

Manual técnico

Proyecto: Sistema para gestión de actividades en el gimnasio del CEET - GymSenApp

Versión: 1.0

SENA

2024

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor(es) | Descripción | Fecha |
| 1.0 | Mariana Lizeth Marín Rojas, Santiago Salamanca Díaz, Marco Antonio Ortíz Ramírez | Creación del documento | Marzo 25 del 2024 |

Tabla de contenido

[1. Introducción 5](#_Toc162032959)

[*1.1. Alcance del documento.* 5](#_Toc162032960)

[*1.2. Generalidades del proyecto.* 5](#_Toc162032961)

[2. Justificación del manual técnico 6](#_Toc162032962)

[3. Objetivos del manual técnico 6](#_Toc162032963)

[4. Requerimientos de conocimiento para manejar el sistema 7](#_Toc162032964)

[*4.1. Generales.* 7](#_Toc162032965)

[*4.2. Técnicos.* 7](#_Toc162032966)

[5. Desarrollo 8](#_Toc162032967)

[*5.1. Generalidades.* 8](#_Toc162032968)

[*5.2. Procesos del sistema.* 8](#_Toc162032969)

[5.2.1. Mapa de procesos (BPMN). 8](#_Toc162032970)

[5.2.2. Requerimientos funcionales y no funcionales. 10](#_Toc162032971)

[5.2.3. Casos de uso. 20](#_Toc162032972)

[5.2.1. Mapa de historias de usuario (Móvil). 30](#_Toc162032973)

[*5.3. Administración de usuarios.* 31](#_Toc162032974)

[*5.4. Arquitectura del sistema.* 31](#_Toc162032975)

[5.4.1. Diagramas UML. 32](#_Toc162032976)

[5.4.2. Base de datos. 35](#_Toc162032977)

[*5.5. Documentación del código fuente.* 38](#_Toc162032978)

[5.5.1. Estructura de carpetas - Frontend. 38](#_Toc162032979)

[5.5.2. Relaciones entre componentes. 41](#_Toc162032980)

[*5.6. Entorno de producción.* 43](#_Toc162032981)

[5.6.1. Descripción de las plataformas. 43](#_Toc162032982)

[5.6.2. Acuerdos a nivel de servicio. 44](#_Toc162032983)

[6. Anexos 45](#_Toc162032984)

[7. Glosario 46](#_Toc162032985)

[8. Referencias 46](#_Toc162032986)

# **1. Introducción**

## ***1.1. Alcance del documento.***

El propósito del presente documento es proporcionar la información principal del software GymSenApp para los administradores del sistema, que se encargarán en algún punto de este, siendo brindada la capacitación mediante un manual técnico del sistema.

En este sentido, se deben tener claras unas nociones generales sobre el proyecto, las cuales permitirán obtener una contextualización idónea sobre el proyecto de software. Veamos pues las consideraciones:

## ***1.2. Generalidades del proyecto.***

El sistema para gestión de actividades en el gimnasio del CEET, llamado “GymSenApp”, es un proyecto formativo el cual busca optimizar los procesos que engloba el uso del gimnasio del centro, permitiendo así una mejor planificación de las rutinas de ejercicios de los aprendices, así como también permite apoyar el acompañamiento y comunicación por parte del personal del gimnasio hacia los aprendices que asistan al lugar.

El sistema está compuesto de tres (3) grupos de actividades generales agrupadas según el tipo de usuario que puede acceder a ellas:

* + Administradores
  + Aprendices
  + Instructores

Cada uno de estos grupos o roles, está compuesto por actividades específicas:

* + Administradores
    - Gestión de anuncios
    - Gestión de usuarios
    - Generación de reportes
  + Aprendices
    - Visualizar anuncios
    - Visualizar recomendaciones saludables
    - Planificar rutinas
  + Instructores
    - Gestionar asistencias

# **2. Justificación** **del manual técnico**

Este manual se está creando debido a su importancia a la hora de darle de alguna forma un final al software, de modo tal que se abren las puertas a que otras personas se unan al proyecto y nos ayuden a hacerlo crecer, por lo que empezará a hacerse necesario explicar a cada nuevo integrante cómo funciona el sistema, qué contiene y demás aspectos importantes del sistema. Ante este escenario, es imperativo preparar documentos que nos sirvan de apoyo para brindarles la capacitación necesaria a los nuevos integrantes y desarrolladores del proyecto, a fin de que conozcan por medio de elementos ajenos a nosotros, los creadores del proyecto, cómo desarrollar sus funciones dentro del sistema; razón por la cual se crea este documento.

# **3. Objetivos del manual técnico**

El manual técnico siendo creado en este documento tiene como objetivo principal brindarles a los nuevos desarrolladores que se unan al equipo, la información necesaria para que puedan desarrollar acciones enfocadas en la mejora del software GymSenApp. Adicionalmente, comprende 2 objetivos específicos:

1. Brindar una contextualización adecuada del software a personas que quieran conocer qué procesos maneja el mismo y cómo está constituido.
2. Brindar en un solo espacio varios componentes que le permitan a todos los desarrolladores del proyecto acceder fácilmente a información que necesiten conocer del sistema sin mucho esfuerzo.

# **4. Requerimientos de conocimiento para manejar el sistema**

## ***4.1. Generales.***

El usuario debe tener conocimientos básicos y hasta un poco específicos sobre Frontend, Backend y bases de datos, así como deben saber interpretar gráficos que acompañan el desarrollo de software, esperando que logren comprender fácilmente diagramas como lo son los UML.

## ***4.2. Técnicos.***

En cuanto a hardware se requerirá mínimo de los elementos básicos que contiene un computador (teclado, mouse, y demás).

En cuanto a software de forma general se requerirá de un sistema operativo en lo posible recientemente sacado, pensando en que se puedan aprovechar las características que estos sistemas operativos brinden; así como se requerirá, por otro lado, de mínimo un navegador de internet. Todo lo anterior enfocado en que se quiera analizar el sistema de forma general, ya desplegado. Si se desea trabajar en el código, se requiere de un editor de código, se requiere tener instalado NodeJS (en lo posible la versión 18.6.6), se requiere tener instalado Angular en el equipo, también se requiere de un emulador de servidor (como Xampp) y de un gestor de bases de datos, en el caso MySQL.

# **5. Desarrollo**

## ***5.1. Generalidades.***

El presente documento especifica los elementos componen al software y que junto con los diagramas correspondientes detallan los procesos que desarrolla el mismo, de tal forma que sirve como guía para entender el sistema, los requerimientos a los que responde, cómo está constituido, entre otros elementos relacionados. Pasemos a analizar los componentes principales:

## ***5.2. Procesos del sistema.***

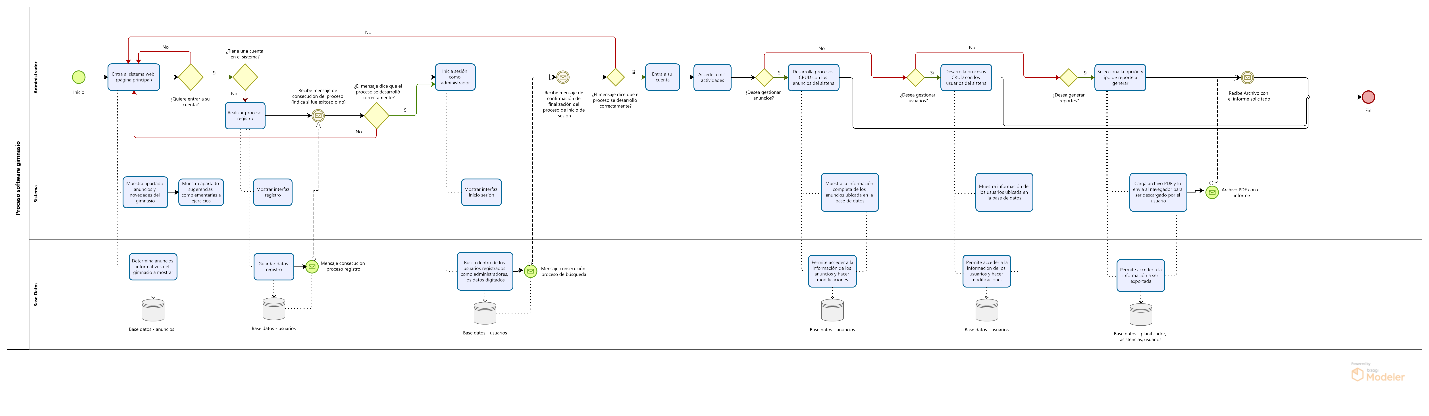
El sistema divide sus actividades en tres áreas, enfocadas en las necesidades identificadas con respecto a los tipos de usuarios que interactúan con el sistema. Con base en lo anterior, podemos pasar a visualizar y comprender los procesos que le competen a cada uno de estos usuarios o actores dentro del sistema:

### **5.2.1. Mapa de procesos (BPMN).**

Enfocándonos en los procesos, encontramos este diagrama, el cual describe detalladamente cada uno de los procesos y actividades que desarrollan los usuarios dentro del sistema, permitiendo entender la lógica de negocio que se maneja y qué elementos deben ser comprendidos para pasar a un entorno en el que se traigan a la realidad las actividades expuestas en los diagramas de este tipo.

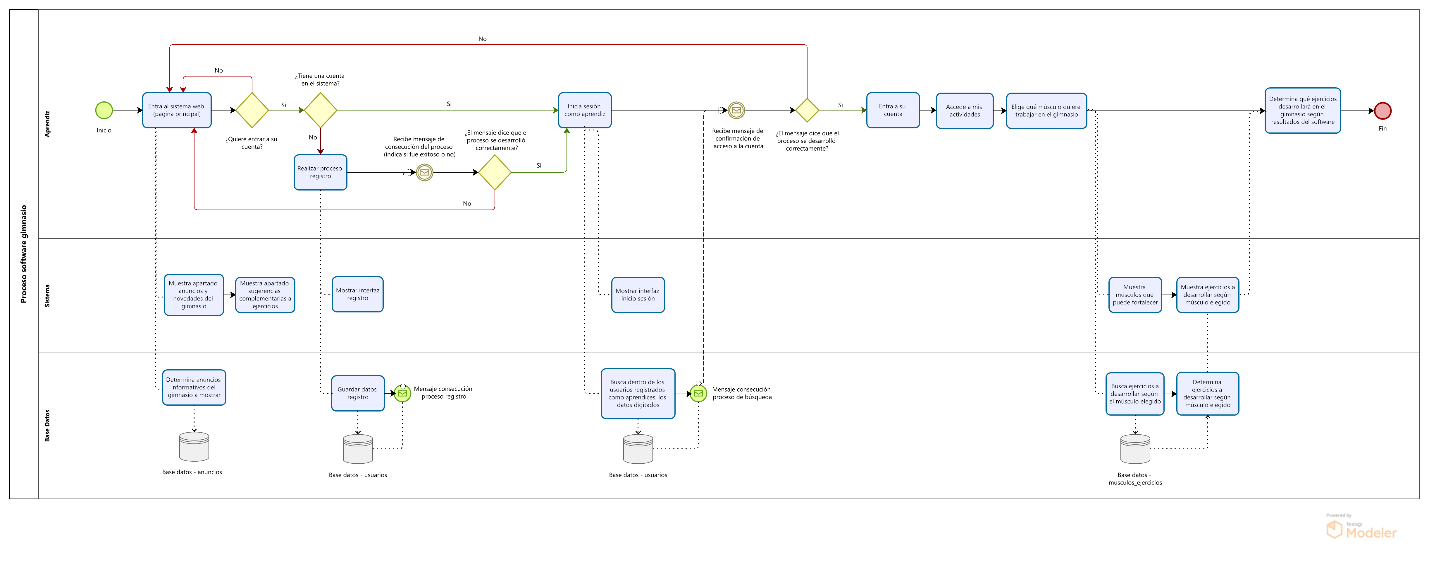
#### ***5.2.1.1. Administrador.***

En cuanto a estos usuarios y los procesos que desarrollarán dentro del sistema, contamos con el siguiente diagrama BPMN:



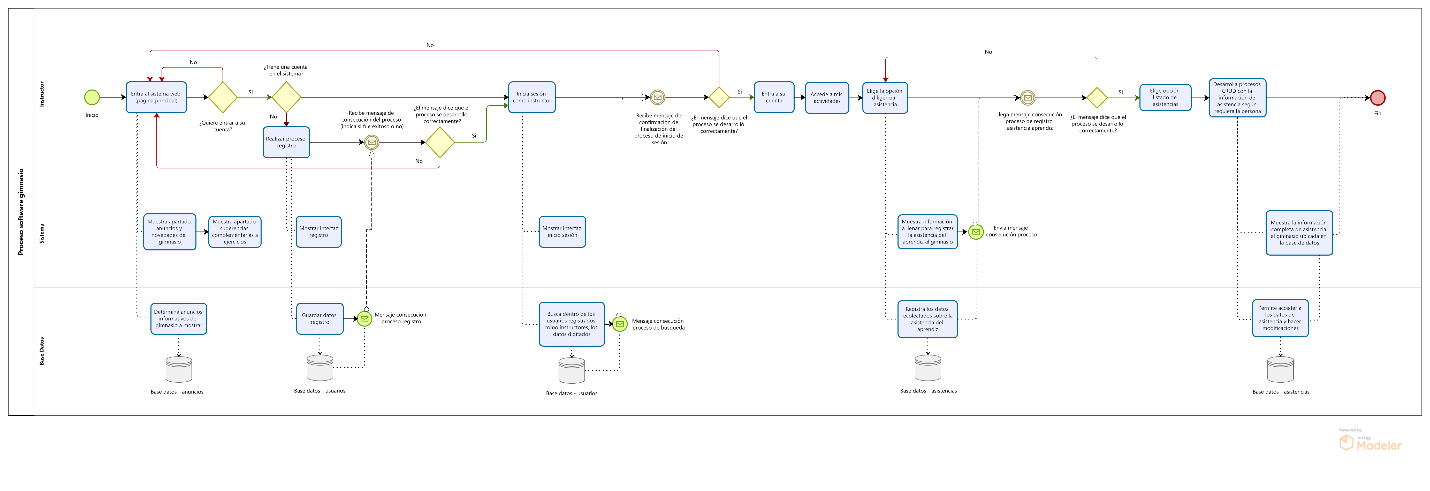
#### ***5.2.1.2. Aprendiz.***

En cuanto a estos usuarios y los procesos que desarrollarán dentro del sistema, contamos con el siguiente diagrama BPMN:



#### ***5.2.1.3. Instructor.***

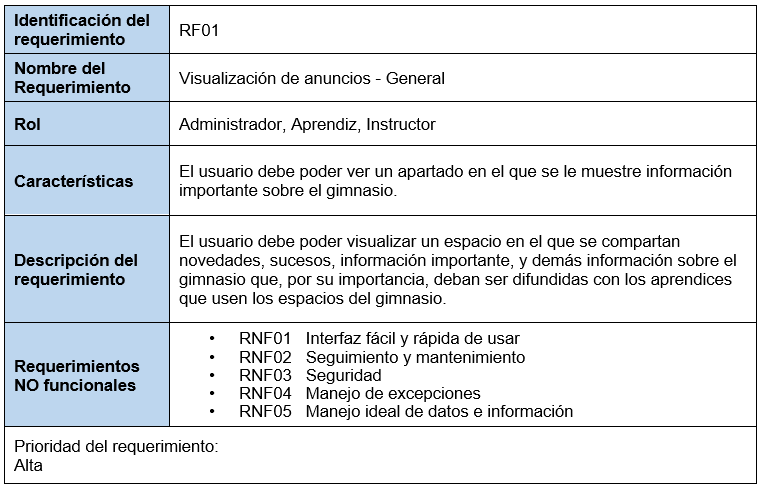
En cuanto a estos usuarios y los procesos que desarrollarán dentro del sistema, contamos con el siguiente diagrama BPMN:

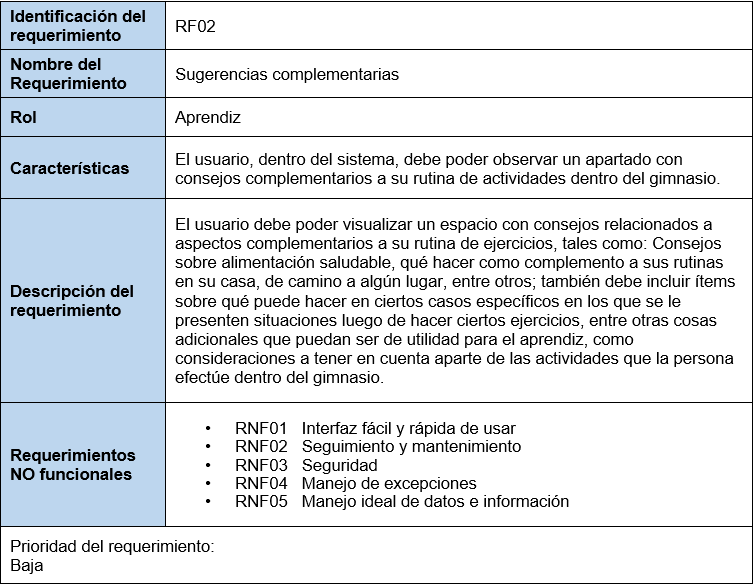


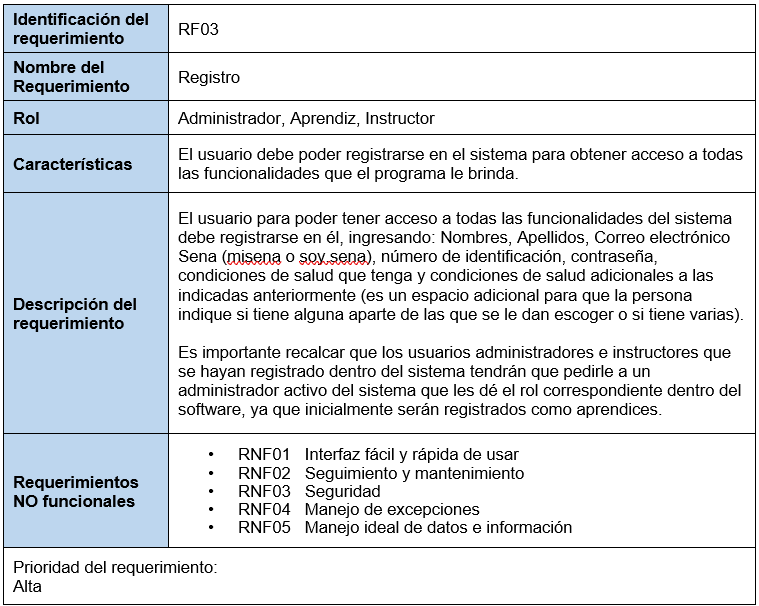
### **5.2.2. Requerimientos funcionales y no funcionales.**

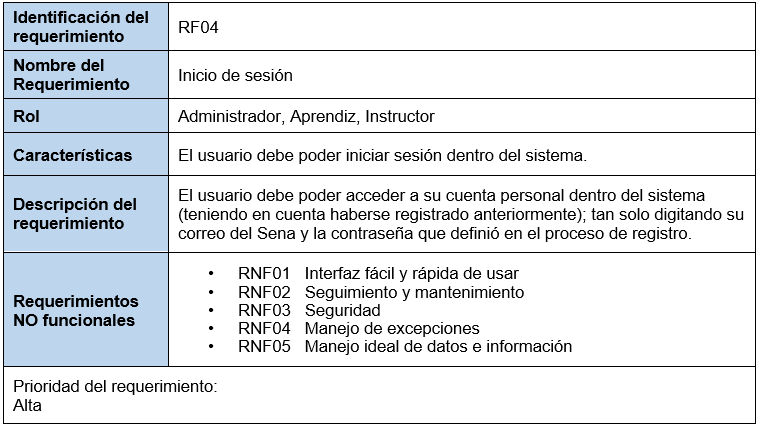
Ahora, veamos los requerimientos de los stakeholders del proyecto:

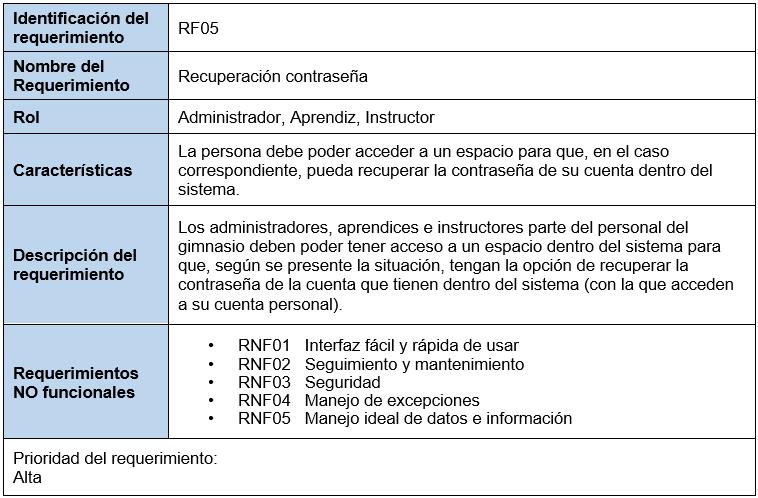
**Los requerimientos funcionales son los siguientes:**

