



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

**CONTROL DE LA SALUD A TRAVÉS DE LA MEDICINA DEPORTIVA
UTILIZANDO UN SISTEMA BASADO EN CONOCIMIENTO Y MACHINE
LEARNING**

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

LIC (A). MATEMÁTICAS APLICADAS Y COMPUTACIÓN

P R E S E N T A:

ARENAS MARTÍNEZ AMANDA RUBÍ

GARCÍA RODRÍGUEZ LUIS EDUARDO

DIRECTOR DE TESIS

DR. EDUARDO ELOY LOZA PACHECO



Santa Cruz Acatlán, Estado de México, 2021



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Índice

Capítulo I. Introducción	6
1.1 Introducción	6
1.2 Objetivo general	8
1.3 Objetivos específicos	8
1.4 Hipótesis	8
1.5 Justificación	9
1.6 Organización de la tesis	10
Capítulo II. Marco Teórico	14
2.1 Medicina del deporte	14
2.1.1 Nutrición	14
2.1.2 Ejercicio	14
2.2 Componentes de un sistema basado en conocimiento	15
2.2.1 Representación del conocimiento	15
2.2.2 Motor de búsqueda	18
2.3 Motor de inferencia	19
2.4 Machine learning	20
2.5 Redes Neuronales	21
2.6 Lenguajes de programación	23
2.6.1 Swift	24
2.6.2 XCode	24
2.7 Tecnologías de Machine Learning	25
2.7.1 Core Model	25
2.7.2 Core ML	25
2.7.3 Core Motion	26
2.8 Base de Datos	26
2.9 Herramientas de Control de Versiones	28

Capítulo III. Análisis y Diseño	31
3.1 Documento de requerimientos inicial	31
3.2 Análisis de requerimientos funcionales y no funcionales	32
3.2.1 Requerimientos funcionales	32
3.2.2 Requerimientos no funcionales	33
3.3 Casos de Uso	34
3.4 Descripción de diagramas de uso	34
3.6 Diseño de la base de datos	42
3.7 Diseño de la aplicación	43
Capítulo IV. Implementación	50
4.1 Implementación de la base de datos	50
4.2 Implementación de la aplicación	55
4.5 Integración	75
Capítulo V. Pruebas y resultados	88
5.1 Pruebas de la aplicación	88
5.2 Pruebas de la funcionalidad (UX)	100
5.3 Sigüientes pasos (Entrevista con usuarios)	118
Capítulo VI. Conclusiones y trabajos futuros	120
Referencias	123
Glosario	127
Anexos	128

Resumen

El presente trabajo trata sobre la implementación de una aplicación que tiene un enfoque orientado al control de la salud a través de la medicina deportiva utilizando un sistema basado en conocimiento y tecnología de machine learning. Desde el punto de vista tecnológico se desarrolla un sistema que nos permita supervisar la salud a través de la medicina del deporte mediante la tecnología Core ML que aplica el machine learning. Para llegar a esto, se ha utilizado el método tradicional de desarrollo, el cual conlleva, las etapas de investigación, análisis de requerimientos del sistema, diseño e implementación de estos requerimientos, pasando por la etapa pruebas y llegando a los resultados. Finalmente, se espera que este documento sea de utilidad para los alumnos de la carrera de Matemáticas Aplicadas y Computación y cualquier persona ajena a la misma, para que pueda ocupar y aprovecharla para fines de entendimiento o en trabajos futuros propios.

Abstract

The present work shows an application that has an approach oriented to the control of health through sports medicine using a Knowledge-based system and machine learning technology. The main objective is to develop a system that allows us to monitor health through sports medicine based on Core ML and machine learning to carry out the corresponding analysis with the data obtained. The work employs a traditional development method, which involves the research stage, analysis of system requirements, implementation of these requirements, going through the testing stage, and reaching the results. Finally, it is expected that this document will be helpful for students of the Applied Mathematics and Computer Science degree and anyone outside of it so that they can use and take advantage of it for purposes of understanding or in future work of their own.

Capítulo I. Introducción

1.1 Introducción

Hoy en día, según la Organización Mundial de la Salud existen diversos padecimientos crónicos degenerativos como es la diabetes y la obesidad. La medicina del deporte es útil para dar un seguimiento puntual de la nutrición y los ejercicios a realizar. La obesidad y la diabetes son una epidemia presente en México y todo el mundo. La relación de ambos padecimientos es una situación preocupante, ya que 7 de cada 10 adultos y 1 de cada 3 niños tienen sobrepeso y obesidad, aproximadamente 422 millones de personas sufren diabetes en el mundo, además de que la diabetes causa cada año alrededor de 1.5 millones de muertes, con información de la página oficial de OMS [1]. La obesidad incrementa el riesgo de padecer enfermedades cardiovasculares y también algunos tipos de cáncer. A su vez, la obesidad es uno de los factores principales de padecer diabetes, la cual tiene complicaciones severas como la ceguera, amputación de extremidades, infecciones recurrentes, nefropatía diabética, entre otros. Para tal efecto, los sistemas médicos expertos juegan un papel importante en este panorama, ejemplo de ello podemos rastrearlo mediante aplicaciones utilizando inteligencia artificial particularmente machine learning. Existen diferentes tipos de sistema experto diseñados para facilitar el trabajo de algunos profesionales, por ejemplo existe uno que mide las probabilidades de que una persona sufre una angina de pecho en España, también está “MYCIN” el cual es capaz de diagnosticar enfermedades infecciosas de la sangre y recetar los antibióticos apropiados, por otro lado “PUFF” es especialista en enfermedades pulmonares y es capaz de diagnosticar enfermedades pulmonares, y “CADUCEUS” es utilizado para la realización de diagnósticos en medicina interna. Adicionalmente, las tecnologías de la información enfocadas en machine learning han sido benéficas para el ser humano en el campo de la salud, pasando desde que podemos detectar el cáncer con una simple foto tomada desde nuestro teléfono inteligente hasta ir descartando síntomas mediante cuestionarios muy específicos y luego ser tratados por un médico. Existen diferentes aplicaciones representativas que cuidan la salud, ejemplo de ellos es una aplicación llamada “Rec Style”, indica

cómo cuidar tu alimentación de manera segura y gratuita. Tecnologías como Android y Swift que son desarrolladas por Google y Apple respectivamente, existen para que se pueda desarrollar estas aplicaciones de la mejor manera y garantizando al usuario que tendrá la mejor atención. Sin embargo, se requiere una aplicación para prevenir este tipo de padecimientos como se menciona al inicio de la introducción, ya que si bien, existen sistemas que controlan el cómo se debe comer sanamente o controlar la alimentación a través de las calorías, o simplemente dar tips o rutinas de ejercicios, no existe hoy en día un sistema que nos permita unificar todo para dar una mejor observación de todos estos puntos antes mencionados relacionando así la medicina del deporte y la nutrición. También es importante observar que todas estas apps y sistemas puedan dar una solución a este problema. Por lo que el presente trabajo busca controlar la salud a través de sistemas basados en conocimientos y machine learning para la medicina del deporte para así dar una solución a este tipo de problemas lo cual implica controlar el exceso de grasas y promover la educación física.

1.2 Objetivo general

El objetivo es desarrollar un sistema que nos permita supervisar la salud a través de la medicina del deporte mediante un sistema basado en Core ML y machine learning para así realizar el análisis correspondiente con los datos obtenidos.

1.3 Objetivos específicos

1. Diseñar el sistema de control de salud utilizando medicina del deporte.
2. Diagrama de componentes de base de datos e integración, describir los elementos de medicina del deporte.
3. Diseño de la representación de conocimiento del sistema basado en conocimiento.
4. Diseño e implementación del modelo de datos (base de datos).
5. Diseño e implementación de la aplicación móvil.
6. Diseño del modelo de machine learning.
7. Implementación del sistema basado en conocimiento.
8. Implementación del sistema de control de salud utilizando "Core Motion".
9. Integración del sistema.

1.4 Hipótesis

Es posible desarrollar un sistema que nos permita supervisar la salud a través de la medicina del deporte mediante un sistema basado en Core ML y machine learning para así realizar el análisis correspondiente con los datos obtenidos.

1.5 Justificación

El presente trabajo de investigación aborda el estudio del control de la salud a través de la medicina deportiva utilizando un sistema basado en conocimiento y machine learning. El siguiente estudio es relevante dado que el control de la salud es muy importante hoy en día por los siguientes elementos que es la nutrición y el ejercicio lo cual nos permite estar más sanos, de igual manera nos permite llevar un control adecuado de esta salud. La inteligencia artificial y la tecnología de cómputo son importantes para el control automatizado de esto. Los sistemas basados en conocimientos nos permiten llevar un rastreo de estos elementos; por último, con la realidad aumentada nos permiten dar un seguimiento del ejercicio de una persona y sugerir mejoras a partir de eso. Con este fin el control de salud a través de la medicina deportiva utilizando un sistema basado en conocimiento y machine learning, se hará mediante la descripción de lo que es medicina del deporte y el análisis y diseño de un sistema inteligente. El diseño de la representación del conocimiento, el diseño de la base de datos y la aplicación. Después se realizará la implementación de la base de datos y la aplicación. Y se implementará un modelo de machine learning utilizando la biblioteca Core Motion y se hará el entrenamiento de este modelo con un conjunto de datos.

1.6 Organización de la tesis

Capítulo 1. Introducción:

Objetivo: Describir el problema y delimitar los objetivos de este, mediante la descripción del dominio del problema y los objetivos.

Resumen: En la introducción se realiza el planteamiento del problema, con el fin de delimitar el dominio de la presente investigación. Adicionalmente se describen los objetivos particulares del trabajo. Por otro lado, se realiza una argumentación que justifica la viabilidad del trabajo.

Capítulo 2. Marco Teórico:

Objetivo: Describir el caso de aplicaciones, los conceptos teóricos, así como los elementos científicos y tecnológicos que involucra el presente trabajo, como la inteligencia artificial, Machine Learning y la tecnología XCode, Swift, CoreML entre otras.

Resumen: El capítulo aborda los conceptos fundamentales de medicina del deporte con el fin de utilizar este tema como caso de uso. Además, se revisan los temas de sistemas basados en conocimiento, representación del conocimiento y algoritmos de razonamiento. Más adelante se mencionan los conceptos de aprendizaje de máquinas. Finalmente se concluye con una revisión sobre las tecnologías utilizadas en el presente trabajo que es el lenguaje Swift. La tecnología XCode, así como el framework CoreML y las bibliotecas de Machine Learning de Apple.

En este capítulo se podrá apreciar parte de la teoría aplicada para poder abordar el tema a tratar en este documento. Se incluirán parte de algunas definiciones, además de que se hablará sobre varios aspectos que son considerados durante capítulos posteriores para así entender la problemática a tratar.

Capítulo 3. Análisis y Diseño:

Objetivo: Realizar la obtención del conocimiento, análisis, síntesis, diseño del modelado de los datos y de la aplicación móvil.

Resumen: Para el presente capítulo se realiza un análisis y síntesis de los elementos necesarios para la construcción del trabajo. Se realizará el análisis de los datos, sobre dietas, ejercicios y temas relacionados con medicina del deporte. Con el propósito de extraer las características y variables necesarias para introducir en el modelo de Machine Learning. Adicionalmente se realizará un modelado de datos, para determinar los diagramas necesarios como son los casos de uso del sistema, el diagrama de clases, entre otros necesarios. Finalmente se realizará el diseño de la aplicación móvil, orientada a una estrategia de movilidad para el usuario.

Para este capítulo se dará a conocer un estudio del problema que se aborda en este trabajo junto con su debido desarrollo. Se definirá el problema planteado utilizando el debido análisis y se procederá a implementar una solución.

Así mismo, se analizará la propuesta como solución examinando la viabilidad, su ejecución y demás elementos para llevar a cabo la misma.

Después, se hará la respectiva interpretación del análisis de requerimientos de software ya obtenidos, para así crear un modelo que sea capaz de traducirlos a un sistema que satisfaga la solución. Esto para dar una mejor explicación a la idea de lo que se quiere hacer y lo que se desea que haga el software, para que así sea más legible a una tercera persona. Adicionalmente, se detalla la forma en la que se trabajará con la metodología seleccionada que organiza y estructura las etapas del desarrollo del presente trabajo.

Capítulo 4. Implementación:

Objetivos: Implementar el modelo de representación del conocimiento, los algoritmos y definir el modelo de machine learning para el caso de medicina del deporte.

En el presente trabajo se busca implementar las tecnologías como CoreML y otras de machine learning con el propósito de ayudar a la medicina del deporte.

Resumen: En el siguiente capítulo principalmente se procederá a implementar todo el análisis y diseño que se describió en el capítulo anterior, en el código correspondiente con el fin de obtener un sistema funcional. Se detalla el código a realizar y su respectiva justificación. Por último, se toman en cuenta las consideraciones necesarias en la codificación para así mejorar la experiencia del usuario en dicho sistema.

Capítulo 5. Pruebas y resultados:

Objetivo: Probar el modelo de Machine Learning, así como el funcionamiento de cada uno de los algoritmos y modelo de la aplicación, así como su integración total.

Resumen: Durante este capítulo se hablará sobre las pruebas planeadas y realizadas sobre este sistema. Estas pruebas podrán afirmar que el software producido se realiza de la forma en que se espera, sin errores y cumpliendo los requerimientos obtenidos anteriormente.

Adicionalmente, se describe la forma en que la aplicación se utiliza para que así el usuario pueda conocer el funcionamiento y características del software que está utilizando.

Capítulo 6. Conclusiones y trabajos futuros:

Objetivo: Mostrar las conclusiones a partir de la información recabada en la parte de pruebas y resultados, diseño y análisis e implementación, con el objetivo de demostrar que la hipótesis propuesta es verdadera.

Resumen: Por último, en este capítulo se discute sobre el producto ya terminado y sus resultados. Se describe el comportamiento del sistema dado que ya ha sido utilizado por los usuarios. Y se discute la veracidad de la hipótesis planteada en el capítulo 1.

Capítulo II. Marco Teórico

2.1 Medicina del deporte

La medicina deportiva o del deporte, estudia la aptitud del deporte, trabajando la prevención como el tratamiento de patologías derivadas de la misma práctica. Así pues, esta especialidad se encarga del cuidado, asesoramiento, diagnóstico y tratamiento necesario para la práctica de forma segura [2]. Entre la que destacan los siguientes elementos: nutrición y ejercicio.

2.1.1 Nutrición

La nutrición se refiere a los nutrientes que componen los alimentos, implica los procesos que suceden en el cuerpo después de comer, es decir la obtención, asimilación y digestión de los nutrimentos por el organismo [3]. Así mismo, la nutrición humana es la ciencia que investiga la relación entre los alimentos consumidos por el hombre y la salud (enfermedades), para buscar el bienestar y la preservación de la salud humana [4].

