Imagen 1)



Figura 4.53. Rol Participante - Pantalla 1 - Inicial - Imagen1.

En esta pantalla se permite ver fotografías de competidores bolivianos en eventos internacionales.

4.8.2. Pantalla 1 - Pantalla Inicial (Imagen 2)

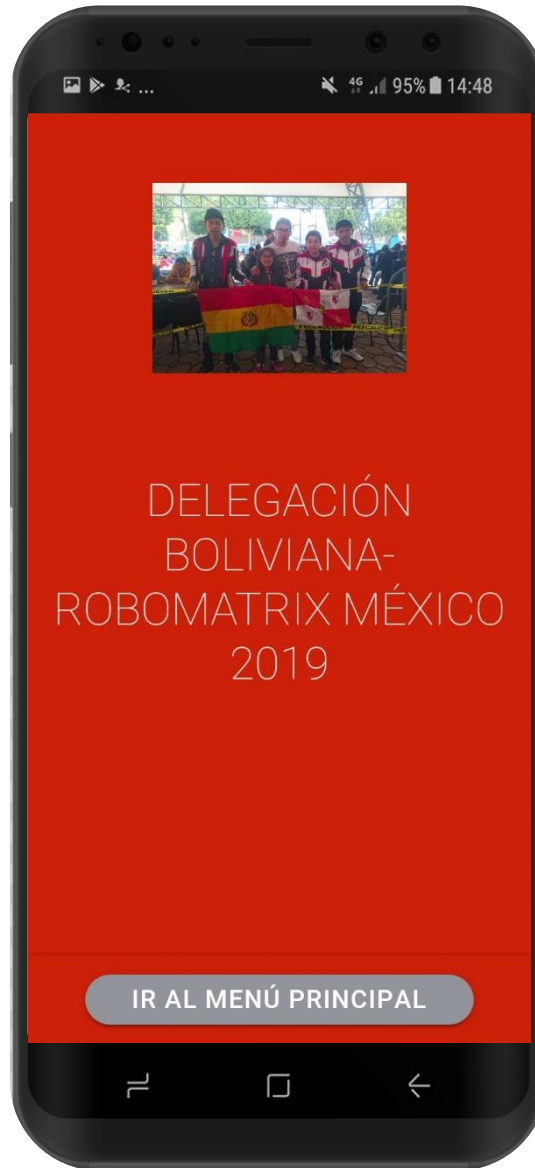


Figura 4.54. Rol Participante - Pantalla 1 - Inicial - Imagen 2.

En esta pantalla se permite ver fotografías de competidores bolivianos en eventos internacionales.

4.8.3. Pantalla 1 - Pantalla Inicial (Imagen 3)



Figura 4.55. Rol Participante - Pantalla 1 - Inicial - Imagen 3.

En esta pantalla se permite ver fotografías de competidores bolivianos en eventos internacionales.

4.8.4. Pantalla 2 - Eventos

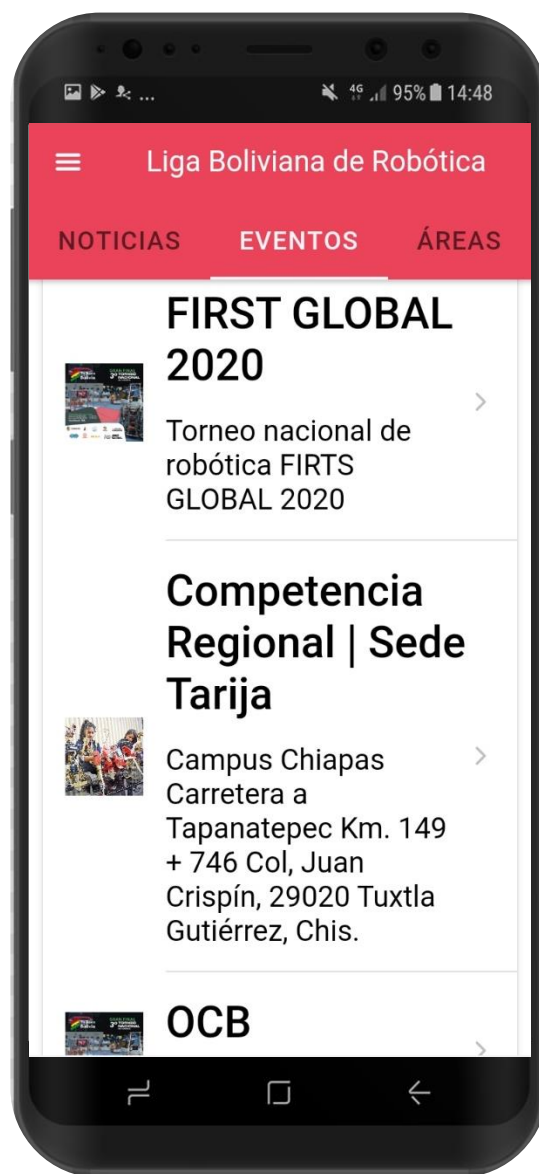


Imagen 4.56. Rol Participante - Pantalla 2 - Eventos.

En esta pantalla se permite ver el listado de eventos relacionados con la robótica y haciendo click en cada uno de sus cuadros, se permite ver más información a detalle.