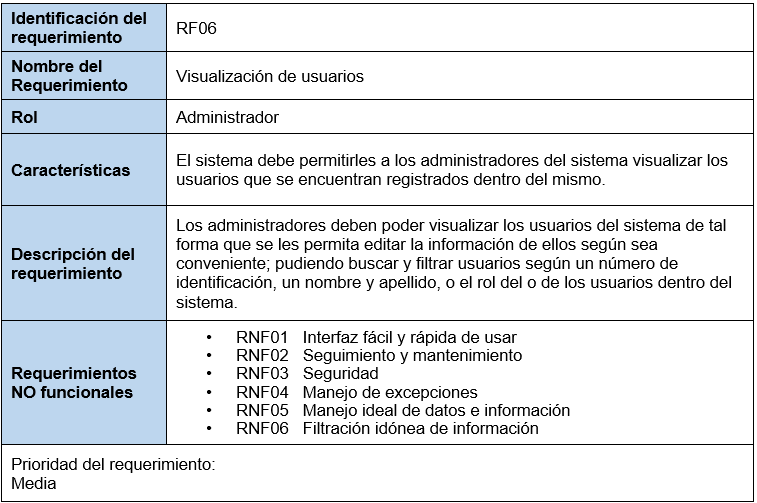


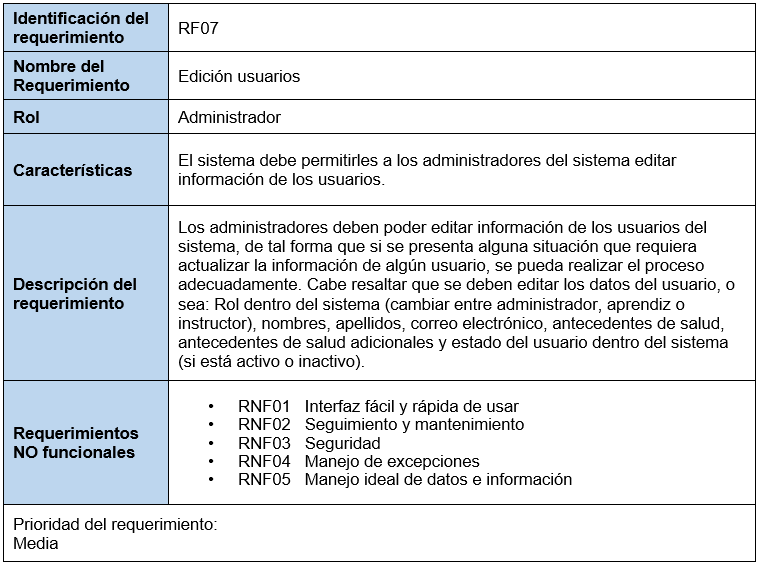




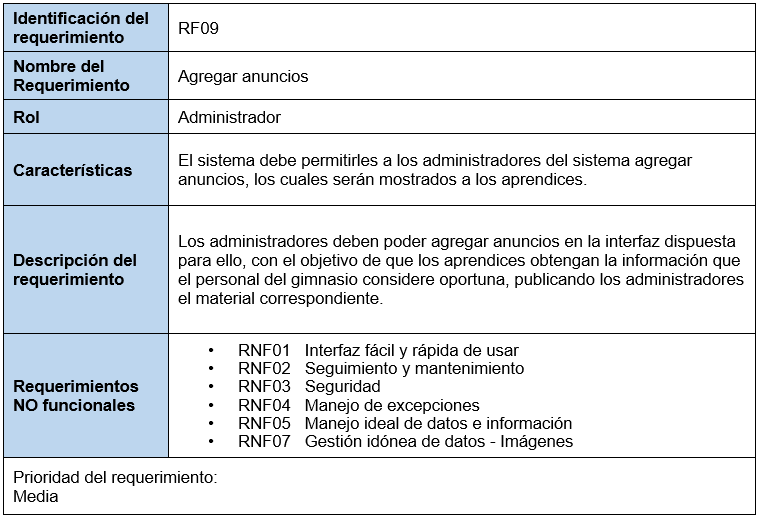


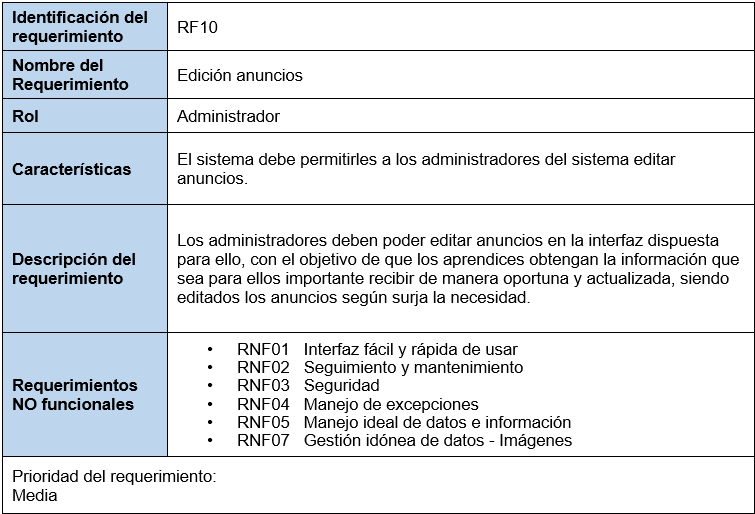


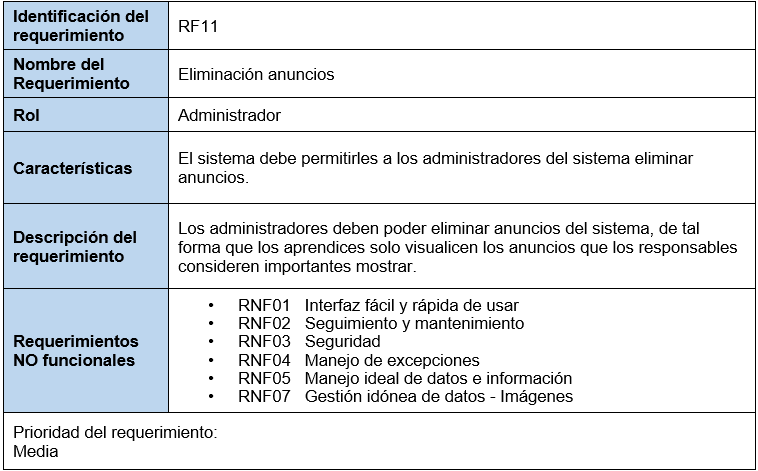


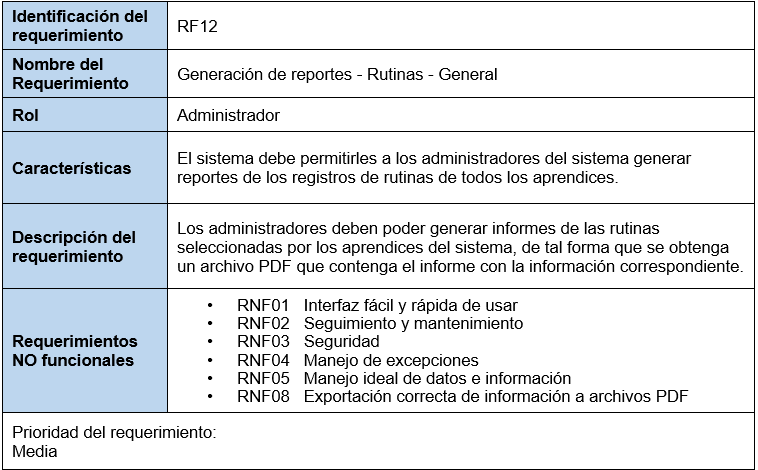


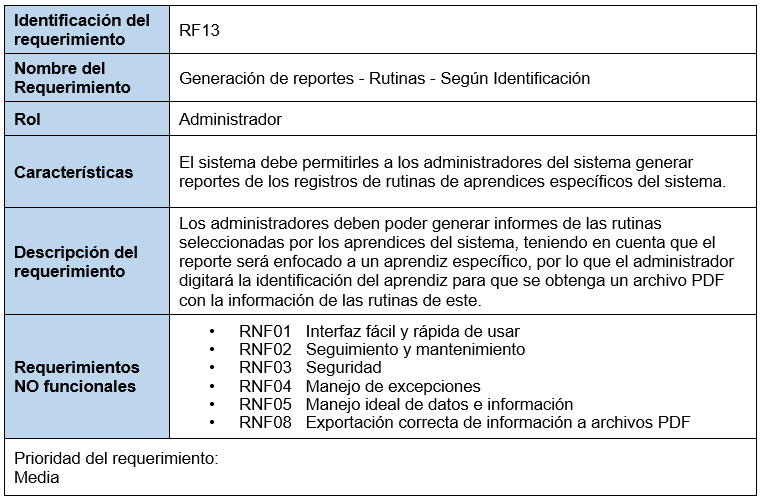


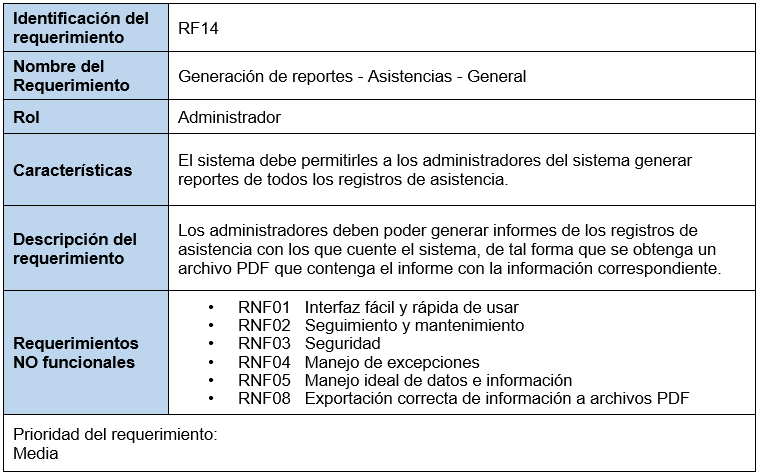


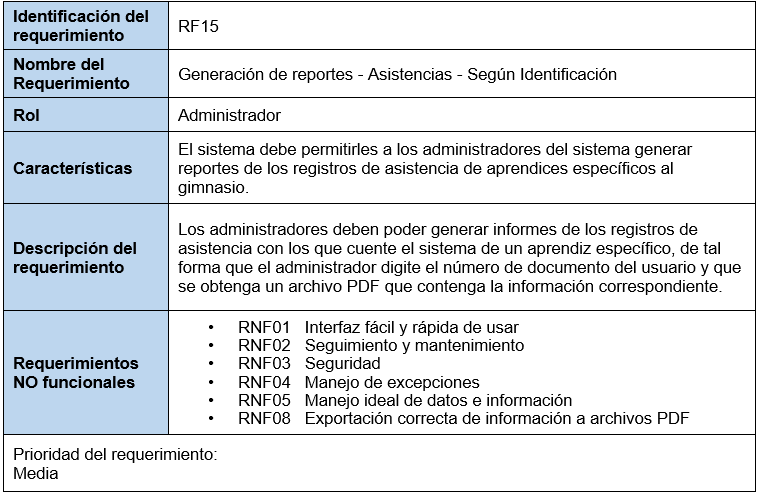


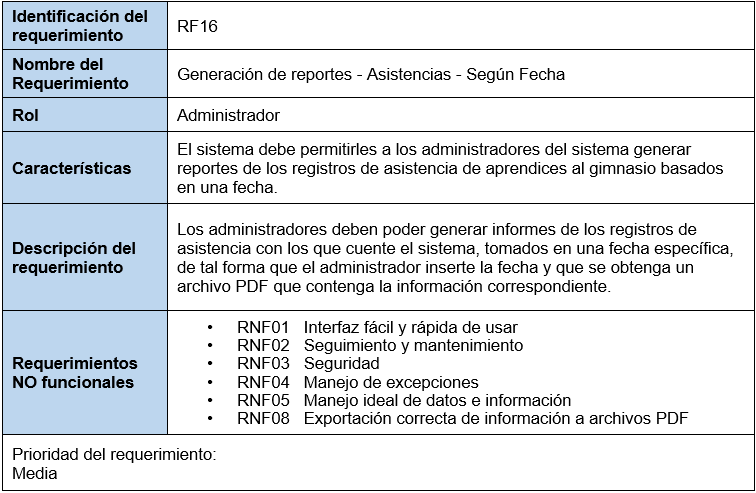


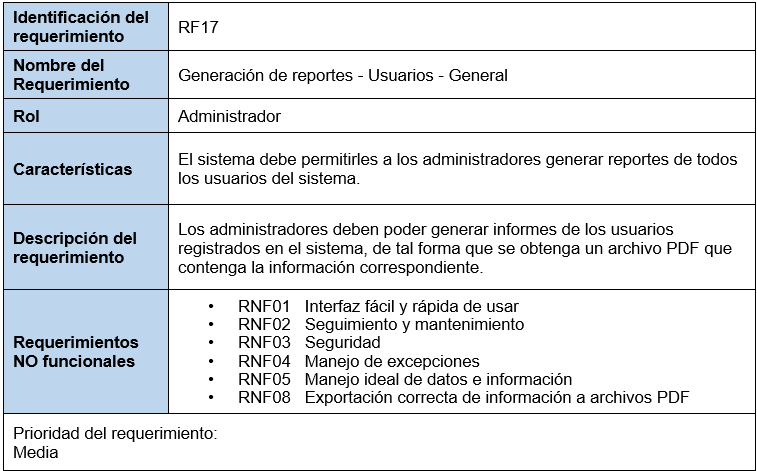




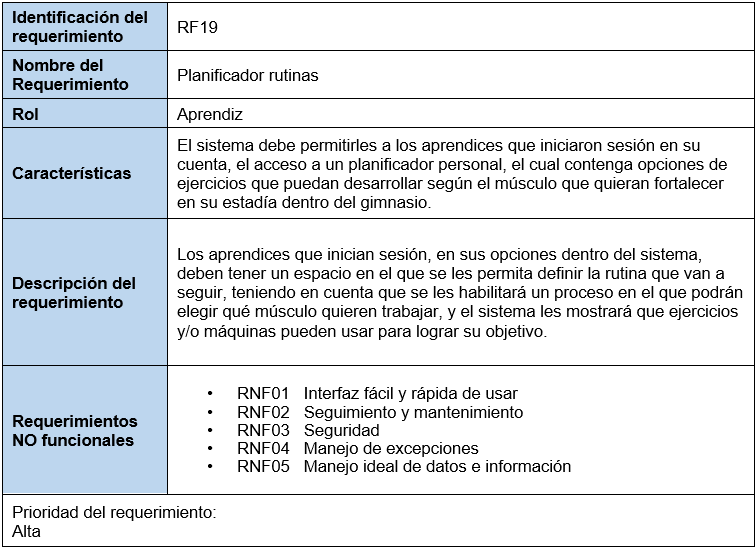


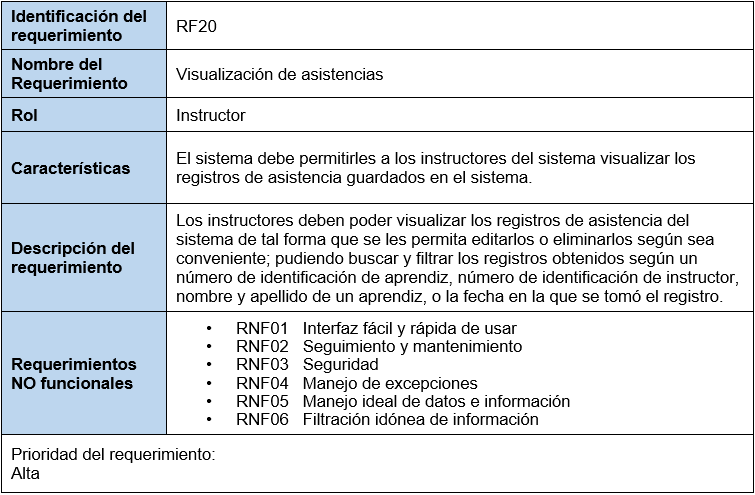


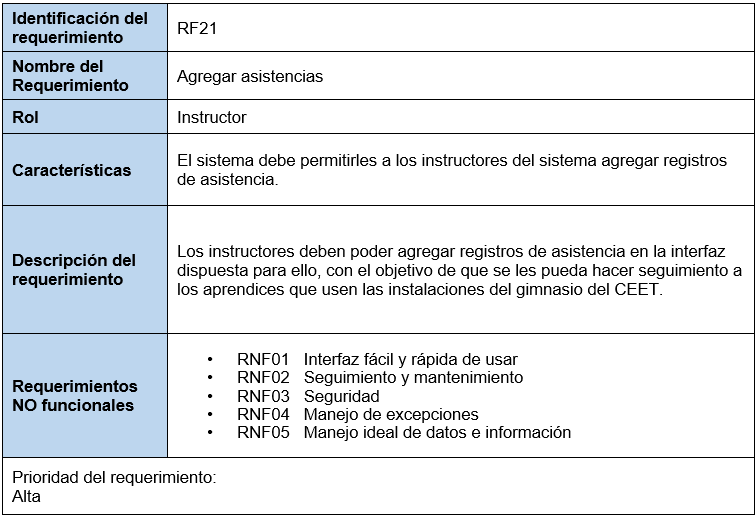


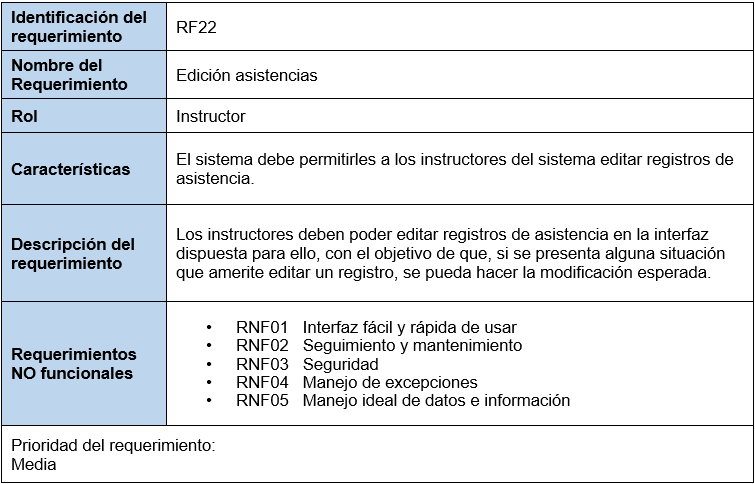


