

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA SISTEMAS

DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA ADMINISTRACIÓN DE UNA ESCUELA DE BALONCESTO

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

EDGAR DAVID GUAMÁN RUBIO

edgar.guaman01@epn.edu.ec

BRYAN ANDRÉS VIZUETE CHÁVEZ

bryan.vizquete@epn.edu.ec

DIRECTOR: MSC. MARÍA MONSERRATE INTRIAGO PAZMIÑO

monserrate.intriago@epn.edu.ec

Quito, julio 2022

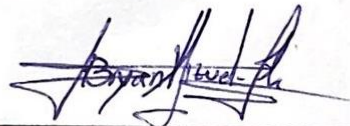
DECLARACIÓN

Nosotros, **Edgar David Guamán Rubio** y **Bryan Andrés Vizúete Chávez**, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.



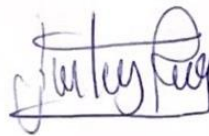
Edgar David Guamán Rubio



Bryan Andrés Vizúete Chávez

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por **Edgar David Guamán Rubio** y **Bryan Andrés Vizúete Chávez**, bajo mi supervisión.



MSc. María Monserrate Intriago Pazmiño
DIRECTORA DEL PROYECTO

DEDICATORIA – EDGAR GUAMÁN

Dedico este trabajo a mi familia, en especial a mis abuelitos Jorge Ramón Rubio Duque y Clara Elena Orozco, quienes me han brindado su amor y me han apoyado a lo largo de toda mi vida sin pedir nada a cambio.

Mención especial para mi mami Susana Rubio, que me ha acompañado en momentos de felicidad y tristeza, dándome siempre una palabra de aliento que me ha permitido superar todo tipo de obstáculos.

También debo agradecer a mi padre Edgar Guamán Córdova, que forjó mi carácter y me supo transmitir todas las vivencias y conocimiento que él acumuló en toda su vida.

Para el resto de mi familia, tíos, tías, primos, primas, etc., que no me tomen como ejemplo, pero que vean que, con esfuerzo y dedicación, se pueden alcanzar los objetivos que uno se plantea.



Edgar David Guamán Rubio

DEDICATORIA – BRYAN VIZUETE

A mis padres Pablo y Esthela por su cariño y motivación para cumplir con este sueño tan anhelado ya que han sido los pilares fundamentales de mi motivación y esfuerzo, gracias al apoyo que me han brindado durante toda la carrera he podido lograr llegar a concluir esta etapa de mi formación profesional.

Para ustedes con todo mi amor.

A handwritten signature in black ink, appearing to read 'Bryan Vizueté Chávez', with stylized flourishes and a horizontal line underneath.

Bryan Andrés Vizueté Chávez

AGRADECIMIENTO – EDGAR GUAMÁN

A la Escuela Politécnica Nacional, por tener profesores de altísima calidad que me supieron transmitir su conocimiento para poder aplicarlo en la vida laboral.

A mis amigos de la facultad, con los que compartí momentos muy amenos y que con su sola presencia hacían que la vida universitaria sea mucho más llevadera y digna de recordar.

También agradezco a mi directora de tesis MSc. Monserrate Intriago, por su paciencia, su apoyo y motivación para lograr terminar este trabajo de titulación.

Edgar David Guamán Rubio

AGRADECIMIENTO – BRYAN VIZUETE

A mis padres Pablo y Esthela un infinito agradecimiento por darme la mejor educación, inculcándome sus valores y enseñanzas, gracias a ustedes soy la persona que soy, por siempre creer en mí apoyándome en cada paso que doy.

A mis hermanos, por todas sus palabras de aliento para seguir adelante y no rendirme por siempre estar ahí pendientes de mí.

A la MSc. Monserrate Intriago, directora de tesis, por todo su apoyo y paciencia para culminar con este proceso de titulación

A mi alma mater la Escuela Politécnica Nacional y a todos mis profesores, por brindarme esos conocimientos necesarios y experiencias para mi desarrollo profesional.

A mi familia y amigos por estar presentes en todo el desarrollo de mi vida, por el compartir momentos y experiencias inolvidables.

Bryan Andrés Vizúete Chávez

ÍNDICE DE CONTENIDO

RESUMEN	1
ABSTRACT	2
Capítulo 1. Introducción	3
1.1. Planteamiento del problema	3
1.2. Justificación teórica	5
1.3. Justificación metodológica	5
1.4. Justificación práctica	6
1.5. Objetivos	7
1.5.1. Objetivo general	7
1.5.2. Objetivos específicos	7
1.6. Marco de referencia	7
1.7. Alcance	8
1.8. Marco teórico	9
1.8.1. Aplicación móvil	9
1.8.2. Sistema operativo móvil	9
1.8.2.1. Comparación de sistemas operativos móviles	9
1.8.3. Framework para desarrollo móvil	11
1.8.4. Gestión administrativa	15
1.8.5. Metodología ágil	15
1.9. Estructura del documento	15
Capítulo 2. Metodología	16
2.1. Herramientas y tecnologías utilizadas	16
2.2. Historias de usuario	18
2.3. Arquitectura de la aplicación	19
2.4. Desarrollo de la aplicación	20

2.4.1. Recopilación de requisitos	20
2.4.2. Características del aplicativo	20
2.4.3. Product backlog	21
2.4.4. Sprint 0	29
2.4.4.1. Objetivos del sprint	29
2.4.4.2. Planificación del sprint	30
2.4.4.3. Implementación	31
2.4.4.4. Sprint review	34
2.4.5. Sprint 1	35
2.4.5.1. Objetivos del sprint	35
2.4.5.2. Planificación del sprint	35
2.4.5.3. Implementación	40
2.4.5.4. Sprint review	49
2.4.6. Sprint 2	52
2.4.6.1. Objetivo del sprint	52
2.4.6.2. Planificación del sprint	52
2.4.6.3. Implementación	54
2.4.6.4. Sprint review	58
2.4.7. Sprint 3	60
2.4.7.1. Objetivo del sprint	60
2.4.7.2. Planificación del sprint	60
2.4.7.3. Implementación	62
2.4.7.4. Sprint review	64
2.4.8. Sprint 4	66
2.4.8.1. Objetivo del sprint	66
2.4.8.2. Planificación del sprint	66
2.4.8.3. Implementación	70

2.4.8.4. Sprint review	73
2.4.9. Sprint 5	75
2.4.9.1. Objetivo del sprint	75
2.4.9.2. Planificación del sprint	76
2.4.9.3. Implementación	77
2.4.9.4. Sprint review	80
2.4.10. Sprint 6.....	81
2.4.10.1. Objetivo del sprint.....	81
2.4.10.2. Planificación del sprint	81
2.4.10.3. Implementación	83
2.4.10.4. Sprint review	85
Capítulo 3. Resultados y discusión	87
3.1. Pruebas de usabilidad	87
3.2. Sistema de escala de usabilidad.....	88
3.3. Casos de prueba	89
Capítulo 4. Conclusiones y recomendaciones.....	98
4.1. Conclusiones	98
3.2. Recomendaciones.....	99
Glosario de términos.....	100
Referencias bibliográficas	102
Anexos.....	104

ÍNDICE DE TABLAS

Tabla 1. Comparación de sistemas operativos móviles	9
Tabla 2. Comparación de aplicaciones nativas y móviles.....	14
Tabla 3. Herramientas para el desarrollo de la aplicación móvil.....	17
Tabla 4. Estructura de una historia de usuario.....	18
Tabla 5. Product backlog	21
Tabla 6. Historia de usuario MONSAPP-1.....	30
Tabla 7. Historia de usuario MONSAPP-2.....	31
Tabla 8. Historia de usuario MONSAPP-3.....	31
Tabla 9. Revisión sprint 0.....	34
Tabla 10. Historia de usuario MONSAPP-4.....	36
Tabla 11. Historia de usuario MONSAPP-5.....	37
Tabla 12. Historia de usuario MONSAPP-6.....	37
Tabla 13. Historia de usuario MONSAPP-7.....	38
Tabla 14. Historia de usuario MONSAPP-8.....	38
Tabla 15. Historia de usuario MONSAPP-9.....	38
Tabla 16. Historia de usuario MONSAPP-10.....	39
Tabla 17. Historia de usuario MONSAPP-11.....	40
Tabla 18. Revisión sprint 1.....	50
Tabla 19. Historia de usuario MONSAPP-12.....	53
Tabla 20. Historia de usuario MONSAPP-13.....	53
Tabla 21. Historia de usuario MONSAPP-14.....	54
Tabla 22. Revisión sprint 2.....	59
Tabla 23. Historia de usuario MONSAPP-15.....	60
Tabla 24. Historia de usuario MONSAPP-16.....	61
Tabla 25. Historia de usuario MONSAPP-17.....	61

Tabla 26. Revisión sprint 3.....	65
Tabla 27. Historia de usuario MONSAPP-18.....	67
Tabla 28. Historia de usuario MONSAPP-19.....	67
Tabla 29. Historia de usuario MONSAPP-20.....	68
Tabla 30. Historia de usuario MONSAPP-21.....	68
Tabla 31. Historia de usuario MONSAPP-22.....	69
Tabla 32. Historia de usuario MONSAPP-23.....	69
Tabla 33. Historia de usuario MONSAPP-24.....	69
Tabla 34. Revisión sprint 4.....	74
Tabla 35. Historia de usuario MONSAPP-25.....	76
Tabla 36. Historia de usuario MONSAPP-26.....	77
Tabla 37. Historia de usuario MONSAPP-27.....	77
Tabla 38. Revisión sprint 5.....	81
Tabla 39. Historia de usuario MONSAPP-28.....	82
Tabla 40. Historia de usuario MONSAPP-29.....	82
Tabla 41. Revisión sprint 6.....	86
Tabla 42. Preguntas del cuestionario de usabilidad	88
Tabla 43. Caso de prueba 1.....	89
Tabla 44. Caso de prueba 2.....	90
Tabla 45. Caso de prueba 3.....	90
Tabla 46. Caso de prueba 4.....	91
Tabla 47. Caso de prueba 5.....	92
Tabla 48. Caso de prueba 6.....	92
Tabla 49. Caso de prueba 7.....	93
Tabla 50. Caso de prueba 8.....	93
Tabla 51. Caso de prueba 9.....	94
Tabla 52. Caso de prueba 10.....	94

Tabla 53. Caso de prueba 11.....	95
Tabla 54. Caso de prueba 12.....	95
Tabla 55. Caso de prueba 13.....	96
Tabla 56. Caso de prueba 14.....	96

ÍNDICE DE FIGURAS

Figura 1. Framework Cordova.....	11
Figura 2. Framework IONIC.	12
Figura 3. Arquitectura de la aplicación.	19
Figura 4. Planificación del sprint 0.....	30
Figura 5. Sitio de descarga de Visual Studio Code.	32
Figura 6. Vista del repositorio en GitHub.....	32
Figura 7. Página de descarga de NodeJS.....	33
Figura 8. Vista del proyecto en Firebase.....	33
Figura 9. Finalización del sprint 0.....	34
Figura 10. Planificación del sprint 1.....	36
Figura 11. Bosquejos de las pantallas de la aplicación móvil.	41
Figura 12. HTML para registrar un nuevo usuario.	42
Figura 13. Servicio para crear un nuevo usuario.....	43
Figura 14. Métodos para crear o actualizar un usuario.....	43
Figura 15. Método para validar una cuenta.	44
Figura 16. HTML para iniciar sesión.....	44
Figura 17. Servicio para iniciar sesión.....	45
Figura 18. HTML para editar los datos del usuario.....	45
Figura 19. Servicio para obtener los datos del usuario desde la base.....	46
Figura 20. Interfaz gráfica de la aplicación.	47
Figura 21. Interfaz gráfica de inicio de sesión y creación de usuarios.	48
Figura 22. Barra lateral de la aplicación móvil.....	49
Figura 23. Finalización del sprint 1.....	50
Figura 24. Planificación del sprint 2.....	53
Figura 25. HTML para la creación de un partido.	55

Figura 26. HTML para la creación de un entrenamiento.....	55
Figura 27. HTML para el manejo de jugadores	56
Figura 28. Vistas del módulo de entramiento.	57
Figura 29. Vistas para listas jugadores y editarlos.	58
Figura 30. Finalización del sprint 2.....	59
Figura 31. Planificación del sprint 3.....	60
Figura 32. HTML para listar partidos.	62
Figura 33. Servicio para el manejo de usuarios.....	63
Figura 34. Método para mostrar partidos y entrenamientos por jugador.....	63
Figura 35. Interfaz para comunicarse con el administrador.	64
Figura 36. Revisión del sprint 3.	65
Figura 37. Planificación del sprint 4.....	67
Figura 38. HTML para editar datos del usuario.	70
Figura 39. Servicio para editar datos del entrenador.	70
Figura 40. Interfaz de partidos y entrenamientos.	71
Figura 41. Uso de Google Maps en la aplicación.	72
Figura 42. Partidos asignados a un jugador.	72
Figura 43. Iconos para acceder a las redes sociales.....	73
Figura 44. Revisión del sprint 4.	74
Figura 45. Planificación del sprint 5.....	76
Figura 46. HTML para ingresar un equipo rival.	78
Figura 47. HTML para recuperar la contraseña.	78
Figura 48. HTML con las opciones de la barra lateral.	79
Figura 49. Interfaz para crear equipos rivales.	80
Figura 50. Revisión del sprint 5.	80
Figura 51. Planificación del sprint 6.....	82
Figura 52. HTML para el reporte de jugadores.....	83

Figura 53. HTML para el reporte de torneos.	84
Figura 54. Vista del reporte de jugadores.....	85
Figura 55. Revisión del sprint 6.....	86

RESUMEN

En el presente trabajo de titulación se desarrolla una aplicación móvil para la gestión de la escuela de baloncesto Monsters. Esta aplicación permite crear, editar y eliminar jugadores de baloncesto, de entrenadores, de partidos, de equipos rivales y de torneos.

El aplicativo fue desarrollado haciendo uso del marco de trabajo Scrum. Este marco de trabajo, que pertenece a las metodologías ágiles, permitió aplicar buenas prácticas a lo largo de todo el proceso de desarrollo del aplicativo. La utilización de roles, artefactos, flujos de trabajo y documentos propios de Scrum facilitó la gestión de cambios e inclusión de nuevos requisitos al proceso de desarrollo. Este proyecto fue dividido en sprints o iteraciones; cada sprint tuvo una duración de dos semanas y al final del proceso de desarrollo se contabilizó 5 sprints.

Para el desarrollo de las funcionalidades y de la aplicación en si se utilizó el lenguaje de programación Typescript, que es un superconjunto de Javascript. Para el manejo y persistencia de datos, se utilizó Firebase RealTime Database, una base de datos norelacional ofrecida por Google. Finalmente, para la compilación a una aplicación móvil, se hizo uso del marco de trabajo Ionic, el cual permite el desarrollo de aplicaciones híbridas.

Por último, para el proceso de adaptación del aplicativo, el cual fue mucho más fácil gracias a las entregas de un prototipo al final de cada sprint, se aplicó pruebas de usabilidad y se dio un tutorial de todas las funcionalidades a todos los miembros de la escuela Monsters.

Palabras clave: baloncesto, Scrum, Ionic, Typescript, Firebase, aplicación móvil.

ABSTRACT

In this academic article, a mobile application is developed for the management of the Monsters basketball school. This application allows you to create, edit and delete basketball players, coaches, games, rival teams, and tournaments.

The application was developed using the Scrum framework. This framework, which belongs to agile methodologies, allowed the application of good practices throughout the entire development process of the application. The use of Scrum roles, artifacts, workflows, and documents facilitated the management of changes and the inclusion of new requirements in the development process. This project was divided into sprints or iterations; each sprint lasted two weeks and at the end of the development process 5 sprints were counted.

For the development of the functionalities and the application itself, Typescript programming language was used, which is a superset of Javascript. For data management and persistence, Firebase RealTime Database, a non-relational database offered by Google, was used. Finally, for the compilation to a mobile application, the Ionic framework was used, which allows the development of hybrid applications.

Finally, for the process of adapting the application, which was much easier thanks to the delivery of a prototype at the end of each sprint, usability tests were applied and a tutorial on all the functionalities was given to all members of the Monsters school.

Keywords: basketball, Scrum, Ionic, Typescript, Firebase, mobile application.

Capítulo 1. Introducción

1.1. Planteamiento del problema

El deporte es fundamental en la vida de cualquier ser humano, ya que permite a las personas ejercitar su organismo para mantenerlo en un buen estado físico, así como también les permite relajarse, distenderse, despreocuparse de la rutina, liberar tensión y sobre todo distraerse [1].

La Organización Mundial de la Salud (OMS) describe a la actividad física como cualquier movimiento corporal producido por los músculos esqueléticos que exijan un gasto de energía [2]. La OMS también da a conocer que la falta de actividad física representa el 6% de las muertes mundiales. Esta causa solo es superada por la hipertensión (13%) y el consumo de tabaco (9%).

El baloncesto es un deporte que fomenta el desarrollo físico y personal de los niños, jóvenes y adultos. Enseña valores como el compañerismo, el trabajo en equipo, la generosidad, y ayuda al desarrollo del aparato locomotor y el fortalecimiento de los músculos.

La historia del baloncesto en Ecuador según Lic. Juan Albino Astudillo Avilés en su obra "Lauro, historia del deporte ecuatoriano" [3], además de obras, como: Memorias del Deporte, Baloncesto (2011, Ministerio del Deporte) señalan que el origen del baloncesto en el Ecuador es introducido por los norteamericanos en el transcurso de los años 1900-1930 en Portovelo, provincia del El Oro.

A partir de esa década se fueron conformando varias agrupaciones tanto en Quito como en Guayaquil, se construyó el primer coliseo cubierto en el país y a partir de ello se fueron realizando varios torneos nacionales, posteriormente internaciones y fue así como en 1950 Ecuador participa en el primer campeonato mundial de baloncesto masculino.

En la actualidad en el Ecuador existe la Federación Ecuatoriana de Basquetbol (FEB) y varias escuelas de baloncesto. La creación de escuelas deportivas y recreativas para niños y adolescentes tiene el fin de construir una cultura de equidad con igualdad de género, integración social y comunitaria que establezca

un mecanismo de salud preventiva y buen uso del tiempo libre, a través de la educación en valores [4]. Existen escuelas deportivas pertenecientes al municipio del Distrito metropolitano de Quito las cuales se encuentran ubicadas en cada uno de los barrios, parroquias urbanas y rurales del cantón Quito.

El negocio de las academias de baloncesto ha crecido cada vez más debido a la gran cantidad de personas que quieren practicar este deporte de una forma personal y profesional, es así como la competencia de estos centros deportivos hoy en día es muy demandante. Cada escuela tiene su reputación que la caracteriza y diferencia de las demás. Monsters es una escuela de formación de baloncesto cuya misión es formar a sus estudiantes de una forma integral inculcando valores como la disciplina, honestidad y el esfuerzo. Fue fundada en el año 2013, cuenta con alrededor de 60 estudiantes, y actualmente realiza sus entrenamientos en el parque de la Mujer en Quito. La escuela de baloncesto tiene varias categorías y participa en torneos como el de San Juan, Eufrasia, Carolina Sub basquetero y otros torneos que realiza la federación ecuatoriana de baloncesto [5].

En el tema administrativo los procesos de matriculación, cobros y control de asistencia, actualmente se manejan a través de archivos Excel, documentos en papel y archivos de WhatsApp. Esto dificulta realizar un adecuado seguimiento y control a todos los estudiantes.

La escuela no puede implementar rutinas de entrenamiento personalizadas de acuerdo con el nivel de cada estudiante que le permitan monitorear su avance y desempeño, debido a la falta de herramientas tecnológicas que optimicen este proceso.

Los padres de familia no tienen información oportuna referente al calendario de juegos y actividades realizadas por parte de la escuela en sus diferentes categorías, para solucionar estos problemas se presenta el desarrollo de esta aplicación que permitirá optimizar los procesos administrativos de la escuela de baloncesto Monsters, además facilitará a la escuela implementar rutinas de entrenamiento de acuerdo con el nivel del jugador y centralizará la información que necesitan los padres de familia.

1.2. Justificación teórica

En el proceso de desarrollo del aplicativo se hizo uso del framework Ionic, incluyendo también herramientas y procedimientos que provee la ingeniería de software a fin de obtener una aplicación que cumpla con buenas prácticas de desarrollo y a la vez sea atractiva para el usuario final. Se describen y analizan los principales paradigmas del ciclo de vida de la aplicación móvil, desde la cascada tradicional hasta los actuales orientados a objetos, a fin de seleccionar el ciclo de vida adecuado para cada desarrollo, se analizan las actividades de cada uno de los paradigmas [6].

A través de una recopilación y análisis de los principales modelos de desarrollo de software existentes, se puede facilitar la elección de un modelo apropiado para cada proyecto en particular. Una buena elección del paradigma de programación ahorra tiempo y mejora la calidad de los sistemas que se producen [7]. Estos son de gran importancia para el desarrollo de los procesos de una organización ya que ayudan a escoger una estrategia para el desarrollo de software, cada paradigma tiene su propio set de herramientas, métodos y procedimientos, los cuales son expresados de forma clara a la vez que se va definiendo el ciclo de vida de software.

Existen muchos frameworks para el desarrollo de aplicación híbridas como Angular que es un framework de JavaScript de código abierto que está escrito en TypeScript, así también está React que es una biblioteca de JavaScript para construir interfaces de usuario componibles, fomenta la creación de componentes reutilizables para interfaces de usuario que muestran datos que cambian con el tiempo. Inicialmente React se concibió como la V del modelo MVC (Modelo, Vista, Controlador), pero Facebook presentó su propia arquitectura FLUX que se compone de 4 elementos claves: views, stores, actions y dispatcher [8].

1.3. Justificación metodológica

Se ha escogido la metodología Scrum porque la escuela de baloncesto Monsters actualmente no posee ningún sistema que le permita definir a detalle el funcionamiento deseado en la aplicación, por lo cual se requiere una metodología

que facilite tener un contacto permanente con el cliente para ajustar continuamente los requerimientos en cada iteración.

Scrum nos permite realizar entregas parciales y regulares del producto final, priorizadas por el beneficio que aporta al cliente. La metodología se ejecuta en ciclos temporales cortos que pueden ligeramente variar de dos semanas a un mes, y estos ciclos se les conoce normalmente como iteraciones [9].

El proceso de diseño e implementación de esta metodología está dividido en 3 etapas:

- **Planificación de la iteración:** Etapa en donde nos reuniremos con un representante de la empresa para identificar los requerimientos del aplicativo. Así mismo se establecerá una planificación sobre el listado de las tareas prioritarias para la realización en cada iteración.
- **Ejecución:** Se realizarán reuniones diarias de máximo 15 minutos para evaluar la evolución de las tareas designadas y de esta manera cumplir con los objetivos de cada interacción.
- **Inspección y adaptación:** En esta etapa se presenta un resultado de la iteración, teniendo en cuenta posibles cambios que deberán ser adaptados para la siguiente iteración.

1.4. Justificación práctica

La transformación digital móvil es una necesidad y una realidad para las empresas que quieran prosperar entre un público cada vez más centrado y enfocado a un smartphone.