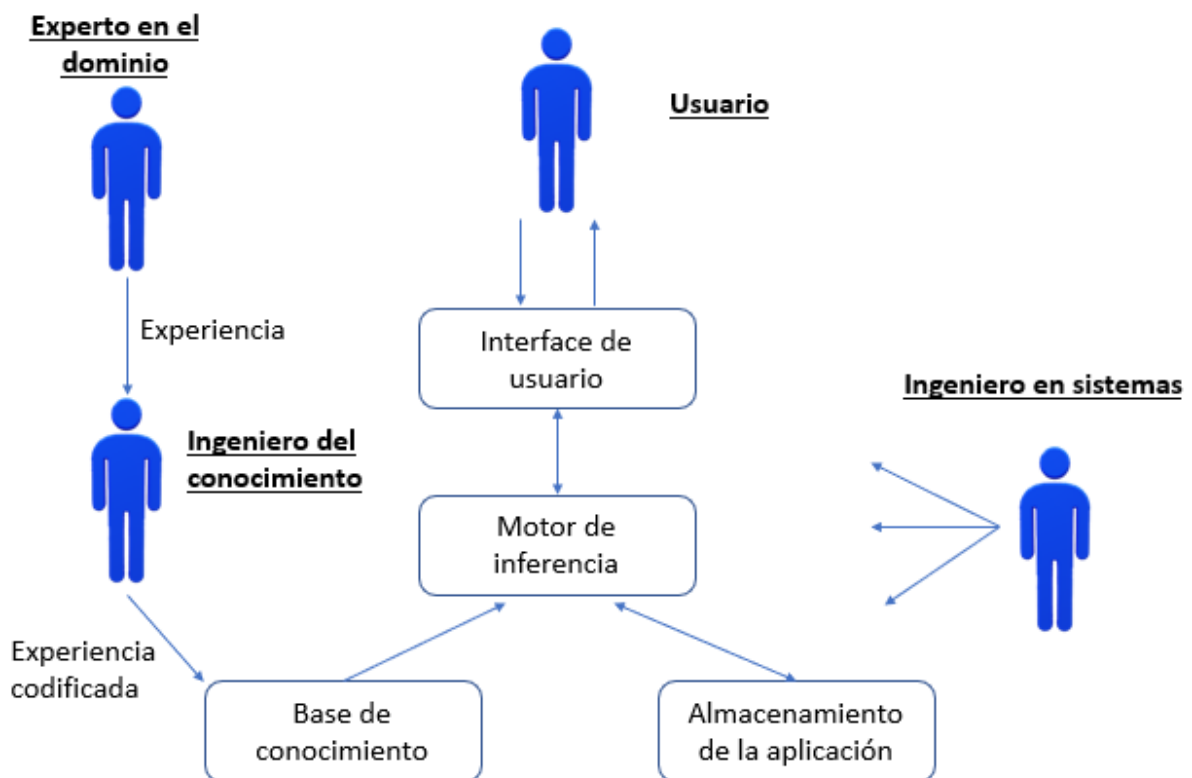
2.1.2 Ejercicio

El ejercicio es la actividad física planificada, estructurada y repetitiva, realizada para recuperar, mantener o mejorar una forma física, empleando un conjunto de movimientos corporales que se realiza para tal fin.

Por consiguiente, se considera ejercicio al conjunto de acciones motoras musculoesqueléticas [5].

2.2 Componentes de un sistema basado en conocimiento

Un sistema basado en conocimiento son diseñados para modelar el comportamiento de un experto en algún campo, está compuesto por una interfaz de usuario y un motor de inferencia, estos sistemas son altamente rentables, los problemas pueden resolverse con conocimiento experto, se tiene acceso a un experto, puede tener una o más soluciones, el problema y el conocimiento cambian rápidamente, un sistema experto no se considera acabado una vez que esté terminado debido a que siempre necesita de ajustes y mejoras por lo tanto siempre se está desarrollando [6].



Componentes de un sistema basado en conocimiento 2.1.

2.2.1 Representación del conocimiento

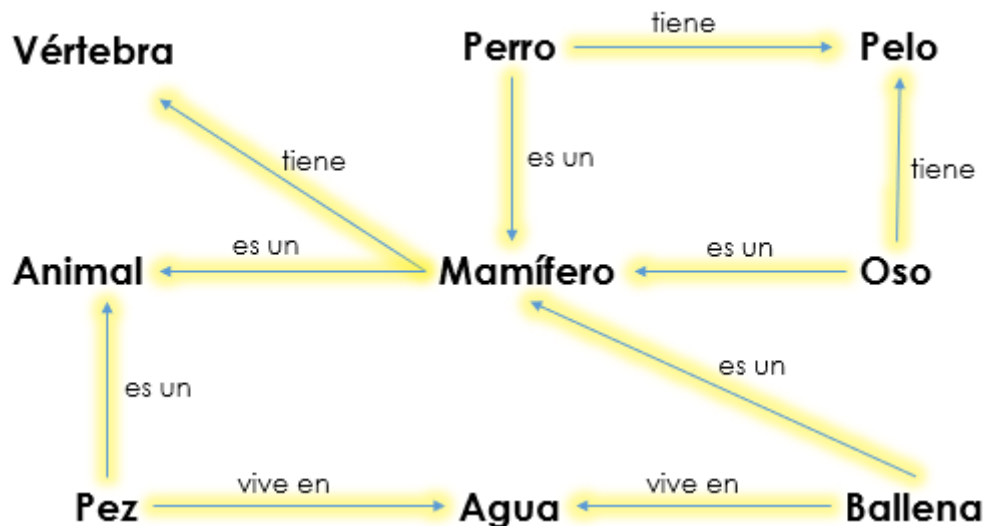
Una representación de conocimiento son estructuras de datos, con los cuales puede capturar los elementos esenciales que se utilizan para representar un problema y con esto podrá saber qué variables son las que se pueden utilizar y así mismo que operadores, para con esto poder crear un algoritmo eficiente. La representación del conocimiento es importante porque nos permite realizar un razonamiento. Algunas de las características de la representación del conocimiento son el alcance, la

granularidad, modularidad y entendimiento, conocimiento explícito y conocimiento implícito. Existen dos tipos de representaciones como la cuantitativa que son la geometría euclidiana y sistema coordenado, y los cualitativos que son los marcos conceptuales y las redes semánticas. El conocimiento para involucrar en los esquemas o estructuras se clasifica en 2 categorías:

- **Objetos:** Todo sistema debería tener información de las propiedades físicas de objetos y conceptos.
- **Dominio:** Es importante especificar las acciones que pueden llevar a cabo los objetos.

Hay quienes consideran algunos elementos más. A continuación, se mencionará los tipos de representación de conocimiento y una breve explicación:

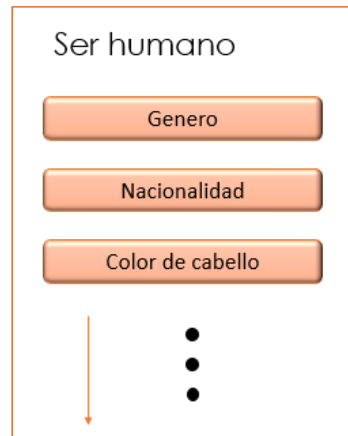
- **Redes semánticas:** Es un gráfico que consiste en nodos que están conectados por bordes, los nodos representan objetos, y los enlaces entre nodos representan relaciones entre esos objetos. Los enlaces suelen estar etiquetados para indicar la naturaleza de la relación. Es una representación comúnmente utilizada en Inteligencia Artificial.



Representación del conocimiento 2.2.1.1.

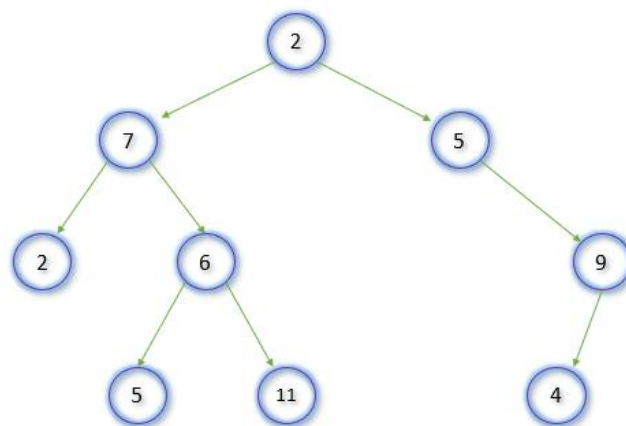
- **Marco conceptual:** Es un desarrollo de redes semánticas y permite expresar la idea de herencia. Consiste en un conjunto de marcos o nodos, que están conectados entre sí por relaciones. Los marcos conceptuales son una

representación orientada a objetos y están compuestos por slots y valores de slots. Cada cuadro describe ya sea una instancia o una clase. Estos cuadros permiten heredar propiedades a más de un marco a esto se le conoce como múltiple herencia.



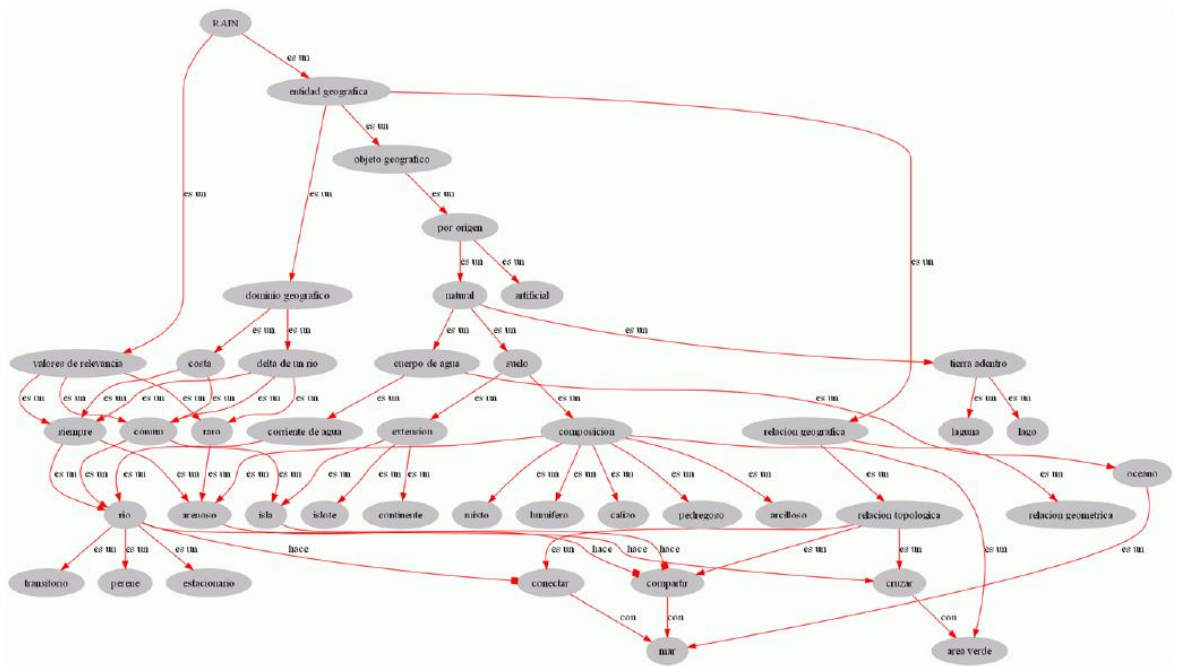
Marco conceptual 2.2.1.2.

- Árboles: Son un tipo de red semántica (o gráfica) en el que cada nodo excepto el nodo raíz, tiene exactamente un predecesor y cada nodo puede tener uno o más sucesores, un nodo que no tiene sucesores se le llama nodo hoja, uno o más nodos hojas son llamados nodos meta, una ruta que nos lleva desde el raíz a nuestra meta se le llama ruta completa (AND Path) y a la que no ruta parcial (OR-path), cuando un nodo se divide en dos nodos se le conoce como bifurcación y si un nodo se divide en n sucesores se le conoce como factor de bifurcación.



Red semántica tipo árbol 2.2.1.3.

- Ontologías: Es una descripción formal del conocimiento como un conjunto de conceptos dentro de un dominio y las relaciones que mantienen entre ellos. Necesitamos especificar componentes como individuos, clases, atributos, relaciones, así como restricciones, reglas y axiomas. Se puede aplicar a un conjunto de hechos individuales para crear una gráfica conectada de conocimiento [7].



Red ontológica 2.2.1.4.

2.2.2 Motor de búsqueda

Un motor de búsqueda es un motor de razonamiento en un sistema informático que busca elementos por ejemplo archivos almacenados en servidores web gracias a su “spider” (Web Crawler). Un ejemplo de esto son los buscadores de internet, algunos buscan sólo en la web, pero otros buscando además en noticias y/o servidores como Gopher, FTP, etc., cuando se pide la información sobre algún tema. Las búsquedas se hacen mediante palabras clave y con árboles jerárquicos por temas. El resultado de esta búsqueda es un listado de direcciones web en los que se mencionan temas relacionados con las palabras clave buscadas.

Los motores de búsqueda crean listados de sitios web utilizando arañas (spiders) que rastrean (crawl) las páginas web, indexan su información y siguen los enlaces de ellas hacia otras páginas web. Las arañas regresan con frecuencia a los sitios ya

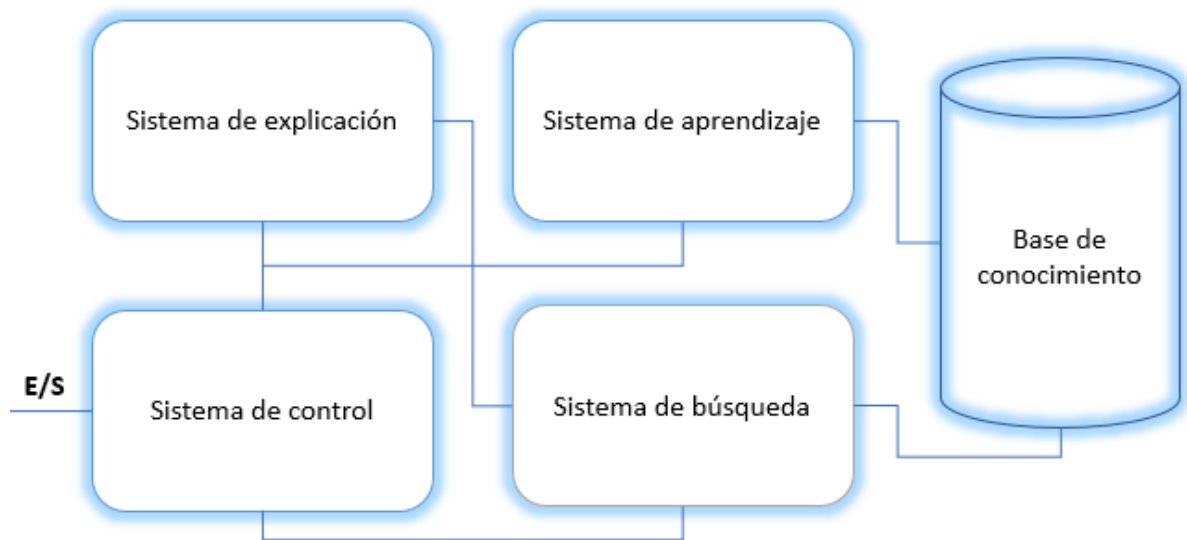
rastreados para comprobar actualizaciones o cambios, todo lo que encuentran queda reflejado en la base de datos del motor de búsqueda.

Existen tres tipos de motores de búsqueda; los de índices temáticos los cuales están compuestos por temas o categorías jerarquizados, los motores de búsqueda por palabras clave, los sistemas mixtos que tienen un directorio además de su motor de búsqueda y los motores de búsqueda semánticos los cuales tienen componentes de inteligencia artificial.

A su vez también existen clases de buscadores, los buscadores jerárquicos (arañas o spiders), directorios, sistemas mixtos (buscador-directorio), metabuscadores, FFA enlaces gratuitos para todos, buscadores verticales [8].

2.3 Motor de inferencia

Es un componente de un sistema experto, se encarga de administrar y controlar lógicamente el manejo y la utilización del conocimiento, el modelo del motor de inferencia es la estrategia de búsqueda para producir el conocimiento requerido. Sus principales funciones es examinar los hechos y las reglas y si es posible añade nuevos hechos, decide el orden en que se hacen las inferencias, verifica la consistencia de los hechos y los datos y actualiza la base de conocimiento. El motor está compuesto por un sistema de explicación, un sistema de control, uno de aprendizaje y otro de búsqueda, todo esto es lo que conforma la base de conocimiento [9].



Motor de inferencia 2.3.

2.4 Machine learning

El aprendizaje automático (Machine learning) también es llamado aprendizaje de máquina, es una parte extremadamente importante de la Inteligencia Artificial, ya que permite a un sistema aprender de los datos en lugar de aprender mediante la programación explícita. No es un proceso sencillo, conforme el algoritmo recopila datos de entrenamiento, es posible producir modelos más precisos basados en datos. Un modelo de machine learning es la salida de información que se genera cuando se entrena un algoritmo con datos. Después del entrenamiento al proporcionar un modelo con una entrada, se le dará una salida. Las técnicas de machine learning son necesarias para mejorar la precisión de los modelos predictivos, dependiendo del problema que se esté atendiendo existen diferentes enfoques basados en el tipo y volumen de los datos. Existen 4 categorías:

- ❑ **Aprendizaje Supervisado:** Comienza con un conjunto establecido de datos (patrones de entrada) y una función de clasificación. Intenta encontrar patrones en datos que se puedan aplicar a un proceso por ejemplo de analítica de datos, estos datos tienen características etiquetadas que definen el significado de los datos. Por ejemplo, CoreML cuenta con una base de datos basada en imágenes y descripciones que distingue entre millones de animales. Cabe mencionar que nuestro trabajo utiliza el aprendizaje supervisado.