4.8.5. Pantalla 3 - Detalle del Evento

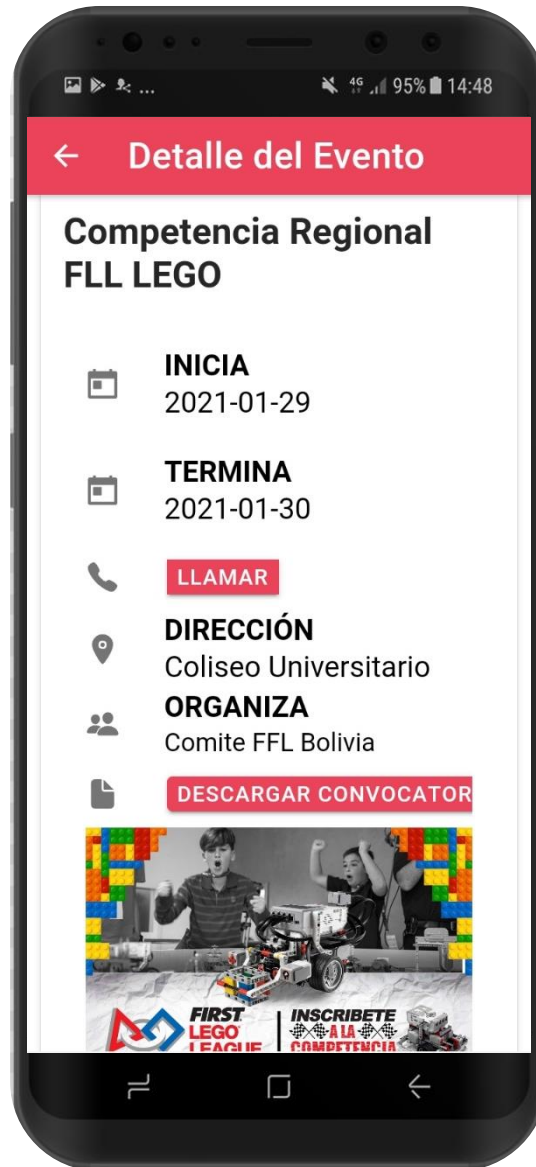


Figura 4.57. Rol Participante - Pantalla 3 - Detalle del Evento.

En esta pantalla se permite visualizar toda la información necesaria para participar en un evento.

4.8.6. Pantalla 4 - Áreas



Figura 4.58. Rol Participante - Pantalla 4 - Áreas.

En esta pantalla se permite ver el listado de áreas que puede contemplar un evento y se puede acceder a información específica de cada área.

4.8.7. Pantalla 5 - Detalle de Área



Figura 4.59. Rol Participante - Pantalla 5 - Detalle de Área.

En esta pantalla se permite descargar las instrucciones específicas para un área de la competencia.

4.8.8. Pantalla 6 - Noticias



Figura 4.50. Rol Participante - Pantalla 6 - Noticias.

En esta pantalla se permite ver un listado de noticias referente a la competencia en desarrollo o publicaciones relacionadas con cursos, conferencias y próximos eventos de competencia a realizarse.

4.8.9. Pantalla 7 - Detalle Anuncio



Figura 4.51. Rol Participante - Pantalla 7 – Detalle Anuncio.

En esta pantalla se permite ver información ampliada sobre cada noticia o anuncio publicado.

4.8.10. Pantalla 8 - Menú

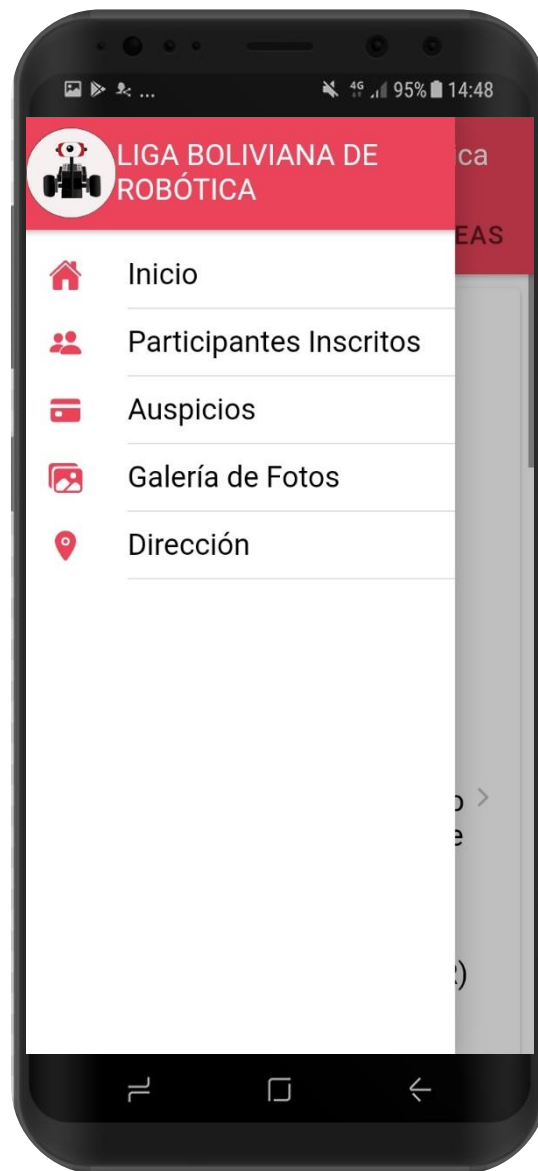


Figura 4.52. Rol Participante - Pantalla 8 - Menú.

