

# UNIVERSIDAD DEL BÍO-BÍO

Facultad de Ciencias Empresariales  
Departamento Sistemas de Información



UNIVERSIDAD DEL BÍO-BÍO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO DE EJECUCIÓN EN  
COMPUTACIÓN E INFORMÁTICA

“Desarrollo de Sistema Web y Aplicación Móvil  
Android para el apoyo a la gestión y progresión  
Deportiva Gimnasio Esparta.”

Alumnos: Nicolás Llanos M.

Profesor Guía: Karina Rojas C.

Profesor Informante: Elizabeth Grandón T.

Concepción, Abril 2019

## Índice

Introducción	2
Índice de ilustraciones	4
Capítulo 1: Definición de la empresa o institución	5
Capítulo 2: Definición proyecto	12
Capítulo 3: Software para el desarrollo	16
Capítulo 4: Análisis - Especificación de requerimientos de Software	18
Capítulo 5: Análisis de factibilidad	30
Capítulo 6: Análisis – Casos de uso – DFD y modelamiento de datos	36
Capítulo 7: Diseño	96
Capítulo 8: Pruebas	107
Capítulo 9: Plan de implementación y puesta en marcha	115
Capítulo 10: Conclusiones	118
Bibliografía	120
Anexos	122

---

## **INTRODUCCIÓN**

---

Este Informe se presenta para dar conformidad a los requisitos exigidos por la Universidad del Bío-Bío para optar al título de Ingeniero de Ejecución en Computación e Informática y para dar a conocer los resultados del proyecto de Software “Desarrollo de Sistema Web y Aplicación Móvil Android para el apoyo a la gestión y progresión Deportiva Gimnasio Esparta”, el que hará aportes a la administración y al progreso deportivo de sus respectivos clientes a través del uso de Tecnologías de la Información.

El contenido presentado en este documento lo conforman 12 capítulos detallados a continuación:

El capítulo 1 presenta los datos que identifican a la Empresa, como la descripción de esta, la representación del área de estudio y las problemáticas que presenta. En el capítulo 2 se establecen los objetivos del proyecto (generales y específicos), el ambiente de Ingeniería de Software (Metodología de desarrollo del proyecto de software), definiciones, siglas y abreviaciones que se ocuparán en el documento. El capítulo 3 define los atributos y funcionalidades del Software que se obtendrá como resultado del proyecto. Contiene elementos como alcances, objetivos del sistema, interfaces de hardware, interfaces de usuario, interfaces de software, entre otros, y los requerimientos propiamente tal.

En el capítulo 4 se realiza el estudio de factibilidad del proyecto (factibilidad técnica, operativa y económica).

En el capítulo 5 se exponen los diagramas idóneos al Software y al flujo de Información de este. Define elementos como Diagramas de Base de Datos, especificación de Casos de Uso y Diagrama del Proceso de Negocio una vez implementado el Sistema. El capítulo 6 contiene el Diseño Físico de la Base de Datos, diseño interfaz y navegación (Mockups).

En el capítulo 7 e incluyen módulos, reglas, funciones, etc. creadas o modificadas en el framework.

El capítulo 8 engloba todas las pruebas que serán aplicadas al Software. En el capítulo 9 se definen estrategias/planes para la implementación y puesta en marcha del Sistema.

En el capítulo 10 se dan a conocer las Conclusiones obtenidas al finalizar el proyecto. El capítulo 11 muestra la Bibliografía utilizada para apoyar el Desarrollo del proyecto. Por último, en el capítulo 12 se adjuntan anexos.

---

## ÍNDICE DE ILUSTRACIONES

---

Ilustración 1: Estructura Organizacional Gimnasio Esparta.....	6
Ilustración 2: BPMN Proceso de Inscripción de Clientes .....	10
Ilustración 3: BPMN Proceso de Registro de Evaluación/ progresión Física.....	11
Ilustración 4: BPMN Proceso de Registro de Asistencia de Clientes .....	11
Ilustración 5: Diagrama Caso de Uso Cliente, Nutricionista, Profesor.....	38
Ilustración 6: Diagrama Caso de Uso Administrador, Profesor .....	39
Ilustración 7: Modelamiento de Datos .....	96
Ilustración 8: Diseño Físico de la Base de Datos.....	96
Ilustración 9: Diseño de Interfaz Sistema Web (Login) .....	97
Ilustración 10: Diseño de Interfaz Área de Trabajo Sistema Web .....	98
Ilustración 11: Diseño de Interfaz Aplicación Móvil (Login) .....	100
Ilustración 12: Diseño de Interfaz Aplicación Móvil (Menú Principal).....	101
Ilustración 13: Diseño de Interfaz Aplicación Móvil (Rutinas, Ejercicios, Información Nutricional, Progresión Física, Información de Membresía, Asistencias .....	102
Ilustración 14: Jerarquía de Menú Sistema Web.....	105
Ilustración 15: Jerarquía de Menú aplicación móvil android.....	106

---

## **CAPÍTULO 1: DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN**

---

### **1.1 Descripción de la empresa**

El presente proyecto se desarrollará para el Gimnasio Esparta, ubicado en Manuel Rodríguez 2150, Chiguayante, Bío-Bío, Chile. Dicha dependencia está destinada a realizar diversos tipos de actividades físicas guiadas por un profesional a cargo. Cabe mencionar que diariamente asisten dos entrenadores personales.

Se cuenta con diversos tipos de servicios: sala de musculación, sala cardiovascular, baños, recepción y área para clases grupales como zumba, pilates, reductivo abdominal, entre otros. Estas actividades están dirigidas a personas de todas las edades, dependiendo de sus respectivos objetivos, ya sea recreativo, estético, de salud o de perfeccionamiento deportivo.

El gimnasio Esparta percibe sus ingresos a través del proceso de inscripción, los que son alrededor de 100 personas. Se realiza control de asistencia de los clientes de forma manual, la cual es realizada por el profesor o profesional a cargo. El sistema de recaudación se realiza en efectivo y no se tiene la intención de su modificación.

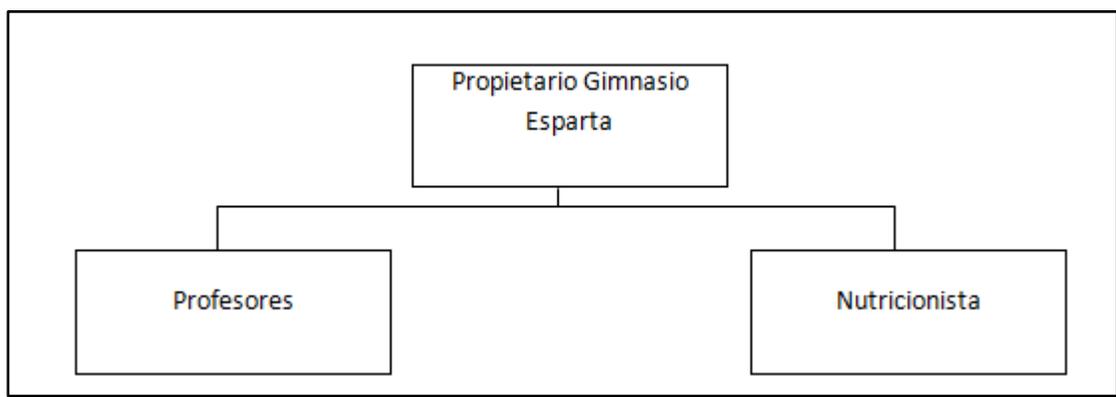
### **1.2 Descripción del área de estudio**

Los objetivos de la Empresa son los siguientes:

- Promover el ejercicio con un ambiente grato.
- Brindar un servicio integral el cual se traduce tanto en el cuidado de la salud, la actividad física, la estética, como en la recreación de cada uno de los clientes, buscando de esta manera realizar todos los esfuerzos necesarios para poner en

primer lugar a la persona y brindarle la mejor comodidad dentro de la organización.

- Crear una oferta de programas que responda con calidad a la necesidad de todos aquellos interesados en su desarrollo personal.
- Captar la mayor cantidad de clientes.
- Motivar e instruir a los Empleados.



**Ilustración 1:** Estructura Organizacional Gimnasio Esparta, Elaboración propia.

La ilustración representa la estructura organizacional de la Empresa, lo cual está compuesta por Profesores del área deportiva, un experto del área de la Salud (Nutricionista) y por el Propietario o dueño del Gimnasio.

**Dueño/Propietario:** El gimnasio cuenta con un propietario, sus funciones se describen a continuación:

**1) Gestión del gimnasio Esparta:**

- Verificar rentabilidad, establecer políticas de mejoramiento continuo en los programas para los respectivos clientes.
- Disponer de un equipamiento óptimo según las necesidades establecidas, técnicas de entrenamiento y acondicionamiento físico.

- Inscripción de clientes.
  - Control de Deudas.
  - Gestionar Pagos.
- 2) **Contratar o terminar contratos:** Con Empleados según las necesidades con las que se encuentre la dependencia Deportiva.
  - 3) **Determinar nuevos focos de Mercado:** Establecer estrategias de Marketing para la incorporación de nuevos clientes.
  - 4) **Orientación al cliente:** Función fundamental es mantener a los clientes satisfechos acogiendo consultas, sugerencias, y eventuales reclamos para dar pronta solución a cualquier insatisfacción por parte de ellos.
  - 5) **Establecer planes:** Instaurar promociones atractivas que atraigan al público para su participación.
  - 6) **Rol de Profesor:** Realizar evaluaciones generales de admisión a los clientes, supervisar a los clientes que se encuentran entrenando, diseñar rutinas, llevar un control del progreso de las personas inscritas, instruir o capacitar a la gente participante en la elaboración de sus respectivas rutinas.

**Profesores:** El gimnasio cuenta con profesores o entrenadores deportivos, sus funciones se describen a continuación:

- 1) **Funciones principales:** Realizar evaluaciones generales a los clientes, supervisar a los clientes que se encuentran entrenando, diseñar rutinas, llevar un control del progreso de las personas inscritas, instruir o capacitar a la gente participante en la elaboración de sus respectivas rutinas.

**Nutricionista:** El gimnasio cuenta con un profesional del área de la salud (nutricionista), sus funciones se describen a continuación:

- 1) Contribuye a conocer hábitos de alimentación, horarios, preferencias alimentarias, entre otras.
- 2) Ayuda a calcular la cantidad de calorías y nutrientes consumidos en la dieta de los clientes, lo que posibilita la adecuación de estos según los requerimientos personales.
- 3) Permite la creación de un plan de alimentación y menú personalizado, con recetas de acuerdo a las necesidades de cada clientes.
- 4) Realiza educación sobre: etiquetado nutricional de los alimentos, alimentos dietéticos o light, nutrientes: macro y micronutrientes, porciones de alimentos, entre otros
- 5) Facilita la entrega de material de apoyo para el tratamiento nutricional de la persona(recetarios de cocina, listas de supermercado, consejos de alimentación saludable, entre otros)

### **1.3 Descripción de la problemática**

Las diversas problemáticas que presenta el Gimnasio Esparta de Chiguayante apuntan principalmente a que no cuenta con un sistema de información computacional que facilite el registro de pagos e inscripción de clientes, control de acceso, tiempo invertido en entrenamiento, control de activos fijos, información de clientes (índices corporales, información nutricional, entre otros), rutina de cada uno de sus clientes, etc.

#### **- Respecto del registro de información:**

Actualmente el registro de la información se realiza de forma manual, lo que implica ingresar todos los datos en un “libro de registros”. Esto a su vez genera dificultad para la organización de la información, haciendo el proceso lento lo que no permite al sistema ser eficiente a la hora de consultar los registros. Además, el hecho de llevar

toda la información en el libro implica un alto riesgo, ya que no se cuenta con un sistema de respaldo que permita consultar en caso de pérdida de los registros

- **Respecto del registro de clientes y pagos:**

Se ha evidenciado que el escaso control termina generando pérdidas en las utilidades y en el servicio propiamente tal. De acuerdo a lo anterior los principales nudos críticos que se observan son el ingreso de los clientes que asisten y hacen uso de los servicios prestados sin pago, ya que como no se cuenta con registros eficientes no se tiene mayor claridad del inicio y/o término de los planes de sus clientes.

- **Respecto a las solicitudes de los Clientes:**

Actualmente asisten durante el día dos entrenadores personales, lo que implica que no pueden atender las consultas de todos los clientes en un momento determinado.

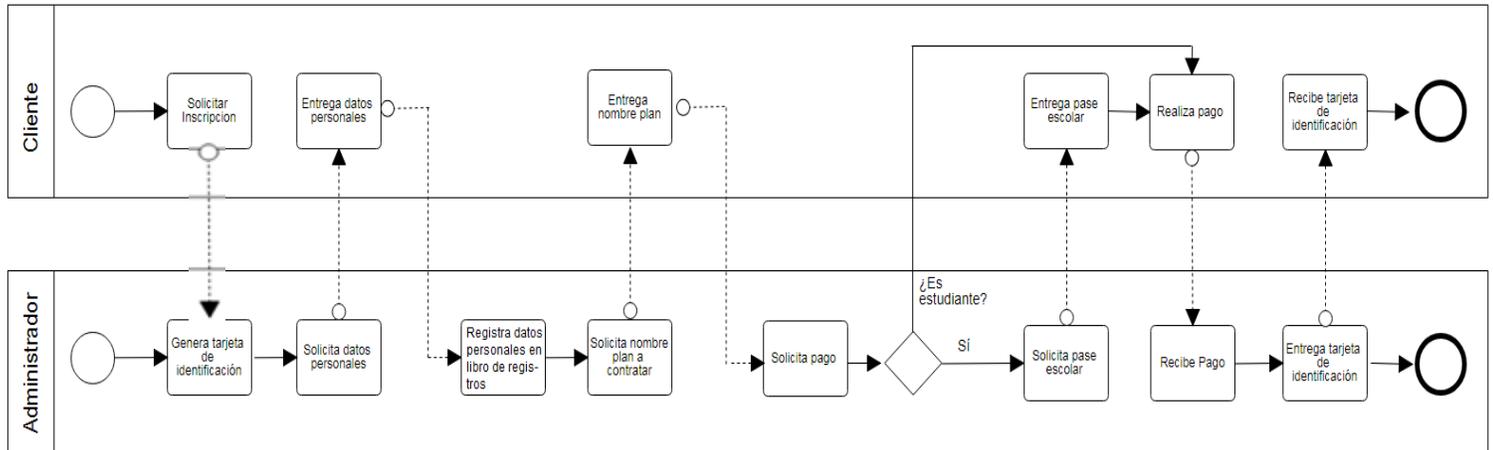
- **Respecto a la generación de informes:**

Para medir la evolución de los clientes inscritos en los programas de entrenamiento se cuenta con fichas de cada usuario que describen el peso, talla, índice de masa corporal, perímetros, entre otros. Las que deben ser almacenadas manualmente, en cuadernos de registro o en formato digital (Word). Lo anterior dificulta la medición progresiva de los usuarios debido a la búsqueda engorrosa que se debe hacer, por tanto no es de fácil acceso ni para el dispositivo ni para los usuarios. El mismo formato se usa para asignar rutinas a los usuarios, situación que se complejiza cada vez que se debe actualizar la información mes a mes o de acuerdo al criterio del profesional que se encuentre a cargo.

- **Respecto al control de activos fijos:**

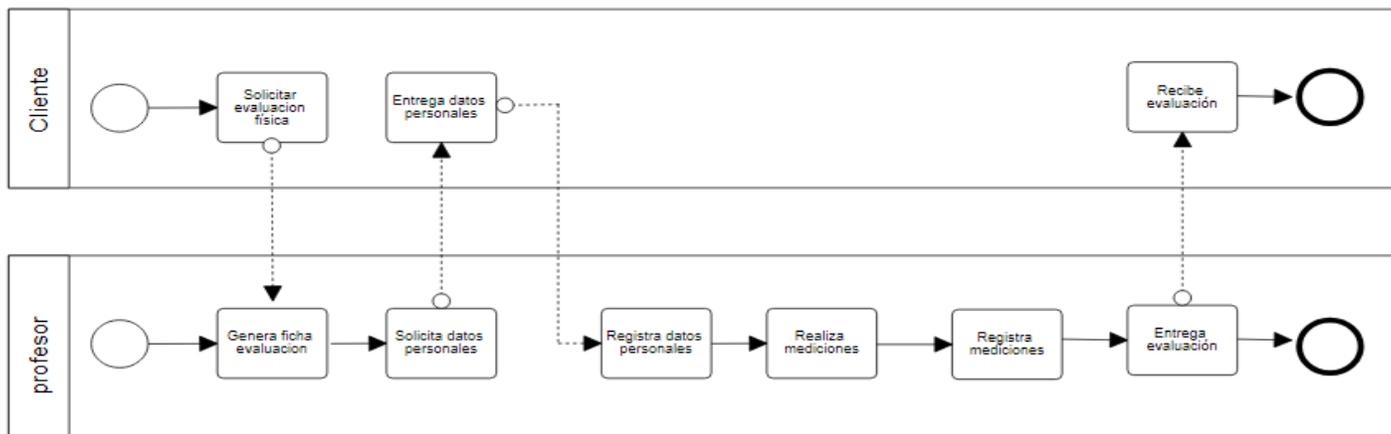
En la dependencia no se cuenta con un inventario de activo fijo lo que impide llevar

un control de las máquinas e implementos deportivos, por lo cual se desconoce si hay pérdidas por deterioro, bajas o robos del activo. Esto podría generar a largo plazo la pérdida del prestigio y disminución en las utilidades del negocio.



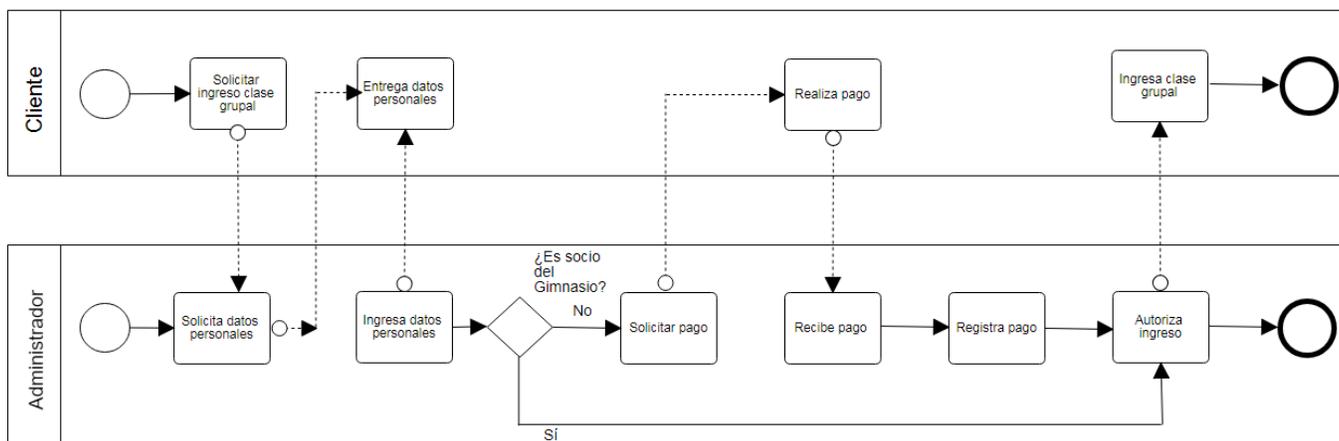
**Ilustración 2:** BPMN Proceso de Inscripción de Clientes, Elaboración propia (2019).

La ilustración representa el flujo de información al realizar el proceso de inscripción de un nuevo Cliente al gimnasio.



**Ilustración 3:** BPMN Proceso de Registro de Evaluación/ progresión Física, Elaboración propia (2019).

La ilustración representa el flujo de información al realizar el proceso de registro de evaluaciones o progresiones Físicas.



**Ilustración 4:** BPMN Proceso de Registro de Asistencia de Clientes, Elaboración propia (2019).

La ilustración representa el flujo de información al realizar el proceso de registro de asistencias al gimnasio.

---

## **CAPÍTULO 2: DEFINICIÓN PROYECTO**

---

### **2.1 Objetivos del proyecto**

#### **Objetivo General**

Desarrollar un sistema web complementado con una aplicación móvil android que permita realizar las tareas administrativas y llevar un seguimiento deportivo con el fin de apoyar la gestión eficiente de los recursos, el funcionamiento, la toma de decisiones y el progreso de las respectivas metas de cada cliente que asiste al gimnasio Esparta.

#### **Objetivos Específicos**

- Analizar la situación actual del gimnasio Esparta, para así plantear soluciones a sus respectivos problemas.
- Determinar los requerimientos para el funcionamiento del sistema.
- Desarrollar la base de datos.
- Desarrollar un sistema web que facilite los procesos de inscripción, control, recursos de la empresa y progresiones deportivas de los clientes.
- Desarrollar una aplicación móvil android de apoyo para los clientes del gimnasio.
- Probar el sistema en un ambiente de datos reales.

### **2.2 Ambiente de Ingeniería de Software**

Tras un análisis realizado previamente se considera utilizar la metodología de Desarrollo Iterativo e Incremental debido a las características que presenta. La idea principal es basarse en la idea de desarrollar una implementación inicial, mostrárselo al cliente y desarrollar sucesivas versiones hasta obtener el sistema requerido. Cada incremento o versión del sistema incluirá una nueva funcionalidad. Normalmente las primeras versiones contendrán las funcionalidades más importantes o más urgentes. Por

otra parte desarrollar el software incrementalmente hace que sea más fácil y económico realizar cambios.

### **Características de la Metodología**

**Metodología de Desarrollo Iterativo e Incremental:** “Es un proceso de software creado en respuesta a las debilidades del modelo tradicional de Cascada.

Básicamente este modelo de desarrollo, es un conjunto de tareas agrupadas en pequeñas etapas repetitivas (iteraciones). Cada versión tiene sentido para el cliente.

Lo que busca es que en cada iteración son los componentes que logren evolucionar el producto dependiendo de los completados de las iteraciones anteriores, agregando más opciones de requisitos y logrando así un mejoramiento mucho más completo.

Los pasos claves en el proceso son iniciar con una implementación simple de los requerimientos del sistema, e iterativamente mejorar la serie evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema. El proceso en sí mismo consiste en crear una versión del sistema. La meta de esta etapa es crear un producto con el que el usuario pueda interactuar, y por ende retroalimentar el proceso siguiente.” (Rivas, 2010).

### **Ventajas Metodología de Desarrollo Iterativo e Incremental**

“Las ventajas de la Metodología son las siguientes:

- Resolución de problemas de alto riesgo en tiempos tempranos del proyecto.
- Visión de avance en el desarrollo desde las etapas iniciales del desarrollo.

- Obtención del feedback del usuario lo antes posible, para orientar el desarrollo al cumplimiento de sus necesidades y realizar todas las adaptaciones identificadas para cumplir con los objetivos planteados.
- Menor tasa de fallo del proyecto, mejor productividad del equipo, y menor cantidad de defectos, según demuestran estudios realizados sobre proyectos que han aplicado esta técnica.
- No hace falta que los requisitos estén totalmente definidos al inicio del desarrollo, sino que se pueden ir refinando en cada una de las iteraciones.
- Permite manejar la complejidad del proyecto, apuntando a la resolución de los problemas por partes.
- El aprendizaje y experiencia del equipo iteración tras iteración, mejora exponencialmente el trabajo, aumenta la productividad y permite optimizar el proceso en el corto plazo.
- El trabajo iterativo deja una experiencia en el equipo que permite ir ajustando y mejorando las planificaciones, logrando menores desvíos en la duración total del proyecto.” (Cruz, 2007).

### 2.3 Definiciones, Siglas y Abreviaciones

**HTML:** (HyperText Markup Language), Lenguaje de Marcas de Hipertexto.

**CSS:** (Cascading Style Sheets), Hojas de estilo cascada para el diseño de interfaces.

**UML:** Lenguaje Unificado de Modelado para la realización de los casos de Usos correspondientes para los requerimientos funcionales.

**PHP:** Término que será utilizado para referirnos al lenguaje de programación de propósito general de código del lado del servidor diseñado para el desarrollo web de contenido dinámico.

**Mysql:** Palabra utilizada para dar mención a un sistema de gestión de bases de datos relacional.

**Framework:** Concepto utilizado para dar alusión a un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación.

**Laravel:** Término que será ocupado cuando demos mención al framework utilizado en el desarrollo del proyecto de software.

**Javascript:** Acrónimo para hacer referencia al lenguaje de programación que permite realizar actividades complejas en un sistema web.

**CASE:** Es un acrónimo para Computer-Aided Software Engineering, y lo utilizaremos para referirnos a un conjunto de herramientas que nos facilitaran el desarrollo del software.

---

### **CAPÍTULO 3: SOFTWARE PARA EL DESARROLLO**

---

Para el desarrollo del Sistema se definieron diversas alternativas de software a utilizar, el cual se listarán a continuación:

- Entornos de desarrollo Web: Xampp, WampServer.
- Gestores de Base de Datos: MySQL Workbench, PostgreSQL, Sqlite, SQL Server
- Framework: CodeIgniter, Symfony 4, Laravel, Yii.
- Software para el desarrollo móvil Android: Android Studio, B4A.

Se analizaron las diversas opciones para cada necesidad y se tomaron las siguientes decisiones con respecto al software a utilizar:

- Entorno de desarrollo Web: Se utilizará Xampp debido a que el desarrollador ha utilizado Xampp en proyectos anteriores de desarrollo web.
- Gestor de Base de Datos: Luego de analizar entre MYSQL Workbench y SQL Server, ya que el desarrollador ha trabajado con estos gestores de base de datos, se decidió optar por MYSQL Workbench, ya que ofrece la posibilidad de diseñar las bases de datos de forma visual (más sencilla).
- Framework: Se tomó la decisión de trabajar con Laravel, ya que el desarrollador tienen experiencia con esta herramienta, sin embargo, contaba con conocimientos en el framework CodeIgniter. Con respecto al resto de herramientas mencionadas no se contaba con la experiencia o conocimientos.
- Software para el desarrollo Móvil: Se analizaron las diversas alternativas, sin embargo se decidió optar por Android Studio debido a que cuenta con poderosas herramientas para el desarrollo móvil, además su lenguaje es JAVA, por lo que se cuenta con los conocimientos necesarios para construir aplicaciones móviles. Por otra parte, B4A utiliza el lenguaje BASIC, por lo cual no se tiene conocimiento por parte del desarrollador.

Por último, se trabajará con GITHUB, lo cual permite almacenar en la nube el código generado por parte de los desarrolladores y llevar un control de los cambios que ambos han proporcionado.

A continuación se describe el conjunto de software seleccionado para el desarrollo del sistema, junto con sus versiones:

- Xampp 3.1.0: “WampServer es un entorno de Desarrollo Web para Windows con el que se puede crear aplicaciones Web con Apache, PHP y Base de Datos MySQL. También incluye PHPMyAdmin y SQLiteManager para manejar las Bases de Datos”. (Ledesma, 2013)
- PHP Script Language 5.4.2: “PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica”. (Camargo, 2011)
- MySQL Workbench 8.0: “MySQL Workbench es una poderosa herramienta que ofrece la posibilidad de diseñar base de datos en forma visual, es decir, podemos “dibujarla” y luego automáticamente crea los scripts necesarios para su creación en el servidor; además, permite documentar y manejar nuestras Bases de Datos”. (Benitez, 2013)
- Framework LARAVEL 5.0: “Se trata de framework de desarrollo con una curva de aprendizaje muy rápida y que maneja una sintaxis expresiva, elegante, con el objetivo de eliminar la molestia del desarrollo web facilitando las tareas comunes, como la autenticación, enrutamiento, sesiones y caché. Proporciona potentes herramientas necesarias para construir aplicaciones robustas y que puede ser utilizado tanto para proyectos a nivel empresarial como para proyectos más sencillos, lo que significa que es perfecto para todos los tipos de proyectos”. (Lobo, 2017)

- ANDROID STUDIO: “Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Cuenta con algunas herramientas que nos facilitarán el desarrollo de las aplicaciones, como por ejemplo el poder previsualizar las aplicaciones en diferentes smartphones y tablets para saber cómo está quedando el código que nosotros estamos editando, y cómo se ve en los diferentes tipos de pantalla que existen”. (desdelinux, 2018).
- GITHUB: “GitHub es un sitio web y un servicio en la nube que ayuda a los desarrolladores a almacenar y administrar su código, al igual que llevar un registro y control de cualquier cambio sobre este código”. (August, 2018)

---

## **CAPÍTULO 4: ANÁLISIS - ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE**

---

### **4.1 Alcances**

El sistema será un conjunto de soluciones informáticas la cual lo componen un sistema web, aplicación móvil y sistema de acceso dactilar. La solución web podrá registrar clientes, personal, implemento

s deportivos, planes, clases, planes nutricional, progreso de clientes y rutinas. Por otro lado, la solución móvil permitirá al cliente tener acceso a rutinas asignadas, evaluación nutricional, progresión física, ejercicios, revisar datos de membresía, además modificar su perfil. Y por último, el sistema de acceso dactilar, permitirá llevar un control de ingreso de los clientes al gimnasio.

### **4.2 Objetivo del software**

- Minimizar el tiempo en la elaboración de informes necesarios a través del sistema web para la toma de decisiones del gimnasio y progresiones deportivas de cada cliente.

- Registrar los contratos adquiridos por los clientes.-
- Mantener y actualizar la información de usuarios, clientes, contratos, evaluaciones, planes, control de activos fijos.
- Registrar datos del personal del gimnasio.-
- Llevar un control de deudas por parte de los clientes.-
- Proveer un control de acceso al gimnasio.
- Asignar rutinas de ejercicios a usuarios.
- Realizar evaluaciones nutricionales a usuarios.
- Proveer información sobre clases, planes, entre otros, a futuros clientes.

### **4.3 Descripción Global del Producto**

#### **4.3.1 Interfaz de usuario**

La interfaz deberá ser de fácil acceso y uso, de colores sobrios e iconos bien marcados que destaquen e identifique de manera clara la funcionalidad, junto con su respectivo nombre, esta estará diseñada para un usuario que posea un nivel básico en computación que pueda realizar las tareas de la manera más sencilla posible.

#### **4.3.2 Consideraciones ambientales**

El sistema será utilizado tanto por el dueño del gimnasio, empleados, nutricionistas y clientes, los cuales tendrán acceso a las características o funcionalidades correspondiente a cada perfil.

#### 4.4 Requerimientos Específicos

Acrónimos utilizados en la definición de requerimientos:

**RF\_XX:** RF corresponde a los requerimientos funcionales, XX representa el número de requerimiento funcional, como por ejemplo, 100\_RF, corresponde al requerimiento funcional número 100.

**RNF\_XX:** RNF corresponde a los requerimientos no funcionales, XX representa el número de requerimiento no funcional, como por ejemplo, 100\_NF, corresponde al requerimiento no funcional número 100.

##### 4.4.1 Requerimientos Funcionales del sistema

ID	Nombre	Descripción
<b>RF_01</b>	Registrar Cliente	Permite registrar datos personales de cada cliente.
<b>RF_02</b>	Modificar Cliente	Permite modificar datos personales de un cliente ya registrado.
<b>RF_03</b>	Asignar estado a Cliente	Permite asignar estado a clientes dependiendo su inactividad o actividad en el gimnasio ( <b>0:</b> Inactivo, <b>1:</b> Activo)
<b>RF_04</b>	Registrar Profesor	Permite registrar datos personales de profesores.
<b>RF_05</b>	Modificar Profesor	Permite modificar datos personales de profesores.
<b>RF_06</b>	Asignar estado a Profesor	Permite asignar estado a profesores dependiendo su situación actual en la

		Empresa ( <b>0</b> : Inactivo, <b>1</b> : Activo)
<b>RF_07</b>	Registrar Nutricionista	Permite registrar datos personales de nutricionistas.
<b>RF_08</b>	Modificar Nutricionista	Permite modificar datos personales de nutricionistas.
<b>RF_09</b>	Asignar estado a Nutricionista	Permite asignar estado a profesores dependiendo su situación actual en la Empresa ( <b>0</b> : Inactivo, <b>1</b> : Activo)
<b>RF_10</b>	Registrar activos fijos	Permite ingresar los activos fijos de la empresa.
<b>RF_11</b>	Dar de baja activo fijo	Permite dar de baja un activo fijo ya registrado.
<b>RF_12</b>	Ingresar estado a un activo fijo	Permite ingresar el estado en el que se encuentra el activo fijo (deterioro, disponible, mantención)
<b>RF_13</b>	Añadir Rutina	Permite crear una rutina a un cliente determinado
<b>RF_14</b>	Modificar Rutina	Permite modificar una rutina de ejercicios ya registrada.
<b>RF_15</b>	Eliminar Rutina	Permite eliminar una rutina ya registrada.
<b>RF_16</b>	Asignar ejercicios a Rutinas	Permite asignar los ejercicios a la rutina correspondiente.

<b>RF_17</b>	Registrar Plan de Membresía	Permite crear planes de membresía.
<b>RF_18</b>	Modificar Plan de Membresía	Permite modificar un plan ya creado.
<b>RF_19</b>	Asignar Clases a Planes	Permite asignar clases a los respectivos planes.
<b>RF_20</b>	Administrar Pagos	Permitirá administrar pagos respectivos de cada cliente (registrar, modificar).
<b>RF_21</b>	Administrar Información de Progresión Física	Permitirá ingresar datos físicos de cada cliente para llevar un registro de su progresión.
<b>RF_22</b>	Modificar Información de Progresión Física	Permitirá modificar datos físicos de cada clientes.
<b>RF_23</b>	Eliminar Información de Progresión Física	Permitirá eliminar datos físicos de cada clientes.
<b>RF_24</b>	Añadir Plan Nutricional a cliente	Permitirá al nutricionista asignarle un plan a un respectivo cliente.
<b>RF_25</b>	Modificar Plan Nutricional	Permitirá modificar un plan nutricional ya asignado.
<b>RF_26</b>	Eliminar Plan Nutricional	Permitirá eliminar un plan nutricional ya registrado.

<b>RF_27</b>	Añadir Fotografía de Perfil	Permite a los clientes añadir su propia foto de perfil.
<b>RF_28</b>	Registrar Acceso	Permite al administrador registrar asistencia a clases.
<b>RF_29</b>	Generar Reportes	Permite al propietario generar siguientes reportes que apoyen a la toma de decisiones.
<b>RF_30</b>	Crear Clases	Permite crear clases del Gimnasio.
<b>RF_31</b>	Modificar Clases	Permite modificar clases ya existentes.
<b>RF_32</b>	Asignar Estado a Clases	Permite asignar estado a clases dependiendo su situación actual en la Empresa ( <b>0</b> : Inactivo, <b>1</b> : Activo)
<b>RF_33</b>	Asignar Horarios a Clases	Permite ingresar horarios a clases.
<b>RF_34</b>	Asignar Profesores a Clases	Permite asignar los profesores a las respectivas clases.
<b>RF_35</b>	Crear Usuario	Permite crear un usuario con su respectivo rol o privilegio.
<b>RF_36</b>	Modificar Usuario	Permite modificar un usuario.
<b>RF_37</b>	Ingresar Inscripción	Permite al administrador registrar inscripciones a clientes.
<b>RF_38</b>	Modificar Inscripción	Permite al administrador modificar

		inscripciones a clientes.
<b>RF_39</b>	Desactivar Inscripción	Permite al administrador desactivar una inscripción con el fin de generar nueva inscripción a un cliente.
<b>RF_40</b>	Enviar Correo	Permitirá al administrador/propietario del gimnasio enviar correos a clientes sobre deudas.

#### 4.4.2 Requerimientos No Funcionales del sistema

<b>Id</b>	<b>Nombre</b>	<b>Descripción</b>
<b>RNF_01</b>	Tiempo	El tiempo de respuesta será acorde a lo que necesita la empresa.
<b>RNF_02</b>	Estable	El sistema estará en línea todo el tiempo. Y con ayuda de un generador.
<b>RNF_03</b>	Seguridad	El sistema administrará los datos ingresados de forma que no existan datos incoherentes, duplicados o no correspondientes.
<b>RNF_04</b>	Usabilidad	El sistema y su interfaz serán simples e intuitivos para lograr así una mejor interacción sin problemas.
<b>RNF_05</b>	Mantenibilidad	El sistema será flexible, lo que le permitirá evolucionar, agregar o quitar funcionalidades a futuro sin mayores problemas.

#### 4.4.3 Interfaces externas de entrada

<b>Identificador</b>	<b>Nombre del ítem.</b>	<b>Detalle de Datos contenidos en ítem</b>
----------------------	-------------------------	--

DE_01	Clientes	RUT, NOMBRE, APELLIDO PATERNO, APELLIDO MATERNO, TELEFONO, EMAIL, DIRECCIÓN, NACIONALIDAD, FECHA DE NACIMIENTO, GÉNERO, FOTOGRAFIA.
DE_02	Profesores	RUT, NOMBRE, APELLIDO PATERNO, APELLIDO MATERNO, TELEFONO, EMAIL, DIRECCIÓN, NACIONALIDAD, FECHA DE NACIMIENTO, GÉNERO, FOTOGRAFIA
DE_03	Nutricionistas	RUT, NOMBRE, APELLIDO PATERNO, APELLIDO MATERNO, TELEFONO, EMAIL, DIRECCIÓN, NACIONALIDAD, FECHA DE NACIMIENTO, GÉNERO, FOTOGRAFIA
DE_04	Planes	NOMBRE, PRECIO, DESCRIPCION, DURACION, DIAS GRATIS, OBJETIVO, ESTADO.
DE_05	Activos Fijos	NOMBRE, TIPO, SUBCATEGORÍA, ESTADO
DE_06	Clases	NOMBRE, OBJETIVO, DESCRIPCION, ESTADO.
DE_07	Progresión Física	PESO CORPORAL, ESTATURA, SUBESCAPULAR, TRICIPITAL, BICIPITAL, SUPRA ILIACO, HOMBROS, PECHO, BÍCEPS RELAJADO, BÍCEPS CONTRAÍDO, MUÑECA, CINTURA, CADERA, MUSLO, GEMELO, ABDOMINAL.
DE_08	Plan Nutricional	NOMBRE, DESCRIPCION, OBJETIVO.
DE_09	Rutina	NOMBRE, DESCRIPCION.
DE_10	Pago	MONTO, FECHA, ESTADO CANCELADO, FECHA DE CANCELADO, HORA DE CANCELACION

DE_11	Privilegio	NOMBRE
DE_12	Horario	HORA DE INICIO, HORA DE FINALIZACIÓN
DE_13	Inscripción	FECHA DE INICIO, MONTO, FECHA DE FINALIZACION, FECHA DE INGRESO, HORA DE INGRESO.
DE_14	Día	NOMBRE

#### 4.4.4 Interfaces externas de Salida

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
IS_01	Reporte de Clientes	RUT, NOMBRE, APELLIDO PATERNO, APELLIDO MATERNO, TELEFONO, EMAIL, DIRECCIÓN, NACIONALIDAD, FECHA DE NACIMIENTO, GÉNERO, FOTOGRAFIA.	Archivo PDF Impresora Pantalla
IS_02	Reporte de Profesores	RUT, NOMBRE, APELLIDO PATERNO, APELLIDO MATERNO, TELEFONO, EMAIL, DIRECCIÓN, NACIONALIDAD, FECHA DE NACIMIENTO, GÉNERO, FOTOGRAFIA	Archivo PDF Impresora Pantalla
IS_03	Reporte de Activos Fijos	NOMBRE, TIPO, SUBCATEGORÍA, ESTADO	Archivo PDF Impresora

			Pantalla
IS_04	Reporte de Deudores	NOMBRE, DEUDA	Archivo PDF Impresora Pantalla
IS_05	Reporte de Asistencias	NOMBRE CLASE, FECHA INGRESO, HORA INGRESO	Archivo PDF Impresora Pantalla
IS_06	Reporte de Clases	NOMBRE, OBJETIVO, DESCRIPCION, ESTADO.	Archivo PDF Impresora Pantalla
IS_07	Reporte de Planes	NOMBRE, DESCRIPCION, OBJETIVO, PRECIO, ESTADO, DURACION.	Archivo PDF Impresora Pantalla
IS_08	Reporte de Ingresos	MES, INSCRITOS, INGRESOS	Archivo PDF Impresora Pantalla
IS_09	Reporte de Progresiones Físicas	NOMBRE, APELLIDO PATERNO, APELLIDO MATERNO, FECHA DE EVALUACIÓN, PESO CORPORAL, ESTATURA, PLIEGUE SUBESCAPULAR, PLIEGUE TRICIPITAL, PLIEGUE BICIPITAL,	Archivo PDF Impresora Pantalla

---

PLIEGUE SUPRA ILIACO, PLIEGUE HOMBROS, PLIEGUE PECHO, PLIEGUE BICEPS RELAJADO, PLIEGUEBICEPS CONTRAIDO, PLIEGUE MUÑECA, PLIEGUE CINTURA, PLIEGUE CADERA, PLIEGUE MUSLO, PLIEGUE GEMELO, ÍNDICE DE MASA CORPORAL, PORCENTAJE DE GRASA.			
IS_10	Reporte de Planes Nutricionales	NOMBRE, DESCRIPCION, OBJETIVO.	Archivo PDF Impresora Pantalla

---

#### 4.4.5 Atributos del producto

##### **Usabilidad - Operabilidad.**

Los mensajes de error serán claros para el usuario. Es decir que en el 100% de los mensajes de error, relacionados con el ingreso y procesamiento de datos, se indicará claramente el error, la causa y la solución. Se utilizará un lenguaje que le resulte familiar a los usuarios.