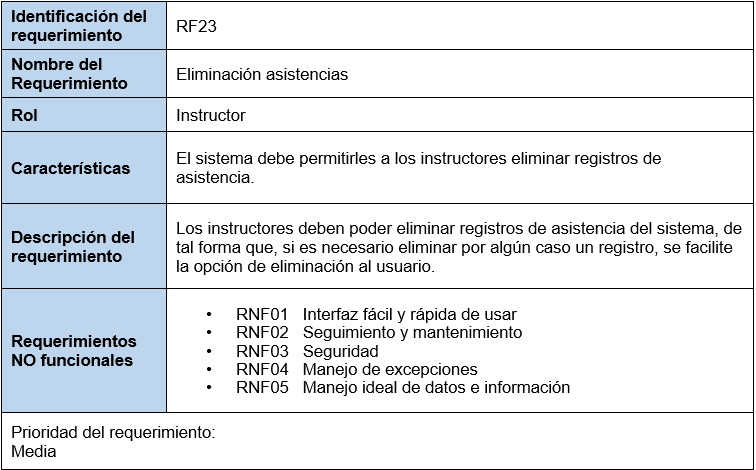




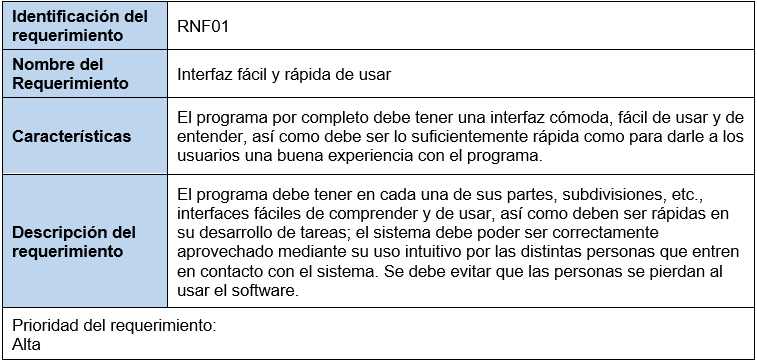


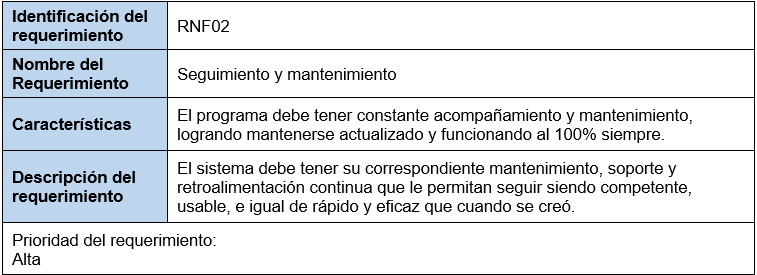


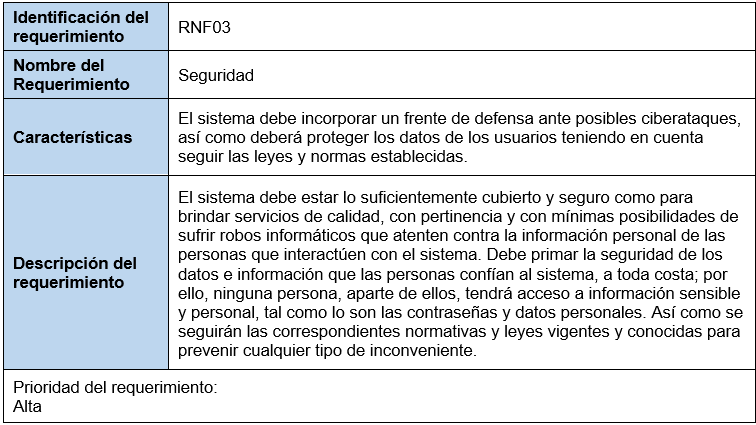




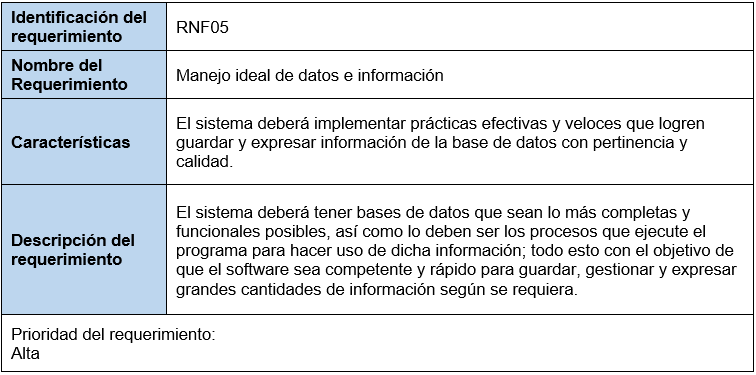
**Los requerimientos no funcionales son los siguientes:**

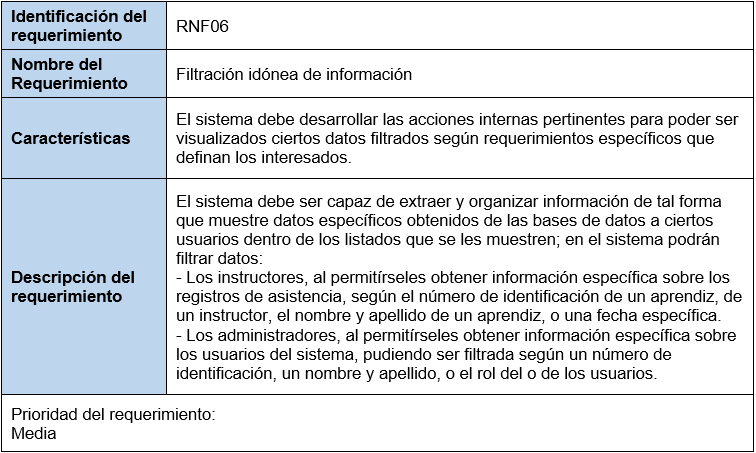


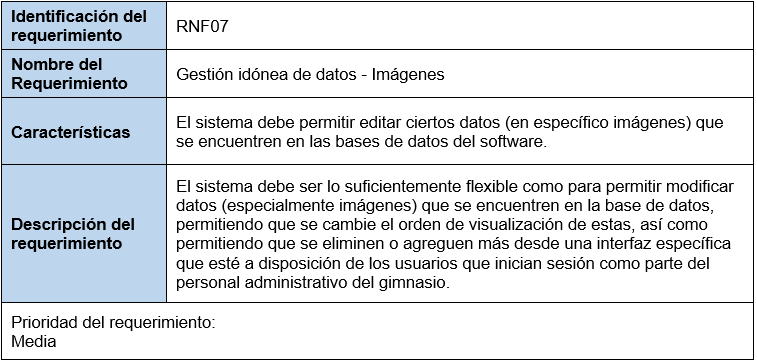


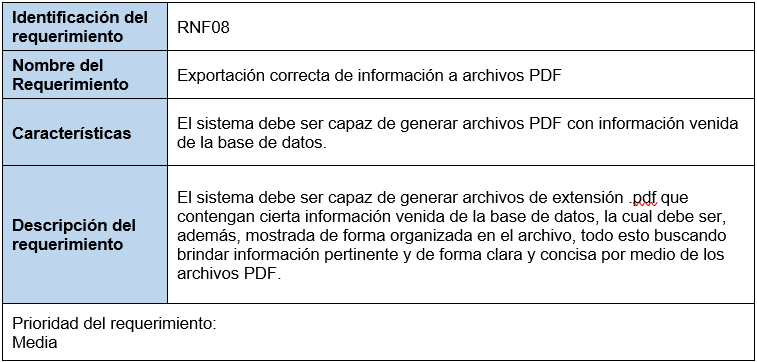








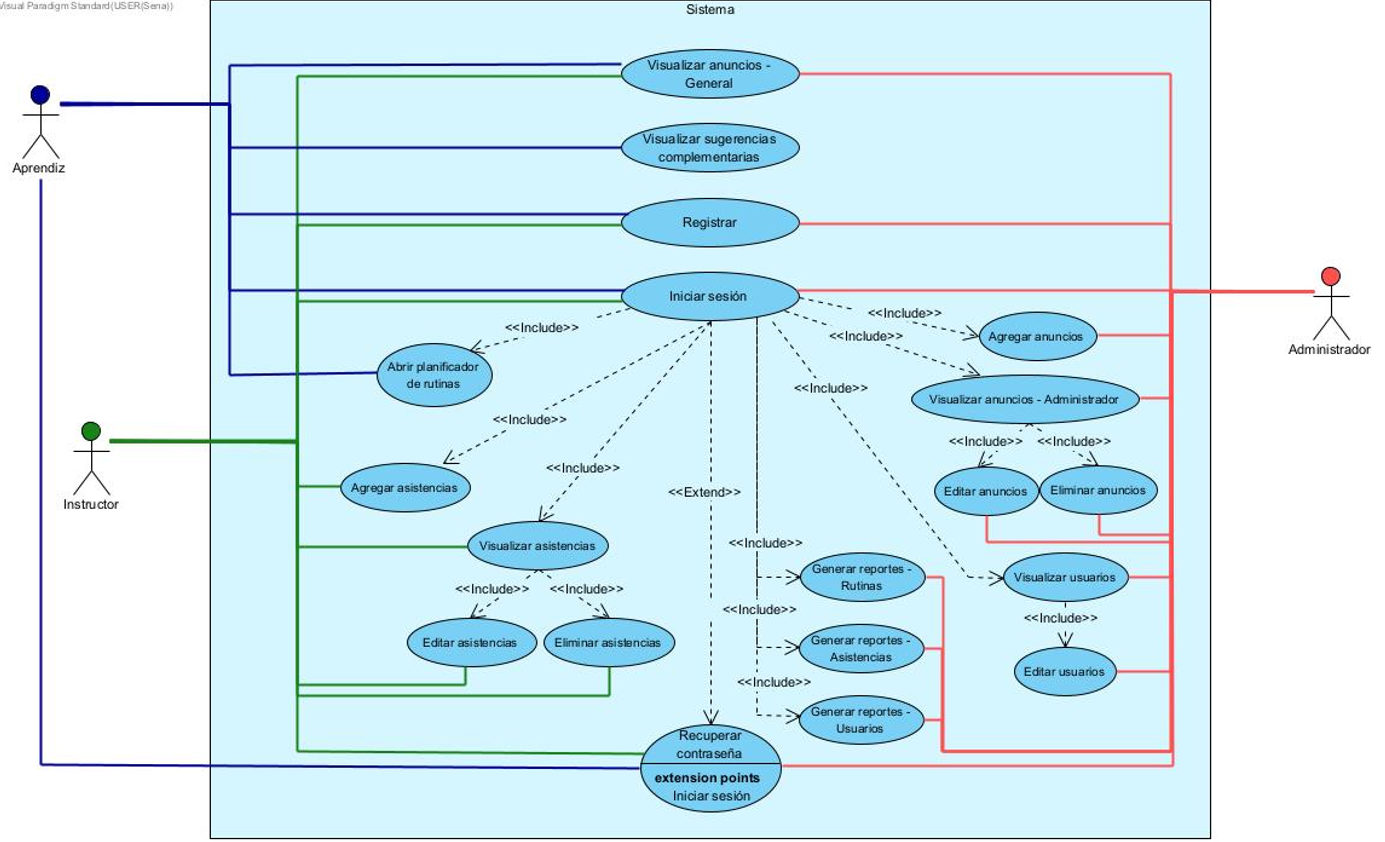




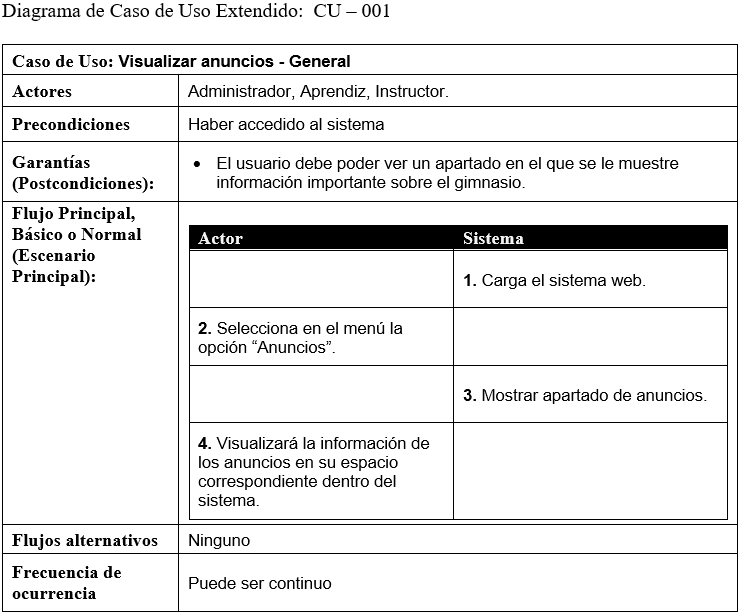
### **5.2.3. Casos de uso.**

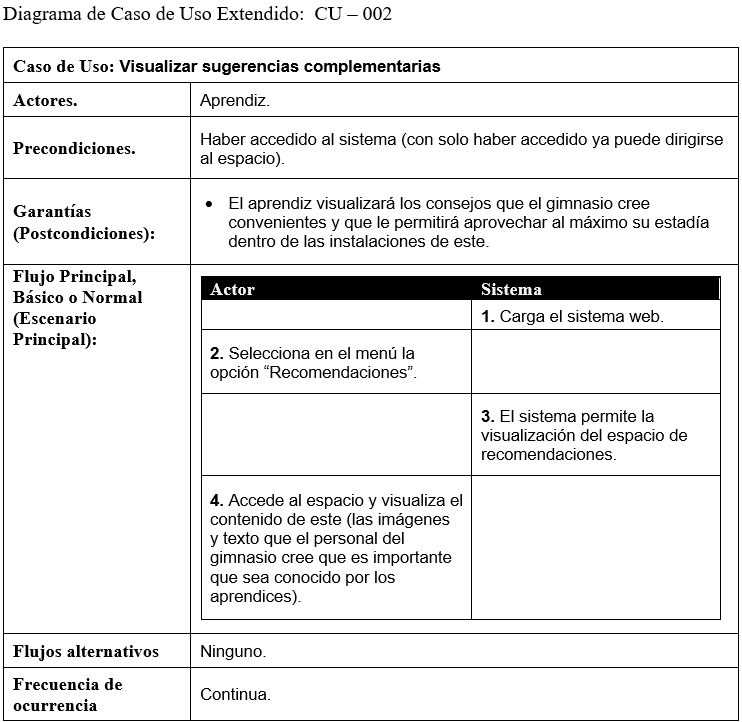
Ahora, veamos el diagrama de casos de uso y los casos de uso extendidos del proyecto:

#### ***5.2.3.1. Diagrama de casos de uso.***

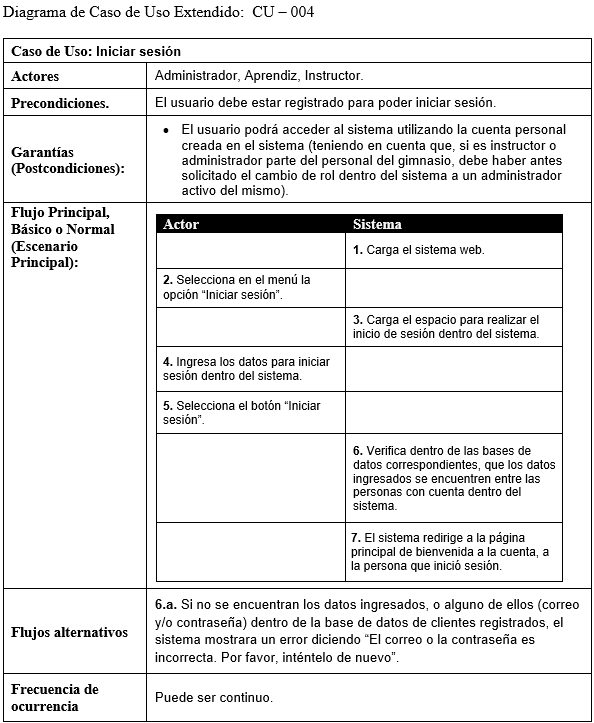


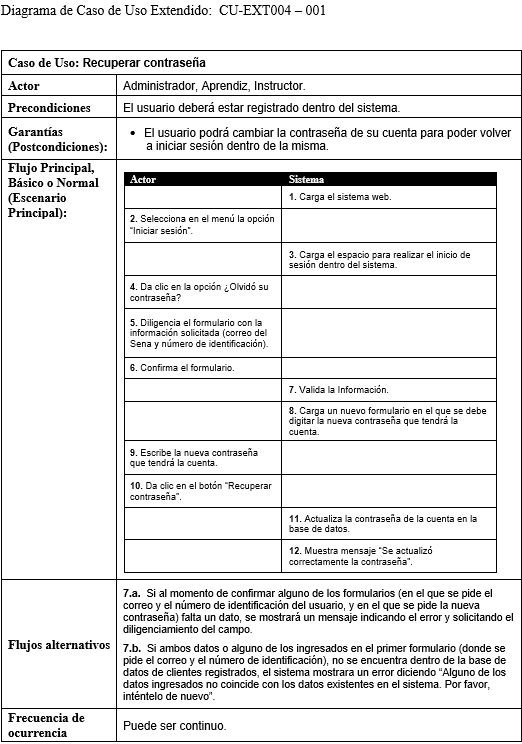
#### ***5.2.3.2. Casos de uso extendido.***