Según un estudio de GuiaLocal.com, el uso de dispositivos móviles creció 61% en países de América Latina, mientras que el uso de computadores para conectarse a Internet registra una caída de 11.3%. Ecuador ocupa el quinto lugar en las estadísticas de uso de dispositivos móviles con un 75% de crecimiento anual [10]. Según los últimos datos del Instituto Nacional de Estadística y Censos (INEC), el 76,8% de la población utiliza los dispositivos móviles para el aprendizaje y la educación [11].

Debido a que la mayoría de las funciones requeridas por el cliente se realizan al aire libre y los usuarios tienen acceso a un teléfono inteligente, se ha determinado la pertinencia de desarrollar una aplicación móvil. También, al realizar una aplicación híbrida se puede abarcar la mayor cantidad de usuarios posibles.

La ventaja de utilizar un aplicativo móvil dentro de una organización es que facilita la optimización de los procesos y a su vez es un atractivo para atraer a nuevos clientes, también ayuda a la caracterización de una organización en cuanto al servicio que ofrece y como lo ofrece. Es así como se pretende crear una aplicación llamativa, fácil de utilizar y que brinde al usuario la mejor experiencia posible.

1.5. Objetivos

1.5.1. Objetivo general

- Desarrollar un aplicativo móvil para dispositivos Android y iOS que permita administrar los procesos de las escuelas de baloncesto Monsters.

1.5.2. Objetivos específicos

- Analizar la situación actual de los procesos de la escuela de baloncesto Monsters.
- Definir los requerimientos funcionales para el desarrollo de la aplicación móvil.
- Diseñar la aplicación móvil.
- Desarrollar el sistema.
- Realizar las pruebas funcionales del aplicativo móvil.
- Publicar la aplicación.

1.6. Marco de referencia

Las aplicaciones móviles hoy en día son un requisito comercial para añadir gestión en las fuerzas de ventas, logrando así de manera directa un crecimiento económico notable y al mismo tiempo una superación empresarial.

El Instituto de Investigación de informática realizó un artículo científico para identificar el crecimiento de los dispositivos móviles y de bajo costo, teniendo como ventaja que la aplicación es híbrida, apta para el sistema Android e iOS como respuesta a la necesidad de desarrollarla para ambas plataformas, por lo que, estas aplicaciones actualmente están siendo novedosas en diferentes enfoques; los autores mencionaron que las WPA son de mejor opción por lo que se instalan y se ejecutan desde la tienda de aplicaciones de Google, su modo de funcionamiento es sin internet, además se puede incorporar características como las notificaciones y se la puede instalar en computadoras de escritorio, por lo que, vieron factible unificarlas sin importar el dispositivo y sistema operativo.

Según su trabajo de investigación, el cual plantea desarrollar una aplicación móvil para ayudar en la gestión deportiva de un equipo de baloncesto, ayudando también a llevar la información de los partidos disputados por parte del equipo de baloncesto, esta aplicación está desarrollada en un lenguaje nativo que es Android, concluyendo que con el desarrollo de una aplicación móvil para la gestión de baloncesto ayuda significativamente en los procesos y disminuye costos para el equipo ya que se necesita menos personal para llevar la gestión de los partidos y entrenamientos que disputa el equipo de baloncesto.

1.7. Alcance

En el presente trabajo se desarrolla una aplicación móvil híbrida para la gestión administrativa en una escuela de baloncesto. Al decir gestión administrativa se hace referencia al manejo de jugadores, que abarca la creación, edición, actualización y eliminación de estos, lo mismo en el caso de los entrenadores. Así mismo también se mejora el proceso de comunicación entre los entrenadores y los jugadores, ya que se implementa el control de partidos y entrenamientos.

Incluso va más allá, dando la posibilidad de una comunicación con el gerente de la escuela de baloncesto, ya sea mediante un mensaje de texto, una llamada o un correo electrónico. Otra característica que resalta en el aplicativo es la posibilidad de acceder a las redes sociales y también ver los campeonatos de la escuela. Este aplicativo beneficia directamente a todos los miembros de la escuela, desde el

gerente general, los entrenadores, y hasta los jugadores que practican este deporte.

Se puede afirmar que va a ser una herramienta para optimizar el proceso de la gestión de la información. Además, el diseño será intuitivo y de fácil de uso para los usuarios que la utilicen de modo que, se optimice el tiempo en las tareas al administrar la información.

1.8. Marco teórico

1.8.1. Aplicación móvil

Se define como un software escrito para diferentes dispositivos móviles, mismos que ejecutan una tarea específica como por ejemplo un reproductor musical o de video, juego, programa, calendario, etc. Para acceder el usuario deberá descargarlo e instalarlo en su teléfono o tableta.

1.8.2. Sistema operativo móvil

Permite el control del dispositivo móvil, similar a un sistema instalado en un computador, sea este Windows, Linux, etc. Sin embargo, estos sistemas operativos móviles son más simples y orientados a una conexión inalámbrica. Entre las características se menciona [12]:

- Brindar un entorno conveniente para trabajar.
- Proveer de una distribución correcta de los recursos.
- Eficiencia en el uso del hardware.
- Controlar y administrar la ejecución de los diferentes programas.

1.8.2.1. Comparación de sistemas operativos móviles

Tabla 1. Comparación de sistemas operativos móviles

Sistema Operativo	Android	iOS	Windows Phone
Kernel	Linux	OS X	Windows NT

Tipos de SO	Abierto	Cerrado	Cerrado
Lenguaje de Programación nativo	Java	Objective C	C#
Seguridad	Muy Buena	Susceptible a Malware	Muy buena
Adaptabilidad	Excelente	Excelente	Excelente
Multitarea	Si	Si	Si
Estándar soportado	GSM, CDMA	GSM, CDMA	GSM, CDMA
Hardware	Amplia gama de dispositivos	iPhone, iPad, iPod touch	Limitada gama de dispositivos
Programa de productividad	Google Docs	iWork	Office Mobile
Tienda de software	Google Play	App Store	Marketplace
Apps	500000+	650000+	100000+
Soporte para tableta	Si	Si	Si
Expansión de almacenamiento	Micro SD	No	No
Soporte en la Nube	Google Cloud	iCloud	SkyDrive
Interfaz de Usuario	Más técnico	Fácil	Fácil
Asistente de voz	Google Assistant	Siri	Cortana
Personalización	Amplia	Limitada	Ninguna
Vida de la batería	Media duración	Media duración	Media duración

Fuente: [12]

1.8.3. Framework para desarrollo móvil

Un framework o marco de trabajo es una estructura sobre la que puede construir software. Los marcos de trabajo generalmente se asocian con un lenguaje de programación específico y se adaptan a diferentes tipos de tareas. En lo que respecta a desarrollo móvil, cada framework cuenta con todas las herramientas para visualizar la aplicación que se está construyendo e inclusive corregirla en tiempo real. Algunos de los frameworks son:

CORDOVA

Es uno de los primeros marcos de trabajo híbrido, permite utilizar JavaScript y HTML5 para crear aplicaciones rápidas, robustas y sencillas con funcionalidad en cualquier sistema operativo [13].

El aplicativo desarrollado en Cordova se presenta como una aplicación web normal, que se usará a través de un navegador, y para que ésta sea utilizada en dispositivos móviles será encapsulada y compilada para Android y iOS. La manera de trabajar de Cordova es la siguiente:

- Construcción de la aplicación usando estándares web.
- Combinar Cordova con aplicaciones web, accediendo a características nativas.
- Configurar la app para las plataformas.

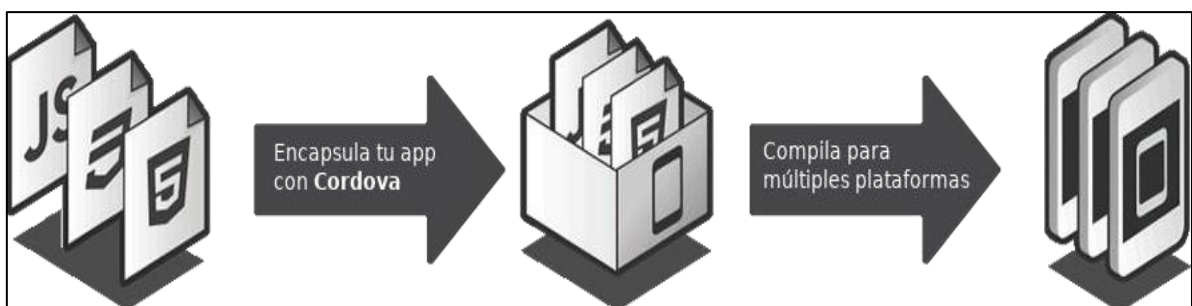


Figura 1. Framework Cordova.

Fuente: [14]

FLUTTER

Es un framework de código abierto que usa el lenguaje de programación Dart y fue desarrollado por Google. Se procura que siempre que se pueda, se reutilice código, de manera que la aplicación se torna ágil y rentable. Entre sus características se mencionan [15].

- Experiencia nativa: ayuda a los programadores en situaciones críticas como la navegación, fuentes, desplazamientos, íconos.
- Desarrollo ágil: por la variedad de elementos que proporciona permite que el desarrollador trabaje de manera más rápida en la aplicación.
- Interfaz amigable con el usuario: la interfaz que se creará debe ser atractiva e intuitiva para el usuario, brindando experiencia al momento de manejar la aplicación.

IONIC

IONIC es un framework de acceso libre y código abierto que permite combinar varias tecnologías para la creación de aplicaciones móviles de manera rápida y sencilla.

Admite el desarrollo de aplicaciones con el uso de tecnologías web, las cuales comúnmente son: JavaScript, HTML y/o estilos CSS. Para que las aplicaciones se puedan empaquetar con Android y iOS se usa Cordova [16].

La implementación de una aplicación desarrollada en IONIC permite que sea estable, sencilla y amigable con el usuario.

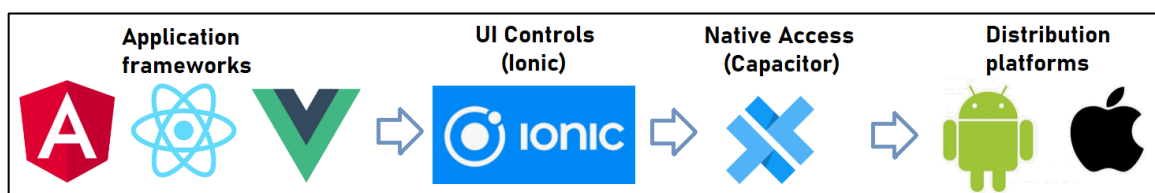


Figura 2. Framework IONIC.

Fuente: [17]

Según Esaú Abril [18] entre las ventajas de utilizar IONIC están:

- Se puede compilar para los sistemas Android y iOS.
- Soporta frameworks populares como Angularjs, Reactjs y Vuejs.
- Permite la reutilización de código.
- El desarrollo e implementación de una aplicación realizada en este framework es única tanto para Android, iOS o para la web.

REACT NATIVE

Es un framework JavaScript para crear aplicaciones reales nativas para iOS y Android, basado en la librería de JavaScript React para la creación de componentes visuales, cambiando el propósito de estos para, en lugar de ser ejecutados en navegador, correr directamente sobre las plataformas móviles nativas. Nos proporciona las siguientes funcionalidades:

- Compatibilidad multiplataforma haciendo uso del mismo código base.
- Funcionalidad nativa mejorando el rendimiento de las aplicaciones sin el uso de un WebView.
- Actualizaciones instantáneas tanto para desarrollo y/o test.
- Sencilla curva de aprendizaje ya que es intuitivo tanto para los expertos en JS o incluso para novatos [19].

XAMARIN

Desarrollado por Microsoft, gratuito y de código abierto. Las características más relevantes están [15]:

- Enlaces para SDK, su producción de errores es menor en tiempo de ejecución.
- Permite el uso de un solo código de programación mediante el enlace de bibliotecas nativas.
- Uso de lenguaje moderno como C#, funciones de lambdas, LINQ, programación en paralelo.
- Acceso a bibliotecas de clases.
- IDE moderno: control del código, administración de proyectos, plantillas, solución de errores.

- Multiplataforma: se comparte hasta el 90% del código, reduciendo tiempo y costo de desarrollo.

Tabla 2. Comparación de aplicaciones nativas y móviles.

Comparativa	Aplicaciones Nativas	Aplicaciones Híbridas
Curva de aprendizaje	Se requiere tiempo y mucha dedicación para aprender a detalle cada lenguaje de programación específico para cada sistema operativo.	El desarrollo de aplicaciones híbridas usualmente es más fácil, ya que se apoya en tecnologías web.
Exportación a distintas plataformas	Se requiere programar una aplicación para cada plataforma ya que manejan distintos lenguajes de desarrollo.	Con las aplicaciones híbridas se dan ventajas de facilidad y versatilidad para exportar la aplicación a las distintas plataformas.
Coste del desarrollo	Mayor costo ya que se desarrolla una aplicación para cada plataforma.	Menor costo debido a que una sola aplicación es desarrollada para todas las plataformas.
Facilidad de encontrar desarrolladores	Hay que buscar un desarrollador para cada plataforma.	Al ser desarrolladas con tecnologías web se cuenta con una gran cantidad de desarrolladores.
Acceso a características del dispositivo	Las aplicaciones nativas tienen el acceso total a los periféricos del dispositivo.	Las aplicaciones tienen acceso limitado o incompatibilidad con ciertos periféricos.
Aspecto visual y experiencia del usuario	Las aplicaciones nativas ofrecen la mejor experiencia de usuario y un aspecto visual inmejorable.	Las aplicaciones híbridas utilizan HTML5 y CSS para simular todo tipo de comportamientos similares a las aplicaciones nativas.

1.8.4. Gestión administrativa

Ciencia que estudia a las organizaciones, así como las técnicas de dirección, control, organización de los recursos para definir los objetivos de una empresa. La importancia de la gestión administrativa consiste en preparar a la organización y disponerla para actuar, pero de manera anticipada, contemplando todos los medios y procedimientos que necesita para cumplir con sus objetivos y disminuir los efectos negativos o posibles problemas [12].

1.8.5. Metodología ágil

Una metodología ágil es una forma de gestionar un proyecto dividiéndolo en varias fases. Implica la colaboración constante con las partes interesadas y la mejora continua en cada etapa. Una vez que comienza el trabajo, los equipos pasan por un proceso de planificación, ejecución y evaluación [20].

Entre sus ventajas se detalla [21]:

- Mejora la calidad del producto.
- Mayor satisfacción al cliente.
- Mayor motivación al personal.
- Trabajo grupal.
- Uso métricas.
- Mayor capacidad y control de predicción.
- Reducción en los costes

Así mismo, se menciona las 4 categorías de las metodologías ágiles [9]:

- Entrega de valor.
- Colaboración del negocio.
- Dinámica grupal y de cultura.
- Retrospectivas y aprendizaje de forma continua.

1.9. Estructura del documento

El presente documento está estructurado según las normas APA las cuales constan de una portada, un resumen/abstract, un cuerpo del documento que viene a ser la introducción, metodología a utilizarse, resultados y la discusión. Además, el documento contiene todas las referencias, tablas, y figuras utilizadas.

Capítulo 2. Metodología

Para el desarrollo del presente trabajo se realizó una entrevista al propietario de la escuela de baloncesto Monsters con el fin de establecer la situación actual de la escuela y hacer un levantamiento de los procesos de ésta.

Posteriormente en una segunda entrevista se definió los requerimientos funcionales que debe cumplir la aplicación para optimizar los procesos que el cliente determine, estableciendo un conjunto de características que conformarán el backlog inicial. Estas características se crearán como historias de usuario.

Se diseñó un mockup que incluya las pantallas principales de la aplicación y su navegabilidad. Con la aprobación de cliente se procederá a iniciar el desarrollo de la aplicación.

Al inicio de cada sprint se planificará el conjunto de historias de usuario a entregar, el esfuerzo requerido para culminar cada una y de paso su prioridad. Por otra parte, al final de cada sprint se ejecutarán pruebas funcionales y se realizarán entregas parciales al cliente, de manera que se pueda cumplir con todas las historias de usuario y detectar errores que modificarán el backlog.

Este proceso se ejecuta de forma iterativa hasta culminar todo el desarrollo e implementación de las historias de usuario del backlog. Una vez que el cliente se encuentra satisfecho con el producto se publicará la aplicación en Google Play Store.

2.1. Herramientas y tecnologías utilizadas

Para el desarrollo de la aplicación móvil se utilizaron las siguientes herramientas y tecnologías.

Tabla 3. Herramientas para el desarrollo de la aplicación móvil.

Herramienta	Descripción
 Git	Git se utiliza para realizar un seguimiento de los cambios en el código fuente, lo que permite que varios desarrolladores trabajen juntos en el desarrollo no lineal.
 GitHub	GitHub es una plataforma de desarrollo de software en línea que se utiliza para almacenar, rastrear y colaborar en proyectos de software.
 Balsamiq Wireframes	Balsamiq Wireframes es una herramienta para crear prototipos que muestren una interfaz de usuario en desarrollo.
 Ionic	Un conjunto de herramientas móviles de código abierto para crear experiencias de aplicaciones web y nativas multiplataforma de alta calidad.
 Angular	Angular es un framework basado en JavaScript de código abierto del lado del cliente, que se utiliza para crear aplicaciones personalizadas en HTML, CSS y Typescript.
 Node	Node.js es un entorno de tiempo de ejecución del lado del servidor de código abierto creado en el motor JavaScript V8 de Chrome.
 TypeScript	TypeScript es tanto un lenguaje como un conjunto de herramientas. Además, es un lenguaje compilado fuertemente tipado, orientado a objetos.

 Visual Studio Code	Editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. Soporta muchos lenguajes y posee una gran cantidad de útiles extensiones.
 Android Studio	Android Studio es un IDE poderoso y sofisticado, diseñado con el propósito de desarrollar, probar y empaquetar aplicaciones de Android.
 Firebase	Firebase provee un backend como servicio. Tiene herramientas de autenticación, base de datos, pruebas, notificaciones y alojamiento.

Elaborado por: Autores

Fuente Análisis

2.2. Historias de usuario

Para la elaboración del presente proyecto se ha definido historias de usuario, las mismas que explican de una forma general la funcionabilidad del software desde la perspectiva del cliente, la estructura utilizada en las historias de usuario se encuentra en la siguiente tabla:

Tabla 4. Estructura de una historia de usuario.

Código		Esfuerzo	
Nombre			
Descripción			
Prioridad			
Criterios de aceptación:			

Elaborado por: Autores

Fuente: Análisis

2.3. Arquitectura de la aplicación

La aplicación presenta una arquitectura cliente-servidor, la misma que realiza la comunicación entre los distintos componentes mediante servicios REST. Asimismo, la base de datos que utiliza es una que pertenece a la categoría de no relacional. Este tipo de base de datos no tiene un esquema de entidad – relación y no utiliza el formato de tablas para almacenar los datos de la aplicación móvil.

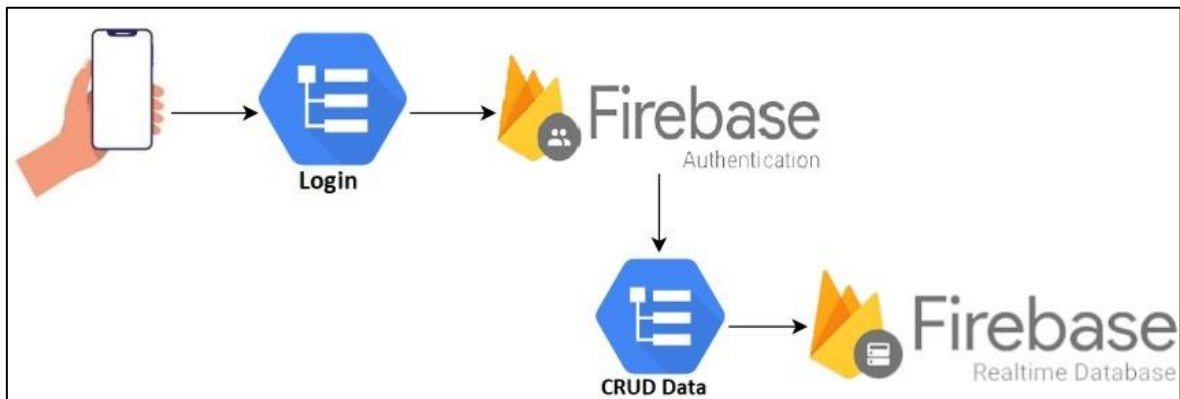


Figura 3. Arquitectura de la aplicación.

Eventos (Login y CRUD Data): El usuario realiza una petición desde la aplicación móvil. La primera petición que se envía es la de autenticación, Firebase determina si el usuario es uno válido y luego establece su rol, el cual puede ser administrador, jugador o entrenador.

Una vez autenticado el usuario y definido su rol, las siguientes peticiones que se envían corresponden a la manipulación de los datos, ya sea de jugadores, entrenadores, entrenamientos, equipos rivales, torneos o encuentros deportivos. Estas peticiones están relacionada a un método HTTP (GET, POST, PUT y DELETE).

Si se quiere conocer todos los jugadores, se envía el método GET a la colección de jugadores. Si se crea un jugador, se envía el método POST con todos los datos del nuevo jugador. Si se edita uno o todos los datos de un jugador, se envía el método PUT junto con el id del jugador. Finalmente, si se elimina un jugador, se envía el método DELETE junto con el id del jugador. Este proceso es el mismo para el resto de información (entrenadores, partidos, torneos, etc.).

Firestore authentication: Es un servicio que interpreta la petición de la aplicación móvil que contiene el usuario y la contraseña. Al decir interpreta, se hace referencia a que emite un error si encuentra inconsistencias o errores en el usuario y la contraseña enviados.

Si la información es correcta, se asigna el rol que el usuario tiene registrado en la aplicación. De acuerdo con el rol asignado, dicho usuario puede acceder a ciertos módulos en la aplicación (partidos, entrenadores, jugadores, etc.) y modificar ciertos datos que corresponden a esos módulos.

Firestore database: Permite tener una base de datos no relacional en la cual se almacena los datos de la aplicación móvil. Al ser no relacional, no se puede hablar de tablas, en su lugar se utiliza el término colecciones. Las colecciones que se tiene para la escuela de baloncesto son jugadores, entrenadores, partidos, entrenamientos, equipos rivales y torneos.

2.4. Desarrollo de la aplicación

En la presente sección se detalla los procesos involucrados en el desarrollo de la aplicación móvil, entre ellos la recopilación de requisitos para la creación del product backlog, la elección de las herramientas de desarrollo, la distribución de las historias de usuario en los distintos sprints y el detalle de la ejecución de éstas.

2.4.1. Recopilación de requisitos

Para la recopilación de requisitos se llevó a cabo un análisis de las necesidades que posee la escuela de baloncesto en cuanto a su administración, manejo de estudiantes, organización de entrenamientos y notificación de encuentros deportivos. Estos requisitos se transformaron en historias de usuario para el backlog del proyecto.

2.4.2. Características del aplicativo

En base a las reuniones que se ha establecido con la escuela de baloncesto Monster se ha llegado a definir una serie de requisitos funcionales que debería tener el aplicativo los cuales son:

- Roles de usuario: Se definió 3 roles para poder ingresar como usuario administrador, usuario jugador y usuario entrenador.
- Inicio de sesión: Una pantalla para poderse autenticar dependiendo de los roles mencionados anteriormente.
- Administración de jugadores: Esta funcionalidad permite la creación, edición y eliminación de jugadores.
- Administración de entrenadores: Esta característica permite la creación, edición y eliminación de entrenadores.
- Administración de partidos: Esta funcionalidad permite la creación, edición y eliminación de partidos. También se tiene la posibilidad de seleccionar los jugadores que asistirán a ese partido.
- Envío de notificaciones: Esta funcionalidad permite enviar una notificación a todos los jugadores seleccionados al momento de crear un partido.
- Uso de mapas: Esta característica permite visualizar la ubicación de los lugares donde se jugarán los partidos.
- Comunicación vía llamada y correo electrónico: Esta funcionalidad permite contactar a un jugador o entrenador en caso de ser necesario.
- Información de la escuela de baloncesto: Esta característica permite conocer brevemente la participación de la escuela en diferentes campeonatos.