En esta pantalla se puede acceder al menú principal y seleccionar la opción requerida.

4.8.11. Pantalla 9 - Participantes Inscritos

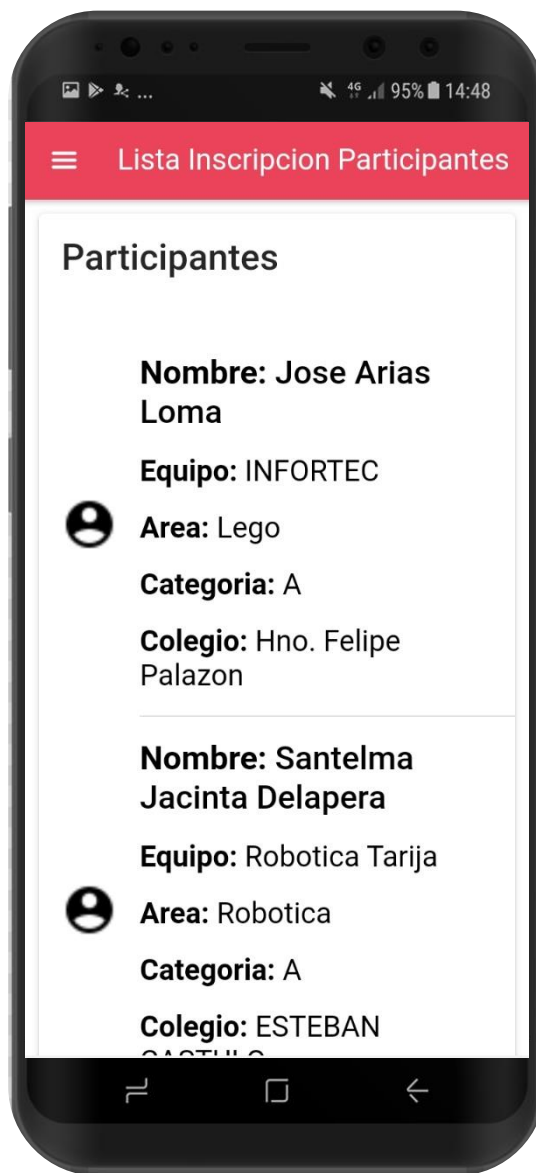


Figura 4.53. Rol Participante - Pantalla 9 - Participantes Inscritos.

En esta pantalla se permite ver un listado general de todos los participantes inscritos con sus respectivas áreas y categorías.

4.8.12. Pantalla 10 - Auspicios

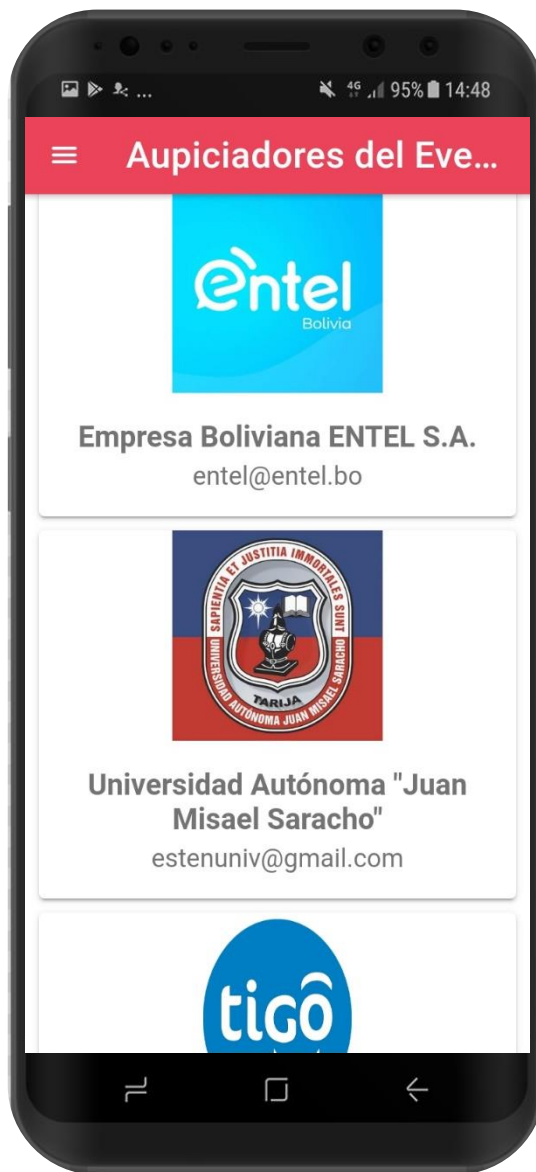


Figura 4.54. Rol Participante - Pantalla 10 - Auspicios.

En esta pantalla los participantes pueden conocer el listado de las instituciones y empresas que auspician la competencia de robótica.