Se visualizará sólo contenido relevante con el fin de que el usuario solo se fije en lo realmente importante. El sistema contará con accesos directos en pantalla de inicio con el fin de facilitar la usabilidad. Se realizará capacitación a usuarios de la Empresa sobre usabilidad del Sistema

##### **Seguridad.**

El propietario de la empresa dio a entender que es de suma importancia mantener un control de acceso al sistema. Para ello, se implementará un sistema de cuentas de usuario con login y password, considerando los permisos para cada uno de ellas. El acceso a la base de datos se encuentra protegido mediante login y password. Las contraseñas se encuentran encriptadas, además podrá ser recuperada con el correo que tiene asociado el usuario.

**Fiabilidad:** El dueño de la empresa debe tener el 100% de confianza en el software, es por ello que en cada entrega del software, se deberá hacer pruebas de aceptación realizadas por el encargado respectivo.

**Diseño Gráfico:** El propietario de la empresa establece que el diseño de la interfaz del software, deberá enmarcarse en los colores que conforman el logo de la empresa.

---

## CAPÍTULO 5: ANÁLISIS DE FACTIBILIDAD

---

Este estudio nos permitirá determinar la infraestructura tecnológica y la capacidad técnica que implica la implantación del sistema solicitado, así como también el grado de aceptación que la propuesta genera en la empresa, además de los costos y beneficios que pudiera tener. Este estudio nos permite determinar las posibilidades de diseñar el sistema propuesto por nuestro cliente y su puesta en marcha.

Los aspectos a considerar para analizar la factibilidad del sistema fueron clasificados en tres áreas: *Técnico, Operativo, Económico*.

### 5.1 Factibilidad técnica

Para tener conocimiento si el sistema es factible técnicamente, se respondieron y evaluaron las siguientes interrogantes:

¿Existe o se puede adquirir la tecnología o herramientas necesarias para desarrollar el Sistema solicitado?

Si, existe la tecnología necesaria para desarrollar el Software e implementarlo, los cuales se mencionan a continuación:

A continuación se muestra la descripción del hardware a utilizar (desarrollador) para el posterior diseño, construcción y puesta en marcha del sistema.

	<b>HP 1000</b>
<b>Procesador</b>	Intel Core i3
<b>Memoria RAM</b>	8 Gb
<b>Sistema Operativo</b>	Windows 7 64 bits

<b>Gráficos</b>	Intel HD Graphics AMD Radeon HD 7400M Series
<b>Almacenamiento</b>	500 GB

	<b>Motorola moto g</b>
<b>Procesador</b>	Procesador de 4 Núcleos a 1,2 GHz
<b>Memoria RAM</b>	1 Gb
<b>Sistema Operativo</b>	Android 4.3 Jelly Bean
<b>Almacenamiento</b>	16 GB

### **Conocimientos del desarrollador**

El desarrollador que se presenta en el informe y proyecto, cuenta con los conocimientos necesarios para el desarrollo e implementación del sistema propuesto debido a los años de estudio e experiencia en programación web, móvil y conocimientos en ingeniería de software.

Por otra parte, se cuenta con el software necesario mencionado en el capítulo 3, ya que se encuentran disponibles de forma gratuita para la descarga en la web.

### **Conclusiones:**

Como resultado de este estudio técnico se determinó que la situación actual del desarrollador es óptima, ya que se cuenta con la infraestructura tecnológica de hardware y software necesaria para el desarrollo y puesta en marcha del sistema.

## **5.2 Factibilidad operativa**

Según las personas que trabajan en la Empresa, que fueron encuestadas por parte del desarrollador, deseaban mejoras en su servicio, debido a la situación en la que se encuentra el Gimnasio actualmente (problemáticas descritas en el capítulo 1), por lo que mencionaban que sería de gran ayuda informatizar los procesos mediante la implementación de un sistema que fuera de “fácil uso”.

Ahora bien, los usuarios del Sistema (a quienes va dirigido el sistema) tienen un conocimiento de usuario intermedio, por lo que manejan con facilidad variadas aplicaciones. Debido a lo anterior, no se espera mayor complicación en la adaptación e incorporación del sistema. Además para facilitar la usabilidad del sistema, el desarrollador pondrá énfasis en la creación de una interfaz intuitiva e amigable para los usuarios.

### **Conclusiones:**

Desde el punto de vista de la factibilidad operativa el sistema es factible, ya que cuentan con los conocimientos necesarios para la usabilidad/manejo de este, añadiendo además que la interfaz será lo más amigable e intuitiva posible.

## **5.3 Factibilidad económica**

### Sistema Actual:

En el sistema actual, el dueño del gimnasio es quien registra todo tipo de información (clientes, planes, pagos, etc.) de forma “manual” ya sea en libros o en planillas excel, lo cual conlleva una gran pérdida de tiempo en funciones al querer información inmediata, además de arriesgarse de tener pérdidas de información o dinero, todo esto trayendo una

gran confusión en tareas tan simples como saber si algún cliente ha pagado su membresía.

Valor UF: \$27.557 (1 de Marzo 2019).  
 Valor UF por hora Ingeniero: 0,5 UF. (\$13.778)  
 Valor UF por hora Analista: 0,4 UF. (\$11.022)  
 Valor UF por hora Programador: 0,3 UF. (\$8.267)

Costo de mano de obra para el desarrollo.

	Mes (40 horas)	Semana (10 horas)
Ingeniero	\$551.140	\$137.785
Analista	\$440.912	\$110.228
Programador	\$330.684	\$82.671

Actividades relacionadas al proyecto.

Actividad	Tiempo (semanas)	Precedente	Recursos Humanos	Honorarios
Análisis de requisitos	2	-	Ingeniero, Analista	\$496.026
Diseño del sistema	3	Análisis de requisitos	Analista	\$330.684
Codificación	5	Diseño del sistema	Programador	\$413.355

Pruebas	2	Codificación	Analista, Programador	\$385.798
Implementación	2	Pruebas	Ingeniero, Analista, Programador	\$661.368
<b>TOTAL</b>	<b>14</b>			<b>\$2.287.231</b>

Para evaluar la factibilidad económica del proyecto, se consideraran los distintos costos tanto de implementación del sistema, habilitación de espacio físico y como costos mensuales.

<b>Costos de Implementación del Sistema</b>	
Costo mano de obra	\$2.287.231

<b>Costos de habilitación de Espacio Físico</b>	
Mueble Escritorio	\$ 70.000
Silla para Escritorio	\$ 35.000
<b>Costo Total de Habilitación</b>	<b>\$ 105.000</b>

<b>Costos Mensuales</b>	
Energía por consumo de PC	\$ 6.000
2 Resma de 500 hojas	\$ 7.980
Costo Internet	\$ 22.500
Hosting	\$ 20.000
Dominio	\$ 9.990

<b>Total</b>	<b>Costo</b>	<b>\$</b>
<b>Mensual</b>		<b>66.470</b>

Este proyecto está diseñado para colaborar en la gestión y administración del gimnasio Esparta, por lo cual el costo de implementación del sistema no será considerado un costo para la empresa, ya que ha sido una colaboración mutua con el fin de beneficiar a cada una de las partes involucradas.

Considerando que el costo de implementación será una donación, el gimnasio deberá asumir aquellos costos de habilitación de espacio físico además de aquellos costos mensuales básicos para el normal funcionamiento de la instalación, como energía e internet. Por lo tanto, en conclusión, podemos mencionar que económicamente el proyecto es factible.

Dada la importancia y las ventajas que otorga la implementación del sistema, es que el centro deportivo ha aceptado poner en marcha este proyecto, solucionando así falencias y debilidades que fueron planteadas por el administrador, que dieron origen a la creación de un sistema que busca entregar las herramientas para la toma de decisiones, gestión e innovación del negocio. Además de la importancia que presenta para la administración del centro deportivo, no podemos olvidar la importancia que esta tiene para los usuarios del gimnasio Esparta, ya que podrán optar de forma fácil y expedita a información que permitirá ver el crecimiento y desarrollo personal de los clientes a lo largo del tiempo, además de otros factores que podrán ser medidos.

Finalmente, podemos concluir que la implementación del sistema impacta positivamente tanto a la organización (como por ejemplo en la toma de decisiones de inversión, contratación de profesionales, impartición de clases adicionales, control financiero, etc.) como a los usuarios (como por ejemplo en el conocimiento de índices corporales,

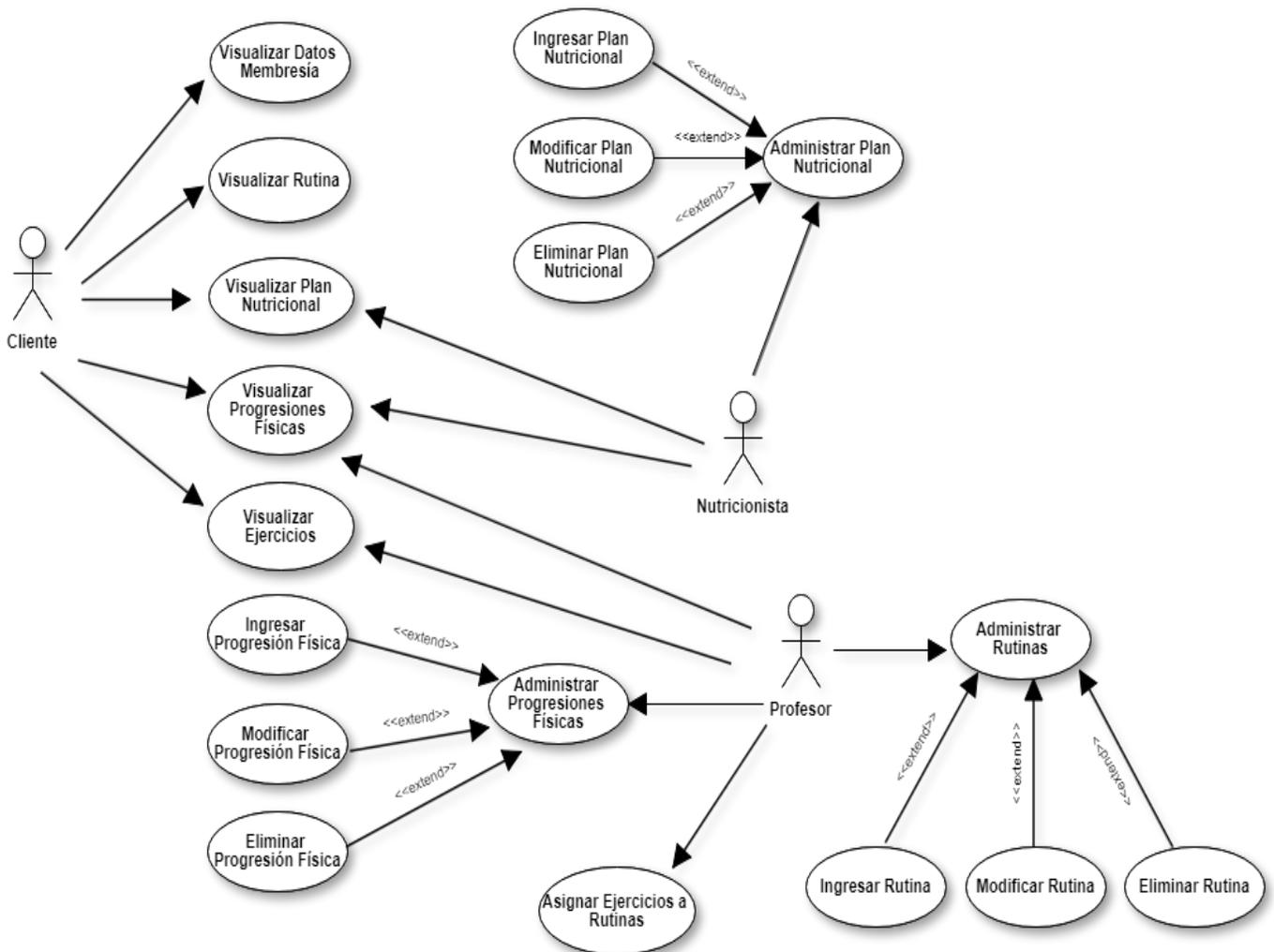
horarios de clases, situación de deuda, rutina deportiva, planes nutricionales, etc.) cumpliendo con esto el enfoque orientado al cliente.

---

**CAPÍTULO 6: ANÁLISIS – CASOS DE USO – DFD Y MODELAMIENTO DE DATOS**

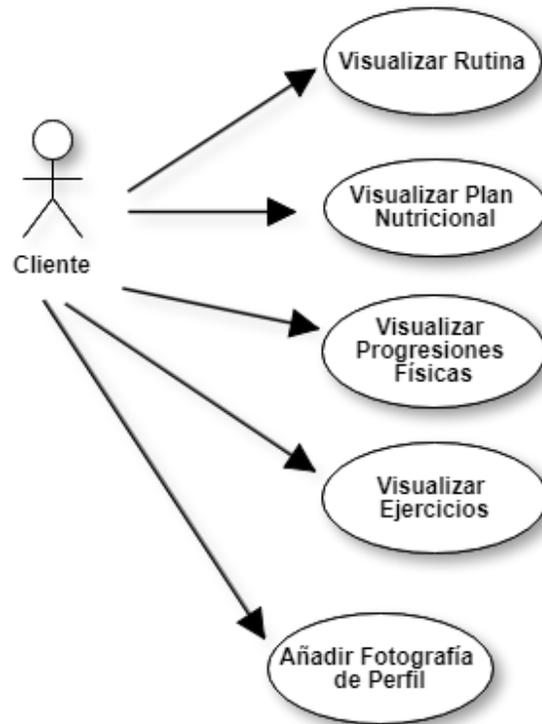
---

**6.1.1 Diagrama de casos de uso Sistema Web**





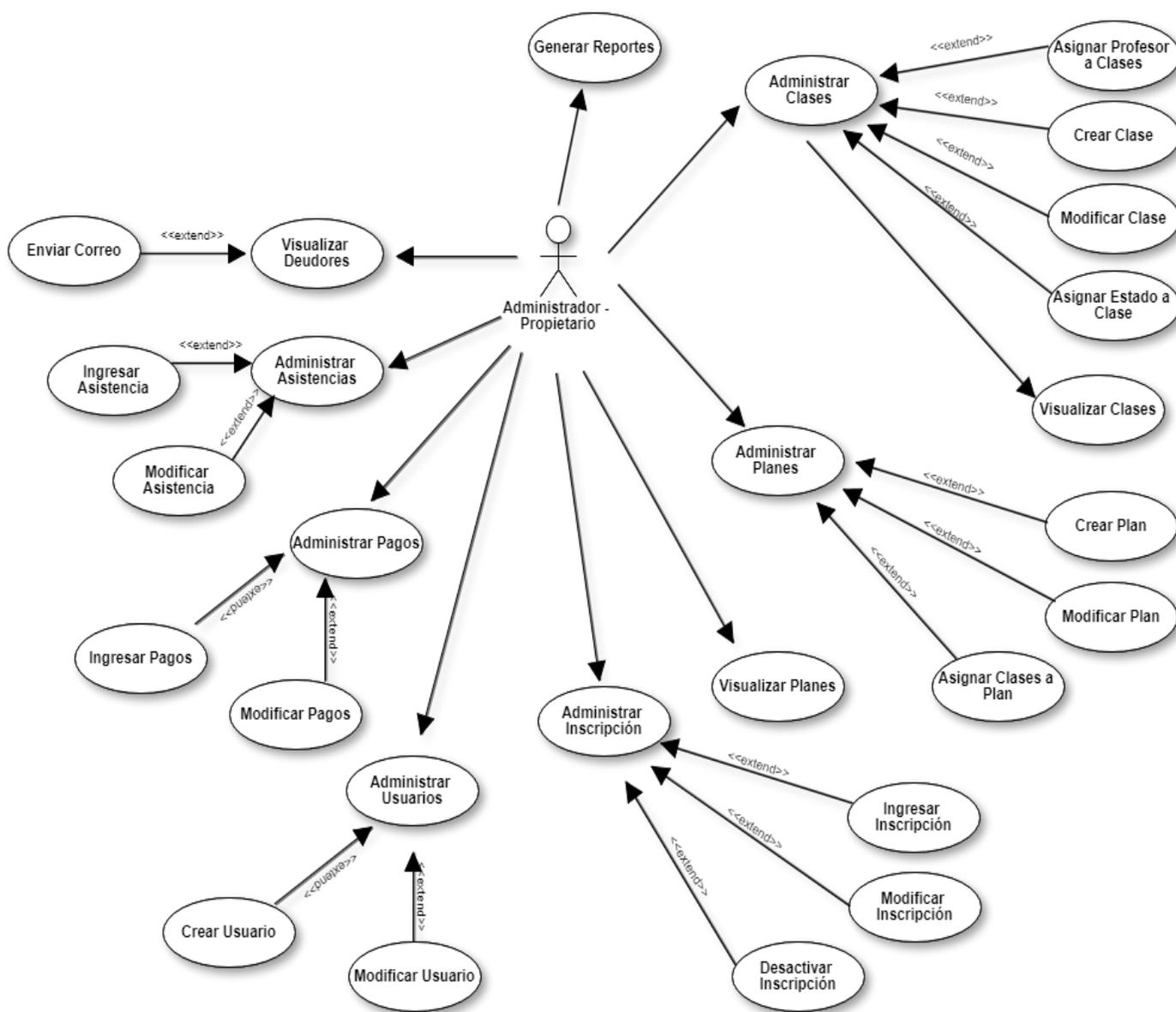
### 6.1.2 Diagrama de casos de uso Aplicación Móvil



**Ilustración 5:** Diagrama Caso de Uso Cliente, Nutricionista, Profesor, Elaboración Propia (2019).

La ilustración representa la interacción de los usuarios con el sistema. Representa cada una de las funcionalidades que pueden realizar los diversos actores (nutricionista, profesor, cliente).





**Ilustración 6:** Diagrama Caso de Uso Administrador.

La ilustración representa la interacción del administrador/propietario con el sistema. Representa cada una de las funcionalidades que puede llevar a cabo

**6.1.2 Actores**

<b>Actor</b>	<b>Cargo o Funciones</b>	<b>Nivel de conocimientos técnicos</b>	<b>Nivel de privilegio</b>
<p><b>Administrador</b> <b>-Propietario</b></p>	<p>- Administrador - Profesor deportivo</p>	<p>Medio</p>	<p>Alto</p> <p><u>Funcionalidades:</u></p> <ul style="list-style-type: none"> <li>- Administrar Personal</li> <li>- Administrar Clientes</li> <li>- Visualizar Clientes</li> <li>- Visualizar Personal</li> <li>- Administrar Activos Fijos</li> <li>- Visualizar Activos Fijos</li> <li>- Administrar Clases</li> <li>- Visualizar Clases</li> <li>- Administrar Planes</li> <li>- Visualizar Planes</li> <li>- Generar Reportes</li> <li>- Visualizar Deudores</li> <li>- Administrar Inscripciones</li> <li>- Enviar Correo a Deudores</li> <li>- Administrar Pagos</li> </ul>

			<ul style="list-style-type: none"> <li>- Administrar Asistencias</li> </ul>
<b>Profesor</b>	Personal a cargo de supervisar y apoyar la evolución deportiva de los respectivos clientes del gimnasio.	Medio	<ul style="list-style-type: none"> <li>- Administrar Rutinas</li> <li>- Administrar Información de Progresión Física</li> <li>- Visualizar Información de Progresión Física</li> <li>- Asignar Ejercicios a Rutinas</li> <li>- Visualizar Ejercicios</li> </ul>
<b>Nutricionista</b>	Profesional del área de la salud encargado de apoyar en el ámbito nutricional a los respectivos clientes del gimnasio.	Medio	<p>Medio</p> <p><u>Funcionalidades:</u></p> <ul style="list-style-type: none"> <li>- Administrar Planes Nutricionales</li> <li>- Visualizar Planes Nutricionales</li> <li>- Visualizar Información de Progresión Física</li> </ul>
<b>Cliente</b>	Cliente de la empresa	Básico	<p>Bajo</p> <p><u>Funcionalidades:</u></p> <ul style="list-style-type: none"> <li>- Visualizar Datos Membresía (Sistema Web)</li> <li>- Asignar Fotografía Perfil (Aplicación)</li> </ul>

			<p>Móvil)</p> <ul style="list-style-type: none"> <li>- Visualizar Rutina (Sistema Web – Aplicación Móvil)</li> <li>- Visualizar Plan Nutricional (Sistema Web – Aplicación Móvil)</li> <li>- Visualizar Ejercicios (Sistema Web – Aplicación Móvil)</li> <li>- Visualizar Progresiones Físicas (Sistema Web – Aplicación Móvil)</li> </ul>
--	--	--	--

### 6.1.3 Especificación de los Casos de Uso Sistema Web

#### 6.1.3.1 Caso de Uso: Administrar clientes - Registrar Cliente

- Descripción: El administrador y los profesores podrán registrar a un nuevo cliente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Clientes - Registrar Clientes.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador a cargo ingresa Rut, Nombre, Apellido Materno, Apellido Paterno, Teléfono, Email, Dirección, Nacionalidad,	2 El sistema valida que los datos ingresados sean correctos y que los datos no existan en la Base de Datos.

Fecha de Nacimiento, Género, Patología y Alergia (en caso de contener) .	
3 Presiona “Registrar Cliente”	4 El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador no ingresa todos los datos solicitados y presiona “Registrar Cliente”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
1(b) El administrador ingresa un rut ya existente.	2(b) El sistema mostrará un mensaje indicando que el rut ya existe.

- Post-Condiciones: El sistema mostrará un mensaje “Cliente registrado con éxito”.

### 6.1.3.2 Caso de Uso: Administrar Clientes - Editar

- Descripción: El administrador y los profesores podrán editar un Cliente existente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Clientes – Listar Clientes en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los clientes existentes.
2 El administrador presiona el	.

botón “Editar” del cliente que desea modificar a través de la lista completa de clientes o de una búsqueda o filtro por Nombre-Apellido Paterno o por estado (inactivo/activo)	El sistema muestra todos los datos existentes de la persona a modificar.
4 El administrador modifica el campo respectivo.	5 El sistema mostrará una alerta “Está seguro de que desea modificar?”.
6 El administrador presiona “aceptar”	7 El sistema almacena los datos modificados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a)El administrador borra la información en un campo y presiona “Guardar”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Cliente modificado con éxito”.

### 6.1.3.3 Caso de Uso: Administrar clientes - Cambiar estado de cliente

- Descripción: El administrador podrá asignar un estado a un cliente existente.
- Pre-Condiciones: : Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Clientes – Listar Clientes en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los clientes

	existentes.
2 El administrador presiona el botón “Dejar Inactivo” a través de la lista completa de clientes o de una búsqueda o filtro por Nombre-Apellido Paterno o por estado (inactivo o activo) en caso de que el cliente no se encuentre asistiendo al gimnasio o presiona el botón “Dejar activo” en caso de que un cliente vuelva a incorporarse al gimnasio.	3 El sistema muestra un mensaje de alerta “¿Está seguro de que desea cambiar de estado?”
4 El administrador presiona “aceptar”.	5 El sistema cambia de estado al cliente de la Base de Datos.

- Post-Condiciones: El sistema mostrará un mensaje “Estado asignado con éxito”.

#### 6.1.3.4 Caso de Uso: Visualizar Clientes

- Descripción: El administrador y los profesores podrán listar los clientes existentes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Clientes - Listar Clientes
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los clientes existentes.

<p>2 El administrador/profesor filtra la búsqueda de acuerdo a su necesidad, ya sea por Nombre-Apellido Paterno o por Estado (activo/inactivo).</p>	<p>3 El sistema muestra los clientes de acuerdo al filtro seleccionado por parte del administrador.</p>
<p>El administrador/profesor presiona sobre el botón “Ver Datos” sobre el cliente que desea consultar.</p>	<p>El sistema despliega los datos correspondientes al cliente (Rut, Nombres, Apellidos, fotografía, etc.)</p>

- Post-Condiciones: El sistema muestra la lista de clientes existentes

#### 6.1.3.5 Caso de Uso: Administrar Profesores - Registrar

- Descripción: El administrador podrá registrar a un nuevo personal (profesor, administrador o nutricionista).
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Personal - Registrar Profesores en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
<p>1 El administrador ingresa rut, Nombre, Apellido Materno, Apellido Paterno, Teléfono, Email, Dirección, Nacionalidad, Fecha de Nacimiento, Fotografía. y selecciona género (masculino o femenino).</p>	<p>2 El sistema valida que los datos ingresados sean correctos y que los datos no existan en la Base de Datos.</p>
<p>3 Presiona “Registrar</p>	<p>4 El sistema almacena los</p>

Profesor”.	datos ingresados en la Base de Datos.
------------	---------------------------------------

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador no ingresa todos los datos solicitados y presiona “Registrar Profesor”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
1(b) El administrador ingresa un rut ya existente.	2(b) El sistema mostrará un mensaje indicando que el rut ya existe.

- Post-Condiciones: El sistema mostrará un mensaje “Profesor registrado con éxito”.

### 6.1.3.6 Caso de Uso: Administrar Profesores - Editar

- Descripción: El administrador podrá editar un Personal existente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Profesores – Listar Profesores en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todo los profesores existentes.
2 El administrador presiona el botón “Editar” del profesor que desea modificar a través de la lista completa de personas o de una búsqueda o	3 El sistema muestra todos los datos existentes de la persona. El sistema muestra todos los datos existentes de la persona a modificar.

filtro por Nombre-Apellido Paterno.	
4 El administrador modifica el campo respectivo.	5 El sistema mostrará una alerta “Está seguro de que desea modificar?”.
6 El administrador presiona “aceptar”	7 El sistema almacena los datos modificados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a)El administrador borra la información en un campo y presiona “Registrar Profesor”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Profesor registrado con éxito”.

### 6.1.3.7 Caso de Uso: Administrar Profesores - Cambiar estado de profesor

- Descripción: El administrador podrá asignar un estado a un personal existente.
- Pre-Condiciones: : Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Profesores – Listar Profesores en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los Profesores existentes.
2 El administrador presiona el botón “Dejar Inactivo” a través de la lista completa de	3 El sistema muestra un mensaje de alerta “¿Está seguro de que desea cambiar

clientes o de una búsqueda o filtro por Nombre-Apellido en caso de que el profesor no se encuentre asistiendo al gimnasio o presiona el botón “Dejar activo” en caso de que un profesor vuelva a incorporarse al gimnasio.	de estado?”
4 El administrador presiona “aceptar”.	5 El sistema cambia de estado al cliente de la Base de Datos.

- Post-Condiciones: El sistema mostrará un mensaje “Profesor eliminado con éxito”.

#### 6.1.3.8 Caso de Uso: Listar Profesores

- Descripción: El administrador podrá listar los clientes existentes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Profesores - Listar Profesores
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los profesores existentes.
2 El administrador filtra la /búsqueda de acuerdo a su necesidad ya sea por Nombre-Apellido.	3 El sistema muestra el personal de acuerdo al filtro seleccionado por parte del administrador.
El administrador presiona sobre el botón “Ver Datos”	El sistema despliega los datos correspondientes a la persona

sobre el profesor que desea consultar.	(Rut, Nombres, Apellidos, Fotografía, etc.)
--	---

- Post-Condiciones: El sistema muestra la lista de Profesores existentes.

### 6.1.3.9 Caso de Uso: Administrar Activos Fijos - Registrar

- Descripción: El administrador podrá registrar un nuevo Activos Fijo.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Activos Fijos – Añadir Activos Fijo en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador ingresa Nombre, estado, tipo y subcategoría del Activos Fijo.	2 El sistema valida que los datos ingresados sean correctos.
3 Presiona “Agregar Activos Fijo”.	4 El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a)El administrador no ingresa todos los datos solicitados y presiona “Agregar Activos Fijo”.	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Activos Fijo registrado con éxito”.

### 6.1.3.10 Caso de Uso: Administrar Activos Fijos - Editar

- Descripción: El administrador podrá editar un activo fijo existente.

- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Activos Fijos – Listar Activos Fijos en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los Activos Fijos existentes.
2 El administrador presiona el botón “Editar” del activo fijo que desea modificar a través de la lista completa de Activos Fijos o de una búsqueda o filtro por nombre o por estado (deterioro, en mantención o disponible).	3 El sistema muestra todos los datos existentes del activo fijo.
4 El administrador modifica el campo respectivo.	5 El sistema mostrará una alerta “Está seguro de que desea modificar?”.
6 El administrador presiona “aceptar”	7 El sistema almacena los datos modificados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a)El administrador borra la información en un campo y presiona “Agregar activo fijo”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Activos Fijo modificado con éxito”.

**6.1.3.11 Caso de Uso: Administrar Activos Fijos - Dar de Baja**

- Descripción: El administrador podrá eliminar un activo fijo existente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Activos Fijos – Listar Activos Fijos en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los Activos Fijos existentes.
2 El administrador presiona el botón “Dar de Baja” del activo fijo que desea eliminar a través de la lista completa de Activos Fijos o de una búsqueda o filtro por nombre o estado (deterioro, disponible, en mantención).	3 El sistema muestra un mensaje de alerta “¿Está seguro de que dar de baja?”.
4 El administrador presiona “aceptar”	5 El sistema elimina el activo fijo de la Base de Datos.

- Post-Condiciones: El sistema mostrará un mensaje “Activos Fijo eliminado con éxito”.

**6.1.3.12 Caso de Uso: Administrar Activos Fijos - Asignar estado a Activo Fijo**

- Descripción: El administrador podrá asignar un estado a un activo fijo existente.
- Pre-Condiciones: : Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Activos Fijos – Listar Activos Fijos en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema

	1 El sistema muestra la lista de todos los Activos Fijos existentes.
2 El administrador presiona el botón “Asignar Estado” a través de la lista completa de Activos Fijos o de una búsqueda por Nombre	3 El sistema muestra un mensaje de alerta “¿Está seguro de que desea asignar estado?”
4 El administrador presiona “aceptar”.	5 El sistema ingresa el estado al activo fijo de la Base de Datos.

- Post-Condiciones: El sistema mostrará un mensaje “Estado asignado con éxito”.

### 6.1.3.13 Caso de Uso: Listar Activos Fijos

- Descripción: El administrador podrá listar los Activos Fijos existentes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Activos Fijos – Listar Activos Fijos.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los Activos Fijos existentes
2 El administrador filtra la búsqueda de acuerdo a su necesidad.	4 El sistema muestra los Activos Fijos de acuerdo al filtro seleccionado por parte del propietario.

- Post-Condiciones: El sistema mostrará la lista de Activos Fijos existentes.

**6.1.3.14 Caso de Uso: Administrar Planes - Registrar**

- Descripción: El administrador podrá añadir un nuevo plan.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Planes – Agregar Plan en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador ingresa Nombre y precio, duración y días gratis.	2 El sistema valida que los datos ingresados sean correctos y que el nombre del plan no exista en la Base de Datos.
3 Presiona “Agregar Plan”.	4 El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a)El administrador no ingresa todos los datos solicitados y presiona “Agregar Plan”.	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
1(b)El administrador ingresa un plan ya existente.	2(b) El sistema mostrará un mensaje indicando que el plan ya existe.

- Post-Condiciones: El sistema mostrará un mensaje “Plan registrado con éxito”.

**6.1.3.15 Caso de Uso: Administrar Planes - Editar**

- Descripción: El propietario podrá editar un plan ya existente.

- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Planes – Listar Planes en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los planes existentes.
2 El administrador presiona el botón “Editar” del plan que desea modificar a través de la lista completa de planes o de una búsqueda o filtro por nombre.	3 El sistema muestra todos los datos existentes del plan a modificar.
4 El administrador modifica el campo respectivo.	5 El sistema mostrará una alerta “Está seguro de que desea modificar?”.
6 El administrador presiona “aceptar”	7 El sistema almacena los datos modificados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a)El administrador borra la información en un campo y presiona “Guardar”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
1(b) El administrador modifica el nombre del plan por un nombre de un plan ya existente.	2(b) El sistema mostrará un mensaje indicando que el plan ya existe.

- Post-Condiciones: El sistema mostrará un mensaje “Plan modificado con éxito”.

### 6.1.3.16 Caso de Uso: Listar Planes

- Descripción: El administrador podrá listar los planes existentes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Planes – Listar Planes en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los planes existentes.
2 El administrador filtra la búsqueda de acuerdo a su necesidad.	3 El sistema muestra el personal de acuerdo al filtro seleccionado por parte del propietario

- Post-Condiciones: El sistema mostrará la lista de planes existentes.

### 6.1.3.17 Caso de Uso: Listar asistencia de Clientes

- Descripción: El administrador podrá listar las asistencias de los clientes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Listar Asistencias.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todas las asistencias de clientes.
2 El administrador filtra la búsqueda de acuerdo a su necesidad	3 El sistema muestra las asistencias de acuerdo al filtro seleccionado por parte del administrador.

- Post-Condiciones: El sistema mostrará la lista de asistencias existentes.

### 6.1.3.18 Caso de Uso: Administrar Clases - Registrar

- Descripción: El administrador podrá añadir una nueva Clase.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Clases – Añadir Clase en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador ingresa nombre, descripción, hora de inicio, hora de finalización y selecciona días en que se impartirán las clases.	2 El sistema valida que los datos ingresados sean correctos y que el nombre de la clase no exista en la Base de Datos, además que no se esté ingresando una clase en un horario ya existente correspondiente a otra clase.
3 Presiona “Agregar Clase”.	4 El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a)El administrador no ingresa todos los datos solicitados y presiona “Agregar Clase”.	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
El administrador ingresa un horario en el ya se encuentra	El sistema muestra un mensaje de alerta “Existe una

designado en otra clase.	clase en el mismo horario”.
1(b)El administrador ingresa una clase ya existente.	2(b) El sistema mostrará un mensaje indicando que la clase ya existe.

- Post-Condiciones: El sistema mostrará un mensaje “Plan registrado con éxito”.

### 6.1.3.19 Caso de Uso: Administrar Clases - Editar

- Descripción: El propietario podrá editar una clase existente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Clases – Listar Planes en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos las clases existentes.
2 El administrador presiona el botón “Editar” de la clase que desea modificar a través de la lista completa de clases o de una búsqueda por nombre.	3 El sistema muestra todos los datos existentes de la clase a modificar.
4 El administrador modifica el campo respectivo.	5 El sistema mostrará una alerta “Está seguro de que desea modificar?”.
6 El administrador presiona “aceptar”	7 El sistema almacena los datos modificados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
----------	------------

1(a) El administrador borra la información en un campo y presiona “Guardar”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
1(b) El administrador modifica el nombre de la clase por un nombre de una clase ya existente.	2(b) El sistema mostrará un mensaje indicando que la clase ya existe.

- Post-Condiciones: El sistema mostrará un mensaje “Clase modificada con éxito”.

#### 6.1.3.20 Caso de Uso: Administrar Clases -

- Descripción: El administrador podrá
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Clases– Listar Clases en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todas las clases existentes.
2	3
4	5

- Post-Condiciones: El sistema mostrará un mensaje “”.

#### 6.1.3.21 Caso de Uso: Listar Clases

- Descripción: El administrador podrá listar las clases existentes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Clases - Listar Clases en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todas las clases existentes
2 El administrador filtra la búsqueda de acuerdo a su necesidad.	3 El sistema muestra las clases de acuerdo al filtro aplicado por parte del administrador.

- Post-Condiciones: El sistema mostrará la lista de clases existentes.

### 6.1.3.22 Caso de Uso: Administrar Rutinas – Crear Rutina

- Descripción: El profesor o administrador podrá crear una Rutina a un cliente en específico.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y seleccionar Rutinas – Crear Rutina.
- Flujo de Eventos Básicos:

Al actor	El sistema
1)El profesor o administrador ingresa Rut y presiona botón “Buscar”	2) El sistema valida que el Rut exista y abre formulario correspondiente.
3) Ingresar Nombre de rutina y una breve descripción, luego presiona “Agregar Rutina”.	4) El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1a) El administrador o profesor no ingresa todos los datos solicitados y presiona “Añadir Rutina”	2a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
1b) El administrador o profesor ingresa un Rut no existente.	2b) El sistema mostrará un mensaje indicando que el Rut no existe.

1c) El administrador o profesor ingresa un nombre de rutina ya existente	2c) El sistema mostrará un mensaje indicando que la rutina ya existe.
--	---

- Post-Condiciones: El sistema mostrará un mensaje “Rutina Ingresada con éxito”.

### 6.1.3.23 Caso de Uso: Administrar Rutinas – Modificar Rutina

- Descripción: El profesor o administrador podrá modificar una Rutina a un cliente en específico.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Rutinas – Listar Rutinas.
- Flujo de Eventos Básicos:

Al actor	El sistema
1) El profesor o administrador selecciona Administrar Rutinas – Listar Rutinas	2) El sistema mostrará la lista de rutinas, junto con el buscador de rutinas a través del nombre del cliente respectivo.
3) Ingresar el nombre del cliente al cual desea modificar la rutina a través del buscador o a través de la lista de rutinas y presiona el botón “Editar”.	4) El sistema muestra los formularios correspondientes a la rutina seleccionada (Nombre, Descripción).
5) Modifica el dato correspondiente (nombre o descripción de la rutina) y luego presiona en botón “Editar Rutina”	6) El sistema modificará la rutina de la base de datos y mostrará un mensaje “Rutina modificada con éxito”

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador o profesor borra todos los campos y luego presiona “Editar Rutina”.	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

1(b) Ingresa Nombre de rutina ya existente.	2(b) El sistema muestra un mensaje de alerta “Rutina ya existe”.
---	--

- Post-Condiciones: El sistema mostrará un mensaje “Rutina Modificada con éxito”.

#### 6.1.3.24 Caso de Uso: Administrar Rutinas – Eliminar Rutina

- Descripción: El profesor o administrador podrá eliminar una Rutina existente a un cliente en específico.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Rutinas – Listar Rutinas.
- Flujo de Eventos Básicos:

Al actor	El sistema
1)El profesor o administrador selecciona Administrar Rutinas – Listar Rutinas	2) El sistema mostrará la lista de rutinas, junto con el buscador de rutinas a través del nombre del cliente respectivo.
3) Ingresa el nombre del cliente al cual desea eliminar la rutina a través del buscador o a través de la lista de rutinas y presiona en el botón “Eliminar”	4) El sistema muestra un mensaje de confirmación “¿Está seguro de que desea eliminar?”
5) Presiona sobre el botón “Aceptar”.	6) El sistema eliminará la rutina de la Base de datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El profesor o administrador selecciona Administrar Rutinas – Listar Rutinas	2(a) El sistema mostrará la lista de rutinas, junto con el buscador de rutinas a través del nombre del cliente

	respectivo.
1(b) Ingresar el nombre del cliente al cual desea eliminar la rutina a través del buscador o a través de la lista de rutinas y presionar en el botón "Eliminar".	2(b) El sistema muestra un mensaje de confirmación "¿Está seguro de que desea eliminar?"
2(c) Presionar sobre el botón "Cancelar"	3(c) El sistema mostrará el Listado de Rutinas.

- Post-Condiciones: El sistema mostrará un mensaje "Rutina eliminada con éxito".