2.4.3. Product backlog

En la lista de producto se listarán todos los requisitos transformados en historias de usuario, los cuales se van a implementar durante el desarrollo de la aplicación móvil. En la siguiente tabla se muestra el producto backlog ordenado en base a cada sprint:

Tabla 5. Product backlog

Código	Nombre	Descripción	Prioridad
MONSAPP-1	Como equipo de desarrollo crear una organización en	Crear una organización en GitHub para el	Alta

	GitHub y un repositorio.	desarrollo de la aplicación móvil. Dentro de esta organización estará el repositorio principal de la aplicación.	
MONSAPP-2	Como equipo de desarrollo instalar, configurar y probar el ambiente de desarrollo.	Preparar todos los dispositivos y herramientas para el desarrollo del proyecto.	Alta
MONSAPP-3	Como equipo de desarrollo migrar el framework de React Native a Ionic.	Cambiar el uso del framework debido a un menor peso de la aplicación, comportamiento nativo y eficiencia en el uso de recursos.	Alta
MONSAPP-4	Como equipo de desarrollo diseñar el bosquejo (Mockup) de la interfaz de usuario.	Crear un bosquejo que permita navegar por todas las pantallas de la aplicación.	Alta
MONSAPP-5	Como administrador	Implementar la navegación stack,	Alta

	disponer de la navegación de toda la aplicación para mi perfil correspondiente.	bottom tap, y sidebar para poder visualizar el flujo entre pantallas.	
MONSAPP-6	Como entrenador disponer de la navegación de toda la aplicación para mi perfil correspondiente.	Implementar la navegación stack, bottom tap, y sidebar para poder visualizar el flujo entre pantallas.	Alta
MONSAPP-7	Como jugador disponer de la navegación de toda la aplicación para mi perfil correspondiente.	Implementar la navegación stack, bottom tap y sidebar para poder visualizar el flujo entre pantallas.	Alta
MONSAPP-8	Como usuario de la aplicación poder iniciar sesión en la aplicación.	Crear una pantalla para iniciar sesión en la aplicación mediante un correo electrónico y una contraseña.	Alta
MONSAPP-9	Como administrador registrar cuentas de usuarios para	Crear una pantalla para poder registrar un correo	Alta

	jugadores.	electrónico, contraseña y datos personales del usuario.	
MONSAPP-10	Como administrador registrar cuentas de usuarios para entrenadores.	Crear una pantalla para poder registrarun correo electrónico, contraseña y datos personales del usuario.	Alta
MONSAPP-11	Como administrador registrar cuentas de usuarios para administradores.	Crear una pantalla para poder registrarun correo electrónico, contraseña y datos personales del usuario.	Alta
MONSAPP-12	Como entrenador poder crear entrenamientos.	Implementar una pantalla para poder agregar un entrenamiento.	Alta
MONSAPP-13	Como usuario entrenador poder ver la lista de	Implementar una pantalla para mostrar a los jugadores de la	Media

	jugadores.	escuela, los cuales se mostrarán como una lista.	
MONSAPP-14	Como administrador enlazar la lista de jugadores con una opción para editar un jugador a la vez.	Implementar una pantalla para editarlos datos personales del jugador la cual se navegue desde la pantalla de jugadores.	Media
MONSAPP-15	Como jugador podereditar mis propios datos.	Implementar una pantalla para editarlos datos personales del jugador la cual se navegue desde la pantalla de jugadores.	Media
MONSAPP-16	Como usuario podercomunicarse con uncontacto.	Implementar una pantalla para poder llamar, enviar un mensaje de texto o un correo electrónico a un jugador o	Media

		entrenador.	
MONSAPP-17	Como administrador agregar un partido de baloncesto.	Implementar una pantalla para poder agregar encuentros deportivos con los siguientes datos: fecha del encuentro, hora del encuentro y equipo rival. Esta pantalla solo estará disponible para el usuario administrador.	Alta
MONSAPP-18	Como usuario visualizar la lista de partidos.	Implementar una pantalla para mostrar la lista de partidos.	Media
MONSAPP-19	Como administrador y entrenador ver la lista de entrenadores.	Implementar una pantalla para mostrar a los entrenadores, los cuales preparan a todas las categorías de la escuela.	Media

MONSAPP-20	Como administrador modificar datos de unentrenador.	Implementar una pantalla para modificar la información de los entrenadores, desde el rol administrador.	Media
MONSAPP-21	Como entrenador modificar mis propios datos.	Implementar una pantalla para modificar la información de los entrenadores, desde el rol entrenador.	Baja
MONSAPP-22	Como usuario tener en la pantalla lateral un botón para acceder a las redes sociales.	Implementar los iconos de las redes sociales Facebook e Instagram.	Baja
MONSAPP-23	Como usuario jugador visualizar la ubicación de los partidos de la escuela.	Mostrar la ubicación mediante el uso de Google Maps de las canchas donde se desarrollará el	Baja

		encuentro.	
MONSAPP-24	Como administrador poder seleccionar a jugadores específicos para un partido o entrenamiento.	Mostrar en el rol jugador los partidos o entrenamientos a los cuales fue seleccionado.	Baja
MONSAPP-25	Como usuario administrador agregar un equipo rival.	Implementar una pantalla para poder agregar equipos rivales deportivos con los siguientes datos: nombre, provincia, cantón y estado.	Media
MONSAPP-26	Como usuario del sistema poder recuperar la contraseña.	Implementar la funcionalidad de recuperar la clave en la pantalla de inicio de sesión.	Alta
MONSAPP-27	Como usuario tener en la barra lateral acceso a la misión, visión y campeonatos de la	Implementar tres botones en la barra lateral que me permitan ver la misión, visión y	Baja

	escuela.	campeonatos de la escuela.	
MONSAPP-28	Como usuario administrador agregar un torneo.	Implementar una pantalla para poder agregar torneos con los siguientes datos: nombre, fecha inicio, fecha fin y resultado.	Media
MONSAPP-29	Como usuario administrador tener la opción de reportes.	Implementar la opción de reportes que me permita ver un reporte de los torneos y un reporte detallado de los jugadores.	Media

Elaborado por: Autores

Fuente: Análisis

2.4.4. Sprint 0

2.4.4.1. Objetivos del sprint

Los Objetivos de este sprint son:

- Crear un repositorio para el manejo de versiones.
- Configurar y probar el ambiente de desarrollo.
- Seleccionar el framework para el desarrollo del aplicativo.

2.4.4.2. Planificación del sprint

A continuación, se muestra en la figura 4 la planificación del sprint 0, haciendo uso de la herramienta Trello:

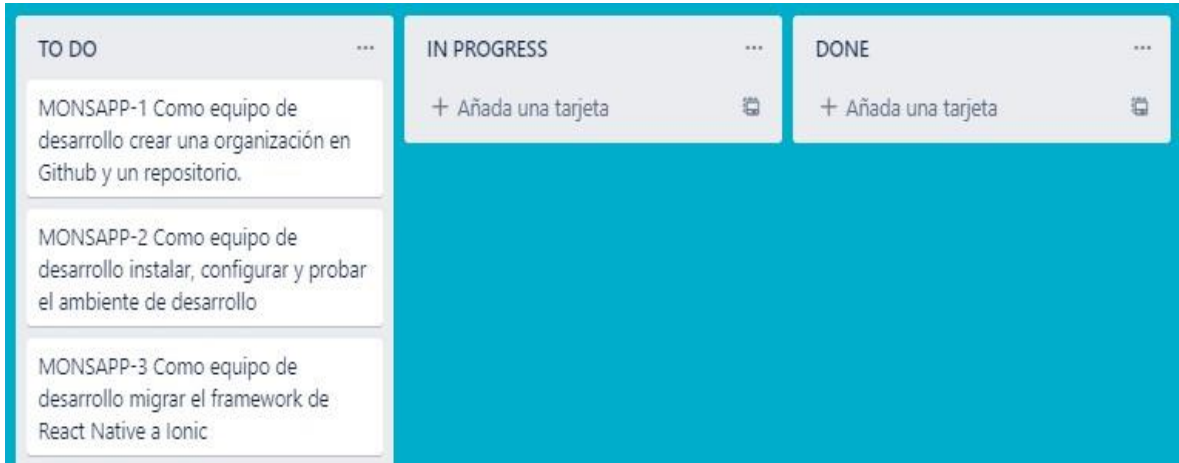


Figura 4. Planificación del sprint 0.

Las siguientes son las historias de usuario que se trabajaron en este sprint:

Tabla 6. Historia de usuario MONSAPP-1

Código:	MONSAPP-1	Esfuerzo:	1
Nombre:	Como equipo de desarrollo crear una organización en GitHub y un repositorio.		
Descripción:	Crear una organización en GitHub para el desarrollo de la aplicación móvil. Dentro de esta organización estará el repositorio principal de la aplicación.		
Prioridad:	Alta		
Criterios de aceptación:			
La organización debe ser de carácter privado y solo debe contener el repositorio del aplicativo móvil.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 7. Historia de usuario MONSAPP-2

Código:	MONSAPP-2	Esfuerzo:	3
Nombre:	Como equipo de desarrollo instalar, configurar y probar el ambiente de desarrollo.		
Descripción:	Preparar todos los dispositivos y herramientas para el desarrollo del proyecto		
Prioridad:	Alta		
Criterios de aceptación:			
El ambiente debe tener todas las herramientas y recursos necesarios para poder desarrollar el proyecto.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 8. Historia de usuario MONSAPP-3

Código:	MONSAPP-3	Esfuerzo:	3
Nombre:	Como equipo de desarrollo migrar el framework de React Native a Ionic		
Descripción:	Cambiar el uso del framework debido a un menor peso de la aplicación, comportamiento nativo y eficiencia en el uso de recursos.		
Prioridad:	Alta		
Criterios de aceptación:			
El marco de trabajo debe ser para desarrollo móvil y debe hacer un uso eficiente de los recursos.			

Elaborado por: Autores

Fuente: Historias de usuario

2.4.4.3. Implementación

Para el presente proyecto se hizo uso de Visual Studio Code como IDE (Entorno de desarrollo integrado) para el desarrollo de la aplicación móvil. Posee muchas extensiones para mejorar en la escritura de código, hacer seguimiento de errores,

dar formato al código, etc. Al ser propiedad de Microsoft, continuamente se está actualizando y trayendo nuevas características y/o mejoras.

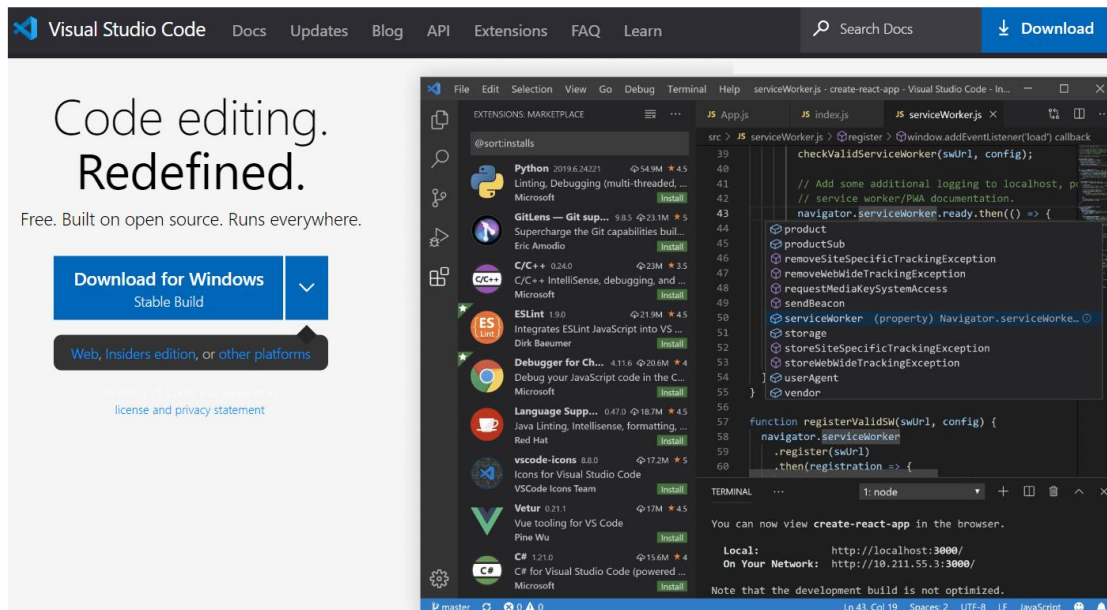


Figura 5. Sitio de descarga de Visual Studio Code.

Uso de GitHub para el almacenamiento del código del aplicativo. Dentro de éste creamos un repositorio para el control y manejo de versiones del proyecto; esto como un respaldo en caso de que pase algo con el ordenador.

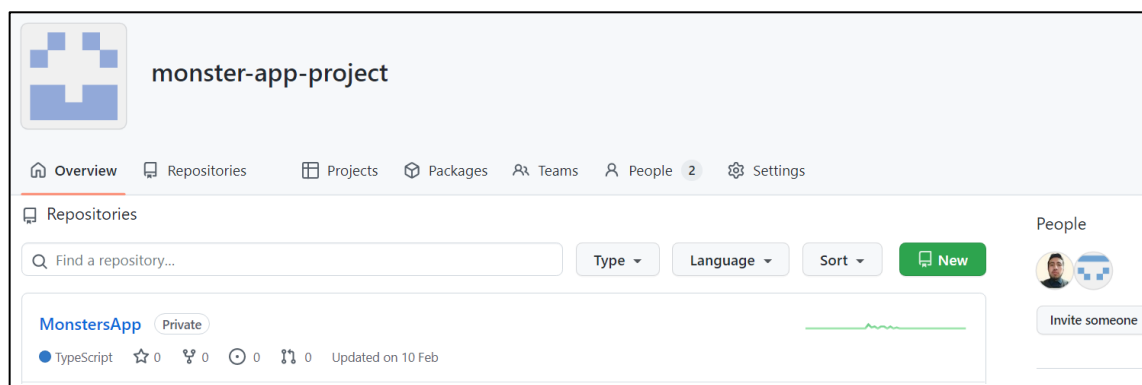


Figura 6. Vista del repositorio en GitHub.

También se hace uso de NodeJS, el cual se descarga por medio de la página oficial que es nodejs.org/es.

Cuando se descargue procedemos a instalar y cuando finalice procedemos a abrir en el editor de código Visual Studio Code la misma que nos permitirá abrir la terminal para ver los procesos de la aplicación móvil.

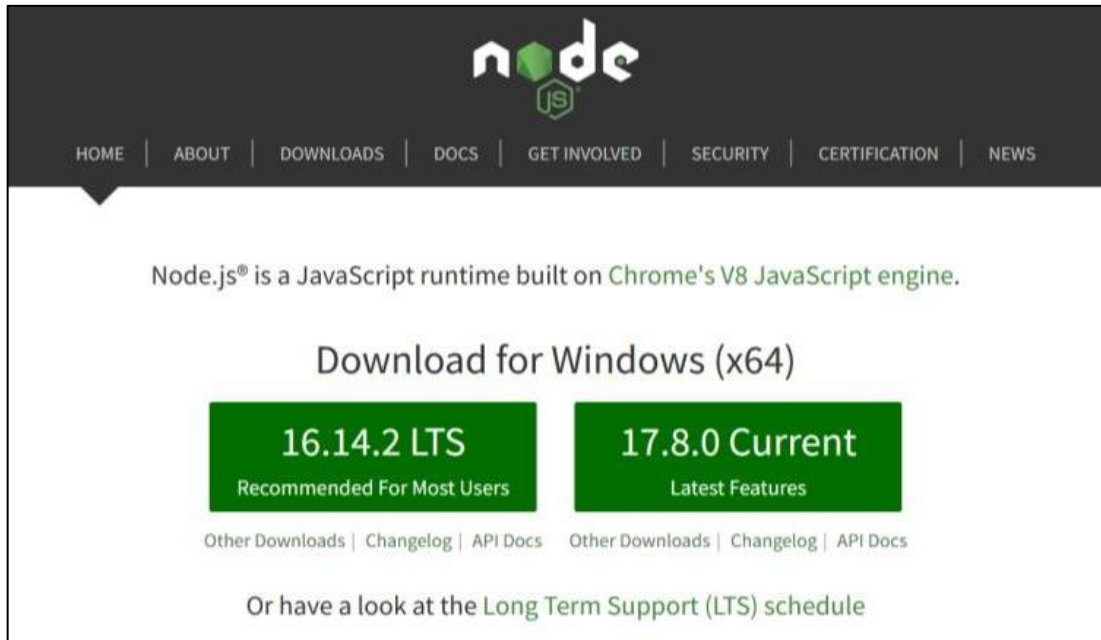


Figura 7. Página de descarga de NodeJS.

Finalmente, para poder usar los servicios de Firebase, los cuales son de autenticación y de almacenamiento de datos, se debe crear un proyecto dentro de este servicio de Google, al cual accederemos mediante un API-Key.

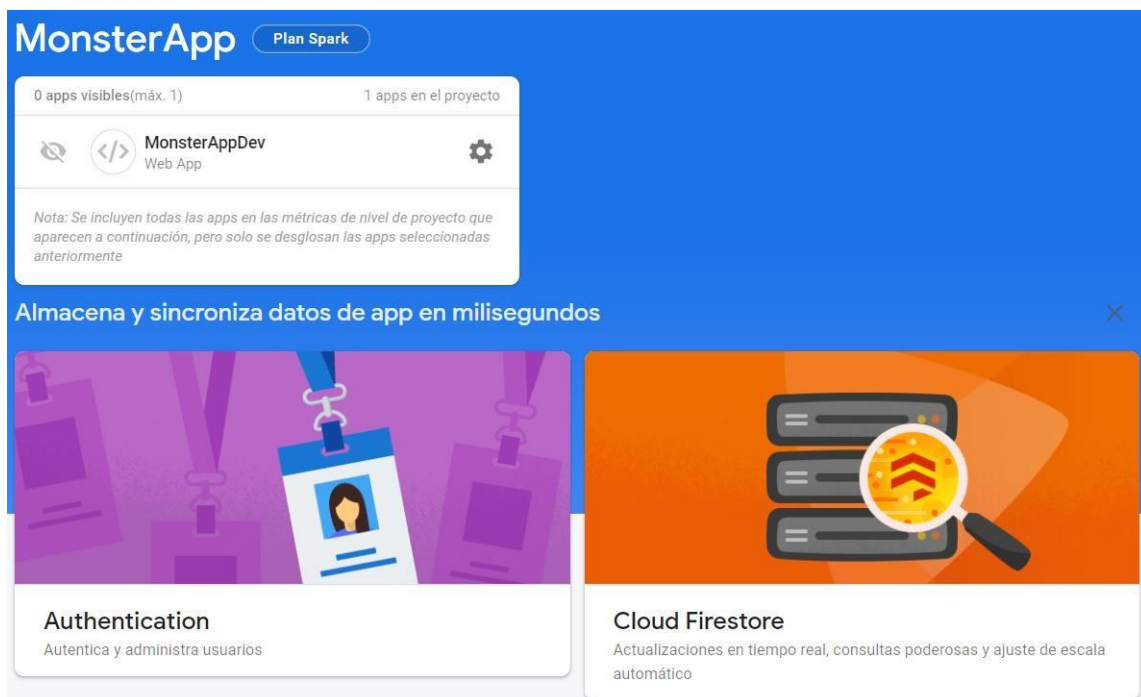


Figura 8. Vista del proyecto en Firebase.

2.4.4.4. Sprint review

Al final del sprint 0 se logró cumplir con los objetivos planteados, se definió una organización con su respectivo repositorio para el alojamiento de la aplicación y control de versiones, se instaló y configuró todas las herramientas necesarias para el desarrollo de la aplicación y finalmente se eligió el framework que se ajustó a las necesidades de la aplicación.

En la figura siguiente se muestra la captura de la revisión del sprint 0 en la herramienta Trello.

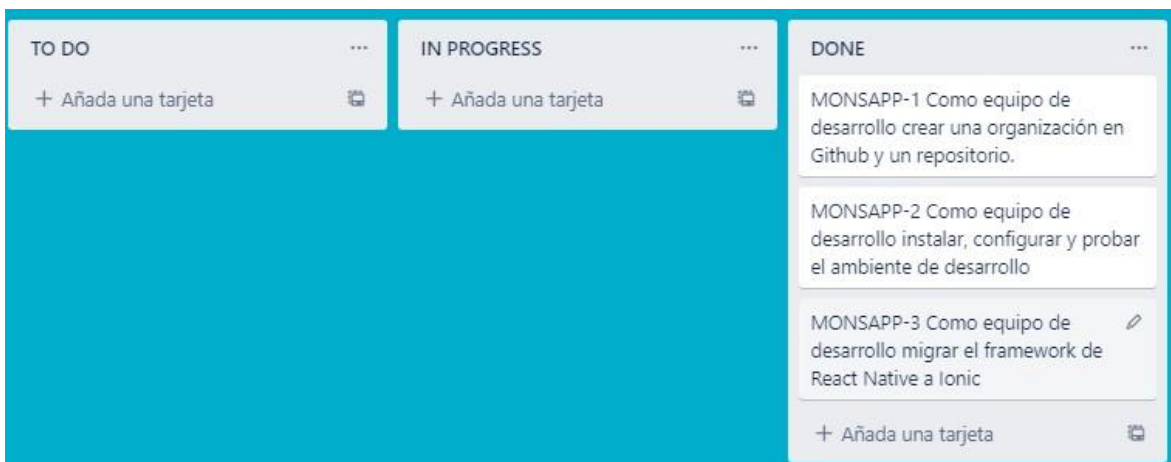


Figura 9. Finalización del sprint 0.

En la siguiente tabla se puede observar las historias de usuario que se implementó en el sprint 0.

Tabla 9. Revisión sprint 0.

Código	Nombre	Observaciones	Esfuerzo inicial	Esfuerzo final	Cumplido
MONSAPP-1	Como equipo de desarrollo crear una organización en GitHub y un repositorio.	Ninguna	1	1	Si
MONSAPP-2	Como equipo de desarrollo instalar,	Ninguna	3	3	Si

	configurar y probar el ambiente de desarrollo.				
MONSAPP-3	Como equipo de desarrollo migrar el framework de React Native a Ionic.	Ninguna	3	3	Si

2.4.5. Sprint 1

2.4.5.1. Objetivos del sprint

Los Objetivos de este sprint son:

- Diseñar el boceto de la aplicación (Mockups) de la interfaz de usuario.
- Implementar la navegación en toda la aplicación en sus diferentes roles de usuario.
- Crear la pantalla de inicio de sesión de la aplicación.
- Registrar cuentas de usuarios para administradores, entrenadores y jugadores.