4.8.13. Pantalla 11 - Galería de Fotos

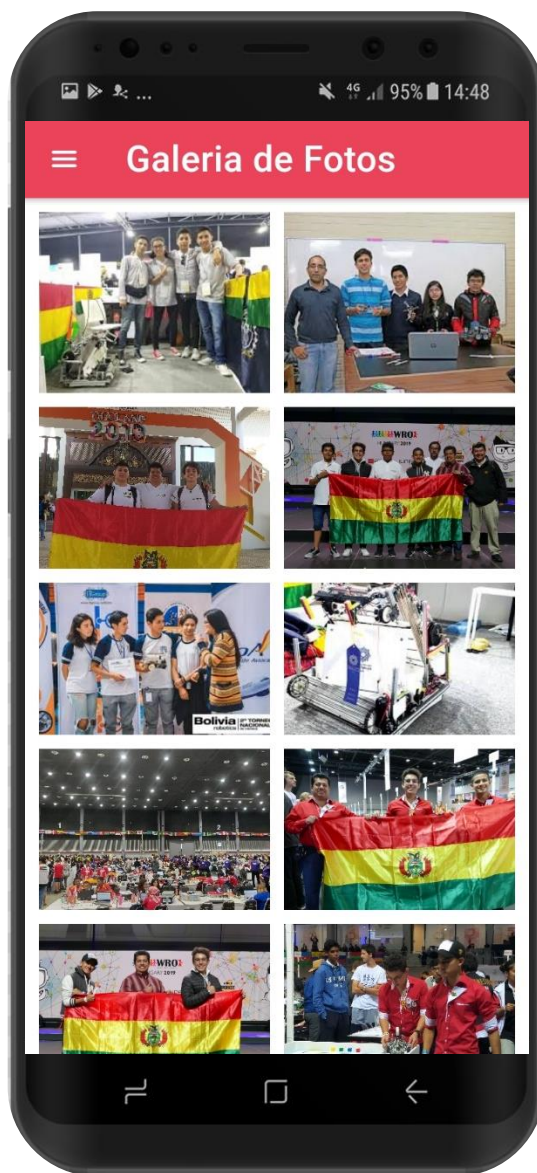


Figura 4.55. Rol Participante - Pantalla 11 - Galería de Fotos.

En esta pantalla los participantes pueden ver fotografías destacadas de competencias anteriores recordando a los equipos ganadores en pruebas nacionales o internacionales.

4.8.14. Pantalla 12 - Foto Ampliada

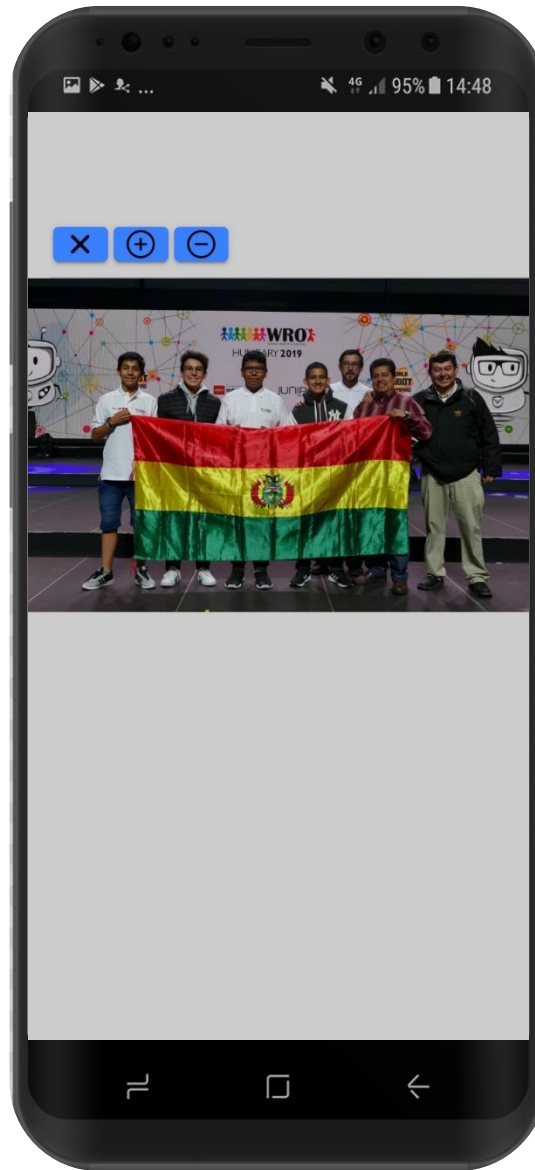


Figura 4.56. Rol Participante - Pantalla 12 - Foto Ampliada.

En esta pantalla se puede ver una fotografía de la galería en un tamaño aumentado. Al colocarse el celular en posición horizontal la fotografía puede aumentar una vez más su dimensión.

