

**UNIVERSIDAD RICARDO PALMA**  
**FACULTAD DE INGENIERÍA**  
**PROGRAMA DE TITULACIÓN POR TESIS**  
**ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA**



**MYTRAININGGOAL: FACILITAR LA ADOPCIÓN DE UN ESTILO  
DE VIDA SANO EN LOS CLIENTES DEL GIMNASIO PERSONAL  
TRAINING, MEDIANTE LA UTILIZACIÓN DE UN SISTEMA MÓVIL  
CON ANDROID, APLICANDO EL CONCEPTO DE CLEAN  
ARCHITECTURE**

**TESIS**  
**PARA OPTAR EL TÍTULO PROFESIONAL DE**  
**INGENIERO INFORMÁTICO**

**PRESENTADA POR**

**Bach. DELLEPIANE ESPINOZA SERGIO GIANFRANCO**

**Bach. DÍAZ ALVARADO HENRY ALINSON**

**ASESOR: Lic. WARTHON VARELA RENZO**  
**ASESOR: Mg. LINÁREZ COLOMA HUMBERTO**

**LIMA - PERÚ**

**2016**

## **DEDICATORIA**

A mis padres Michael y Gladis  
A mis hermanas Pamela y Gianella  
A mi novia Karen  
A mi compañero de tesis Sergio.

A mis padres Sergio y Silvia  
A Juan, el cual me quiere como un hijo  
y yo como a un padre  
A mi hermano Jorge  
A mi compañero de tesis Henry.

## **AGRADECIMIENTOS**

En primer lugar agradecer a Dios por darnos fuerza, salud e inteligencia para poder llegar hasta este momento tan importante en nuestras vidas.

En segundo lugar agradecer a nuestros padres por su constante apoyo y preocupación por que alcancemos metas, sueños y objetivos desde que empezamos el colegio, pasando por la universidad y por todo lo que nos tiene preparado el futuro.

Agradecer a nuestros asesores, el Lic. Renzo Warthon y el Mg. Humberto Linarez por la confianza que depositaron en nosotros, por el apoyo y asesoría para llevar todo hacia delante de la mejor manera y por todos los conocimientos que nos transmitieron.

Agradecer a mi compañero de tesis, Sergio, por su constante apoyo y por sus enseñanzas. De igual manera agradecer a mi compañero y amigo Henry, por todos los momentos buenos y malos en el desarrollo de esta Tesis, por su comprensión y apoyo en los momentos más difíciles.

# ÍNDICE

RESUMEN .....	viii
ABSTRACT .....	ix
INTRODUCCIÓN.....	1
CAPITULO I: VISIÓN DEL PROYECTO .....	2
1.1 Antecedentes del Problema .....	2
1.2 Identificación del Problema .....	3
1.2.1 Problema Central .....	3
1.2.2 Problemas Específicos.....	5
1.3 Objetivos .....	6
1.3.1 Objetivo General .....	6
1.3.2 Objetivos Específicos.....	6
1.4 Descripción y Sustentación de la Solución .....	7
1.4.1 Descripción de la Solución.....	7
1.4.2 Justificación de la Realización del Proyecto .....	8
CAPITULO II: MARCO CONCEPTUAL.....	9
2.1. Marco Conceptual .....	9
2.2 Estado del Arte.....	19
2.2.1 Trabajos Realizados .....	19
2.2.2 Herramientas para la Implementación .....	21
2.2.3 Definición de Términos .....	22
CAPITULO III: DESARROLLO DEL PROYECTO .....	24
3.1 Alcance del Proyecto:.....	24
3.1.1 Descripción del alcance del producto. ....	24
3.1.2 Criterios de aceptación del producto.....	25
3.1.3 Estructura de Desglose del Trabajo y Entregables. ....	26
3.1.4 Las exclusiones del proyecto.....	27

3.1.5 Restricciones del proyecto.....	27
3.1.6 Supuestos del proyecto.....	27
CAPITULO IV: DESARROLLO DEL PRODUCTO .....	28
4.1 Modelado de Negocio.....	28
4.1.1 Modelo AS IS .....	28
4.1.2 Modelo TO BE. ....	29
4.2 Requerimientos del Producto/Software. ....	31
4.2.1 Requerimientos Funcionales.....	31
4.2.2 Requerimientos No Funcionales. ....	33
4.3 Arquitectura del Producto/Software. ....	33
4.3.1 Vista de Casos de Uso.....	33
4.3.2 Vista de Procesos.....	41
4.3.3 Vista Lógica. ....	41
4.3.4 Vista de Componentes.....	47
4.3.5 Vista de Despliegue. ....	50
4.4 Diseño de la solución. ....	52
4.4.1 Realización de Casos de Uso.....	52
4.4.2 Prototipos.....	72
CAPITULO V: RESULTADOS.....	80
CONCLUSIONES Y RECOMENDACIONES .....	81
REFERENCIAS BIBLIOGRÁFICAS .....	82
ANEXOS .....	84

## ÍNDICE DE FIGURAS

FIGURA 1-1 MACROPROCESOS DEL NEGOCIO .....	5
FIGURA 2-1 LOGO DE MYBATIS.....	11
FIGURA 2-2 LOGO DE ANDROID .....	12
FIGURA 2-3 HISTORIAS DE USUARIO .....	13
FIGURA 2-4 LOGO DE MYFITNESSPAL .....	20
FIGURA 2-5 APP DIET ASSISTANT.....	20
FIGURA 2-6 LOGO DIETAS PARA ADELGAZAR .....	21
FIGURA 4-1 MODELO AS IS .....	28
FIGURA 4-2 MODELO TO BE .....	30
FIGURA 4-3 DIAGRAMA DE PAQUETES DEL SISTEMA.....	35
FIGURA 4-4 DIAGRAMA DE ACTORES DEL SISTEMA.....	36
FIGURA 4-5 DIAGRAMA GENERAL DE CASOS DE USO DEL SISTEMA.....	40
FIGURA 4-6 VISTA LÓGICA DEL SISTEMA .....	45
FIGURA 4-7 DIAGRAMA DE ENTIDAD - RELACIÓN.....	46
FIGURA 4-8 DIAGRAMA DE COMPONENTES.....	50
FIGURA 4-9 DIAGRAMA DE DESPLIEGUE.....	51
FIGURA 4-10 PANTALLA DE BIENVENIDA.....	72
FIGURA 4-11 PANTALLA DE LOGIN .....	73
FIGURA 4-12 PANTALLA DE MENU DE OPCIONES .....	74
FIGURA 4-13 PANTALLA DE CONSULTA PROGRAMAS DEL GIMNASIO ...	75
FIGURA 4-14 PANTALLA DEL DETALLE DE UN PROGRAMA.....	76
FIGURA 4-15 PANTALLA DE MIS DIETAS .....	77
FIGURA 4-16 PANTALLA DE MI PROGRESO.....	78
FIGURA 4-17 PANTALLA DE CONSULTA CALENDARIO.....	79

## ÍNDICE DE TABLAS

TABLA 2-1 PRINCIPIOS SOLID DE CLEAN ARCHITECTURE.....	9
TABLA 2-2 CAPAS DE CLEAN ARCHITECTURE .....	10
TABLA 2-3 GUÍA DE ALIMENTOS PARA COMER SANAMENTE .....	16
TABLA 2-4 RECOMENDACIONES CALÓRICAS SEGÚN EDAD, SEXO Y PESO .....	17
TABLA 2-5 RECOMENDACIONES DE ALIMENTOS POR LA PIRÁMIDE ALIMENTICIA.....	18
TABLA 3-1 HITOS DEL PROYECTO .....	26
TABLA 4-1 DESCRIPCIÓN DE ACTIVIDADES DEL MODELO TO BE .....	31

## **RESUMEN**

Este proyecto de tesis consiste en el diseño e implementación de un sistema móvil basado en la arquitectura limpia para el uso de los clientes del gimnasio Personal Training, con el fin de que lleven un estilo de vida saludable.

Por otro lado, el gimnasio cuenta con una barra de comida, en la cual los clientes pueden encontrar alimentos sanos, óptimos para su alimentación.

Debido a la falta de información y falta de comunicación entre el personal del gimnasio, el cliente no está al tanto de los alimentos de la barra.

Actualmente la mayoría de los sistemas se desarrollan sobre una arquitectura básica, poco flexibles a los cambios, debido a ello se propone que nuestro sistema se base en una arquitectura limpia, muy flexible.

Teniendo en cuenta los problemas identificados, se propone un único sistema móvil en android, utilizando la arquitectura limpia.

Por último, el sistema permitirá al cliente tener información sobre sus pesos, medidas, resultados de evaluaciones en el momento que desee. Solo tiene que tomar su teléfono móvil y abrir nuestra aplicación.

Palabras Claves: Arquitectura Limpia, Aplicación Móvil, Vida Sana.



## **ABSTRACT**

This thesis project consists of the design and implementation of a mobile system based on clean android architecture for use by customers Personal Training gym, so they have a healthy lifestyle.

On the other hand, the gym has a food bar, where customers can find healthy diets and optimal for their food.

Due to lack of information and lack of communication between staff of the gym, the customer is not aware of this bar food.

Currently most systems are developed based on an architecture for walks flexible because of this we propose to base our system on a clean, very flexible architecture.

Given the problems identified in this thesis the implementation of a single mobile android system is proposed, using the clean architecture.

Finally, the system will allow the customer to have information about their weights, measurements, test results at the time you want. Just enter your mobile phone and open the application.

Keywords: Architecture Clean, Mobile Application, Fitness.

## INTRODUCCIÓN

El gimnasio Personal Training, que brinda diversos programas de ejercicios para cada perfil, tiene un único local, situado en el distrito de San Borja, además cuenta con nutricionistas y trainers dispuestos a ayudar a sus clientes. En varias ocasiones por no contar con la información de sus exámenes médicos, medidas y pesos, el cliente pierde el conocimiento de estos resultados.

Se ha dividido el proyecto en 5 etapas. El primer capítulo se centra en la descripción del problema y la justificación de como poder solucionar mediante el desarrollo del sistema. El segundo capítulo está centrado en la exposición de las tecnologías y software a usarse en todo el proyecto. El tercer capítulo detalla el alcance del proyecto, así como también las exclusiones y restricciones. El cuarto capítulo muestra el diseño de la arquitectura del sistema, en un primer lugar se muestra el modelado del sistema y de la base de datos. Así mismo se muestran las interfaces y módulos que se desarrollaron con la finalidad de lograr un único sistema móvil basado en la arquitectura limpia.

Por último, se presentan las conclusiones y recomendaciones del presente trabajo, además de proponer algunos trabajos futuros que permitan optimizar el diseño de la arquitectura del sistema para una posible implementación en cualquier otro gimnasio.

# CAPITULO I: VISIÓN DEL PROYECTO

## 1.1. ANTECEDENTES DEL PROBLEMA

Los sistemas de información en la actualidad más comunes son vía web y vía móvil. Dada la coyuntura actual, más personas están usando sus smartphones y/o smartwatches para realizar sus actividades rutinarias.

Muchas personas vienen preocupándose por su salud, por las desorbitantes estadísticas que presenta la OMS, como señala Ayala (2013):

“Datos divulgados por la Organización Mundial de la Salud (OMS) en Bolivia indican que la prevalencia de estos males ha aumentado como consecuencia de los malos hábitos alimenticios. Por ejemplo, Santa Cruz presenta los índices más altos de diabetes con 300.000 personas (un 15%+-)”. (p. 25)

Bolivia solo es un ejemplo de lo que puede ocurrir si no se tiene un peso controlado y una buena alimentación, en el Perú según el diario El Comercio (2014, Sección Salud): “Tres de cada cinco peruanos tienen sobrepeso u obesidad”... “La organización mundial de la salud (OMS) recomienda hacer 30 minutos de ejercicio al día en ingerir, como máximo 25 gramos de azúcar (unas seis cucharaditas de café) para evitar desarrollar enfermedades”.

Es por este motivo que las personas, para evitar problemas de salud deciden realizar alguna actividad física cardiovascular (correr, nadar, bailar, etc.) de manera personal o concurriendo a un centro de ejercicios físicos conocido como gimnasio. Por otro lado, las personas al inscribirse a un gimnasio tienen la idea errónea que al realizar ejercicio pueden alimentarse de una manera desordenada y no seguir la dieta que se le recomienda de manera estricta.

Estos malos hábitos incurren en que, aunque los clientes del gimnasio realizan los ejercicios físicos que se le indican, no tienen la fuerza de voluntad para acatar fielmente la dieta entregada por el nutricionista, o si es que la tienen o intentan tenerla, no tienen opciones de alimentación para poder seguir la dieta de manera correcta y estricta.

Según Flor, Granda y Revelo (2005):

“Para lograr el máximo rendimiento físico, es conveniente tener estilos de vida adecuados (no fumar, no consumir alcohol y un descanso adecuado nocturno), tener una dieta variada y equilibrada, que se ajuste en cantidad y calidad a las necesidades de cada deportista. Alimentarse correctamente y tener un entrenamiento adecuado garantiza un crecimiento correcto del organismo, contribuye a la conservación de la salud”. (p. 119)

El presente proyecto está dirigido al desarrollo de un sistema de información utilizando el medio móvil con el fin de llevar un control de la vida sana de las personas que se inscriben a un gimnasio tanto en temas clínicos (exámenes), nutricionales y de control de peso, además de facilitarles el proceso de obtención de la meta que dichas personas se han propuesto al inscribirse en el gimnasio.

## **1.2. IDENTIFICACIÓN DEL PROBLEMA**

### **1.2.1. PROBLEMA CENTRAL**

En la actualidad en el gimnasio, no existe un seguimiento efectivo sobre el proceso evolutivo de la persona que acude al mismo. Dicho seguimiento debe comprender:

- **La rama de control de peso:**

Si bien es cierto que en el gimnasio hay un control de peso cada 2 días, no hay manera de que el cliente sepa su evolución de su peso en el gimnasio, esto ayudaría a la motivación del cliente cada vez que revisa su evolución y ve sus resultados (Ver Figura 1-1).

- **La rama de exámenes médicos:**

En el gimnasio se realizan 2 exámenes médicos para ver el estado de salud del cliente que son: examen de sangre y examen cardiológico. Luego de realizados estos exámenes, el cliente no puede consultar a lo largo de su membresía los resultados de estos exámenes.

- **La rama de alimentación efectiva:**

Al suscribirse al gimnasio, el cliente debe tener una cita con un nutricionista, este servicio es ofrecido por el mismo gimnasio. El nutricionista le realiza los controles de peso respectivos y le suscribe una dieta que el cliente debe seguir. El problema se encuentra en que el cliente tiene impreso o por correo su dieta y puede traspapelarse o perderse. Por otro lado, el único canal de comunicación con el nutricionista es vía correo, no existe un medio que indique si el nutricionista se encuentra disponible o no para atender vía correo al cliente.

Por otro lado, el cliente, al tener una dieta estricta que debe cumplir, no tiene alternativas de alimentación, con la salvedad que le realice las preguntas respectivas a su nutricionista. Por ende, el cliente no puede realizar una toma de decisión correcta ante alguna posibilidad de elección alterna a su dieta establecida.

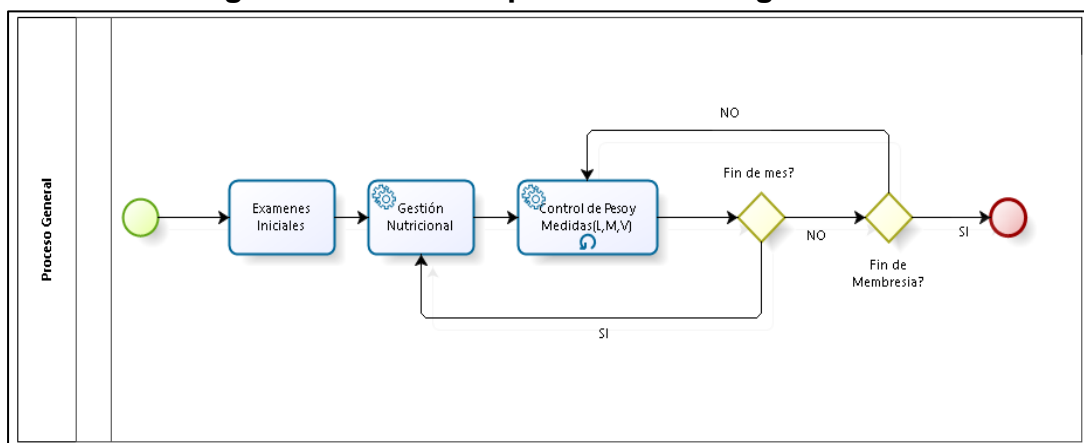
Al no haber dicho seguimiento efectivo y tener dicha información centralizada y de fácil acceso para el cliente, el cliente no puede ver su evolución y le toma tiempo realizar las consultas ya que solo al acercarse a los módulos respectivos de cada rama, puede obtener la información que él podría necesitar de alta disponibilidad.

## 1.2.2. PROBLEMAS ESPECÍFICOS

Se tiene los siguientes problemas específicos:

- a) Poca disponibilidad de información de resultados.
- b) La dieta no está al alcance en cualquier momento.
- c) Poca comunicación con el Nutricionista.
- d) Incertidumbre de pérdida calórica cardiovascular.

**Figura N° 1-1: Macroprocesos del Negocio**



Fuente: Elaboración propia

- Desarrollo de un sistema basado en una arquitectura poco flexible; habíamos escuchado sobre el término Clean Code, pero siendo mucho más específicos: Clean Architecture. Al indagar sobre esta nueva arquitectura que se está aplicando de manera reciente en los proyectos, no encontramos algún proyecto de donde podamos “guiarnos” para poder construir la arquitectura limpia en mención, todos los proyectos

construidos entrelazan sus clases de tal manera que si alguna es removida del proyecto, se debe refactorizar todo el proyecto, generando una pérdida de tiempo.

Este es un gran problema que aqueja a varios proyectos, el no tener una arquitectura robusta y escalable que permita realizar arreglos en el proyecto o correcciones sin tener que realizar demasiado trabajo.

Pudimos ver dicho problema al intentar realizar un cambio de un proyecto y nos dimos con la sorpresa que al realizar dicho cambio, la mitad del proyecto tuvo que ser refactorizado ya que la clase que fue cambiada tenía demasiados lazos con otras clases y así sucesivamente.

## **1.3. OBJETIVOS**

### **1.3.1. OBJETIVO GENERAL**

Desarrollar un aplicativo en android para facilitar la adopción de un estilo de vida sano en los clientes del gimnasio Personal Training.

### **1.3.2. OBJETIVOS ESPECÍFICOS**

Al plantear el objetivo general y evaluarlo, hemos encontrado los siguientes objetivos específicos que ayudarán a solucionar de una manera más efectiva el objetivo general antes descrito:

- a) Desarrollar el módulo de Recomendaciones Alimenticias para ofrecerle opciones alimenticias al deportista a raíz de la dieta que el nutricionista le ha recomendado tomar, haciendo comparaciones calóricas de los alimentos sustitutos con los alimentos de la dieta.

- b) Desarrollar el módulo de Información Personal para ofrecerle su información alimentaria, de peso y de salud de manera siempre disponible, para que el deportista pueda ver su avance y evolución en el gimnasio.
- c) Desarrollar el módulo de Consultas Alimentarias para ofrecerle siempre que productos de la barra alimentaria puede consumir de acuerdo al programa en el que se encuentre inscrito y que opciones son las más acordes para llegar a su meta de manera correcta y saludable.
- d) Desarrollar el módulo de Consultas de Evaluaciones para diagnosticar al deportista que tipo de ejercicios puede realizar y cuáles no de acuerdo a los resultados de sangre y cardiológicos que haya obtenido al registrarse en el gimnasio, indicar también al deportista cuáles realizar o no si posee alguna imposibilidad muscular.
- e) Informar al deportista cuántas calorías ha perdido de acuerdo al esfuerzo que utilice en cada sesión de entrenamiento.

## **1.4. DESCRIPCIÓN Y SUSTENTACIÓN DE LA SOLUCIÓN**

### **1.4.1. DESCRIPCIÓN DE LA SOLUCIÓN**

Lo que se pretende cubrir y lograr en este proyecto son los siguientes puntos importantes:

- Lo primero es automatizar el proceso de consultas de alimentos que posee la barra de alimentos del gimnasio, dicha automatización está contenida dentro de los módulos de Gestión de Alimentos y Consultas.



- Lo segundo es automatizar el proceso de control evaluaciones tanto cardiológicas como físicas que se realizan anual y semanal respectivamente. Esta automatización está contenida dentro del módulo de Consulta de Evaluaciones.
- Lo tercero es la construcción y despliegue del módulo de Recomendaciones Alimenticias; en este módulo se le podrá recomendar a los clientes diferentes alimentos según la dieta que se le asigne.

#### **1.4.2. JUSTIFICACIÓN DE LA REALIZACIÓN DEL PROYECTO**

Debido a la innovación del proyecto MyTrainingGoal, se facilitará la adopción de un estilo de vida sano en los clientes del gimnasio Personal Training, mediante el desarrollo de un aplicativo en android. Así como también se tendrá mayor acogida de nuevos clientes, lo que significa mayores ingresos para el gimnasio.

Al utilizar los principios de la arquitectura limpia, se podrá realizar cualquier cambio en alguna clase del proyecto, sin necesidad de preocuparse porque el cambio pueda dañar otra clase.

## CAPÍTULO II: MARCO CONCEPTUAL

### 2.1. MARCO CONCEPTUAL

#### CONCEPTO: CLEAN ARCHITECTURE

Según Pablo Carcelén Amores (2014):

“A real case of a cognitive multiplayer game: design and development of the iOS client in Swift and its cloud architecture. The clean architecture encourages the use case oriented design. This means that everything begin with a use case (a business rule specific to the application) that is independent from the rest. So one could for example change the database or the delivery mechanism without affecting the core of the application.