2.4.5.2. Planificación del sprint

A continuación, se muestra en la figura 10 la planificación del sprint 1, haciendo uso de la herramienta Trello:

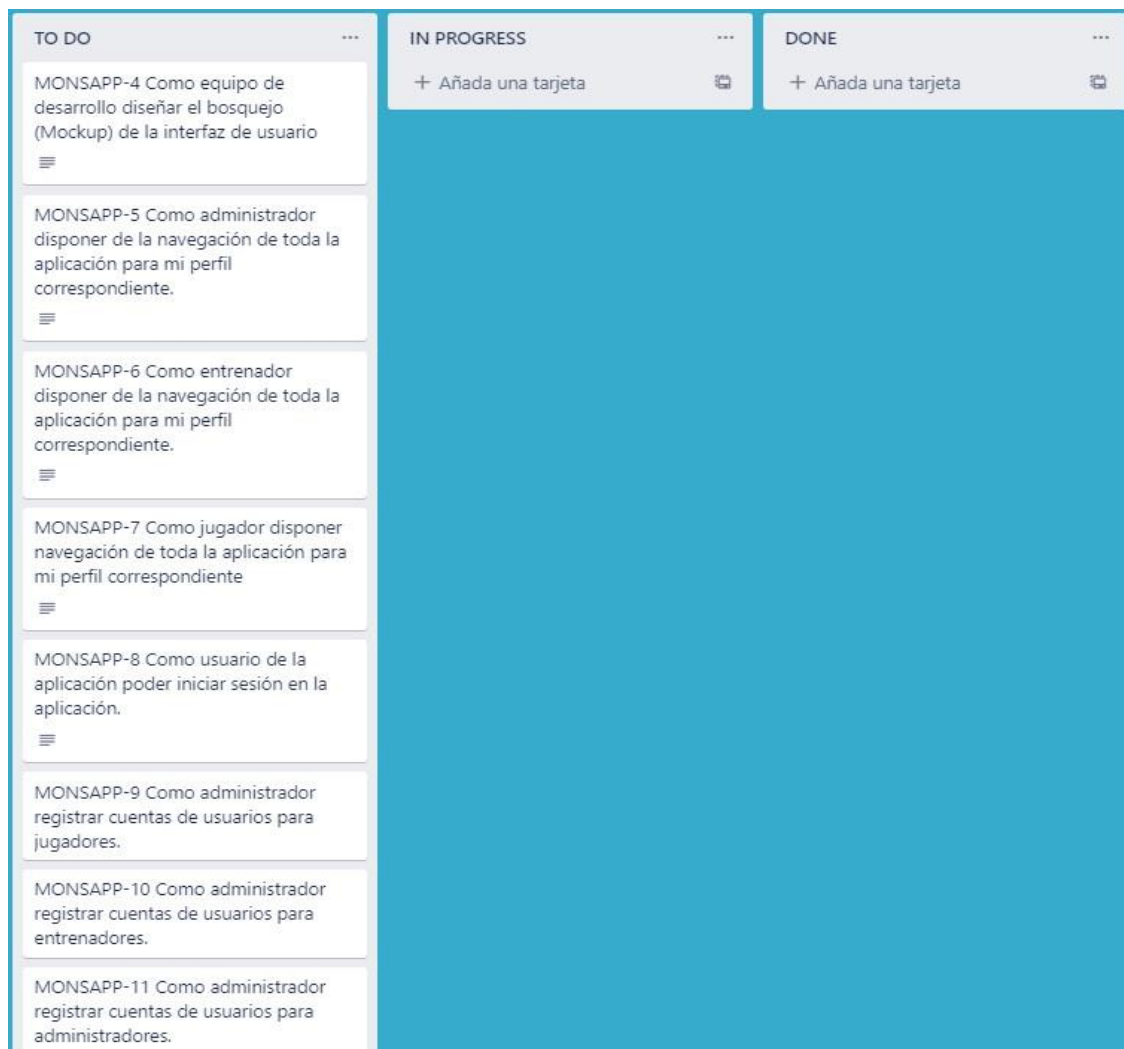


Figura 10. Planificación del sprint 1.

Las siguientes son las historias de usuario que se trabajaron en este sprint:

Tabla 10. Historia de usuario MONSAPP-4.

Código:	MONSAPP-4	Esfuerzo:	1
Nombre:	Como equipo de desarrollo diseñar el bosquejo (Mockup) de la interfaz de usuario		
Descripción:	Crear un bosquejo que permita navegar por todas las pantallas de la aplicación.		
Prioridad:	Alta		
Criterios de aceptación:	El bosquejo tiene ser comprensible, que permita tener una vista general de las pantallas que se va a realizar.		

Elaborado por: Autores
Fuente: Historias de usuario

Tabla 11. Historia de usuario MONSAPP-5.

Código:	MONSAPP-5	Esfuerzo:	2
Nombre:	Como administrador disponer de la navegación de toda la aplicación para mi perfil correspondiente.		
Descripción:	Implementar la navegación stack, bottom tap, y sidebar para poder visualizar el flujo entre pantallas.		
Prioridad:	Alta		
Criterios de aceptación:			
Validar que todos los tipos de navegación funcionen correctamente para el rol administrador.			

Elaborado por: Autores
Fuente: Historias de usuario

Tabla 12. Historia de usuario MONSAPP-6.

Código:	MONSAPP-6	Esfuerzo:	2
Nombre:	Como entrenador disponer de la navegación de toda la aplicación para mi perfil correspondiente.		
Descripción:	Implementar la navegación stack, bottom tap, y sidebar para poder visualizar el flujo entre pantallas.		
Prioridad:	Alta		
Criterios de aceptación:			
Validar que todos los tipos de navegación funcionen correctamente para el rol entrenador.			

Elaborado por: Autores
Fuente: Historias de usuario

Tabla 13. Historia de usuario MONSAPP-7.

Código:	MONSAPP-7	Esfuerzo:	2
Nombre:	Como jugador disponer de la navegación de toda la aplicación para mi perfil correspondiente.		
Descripción:	Implementar la navegación stack, bottom tap, y sidebar para poder visualizar el flujo entre pantallas.		
Prioridad:	Alta		
Criterios de aceptación:			
Validar que todos los tipos de navegación funcionen correctamente para el rol jugador.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 14. Historia de usuario MONSAPP-8.

Código:	MONSAPP-8	Esfuerzo:	5
Nombre:	Como usuario de la aplicación poder iniciar sesión en la aplicación.		
Descripción:	Crear una pantalla para poder iniciar sesión en la aplicación mediante un correo electrónico y una contraseña.		
Prioridad:	Alta		
Criterios de aceptación:			
El correo electrónico debe existir en la base de datos y la contraseña debe corresponder al correo ingresado.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 15. Historia de usuario MONSAPP-9.

Código:	MONSAPP-9	Esfuerzo:	5
Nombre:	Como administrador registrar cuentas de usuarios para jugadores.		

Descripción:	Crear una pantalla para poder registrar un correo electrónico, contraseña y datos personales del jugador.
Prioridad:	Alta
Criterios de aceptación:	
En la pantalla de registro de un jugador se debe llenar los siguientes campos:	
- Nombre	- Apellido
- Celular	- Edad
- Apodo	- Peso
- Estatura	- Número de camiseta
- Categoría	- Sexo
- Estado	- Correo
- Contraseña	

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 16. Historia de usuario MONSAPP-10.

Código:	MONSAPP-10	Esfuerzo:	5
Nombre:	Como administrador registrar cuentas de usuarios para entrenadores.		
Descripción:	Crear una pantalla para poder registrar un correo electrónico, contraseña y datos personales del entrenador.		
Prioridad:	Alta		
Criterios de aceptación:			
En la pantalla de registro de un administrador se debe llenar los siguientes campos:			
- Nombre	- Apellido		
- Nacionalidad	- Edad		
- Celular	- Sexo		
- Estado	- Correo electrónico		
- Contraseña			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 17. Historia de usuario MONSAPP-11.

Código:	MONSAPP-11	Esfuerzo:	5
Nombre:	Como administrador registrar cuentas de usuarios para administradores.		
Descripción:	Crear una pantalla para poder registrar un correo electrónico, contraseña y datos personales del usuario.		
Prioridad:	Alta		
Criterios de aceptación:			
<p>En la pantalla de registro cuenta de un administrador se debe llenar los siguientes campos:</p> <ul style="list-style-type: none"> - Nombre - Apellido - Correo electrónico 			

Elaborado por: Autores

Fuente: Historias de usuario

2.4.5.3. Implementación

Se realizó el bosquejo de las pantallas de la aplicación móvil la cual será intuitiva y fácil de utilizar por los distintos tipos de usuario. Ésta cuenta con unas pantallas de navegación en la cual podremos acceder a todas las funciones de la aplicación de una forma sencilla y rápida.

A continuación, en las siguientes figuras se muestra los mockups de la aplicación:



Figura 11. Bosquejos de las pantallas de la aplicación móvil.

A continuación, se puede observar el HTML para registrar nuevos usuarios en la aplicación móvil el mismo que cuenta con los campos de nombre, apellido, email, contraseña, apellido, edad, apodo, peso, teléfono, estatura, categoría número estado, nacionalidad los cuales son obligatorios al momento de registrar un nuevo usuario.

```
1 <form #userForm="ngForm" class="ion-padding ion-margin" (ngSubmit)="onSubmit(userForm.value)" novalidate>
2
3 <ion-toolbar *ngIf="!create" color="primary">
4   <ion-button (click)="onCancel()" fill="clear" color="success">
5     REGRESAR
6   </ion-button>
7 </ion-toolbar>
8 <ion-item>
9   <ion-label position="floating">Nombres</ion-label>
10  <ion-input type="text" [(ngModel)]="user.name" name="name" required</ion-input>
11 </ion-item>
12 <ion-item>
13   <ion-label position="floating">Apellido</ion-label>
14   <ion-input type="text" [(ngModel)]="user.apellido" name="apellido" required</ion-input>
15 </ion-item>
16 <ion-item>
17   <ion-label position="floating">Nacionalidad</ion-label>
18   <ion-input type="text" [(ngModel)]="user.nacionalidad" name="nacionalidad" required</ion-input>
19 </ion-item>
20 <ion-item>
21   <ion-label position="floating">Edad</ion-label>
22   <ion-input type="text" [(ngModel)]="user.edad" name="edad" required</ion-input>
23 </ion-item>
24 <ion-item>
25   <ion-label position="floating">Telefono</ion-label>
26   <ion-input type="tel" [(ngModel)]="user.telefono" name="telefono" required</ion-input>
27 </ion-item>
28 <ion-item>
29   <ion-label position="floating">Apodo</ion-label>
30   <ion-input type="text" [(ngModel)]="user.apodo" name="apodo" required</ion-input>
```

Figura 12. HTML para registrar un nuevo usuario.

Servicio que llama el API REST de la aplicación móvil para crear un nuevo usuario donde mandamos a llenar en la base de datos el id que es automático, nombre, apellido, email, contraseña, apellido, edad, apodo, peso, teléfono, estatura, categoría número estado, nacionalidad.

```

6 import { HttpClient } from '@angular/common/http';
7 import { Entre } from '../models/entre.model';
8
9 @Injectable({
10   providedIn: 'root',
11 })
12 export class LeagueService {
13   private currentMatch: Match;
14   private currentEntre: Entre;
15   constructor(
16     private firestore: AngularFirestore,
17     private http: HttpClient
18   ) {}
19
20   addPlayer(user: User, uid: string) {
21     this.firestore.collection('users').doc(uid).update({
22       uid,
23       name: user.name,
24       email: user.email,
25       password: user.password,
26       apellido: user.apellido,
27       edad: user.edad,
28       apodo: user.apodo,
29       peso: user.peso,
30       telefono: user.telefono,
31       estatura: user.estatura,
32       categoria: user.categoria,
33       numero: user.numero,
34       estado: user.estado,
35       nacionalidad: user.nacionalidad,

```

Figura 13. Servicio para crear un nuevo usuario.

A continuación, se puede apreciar los métodos que nos permiten crear un nuevo usuario o editar los datos de un usuario:

```

    });
  }

  createAccount(user: User) {
    return this.afAuth
      .createUserWithEmailAndPassword(user.email, user.password)
      .then((data) => {
        this.addRoles(user.role, data.user.uid);

        this.leagueServ.addPlayer(user, data.user.uid);
      });
  }

  updateAccount(newUser: User, oldUser: User) {
    console.log(newUser);

    return this.afAuth
      .signInWithEmailAndPassword(oldUser.email, oldUser.password)
      .then((data) => {
        data.user.updateEmail(newUser.email);
        data.user.updatePassword(newUser.password);

        this.leagueServ.updatePlayer(newUser);
      });
  }
}

```

Figura 14. Métodos para crear o actualizar un usuario.

Método para validar cuenta la misma que será validada por el correo electrónico donde le llegará un mensaje de confirmación al correo registrado al momento de ingresar los datos para registrar un nuevo usuario.

```

33
34     constructor(
35         private authService: AuthService,
36         private toaster: ToastController,
37         private league: LeagueService,
38         private navCtrl: NavController
39     ) {}
40
41     ngOnInit() {}
42
43     onSubmit() {
44
45         this.authService
46             .createAccount(this.user)
47             .then(() => {
48                 this.navCtrl.navigateForward('login');
49                 this.presentToastWithOptions('Cuenta creada con éxito!', false);
50             })
51             .catch((err) => {
52                 this.presentToastWithOptions(err, true);
53             });
54     }
55
56     private async presentToastWithOptions(msg: string, error: boolean) {
57         let color = error ? 'danger' : 'success';
58
59         const toast = await this.toaster.create({
60             message: msg,
61             duration: 1000,
62             position: 'bottom',

```

Figura 15. Método para validar una cuenta.

HTML para iniciar sesión el mismo que tiene la propiedad “required” en las cajas de texto donde no podrá ingresar a la aplicación mientras los campos requeridos no se encuentren llenos. De igual forma tiene la propiedad “placeholder” la misma que mostrara en la caja de texto que tipo de información tiene que ingresar.

```

login.page.html login.page.html X vision.page.html app.component.html tab3.page.html T9 contact.service.ts T8 ta
app > login > login.page.html > ion-content > div.logindiv > ion-card > form > ion-grid > ion-row.ion-justify-content > ion-col.ion-text-
<form #loginForm="ngForm">
  <ion-grid>
    <ion-row class="ion-justify-content">
      <ion-col size="12" class="ion-text-center">
        <ion-input [(ngModel)]="username" required name="username" type="text" placeholder="E-mail">
      </ion-col>
      <ion-col size="12" class="ion-text-center">
        <ion-input [(ngModel)]="password" required name="password" type="password"
          placeholder="Password"></ion-input>
      </ion-col>
      <ion-col class="ion-text-center">
        <ion-button type="submit" class="ion-text-center" [disabled]="!loginForm.valid"
          (click)="checkLogin()">
          LOGIN
        </ion-button>
      </ion-col>
    </ion-row>
    <ion-row class="ion-justify-content">
      <ion-col class="ion-text-center">
        <a (click)="gotoUserCreate()">
          CREAR CUENTA
        </a>
      </ion-col>
    </ion-row>
  </ion-grid>
</form>
</ion-card>

```

Figura 16. HTML para iniciar sesión.

Servicio para iniciar sesión, el cual validará el usuario y la contraseña en la base de datos. De la misma forma tiene la propiedad de StoreOnSession, la cual permite

almacenar en memoria temporal del teléfono la sesión del usuario que inicia para mostrar sus datos.

```
login(username: string, password: string) {
  return new Promise((resolve, reject) => {
    this.afAuth
      .signInWithEmailAndPassword(username, password)
      .then((data) => {
        this.firestore
          .collection('users')
          .doc(data.user.uid)
          .get()
          .subscribe((doc) => {
            resolve(true);
            this.isLogged = true;
            let user: any = doc.data();
            this.storeOnSession(user);
          });
      })
      .catch((error) => {
        reject(error);
        this.isLogged = false;
        return error;
      });
  });
};
```

Figura 17. Servicio para iniciar sesión.

HTML para editar datos, el usuario podrá editar sus datos personales. El mismo HTML es utilizado para los usuarios entrenador y jugador. Este HTML cuenta con la propiedad *ngIf la cual verifica que rol es para mostrar que datos puede editar.

```
<ion-item>
  <ion-label position="floating">Correo electrónico del jugador</ion-label>
  <ion-input type="email" [(ngModel)]="user.email" name="email" required</ion-input>
</ion-item>
<ion-item>
  <ion-label position="floating">Contraseña</ion-label>
  <ion-input type="password" [(ngModel)]="user.password" name="password" required</ion-input>
</ion-item>
<ion-item *ngIf="!user.role">
  <ion-label>Role</ion-label>
  <ion-select [(ngModel)]="user.role" name="role" required>
    <ion-select-option value="user">Jugador</ion-select-option>
    <ion-select-option value="entre">Entrenador</ion-select-option>
  </ion-select>
</ion-item>
<ion-item *ngIf="user.role=='user'">
  <ion-label>Role</ion-label>
  <ion-select [(ngModel)]="user.role" name="role" required>
    <ion-select-option value="user">Jugador</ion-select-option>
  </ion-select>
</ion-item>
```

Figura 18. HTML para editar los datos del usuario.

El HTML corresponde a una pantalla, para que funcione esta pantalla, necesita de un servicio, el cual va a validar y guardar los datos del usuario.

```

createAccount(user: User) {
  return this.afAuth
    .createUserWithEmailAndPassword(user.email, user.password)
    .then((data) => {
      this.addRoles(user.role, data.user.uid);

      this.leagueServ.addPlayer(user, data.user.uid);
    });
}

updateAccount(newUser: User, oldUser: User) {
  console.log(newUser);

  return this.afAuth
    .signInWithEmailAndPassword(oldUser.email, oldUser.password)
    .then((data) => {
      data.user.updateEmail(newUser.email);
      data.user.updatePassword(newUser.password);

      this.leagueServ.updatePlayer(newUser);
    });
}

private addRoles(role: string, uid: string) {
  this.firestore.collection('users').doc(uid).set({ role: role });
}
}

```

Figura 19. Servicio para obtener los datos del usuario desde la base.

En la figura 20 podemos visualizar que cuando se inicia en la página principal de la aplicación se muestra el nombre del usuario ya sea entrenador o jugador (ver cuadro No.1).

De la misma manera, cuando el usuario es entrenador poder ver dos opciones que son la de jugadores y entrenamientos (ver cuadro No. 2) lo mismo que cuando el usuario entrenador dar clic en el botón jugadores, puede ver la lista de jugadores que tiene actualmente y si le da clic sobre entrenamientos puede ingresar un nuevo entrenamiento o ver los entrenamientos asignados que tiene.

La segunda interfaz gráfica pertenece al usuario jugador en el cual podrá ver la lista de los próximos partidos y entrenamientos pendientes (ver cuadro No. 3). La aplicación también cuenta con un menú de navegación mediante tabs, las cuales se muestran dependiendo del usuario que ingresa a la app. (cuadro No. 4).

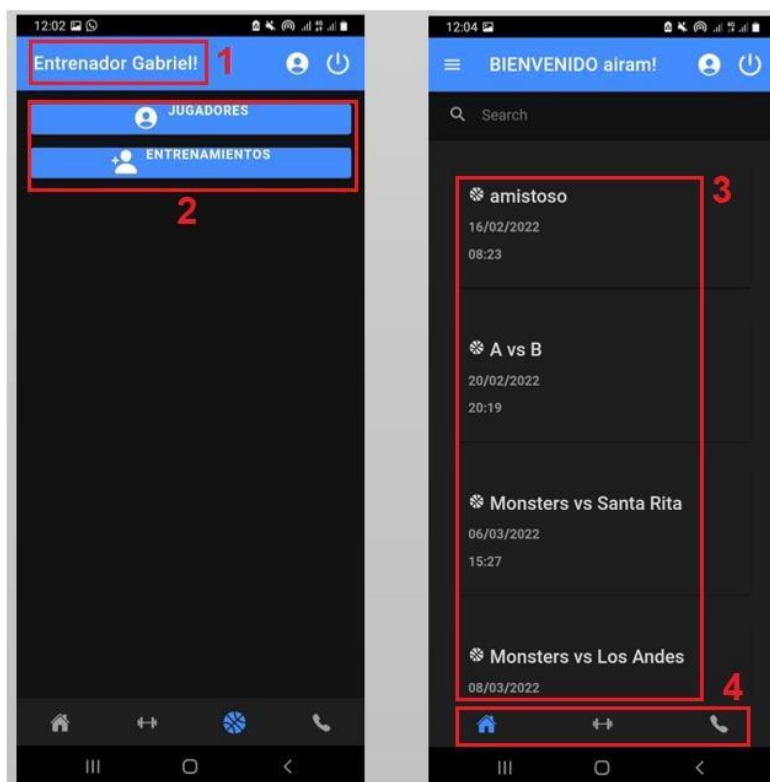


Figura 20. Interfaz gráfica de la aplicación.

En la figura 21 se puede observar que la primera pantalla es para iniciar sesión en la app móvil, la cual tiene campos requeridos, que mientras el usuario y contraseña no estén llenos no se activara el botón de Login (ver cuadro No. 1), cuando finalmente el botón se activa el usuario le da clic el correo y la contraseña es validado si es correcto ingresa a la página principal de la aplicación caso contrario le muestra un mensaje de correo electrónico o contraseña incorrectos.

Cuando le da clic en el texto de crear nueva cuenta le redirige a una nueva pantalla en la cual le solicita el correo electrónico, contraseña, el tipo de rol y los nombres (ver cuadro No. 2) de la misma forma tiene el validador de campos, mientras no esté lleno con información no le permite dar clic sobre el botón crear.

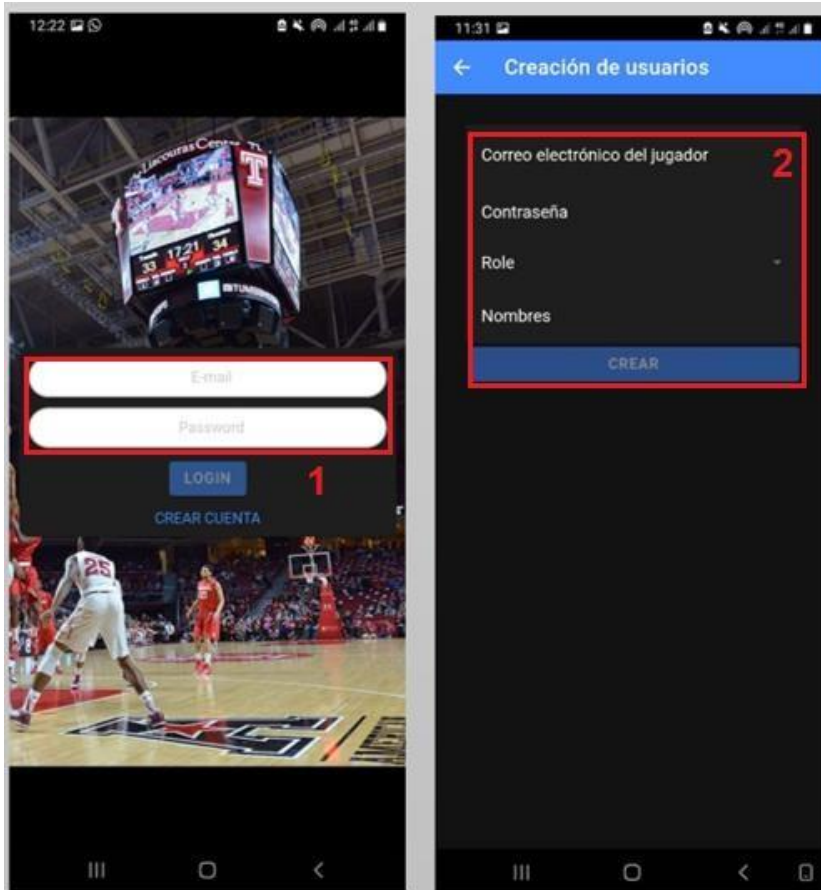


Figura 21. Interfaz gráfica de inicio de sesión y creación de usuarios.

