

**SP – TRAINER, APLICACIÓN PARA GESTIONAR
GRUPOS DE ENTRENAMIENTO EN LA CIUDAD DE
SANTA MARTA**

**SP – TRAINER, APLICACIÓN PARA GESTIONAR GRUPOS DE
ENTRENAMIENTO EN LA CIUDAD DE SANTA MARTA**



Integrantes

Dayanis Pacheco & Luigy Márquez.

Asesores

Dianalin Neme Prada

Ivan Romero

Miguel Felizzola

Elkin Quintero

Jose Sánchez

Universidad Antonio Nariño.

Bogotá D.C.

2020



Tabla de Contenido

ii

Capítulo 1. Introducción	1
Capítulo 2. Marco Teórico	2
2.1 ¿Qué es Software?.....	2
2.2 Ingeniería del Software	2
2.3 Entrenador Personal.....	3
2.4 Java.....	3
2.5 Angular (framework)	4
2.6 MySQL	4
Capítulo 3. Estado del Arte.....	6
3.1 Resumen del estado del arte.....	6
3.2 Impacto.....	9
3.3 Componente de innovación.....	9
Capítulo 4. Formulación del Problema.....	10
Capítulo 5. Objetivos.....	13
5.1 Objetivo General.....	13
5.2 Objetivos Específicos.....	13
Capítulo 6. Metodología.....	14
6.1. Planificación de Proceso SCRUM.....	14
6.2. Definición de Equipo	15
6.3. Definición de los objetivos de las iteraciones (Sprint Backlog).....	15
6.4. Planeación de las iteraciones (Sprints)	17
Capítulo 7. Proceso de Software.....	18
7.1 Requerimientos Funcionales y No Funcionales.....	18
7.1.1 Requerimientos Funcionales.....	18
7.1.2 Requerimientos No Funcionales	33
7.2 Diseño y Arquitectura.....	35
7.2.1 Diagrama de Casos de Usos.....	35
7.2.2 Diagrama de Clases.....	36
7.2.3 Diagrama de Secuencia.....	37
7.2.4 Diagrama de Componentes.....	39
7.2.5 Diagrama de Despliegue.....	39
7.2.6 Arquitectura de Alto Nivel.....	40
7.3 Construcción.....	43
7.4 Pruebas.....	49
7.4.1 Pruebas de Caja Negra.....	49
7.4.2 Pruebas de APIs.....	52
7.5 Instalación y Configuración.....	54
7.5.1 Despliegue de Microservicios en DOCKER.....	54
7.5.2 Despliegue de Aplicativo WEB en APACHE TOMCAT 9.0.34.....	55
Capítulo 8. Conclusión.....	56



Capítulo 9. Referencias.....	57
Capítulo 10. Anexos.....	58...
10.1 Metodología de Desarrollo Seguro SDL.....	58 ⁱⁱⁱ
10.1.1 Etapa de Formación.....	58
10.1.2 Etapa de Requerimientos.....	58
10.1.3 Etapa de Diseño.....	59
10.1.4 Etapa de Implementación.....	60
10.1.5 Etapa de Comprobación.....	61
10.1.6 Etapa de Release.....	65

**Lista de tablas**

Tabla 1. Requerimiento Funcional Registrar Cuenta.....	18
Tabla 2. Requerimiento Funcional Iniciar Sesión.....	19
Tabla 3. Requerimiento Funcional Consulta menú de opciones Entrenador.....	20
Tabla 4. Requerimiento Funcional Consultar menú de opciones Cliente.....	20
Tabla 5. Requerimiento Funcional Actualizar datos de usuario.....	21
Tabla 6. Requerimiento Funcional Administrar Grupo de Entrenamiento.....	22
Tabla 7. Requerimiento Funcional Crear Grupo de Entrenamiento.....	23
Tabla 8. Requerimiento Funcional Editar Grupo de Entrenamiento.....	24
Tabla 9. Requerimiento Funcional Administrar Sesiones de Entrenamiento.....	25
Tabla 10. Requerimiento Funcional Ver Detalle de Grupo.....	25
Tabla 11. Requerimiento Funcional Administrar Miembros.....	26
Tabla 12. Requerimiento Funcional Visualizar miembros de grupo.....	27
Tabla 13. Requerimiento Funcional Inactivar miembro.....	27
<i>Tabla 14. Requerimiento Funcional Calificar Miembros.....</i>	<i>28</i>
Tabla 15. Requerimiento Funcional Validar Solicitudes de Inscripción.....	29
Tabla 16. Requerimiento Funcional Buscar Grupo de Entrenamiento.....	30
Tabla 17. Requerimiento Funcional Ver Detalle de Grupo.....	30
Tabla 18. Requerimiento Funcional Realizar Solicitud de Inscripción.....	31
Tabla 19. Requerimiento Funcional Consultar Mis Grupos.....	32
<i>Tabla 20. Requerimiento Cancelar Inscripción.....</i>	<i>33</i>
Tabla 21. Requerimiento Funcional Calificar Entrenador.....	33
Tabla 22. Requerimiento No Funcional Ambiente Móvil.....	34
Tabla 23. Requerimiento No Funcional Versión Lenguaje Aplicación.....	34
Tabla 24. Requerimiento No Funcional Encriptación de Contraseña.....	34
Tabla 25. Requerimiento No Funcional Navegador Web.....	34
Tabla 26. Requerimiento No Funcional de Inicio de Sesión.....	34
Tabla 27. Requerimiento No Funcional Desarrollo de Front web en Angular.....	35
Tabla 28. Caso de Prueba 1 Registro de Usuario.....	50
Tabla 29. Caso de Prueba 2 Crear Grupo.....	51
Tabla 30. Caso de Prueba 3 Solicitud de Inscripción.....	51
Tabla 31. Prueba de Inyección SQL en Login.....	62
Tabla 32. Prueba de XSS en formulario de registro.....	63
Tabla 33. Prueba de XSS en formulario de registro de grupos.....	64



Lista de figuras

v

Figura 1. Plataforma BookFit	6
Figura 2. Plataforma GymSync.....	7
Figura 3. Plataforma Fitpal	8
Figura 4. Plataforma Bienestapp.....	8
Figura 5. Diagrama de Casos de Usos– Fuente propia	36
Figura 6. Diagrama de Clase – Fuente propia.....	37
Figura 7. Diagramas de Secuencia– Fuente propia.....	38
Figura 8. Diagrama de Componentes – Fuente propia	39
Figura 9. Diagrama de Despliegue – Fuente propia	40
Figura 10. Esquema Arquitectura Microservicios	42
Figura 11. Esquema de Base de Datos.....	43
Figura 12. Spring Eureka y Microservicios	45
Figura 13. Aplicación WEB.....	47
Figura 14. Aplicativo Móvil	47
Figura 15. Código fuente en SonarQube.....	48
Figura 16. SonarLint en IDE.....	48
Figura 17. Pruebas de API USER-SERVICE	52
Figura 18. Pruebas de API GROUP-TRAINER-SERVICE	53
Figura 19. Pruebas de API NOTIFICATION-SERVICE	53
Figura 20. Imágenes Docker de Microservicios	54
Figura 21. Microservicios en Docker funcionando.....	55
Figura 22. WAR en Carpeta de Apache Tomcat	55
Figura 23. Iniciando apache tomcat	55



Capítulo 1.

Introducción

Son muchos los beneficios que aporta mantener una rutina diaria de ejercicios, como mejorar la forma y resistencia física, aumentar la autoestima, reducir el nivel de depresión, rebajar la tensión de estrés, etc. Algunos de los cuales se presentan como males de la vida moderna, que cuyas causas en ocasiones suele ser relacionada con la vida agitada del mundo de hoy. Como consecuencia de lo anterior, últimamente se ha presentado un crecimiento en los negocios relacionados con el bienestar y la salud.

En Colombia se ha venido impulsando el negocio de los gimnasios y entrenadores personales, “según la International Health, Racquet & Sportsclub Association, en el país operan unos 1.500 gimnasios”(«Crecimiento de los gimnasios en Colombia - Archivo Digital de Noticias de Colombia y el Mundo desde 1.990 - eltiempo.com» s. f.) .

En Santa Marta (Ciudad Ubicada en el Norte de Colombia) la construcción de una red de parques con espacios biosaludables también ha propiciado el crecimiento de este sector, impulsando a los ciudadanos a realizar más actividades físicas.

En el presente proyecto se propone el desarrollo de una aplicación que permita gestionar grupos de entrenamiento en la ciudad de Santa Marta. El objetivo principal es construir una herramienta que le permita a la ciudadanía tener canales de acceso a entrenadores personales para que los guíen en sus ejercicios.

Capítulo 2.

Marco Teórico

2.1 ¿Qué es Software?

La palabra “software” especifica el conjunto de programas informáticos que se desarrollan en el entorno de una computadora y que normalmente se definen en tres tipos diferentes:

- “Programas de control: Que inspeccionan y supervisan la ejecución de todas las trabajos y procesos que tienen lugar en la ordenadora (ejemplo: el Sistemas Operativo).
- Programas de Proceso: Son utilizados por el usuario para crear programas (ejemplo: compiladores, intérpretes, etc.).
- Programas de aplicación: Son los desarrollados por y para el usuario de la computadora para solucionar problemas específicos.” («Software» 2019)

2.2 Ingeniería del Software

Ingeniería de software es el estudio de la ciencia y las matemáticas mediante la cual la capacidad de los equipos computacionales se hace útil a personas a través de software de computación, procedimientos y documentación asociada.

La ingeniería de Software incluye la aplicación práctica del conocimiento científico en el diseño y creación de programas para computadoras y la documentación agrupada requerida para desarrollarlos, operarlos y mantenerlos. Esta define modelos de desarrollo de software para la elaboración de software de calidad.

La ingeniería de software combina:

- Métodos completos para todas las fases del desarrollo de software.
- Mejores herramientas para automatizar estos métodos.
- Bloques de construcción más potentes para la implementación del Software.
- Mejores técnicas para la garantía de calidad del software.
- Filosofía predominante para coordinación, control y gestión.

2.3 Entrenador Personal.

Es un profesional titulado en ciencias de la actividad física, fitness y el deporte, que es capaz de prescribir ejercicios, motiva y fija metas de forma individualizada y grupales, teniendo en cuenta las condiciones físicas y las metas de cada uno de sus clientes o usuario.(«Entrenador Personal, ¿Qué es? ¿Beneficios? ¿Desventajas?» 2015)

2.4 Java.

Es uno de los lenguajes de programación más usados en el mundo, principalmente por aplicaciones de cliente-servidos ya que es rápido, seguro y fiable. Este se rige por sus cinco objetivos:

1. “Debería usar el paradigma de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.”(«Java (lenguaje de programación)» 2019)



Java es un lenguaje multiplataforma que nos va a permitir construir nuestra aplicación⁴ para cualquier sistema operativo, también contiene una comunidad muy grande, por el cual se puede encontrar mucha documentación y soporte. Es gratuita y de código abierto, y también es usada por grandes empresas como Google, Netflix y Amazon

2.5 Angular (framework)

Es un framework gratuito y Open Source, para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. “Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC)”.(«Angular (framework)» 2019).

Angular tiene una serie de características que ofrecen buenos beneficios a la hora de desarrollar una aplicación, como:

- Mejorar la productividad con su arquitectura MVC.
- Menos errores de código.
- Calidad de Software.
- Opensource.

2.6 MySQL

“MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo,¹² y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.” («MySQL»)



MySQL cuenta con muchas características como: Transacciones, claves foráneas,⁵ conectividad segura, replicación, búsqueda e indexación de campos de texto, que nos van a permitir tener un mejor rendimiento en nuestra aplicación.

Capítulo 3. Estado del Arte

3.1 Resumen del estado del arte

En los últimos años las empresas dedicadas a ofrecer entrenamientos personalizados están buscando atraer y buscar nuevos clientes a través de las nuevas oportunidades que brindan las tecnologías móviles y web, por ejemplo.

BookFit Manager – Manage your Fitness Classes on the Move

“BookFit es un emprendimiento del Reino Unido que se enfoca en darles la facilidad a los entrenadores personales de manejar toda la logística de sus clases” (Cristina González Paredes, Paula Muriel Amador, y Silvana Hoyos 2016). Les permite publicar sus clases, con hora y lugar y hacerlas visibles a sus clientes para que estos las reserven y/o cancelen a su gusto. Por otro lado, el entrenador tiene acceso en tiempo real a la información de la clase que publicó, es decir, cuántas personas se han inscrito, cuántos cupos quedan etc. También el entrenador puede enviar mensajes a sus alumnos/clientes. “En cuanto al precio se puede decir que manejan planes desde niveles básicos hasta Pro: manejar 2 – 10 clases por semana □ £10 / mes (\$37.910 COP aprox.) “ («BookFit | Fitness booking app for Personal Trainers & Fitness Instructors» s. f.)

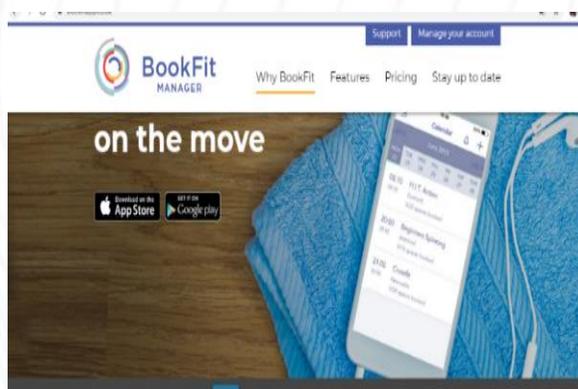


Figura 1. Plataforma BookFit

GymSync – Book your Fitness Classes with Ease, Online or on your Mobile

Se enfoca en prestar un servicio webbased y de aplicación móvil a cualquier organización que provea clases de entrenamiento físico y quiera optimizar el sistema de reserva de clases volviéndolo más eficiente, conveniente y flexible para sus miembros. Mediante la aplicación las instituciones pueden reducir costos de administración de clases, maximizar asistencia a cada clase y rastrear y reportar la asistencia de los clientes de manera más efectiva. («GymSync | Exercise Class Booking App for Gyms & Leisure Clubs» s. f.)

GymSync da 30 días de prueba gratis y de ahí en adelante cobra una mensualidad de £30 por mes (\$113.730 COP aprox.).

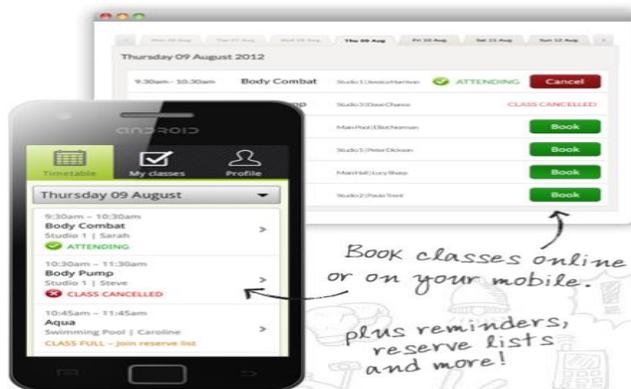


Figura 2. Plataforma GymSync

Fitpal – Miles de clases Fitness una única suscripción.