4.8.15. Pantalla 13 - Dirección Evento

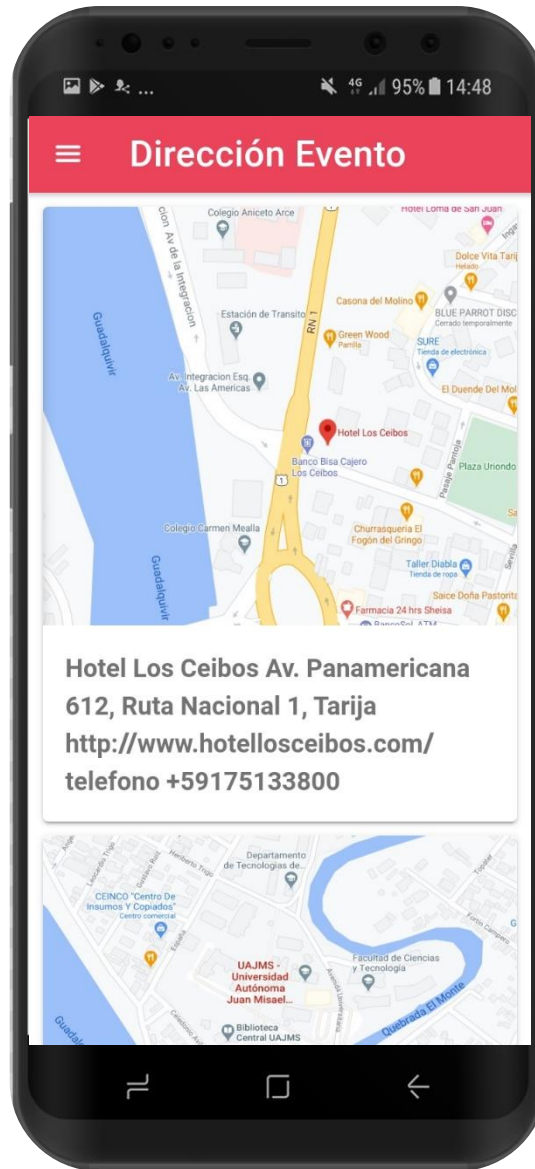


Figura 4.57. Rol Participante - Pantalla 13 - Dirección Evento.

En esta pantalla los participantes pueden ver la imagen y descripción de una dirección importante, para la competencia en desarrollo.

IMPLEMENTACIÓN

5. IMPLEMENTACIÓN

5.1. Sección Código de Logueo

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { IonSlides, ToastController, LoadingController } from '@ionic/angular';
import { User } from 'src/app/interfaces/user';
import { AuthService } from 'src/app/services/auth.service';
@Component({
  selector: 'app-login',
  templateUrl: './login.page.html',
  styleUrls: ['./login.page.scss'],
})
export class LoginPage implements OnInit {
  //@ViewChild(IonSlides) slides: IonSlides;
  public userLogin: User = {};
  public userRegister: User = {};
  private loading: any;
  constructor(private loadingCtrl: LoadingController,
    private toastCtrl: ToastController,
    private authService: AuthService
  ) { }
  ngOnInit() {
  }
  /*segmentChanged(event: any) {
    if (event.detail.value == "login") {
      this.slides.slidePrev();
    } else {
      this.slides.slideNext();
    }
  }
}
```

```

    }
*/
//funciones creadas
async login() {
    await this.presentLoading();
    try {
        await this.authService.login(this.userLogin);
    } catch (error) {
        console.error(error);
        this.presentToast(error.message);
    } finally {
        this.loading.dismiss();
    }
}
async register() {
    await this.presentLoading();
    try {
        await this.authService.register(this.userRegister);
    } catch (error) {
        console.error(error);
        this.presentToast(error.message);
    } finally {
        this.loading.dismiss();
    }
}
} //fin register
//funcion para loading
async presentLoading() {
    this.loading = await this.loadingCtrl.create({ message: 'Por favor, Aguarde...' });
    return this.loading.present();
}
async presentToast(message:string) {

```

```

const toast = await this.toastCtrl.create({ message,duration:2000});
toast.present();
}

```

5.2 Sección Código de Organizadores

```

import { Injectable } from '@angular/core';
import { AngularFirestore, AngularFirestoreCollection } from '@angular/fire/firestore';
import { map } from 'rxjs/operators';
import { Organizador } from '../interfaces/organizador';
@Injectable({
  providedIn: 'root'
})
export class OrganizadoresService {
  private organizadorsCollection: AngularFirestoreCollection<Organizador>;
  constructor(private afs1:AngularFirestore) {
    this.organizadorsCollection = this.afs1.collection<Organizador>('organizador');
  }
  //creamos las operaciones basicas de un abm para Admin
  getOrganizadores() {
    return this.organizadorsCollection.snapshotChanges().pipe(
      map(actions=>{
        return actions.map(a=>{
          const data=a.payload.doc.data();
          const id=a.payload.doc.id;
          return {id, ...data};
        });
      })
    );
  }
  getOrganizador(id:string) {

```

```

    return this.organizadoresCollection.doc<Organizador>(id).valueChanges();
  }
  addOrganizador(organizador:Organizador) {
    return this.organizadoresCollection.add(organizador);
  }
  updateOrganizador(id:string,organizador:Organizador) {
    return this.organizadoresCollection.doc<Organizador>(id).update(organizador);
  }
  deleteOrganizador(id:string) {
    return this.organizadoresCollection.doc(id).delete();
  }
}

```

5.3 Sección Código de Tutor

```

import { Injectable } from '@angular/core';
import { AngularFireStore, AngularFireStoreCollection } from '@angular/fire/firestore';
import { map } from 'rxjs/operators';
import { Organizador } from '../interfaces/organizador';
@Injectable({
  providedIn: 'root'
})
export class OrganizadoresService {
  private organizadoresCollection: AngularFireStoreCollection<Organizador>;
  constructor(private afs1:AngularFirestore) {
    this.organizadoresCollection = this.afs1.collection<Organizador>('organizador');
  }
  //creamos las operaciones basicas de un abm para Admin
  getOrganizadores() {
    return this.organizadoresCollection.snapshotChanges().pipe(
      map(actions=>{

```