Los principios en que se basa esta arquitectura se pueden apreciar en la Tabla 2-1” (p. 48):

**Tabla 2-1: Principios SOLID de Clean Architecture**

Inicial	Acrónimo	Concepto
S	SRP	Principio de responsabilidad única (Single responsibility principle): La noción de que un objeto solo debería tener una única responsabilidad.
O	OCP	Principio de abierto/cerrado (Open/closed principle): La noción de que las “entidades de software deben estar abiertas para su extensión, pero cerradas para su modificación”.
L	LSP	Principio de sustitución de Liskov (Liskov substitution principle): La noción de que los “objetos de un programa deberían ser reemplazables por instancias de sus subtipos sin alterar el correcto funcionamiento del programa”.
I	ISP	Principio de segregación de la interfaz (Interface segregation principle): La noción de que “muchas interfaces cliente específicas son mejores que una interfaz de propósito general”.
D	DIP	Principio de inversión de la dependencia (Dependency inversion principle): La noción de que se debe “depender de

		abstracciones, no depender de implementaciones”. La Inyección de Dependencias es uno de los métodos que siguen este principio.
--	--	---

Fuente: Pablo Carcelén, Clean Architecture, 2014

Lo que el autor de esta tesis nos quiere dar a entender es que la arquitectura limpia se basa en el diseño de orientación a casos de uso. Todo debe iniciar con un caso de uso; es por eso que si nosotros desearíamos cambiar el tipo de carga de imágenes de Picasso a Glide, pues dicho cambio no debería afectar al sistema ni ocasionar una caída del mismo.

Este mismo autor nos muestra la siguiente imagen (Ver anexo 2).

Interpretando el gráfico que encontramos en la parte superior, nos quiere dar a entender que las clases debe acceder a las clases desde la capa superior hacia la inferior, más no realizarlo de manera inversa ya que las clases se entrelazarían demasiado y no podría por ejemplo cambiar el motor de lectura de imágenes ya que estaría entrelazado con otras clases.

Pablo Carcelén Amores (2014), nos explica con más detalles el significado de cada capa en la siguiente Tabla 2-2 (p. 55):

**Tabla 2-2: Capas de Clean Architecture**

Capas	Descripción
<b>Views</b>	Contenido de las pantallas originadas en el presentador. Es un mecanismo de entrega, que no se puede reutilizar entre plataformas.
<b>Presenter</b>	Se centran en hacer que las respuestas del interactor estén listas para mostrarse. Transforma los modelos del interactor en modelos de vista. Se trata de una pieza importante, ya que puede ser reutilizado entre las plataformas (a diferencia de la vista, que son específicos de una plataforma).
<b>Interactor</b>	Son los encargados de modelar un caso de uso, contienen las reglas de negocio y por lo general actúa como un mediador entre la base de datos y el presentador.
<b>Entities</b>	Son los objetos del modelo manipulados por los indicadores, estos objetos nunca llegan a la capa de

	presentación.
<b>Routing</b>	Representan cómo fluyen las pantallas en una aplicación. Cuando una vista se va a presentar, es responsable de hacer que aparezca en la pantalla.

Fuente: Pablo Carcelén, Clean Architecture, 2014

Clean Architecture plantea que la programación adopte, de tal manera que la programación sea más entendible y que pueda ser escalable y robusta.

## CONCEPTOS: MYBATIS

Según la página web oficial de MyBatis:

“Es un framework de persistencia que soporta SQL, procedimientos almacenados y mapeos avanzados. MyBatis elimina casi todo el código JDBC, el establecimiento manual de los parámetros y la obtención de resultados. MyBatis puede configurarse con XML o anotaciones y permite mapear mapas y POJOs (Plain Old Java Objects) con registros de base de datos”.

Se puede apreciar el logo oficial de MyBatis en la Figura 2-1.

**Figura N° 2-1: Logo de MyBatis**



Fuente: Página Web Oficial de MyBatis

## CONCEPTOS: ANDROID

Según la página web oficial de Android:

“Android, the world's most popular mobile platform. Android powers hundreds of millions of mobile devices in more than

190 countries around the world. It's the largest installed base of any mobile platform and growing fast—every day another million users power up their Android devices for the first time and start looking for apps, games, and other digital content.

Android gives you a world-class platform for creating apps and games for Android users everywhere, as well as an open marketplace for distributing to them instantly”.

Éste extracto obtenido de la página antes mencionada nos dice que Android es la mayor base instalada de cualquier plataforma móvil. Esta base tiene un crecimiento muy rápido.

Android está empujando constantemente los límites de hardware y software hacia adelante, para traer nuevas capacidades para los usuarios y desarrolladores. Para los desarrolladores, la rápida evolución de la tecnología Android le permite mantenerse al frente con aplicaciones potentes y diferenciadas. El logo de android es un pequeño androide de color verde, ver Figura 2-2.

Android le da acceso a las últimas tecnologías e innovaciones a través de una multitud de factores de forma de dispositivos, arquitecturas entre otros. Desde el procesamiento de múltiples núcleos y gráficos de alto rendimiento para sensores con tecnología de última generación, pantallas táctiles, vibrantes y emergentes tecnologías móviles.

**Figura N° 2-2: Logo de Android**



Fuente: Página Web Oficial de Android

## CONCEPTO: HISTORIA DE USUARIO

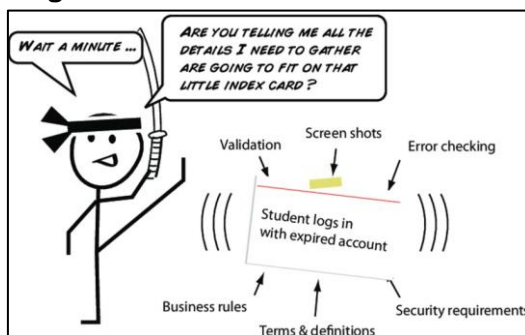
Según la página web oficial de PMO:

“Las metodologías ágiles como Scrum utilizan las historias de usuario como el instrumento principal para identificar los requerimientos de usuario”.

Las historias de usuario son descripciones cortas y simples de una funcionalidad, escritas desde la perspectiva de la persona que necesita una nueva capacidad de un sistema, por lo general el usuario, área de negocio o cliente. (Ver Figura 2-3).

Típicamente las historias siguen una plantilla simple: Yo como un [Rol], necesito [Descripción de la funcionalidad], con la finalidad de [Descripción de la consecuencia].

**Figura N° 2-3: Historias de Usuarios**



Fuente: Página Web PMO (<http://www.pmoinformatica.com/>)

## CONCEPTO: CALORÍA

Según Carol Hodges (1994):

“Las calorías son una unidad de medida para determinar la cantidad de energía que proviene de los alimentos; para eso se utilizan las kilocalorías, aunque en algunos lugares son usados los kilo-joules. De ahí que una caloría sea igual a 4.2 kilo joules. Cuando se consumen menos calorías de las necesarias se tiene como resultado un bajo peso de la persona. Cuando la ingesta calórica es mayor a la necesaria la persona adquirirá un sobrepeso. Es muy recomendable

tener una alimentación balanceada para adquirir la ingesta calórica necesaria y poder mantener un peso corporal estable. Para saber la ingesta necesaria de calorías y mantener el peso de una persona, Hodges recomienda lo siguiente:

- Hombres: el peso en libras multiplicado por 11.
- Mujeres: el peso en libra por 10". (p. 235)

Es importante optimizar la actividad física de la persona, la masa muscular y los factores genéticos, para poder saber que tan efectiva puede resultar estas fórmulas.

Como dice Judith E. Brown (1999):

“El valor de calorías necesarias para un buen metabolismo dependerá de la actividad que se realice. El costo de energía de apoyo para una persona inactiva es del 30% de las calorías necesarias. Para alguien con actividad promedio es del 50%, y de mucha actividad del 75%. Estas calorías necesarias son aproximadamente de 2145 calorías por día.

Los carbohidratos, lípidos y proteínas contienen calorías y proveen energía al cuerpo humano. Se ha comprobado que por un gramo de carbohidratos se obtienen cuatro calorías, un gramo de lípidos equivalen nueve calorías y un gramo de proteínas aporta cuatro calorías”. (p. 75)

## **CONCEPTO: EQUIVALENCIAS ALIMENTARIAS**

Según Judith E. Brown (1999):

“Para lograr un balance nutricional es recomendable consumir al día alimentos de los principales grupos alimenticios:

- Leche, yogurt, queso
- Carnes, aves, pescado, frijoles, huevos, nueces
- Vegetales, frutas
- Pan, cereal, arroz y pastas". (p. 109)

Wardlaw y Kessel (2002) nos dicen que:

“es recomendable agregar grasas a la dieta con moderación con tal de enfatizar los sabores y ayudar al aporte de otros nutrientes, como ciertas vitaminas y ácidos grasos esenciales.

El comer moderadamente requiere una planeación de la dieta que va a ingerir al día, por lo que es fundamental no excederse en el consumo de fuentes de nutrientes. Un buen método es realizar una combinación de grupos alimenticios: si en un día se consumen en exceso alimentos de un grupo nutricional predominante, el resto del día deberá consumir alimentos de los otros grupos para tratar de llegar a un balance de recuperación. Un ejemplo de lo anterior sería que a cierta hora del día si se consumen hamburguesas con queso y tocino postres, refrescos y otras bebidas, combinarlos con frutas, ensaladas, el resto del día. Resulta más recomendable moderar su consumo, que el eliminar su ingesta. Por lo tanto, la variedad, balance y la moderación controlan y prevén riesgos sustanciales relacionados con enfermedades de tipo nutricional”. (p. 59)

Por otro lado Brown (1999), indica que:

“el monto diario de alimentos necesarios para los seres humanos depende de muchos factores como la edad, sexo, estatura, peso y actividad física; por tal motivo existen diferentes tipos de tablas nutricionales que se



basan en uno o varios factores influyentes en los requerimientos nutricionales por día. Brown, muestra una tabla en la que los requerimientos nutricionales se dividen en cuatro grupos de alimentos a los que influyen la edad, talla corporal y el nivel de actividad. Se mide por porciones y una forma de saber aproximadamente cuantas porciones consumir es a través de la edad: A menor edad menos porciones y a mayor edad más porciones (ver Tabla 2-3)". (p. 130)

**Tabla 2-3: Guía de alimentos para comer sanamente**

<b>PORCIONES POR DIA</b>	
<b><u>CEREALES</u></b>	
5 – 12 Porciones	
<b><u>VEGETALES Y FRUTAS</u></b>	
5 – 10 Porciones	
<b><u>PRODUCTOS LACTEOS</u></b>	
De 2-3 años	4 – 9 porciones
De 3-4 años	10 – 16 porciones
Mujeres	3 – 4 porciones
Hombres	3 – 4 porciones
<b><u>CARNICOS Y ALTERNATIVOS</u></b>	
2 – 3 Porciones	

Fuente: Judith E. Brown, Nutrition Now, 1999

Wardlaw y Kessel (2002) hablan sobre:

“la referencia de administración de nutrientes [RNI’s] la cual fue adoptada en Canadá en 1991 y esta es usada por varios profesionales. La Tabla 2-4 muestra el número de calorías necesarias por día con respecto a la edad y el peso basado en la RNI’s” (p. 80):

**Tabla 2-4: Recomendaciones calóricas según la edad, sexo, y peso**

Edad	Sexo	Peso (Kg)	Kcal/día
<b>Bebes (meses)</b>			
0 - 2	Ambos	4.5	500
3 - 5	Ambos	7.0	700
6 - 8	Ambos	8.5	800
9 - 11	Ambos	9.5	950
<b>Niños y adultos (años)</b>			
1	Ambos	11	1100
2 - 3	Ambos	14	1300
4 - 6	Ambos	18	1800
7 - 9	Hombre	25	2200
	Mujer	25	1900
10 - 12	Hombre	34	2500
	Mujer	36	2200
13 - 15	Hombre	50	2800
	Mujer	48	2200
16 - 18	Hombre	62	3200
	Mujer	53	2100
19 - 24	Hombre	71	3000
	Mujer	58	2100
25 - 49	Hombre	74	2700
	Mujer	59	1900
50 - 74	Hombre	73	2300
	Mujer	63	1800
75 +	Hombre	69	2000
	Mujer	64	1500

Fuente: Wardlaw y Kessel, Perspectives in Nutrition, 2002

Alicia Bernard (2005), nos explica:

“otra forma de mostrar los requerimientos de la ingesta diaria de alimentos es la Pirámide Alimenticia en la cual se miden los alimentos por porciones y se dividen en seis grupos de alimentos, esta se puede ver en la Tabla 2-5”.  
(p. 39)

**Tabla 2-5: Recomendaciones de alimentos por la pirámide alimenticia**



Fuente: Alicia Bernard, 2005

Es necesario considerar la importancia que adquiere el conocer la distribución del aporte calórico total de los alimentos consumidos a lo largo del día, con el fin de poder fijar una recomendación de ingesta diaria que no presente excedentes calóricos y se adecue a los requerimientos de la persona según la actividad laboral que realice. Esto es, valorar el nivel de calorías ingerido durante el desayuno, la comida y la cena.

Bernard (2005), comenta:

“que del total de calorías ingeridas por un trabajador, el 35% podría ser aportado por el desayuno, el 50% por la comida y un 15% restante por la cena. Por ejemplo, en el caso de un trabajador con edad entre 25 y 50 años que realice una actividad moderada, su ingesta de calorías deberá estar en un promedio de 2600 kilocalorías aproximadamente. Por lo tanto, al aplicar los porcentajes 910 kilocalorías deberán ser ingeridas durante el desayuno, 1300 kilocalorías durante la comida y 390 en la cena”. (p. 57)

## **CONCEPTO: JAVA**

Según la página web oficial de SUN:

“Java es un lenguaje de programación orientado a objetos creado por Sun Microsystems en el año 1991, para poder funcionar en distintos tipos de procesadores”.

El código Java, una vez compilado, puede llevarse sin modificación alguna sobre cualquier máquina, y ejecutarlo. Esto se debe a que el código se ejecuta sobre una máquina virtual, la Java Virtual Machine (JVM), que se encarga de interpretar el código (archivos compilados .class) y convertirlo a código particular de la CPU en donde esté corriendo (siempre que se soporte dicha máquina virtual). Sun Microsystems define tres plataformas en un intento por cubrir distintos entornos de aplicación. Así, ha distribuido muchas de sus APIs de forma que pertenezcan a cada una de las plataformas: Java ME (Java Platform, Micro Edition) o J2ME, orientada a entornos de limitados recursos, como teléfonos móviles, PDAs, etc. Java SE (Java Platform, Standard Edition) o J2SE, para entornos de gama media y estaciones de trabajo. Aquí se sitúa al usuario de un PC de escritorio. Java EE (Java Platform, Enterprise Edition) o J2EE, orientada a entornos distribuidos empresariales o de Internet.

## **2.2. ESTADO DEL ARTE**

### **2.2.1. TRABAJOS REALIZADOS**

Se investigaron acerca de aplicaciones Android relacionadas con dietas y nutrición:

- **Contador de Calorías – MyFitnessPal:**

Es una aplicación Android para perder peso, tiene un contador de calorías rápido y fácil de usar, además cuenta con una base de datos de alimentos. Esta aplicación ayuda a controlar los principales nutrientes: calorías, grasas, proteínas, carbohidratos, fibra y colesterol en los alimentos de la dieta, además de controlar el progreso en la misma.

Además esta aplicación cuenta con ejercicios, registra entrenamiento vascular, muscular, se puede apreciar el logo de la aplicación en la Figura 2-4.

**Figura N° 2-4: Logo MyFitnessPal**



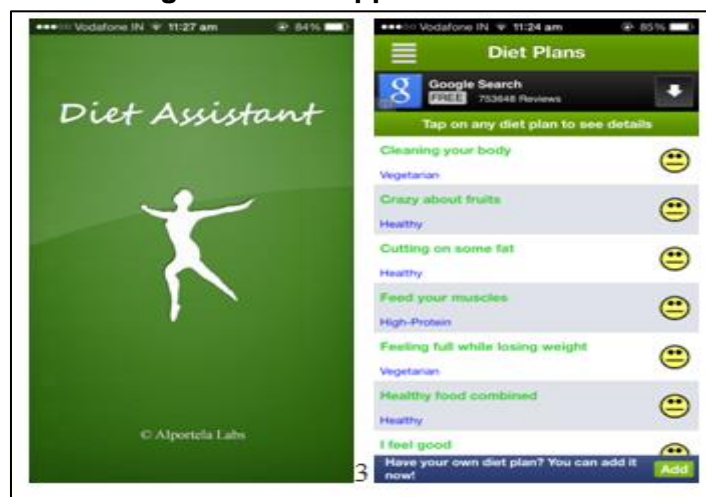
Fuente: Página Web Oficial de Myfitnesspal

- **Diet Assistant:**

Es una aplicación Android para asignación y control de dietas, en la cual el usuario puede fijar objetivos de peso a perder a través de las dietas, tener control de las dietas y de su progreso en esta, esta aplicación presenta una gran variedad de dietas para diferentes fines.(Ver Figura 2-5).

Esta aplicación también provee información acerca del valor nutricional de los alimentos de las dietas, como calorías, proteínas, grasas, etc.

**Figura N° 2-5: App Diet Assistant**



Fuente: Página Web Oficial de Diet Assistant

- **Dietas para adelgazar:**

Es un aplicación Android informativa, ayuda a seguir una dieta indicando el menú del día, además también genera una lista de compras de los ingredientes necesarios para la dieta escogida, en esta aplicación se puede registrar constantemente el peso para poder visualizar el progreso de uno en la dieta, además puede calcular el índice de masa corporal del usuario. (Ver Figura 2-6).

Esta aplicación provee distintos tipos de dietas como dietas para bajar peso, dietas rápidas, dietas para una alimentación sana, dietas vegetarianas y dietas para deportistas.

**Figura N° 2-6: Logo Dietas para Adelgazar**



Fuente: Página Web Oficial de Dietas para Adelgazar

## **2.2.2. HERRAMIENTAS PARA LA IMPLEMENTACIÓN**

Para este proyecto se utilizarán las siguientes herramientas:

1. Android: Se utilizará desde la versión 4.4 ya que estas versiones tienen compatibilidad con material design y, en la actualidad, son las versiones más usadas en el mercado.
2. MySQL: Se decidió utilizar el motor de base de datos MySQL ya que en nuestro sistema utilizaremos para el desarrollo únicamente software libre.

Según la página oficial de MySQL: “MySQL es la base de datos de código abierto número uno del mundo, es la base de datos número uno para Web y es una excelente base de datos embebida. Más de 3.000 ISVs y OEMs, incluyendo 8 de los 10 mayores, y 17 de los 20 principales proveedores de software de todo el mundo confían en MySQL como base de datos de sus productos”.

3. Netbeans: Se utilizará este IDE de desarrollo para los servicios web ya que el equipo de la tesis la conoce para el desarrollo de dichos web services y por el tiempo reducido no podemos realizar el estudio de algún otro IDE.
4. Java: Se utilizará java ya que el equipo de desarrollo de la tesis solo maneja este lenguaje para desarrollar los web services y java es considerado software libre.
5. Rational Rose: Se utilizará Rational Rose para el diseño de los diagramas del modelado del sistema. Además el equipo de tesis conoce el software y, para el tiempo planteado, puede modelar en este software sin ningún problema.

### **2.2.3.DEFINICIÓN DE TÉRMINOS**

Se tienen los siguientes términos:

- **Clean Architecture:** arquitectura limpia, basado en el principio SOLID.
- **Joules:** unidad del Sistema Internacional de Unidades para energía en forma de calor (Q) y trabajo (W).
- **CUS:** Abreviatura de Casos de Uso.

- **Entity:** representa una "cosa", "objeto" o "concepto" del mundo real con existencia independiente, es decir, se diferencia únicamente de otro objeto o cosa, incluso siendo del mismo tipo, o una misma entidad.

- **String:** es una secuencia ordenada (de longitud arbitraria, aunque finita) de elementos que pertenecen a un cierto lenguaje formal o alfabeto análogas a una fórmula o a una oración. En general, una cadena de caracteres es una sucesión de caracteres (letras, números u otros signos o símbolos).

- **Smartphone:** es un tipo de teléfono móvil construido sobre una plataforma informática móvil, con mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional.

- **APK (Android Application Package):** es un paquete para el sistema operativo Android. Este formato es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android para teléfonos inteligentes y tabletas, así como también algunas distribuciones enfocadas a su uso en ordenadores personales de escritorio y portátiles (Note y Netbook respectivamente) ejemplo: Remix OS.



## **CAPÍTULO III: DESARROLLO DEL PROYECTO**

### **3.1. ALCANCE DEL PROYECTO**

#### **3.1.1. DESCRIPCIÓN DEL ALCANCE DEL PRODUCTO**

El producto contará con las siguientes funcionalidades:

- Se mostrará la información de todos los programas que el gimnasio posee (Fuxion Xtreme, Muscle Xtreme, RPM 50, Abs): Se mostrará la información de cada programa.
- Se mostrará el perfil del usuario en donde se detallarán sus datos básicos, el nombre de su nutricionista asignado, el programa en el cual se ha inscrito y la fecha de fin de su membresía.
- Se mostrará la dieta que el nutricionista le ha asignado y el historial de dietas que ha tenido durante toda su estancia en el gimnasio.
- Se mostrará el histórico de evaluaciones físicas (Peso y completas) realizadas tanto por la persona encargada de los pesajes en los días determinados por el gimnasio y por el nutricionista asignado.
- Se mostrarán opciones de alimentos a la dieta que el nutricionista le ha asignado, para que el usuario pueda tener opciones de alimentación sana.
- Se mostrarán los productos que la barra de alimentos, ubicada en el gimnasio, le ofrece para consumir.

- El usuario podrá mantener una conversación con su nutricionista en un horario pactado vía un chat que el aplicativo le mostrará.
- Se mostrarán las experiencias que los usuarios dejan en el gimnasio al término de su membresía en el gimnasio.
- Se mostrarán los exámenes médicos que se realizaron al usuario al inicio de su membresía en el gimnasio.

### **3.1.2. CRITERIOS DE ACEPTACIÓN DEL PRODUCTO**

El aplicativo para los clientes del gimnasio deberá estar aprobado en cada etapa del proyecto, generando un acta de aprobación y la minuta de la reunión donde se expuso los resultados de la etapa para poder iniciar la siguiente.

Haciendo una comparativa con el aplicativo “Dietas para adelgazar”, el aplicativo estudiado si bien presenta varias dietas, el usuario no sabe, a ciencia cierta, si las dietas que se presentan en este aplicativo son evaluadas por algún nutricionista o si lo fueran, pues se sabe bien que una dieta pues no funciona para todos.