Fitpal es un sitio web que funciona en once ciudades del país, el cual permite a los usuarios acceder a varios centros de Fitness y entrenadores independientes pagando una única suscripción. “Con el pago de la suscripción mensual los usuarios pueden tomar clases ilimitadas en ese periodo de tiempo. Por otro lado, las clases de entrenadores independientes si se pueden tomar cuantas veces quieran. La suscripción mensual cuesta

\$189.000 COP y los pasos a seguir para el uso de la herramienta es suscribirse, explorar, reservar y entrenar”.(«Fitpal. | Los mejores gimnasios y centros fitness - 1 suscripción» s. f.)

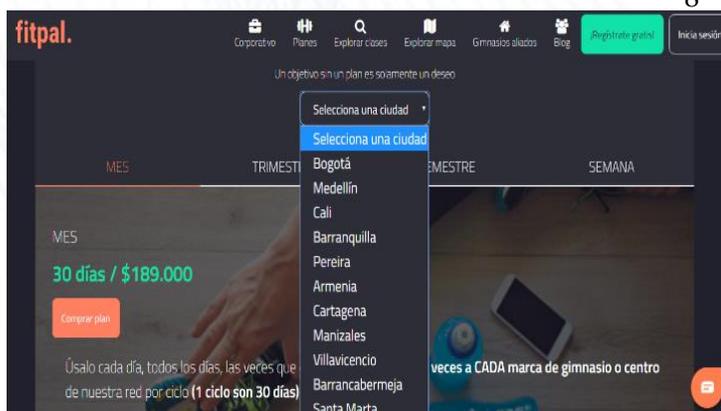


Figura 3. Plataforma Fitpal

Bienestapp

Bienestapp, la aplicación con la que podrás alcanzar una vida feliz y saludable con cientos de entrenadores personales que llegan a tu casa presionando un botón. “Con esta aplicación el entrenador personal te ejercitará durante 60 minutos como máximo, es decir que no necesitas 2 o 3 horas por día para ejercitarte sino menos de 60 minutos pues con tu entrenador personal logras tu meta más rápida y de manera profesionalmente dirigida. Tiene un costo la suscripción”. («Entrenador personal y tus clases favoritas, a un botón |

bienestapp» s. f.)



Figura 4. Plataforma Bienestapp.

3.2 Impacto.

Se va a construir un aplicativo web y móvil, que busque:

- Permitir que los entrenadores puedan acceder a nuevas tecnologías de información, con la finalidad de darse mayor visibilidad en el mercado.
- Ser una herramienta para aquellos entrenadores que desean atraer usuarios.
- Que estos tengan un mayor acceso de la información sobre los diferentes grupos de entrenamiento, logrando que estos participen en estas actividades.
- Agilidad en los procesos de inscripción y participación en actividades físicas.
- Generar acercamiento de las personas a intereses comunes.

3.3 Componente de innovación

Destacamos los siguientes puntos como componente de innovación.

- **Notificaciones:** La aplicación contara con un sistema de múltiples notificaciones para informar aspectos relevantes dentro de la misma, como creación de nuevos grupos de entrenamiento, solicitudes de inscripción, etc.
- **Calificación:** Dentro del sistema de podrá calificar tanto al entrenador como a los usuarios, esto va a permitir al entrenar tener un mejor rango en la aplicación y tener una idea de los usuarios que solicitan inscribirse a sus grupos.
- **Gratuita:** No será necesario realizar un pago para registrarse dentro de la aplicación y utilizar sus funcionalidades.
- **Aplicativo Móvil:** La aplicación tendrá una versión móvil, con funcionalidades limitadas, para permitirle al usuario utilizar las funciones más importantes desde su celular.

Capítulo 4. Formulación del Problema.

La actividad física es importante para tener un estilo de vida saludable, la Organización Mundial de Salud (OMS) “estima que esta disminuye en un 30% las complicaciones cardiovasculares de la enfermedad coronaria, el 27% de los casos de diabetes tipo 2 y el 25% de los cánceres de seno y colon, entre otras enfermedades. Además, contribuye a la reducción de obesidad”. («Menos de la mitad de los adultos colombianos hace actividad física» s. f.)

En la actualidad en Colombia la actividad física ha tomado mucha popularidad, se ha tenido un amplio crecimiento de más “460.000 afiliados en diferentes gimnasios y más el 10% de ingresos de entrenadores personales” («Crece el sector fitness en el país - Colombia» s. f.), según la Asociación Internacional de la Salud, Raqueta y Clubes Deportivos. Este interés está impulsando el negocio de los gimnasios o centros deportivos en el país.

Por otro lado, en la ciudad de Santa Marta (Ciudad que hace parte del territorio Colombiano) se ha presentado un crecimiento en los últimos 10 años en cuanto a infraestructura, esto se ve reflejado en la creación de “una red de 46 parques que están distribuidos en áreas como metropolitanos, distritales, barriales y de localidades, unos de los proyectos que ha tenido mayor impacto en la ciudadanía samaria “ (sshroot 2018). El cual su principal característica son los espacios biosaludables, que fomentan a la participación de los ciudadanos en actividades deportivas, aprovechamiento del tiempo



libre y estilos de vida saludable, a través de la utilización de equipos o máquinas para la¹¹ realización de ejercicios físicos. Estos espacios han propiciado a la participación de los ciudadanos en actividades físicas, que en muchas ocasiones no poseen el dinero o tiempo suficiente para ir a un gimnasio o contratar a un entrenador personal, siendo estos lugares de mucha ayuda para realizar sus prácticas sin costo ni tiempo estipulado. Esto ha impulsado a que entrenadores o instructores con conocimientos en educación física, entrenen o guíen a las personas a realizar sus actividades en estos lugares. Muchos de estos entrenamientos se presentan de manera informal, donde no se tiene un control del personal que asiste. De igual forma debido a la poca divulgación de estas actividades, algunos ciudadanos ejercen sus entrenamientos sin orientación alguna. Actualmente existen algunas aplicaciones que brindan servicios de entrenamiento, pero muchas veces ofrecen un servicio de entrenamiento personalizado o invitan al usuario a inscribirse en un gimnasio, adicionalmente también para acceder al uso de estas herramientas se debe pagar una cuota mensual.

Teniendo en cuenta lo anterior, existe la necesidad de construir herramientas que le permitan a las personas acceder a los servicios de entrenamiento, y a los entrenadores a ofrecer sus servicios de una forma grupal, esto de forma gratuita para que todos tengan la oportunidad de acceder a estos beneficios.



Basándonos en lo anterior podemos plantear el siguiente interrogante: ¿Es posible¹² desarrollar una aplicación que permita gestionar grupos de entrenamiento en la ciudad de Santa Marta y servir de medio para acceder a estos servicios?

Capítulo 5. Objetivos.

5.1 Objetivo General

Desarrollar una aplicación que permita gestionar grupos de entrenamientos en cualquier sitio en la ciudad de Santa Marta, además que les permita a los entrenadores llevar control de sus sesiones de entrenamiento y clientes inscriptos en sus grupos.

5.2 Objetivos Específicos

- ✓ Construir un aplicativo web que sirva como canal de acceso a servicios de entrenamientos independientes o particulares en la ciudad de Santa Marta.
- ✓ Desarrollar una aplicación móvil para Sistema Operativo Android que le permita al usuario poder utilizar las funciones más relevantes del sistema en su celular, como búsqueda de grupos, realizar y aprobar solicitudes de inscripción y consultar información de sus grupos.
- ✓ Permitir al entrenador gestionar sus grupos de entrenamiento y aceptar o rechazar solicitudes de inscripción a esos grupos.
- ✓ Aplicar controles mínimos de seguridad para proporcionarle al usuario una aplicación confiable.
- ✓ Definir una arquitectura de software robusta y acorde a los conceptos dados en la especialización.

Capítulo 6. Metodología.

Para la elaboración de este proyecto se utilizó la metodología SCRUM, que es una metodología de desarrollo ágil, su objetivo principal es controlar y planificar proyectos que presentan grandes cambios a última hora, y se basa principalmente en desarrollar un proyecto en entrega parciales.

Se definió utilizar esta metodología porque nos ofrece una serie de ventajas que nos permitirá llevar nuestro proyecto de una forma exitosa, como:

- Organización de las actividades del proyecto.
- Tendremos un control de las actividades que realizaremos cada día (Sprints)
- Reuniones para realizar el seguimiento de las tareas que ejecutamos.
- Identificar problemas que se estén presentando en el proyecto para mitigarlos en las siguientes etapas.
- Respuesta rápida a los cambios que se puedan presentar a lo largo del proyecto.

6.1. Planificación de Proceso SCRUM

La metodología SCRUM requiere una buena planificación, el desarrollo de este proyecto se dividirá en ciclos (iteraciones) de máximo 30 días, realizando reuniones cortas de 15 minutos cada día, para verificar el progreso, analizar los problemas y determinar un plan de acción para solucionarlos.

6.2. Definición de Equipo

Debido a que este proyecto será realizado por dos personas (Dayanis Pacheco y Luigi Márquez), estas se encargaran de todos los roles necesarios como Product Owner, Scrum Master y Scrum Developer.

6.3. Definición de los objetivos de las iteraciones (Sprint Backlog)

Sprint 1: En el primer Sprint se realizarán los siguientes ítems:

- Creación de la estructura de base de datos.
- Creación de Arquitectura Base.
- Creación del Módulo de inicio de Sesión.
- Creación del Módulo de Registro de Usuario
- Creación del Módulo de Actualización de Datos
- Pruebas de Módulos.
- Despliegue e instalación.

Sprint 2: En el segundo Sprint se realizarán los siguientes ítems:

- Creación de Módulo de Administración de Grupos.
- Creación del Módulo de Sesiones de Grupo.
- Pruebas de módulos.
- Despliegue y actualización.

Sprint 3: En el tercer sprint se realizarán los siguientes ítems:

- Creación del Módulo de Búsqueda de Grupos.
- Creación del Módulo de Administración de Clientes.
- Pruebas de módulos.

- Despliegue y actualización.

Sprint 4: En el cuarto sprint se realizarán los siguientes ítems:

- Creación de Modulo de Mis Grupos para el cliente.
- Creación de Funcionalidad de Calificaciones de Miembros y Entrenadores.
- Pruebas de Módulos.
- Despliegue y actualización.

Sprint 5: En el quinto sprint se realizarán los siguientes ítems:

- Creación del Módulo de Notificaciones.
- Prueba del Módulo.
- Despliegue y actualización.

Sprint 6: En el sexto Sprint, se comienza a desarrollar la parte Móvil del Proyecto, se realizan los siguientes puntos:

- Módulo de Inicio de Sesión Aplicativo Móvil.
- Módulo de Búsqueda de Grupos Aplicativo Móvil.
- Pruebas de Módulos.

Sprint 7: En el séptimo Sprint, se realizan los siguientes ítems del aplicativo móvil.

- Módulos de Administración de Clientes para el entrenador Movil.
- Módulos de mis Grupos para el cliente.
- Pruebas de Módulos.

Sprint 8: En el último sprint se realizan las actividades finales.

- Funcionalidad para calificar miembros y entrenadores Móvil.
- Pruebas de funcionalidad.

- Generación y publicación de APK de aplicativo Movil.

6.4. Planeación de las iteraciones (Sprints)

Las Iteraciones (Sprints) tendrán una duración de 20 días laborales, y se han distribuido de la siguiente manera:

Sprint 1: Desde el 3 de noviembre de 2019 hasta el 5 de enero de 2020.

Sprint 2: Desde el 6 de enero de 2020 hasta el 19 de enero de 2020.

Sprint 3: Desde el 20 de enero de 2020 hasta 2 de febrero de 2020.

Sprint 4: Desde el 3 de febrero de 2020 hasta 16 de febrero de 2020.

Sprint 5: Desde el 17 enero de 2020 hasta el 9 de abril de 2020.

Sprint 6: Desde el 10 abril de 2020 hasta el 26 de abril de 2020.

Sprint 7: Desde el 27 abril de 2020 hasta el 10 de mayo de 2020.

Sprint 8: Desde el 11 mayo de 2020 hasta el 24 de mayo de 2020.

Capítulo 7. Proceso de Software.

7.1 Requerimientos Funcionales y No Funcionales

7.1.1 Requerimientos Funcionales

Nombre	Registrar Cuenta	RF1
Descripción: El usuario deberá crear una cuenta para ingresar al sistema		
Actores: Entrenador, Cliente		
Precondiciones: Iniciar el aplicativo		
Flujo Normal		
Acción del actor	Respuesta	
<p>El usuario ingresa los siguientes datos</p> <ul style="list-style-type: none"> • Nombre y Apellidos. • Fecha nacimiento. • Celular • Usuario. • Email. • Contraseña • Confirmar contraseña • Sexo • Fecha de nacimiento • Tipo de Usuario (Entrenador, Cliente) • Leer y aceptar términos y condiciones • Presionar el botón Registrar 	Registro exitoso del usuario.	
Flujo Alternativo		
Post condiciones:		
Se realiza el registro correcto del usuario en el sistema.		

Tabla 1. Requerimiento Funcional Registrar Cuenta

Nombre	Iniciar Sesión	RF2
Descripción: El usuario deberá iniciar sesión para ingresar al sistema		
Actores: Entrenador, Cliente		
Precondiciones: Iniciar el aplicativo		
Flujo Normal		
Acción del actor	Respuesta	
El usuario ingresa los siguientes datos <ul style="list-style-type: none"> • Usuario. • Contraseña. • Presionar el botón ingresar 	Ingreso exitoso del usuario.	
Flujo Alternativo		
Post condiciones:		
Se realiza el ingreso correcto del usuario en el sistema.		

Tabla 2. *Requerimiento Funcional Iniciar Sesión*

Nombre	Consultar menú de opciones	RF3
Descripción: El usuario visualizar el menú de opciones		
Actores: Entrenador		
Precondiciones: Iniciar sesión en el aplicativo		
Flujo Normal		
Acción del actor	Respuesta	
El usuario visualiza las siguientes opciones: <ul style="list-style-type: none"> • Actualizar Datos. • Administrar grupo de entrenamiento 	El sistema visualiza las diferentes opciones que puede realizar el usuario.	

<ul style="list-style-type: none"> • Gestionar clientes. • Panel de notificaciones • Cerrar Sesión. 	
Flujo Alternativo	
Post condiciones:	
El usuario puede ingresar algunas de estas opciones.	

Tabla 3. Requerimiento Funcional Consulta menú de opciones Entrenador

Nombre	Consultar menú de opciones	RF4
Descripción: El usuario visualizar el menú de opciones		
Actores: Cliente		
Precondiciones: Iniciar sesión en el aplicativo		
Flujo Normal		
Acción del actor	Respuesta	
El usuario visualiza las siguientes opciones: <ul style="list-style-type: none"> • Actualizar Datos. • Buscar grupo de entrenamiento • Mis grupos • Panel de notificaciones • Cerrar Sesión. 	El sistema visualiza las diferentes opciones que puede realizar el usuario.	
Flujo Alternativo		
Post condiciones:		
El usuario puede ingresar algunas de estas opciones.		