```

    return actions.map(a=>{
      const data=a.payload.doc.data();
      const id=a.payload.doc.id;
      return {id, ...data};
    });
  })
);
}

getOrganizador(id:string) {
  return this.organizadorsCollection.doc<Organizador>(id).valueChanges();
}

addOrganizador(organizador:Organizador) {
  return this.organizadorsCollection.add(organizador);
}

updateOrganizador(id:string,organizador:Organizador) {
  return this.organizadorsCollection.doc<Organizador>(id).update(organizador);
}

deleteOrganizador(id:string) {
  return this.organizadorsCollection.doc(id).delete();
}
}
}

```

5.4. Sección Código de Área

```

import { Injectable } from '@angular/core';
import { Area } from '../interfaces/area';
import { AngularFirestore, AngularFirestoreCollection } from '@angular/fire/firestore';
import { map } from 'rxjs/operators';
@Injectable({
  providedIn: 'root'
})

```

```

export class AreaService {
  private areasCollection: AngularFireCollection<Area>;
  constructor(private afs1:AngularFirestore) {
    this.areasCollection = this.afs1.collection<Area>('area');
  }
  //creamos las operaciones basicas de un abm para Admin
  getAreas() {
    return this.areasCollection.snapshotChanges().pipe(
      map(actions=>{
        return actions.map(a=>{
          const data=a.payload.doc.data();
          const id=a.payload.doc.id;
          return {id, ...data};
        });
      })
    );
  }
  getArea(id:string) {
    return this.areasCollection.doc<Area>(id).valueChanges();
  }
  addArea(area:Area) {
    return this.areasCollection.add(area);
  }
  updateArea(id:string,area:Area) {
    return this.areasCollection.doc<Area>(id).update(area);
  }
  deleteArea(id:string) {
    return this.areasCollection.doc(id).delete();
  }
}

```


5.5. Sección Código de Anuncios

```
import { Injectable } from '@angular/core';
import { Anuncio } from '../interfaces/anuncio';
import { AngularFireCollection, AngularFire } from '@angular/fire/firestore';
import { map } from 'rxjs/operators';
@Injectable({
  providedIn: 'root'
})
export class AnuncioService {
  private anunciosCollection: AngularFireCollection<Anuncio>;
  constructor(private afs2: AngularFire) {
    this.anunciosCollection = this.afs2.collection<Anuncio>('anuncio');
  }
  //creamos las operaciones basicas de un abm para Admin
  getAnuncios() {
    return this.anunciosCollection.snapshotChanges().pipe(
      map(actions=>{
        return actions.map(a=>{
          const data=a.payload.doc.data();
          const id=a.payload.doc.id;
          return {id, ...data};
        });
      })
    );
  }
  getAnuncio(id:string) {
    return this.anunciosCollection.doc<Anuncio>(id).valueChanges();
  }
  addAnuncio(anuncio:Anuncio) {
    return this.anunciosCollection.add(anuncio);
  }
}
```

```

}
updateAnuncio(id:string,anuncio: Anuncio) {
  return this.anunciosCollection.doc<Anuncio>(id).update(anuncio);
}
deleteAnuncio(id:string) {
  return this.anunciosCollection.doc(id).delete();
}
}
}

```

5.6 Sección Código de Ubicación

```

import { Injectable } from '@angular/core';
import { AngularFireCollection, AngularFire } from '@angular/fire/firestore';
import { Ubicar } from '../interfaces/ubicar';
import { map } from 'rxjs/operators';
@Injectable({
  providedIn: 'root'
})
export class UbicarService {
  private ubicarsCollection: AngularFireCollection<Ubicar>;
  constructor(private afs2: AngularFire) {
    this.ubicarsCollection = this.afs2.collection<Ubicar>('ubicar');
  }
  //creamos las operaciones basicas de un abm para Admin
  getUbicars() {
    return this.ubicarsCollection.snapshotChanges().pipe(
      map(actions=>{
        return actions.map(a=>{
          const data=a.payload.doc.data();
          const id=a.payload.doc.id;
          return {id, ...data};
        });
      })
    );
  }
}

```

```

    });
  })
);
}

getUbicar(id:string) {
  return this.ubicarsCollection.doc<Ubicar>(id).valueChanges();
}
addUbicar(ubicar:Ubicar) {
  return this.ubicarsCollection.add(ubicar);
}
updateUbicar(id:string,ubicar: Ubicar) {
  return this.ubicarsCollection.doc<Ubicar>(id).update(ubicar);
}
deleteUbicar(id:string) {
  return this.ubicarsCollection.doc(id).delete();
}
}

```

5.7. Sección Código de Departamento

```

import { Injectable } from '@angular/core';
import { AngularFireCollection, AngularFirestore } from '@angular/fire/firestore';
import { Departamento } from '../interfaces/departamento';
import { map } from 'rxjs/operators';
@Injectable({
  providedIn: 'root'
})
export class DepartamentoService {
  private departamentosCollection: AngularFireCollection<Departamento>;
  constructor(private afs1:AngularFirestore) {

```