A diferencia de nuestra aplicación, que la dieta será construida por un nutricionista y cada dieta será diferente para cada usuario, es decir estas dietas no serán repetitivas

La única igualdad entre nuestro aplicativo y el aplicativo “Dietas para adelgazar” es que se manejarán recordatorios por cada comida para que el usuario no olvide que debe comer a determinadas horas indicadas por el nutricionista

Luego comparando nuestro aplicativo con el aplicativo “Diet Assistant”, nos dimos cuenta que tiene la misma funcionalidad que el aplicativo “Dietas para adelgazar”, es decir que en este aplicativo el

usuario mismo planea su dieta, volviendo al mismo dilema en el que una dieta no funciona para todas las personas.

La única igualdad entre nuestro aplicativo y el aplicativo “Diet Assistant” es que mantienen un histórico de fotos por cada evaluación que el usuario se realiza; en el caso de nuestro aplicativo pues la evaluación la realiza el mismo gimnasio y dicho histórico se muestra en el perfil del usuario.

### **3.1.3. ESTRUCTURA DE DESGLOSE DEL TRABAJO Y ENTREGABLES**

El proyecto será un aplicativo móvil para los clientes del gimnasio Personal Training, por lo cual los entregables serán los siguientes:

- Prototipos del aplicativo
- Paquetes de instalación
- Documentación general del proyecto generada en cada etapa en un documento del tipo .pdf.

Como se muestra en la Tabla 3-1, éstos serían nuestros hitos del proyecto:

**Tabla 3-1: Hitos del Proyecto**

Hito	Fecha tope
Creación del WBS	08/05/2016
Fin Módulo Gestión	14/05/2016
Fin Módulo Documento de Tesis	20/08/2016
Fin Módulo Análisis y Diseño	24/09/2016
Fin Módulo Construcción	30/09/2016
Fin Ajustes de la Documentación	02/10/2016
Fin Despliegue	04/11/2016

Fuente: Elaboración propia

### **3.1.4. EXCLUSIONES DEL PROYECTO**

El proyecto no abarca el módulo de contador de calorías realizadas luego de realizar alguna actividad física mediante el uso del smartwatch, debido a que el tiempo es muy corto para la implementación.

### **3.1.5. RESTRICCIONES DEL PROYECTO**

El aplicativo solo funciona para dispositivos móviles con sistema operativo Android 4.4 y superiores.

El aplicativo solo será para smartphones y no para tablets.

El aplicativo solo se verá en modo vertical y no en horizontal.

El aplicativo solamente funcionará si tiene acceso a internet, por ende no funcionará de manera offline.

### **3.1.6. SUPUESTOS DEL PROYECTO**

El proyecto se llevará a cabo bajo los siguientes supuestos:

- Se cuenta con todas las herramientas de software necesarias para el desarrollo y diseño del aplicativo.
- Se cuenta con todas las herramientas necesarias para un correcto testeo del aplicativo.
- El gimnasio otorgará accesos a sus servidores para la instalación de archivo compilado de los web services.
- El gimnasio adquirirá una cuenta en el google play store para subir el aplicativo.

# CAPÍTULO IV: DESARROLLO DEL PRODUCTO

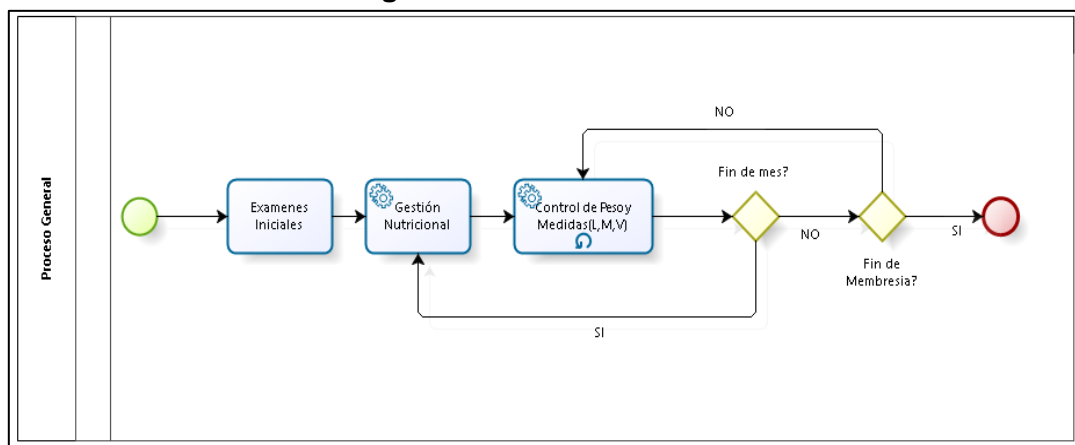
## 4.1. MODELADO DEL NEGOCIO

### 4.1.1. MODELO AS IS

En la actualidad los procesos generales del negocio son los siguientes:

- Realizar los exámenes iniciales, con el fin de registrar los resultados para tomarlos como puntos de partida.
- Luego el cliente tiene que asistir mensualmente al nutricionista para su control nutricional.
- Los días lunes, miércoles y viernes el cliente pasará los controles de peso y medidas.
- Todos estos procesos se llevan a cabo mientras no se termine la membresía del cliente en el gimnasio (Ver Figura 4-1).

Figura N° 4-1: Modelo AS IS



Fuente: Elaboración propia

#### **4.1.2. MODELO TO BE**

El modelo TO BE, muestra el proceso automatizado de Consulta de Dietas, el cual se puede apreciar en las siguientes dos figuras (Ver figuras 4-2):

El proceso comienza cuando el Nutricionista ingresa a la opción Registrar dietas, paso seguido el sistema muestra el formulario del Registro de la dieta Principal. El Nutricionista registra la dieta y selecciona el botón aceptar, luego le aparecerá el mensaje “Se realizó el registro exitosamente”.

El sistema mostrará el formulario del Registro de dieta Alternativa, al igual que en la actividad anterior, el nutricionista procede a registrar la dieta y selecciona el botón aceptar. Se muestra el mensaje “Se realizó el registro exitosamente”. Mediante el Backend se notifica el registro de las nuevas dietas al Cliente.

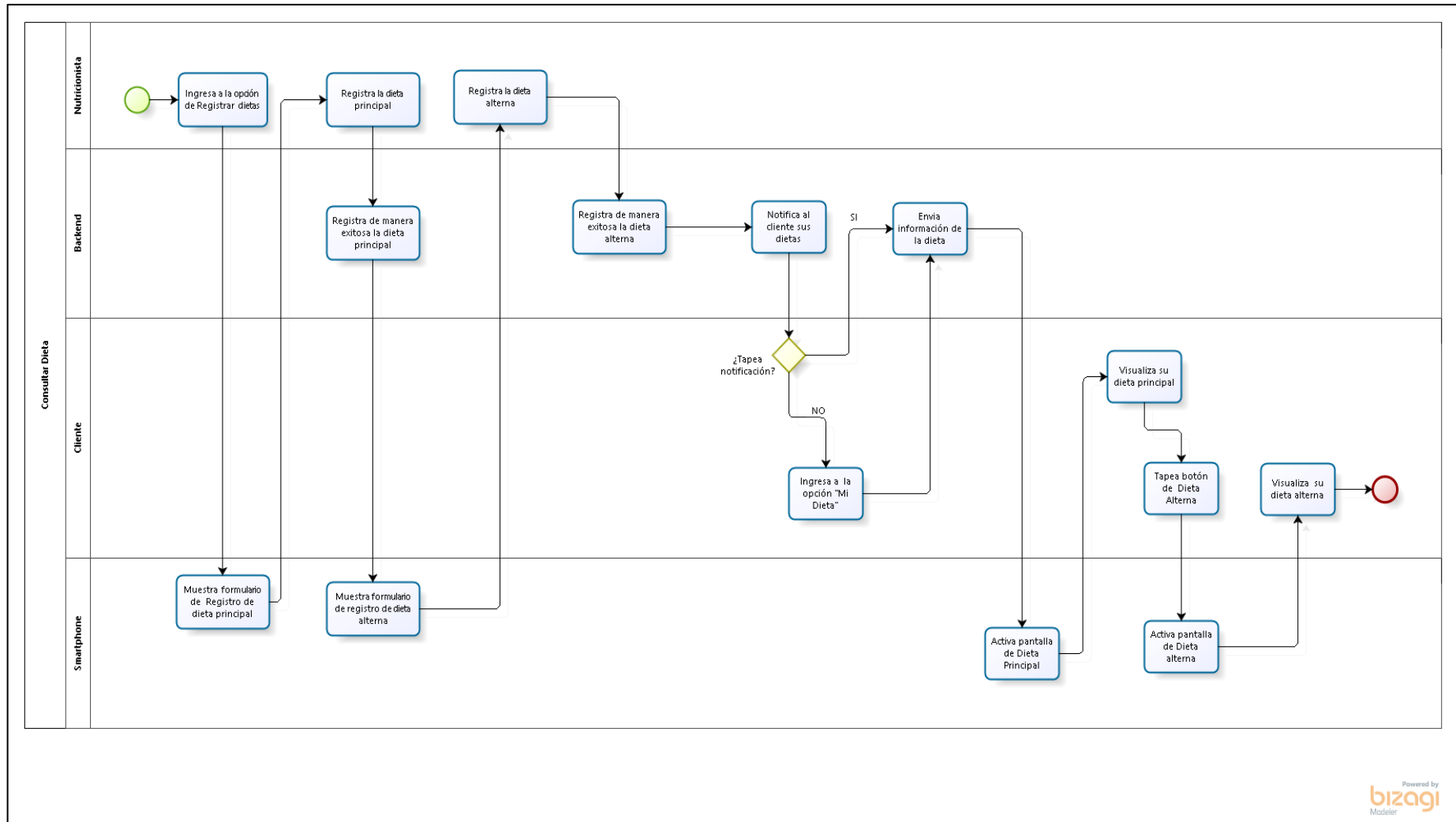
El cliente recibe la notificación de las dietas registradas:

Si tapinga la notificación, el backend envía la información de la dieta registrada.

Si no tapinga la notificación, debe ingresar a la opción Mi Dieta, al momento de ingresar, el backend envía la información de la dieta registrada.

Se activa la pantalla Dieta Principal, el cliente visualiza su dieta principal, luego tapinga la dieta alternativa y se activa la pantalla Dieta Alternativa. El cliente visualiza su dieta alternativa y termina el proceso.

Figura N° 4-2: Modelo TO BE



Fuente: Elaboración propia

En la Tablas 4-1, se describe cada actividad del proceso:

**Tabla 4-1: Descripción de actividades del modelo TO BE**

Nro.	Actividad	Input	Output	Dependencias
1	Ingresa a la opción de registrar dietas			Actividad Nro. 2
2	Muestra el formulario de Registro de Dieta Principal		Formulario	Actividad Nro. 1 y 3
3	Registra la dieta principal	Dieta Principal	Dieta Principal	Actividad Nro. 2 y 4
4	Registra de manera exitosa la dieta principal	Dieta Principal	Mensaje	Actividad Nro. 3 y 5
5	Muestra el formulario de Registro de Dieta Alterna		Formulario	Actividad Nro. 4 y 6
6	Registra la dieta alterna	Dieta Alterna	Dieta Alterna	Actividad Nro. 5 y 7
7	Registra de manera exitosa la dieta alterna	Dieta Alterna	Mensaje	Actividad Nro. 6 y 8
8	Notifica al cliente sus dietas	Dieta	Notificación	Actividad Nro. 7 y 9
9	Ingresa a la opción "Mi Dieta"		Formulario	Actividad Nro. 8 y 10
10	Envía la información de la dieta	Dieta	Dieta	Actividad Nro. 9 y 11
11	Activa pantalla de dieta principal		Formulario	Actividad Nro. 10 y 12
12	Visualiza su dieta principal		Dieta Principal	Actividad Nro. 11 y 13
13	Tapea botón de dieta alterna		Formulario	Actividad Nro. 12 y 14
14	Activa pantalla de dieta alterna		Formulario	Actividad Nro. 13 y 15
15	Visualiza su dieta alterna		Dieta Alterna	Actividad Nro. 14

Fuente: Elaboración propia

## 4.2. REQUERIMIENTOS DEL PRODUCTO/SOFTWARE

### 4.2.1. REQUERIMIENTOS FUNCIONALES

- RF01: El sistema permitirá consultar cada programa del gimnasio.
- RF02: El sistema deberá mostrar una pequeña descripción de cada programa al seleccionar la imagen del mismo.
- RF03: El sistema deberá mostrar el perfil de cada usuario.



- RF04: El sistema permitirá poder cambiar la contraseña del usuario.
- RF05: El sistema permitirá consultar las dietas recomendadas por el nutricionista al usuario.
- RF06: El sistema mostrará un máximo de 10 dietas equivalentes a cada usuario.
- RF07: El sistema mostrará en la parte inferior de cada dieta las recomendaciones nutricionales correspondientes.
- RF08: El sistema deberá mostrar el avance de medidas de cada parte del cuerpo mediante una imagen sencilla y amigable.
- RF09: El sistema deberá mostrar un histórico de medidas del usuario en la parte inferior.
- RF10: El sistema permitirá congelar los días que el usuario decida no asistir para que no sean días perdidos, todo esto lo debe realizar en la opción Calendario.
- RF11: El sistema permitirá reservar una cita con el nutricionista a través de la opción Calendario.
- RF12: El sistema deberá notificará usuario ante cualquier evento que se genere con una alerta push en su teléfono móvil.
- RF13: El sistema deberá contar con la opción de realizar encuestas, con el fin de validar la satisfacción de los usuarios por el uso del sistema y de cómo éste ayuda en su vida deportiva.
- RF14: El sistema manejará el tipo de información tales como: nombres, apellidos, edad, altura, peso, tallas, etc.).

## **4.2.2. REQUERIMIENTOS NO FUNCIONALES**

- RNF01: El sistema garantizará la confidencialidad y la seguridad de la información de los usuarios del gimnasio.
- RNF02: La disponibilidad del sistema será de un 100% para los usuarios del gimnasio, el sistema estará disponible 24 horas al día los 7 días de la semana garantizando así un esquema adecuado. Debido a que estará en el Google Play Store, podrá ser descargada cuando se desee.
- RNF03: El sistema tendrá un manual de usuario, en donde se describe detalladamente las opciones del mismo.
- RNF04: El sistema contará con una interfaz de uso sencilla y amigable.

## **4.3. ARQUITECTURA DEL PRODUCTO/SOFTWARE**

### **4.3.1. VISTA DE CASOS DE USO**

#### **4.3.1.1. DIAGRAMA DE PAQUETES**

El sistema contará con 3 paquetes, los cuales son:

##### **- Paquete de Seguridad:**

El paquete seguridad, es el paquete de partida, ya que sin él no podrían iniciar los otros dos paquetes (Administración y Consultas). (Ver Figura 4-3).

En este paquete se encuentran los casos de uso:

- \* Iniciar Sesión
- \* Olvide mi Contraseña

### **- Paquete de Administración:**

El paquete Administración depende del paquete Seguridad y sirve como base para el paquete Consultas, éste paquete es monitoreado por el actor SGC, el cual nos brinda toda la información para realizar las consultas. (Ver Figura 4-3).

En este paquete se encuentran los siguientes casos de uso:

- \* Administrar Alimentos
- \* Administrar Evaluaciones
- \* Administrar Evaluaciones Iniciales
- \* Administrar Dietas

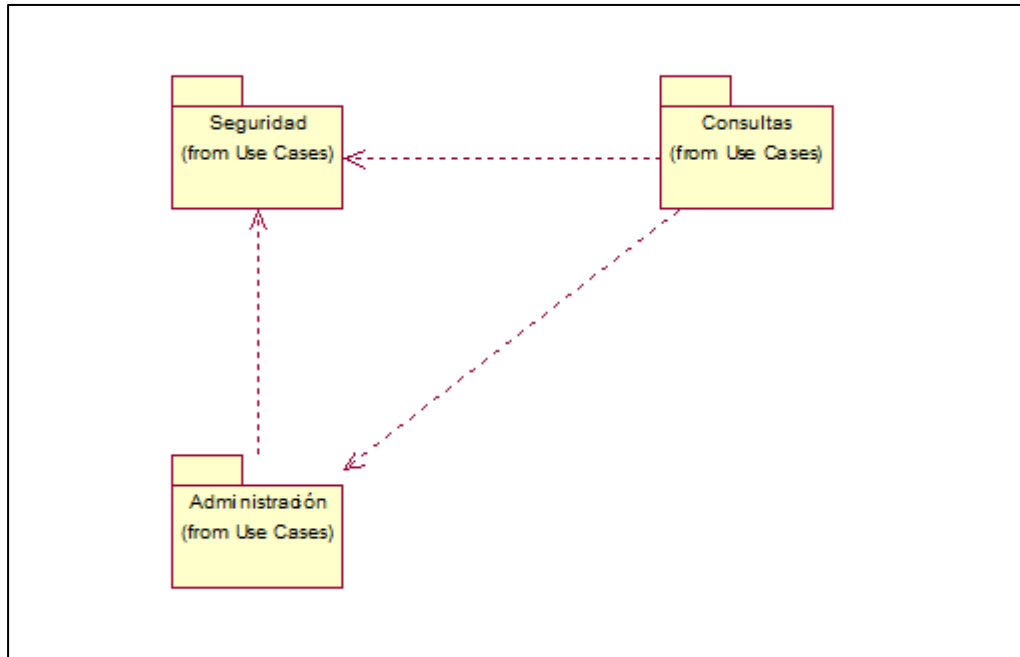
### **- Paquete de Consultas:**

El Paquete Consultas depende tanto del paquete Seguridad como del paquete Administración. Este paquete es el más importante del proyecto, puesto que se están automatizando todos sus Casos de Uso. (Ver Figura 4-3).

En este paquete se encuentran los siguientes casos de uso:

- \* Consultar Evaluaciones
- \* Consultar Evaluaciones Iniciales
- \* Consultar Programas Gimnasio
- \* Consultar Dietas
- \* Consultar Calendario
- \* Consultar Tienda
- \* Llenar Encuesta
- \* Consultar Notificaciones

**Figura N° 4-3: Diagrama de Paquetes del Sistema**



Fuente: Elaboración propia

#### **4.3.1.2. DIAGRAMA DE ACTORES**

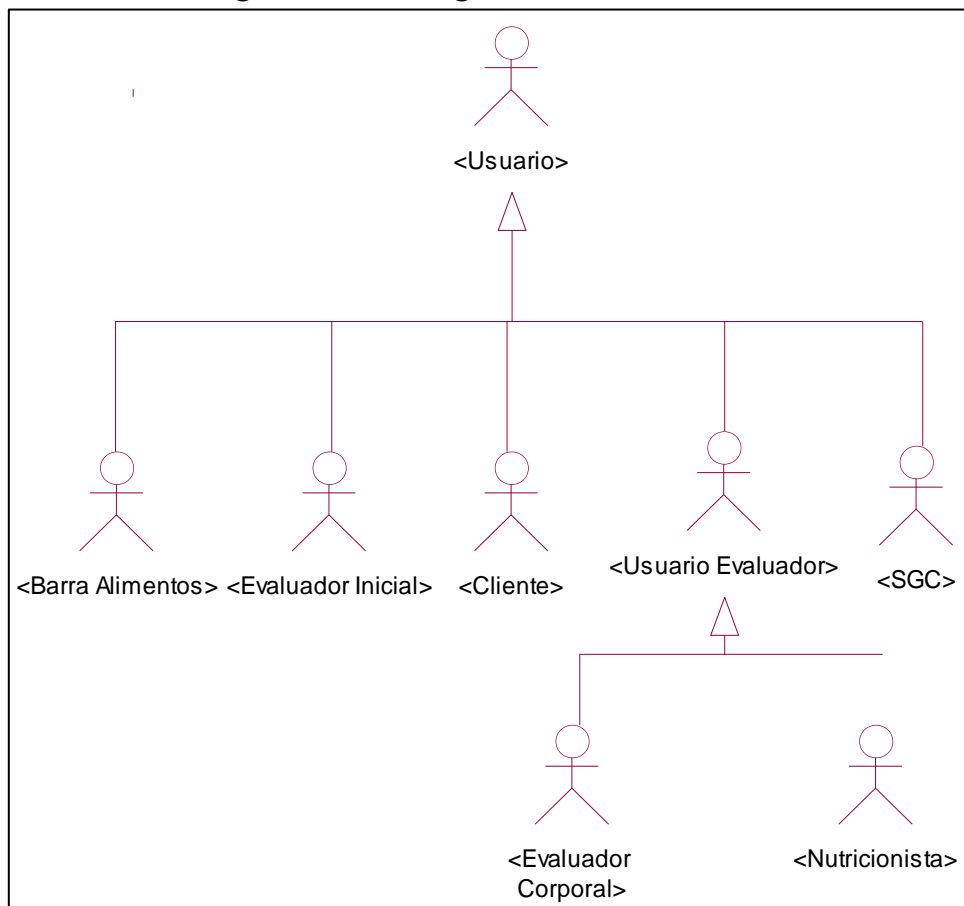
Basándose en los roles del proceso, se han identificado los siguientes actores del sistema:

- Usuario: Es el actor que generaliza a todos los demás actores, tiene el perfil de un administrador.
- Nutricionista: Es el actor encargado de registrar las dietas y responder las consultas del chat.
- Barra de Alimentos: Es el actor encargado de dar mantenimiento a los alimentos para las futuras dietas.
- Evaluador Corporal: Es el actor encargado de gestionar las evaluaciones corporales de todos los clientes.
- Evaluador Inicial: Es el actor encargado de gestionar sólo las evaluaciones iniciales de los clientes nuevos.

- Cliente: Actor básico con permisos de Consulta de todos sus exámenes y dietas.
- Usuario Evaluador: Es la generalización de los usuarios: Evaluador Corporal y Nutricionista.
- SGC: Actor del sistema web del Gimnasio, es un sistema externo el cual se encarga de proporcionar la data a mostrar en la aplicación mediante peticiones a su servidor. Además de soportar las operaciones que se deseen realizar en la aplicación.

En la figura 4-4, se muestra como están generalizados los actores del sistema en el actor Usuario:

**Figura N° 4-4: Diagrama de Actores del Sistema**



Fuente: Elaboración propia

### **4.3.1.3. DIAGRAMA DE CASOS DE USO**

Diagrama general de casos de uso en donde se observa la relación y distribución de los actores con sus respectivos casos de uso.

#### **\* CUS\_Iniciar Sesión:**

En este caso de uso se realiza el logueo al aplicativo, ingresando el código de usuario y la contraseña.