Tabla 4. Requerimiento Funcional Consultar menú de opciones Cliente

Nombre	Actualizar datos	RF5
Descripción: El usuario deberá actualizar sus datos en el sistema.		
Actores: Entrenador, Cliente		
Precondiciones: Ingresar a sistema		
Flujo Normal		
Acción del actor	Respuesta	
<p>El usuario actualiza los siguientes datos</p> <ul style="list-style-type: none"> • Alto. • Nombres • Apellidos • Fecha de nacimiento • Celular • Estatura • Peso • Dirección • Perfil • Presionar el botón actualizar 	<p>Actualización exitosa de datos usuario.</p>	
Flujo Alternativo		
<ul style="list-style-type: none"> • Presionar el botón cambiar contraseña. • Solicita contraseña anterior. • Solicita contraseña nueva. • Solicita confirmar contraseña nueva. 	<p>Contraseña cambiada con éxito</p>	
Post condiciones:		
Se realiza la correcta actualización de datos del usuario en el sistema.		

Tabla 5. Requerimiento Funcional Actualizar datos de usuario

Nombre	Administrar Grupos de Entrenamiento	RF6
Descripción: El usuario deberá consultar sus grupos de entrenamiento		
Actores: Entrenador		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú Administrar Grupos de Entrenamiento		
Flujo Normal		
Acción del actor	Respuesta	
El usuario visualiza la siguiente información. <ul style="list-style-type: none"> • Botón crear grupo de entrenamiento. • Listado de grupos de entrenamientos 	Listado de Grupos creados por el usuario	
Flujo Alternativo		
Post condiciones: <ul style="list-style-type: none"> • Crear Grupo de Entrenamiento • Editar Grupo de Entrenamiento • Ver detalle de Grupo • Publicar Grupo • Administrar Sesiones de entrenamiento 		

Tabla 6.Requerimiento Funcional Administrar Grupo de Entrenamiento

Nombre	Crear Grupo de Entrenamiento	RF7
Descripción: El usuario deberá crear un grupo de entrenamiento.		
Actores: Entrenador		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú Administrar grupo, presionar el botón Crear grupo de entrenamiento.		
Flujo Normal		
Acción del actor	Respuesta	

<p>El usuario ingresa los siguientes datos</p> <ul style="list-style-type: none"> • Nombre del grupo. • Descripción del grupo. • Cantidad de cupos. • Activo • Fecha inicial • Fecha final • Descripción • Presionar el botón Guardar 	<p>Creación del grupo exitoso.</p>
<p>Flujo Alternativo</p>	
<p>Post condiciones:</p> <p>Se realiza el registro de un grupo de entrenamiento en el sistema.</p>	

Tabla 7. Requerimiento Funcional Crear Grupo de Entrenamiento

<p>Nombre</p>	<p>Editar Grupo de Entrenamiento</p>	<p>RF8</p>
<p>Descripción: El usuario deberá editar un grupo de entrenamiento.</p>		
<p>Actores: Entrenador</p>		
<p>Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú Administrar grupo, presionar la opción editar sobre un grupo de entrenamiento en el listado.</p>		
<p>Flujo Normal</p>		
<p>Acción del actor</p>	<p>Respuesta</p>	
<p>El usuario actualiza los siguientes datos</p> <ul style="list-style-type: none"> • Nombre del grupo. • Descripción del grupo. • Cantidad de cupos. • Activo • Fecha inicial • Fecha final • Descripción 	<p>Actualización del grupo exitoso.</p>	

<ul style="list-style-type: none"> • Presionar el botón Guardar 	
Flujo Alternativo	
Post condiciones:	
Se realiza el registro de un grupo de entrenamiento en el sistema.	

Tabla 8. Requerimiento Funcional Editar Grupo de Entrenamiento

Nombre	Administrar Sesiones para Grupo de Entrenamiento	RF9
Descripción: El usuario deberá crear una sesión de entrenamiento para un grupo.		
Actores: Entrenador		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú Administrar grupo, presionar la opción administrar sesión de entrenamiento sobre un grupo de en el listado.		
Flujo Normal		
Acción del actor	Respuesta	
<p>El usuario ingresa visualiza la siguiente información.</p> <ul style="list-style-type: none"> • Nombre sesión • Día inicial • Día final • Hora inicio. • Hora fin. • Activo • Dirección • Descripción • Sitio por defecto • Botón Guardar • Botón Cancelar • Lista de Sesiones del grupo con opciones de editar y eliminar. 	Ninguna	

Flujo Alternativo	
Post condiciones:	
Se realiza el registro de una sesión en un grupo de entrenamiento.	

Tabla 9. Requerimiento Funcional Administrar Sesiones de Entrenamiento

Nombre	Ver detalle de Grupo de Entrenamiento	RF10
Descripción: El usuario deberá ver el detalle de un grupo de entrenamiento.		
Actores: Entrenador		
Precondiciones: Iniciar sesión en el aplicativo, Consultar Grupos de Entrenamiento, Presionar clic en el botón ver detalle de grupo de entrenamiento.		
Flujo Normal		
Acción del actor	Respuesta	
<p>El usuario visualizar los siguientes datos</p> <ul style="list-style-type: none"> • Nombre del grupo. • Descripción del grupo. • Cantidad de cupos/Cantidad de miembros • Activo • Fecha inicial • Fecha final • Listado de Sesiones de Entrenamientos • Lista de Miembros 	Ver datos del grupo	
Flujo Alternativo		
Post condiciones:		

Tabla 10. Requerimiento Funcional Ver Detalle de Grupo

Nombre	Administrar miembros	RF11
Descripción: El usuario deberá administrar miembros.		
Actores: Entrenador		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú administrar clientes, dar clic sobre el botón administrar miembros en un grupo.		
Flujo Normal		
Acción del actor	Respuesta	
El usuario puede realizar las siguientes acciones:		
<ul style="list-style-type: none"> • Visualizar miembros de un grupo. • Ver Solicitudes de inscripción. 		
Flujo Alternativo		
Post condiciones:		

Tabla 11.Requerimiento Funcional Administrar Miembros

Nombre	Visualizar miembros de grupo	RF12
Descripción: El usuario deberá visualizar los miembros de un grupo.		
Actores: Entrenador		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú administrar clientes, presionar sobre la opción visualizar miembros de un grupo		
Flujo Normal		
Acción del actor	Respuesta	
El usuario puede realizar las siguientes acciones:		
<ul style="list-style-type: none"> • Seleccionar Grupo de entrenamiento. 		

<ul style="list-style-type: none"> • Visualizar listado de miembros. 	
Flujo Alternativo	
Post condiciones:	

Tabla 12. Requerimiento Funcional Visualizar miembros de grupo

Nombre	Inactivar miembros de grupo	RF13
Descripción: El usuario podrá inactivar los miembros de un grupo.		
Actores: Entrenador		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú administrar clientes, presionar sobre la opción Administrar Miembros en un grupo		
Flujo Normal		
Acción del actor	Respuesta	
<p>El usuario puede realizar las siguientes acciones:</p> <ul style="list-style-type: none"> • Seleccionar un miembro • Presionar el botón inactivar 	Inactivación del miembro exitosamente	
Flujo Alternativo		
Post condiciones:		
El miembro es inactivado del grupo		

Tabla 13. Requerimiento Funcional Inactivar miembro

Nombre	Calificar miembros de grupo	RF14
Descripción: El usuario podrá calificar los miembros de un grupo.		

Actores: Entrenador	
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú administrar clientes, presionar sobre la opción Visualizar Miembros sobre un grupo	
Flujo Normal	
Acción del actor	Respuesta
<p>El usuario puede realizar las siguientes acciones:</p> <ul style="list-style-type: none"> • Seleccionar un miembro • Calificar miembro. 	Calificación del miembro exitosamente
Flujo Alternativo	
Post condiciones:	
El miembro es calificado	

Tabla 14. Requerimiento Funcional Calificar Miembros

Nombre	Validar Solicitudes de inscripción	RF15
Descripción: El usuario podrá validar las solicitudes de usuario para inscribirse en un grupo.		
Actores: Entrenador		
Precondiciones Iniciar sesión en el aplicativo, ingresar a la opción de menú administrar clientes, presionar sobre la opción Administrar Solicitudes de inscripción sobre un grupo.		
Flujo Normal		
Acción del actor	Respuesta	
El usuario puede realizar las siguientes acciones:	Inscripción o rechazado del miembro exitoso.	

<ul style="list-style-type: none"> • Seleccionar una solicitud de inscripción. • Ver información del usuario. • Aceptar Solicitud. • Rechazar Solicitud. • Ingresar motivo de rechazo en caso de rechazar solicitud • Presionar el botón finalizar. 	
Flujo Alternativo	
Post condiciones:	
El miembro es agregar el grupo o rechazado según la acción del usuario	

Tabla 15. Requerimiento Funcional Validar Solicitudes de Inscripción

Nombre	Buscar grupo de entrenamiento	RF16
Descripción: El usuario podrá buscar grupos de entrenamiento.		
Actores: Cliente.		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú Buscar grupos de entrenamiento		
Flujo Normal		
Acción del actor	Respuesta	
El usuario puede realizar las siguientes acciones: <ul style="list-style-type: none"> • Buscar por nombre. • Buscar por categoría • Buscar por entrenador • Presionar el botón buscar 	Listado de grupos de entrenamiento.	
Flujo Alternativo		

<p>Post condiciones:</p> <ul style="list-style-type: none"> • Mostrar listado de grupos de entrenamiento • Seleccionar Grupo de Entrenamiento.

Tabla 16.Requerimiento Funcional Buscar Grupo de Entrenamiento

Nombre	Ver detalle de grupo (cliente).	RF17
Descripción: El usuario podrá ver la información detallada de un grupo de entrenamiento.		
Actores: Cliente.		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú Buscar grupos de entrenamiento, Seleccionar Grupo de Entrenamiento.		
Flujo Normal		
Acción del actor	Respuesta	
<p>El usuario puede visualizar la siguiente información:</p> <ul style="list-style-type: none"> • Nombre del Grupo • Descripción del Grupo. • Fecha inicial y final • Cantidad de cupos disponibles. • Rating entrenador. • Sitio de entrenamiento. • Listado de sesiones de entrenamiento. 		
Flujo Alternativo		
Post condiciones:		
Se realizar una solicitud de inscripción.		

Tabla 17.Requerimiento Funcional Ver Detalle de Grupo

Nombre	Realizar solicitud de inscripción.	RF18
---------------	------------------------------------	------

Descripción: El usuario podrá realizar una solicitud de inscripción a un grupo.	
Actores: Cliente.	
Precondiciones: Iniciar sesión en el aplicativo, Buscar Grupo de Entrenamiento, Ver detalle.	
Flujo Normal	
Acción del actor	Respuesta
<p>El usuario puede realizar las siguientes acciones:</p> <ul style="list-style-type: none"> • Ingresar motivo de solicitud de inscripción. • Presionar el botón solicitar inscripción. 	Solicitud realizara exitosamente.
Flujo Alternativo	
Post condiciones:	
Se realizar una solicitud de inscripción para que el entrenador la valide.	

Tabla 18.Requerimiento Funcional Realizar Solicitud de Inscripción

Nombre	Consular Mis Grupos de Entrenamiento.	RF19
Descripción: El usuario podrá consulta sus grupos de entrenamiento		
Actores: Cliente.		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de Mis Grupos		
Flujo Normal		
Acción del actor	Respuesta	
El usuario visualiza la siguiente información:		

<ul style="list-style-type: none"> • Ver listado de grupo de entrenamiento en los cuales es miembros. • Opción cancelar inscripción. • Opción calificar entrenador. 	
Flujo Alternativo	
Post condiciones:	
<ul style="list-style-type: none"> • Ver información de un grupo de entrenamiento • Cancelar Inscripción • Calificar Entrenador 	

Tabla 19. Requerimiento Funcional Consultar Mis Grupos

Nombre	Cancelar Inscripción.	RF20
Descripción: El usuario podrá cancelar su inscripción en un grupo de entrenamiento		
Actores: Cliente.		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú mis Grupos		
Flujo Normal		
Acción del actor	Respuesta	
<p>El usuario puede realizar las siguientes acciones:</p> <ul style="list-style-type: none"> • Presionar la opción cancelar inscripción sobre un grupo de entrenamiento. • Presionar el botón Aceptar. 	Cancelación de Inscripción Realizada correctamente.	
Flujo Alternativo		

Post condiciones:	
El usuario es inactivado permanentemente del grupo de entrenamiento y se envía una notificación al entrenador informándolo.	

Tabla 20. Requerimiento Cancelar Inscripción

Nombre	Calificar Entrenador de Grupo.	RF21
Descripción: El usuario podrá calificar a un entrenador de un grupo de entrenamiento.		
Actores: Cliente.		
Precondiciones: Iniciar sesión en el aplicativo, ingresar a la opción de menú mis Grupos, Seleccionar Grupo, presionar sobre la opción calificar entrenador		
Flujo Normal		
Acción del actor	Respuesta	
El usuario puede realizar las siguientes acciones: <ul style="list-style-type: none"> • Ingresar calificación entrenador. • Presionar el botón Finalizar. 	Calificación del entrenador correctamente.	
Flujo Alternativo		
Post condiciones:		
El entrenador es calificado.		

Tabla 21. Requerimiento Funcional Calificar Entrenador

7.1.2 Requerimientos No Funcionales

Nombre	Ambiente Móvil	RNF1
Tipo	Necesario	
Descripción	Aplicación Móvil.	

Criterios de aceptación	La aplicación debe ejecutarse en celulares con sistema operativo Superior a Android 9.
--------------------------------	--

Tabla 22. Requerimiento No Funcional Ambiente Móvil

Nombre	Versión lenguaje aplicación.	RNF2
Tipo	Necesario	
Descripción	Aplicación.	
Criterios de aceptación	La aplicación debe ejecutarse sobre Java versión 8	

Tabla 23. Requerimiento No Funcional Versión Lenguaje Aplicación

Nombre	Encriptación de Contraseña.	RNF3
Tipo	Necesario	
Descripción	Base de Datos.	
Criterios de aceptación	Se debe realizar guardado de contraseña con encriptación SHA256 en la Base de Datos	

Tabla 24. Requerimiento No Funcional Encriptación de Contraseña.

Nombre	Navegador Web.	RNF4
Tipo	Necesario	
Descripción	Aplicación web.	
Criterios de aceptación	Debe permitir el acceso a los diferentes navegadores web actuales.	

Tabla 25. Requerimiento No Funcional Navegador Web.

Nombre	Log de inicio de sesión	RNF5
Tipo	Necesario	
Descripción	Aplicación.	
Criterios de aceptación	Se tendrá un registro de cuantas veces ingreso el usuario y si fue exitoso o fallido el ingreso.	

Tabla 26. Requerimiento No Funcional de Inicio de Sesión

Nombre	Desarrollo de Front web en Angular	RNF6
Tipo	Necesario	

Descripción	Aplicación web.
Criterios de aceptación	Se va a desarrollar el front mediante el framework de Angular 8.