### 6.1.3.25 Caso de Uso: Listar Ejercicios

- Descripción: Los Clientes, Administradores y Profesores podrán listar las los ejercicios existentes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Ejercicios - Listar Ejercicios en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todas los ejercicios existentes
2 El administrador, profesor o cliente filtra la búsqueda de acuerdo a su necesidad, ya sea por nombre de ejercicio o por clasificación del ejercicio (hombros, pecho, oblicuos, etc.).	3 El sistema muestra los ejercicios de acuerdo al filtro aplicado por parte del usuario.
El usuario presiona sobre el	El sistema despliega los datos

botón “Ver Descripción” sobre el ejercicio que desea consultar.	correspondientes a ejercicio (descripción, fotografía, video, etc).
---	---

- Post-Condiciones: El sistema mostrará la lista de ejercicios existentes.

### 6.1.3.26 Caso de Uso: Administrar Plan Nutricional – Crear Plan Nutricional

- Descripción: El Nutricionista podrá crear una Rutina a un cliente en específico.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y seleccionar Plan Nutricional – Añadir Plan Nutricional.
- Flujo de Eventos Básicos:

Al actor	El sistema
El/la Nutricionista ingresa Rut y presiona botón “Buscar”	2) El sistema valida que el Rut exista y abre formulario correspondiente.
3) Ingresas Nombre de rutina y descripción del Plan Nutricional, luego presiona “Agregar Plan Nutricional”.	4) El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1a) El/la Nutricionista no ingresa todos los datos solicitados y presiona “Agregar Plan Nutricional”	2a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
1b) El/la Nutricionista ingresa un Rut no existente.	2b) El sistema mostrará un mensaje indicando que el Rut no existe.
1c) El/la Nutricionista ingresa un nombre de plan nutricional ya existente	2c) El sistema mostrará un mensaje indicando que la rutina ya existe.

- Post-Condiciones: El sistema mostrará un mensaje “Plan Nutricional Ingresado con éxito”.

### 6.1.3.27 Caso de Uso: Administrar Plan Nutricional – Modificar Plan Nutricional

- Descripción: El Nutricionista podrá modificar un plan nutricional a un cliente en específico.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Plan Nutricional – Listar Plan Nutricional.
- Flujo de Eventos Básicos:

Al actor	El sistema
2)El/la Nutricionista selecciona Plan Nutricional – Listar Plan Nutricional	2) El sistema mostrará la lista de planes nutricionales, junto con el buscador de planes a través del nombre del cliente respectivo.
3) Ingresar el nombre del cliente al cual desea modificar el plan nutricional a través del buscador o a través de la lista de planes y presiona el botón “Editar”.	4) El sistema muestra los formularios correspondientes al plan seleccionado (Nombre, Descripción).
5) Modifica el dato correspondiente (nombre o descripción del plan) y luego presiona en botón “Guardar”	6) El sistema modificará el plan nutricional de la base de datos y mostrará un mensaje “Plan Nutricional modificado con éxito”

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El/la Nutricionista borra todos los campos y luego presiona “Guardar”.	2(a) El sistema muestra un mensaje de alerta “Todos los datos son

	obligatorios”.
1(b) Ingresa Nombre de rutina ya existente.	2(b) El sistema muestra un mensaje de alerta “Plan Nutricional ya existe”.

- Post-Condiciones: El sistema mostrará un mensaje “Plan Nutricional Modificado con éxito”.

### 6.1.3.28 Caso de Uso: Administrar Rutinas – Eliminar Plan Nutricional

- Descripción: El Nutricionista podrá eliminar un Plan Nutricional existente a un cliente en específico.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Plan Nutricional – Listar Plan Nutricional.
- Flujo de Eventos Básicos:

Al actor	El sistema
El/la Nutricionista selecciona Plan Nutricional – Listar Plan Nutricional	2) El sistema mostrará la lista de planes, junto con el buscador de planes nutricionales a través del nombre del cliente respectivo.
3) Ingresa el nombre del cliente al cual desea eliminar el plan a través del buscador o a través de la lista de planes nutricionales y presiona en el botón “Eliminar”	4) El sistema muestra un mensaje de confirmación “¿Está seguro de que desea eliminar?”
5) Presiona sobre el botón “Aceptar”.	6) El sistema eliminará el plan nutricional de la Base de datos.

- Post-Condiciones: El sistema mostrará un mensaje “Plan Nutricional eliminado con éxito”.

**6.1.3.29 Caso de Uso: Listar Plan Nutricional**

- Descripción: El Nutricionista podrá listar los planes nutricionales existentes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Plan Nutricional - Listar Plan Nutricional en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todas los planes existentes
2 El/la Nutricionista filtra la búsqueda ya sea por nombre o realiza la búsqueda por el listado completo de planes.	3 El sistema muestra los planes nutricionales de acuerdo al filtro aplicado por parte del nutricionista.

- Post-Condiciones: El sistema mostrará la lista de planes nutricionales existentes.

**6.1.3.30 Caso de Uso: Administrar Progresión Física – Añadir Progresión Física**

- Descripción: El administrador/profesor podrá añadir progresiones físicas a un cliente en específico.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y seleccionar Progresión Física Clientes – Añadir Progresión Física.
- Flujo de Eventos Básicos:

Al actor	El sistema
1)El administrador o profesor ingresa Rut y presiona botón “Buscar”	2) El sistema valida que el Rut exista y abre formulario correspondiente.
3) Ingresa Peso Corporal, Estatura, Medida Subescapular, Tricipital, Bicipital, Supra Iliaco, Hombros, Pecho, Biceps Relajado, Biceps	4) El sistema almacena los datos ingresados en la Base de Datos.

Contraído, Muñeca, Cintura, Cadera, Muslo, Gemelo.	
--	--

- Flujo de Eventos Alternativo:

Al actor	El sistema
1a) El administrador/profesor no ingresa todos los datos solicitados y presiona “Agregar Progresión Física”	2a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
1b) El administrador/profesor ingresa un Rut no existente.	2b) El sistema mostrará un mensaje indicando que el Rut no existe.

- Post-Condiciones: El sistema mostrará un mensaje “Progresión Física ingresada con éxito”.

### 6.1.3.31 Caso de Uso: Administrar Progresión Física – Modificar Progresión Física

- Descripción: El Administrador/profesor podrá modificar un plan nutricional a un cliente en específico.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Progresión Física Clientes – Listar Progresión Física.
- Flujo de Eventos Básicos:

Al actor	El sistema
	2) El sistema mostrará la lista de progresiones, junto con el buscador de progresos a través del nombre del cliente respectivo.
3) Ingresar el nombre del cliente al cual desea modificar el progreso a través del buscador o a través de la lista de	4) El sistema muestra los formularios correspondientes al plan seleccionado (mediciones).

progresiones físicas y presiona el botón “Editar”.	
5) Modifica el dato correspondiente y luego presiona en botón “Guardar”	6) El sistema modificará la progresión física de la base de datos y mostrará un mensaje “Progresión Física modificada con éxito”

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El profesor/administrador borra todos los campos y luego presiona “Guardar”.	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Progresión física modificada con éxito”.

### 6.1.3.32 Caso de Uso: Administrar Progresión Física – Eliminar Progresión Física

- Descripción: El profesor/administrador podrá eliminar una Progresión Física existente a un cliente en específico.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Progresión Física Clientes – Listar Progresión Física.
- Flujo de Eventos Básicos:

Al actor	El sistema
	2) El sistema mostrará la lista de progresiones Físicas, junto con el buscador de progresiones a través del nombre del cliente respectivo.
3) Ingresar el nombre del cliente al cual desea eliminar la progresión a través del buscador o a través de la lista de	4) El sistema muestra un mensaje de confirmación “¿Está seguro de que desea eliminar?”

planes progresiones y presiona en el botón “Eliminar”	
5) Presiona sobre el botón “Aceptar”.	6) El sistema eliminará la progresión física de la Base de datos.

- Post-Condiciones: El sistema mostrará un mensaje “Progresión Física eliminada con éxito”.

### 6.1.3.33 Caso de Uso: Listar Progresión Física

- Descripción: El administrador, profesores y clientes podrán listar las progresiones físicas.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Progresión Física Clientes - Listar Progresión Física
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todas las progresiones físicas existentes.
2 El administrador/profesor filtra la búsqueda de acuerdo a su necesidad, ya sea por Nombre-Apellido Paterno.	3 El sistema muestra las progresiones de acuerdo al filtro seleccionado por parte del administrador o profesor.
El administrador/profesor presiona sobre el botón “Ver Datos” sobre el cliente que desea consultar.	El sistema despliega las mediciones correspondientes a la progresión física.

- Post-Condiciones: El sistema muestra la lista de progresiones físicas existentes.

### 6.1.3.34 Caso de Uso: Listar Datos de Membresía

- Descripción: Los Clientes podrán listar sus datos de membresía existentes.
- Pre-Condiciones: Haber ingresado a la aplicación mediante su cuenta personal y haber seleccionado Progresión Física en el menú Principal.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra los datos correspondientes del cliente, como fecha de nacimiento, nombre, apellidos, etc. Junto con los datos del plan contratado, como fecha de inicio de inscripción, fecha de pago, etc.

Post-Condiciones: El sistema mostrará los datos de membresía existentes junto con sus datos personales (clientes).

### 6.1.3.35 Caso de Uso: Administrar nutricionistas - Registrar Nutricionista

- Descripción: El administrador podrá registrar a un nuevo cliente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Nutricionistas - Registrar nutricionista.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador ingresa Rut, Nombre, Apellido	2 El sistema valida que los datos ingresados sean

Materno, Apellido Paterno, Teléfono, Email, Dirección, Nacionalidad, Fecha de Nacimiento, Género, Patología y Alergia (en caso de contener) .	correctos y que los datos no existan en la Base de Datos.
3 Presiona “Registrar Nutricionista”	4 El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador no ingresa todos los datos solicitados y presiona “Registrar Nutricionista”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.
1(b) El administrador ingresa un rut ya existente.	2(b) El sistema mostrará un mensaje indicando que el rut ya existe.

- Post-Condiciones: El sistema mostrará un mensaje “Nutricionista registrado con éxito”.

### 6.1.3.36 Caso de Uso: Administrar nutricionistas - Editar

- Descripción: El administrador podrán editar un Nutricionista existente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Nutricionista – Listar Nutricionistas en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
----------	------------

	1 El sistema muestra la lista de todos los nutricionistas existentes.
2 El administrador presiona el botón “Editar” del nutricionista que desea modificar a través de la lista completa de nutricionistas o de una búsqueda o filtro por Nombre-Apellido Paterno o por estado (inactivo/activo)	El sistema muestra todos los datos existentes de la persona a modificar.
4 El administrador modifica el campo respectivo.	5 El sistema mostrará una alerta “Está seguro de que desea modificar?”
6 El administrador presiona “aceptar”	7 El sistema almacena los datos modificados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a)El administrador borra la información en un campo y presiona “Guardar”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Nutricionista modificado con éxito”.

### 6.1.3.37 Caso de Uso: Administrar nutricionistas - Cambiar estado de nutricionista

- Descripción: El administrador podrá asignar un estado a un nutricionista existente.

- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Nutricionista – Listar Nutricionistas en el menú.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los clientes existentes.
2 El administrador presiona el botón “Dejar Inactivo” a través de la lista completa de clientes o de una búsqueda o filtro por Nombre-Apellido Paterno o por estado (inactivo o activo) en caso de que el cliente no se encuentre asistiendo al gimnasio o presiona el botón “Dejar activo” en caso de que un cliente vuelva a incorporarse al gimnasio.	3 El sistema muestra un mensaje de alerta “¿Está seguro de que desea cambiar de estado?”
4 El administrador presiona “aceptar”.	5 El sistema cambia de estado al cliente de la Base de Datos.

- Post-Condiciones: El sistema mostrará un mensaje “Estado asignado con éxito”.

#### 6.1.3.38 Caso de Uso: Visualizar Nutricionistas

- Descripción: El administrador podrán listar los Nutricionistas existentes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Nutricionista - Listar Nutricionistas

- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los clientes existentes.
2 El administrador filtra la búsqueda de acuerdo a su necesidad, ya sea por Nombre-Apellido Paterno o por Estado (activo/inactivo).	3 El sistema muestra los clientes de acuerdo al filtro seleccionado por parte del administrador.
El administrador presiona sobre el botón “Ver Datos” sobre el nutricionista que desea consultar.	El sistema despliega los datos correspondientes al nutricionista (Rut, Nombres, Apellidos, fotografía, etc.)

- Post-Condiciones: El sistema muestra la lista de nutricionistas existentes

### 6.1.3.39 Caso de Uso: Administrar Inscripciones - Registrar Inscripción

- Descripción: El administrador podrá registrar una inscripción a un cliente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Inscripciones - Registrar Inscripción.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador selecciona el cliente a través de la lista o a través de búsqueda por Nombre-Apellidos, ingresa la fecha de inscripción, y	2 El sistema valida que los datos ingresados sean correctos y que los datos no existan en la Base de Datos.

selecciona el plan a contratar.	
3 Presiona “Registrar Inscripción”	4 El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador no ingresa todos los datos solicitados y presiona “Registrar Inscripción”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Inscripción registrada con éxito”.

#### 6.1.3.40 Caso de Uso: Administrar Inscripciones - Modificar Inscripción

- Descripción: El administrador podrá modificar una inscripción a un cliente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Inscripciones - Modificar Inscripción.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador selecciona el cliente a través de la lista o a través de búsqueda por Nombre-Apellidos, ingresa la fecha de inscripción, y selecciona el plan a cambiar.	2 El sistema valida que los datos ingresados sean correctos y que los datos no existan en la Base de Datos.

3 Presiona “Modificar Inscripción”	4 El sistema almacena los datos ingresados en la Base de Datos.
------------------------------------	---

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador no ingresa todos los datos solicitados y presiona “Modificar Inscripción”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Inscripción modificada con éxito”.

#### 6.1.3.41 Caso de Uso: Administrar Asistencias - Registrar Asistencia

- Descripción: El administrador podrá registrar una asistencia de un cliente a una clase respectiva.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Asistencias - Registrar Asistencia.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador selecciona el cliente a través de la lista o a través de búsqueda por Nombre-Apellidos, selecciona la clase a la que asistirá a través de la lista o a través de búsqueda por nombre, ingresa la fecha de ingreso y hora de ingreso.	2 El sistema valida que los datos ingresados sean correctos y que los datos no existan en la Base de Datos.

3 Presiona “Registrar Asistencia”	4 El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador no ingresa todos los datos solicitados y presiona “Registrar Asistencia”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Asistencia registrada con éxito”.

#### 6.1.3.42 Caso de Uso: Administrar Asistencias - Modificar Asistencia

- Descripción: El administrador podrá modificar una asistencia de un cliente.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Asistencias - Modificar Asistencia.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador selecciona el cliente a través de la lista o a través de búsqueda por Nombre-Apellidos, ingresa la fecha de ingreso y hora de ingreso a modificar.	2 El sistema valida que los datos ingresados sean correctos y que los datos no existan en la Base de Datos.
3 Presiona “Modificar Asistencia”	4 El sistema almacena los datos ingresados en la Base de

	Datos.
--	--------

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador no ingresa todos los datos solicitados y presiona “Modificar Asistencia”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Asistencia modificada con éxito”.

#### 6.1.3.43 Caso de Uso: Administrar Pagos - Registrar Pagos

- Descripción: El administrador podrá registrar los pagos de membresías correspondientes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Pagos - Registrar Pagos.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador selecciona el cliente a través de la lista o a través de búsqueda por Nombre-Apellidos, ingresa la hora de pago, fecha de pago.	2 El sistema valida que los datos ingresados sean correctos y que los datos no existan en la Base de Datos.
3 Presiona “Registrar Pago”	4 El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador no ingresa todos los datos solicitados y presiona “Registrar Pago”	2(a) El sistema muestra un mensaje de alerta “Todos los datos son obligatorios”.

- Post-Condiciones: El sistema mostrará un mensaje “Pago ingresado con éxito”.

#### 6.1.3.44 Caso de Uso: Administrar Pagos - Modificar Pagos

- Descripción: El administrador podrá modificar los pagos de membresías correspondientes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Pagos - Modificar Pagos.
- Flujo de Eventos Básicos:

Al actor	El sistema
1 El administrador selecciona el cliente a través de la lista o a través de búsqueda por Nombre-Apellidos, ingresa la hora de pago, fecha de pago a modificar.	2 El sistema valida que los datos ingresados sean correctos y que los datos no existan en la Base de Datos.
3 Presiona “Modificar Pago”	4 El sistema almacena los datos ingresados en la Base de Datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) El administrador no	2(a) El sistema muestra un

ingresa todos los datos solicitados y presiona “Modificar Pago”	mensaje de alerta “Todos los datos son obligatorios”.
---	---

Post-Condiciones: El sistema mostrará un mensaje “Pago modificado con éxito”.

#### 6.1.3.45 Caso de Uso: Visualizar Deudores

- Descripción: El administrador podrá visualizar los deudores correspondientes.
- Pre-Condiciones: Haber ingresado al sistema mediante su cuenta personal y haber seleccionado Deudores - Visualizar Deudores.
- Flujo de Eventos Básicos:

Al actor	El sistema
	2 El sistema muestra el listado de los clientes que tienen deudas y ofrece la opción de contactarlo por correo.

- Flujo de Eventos Alternativo:

Al actor	El sistema

Post-Condiciones: El sistema mostrará el listado de deudores.

## 6.1.4 Especificación de los Casos de Uso Aplicación Móvil Android

### 6.1.4.1 Caso de Uso: Listar Ejercicios

- Descripción: Los Clientes podrán listar las los ejercicios existentes.
- Pre-Condiciones: Haber ingresado a la aplicación mediante su cuenta personal y haber seleccionado Listado Ejercicios en el menú Principal.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todas los ejercicios existentes
2 El cliente filtra la búsqueda de acuerdo a su necesidad, ya sea por nombre de ejercicio o por clasificación del ejercicio (hombros, pecho, oblicuos, etc.).	3 El sistema muestra los ejercicios de acuerdo al filtro aplicado por parte del usuario.
El usuario presiona sobre el ejercicio que desea consultar de la lista.	El sistema despliega los datos correspondientes al ejercicio (descripción, fotografía, video, etc).

- Post-Condiciones: El sistema mostrará la lista de ejercicios existentes.

### 6.1.4.2 Caso de Uso: Listar Rutinas

- Descripción: Los Clientes podrán listar sus rutinas existentes.
- Pre-Condiciones: Haber ingresado a la aplicación mediante su cuenta personal y haber seleccionado Mis Rutinas en el menú Principal.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todas las rutinas existentes correspondientes al cliente agrupados por fecha.
2) El cliente presiona sobre la rutina que desea consultar de la lista.	3 El sistema despliega los datos correspondientes a las rutinas (nombre, descripción) junto con los ejercicios asignados por día.

- Post-Condiciones: El sistema mostrará la lista de rutinas existentes.

#### 6.1.4.3 Caso de Uso: Listar Progresión Física

- Descripción: Los Clientes podrán listar sus progresiones físicas existentes.
- Pre-Condiciones: Haber ingresado a la aplicación mediante su cuenta personal y haber seleccionado Progresión Física en el menú Principal.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todas las progresiones existentes correspondientes al cliente agrupados por fecha.
El cliente presiona sobre progresión que desea consultar de la lista.	3 El sistema despliega los datos correspondientes a la progresión (mediciones, estatura, índice de masa corporal, porcentaje de grasa, etc.).

- Post-Condiciones: El sistema mostrará la lista de progresiones físicas existentes.

#### 6.1.4.4 Caso de Uso: Listar Información Nutricional

- Descripción: Los Clientes podrán listar sus planes nutricionales existentes.
- Pre-Condiciones: Haber ingresado a la aplicación mediante su cuenta personal y haber seleccionado Info Nutricional en el menú Principal.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1 El sistema muestra la lista de todos los planes nutricionales correspondientes al cliente.
2) El cliente presiona sobre el plan que desea consultar de la lista .	3 El sistema despliega los datos correspondientes al plan (nombre del plan, descripción plan nutricional.).

- Post-Condiciones: El sistema mostrará la lista de planes nutricionales existentes.

## 6.2 Diagramas de Flujo de Datos

### 6.2.1 Diccionario de Datos

#### Tabla Personas

NOMBRE	TIPO DE DATO	PRIMARIA	FORANEA
PERSONA_RUT	VARCHAR (12)	SI	NO
NOMBRE	VARCHAR (50)	NO	NO
APELLIDO_PATER	VARCHAR (50)	NO	NO

NO			
APELLIDO_MATERNO	VARCHAR (50)	NO	NO
DIRECCION	VARCHAR (200)	NO	NO
TELEFONO	INT (11)	NO	NO
FECHA_NACIMIENTO	DATE	NO	NO
EMAIL	VARCHAR (80)	NO	NO
NACIONALIDAD	VARCHAR (50)	NO	NO
ESTADO	INT	NO	NO
FOTOGRAFIA	VARCHAR (500)	NO	NO

**Tabla propietario**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
PROPIETARIO_GIMNASIO	VARCHAR (30)	NO	NO

**Tabla Administrador**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
ADMINISTRADOR_CARGO	VARCHAR (30)	NO	NO

**Tabla Profesor**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
PROFESOR_PROFESION	VARCHAR (30)	NO	NO

**Tabla Cliente**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
CLIENTE_PROFESION	VARCHAR (30)	NO	NO

**Tabla Nutricionista**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
NUTRICIONISTA_ESPECIALIDAD	VARCHAR (30)	NO	NO

**Tabla Empresa**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
ID_EMP	INT (10)	SI	NO
NOMBRE_EMP	VARCHAR (200)	NO	NO

DIRECCION_EMP	VARCHAR (200)	NO	NO
TELEFONO_EMP	INT (11)	NO	NO
EMAIL_EMP	VARCHAR (50)	NO	NO

**Tabla Ejercicios**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
ID_EJERCICIO	INT (10)	SI	NO
NOMBRE	VARCHAR (255)	NO	NO
FOTOGRAFIA	VARCHAR (500)	NO	NO
VIDEO	VARCHAR (500)	NO	NO
DESCRIPCION	VARCHAR (255)	NO	NO
CLASIFICACION	VARCHAR (50)	NO	NO

**Tabla Activos\_fijos**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
ID_ACTIVO_FIJO	INT (10)	SI	NO
NOMBRE	VARCHAR (255)	NO	NO
ESTADO	VARCHAR (1)	NO	NO
TIPO	VARCHAR (50)	NO	NO

FECHA_INGRESO	DATE	NO	NO
SUBCATEGORIA	VARCHAR (200)	NO	NO

**Tabla Plan**

NOMBRE	TIPO DE DATO	PRIMARIA	FORANEA
PLAN_CODIGO	INT (10)	SI	NO
PLAN_NOMBRE	VARCHAR (30)	NO	NO
PLAN_DESCRIPCION	VARCHAR (200)	NO	NO
PLAN_OBJETIVO	VARCHAR (200)	NO	NO
PLAN_PRECIO	INT (11)	NO	NO
PLAN_ESTADO	BOOLEAN	NO	NO
PLAN_DURACION	INT (10)	NO	NO
PLAN_PRECIO_MENSUAL	INT (15)	NO	NO
DIAS_GRATIS	INT (10)	NO	NO

**Tabla Clase**

NOMBRE	TIPO DE DATO	PRIMARIA	FORANEA
ID_CLASE	INT (10)	SI	NO

CLASE_NOMBRE	VARCHAR (50)	NO	NO
CLASE_OBJETIVO	VARCHAR (100)	NO	NO
DESCRIPCION	VARCHAR (255)	NO	NO
ESTADO	INT (1)	NO	NO

**Tabla Rutinas**

NOMBRE	TIPO DE DATO	PRIMARIA	FORANEA
ID_RUTINA	INT (10)	SI	NO
NOMBRE_RUTINA	VARCHAR (255)	NO	NO
DESC_RUTINA	VARCHAR (255)	NO	NO

**Tabla Pago**

NOMBRE	TIPO DE DATO	PRIMARIA	FORANEA
PAGO_CODIGO	INT (10)	SI	NO
PAGO_MONTO	INT (15)	NO	NO
PAGO_FECHA	DATE	NO	NO
PAGO_CANCELADO	BOOLEAN	NO	NO
PAGO_FECHA_CA	DATE	NO	NO

NCELADO			
PAGO_HORA_CAN CELADO	TIME	NO	NO

**Tabla Privilegio**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
COD_PRIV	VARCHAR (2)	SI	NO
PRIV_NOMBRE	VARCHAR (30)	NO	NO

**Tabla Plan Nutricional**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
PLAN_NUTRI_COD IGO	INT (10)	SI	NO
NOMBRE	VARCHAR (30)	NO	NO
DESCRIPCION	VARCHAR (200)	NO	NO
OBJETIVO	VARCHAR (100)	NO	NO

**Tabla Progresos**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
---------------	---------------------	-----------------	----------------

ID_PROGRESO	INT (10)	SI	NO
PESO_CORPORAL	DOUBLE	NO	NO
ESTATURA	INT (11)	NO	NO
SUBESCAPULAR	VARCHAR (30)	NO	NO
TRICIPITAL	VARCHAR (30)	NO	NO
BICIPITAL	VARCHAR (30)	NO	NO
SUPRA_ILIACO	VARCHAR (30)	NO	NO
HOMBROS	DOUBLE	NO	NO
PECHO	DOUBLE	NO	NO
BICEPS_RELAJAD O	DOUBLE	NO	NO
BICEPS_CONTRAI DO	DOUBLE	NO	NO
MUNECA	DOUBLE	NO	NO
CINTURA	DOUBLE	NO	NO
CADERA	DOUBLE	NO	NO
MUSLO	DOUBLE	NO	NO
GEMELO	DOUBLE	NO	NO
FECHA_EVALUAC ION	DATE	NO	NO

**Tabla Usuario**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
USUARIO_LOGIN	VARCHAR (50)	SI	NO
USUARIO_PASSW ORD	VARCHAR (50)	NO	NO

**Tabla Día**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
ID_DIA	INT (10)	SI	NO
NOMBRE	VARCHAR (20)	NO	NO

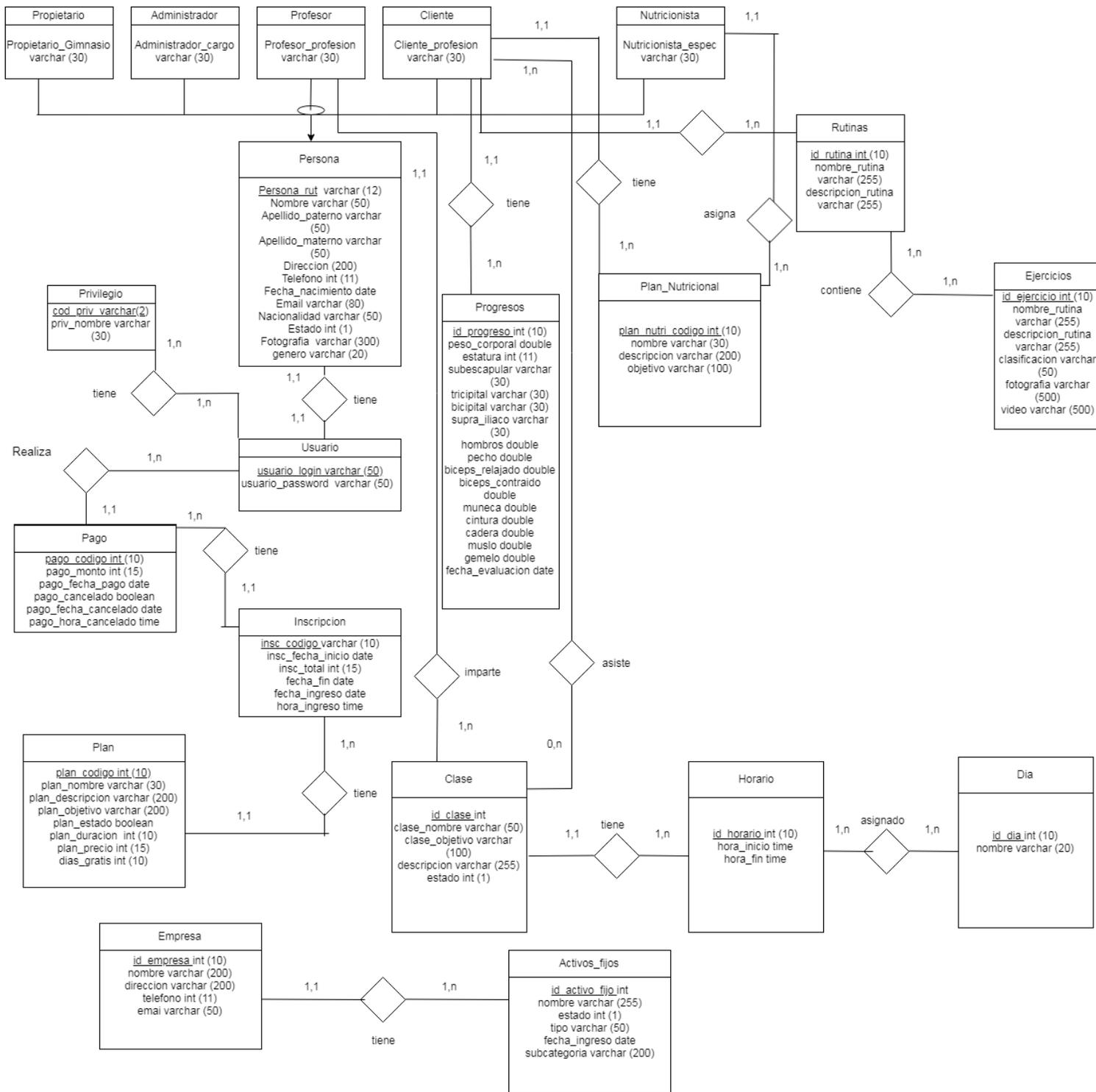
**Tabla Horario**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
ID_HORARIO	INT (10)	SI	NO
HORA_INICIO	TIME	NO	NO
HORA_FIN	TIME	NO	NO

**Tabla Inscripcion**

<b>NOMBRE</b>	<b>TIPO DE DATO</b>	<b>PRIMARIA</b>	<b>FORANEA</b>
INSC_CODIGO	VARCHAR (10)	SI	NO
INSC_FECHA_INICIO	DATE	NO	NO
INSC_TOTAL	INT (15)	NO	NO
FECHA_FIN	DATE	NO	NO
FECHA_INGRESO	DATE	NO	NO
HORA_INGRESO	TIME	NO	NO

### 6.3 Modelamiento de datos

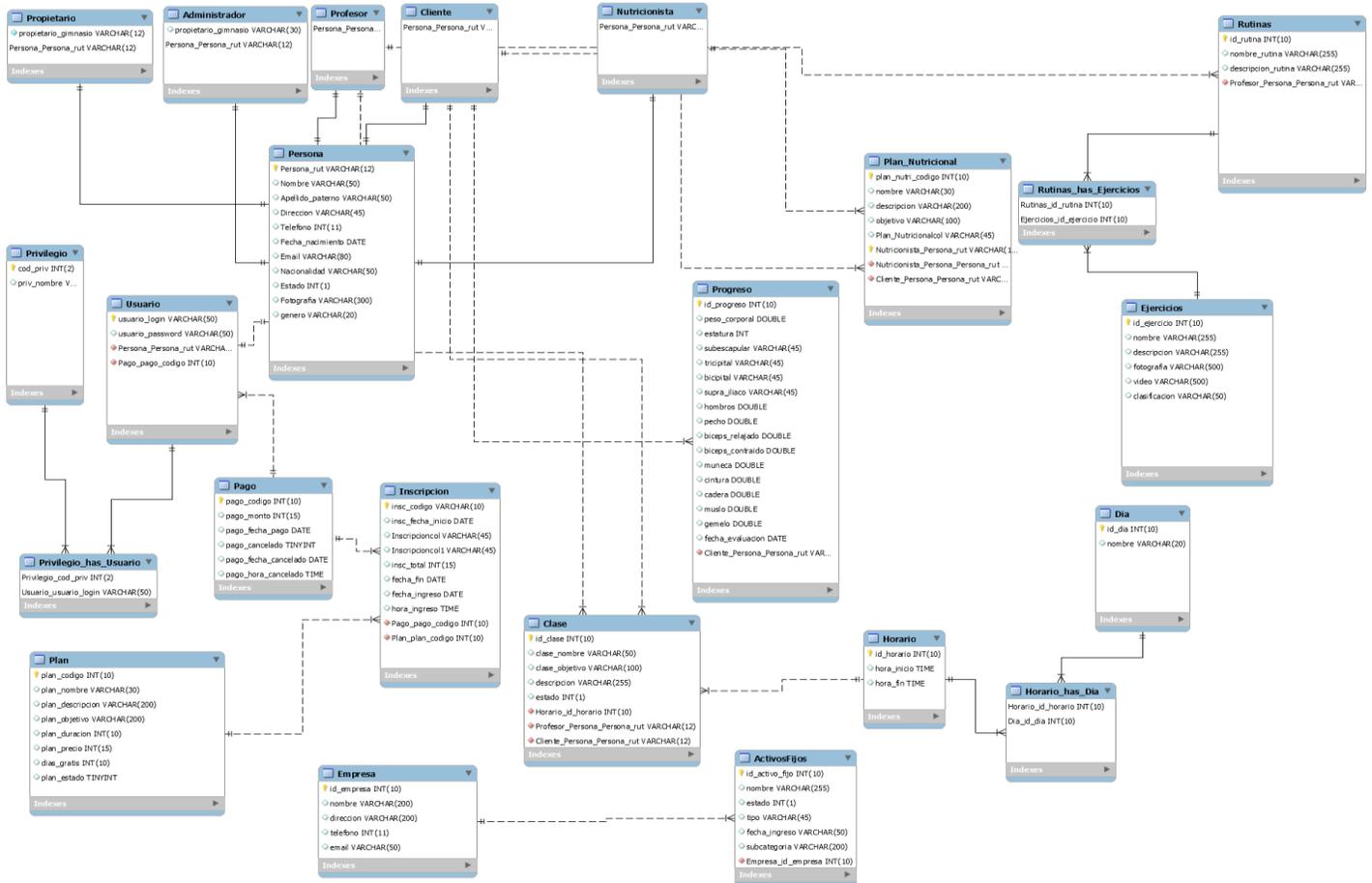


**Ilustración 7:** Modelamiento de Datos, Elaboración Propia (2019).

La ilustración representa la estructura lógica de la base de datos, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos.

**CAPÍTULO 7: DISEÑO**

**7.1 Diseño de Físico de la Base de datos**



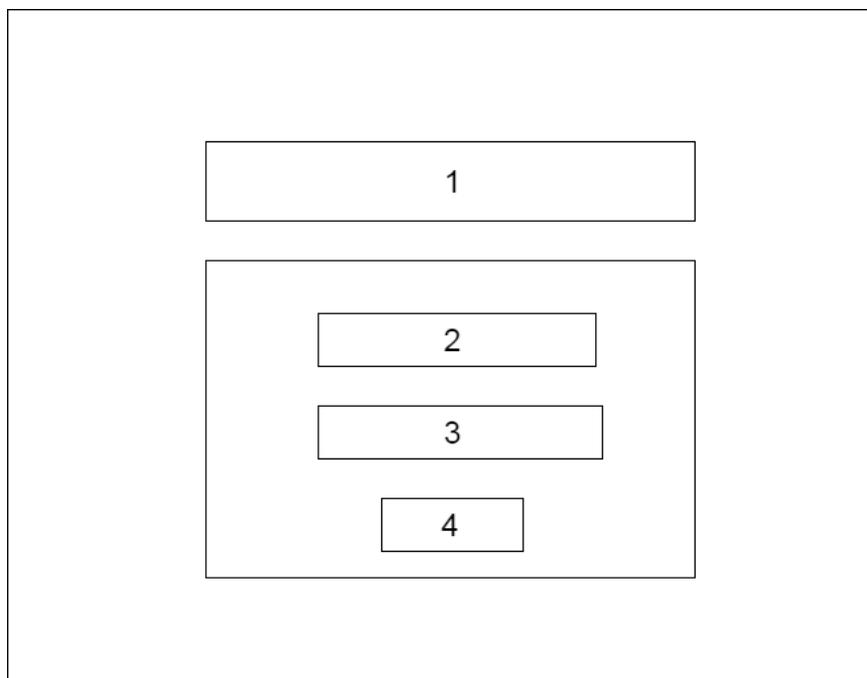
**Ilustración 8:** Diseño Físico de la Base de Datos, Elaboración Propia (2019).

La ilustración representa cómo se construirá el modelo en la base de datos.

Un modelo de base de datos física muestra todas las estructuras de tabla, incluidos el nombre de columna, el tipo de datos de columna, las restricciones de columna, la clave principal, la clave externa y las relaciones entre las tablas.

## 7.2 Diseño interfaz y navegación (Mockups)

### Interfaz Sistema Web



**Ilustración 9:** Diseño de Interfaz Sistema Web (Login), Elaboración Propia (2019).

### Sistema Web - Pantalla Inicio de Sesión

#### Área 1:

- Incluye nombre del sistema, junto con el logo oficial de la Empresa. El color de fondo será de color Rojo (color institucional).

#### Área 2:

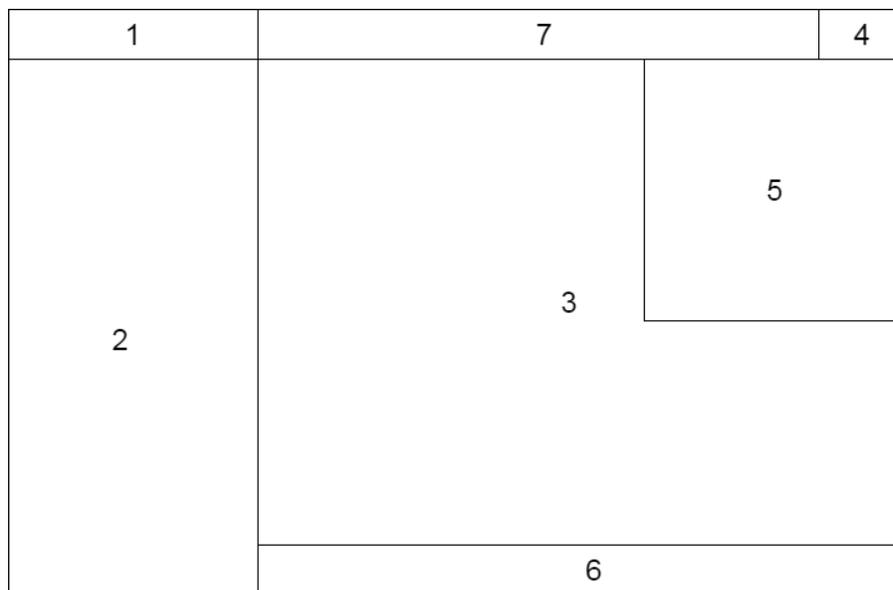
- Contiene cuadro de texto de Ingreso de Email de Usuario.

**Área 3:**

- Contiene cuadro de texto de Ingreso de Contraseña,

**Área 4:**

- Contiene Botón de Ingresar al sistema “INGRESAR”.



**Ilustración 10:** Diseño de Interfaz Área de Trabajo Sistema Web, Elaboración Propia (2019).

**Sistema Web - Área de Trabajo**

**Área 1:**

- Incluye nombre del sistema, junto con el logo oficial de la Empresa. El color de fondo será de color Rojo (color institucional).

**Área 2:**

- Barra de secciones del sistema, como Inicio, Personal, Clientes, Planes, etc. El color de fondo será de color Gris (color institucional).

**Área 3:**

- Contenido de la sección seleccionada, despliegue e ingreso de datos (FORMULARIOS), contiene Botones de navegación (BUSCAR), Botones de opción general (CREAR, EDITAR, ELIMINAR, AGREGAR, ETC).

**Área 4:**

- Contiene imagen de perfil, junto con el nombre del Usuario logueado. El color de fondo será color Rojo (color institucional).

**Área 5:**

- Contenido que se genera al presionar el Área 4, contiene imagen de perfil, junto con el nombre del usuario y tipo (profesor, cliente, nutricionista o administrador) logueado y Botones de cerrar Sesión, perfil (mostrará datos personales) y de cambiar imagen de perfil. El color de fondo será color Rojo (color institucional).