#### **\* CUS\_Olvide mi Contraseña:**

Este caso de uso se realiza si al momento de iniciar sesión se muestra un error en la contraseña. El usuario puede seleccionar la opción “Olvide mi Contraseña”, para que puede cambiar su contraseña.

#### **\* CUS\_Administrar Alimentos:**

En este caso de uso se realiza el registro, modificación y eliminación de los alimentos que podrán ser consultados en la barra de alimentos para poder crear las dietas.

#### **\* CUS\_Administrar Evaluaciones:**

En este caso de uso se registran, modifican y eliminan las evaluaciones de cada cliente, para que luego puedan ser consultadas y pueda ver cómo va su evolución en el gimnasio.

#### **\* CUS\_Administrar Evaluaciones Iniciales:**

En este caso de uso se registran, modifican y eliminan las evaluaciones iniciales de cada cliente, para que luego puedan ser consultadas.

\* CUS\_Administrar Dietas:

En este caso de uso se registran, modifican y eliminan las dietas de cada usuario, para que luego puedan ser consultadas y/o comparadas por los clientes.

\* CUS\_Consultar Evaluaciones:

En este caso de uso se realizan las consultas de las evaluaciones registradas de cada usuario, con el fin de ver su evolución.

\* CUS\_Consultar Evaluaciones Iniciales:

En este caso de uso se consultan las evaluaciones iniciales anteriormente registradas.

\* CUS\_Consultar Programas Gimnasio:

En este caso de uso se puede consultar el detalle de cada programa que brinda el gimnasio, con el fin de orientar a los clientes al momento de la inscripción.

\* CUS\_Consultar Dietas:

En este caso de uso se puede consultar hasta un máximo de 5 dietas por usuario para poder escoger la más adecuada.

\* CUS\_Consultar Calendario:

En este caso de uso se puede congelar los días que el cliente no pueda asistir con el fin de no perderlos y recuperarlos en otras fechas.

\* CUS\_Consultar Tienda:

En este caso de uso se realiza la consulta y generación de un ticket sobre la compra de un producto de la barra de alimentos.

\* CUS\_Llenar Encuesta:

En este caso de uso se realiza el llenado de una encuesta de satisfacción del aplicativo, con el fin de saber si le fue de mucha ayuda o no.

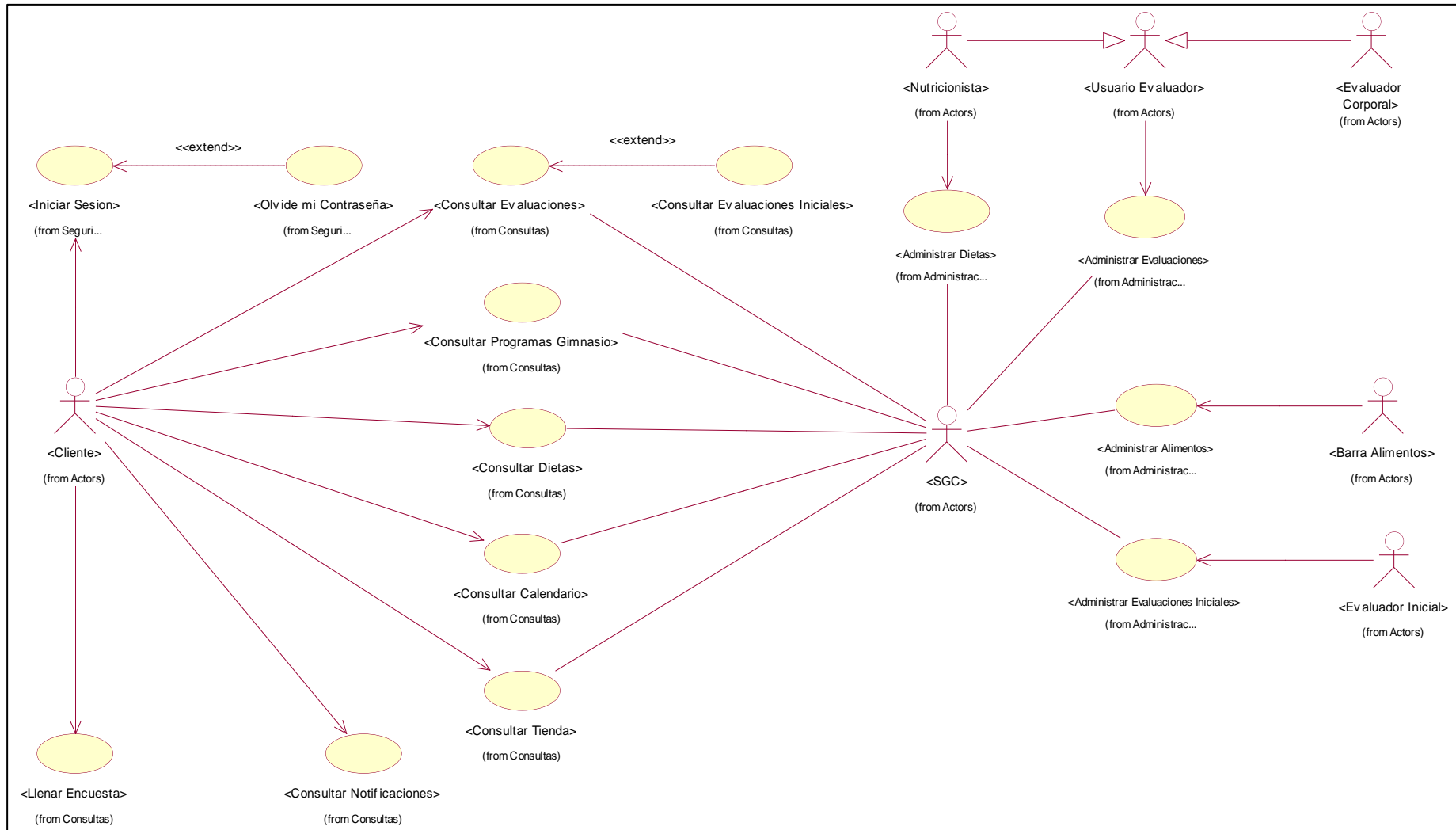
\* CUS\_Consultar Notificaciones:

En este caso de uso se realiza la consulta de las diferentes notificaciones que reciba el cliente, puede recibir notificaciones sobre evaluaciones, dietas, programas nuevos, etc.

A continuación se muestra la Figura 4-5, en donde se observa las relaciones entre casos de uso y actores del sistema:



**Figura N° 4-5: Diagrama General de Casos de Uso del Sistema**



Fuente: Elaboración propia

### 4.3.2. VISTA DE PROCESOS

En la vista de procesos se tienen los modelos AS IS y TO BE del negocio.

El modelo AS IS, nos muestra como son en la actualidad los procesos del gimnasio, estos son: exámenes iniciales, control de peso y medidas y exámenes nutricionales.

El modelo TO BE, nos muestra cómo es que serán automatizados cada uno de los procesos.

Ambos modelos están detallados en el punto 4.1.

### 4.3.3. VISTA LÓGICA

Para proporcionar una base para comprender la estructura y la organización del diseño del sistema, se realizó la vista lógica de las clases utilizadas en el sistema. (Ver Figura 4-6).

Se tienen las siguientes clases relacionadas:

- E\_Usuario: la cual tiene los siguientes atributos:

- \* idUsuario, de tipo Entero.
- \* nombre, de tipo String.
- \* apellidos, de tipo String.
- \* dirección, de tipo String.
- \* teléfono, de tipo String.
- \* celular, de tipo String.
- \* foto\_perfil, de tipo String.
- \* foto\_frente, de tipo String.
- \* token, de tipo String.
- \* horarioEntrenamiento, de tipo Date.

- E\_TipoUsuario: la cual tiene los siguientes atributos:

- \* idTipoUsuario, de tipo Entero.
- \* nombreTipoUsuario, de tipo String.

- E\_CitaNutricional: la cual tiene los siguientes atributos:
  - \* idCitaNutricional, de tipo Entero.
  - \* fechaCitaNutricional, de tipo Date.
  - \* horaCitaNutricional, de tipo Datetime.
  
- E\_PlanNutricional: la cual tiene los siguientes atributos:
  - \* idPlanNutricional, de tipo Entero.
  - \* cantidadAlimentos, de tipo Entero.
- E\_Alimentos: la cual tiene los siguientes atributos:
  - \* idAlimento, de tipo Entero.
  - \* nombreAlimento, de tipo String.
  - \* calorías, de tipo Double.
  
- E\_HistorialPlanNutricional: la cual tiene los siguientes atributos:
  - \* idHistorialPlanNutricional, de tipo Entero.
  - \* nombrePlanNutricional, de tipo String.
  - \* fechaPlanNutricional, de tipo Date.
  - \* horaPlanNutricional, de tipo Datetime.
  
- E\_ProgramaEjercicio: la cual tiene los siguientes atributos:
  - \* idProgramaEjercicio, de tipo Entero.
  - \* nombreProgramaEjercicio, de tipo String.
  - \* fraseProgramaEjercicio, de tipo String.
  - \* nivelFuerza, de tipo Entero.
  - \* nivelResistencia, de tipo Entero.
  - \* nivelIntensidad, de tipo Entero.
  - \* detalleProgramaEjercicio, de tipo String.
  
- E\_BarraAlimentos: la cual tiene los siguientes atributos:
  - \* idBarraAlimentos, de tipo Entero.
  - \* nombre, de tipo String.
  - \* horaApertura, de tipo Datetime.

- \* horaCierre, de tipo Datetime.

- E\_Productos: la cual tiene los siguientes atributos:

- \* idProducto, de tipo Entero.

- \* nombreProducto, de tipo String.

- \* calorías, de tipo Double.

- \* precio, de tipo Double.

- E\_TipoProducto: la cual tiene los siguientes atributos:

- \* idTipoProducto, de tipo Entero.

- \* nombreTipoProducto, de tipo String.

- E\_Dias: la cual tiene los siguientes atributos:

- \* idDia, de tipo Entero.

- \* nombreDia, de tipo String.

- E\_TipoEvaluacionFisica: la cual tiene los siguientes atributos:

- \* idTipoEvaluacionFisica, de tipo Entero.

- \* nombreTipoEvaluacionFisica, de tipo String.

- E\_EvaluacionFisica: la cual tiene los siguientes atributos:

- \* idEvaluacionFisica, de tipo Entero.

- \* peso, de tipo Double.

- \* brazo, de tipo Double.

- \* muñeca, de tipo Double.

- \* cuello, de tipo Double.

- \* pecho, de tipo Double.

- \* cintura, de tipo Double.

- \* gluteos, de tipo Double.

- \* muslos, de tipo Double.

- \* pantorrilla, de tipo Double.

- \* hombros, de tipo Double.

- \* fechaEvaluacionFisica, de tipo Date.

- E\_HistorialEvaluacionFisica: la cual tiene los siguientes atributos:

- \* idHistorialEvaluacionFisica, de tipo Entero.
- \* fechaHistorialEvaluacionFisica, de tipo Date.
- \* horaHistoralEvaluacionFisica, de tipo Datetime.

- E\_TipoEvaluacionInicial: la cual tiene los siguientes atributos:

- \* idTipoEvaluacionInicial, de tipo Entero.
- \* nombreTipoEvaluacionInicial, de tipo String.

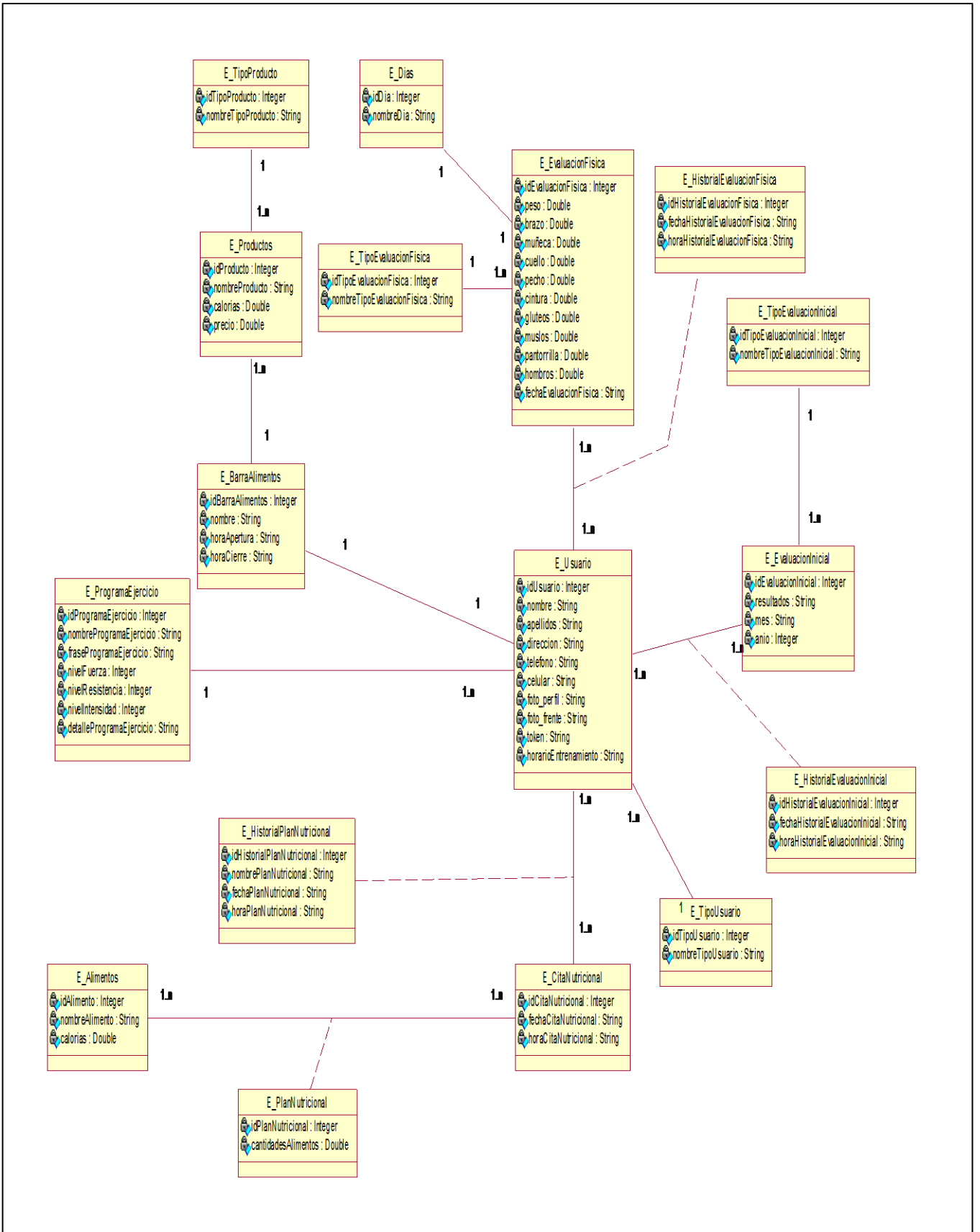
- E\_EvaluacionInicial: la cual tiene los siguientes atributos:

- \* idEvaluacionInicial, de tipo Entero.
- \* resultados, de tipo String.
- \* mes, de tipo Entero.
- \* año, de tipo Entero.

- E\_HistorialEvaluacionInicial: la cual tiene los siguientes atributos:

- \* idHistorialEvaluacionInicial, de tipo Entero.
- \* fechaHistorialEvaluacionInicial, de tipo Date.
- \* horaHistorialEvaluacionInicial, de tipo Datetime.

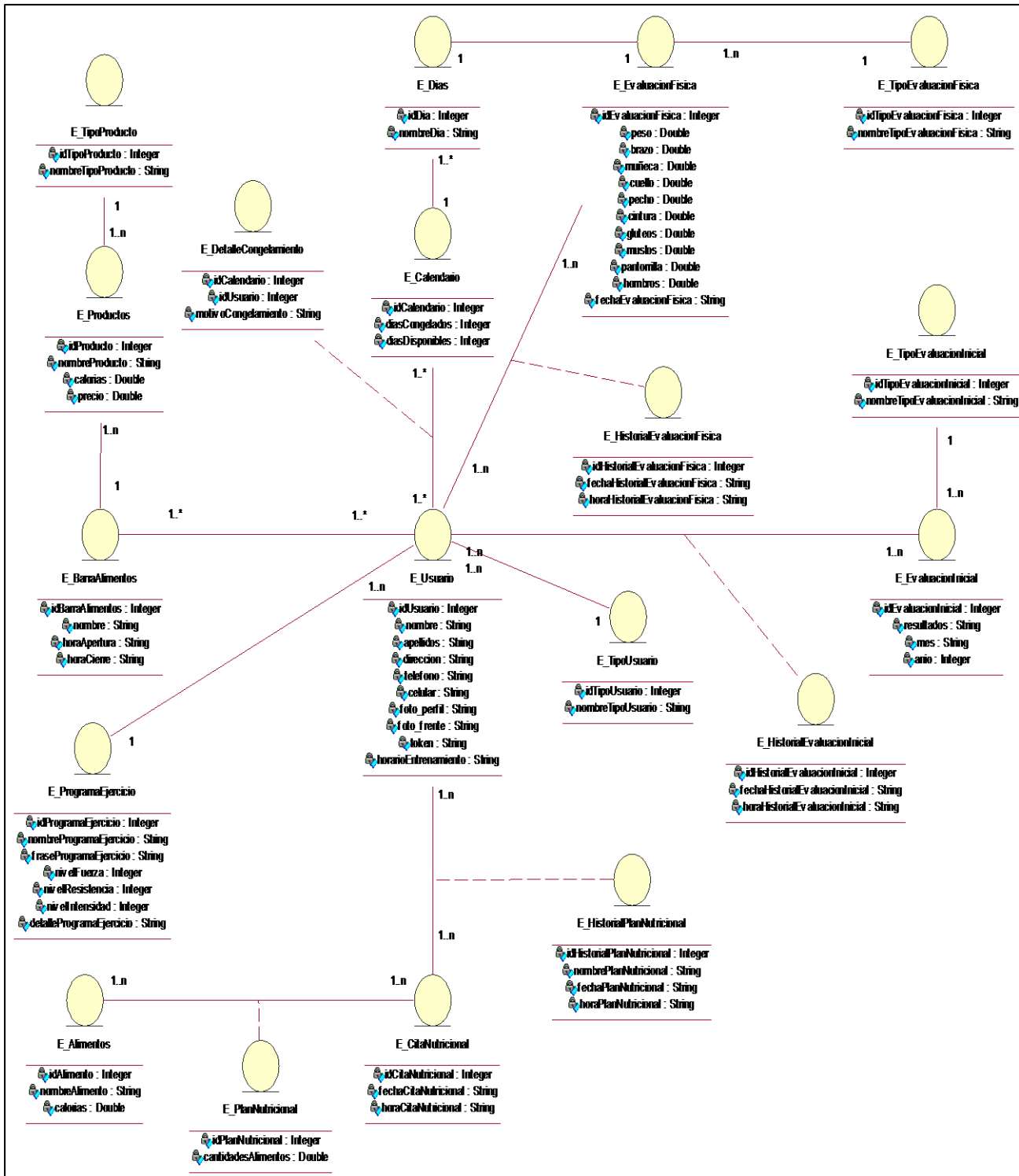
Figura N° 4-6: Vista Lógica del Sistema



Fuente: Elaboración propia

Además se tiene la Figura 4-7, de entidad- relación, la cual nos detalla las relaciones existentes entre las entidades del sistema:

Figura N° 4-7: Diagrama de Entidad-Relación



Fuente: Elaboración propia

#### 4.3.4. VISTA DE COMPONENTES

En el siguiente diagrama podemos ver las dependencias que hay entre los componentes de nuestro sistema para poder asegurar el funcionamiento adecuado del mismo (Ver Figura 4-8).

Se tienen los siguientes componentes:

- **APK:**

El APK es nuestro ejecutable, el cual depende de varias librerías para su correcto funcionamiento. Si una de estas librerías o llámense componentes es retirado del sistema, el proyecto no podría ejecutarse y fallaría al momento de su compilación. Finalmente, el APK depende del sistema SGC para que la información proporcionada por el SGC pueda mostrarse en la pantalla que se desee.

- **SGC:**

Llamado Sistema de Gestión de Clientes, es un sistema externo el cual se encarga de proporcionar la data a mostrar en la aplicación mediante peticiones a su servidor. Además de soportar las operaciones que se deseen realizar en la aplicación.

- **Support AppCompact versión 7:**

Biblioteca Android importada en el Gradle para poder utilizar determinados componentes para versiones antiguas, su propio nombre lo dice: Biblioteca de soporte y compatibilidad versión 7. Mediante esta biblioteca podemos utilizar la compatibilidad de versiones menores a la 4.0 los siguientes componentes: ActionBar, AppCompatActivity, AppCompatActivity y ShareActionProvider.



- **Support App Compact versión 4:**

Biblioteca Android importada en el Gradle para poder utilizar los componentes a partir de la versión 1.6 de Android hacia adelante. Mediante esta biblioteca podemos utilizar uno de los componentes más utilizados: Los Fragments.

- **Support Multidex:**

Biblioteca importada en el gradle la cual se utiliza para indicar al apk que se utilizarán métodos de más de 65 000 líneas de código. Sin esta biblioteca, el apk no reconocerá más de 65 000 líneas de código y no podrá ser ejecutado.

- **Picasso:**

Biblioteca importada en el gradle la cual permite cargar imágenes en los componentes necesarios y poder aplicarles las transformaciones que se desee. Esta biblioteca soluciona los problemas de errores de tamaño de imágenes.

- **Butterknife:**

Biblioteca importada en el gradle la cual permite evitar escribir muchas líneas de código para instanciar los componentes y soluciona el problema realizando inyección de vistas. Es una librería muy utilizada para instanciar componentes de una manera clara y sencilla.

- **Support Design versión 23:**

Biblioteca Android importada en el gradle la cual permite manejar los problemas de diseño en android, implementando las nuevas características de Material Design y hacer que las apps se vean más elegantes y vistosas.

- **Gson v2.6.2:**

Biblioteca importada en el gradle la cual permite convertir una cadena de texto en formato JSON a un objeto que

mantenga la misma estructura de dicha cadena, de la misma manera, permite convertir un objeto a una cadena de texto. Esto permite una mayor facilidad en el manejo de la data en la aplicación.

- **Retrofit:**

Biblioteca importada en el gradle la cual es un cliente RESTFULL, esta biblioteca usa anotaciones para describir peticiones HTTP.