Tabla 27. Requerimiento No Funcional Desarrollo de Front web en Angular

7.2 Diseño y Arquitectura

7.2.1 Diagrama de Casos de Usos

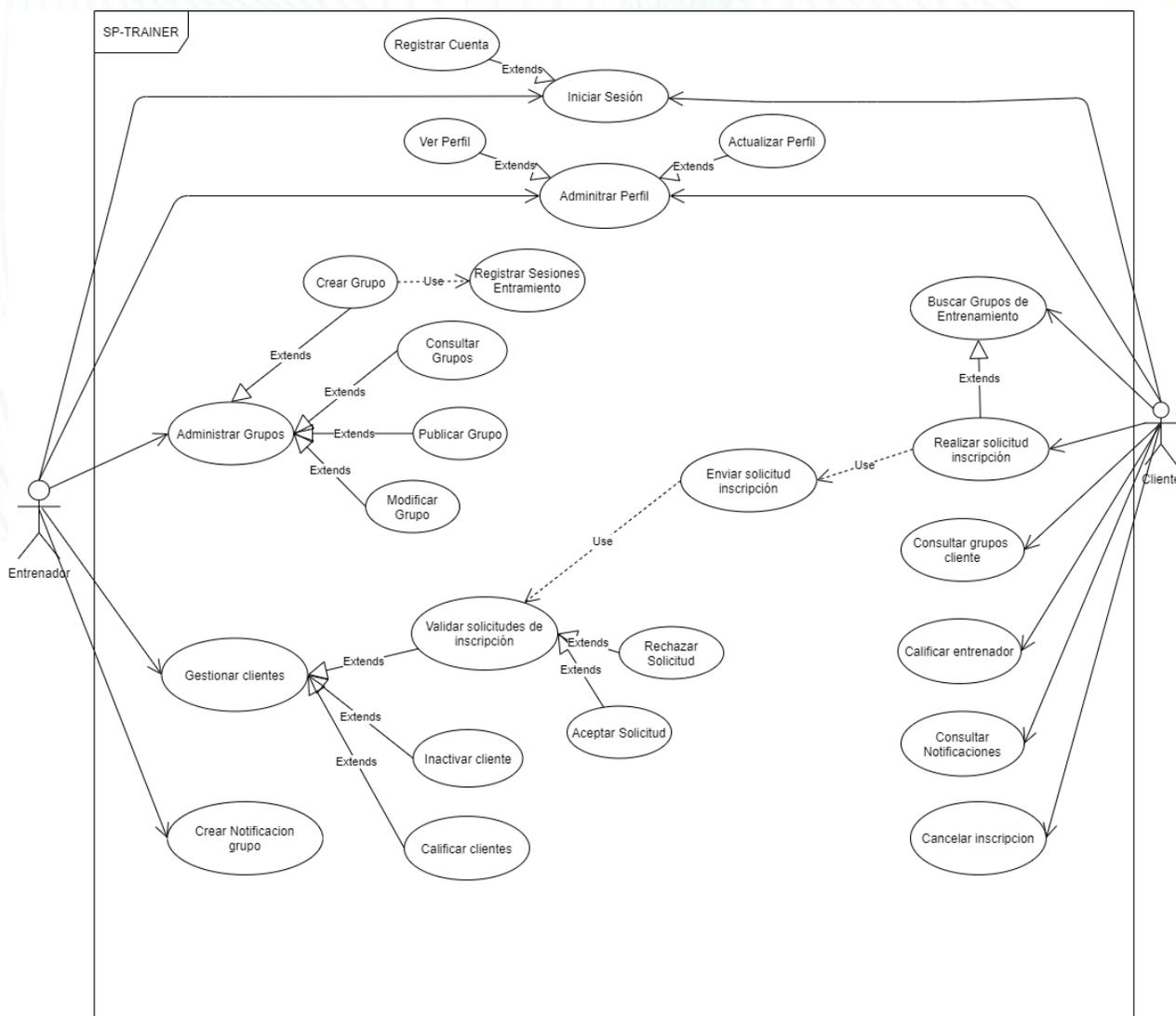


Figura 5. Diagrama de Casos de Usos– Fuente propia

En la figura 5 se muestra el Diagrama de Casos de Uso arquitecturalmente relevante, donde se indica como el usuario interactúa con el sistema, en este se puede ver principalmente las operaciones que puede realizar el entrenador y el cliente, como administrar grupo y buscar grupo, entre otras, se emplean extensiones e inclusiones en los casos pertinentes.

7.2.2 Diagrama de Clases

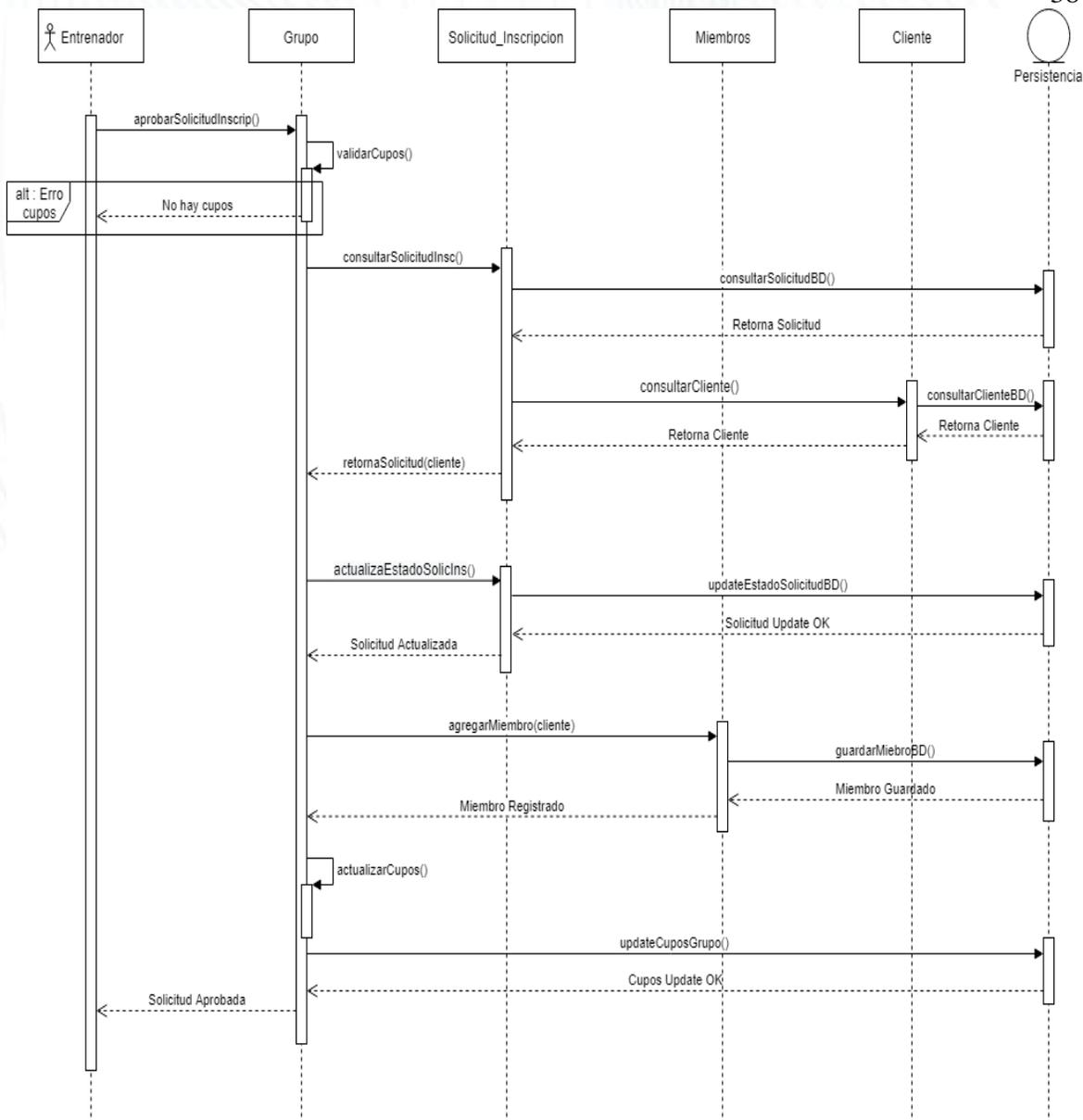


Figura 7. Diagramas de Secuencia– Fuente propia

En la figura 7 se muestra el Diagrama de Secuencia más relevante, donde se representa un proceso en particular, la aprobación de una solicitud de inscripción, en este se muestra la secuencia de dicho proceso, como inicia desde el entrenador, de valida cupos y se agrega el miembro al grupo de entrenamiento.

7.2.4 Diagrama de Componentes

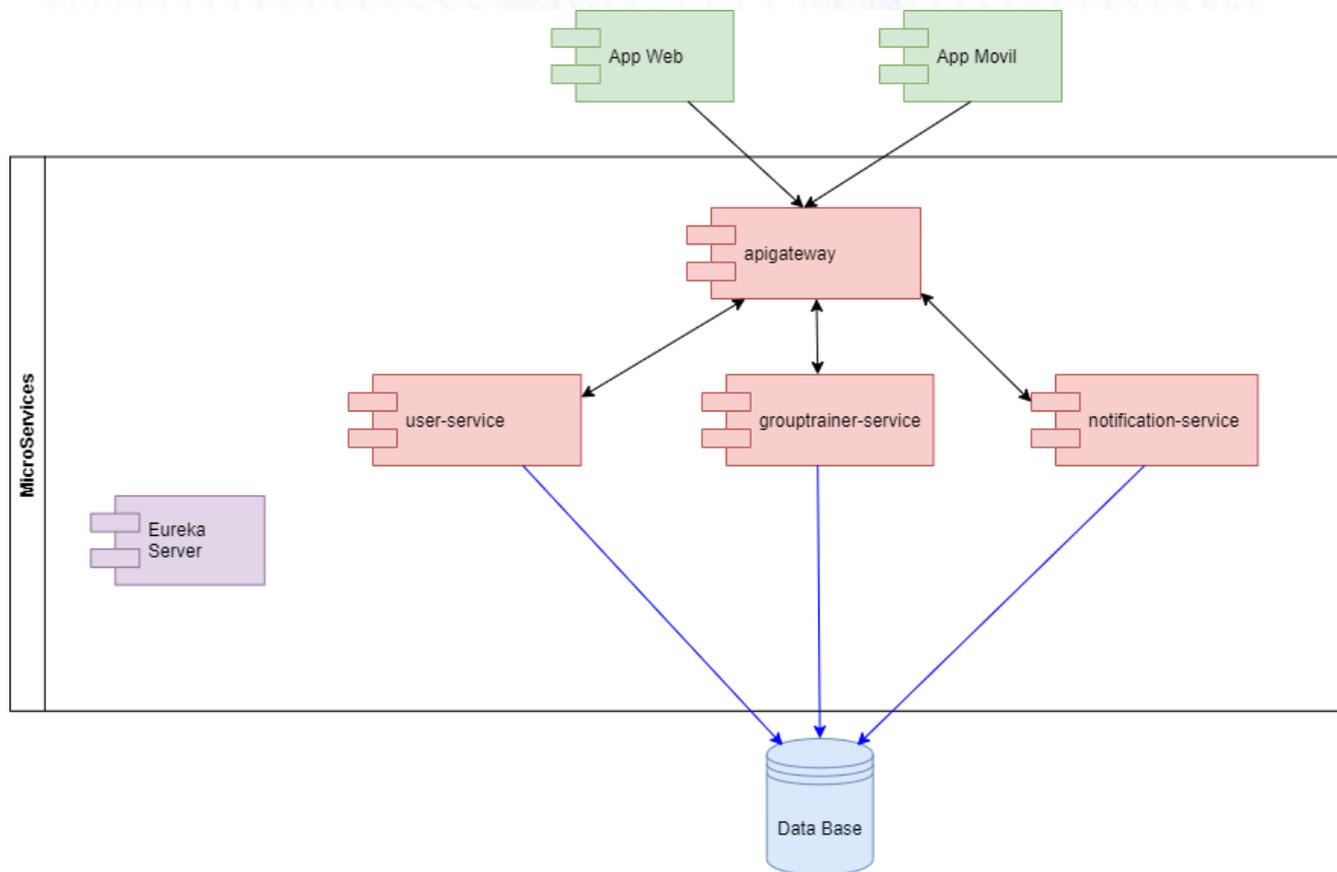


Figura 8. Diagrama de Componentes – Fuente propia

En la figura 8 se muestra el Diagrama de Componentes, donde se representa los componentes que hacen parte del sistema, se logra observar los microservicios y la forma en que interactúan

7.2.5 Diagrama de Despliegue

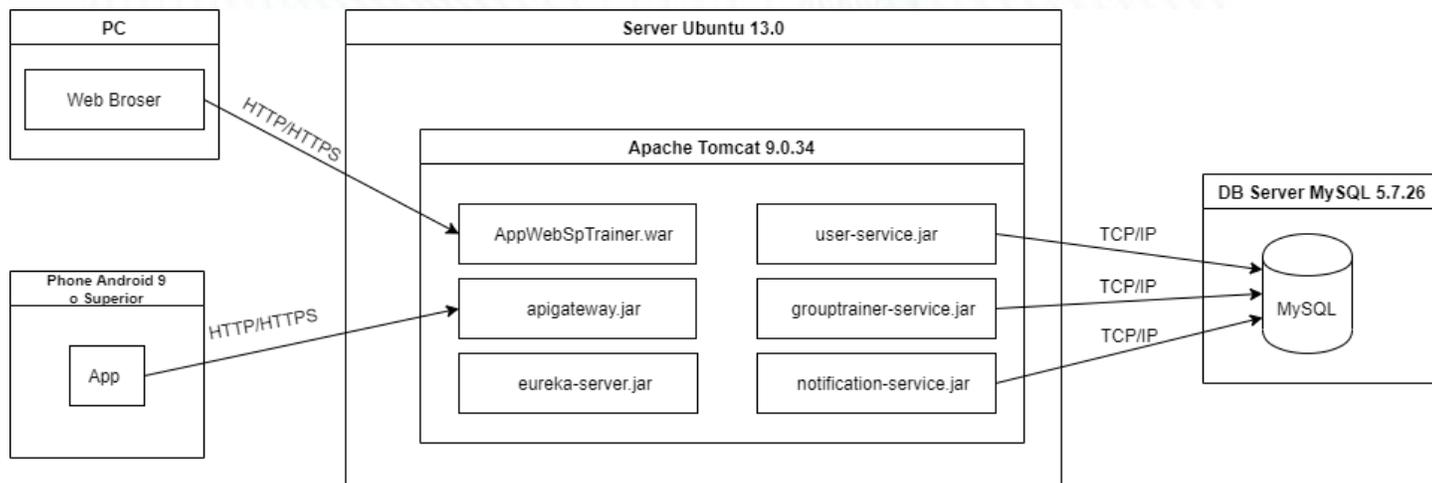


Figura 9. Diagrama de Despliegue – Fuente propia

En la figura 9 se muestra el Diagrama de Despliegue, donde se evidencia el servidor que se usara, el servidor de aplicaciones, protocolos de comunicación y artefactos que deben ser desplegados.