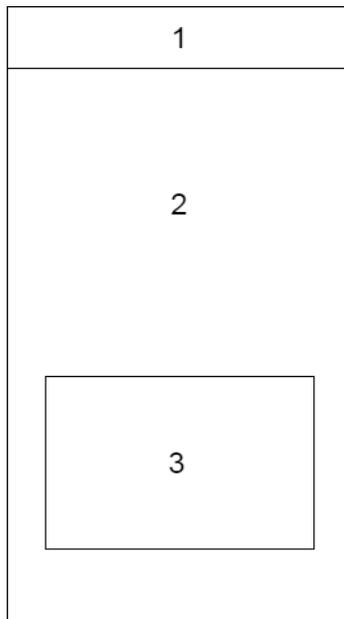
**Área 6:**

- Contiene nombre del sistema, junto con datos de la Empresa y versión del Software. El color de fondo será de color Gris (color institucional).

**Área 7:**

- Contiene Botón para visualizar o esconder nombre de las secciones del Área 2. El color de fondo será color Rojo (color institucional).

### Interfaz Aplicación Móvil



**Ilustración 11:** Diseño de Interfaz Aplicación Móvil (Login), Elaboración Propia (2019).

### Aplicación Móvil - Inicio de Sesión

#### Área 1:

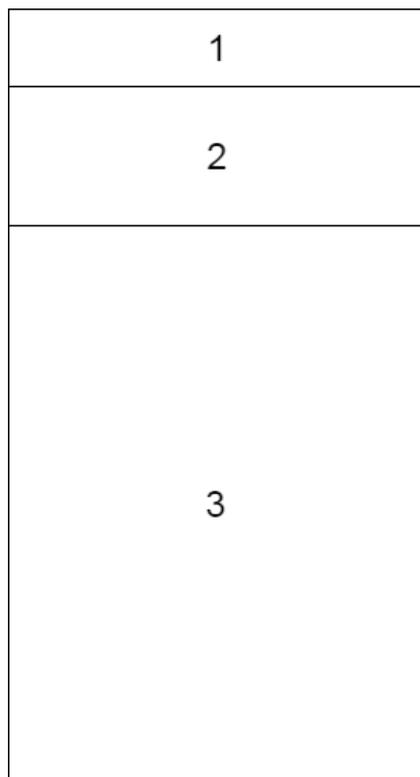
- Incluye nombre del sistema, junto con el logo oficial de la Empresa. El color de fondo será de color Rojo (color institucional).

#### Área 2:

- Contiene imagen de fondo.

#### Área 3:

- Contenido de formulario ingreso de Sesión, contiene Botones para el ingreso de email, contraseña e ingresar. Los Botones para el ingreso de datos será de color transparente, y el botón para ingresar será de color rojo (color institucional).



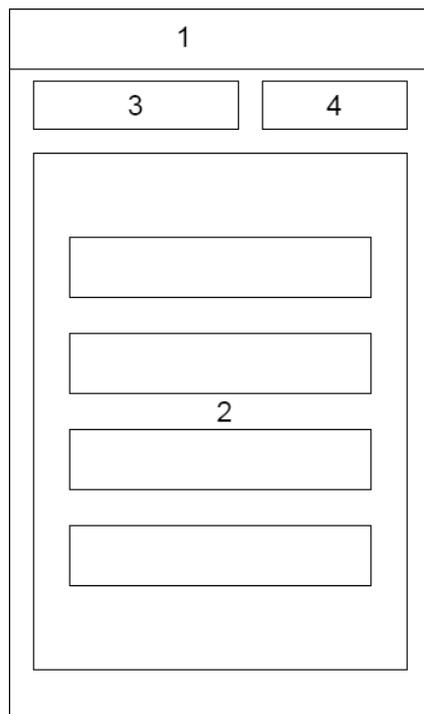
**Ilustración 12:** Diseño de Interfaz Aplicación Móvil (Menú Principal), Elaboración Propia (2019).

### **Menú Principal**

Área 1: Incluye nombre del sistema, junto con el logo oficial de la Empresa. El color de fondo será de color Rojo (color institucional), al costado derecho un botón que contenga las opciones de Añadir foto de perfil, abrir facebook, instagram del gimnasio e cerrar sesión.

Área 2: Contiene imagen de perfil del cliente junto con datos personales.

Área 3: Contiene imágenes junto con sus respectivos nombres de las funcionalidades que podrán consultar los clientes del gimnasio (información nutricional, rutinas, listado de ejercicios, progresión física, información de membresía e asistencias).



**Ilustración 13:** Diseño de Interfaz Aplicación Móvil (Rutinas, Ejercicios, Información Nutricional, Progresión Física, Información de Membresía, Asistencias), Elaboración Propia (2019).

Interfaz para los siguientes Menús: Mis Rutinas - Listado de Ejercicios - Información Nutricional - Progresión Física, Información de Membresía y Asistencias.

**Área 1:**

- Incluye nombre del sistema, junto con el logo oficial de la Empresa. El color de fondo será de color Rojo (color institucional).

**Área 2:**

- Menú Mis Rutinas: Contiene listado de las rutinas asignadas a cada cliente agrupadas por fecha de forma ascendente.
- Menú Listado de Ejercicios: Contiene listado completo de ejercicios.

- Menú Info Nutricional: Contiene listado de información/planes nutricionales asignadas a cada cliente agrupadas por fecha de forma ascendente.
- Menú Progresión Física: Contiene listado de progresiones físicas asignadas a cada cliente agrupadas por fecha de forma ascendente.
- Menú Asistencias: Contiene listado de asistencias del cliente agrupadas por fecha de forma ascendente.

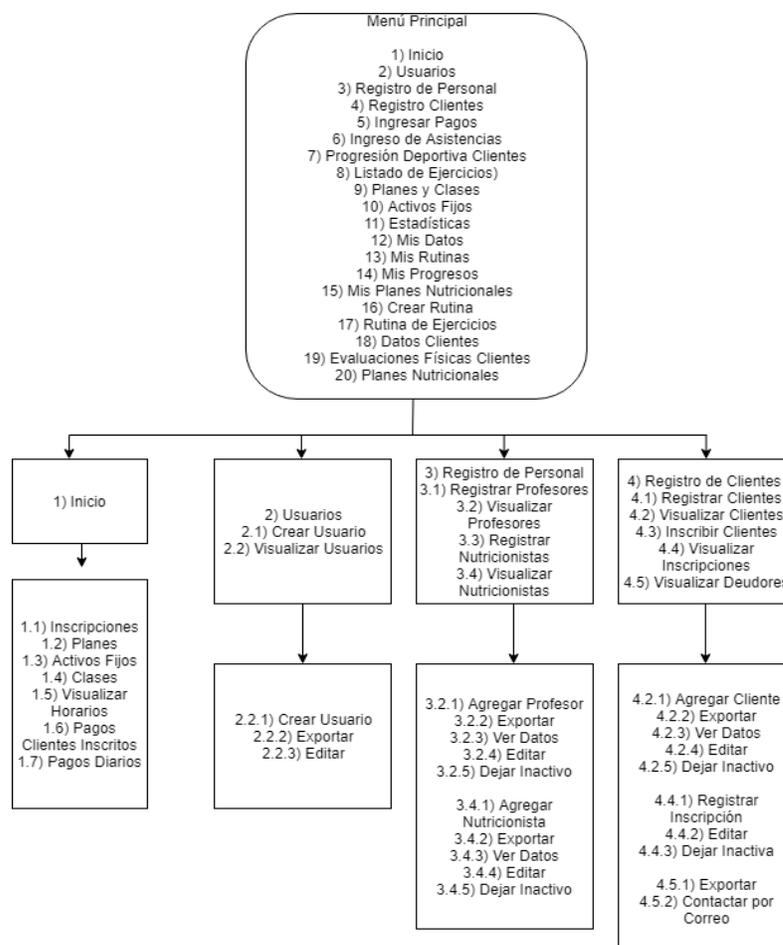
**Área 3:**

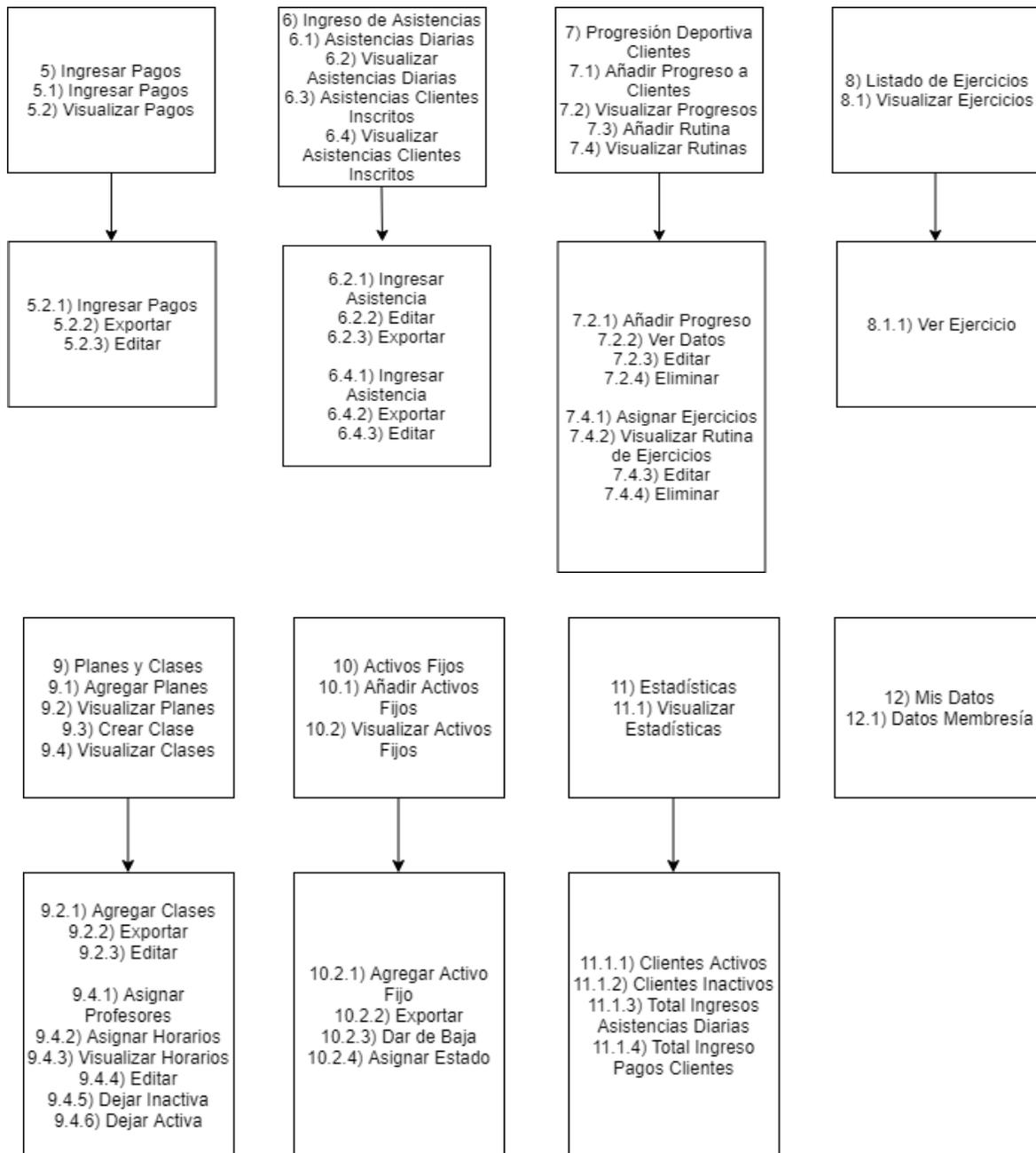
- Contiene cuadro de texto para realizar búsqueda por nombre para menú de listado de ejercicios.

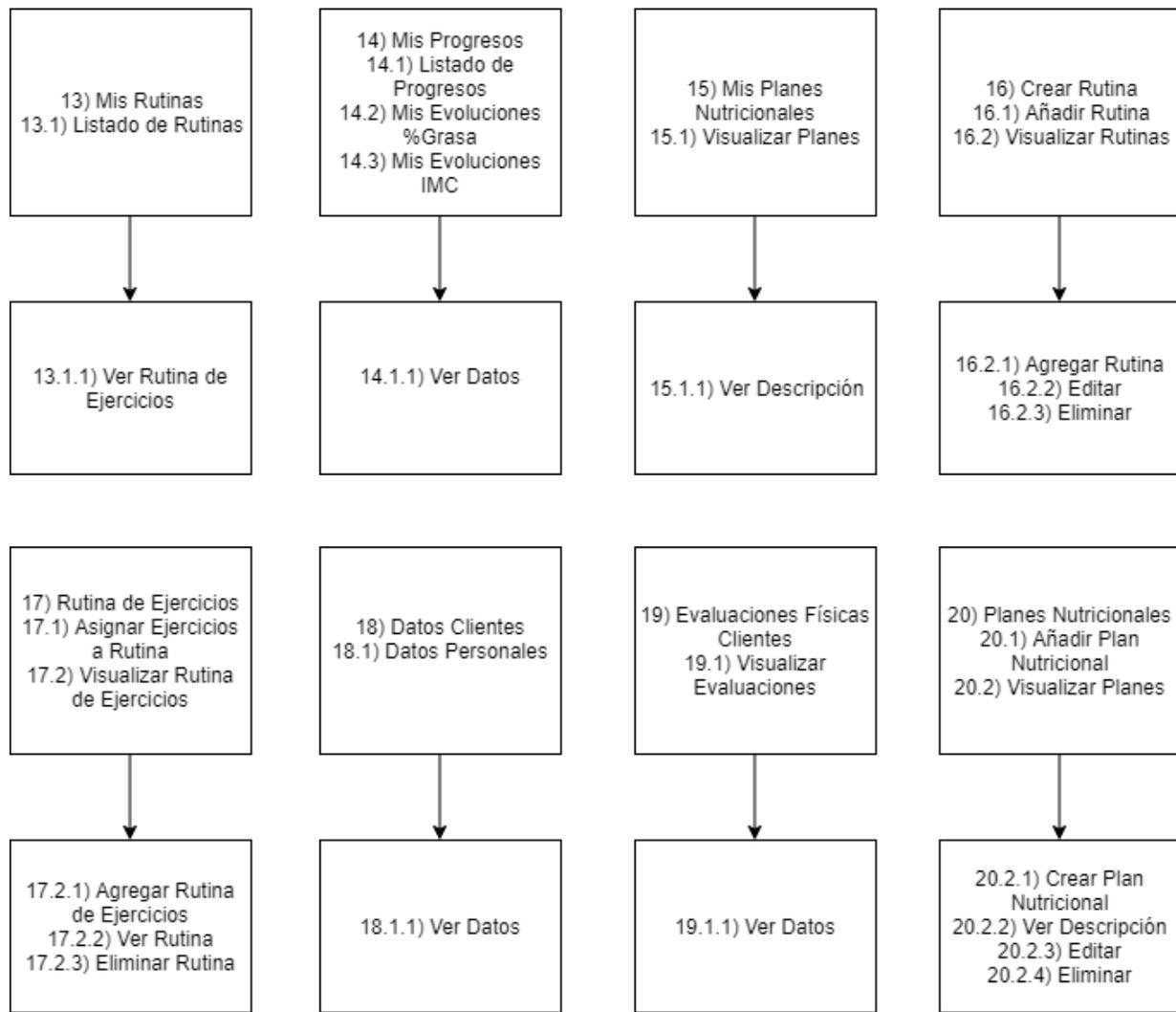
**Área 4:**

- Contiene lista de tipos de ejercicios (pecho, hombros, oblicuos, etc.) con el fin de realizar búsqueda más eficiente para menú de listado de ejercicios.

**Jerarquía de Menú General Sistema Web**





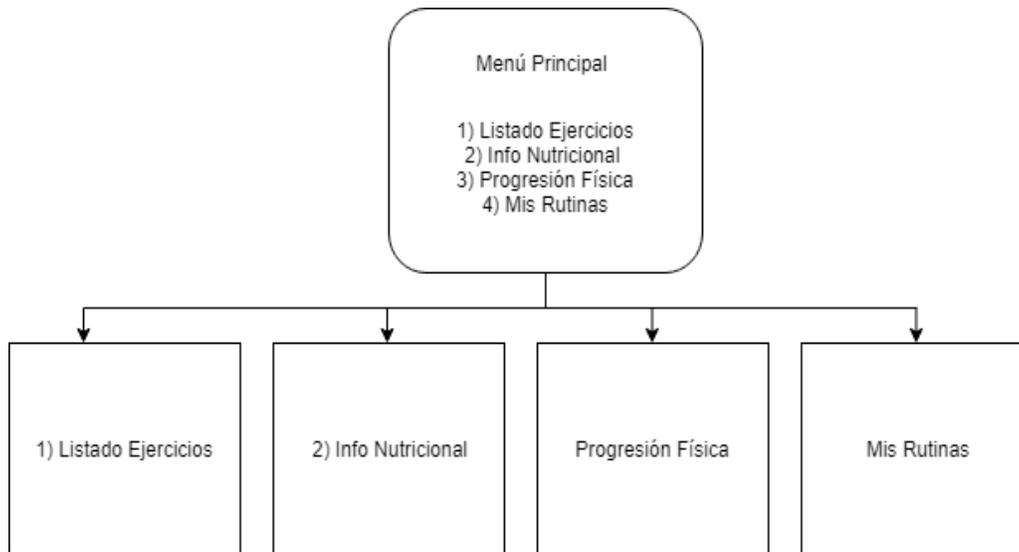


**Ilustración 14:** Jerarquía de Menú Sistema Web, Elaboración Propia (2019).

La Ilustración representa la jerarquía de Menú del Sistema Web, como estará organizando y que secciones contendrá.

Cabe mencionar que dependiendo del tipo de usuario, ya sea profesor, administrador (propietario), cliente o nutricionista contará con sus respectivos menús (funcionalidades) de la ilustración anterior descritos en Diagrama de casos de usos.

### Jerarquía de Menú aplicación móvil android



**Ilustración 15:** Jerarquía de Menú aplicación móvil android, Elaboración Propia (2019).

La Ilustración representa la jerarquía de Menú de la aplicación móvil android, cómo estará organizando y que secciones contendrá.

---

## **CAPÍTULO 8: PRUEBAS**

---

### **Objetivos de las pruebas:**

“El plan de pruebas de Software se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de Validación y Verificación de los requerimientos funcionales y no funcionales. Además, a través del plan de pruebas se puede continuar con la trazabilidad de los requerimientos, con lo cual el grupo de trabajo, identifica el porcentaje de avance que se ha logrado hasta cierto momento.

Al desarrollar el plan de pruebas, se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y se asegura la calidad del producto que se está entregando al cliente”. (Loaiza & Zorro, 2010)

Se determinaron diversas pruebas a realizar por parte del desarrollador para corroborar o verificar el correcto funcionamiento del sistema, a continuación se describen los tipos de pruebas que se realizaron:

### **Pruebas realizadas:**

#### **1) Prueba de aceptación de requerimientos**

Esta etapa se realizó de manera conjunta entre el personal del Gimnasio Esparta y el desarrollador (entrevistas), de la cual se efectuaron posteriores revisiones, mejoras y aceptación de estos requerimientos por parte de la organización.

#### **2) Prueba de validación de diseño del sistema**

En esta prueba de diseño, consistió en verificar que el diseño visual del sistema cumple con los requisitos, gustos y necesidades del Gimnasio Esparta. Esto incluye que se deberá respetar un acuerdo previo en cuanto al diseño del sistema en aspectos como logos, colores institucionales, fuentes o cualquier otro requisito de diseño solicitado por parte del

propietario.

### 3) Prueba de unidad

La prueba de unidad se centra en el módulo, es decir, se prueba cada una de las funciones del sistema. Usando la descripción del diseño detallado como guía, se prueban los caminos más importantes con el fin de descubrir y corregir errores.

#### Plan de Pruebas Unitarias:

Descripción Casos de Prueba	Condición de prueba	Pre-Condición Dato de Prueba	Dato de Prueba	Acciones de ejecución	Resultado Esperado
Caso de Uso Ingresar al sistema (App)	Debe contar con la aplicación instalada en el dispositivo móvil y abrir la aplicación.	Debe estar registrado el correo y contraseña en la base de datos.	Caja de texto Ingrese Email : javier@gmail.com  Caja de texto Ingrese Contraseña : asd	1) Abrir la aplicación- >SpartaGym  2) Ingresar correo javier@gmail.com , luego contraseña asd y presionar botón INGRESAR	Si el usuario y contraseña están correctos debe mostrar el menú principal de la aplicación con las siguientes opciones (junto con su fotografía de perfil e información personal):  - Información nutricional, Rutinas, Listado de ejercicios, Progresión Física, Información de membresía e Asistencias  Si el correo y contraseña no están escritos correctamente

					o no existen en la BD debe desplegar el mensaje: "Usuario y/o Pass incorrectos"
Caso de uso Ingresar al sistema (Sistema Web)	Debe ingresar al sistema web mediante la página web promocional (estar en pantalla de logueo).	Debe estar registrado el correo y contraseña en la base de datos.	Caja de texto Ingrese Email : javier@gmail.com  Caja de texto Ingrese Contraseña : asd	1) Ingresar a la Página Web promocional del gimnasio Esparta. 2) Ingresar al Sistema Web. 3) Ingresar correo javer@gmail.com , luego contraseña asd y presionar botón INGRESAR.	Si el usuario y contraseña están correctos debe mostrar el menú principal de la aplicación con las funcionalidades correspondientes a cada usuario, por ejemplo en el caso del cliente:  - Información nutricional, Rutinas, Listado de ejercicios, Progresión Física, Información de membresía e Asistencias  Si el correo y contraseña no están escritos correctamente o no existen en la BD debe desplegar el mensaje: "Usuario y/o Pass incorrectos"
Caso de Uso Registrar Cliente (Profesores, Administrador )	Debe haber ingresado al sistema mediante su cuenta personal.	No debe estar registrado el Rut a ingresar en la Base de Datos, haber seleccionado Clientes - Registrar	Caja de texto Rut: 19087734, Nombre: Nicolás Llanos, Apellido Paterno: Llanos, Apellido	1) Ingresar al Sistema Web mediante cuenta personal. 2) Seleccionar Clientes - Registrar Clientes en el Menú.	Si el rut no existe en la Base de Datos y se ingresan todos los datos el sistema muestra el siguiente mensaje : "Cliente Registrado con Éxito" En caso de que se

		Clientes en el Menú.	Materno: Perez, Dirección: Calle 1 casa 103, Valle la Piedra, Chiguayante, Teléfono: 98635522, Email: Nicolas.llanos m@gmail.com, Nacionalidad: Chilena, Alergia: No contiene, Patología: No contiene, selecciona fecha de nacimiento: 1995-18-03, genero: masculino.	3) Ingresar rut, Nombre, Apellido Paterno, Apellido Materno, Direccion, Telefono, Email, Nacionalidad, Alergia, Patologia, seleccionar fecha de nacimiento y genero. 4) Presionar "Registrar Cliente"	haya ingresado un rut que ya existe en la Base de Datos el sistema muestra una alerta indicando que el rut ya existe.
Caso de Uso Modificar Cliente (Profesores, Administrador )	Debe haber ingresado al sistema mediante su cuenta personal.	Haber seleccionado Clientes - Listar Clientes en el Menú. Haber presionado sobre el botón "Editar" en el cliente respectivo. El cliente debe estar registrado en la Base de Datos.	Ingresar Nicolás Llanos en caja de texto para búsqueda de clientes o situarse en el cliente mediante lista completa de clientes. Nombre: Nicolás Llanos, Apellido Paterno: Llanos, Apellido Materno: Perez,	1) Ingresar al Sistema Web mediante cuenta personal. 2) Seleccionar Clientes - Listar Clientes en el Menú. 3) Situarse en el cliente mediante la lista completa de clientes o mediante la búsqueda a través del cuadro de texto. 4) Presionar "Editar" sobre el	El sistema muestra un mensaje "Cliente modificado con éxito" En caso de que haya borrado el campo y no haya ingresado ningún número el sistema muestra el siguiente mensaje: "Todos los Campos son obligatorios".

			<p>Dirección: Calle 1 casa 103, Valle la Piedra, Chiguayante, Teléfono: 98635522, Email: Nicolas.llanos m@gmail.com, Nacionalidad: Chilena, Alergia: No contiene, Patología: No contiene, selecciona fecha de nacimiento: 1995-18-03, genero: masculino. Editar el campo Teléfono Borrando 98635522 e ingresar 67229123</p>	<p>cliente 5) Modificar el cuadro de Texto Teléfono. 6) Presionar "GUARDAR"</p>	
<p>Caso de Uso Asignar Estado a Cliente (Administrador)</p>	<p>Debe haber ingresado al sistema mediante su cuenta personal.</p>	<p>Haber seleccionado Clientes - Listar Clientes en el Menú. El usuario debe estar registrado en la Base de Datos.</p>	<p>Ingresar Nicolás Llanos en caja de texto para búsqueda de clientes o situarse en el cliente mediante lista completa de clientes.</p>	<p>1) Ingresar al Sistema Web mediante cuenta personal. 2) Seleccionar Clientes - Listar Clientes en el Menú. 3) Situarse en el cliente mediante la lista completa de clientes o mediante la</p>	<p>El sistema muestra un mensaje "Estado asignado con éxito"</p>

				<p>búsqueda a través del cuadro de texto.</p> <p>4) Presionar sobre el botón "Dejar Inactivo"</p> <p>5) Presionar "Aceptar" luego del mensaje de confirmación "¿Está seguro de que desea cambiar de estado?."</p>	
<p>Caso de Uso Listar Clientes (Administrador - Profesor)</p>	<p>Debe haber ingresado al sistema mediante su cuenta personal.</p>	<p>Haber seleccionado Clientes - Listar Clientes en el Menú.</p> <p>Deben haber registros de clientes en la Base de Datos</p>	<p>Ingresar Nicolás Llanos en caja de texto para búsqueda de cliente en específico.</p>	<p>1) Ingresar al Sistema Web mediante cuenta personal.</p> <p>2) Seleccionar Clientes - Listar Clientes en el Menú.</p> <p>3) Seleccionar Ver Datos</p>	<p>El sistema muestra el Rut, Nombre, Apellido Paterno, Apellido Materno y Estado del Cliente Nicolas Llanos, en caso de seleccionar en el Botón "Ver Datos" muestra todos los datos correspondientes al cliente como fotografía de perfil, fecha de nacimiento, etc. En caso de no buscar al Cliente en específico Nicolas Llanos muestra la lista de todos los clientes registrados en la Base de Datos junto con sus datos básicos (Rut, Nombre, Apellido Paterno, Apellido Materno y Estado.</p>
Caso de Uso	Debe haber	Haber	Ingresar	1) Ingresar al	El sistema despliega

<p>Crear Plan Nutricional (Nutricionista)</p>	<p>ingresado al sistema mediante su cuenta persona</p>	<p>seleccionado Plan Nutricional - Crear Plan Nutricional Deben haber registros de clientes en la Base de Datos</p>	<p>19087734 en Cuadro de Texto Rut. Presionar "BUSCAR" Ingresar Nombre Plan Nutricional: Plan Nutricional Nicolas Llanos, Ingresar Descripción: Plan nutricional orientado al tipo de alimentación para el aumento de masa muscular.</p>	<p>Sistema Web mediante cuenta personal. 2) Seleccionar Plan Nutricional - Crear Plan Nutricional 3) Ingresar Rut 19087734 4) Presionar "BUSCAR" 5) Ingresar Nombre de Plan Nutricional 6) Ingresar Descripción Plan Nutricional 7) Presionar Agregar Plan Nutricional.</p>	<p>los campos Nombre de plan nutricional y Descripción en caso de que e rut exista para poder ingresar el plan correspondiente, en caso contrario el sistema despliega el siguiente mensaje "El rut no existe". Luego de Ingresar los datos en los formularios y haber presionado "Agregar Plan Nutricional" el sistema despliega el siguiente mensaje "Plan Nutricional ingresado con exito".</p>
---	--	---	--	---	--

**Pruebas que se desarrollarán a posterior:**

**4) Prueba de integración**

Son aquellas que se ejecutan posteriormente a la aprobación de las pruebas anteriores. Describen a las pruebas de todos los elementos unitarios que forman el sistema. Reside en realizar pruebas para comprobar que el gran conjunto de partes del sistema funcionan de forma conjunta y sin errores.

**5) Prueba de sistema**

Verifica que cada elemento encaje de forma adecuada y que se alcance la funcionalidad y el rendimiento del sistema total, además de comprobar que el sistema satisface los requerimientos funcionales y no funcionales especificados anteriormente para validar la aceptación de los usuarios del sistema.

En conclusión, podemos mencionar que primeramente se cumplieron con los requerimientos solicitados por parte del personal del Gimnasio, el cual se analizaron en conjunto con el desarrollador, esto no quiere decir que no surjan nuevas necesidades antes de la implementación e puesta en marcha del sistema.

Por otra parte, luego de analizar diversas posibilidades de diseño e interfaz del software, se optó y se decidió por el más adecuado, el cual también se trabajó en conjunto con el personal de la Empresa, se definieron colores, logos, fuentes, etc.

Además se realizaron pruebas unitarias a funcionalidades para validar su correcto funcionamiento a través de los errores generados.

Por último, se realizarán dos tipos de pruebas una vez realizadas todas las pruebas unitarias pertinentes para cada funcionalidad, denominadas pruebas de integración (para comprobar que las partes del sistema funcionen en conjunto de manera correcta) y prueba del sistema para validar que el sistema cumpla con los requerimientos funcionales y no funcionales.

---

## **CAPÍTULO 9: PLAN DE IMPLEMENTACIÓN Y PUESTA EN MARCHA**

---

El método para la implementación y puesta en marcha se realizará de la siguiente manera: Se instalará el sistema, sin interrumpir la forma en la que se realizan las tareas actualmente (Registros a mano). Ambas se utilizarán simultáneamente, hasta que se compruebe fehacientemente que el sistema Informático funciona adecuadamente. Es recomendable utilizar este método debido a que el sistema impactará en forma importante en el Negocio. De esta forma se contará con una mayor seguridad.

### **Acondicionamiento de instalaciones**

El objetivo es adaptar las instalaciones donde será implantado el sistema, teniendo las condiciones óptimas de hardware, software y red que permitan la operatividad del sistema.

Etapas:

- Se evaluarán las condiciones del espacio físico de la institución deportiva, se diseñará la distribución del equipo informático, de la red y el mobiliario dentro del sector.
- Diseñar la distribución del equipo informático dentro del local y el mobiliario.
- Adecuar el local: infraestructura física que ofrezca seguridad al equipo contra situaciones climatológicas.
- Instalación del software: Se instalará la base de datos y sistema web en hosting.

### **Capacitación a personal**

Se realizará una capacitación al personal con el fin de que los usuarios del sistema tengan conocimiento de las funciones y la operación aplicada específicamente a cada uno de los procesos de negocio de las áreas involucradas, acorde a los requerimientos de la organización.

**Temas a abordar en la respectiva capacitación:**

TEMA	DURACION	MODALIDAD	ENCARGADO
Descripción de módulos	1 hora	Presencial	Desarrollador Nicolás Llanos m.
Inicio de sesión	30 minutos	Presencial	Desarrollador Nicolás Llanos m.
Gestión de usuarios	30 minutos	Presencial	Desarrollador Nicolás Llanos m.
Operaciones del sistema	1 hora y 30 minutos	Presencial	Desarrollador Nicolás Llanos m.
Consultas del sistema	1 hora	Presencial	Desarrollador Nicolás Llanos m.

**Contenidos a abordar**

- Descripción de Módulos: Espacio en el cual se llevará a cabo la introducción y descripción de cada uno de los módulos del sistema a los usuarios finales, resolviendo todas y cada una de las dudas que se presenten.
- Inicio de Sesión: Modulo el cual está diseñado para ingresar al sistema a través de una cuenta y su respectiva contraseña, los ingresos al sistema serán controlados por medio de perfiles los cuales tendrán ciertas restricciones para realizar acciones en el sistema.

- Gestión de Usuarios: Modulo el cual fue diseñado para la creación, edición de usuarios y perfiles dentro del sistema.
- Operaciones del sistema: Modulo el cual fue diseñado para la creación y manejo de las funcionalidades requeridas.
- Consultas del sistema: Modulo el cual fue diseñado para realizar reportes.

Cabe mencionar que además de la capacitación realizada por parte del Desarrollador, los usuarios finales contarán con su respectivo manual de usuario, el cual se puede visualizar en el capítulo de Anexos.

---

## CAPÍTULO 10: CONCLUSIONES

---

El proyecto “Desarrollo de Sistema Web y Aplicación Móvil Android para el apoyo a la gestión y progresión Deportiva Gimnasio Esparta” no solo formó una parte de la organización, sino que también generó necesidades de definir la estructura interna en la organización de manera clara y comprensible a cualquier persona, pertenezca o no a la organización, generando un valor agregado que en un principio no estaba contemplada. Esto nos permitió participar de un proceso del cual pocas veces hay oportunidades de estar involucrado, lo que significó un gran aprendizaje tanto personal como profesional, para todas las partes involucradas, siendo todos beneficiados de este proceso.

En conclusión se puede afirmar que se cumplieron con los objetivos del proyecto, así también como la implementación de los requerimientos más urgentes o prioritarios mencionadas por parte del personal del gimnasio. Sin embargo, cabe destacar que quedan funcionalidades pendientes como el registro de asistencias con huella dactilar, control de accesos con torniquete y mejorar aquellas funcionalidades que corresponden a la progresión deportiva de los clientes, el cual se trabajará en conjunto entre el desarrollador y personal de la Empresa para su posterior implementación.

### **Beneficios esperados:**

Desde el punto de vista de la organización:

- Eficiencia y centralización de la información:

Con la ayuda de Sistema de Apoyo a la Gestión, la búsqueda de información se hará con rapidez, está siempre estará respaldada en la Base de Datos, centralizada, óptima, precisa, para cuando la necesiten.

- Seguridad: Tendrán a su disposición un sistema seguro, en donde todos los niveles de la Pirámide Organizacional se verán directa o indirectamente beneficiados con él.
- Apoyo y seguimiento de avances de Clientes: Con la ayuda de la Aplicación Móvil Android se podrá llevar seguimientos de los avances/progresión deportiva-física que han tenido los clientes cada cierto tiempo designado por los profesionales a cargo con el fin de analizar y tomar decisiones con respecto a la persona.

#### Beneficios obtenidos:

Desde el punto de vista del Equipo de Desarrollo: Como se trata de un proyecto de desarrollo que impacta a la organización de manera importante, lo aprendido durante la realización, no solo es la parte técnica de desarrollo en sí, sino que también, el entendimiento de cómo funciona una empresa, los canales de comunicación, las jerarquías, que en este caso no están definidas, pero gracias al trabajo en conjunto, se realizó la definición de cada cargo, de los alcances de estos, así como también las responsabilidades. En la parte de desarrollo, se trabajó fuertemente utilizando, renovando y adquiriendo conocimiento en lenguajes, bases de datos, y en el desarrollo del informe final. Pero lo más importante es el aprender a trabajar en equipo, ya que esa experiencia hace una diferencia importantísima entre un muy buen profesional y un profesional regular.

Expertise: Sirvió para practicar y ampliar los conocimientos de los lenguajes de programación utilizados, de una base de datos para, así, adquirir experiencia en el ámbito de desarrollo y administración de proyectos.

Adquisición de Valores: Desarrollar este proyecto fue una buena oportunidad para socializar con los usuarios del sistema y, así, hacerlos partícipes de este. Se cumplió a cabalidad con la responsabilidad, puntualidad y el tratar de resolver las problemáticas de la mejor manera posible.

Actualización de Currículum: Fue una buena oportunidad para crecer como estudiantes y como profesionales, para adquirir experiencia el campo de la ingeniería desarrollando un proyecto informático de principio a fin.

---

## BIBLIOGRAFÍA

---

Ingenieros, I. N. G. S. (2014, 25 diciembre). ¿QUE ES WAMP SERVER? Recuperado 4 enero, 2019, de <https://abdonflores.wordpress.com/2013/11/19/que-es-wamp-server/>

Colaboradores de Wikipedia. (2019, 3 enero). PHP - Wikipedia, la enciclopedia libre. Recuperado 4 enero, 2019, de <https://es.wikipedia.org/wiki/PHP>

Tecnopedia. (s.f.-b). Mysql Workbench 5. Recuperado 4 enero, 2019, de <http://www.tecnopedia.net/mysql/mysql-workbench-5/>

Colaboradores de Wikipedia. (2019b, 2 enero). Laravel - Wikipedia, la enciclopedia libre. Recuperado 4 enero, 2019, de <https://es.wikipedia.org/wiki/Laravel>

Brian Jackson, B. J. (2018, 21 agosto). ¿Qué es GitHub? Una Guía para Principiantes sobre GitHub. Recuperado 4 enero, 2019, de <https://kinsta.com/es/base-de-conocimiento/que-es-github/>

Developers. (s.f.). Conoce Android Studio | Android Developers. Recuperado 4 marzo, 2019, de <https://developer.android.com/studio/intro/?hl=es-419>

Carvajal, V., & Jimenez, L. (2010). Plan de Pruebas de Software. Recuperado 11 marzo, 2019, de <http://www.alfarosolis.com/>

Daniel Rivas, D. R. (s.f.). La metodología de desarrollo iterativo y creciente. Recuperado 30 abril, 2019, de

[https://www.academia.edu/7964110/La\\_metodologia\\_de\\_desarrollo\\_iterativo\\_y\\_creciente](https://www.academia.edu/7964110/La_metodologia_de_desarrollo_iterativo_y_creciente)

Cruz, C. S. (s.f.). MODELOS - Intro Ingenieria Software. Recuperado 1 mayo, 2019, de <http://scruz334.blogspot.es/tags/modelos/>

Anibal Ledesma, A. L. (2013, 29 noviembre). ¿Que es WampServer. Recuperado 1 mayo, 2019, de [https://prezi.com/h0j-\\_95vhkj2/que-es-wampserver/](https://prezi.com/h0j-_95vhkj2/que-es-wampserver/)

Benítez, A. B. (2013, 27 septiembre). MySQL: Optimización de la configuración del servidor y utilidades interesantes.. Recuperado 1 mayo, 2019, de <https://www.jabenitez.com/2013/09/27/mysql-optimizacion-de-la-configuracion-del-servidor-y-utilidades-interesantes/>

Camargo, R. C. (s.f.). ¿QUÉ ES PHP? Recuperado 1 mayo, 2019, de <https://es.calameo.com/books/000894483d5bb4fd470e5>

Rafael Lobo, R. L. (s.f.). Framework Laravel. Recuperado 1 mayo, 2019, de <https://es.slideshare.net/RafaelLobo4/framework-laravel>

---

**ANEXOS**

---

**ANEXO 1: FICHA EMPRESA**

Datos de la empresa:

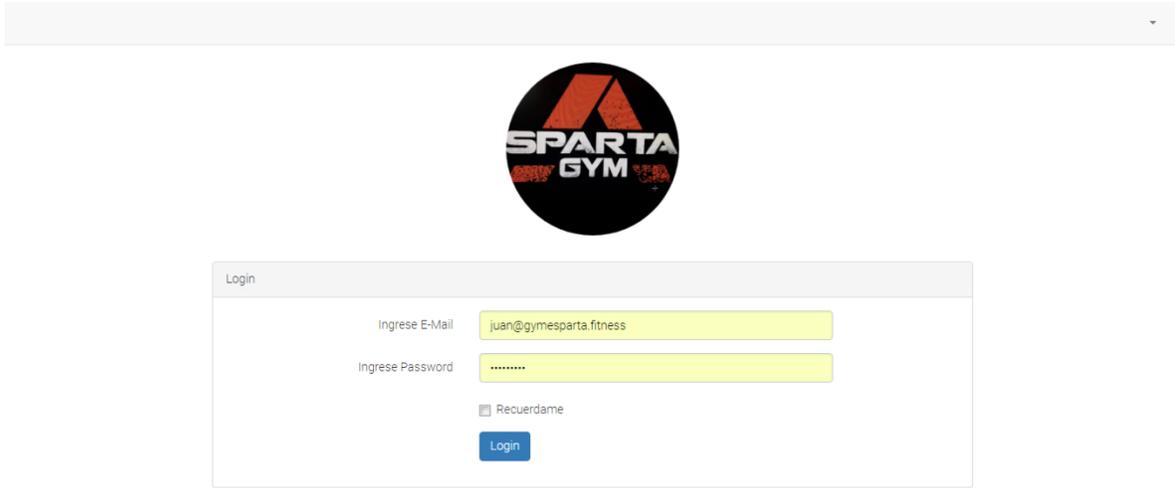
Nombre	Gimnasio Esparta
Giro comercial	Organización de Deportes / Centro Fitness
Dirección	Manuel Rodríguez 2150, Chiguayante
Teléfono	+569 82407367
Página Web	
No de empleados	4

Datos de la persona de contacto en la empresa:

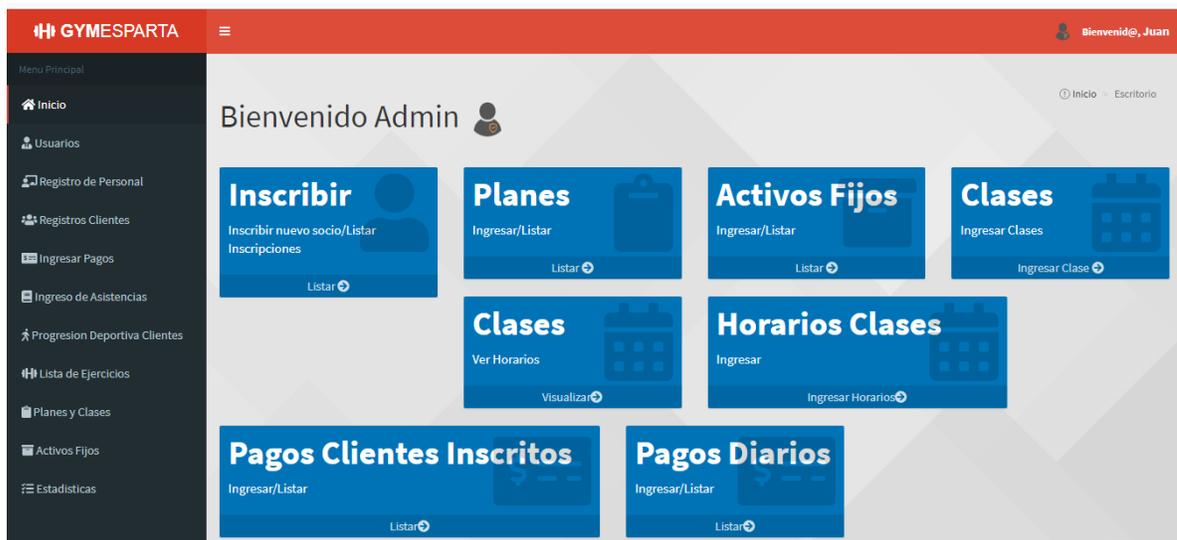
Nombre	Diego Felipe Montanares
Cargo	Propietario/profesor
E-mail	<a href="mailto:gymesparta3.0@gmail.com">gymesparta3.0@gmail.com</a>
Teléfono	+569 82407367

## ANEXO 2: MANUAL DE USUARIO

### Funcionalidades Administrador



- 1) A continuación se muestra la pantalla de logueo al sistema, lo cual se debe ingresar el e-mail y contraseña correspondiente al administrador.