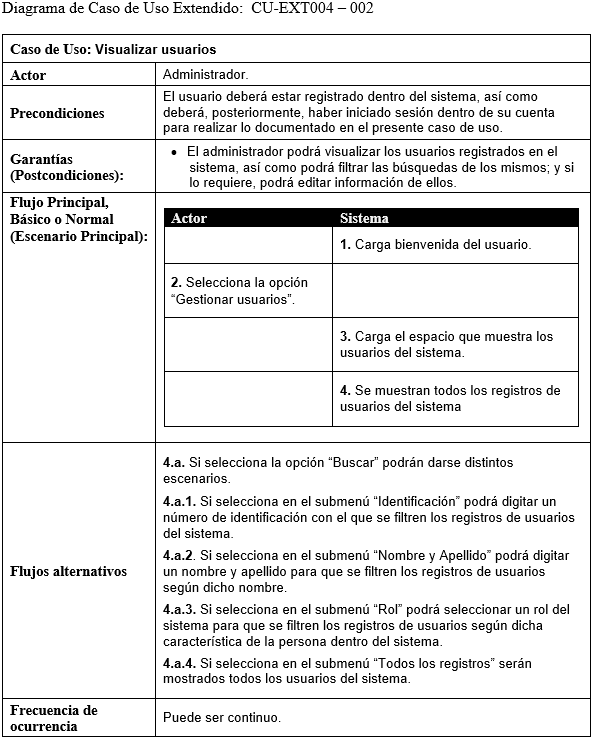


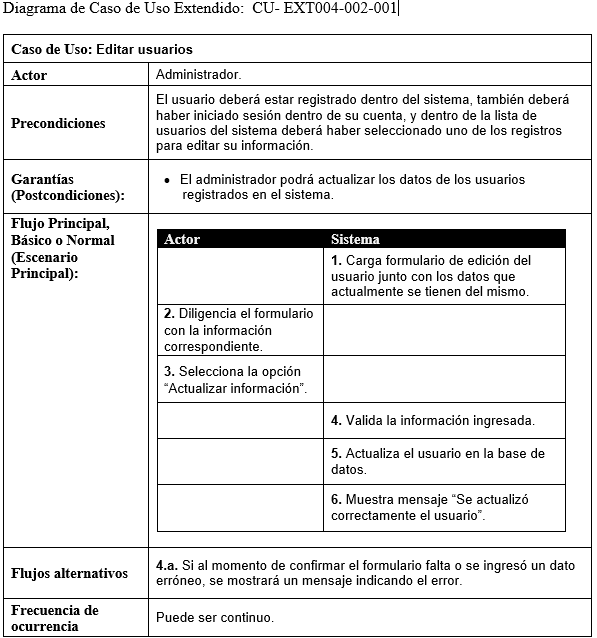


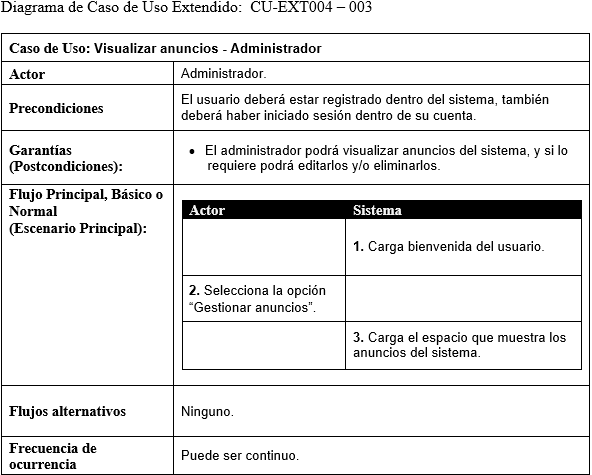


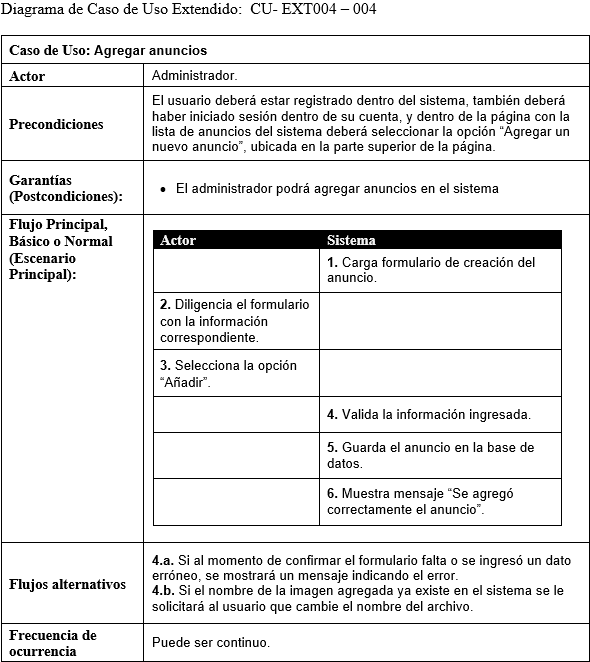


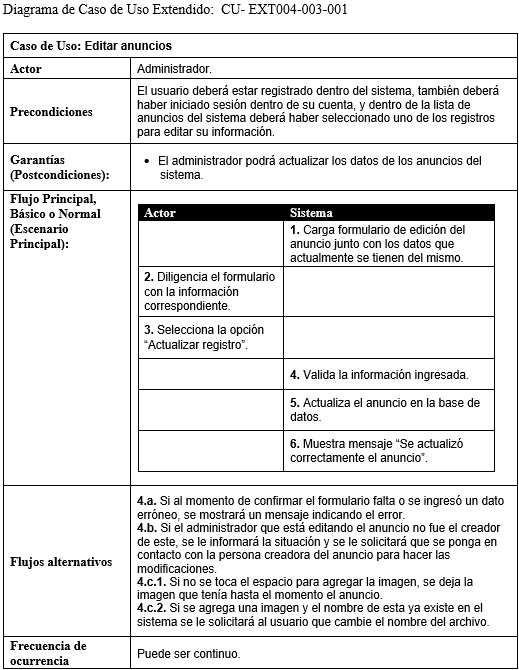


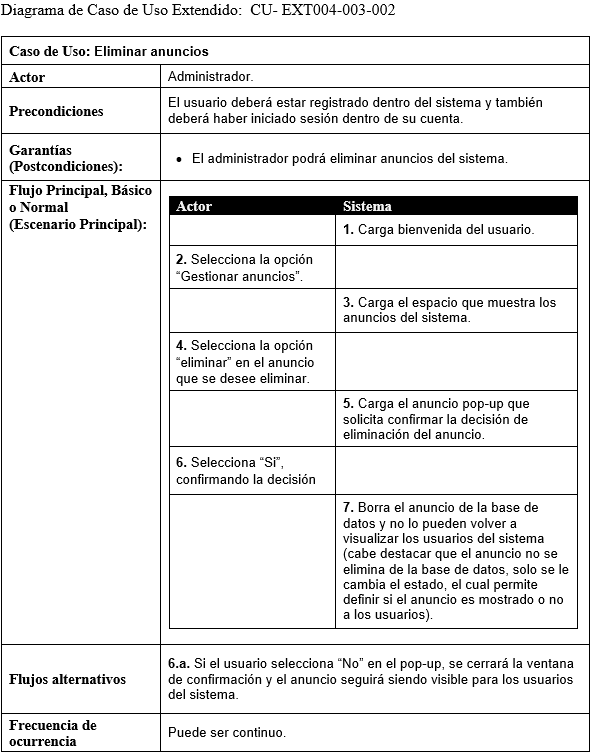


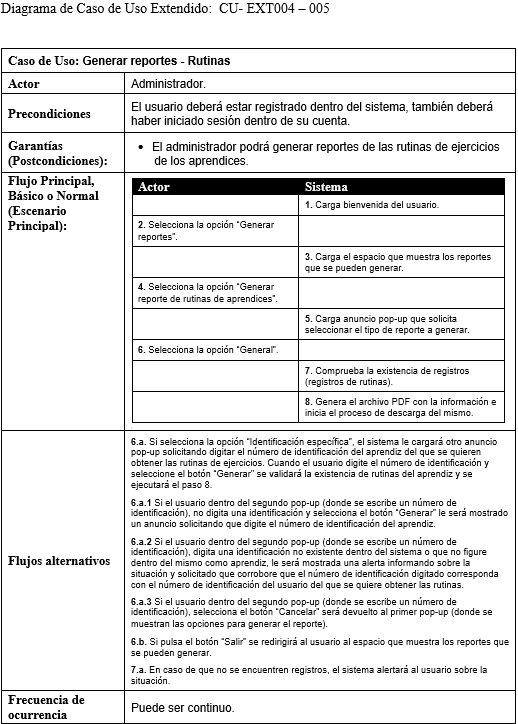




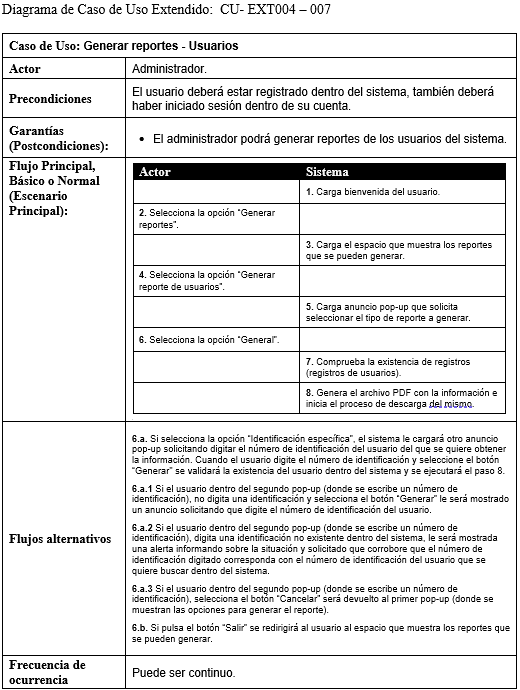


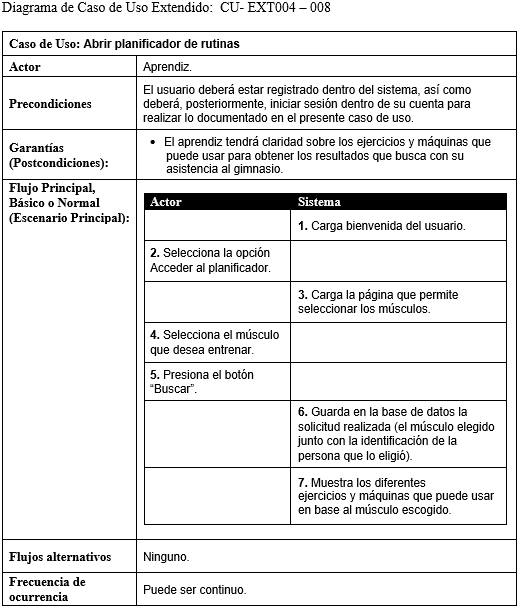


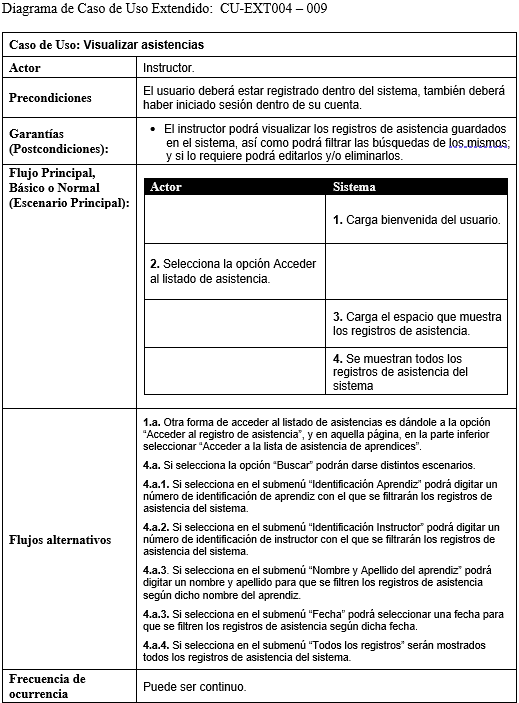


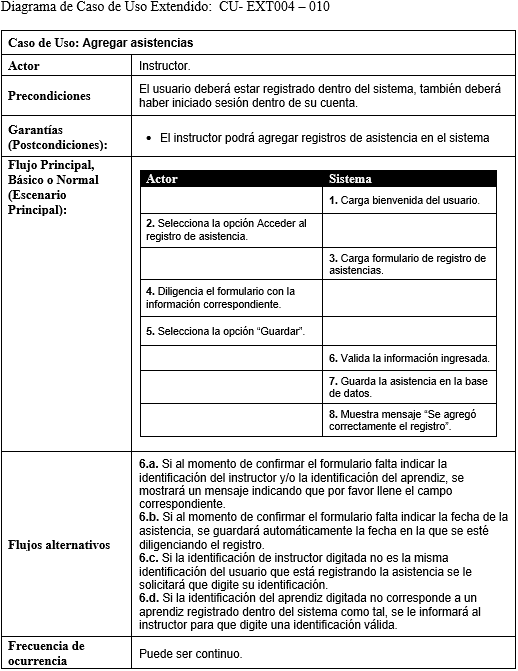


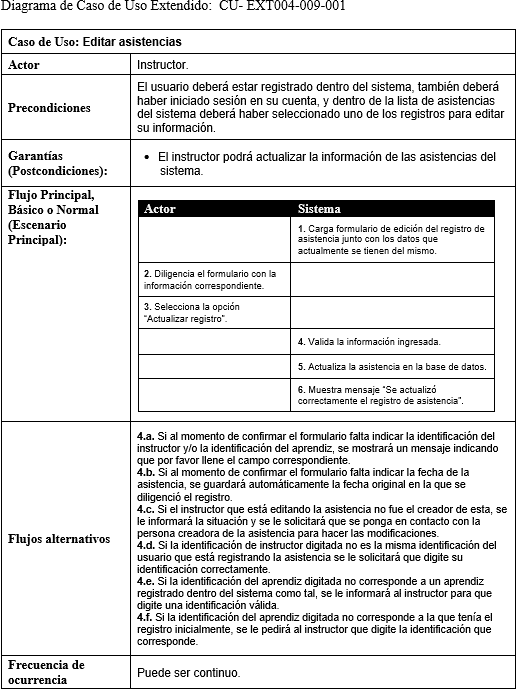


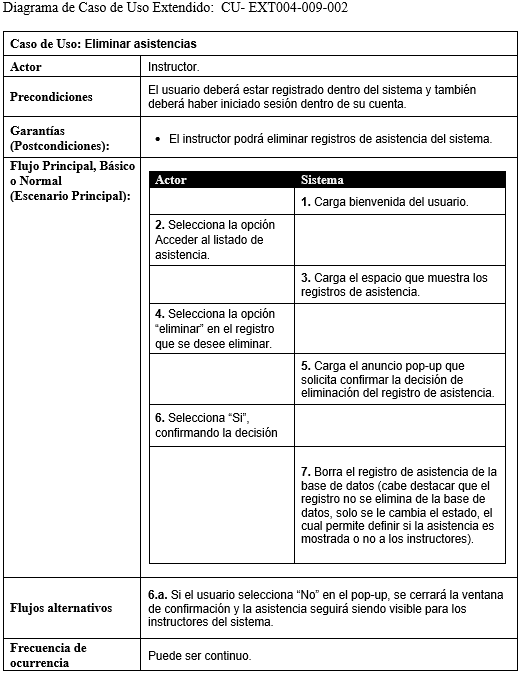






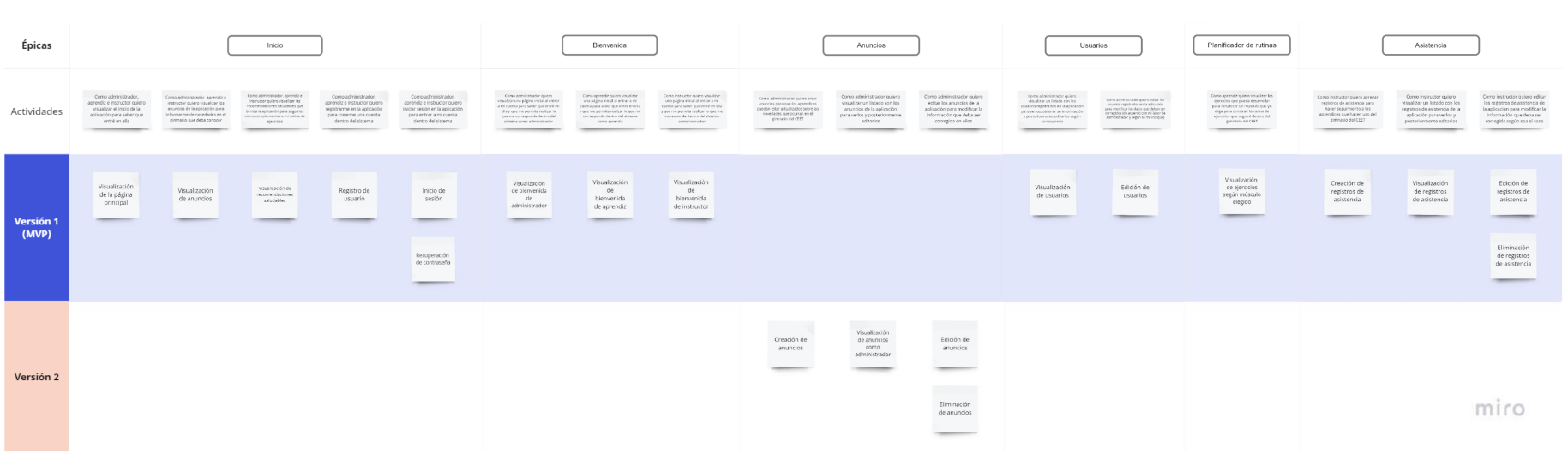






### **5.2.1. Mapa de historias de usuario (Móvil).**

Aquí se presenta el mapa de historias de usuario (user story mapping) aplicado cuando se desarrolló el proyecto para entornos móviles, el cual brinda una perspectiva general sobre los procesos que maneja el sistema:



## ***5.3. Administración de usuarios.***

Continuando con el proceso de descripción del sistema, este (como lo mencionamos anteriormente) se divide en tres grupos de actividades, separadas de tal forma que cada una se agrupe de acuerdo con los usuarios a los que les corresponda el grupo de tareas; las divisiones son: Administradores, Aprendices, Instructores; de esta manera se clasifican por roles las actividades a realizar por los actores del sistema.

De acuerdo a lo anterior, debemos aclarar qué personas se encargarán de lo relacionado a la gestión de usuarios: los que desarrollarán estas acciones serán los administradores del sistema, de tal forma que ellos podrán editar los datos de los usuarios del sistema; ellos serán los encargados de cambiar información de los usuarios, así como son los encargados de cambiarle el rol a estos últimos, de modo tal que todas las personas que vayan a usar el sistema se tienen que registrar dentro del aplicativo, y los usuarios que deban realizar actividades de gerencia (administradores) o los instructores del sistema, tendrán que hablar con las personas que figuran como administradores dentro del sistema para que estos se encarguen de cambiarles el rol dentro del aplicativo. Por otro lado, cualquier novedad respecto a los usuarios, que tengan que ver con alguna suspensión, expulsión o suceso que amerite bloquear el acceso al sistema por parte de un usuario, deberá ser atendido por un administrador, de tal forma que se le comente el caso y se actúe en consecuencia.

## ***5.4. Arquitectura del sistema.***

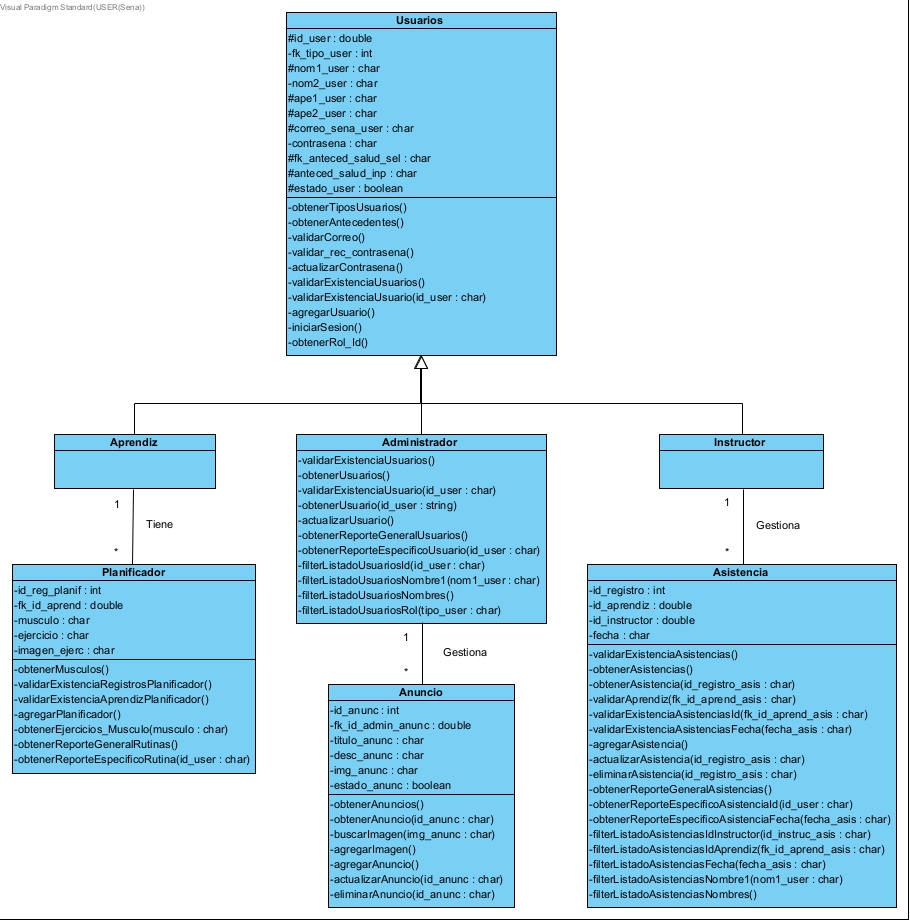
Ahora, le invitamos a conocer los distintos diagramas y componentes de arquitectura que sustentan el sistema, sus funciones y la forma en que opera:

### **5.4.1. Diagramas UML.**

En cuanto a estos diagramas, nos parece primordial que conozca los siguientes, enfocados al sistema y su forma de operar:

#### ***5.4.1.2. Diagrama de clases.***

El diagrama de clases es el siguiente:

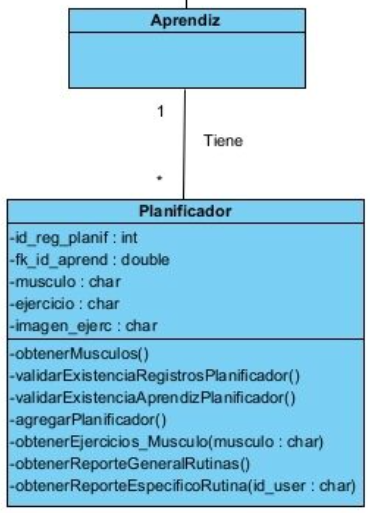


Ahora, se presenta en detalle cada clase:

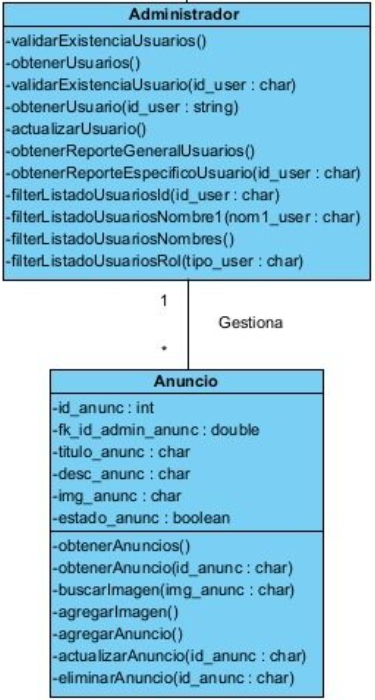
**Clase Usuarios:**

****

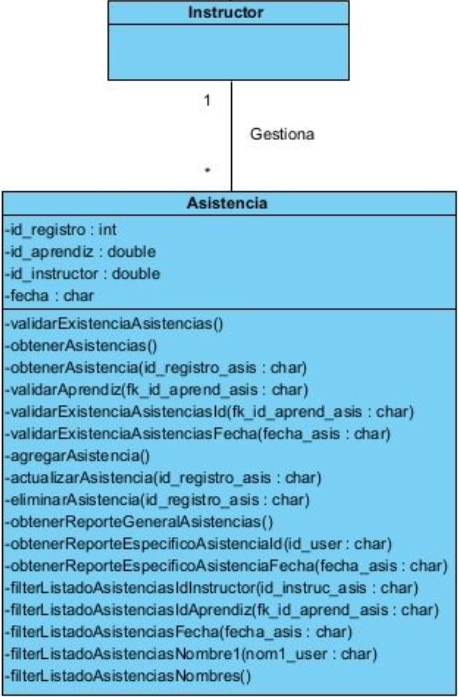
**Clases Actividades de aprendiz (Clase Aprendiz, Clase Planificador):**



**Clases Actividades de administrador (Clase Administrador, Clase Anuncio):**

****

**Clases Actividades de instructor (Clase Instructor, Clase Asistencia):**

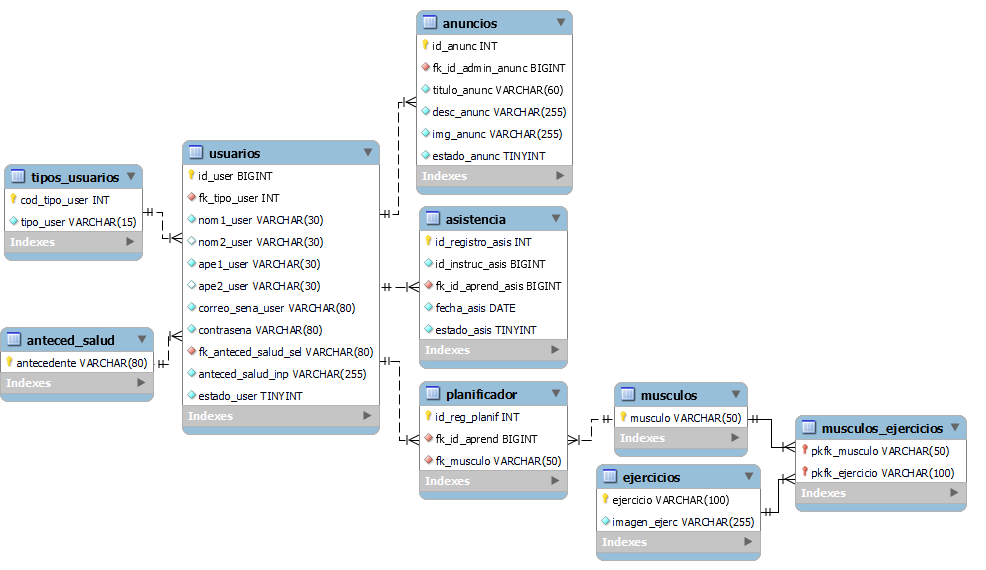
****

#### ***5.4.1.4. Diagrama de despliegue.***

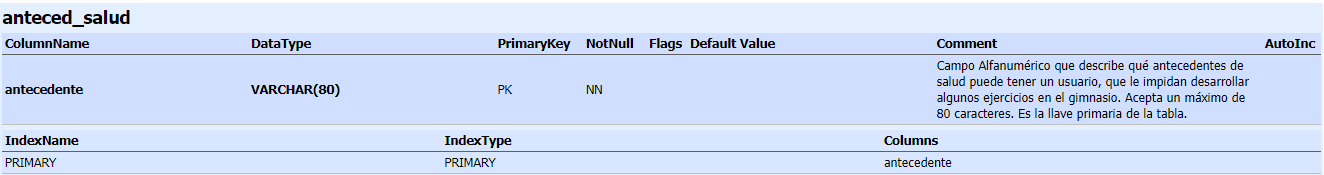
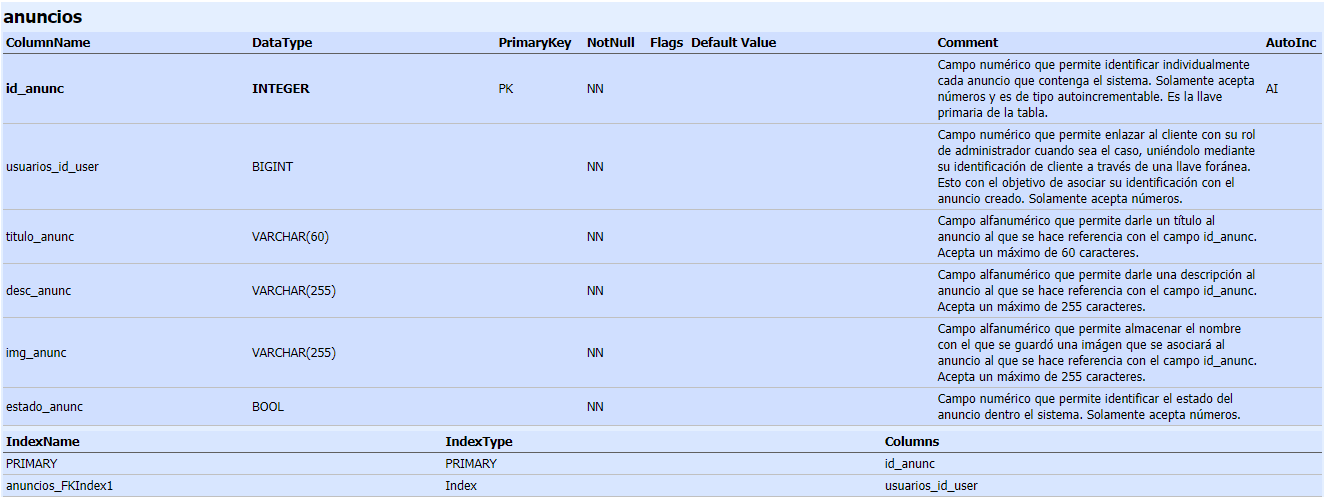
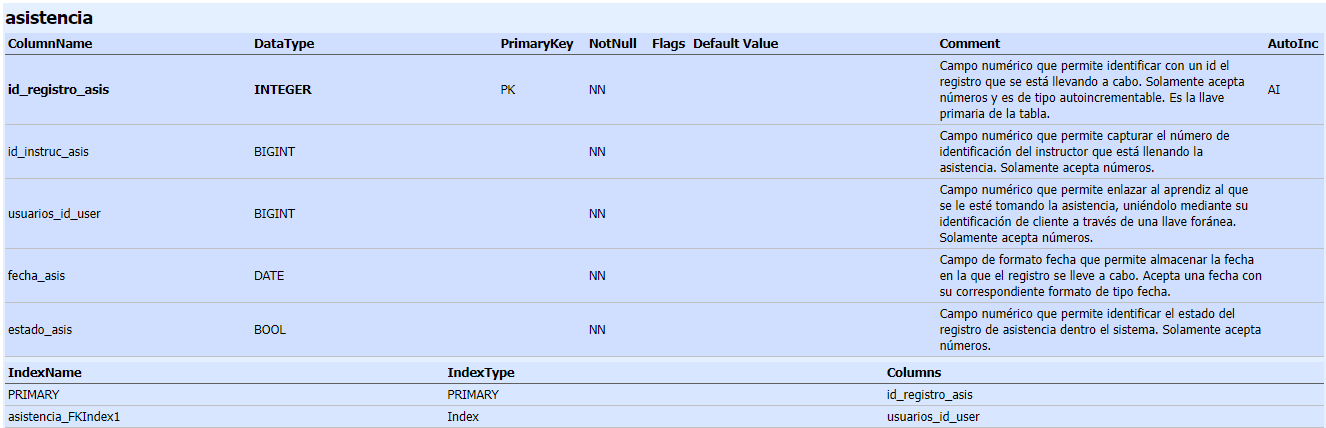
### **5.4.2. Base de datos.**

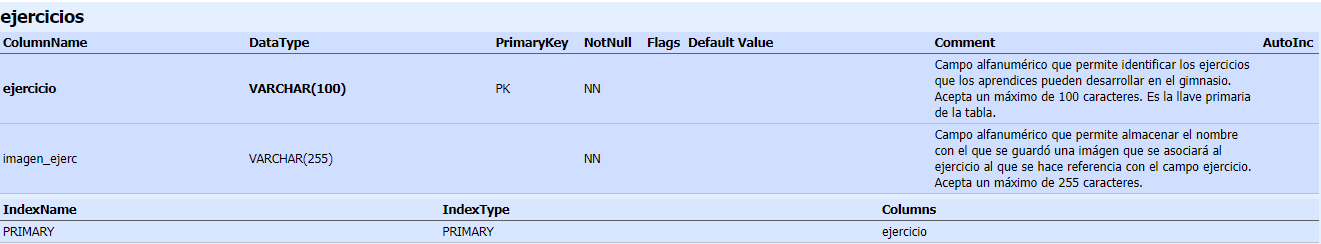
En cuanto a arquitectura de la base de datos, le invitamos a que conozca el modelo relacional del proyecto y el diccionario de datos de este:

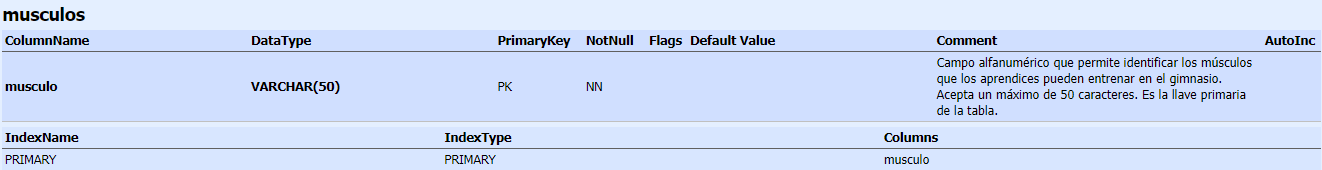
#### ***5.4.2.1. Modelo relacional.***

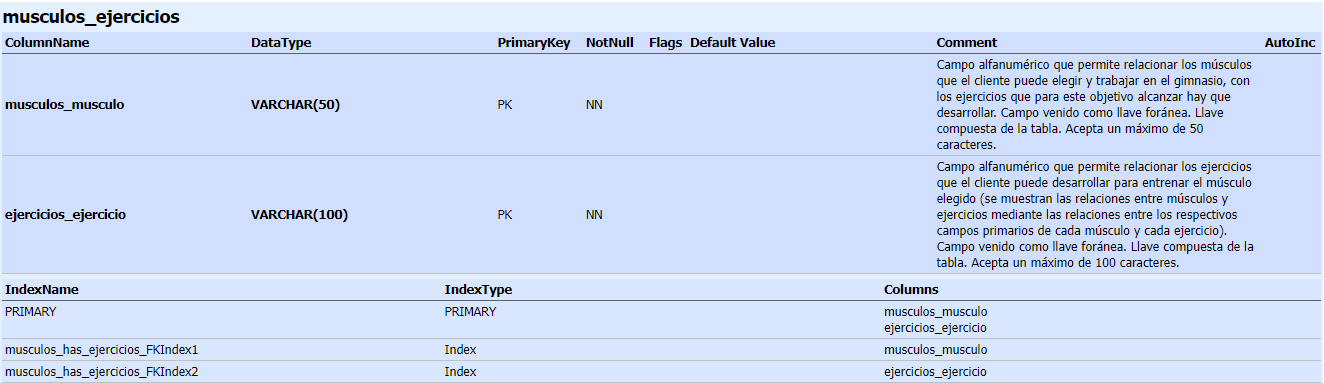
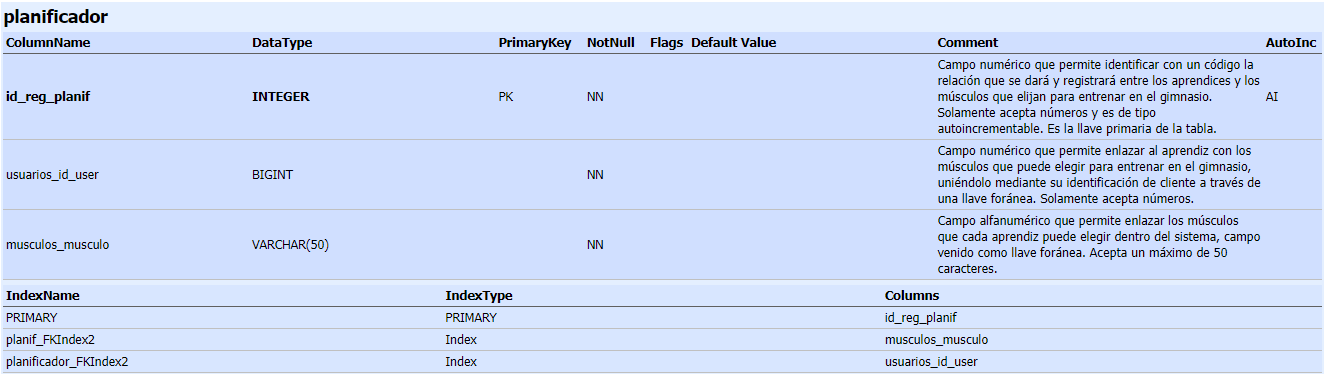