7.2.6 Arquitectura de Alto Nivel.

Para el desarrollo de este proyecto se utilizó el patrón de arquitectura de Microservicios, este patrón consiste en construir una aplicación como un conjunto de múltiples unidades separadas (componentes de servicio) que se pueden desarrollar, probar y desplegar de manera independiente uno de otro, donde normalmente se comunican entre ellos por mecanismos ligeros (API HTTP) y cada uno de estos servicios contiene una funcionalidad específica de la aplicación.

Se definió utilizar este patrón porque nos ofrece una serie de ventajas a nivel de arquitectura del sistema como:

- Servicios pequeños e independientes (principio de responsabilidad única).

- Unidades de despliegue pequeñas.
- Multitecnología: Cada servicio se podría realizar con una tecnología diferente.
- Agilidad: Rápida respuesta al cambio debido a que están separados y desacoplados.
- Escalabilidad: Debido a que son servicios que se encuentran separados se pueden escalar individualmente permitiendo tener más precisión.

Para su implementación se tuvieron en cuenta 3 microservicios principales y un api Gateway:

- **USER-SERVICE:** Servicio encargado de las funcionalidades de usuario, creación, modificación y logeo.
- **GROUP-TRAINER-SERVICE:** Servicio encargado de las funcionalidades de los grupos de entrenamiento, creación, modificación, sesiones de entrenamiento, búsqueda de grupos, registro de miembros.
- **NOTIFICATION-SERVICE:** Servicio encargado de las funcionalidades de notificaciones, guardar, consultar, etc.

A continuación, se muestra un esquema de la arquitectura definida con las tecnologías que fueron utilizadas para su implementación y que son detalladas en la sección de Construcción.

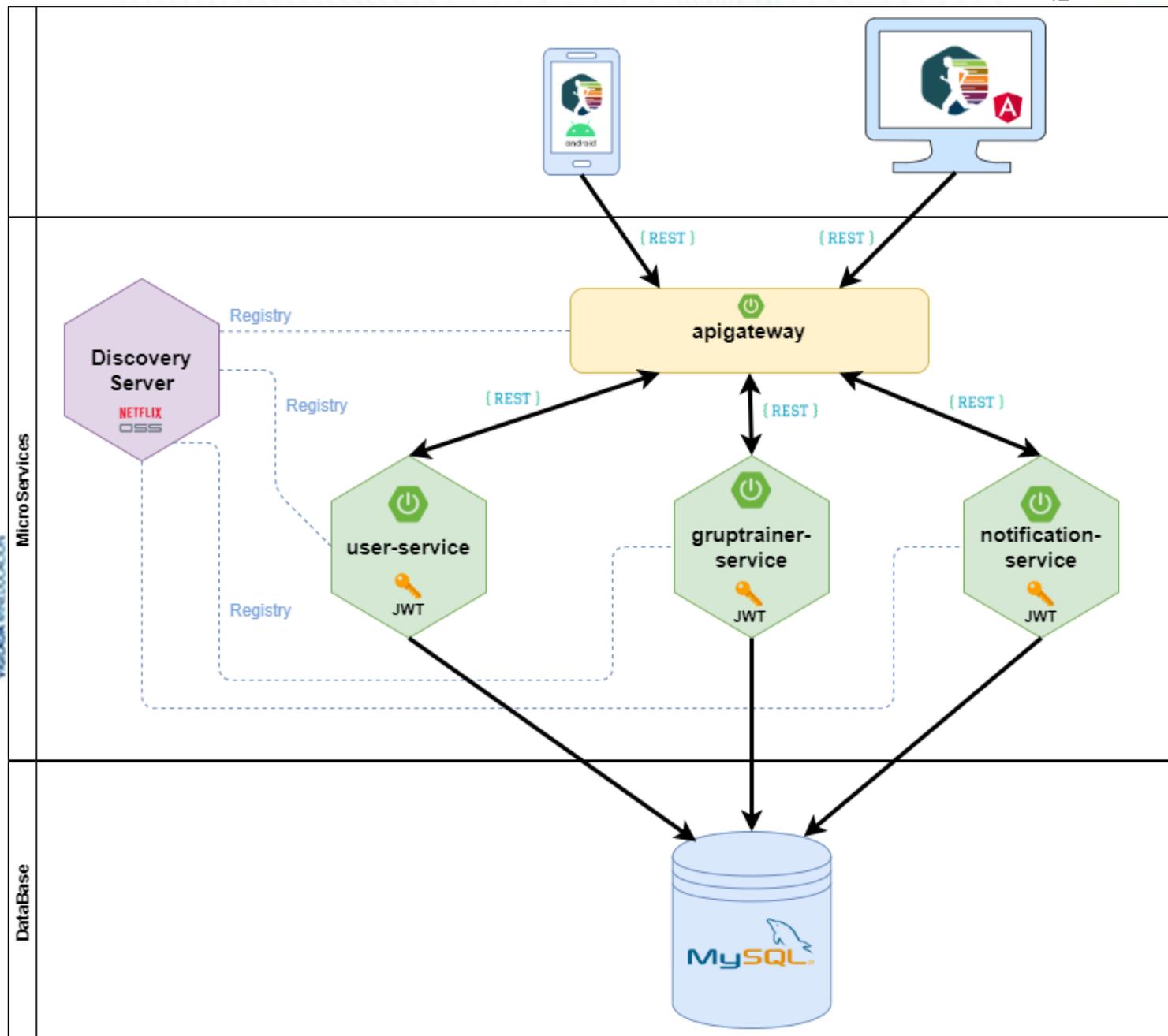


Figura 10. Esquema Arquitectura Microservicios

7.3 Construcción.

Base de Datos.

El motor de base de datos empleado para el almacenamiento de la información del software fue MYSQL versión 5.7.26, se crearon 10 tablas para el registro de toda la información necesaria en la base de datos, a continuación, se muestra un esquema de la base de datos.

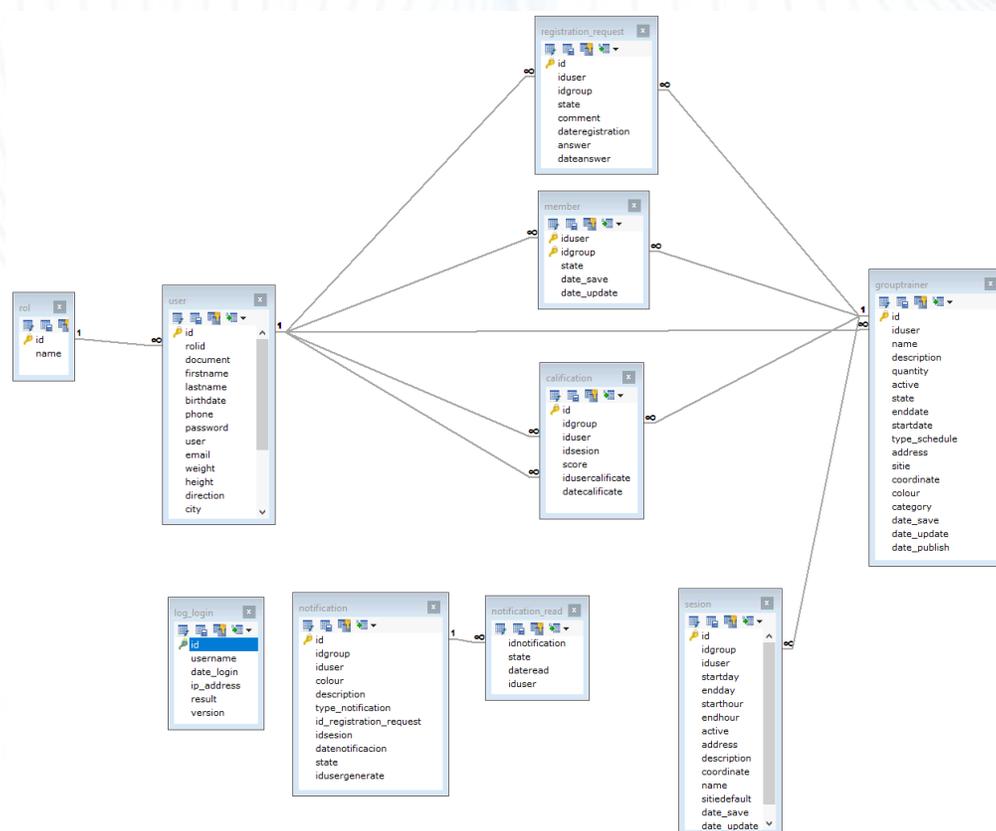


Figura 11. Esquema de Base de Datos

Microservicios.

Como se mencionó en la sección de Arquitectura de Alto Nivel, se crearon tres microservicios, USER-SERVICE, GROUP-TRAINER-SERVICE y NOTIFICATION-

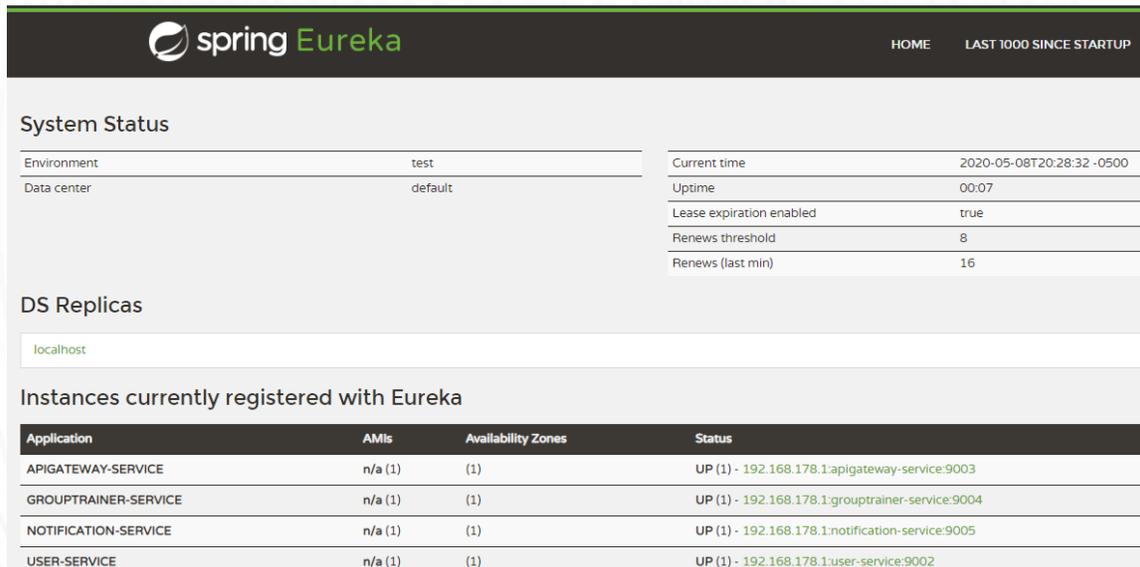


SERVICE, que son descritos en dicha sección, de igual forma se creó un API-⁴⁴ GATEWAY para permitir la comunicación entre los microservicios, la aplicación web y móvil.

Para la construcción de la arquitectura de microservicios definimos un modelo de implementación para cada uno de los componentes. Para ello hicimos uso del stack tecnológico de Spring Cloud y Netflix OSS:

- **Lenguaje de Programación:** JAVA versión 8.
- **Microservicios:** Son aplicaciones Spring Boot con controladores Spring MVC.
- **SwaggerUI:** Utilizamos Swagger para documentar y definir nuestro API.
- **Config Server:** Microservicio basado en Spring Cloud Config.
- **Git:** Se utilizó como repositorio.
- **Registry / Discovery Service:** Microservicio basado en Eureka de Netflix OSS.
- **Load Balancer:** Utilizamos Ribbon de Netflix OSS que ya viene integrado en REST-template de Spring y Web Client.
- **Json Web Token:** Como seguridad, cada microservicio tiene autenticación jwt para que se puede acceder a él y consumir sus métodos.

Spring Cloud Netflix Eureka es una librería donde principalmente se tiene un servicio (Discovery Server) que se comporta como servidor y se encarga de registrar y localizar todos los microservicios existentes, ofreciendo datos relevantes de cada uno, como estado, instancias, etc. Adicionalmente nos permite realizar balanceo de carga y se puede integrar con spring boot.



The screenshot shows the Spring Eureka dashboard. At the top, there is a navigation bar with the Spring Eureka logo and links for 'HOME' and 'LAST 1000 SINCE STARTUP'. Below this is the 'System Status' section, which is divided into two columns of key-value pairs. The first column shows 'Environment: test' and 'Data center: default'. The second column shows 'Current time: 2020-05-08T20:28:32 -0500', 'Uptime: 00:07', 'Lease expiration enabled: true', 'Renews threshold: 8', and 'Renews (last min): 16'. Below the system status is the 'DS Replicas' section, which shows 'localhost'. The main part of the dashboard is 'Instances currently registered with Eureka', which contains a table with the following data:

Application	AMIs	Availability Zones	Status
APIGATEWAY-SERVICE	n/a (1)	(1)	UP (1) - 192.168.178.1.apigateway-service:9003
GROUPTRAINER-SERVICE	n/a (1)	(1)	UP (1) - 192.168.178.1.grouptrainer-service:9004
NOTIFICATION-SERVICE	n/a (1)	(1)	UP (1) - 192.168.178.1.notification-service:9005
USER-SERVICE	n/a (1)	(1)	UP (1) - 192.168.178.1.user-service:9002

Figura 12. Spring Eureka y Microservicios

Aplicación WEB.

La aplicación WEB fue desarrollada con el framework Angular 8, se utilizó la librería HttpClient para comunicarse con el API-GATEWAY. Dentro de la aplicación manejaron dos tipos de Roles, Entrenador y Cliente, a continuación, describimos brevemente las funcionalidades:

Funcionalidades para ambos roles.

- **Registro de Usuario:** Formulario de Solicitud de datos para el registro del usuario en la aplicación.
- **Inicio de Sesión:** Ingreso del usuario al sistema por medio del ingreso de los datos de usuario o correo y contraseña.
- **Actualizar Datos:** Formulario para actualizar datos de usuario.
- **Consultar Notificaciones:** Consulta de notificaciones del usuario.

Funcionalidades para el Entrenador.

- **Administrar Grupos:** Creación y modificación de grupos y sesiones de entrenamiento.
- **Administrar Clientes:** Administrar solicitudes de inscripción, consultar miembros de grupo y calificar miembros de grupos.

Funcionalidades para el Cliente:

- **Mis Grupos:** Consultar los grupos al cual se encuentra inscrito y calificar al entrenador.
- **Buscar Grupos de Entrenamiento:** Buscar grupos de entrenamiento y solicitar inscribirse si se encuentra interesado.

Se utilizó la librería de Google Maps para el geoposicionamiento del usuario y de los sitios de entrenamiento y poder establecer la cercanía entre los dos puntos.

En los formularios donde se obtiene información relevante se emplean validaciones de datos; campos vacíos, cantidad de caracteres, solo números o letras y evitar caracteres especiales, todos estos controles se implementaron para evitar vulnerabilidades. Adicional, un control de seguridad empleado fue la encriptación de la contraseña en SHA256 antes de enviar la petición, con esto logramos que la contraseña que ingresa el usuario no sea visible cuando se envían los datos al API-GATEWAY, también se encriptó los datos que se utilizan en la aplicación mientras el usuario navega (Local Storage).