```

    this.departamentosCollection = this.afs1.collection<Departamento>('departamento');
  }
//creamos las operaciones basicas de un abm para Admin
getDepartamentos() {
  return this.departamentosCollection.snapshotChanges().pipe(
    map(actions=>{
      return actions.map(a=>{
        const data=a.payload.doc.data();
        const id=a.payload.doc.id;
        return {id, ...data};
      });
    })
  );
}
getDepartamento(id:string) {
  return this.departamentosCollection.doc<Departamento>(id).valueChanges();
}
addDepartamento(departamento:Departamento) {
  return this.departamentosCollection.add(departamento);
}
updateDepartamento(id:string,departamento:Departamento) {
  return this.departamentosCollection.doc<Departamento>(id).update(departamento);
}
deleteDepartamento(id:string) {
  return this.departamentosCollection.doc(id).delete();
}
}

```

5.8. Sección Código de Aplicación Cliente

Pantalla Bienvenida

```
<ion-content [fullscreen]="true">
<ion-slides>
  <ion-slide *ngFor="let slide of slides">
    <div class="slide-img-padding">
      <img [src]="slide.img">
    </div>
    <div>
      <h3 [innerHTML]="slide.titulo"></h3>
    </div>
  </ion-slide>
</ion-slides>

</ion-content >

<ion-footer no-border color="red">
<ion-grid>
  <ion-row>
    <ion-col size="10" offset="1">
      <ion-button color="medium"
        expand="full"
        shape="round"
        routerLink="/segmento"
      >
        Ir al Menu Principal
      </ion-button>
    </ion-col>
  </ion-row>
  <!--ion-row>
    <ion-col size="10" offset="1">
      <ion-button expand="full"
```

```

        fill="clear"
        mode="ios"
        class="btn-primary"
        routerLink="/regilla">
    <ion-label color="dark">Sign in</ion-label>
</ion-button>
</ion-col>
</ion-row-->
</ion-grid>
</ion-footer>

```

5.9. Sección Código de Menú Principal

```

async getEvetos(){
    let loader=this.loadingCtrl.create({
        message:"Espere por Favor.."
    });
    (await loader).present();
    try{
        // aqui hacemos la consulta de todos datos
        this.firestore
            .collection("evento")
            .snapshotChanges()
            .subscribe(data=>{
                this.eventos=data.map(e=>{
                    return{
                        id:e.payload.doc.id,
                        nombre:e.payload.doc.data()["nombre"],
                        informacion:e.payload.doc.data()["informacion"],
                        fechainicio:e.payload.doc.data()["fechainicio"],
                        fechafinal:e.payload.doc.data()["fechafinal"],
                    }
                })
            })
    }
}

```

```

        hora:e.payload.doc.data()["hora"],
        direccion:e.payload.doc.data()["direccion"],
        logotipo:e.payload.doc.data()["logotipo"]
    }
  })
});
(await loader).dismiss();
}catch(e){
  this.showToast(e);
}
}

```

5.10. Sección Código para Detalle Evento

```

constructor(
  private activeRoute:ActivatedRoute,
  private loadingCtrl:LoadingController,
  private toastCtrl:ToastController,
  private firestore:AngularFirestore
) {
  this.id=this.activeRoute.snapshot.paramMap.get('id');
}
ngOnInit() {

  this.getEventoId(this.id);
}
async getEventoId(id:string){
  let loader=this.loadingCtrl.create({
    message:"Espere por Favor.."
  });
  (await loader).present();

```

```

this.firestore
.doc("evento/" + id)
.valueChanges()
.subscribe(data=>{
  this.evento.actividad=data["actividad"];
  this.evento.celular=data["celular"];
  this.evento.direccion=data["direccion"];
  this.evento.fechafinal=data["fechafinal"];
  this.evento.fechainicio=data["fechainicio"];
  this.evento.hora=data["hora"];
  this.evento.informacion=data["informacion"];
  this.evento.logotipo=data["logotipo"];
  this.evento.nombre=data["nombre"];
  this.evento.organiza=data["organiza"];
  this.evento.userId=data["userId"];
});
(await loader).dismiss();
}

```

5.11. Sección Código de Área

```

async getAreas(){
  let loader=this.loadingCtrl.create({
    message:"Espere por Favor.."
  });
  (await loader).present();

  try{
    // aqui hacemos la consulta de todos datos
    this.firestore
      .collection("area")

```



```

.snapshotChanges()
.subscribe(data=>{
  this.areas=data.map(e=>{
    return{
      id:e.payload.doc.id,
      nombre:e.payload.doc.data()["nombre"],
      detalle:e.payload.doc.data()["detalle"],
      logo:e.payload.doc.data()["logo"],
    }
  })
});
(await loader).dismiss();
}catch(e){
  this.showToast(e);
}
}
}

```

5.12 Sección Código para Detalle Área

```

async getAreaId(id:string){
  let loader=this.loadingCtrl.create({
    message:"Espere por Favor.."
  });
  (await loader).present();
  this.firestore
    .doc("area/" + id)
    .valueChanges()
    .subscribe(data=>{
      this.area.archivo=data["archivo"];
      this.area.detalle=data["detalle"];
    }
  );
}

```

```

    this.area.fecha=data["fecha"];
    this.area.logo=data["logo"];
    this.area.nombre=data["nombre"];
    this.area.userId=data["userId"];

    });
    (await loader).dismiss();
  }

```

5.13. Sección Código de Archivos

```

openUrl(dir:string, target:string){
  const url=dir;
  //this.browser.create('https://www.google.com','_self');
  this.browser.create(url,target);
  console.log(url);
}

```

5.14. Sección Código de Noticias

```

async getNoticias(){
  let loader=this.loadingCtrl.create({
    message:"Espere por Favor.."
  });
  (await loader).present();
  try{
    // aqui hacemos la consulta de todos datos
    this.firestore
      .collection("anuncio")

```