- 2) Luego de ingresar con su cuenta personal, el sistema mostrará un listado de funcionalidades a escoger como menú de inicio.

**HI GYMESPARTA** Bienvenido, Juan

**INGRESAR NUEVO PROFESOR**

Datos Usuario

**Rut:**

**Profesion:**

**Nombre:**

**Apellido Paterno:**

**Apellido Materno:**

**Telefono:**

**Email:**

**Nacionalidad:**

**Fecha de Nacimiento:**

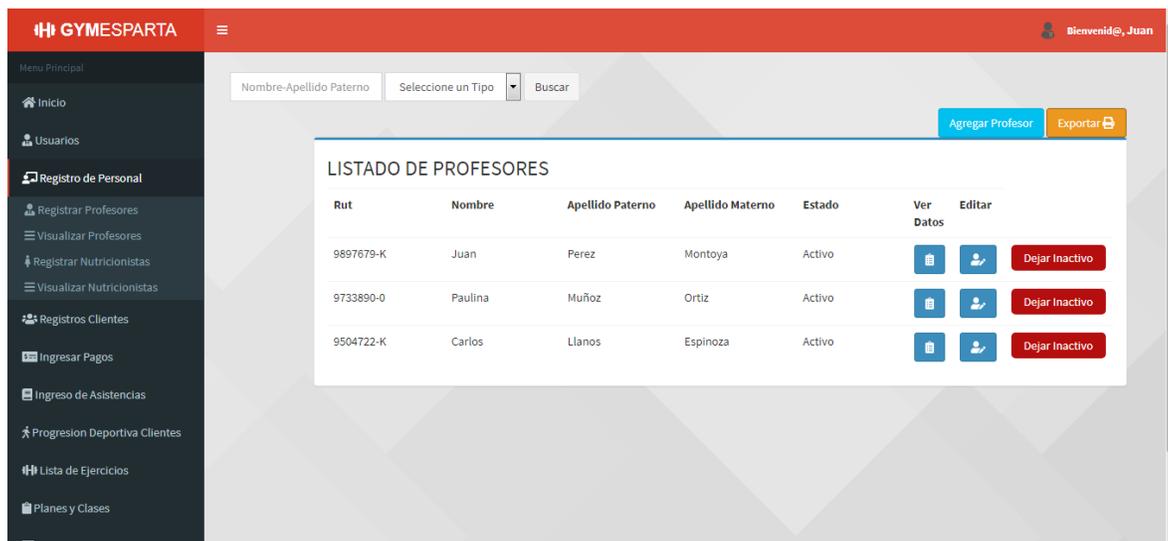
**Selecciona genero:**  
 Masculino  Femenino

Direccion Usuario

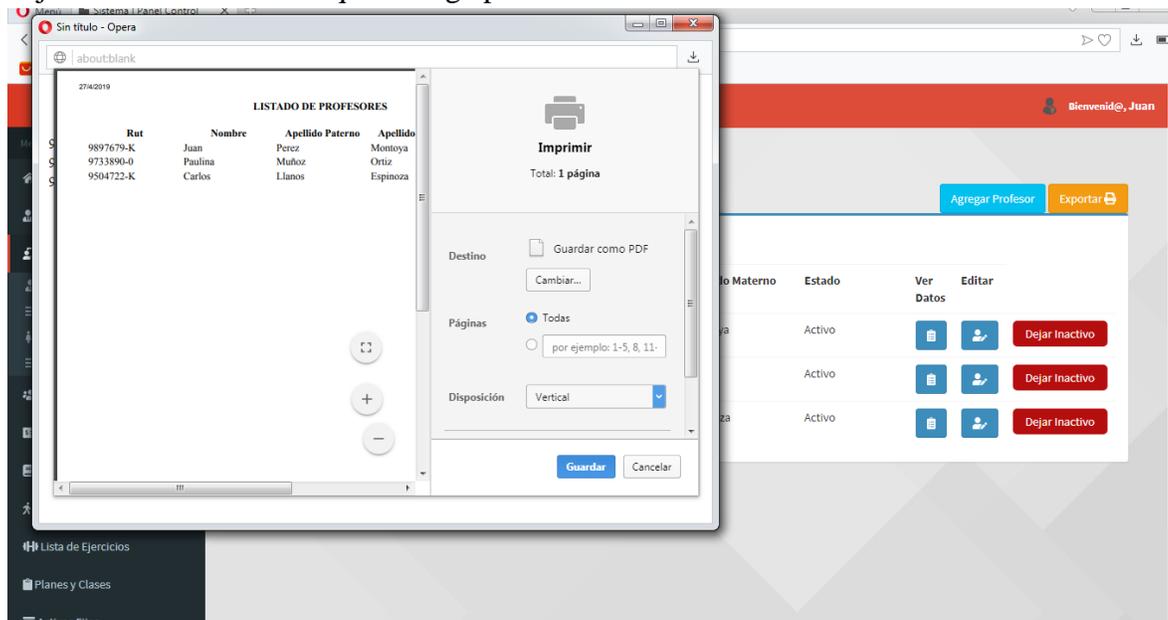
**Direccion:**

Agregar Profesor

- 3) Las siguientes pantallas muestran la opción de registrar profesores, el cual como primer lugar se debe seleccionar “Registro de Personal” y luego Registrar Profesores. El administrador debe ingresar los datos correspondientes al profesional, y por ultimo debe presionar “Agregar Profesor”.



- 4) A continuación, se visualiza el listado de profesores registrados, junto con las opciones respectivas, como agregar otro profesor, ver los datos del profesor a través de una búsqueda por nombre-apellido o por tipo (inactivo, activo), editar los datos, exportar y dejar inactivo en caso de que no siga perteneciendo a la institución.



- 5) La siguiente pantalla muestra los reportes que se pueden visualizar a través de presionar sobre el botón “Exportar” en el menú de visualizar Profesores, imprimir o guardar correspondiente a los datos de los profesores registrados en el sistema.

**GYMESPARTA** Bienvenid@, Juan

**INGRESAR NUEVO NUTRICIONISTA**

Datos Usuario

Rut:

Nombre:

Apellido Paterno:

Apellido Materno:

Telefono:

Email:

- 6) La siguiente pantalla muestra la opción de registrar nutricionistas, el cual como primer lugar se debe seleccionar “Registro de Personal” y luego Registrar Nutricionistas. El administrador debe ingresar los datos correspondientes al nutricionista, y por ultimo debe presionar “Agregar Nutricionista”.

**GYMESPARTA** Bienvenid@, Juan

Nombre-Apellido Paterno Seleccione un Tipo Buscar

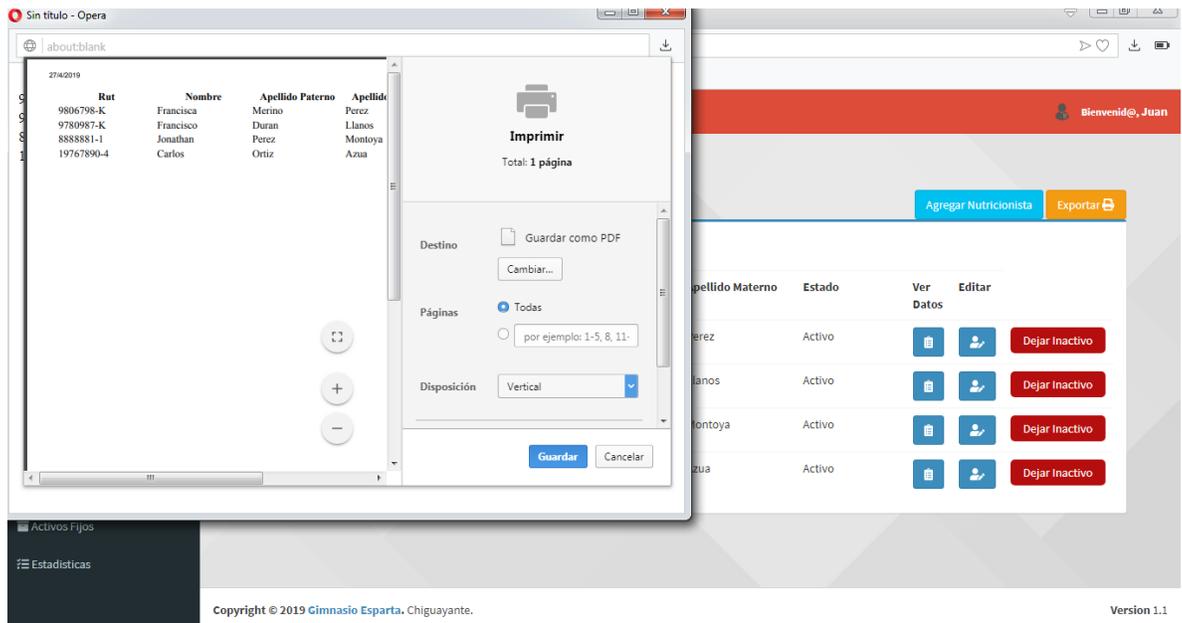
Agregar Nutricionista Exportar

**LISTADO DE NUTRICIONISTAS**

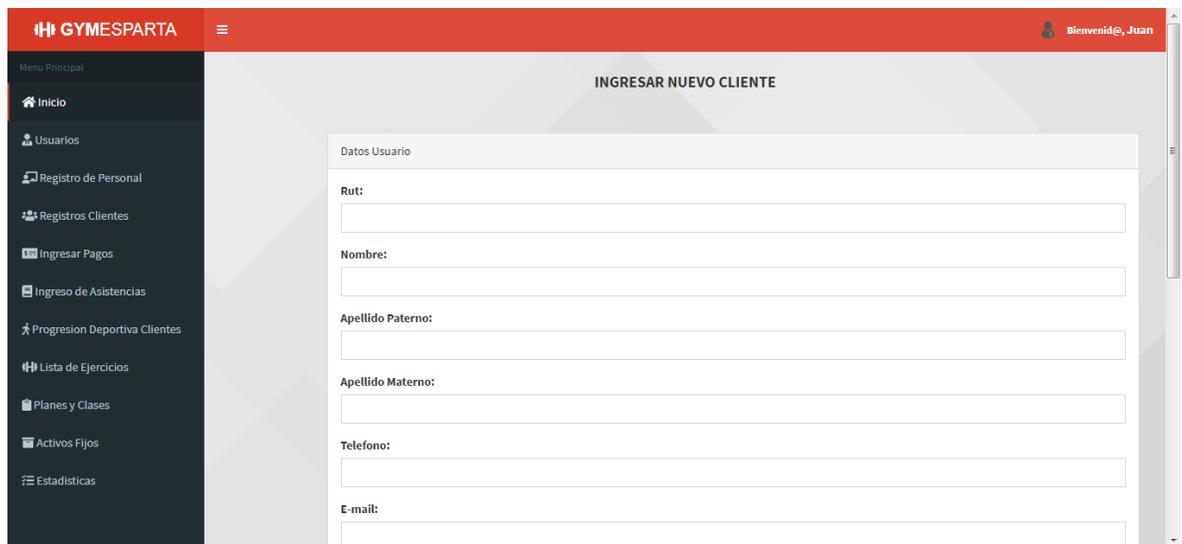
Rut	Nombre	Apellido Paterno	Apellido Materno	Estado	Ver Datos	Editar
9806798-K	Francisca	Merino	Perez	Activo		Dejar Inactivo
9780987-K	Francisco	Duran	Llanos	Activo		Dejar Inactivo
8888881-1	Jonathan	Perez	Montoya	Activo		Dejar Inactivo
19767890-4	Carlos	Ortiz	Azua	Activo		Dejar Inactivo

Copyright © 2019 Gimnasio Esparta. Chiguayante. Version 1.1

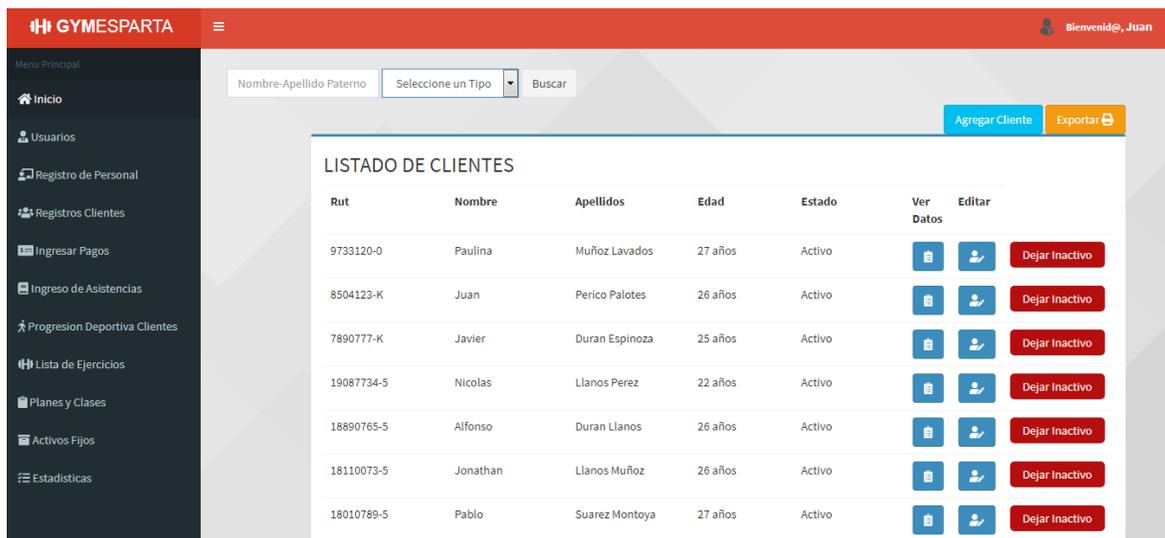
- 7) A continuación, se visualiza el listado de nutricionistas registrados, junto con las opciones respectivas, como agregar otro nutricionista, ver los datos del nutricionista a través de una búsqueda por nombre-apellido o por tipo (inactivo, activo), editar los datos, exportar y dejar inactivo en caso de que no siga perteneciendo a la institución.



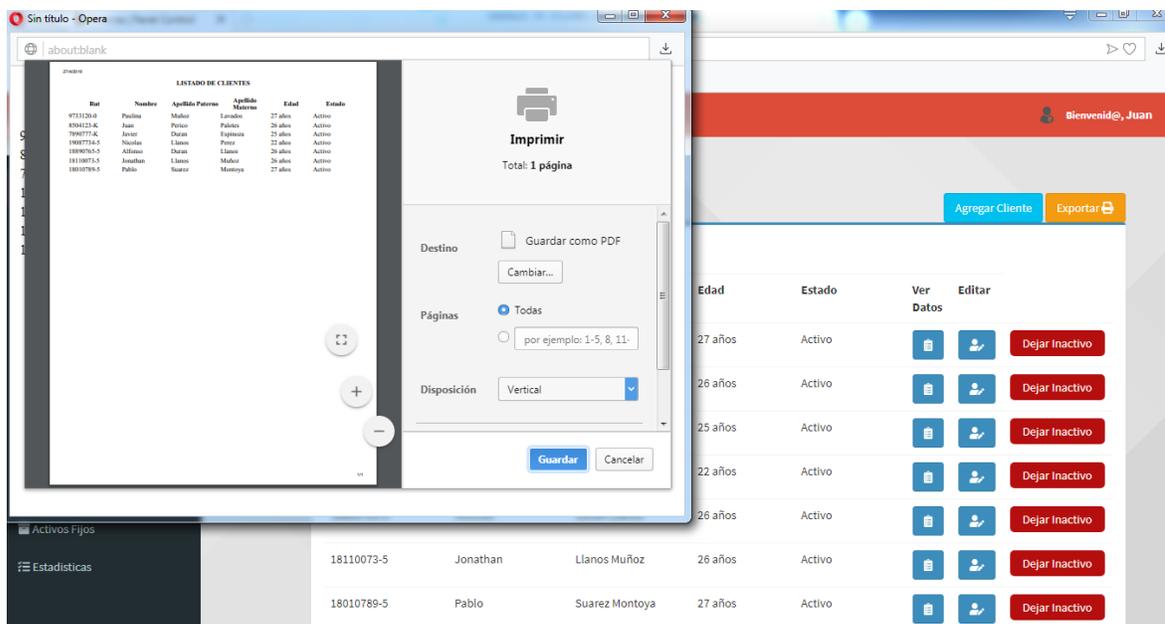
- 8) La siguiente pantalla muestra los reportes que se pueden visualizar a través de presionar sobre el botón “Exportar” en el menú de visualizar Nutricionistas, imprimir o guardar correspondiente a los datos de los nutricionistas registrados en el sistema.



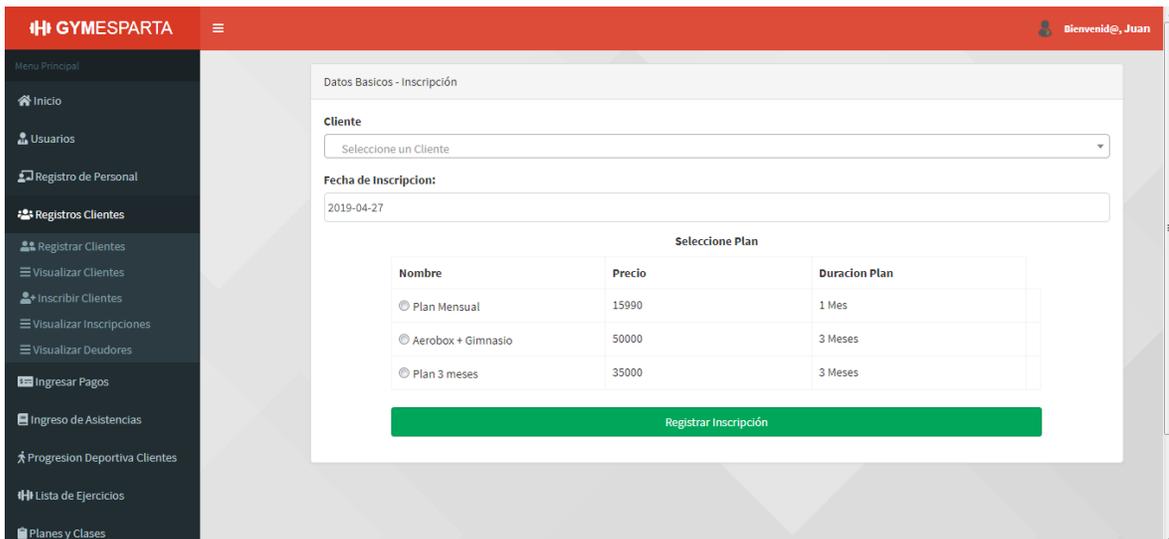
- 9) La siguiente pantalla muestra la opción de registrar clientes, el cual como primer lugar se debe seleccionar “Registros Clientes” y luego Registrar Clientes. El administrador debe ingresar los datos correspondientes al cliente, y por ultimo debe presionar “Agregar Cliente”.



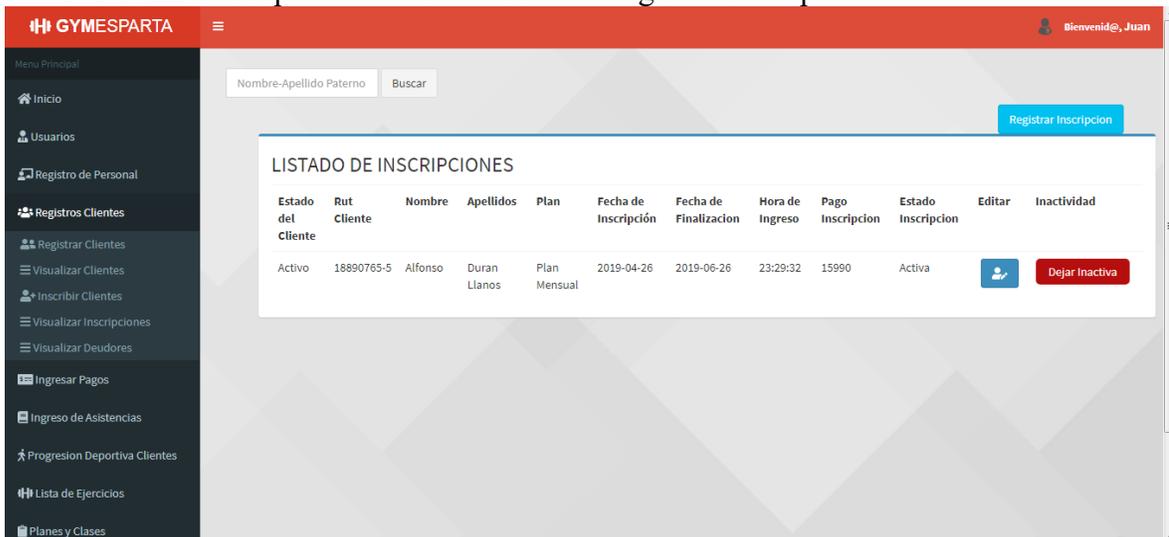
10) A continuación, se visualiza el listado de clientes registrados, junto con las opciones respectivas, como agregar otro cliente, ver los datos del cliente a través de una búsqueda por nombre-apellido o por tipo (inactivo, activo), editar los datos, exportar y dejar inactivo en caso de que no siga asistiendo a la institución.



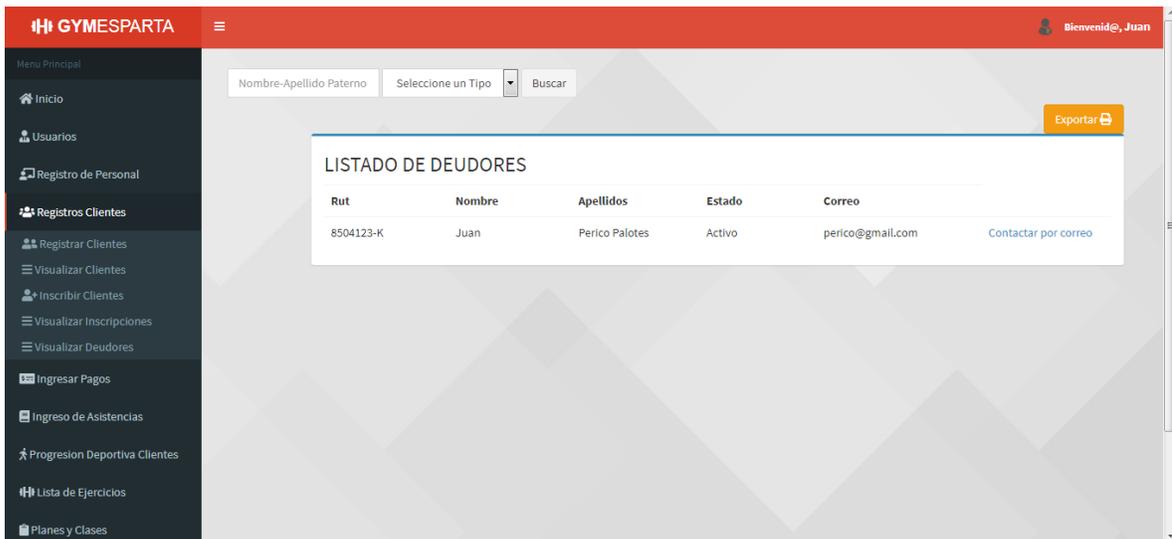
11) La siguiente pantalla muestra los reportes que se pueden visualizar a través de presionar sobre el botón “Exportar” en el menú de visualizar Clientes, imprimir o guardar correspondiente a los datos de los clientes registrados en el sistema.



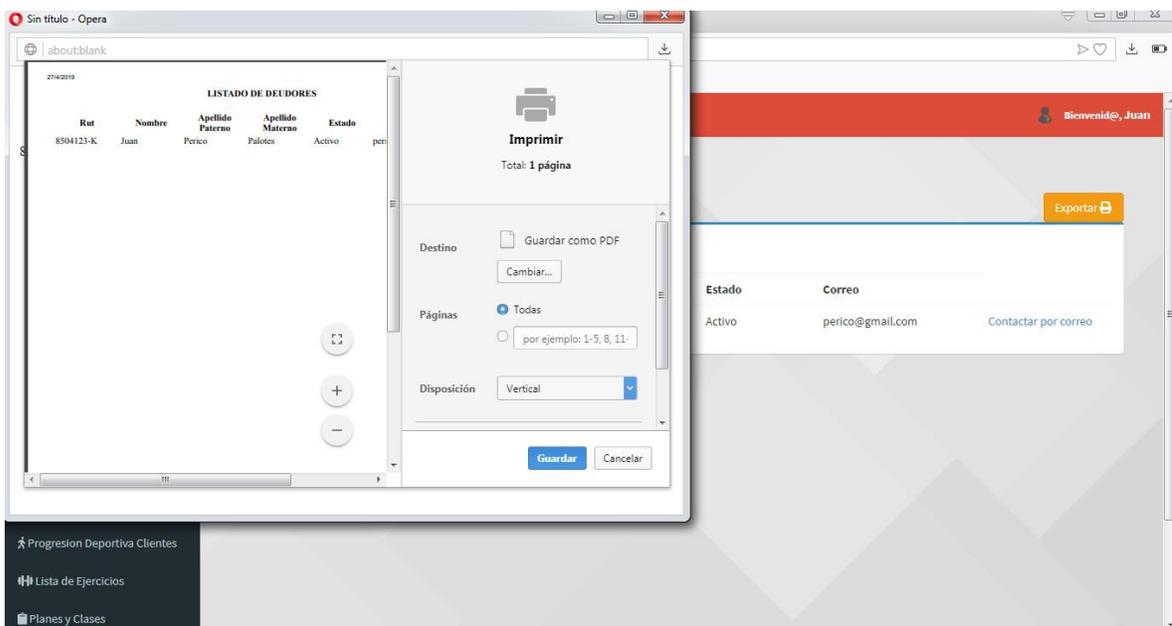
12) La siguiente pantalla muestra la opción de registrar inscripciones a clientes, el cual como primer lugar se debe seleccionar “Registros Clientes” y luego Inscribir Clientes. El administrador debe seleccionar un Cliente o ingresar el nombre junto con sus apellidos en el buscador, debe seleccionar además la fecha, y el plan a contratar. Por ultimo debe presionar sobre el botón “Registrar Inscripción”.



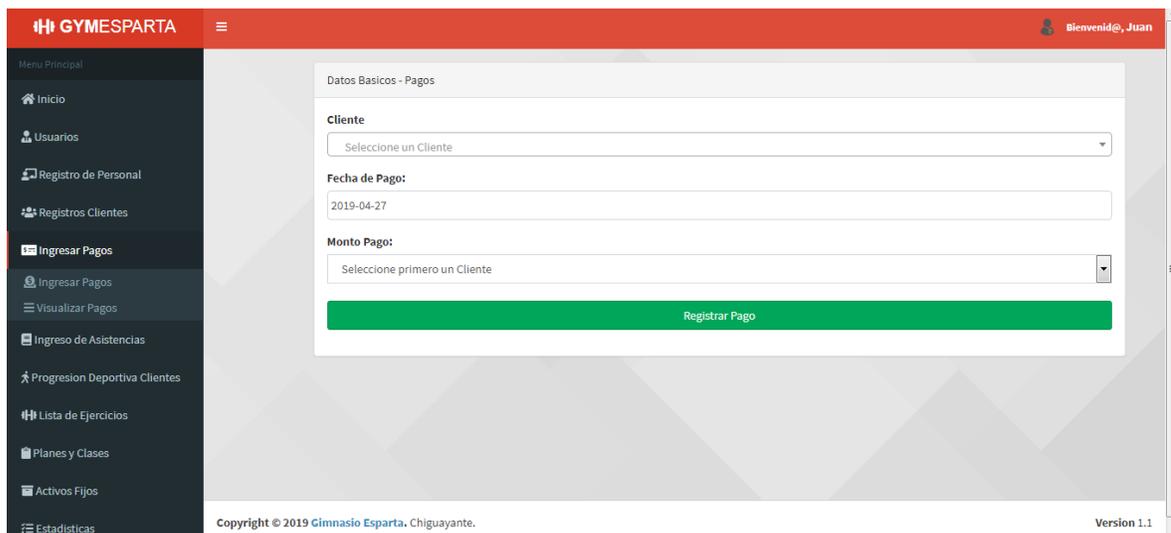
13) A continuación, se visualiza el listado de inscripciones, junto con las opciones respectivas, como agregar otra inscripción, editar la inscripción, o dejar inactiva una inscripción para ingresar una nueva inscripción a un cliente.



14) La siguiente pantalla muestra el listado de deudores junto con la opción de contactarlo por correo para informar sobre su atraso y visualizar los reportes a través del botón “Exportar”.

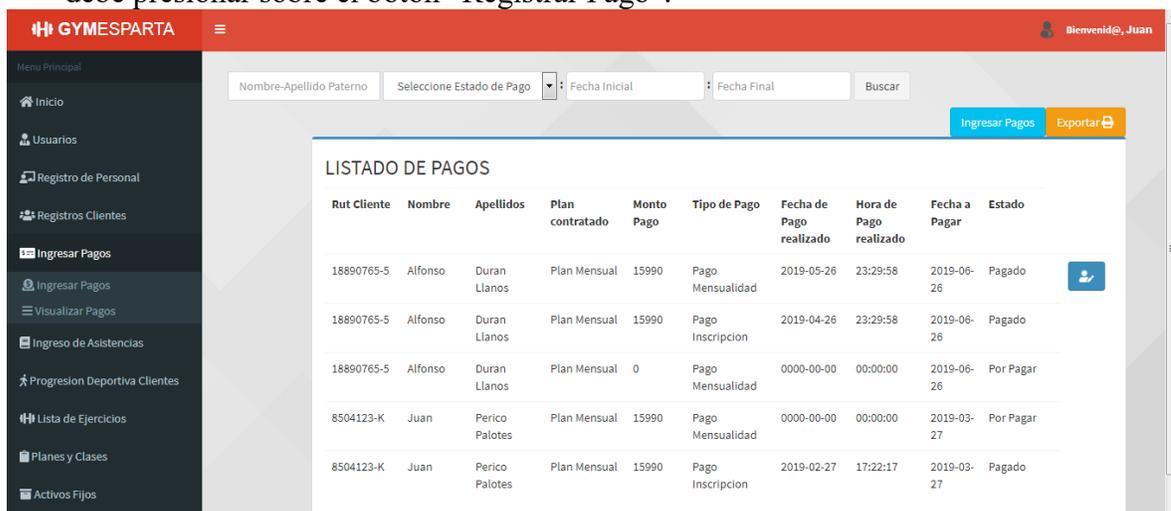


15) La siguiente pantalla muestra los reportes que se pueden visualizar a través de presionar sobre el botón “Exportar” en el menú de visualizar Deudores, imprimir o guardar correspondiente a los datos de los deudores.

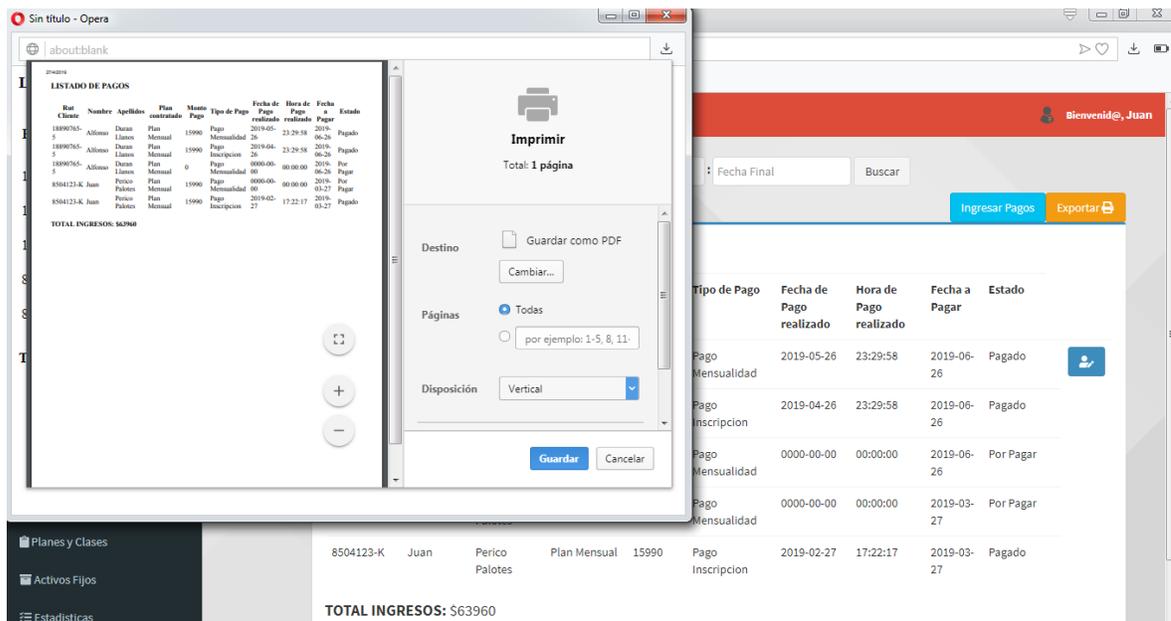


16) La siguiente pantalla muestra la opción de registrar pagos a clientes, el cual como primer lugar se debe seleccionar “Pagos” y luego Ingresar Pagos.

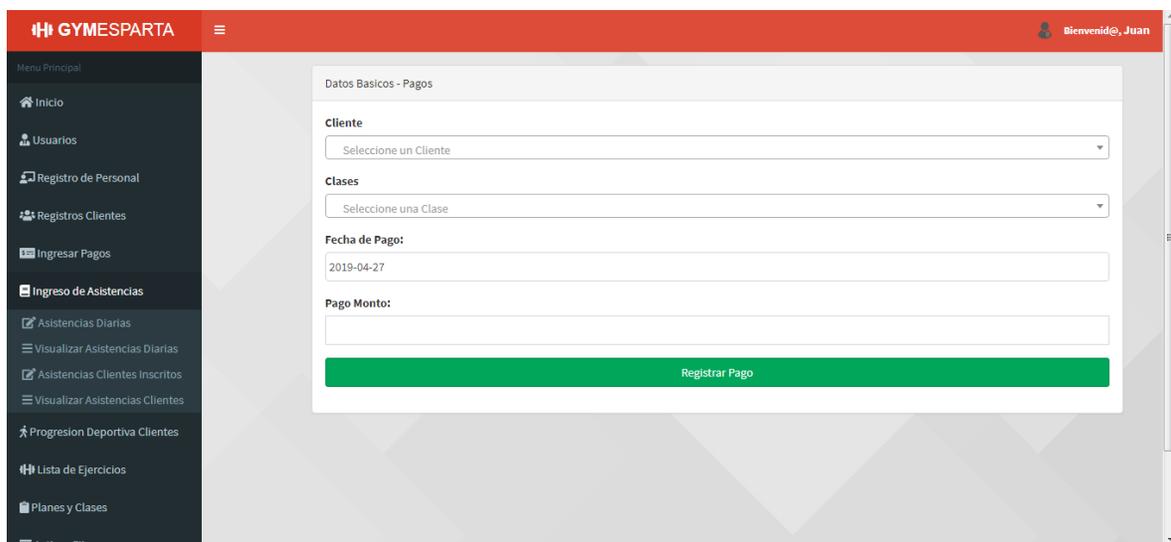
El administrador debe seleccionar un Cliente o ingresar el nombre junto con sus apellidos en el buscador, debe seleccionar además la fecha de pago. Por ultimo debe presionar sobre el botón “Registrar Pago”.



17) La siguiente pantalla muestra el listado de pagos junto con la opción de editarlo, ingresar otro pago, o visualizar los reportes a través del botón “Exportar”. Por otra parte se pueden visualizar los ingresos por rango de fechas, buscar pagos por estado (pagados, por pagar).



18) La siguiente pantalla muestra los reportes que se pueden visualizar a través de presionar sobre el botón “Exportar” en el menú de visualizar Pagos, imprimir o guardar correspondiente a los datos de los pagos.

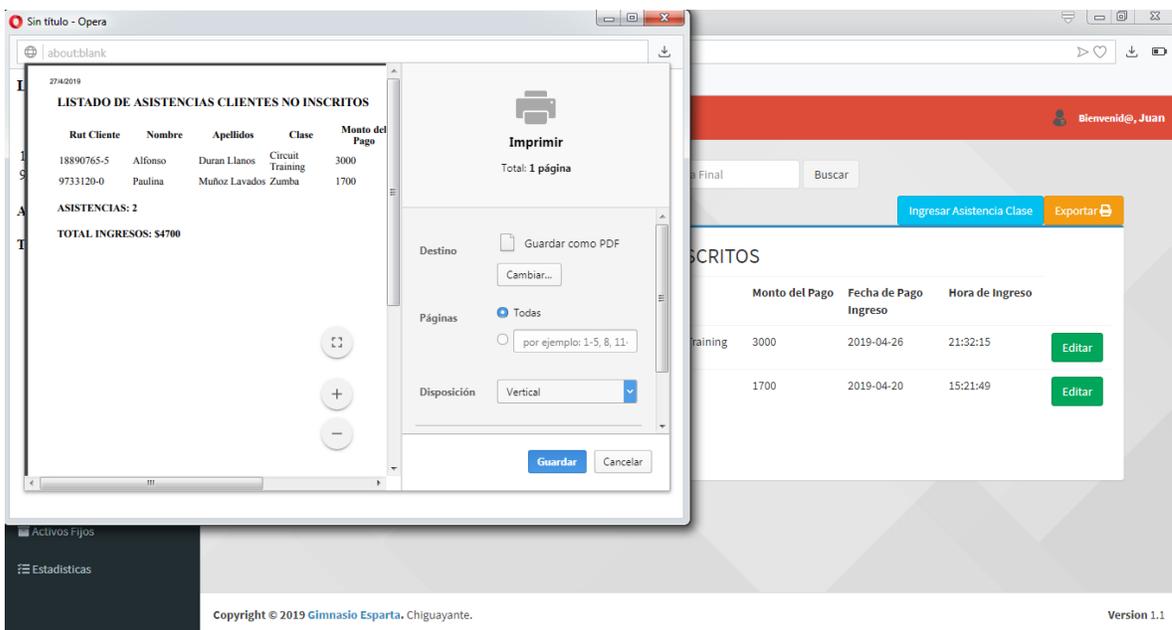


19) La siguiente pantalla muestra la opción de registrar asistencias a clases aquellos que no registran inscripción, el cual como primer lugar se debe seleccionar “Ingreso de Asistencias” y luego Asistencias Diarias.