- **Okhttp3:**

Biblioteca importada en el gradle la cual se utiliza para realizar peticiones a los servicios web en formato RESTFULL. Esta biblioteca trabaja en conjunto con Retrofit.

- **RecyclerView:**

Biblioteca importada en el gradle la cual nos proporciona el componente llamado RecyclerView, que es una versión mejorada del componente ListView. El RecyclerView nos facilita la carga de data en la lista, recicla las vistas de manera eficiente sin necesidad de escribir un código muy complejo para dicho cometido y además la facilidad de implementar mejoras como es el caso del PullToRefresh.

- **CardView:**

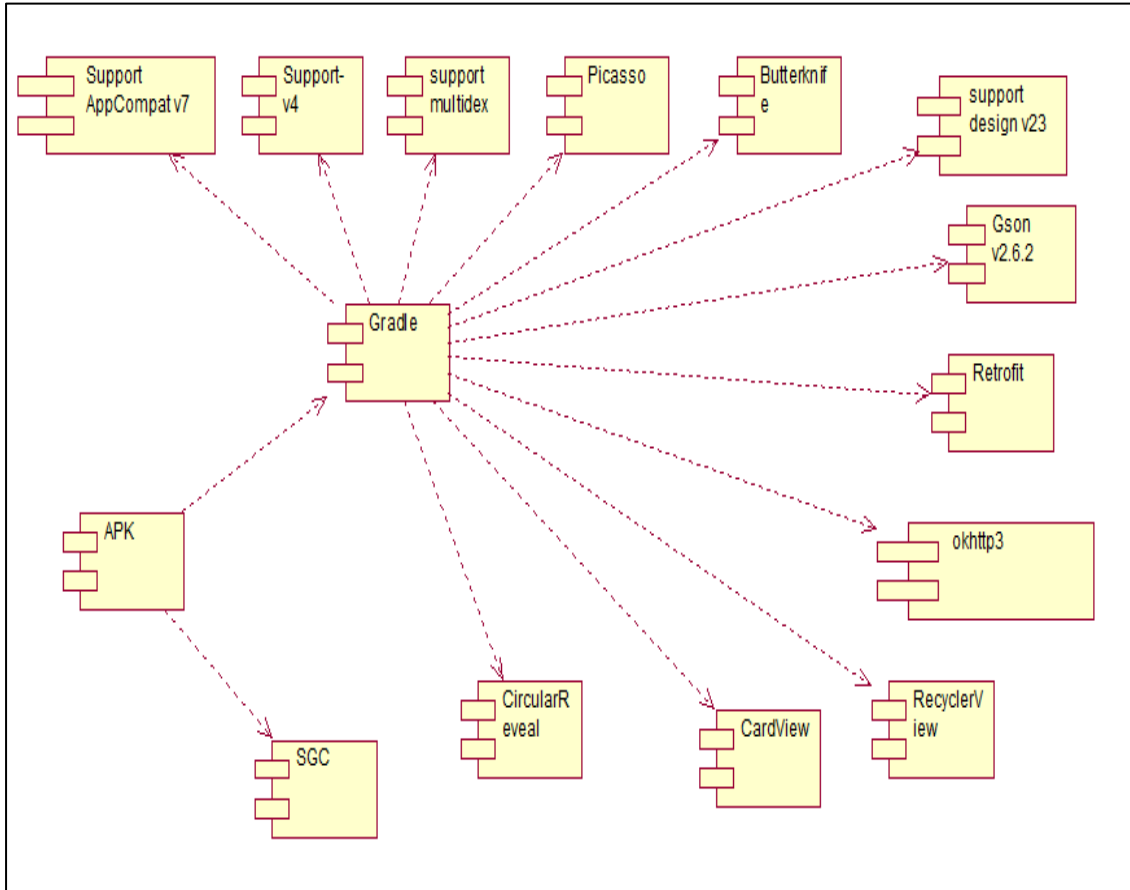
Biblioteca importada en el gradle la cual nos proporciona el componente llamado CardView. Este componente es una mejora a los view comunes que se utilizan en las listas. Da un aspecto de tarjeta a dichos View y crea un efecto más estético en la presentación de las listas de datos.

- **CircularReveal:**

Biblioteca importada en el gradle la cual nos da un efecto visual mucho más amigable el cual consiste en ocultar o

mostrar determinado elemento con un efecto de revelación u ocultación ayudado de un fondo negro de manera circular.

**Figura N° 4-8: Diagrama de Componentes**



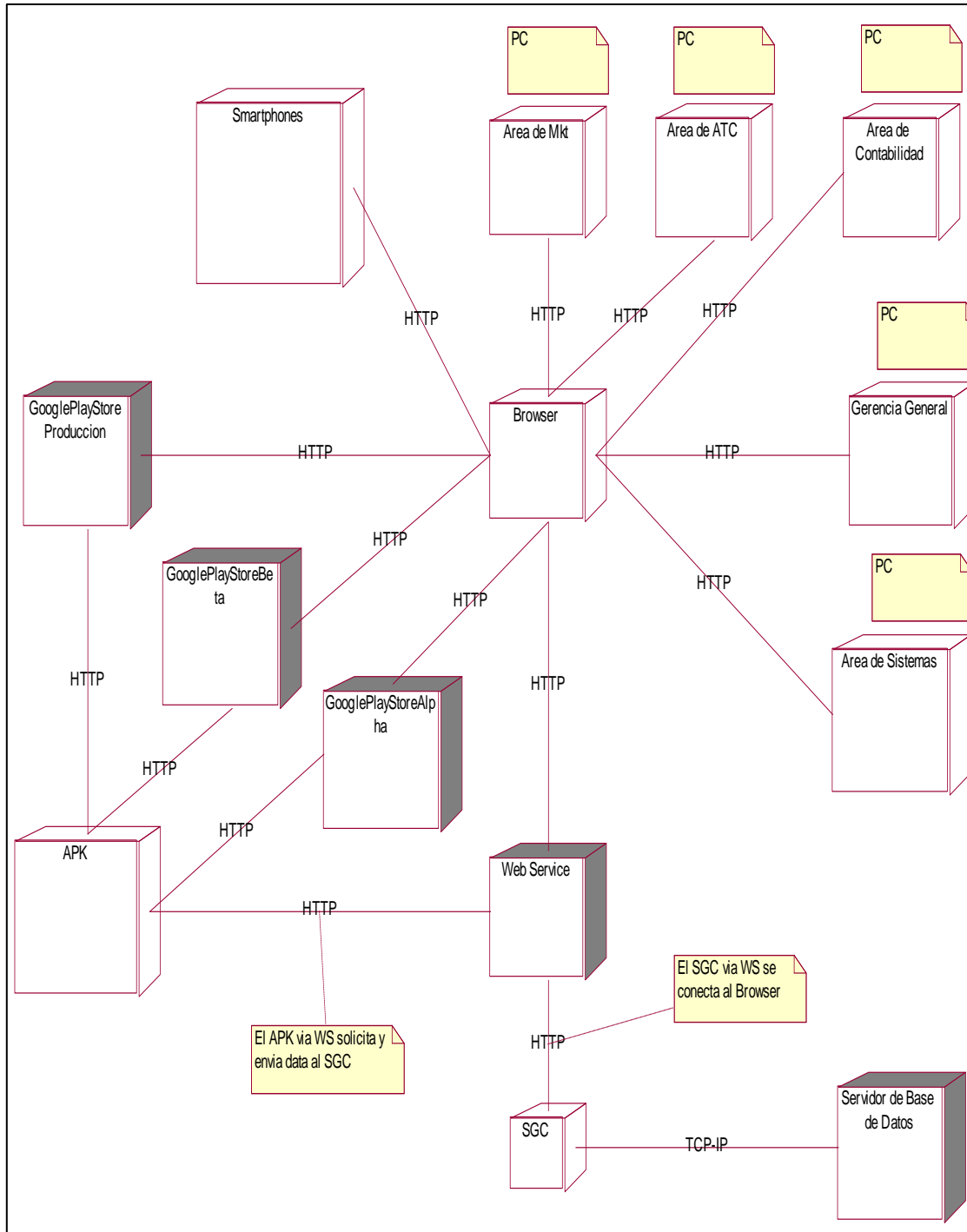
Fuente: Elaboración propia

#### 4.3.5. VISTA DE DESPLIEGUE

En el siguiente diagrama se puede apreciar como desplegar el sistema tanto en la web como en el Store, llámese Google Play Store. Esto con el fin de ser utilizado por cualquier persona. En el caso del aplicativo se puede descargar de una tienda dependiendo del lenguaje, en el caso de la web, cualquier persona puede ingresar a la URL y poder ver el contenido que se tiene (Ver Figura 4-9).

El diagrama describe la forma de cómo se desplegó el sistema utilizando HTTP y TCP-IP.

Figura N° 4-9: Diagrama de Despliegue



Fuente: Elaboración propia

## 4.4. DISEÑO DE LA SOLUCIÓN

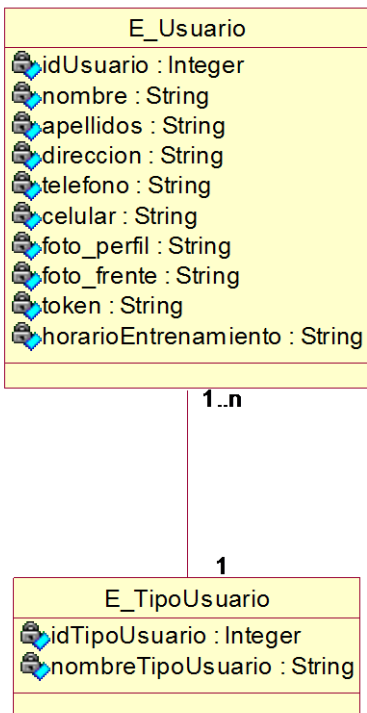
### 4.4.1. REALIZACIÓN DE CASOS DE USO

Historia de Usuario	
<b>Número: 1</b>	<b>Usuario:</b> Cliente
<b>Nombre historia:</b> Inicio de Sesión	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Objetivos:</b>  Poder iniciar sesión en el aplicativo sin ninguna dificultad.	
<b>Pre Condiciones:</b>  Al inscribirse en el gimnasio, se le brinda un código de usuario y una contraseña.	
<b>Descripción:</b>  El cliente ingresa el código recibido vía e-mail en el campo "Ingresar Código" de la pantalla "Login". Luego selecciona el botón Ingresar. Si el código ingresado es correcto, se mostrará el Menú Principal. Si el código ingresado es incorrecto, le aparecerá el mensaje: "Ingresar código correcto".	
<b>Observaciones:</b>  El código se envía al Cliente cuando es registrado en el sistema web del gimnasio.	
<b>Prototipos:</b>	



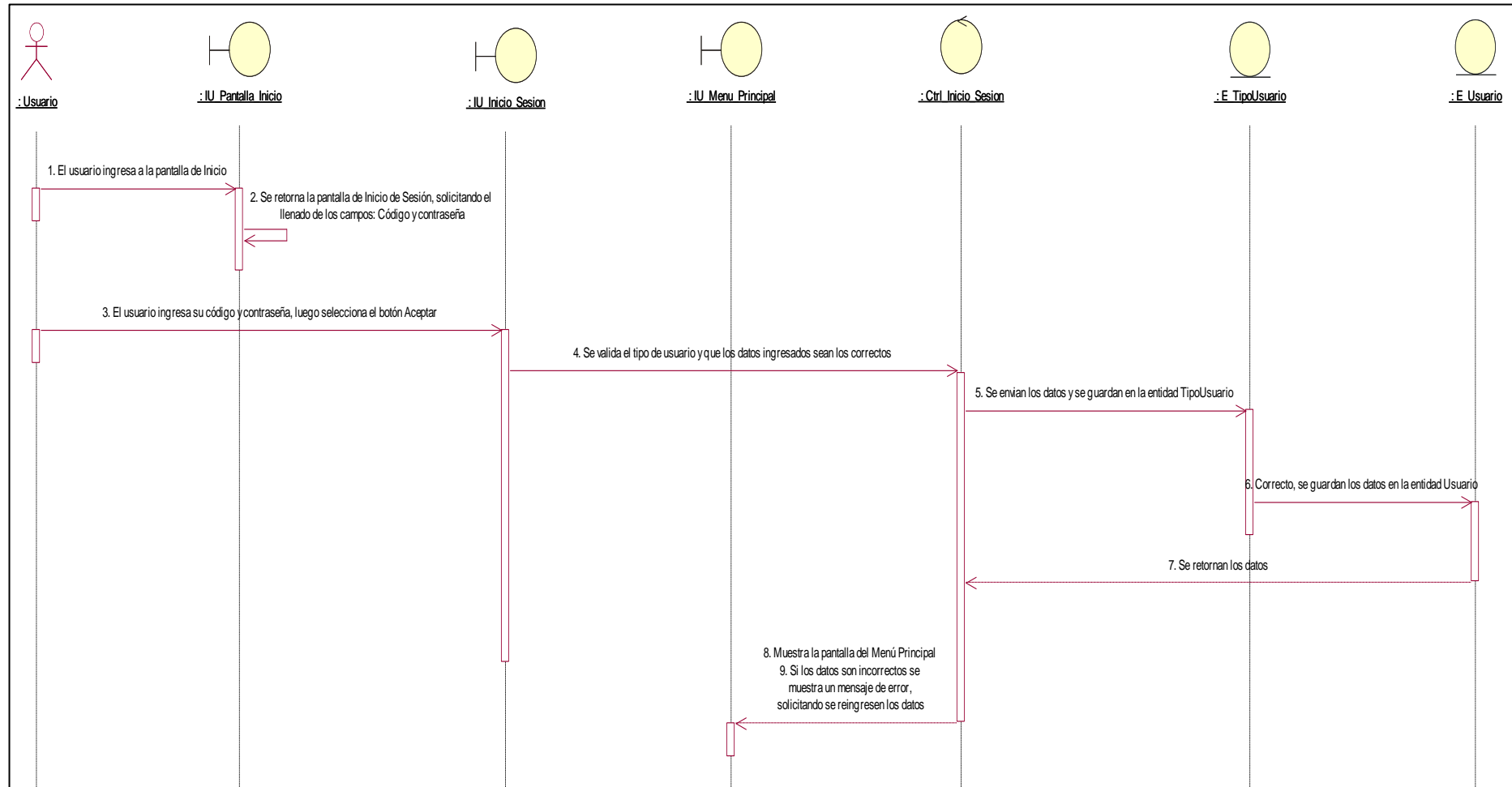
**Diagrama de Clases:**

Se utilizarán las siguientes clases:



## Diagrama de Secuencia:

Se muestra diagrama de secuencia:

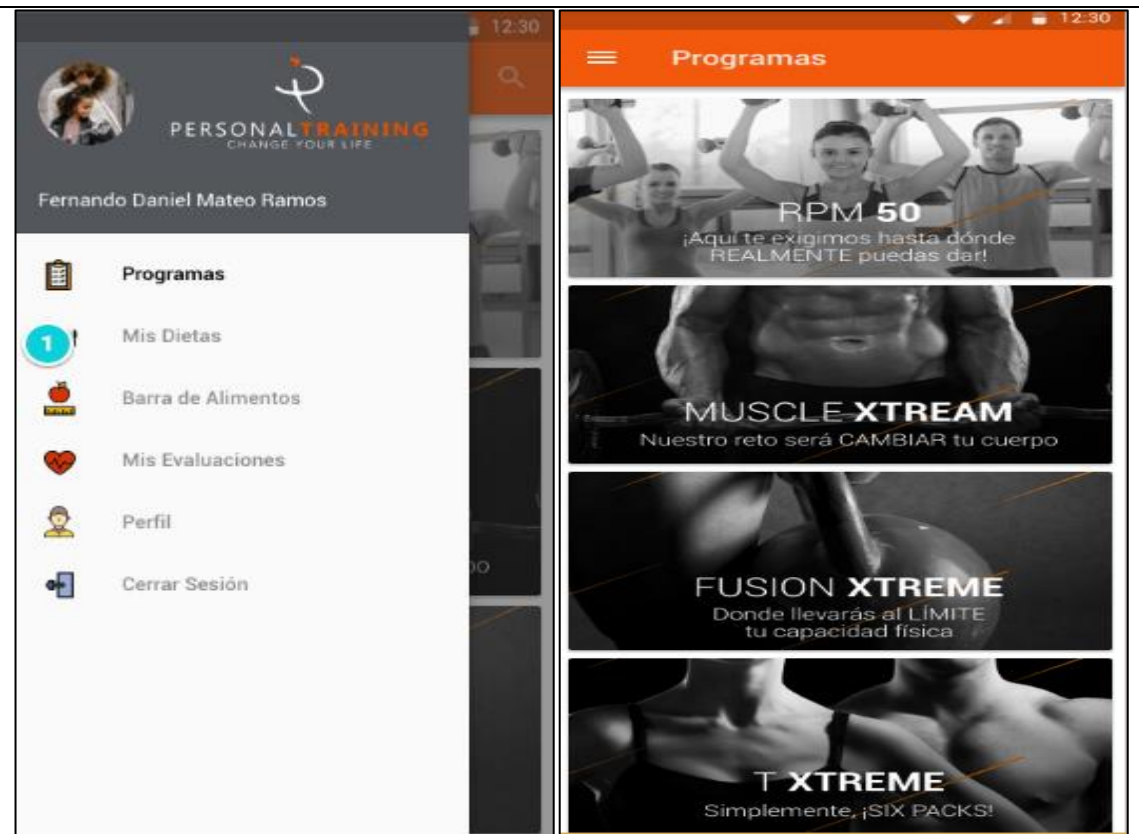


**Post Condiciones:**

En el caso que la contraseña sea la incorrecta, el aplicativo le permite poder cambiar la contraseña, para lo cual debe seguir unos pasos simples que se le enviará a su e-mail.

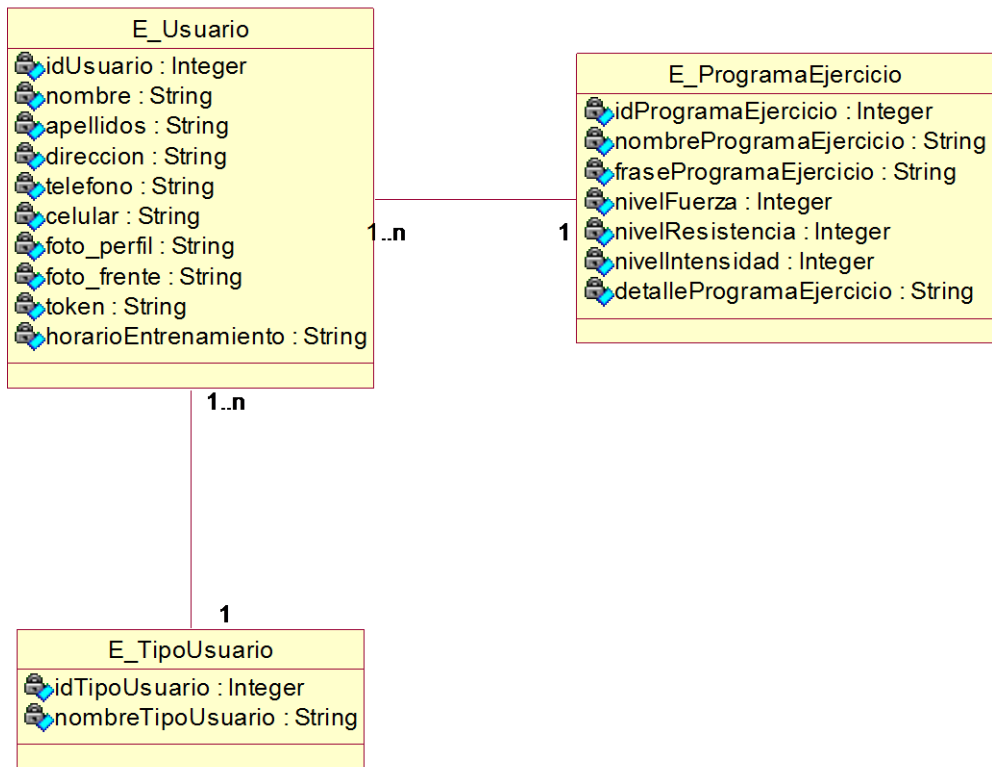


<b>Historia de Usuario</b>	
<b>Número: 2</b>	<b>Usuario:</b> Cliente
<b>Nombre historia:</b> Consultar Programas Gimnasio	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Objetivos:</b>  Consultar cada descripción de los programas que brinda el gimnasio, para poder orientarse mejor y decidir de forma correcta a cual inscribirse.	
<b>Pre Condiciones:</b>  Haber iniciado sesión correctamente.	
<b>Descripción:</b>  El cliente selecciona la opción "Programas" del Menú Principal. Se mostrará todos los programas vigentes que tiene el gimnasio: <ul style="list-style-type: none"> <li>- RPM 50</li> <li>- MUSCLE XTREAM</li> <li>- FUSION XTREME</li> <li>- T-XTREME</li> </ul> Al seleccionar cada programa, se mostrará el detalle completo del mismo.	
<b>Observaciones:</b>  Ninguna.	
<b>Prototipos:</b>	