- ❑ Aprendizaje no supervisado: Es utilizado cuando el problema requiere una cantidad masiva de datos que no están etiquetados. Requiere algoritmos que clasifican o bien, agrupan los datos con base en los patrones que encuentra para poder clasificarlos según su significado. Lleva a cabo un proceso iterativo analizando los datos con el menor grado de intervención humana. Una de sus aplicaciones más populares se utiliza con la tecnología de detección de spam en e-mails.
- ❑ Aprendizaje reforzado (Reinforcement learning): Es un modelo de aprendizaje conductual. El algoritmo recibe retroalimentación o estímulo del análisis de datos, conduciendo al usuario hacia el mejor resultado. El sistema aprende a través de la prueba y error, por lo tanto, una secuencia de decisiones exitosas conduce al fortalecimiento del proceso, porque es el que resuelve el problema de manera más efectiva.
- ❑ Deep learning: Incorpora las redes neuronales en capas sucesivas para aprender de los datos de manera iterativa. Es útil cuando se trata de aprender patrones de datos no estructurados. Se utilizan a menudo en el reconocimiento de imágenes, voz y aplicaciones de visión de computadora.

Actualmente existen muchas herramientas tanto privadas como de código abierto para poder desarrollar tecnologías de Machine learning [10].

2.5 Redes Neuronales

Una red neuronal es un modelo formal que emula el modo en que el cerebro humano procesa la información. En redes neuronales existen conceptos básicos para poder abordarlas, los cuales son:

- Perceptrón: Es una simulación de una neurona desarrollada en los años 50 por McCulloch y Pitts. El funcionamiento es sencillo, ya que su principal objetivo es la de poder generar un artefacto que aprenda para después poder clasificar datos de un conjunto. El entrenamiento del perceptrón no es más que el de realizar un conjunto de iteraciones denominadas épocas donde no se comete ningún error después de estas. El proceso comienza con valores aleatorios de los pesos y se van modificando estos según la diferencia entre los valores deseados y los calculados por la red, utilizando el triángulo de

Rosenblatt. En la siguiente imagen obtenemos el modelo matemático de un perceptrón:

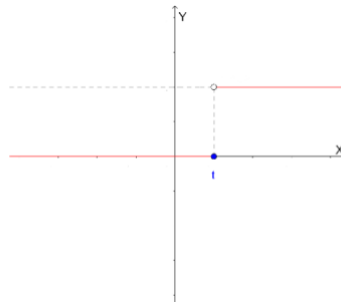
$$X = \sum_{i=1}^n p_i X_i$$
$$\text{Escalon}(X) = \begin{cases} 1 & \text{para } X > t \\ 0 & \text{para } X \leq t \end{cases}$$
$$p_x^* \leftarrow p_i + (a * x_i * e)$$

donde X es la entrada total del perceptrón

escalon(X) es la salida del perceptrón

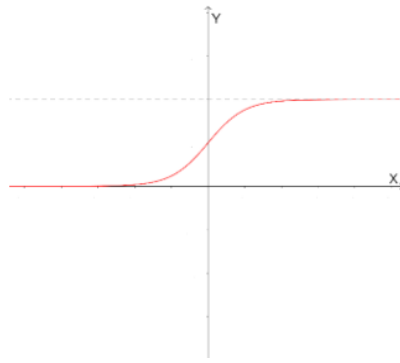
Ecuación de un perceptrón 2.5.1 [11].

- Funciones lineales: Las funciones lineales son usadas cuando el tipo del problema es linealmente separable.



Funciones lineales 2.5.2 [12].

- Funciones no lineales: Las no lineales son utilizadas en la mayoría de los problemas, aquellos que son linealmente no separables. La función escalón se utiliza en la mayoría de los problemas de redes neuronales.



Funciones no lineales 2.5.3 [13].

Una red neuronal puede aproximar una amplia gama de modelos predictivos con demandas mínimas sobre la estructura y asunción de modelos. La forma de las relaciones está determinada durante el proceso de aprendizaje. Si una relación lineal entre el objetivo y los predictores es apropiada, los resultados de la red neuronal deben aproximarse mucho a los del modelo lineal tradicional. Si una relación no lineal es más apropiada, la red neuronal aproxima automáticamente la estructura de modelo "correcta" [14].

2.6 Lenguajes de programación

Un lenguaje de programación es un conjunto de símbolos (alfabeto), gramática (palabras), reglas sintácticas y semánticas que indican cómo hacer un programa que puede ser procesado por una máquina formal [15]. Comúnmente encontramos diferentes lenguajes de programación que resuelven diferentes necesidades, como son los lenguajes estructurados, los lenguajes orientados a objetivos, los lenguajes imperativos entre otros. Dentro de los lenguajes de última generación, encontramos lenguajes como Python, Clojure y Swift. El lenguaje que se utilizará en este trabajo será Swift gracias a las prestaciones y la tecnología que tiene a su alrededor como es la programación más sencilla en inteligencia artificial (Corte ML y ARKit). A continuación, se describe brevemente el lenguaje de programación Swift.

2.6.1 Swift

Swift fue presentado por el vicepresidente de ingeniería de software Craig Federighi en la WWDC (World Wide Developer Conference) en el año 2014 [16]. Swift es un lenguaje de programación de alto nivel creado por Apple para el desarrollo de apps en los diferentes dispositivos que tienen, ya sea el iPhone, iPad, Mac, etc. Es un lenguaje de alto nivel, de propósito general (multipropósito y multiparadigma), además de eso, cuenta con muchas más características, una de ellas es que es seguro, ya que obliga a los programadores a cumplir ciertas restricciones para cumplir con los protocolos de seguridad. Su última versión es la 5.2, fue presentada el 2019 en la WWDC.



Lenguaje Swift 2.6.1 [17].

2.6.2 XCode

Introducido en octubre de 2003 junto a la versión 10.3 de Mac OS X, XCode es el entorno de desarrollo integrado (IDE) creado por Apple que trabaja de manera conjunta con Interface Builder (herramienta gráfica para crear interfaces de usuario) y que se puede descargar de manera gratuita desde la Mac App Store. Este conjunto de herramientas además incluye una colección de compiladores del proyecto GNU (GCC) y puede compilar código C, C++, Objective-C, Objective-C++, AppleScript y Java gracias a una gran gama de modelos de programación. Además, hay que destacar que por el momento XCode es el único software disponible para escribir código Swift [18].



XCode 2.6.2 [19].

2.7 Tecnologías de Machine Learning

La tecnología machine learning es esencial en el desarrollo de aplicaciones, ya sean de escritorio, móviles o web, con funcionalidades altamente complejas que se desarrollan íntegramente en diferentes ámbitos como la medicina, ingenierías, etc. Algunos ejemplos de ello es que se puede trabajar con la detección facial, el seguimiento de gestos o muecas e incluso detección de movimiento. Ya que el aprendizaje automático está implementado en el software en nuestros teléfonos, automóviles, hogares, etc. La mayoría de las industrias que trabajan con grandes cantidades de datos han reconocido el valor de la tecnología de machine learning ya que las organizaciones pueden trabajar de manera más eficiente o lograr ventajas significativas sobre su competencia a través de los datos analizados [20].

2.7.1 Core Model

Core Model ó un modelo central es el resultado de aplicar un algoritmo de aprendizaje automático a un conjunto de datos de entrenamiento [21].

2.7.2 Core ML

Core ML es un marco de aprendizaje automático, proporciona una representación unificada para todos los modelos, se utiliza para integrar modelos de aprendizaje en la aplicación. La aplicación puede utilizar las API's de Core ML y los datos del usuario para hacer predicciones y así entrenar o ajustar modelos, todo en el dispositivo del usuario. Core ML es la base de los marcos y la funcionalidad específica del dominio ya que admite visión para analizar imágenes, lenguaje natural para procesar el texto, voz para convertir audio en texto y análisis de sonido para identificar sonidos en audio. Se construye sobre primitivas de bajo nivel como Accelerate y BNNS, así como Metal Performance Shaders [22].



Función de Core ML 2.7.2 [23].

2.7.3 Core Motion

Core Motion es una biblioteca relacionada con el movimiento y el entorno desde el hardware integrado de los dispositivos iOS, incluidos los acelerómetros, giroscopios, podómetro, magnetómetro y barómetro. Utiliza este marco para acceder a datos generados por el hardware para poder usarlos en la aplicación. Muchos servicios le permiten acceder tanto a los valores sin procesar registrados por el hardware como a una versión procesada de estos valores. Por ejemplo, un juego podría utilizar datos de acelerómetro y giroscopio para controlar el comportamiento del juego en pantalla [24].

2.8 Base de Datos

El objetivo general de una base de datos es almacenar información de manera persistente y recuperar la información de una manera rápida, así como relacionar hechos y situaciones que previamente estaban separados en términos conceptuales. Una base de datos resulta exitosa, si cumple con los objetivos y las necesidades de los usuarios. Esto radica en la raíz del diseño de la base de datos. Por ello, la presencia de especialistas es crucial [25]. El punto fundamental de todas las bases de datos es que, actualmente no hay un entorno empresarial donde no se ocupe el almacenamiento de datos y que a su vez estas estén relacionadas entre sí [26].

Por lo anterior, la gestión de base de datos ha evolucionado la forma en que percibimos el mundo, desde una aplicación computacional hasta una parte esencial de un entorno de sistemas moderno [27].

Actualmente existen gestores de bases de datos que ayudan al manejo de grandes conjuntos de datos, así mismo, proporcionan acceso sencillo y eficaz a los datos almacenados, aportando elementos de certeza a la información como es integridad y consistencia en los datos, etc. A continuación, se muestran ejemplos de gestores de base de datos relevantes que nos ayudan a la manipulación de los datos:

MySQL es un gestor de base de datos relacional de código abierto que nos permite almacenar grandes cantidades de información. Este gestor es habitualmente utilizado para el *desarrollo web* por presentar rapidez en lectura ya que generalmente los desarrolladores de aplicaciones web no generan un elevado número de modificaciones de datos, sino más bien de *lectura frecuente de datos* [28].



Gestor de Base de Datos MySQL [29].

Oracle Database está formado por una colección de datos que se tratan como una sola unidad. El propósito de este gestor es almacenar y recuperar información relacionada entre sí. Como característica principal se tiene que, es capaz de gestionar y controlar una *gran cantidad de contenidos*, además de que, es un gestor multiplataforma, disponible para diferentes sistemas operativos como lo es Windows, Linux o Unix, permitiendo así, tener copias de las bases de datos productivas en lugares lejanos a la ubicación principal [30].



Gestor de Base de Datos MySQL 2.8.1 [31].

SQLite es un gestor de base de datos que actúa como un repositorio central de datos. Este gestor permite almacenar información de manera sencilla, eficiente, rápida y en equipos con *poca disponibilidad* de hardware como puede ser un teléfono celular o una computadora de gama baja, esto permite que soporte consultas convencionales desde las más básicas hasta las más complejas de SQL (*Structured Query Language*), además de que, se puede implementar en *dispositivos móviles* o de escritorio sin la necesidad de realizar procesos complejos de importación y exportación de datos.



Gestor de Base de Datos SQLite 2.8.1 [30].

2.9 Herramientas de Control de Versiones

Actualmente existen herramientas de control de versiones que ayudan a compartir y distribuir código entre los desarrolladores, generando así copias locales del proyecto para que puedan tener su propio entorno de desarrollo y publicar cambios dentro del repositorio central. A continuación, se da una breve descripción del proceso de control de versiones de manera lineal:

- **Creación:** Crear un repositorio central, donde cada desarrollador del proyecto tendrá acceso, configurándolo con su respectivo nombre y autenticación, si será privado o no, etc.
- **Probando el repositorio:** Subir la primera versión de la aplicación, verificando que todo funcione correctamente.
- **Cambios:** Dado que ya se creó el repositorio, se ha subido la primera versión de la aplicación y se tiene una copia local del proyecto, se procede hacer a

subir cambios por módulo, tomando en cuenta que, si se trabaja de forma colaborativa, se reparten los módulos para que se pueda trabajar de manera concurrente.

- Actualización: Por último, para estar en sincronía con los demás desarrolladores de software, se actualiza el proyecto que existe en el repositorio remoto respecto del local, así se puede seguir editando con una copia actualizada del sistema.

En general, las herramientas de control de versiones se basan en versionar archivos y carpetas en un repositorio, por lo que sus características principales, se pueden revisar cambios, deshacer cambios, marcar una revisión, etc.

Algunos ejemplos de herramientas de control de versiones de código son:

Git como sistema de control de versiones, fue creado por Linus Torvalds en 2005, cuenta con un *sistema distribuido* en el que cada desarrollador tiene una copia de manera local. En esta tecnología se trabaja mediante ramas, donde cada rama es destinada para un desarrollador o equipo. Git permite guardar un historial de cambios, así, se puede saber quién y cuándo se realizó la modificación, gracias a esta distribución, los desarrolladores pueden modificar el código fuente y hacer cambios sin la necesidad de tener conectividad [31].



Git 2.9.1 [32]

Otra es Azure DevOps como tecnología de versionamiento, permite que los roles que anteriormente estaban aislados, tanto desarrollo, calidad y seguridad, etc., se coordinan para crear productos de alta calidad, haciendo eficaz la recuperación de versiones anteriores. Este sistema de control de versiones es *centralizado*, esto significa que, el histórico de cambios está disponible únicamente en el servidor y se puede ver solamente cuando se está conectado, lo cual lo vuelve un entorno de desarrollo integrado. Así mismo, la creación de una rama de código requiere estar *conectado al servidor* [33].



Azure Devops 2.9.2 [34]

Bitbucket es una herramienta de control de versiones que permite enlazar archivos y carpetas locales a un alojamiento de su propio repositorio. Este manejador se puede clasificar como un entorno de colaboración de código de tipo *distribuido*. Su principal diferencia entre las herramientas de versionamiento que actualmente existen es que, se pueden crear repositorios privados ilimitados gratuitos [35].



Bitbucket 2.9.3 [36].

Capítulo III. Análisis y Diseño

El presente capítulo muestra el análisis y el diseño de la aplicación. Como es el documento requerimiento inicial, la descripción de los casos de uso, el modelado de datos, así como la presentación (sketch) de la aplicación [37].

3.1 Documento de requerimientos inicial

Analizando los requerimientos iniciales, se puede apreciar que se pueden dividir en requerimientos funcionales y no funcionales definidos en base a la medicina del deporte, los cuales permitirán formar un **plan** de salud, lo cual incluye ejercicios y nutrición para el usuario y así tener un mejor **control** de la salud facilitando el acceso por medio de una aplicación móvil. Como estudiantes observamos la carencia que existe en las personas sobre su control de salud, es por eso que se desarrolló un sistema, el cual permite al usuario poder tener una rutina de ejercicio y un mejor control de su peso, así mismo también poder medir sus calorías.

La aplicación móvil muestra una pantalla con dos secciones, las cuales serán ejercicio y nutrición, de las cuales se componen cada una por lo siguiente:

Ejercicio	Nutrición
Rutinas	Recetas
Comparación	Calculadora de nutrición
Consejos	Control de calorías
	Dietas

Secciones en las que se divide la aplicación 3.1.

3.2 Análisis de requerimientos funcionales y no funcionales

A partir de lo anterior, es necesario incorporar los elementos antes mencionados en requerimientos funcionales y no funcionales [35].