En la figura 22 se puede visualizar que se cuenta con un menú lateral (ver cuadro No. 1) donde se puede observar la misión, visión, campeonatos y redes sociales. Al momento de dar clic sobre la opción campeonatos se dirigirá a otra pantalla donde podremos visualizar los campeonatos obtenidos por la escuela de baloncesto Monsters (ver cuadro No. 2).

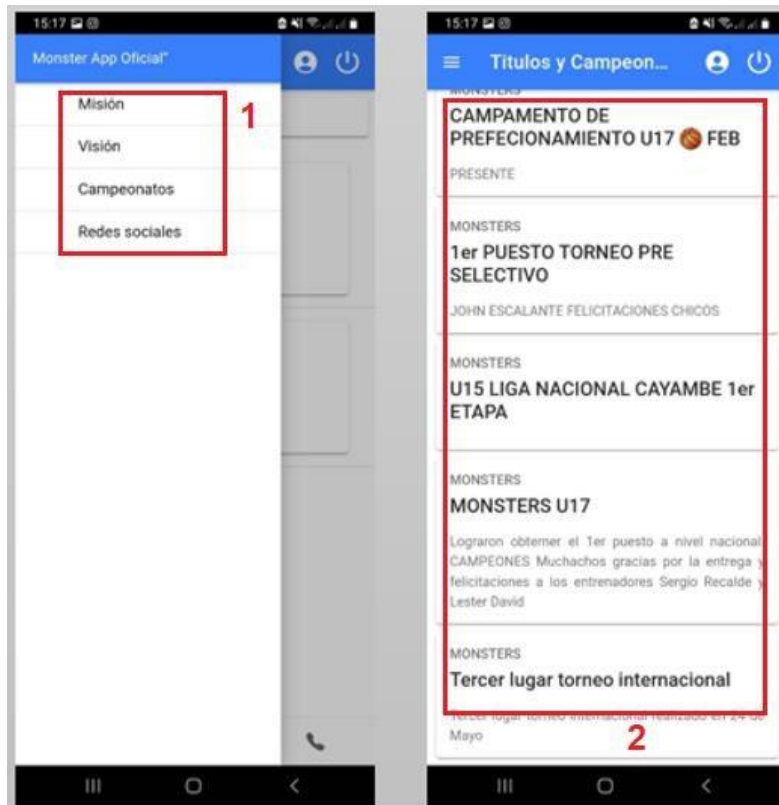


Figura 22. Barra lateral de la aplicación móvil.

2.4.5.4. Sprint review

Al final del sprint 1 se logró cumplir con los objetivos planteados, se logró diseñar el bosquejo de la aplicación con todas las interfaces de usuario, se implementó la navegación de la aplicación para los diferentes roles, se implementó la pantalla de inicio de sesión de la aplicación y finalmente se implementó la pantalla de registrar cuentas para los diferentes tipos de usuarios.

En la figura siguiente se muestra la captura de la revisión del sprint 1 en la herramienta Trello.

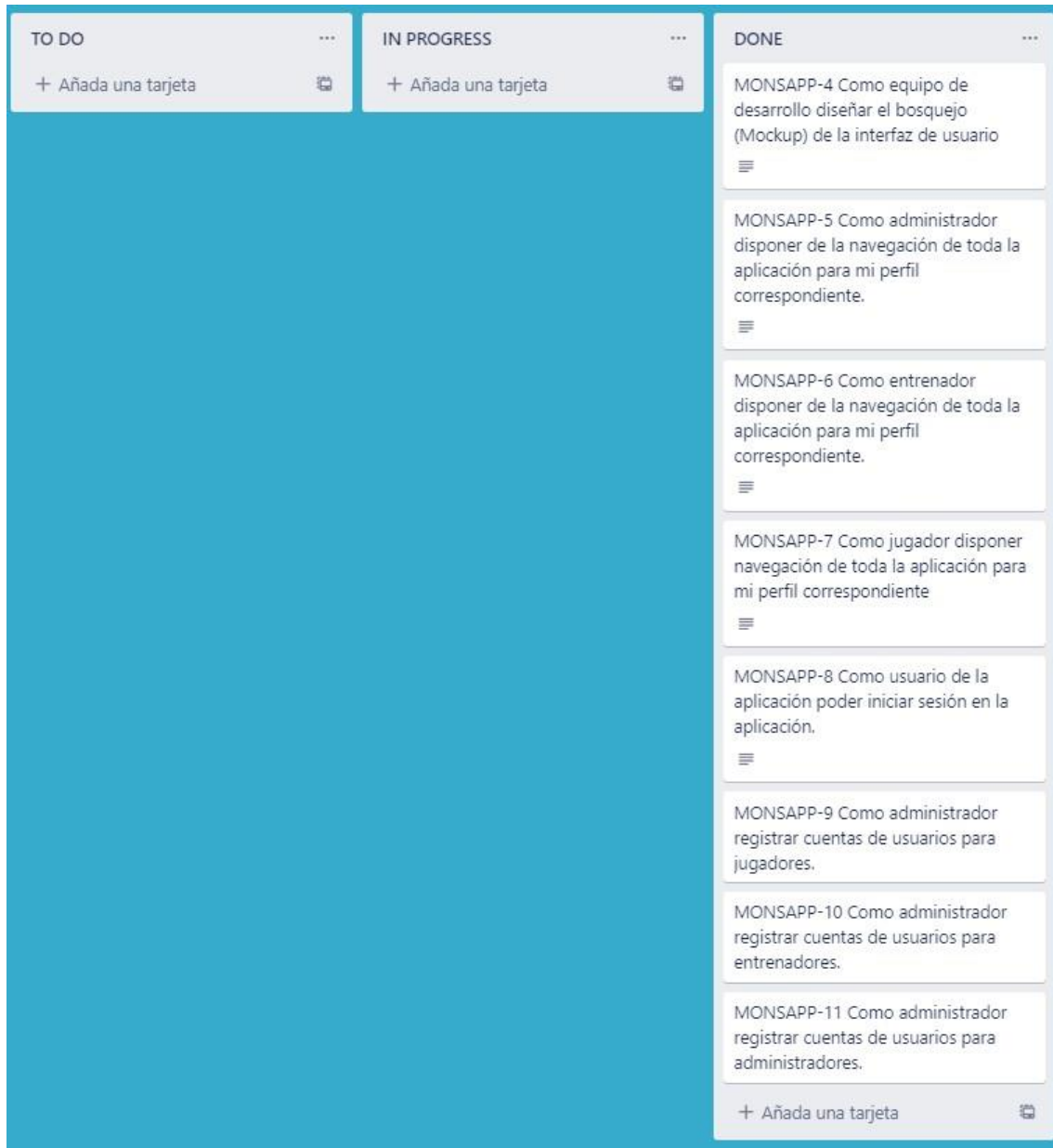


Figura 23. Finalización del sprint 1.

En la tabla 18 se puede observar las historias de usuario que se implementó en este sprint.

Tabla 18. Revisión sprint 1.

Código	Nombre	Observaciones	Esfuerzo inicial	Esfuerzo final	Cumplido
MONSAPP-4	Como equipo de desarrollo diseñar el bosquejo (Mockup) de la interfaz de usuario	Ninguna	1	1	Si

	bosquejo (Mockup) de la interfaz de usuario.				
MONSAPP-5	Como administrador disponer de la navegación de toda la aplicación para mi perfil correspondiente.	Ninguna	2	2	Si
MONSAPP-6	Como entrenador disponer de la navegación de toda la aplicación para mi perfil correspondiente.	Ninguna	2	2	Si
MONSAPP-7	Como jugador disponer de la navegación de toda la aplicación para mi perfil correspondiente.	Ninguna	2	2	Si
MONSAPP-8	Como usuario de la aplicación poder iniciar sesión en la aplicación.	Ninguna	5	5	Si

MONSAPP-9	Como administrador registrar cuentas de usuario para jugadores.	Ninguna	5	5	Si
MONSAPP-10	Como administrador registrar cuentas de usuario para entrenadores.	Ninguna	5	5	Si
MONSAPP-11	Como administrador registrar cuentas de usuario para administradores.	Ninguna	5	5	Si

2.4.6. Sprint 2

2.4.6.1. Objetivo del sprint

Los objetivos de este sprint son:

- Implementar una pantalla para agregar un entrenamiento.
- Implementar una pantalla para mostrar a los jugadores de la escuela.
- Diseñar una pantalla para editar los datos personales del jugador.
- Diseñar una pantalla para agregar encuentros deportivos.

2.4.6.2. Planificación del sprint

A continuación, se muestra en la figura 24 la planificación del sprint 2, haciendo uso de la herramienta Trello:

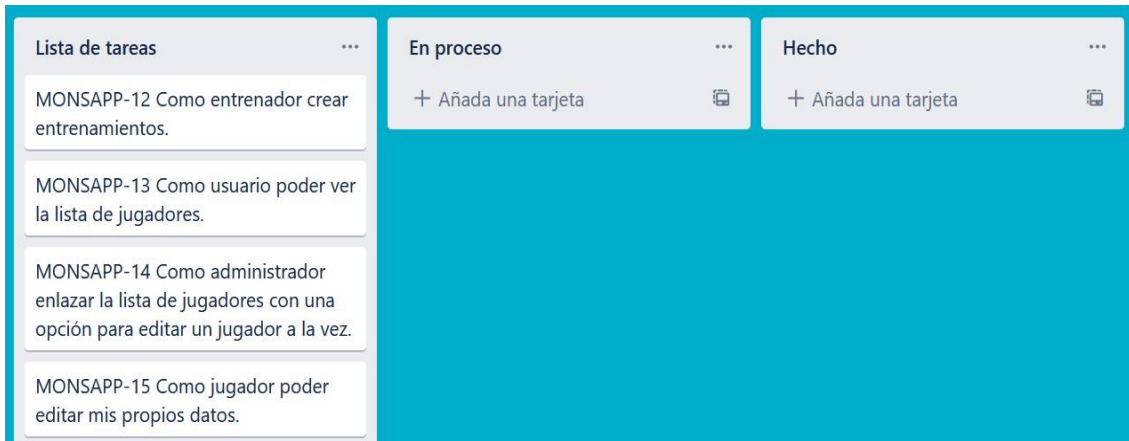


Figura 24. Planificación del sprint 2.

Las siguientes son las historias de usuario que se trabajaron en este sprint:

Tabla 19. Historia de usuario MONSAPP-12.

Código:	MONSAPP-12	Esfuerzo:	2
Nombre:	Como entrenador crear entrenamientos		
Descripción:	Implementar una pantalla para agregar un entrenamiento.		
Prioridad:	Alta		
Criterios de aceptación:			
La pantalla tiene que ser amigable con el entrenador para que no sea complejo crear los nuevos entrenamientos.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 20. Historia de usuario MONSAPP-13.

Código:	MONSAPP-13	Esfuerzo:	3
Nombre:	Como usuario ver la lista de jugadores		
Descripción:	Implementar una pantalla para mostrar a los jugadores de la escuela, los cuales se mostrarán como una lista.		
Prioridad:	Media		
Criterios de aceptación:			
La lista que se mostrará en pantalla deberá tener el nombre y apellido de los jugadores de la escuela registrados.			

Elaborado por: Autores
Fuente: Historias de usuario

Tabla 21. Historia de usuario MONSAPP-14.

Código:	MONSAPP-14	Esfuerzo:	2
Nombre:	Como administrador enlazar la lista de jugadores con una opción para editar un jugador a la vez.		
Descripción:	Implementar una pantalla para editar los datos personales del jugador la cual se navegue desde la pantalla de jugadores.		
Prioridad:	Media		
Criterios de aceptación:			
El administrador podrá editar los datos de los jugadores navegando a través de lapantalla jugadores la misma que deberá ser amigable con el usuario al momento de editar los datos.			

Elaborado por: Autores
Fuente: Historias de usuario

2.4.6.3. Implementación

Se realizó la implementación de las pantallas antes mencionadas en las historias de usuario. Estas permitirán al usuario final tener una mayor versatilidad al momento de utilizar la aplicación móvil.

En la figura 25 se puede apreciar el código HTML para crear nuevo partido donde se hace uso de la directiva de Angular ngModel para establecer los atributos que componen al modelo "partido". Estos atributos son el nombre del partido, la fecha y la hora de este, y también los jugadores seleccionados y no seleccionados para dicho partido.

```

<ion-item>
  <ion-label position="floating">
    Nombre del juego :</ion-label>
    <ion-input [(ngModel)]="match.name" name="name" required></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Seleccione una fecha :</ion-label>
    <ion-datetime displayFormat="DD MMM YYYY" [(ngModel)]="match.date" name="date"
      monthShortNames="jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec"></ion-datetime>
  </ion-item>

  <ion-item class="villeSearch">
    <ion-label position="floating">Ciudad :</ion-label>
    <ion-input class="villeSearchInp" (click)="onClick()" (ionBlur)="onBlur()"
      (ionChange)="searchCity(match.city)" [(ngModel)]="match.city" name="city" required></ion-input>
    <ion-list class="listAutocomplete">
      <ion-item *ngFor="let city of cityData; let i = index">
        <div (click)="cityClick(city.nom)">
          <ion-label>{{city.nom}}</ion-label>
        </div>
      </ion-item>
    </ion-list>
  </ion-item>

```

Figura 25. HTML para la creación de un partido.

A continuación, se muestra el HTML para crear nuevo entrenamiento el cual se ingresará el nombre del entrenamiento, en una lista desplegable se escogerá el tipo de dificultad del entrenamiento que va desde fácil, media y difícil. De la misma forma mediante una lista desplegable se elegirá el tiempo que durará el entrenamiento que mínimo 15 minutos y máximo 60 minutos.

```

1 <form #matchForm="ngForm" (ngSubmit)="onSubmit()">
2   <ion-grid>
3
4     <ion-item>
5       <ion-label position="floating">
6         Entrenamiento :</ion-label>
7         <ion-input [(ngModel)]="match.name" name="name" required></ion-input>
8     </ion-item>
9
10    <ion-item>
11      <ion-label>Dificultad</ion-label>
12      <ion-select [(ngModel)]="match.dificultad" name="dificultad" required>
13
14        <ion-select-option value="facil">Facil</ion-select-option>
15        <ion-select-option value="media">Media</ion-select-option>
16        <ion-select-option value="difícil">Difícil</ion-select-option>
17      </ion-select>
18    </ion-item>
19
20    <ion-item>
21      <ion-label>Duracion</ion-label>
22      <ion-select [(ngModel)]="match.duracion" name="duracion" required>
23
24        <ion-select-option value="15">15 min</ion-select-option>
25        <ion-select-option value="30">30 min</ion-select-option>
26        <ion-select-option value="45">45 min</ion-select-option>
27        <ion-select-option value="60">60 min</ion-select-option>
28      </ion-select>
29    </ion-item>

```

Figura 26. HTML para la creación de un entrenamiento.

Seguidamente se observa el HTML para seleccionar jugadores que será utilizado tanto para los partidos a disputarse, como para los entrenamientos que planifica la

escuela. Cargamos la lista de los jugadores seleccionados y no seleccionados mediante la directiva ngFor y se puede quitar un jugador ya seleccionado mediante la función removePlayer y agregar un jugador mediante la función addPlayer.

```
<ion-list class="ion-margin">
  <ion-label position="floating">jugadores seleccionados :</ion-label>
  <ion-item>
    <ion-row size="12">
      <div class="ion-wrap" *ngFor="let player of match.players">
        <ion-chip class="player-chip" (click)="removePlayer(player)">
          <ion-avatar>
            <ion-icon name="person-circle" size="large"></ion-icon>
          </ion-avatar>
          <ion-label>{{player.name}}</ion-label>
        </ion-chip>
      </div>
    </ion-row>
  </ion-item>
</ion-list>
<ion-list class="ion-margin">
  <ion-label position="floating">Jugadores no seleccionados :</ion-label>
  <ion-item>
    <ion-row size="12">
      <div class="ion-wrap" *ngFor="let player of nonSelectedPlayers">
        <ion-chip class="player-chip" (click)="addPlayer(player)">
          <ion-avatar>
            <ion-icon name="person-circle" size="large"></ion-icon>
          </ion-avatar>
          <ion-label>{{player.name}}</ion-label>
        </ion-chip>
      </div>
    </ion-row>
  </ion-item>
</ion-list>
```

Figura 27. HTML para el manejo de jugadores.

En la figura 28 podemos visualizar 3 interfaz gráficas, la primera nos muestra los botones que se le mostrara al rol de entrenador (ver cuadro No. 1) los cuales permitirán ingresar un nuevo jugador y entrenamiento.

En la segunda interfaz gráfica tenemos la lista de los entrenamientos pendientes (ver cuadro No. 2) el cual tiene nombre, fecha y la hora en la que está programado el entrenamiento el mismo que al dar clic nos dirigirá a otra pantalla donde podremos visualizar la lista de jugadores que están convocados.

El mapa de la ubicación de la cancha (ver cuadro No. 4) donde se tiene un botón que sirve para abrir en Google Maps y marcar la ruta de la cancha, finalmente la app móvil cuenta con un buscador donde se podrá buscar por nombre de entrenamiento (ver cuadro No. 3).

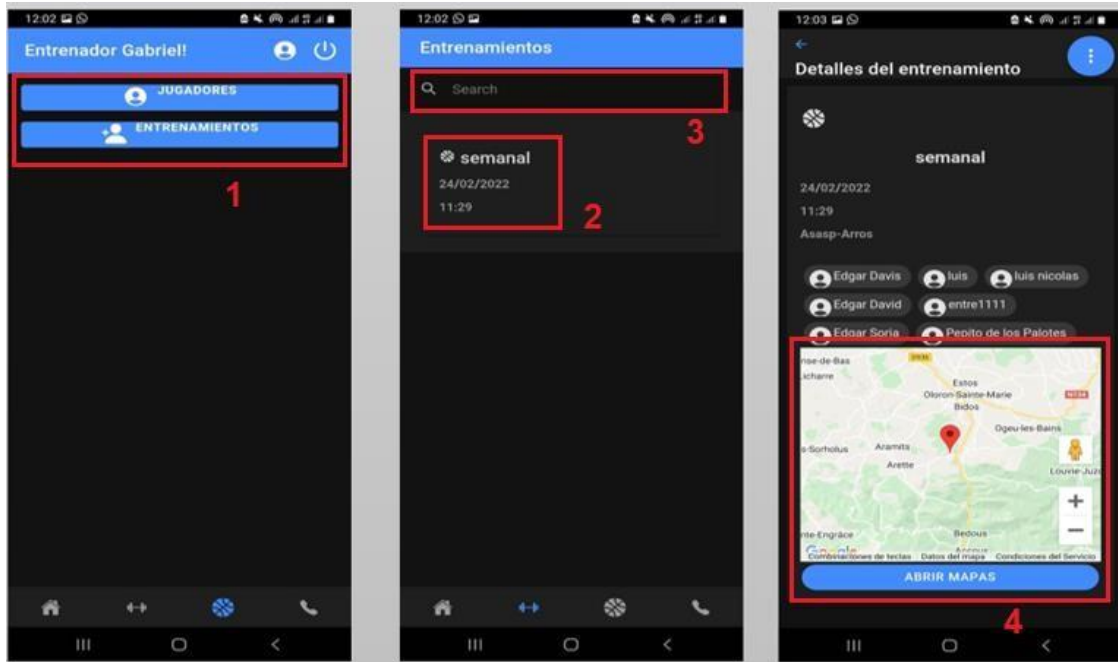


Figura 28. Vistas del módulo de entrenamiento.

En la figura 29 se puede visualizar en pantalla la lista de jugadores que están registrados para el entrenamiento (ver cuadro No. 1), donde tendremos el nombre del entrenamiento, el tiempo que dura y el nivel de dificultad del entrenamiento; de igual forma se puede acceder a una lista de reproducción para ver diversos videos acordes al nivel.

En la segunda interfaz gráfica podemos visualizar la pantalla de editar datos tanto del entrenado como del jugador (ver cuadro No 2).

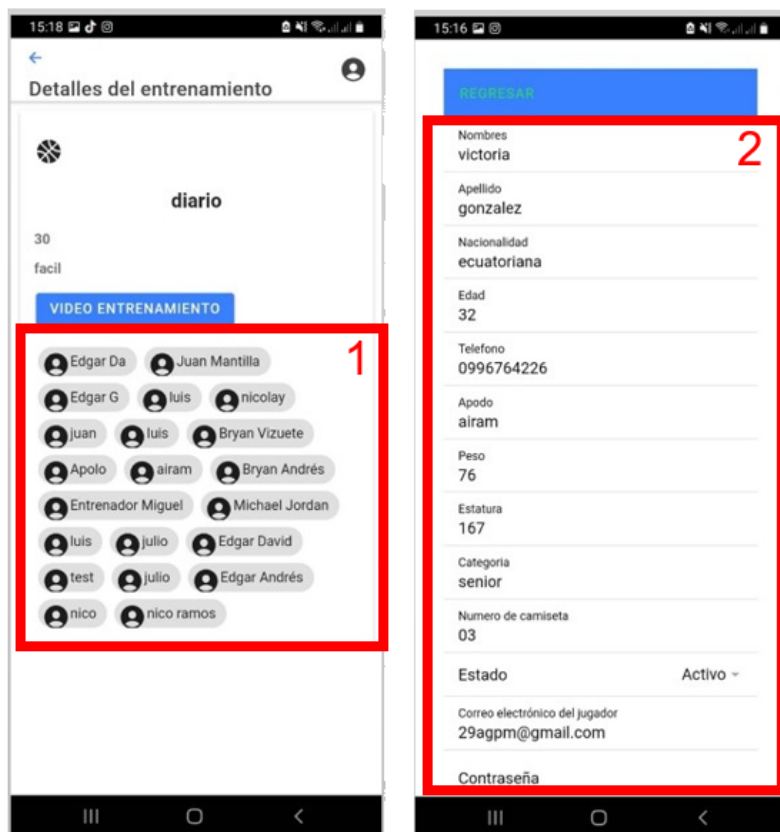


Figura 29. Vistas para listas jugadores y editarlos.

2.4.6.4. Sprint review

Al final del sprint 2 se logró cumplir con la mayoría de los objetivos planteados, los mismos que fueron implementar la pantalla para agregar un nuevo entrenamiento por parte del entrenador, implementar una pantalla para visualizar los jugadores. De la misma forma quedó pendiente la historia de usuario con el código MONSAPP-15 en la cual el usuario jugador edita sus datos personales.

No se cumplió con todas las historias de usuario del presente sprint por lo tanto se plantea para su culminación en el siguiente sprint.