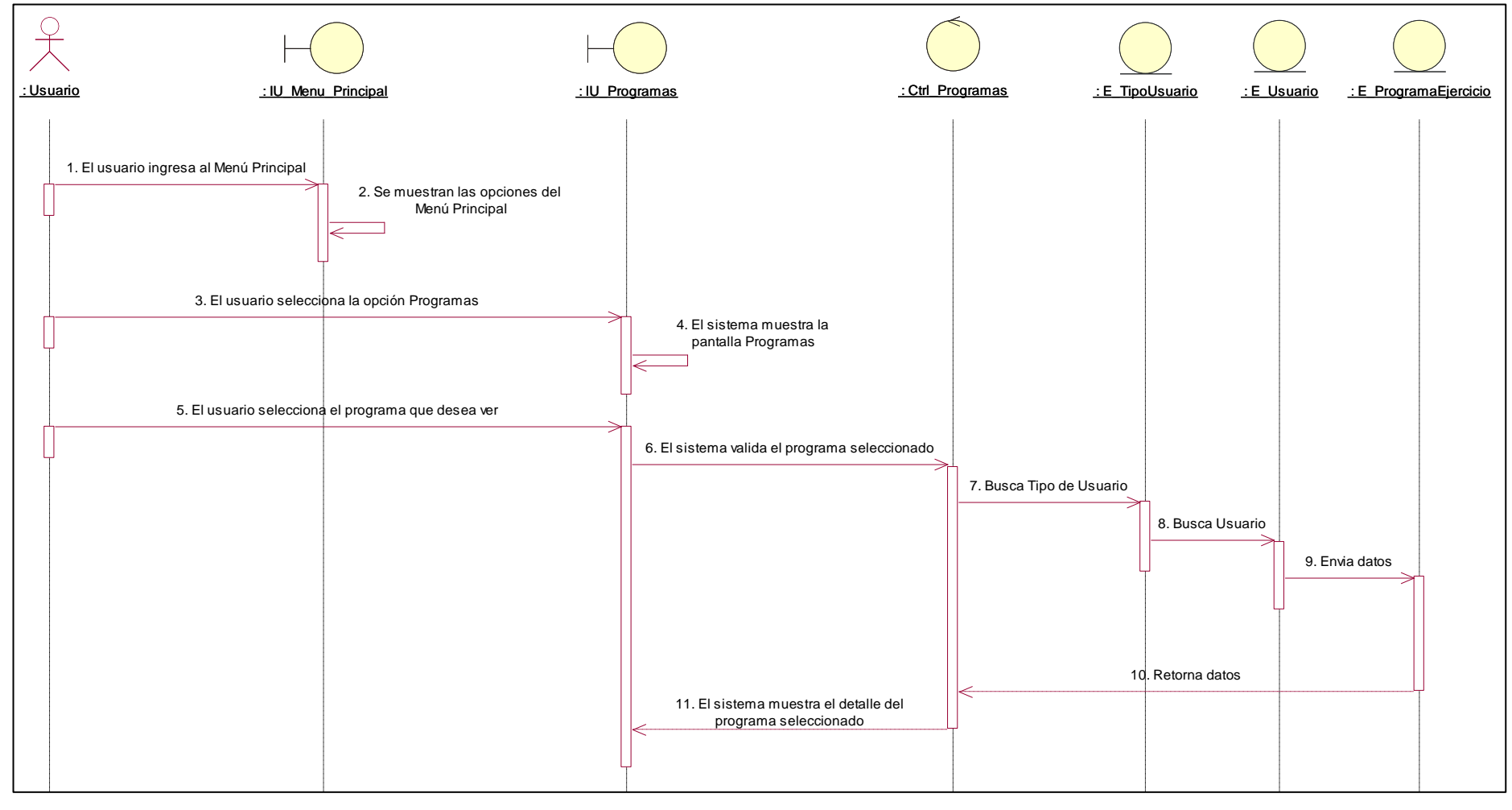
## Diagrama de Clases:

Se utilizarán las siguientes clases:



## Diagrama de Secuencia:

Se muestra diagrama de secuencia:



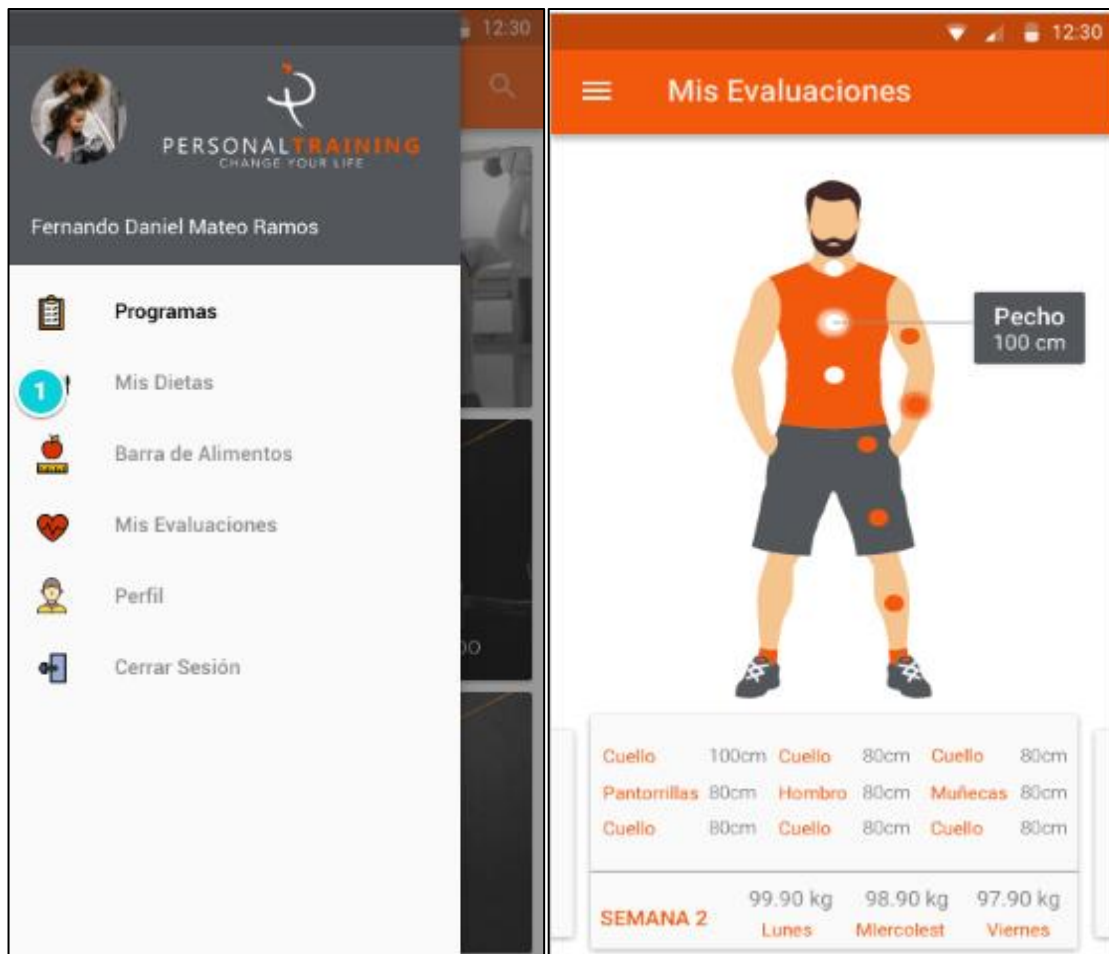
**Post Condiciones:**

Ninguna.

Historia de Usuario	
<b>Número: 3</b>	<b>Usuario:</b> Cliente
<b>Nombre historia:</b> Consultar Evaluaciones	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Objetivos:</b>  Consultar el progreso que tiene un usuario semanalmente, esto con el fin de ver si su esfuerzo y constancia está dando frutos.	
<b>Pre Condiciones:</b>  Haber iniciado sesión correctamente. Haber asistido a los controles semanales en el gimnasio.	
<b>Descripción:</b>  El cliente selecciona la opción "Progreso" del Menú Principal. Se mostrará la pantalla "Progreso", que contiene un prototipo de la persona con puntos blancos y naranjas en ciertas partes del cuerpo como por ejemplo: Pecho, cuello, pantorrillas, hombro, muñecas, muslo, etc. Al momento de seleccionar cada punto, podrá consultar la medida que tiene actualmente, y en la parte inferior podrá apreciar el progreso que viene teniendo semana a semana.	
<b>Observaciones:</b>  En la parte inferior además se puede apreciar el progreso del peso	

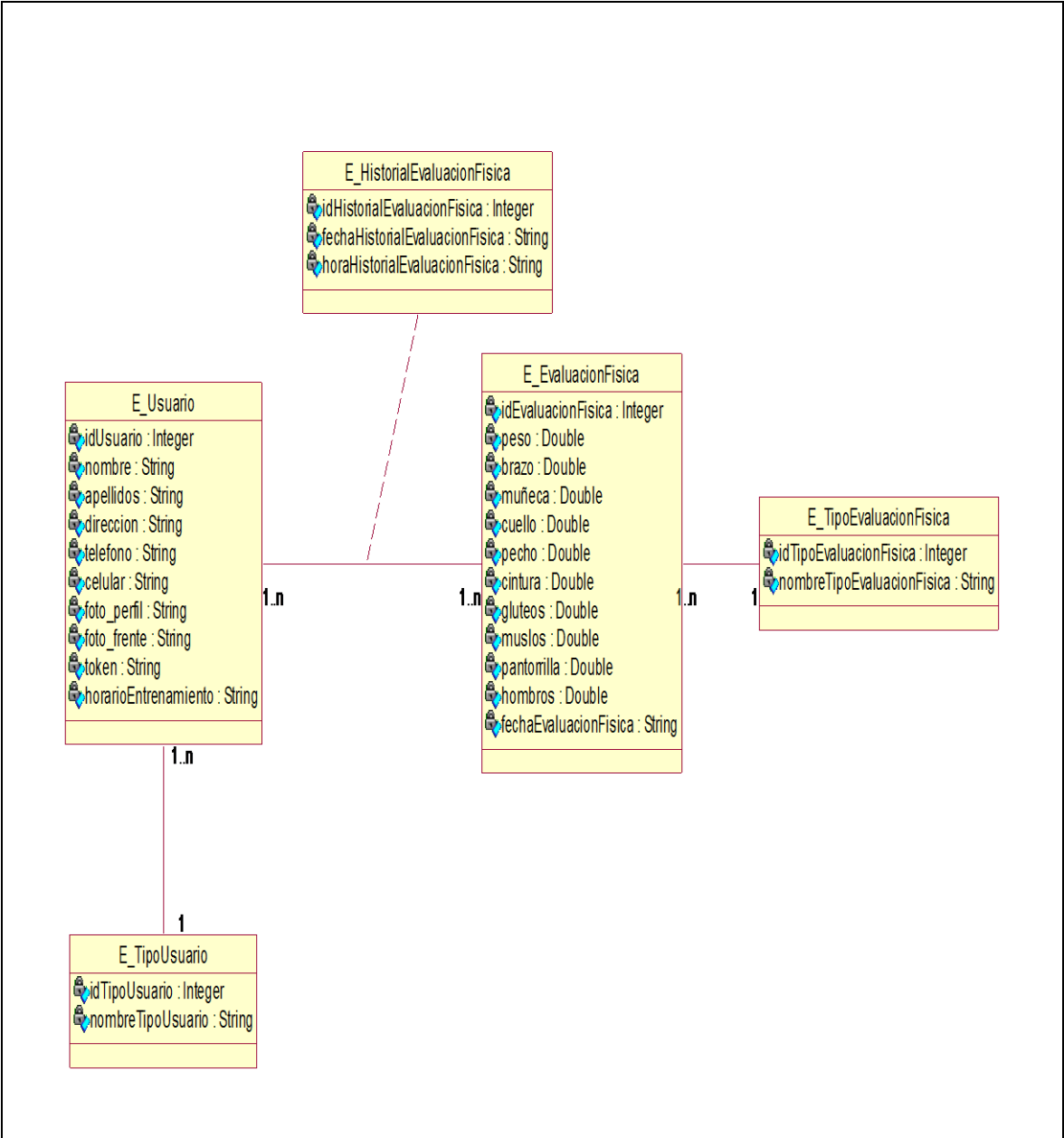
semanalmente.

### Prototipos:



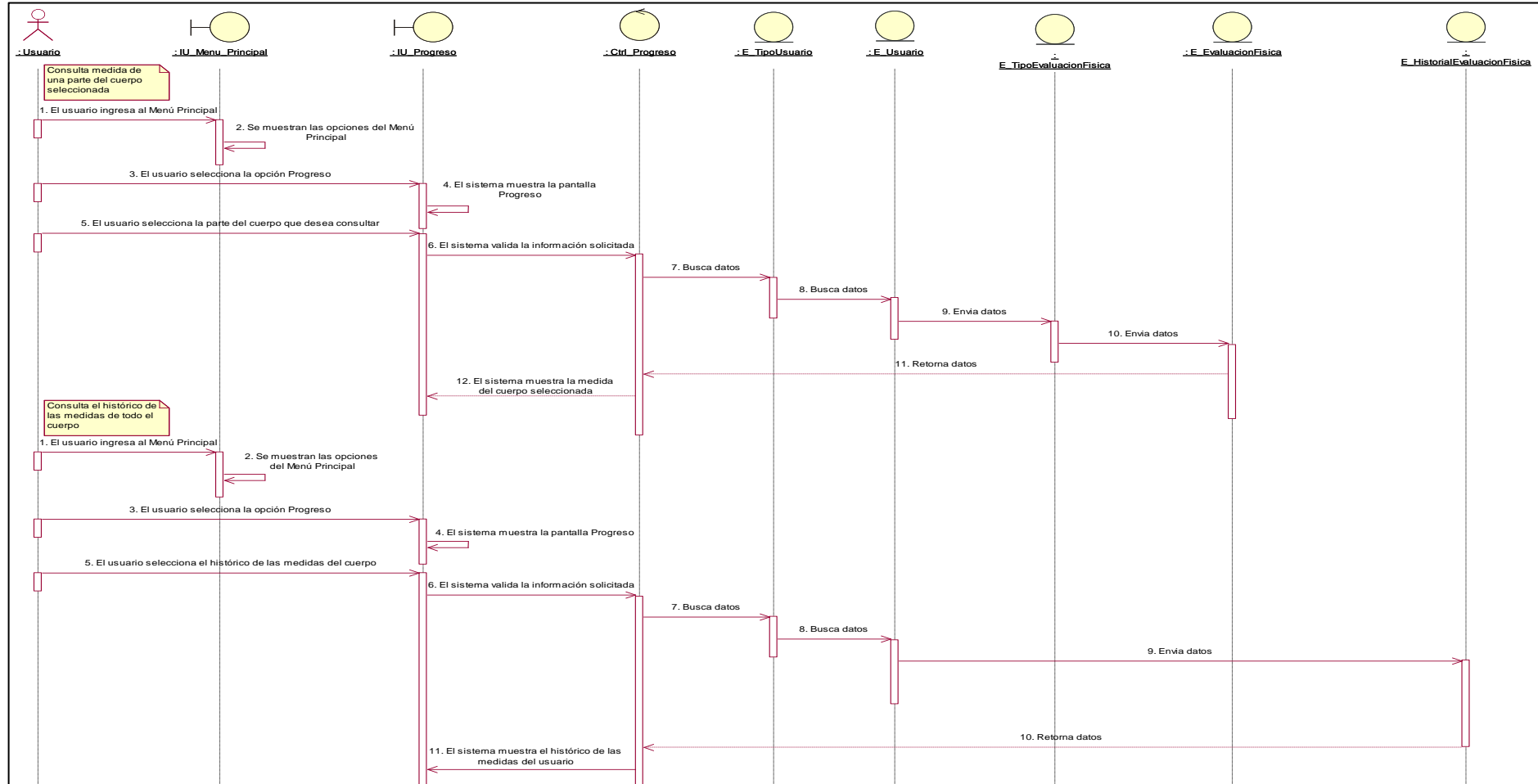
### Diagrama de Clases:

Se utilizarán las siguientes clases:



## Diagrama de Secuencia:

Se muestra el diagrama de secuencia de la consulta de medida por alguna parte del cuerpo seleccionada:





**Post Condiciones:**

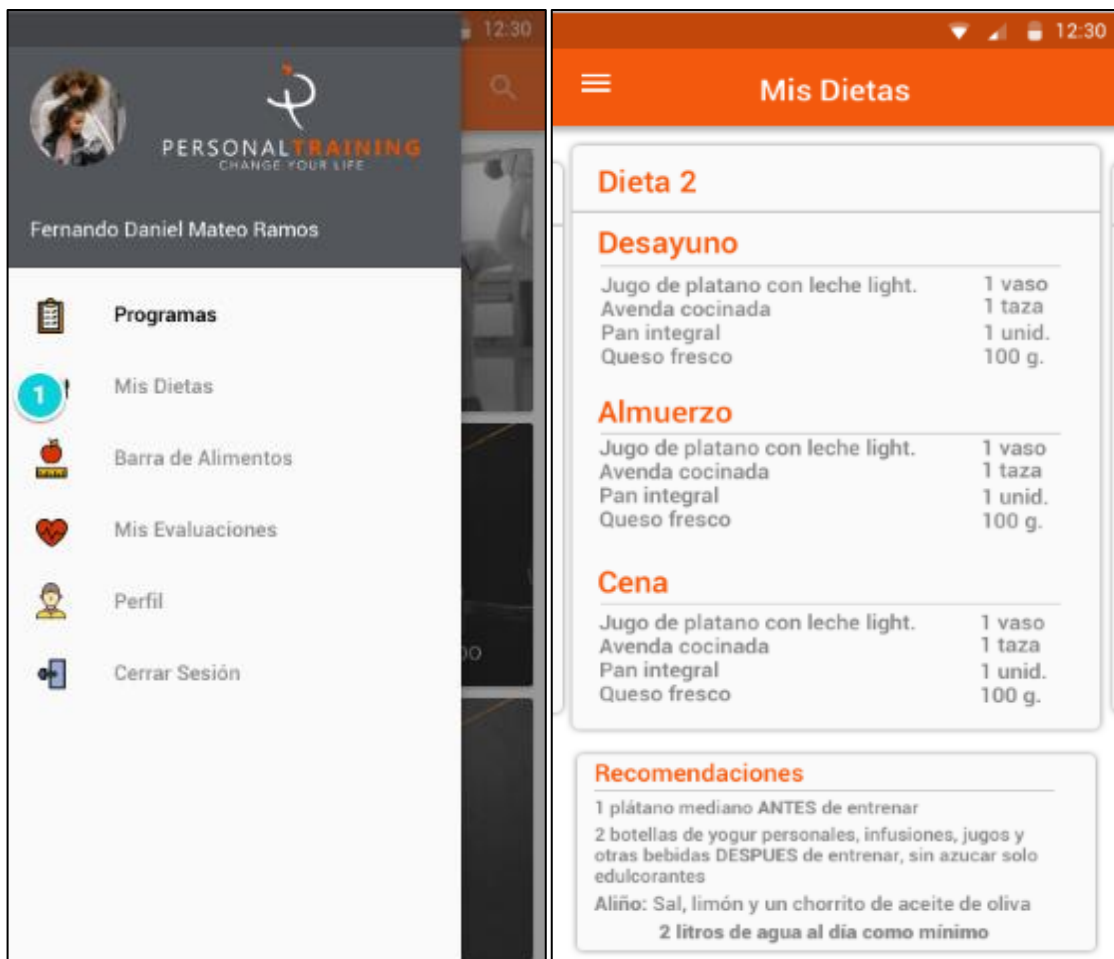
El usuario que no tenga información para mostrar, es debido a que no asistió a los controles semanales.

Historia de Usuario	
<b>Número: 4</b>	<b>Usuario:</b> Cliente
<b>Nombre historia:</b> Consultar Mis Dietas	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Objetivos:</b>  Consultar las dietas recetadas por el nutricionista, según previa cita realizada. Se mostrará 5 dietas equivalentes para que el usuario pueda escoger la que mejor le parezca.	
<b>Pre Condiciones:</b>  Haber iniciado sesión correctamente.  Haber asistido a la cita con el nutricionista.	
<b>Descripción:</b>  El cliente selecciona la opción “Mis Dietas” del Menú Principal.  Se mostrará la pantalla “Mis Dietas”, en la parte superior de la pantalla se mostrará las dietas por las que el cliente puede optar. Cada dieta se divide en: desayuno, almuerzo y cena. El detalle de cada dieta está bien detallada con sus medidas respectivas.	

## Observaciones:

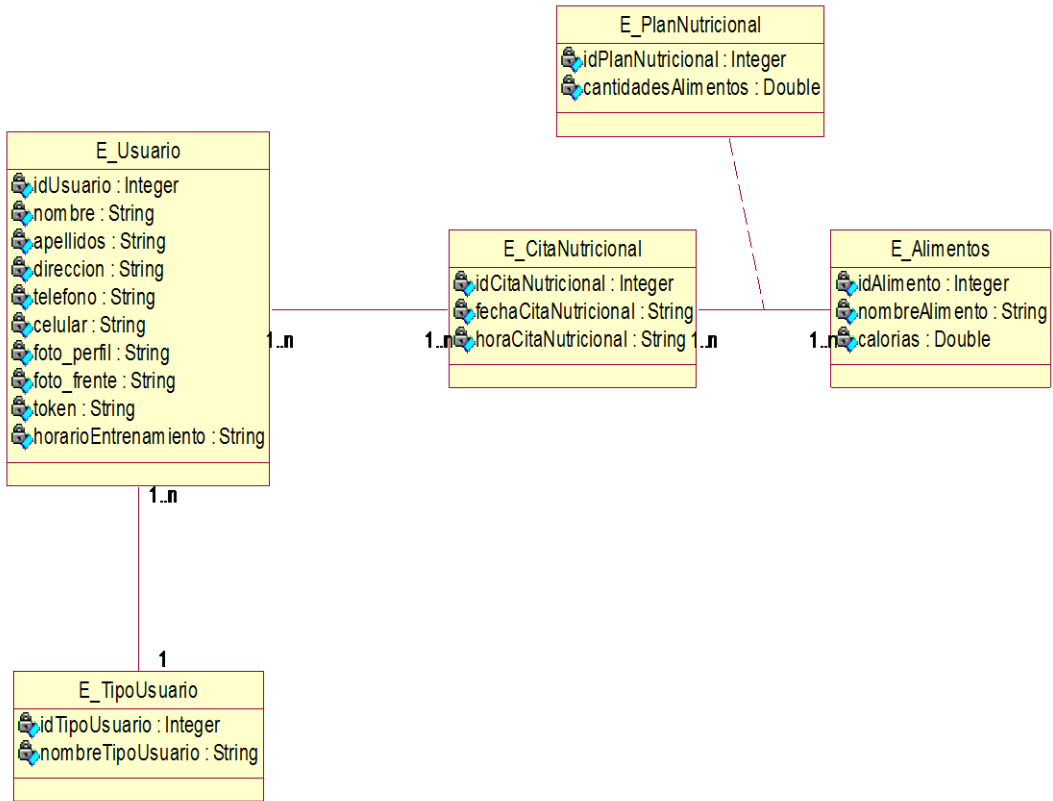
En la parte inferior además se puede apreciar las Recomendaciones Generales que el cliente debe tomar para cada día.