3.2.1 Requerimientos funcionales

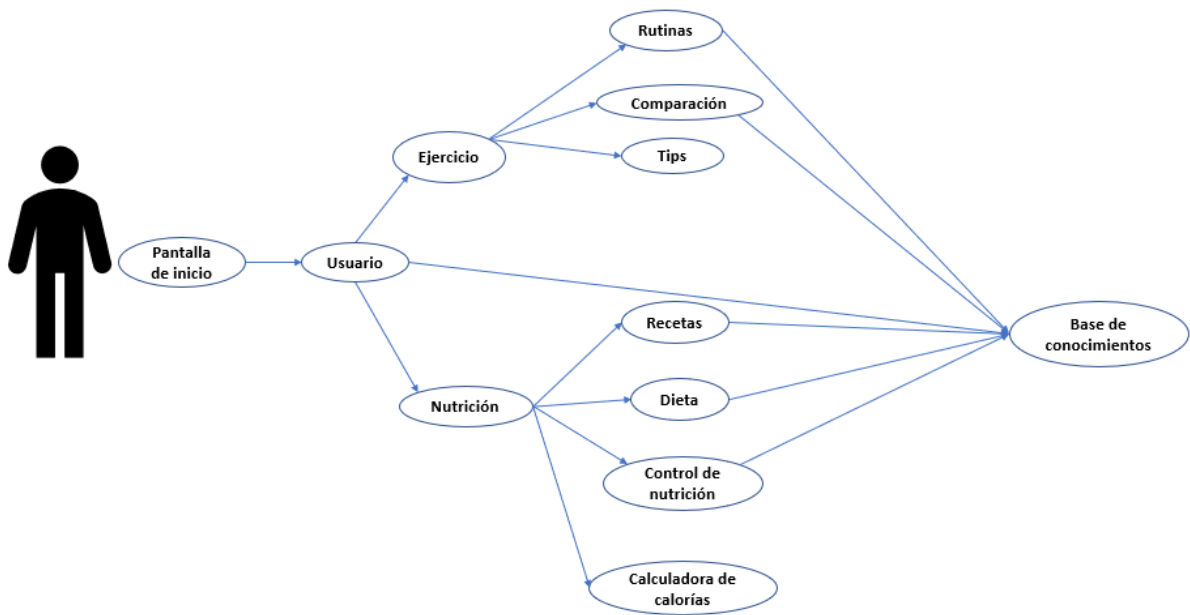
- La aplicación debe ser disponible para iOS y contar con una versión mínima de 10.1
- Se requiere pedir al usuario permiso para acceder a su cámara.
- Se requiere pedir al usuario permiso para almacenar datos.
- Para el desarrollo de nuestra aplicación es indispensable contar con una base de datos.
- El usuario debe contar con alguna cuenta de correo.
- Para las vistas de nuestra aplicación es indispensable contar con un botón que manda a la vista anterior.
- La aplicación móvil debe de tener una pantalla de inicio el cual contenga dos funcionalidades para que el usuario pueda elegir entre ejercicio y nutrición.
- Las imágenes tomadas con el dispositivo móvil se almacenarán en el teléfono.
- La aplicación móvil contará con un menú principal los cuales a su vez contarán con menús secundarios.
- En caso de que la elección del usuario sea ejercicio se mostrará un submenú con tres opciones las cuales serán rutinas, comparación y consejos.
- Para la sección de rutinas se debe dividir en 4 formas según la parte del cuerpo que se desee trabajar podrán ser brazo, abdomen, pierna y glúteos.
- En cada una de estas opciones se desea mostrar imágenes ilustrativas de los ejercicios en forma de lista donde cada imagen contendrá a su vez 2 botones de retroceder y avanzar respectivamente y una breve descripción del ejercicio a realizar.
- Para la parte de comparación forzosamente el usuario se tendrá que tomar una foto para que el sistema lo compare con la forma correcta de hacerlo y así decir si está bien hecho o no el ejercicio con su porcentaje de ejercicio respectivamente de cada categoría.

- En la parte sobre consejos se da a conocer tips para realizar las rutinas de una mejor manera.
- En caso de que la elección del usuario sea Nutrición se mostrará un submenú con 4 opciones que serán recetas, control de nutrición, calculadora de calorías y dieta.
- Para la sección de recetas se mostrarán diferentes imágenes con su respectivo nombre del platillo y así el usuario podrá elegir el que quiera visualizar.
- Para la sección de control de nutrición se mostrará una imagen con sus diferentes características y así el usuario va a saber cuanto está consumiendo por porción.
- Para la sección calculadora de calorías se le solicitará al usuario que llene campos como edad, altura, sexo y peso, ya que con esto se calculara si las kcal que consume están dentro del rango predefinido.
- Para la sección dietas se considera que se debe de tener un catálogo de los diferentes tipos de dietas que el usuario pueda elegir según lo requiera.

3.2.2 Requerimientos no funcionales

- Al ejecutar la aplicación mostrar una imagen con el logo de la aplicación.
- Se requiere pedir al usuario permiso para acceder a su ubicación.
- Se requiere pedir al usuario permiso para acceder a sus contactos y/o llamadas.
- El dispositivo deberá tener conexión a internet en todo momento para usar la aplicación.

3.3 Casos de Uso



Casos de uso del sistema 3.2.

3.4 Descripción de diagramas de uso

CU-MDAIOS-001

1	Breve descripción: Inicio de la aplicación
2	Actores: Usuario
3	Flujo de eventos: El sistema muestra una imagen representativa a la aplicación.
4	Requerimiento especial: Otorgar los permisos necesarios a la aplicación.
5	Precondición: Tener un dispositivo iOS. Tener la versión 10.1 Tener instalada la aplicación móvil.
6	Postcondición: Se redirige al caso de uso CU-MDAIOS-001.1
7	Puntos de extensión: Iniciar sesión mediante correo y contraseña en la aplicación para que el sistema tenga un mejor control sobre el usuario.

CU-MDAIOS-001.1

1	Breve descripción: Pantalla de inicio
2	Actores: Usuario
3	<p>Flujo de eventos: El sistema muestra el logo de la aplicación. El sistema muestra la pantalla para poder iniciar sesión o da la opción de registrarse. Para iniciar sesión se piden dos campos, el de correo y contraseña para así validar los datos. Para la opción de registrarse se piden los campos de nombre, apellido paterno, apellido materno, fecha de nacimiento, padecimientos, correo, contraseña, para que así pueda concluir con éxito el registro o en su defecto cancelar. Ya que se inició correctamente la sesión, el sistema muestra dos campos de ejercicio y nutrición respectivamente. Si el inicio de sesión fue erróneo, se mandará una alerta de datos erróneos.</p>
4	Requerimiento especial: Contar con al menos un correo electrónico de cualquier dominio.
5	<p>Precondición: Haber ejecutado de manera correcta la aplicación y que se muestre la pantalla de inicio.</p>
6	<p>Postcondición: Haber iniciado sesión con éxito Se redirige al caso de uso CU-MDAIOS-001.1.1 o al caso de uso CU-MDAIOS-001.1.2</p>
7	<p>Puntos de extensión: Nos direcciona a la pantalla de ejercicio. Nos direcciona a la pantalla de nutrición.</p>

CU-MDAIOS-001.1.1

1	Breve descripción: Ejercicio
2	Actores: Usuario
3	<p>Flujo de eventos: El sistema muestra el título de la vista en la que actualmente se encuentra. El sistema muestra un menú secundario que incluye rutinas, comparación y tips. Este menú se conforma por imágenes y sus títulos.</p>
4	Requerimiento especial: No aplica
5	<p>Precondición: Haber elegido dentro del menú principal el caso de uso de "Ejercicio" CU-MDAIOS-001.1.1</p>

6	<p>Postcondición: Tener la conexión a la siguiente pantalla dependiendo la elección del usuario para así elegir entre los casos de uso CU-MDAIOS-001.1.1.1, CU-MDAIOS-001.1.1.2 y CU-MDAIOS-001.1.1.3</p>
7	<p>Puntos de extensión: Nos dirige a la vista de rutinas. Nos direcciona a la vista de comparación. Nos direcciona a la vista de tips.</p>

CU-MDAIOS-001.1.2

1	<p>Breve descripción: Nutrición</p>
2	<p>Actores: Usuario</p>
3	<p>Flujo de eventos: El sistema muestra el título de la vista en la que actualmente se encuentra. El sistema muestra un menú secundario que incluye recetas, control de nutrición, calculadora de calorías y dieta. Este menú se conforma por imágenes y sus títulos.</p>
4	<p>Requerimiento especial: No aplica</p>
5	<p>Precondición: Haber elegido dentro del menú principal el caso de uso de “Nutrición” CU-MDAIOS-001.1.2</p>
6	<p>Postcondición: Tener la conexión a la siguiente pantalla dependiendo la elección del usuario para así elegir entre los casos de uso CU-MDAIOS-001.1.2.1, CU-MDAIOS-001.1.2.2, CU-MDAIOS-001.1.2.3 y CU-MDAIOS-001.1.2.4</p>
7	<p>Puntos de extensión: Nos direcciona a la vista de recetas. Nos dirige a la vista de control de nutrición. Nos dirige a la vista de la calculadora de calorías. Nos direcciona a la vista de dieta.</p>

CU-MDAIOS-001.1.1.1

1	<p>Breve descripción: Rutinas</p>
2	<p>Actores: Usuario</p>
3	<p>Flujo de eventos:</p>

	<p>El sistema muestra el título de la vista en la que actualmente se encuentra. El sistema muestra un menú donde se encuentran diferentes opciones como son pierna, brazo, abdomen y glúteos. Este menú se conforma por imágenes y sus títulos. Al momento de seleccionar alguno se abrirá una pantalla que tendrá una rutina específica de la opción antes seleccionada.</p>
4	Requerimiento especial: No aplica
5	Precondición: Haber elegido dentro del menú el caso de uso de “Rutinas” CU-MDAIOS-001.1.1.1
6	Postcondición: Tener la conexión a la siguiente pantalla dependiendo la elección del usuario.
7	Puntos de extensión: Nos direcciona a la vista de pierna. Nos direcciona a la vista de brazo. Nos direcciona a la vista del abdomen. Nos dirige a la vista de glúteos. En cada vista que seleccione podrá encontrar el título de la vista en la que se encuentre actualmente. En cada vista se encuentran 7 secciones correspondientes a cada día de la semana. En estas secciones al elegir una, se muestra el título del día elegido, así mismo se muestra una imagen representativa del ejercicio y una breve descripción de las instrucciones, todo esto por cada una de las rutinas correspondientes. Cuenta con un campo que realiza la acción de regresar al caso de uso CU-MDAIOS-001.1.1

CU-MDAIOS-001.1.1.2

1	Breve descripción: Comparación
2	Actores: Usuario
3	Flujo de eventos: El sistema muestra el título de la vista en la que actualmente se encuentra. El sistema muestra del lado superior izquierdo un botón el cual nos servirá para comparar las imágenes. El sistema muestra la foto que se tomó el usuario o la imagen que se va a comparar. Al momento de seleccionar el botón se hace la comparación. Cuando el sistema termina de comparar la foto, muestra un icono con el porcentaje realizado por cada categoría comparada.
4	Requerimiento especial: Contar con acceso a la galería del dispositivo.

	Contar con una versión mínimo o superior de 10.1
5	Precondición: Haber elegido dentro del menú el caso de uso de “Comparación” CU-MDAIOS-001.1.1.2
6	Postcondición: Cuenta con un campo que realiza la acción de regresar al caso de uso CU-MDAIOS-001.1.1
7	Puntos de extensión: Botón que servirá para comparar las fotos. Imagen del ejercicio que fue cargada previamente por el usuario. Icono que indica los resultados en porcentaje de cada categoría comparada.

CU-MDAIOS-001.1.1.3

1	Breve descripción: Consejos
2	Actores: Usuario
3	Flujo de eventos: El sistema muestra el título de la vista en la que actualmente se encuentra. El sistema muestra el título del tip del día. El sistema muestra una descripción del tip del día.
4	Requerimiento especial: No aplica
5	Precondición: Haber elegido dentro del menú el caso de uso de “Consejos” CU-MDAIOS-001.1.1.3
6	Postcondición: Cuenta con un campo que realiza la acción de regresar al caso de uso CU-MDAIOS-001.1.1
7	Puntos de extensión: Título del tip del día. Texto del tip del día.

CU-MDAIOS-001.1.2.1

1	Breve descripción: Recetas
2	Actores: Usuario
3	Flujo de eventos: El sistema muestra el título de la vista en la que actualmente se encuentra.

	<p>El sistema muestra el título de la receta.</p> <p>El sistema muestra una imagen de la receta.</p> <p>El sistema muestra una breve descripción de la receta.</p>
4	Requerimiento especial: No aplica
5	Precondición: Haber elegido dentro del menú el caso de uso de “Recetas” CU-MDAIOS-001.1.2.1
6	Postcondición: Cuenta con un campo que realiza la acción de regresar al caso de uso CU-MDAIOS-001.1.2
7	Puntos de extensión: Títulos de las recetas. Imágenes de las recetas. Breves descripciones de las recetas.

CU-MDAIOS-001.1.2.2

1	Breve descripción: Control de nutrición
2	Actores: Usuario
3	Flujo de eventos: El sistema muestra el título de la vista en la que actualmente se encuentra. El sistema muestra el título de la receta. El sistema muestra una imagen del alimento requerido. El sistema muestra la información del alimento, mostrando como campos: cantidad en gramos del alimento, las calorías, el número de proteína y las grasas que contiene.
4	Requerimiento especial: No aplica
5	Precondición: Haber elegido dentro del menú el caso de uso de “Control de nutrición” CU-MDAIOS-001.1.2.2
6	Postcondición: Cuenta con un campo que realiza la acción de regresar al caso de uso CU-MDAIOS-001.1.2
7	Puntos de extensión: Título de las recetas. Imágenes de las recetas. Se muestran los campos: Cantidad, calorías, proteína y grasas.

CU-MDAIOS-001.1.2.3

1	Breve descripción: Calculadora de calorías
2	Actores: Usuario
3	Flujo de eventos: El sistema muestra el título de la vista en la que actualmente se encuentra. El sistema muestra cinco diferentes campos que son edad, altura, género, peso, los cuales serán llenados por el usuario en el momento en el que se desee saber el control de consumo de calorías. El sistema muestra un campo donde se da el resultado del índice de masa corporal.
4	Requerimiento especial: Llenar los datos correspondientes
5	Precondición: Haber elegido dentro del menú el caso de uso de “Calculadora de calorías” CU-MDAIOS-001.1.2.3
6	Postcondición: Cuenta con un campo que realiza la acción de regresar al caso de uso CU-MDAIOS-001.1.2
7	Puntos de extensión: Muestra los campos que el usuario ingresó. Muestra el índice de masa corporal.

CU-MDAIOS-001.1.2.4

1	Breve descripción: Dietas
2	Actores: Usuario
3	Flujo de eventos: El sistema muestra el título de la vista en la que actualmente se encuentra. El sistema muestra el título de cada dieta. El sistema muestra una imagen referente a la dieta correspondiente. El sistema muestra una breve descripción de la misma dieta.
4	Requerimiento especial: No aplica
5	Precondición: Haber elegido dentro del menú el caso de uso de “Dietas” CU-MDAIOS-001.1.2.4
6	Postcondición: Cuenta con un campo que realiza la acción de regresar al caso de uso CU-MDAIOS-001.1.2

7

Puntos de extensión:

Muestra el título de cada dieta.

Muestra la imagen de cada dieta.

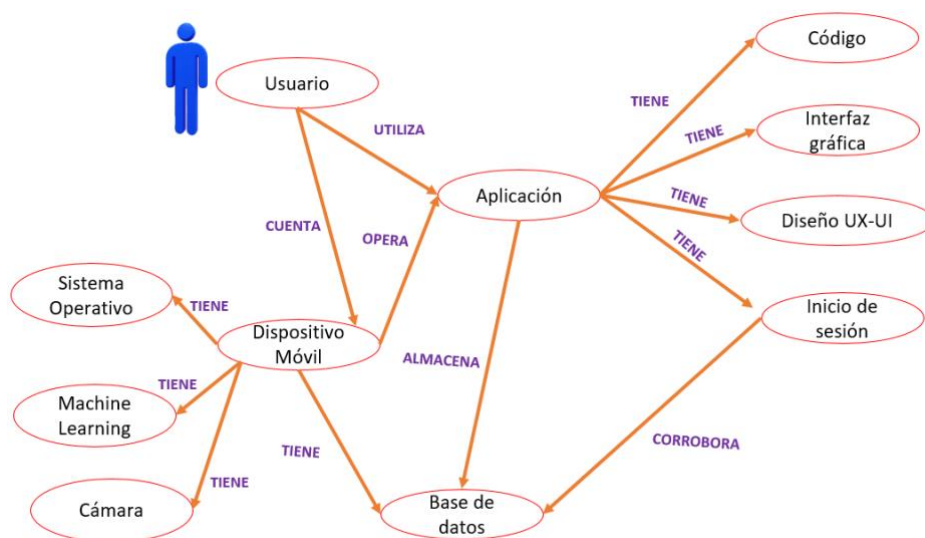
Muestra una breve descripción de cada dieta.