En la siguiente figura se muestra la captura de la revisión del sprint 2 en la herramienta Trello:

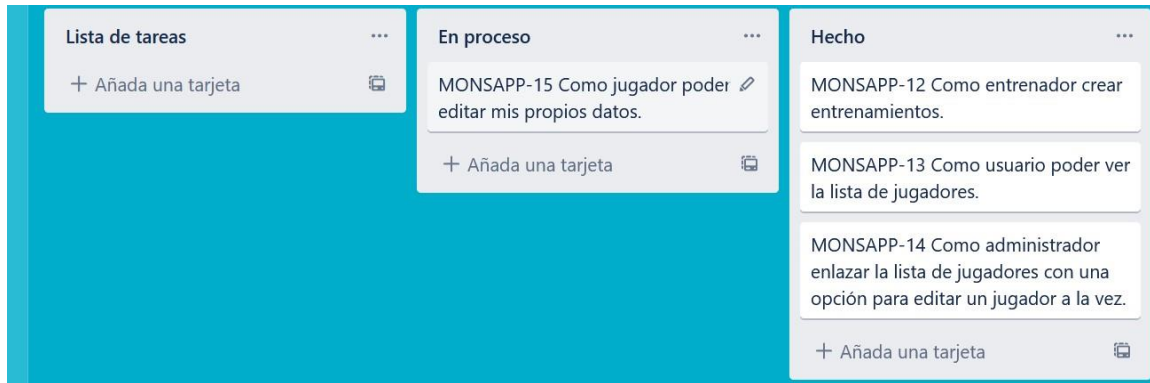


Figura 30. Finalización del sprint 2.

En la tabla 22 se puede observar las historias de usuario que se implementó en este sprint y la que quedó pendiente.

Tabla 22. Revisión sprint 2.

Código	Nombre	Observaciones	Esfuerzo inicial	Esfuerzo final	Cumplido
MONSAPP-12	Como entrenador crear entrenamientos.	Ninguna	2	2	Si
MONSAPP-13	Como usuario poder ver la lista de jugadores.	Ninguna	3	3	Si
MONSAPP-14	Como administrador enlazar la lista de jugadores con una opción para editar un jugador a la vez.	Ninguna	2	2	Si
MONSAPP-15	Como jugador poder editar mis propios datos	Pendiente	3	3	No

2.4.7. Sprint 3

2.4.7.1. Objetivo del sprint

Los objetivos de este sprint son:

- Elaborar una pantalla para poder llamar, enviar un mensaje de texto o un correo electrónico a un jugador o entrenador.
- Crear una pantalla para agregar nuevo partido.
- Crear una pantalla para mostrar la lista de partidos.
- Realizar una pantalla para mostrar a los entrenadores de la escuela.

2.4.7.2. Planificación del sprint

A continuación, se muestra en la figura 31 la planificación del sprint 3, haciendo uso de la herramienta Trello:

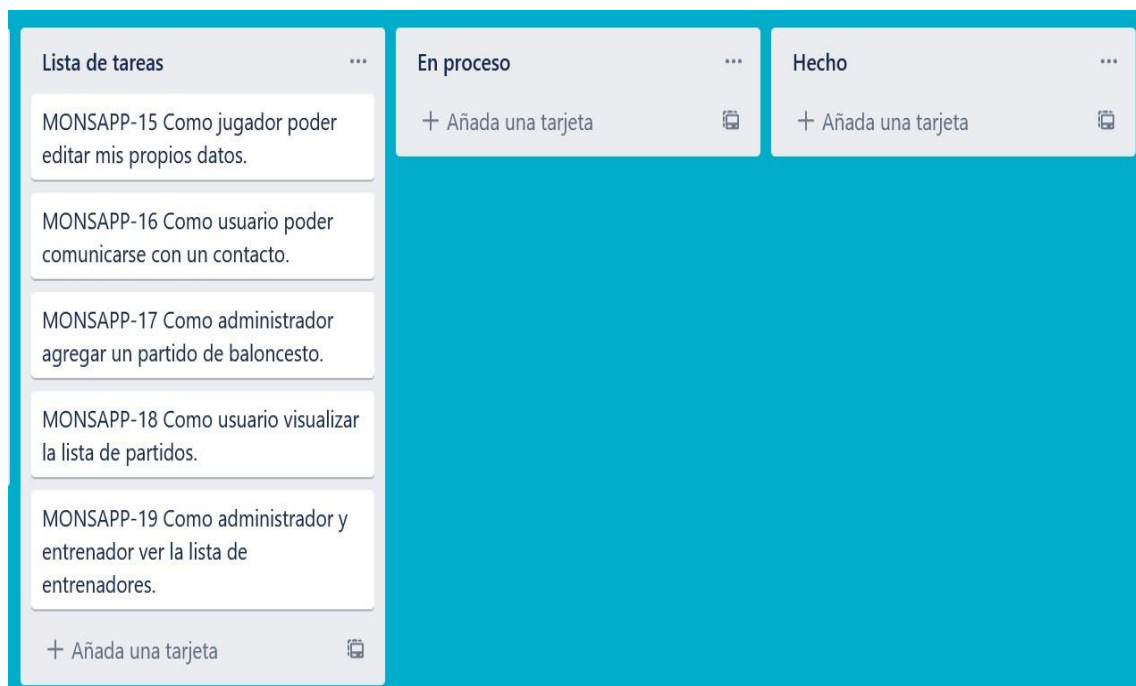


Figura 31. Planificación del sprint 3.

Las siguientes son las historias de usuario que se trabajaron en este sprint:

Tabla 23. Historia de usuario MONSAPP-15.

Código:	MONSAPP-15	Esfuerzo:	3
Nombre:	Como jugador poder editar mis propios datos.		

Descripción:	Implementar una pantalla para editar los datos personales del jugador la cual se navegue desde la pantalla de jugadores
Prioridad:	Media
Criterios de aceptación:	
El usuario jugador podrá editar sus datos personales de su perfil de una manera rápida y fácil.	

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 24. Historia de usuario MONSAPP-16.

Código:	MONSAPP-16	Esfuerzo:	4
Nombre:	Como usuario poder comunicarse con un contacto.		
Descripción:	Implementar una pantalla para poder llamar, enviar un mensaje de texto o un correo electrónico a un jugador o entrenador.		
Prioridad:	Media		
Criterios de aceptación:			
La pantalla para poder comunicarse entrenador - jugador deberá ser amigable con los usuarios facilitando la comunicación de estos.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 25. Historia de usuario MONSAPP-17.

Código:	MONSAPP-17	Esfuerzo:	5
Nombre:	Como administrador agregar un partido de baloncesto.		
Descripción:	Implementar una pantalla para poder agregar encuentros deportivos con los siguientes datos: fecha, hora, resultado, localidad, ubicación y equipo rival.		
Prioridad:	Alta		
Criterios de aceptación:			
La pantalla para poder agregar encuentros deportivos solo será visible para el administrador y permitirá el ingreso de los siguientes datos: - Fecha - Resultado - Hora - Localidad - Ubicación - Rival			

Elaborado por: Autores
Fuente: Historias de usuario

2.4.7.3. Implementación

HTML para listar partidos donde se mostrará la ciudad en la cual se disputará el partido, mediante el ngModel cargaremos la ubicación donde se jugará el partido, de igual forma listaremos la lista de los jugadores.

```
components > forms > match-form > <> match-form.component.html > form > ion-grid > ion-item.villeSearch > ion-list.listAutocomplete
<ion-item class="villeSearch">
  <ion-label position="floating">Ciudad :</ion-label>
  <ion-input class="villeSearchInp" (click)="onClick()" (ionBlur)="onBlur()"
    (ionChange)="searchCity(match.city)" [(ngModel)]="match.city" name="city" required</ion-input>
  <ion-list class="listAutocomplete">
    <ion-item *ngFor="let city of cityData; let i = index">
      <div (click)="cityClick(city.nom)">
        <ion-label>{{city.nom}}</ion-label>
      </div>
    </ion-item>
  </ion-list>
</ion-item>

<ion-input [ngModel]="match.location.latitude" name="latitude" hidden</ion-input>
<ion-input [ngModel]="match.location.longitude" name="longititude" hidden</ion-input>

I
<ion-list class="ion-margin">
  <ion-label position="floating">jugadores seleccionados :</ion-label>
  <ion-item>
    <ion-row size="12">
      <div class="ion-wrap" *ngFor="let player of match.players">
        <ion-chip class="player-chip" (click)="removePlayer(player)">
          <ion-avatar>
            <ion-icon name="person-circle" size="large"></ion-icon>
          </ion-avatar>
          <ion-label>{{player.name}}</ion-label>
        </ion-chip>
      </div>
    </ion-row>
  </ion-item>
</ion-list>
```

Figura 32. HTML para listar partidos.

Servicio remover los jugadores de la lista de ingresada tanto para entrenamiento como para juegos pendientes de igual forma tenemos la función de addPlayer el cual permite ingresar nuevos jugadores a los entrenamientos y partidos.

```

removePlayer(player: User) {
  this.create
    ? this.matchPage.removePlayer(player)
    : this.match.players.splice(this.match.players.indexOf(player), 1);

  this.nonSelectedPlayers.push(player);
}
addPlayer(player: User) {
  this.create
    ? this.matchPage.addPlayer(player)
    : this.nonSelectedPlayers.splice(
      this.nonSelectedPlayers.indexOf(player),
      1
    );
  this.match.players.push(player);
}

searchCity(city: string) {
  this.leagueServ.cityAPI(city).subscribe((data) => {
    this.cityData = data;
    console.log(data);
  });
}
cityClick(city: string) {
  this.match.city = city;
  console.log(city);
}

```

Figura 33. Servicio para el manejo de usuarios.

Método para mostrar al usuario jugador sus próximos partidos y entrenamientos pendientes en las respectivas pantallas.

```

search(evt: any) {
  let value = evt.detail.value;
  console.log(value);
  console.log(this.hiddenMatches);

  if (value == '') {
    for (const match of this.hiddenMatches) {
      this.hiddenMatches.splice(this.hiddenMatches.indexOf(match), 1);
      this.allMatches.push(match);
      this.sortMatches();
    }
  } else {
    for (const match of this.allMatches) {
      if (!match.name.toLowerCase().includes(value.toLowerCase())) {
        this.allMatches.splice(this.allMatches.indexOf(match), 1);
        this.hiddenMatches.push(match);
      }
    }

    for (const match of this.hiddenMatches) {
      if (match.name.toLowerCase().includes(value.toLowerCase())) {
        this.hiddenMatches.splice(
          this.hiddenMatches.indexOf(match),
          1
        );
        this.allMatches.push(match);
        this.sortMatches();
      }
    }
  }
}

```

Figura 34. Método para mostrar partidos y entrenamientos por jugador.

En la figura 35 (ver cuadro No. 1), se elaboró y diseño la pantalla para que los usuarios se puedan comunicar con el entrenador ya sea mediante llamada, mensaje de texto o correo electrónico.

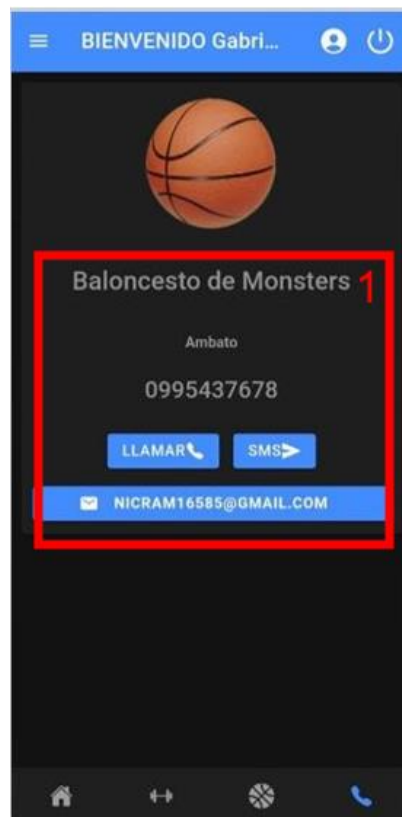


Figura 35. Interfaz para comunicarse con el administrador.

2.4.7.4. Sprint review

Al final del sprint 3 se logró cumplir con la mayoría de los objetivos planteados, los mismos que fueron implementar una pantalla que permite la comunicación mediante llamada, mensaje o correo entre los jugadores y el administrador. Se realizó una pantalla que servirá para que el usuario administrador pueda agregar nuevos partidos por disputarse.

También se cumplió con la historia de usuario pendiente del Sprint 2, la cual era implementar una pantalla para que los jugadores puedan editar sus datos personales. Esta tarea podrán realizarla tanto los entrenadores como los jugadores de la escuela de baloncesto Monsters.

Algo a tomar en cuenta es que quedaron pendientes de realizar dos historias de usuario con los códigos MONSAPP-18 y MONSAPP-19. Las mismas son realizadas para visualizar los partidos y ver la lista de los entrenados, dichas historias antes mencionados serán planificadas y elaboradas en el Sprint 4.



Figura 36. Revisión del sprint 3.

En la tabla 26 se puede observar las historias de usuario que se implementó en este sprint y las que quedaron pendientes.

Tabla 26. Revisión sprint 3.

Código	Nombre	Observaciones	Esfuerzo inicial	Esfuerzo final	Cumplido
MONSAPP-15	Como jugador poder editar mis propios datos.	Ninguna	3	3	Si
MONSAPP-16	Como usuario poder comunicarse con un contacto.	Ninguna	4	4	Si

MONSAPP-17	Como administrador agregar un partido de baloncesto.	Ninguna	5	5	Si
MONSAPP-18	Como usuario visualizar la lista de partidos.	Pendiente	2	2	No
MONSAPP-19	Como administrador y entrenador ver la lista de entrenadores.	Pendiente	2	2	No

2.4.8. Sprint 4

2.4.8.1. Objetivo del sprint

Los objetivos de este sprint son:

- Implementar una pantalla para editar la información de los entrenadores.
- Implementar una pantalla para editar los datos personales.
- Agregar los iconos de las redes sociales de la escuela.
- Mostrar la ubicación mediante Google Maps de las canchas donde se realiza los entrenamientos.
- Enviar notificaciones a los jugadores que han sido seleccionados para un partido.

2.4.8.2. Planificación del sprint

A continuación, se muestra en la figura 37 la planificación del sprint 4, haciendo uso de la herramienta Trello:

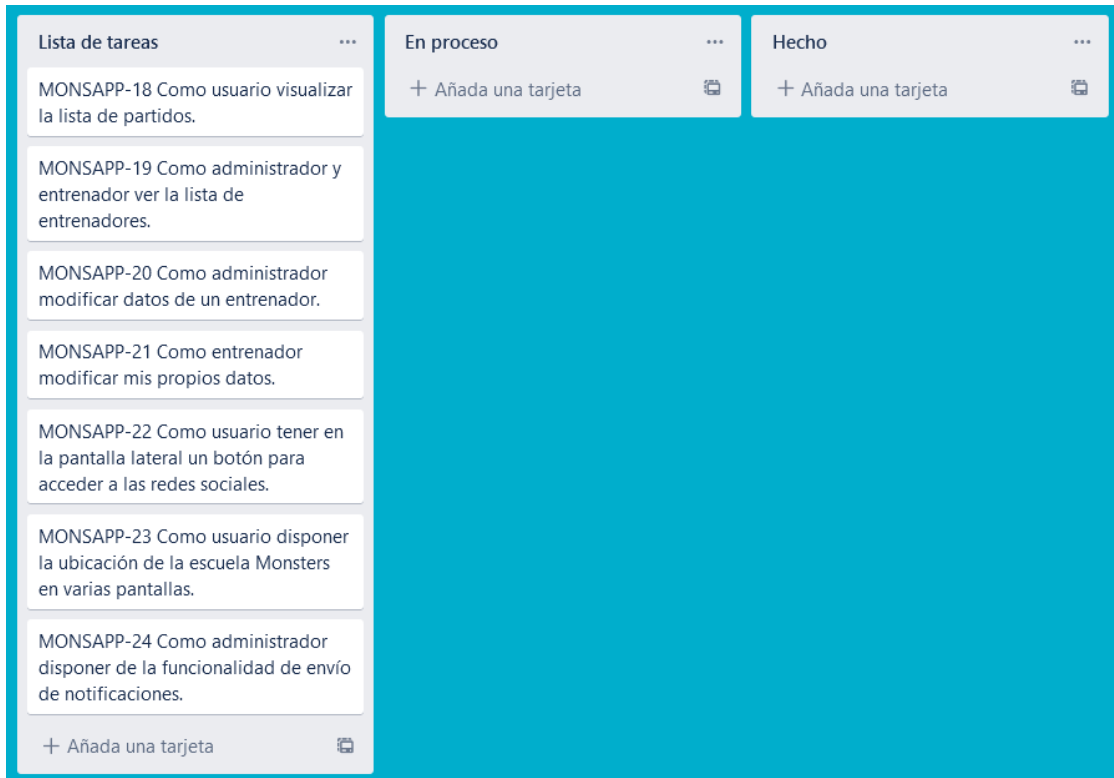


Figura 37. Planificación del sprint 4.

Las siguientes son las historias de usuario que se trabajaron en este sprint:

Tabla 27. Historia de usuario MONSAPP-18.

Código:	MONSAPP-18	Esfuerzo:	2
Nombre:	Como usuario visualizar la lista de partidos.		
Descripción:	Implementar una pantalla para mostrar la lista de partidos.		
Prioridad:	Media		
Criterios de aceptación:	Los usuarios podrán visualizar todos los partidos en forma de una lista, la pantalla tendrá que ser amigable con los usuarios y de fácil manejo.		

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 28. Historia de usuario MONSAPP-19.

Código:	MONSAPP-19	Esfuerzo:	2
Nombre:	Como administrador y entrenador ver la lista de entrenadores.		
Descripción:	Implementar una pantalla para mostrar a los		

	entrenadores de la escuela, los cuales preparan a las distintas categorías de la escuela de baloncesto.
Prioridad:	Media
Criterios de aceptación:	
Los entrenadores y administradores podrán visualizar las categorías existentes en la escuela de baloncesto.	

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 29. Historia de usuario MONSAPP-20.

Código:	MONSAPP-20	Esfuerzo:	3
Nombre:	Como administrador modificar datos de un entrenador.		
Descripción:	Implementar una pantalla para modificar la información de los entrenadores, desde el rol administrador.		
Prioridad:	Media		
Criterios de aceptación:			
El administrador podrá editar los datos de los entrenadores de una forma fácil y amigable con el usuario.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 30. Historia de usuario MONSAPP-21.

Código:	MONSAPP-21	Esfuerzo:	2
Nombre:	Como entrenador modificar mis propios datos.		
Descripción:	Implementar una pantalla para modificar la información de los entrenadores, desde el rol entrenador.		
Prioridad:	Media		
Criterios de aceptación:			
El entrenador podrá editar sus datos personales de una forma fácil y rápida.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 31. Historia de usuario MONSAPP-22.

Código:	MONSAPP-22	Esfuerzo:	1
Nombre:	Como usuario tener en la pantalla lateral un botón para acceder a las redes sociales.		
Descripción:	Implementar los íconos de las redes sociales de Facebook e Instagram.		
Prioridad:	Baja		
Criterios de aceptación:			
Al dar clic en cada uno de los íconos de las redes sociales se dirigirá al perfil en la red social específica de la escuela de baloncesto.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 32. Historia de usuario MONSAPP-23.

Código:	MONSAPP-23	Esfuerzo:	2
Nombre:	Como usuario disponer la ubicación de la escuela Monsters en varias pantallas.		
Descripción:	Mostrar la ubicación mediante el uso de Google Maps de las canchas donde entrena la escuela Monsters.		
Prioridad:	Baja		
Criterios de aceptación:			
La aplicación permitirá visualizar las canchas deportivas donde se tendrán los entrenamientos mediante la geolocalización de Google Maps.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 33. Historia de usuario MONSAPP-24.

Código:	MONSAPP-24	Esfuerzo:	3
Nombre:	Como administrador disponer de la funcionalidad de envío de notificaciones.		
Descripción:	Enviar una notificación de aviso a los jugadores que han sido seleccionados para un partido.		
Prioridad:	Baja		
Criterios de aceptación:			
La aplicación móvil enviara notificaciones avisando a los jugadores que han sido			

Elaborado por: Autores
Fuente: Historias de usuario

2.4.8.3. Implementación

A continuación, se muestra el HTML para editar datos del jugador. También se podrá editar los datos personales de entrenadores de la institución.

```
37 <ion-label position="floating">Estatura</ion-label>
38 <ion-input type="tel" [(ngModel)]="user.estatura" name="estatura" required</ion-input>
39 </ion-item>
40 <ion-item>
41 <ion-label position="floating">Categoria</ion-label>
42 <ion-input type="text" [(ngModel)]="user.categoria" name="categoria" required</ion-input>
43 </ion-item>
44 <ion-item>
45 <ion-label position="floating">Numero de camiseta</ion-label>
46 <ion-input type="tel" [(ngModel)]="user.numero" name="numero" required</ion-input>
47 </ion-item>
48 <ion-item>
49 <ion-label>Estado</ion-label>
50 <ion-select [(ngModel)]="user.estado" name="estado" required>
51
52 <ion-select-option value="activo">Activo</ion-select-option>
53 <ion-select-option value="inactivo">Inactivo</ion-select-option>
54 </ion-select>
55 </ion-item>
56 <ion-item>
57 <ion-label position="floating">Correo electrónico del jugador</ion-label>
58 <ion-input type="email" [(ngModel)]="user.email" name="email" required</ion-input>
59
60
61 </ion-item>
62 <ion-item>
63 <ion-label position="floating">Contraseña</ion-label>
64 <ion-input type="password" minlength="8" [(ngModel)]="user.password" name="password" required</ion-input>
65 </ion-item>
```

Figura 38. HTML para editar datos del usuario.

Servicio para editar datos del entrenador en el cual podemos ver que se envían los datos editados y el id del entrenador.

```
updateAccount(newUser: User, oldUser: User) {
  console.log(newUser);

  return this.afAuth
    .signInWithEmailAndPassword(oldUser.email, oldUser.password)
    .then((data) => {
      data.user.updateEmail(newUser.email);
      data.user.updatePassword(newUser.password);

      this.leagueServ.updatePlayer(newUser);
    });
}
```

Figura 39. Servicio para editar datos del entrenador.

En la figura 40 tenemos 3 interfaces gráficas. En la primera se muestra la lista de los próximos partidos que tiene el jugador (ver cuadro 1). En la segunda interfaz se muestra la lista de los entrenamientos que tiene el jugador (ver cuadro No. 2).

En la tercera interfaz se muestra la lista de los jugadores del próximo encuentro deportivo (ver cuadro No. 3) en el cual podemos visualizar el nombre, la fecha hora, lista de jugadores y ubicación de la cancha en al cual se disputará en encuentro deportivo.