## Prototipos:



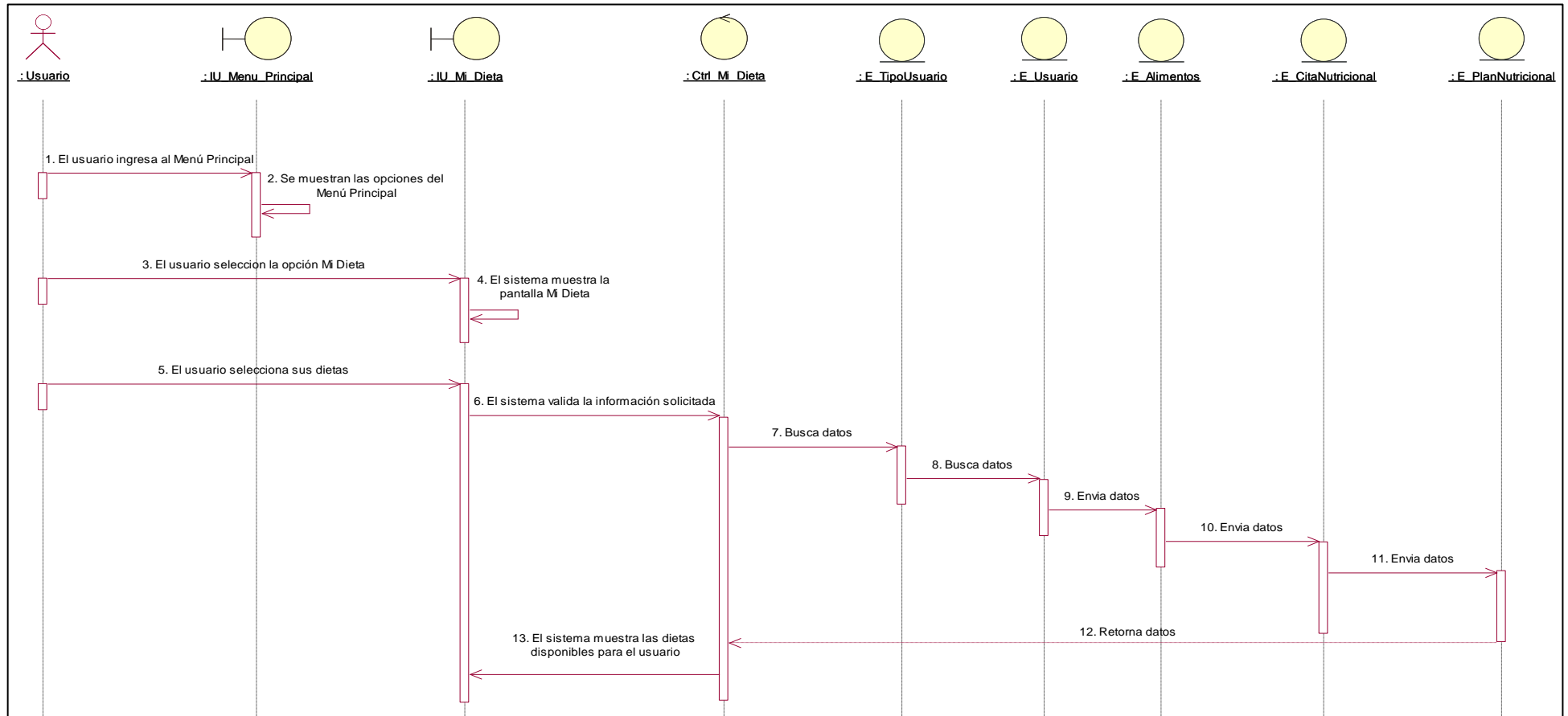
## Diagrama de Clases:

Se utilizarán las siguientes clases:



## Diagrama de Secuencia:

Se muestra diagrama de secuencia:



**Post Condiciones:**

El usuario que no tenga información para mostrar, es debido a que no asistió a la cita con el nutricionista.

**Historia de Usuario****Número: 5****Usuario:** Cliente**Nombre historia:** Consultar Calendario**Prioridad en negocio:**

Alta

**Riesgo en desarrollo:**

Alta

**Objetivos:**

Poder congelar los días que el usuario no pueda asistir al gimnasio por diferentes motivos, esto con el fin de poder recuperarlos en otra fecha y que no sean descontados.

**Pre Condiciones:**

Haber iniciado sesión correctamente.

El usuario tendrá disponibles una cantidad de días de congelamiento, dependiendo de las semanas inscritas en el gimnasio. Por ejemplo si tiene un contrato por 1 mes, no contará con días de congelamiento. Si tiene un contrato de 2 meses, contará con 5 días de congelamiento.

**Descripción:**

El cliente selecciona la opción "Calendario" del Menú Principal.

Se mostrará la pantalla "Calendario", en donde debe seleccionar la opción "+".

Se mostrará un dialogo, en donde el cliente debe elegir la fecha de congelamiento.

El cliente selecciona el botón “Aceptar”.

El cliente deberá confirmar con el gimnasio el motivo de su elección del día de congelamiento.

Al confirmarse el motivo, el cliente visualizará el día de congelamiento elegido, el cual tendrá como icono un copo de nieve debajo del día elegido.

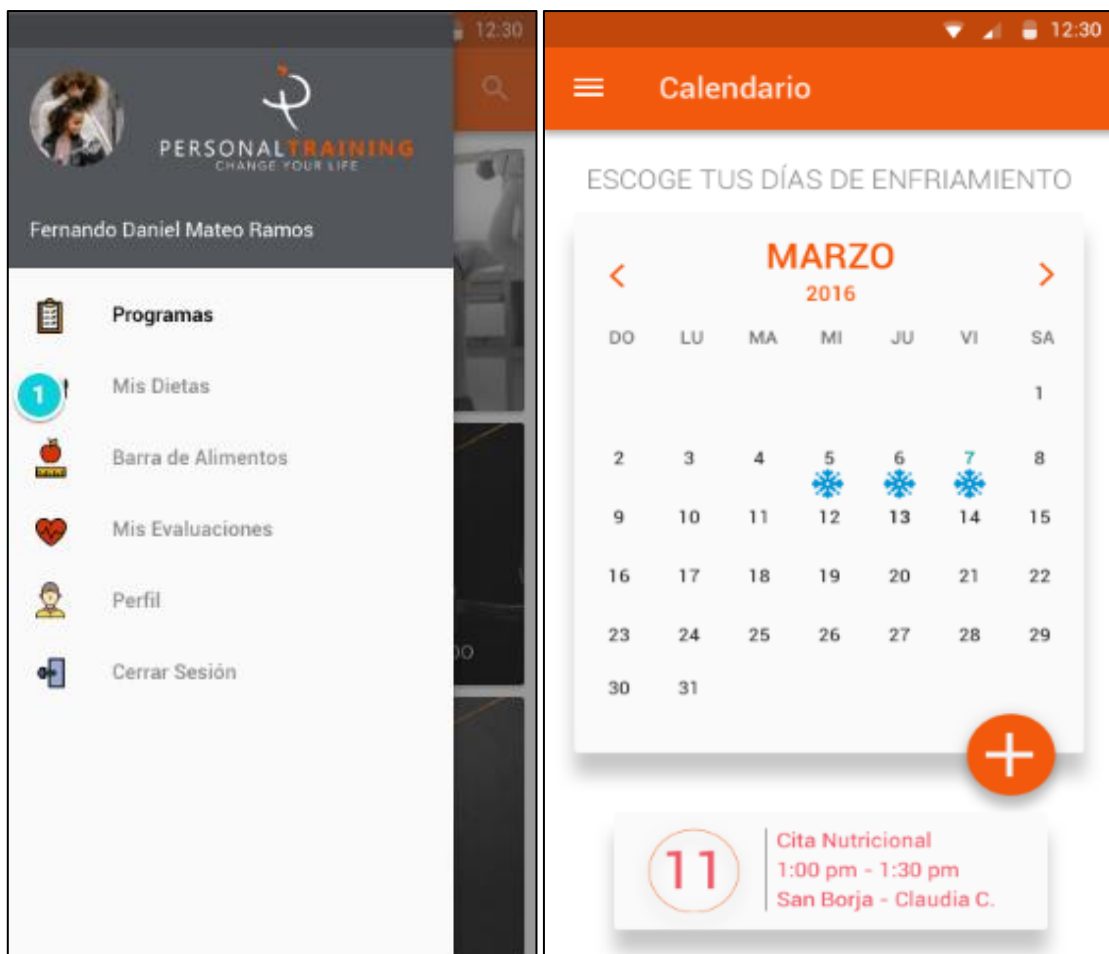
El cliente selecciona un día del calendario.

Se mostrará el detalle de ese día, si no hubiera información se mostrará el mensaje: “No hay información disponible”.

### Observaciones:

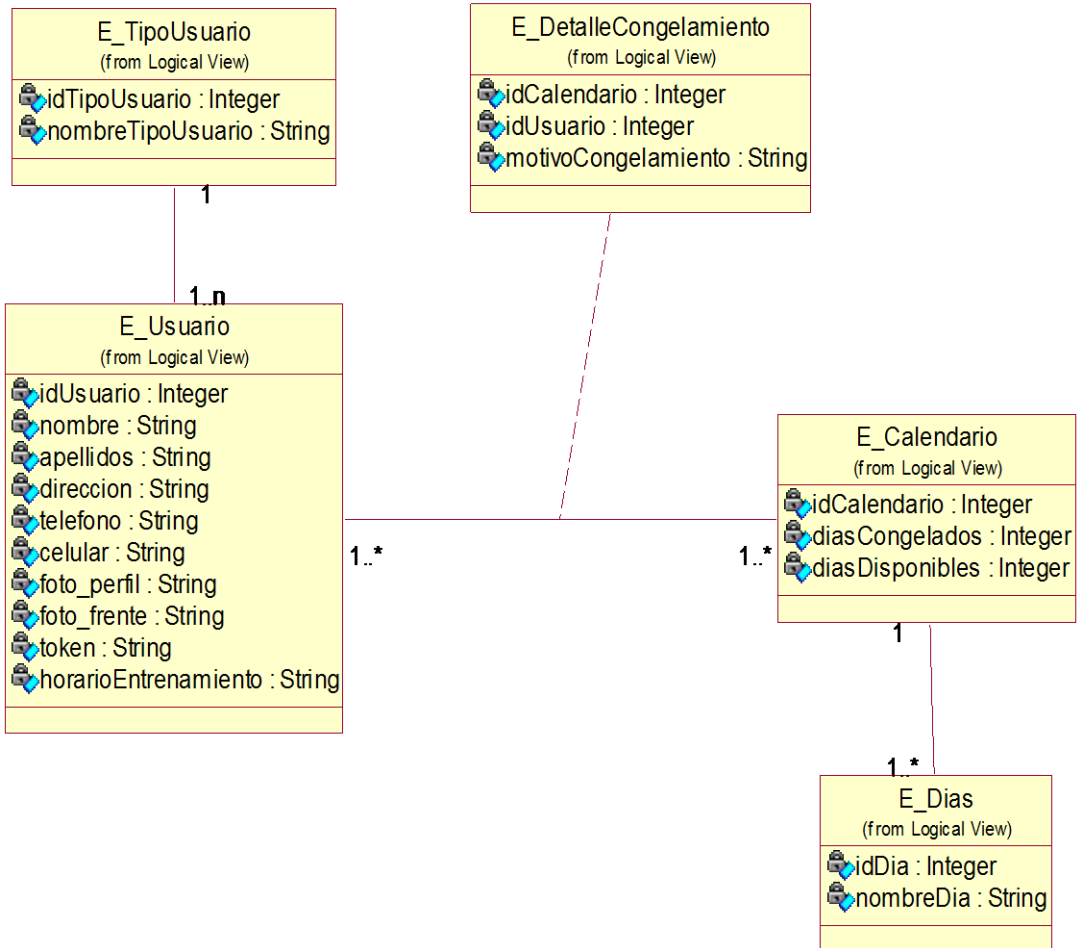
Ninguna.

### Prototipos:



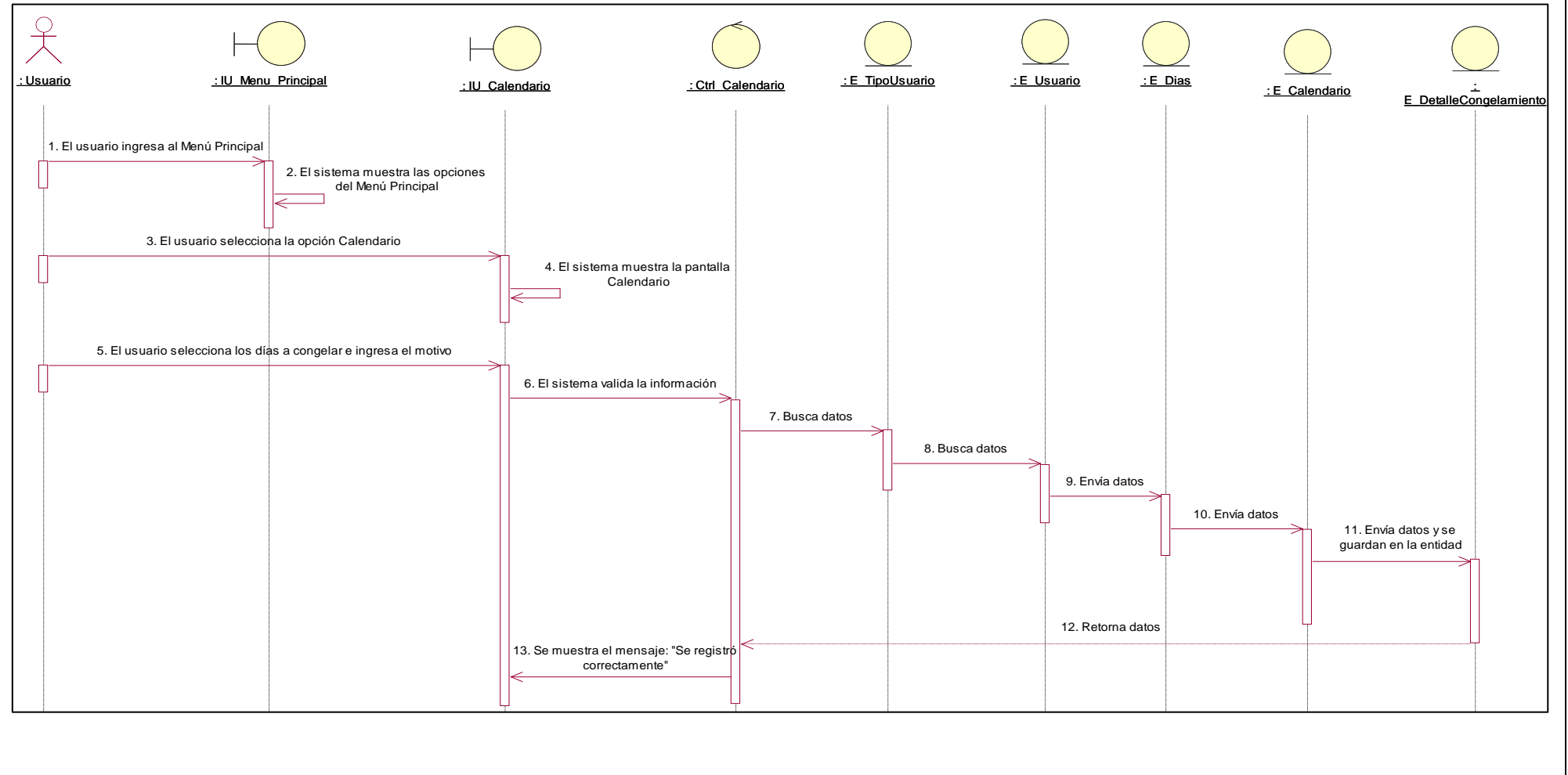
## Diagrama de Clases:

Se utilizarán las siguientes clases:



## Diagrama de Secuencia:

Se muestra diagrama de secuencia:





### Post Condiciones:

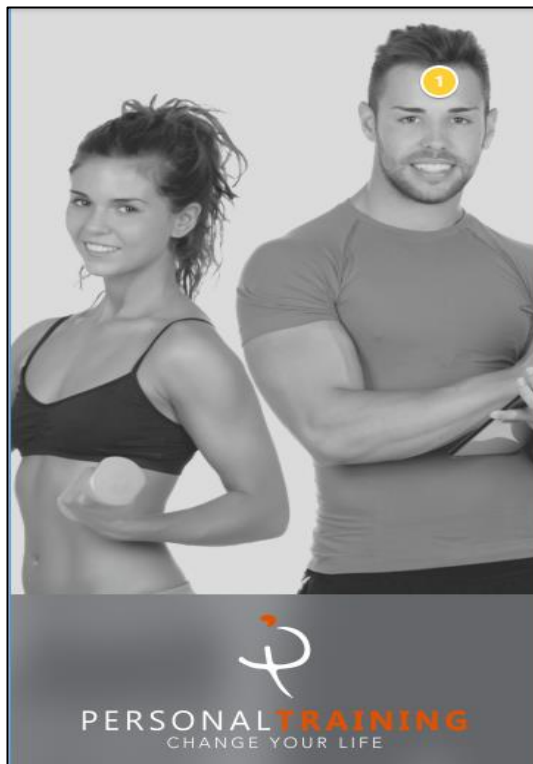
El usuario no podrá exceder el número máximo de días de congelamiento asignados.

## 4.4.2. PROTOTIPOS

Se tiene los siguientes diseños del sistema:

- **Pantalla principal del sistema:** en la Figura 4-10 se aprecia la pantalla de Bienvenida de la aplicación.

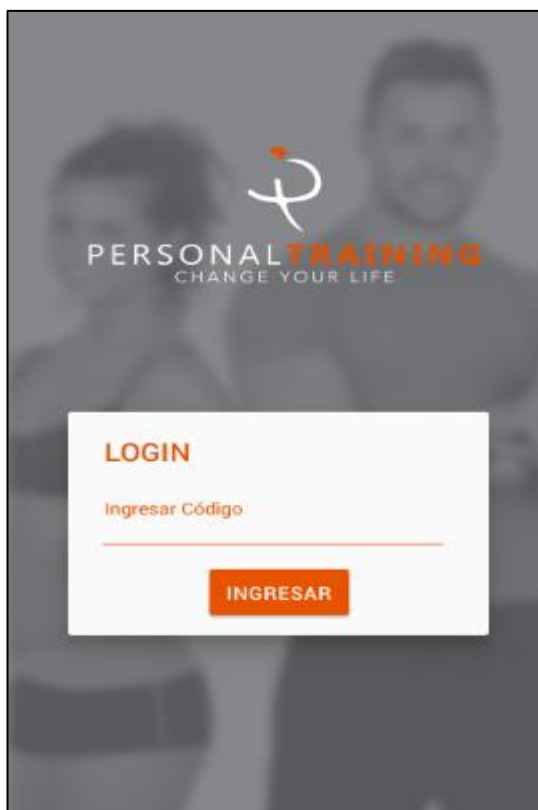
**Figura N° 4-10: Pantalla de Bienvenida**



Fuente: Elaboración propia

- **Pantalla de Login:** en la Figura 4-11 se observa el Login del aplicativo, el cual solicita que se ingrese el código y la contraseña.

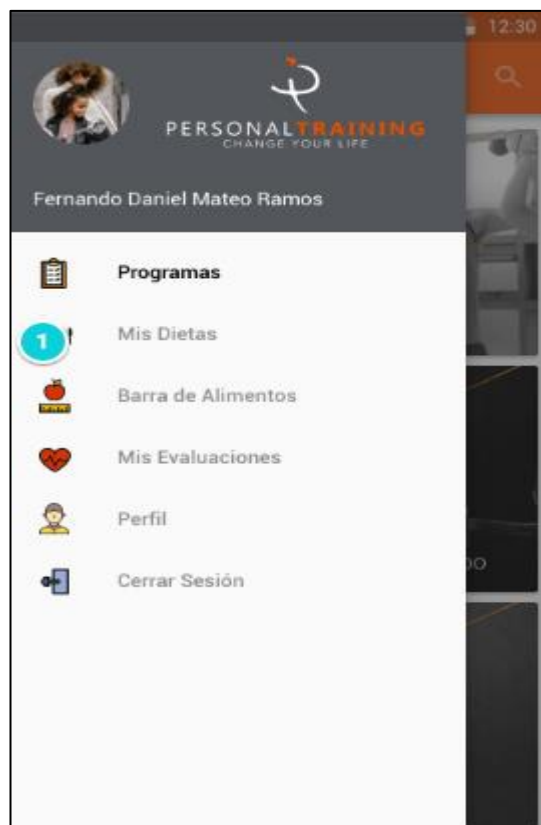
**Figura N° 4-11: Pantalla de Login**



Fuente: Elaboración propia

- Menú principal del sistema: en la Figura 4-12 se observa el Menú de Opciones que tiene el aplicativo.

**Figura N° 4-12: Pantalla de Menú de Opciones**



Fuente: Elaboración propia

- **Opción de Consultas de Programas:** en la Figura 4-13 se observa la Pantalla de la opción Programas, la cual nos muestra un resumen de todos los programas que están disponibles en el gimnasio.

**Figura N° 4-13: Pantalla de Consulta Programas del Gimnasio**



Fuente: Elaboración propia

- **Detalle de Programa:** en la Figura 4-14 se observa el detalle de cada Programa consultado, se tiene una descripción y los niveles de fuerza, resistencia e intensidad.

**Figura N° 4-14: Pantalla del detalle de un programa**



Fuente: Elaboración propia

- **Opción de Mis Dietas:** en la Figura 4-15 se observa las dietas disponibles para cada usuario, las mismas que fueron generados por el Nutricionista, se le brinda un máximo de 5 dietas para que pueda alternarlas, además se muestran algunas recomendaciones en la parte inferior.

**Figura N° 4-15: Pantalla de Mis Dietas**



Fuente: Elaboración propia

- **Mi Progreso:** en la Figura 4-16 se observa el progreso de cada usuario en cuanto a sus medidas y pesos, al seleccionar una parte del cuerpo se podrá consultar la medida que tiene, además en la parte inferior se puede observar un historial semanal de las medidas.