3.5 Diseño de la representación del conocimiento

La representación del conocimiento es importante porque podemos entender con mayor claridad la funcionalidad e interconexión de los conceptos de cualquier sistema que se desee estudiar y entender a profundidad. Además de que es necesario para que este pueda ser procesado por una computadora. Estas características facilitan el aprendizaje de conceptos (etiquetas) que serán nuestras palabras clave y conectores o predicados de las mismas. A continuación, se muestra la representación del conocimiento en una red semántica.

Representación del diseño de la aplicación móvil

La siguiente imagen muestra la relación de la aplicación con las diferentes entidades como es el usuario y el modelo de datos.

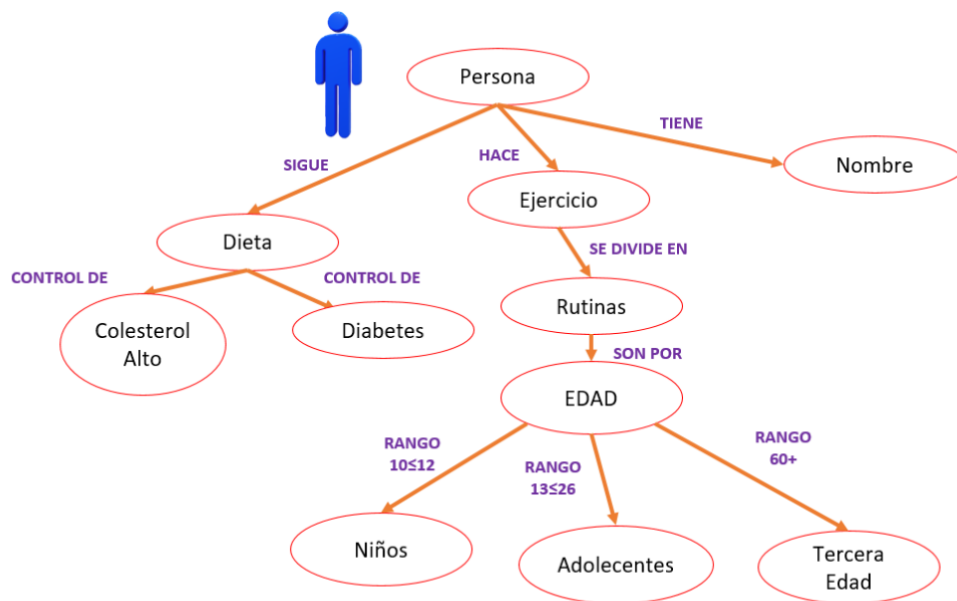


Representación del conocimiento de una aplicación móvil 3.3.

Representación de la Persona.

El siguiente gráfico muestra las características y componentes con las que cuenta una persona. El objetivo es describir el perfil de usuario, con el propósito de que se pueda utilizar la medicina del deporte.

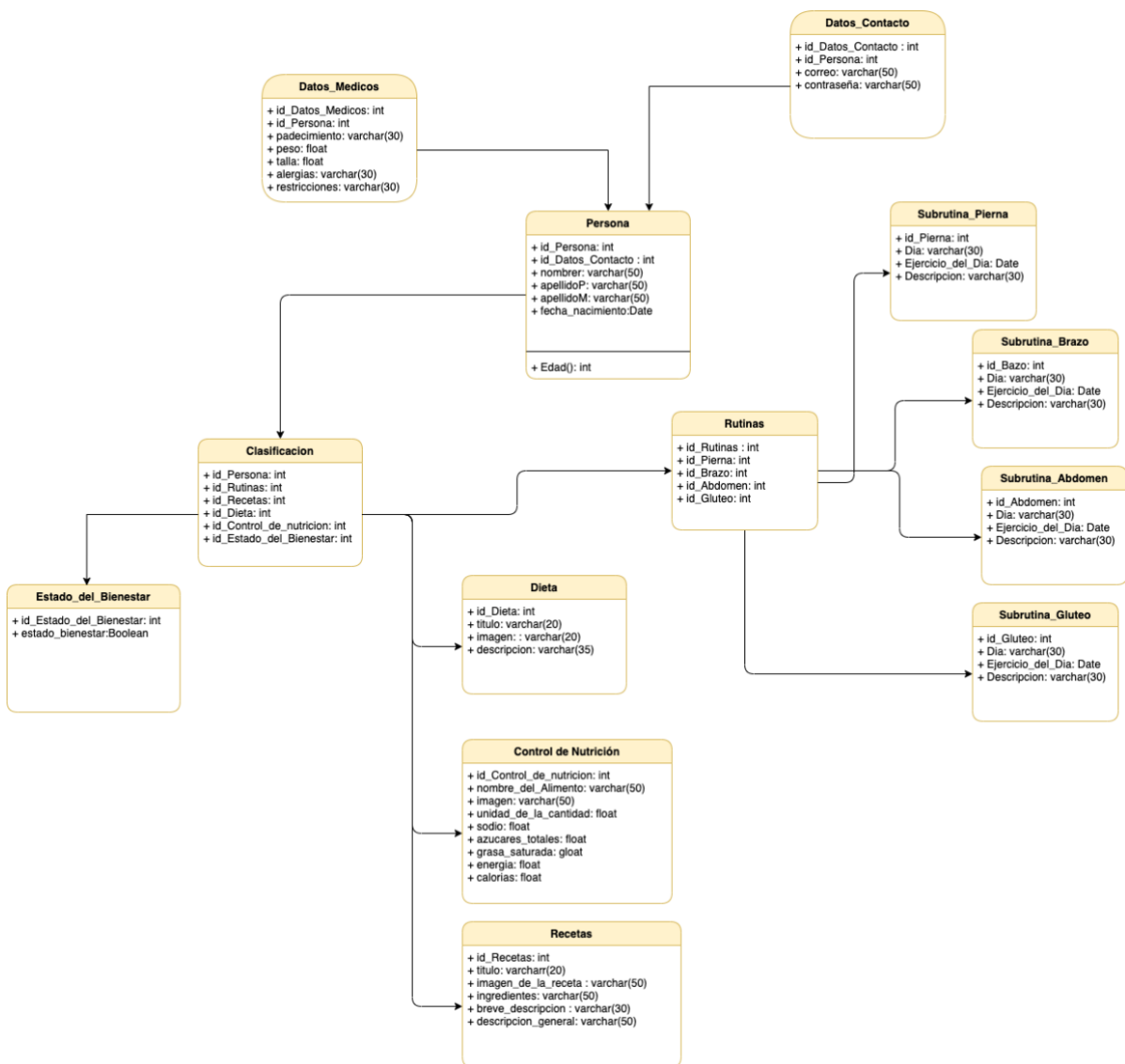
El siguiente grafo describe las características y componentes con las que cuenta nuestro sistema desarrollado en función de una persona, donde se describen atributos y varias acciones con las que puede interactuar una persona dependiendo de su peso, padecimiento o simplemente el ejercicio que desea realizar.



Representación del conocimiento de una persona 3.4.

3.6 Diseño de la base de datos

El diseño de la base de datos es importante, ya que podemos entender y mostrar cuales son nuestras entidades, así mismo los atributos que contienen cada una de ellas y el cómo se relacionan para poder interactuar entre sí o con el sistema. A continuación, se muestra el diagrama utilizando base de datos mediante un modelo de entidad relación aplicando la tercera forma de normalización.



Modelo de base de datos 3.5.

3.7 Diseño de la aplicación

Pantalla de inicio.

En las imágenes (Figura 3.7.1 y 3.7.2) se muestra la vista de pantalla de inicio.

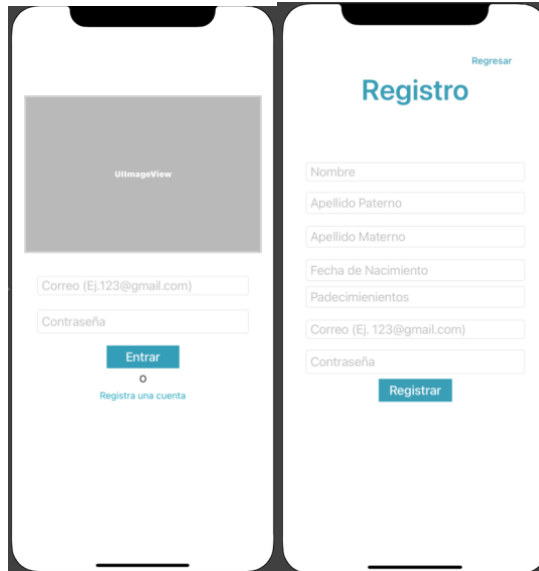


Figura 3.7.1 y Figura 3.7.1.1.

Menú principal.

En esta opción se muestra un menú, se puede elegir entre hacer ejercicio o nutrición.

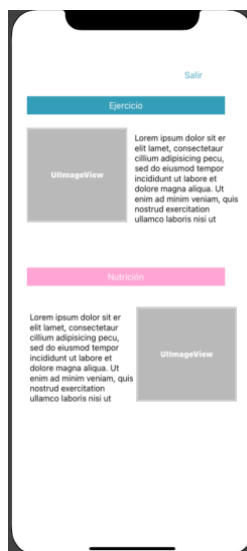


Figura 3.7.2.

Vista de ejercicio.

En la Figura 3.7.3 se muestran las opciones de ejercicio que el usuario puede escoger.



Figura 3.7.3.

Rutinas.

En la opción de rutinas, se puede escoger cualquier rutina dependiendo de qué se quiere realizar.

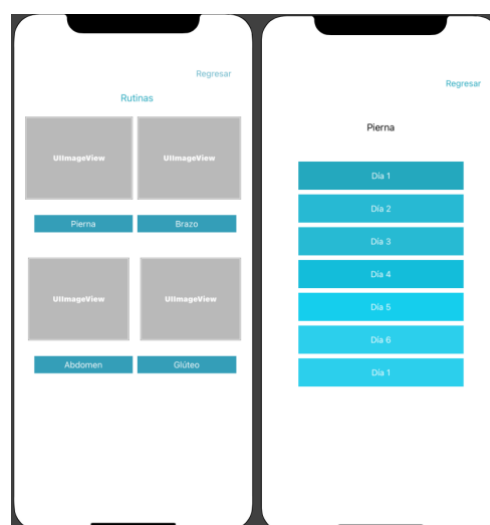


Figura 3.7.3.1 y Figura 3.7.3.1.1.

Comparación.

En la siguiente figura, se puede apreciar la fotografía que se tomó el usuario, y del otro lado el ejercicio que se está realizando para así saber si se está haciendo bien los ejercicios.



Figura 3.7.3.2.

Consejos.

Para la siguiente imagen, se puede observar que cuenta con la opción de poder adquirir diferentes consejos o tips que serán de ayuda para el usuario.

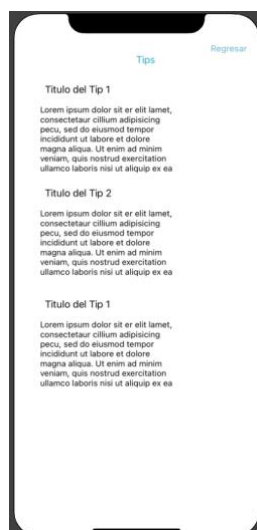


Figura 3.7.3.3.

Nutrición.

Para la siguiente imagen, se puede observar que hay diferentes opciones que el usuario puede elegir en función de la nutrición.

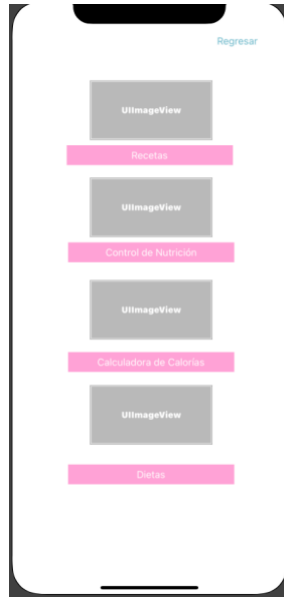


Figura 3.7.4.

Recetas.

Para la siguiente imagen, se puede observar que el usuario puede visualizar diferentes recetas, las cuales al momento de seleccionar tendrá las opciones de preparación.



Figura 3.7.4.1.

Control de nutrición.

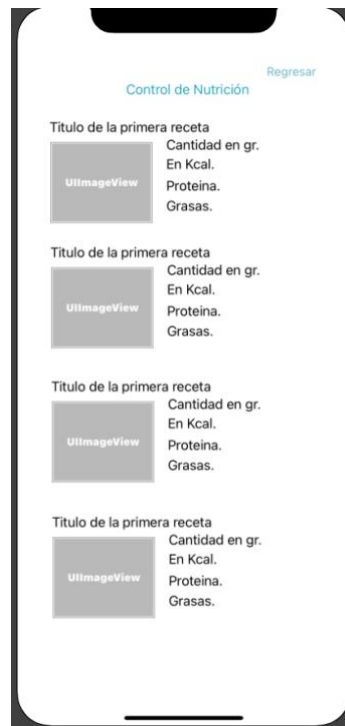


Figura 3.7.4.2.

Calculadora de calorías.

En esta venta, el usuario puede contemplar las calorías que puede consumir diariamente.

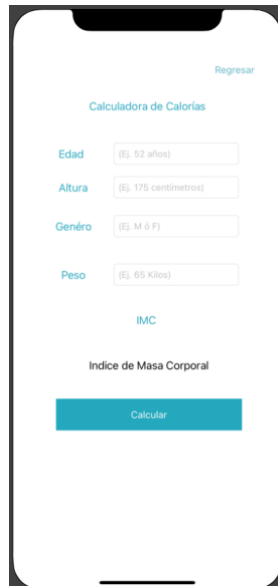


Figura 3.7.4.3.

Dieta.

Para esta venta, el usuario cuenta con diferentes opciones dietas y así escoger una.

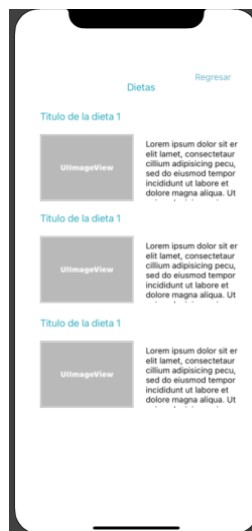


Figura 3.7.4.4.

Capítulo IV. Implementación

En este capítulo abordaremos la implementación del sistema con base a lo planteado en los capítulos anteriores. El desarrollo de nuestro sistema se divide en dos partes, la parte del “back end” y el “front end”, en cuanto al back, nos enfocaremos en la implementación de la base de datos, el código de la aplicación, el entrenamiento del sistema con machine learning y en cuanto al front end, nos enfocamos hacia el área de “UX-UI” orientado al diseño y la experiencia del usuario. Así mismo juntando todos estos componentes podremos lograr la implementación y el funcionamiento de nuestro sistema en óptimas condiciones.

4.1 Implementación de la base de datos

Una base de datos como sabemos es un repositorio organizado de datos almacenados en un equipo de computo. Que es controlada por un sistema de gestión de base de datos (SGBD). Para nuestra aplicación utilizaremos el gestor “DB Browser” [36] que como lo mencionamos es una herramienta para construir una base de datos por lo cual utilizaremos este gestor para el desarrollo de la base de datos de nuestra aplicación. La tecnología de cómputo determinada para nuestra aplicación es SQLite debido a que es un lenguaje nativo del sistema operativo iOS y es el más óptimo en cuanto al almacenamiento y recuperación de datos en las aplicaciones móviles.

A continuación, mencionamos el procedimiento a seguir para poder llevar a cabo la implementación en la aplicación móvil:

- Ingresar a la página oficial de SQLite en la siguiente url: <https://sqlitebrowser.org/>
- Nos ubicamos en la sección que dice “Download” y elegimos la versión que corresponde a nuestro sistema operativo y procedemos a la instalación. En nuestro caso fue MacOS.