```

.snapshotChanges()
.subscribe(data=>{
  this.noticias=data.map(e=>{
    return{
      id:e.payload.doc.id,
      descripcion:e.payload.doc.data()["descripcion"],
      archivo:e.payload.doc.data()["archivo"],
      fecha:e.payload.doc.data()["fecha"],
      imagen:e.payload.doc.data()["imagen"],
    }
  })
});
(await loader).dismiss();
}catch(e){
  this.showToast(e);
}
}

```

5.15. Sección Código para Detalle de Noticia

```

async getAnuncioId(id:string){
  let loader=this.loadingCtrl.create({
    message:"Espere por Favor.."
  });
  (await loader).present();
  this.firestore
    .doc("anuncio/" + id)
    .valueChanges()
    .subscribe(data=>{
      this.anuncio.archivo=data["archivo"];
    }
  );
}

```

```

    this.anuncio.descripcion=data["descripcion"];
    this.anuncio.fecha=data["fecha"];
    this.anuncio.imagen=data["imagen"];
    this.anuncio.nombre=data["nombre"];
    this.anuncio.userId=data["userId"];

    });
    (await loader).dismiss();
  }

```

5.16. Sección Código para Dirección del Evento

```

getGeo() {
  if (!this.post.posicion) {
    this.post.coords = null;
    return;
  }
  this.cargandoGeo = true;
  this.geolocation.getCurrentPosition().then((resp) => {
    // resp.coords.latitude
    // resp.coords.longitude
    this.cargandoGeo = false;
    const coords = `${resp.coords.latitude},${resp.coords.longitude}`;
    console.log(coords);
    this.post.coords = coords;
    /*this.lat = resp.coords.latitude;
    this.lng = resp.coords.longitude;
    console.log(this.lat, this.lng);
    */
  }).catch((error) => {
    console.log('Error getting location', error);
  });
}

```

```

    this.cargandoGeo = false;
  });
  console.log(this.post);
}

```

5.17. Sección Código para Menú Desplegable

```

<ion-app>
  <ion-menu side="start" content-id="principal" menuId="first" >
    <ion-header>
      <ion-toolbar color="danger">
        <ion-avatar slot="start">
          
        </ion-avatar>
        <p> LIGA BOLIVIANA DE ROBÓTICA</p>

        <!--ion-title></ion-title-->
      </ion-toolbar>
    </ion-header>
    <ion-content>
      <ion-list>
        <ion-menu-toggle>
          <ion-item routerLink="/listarea">
            <ion-icon color="danger" slot="start" name="card"></ion-icon>
            Areas
          </ion-item>
          <ion-item routerLink="/listanuncio">
            <ion-icon color="danger" slot="start" name="mail"></ion-icon>
            Anuncios
          </ion-item>
          <ion-item routerLink="/listaparticipa">

```

```

    <ion-icon color="danger" slot="start" name="people"></ion-icon>
    Participantes
  </ion-item>
  <ion-item routerLink="/galeria">
    <ion-icon color="danger" slot="start" name="images"></ion-icon>
    Galeria de Fotos
  </ion-item>
  <ion-item routerLink="/regilla">
    <ion-icon color="danger" slot="start" name="location"></ion-icon>
    Dirección
  </ion-item>
  <ion-item (click)="logout()">
    <ion-icon color="danger" slot="start" name="power"></ion-icon>
    Salir
  </ion-item>
</ion-menu-toggle>
</ion-list>
</ion-content>
</ion-menu>
<ion-router-outlet id="principal"></ion-router-outlet>
</ion-app>

```

CONCLUSIONES
Y
RECOMENDACIONES

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- El sistema móvil cuenta con tres aplicaciones, el primer módulo es para el manejo administrativo del evento, el segundo módulo es para la manipulación de tutor y el tercer módulo es para el manejo del participante, su interfaz es amigable e interactiva para la fácil manipulación.
- La aplicación del módulo administrativo permite el registro de la información de la persona, auspiciadores, áreas, información del evento, anuncios y el registro de la ubicación del evento también permite ver el reporte del participante y reporte del cobro del organizador.
- La aplicación para el módulo tutor permite la inscripción del participante y el registro del comprobante de pago del participante.
- La aplicación en el módulo del participante permite la visualización de la información de la competencia de la Liga Boliviana de Robótica del departamento de Tarija, se puede apreciar las noticias, información del evento, áreas de la competencia y los auspiciadores que apoyan a los diferentes eventos competitivos; también se visualizan a los participantes concursantes, la ubicación del evento de la competencia y una galería de fotografías para destacar los momentos más importantes.

6.2. Recomendaciones

- La aplicación móvil deberá subirse en Google Play Store para su acceso o podrá adquirirse de forma directa en la oficina organizadora, de la Liga Boliviana de Robótica del Departamento de Tarija.
- La aplicación móvil podrá ser instalada en celulares que cuenten con el sistema operativo Android versión 7 en adelante para su mejor funcionamiento.
- La aplicación móvil hace uso sólo de 200 registros gratuitos en la base de datos de la nube, si se llegara a pasar este rango, se tendrá que pagar para el aumento de espacio y tener una óptima funcionalidad.

- Si se desea ampliar otro módulo en el administrador se debe tener en cuenta que sea lo más específico posible según la necesidad.