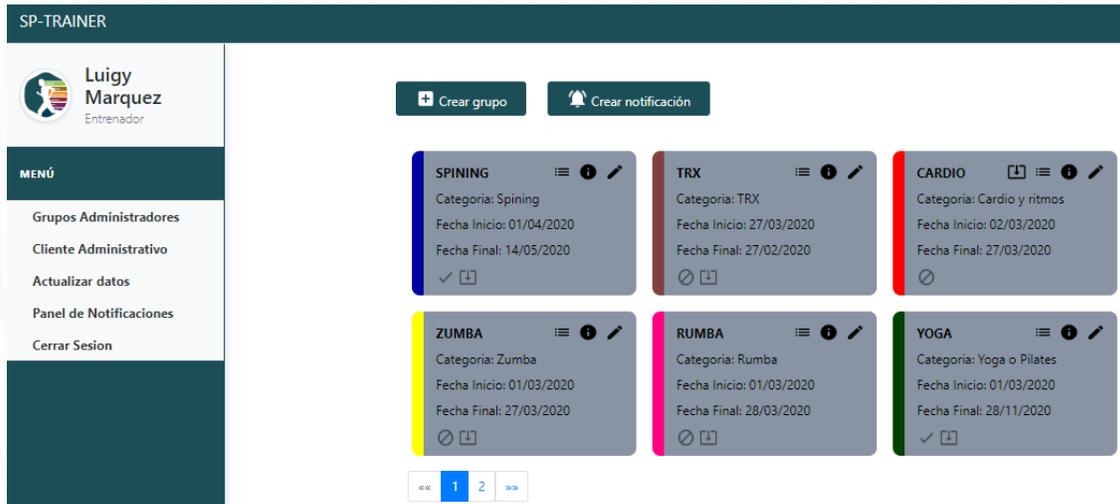


Figura 13. Aplicación WEB

Aplicación MOVÍL.

La aplicación Móvil fue desarrollada en Android JAVA versión 7, se utilizó la herramienta Android Studio. En la app se desarrollaron las mismas funcionalidades que la aplicación WEB, a excepción del módulo de Administración de Grupos.

De igual forma que en el aplicativo WEB se realizó una integración con Google Maps para obtener la ubicación del usuario y mostrarle donde se encontraban las sesiones de entrenamiento de un grupo.



Figura 14. Aplicativo Móvil

Durante el proceso de construcción se utilizaron dos herramientas para medir la calidad del código, SonarQube y SonarLint, esta última fue un componente que integramos en los IDE que utilizamos para el desarrollo del software, IntelliJ IDEA y Android Studio.

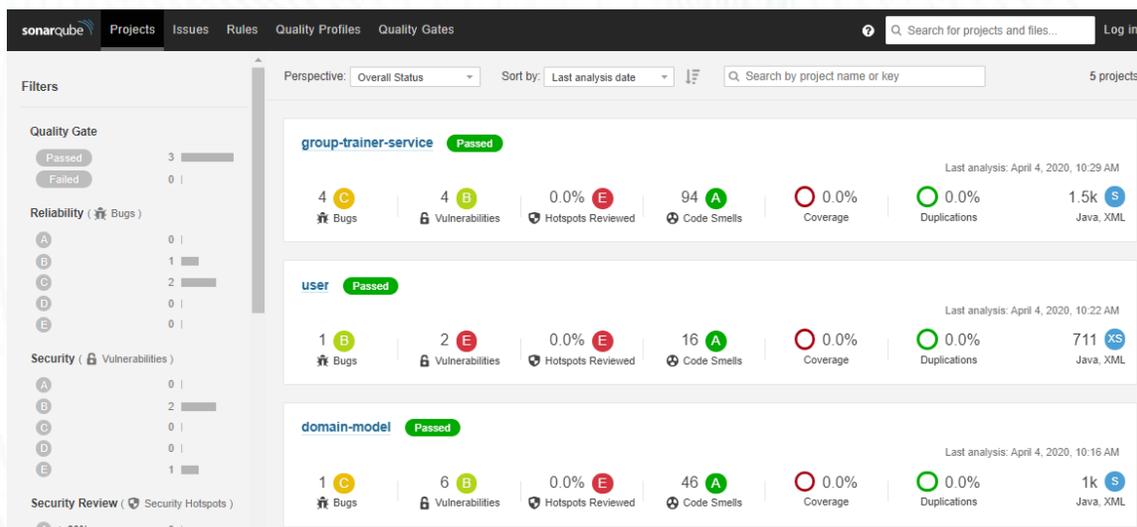


Figura 15. Código fuente en SonarQube

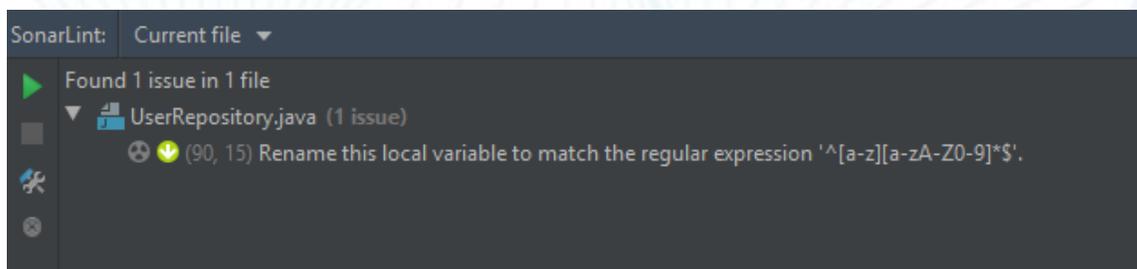


Figura 16. SonarLint en IDE.

7.4 Pruebas.

Las pruebas se realizaron al final y durante el desarrollo del sistema, por cada spring finalizado se ejecutaron pruebas de diferentes ámbitos, principalmente se ejecutaron pruebas de caja negra y pruebas para las APIs, se probaron las funcionalidades más relevantes, la herramienta utilizada para probar las API fue POSTMAN.

7.4.1 Pruebas de Caja Negra.

Caso de Prueba 1 Registro de un Usuario			
Objetivo:	Ingresar a la aplicación web y registrar un Usuario		
Pasos:	1. Ingresar a la aplicación	2. dar clic a registro	3. Llenar los datos con información errada.
	4. Dejar vacíos campos obligatorios	5. Ingresar datos correctamente	6. Guardar Usuario nuevo
Se realiza ingresa datos errados y se evidencia en rojo los datos errados			
Se valida datos únicos de correo y usuario			

<p>Se ingresa los datos ok</p>	<p style="text-align: center;">REGISTRESE Y ACCEDA A TODOS NUESTROS BENEFICIOS</p> <p>Obligatorios*</p> <p>Nombres* <input type="text" value="Dayanis Esther"/></p> <p>Apellidos* <input type="text" value="Ibanez"/></p> <p>Fecha de nacimiento* <input type="text" value="1981-05-20"/></p> <p>Celular* <input type="text" value="3153919637"/></p> <p>Usuario* <input type="text" value="luigi"/></p> <p>E-mail* <input type="text" value="dayanis.deprs@gmail.com"/></p> <p>Contraseña* <input type="password" value="*****"/></p> <p>Confirmar Contraseña* <input type="password" value="*****"/></p> <p>Sexo* <input type="text" value="Masculino"/></p> <p>Tipo de Usuario* <input type="text" value="Cliente"/></p> <p><input checked="" type="checkbox"/> Aceptar Terminos & Condiciones Leer</p> <p style="text-align: center;"> <input type="button" value="Registrarme"/> <input type="button" value="Cancelar"/> </p>
--------------------------------	---

Tabla 28. Caso de Prueba 1 Registro de Usuario

Caso de Prueba 2 Crear un Grupo de Entrenamiento			
Objetivo:	Crear un Grupo de Entrenamiento y validar los campos de entrada.		
Pasos:	1. Ingresar con usuario de entrenador	2. Dar clic a Crear Grupo	3. Presionar clic sobre el botón guardar sin llenar datos
	4. Ingresar datos correctamente y presionar clic sobre el botón guardar.		
Validación de datos faltantes	<p style="text-align: center;">Crear nuevo grupo de entrenamiento</p> <p>Nombre <input type="text" value="Nombre"/> Cantidad cupos <input type="text" value="Cupos"/></p> <p><small>Nombre es requerido</small> <small>Cantidad es requerida</small></p> <p>Color <input type="text" value=""/></p> <p><input type="checkbox"/> Activo</p> <p>Fecha inicial <input type="text" value=""/> Fecha final <input type="text" value=""/></p> <p><small>Fecha inicial es requerida</small> <small>Fecha final es requerido</small></p> <p>Categoría <input type="text" value="Seleccione"/></p> <p><small>Dia inicial es requerido</small></p> <p>Descripción <input type="text" value=""/></p> <p><small>Descripcion es requerida</small></p> <p style="text-align: center;"><input type="button" value="Guardar"/></p>		

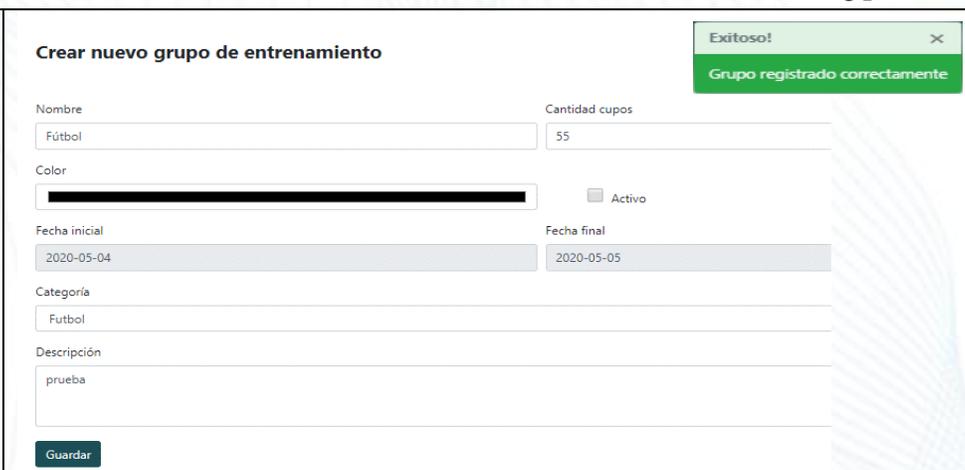
<p>Respuesta de registro correcto</p>	
---------------------------------------	--

Tabla 29. Caso de Prueba 2 Crear Grupo

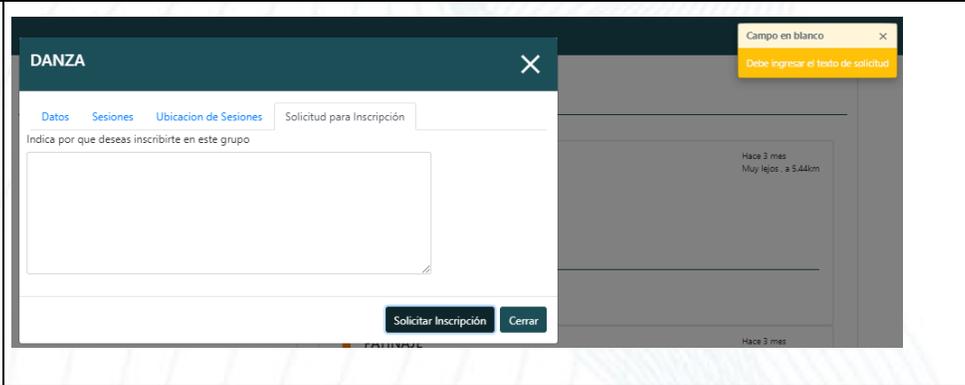
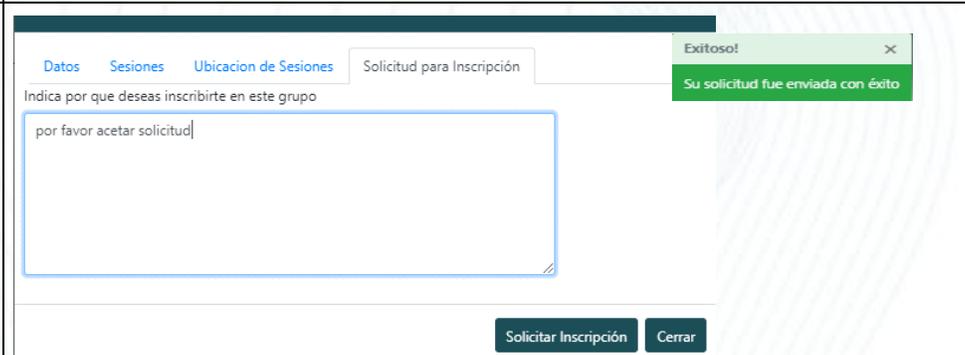
Caso de Prueba 3 Crear solicitud de inscripción			
Objetivo:	Enviar una solicitud de inscripción.		
Pasos:	1. Ingresar al módulo de Buscar Grupo	2. Buscar el grupo de inscripción	3. No ingresar los datos solicitados
	4. Ingresar datos correctamente y presionar clic sobre el botón Solicitar Inscripción.		
Validación de datos faltantes			
Respuesta de registro correcto			

Tabla 30. Caso de Prueba 3 Solicitud de Inscripción

7.4.2 Pruebas de APIs.

Las pruebas de API se ejecutaron sobre cada microservicio, a continuación, se muestran imágenes de la ejecución de las pruebas en POSTMAN de cada microservicio.

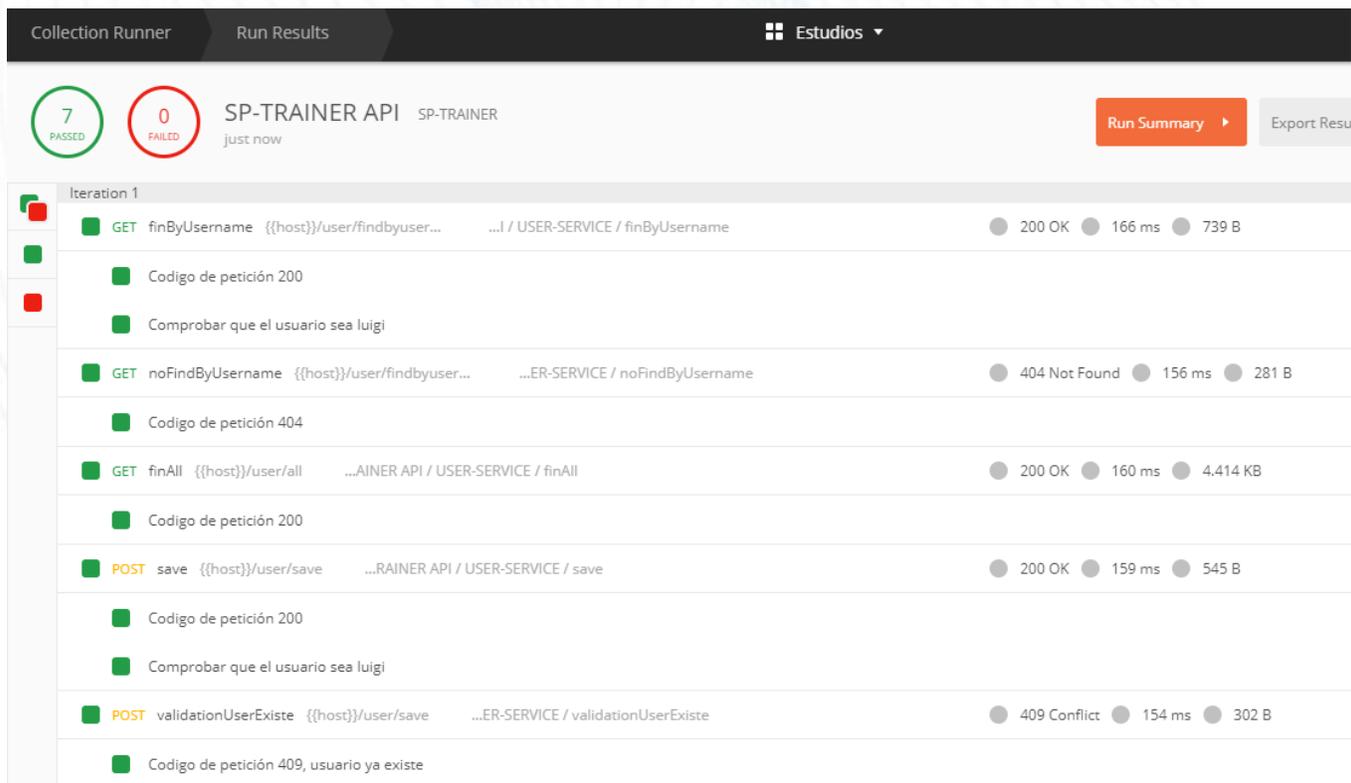


Figura 17. Pruebas de API USER-SERVICE.