Una vez que se ha instalado correctamente la aplicación, procedemos a construir una base de datos, abriendo el gestor de base de datos como se muestra en la siguiente imagen:



Figura 4.1.1

- Al momento de abrir nuestro SGBD, seleccionamos la opción de “New Databases” para crear nuestra base de datos. En la siguiente figura podemos observar nuestro entorno de trabajo de SQLite (Ver figura 4.1.2).

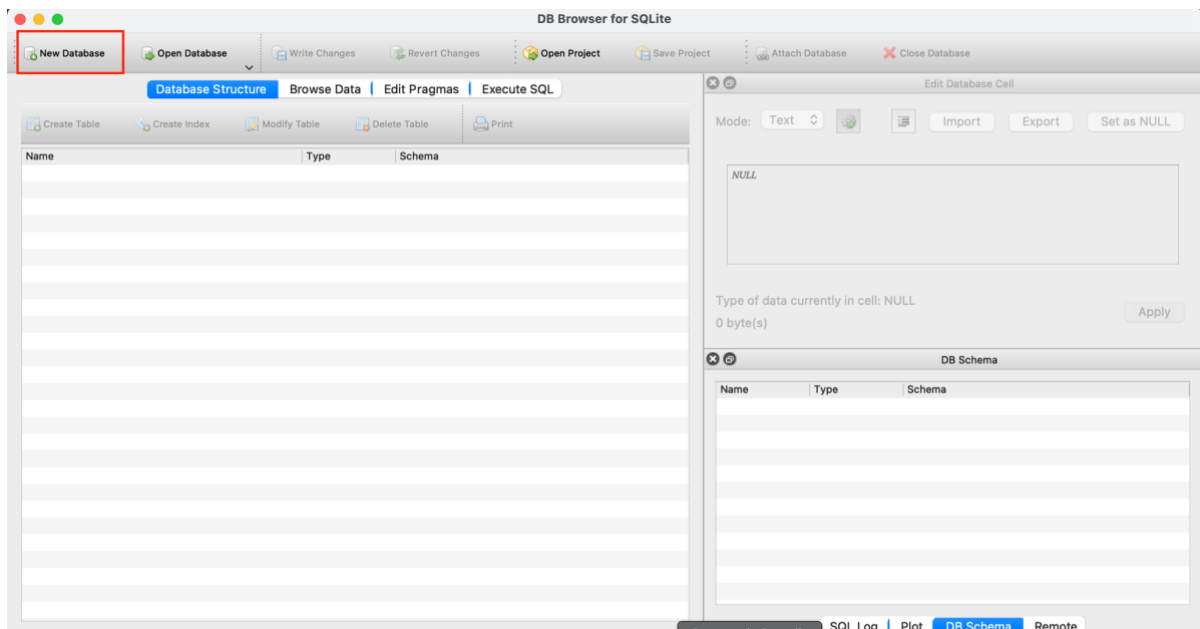


Figura 4.1.2

- A continuación, nos despliega un menú de opciones y nos pide llenar unos campos tales como son el nombre que le daremos a nuestra BD y la ubicación donde se guardará. Es importante señalar que la base de datos debe de llevar el nombre ya sea igual o similar al proyecto del cual estamos desarrollando para evitar confusiones y así mismo localizarla en una carpeta que sea exclusiva para el proyecto.

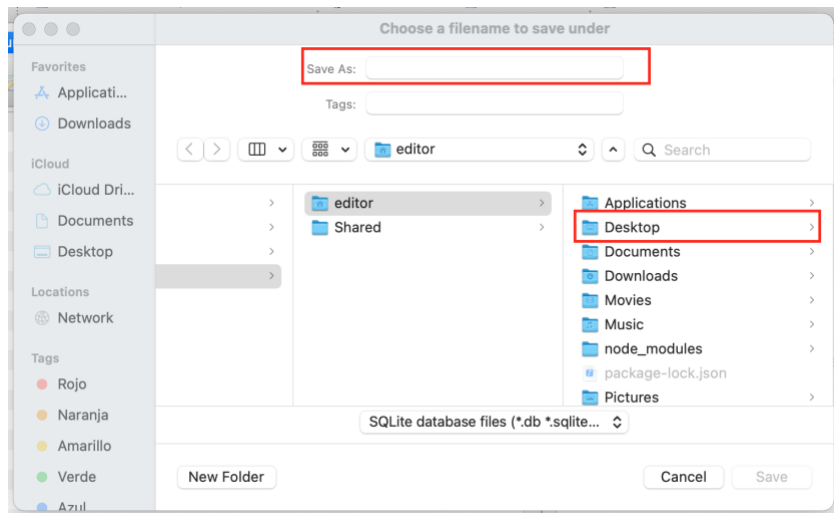


Figura 4.1.3

- Una vez guardados los cambios nos generará una base con el nombre que nosotros asignamos a ella como se muestra en la figura 4.1.4, en la cual podremos trabajar durante todo el proceso.

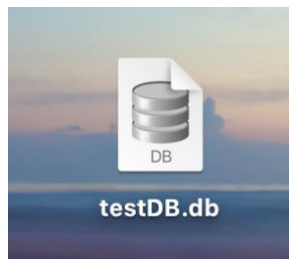


Figura 4.1.4

Procedemos a abrir nuestra BD previamente generado en la cual se puede crear las tablas definidas en la etapa de diseño del diagrama entidad relación. Para poder llevar a cabo la creación de nuestras tablas existen dos maneras de hacerlo, puede ser por medio de interfaz gráfica o mediante la ejecución de consultas “query’s”. Para este trabajo se optó por la interfaz gráfica que nos ofrece una manera de realizar un estudio analítico de la base de datos.

Seleccionamos la opción “Create Table” como se puede ver en el rectángulo café de la imagen 4.1.5:

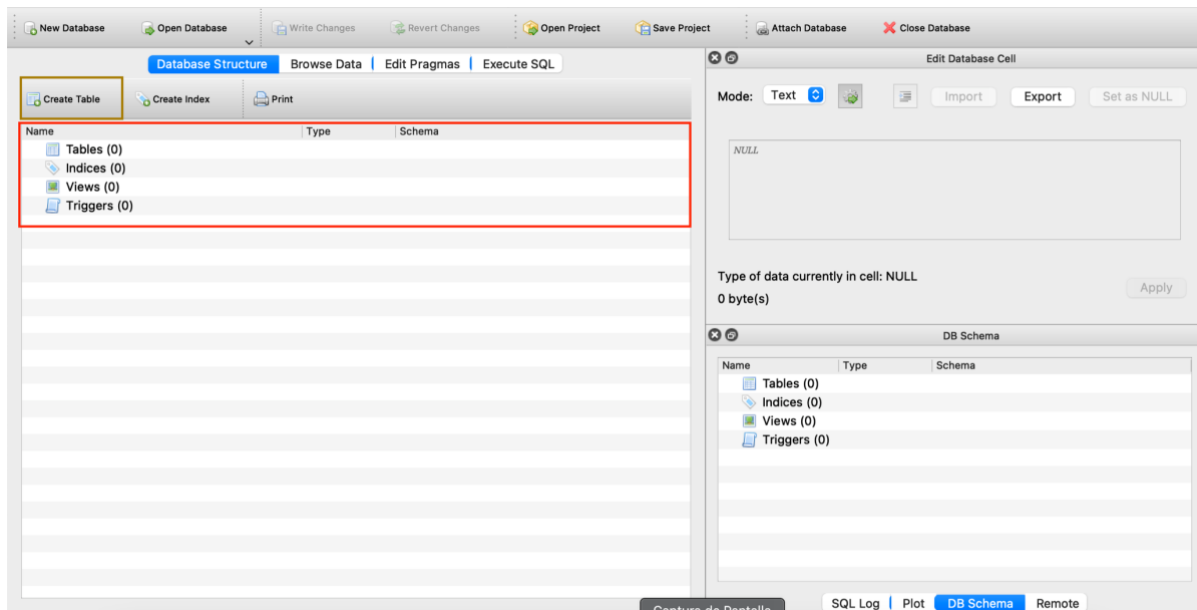


Figura 4.1.5

- Asignamos un nombre a la tabla, agregamos los atributos necesarios seleccionando la opción de “Add field”, definimos el tipo del campo y seleccionamos si es que queremos alguna otra característica. El sistema automáticamente en la parte de abajo te muestra el “query” que ejecutará para poder crear la tabla con todos los datos y características que has seleccionado. Posteriormente damos clic en Ok. En la figura observamos la creación de una tabla Persona que tiene los siguientes atributos: Que es un identificador numérico de persona, y los demás atributos son campos de texto (fecha de nacimiento, padecimiento).

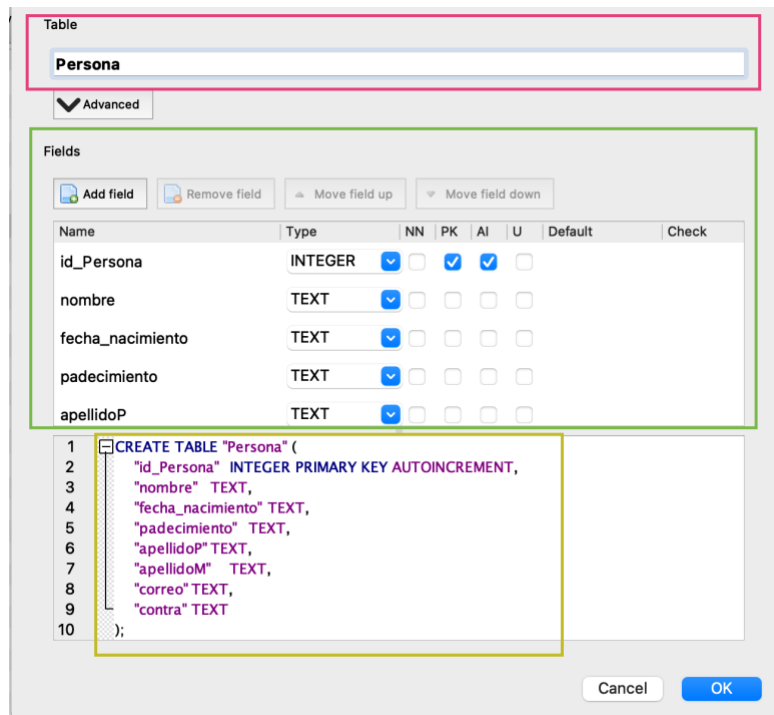


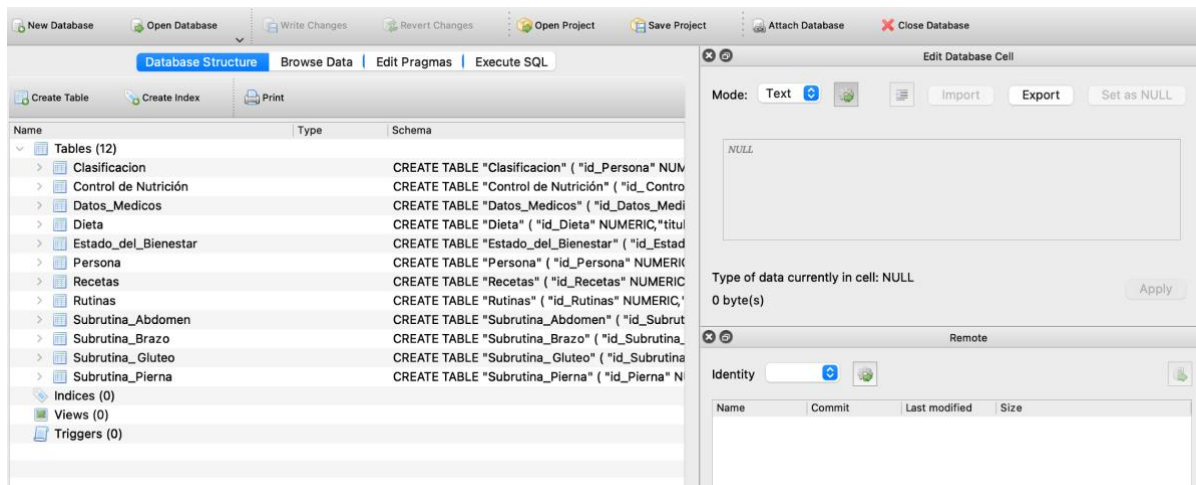
Figura 4.1.6

Para nuestro sistema nosotros utilizamos una base de datos relacional por lo que asignamos las llaves primarias (PK) y las llaves foráneas (FK) de cada tabla que tenemos en la base de datos para poder lograr la relación entre cada una de las tablas que creamos en nuestra BD. A las llaves primarias le agregamos también la característica de que sean auto incrementales (AI) esto para evitar repeticiones al momento de los registros y también para que cada usuario tenga su ID único que será su identificador dentro de nuestro sistema.

Para el aplicativo fue necesario la implementación de 12 tablas, tales como las que se muestran en la imagen del Modelo de base de datos implementada en 4.1.6, donde se pueden ver más a detalle en el anexo A, al final del documento.

Una vez concluida la creación de las tablas, observa que si usted se posiciona sobre Después de haber generado las tablas en nuestro sistema, se puede observar el detalle de nuestra BD solo basta con posicionarse en la sección "Database Structure" y ahí se mostrará todo lo que contiene.

Esta base de datos nos apoya en cuanto al registro de los usuarios y por consiguiente al “logueo” de ellos, así mismo nos ayudará a poder llevar un registro de cada uno de ellos y el poder brindarles una atención más personalizada de acuerdo a su perfil.



Modelo de base de datos implementada 4.1.6

4.2 Implementación de la aplicación

Una aplicación como sabemos, puede ser implementada en distintos lenguajes de programación y existen un sin fin de IDE 's para su desarrollo, existen entornos híbridos los cuales nos permiten desarrollar para ambos sistemas operativos móviles como es Android y iOS. En el caso de iOS, aunque se puede desarrollar en varios entornos, cuenta con su propio *framework*, es el que nosotros utilizaremos para la implementación de nuestro sistema, por lo tanto, será desarrollado en XCode y con el lenguaje nativo que es Swift.

A continuación, mencionaremos los pasos a seguir para poder realizar la implementación de la aplicación:

- Instalamos el programa XCode para Mac, para lo cual es necesario buscarlo en la AppStore¹ [37].
- Una vez instalado, abrimos el programa, como se muestra en la figura 4.2.1:



¹ Para poder instalar el programa de Apple Store, es necesario contar con un Apple ID, el cual se puede generar en: <https://appleid.apple.com/account>

Figura 4.2.1 Dock con la aplicación XCode Instalada.

- Al momento de ejecutarlo muestra la figura 4.2.2 y seleccionamos la opción de “Create a new XCode project”



Figura 4.2.2 Cuadro de Inicio de XCode.

- A continuación, se despliega un menú con un encabezado, donde se muestran distintas opciones, las cuales se pueden elegir según sean tus necesidades. En la sección de “Application” también es posible seleccionar el “Framework & Library”; en nuestro caso seleccionamos solo la opción que dice “App” y damos click en el botón Next. Esta opción nos permite crear una aplicación sin ningún componente extra.

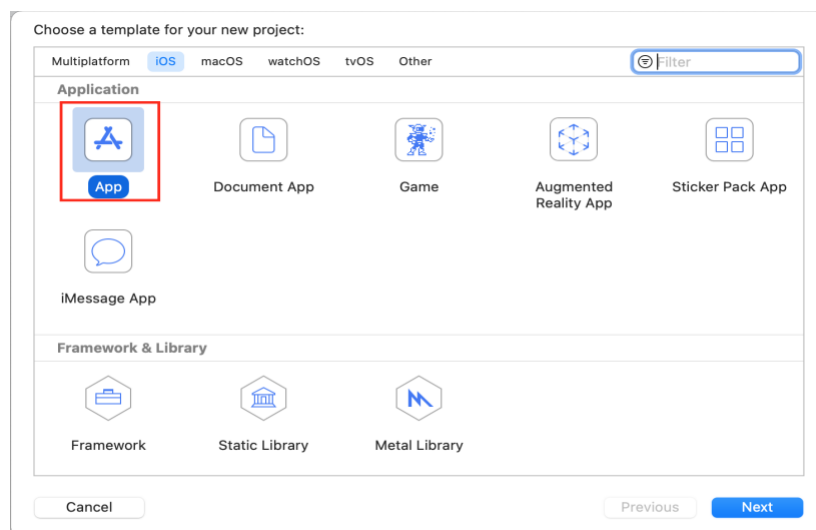


Figura 4.2.3 Cuadro de selección de plantilla de aplicación.