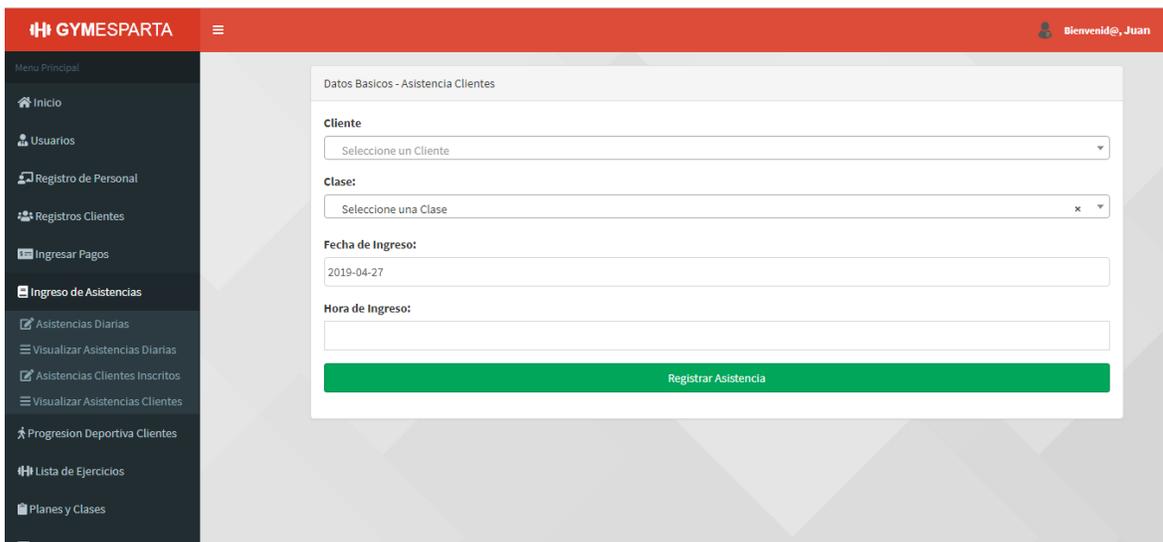
El administrador debe seleccionar un Cliente o ingresar el nombre junto con sus apellidos en el buscador, debe seleccionar además la fecha de pago y la clase a la que asistirá. Por ultimo debe presionar sobre el botón “Registrar Asistencia”.



20) La siguiente pantalla muestra el listado de asistencias junto con la opción de editarlo, ingresar otra asistencia, o visualizar los reportes a través del botón “Exportar”. Por otra parte se pueden visualizar los ingresos y asistencias por rango de fechas o por selección de clase.



21) La siguiente pantalla muestra los reportes que se pueden visualizar a través de presionar sobre el botón “Exportar” en el menú de visualizar Asistencias Diarias, imprimir o guardar correspondiente a los datos de las asistencias.

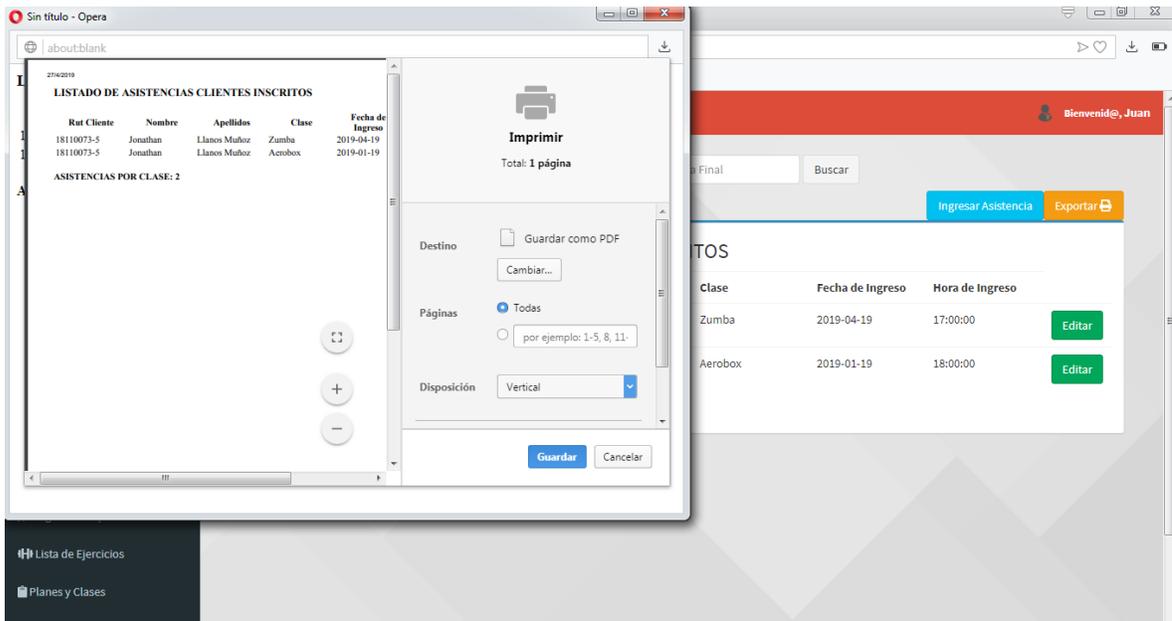


22) La siguiente pantalla muestra la opción de registrar asistencias a clases aquellos que registran inscripción, el cual como primer lugar se debe seleccionar “Ingreso de Asistencias” y luego Asistencias Clientes Inscritos.

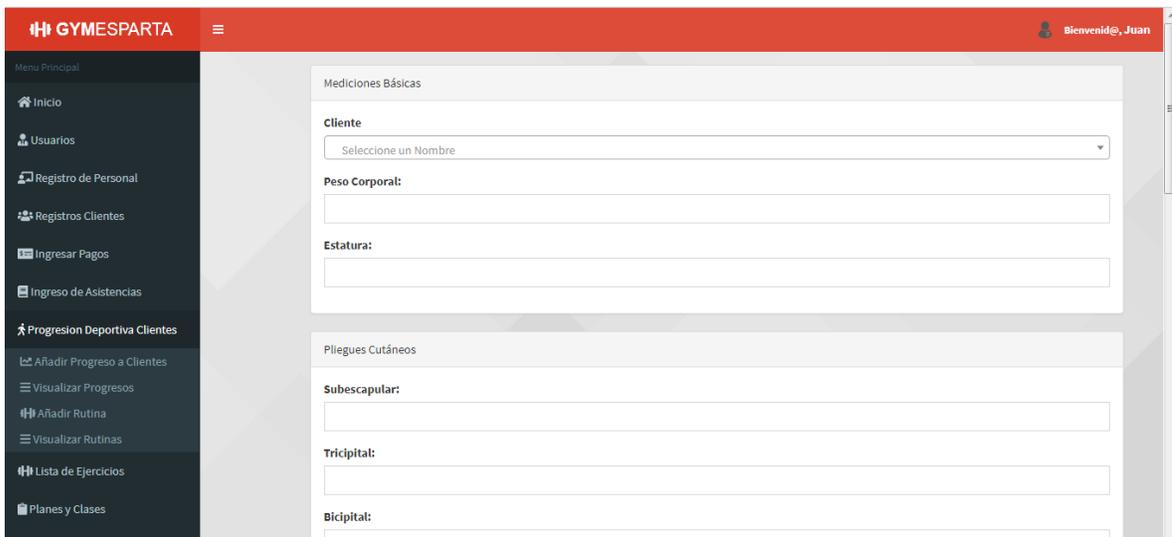
El administrador debe seleccionar un Cliente o ingresar el nombre junto con sus apellidos en el buscador, debe seleccionar además la fecha de pago, hora de ingreso y la clase a la que asistirá. Por ultimo debe presionar sobre el botón “Registrar Asistencia”.



23) La siguiente pantalla muestra el listado de asistencias junto con la opción de editarlo, ingresar otra asistencia, o visualizar los reportes a través del botón “Exportar”. Por otra parte se pueden visualizar las asistencias por rango de fechas o por selección de clase.

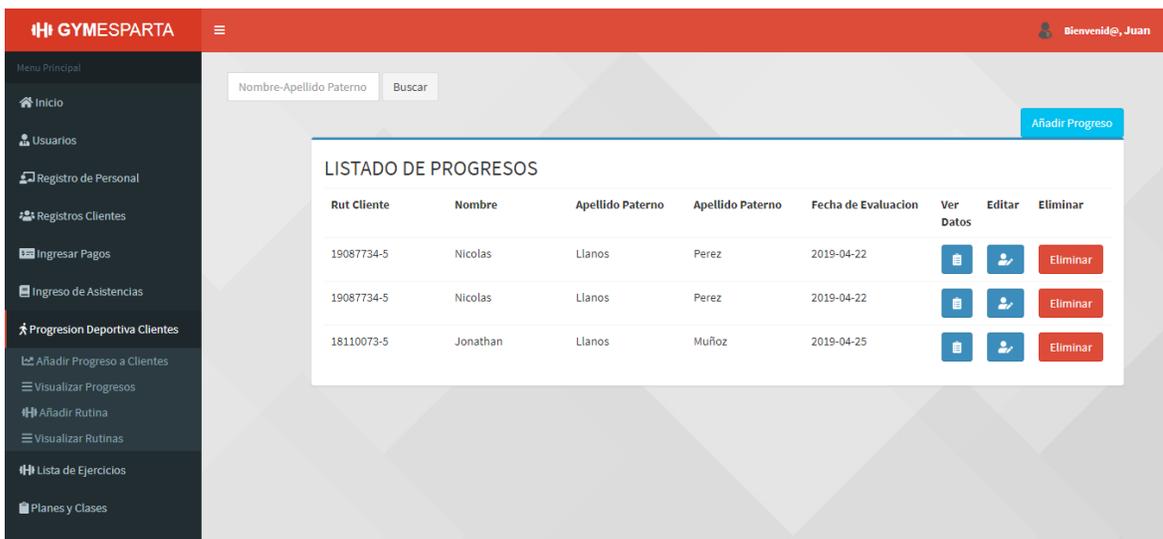


24) La siguiente pantalla muestra los reportes que se pueden visualizar a través de presionar sobre el botón “Exportar” en el menú de visualizar Asistencias Clientes Inscritos, imprimir o guardar correspondiente a los datos de las asistencias.

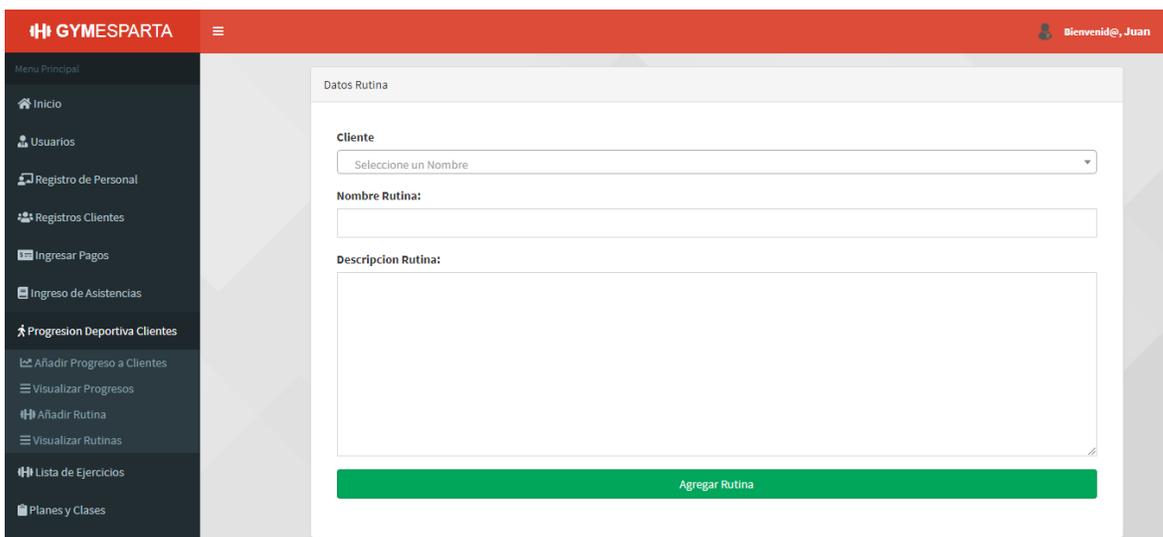


25) La siguiente pantalla muestra el ingreso de Progresos Deportivos a clientes. Se debe seleccionar la opción “Progresión Deportiva Clientes” y luego “Añadir Progreso a Clientes”.

Como primer lugar, se debe seleccionar el cliente o ingresar el nombre con sus apellidos respectivos. A posterior, se deben ingresar las mediciones en los campos correspondientes, como peso corporal, estatura, etc.

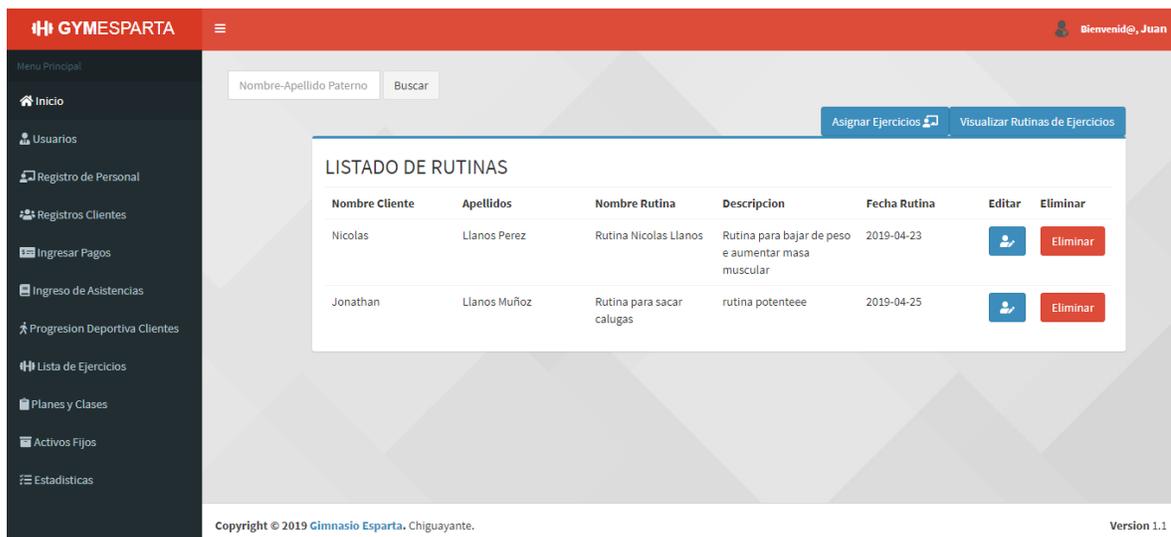


26) La siguiente pantalla muestra el listado de progresiones deportivas junto con la opción de editarlo, ingresar otra progresión, o visualizar los datos de las progresiones y eliminar.

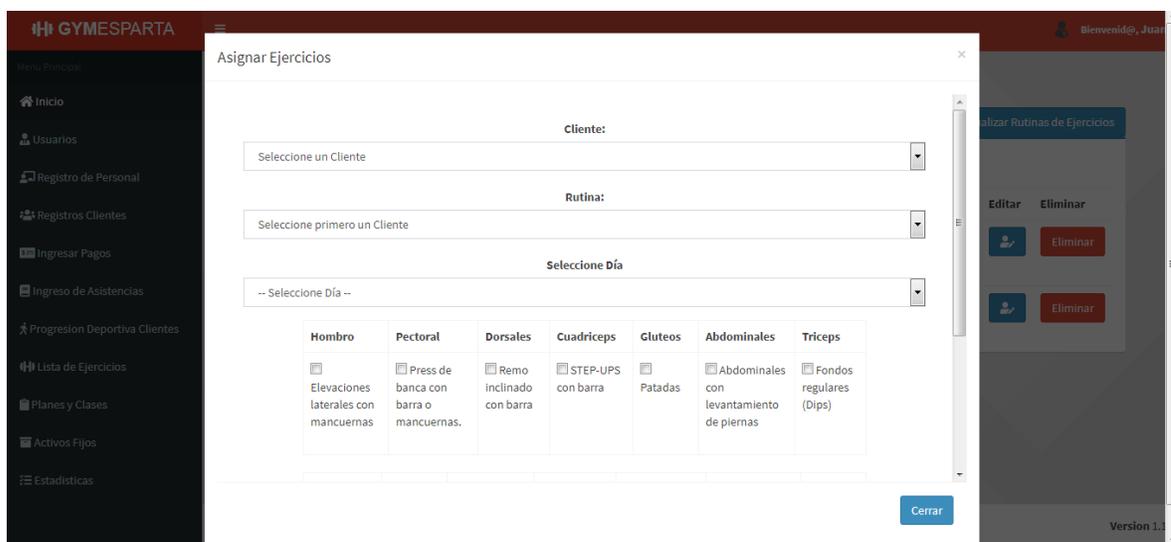


27) La siguiente pantalla muestra la opción de registrar rutinas a clientes, el cual como primer lugar se debe seleccionar “Progresión Deportiva Clientes” y luego “Añadir Rutina”.

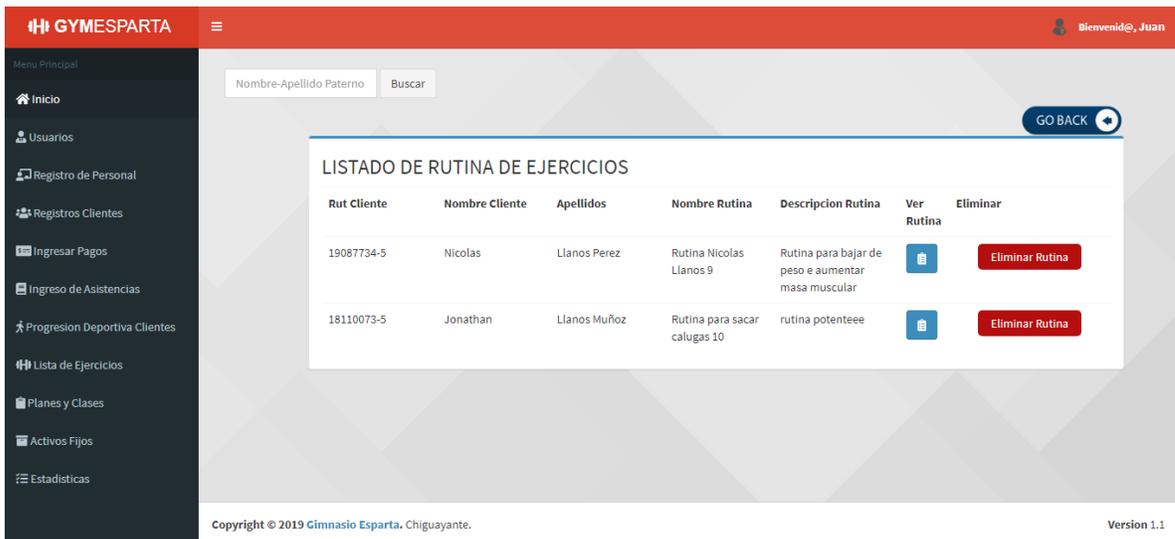
El administrador debe seleccionar un Cliente o ingresar el nombre junto con sus apellidos en el buscador, debe ingresar, además, un Nombre a la rutina y una descripción. Por ultimo debe presionar sobre el botón “Agregar Rutina”.



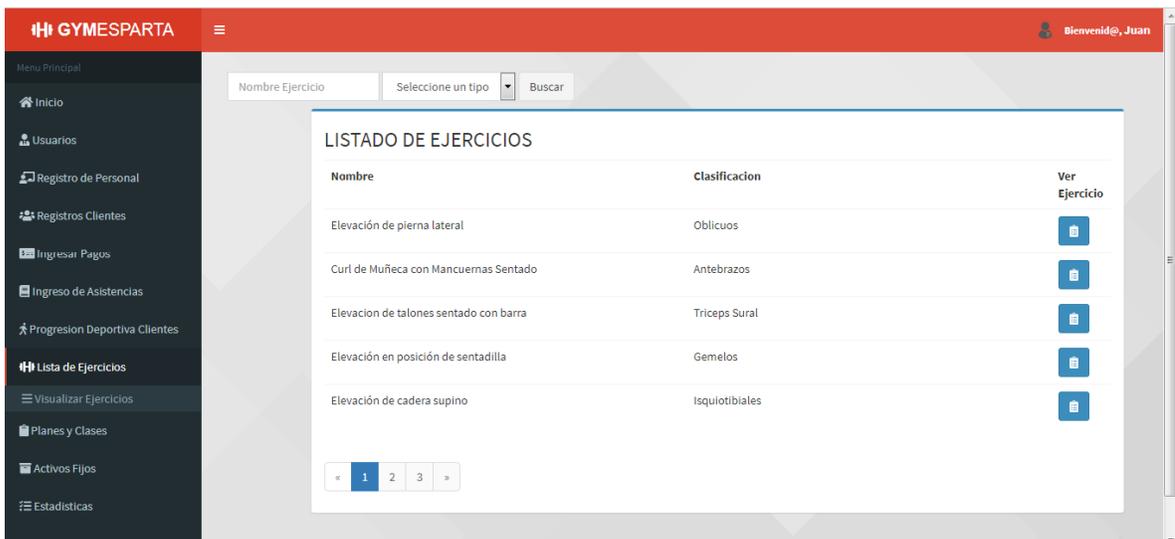
28) La siguiente pantalla muestra el listado de rutinas junto con la opción de editarlo, eliminarlo, asignarle ejercicios a las rutinas y por último, visualizarlas.



29) La siguiente pantalla muestra la pantalla de Asignar Ejercicios a rutinas, lo cual se debe acceder a través de “Visualizar Rutinas”. A posterior, se debe seleccionar un cliente o ingresar el nombre con apellidos, seleccionar la Rutina y por último los ejercicios a realizar.



30) En la siguiente imagen, se visualizan las rutinas junto con sus respectivos ejercicios, lo cual se debe acceder mediante el botón “Ver Rutina”, por otra parte se presenta la opción de Eliminar la Rutina respectiva.



**GYMSPARTA** Bienvenid@, Juan

Nombre Ejercicio

### Descripción Ejercicio

#### Ejercicio: Elevación de pierna lateral



**Descripción:** La elevación de la pierna debe ser controlada en todo momento, no realizar movimientos bruscos con el fin de no perjudicar a los abductores. Como se puede ver en el vídeo es conveniente realizarlo cerca de una pared para mantener la lumbar lo máximo posible.

Cerrar

Ver Ejercicio

**GYMSPARTA** Bienvenid@, Juan

Nombre Ejercicio

### Descripción Ejercicio

coiocano bien el cuerpo.

Numero de Series: 3

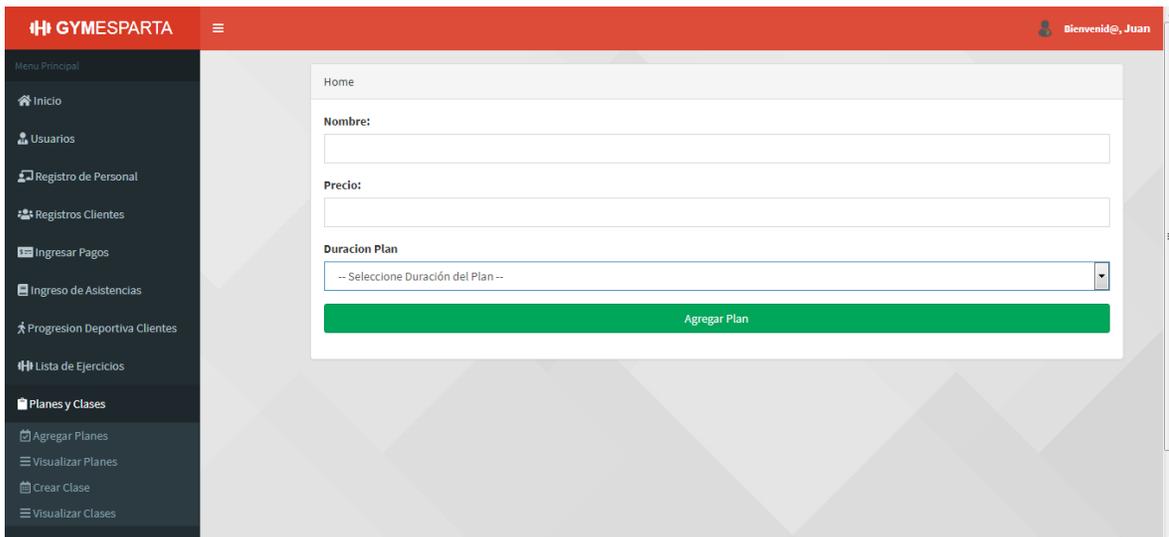
Intervalo de Repeticiones: 15-20



Cerrar

Ver Ejercicio

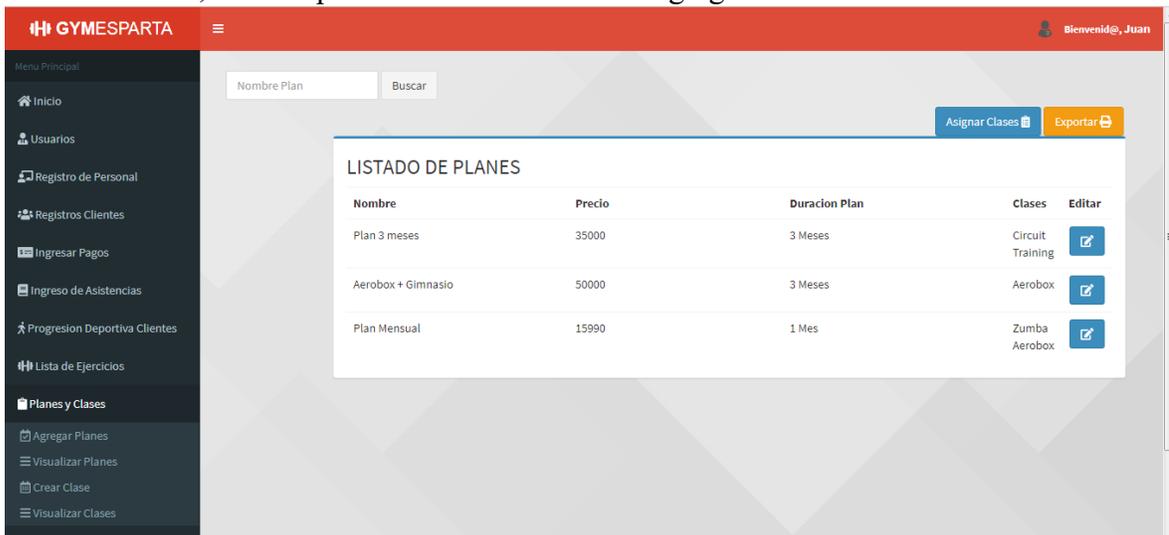
1



31) La siguiente pantalla muestra la opción de registrar planes, el cual como primer lugar se debe seleccionar “Planes y Clases” y luego “Agregar Planes”.

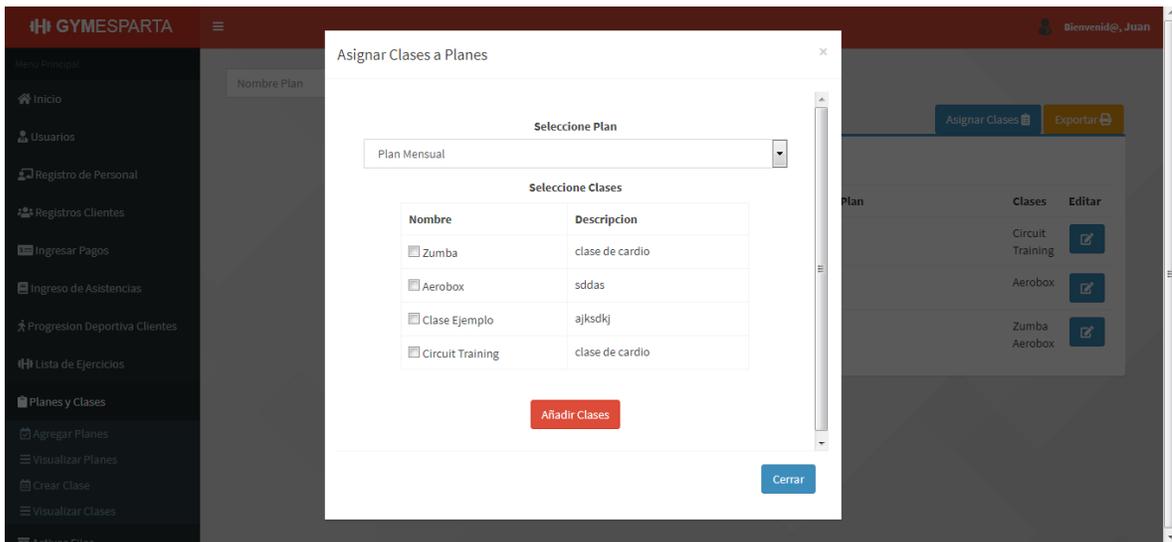
El administrador debe ingresar un nombre, un precio y la duración del plan en meses.

Por último, se debe presionar sobre el botón “Agregar Plan”.

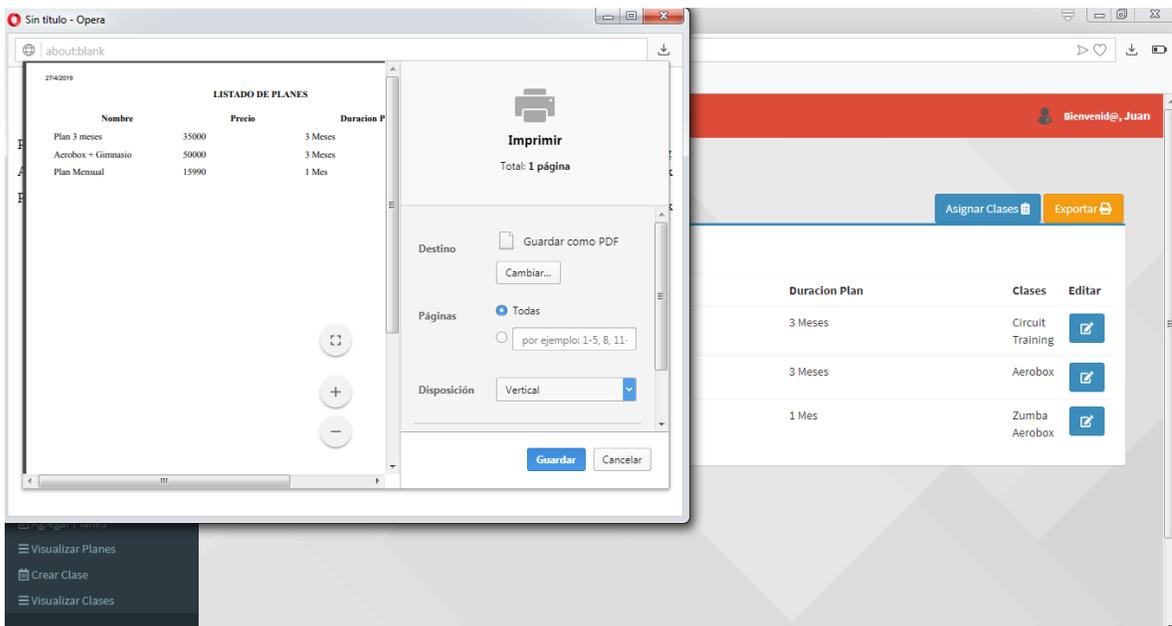


32) La siguiente pantalla muestra el listado de planes junto con la opción de editarlo, ingresar otra clase, o visualizar los reportes a través del botón “Exportar”.

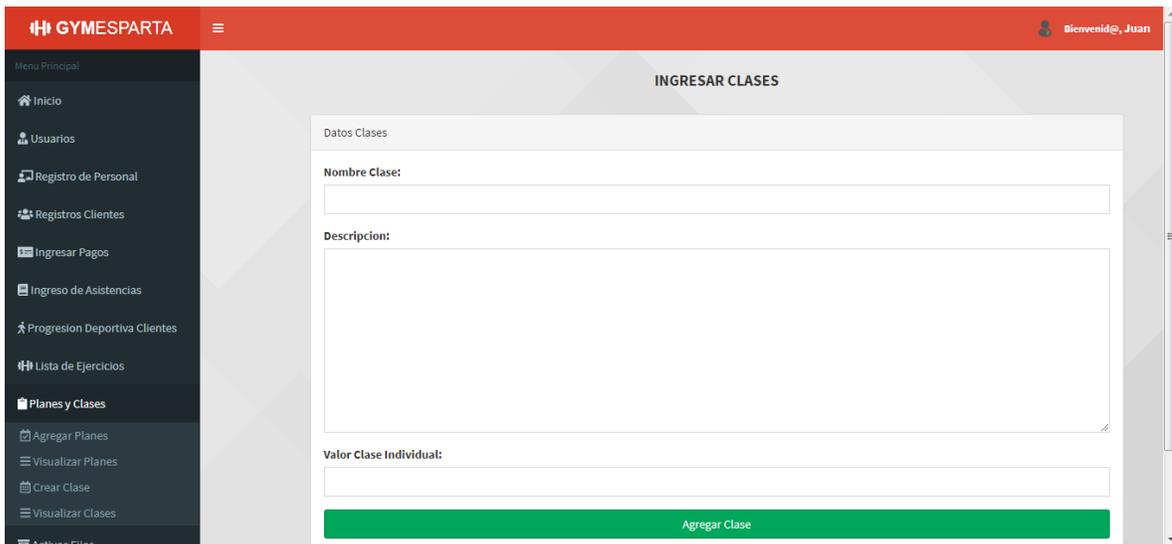
Por otra parte, se pueden asignar las clases a través del botón “Asignar “Clases””.



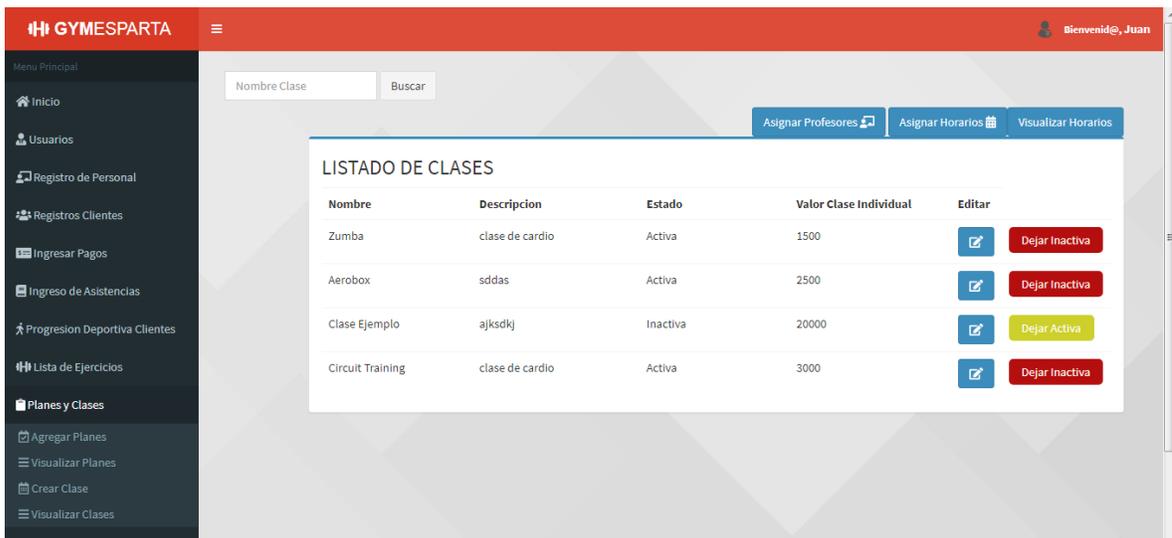
33) La siguiente pantalla muestra la opción de Ingresar las clases respectivas a un plan. Primeramente, se debe seleccionar un plan y luego checkear las clases.



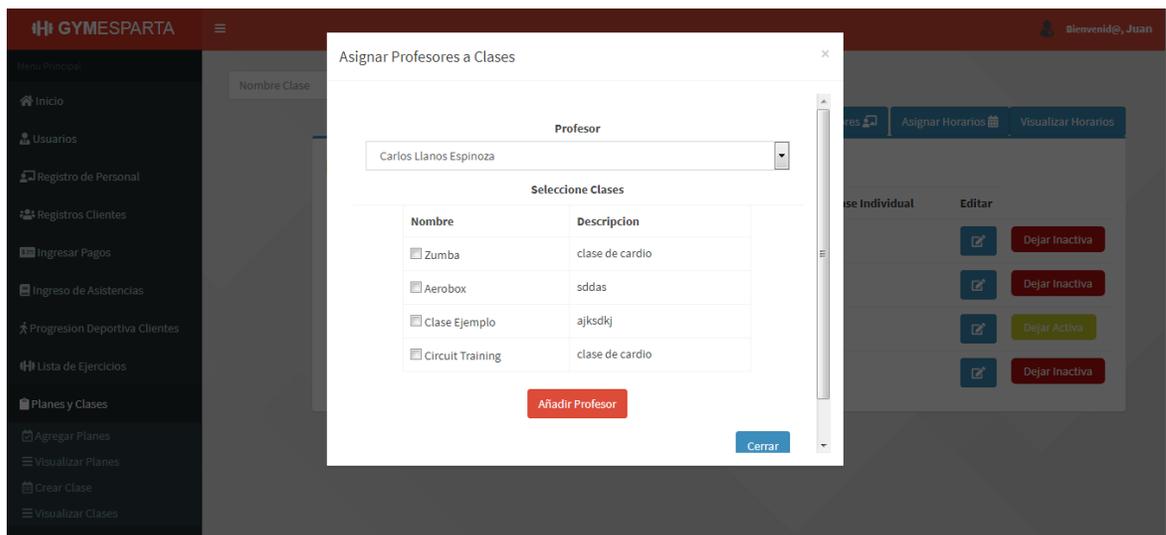
34) La siguiente pantalla muestra los reportes que se pueden visualizar a través de presionar sobre el botón “Exportar” en el menú de visualizar Planes, imprimir o guardar correspondiente a los datos de los planes.