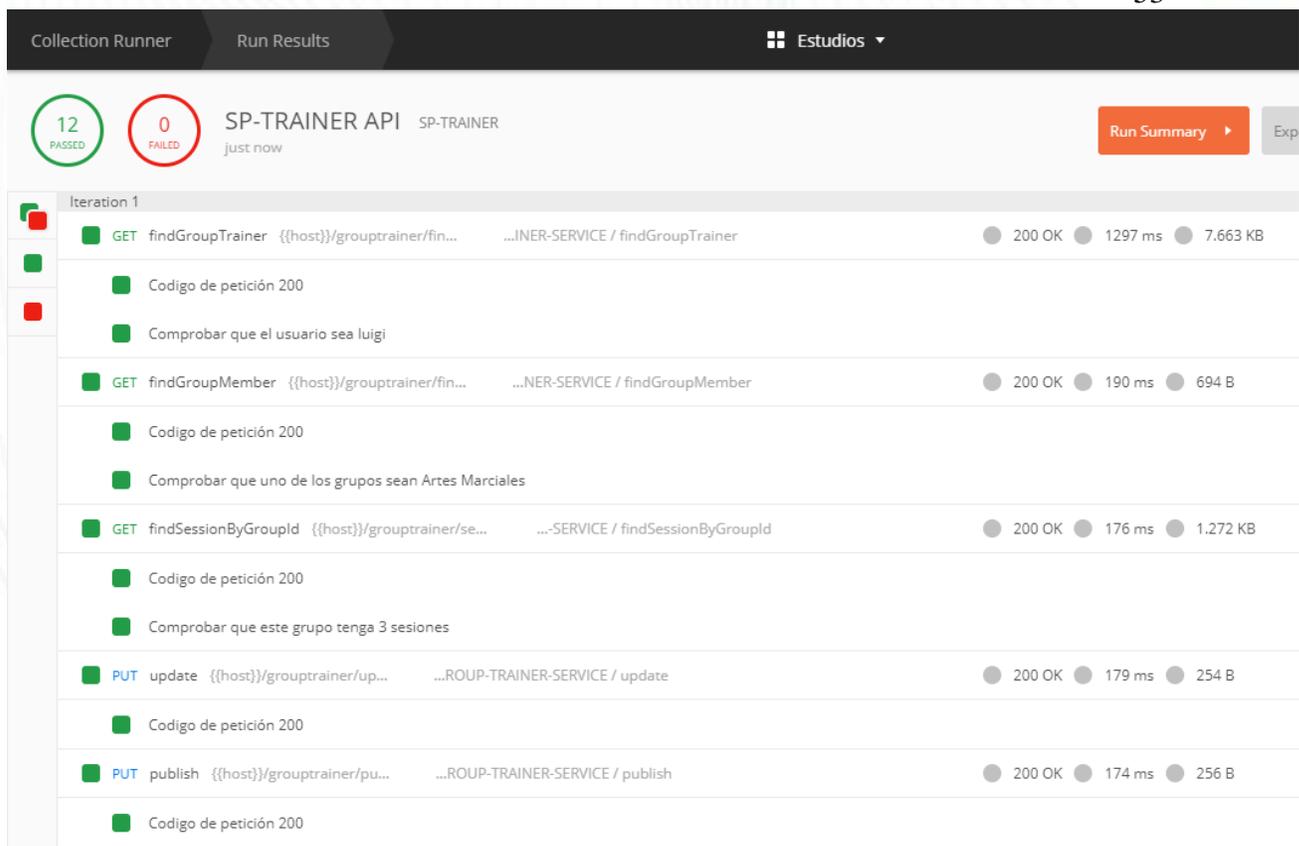


Figura 18. Pruebas de API GROUP-TRAINER-SERVICE

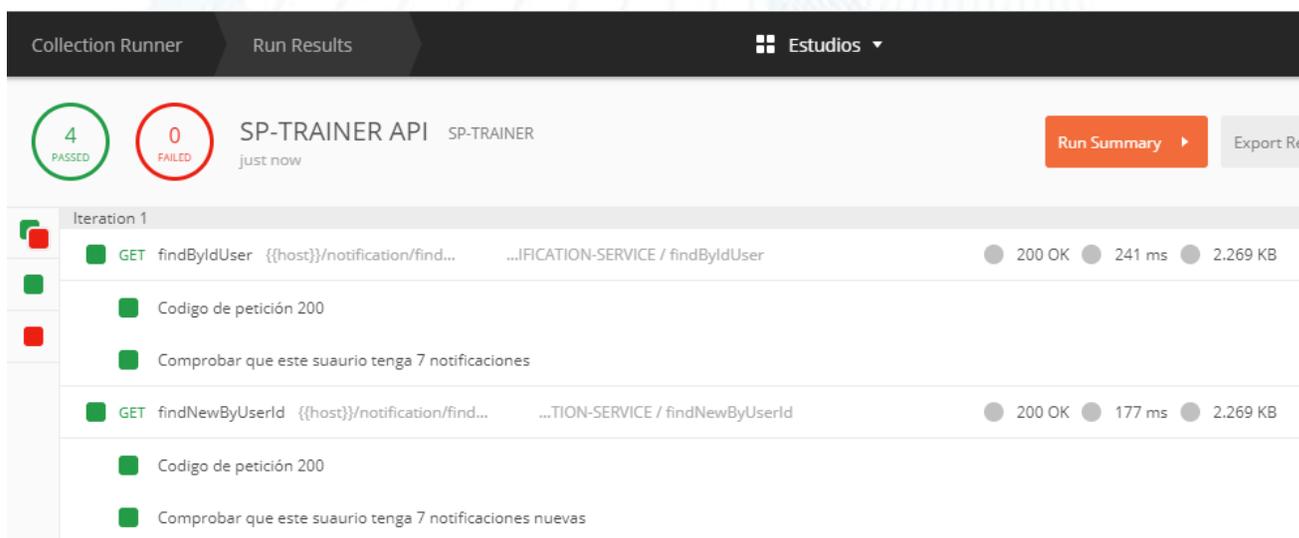


Figura 19. Pruebas de API NOTIFICATION-SERVICE

7.5 Instalación y Configuración.

El proceso de instalación y configuración de la aplicación se realiza sobre un servidor UBUNTU 13.0, es necesario que en dicho servidor se encuentren instalados JAVA versión 8, DOCKER y el servidor de aplicaciones APACHE TOMCAT 9.0.34.

7.5.1 Despliegue de Microservicios en DOCKER.

Para este proceso inicialmente se deben crear las imágenes Docker de los microservicios, para esto debemos ubicarnos dentro de la carpeta de instalación, abrir una terminal y ejecutar los siguientes comandos para crear las imágenes.

```
docker build -t dockersptrainer/discovery-server:01 .
docker build -t dockersptrainer/user-service:01 .
docker build -t dockersptrainer/group-trainer-service:01 .
docker build -t dockersptrainer/notification-service:01 .
docker build -t dockersptrainer/apigateway-service:01 .
```

```
consultor05@consultor05-Aspire-E1-47:~/Documentos/SP-TRAINER/SPTRAINER-BACKED/sptrainer/integration-service$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
dockersptrainer/apigateway-service  01         5c9108d3fbe5  9 seconds ago 695MB
dockersptrainer/notification-service 01         40f700e44f5c 27 seconds ago 699MB
dockersptrainer/group-trainer-service 01         709966316ce9 55 seconds ago 699MB
dockersptrainer/user-service          01         58b781078c24 About a minute ago 698MB
dockersptrainer/discovery-server      01         bb989f59e2d5 3 minutes ago 692MB
hardtech/mysql-store                  latest     ae6b78bedf88 17 months ago 372MB
java                                   8         d23bdf5b1b1b 3 years ago 643MB
consultor05@consultor05-Aspire-E1-47:~/Documentos/SP-TRAINER/SPTRAINER-BACKED/sptrainer/integration-service$
```

Figura 20. Imágenes Docker de Microservicios

Posteriormente ejecutamos los siguientes comandos para correr cada imagen creada e iniciar los microservicios.

```
docker network create sprainer-net
docker run -d --name discovery-server --network=sprainer-net -p 8761:8761 dockersptrainer/discovery-server:01
docker run -d --name user-service --network=sprainer-net -p 9002:9002 dockersptrainer/user-service:01
docker run -d --name group-trainer-service --network=sprainer-net -p 9004:9004 dockersptrainer/group-trainer-service:01
docker run -d --name notification-service --network=sprainer-net -p 9005:9005 dockersptrainer/notification-service:01
docker run -d --name apigateway-service --network=sprainer-net -p 9003:9003 dockersptrainer/apigateway-service:01
```

Inicialmente se debe crear la red en Docker para que cada contenedor este en la misma red, luego debe ser ejecutado el Discovery server inicialmente para que los servicios se registren, después se ejecutan los demás comandos.

```

consultor05@consultor05-Aspire-E1-47:~/Documentos/SP-TRAINER/SPTRAINER-BACKED/sptrainer$ docker ps
CONTAINER ID        IMAGE                                     COMMAND                  CREATED            STATUS              PORTS
3e258ce4b581p     dockersptrainer/apigateway-service:01  "java -Djava.securit..." About a minute ago  Up About a minute  0.0.0.0:9003->9003/tc
860bbf6cc2bdp     dockersptrainer/notification-service:01 "java -Djava.securit..." 2 minutes ago     Up 2 minutes       0.0.0.0:9005->9005/tc
1f5768fa4b14p     dockersptrainer/group-trainer-service:01 "java -Djava.securit..." 2 minutes ago     Up 2 minutes       0.0.0.0:9004->9004/tc
3aee19ed65f4p     dockersptrainer/user-service:01        "java -Djava.securit..." 2 minutes ago     Up 2 minutes       0.0.0.0:9002->9002/tc
b46b66ee122bp     dockersptrainer/discovery-server:01    "java -Djava.securit..." 7 minutes ago     Up 7 minutes       0.0.0.0:8761->8761/tc
1e490354195dp     hardtech/mysql-store                   "docker-entrypoint.s..." 14 months ago     Up 2 hours         33060/tcp, 0.0.0.0:33
05->3306/tcp      mysql-store
consultor05@consultor05-Aspire-E1-47:~/Documentos/SP-TRAINER/SPTRAINER-BACKED/sptrainer$

```

Figura 21. Microservicios en Docker funcionando.

7.5.2 Despliegue de Aplicativo WEB en APACHE TOMCAT 9.0.34.

Para este proceso colocamos el WAR de la aplicación en la ruta *apache-tomcat-*

9.0.34/webapps

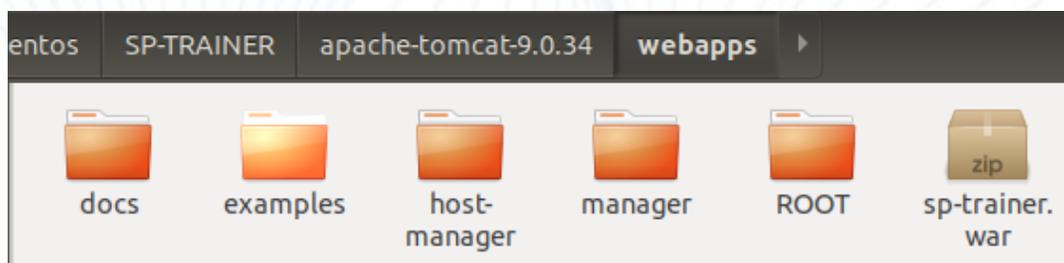


Figura 22. WAR en Carpeta de Apache Tomcat

Posteriormente iniciamos el servidor de aplicaciones ubicándonos en la carpeta

apache-tomcat-9.0.34/bin ejecutando el comando *./startup.sh*

```

root@consultor05-Aspire-E1-47: /home/consultor05/Documentos/SP-TRAINER/apache-tomcat-9.0.34/bin
Archivo Editar Ver Buscar Terminal Ayuda
root@consultor05-Aspire-E1-47:/home/consultor05/Documentos/SP-TRAINER/apache-tomcat-9.0.34/bin# ls
bootstrap.jar      ciphers.bat      configtest.bat  digest.sh        setclasspath.sh  startup.sh        tool-wrapper.sh
catalina.bat      ciphers.sh       configtest.sh   makebase.bat    shutdown.bat     tomcat-juli.jar  version.bat
catalina.sh       commons-daemon.jar daemon.sh        makebase.sh     shutdown.sh      tomcat-native.tar.gz version.sh
catalina-tasks.xml commons-daemon-native.tar.gz digest.bat      setclasspath.bat startup.bat      tool-wrapper.bat
root@consultor05-Aspire-E1-47:/home/consultor05/Documentos/SP-TRAINER/apache-tomcat-9.0.34/bin# ./startup.sh
Using CATALINA_BASE:   /home/consultor05/Documentos/SP-TRAINER/apache-tomcat-9.0.34
Using CATALINA_HOME:   /home/consultor05/Documentos/SP-TRAINER/apache-tomcat-9.0.34
Using CATALINA_TMPDIR: /home/consultor05/Documentos/SP-TRAINER/apache-tomcat-9.0.34/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-oracle
Using CLASSPATH:       /home/consultor05/Documentos/SP-TRAINER/apache-tomcat-9.0.34/bin/bootstrap.jar:/home/consultor05/Documentos/SP-TRAINER/apache-tomcat-9.0.34/bin/tomcat-juli.jar
Tomcat started.
root@consultor05-Aspire-E1-47:/home/consultor05/Documentos/SP-TRAINER/apache-tomcat-9.0.34/bin#

```

Figura 23. Iniciando apache tomcat



Con este proceso la aplicación queda instalada y funcionando correctamente en el⁵⁶ servidor.