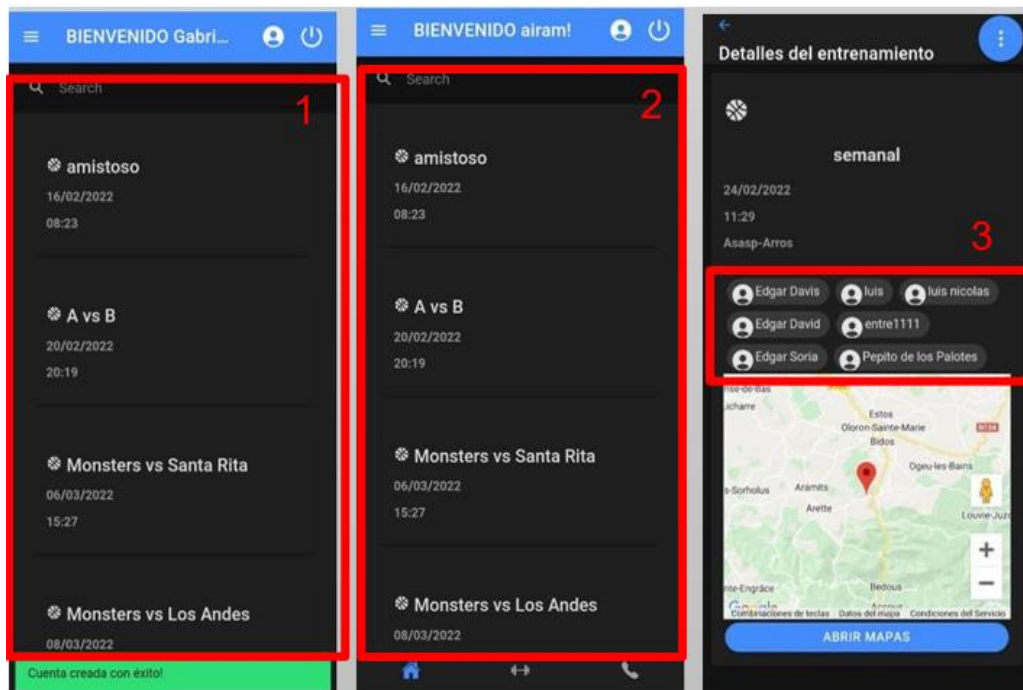


Figura 40. Interfaz de partidos y entrenamientos.

En la figura 41 se muestra el uso de la librería de Google Maps los próximos entrenamientos y partidos pendientes a disputarse (ver cuadro No. 1), como también se muestra los jugadores que fueron seleccionados para participar en el partido (ver cuadro No. 2).

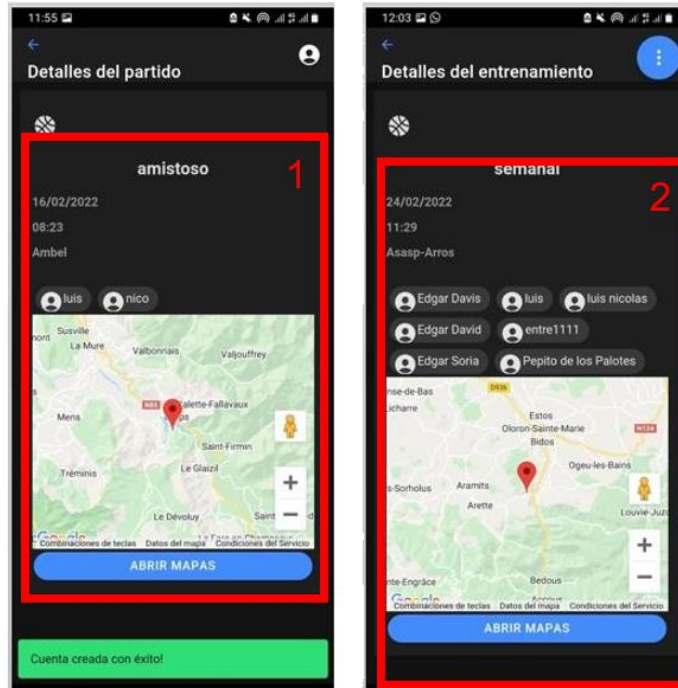


Figura 41. Uso de Google Maps en la aplicación.

En la figura 42 podemos visualizar la interfaz gráfica en la cual se muestra los próximos partidos que tiene asignado un jugador, la misma que tiene un buscador de partidos ya sea por nombre o fecha (ver cuadro No 1).

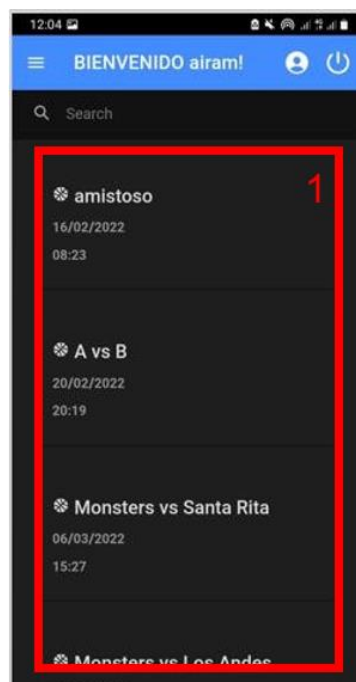


Figura 42. Partidos asignados a un jugador.

En la figura 43 se puede visualizar en pantalla el icono de las redes sociales de la escuela en el cual al dar clic sobre cualquiera de ellos nos redirigirá al perfil de usuario en sus redes sociales (ver cuadro No. 1).



Figura 43. Iconos para acceder a las redes sociales.

2.4.8.4. Sprint review

Al final del sprint 4 se logró cumplir con los objetivos planteados, los mismos que fueron implementar una pantalla donde se podrá modificar los datos de los entrenadores. Esta pantalla será accesible para el administrador y los entrenadores.

También se implementó los botones de las redes sociales de la institución, que al momento de presionar sobre cada uno nos dirige a cada uno de los perfiles oficiales de la escuela de baloncesto Monsters.

Además, se implementó una pantalla donde se puede ver las canchas de los entrenamientos mediante Google Maps y finalmente se programó las notificaciones en la aplicación móvil, las mismas que avisarán a los jugadores que han sido seleccionados para el próximo partido a jugarse.

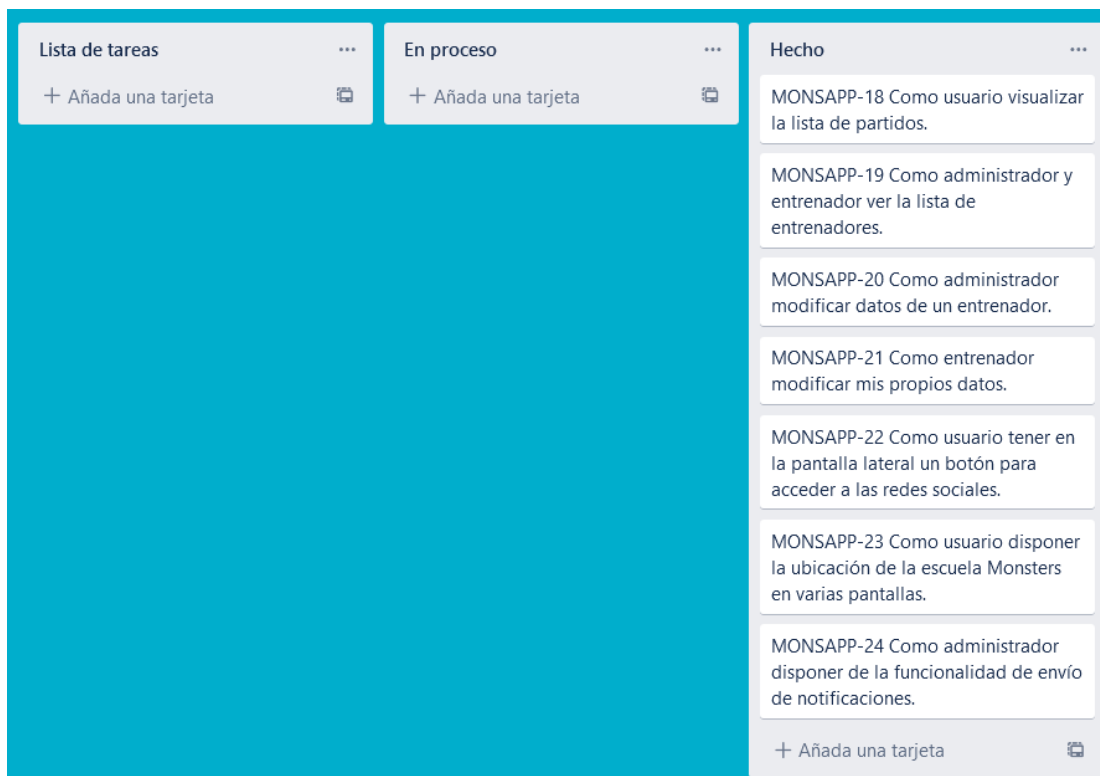


Figura 44. Revisión del sprint 4.

En la tabla 34 se puede observar las historias de usuario que se implementó en este sprint y las que quedaron pendientes.

Tabla 34. Revisión sprint 4.

Código	Nombre	Observaciones	Esfuerzo inicial	Esfuerzo final	Cumplido
MONSAPP-18	Como usuario visualizar la lista de partidos.	Ninguna	2	2	Si
MONSAPP-19	Como administrador y entrenador ver la lista de entrenadores.	Ninguna	2	2	Si
MONSAPP-20	Como administrador	Ninguna	3	3	Si

	modificar datos de un entrenador.				
MONSAPP-21	Como entrenador modificar mis propios datos.	Ninguna	2	2	Si
MONSAPP-22	Como usuario tener en la pantalla lateral un botón para acceder a las redes sociales.	Ninguna	1	1	Si
MONSAPP-23	Como usuario disponer de la ubicación de la escuela Monsters en varias pantallas.	Ninguna	2	2	Si
MONSAPP-24	Como administrador disponer de la función de envío de notificaciones.	Ninguna	3	3	Si

2.4.9. Sprint 5

2.4.9.1. Objetivo del sprint

Los objetivos de este sprint son:

- Implementar una pantalla que permitirá agregar equipos rivales.

- Implementar la funcionalidad de recuperar la clave en la pantalla de inicio de sesión.
- Implementar tres botones en la barra lateral que me permitan ver la misión, visión y campeonatos de la escuela de baloncesto.

2.4.9.2. Planificación del sprint

A continuación, se muestra en la figura 45 la planificación del sprint 5, haciendo uso de la herramienta Trello:

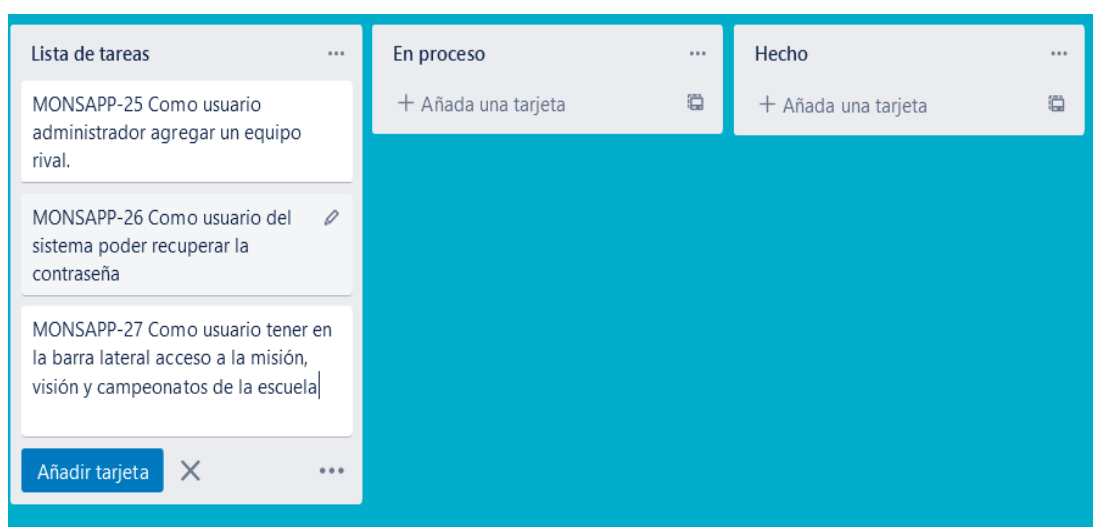


Figura 45. Planificación del sprint 5.

Las siguientes son las historias de usuario que se trabajaron en este sprint:

Tabla 35. Historia de usuario MONSAPP-25.

Código:	MONSAPP-25	Esfuerzo:	2
Nombre:	Como usuario administrador agregar un equipo rival.		
Descripción:	Implementar una pantalla para poder agregar equipos rivales con los siguientes datos: nombre, provincia, cantón y estado.		
Prioridad:	Media		
Criterios de aceptación:	El administrador puede agregar un equipo rival que jugará contra la escuela Monsters.		

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 36. Historia de usuario MONSAPP-26.

Código:	MONSAPP-26	Esfuerzo:	4
Nombre:	Como usuario del sistema poder recuperar la contraseña.		
Descripción:	Implementar la funcionalidad de recuperar la clave en la pantalla de inicio de sesión		
Prioridad:	Alta		
Criterios de aceptación:			
Se tiene una pantalla donde los usuarios pueden recuperar su clave en caso de perderla u olvidarla.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 37. Historia de usuario MONSAPP-27.

Código:	MONSAPP-27	Esfuerzo:	2
Nombre:	Como usuario tener en la barra lateral acceso a la misión, visión y campeonatos del equipo.		
Descripción:	Implementar tres botones en la barra lateral que permitan acceder y conocer la misión, visión y campeonatos logrados por la escuela.		
Prioridad:	Baja		
Criterios de aceptación:			
Se tiene una pantalla donde los usuarios podrán desplegar el menú y visualizar la misión y la visión de la escuela de baloncesto como también podrá ver los campeonatos de baloncesto.			

Elaborado por: Autores

Fuente: Historias de usuario

2.4.9.3. Implementación

A continuación, se observa el HTML para ingresar los datos del equipo rival. Para un óptimo manejo de los datos de los equipos rivales, se implementó un modelo que contiene los atributos de nombre, provincia y cantón al que pertenece el equipo.

```

<form #matchForm="ngForm" (ngSubmit)="onSubmit()">
  <ion-toolbar *ngIf!="create" color="primary">
    <ion-button (click)="onCancel()" fill="clear" color="success">
      <ion-icon name="arrow-back-outline"></ion-icon>
    </ion-button>
    Informacion Equipos
  </ion-toolbar>
  <ion-grid>
    <ion-item>
      <ion-label position="floating">
        Nombre del Equipo :</ion-label>
      <ion-input [(ngModel)]="match.nombre" name="nombre" required></ion-input>
    </ion-item>

    <ion-item >
      <ion-label>Provincia</ion-label>
      <ion-select [(ngModel)]="match.provincia" name="provincia" (ionChange)="selectProvincia($event)" int
        <ion-select-option *ngFor="let provincia of arrayProvincia" [value]="provincia.id">{{provincia.n
      </ion-select>
    </ion-item>
  </ion-grid>
</form>

```

Figura 46. HTML para ingresar un equipo rival.

Para poder recuperar la contraseña en caso de que el usuario no la recuerde, se hace uso de un servicio propio de Firebase, el cual simplemente necesita como parámetro el correo electrónico del usuario para poder enviar un correo con un enlace para que ingrese una nueva contraseña.

```

<ion-title> Recuperar Password</ion-title>
</ion-toolbar>
</ion-header>
<ion-content>
  <div class="logindiv">
    <ion-card>
      <form #loginForm="ngForm" >
        <ion-grid>
          <ion-row class="ion-justify-content">
            <ion-col size="12" class="ion-text-center">
              <ion-label>Ingresa el correo</ion-label>
              <ion-input [(ngModel)]="email" required name="email" type="text">
            </ion-col>

            <ion-col class="ion-text-center">
              <ion-button type="submit" class="ion-text-center"
                (click)="recover()"
                recuperar
              </ion-button>
            </ion-col>
          </ion-row>
        </ion-grid>
      </form>
    </ion-card>
  </div>
</ion-content>

```

Figura 47. HTML para recuperar la contraseña.

Seguidamente se puede ver el HTML correspondiente a la barra lateral, la cual contiene las opciones para dirigirse al menú, la misión, visión y ver los campeonatos obtenidos por la escuela de baloncesto. Se puede apreciar que cada opción tiene su función asignada que realiza la acción al dar clic sobre ésta.

```
<ion-grid>
  <ion-item #sidenav="matSidenav" class="mat-elevation-z8 ">
    <div>
      <button mat-button class="menu-button">
        <ion-icon class="material-icons" onClick="gotoMenu()">
          Menú
        </ion-icon>
      </button>
      <button mat-button class="menu-button">
        <ion-icon class="material-icons" onClick="showMisioScreen()">
          Misión
        </ion-icon>
      </button>
      <button mat-button class="menu-button">
        <ion-icon class="material-icons" onClick="showVisionScreen()">
          Visión
        </ion-icon>
      </button>
      <button mat-button class="menu-button">
        <ion-icon class="material-icons" onClick="showChampionships()">
          Campeonatos
        </ion-icon>
      </button>
    </div>
  </ion-item>
</ion-grid>
```

Figura 48. HTML con las opciones de la barra lateral.

En la figura 49 se presenta la pantalla para agregar un equipo rival. Se puede apreciar los campos nombre, provincia, cantón y estado que deben ser ingresados obligatoriamente (ver cuadro No. 1).

Bajo este formulario, también se muestra una lista con todos los equipos rivales que han sido registrados previamente (ver cuadro No. 2).

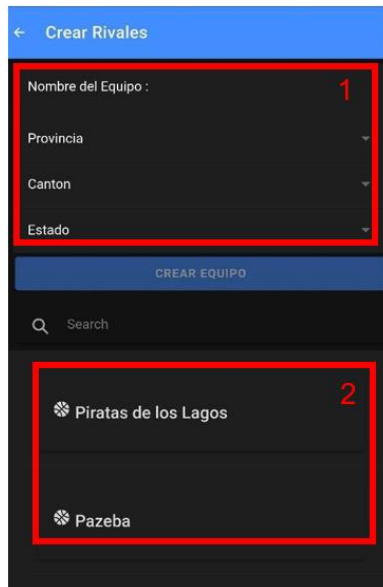


Figura 49. Interfaz para crear equipos rivales.

2.4.9.4. Sprint review

Al final del Sprint 5 se logró cumplir con los objetivos planteados los cuales fueron implementar una pantalla la cual permitirá agregar equipos rivales con los datos de nombre, provincia y el cantón al cual pertenece el equipo. Asimismo, se implementó la función para recuperar la clave de usuario en caso de que se haya perdido u olvidado.

Finalmente se implementó un menú con tres botones en la barra lateral en la cual se puede conocer la misión y la visión de la escuela como también los campeonatos de la escuela de baloncesto Monsters.

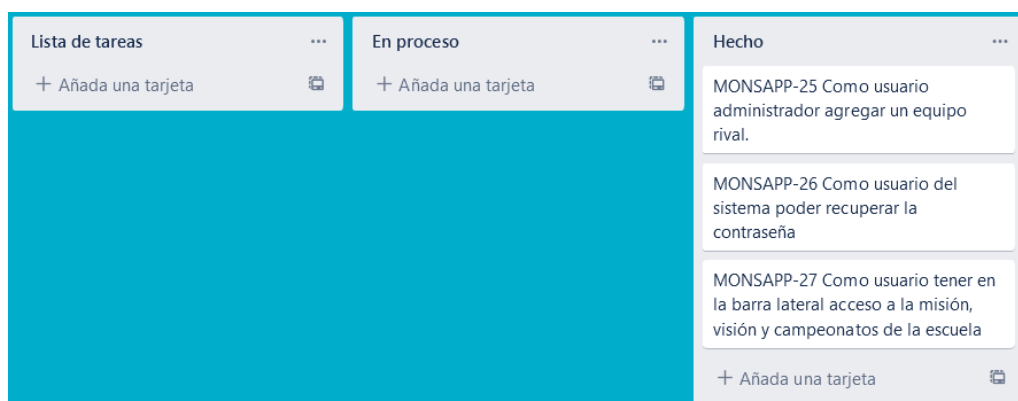


Figura 50. Revisión del sprint 5.

En la tabla 38 se puede apreciar las historias de usuario que se implementó en este sprint.

Tabla 38. Revisión sprint 5.

Código	Nombre	Observaciones	Esfuerzo inicial	Esfuerzo final	Cumplido
MONSAPP-25	Como usuario visualizar la lista de partidos.	Ninguna	2	2	Si
MONSAPP-26	Como usuario del sistema poder recuperar la contraseña.	Ninguna	4	4	Si
MONSAPP-27	Como usuario tener en la barra lateral acceso a la misión, visión y los campeonatos de la escuela.	Ninguna	2	2	Si

2.4.10. Sprint 6

2.4.10.1. Objetivo del sprint

Los objetivos del sprint son:

- Implementar una pantalla que permitirá agregar torneos con la información del nombre, fecha inicio, fecha fin y resultado del torneo.
- Implementar la funcionalidad de reportes. Un reporte que se va a visualizar es con la información de los torneos y otro reporte con toda la información de los jugadores.

2.4.10.2. Planificación del sprint

A continuación, se muestra en la figura 51 la planificación del sprint 6, haciendo uso de la herramienta Trello:

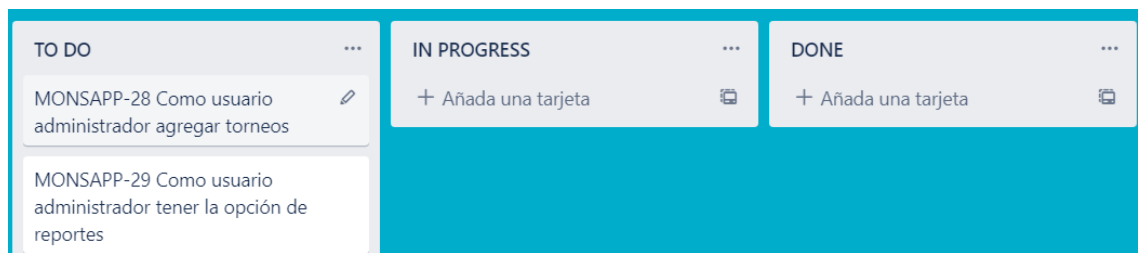


Figura 51. Planificación del sprint 6.

Las siguientes son las historias de usuario que se trabajaron en este sprint:

Tabla 39. Historia de usuario MONSAPP-28.

Código:	MONSAPP-28	Esfuerzo:	4
Nombre:	Como usuario administrador agregar torneos.		
Descripción:	Implementar una pantalla que me permita agregar torneos con los siguientes datos: nombre, fecha inicio, fecha fin y resultado.		
Prioridad:	Media		
Criterios de aceptación:			
Se tiene una pantalla donde se puede agregar torneos.			

Elaborado por: Autores

Fuente: Historias de usuario

Tabla 40. Historia de usuario MONSAPP-29.

Código:	MONSAPP-29	Esfuerzo:	4
Nombre:	Como usuario administrador tener la opción de reportes.		
Descripción:	Implementar la opción de visualizar dos reportes, uno con la información detallada de torneos y otro con toda la información de los jugadores.		
Prioridad:	Media		
Criterios de aceptación:			
Se tienen dos reportes que son visualizados por el administrador con información útil de los torneos y otro con información detallada de los jugadores.			

Elaborado por: Autores

Fuente: Historias de usuario

2.4.10.3. Implementación

Para el tema de reportes, la funcionalidad clave se centra en la pantalla que va a visualizar el administrador. En la figura 52 se puede observar que primero se crea el título del reporte, luego procedemos a establecer qué columnas serán necesarias para el reporte, en este caso, el de jugadores.

```
<ion-header [translucent]="true">
  <ion-toolbar color="secondary">
    <ion-buttons slot="start">
      <ion-menu-button menu="admin" autohide="false"></ion-menu-button>
    </ion-buttons>
    <ion-title>
      Reporte Usuarios
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">

  <ion-grid>
    <ion-row class="ion-row" >
      <ion-col text-wrap class="bgcolor">
        NOMBRES
      </ion-col>
      <ion-col text-wrap class="bgcolor">
        EDAD
      </ion-col>
      <ion-col text-wrap class="bgcolor">
        CAT.
      </ion-col>
      <ion-col text-wrap class="bgcolor">
```

Figura 52. HTML para el reporte de jugadores.