- Posteriormente nos va a mandar una ventana en la cual se puede seleccionar cuales son las opciones que necesitas para tu proyecto. Los datos que nos pide es el nombre del proyecto, el grupo, identificador de la organización, identificador de paquete, interfaz, tecnología de presentación de vistas utilizada y el lenguaje en el que se va a desarrollar el proyecto. Es importante seleccionar las opciones correctamente debido a que con las que se elija, serán con las que se desarrollará el proyecto. Una vez completados los pasos procedemos a dar clic en Next.

Choose options for your new project:

Product Name:

Team:

Organization Identifier:

Bundle Identifier: com.ProductName

Interface:

Life Cycle:

Language:

Use Core Data

Host in CloudKit

Include Tests

Figura 4.2.4 Opciones para un proyecto nuevo.

- Una vez concluidos estos pasos nos abrirá el editor en el cual podremos comenzar a crear el código que llevará la aplicación, por ejemplo:

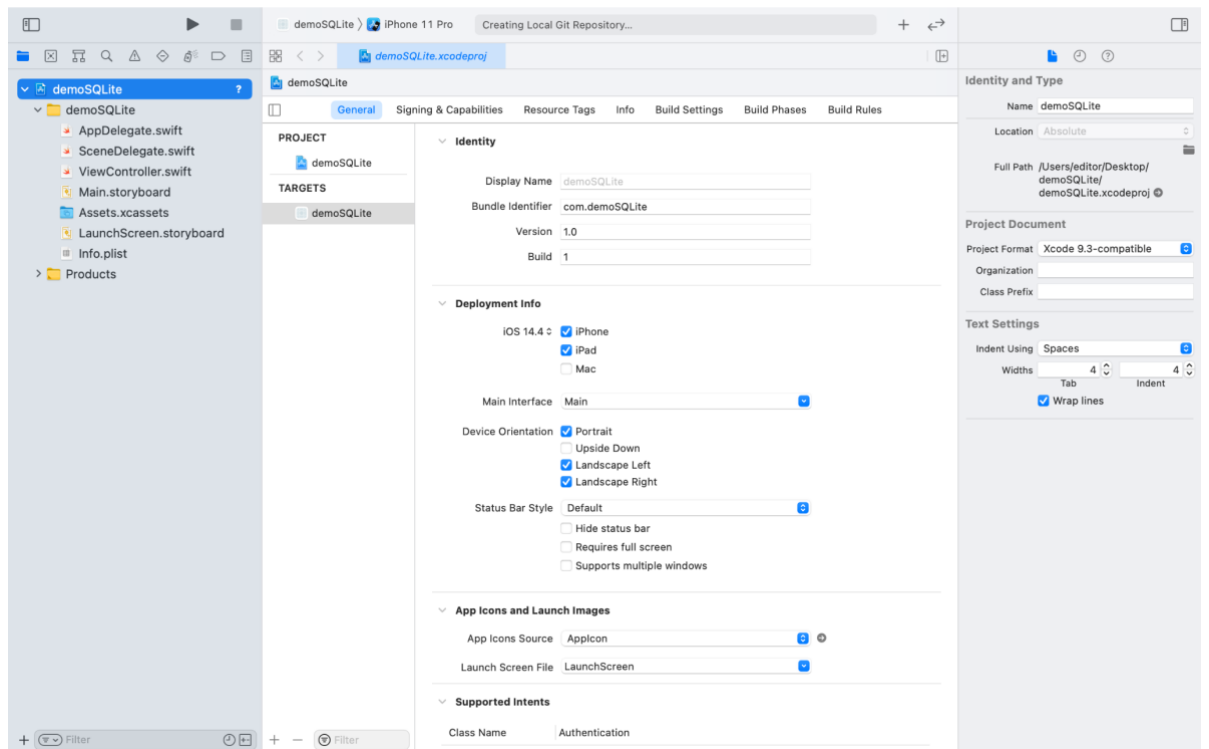


Figura 4.2.5 Opciones generales del proyecto.

Este IDE está basado en un modelo “MVC” (Modelo, Vista y Controlador), este patrón de diseño se basa en separar el desarrollo en tres partes bien diferenciadas.

El *modelo* lo componen los datos de la aplicación y la estructura de estos, en esencia es el esqueleto de la App. Estos son independientes de la vista [38].

La *vista* es lo que la persona que utiliza el sistema puede ver a través de la pantalla con la que interactúa. La vista permite interactuar con el modelo que es la representación de las entidades de la base de datos, pero siempre a través del controlador.

El *controlador* es el encargado de mantener conectados la vista y el modelo, básicamente es el que realiza todo el trabajo ya que la vista le dice al controlador cuando debe actualizar el modelo, es decir, los datos; el modelo comunica al controlador si ha sufrido algún cambio, de esta forma el controlador actualiza la vista.

- En la siguiente imagen podemos observar que nuestras vistas están en el archivo llamado “Main.storyboard” que está ubicado en la parte izquierda de nuestro proyecto, en el grupo de carpetas y archivos. Por lo que al momento de seleccionarlo nos mostrará en la parte central el “Storyboard” cómo lo podemos ver en Figura 4.2.6:

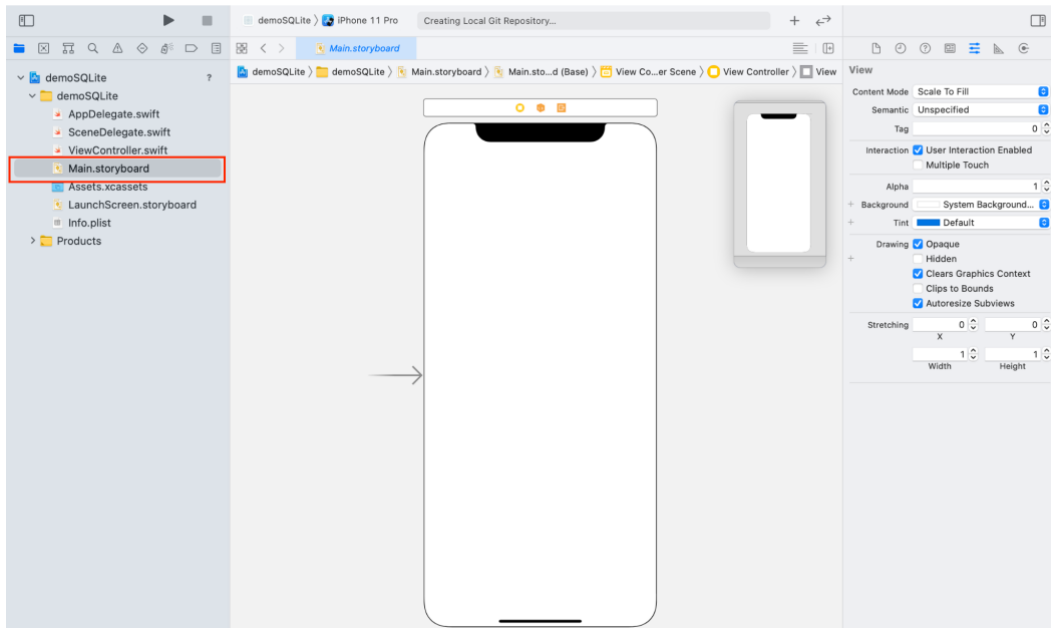


Figura 4.2.6 Storyboard del Proyecto.

- Una vez en el storyboard para implementar el diseño del aplicativo del capítulo anterior, se cuenta con una gran variedad de objetos que Apple ha creado, en los cuales solo es necesario seleccionar el deseado y se creará en el storyboard, en la figura 4.2.7 se puede observar algunas de las opciones disponibles.

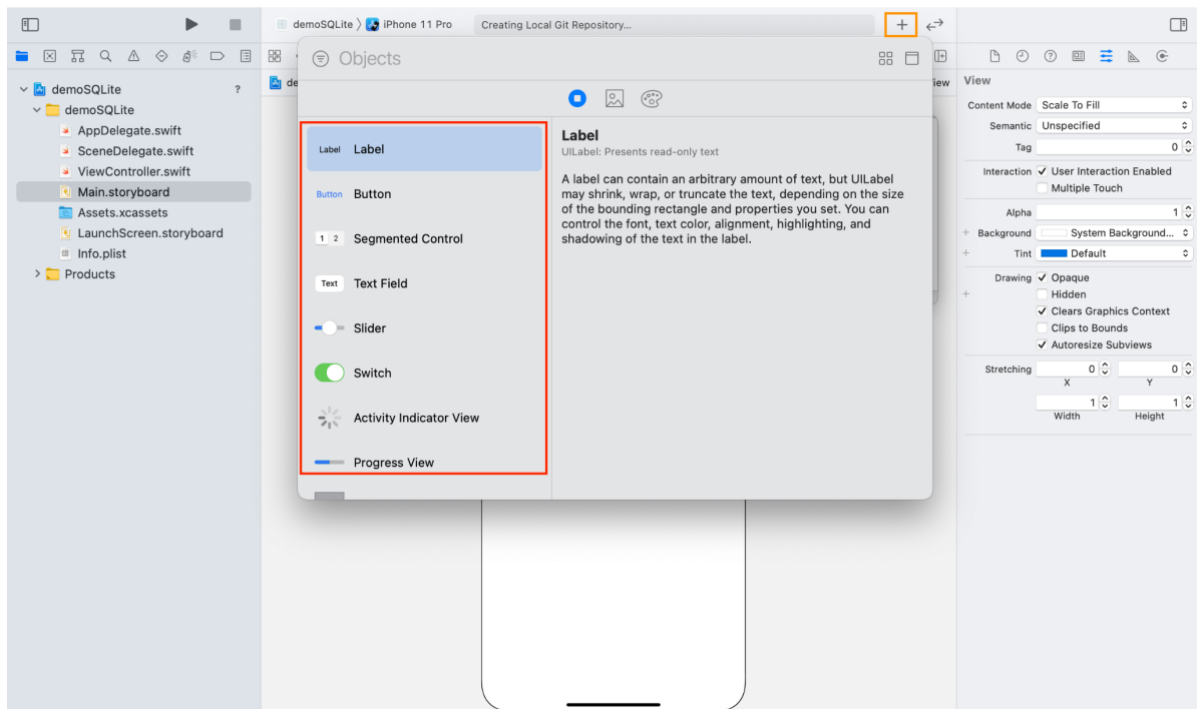


Figura 4.2.7 Opciones de Objetos (Inspector de Objetos).

- Las imágenes de la App, así como el icono y el “LaunchImages” (imagen que se muestra al arrancar la aplicación), se deben añadir a una carpeta llamada “Assets.xcassets”, está situada en el grupo de carpetas y archivos, y si se hace clic en ella puedes ver su contenido como se muestra en el recuadro amarillo de la figura 4.2.8, como podemos observar tenemos 4 carpetas y 4 imágenes fuera de ellas.

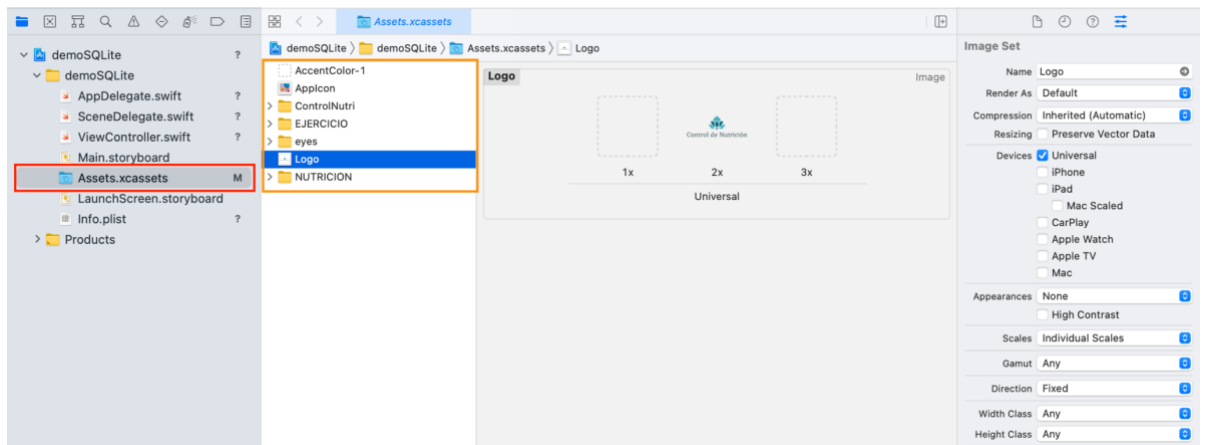


Figura 4.2.8 Assets del Proyecto.

- Comenzaremos implementando la vista llamada “Login”, para esto es necesario seleccionar y arrastrar diferentes objetos, tal es el caso de una imagen donde se encuentra el logo de la aplicación, dos “Text box” que es donde los usuarios podrán ingresar sus datos, uno es para el correo y el otro para la contraseña respectivamente y dos botones, uno para poder entrar al sistema y el otro para poder registrarse si es que no se tiene una cuenta, lo podemos observar en la figura 4.2.9

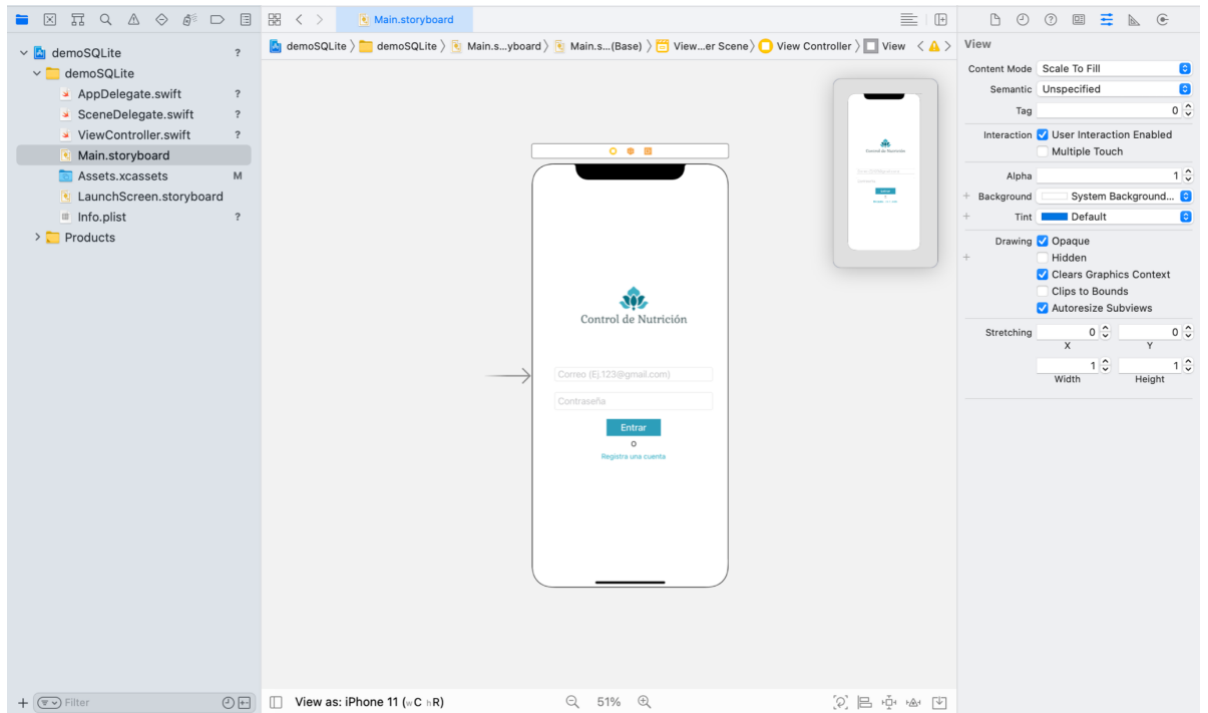


Figura 4.2.9 Login.