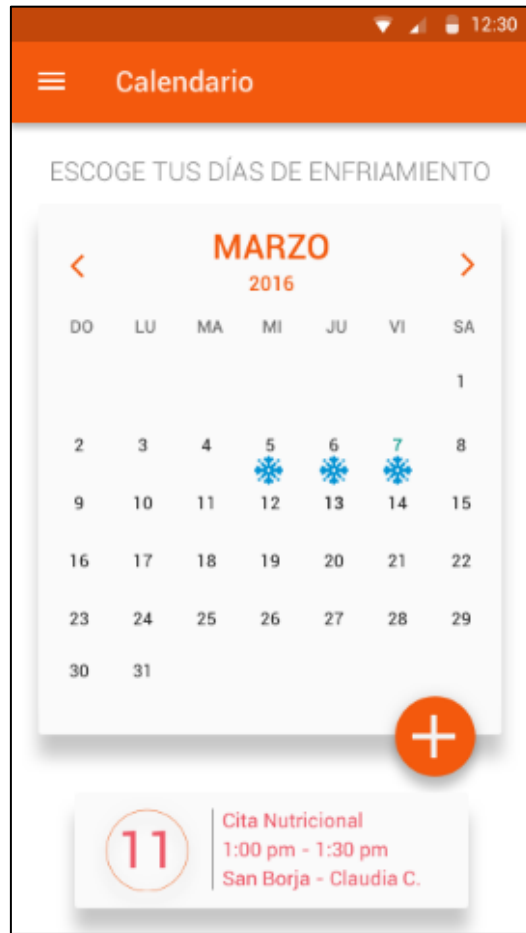
**Figura N° 4-16: Pantalla de Mi Progreso**



Fuente: Elaboración propia

- **Consulta Calendario:** en la Figura 4-17 se observa el Calendario, en el cual se puede seleccionar los días en que un usuario no pueda asistir, con el fin de poder recuperarlos en otro momento.

**Figura N° 4-17: Pantalla de Consulta Calendario**



Fuente: Elaboración propia



## **CAPÍTULO V: RESULTADOS**

Luego de lograr la creación del sistema y su implementación se ha llegado a los siguientes resultados:

- Se logró la implementación de un sistema móvil en android mediante la utilización de la arquitectura limpia.
- Al tener a disposición el código fuente de las capas de la arquitectura limpia, nos permite hacer cualquier modificación sin tener problemas en cualquiera de las otras capas.
- Se comprobó que el sistema al ser muy intuitiva, es de fácil uso y es bastante amigable para el usuario.
- Puesto que el sistema es implementado para el gimnasio Personal Training, un trabajo a futuro es desarrollar el sistema con un grado mayor de personalización y flexibilidad para quien desee hacer uso del mismo, brindado la posibilidad de no limitarlo a una institución; sino, que pueda ser configurado para un entorno externo.

## **CONCLUSIONES Y RECOMENDACIONES**

A partir de la presente investigación, se ha descrito a detalle todo el proceso de desarrollo de software de la presente solución, que en comparación con otras aplicaciones exitosas posee la mayoría de funcionalidades básicas. Todo esto no es más que el resultado de un correcto proceso de desarrollo, por lo cual se recomienda realizar cada una de las partes del proceso de desarrollo de software a fin de garantizar la obtención de un producto de calidad, que solucione el problema del cliente y que supere las expectativas del usuario.

Se ha demostrado que al crear el sistema y asociarla con una nueva tecnología (android) y arquitectura limpia, permitió dar un valor agregado y de mucha importancia para el proyecto. Por ello, se recomienda tomar en cuenta otras posibles utilidades de este producto con el fin de poder aprovechar su valor al máximo.

Del mismo modo, se ha dado a conocer cómo se implementa una solución móvil en android y una arquitectura limpia, esto podrá servir como base para posibles nuevas soluciones a futuro. Por ello, se recomienda tomar en cuenta el diseño del presente proyecto como fundamento para el desarrollo de soluciones más complejas y con un mayor alcance, siendo esto importante ya que el diseño representa la estructura que da sustento al proyecto.

## REFERENCIAS BIBLIOGRÁFICAS

Carol Hodges (1994). Culinary Nutrition for Food Professionals. USA: Great Pacific Books.

Judith Brown (1999). Nutrition Now. USA: West/Wadsworth.

Pablo Carcelén Amores (2014). Clean Architecture. España: IOS-BQ.  
[http://oa.upm.es/40013/6/TFG\\_Pablo\\_Carcelen\\_Amores.pdf](http://oa.upm.es/40013/6/TFG_Pablo_Carcelen_Amores.pdf)

Wardlaw y Kessel (2002). Perspectives in Nutrition. USA: McGraw-Hill.

Alicia Bernard (2005). Los Secretos de la dieta proteica. España: Prumest.

Onostre Ayala Ismael (2013), *Sistema de Información orientado a brindar asistencia de nutrición y salud, basado en tecnología móvil* (Tesis de Pregrado en la carrera de Ingeniería de Sistemas), Universidad Católica Boliviana, Bolivia.

Recuperado de:

<http://www.bibvirtual.ucb.edu.bo/opac/Record/101001323>

Flor, Granda, & Revelo (2005). Rendimiento Físico. España: Fitness.

Biblioteca Virtual, Universidad Católica Boliviana, Bolivia, 2012

<http://www.bibvirtual.ucb.edu.bo/opac/Record/101001323>

Diario El Comercio, Lima, Perú, 03.2014

<http://elcomercio.pe/lima/ciudad/tres-cada-cinco-peruanos-tienen-sobrepeso-obesidad-noticia-1731163>

Diario El Comercio, Lima, Perú, 03.2014

<http://elcomercio.pe/economia/peru/solo-dos-cada-100-peruanos-van-al-gimnasio-noticia-671369>

Bernard Mayta, Project Charter, Apuntes de Análisis de Costes.

Universidad Peruana Union.2007.

[http://www.doccity.com/es/projectcharter\\_ejemplo1/558468/](http://www.doccity.com/es/projectcharter_ejemplo1/558468/)

Página Web Oficial de Android:

<https://developer.android.com/about/android.html?hl=ES>

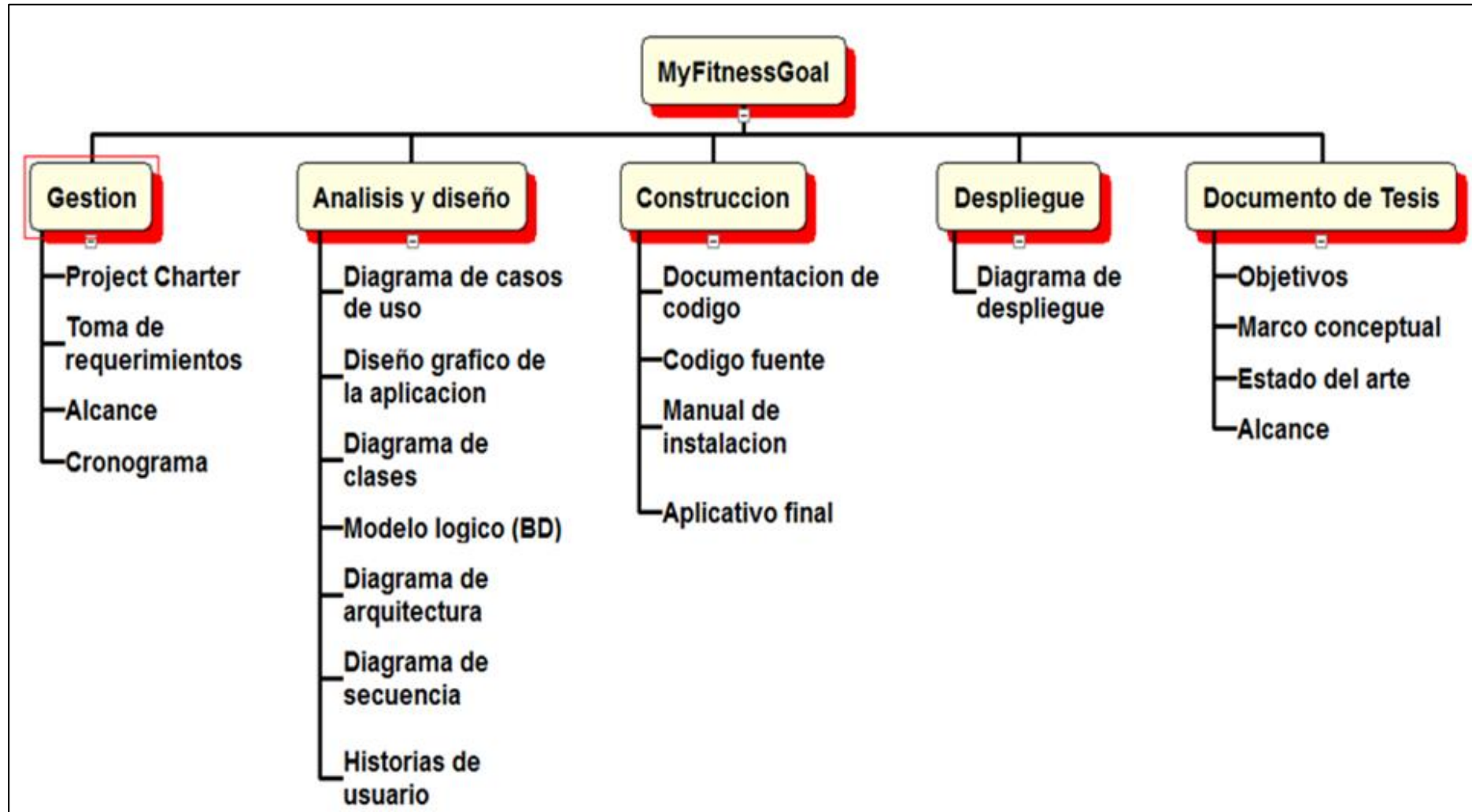
Página Web Oficial de MyBatis:

<http://www.mybatis.org/mybatis-3/es/>

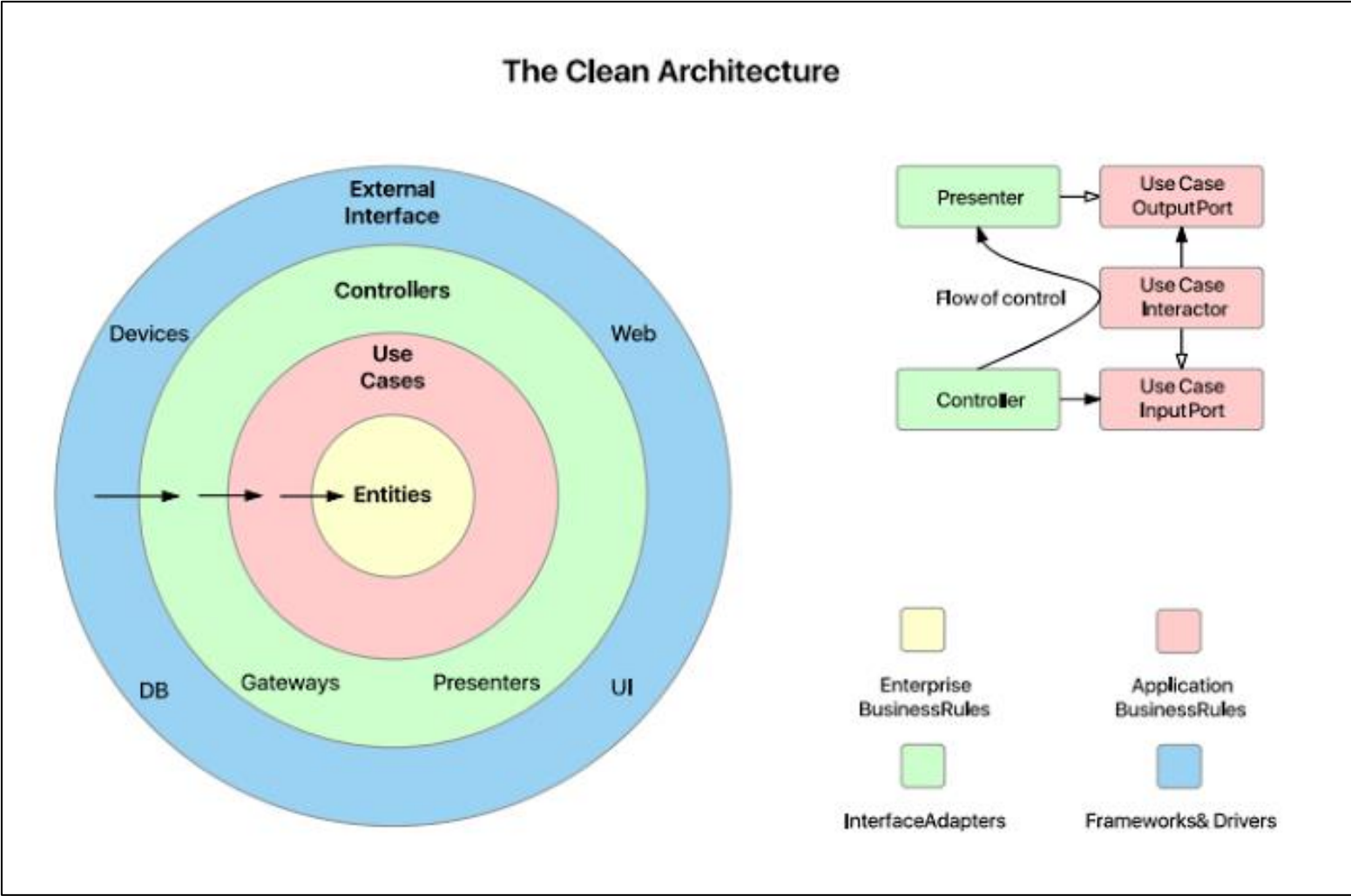
## **ANEXOS**

- Anexo 1      ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO
- Anexo 2      CLEAN ARCHITECTURE
- Anexo 3      CRONOGRAMA DEL PROYECTO

## ANEXO 1: ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO



# ANEXO 2: CLEAN ARCHITECTURE



La arquitectura limpia que se utilizó en el proyecto se basó en las siguientes capas:

- Views: Contenido de las pantallas originadas en el presentador. Es un mecanismo de entrega, que no se puede reutilizar entre plataformas.
- Presenter: Se centran en hacer que las respuestas del interactor estén listas para mostrarse. Transforma los modelos del interactor en modelos de vista. Se trata de una pieza importante, ya que puede ser reutilizado entre las plataformas (a diferencia de la vista, que son específicos de una plataforma).
- Interactor: Son los encargados de modelar un caso de uso, contienen las reglas de negocio y por lo general actúa como un mediador entre la base de datos y el presentador.
- Entities: Son los objetos del modelo manipulados por los indicadores, estos objetos nunca llegan a la capa de presentación.
- Routing: Representan cómo fluyen las pantallas en una aplicación. Cuando una vista se va a presentar, es responsable de hacer que aparezca en la pantalla.



# ANEXO 3: CRONOGRAMA DEL PROYECTO

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		1. MyTrainingGoal	142.06 days?	Mon 02/05/16	Sat 05/11/16	
2		1.1 Gestión	3.56 days	Mon 02/05/16	Sun 08/05/16	
3		1.1.1 Project Charter	2 hrs	Mon 02/05/16	Tue 03/05/16	
4		1.1.2 Toma de Requerimientos	2 days	Tue 03/05/16	Fri 06/05/16	3
5		1.1.3 EDT	1 day	Sat 07/05/16	Sat 07/05/16	4
6		1.1.4 Cronograma	2.5 hrs	Sat 07/05/16	Sun 08/05/16	5
7		1.1.5 Fin Modulo Gestion	0 days	Sun 08/05/16	Sun 08/05/16	6
8		1.2 Documento de Tesis	5 days	Sun 08/05/16	Sat 14/05/16	
9		1.2.1 Objetivos	1 day	Sun 08/05/16	Sun 08/05/16	
10		1.2.1.1 Objetivo general	0.5 days	Sun 08/05/16	Sun 08/05/16	7
11		1.2.1.2 Objetivo específico	0.5 days	Sun 08/05/16	Sun 08/05/16	10
12		1.2.2 Marco Conceptual	1 day	Sun 08/05/16	Mon 09/05/16	9
13		1.2.3 Estado del Arte	1 day	Mon 09/05/16	Wed 11/05/16	12
14		1.2.4 Alcance	2 days	Wed 11/05/16	Sat 14/05/16	13
15		1.2.5 Fin Modulo Documento de Tesis	0 days	Sat 14/05/16	Sat 14/05/16	14
16		1.3 Análisis y Diseño	36.5 days	Sat 14/05/16	Fri 01/07/16	
17		1.3.1 Diagrama de Paquetes	3 hrs	Sat 14/05/16	Sun 15/05/16	15
18		1.3.2 Diagrama de Actores	3 hrs	Sun 15/05/16	Sun 15/05/16	17
19		1.3.3 Diagrama de Casos de Uso	2 wks	Sun 15/05/16	Sun 29/05/16	18
20		1.3.4 Diseño gráfico de la aplicación	2 wks	Sun 15/05/16	Sun 29/05/16	18
21		1.3.5 Historias de Usuario	3 wks	Sun 29/05/16	Tue 21/06/16	19
22		1.3.6 Diagrama de clases	1 day	Tue 21/06/16	Thu 23/06/16	21
23		1.3.7 Modelo Físico(BD)	3 days	Thu 23/06/16	Sun 26/06/16	22
24		1.3.8 Diagrama de Arquitectura	3 days	Sun 26/06/16	Fri 01/07/16	23
25		1.3.9 Fin Modulo Análisis y Diseño	0 days	Fri 01/07/16	Fri 01/07/16	24
26		1.4 Construcción	86.75 days	Fri 01/07/16	Sat 22/10/16	16
27		1.4.1 Aplicativo final	74.75 days	Fri 01/07/16	Thu 06/10/16	25
28		1.4.1.1 Desarrollo del Paquete de Seguridad	5.75 days	Fri 01/07/16	Sat 09/07/16	25
29		1.4.1.1.1 Desarrollo del CUS Inicio de Sesión	0.5 wks	Fri 01/07/16	Sun 03/07/16	25
30		1.4.1.1.2 Desarrollo del CUS Cambiar Contraseña	0.5 wks	Sun 03/07/16	Sat 09/07/16	29
31		1.4.1.2 Desarrollo del Paquete de Administración	23 days	Sat 09/07/16	Sun 07/08/16	30
32		1.4.1.2.1 Desarrollo del CUS Administrar Dietas	1 wk	Sat 09/07/16	Sat 16/07/16	30
33		1.4.1.2.2 Desarrollo del CUS Administrar Evaluaciones	1 wk	Sat 16/07/16	Sat 23/07/16	32
34		1.4.1.2.3 Desarrollo del CUS Administrar Evaluaciones Iniciales	1 wk	Sat 23/07/16	Sun 31/07/16	33
35		1.4.1.2.4 Desarrollo del CUS Administrar Alimentos	1 wk	Sun 31/07/16	Sun 07/08/16	34
36		1.4.1.3 Desarrollo del Paquete de Consultas	46 days	Sun 07/08/16	Thu 06/10/16	35
37		1.4.1.3.1 Desarrollo del CUS Consultar Evaluaciones	0.5 wks	Sun 07/08/16	Thu 11/08/16	35
38		1.4.1.3.2 Desarrollo del CUS Consultar Evaluaciones Iniciales	0.5 wks	Thu 11/08/16	Sun 14/08/16	37
39		1.4.1.3.3 Desarrollo del CUS Consultar Barra Alimentos	1 wk	Sun 14/08/16	Sun 21/08/16	38
40		1.4.1.3.4 Desarrollo del CUS Consultar Programas Gimnasio	1 wk	Sun 21/08/16	Sun 28/08/16	39
41		1.4.1.3.7 Desarrollo del CUS Consultar Dietas	1 wk	Sun 28/08/16	Mon 05/09/16	40
42		1.4.1.3.8 Desarrollo del CUS Consultar Tienda	1 wk	Mon 05/09/16	Tue 13/09/16	41
43		1.4.1.3.9 Desarrollo del CUS Consultar Notificaciones	1 wk	Tue 13/09/16	Wed 21/09/16	42
44		1.4.1.3.10 Desarrollo del CUS Calendario	1 wk	Wed 21/09/16	Wed 28/09/16	43
45		1.4.1.3.11 Desarrollo del CUS Chatear	0.5 wks	Wed 28/09/16	Sun 02/10/16	44
46		1.4.1.3.12 Desarrollo del CUS Llenar Encuesta	0.5 wks	Sun 02/10/16	Thu 06/10/16	45
47		1.4.2 Documentación de código	2 wks	Thu 06/10/16	Sat 22/10/16	27
48		1.4.2 Código fuente	1 hr	Sat 22/10/16	Sat 22/10/16	47
49		1.4.3 Manual de instalación	3 hrs	Sat 22/10/16	Sat 22/10/16	48
50		1.4.4 Pruebas de Software	1 wk	Thu 06/10/16	Fri 14/10/16	27
51		1.4.5 Fin Modulo Construcción	0 days	Fri 14/10/16	Fri 14/10/16	50
49		1.4.3 Manual de instalación	3 hrs	Sat 22/10/16	Sat 22/10/16	48
50		1.4.4 Pruebas de Software	1 wk	Thu 06/10/16	Fri 14/10/16	27
51		1.4.5 Fin Modulo Construcción	0 days	Fri 14/10/16	Fri 14/10/16	50
52		1.5 Pruebas	2 days?	Fri 14/10/16	Sun 16/10/16	
53		1.5.1 Diseño del Plan de Pruebas	0.5 days?	Fri 14/10/16	Sat 15/10/16	51
54		1.5.2 Realización de las Pruebas Unitarias	0.5 days?	Sat 15/10/16	Sat 15/10/16	53
55		1.5.3 Realización de las Pruebas Funcionales	0.5 days?	Sat 15/10/16	Sun 16/10/16	54
56		1.5.4 Fin Modulo Pruebas	0.5 days?	Sun 16/10/16	Sun 16/10/16	55
57		1.6 Ajustes a la documentación	11.5 days	Sun 16/10/16	Sun 30/10/16	
58		1.6.1 Diagrama de secuencia	1 wk	Sun 16/10/16	Sun 23/10/16	56
59		1.6.2 Diagrama de estado	1 wk	Sun 23/10/16	Sun 30/10/16	58
60		1.6.3 Fin Ajustes a la documentación	0 days	Sun 30/10/16	Sun 30/10/16	59
61		1.7 Despliegue	3 days	Sun 30/10/16	Sat 05/11/16	
62		1.7.1 Diagrama de despliegue	3 days	Sun 30/10/16	Sat 05/11/16	59
63		1.7.2 Fin Despliegue	0 days	Sat 05/11/16	Sat 05/11/16	62
64		1.8 Fin Proyecto	0 days	Sat 05/11/16	Sat 05/11/16	62