Capítulo 8. Conclusión.

Durante el proceso de desarrollo de este proyecto se lograron adquirir nuevas habilidades en cuanto al proceso de construcción de un software. No solo se logró definir e implementar la arquitectura de un sistema, sino también a utilizar otras herramientas que nos ayudaron a construir un software de calidad y tener un mejor control a la hora de desarrollar el mismo, como lo fueron GIT, SONARQUBE y POSTMAN.

El diseño del diagrama de casos de uso y de secuencia, junto con los requerimientos, fueron fundamentales para el desarrollo del software, ya que nos permitieron definir todas aquellas funcionalidades que debería tener el sistema para cumplir con su objetivo.

A lo largo del desarrollo de este proyecto, una de las dificultades que se nos presentó, fue el aprendizaje de todas esas tecnologías que nos permitieron construir el software deseado, como lo fue Docker, Spring Cloud y Eureka, Angular y Android Java, pero gracias a la constante investigación continua se lograron aprender e implementar con éxito en nuestro proyecto.

Todo lo aprendido en este proyecto y en la especialización nos van a permitir ser mejores profesionales en nuestra carrera y con el conocimiento adquirido, a contribuir al crecimiento de una organización o crear nuevos proyectos de emprendimiento de calidad.

Capítulo 9. Referencias.

- «Angular (framework)». 2019. En *Wikipedia, la enciclopedia libre*.
[https://es.wikipedia.org/w/index.php?title=Angular_\(framework\)&oldid=118314550](https://es.wikipedia.org/w/index.php?title=Angular_(framework)&oldid=118314550).
- «BookFit | Fitness booking app for Personal Trainers & Fitness Instructors». s. f.
Accedido 13 de septiembre de 2019. <https://www.bookfitapp.co.uk/>.
- «Crece el sector fitness en el país - Colombia». s. f. Accedido 14 de septiembre de 2019.
<http://gerente.com/co/crece-sector-fitness-pais/>.
- «Crecimiento de los gimnasios en Colombia - Archivo Digital de Noticias de Colombia y el Mundo desde 1.990 - eltiempo.com». s. f. Accedido 14 de septiembre de 2019.
<https://www.eltiempo.com/archivo/documento/CMS-16125076>.
- «Entrenador Personal, ¿Qué es? ¿Beneficios? ¿Desventajas?». 2015. CuidatePlus. 23 de abril de 2015. <https://cuidateplus.marca.com/ejercicio-fisico/diccionario/entrenador-personal.html>.
- «Entrenador personal y tu clases favoritas, a un botón | bienestapp». s. f. Accedido 13 de septiembre de 2019. <https://www.bienestapp.com/>.
- «Fitpal.| Los mejores gimnasios y centros fitness - 1 suscripción». s. f. Accedido 13 de septiembre de 2019. <https://www.fitpal.co/>.
- «GymSync | Exercise Class Booking App for Gyms & Leisure Clubs». s. f. Accedido 13 de septiembre de 2019. <https://getgymsync.co.uk/why-gymsync>.
- «Java (lenguaje de programación)». 2019. En *Wikipedia, la enciclopedia libre*.
[https://es.wikipedia.org/w/index.php?title=Java_\(lenguaje_de_programaci%C3%B3n\)&oldid=119249694](https://es.wikipedia.org/w/index.php?title=Java_(lenguaje_de_programaci%C3%B3n)&oldid=119249694).
- «Menos de la mitad de los adultos colombianos hace actividad física». s. f. Accedido 14 de septiembre de 2019. <https://www.minsalud.gov.co/Paginas/Menos-de-la-mitad-de-los-adultos-colombianos-hace-actividad-f%C3%ADsica-.aspx>.
- «[SCRUM] Ser Scrum Master Certified y ¡no morir en el intento! (Parte 1) – El Blog de Pame». s. f. Accedido 14 de septiembre de 2019.
<http://blog.pucp.edu.pe/blog/pamelars/2018/09/03/scrum-ser-scrum-master-certified-y-no-morir-en-el-intento-parte-1/>.
- «Software». 2019. En *Wikipedia, la enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=Software&oldid=119274738>.
- sshroot. 2018. «Red de Paruques». Text. 6 de febrero de 2018.
<https://www.santamarta.gov.co/red-de-paruques>.
- «MySQL». <https://es.wikipedia.org/wiki/MySQL>

Capítulo 10. Anexos.

10.1 Metodología de Desarrollo Seguro SDL.

10.1.1 Etapa de Formación.

El equipo de desarrollo debe formarse en los siguientes cursos

- ✓ JAVA 8
- ✓ ANGULAR 8
- ✓ ARQUITECTURA MICROSERVICIOS
- ✓ SPRING BOOT
- ✓ ANDROID JAVA
- ✓ SPRING CLOUD
- ✓ NETFLIX EUREKA
- ✓ GIT
- ✓ IMPLEMENTACIÓN DE JSON WEB TOKEN

10.1.2 Etapa de Requerimientos.

Se indican los formularios más relevantes de donde se va a obtener la información, así como criterios de aceptación y pruebas que se deben ejecutar

- ✓ Formulario de Registro de Usuarios:

Se obtienen datos personales como teléfono, correo, fecha de nacimiento, contraseña de usuario y nombres.

Umbral de Calidad y límite de errores:

- Validación de Fecha de nacimiento de usuario mayor de edad
- Debe aceptar tratamiento de datos personales (teléfono, correo electrónico)
- Validación de correo electrónico válido
- Contraseña de usuario mínimo 8 caracteres
- Nombres con un máximo de 50 caracteres
- ✓ Formulario de Actualización de Datos:
 - Se obtienen datos personales y ubicación del usuario.

Umbral de Calidad y límite de errores

- Validación de Fecha de nacimiento de usuario mayor de edad
- Nombres con un máximo de 50 caracteres
- Validación de Ubicación con Google Maps
- ✓ Formulario de Registro de Grupos de Entrenamiento:
 - Se obtienen datos de ubicación de los sitios de entrenamiento.

Umbral de Calidad y límite de errores

- Validación de Ubicación con Google Maps

10.1.3 Etapa de Diseño.

Se define la arquitectura de seguridad, estándares criptográficos a utilizar y componentes para evitar ataques conocidos

✓ ARQUITECTURA DE SEGURIDAD

Se define una arquitectura de software basada en Microservicios, para el consumo de las API que exponen cada microservicio se utiliza el tipo de autenticación JSON WEB

TOKEN.

Para la seguridad en el aplicativo se van a manejar ROLES para el acceso de los usuarios a las diferentes funcionalidades

Se implementó el encriptamiento de los datos de navegación del usuario en la página (Local Storage) en SHA256"

✓ COMPONENTES PARA EVITAR ATAQUES CONOCIDOS

Se define el uso en angular de HttpClient para el envío de peticiones, con esto estamos asegurando de que cualquier petición que se ejecute desde la aplicación pase antes por el interceptor

Se definieron controles para ingreso de caracteres para evitar ataques de SQL - INJECTION

10.1.4 Etapa de Implementación

Se define las herramientas a utilizar, las API de terceros que se usan y herramientas de análisis estático de código

- ✓ HERRAMIENTAS
 - JAVA 8
 - ANGULAR 8
 - SPRING BOOT
 - ANDROID

- GIT
- SPRING CLOUD

- ✓ API
 - API de Google Maps

- ✓ ANÁLISIS ESTÁTICO

Se define el uso de la herramienta SonarQube para el análisis estático de código.

10.1.5 Etapa de Comprobación

Se utilizo SonarQube para el análisis y revisión estático de código para búsqueda de vulnerabilidades. También se implementaron pruebas de caja negra para verificar inyección SQL y Cross Site Scriting. A continuación, se muestran las pruebas ejecutas más relevantes.

Caso de Prueba Seguridad 1 Inyección SQL en Login			
Objetivo:	Ingresar a la aplicación web por medio de SQL inyección		
Pasos:	1.Ingresar a la aplicación	2. Ingresar usuario y contraseña	3.Presionar clic sobre la opción ingresar

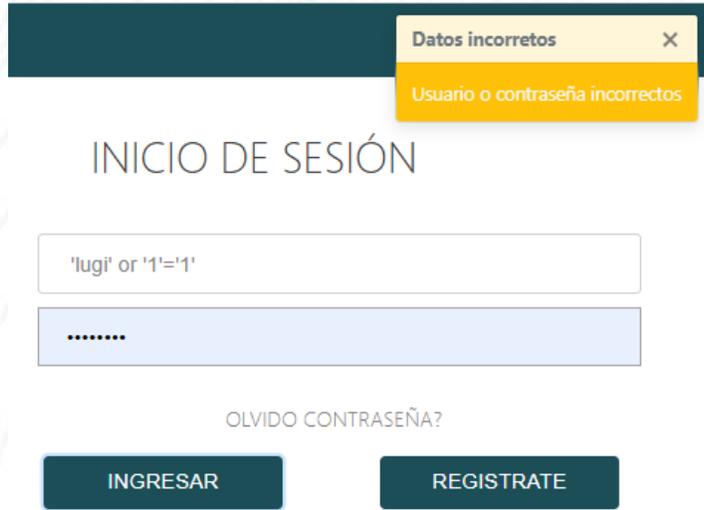
<p>Se digita la sentencia 'lugi' or '1'='1' y una contraseña cualquiera para comprobar el caso de prueba.</p>	
<p>Se valida que al ingresar la sentencia anterior no permite ingresar en la aplicación</p>	

Tabla 31. Prueba de Inyección SQL en Login

Caso de Prueba Seguridad 2 XSS en formulario de Registro		
Objetivo:	Ingresar formulario de registro e inyectar scripts.	
Pasos:	1. Ingresar a la aplicación	2. Ingresar al formulario de registro
		3. Ingresar datos con scripts y presionar clic en registrar.

<p>Se digitan script de inyección en varios campos del formulario.</p>	<h3 style="text-align: center;">REGISTRESE Y ACCEDA A TODOS NUESTROS BENEFICIOS</h3> <p>Obligatorios*</p> <p>Nombres* <input type="text" value="<script>alert('hola')</script>"/></p> <p>Apellidos* <input type="text" value="<script>alert('otra hola')</script>"/></p> <p>Fecha de nacimiento* <input type="text" value="1995-01-03"/></p> <p>Celular* <input type="text" value="3014409872"/></p> <p>Usuario* <input type="text" value="<script>alert('otra hola')</script> "/></p> <p>E-mail* <input type="text" value="lmarquez16@uan.edu.co"/></p> <p>Contraseña* <input type="password" value="....."/></p> <p>Confirmar Contraseña* <input type="password" value="....."/></p> <p>Sexo* <input type="text" value="Femenino"/></p> <p>Tipo de Usuario* <input type="text" value="Cliente"/></p> <p><input checked="" type="checkbox"/> Acepta nuestras Condiciones y políticas de datos</p>
<p>Se valida que no permite la inyección de scripts por que los campos solo permiten que se ingrese números y letras.</p>	<p>Obligatorios*</p> <p>Nombres* <input type="text" value="<script>alert('hola')</script>"/> ❗ <small style="color: red;">El nombre debe tener numero, letras o espacio</small></p> <p>Apellidos* <input type="text" value="<script>alert('otra hola')</script>"/> ❗ <small style="color: red;">El apellido debe tener numero, letras o espacio</small></p> <p>Fecha de nacimiento* <input type="text" value="1995-01-03"/></p> <p>Celular* <input type="text" value="3014409872"/></p> <p>Usuario* <input type="text" value="<script>alert('otra hola')</script>"/> ❗ <small style="color: red;">Debe ingresar solo numeros o letras</small></p> <p>E-mail* <input type="text" value="lmarquez16@uan.edu.co"/></p> <p>Contraseña* <input type="password" value="....."/></p> <p>Confirmar Contraseña* <input type="password" value="....."/></p> <p>Sexo* <input type="text" value="Femenino"/></p> <p>Tipo de Usuario* <input type="text" value="Cliente"/></p> <p><input checked="" type="checkbox"/> Acepta nuestras Condiciones y políticas de datos</p>

Tabla 32. Prueba de XSS en formulario de registro.

Caso de Prueba Seguridad 3 XSS en formulario de registro de grupos	
Objetivo:	Ingresar formulario de registro e inyectar scripts.

<p>Pasos:</p>	<p>1. Iniciar sesión en la aplicación</p>	<p>2. Ingresar al formulario creación de grupos.</p>	<p>3. Ingresar datos con scripts y presionar clic en guardar.</p>
<p>Se digitan script de inyección en varios campos del formulario.</p>	<div data-bbox="435 426 1555 1077"> <h3>Crear nuevo grupo de entrenamiento</h3> <p>Nombre: <input type="text" value="<script>alert('hola')</script>"/></p> <p>Cantidad cupos: <input type="text" value="<script>alert('hola')</script>"/></p> <p>Color: <input type="color" value="black"/></p> <p><input type="checkbox"/> Activo</p> <p>Fecha inicial: <input type="text" value="2020-05-21"/></p> <p>Fecha final: <input type="text" value="2020-09-26"/></p> <p>Categoría: <input type="text" value="Crossfit"/></p> <p>Descripción: <input type="text" value="<script>alert('hola')</script>"/></p> <p><input type="button" value="Guardar"/></p> </div>		
<p>Se valida no permite la inyección de scripts por que los campos solo permiten que se ingrese números y letras.</p>	<div data-bbox="435 1150 1555 1759"> <h3>Crear nuevo grupo de entrenamiento</h3> <p>Nombre: <input type="text" value="<script>alert('hola')</script>"/> ❗</p> <p><small>El nombre debe contener solo letras y numeros</small></p> <p>Cantidad cupos: <input type="text" value="<script>alert('hola')</script>"/> ❗</p> <p><small>Debe ingresar solo numeros</small></p> <p>Color: <input type="color" value="black"/></p> <p><input type="checkbox"/> Activo</p> <p>Fecha inicial: <input type="text" value="2020-05-21"/></p> <p>Fecha final: <input type="text" value="2020-09-26"/></p> <p>Categoría: <input type="text" value="Crossfit"/></p> <p>Descripción: <input type="text" value="<script>alert('hola')</script>"/></p> </div>		

Tabla 33. Prueba de XSS en formulario de registro de grupos.

10.1.6 Etapa de Release

Mitigación de riesgos.

✓ ACCESO NO AUTORIZADO

Bloqueo de cuenta

Posible duración: cuatro horas

✓ ROBO DE DATOS

El borrado de información de terceros y La alteración de la información de terceros: Validar los logs acceso a servidor y poner mayor seguridad a los puertos de acceso.

Posible duración: cuatro horas

✓ DENEGACIÓN DEL SERVICIO

Reiniciar servidor, validar el despliegue en el servidor, revisar log del servidor y realizar seguimiento del servidor para identificar posibles fallas.

Posible duración: treinta Minutos para reiniciar el servidor o validar el despliegue de aplicaciones en el servidor.

Siete días para revisar log del servidor y realizar seguimiento del servidor para identificar posibles fallas

✓ INTENTO FALLIDO DE INGRESO EN APLICACIÓN

Realizar cambio de contraseña

Posible duración: diez minutos