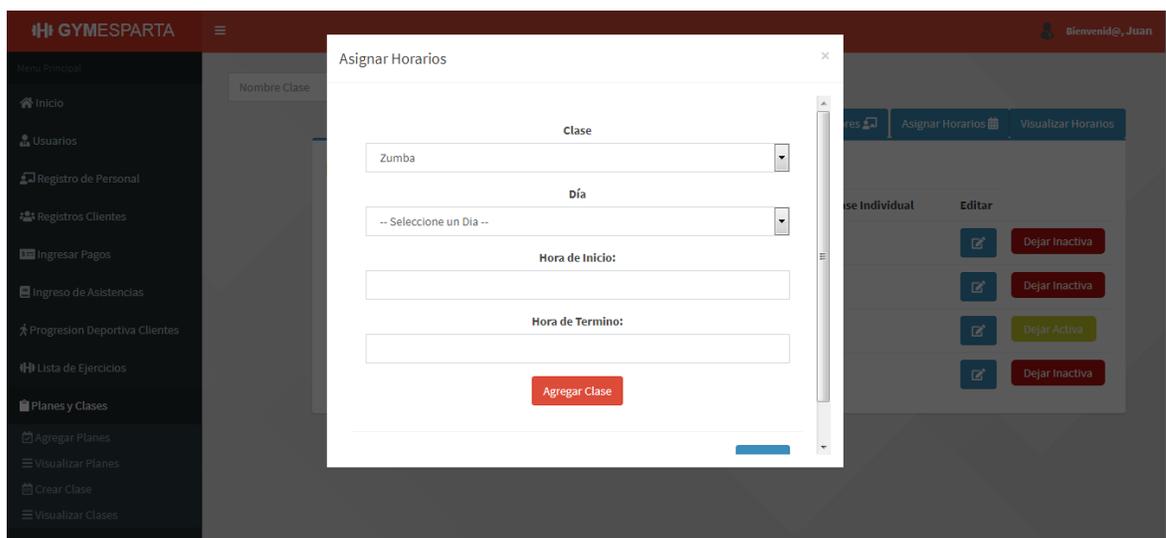
35) La siguiente pantalla muestra la opción de registrar clases, el cual como primer lugar se debe ingresar un nombre, luego una descripción y un valor a la clase para quienes desean asistir y no cuentan con una inscripción. Por último, se debe presionar sobre “Agregar Clase”.



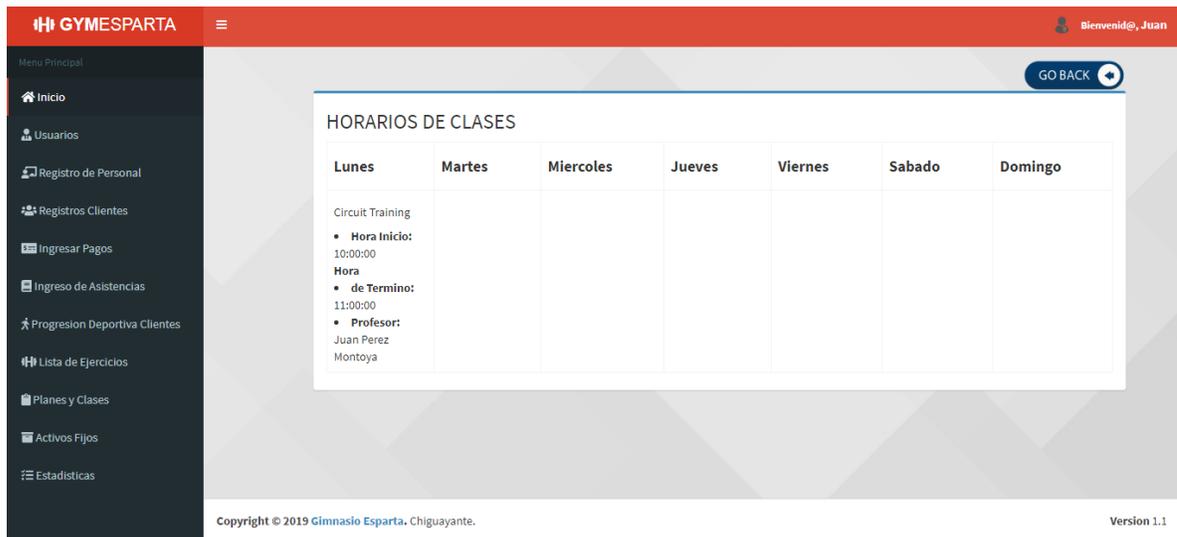
36) La siguiente pantalla muestra el listado de clases junto con la opción de editarla, asignarle un estado (activo e inactivo), asignar profesores a las clases, asignar horarios y visualizar horarios.



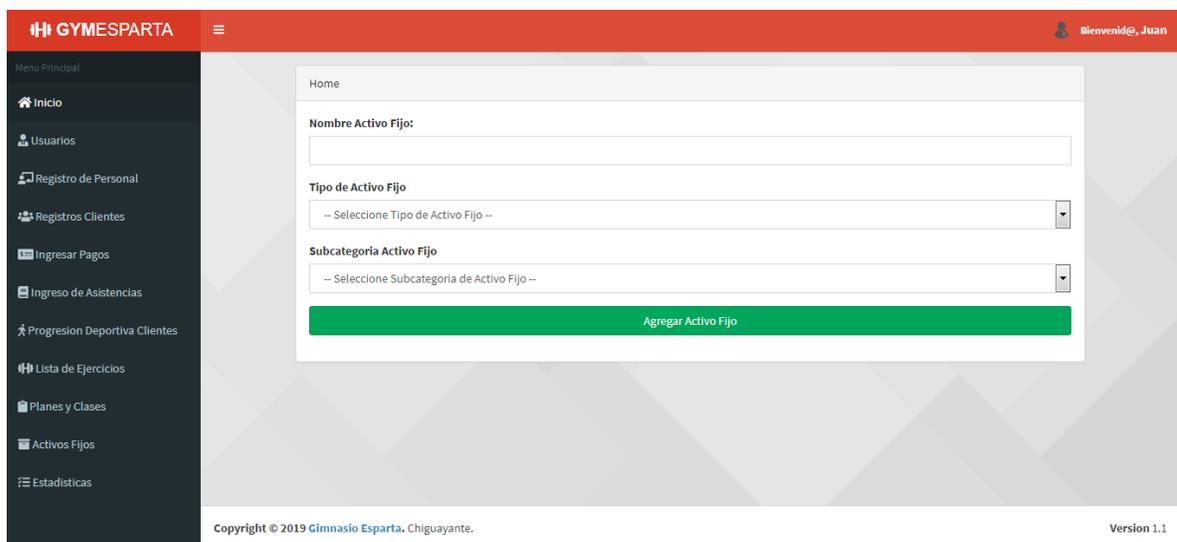
37) La siguiente pantalla muestra la opción de asignar profesores a las clases respectivas. Primeramente, se debe seleccionar un profesor y luego checkear las clases de las cuales se hará cargo.



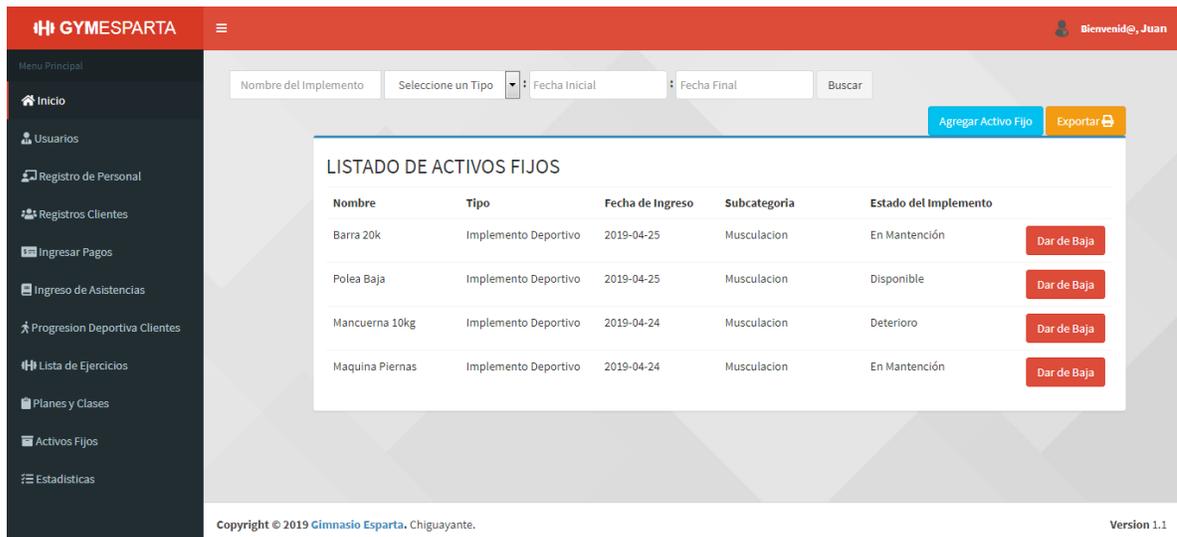
38) La siguiente pantalla muestra la opción de asignar horarios a clases, el cual se ingresa a esta opción a través de “Visualizar Clases”. El administrador debe seleccionar una clase, seleccionar el día y a posterior ingresar el horario de inicio y el horario de término.



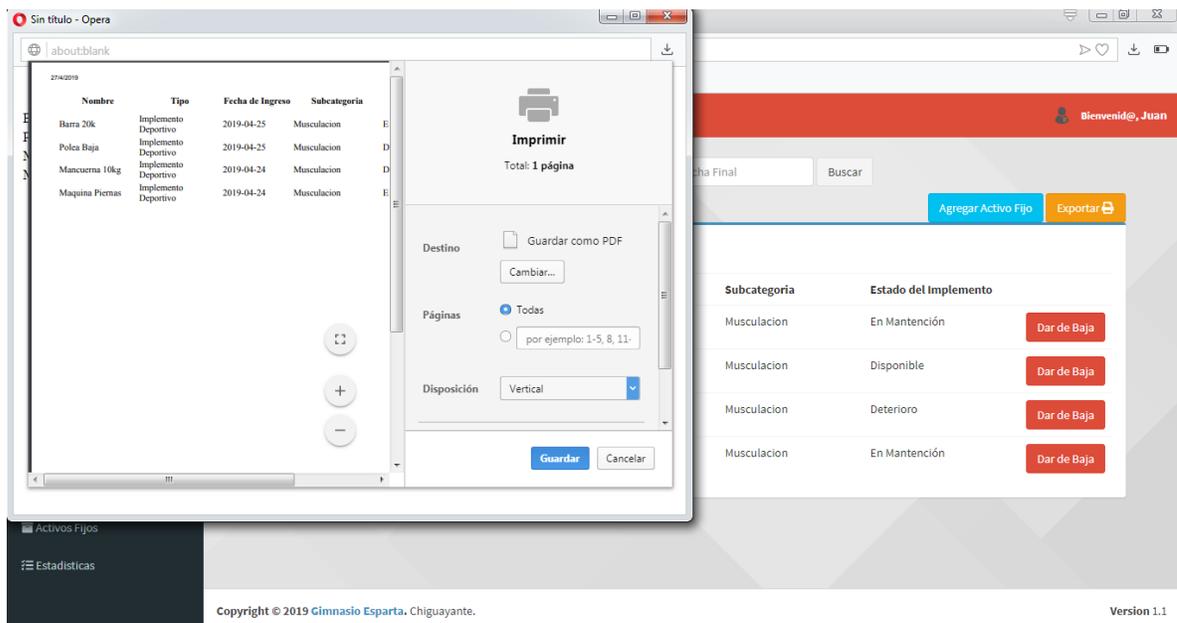
39) La siguiente pantalla muestra los horarios de las clases, luego de haber seleccionado el botón “Visualizar Horarios” en la visualización de clases.



40) La siguiente pantalla muestra la opción de registrar activos fijos, el cual como primer lugar se debe seleccionar “Activos Fijos” y luego Añadir Activos Fijos. El administrador debe ingresar un nombre al activo fijo, luego debe seleccionar el tipo de Activo Fijo y por último, la subcategoría.



41) La siguiente pantalla muestra el listado de activos fijos junto con la opción de darlo de baja, agregar activos fijos, o visualizar los reportes a través del botón “Exportar”. Por otra parte se pueden visualizar los activos fijos que fueron ingresados por rango de fechas, buscar pagos por Tipo o por Nombre.

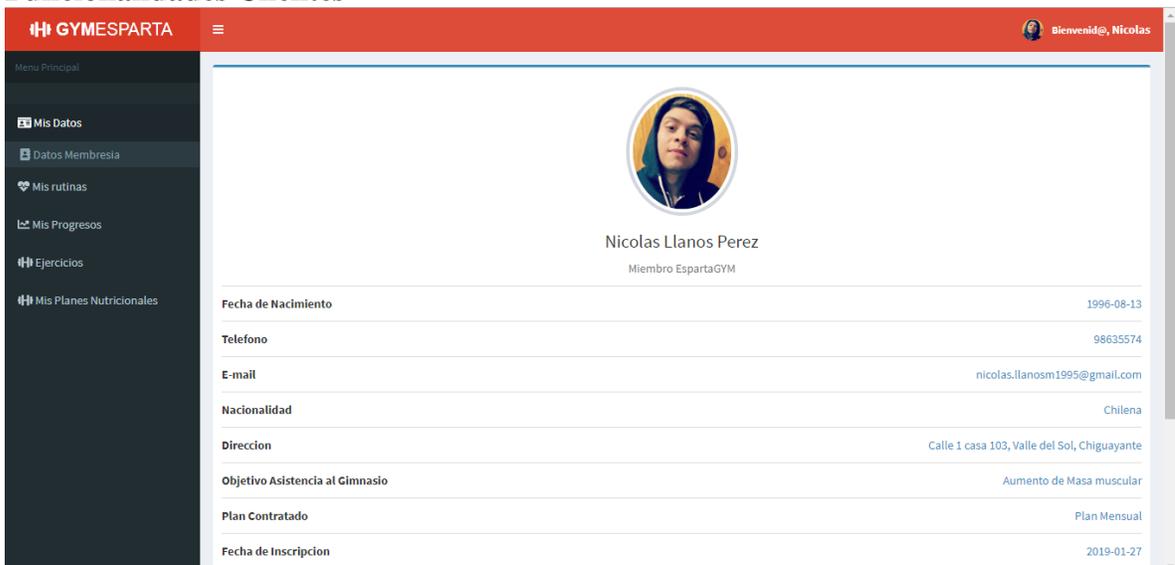


42) La siguiente pantalla muestra los reportes que se pueden visualizar a través de presionar sobre el botón “Exportar” en el menú de visualizar Activos Fijos, imprimir o guardar correspondiente a los datos de los activos fijos.



43) La siguiente pantalla muestra las estadísticas generales, como ingresos totales, cantidad de clientes activos y clientes inactivos, junto con la opción de visualizar los listados correspondientes.

### Funcionalidades Clientes



1) La siguiente pantalla muestra los datos de membresía del cliente, junto con los datos personales. Se debe ingresar a través del menú en la opción “Mis Datos” – “Datos Membresía”.

**GYMESPARTA** Bienvenid@, Nicolas

Menu Principal

- Mis Datos
- Mis rutinas
- Listado de Rutinas
- Mis Progresos
- Ejercicios
- Mis Planes Nutricionales

### LISTADO DE RUTINAS

Nombre Rutina	Descripción Rutina	Fecha de Rutina	Ver Rutina de Ejercicios
Rutina Nicolas Llanos 9	Rutina para bajar de peso e aumentar masa muscular	2019-04-23	
Rutina para sacar calugas 10	rutina potenteee	2019-04-25	

Copyright © 2019 Gimnasio Esparta. Chiguayante. Version 1.1

- 2) La siguiente pantalla muestra el listado de rutinas que fueron asignadas al respectivo cliente, junto con la visualización de los ejercicios que debe realizar. Se debe acceder a este a través del menú “Mis Rutinas” – “Listado de Rutinas”.

**GYMESPARTA** Bienvenid@, Nicolas

Menu Principal

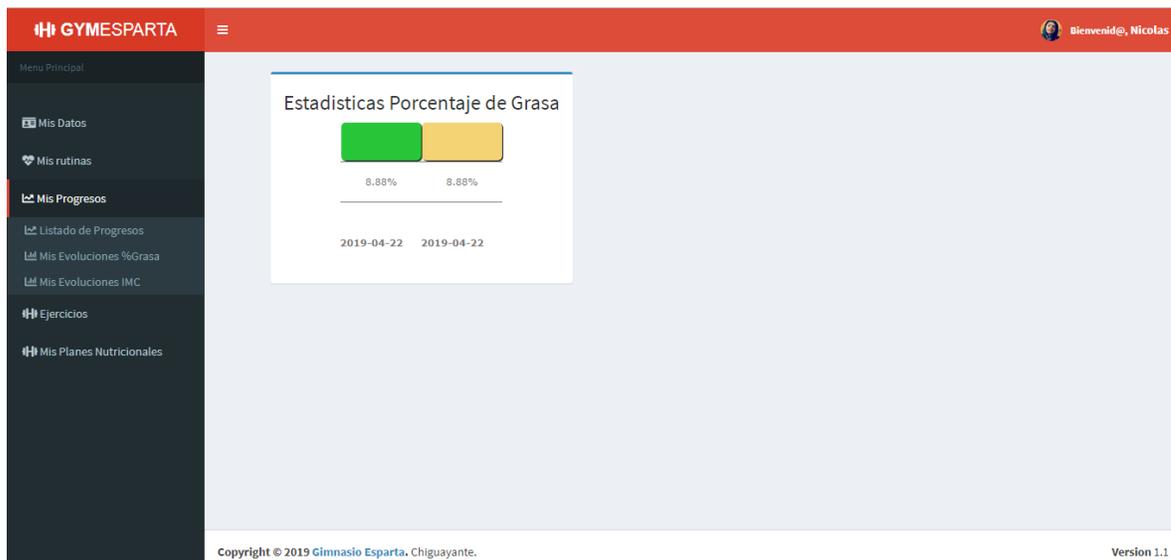
- Mis Datos
- Mis rutinas
- Mis Progresos
- Listado de Progresos
- Mis Evoluciones %Grasa
- Mis Evoluciones IMC
- Ejercicios
- Mis Planes Nutricionales

### LISTADO DE PROGRESOS

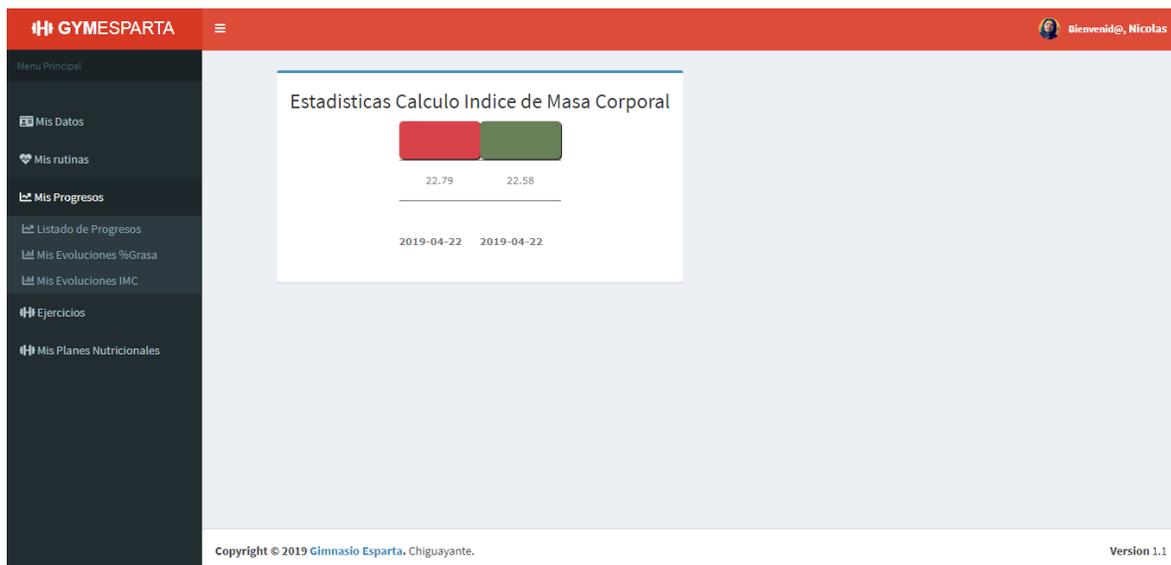
Rut Cliente	Nombre	Apellido Paterno	Apellido Paterno	Fecha de Evaluacion	
19087734-5	Nicolas	Llanos	Perez	2019-04-22	<a href="#">Ver Datos</a>
19087734-5	Nicolas	Llanos	Perez	2019-04-22	<a href="#">Ver Datos</a>

Copyright © 2019 Gimnasio Esparta. Chiguayante. Version 1.1

- 3) La siguiente pantalla muestra el listado de progresos que fueron asignadas al respectivo cliente, junto con la visualización de las mediciones, etc. Se debe acceder a este a través del menú “Mis Progresos” – “Listado de Progresos”.



- 4) La siguiente pantalla muestra gráficos de las respectivas evoluciones que ha tenido el cliente con respecto a su porcentaje de grasa. Se debe acceder a este a través del menú en la opción “Mis Evoluciones %grasa”.



- 5) La siguiente pantalla muestra gráficos de las respectivas evoluciones que ha tenido el cliente con respecto a su índice de masa corporal. Se debe acceder a este a través del menú en la opción “Mis Evoluciones IMC”.

**GYMESPARTA** Bienvenido, Nicolas

Nombre Ejercicio  Seleccione un tipo

### LISTADO DE EJERCICIOS

Nombre	Clasificación	Ver Ejercicio
Elevación de pierna lateral	Oblicuos	<input type="button" value="📄"/>
Curli de Muñeca con Mancuernas Sentado	Antebrazos	<input type="button" value="📄"/>
Elevación de talones sentado con barra	Triceps Sural	<input type="button" value="📄"/>
Elevación en posición de sentadilla	Gemelos	<input type="button" value="📄"/>
Elevación de cadera supino	Isquiotibiales	<input type="button" value="📄"/>

« 1 2 3 »

Copyright © 2019 Gimnasio Esparta, Chiguayante. Version 1.1

**GYMESPARTA** Bienvenido, Nicolas

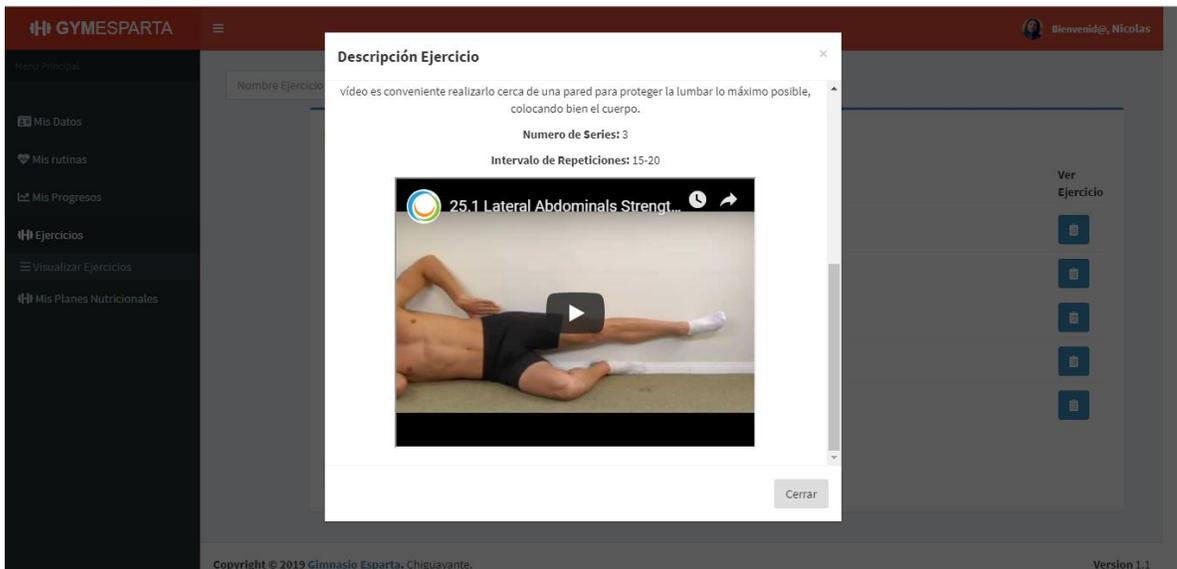
### Descripción Ejercicio

**Ejercicio: Elevación de pierna lateral**

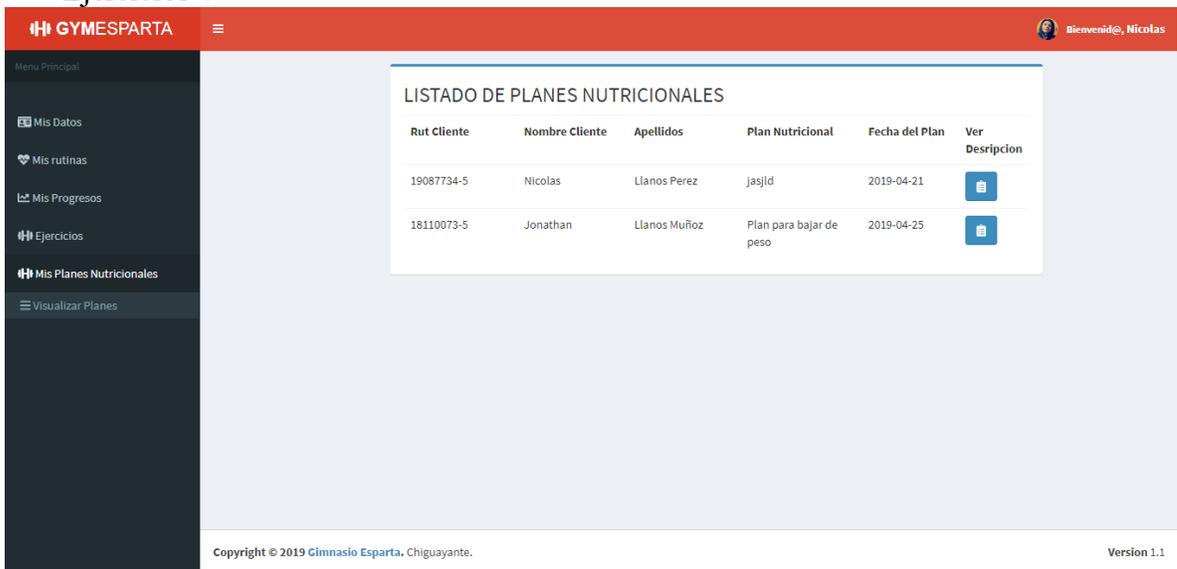


**Descripción:** La elevación de la pierna debe ser controlada en todo momento, no realizar movimientos bruscos con el fin de no perjudicar a los abductores. Como se puede ver en el vídeo es conveniente realizarlo cerca de una pared para proteger la lumbar lo máximo posible, colocando bien el cuerpo.

Copyright © 2019 Gimnasio Esparta, Chiguayante. Version 1.1



- 6) En la siguiente imagen, se visualizan los ejercicios junto con sus respectivos datos (video, imagen, descripción, número de series, intervalo de repeticiones). Se debe acceder a través del menú en la opción “Lista de Ejercicios” – “Visualizar Ejercicios”.



- 7) En la siguiente imagen, se visualizan los planes nutricionales que fueron asignados al cliente respectivo, junto con la descripción de esta.

## Funcionalidades Profesores

- 1) En la siguiente imagen, se visualiza la pantalla de asignación de progresos físicos a clientes, el cual se accede desde el menú “Progreso Clientes” – “Añadir Progreso a Clientes”. Se deben ingresar las mediciones correspondientes y luego presionar sobre el botón “Añadir progreso”.

Rut Cliente	Nombre	Apellido Paterno	Apellido Paterno	Fecha de Evaluacion	Ver Datos	Editar	Eliminar
19087734-5	Nicolas	Llanos	Perez	2019-04-22			
19087734-5	Nicolas	Llanos	Perez	2019-04-22			
18110073-5	Jonathan	Llanos	Muñoz	2019-04-25			
19087734-5	Nicolas	Llanos	Perez	2019-04-29			
19087734-5	Nicolas	Llanos	Perez	2019-04-29			

- 2) En la siguiente imagen, se visualiza el listado de progresos de los clientes, junto con sus respectivas opciones de ver los datos, editar, añadir un nuevo progreso o eliminar.

Se accede a esta a través del menú “Visualizar Progressos”.

Datos Rutina

Cliente

Nombre Rutina:

Descripcion Rutina:

Agregar Rutina

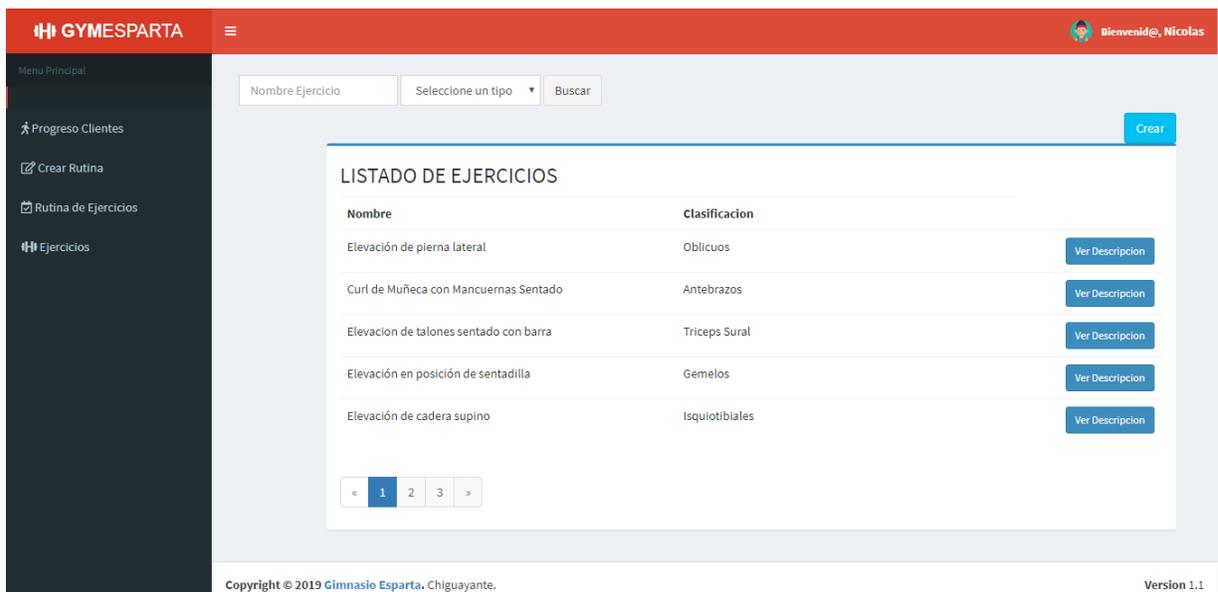
3) En la siguiente imagen, se visualiza la pantalla de asignación de progresos rutinas a clientes, el cual se accede desde el menú “Crear Rutina” – “Añadir Rutina”. Se debe seleccionar un cliente y luego ingresar los datos correspondientes a la rutina (nombre rutina, descripción).

LISTADO DE RUTINAS

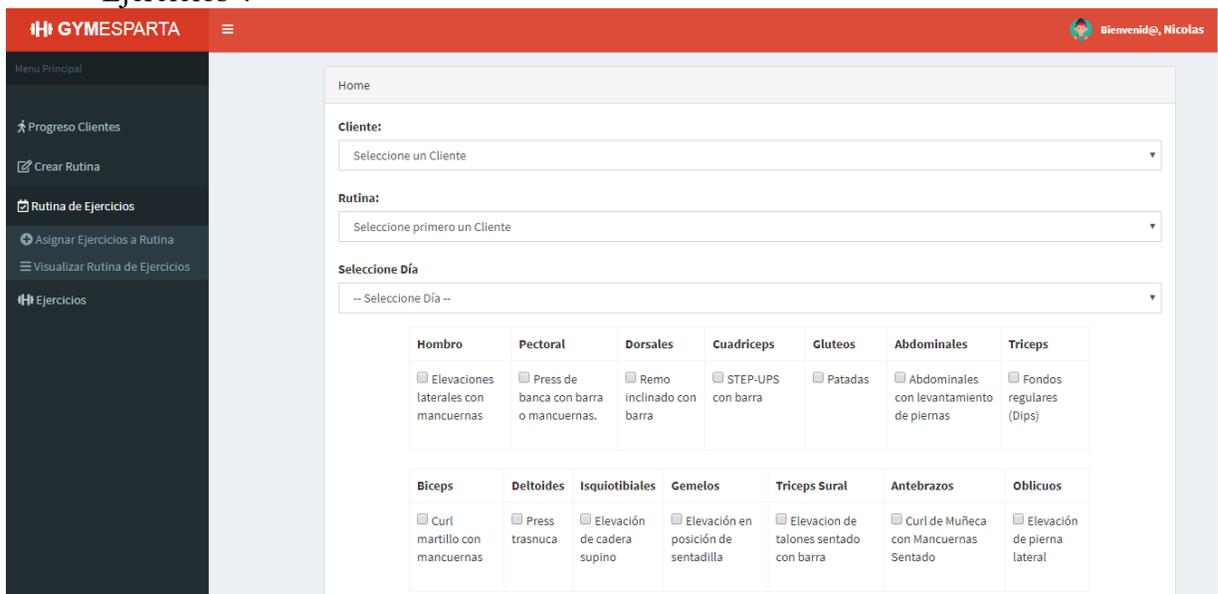
Nombre	Descripcion	Fecha Rutina	Editar	Eliminar
Rutina Nicolas Llanos	Rutina para bajar de peso e aumentar masa muscular	2019-04-23		
Rutina para sacar calugas	rutina potenteee	2019-04-25		

Copyright © 2019 Gimnasio Esparta. Chiguayante. Version 1.1

4) En la siguiente imagen, se visualiza el listado de rutinas de los clientes, junto con sus respectivas opciones de ver los datos, editar, añadir una nueva rutina o eliminar. Se accede a esta a través del menú “Visualizar Rutinas”.



5) En la siguiente imagen, se visualizan los ejercicios junto con sus respectivos datos (video, imagen, descripción, número de series, intervalo de repeticiones). Se debe acceder a través del menú en la opción “Lista de Ejercicios” – “Visualizar Ejercicios”.



6) En la siguiente imagen, se visualiza la pantalla de asignación de ejercicios a rutinas de clientes, el cual se accede desde el menú “Rutina de Ejercicios” – “Asignar Ejercicios a Rutina”. Se debe seleccionar un cliente, luego una rutina y por ultimo asignar los ejercicios de acuerdo al día a ingresar.

Nombre-Apellido Paterno

[Agregar Rutina de Ejercicios](#)

### LISTADO DE RUTINA DE EJERCICIOS

Rut Cliente	Nombre Cliente	Apellidos	Nombre Rutina	Descripción Rutina	Ver Rutina	Eliminar
19087734-5	Nicolas	Llanos Perez	Rutina Nicolas Llanos 9	Rutina para bajar de peso e aumentar masa muscular	<input type="button" value="Ver"/>	<input type="button" value="Eliminar Rutina"/>
18110073-5	Jonathan	Llanos Muñoz	Rutina para sacar calugas 10	rutina potenteee	<input type="button" value="Ver"/>	<input type="button" value="Eliminar Rutina"/>

Copyright © 2019 Gimnasio Esparta. Chiguayante. Version 1.1

7) En la siguiente imagen, se visualiza el listado de ejercicios de rutinas de los clientes, junto con sus respectivas opciones de ver los datos, eliminar rutinas, o ingresar nuevos ejercicios.

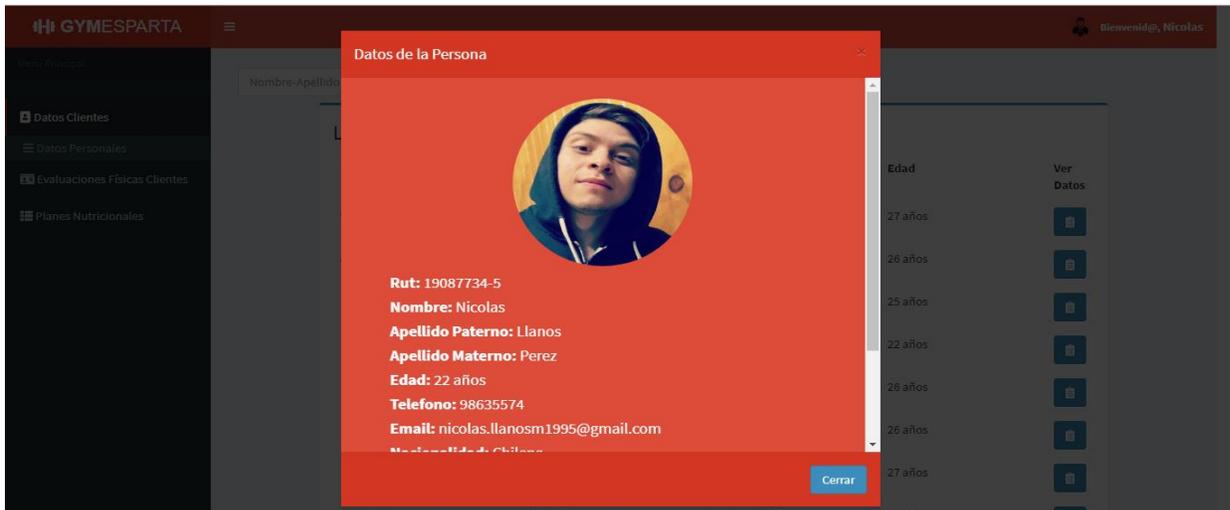
Se accede a esta a través del menú “Visualizar Rutinas de Ejercicios”.

## Funcionalidades Nutricionista

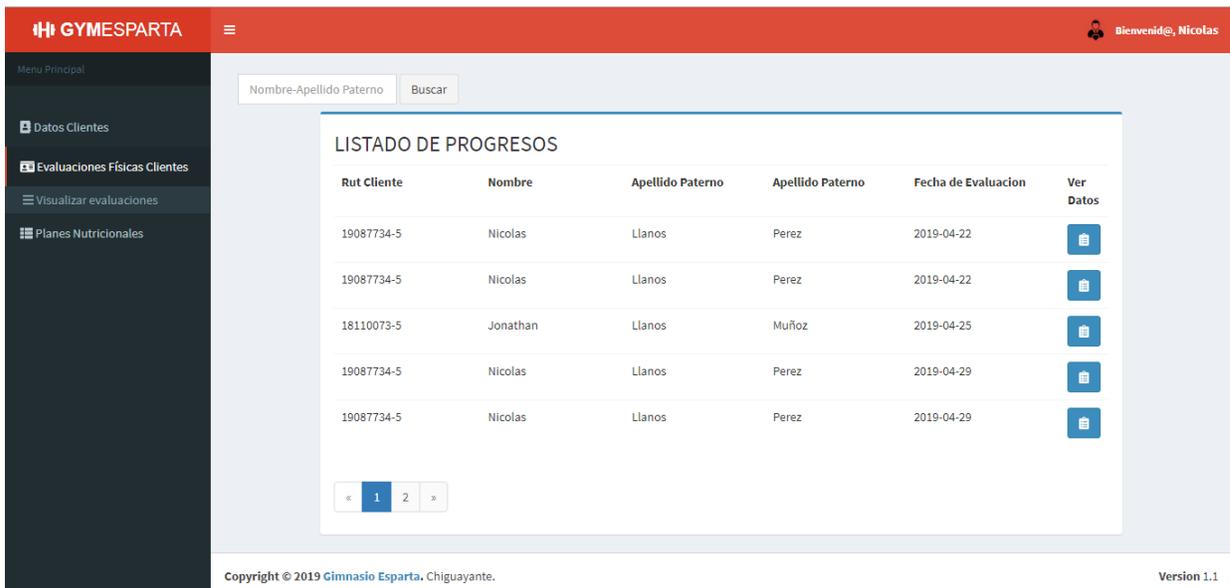
Nombre-Apellido Paterno  Seleccione un Tipo

### LISTADO DE CLIENTES

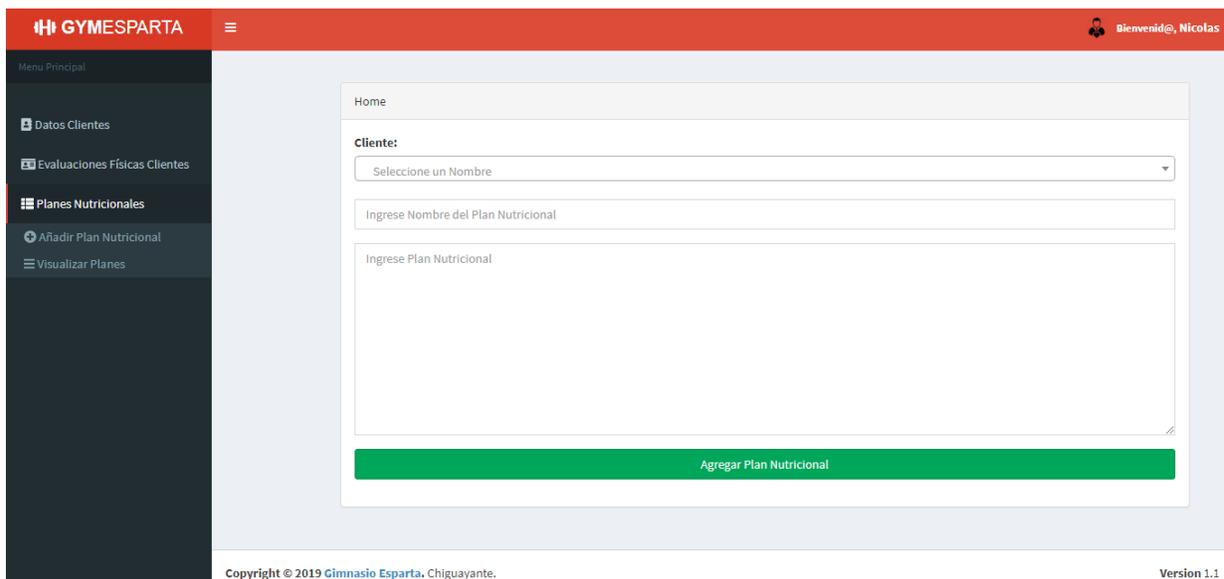
Rut	Nombre	Apellidos	Edad	Ver Datos
9733120-0	Paulina	Muñoz Lavados	27 años	<input type="button" value="Ver"/>
8504123-K	Juan	Perico Palotes	26 años	<input type="button" value="Ver"/>
7890777-K	Javier	Duran Espinoza	25 años	<input type="button" value="Ver"/>
19087734-5	Nicolas	Llanos Perez	22 años	<input type="button" value="Ver"/>
18890765-5	Alfonso	Duran Llanos	26 años	<input type="button" value="Ver"/>
18110073-5	Jonathan	Llanos Muñoz	26 años	<input type="button" value="Ver"/>
18010789-5	Pablo	Suarez Montoya	27 años	<input type="button" value="Ver"/>



1) A continuación, se visualiza el listado de clientes registrados, junto con sus datos personales y deportivos. Se debe ingresar a este a través del menú en la opción “Datos Clientes” – “Datos Personales”.