Algo similar a lo descrito anteriormente ocurre con el reporte de torneos. Lo que cambia en este caso es la información que se trae y las columnas del reporte que se presentan. Las columnas para este reporte son nombre, fecha inicio, fecha fin y resultado del torneo. Todo esto se puede observar en la figura 53.

```
<ion-header [translucent]="true">
  <ion-toolbar color="secondary">
    <ion-buttons slot="start">
      <ion-menu-button menu="admin" autohide="false"></ion-menu-button>
    </ion-buttons>
    <ion-title>
      Reporte Torneos
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">

  <ion-grid>
    <ion-row class="ion-row" >
      <ion-col text-wrap class="bgcolor">
        NOM
      </ion-col>
      <ion-col text-wrap class="bgcolor">
        INICIO
      </ion-col>
      <ion-col text-wrap class="bgcolor">
        FIN
      </ion-col>
      <ion-col text-wrap class="bgcolor">
        RESULTADO
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>
```

Figura 53. HTML para el reporte de torneos.

En la figura 54 se puede apreciar el reporte correspondiente a todos los jugadores de la escuela de baloncesto Monsters. Este reporte tiene las columnas nombre, edad, categoría y número de camiseta.

A primera vista se puede notar que el listado de jugadores no está ordenado alfabéticamente por el apellido del jugador. Por requerimiento del administrador y entrenadores de la escuela, el reporte está ordenado de acuerdo con la categoría a la que pertenece el jugador, ya sea alevín, cadete, infantil, etc.

Reporte de Jugadores			
NOMBRES	EDAD	CAT.	NUM.
Mosquera Amara	10	Alevín	1
Nuñes David	12	Alevín	3
Mosquera Valeria	10	Alevín	2
Jiménez Manuel	10	Alevín	15
Marchesini Jean	14	Cadete	10
Galvez Nicole	15	Cadete	19
Vaca Roger	15	Cadete	12
Díaz Abel	15	Cadete	3
Silva Nidia	12	Infantil	17
Gálvez Ema	13	Infantil	5
Reyes Yina	13	Infantil	13

Figura 54. Vista del reporte de jugadores.

2.4.10.4. Sprint review

Al final del sprint 6 se logró cumplir con los objetivos planteados, los cuales fueron implementar una pantalla para agregar torneos con la información del nombre, fecha inicio, fecha fin y resultado de la escuela Monsters en dicho torneo (final, semifinal, cuartos, etc.).

También se implementó la opción de visualizar dos reportes. Un primer reporte nos muestra la información detallada de todos los torneos en los que la escuela ha participado. Otro torneo contiene toda la información detallada de todos los jugadores ordenado de acuerdo con su categoría.

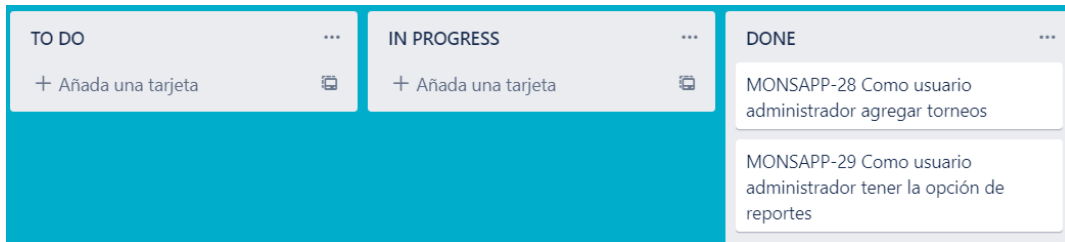


Figura 55. Revisión del sprint 6.

En la tabla 41 se puede observar las historias de usuario que se implementaron en este sprint:

Tabla 41. Revisión sprint 6.

Código	Nombre	Observaciones	Esfuerzo inicial	Esfuerzo final	Cumplido
MONSAPP-28	Como usuario administrador agregar torneos.	Ninguna	4	4	Si
MONSAPP-29	Como usuario administrador tener la opción de reportes.	Ninguna	4	4	Si

Capítulo 3. Resultados y discusión

En este capítulo se presentan los resultados obtenidos en las pruebas de usabilidad. Se definió una encuesta para los casos de usabilidad, la cual sirvió para que los entrenadores puedan realizar pruebas de funcionabilidad más importantes y de uso más frecuente en el sistema.

Las pruebas de usabilidad se realizaron a un grupo de 2 entrenadores y 20 jugadores existentes en la escuela. La encuesta se realizó con el fin de determinar la usabilidad que tendrá la aplicación en la escuela de baloncesto la misma que estuvo laborada con 10 preguntas cualitativas. Las preguntas tuvieron el objetivo de identificar las dificultades, error y mejoras para el sistema.

3.1. Pruebas de usabilidad

Las pruebas de usabilidad de la aplicación móvil se realizaron aplicando una encuesta de tiene lineamientos y métricas de SUS (System Usability Scale).

Beneficios de usar un SUS

SUS se ha convertido en un estándar de la industria, con referencias en más de 1300 artículos y publicaciones. Los beneficios señalados de usar SUS incluyen que:

- Es una escala muy fácil de administrar a los participantes.
- Se puede utilizar en tamaños de muestra pequeños con resultados fiables.
- Es válido: puede diferenciar efectivamente entre sistemas utilizables e inutilizables.

Consideraciones al usar un SUS.

Si está considera el uso de SUS, se debe tener en cuenta lo siguiente:

- El sistema de puntuación es algo complejo.
- Existe la tentación, cuando se mira los puntajes, ya que están en una escala de 0-100, de interpretarlos como porcentajes, pero eso no es correcto.
- La mejor manera de interpretar los resultados consiste en "normalizar" las puntuaciones para producir una clasificación de percentiles.

- SUS no es un diagnóstico: su uso es para clasificar la facilidad de uso del sitio, la aplicación o el entorno que se está probando.

3.2. Sistema de escala de usabilidad

El Sistema de escala de usabilidad nos permite probar el hardware y software de la presente aplicación móvil.

Para realizar la prueba de usabilidad de la aplicación móvil se diseñó un cuestionario con 13 preguntas de opción múltiple para saber el grado de usabilidad de la aplicación móvil.

Tabla 42. Preguntas del cuestionario de usabilidad.

Pregunta	Siendo 1 fácil y 5 difícil				
¿Qué tan complicado le resulta registrarse en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta iniciar sesión en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta crear un jugador en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta crear un partido en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta crear un entrenador en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta crear un entrenamiento en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta editar los datos de un jugador en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta editar los datos de un partido en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta crear un equipo rival en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta crear un equipo rival en la aplicación?	1	2	3	4	5
¿Qué tan complicado le resulta usar la pantalla para llamar, enviar un mensaje y enviar un correo en la aplicación al contacto?	1	2	3	4	5

Pregunta	Siendo 1 probable y 5 poco probable				
¿Utilizaría la aplicación para la administración de la escuela de baloncesto?	1	2	3	4	5
Pregunta	Siendo 1 fácil y 5 difícil				
En términos generales ¿Cómo calificaría la experiencia de uso de la aplicación?	1	2	3	4	5

Para obtener los resultados de usabilidad se tiene que sumar los resultados promediados obtenidos de los cuestionarios realizados a los jugadores, tomando en cuenta las siguientes recomendaciones:

- Las preguntas impares tomaran el valor asignado por el usuario y se le restara 1.
- Las preguntas pares serán de menos 5 el valor asignado por los usuarios.
- Una vez obtenido los resultados, será multiplicado por 2,5.

3.3. Casos de prueba

A continuación, se detalla los casos de prueba que fueron ejecutados por cada usuario.

Tabla 43. Caso de prueba 1.

Caso de prueba 1		
Objetivo del caso de prueba	Registrar cuentas de usuarios para administradores, entrenadores y jugadores.	
Funcionalidad evaluada	Registrar usuario entrenador y jugador.	
Prerrequisitos	Navegar a la pantalla para registrar usuario y llenar datos de los usuarios a registrarse.	
Flujo	Actor	Aplicación
	Ingresar datos.	
	Clic en el botón de Registro.	
		Valida datos.

		Guarda datos.
Resultado esperado	Se abrirá la aplicación y se ingresará los datos del nuevo usuario y se guardaran normalmente.	
Resultado obtenido	Se registró el usuario satisfactoriamente.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 44. Caso de prueba 2.

Caso de prueba 2		
Objetivo del caso de prueba	Uso de la pantalla para agregar un entrenamiento.	
Funcionalidad evaluada	Agregar entrenamiento.	
Prerrequisitos	Navegar a la pantalla de nuevo entrenamiento y llenar los datos para el nuevo entrenamiento.	
Flujo	Actor	Aplicación
	Navegar a la pantalla de nuevo entrenamiento.	
	Ingresar los datos del entrenamiento.	
	Clic en guardar.	
		Valida datos.
		Guarda datos.
	Muestra el nuevo entrenamiento.	
Resultado esperado	Se abrirá la pantalla para agregar un nuevo entrenamiento, se llenará los datos y se guarda el nuevo entrenamiento.	
Resultado obtenido	Datos guardados correctamente.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 45. Caso de prueba 3.

Caso de prueba 3	
Objetivo del caso de prueba	Implementar una pantalla para mostrar a los jugadores de la escuela.

Funcionalidad evaluada	Mostrar jugadores de la escuela.	
Prerrequisitos	Navegar a la pantalla para mostrar los jugadores registrados.	
Flujo	Actor	Aplicación
	Navegar a la pantalla para mostrar jugadores.	
	Consultar la lista de los Jugadores.	
		Muestra jugadores
	Ver la pantalla de jugadores.	
Resultado esperado	Mostrar la pantalla para ver jugadores.	
Resultado obtenido	Lista jugadores	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 46. Caso de prueba 4.

Caso de prueba 4		
Objetivo del caso de prueba	Diseñar una pantalla para editar los datos personales del jugador.	
Funcionalidad evaluada	Editar datos personales de los jugadores.	
Prerrequisitos	Navegar a la pantalla para editar los datos.	
Flujo	Actor	Aplicación
	Navega a la pantalla para editar datos.	
	Cambiar datos.	
	Clic en guardar.	
		Valida datos.
		Guarda datos.
Resultado esperado	Editar datos de los jugadores.	
Resultado obtenido	Datos guardados satisfactoriamente.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 47. Caso de prueba 5.

Caso de prueba 5		
Objetivo del caso de prueba	Diseñar una pantalla para agregar encuentros deportivos.	
Funcionalidad evaluada	Agregar encuentros deportivos.	
Prerrequisitos	Navegar a la pantalla para ingresar un nuevo encuentro deportivo.	
Flujo	Actor	Aplicación
	Navega a la pantalla de ingresar nuevo encuentro.	
	Ingresar datos.	
	Clic en guardar.	
		Valida datos.
		Guarda datos.
Resultado esperado	Se ingresa nuevo encuentro deportivo.	
Resultado obtenido	Nuevo encuentro deportivo ingresado.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 48. Caso de prueba 6.

Caso de prueba 6		
Objetivo del caso de prueba	Elaborar una pantalla para poder llamar, enviar un mensaje de texto o un correo electrónico a un jugador o entrenador.	
Funcionalidad evaluada	Enviar mensaje de texto o correo.	
Prerrequisitos	Navegar a la pantalla para enviar mensaje o correo electrónico.	
Flujo	Actor	Aplicación
	Navega a la pantalla.	
	Enviar mensaje o correo.	
	Clic en enviar.	
		Envía mensaje.

Resultado esperado	Enviara mensaje a los usuarios.
Resultado obtenido	Mensaje enviado con éxito.
Observaciones	El resultado de la prueba fue exitoso.

Tabla 49. Caso de prueba 7.

Caso de prueba 7		
Objetivo del caso de prueba	Diseñar una pantalla para agregar nuevo partido.	
Funcionalidad evaluada	Agregar nuevo partido.	
Prerrequisitos	Navegar a la pantalla para agregar nuevo partido.	
Flujo	Actor	Aplicación
	Navega a la pantalla de nuevo partido.	
	Ingresar datos.	
	Clic en guardar.	
		Valida datos.
		Guarda datos.
Resultado esperado	Se ingresa nuevo encuentro deportivo.	
Resultado obtenido	Nuevo encuentro deportivo ingresado.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 50. Caso de prueba 8.

Caso de prueba 8		
Objetivo del caso de prueba	Crear una pantalla para mostrar la lista de partidos.	
Funcionalidad evaluada	Mostrar la lista de partidos.	
Prerrequisitos	Navegar a la pantalla para listar partidos	
Flujo	Actor	Aplicación
	Navega a la pantalla.	
	Consulta lista de partido.	
		Muestra lista de

		partidos.
Resultado esperado	Listar partidos.	
Resultado obtenido	Partidos mostrados en pantalla.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 51. Caso de prueba 9.

Caso de prueba 9		
Objetivo del caso de prueba	Realizar una pantalla para mostrar a los entrenadores de la escuela.	
Funcionalidad evaluada	Mostrar entrenadores de la escuela.	
Prerrequisitos	Navegar a la pantalla para mostrar entrenadores.	
Flujo	Actor	Aplicación
	Navegar a la pantalla para mostrar entrenadores.	
	Clic en ver entrenadores.	
		Muestra entrenadores.
	Ver la pantalla de entrenadores.	
Resultado esperado	Mostrar la pantalla para ver entrenadores.	
Resultado obtenido	Lista de entrenadores.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 52. Caso de prueba 10.

Caso de prueba 10		
Objetivo del caso de prueba	Implementar una pantalla para editar la información de los entrenadores.	
Funcionalidad evaluada	Editar la información de los entrenadores.	
Prerrequisitos	Navegar a la pantalla para editar entrenadores.	
Flujo	Actor	Aplicación

	Navega a la pantalla.	
	Ingresa cambios.	
	Clic en guardar datos.	
		Validar datos.
		Guarda datos.
	Muestra datos.	
Resultado esperado	Editar datos de los entrenadores.	
Resultado obtenido	Datos editados.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 53. Caso de prueba 11.

Caso de prueba 11		
Objetivo del caso de prueba	Implementar una pantalla para editar los datos personales.	
Funcionalidad evaluada	Editar datos personales.	
Prerrequisitos	Navegar a la pantalla para editar datos.	
Flujo	Actor	Aplicación
	Navega a la pantalla.	
	Ingresa cambios.	
	Clic en guardar datos.	
		Validar datos.
		Guarda datos.
	Muestra datos.	
Resultado esperado	Editar datos personales.	
Resultado obtenido	Datos editados.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 54. Caso de prueba 12.

Caso de prueba 12	
Objetivo del caso de prueba	Agregar los íconos de las redes sociales de la escuela de baloncesto.

Funcionalidad evaluada	Agregar íconos de redes sociales.	
Prerrequisitos	Navegar a la red social Instagram y Facebook.	
Flujo	Actor	Aplicación
	Navegar a la pantalla de redes sociales.	
		Enlaza a las redes sociales.
	Muestra redes sociales.	
Resultado esperado	Mostrar los íconos de redes sociales.	
Resultado obtenido	Ver redes sociales.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 55. Caso de prueba 13.

Caso de prueba 13		
Objetivo del caso de prueba	Mostrar la ubicación mediante Google Maps de las canchas donde se realiza los entrenamientos.	
Funcionalidad evaluada	Ver ubicación de Google Maps.	
Prerrequisitos	Ver ubicación.	
Flujo	Actor	Aplicación
	Ingresa una ubicación.	
		Busca la ubicación
	Muestra la ubicación.	
Resultado esperado	Mostrar la ubicación de las canchas.	
Resultado obtenido	Muestra ubicación.	
Observaciones	El resultado de la prueba fue exitoso.	

Tabla 56. Caso de prueba 14.

Caso de prueba 14	
Objetivo del caso de prueba	Mostrar los partidos y entrenamientos a los jugadores que han sido seleccionados.

Funcionalidad evaluada	Mostrar partidos y entrenamientos asignados a un jugador.	
Prerrequisitos	Navegar a la pantalla de partidos y de entrenamientos.	
Flujo	Actor	Aplicación
	Navega a la pantalla de partidos y/o notificaciones.	
	Ver partido/entrenamiento asignado al jugador.	
		Muestra partido / entrenamiento correspondiente.
Resultado esperado	Partidos/entrenamientos solo por jugador.	
Resultado obtenido	Se muestran los partidos / entrenamientos solo para el jugador específico.	
Observaciones	El resultado de la prueba fue exitoso.	

Capítulo 4. Conclusiones y recomendaciones

4.1. Conclusiones

- Con la investigación bibliográfica en las diferentes bases de datos electrónicas, libros, trabajos investigativos, repositorios universitarios se logró recolectar la información necesaria para sustentar el desarrollo de la investigación.
- Se realizó una investigación de campo a la escuela deportiva de basquetbol MONSTERS la cual permitió recolectar la información por medio de una entrevista al director de la escuela y encuestas a los jugadores y entrenadores de la institución, se determinó los requerimientos para resolver la problemática que existe en la institución.
- Mediante el desarrollo de la aplicación móvil se resolvió la problemática, ya que la misma llevará la información y así minimizando el esfuerzo de los usuarios de la escuela ya que los mismos podrán hacer uso de esta y consultar toda la información necesaria para crear un entrenamiento, convocar jugadores, saber cuándo les toca jugar el próximo partido, registrar nuevos entrenadores, registrar nuevos jugadores y ya no se llevará de una forma manual como se realizaba normalmente.
- El uso de la metodología Scrum en el presente proyecto facilitó el desarrollo de la aplicación móvil ya que se realizó de una forma ordenada ya que se realizó iteraciones las mismas que fueron divididas en Sprint y así se evitó la sobrecarga de trabajo al equipo de desarrollo.
- La aplicación de pruebas unitarias al código hace que el aplicativo llegue al usuario final con la menor cantidad de errores. Otro indicador que es clave para la validez de nuestra aplicación es el resultado que se obtiene al momento de aplicar pruebas de usabilidad a los usuarios

3.2. Recomendaciones

- Se recomienda continuar desarrollando aplicaciones móviles para facilitar las tareas diarias de los usuarios ya que la tecnología está en constante evolución y cada vez nos facilita la utilización de los dispositivos móviles para realizar las tareas cotidianas y nos ahorra tiempos y costos.
- Se recomienda desarrollar aplicaciones móviles deportivas ya que en la actualidad facilita tanto a los entrenadores como a los jugadores ya que con esto les tiene en constante información y les tiene pendientes de sus próximos encuentros deportivos y entrenamientos.
- Se recomienda realizar pruebas en distintos dispositivos para poder ver su usabilidad y su forma de adaptarse a los distintos tipos de pantallas de los dispositivos móviles.
- Se recomienda utilizar la metodología de desarrollo Scrum para el desarrollo de los proyectos, ya que la misma se realiza por medio de sprints y así podemos organizar y distribuir de mejor manera las tareas con todo el equipo de desarrollo.

Glosario de términos

- **Aplicación Móvil:** es un programa que se descarga e instala en el dispositivo móvil de un usuario.
- **Backlog:** es un listado de todas las tareas que se pretenden hacer durante el desarrollo de un proyecto.
- **Framework:** es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado.
- **Ionic:** es una herramienta, gratuita y de código abierto, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS.
- **Scrum:** es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto.
- **Sprint:** es el corazón de Scrum, un intervalo de tiempo de máximo un mes.
- **Base de datos:** aplicación informática en la cual se almacena información de las aplicaciones web y móviles, las mismas que pueden ser SQL o NoSQL.
- **NoSQL:** son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación.
- **MongoDB:** es una base de datos la cual guarda la estructura de los datos en un documento JSON, la misma que no tiene un esquema definido.
- **Firebase:** es la nueva y mejorada plataforma de desarrollo móvil en la nube de Google.

- **Back-end:** es la parte lógica de la aplicación móvil donde se realizan los procesos, peticiones y conexiones de la base de datos.
- **Front-end:** es la parte de diseño y elementos gráficos de la aplicación móvil, es la cual muestra la aplicación móvil de una forma visiblemente estructurada al usuario final.
- **HTML:** es el Lenguaje de marcado de hipertexto donde se realiza la estructuración del Front-end de la aplicación.
- **CSS:** son las hojas de estilo en cascada donde se le da forma y color a la aplicación móvil.
- **JavaScript:** es un lenguaje de secuencias de comandos que permite realizar contenido más dinámico y atractivo para facilitar al usuario final el funcionamiento de la aplicación.
- **Herramienta:** son programas o instrucciones usadas para realizar otras tareas de una forma más rápida y sencilla.
- **Implementar:** es la instalación de una aplicación informática, ejecutando un plan, estándar o algoritmo.
- **Método:** secuencia de actividades orientadas a una meta común.
- **Metodología:** es el conjunto de métodos que se cumplen en una investigación para llegar al objetivo.
- **Requerimiento:** características que se desea que tenga la aplicación móvil.

Referencias bibliográficas

- [1] Artes-Deporte, «Importancia del deporte,» 2020. [En línea]. Available: <https://www.importancia.org/deporte.php#:~:text=La%20importancia%20d el%20deporte%20es,tensi%C3>.
- [2] World Health Organization, «Recomendaciones mundiales sobre actividad física para la salud,» 2010. [En línea].
- [3] J. Astudillo, «Historia del deporte ecuatoriano,» 27 Junio 2012. [En línea]. Available: <http://juanastudilloaviles.blogspot.com/2012/06/lauro-historia-del-deporte.HTML>.
- [4] Quito.gob, 24 Octubre 2020. [En línea]. Available: http://www7.quito.gob.ec/mdmq_ordenanzas/Administraci%C3%B3n%202019-2023/Comisiones%20del%20Concejo%20Metropolitano/Deporte%20y%20Recreaci%C3%B3n/2019/2019-09-26/Punto%203/Escuelas%20Deportivas%20Baloncesto.pdf.
- [5] M. Alban, «Drive,» 5 Noviembre 2020. [En línea]. Available: <https://drive.Google.com/drive/u/0/folders/1bZ4hxS-S13inNqyKmAxmENS0i4I0-Om1>.
- [6] Z. Cataldi, F. Lage, R. Pessacq y R. García, «In Proceedings del V Congreso Internacional de Ingeniería Informática,» 1999.
- [7] J. Gómez y M. Fuentes, «Taxonomía de los modelos y metodologías de desarrollo de software más utilizados,» *Universidades*, vol. 62, nº 52, pp.37-47, 2021.
- [8] A. Paul y A. Nalwara, «Resolviendo problemas de manera diferente,» *React Native para el desarrollo de iOS*, 2016.
- [9] Understanding MVC , «Architecture with React,» 2020. [En línea]. Available: <https://medium.com/createdd-notes/understanding-mvc->

architecture-with-react-6cd38e91fehd.

- [10] Proyectos Ágiles, «Qué es SCRUM. Proyectos Ágiles,» 2020. [En línea].
Available: <https://proyectosagiles.org/que-es-scrum/>.
- [11] G. Terán, V. Oña, J. Cobos y D. Miniguano, «Incidencia de dispositivos

Anexos

ANEXO 1: ENCUESTA

<https://docs.Google.com/forms/d/1WI27KN1UkirwVE02-bCduylfdgdsIN1kfEweYaiZCU4/edit?ts=62602f94>