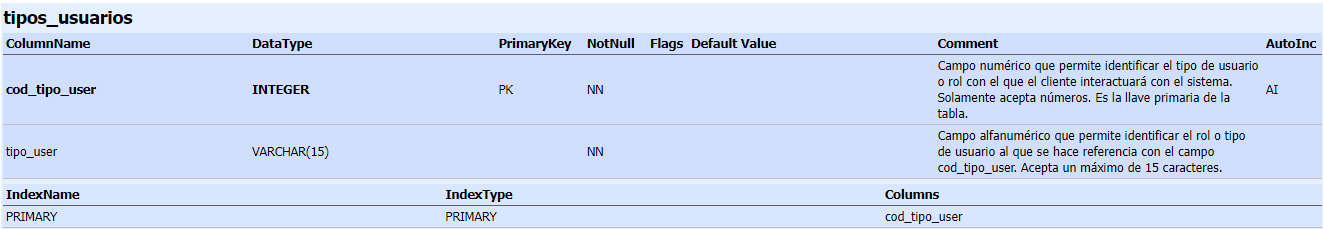
#### ***5.4.2.2. Diccionario de datos.***

Las siguientes ilustraciones corresponden a las descripciones de los elementos (entendidos como tablas y campos) del modelo relacional:







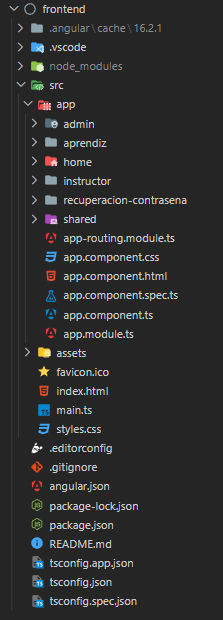




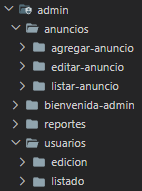
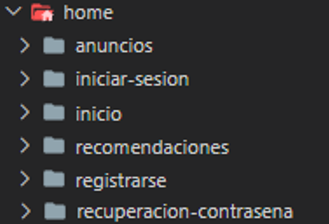
## ***5.5. Documentación del código fuente.***

En este punto, se le mostrará la estructura del código fuente, de tal forma que entienda cómo está organizado el sistema en cuanto a carpetas:

### **5.5.1. Estructura de carpetas - Frontend.**

La estructura general del Frontend es la siguiente:

La estructura de cada una de las subcarpetas dentro de “src/app” (o sea, las carpetas “admin”, “aprendiz”, “home”, “instructor”, “shared”) es la siguiente:



Cada una de las carpetas que se encuentran dentro de las anteriores subcarpetas tienen la siguiente estructura:



De tal modo que cada carpeta comprende un archivo de estilos (.css), un archivo html, un archivo spect.ts (para pruebas) y un component.ts (con la lógica del componente).

Ahora, explicando más a detalle las carpetas:

* Las carpetas “admin”, “aprendiz”, “instructor”, son las carpetas que contienen dentro los componentes de las actividades que cada uno de estos usuarios debe desarrollar dentro del sistema.
* La carpeta “home” contiene los componentes del incio del sistema (inicio general, anuncios, recomendaciones saludables, registro, inicio de sesión).
* La carpeta shared contiene componentes que se usan en, o conectan, con varios componentes, a continuación se explican más a fondo:
  + **Footer**: Componente del footer de las páginas.
  + **Navbar’s**: las carpetas que empiezan por navbar contienen las barras de navegación del sistema, teniendo en cuenta que la carpeta “navbar-general” contiene la barra de navegación inicial del sistema, las carpetas “navbar-log-administrador”, “navbar-log-aprendiz” y “navbar-log-instructor” contienen las barras de navegación para cada usuario del sistema cuando haya iniciado sesión y, por último, el componente “navbars” contiene las condicionales que le permiten al sistema determinar cuál barra de navegación se le debe mostrar al usuario.
  + **Service’s**: Las carpetas que empiezan por “service”, menos la carpeta “service-session\_storage”, contienen los servicios que le permiten al sistema conectar con la API, de tal forma que contienen los nombres referentes a la clase a la que responden para realizar sus funcionalidades de conexión con el Backend. En el caso de la carpeta “service-session\_storage”, este componente contiene un conjunto de métodos que permiten gestionar lo referente a las variables de sesión del sistema, de tal forma que se determinó correcto dejar disponible en un solo lugar los métodos para accionar en conjunto con las variables de sesión, siendo necesario solo llamar métodos para realizar las ejecutar las acciones esperadas.

### **5.5.2. Relaciones entre componentes.**

Este sistema, como explicamos anteriormente, contiene servicios dentro de la carpeta “shared”, estos nos permiten crear las conexiones necesarias con la API, sirviendo como los principales puentes entre el Backend y el Frontend; razón por la cual se hace importante tenerlos en cuenta para cuando se piense realizar modificaciones al sistema. A modo de explicación, a continuación, se describe brevemente de qué se encarga cada servicio que conecta con la API:

* **service-anuncio**: Se encarga de lo relacionado a los procesos CRUD de los anuncios del sistema.
* **service-asistencia**: Se encarga de lo relacionado a los procesos CRUD de los registros de asistencia del sistema.
* **service-planificador**: Se encarga de lo relacionado a los procesos CRUD del planificador del sistema.
* **service-usuario**: Se encarga de lo relacionado a los procesos CRUD de los usuarios del sistema.

Por otro lado, a continuación, se describe el contenido de las carpetas o componentes referentes a los usuarios a los que corresponden de acuerdo con sus roles dentro del sistema:

* **admin**: contiene carpetas:
  + **anuncios**: Conteniendo los componentes:
    - *agregar-anuncio*: Componente relacionado a la creación de anuncios para el sistema.
    - *editar-anuncio*: Componente relacionado a la edición de anuncios del sistema.
    - *listar-anuncio*: Componente relacionado al listado de anuncios mostrado a los administradores del sistema.
  + **bienvenida-admin**: Componente relacionado a la página mostrada apenas el usuario inicia sesión dentro del sistema.
  + **reportes**: Componente relacionado a los reportes que pueden generar estos usuarios dentro del sistema.
  + **usuarios**: Conteniendo los componentes:
    - *edición*: Componente relacionado a la edición de usuarios del sistema.
    - *listado*: Componente relacionado al listado mostrado a los administradores con los usuarios del sistema.
* **aprendiz**: contiene carpetas:
  + **bienvenida-aprendiz**: Componente relacionado a la página mostrada apenas el usuario inicia sesión dentro del sistema.
  + **Planificador**: Componente relacionado al planificador de rutinas con el que cuentan los aprendices del sistema.
* **instructor**: contiene carpetas:
  + **asistencia**: Conteniendo los componentes:
    - *agregar-asistencia*: Componente relacionado a la creación de registros de asistencia en el sistema.
    - *editar- asistencia*: Componente relacionado a la edición de registros de asistencia del sistema.
    - *listar- asistencia*: Componente relacionado al listado de registros de asistencia mostrado a los instructores del sistema.
  + **bienvenida-instructor**: Componente relacionado a la página mostrada apenas el usuario inicia sesión dentro del sistema.

## ***5.6. Entorno de producción.***

Ahora, se van a describir temáticas referentes al despliegue del sistema y a su entorno de producción.

### **5.6.1. Descripción de las plataformas.**

En el caso de este proyecto de formación, se ejecutó el despliegue de tal forma que los componentes del sistema quedaron alojados en tres diferentes servicios con tres diferentes hosts: La base de datos quedó alojada en un servidor de HelioHost, la API se alojó en Render, y el Frontend se alojó en un servicio de Netlify.

Cada uno de estos servidores tienen sus pros y contras, los cuales se limitan aún más teniendo en cuenta que se desplegó usando los planes gratuitos de las plataformas, por lo que hay ciertas condiciones que se deben cumplir para que el sistema siga siendo funcional en la nube. Entrando en detalles, la base de datos cuenta con un máximo de memoria tanto de disco duro como de memoria RAM, pero contiene el correspondiente gestor de bases de datos para MySQL, con lo cual basta para hacer funcionar la base de datos desde la nube. En el caso de la API en Render, esta realiza eficazmente sus actividades, tiene una muy buena integración con Git y GitHub, así como con NodeJS lo que permite dejar funcionando nuestra API de la forma esperada (ya que está construida con NodeJS, y además se facilita el despliegue del componente por medio del uso del repositorio de Git); pero esta herramienta tiene una gran desventaja, y es que el servidor se cierra cada cierto tiempo de inactividad, y al querer volver a usarlo hay que realizarle un reinicio al servicio, por lo que hay que esperar a que vuelva a ejecutarse; este es un gran problema teniendo en cuenta que si usuarios reales acceden al sistema, pueden llegar a sufrir inconvenientes al usar el software si la API entra en inactividad, teniendo que esperar un tiempo prudencial mientras el sistema se reestablece, por lo que aclaramos que el despliegue desarrollado al momento se tiene pensado para que sea tipo prueba, no es recomendable ponerlo a circular tal y como está, evitando que sea usado por usuarios reales. Por otro lado, el Frontend alojado en Netlify, consideramos que está cumpliendo con su propósito a cabalidad, la herramienta permite usar correctamente el sistema no importando que haya sido construido con Angular; integra correctamente NodeJS y además repositorios de Git y GitHub, considerados elementos que le dan un plus al host para considerar seguirlo usando en un futuro.

De acuerdo a lo anterior, es muy importante tener en cuenta que esperamos que, si en algún punto se continúa con el sistema, será imperativo definir donde se alojará el mismo de forma definitiva, buscando que los usuarios tengan una buena, o mejor, una excelente experiencia usando el sistema; ya que hay algunos problemas con los hosts que usamos actualmente que, si bien permiten desplegar el sistema, no es rentable a futuro, por lo que habrá que indagar en este tema con más recursos para determinar en qué host o hosts de desplegarán los componentes del sistema cuando se espere recibir grandes cantidades de usuarios en el sistema.

Aclarado lo anterior, le invitamos a acceder al siguiente link que contiene el aplicativo web ya desplegado, no sin antes solicitarle que tenga en cuenta las limitaciones comentadas sobre la API referente al cierre del servicio, y los componentes de almacenamiento de la base de datos: <https://gymsenapp.netlify.app/>

### **5.6.2. Acuerdos a nivel de servicio.**

Con relación a este término, nos amparamos bajo los lineamientos de cada empresa en cuanto a lo que respecta a los planes y servicios acordados con las mismas, de tal forma que estas empresas dentro de los planes ofertados nos brindan las métricas y límites que el plan escogido abarca, permitiéndonos tener claras todas aquellas directrices bajo las cuales nos prestan el servicio hosting. Lo anteriormente mencionado aplica también para lo relacionado a la gestión de sucesos inesperados que le ocurran en cualquier momento al sistema, proyectando que en caso de que alguna falla de este tipo ocurra, se deben tener presentes los canales de atención de los hosts (si es que el problema tiene que ver con ellos), de tal forma que se debe asegurar que se tendrá una directa forma de comunicación con dichas empresas ante cualquier eventualidad que pueda surgir.

# **6. Anexos**

Como elementos adicionales al presente documento, le presentamos los siguientes documentos y complementos anexos, a fin de que obtenga (según lo requiera) un mayor detalle de los elementos mostrados en este manual:

* Imágenes del diagrama BPMN del sistema, en los archivos llamados “BPMN\_Administrador.png”, “BPMN\_Aprendiz.png”, “BPMN\_Instructor.png” (cada una correspondiente a cada rol del sistema).
* Archivo con los requerimientos del sistema, en el archivo llamado “RequerimientosGSA.docx”.
* Imagen con el diagrama de casos de uso del sistema, en el archivo llamado “CasosDeUsoGSA.jpg”.
* Archivo con los casos de uso extendidos del sistema, en el archivo llamado “CasosDeUsoExtendidoGSA.docx”.
* Archivo con el mapa de historias de usuario para el proyecto móvil, en el archivo llamado “userStoryMapping.pdf”.
* Imagen con el diagrama de clases del sistema, en el archivo llamado “DiagramaClasesGSA.jpg”.
* Imagen con el diagrama de despliegue o distribución del sistema, en el archivo llamado “DiagramaDistribucionGSA.png”.
* Archivo con el modelo relacional del sistema, en el archivo llamado “MRGymSenApp.png”.
* Archivo con el diccionario de datos del sistema, en el archivo llamado “DiccionarioDatosGSA.html”.
* Repositorio de GitHub con la documentación y lo realizado trimestre a trimestre en cuanto a entregas del software en la dirección: <https://github.com/marcooolfoxygamer/Proyecto-GymSenApp>.
* Repositorio de GitHub con el Frontend del sistema, en la siguiente dirección: <https://github.com/marcooolfoxygamer/frontend_web_GSA>.
* Repositorio de GitHub con el Backend del sistema, en la siguiente dirección: <https://github.com/marcooolfoxygamer/backend_GSA>.

# **7. Glosario**

En este documento se utilizan los siguientes términos abreviados:

* GymSenApp: Nombre del software

# **8. Referencias**

El presente documento se basa en los siguientes referentes:

* IEEE 1063 - Standard for Software User Documentation. Recuperado de: <https://ieeexplore.ieee.org/document/974401>
* glucio. (2020, Agosto 26). Manuales de Usuario. IEEE 1063 2001. YouTube. <https://www.youtube.com/watch?v=1VMPM0hYzak>
* Luis Velasco. (2015, Agosto 22). Manual tecnico y Manual de usuario - Ingenieria del software. YouTube. <https://www.youtube.com/watch?v=LFfbxJK46l0>