2) A continuación, se visualiza el listado de progresos deportivos o físicos de los clientes con el fin de proporcionar la información nutricional adecuada a los clientes. Se debe ingresar a este a través del menú en la opción “Evaluaciones Físicas Clientes” – “Visualizar evaluaciones”.

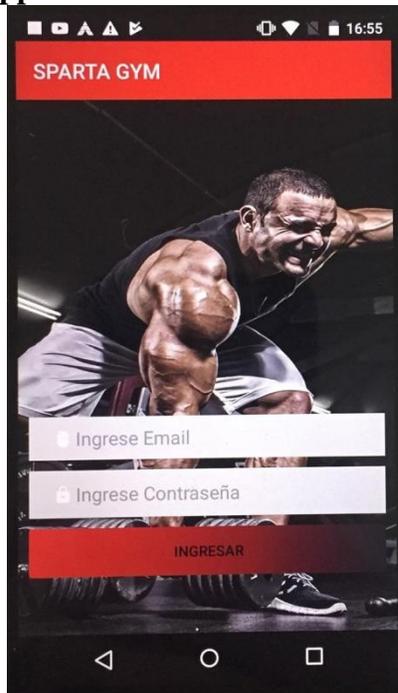


3) En la siguiente imagen, se visualiza la pantalla de asignación de planes nutricionales a clientes, el cual se accede desde el menú “Planes Nutricionales” – “Añadir Plan Nutricional”. Se debe seleccionar un cliente y luego ingresar los datos correspondientes al plan nutricional (nombre plan, descripción).

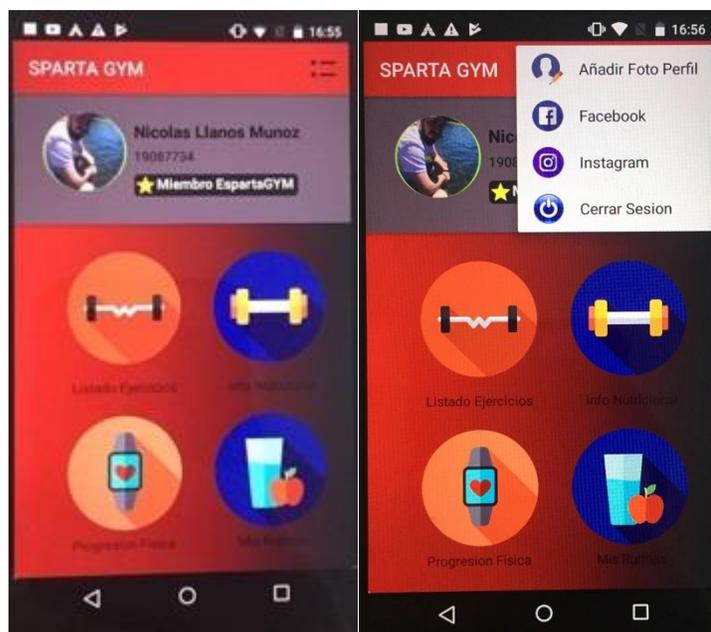


4) En la siguiente imagen, se visualiza el listado de planes nutricionales de los clientes, junto con sus respectivas opciones de ver los datos, eliminar planes, editar planes, o ingresar nuevos planes nutricionales. Se accede a esta a través del menú “Visualizar Planes”.

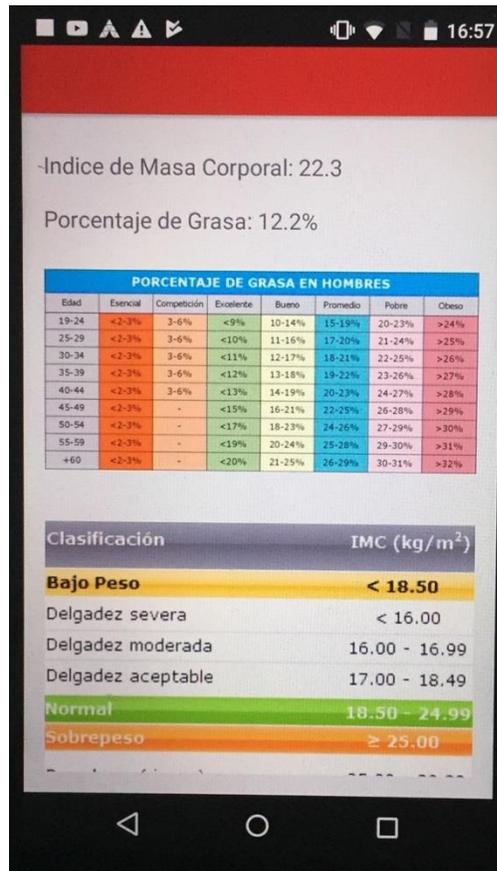
## Funcionalidades Profesores App móvil



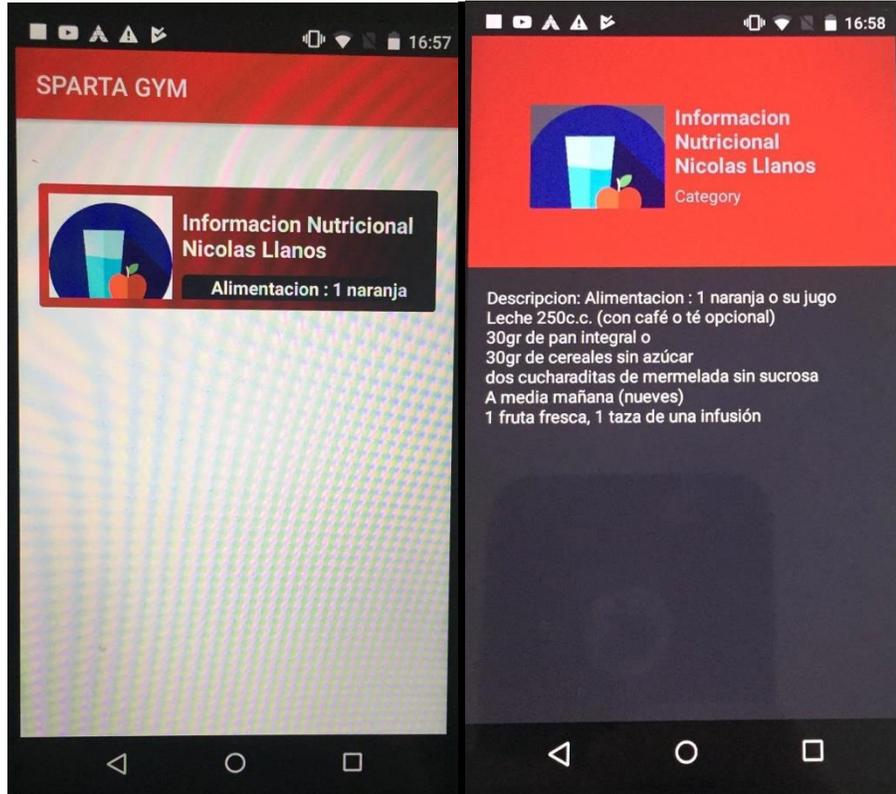
- 1) A continuación, se visualiza la pantalla de inicio de Sesión de la aplicación móvil, el cual se debe ingresar el email y contraseña correspondiente a cada usuario de los clientes.



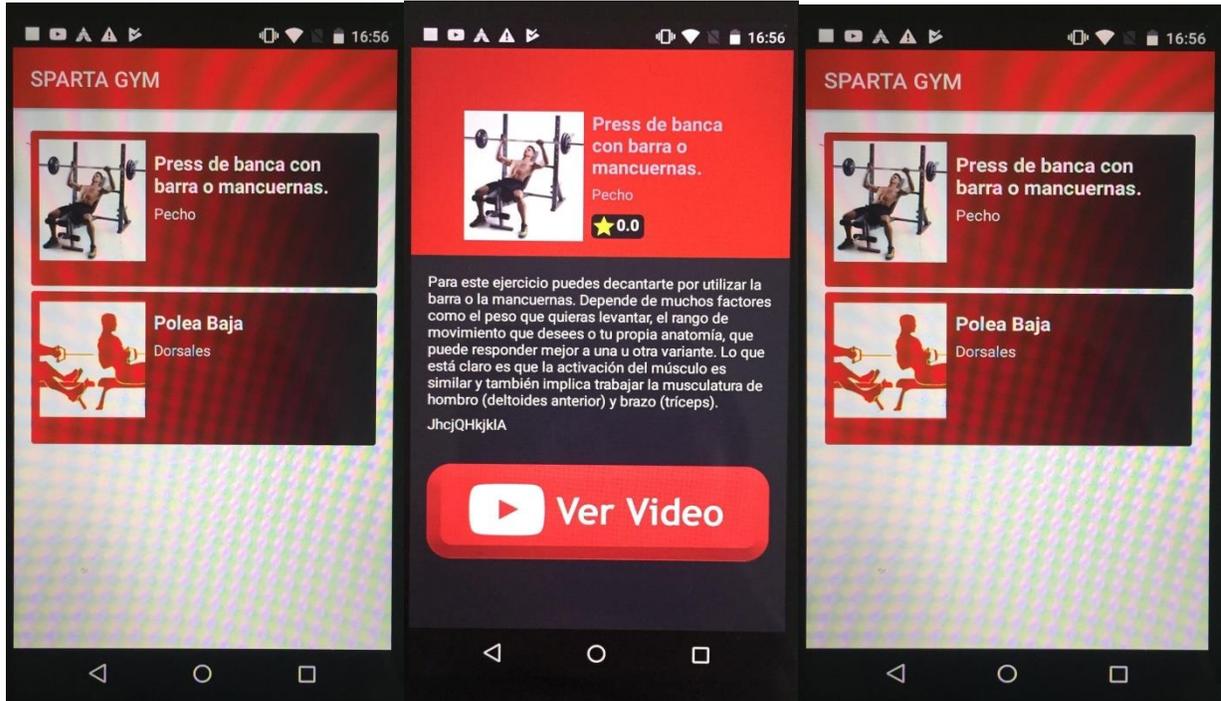
- 2) La siguiente pantalla muestra el menú principal luego del ingreso de sesión, el cual contiene las opciones para visualizar el listado de ejercicios, información nutricional, progresión física, rutinas de ejercicios. Por otra parte cuenta con un menú superior que contiene las opciones de añadir foto de perfil, Facebook institucional, instagram institucional y cerrar la sesión.



- 3) La siguiente pantalla muestra el menú de progresiones físicas de cada cliente con sus respectivas mediciones, lo cual se accede a este a través del menú principal “Progresiones Físicas”.



- 4) La siguiente pantalla muestra el menú de Información nutricional, lo cual se accede a través del menú principal “Info Nutricional”.



- 1) La siguiente pantalla muestra el menú Ejercicios, contiene información del ejercicio, descripción junto con su respectivo video explicativo. Se accede a través del menú principal "Listado Ejercicios".

**ANEXO 3: CODIGO**

## Controlador Clientes

```

<?php namespace App\Http\Controllers;

use App\Http\Requests;
use App\Http\Controllers\Controller;
use App\Http\Requests\ClientesForm;
use Illuminate\Http\Request;
use App\Clientes;
use App\Progreso;
use DB;
use Storage;

class ClientesController extends Controller {

    public function __construct()
    {
        $this->middleware('auth');
    }

    /**
     * Display a listing of the resource.
     *
     * @return Response
     */
    public function index(Request $request)
    {

        //genera listado de cliente en la vista, con la opcion
        de realizar búsquedas por nombre o por estado del cliente.
        $clientes = Clientes::search($request->name)-
        >type($request->get('estado'))->orderBy('rut_cl','DESC')->paginate(20);

        return view ('admin.clientes.inicio')->with('clientes',$clientes);

    }
}

```

```

/**
 * Show the form for creating a new resource.
 *
 * @return Response
 */
public function create()
{
    //ruta para crear clientes
    return view('admin.clientes.crear');
}

/**
 * Store a newly created resource in storage.
 *
 * @return Response
 */
public function store(ClientesForm $request)
{
    //valida que se soliciten obligatoriamente los datos,
    que los datos sean coherentes y no esten duplicados.
    $this->validate($request, [
        'rut_cl' => 'required|unique:clientes',
        'email_cliente' => 'required|unique:clientes|email',
        'nombre_cliente' => 'required',
        'ap_pat_cliente' => 'required',
        'ap_mat_cliente' => 'required',
        'telefono_cliente' => 'required|numeric',
        'nacionalidad_cliente' => 'required',
        'direccion_cliente' => 'required',
        'objetivo' => 'required'
    ]);

    //crea el cliente
    $clientes = new \App\clientes;

    //datos a ingresar
    $contra = \Request::input('rut_cl');
    $nombresito = \Request::input('nombre_cliente');
    $clientes->rut_cl = \Request::input('rut_cl');

```

```

        $clientes->nombre_cliente =
\request::input('nombre_cliente');
        $clientes->ap_pat_cliente =
\request::input('ap_pat_cliente');
        $clientes->ap_mat_cliente =
\request::input('ap_mat_cliente');
        $clientes->telefono_cliente =
\request::input('telefono_cliente');
        $clientes->email_cliente =
\request::input('email_cliente');
        $clientes->huella_cliente = 0;
        //$clientes->contrasena_cliente =
\hash::make(\request::input('contraseña'));
        //$clientes->contrasena_cliente =
\hash::make($nombresito.substr($contra, -4));
        $clientes->nacionalidad_cliente =
\request::input('nacionalidad_cliente');
        $clientes->fecha_nac_cliente =
\request::input('fecha');
        $clientes->sexo_cliente =
\request::input('sexo_cliente');
        //$clientes->alergia_cliente =
\request::input('alergia');
        //$clientes->patologia_cliente =
\request::input('patologia');
        $clientes->fotografia_cliente = 0;
        $clientes->estado = 1;
        $clientes->direccion_cliente =
\request::input('direccion_cliente');
        $clientes->objetivo = \request::input('objetivo');

        //guarda el cliente en la base de Datos
        $clientes->save();

        \session::flash('mensaje','Cliente agregado con
        exito');

        return redirect('clientes');
    }

```

```

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function show($rut_cl)
{
    //

}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function edit($rut_cl)
{
    //ruta para editar un cliente
    return view('admin.clientes.createUpdate')-
>with('clientes', \App\Clientes::find($rut_cl));
}

/**
 * Update the specified resource in storage.
 *
 * @param int $id
 * @return Response
 */
public function update(Request $request, $rut_cl)
{
    //
    //valida que se soliciten obligatoriamente los datos,
que los datos sean coherentes y no esten duplicados.

    $this->validate($request, [
        'email_cliente' => 'required|email',
        'nombre_cliente' => 'required',
        'ap_pat_cliente' => 'required',
        'ap_mat_cliente' => 'required',
    ]);
}

```

```

        'telefono_cliente' => 'required|numeric',
        'nacionalidad_cliente' => 'required',
        'direccion_cliente' => 'required',
        'objetivo' => 'required'

    ]);

        $clientes = \App\Cientes::find($rut_cl);

        $clientes->nombre_cliente =
    \Request::input('nombre_cliente');
        $clientes->ap_pat_cliente =
    \Request::input('ap_pat_cliente');
        $clientes->ap_mat_cliente =
    \Request::input('ap_mat_cliente');
        //$clientes->direccion_cliente =
    \Request::input('direccion');
        $clientes->telefono_cliente =
    \Request::input('telefono_cliente');
        $clientes->email_cliente =
    \Request::input('email_cliente');
        //$clientes->contrasena_cliente =
    \Hash::make(\Request::input('contraseña'));
        $clientes->nacionalidad_cliente =
    \Request::input('nacionalidad_cliente');
        //$clientes->sexo_cliente = \Request::input('sexo');
        //$clientes->alergia_cliente =
    \Request::input('alergia');
        //$clientes->patologia_cliente =
    \Request::input('patologia');
        $clientes->direccion_cliente =
    \Request::input('direccion_cliente');
        $clientes->objetivo = \Request::input('objetivo');

        //guarda las modificaciones en la base de datos
        $clientes->save();

        //redirige a la vista de lista de clientes
        return redirect()->route('clientes.index', ['post' =>
    $rut_cl])->with('mensaje', 'Cliente Modificado con exito');
    }

    /**

```

```

        * Remove the specified resource from storage.
        *
        * @param int $id
        * @return Response
        */
public function destroy($rut_cl)
{
    //
    $clientes= \App\Cientes::find($rut_cl);

    $clientes->delete();

    return redirect()->route('clientes.index')-
>with('message', 'Clientes deleted');
}

public function EditarForm(){
    return view('admin.clientes.inicio');
}

public function EditaraInactivo(Request $request){
    $rut = $request->input('rut_cliente_dejar_activo');

    /*
    DB::table('clientes')->update(['estado'=>0]);
    echo "Estado editado";*/

    DB::table('clientes')
->where('rut_cl', $rut)
->update(['estado' => 0]);

    return redirect()->action('ClientesController@index');
}

```

```

public function EditaraActivo(Request $request){
    $rut = $request->input('rut_cliente');

    /*
    DB::table('clientes')->update(['estado'=>0]);
    echo "Estado editado";*/

    DB::table('clientes')
->where('rut_cl', $rut)
->update(['estado' => 1]);

    return redirect()->action('ClientesController@index');
}

```

```

}

```

## Modelo

```

<?php namespace App;
use Carbon\Carbon;

use Illuminate\Database\Eloquent\Model;

class Clientes extends Model {

    //
    public $timestamps = false;
    protected $table = 'clientes';
    protected $primaryKey = 'rut_cl';
    //protected $
    protected $fillable = ['rut_cl','nombre_cliente',
'ap_pat_cliente',

```

```

'ap_mat_cliente','telefono_cliente','email_cliente','huella_cliente','contrasena_cliente','na
cionalidad_cliente','fecha_nac_cliente','sexo_cliente','fotografia_cliente','estado','direcci
on','objetivo'];

        protected $guarded = ['rut_cl'];

/*
        protected $fillable = ['rut_cl','nombre_cliente',
'ap_pat_cliente',
'ap_mat_cliente','telefono_cliente','email_cliente','huella_cliente','contrasena_cliente','na
cionalidad_cliente','fecha_nac_cliente','sexo_cliente','alergia_cliente','patologia_cliente',
fotografia_cliente','estado'];
*/

        public function getFullNameAttribute()
        {
            return $this->nombre_cliente.".$this->ap_pat_cliente;
        }

        public function scopeSearch($query,$name){

            if(trim($name) != "")
            {
                //return $query->where('nombre_p','LIKE',"%$name%");
                $query->where(\DB::raw("CONCAT(nombre_cliente,'
'ap_pat_cliente)'),"LIKE","%$name%");
            }
        }

        public function progreso() //id
        {
            return $this->hasOne('App\Progreso','rut_cl','rut_cl');
        }

        public function scopeType($query,$estado){
            $estados = config('options4.estados');

            if($estado != "" && isset($estados[$estado])){
                $query->where('estado',$estado);
            }
        }
    
```

```
}
```

```
public function getAgeAttribute()
{
    return Carbon::parse($this->attributes['fecha_nac_cliente']->age;
}
```

```
public function rutinas(){
    return $this->hasMany('App\Rutinas','id_rutina','id_rutina');
}
```

```
}
```

```
Vista para crear clientes
@extends('admin.layout')
```

```
@section('content')
```

```
<!--
```

```
<h5> Fecha: <?php $dt = new DateTime(); echo $dt->format('Y/m/d H:i:s'); ?></h5>-->
```

```
<CENTER><h4><b>INGRESAR NUEVO CLIENTE</b></h4><BR></CENTER>
```

```

                                <div class="container">
    @if($errors->has())
        <div class="alert alert-danger" role="alert">
            @foreach ($errors->all() as $error)
                <div>{{ $error }}</div>
            @endforeach
        </div>

```

```

@endif </br>

<div class="container">
  <div class="row">
    <div class="col-md-10 col-md-offset-1">
      <div class="panel panel-default">
        <div class="panel-heading">Datos Usuario</div>
        <div class="panel-body">

@if(isset($clientes))
  {!! Form::model($clientes, ['route' => ['clientes.update', $clientes->rut_cl], 'method' =>
'patch']) !!}
@else
  {!! Form::open(['route' => 'clientes.store','method' => 'post', 'novalidate']) !!}
@endif

          {!!Form::open(['route' => 'clientes.store'])!!}

          <!--
            <div class="form-group">
              {!!Form::label('rut_cl','Rut: ')!!}
              {!!Form::text('rut_cl',null,
[id=>'rut','required' => 'required','class'=>'form-control', 'placeholder'=>'Ingrese rut sin
digito verificador ej: 12345678'])!!}
            </div-->
            <div class="form-group" >

              {!!Form::label('rut_cl','Rut: ')!!}

              <input type="text" id="rut" name="rut_cl" required oninput="checkRut(this)"
class="form-control">
            </div>

          <div class="form-group" >
            {!!Form::label('nombre_cliente','Nombre:
')!!}
            {!!Form::text('nombre_cliente',null,
[id=>'nombre','required' => 'required','class'=>'form-control', 'placeholder'=>''])!!}
          </div>

```

```

        <div class="form-group">
            {!!Form::label('ap_pat_cliente','Apellido
Paterno: ')!!}
            {!!Form::text('ap_pat_cliente',null,
['id'=>'ap_pat','required' => 'required','class'=>'form-control', 'placeholder'=>"])!!}
        </div>

        <div class="form-group">
            {!!Form::label('ap_mat_cliente','Apellido
Materno: ')!!}
            {!!Form::text('ap_mat_cliente',null,
['id'=>'ap_mat','required' => 'required','class'=>'form-control', 'placeholder'=>"])!!}
        </div>

        <div class="form-group">
            {!!Form::label('telefono_cliente','Telefono:
')!!}
            {!!Form::text('telefono_cliente',null,
['id'=>'telefono','required' => 'required','class'=>'form-control', 'placeholder'=>"])!!}
        </div>

        <div class="form-group">
            {!!Form::label('email_cliente','E-mail: ')!!}
            {!!Form::email('email_cliente',null,
['id'=>'email','required'=>'required','class'=>'form-control', 'placeholder'=>"])!!}
        </div>

        <!--
        <div class="form-group">
            {!!Form::label('contraseña','Contraseña: ')!!}
            {!!Form::text('contraseña',null,
['id'=>'pass','class'=>'form-control', 'placeholder'=>"])!!}
        </div-->

        <div class="form-group">
            {!!Form::label('nacionalidad_cliente','Nacionalidad:
')!!}
    
```

```

                                {!!Form::text('nacionalidad_cliente',null,
[id]=>'nacionalidad','required' => 'required','class'=>'form-control', 'placeholder'=>')]!!}

                                </div>

<div class="form-group">
{!!Form::label('fecha','Fecha de Nacimiento: ')}
{!!      Form::text('fecha',      null,      ['class'=>'form-control      datepicker',
'id'=>'fecha']),old('fecha')}
</div>
<script type="text/javascript" src="/dist/jquery-ui/external/jquery/jquery.js"></script>
<script type="text/javascript" src="/dist/jquery-ui/jquery-ui.min.js"></script>
<script type="text/javascript">
var fecha = $("#fecha").val();

    $('<div class="form-group">
    datepicker({
    dateFormat: "yy-mm-dd",
    language: "es",
    autoclose: true,
    changeMonth: true,
    changeYear: true,
    yearRange: '-100:+0'
    }).datepicker('setDate', new Date(fecha));

</script>

<div class="form-group">
<label for="Sexo">Selecciona genero:</label>
<br>
{!! Form::radio('sexo_cliente', 'Masculino', true) !!}
Masculino

{!! Form::radio('sexo_cliente', 'Femenino', true) !!}
Femenino
</div>

<!--
                                <div class="form-group">
                                {!!Form::label('alergia','Alergia: ')}
                                {!!Form::text('alergia',null,
[id]=>'alergia','class'=>'form-control', 'placeholder'=>')]!!}
                                </div>

```

```

                <div class="form-group">
                    {!!Form::label('patologia','Patologia: ')}
                    {!!Form::text('patologia',null,
[id'=>'patologia','class'=>'form-control', 'placeholder'=>'])!!}
                </div> -->
<!--
                <div class="form-group">
                    <label class="col-md-4 control-label">Nuevo Archivo</label>
                    <input type="file" class="form-control" name="file" >
                </div-->

</div></div></div></div></div>

<div class="container">
    <div class="row">
        <div class="col-md-10 col-md-offset-1">
            <div class="panel panel-default">
                <div class="panel-heading">Direccion Usuario</div>
                <div class="panel-body">

<div class="form-group">
                    {!!Form::label('direccion_cliente','Direccion:
')!!}
                    {!!Form::text('direccion_cliente',null,
[id'=>'direccion_cliente','required' => 'required','class'=>'form-control',
'placeholder'=>'])!!}
                </div>

                <div class="form-group">
                    {!!Form::label('objetivo','Objetivo Asistencia al Gimnasio: ')}
                    {!!Form::text('objetivo',null, [id'=>'objetivo','required' =>
'required','class'=>'form-control', 'placeholder'=>'])!!}
                </div>

```

```
<div class="form-group" onclick="return confirm('Esta seguro?')">
  {!! Form::submit('Agregar Cliente', ["class" => "btn btn-success btn-block"]) !!}
</div>
```

```
{!! Form::close() !!}
</div></div></div></div></div></div>
```

```
@endsection
```

Vista para la modificación de datos del cliente

```
@extends('admin.layout')
```

```
@section('content')
```

```
@if(count($errors)>0)
<div class="alert alert-danger" role="alert">
  <ul>
    @foreach($errors->all()as $messages)
      <li>{{ $messages }}</li>
    @endforeach
  </ul>
</div>
@endif
```

```
<div class="container">
  <div class="row">
    <div class="col-md-10 col-md-offset-1">
      <div class="panel panel-default">
        <div class="panel-heading">Home</div>
```

```
@if(isset($clientes))
  {!! Form::model($clientes, ['route' => ['clientes.update', $clientes->rut_cl], 'method' =>
'patch']) !!}
@else
  {!! Form::open(['route' => 'clientes.store','method' => 'post', 'novalidate']) !!}
@endif
```

```

<div class="panel-body">
  {!! Form::open(['route' => 'clientes.store']) !!}

  <div class="form-group">
    {!!Form::label('rut_cl','Rut Cliente: ')!!}
    {!!Form::text('rut_cl',null, ['class'=>'form-control','disabled'=>'disabled'])!!}
  </div>

  <div class="form-group">
    {!!Form::label('nombre_cliente','Nombre: ')!!}
    {!! Form::text('nombre_cliente', null,
    ['class'=>'form-control', 'placeholder'=>''])
    !!}
  </div>

  <div class="form-group">
    {!!Form::label('ap_pat_cliente','Apellido Paterno: ')!!}
    {!! Form::text('ap_pat_cliente', null,
    ['class'=>'form-control', 'placeholder'=>''])
    !!}
  </div>

  <div class="form-group">
    {!!Form::label('ap_mat_cliente','Apellido Materno: ')!!}
    {!! Form::text('ap_mat_cliente', null,
    ['class'=>'form-control', 'placeholder'=>''])
    !!}
  </div>

  <div class="form-group">
    {!!Form::label('telefono_cliente','Telefono: ')!!}
    {!! Form::text('telefono_cliente', null,
    ['class'=>'form-control', 'placeholder'=>''])
    !!}
  </div>

  <div class="form-group">
    {!!Form::label('email_cliente','Email: ')!!}

```

```
{!! Form::text('email_cliente', null,
[class'=>'form-control', 'placeholder'=>"])
!!}
</div>
```

```
{!!Form::label('nacionalidad_cliente','Nacionalidad: ')!!}
<div class="form-group">
{!! Form::text('nacionalidad_cliente', null,
[class'=>'form-control', 'placeholder'=>"])
!!}
</div>
```

```
{!!Form::label('objetivo','Objetivo asistencia al Gimnasio: ')!!}
<div class="form-group">
{!! Form::text('objetivo', null,
[class'=>'form-control', 'placeholder'=>"])
!!}
</div>
```

```
{!!Form::label('direccion_cliente','Direccion: ')!!}
<div class="form-group">
{!! Form::text('direccion_cliente', null,
[class'=>'form-control', 'placeholder'=>"])
!!}
</div>
```

```
<script type="text/javascript" src="/dist/jquery-ui/external/jquery/jquery.js"></script>
<script type="text/javascript" src="/dist/jquery-ui/jquery-ui.min.js"></script>
```

```
<!--
<div class="form-group">
<label for="Foto">Subir Fotografía:</label>
```

```
<input type="file" class="form-control" name="file" >
</div>
```

```
<div class="form-group">
  {!! Form::text('fotografia_cliente', null,
  ['class'=>'form-control', 'placeholder'=>'Ingrese Fotografia'])
  !!}
</div-->
```

```
<div class="form-group" onclick="return confirm('Esta seguro?')">
  {!! Form::submit('Guardar', ["class" => "btn btn-success btn-block"]) !!}
</div>
```

```
{!! Form::close() !!}
```

```
</div>
```

```
@endsection
```

Vista Boton Modal para visualizar datos de clientes

```
<style>.centrado2{
  margin:10px auto;
  display:block;
  width:200px;
  height:200px;
  border-radius:150px;
}</style>
```

```
<div class="modal fade" tabindex="-1" id="modal-{{ $Cliente->rut_cl }}" role="dialog"
>
```

```
<div class="modal modal-danger fade in" id="modal-danger" style="display: block;
padding-right: 17px;">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
```

```

        <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span
        aria-hidden="true">&times;</span></button>
        <h4 class="modal-title">Datos de la Persona</h4>
</div>

```

```

<div class="modal-body" style="max-height: calc(100vh - 210px);overflow-y:
auto"; align="center">

```

```

        <p> </p>

```

```

        <ul><h4>
        <H4 align="left">
                <p> <b>Rut:</b> {{ $Cliente->rut_cl }}</p>
                <p> <b>Nombre:</b> {{ $Cliente-
>nombre_cliente }}</p>

```

```

        <p> <b>Apellido Paterno:</b> {{ $Cliente->ap_pat_cliente }}</p>
        <p> <b>Apellido Materno:</b> {{ $Cliente->ap_mat_cliente }}</p>
        <p> <b>Edad:</b> {{ $Cliente->age }} años</p>
        <p> <b>Telefono:</b> {{ $Cliente->telefono_cliente }}</p>
        <p> <b>Email:</b> {{ $Cliente->email_cliente }}</p>
        <p> <b>Nacionalidad:</b> {{ $Cliente->nacionalidad_cliente }}</p>
        <p> <b>Fecha de Nacimiento:</b> {{ $Cliente->fecha_nac_cliente }}</p>
        <p> <b>Genero:</b> {{ $Cliente->sexo_cliente }}</p>
        <p> <b>Direccion:</b> {{ $Cliente->direccion_cliente }}</p>
        <p> <b>Objetivo Asistencia al Gimnasio:</b> {{ $Cliente-
>objetivo }}</p>

```

```

        <?php $name = DB::select("SELECT inscripciones.id_plan,
inscripciones.rut_cl, planes.nombre_plan
FROM inscripciones
LEFT JOIN planes ON inscripciones.id_plan = planes.id_plan
LEFT JOIN clientes ON inscripciones.rut_cl = clientes.rut_cl
WHERE clientes.rut_cl = '$Cliente->rut_cl' AND inscripciones.estado = '1'") ?>

```

```

        <p><b>Plan Contratado:</b> @foreach($name as $nam)
<?php echo $nam->nombre_plan?>
@endforeach </a>

```

```

        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-primary" data-
dismiss="modal">Cerrar</button>
        </div>
    </div>
</div>

```

```
</div>
```

```
Vista Listado de Clientes
@extends('admin.layout')
```

```
@section('content')
```

```
@if(Session::has('mensaje'))
    <div class="alert alert-success"><em> {!! session('mensaje') !!}</em></div>
@endif
```

```
<!-- Content Header (Page header) -->
```

```
<!--
```

```

    <section class="content-header">
        <ol class="breadcrumb">
            <li><a href="#"><i class="fa fa-dashboard"></i> Inicio</a></li>
            <li class="active">Clientes</li>
        </ol>
    </section>-->

```

```
<style type="text/css">
```

```

#boton1 {
    padding:5px 15px;
    background:#B50E0E;
    border:0 none;
    color:white;
    cursor:pointer;
    -webkit-border-radius: 5px;
    border-radius: 5px;
}
</style>

```

```
<style type="text/css">
```

```
#boton2 {
  padding:5px 15px;
  background:#CDCF2A;
  border:0 none;
  color:white;
  cursor:pointer;
  -webkit-border-radius: 5px;
  border-radius: 5px;
}
</style>
```

```
<style type="text/css">
```

```
#botonbuscar {

}
</style>
```

```
<!--
```

```
<div class="container">
<div class="row">
<div class="col-md-3">-->
```

```
<!--
```

```
{!! Form::open ([!route' => 'clientes.index', 'method' => 'GET', 'class' => 'navbar-form
navbar-left', 'role' => 'search'])!!}-->
```

```
{!!Form::model(Request::all(), [!route' => 'clientes.index', 'method' => 'GET', 'class'
=> 'navbar-form navbar-left', 'role' => 'search'])!!}
```

```
<div class="form-group">
  {!!Form::text('name',null,[!class'=>'form-control','placeholder'=>'Nombre-Apellido
Paterno'])!!}
  {!!Form::select('estado',config('options4.estados'),null,[!class' => 'form-control'])!!}
</div>
```

```

<button type="submit" class="btn btn-default" id="botonbuscar">Buscar</button>
{!! Form::close()!!}

<!--
<div style="padding: 25px 30px; background: rgb(255, 255, 255); z-index: 999999;
font-size: 16px; font-weight: 600;"><a class="float-right" href="#" data-
toggle="tooltip" data-placement="left" title="" style="color: rgb(255, 255, 255); font-
size: 30px;"></a><a style="color: rgba(200, 255, 255, 0.9);"></a></div>-->
<BR><BR>
<div class="container">
<div class="row">
<div class="col-md-10 col-md-offset-1" align="right">

{!! Html::link(route('clientes.create'), 'Agregar Cliente', array('class' => 'btn btn-info
btn-md')) !!}
<button id="btnPrint" class="btn btn-warning btn-md">Exportar <i class="fas fa-
print"></i></button>

</div>

<div id="dvCont" style="display:none;">
<CENTER><h4><b>LISTADO DE CLIENTES</b></h4></CENTER>
@if(!$clientes->isEmpty())
<table class="table table-bordered">
<tr>
<th>Rut</th>
<th>Nombre</th>
<th>Apellido Paterno</th>
<th>Apellido Materno</th>
<th>Edad</th>
<th>Estado</th>
</tr>
@foreach ($clientes as $Cliente)
<tr>
<td width="500">{{ $Cliente->rut_cl }}</td>
<td width="500">{{ $Cliente->nombre_cliente }}</td>
<td width="500">{{ $Cliente->ap_pat_cliente }}</td>
<td width="500">{{ $Cliente->ap_mat_cliente }}</td>
<td width="500">{{ $Cliente->age }} años</td>

@if($Cliente->estado == 1) <td width="500">{{ "Activo" }}</td>
@elseif($Cliente->estado == 0) <td width="500">{{ "Inactivo" }}</td>

```

```

        @endif
    @endforeach
    </tr>
</table>
@endif

</div>

<div class="col-md-10 col-md-offset-1">
<div class="box box-primary col-md-10 ">
<h3 class="box-title">LISTADO DE CLIENTES</h3>
    @if(!$clientes->isEmpty())
        <table class="table table-hover">
            <tr>
                <th>Rut</th>
                <th>Nombre</th>
                <th>Apellidos</th>
                <th>Edad</th>
                <th>Estado</th>
                <th>Ver Datos</th>
                <th>Editar</th>
            </tr>
            @foreach ($clientes as $Cliente)
                <tr>
                    <td width="500">{{ $Cliente->rut_cl }}</td>
                    <td width="500">{{ $Cliente->nombre_cliente }}</td>
                    <td width="500">{{ $Cliente->ap_pat_cliente }} {{ $Cliente-
                    >ap_mat_cliente }}</td>
                    <td width="500">{{ $Cliente->age }} años</td>
                    <!--<td width="500">{{ $Cliente->estado }}</td-->

                    @if($Cliente->estado == 1) <td width="500">{{ "Activo" }}</td>
                    @elseif($Cliente->estado == 0) <td width="500">{{ "Inactivo" }}</td>
                    @endif
                    <!--<td width="500"></td-->

                </tr>
            @endforeach
        </table>

        <!--AQUI LLAMADA AL MODAL-->

        <td><a href="#" data-target="#modal-{{ $Cliente->rut_cl }}" data-
        toggle="modal" title="Mostrar">
            <button type="button" class="btn btn-primary btn-md" data-toggle="modal"><i
            class="fas fa-clipboard-list"></i></button>
    
```

```

</a></td>

<!--
<td width="60" align="center">
    {!! Html::link(route('clientes.edit', $Cliente->rut_cl), null, array('class' =>
'fas fa-user-edit')) !!}
</td-->

<td width="60" align="center"><a href="{!! route('clientes.edit', $Cliente-
>rut_cl) !!}">
    <button type="button" class="btn btn-primary btn-md"><i class="fas fa-
user-edit"></i></button></td>

@if($Cliente->estado == 1)
<td>
<form id="form1" name="form1" action="/inicio2" method="post">
    <input type="hidden" name="_token" value = "<?php echo csrf_token();
?>">

    <input type="hidden" name="rut_cliente_dejar_activo"
id="rut_cliente_dejar_activo" value="<?php echo $Cliente->rut_cl?>">

    <input type="submit" value="Dejar Inactivo" id="boton1"
onclick="return confirm('Esta seguro?')"/>

</form></td>

@elseif($Cliente->estado == 0)
<td>
<form id="form2" name="form2" action="/inicio" method="post">
    <input type="hidden" name="_token" value = "<?php echo csrf_token();
?>">

    <input type="hidden" name="rut_cliente" id="rut_cliente" value="<?php
echo $Cliente->rut_cl?>">

    <input type="submit" value="Dejar Activo" id="boton2" onclick="return
confirm('Esta seguro?')"/>

</form></td>

```

@endif

</tr>

@include('admin.clientes.botonmodal')

@endforeach

</table>

</div></div>

@endif

<!--{!! \$clientes->render() !!}-->

{!! \$clientes->appends(Request::only(['name','estado']))->render()!!}

</div>

</div>

</div>

@endsection