- Una vez creada la vista de “Login” se crea la vista de “Registro”. Para poder realizar esto se tiene que crear un nuevo objeto de tipo “View Controller”, este objeto se crea de manera similar a un botón o un label, se busca por el nombre del objeto (rectángulo amarillo) y se muestra en la lista de objetos (rectángulo rojo) así como se muestra en la figura 4.2.1.1

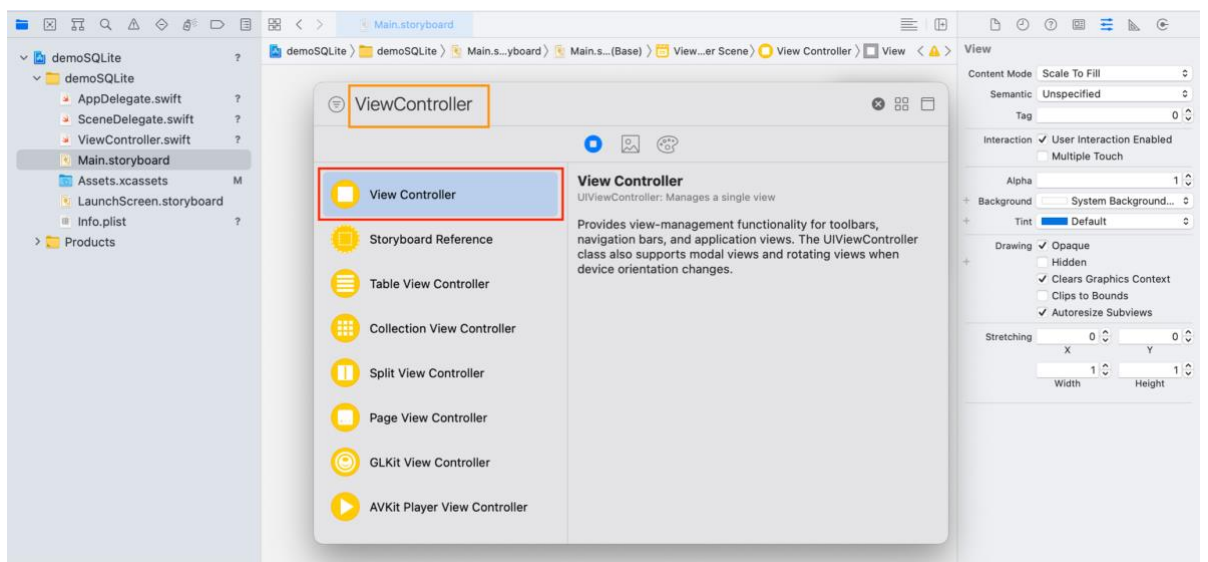


Figura 4.2.1.1

- Por lo que arrastrando el “View Controller” junto con varios objetos más, los cuales son necesarios para nuestra vista como los son un botón y respectivos “Text boxes” se tiene la siguiente Figura 4.2.1.2:

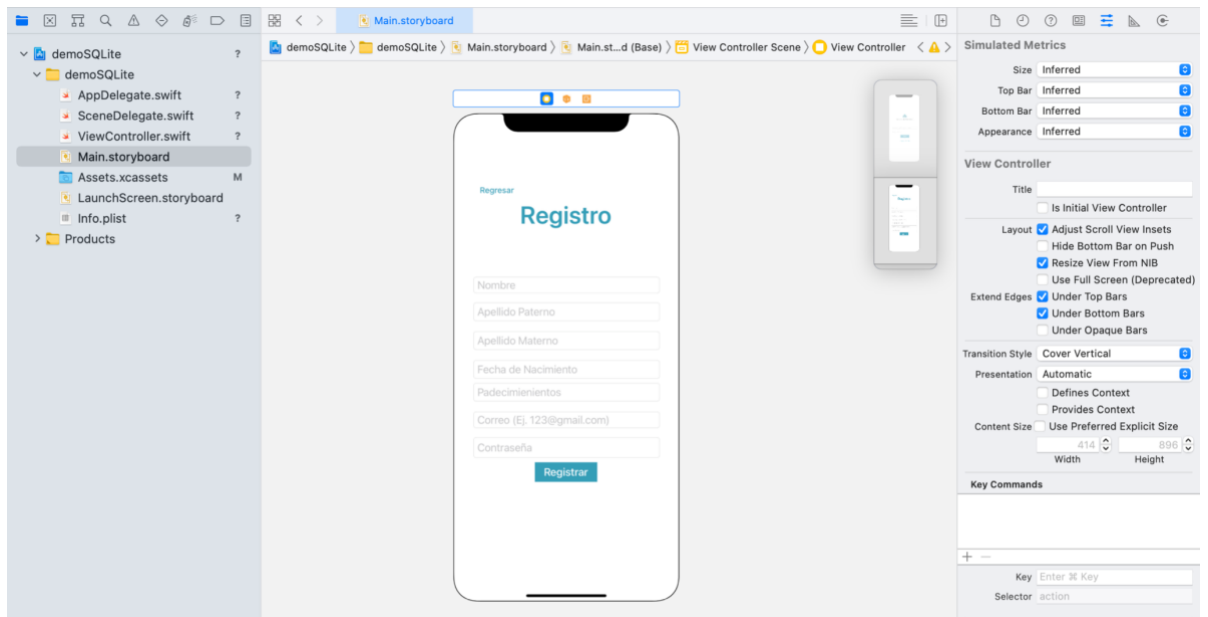


Figura 4.2.1.2 Registro

- Ahora bien, para conectar ambas vistas, tanto la de “Login” como la de Registro a través del botón “Registrar”, hay que hacer clic sobre este mismo botón y pulsando “ctrl”, sin soltar el clic se arrastra el cursor a la otra vista (Registro) y dar clic en donde dice “show”, como se puede observar en la siguiente figura:

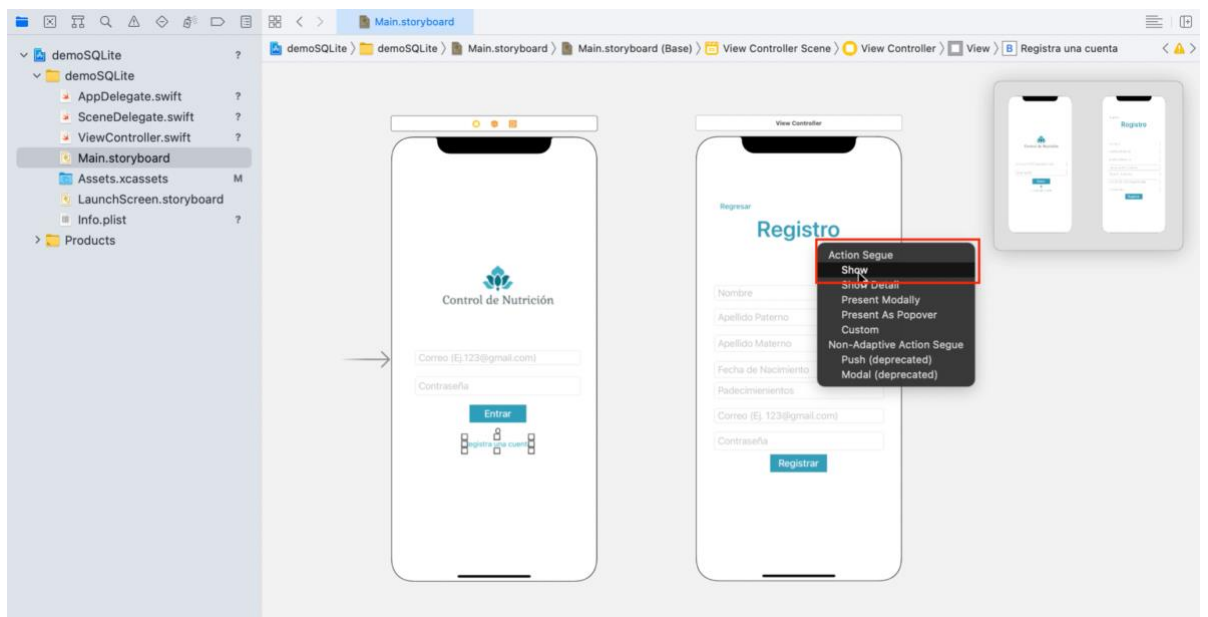


Figura 4.2.1.3 Conexión de Vista de Login con Registro.

- Así mismo, se debe crear una conexión bilateral, ya que por el momento tenemos una vista relacionada con solamente un sentido, que va de Login a Registro, tal es el caso de:

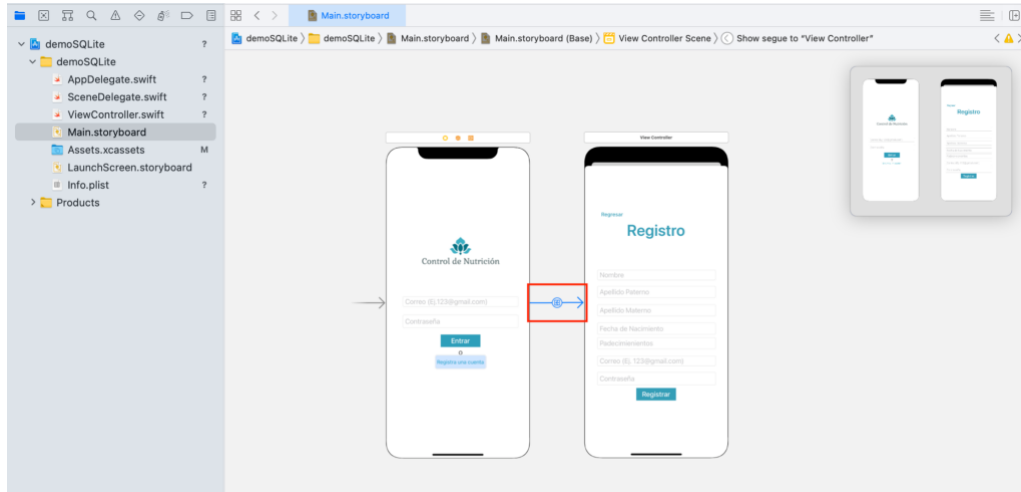


Figura 4.2.1.4 Conexión de un solo sentido.

- Posteriormente para generar la conexión bilateral, el botón que se encuentra en la vista de registro, llamado “Registrar”, hay que hacer clic sobre este mismo botón y pulsando “ctrl”, sin soltar el clic se arrastra a la otra vista, para así generar un flujo de “Login”- “Registro” - “Login”.

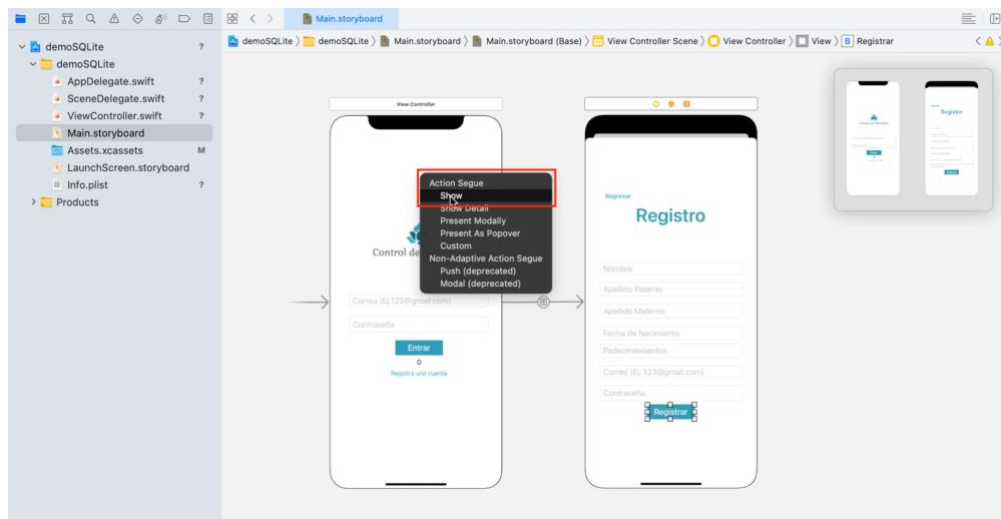


Figura 4.2.1.5 Conexión de Vista de Registro con Login.

- Una vez realizado este paso se puede observar la asociación de ambas vistas, como lo podemos observar en la siguiente imagen:

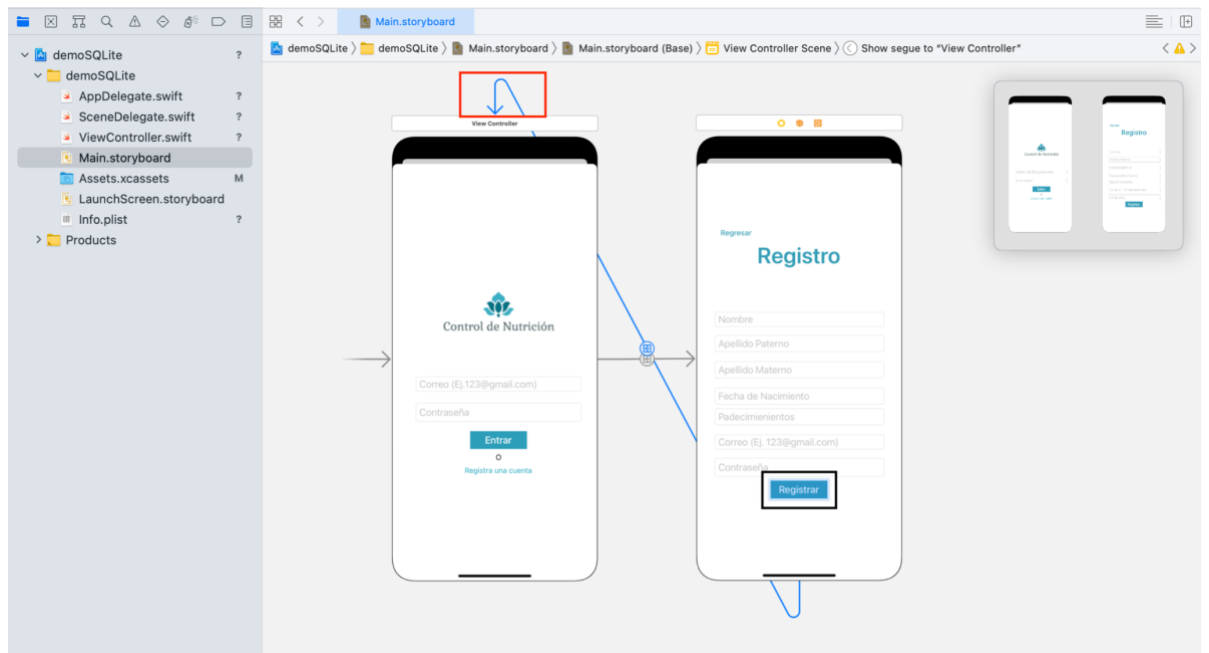


Figura 4.2.1.6 Conexión de ambos sentidos.

Las demás imágenes implementadas de este aplicativo se pueden ver más detalladas en la sección de anexos, específicamente en el Anexo B.

4.3 Implementación del modelo de machine learning

Para la implementación del modelo de machine learning es importante verificar cuales son las necesidades de nuestro proyecto ya que hoy en día existen muchos modelos los cuales tienen diversas funciones. Para poder llevar a cabo el entrenamiento es necesario proporcionar datos de los cuales pueda aprender el algoritmo. En nuestro caso utilizaremos la tecnología de CoreML.

A continuación, mencionaremos los pasos a seguir para poder realizar la implementación del modelo que ocuparemos en la aplicación:

- Para poder crear nuestro modelo, es importante mencionar que se requiere abrir la aplicación Machine Learning de Apple. La aplicación viene instalada, en la suite de herramientas de XCode, seleccionamos la aplicación Core ML como se muestra en la figura 4.3.1.



Figura 4.3.1 Dock con la aplicación de entrenamiento CoreML.

- Después de ejecutar la aplicación observamos el cuadro de diálogo que se muestra en la figura 4.3.2 donde nos solicita seleccionar la ubicación de nuestro proyecto para así después generar un nuevo documento en la ruta previamente seleccionada.

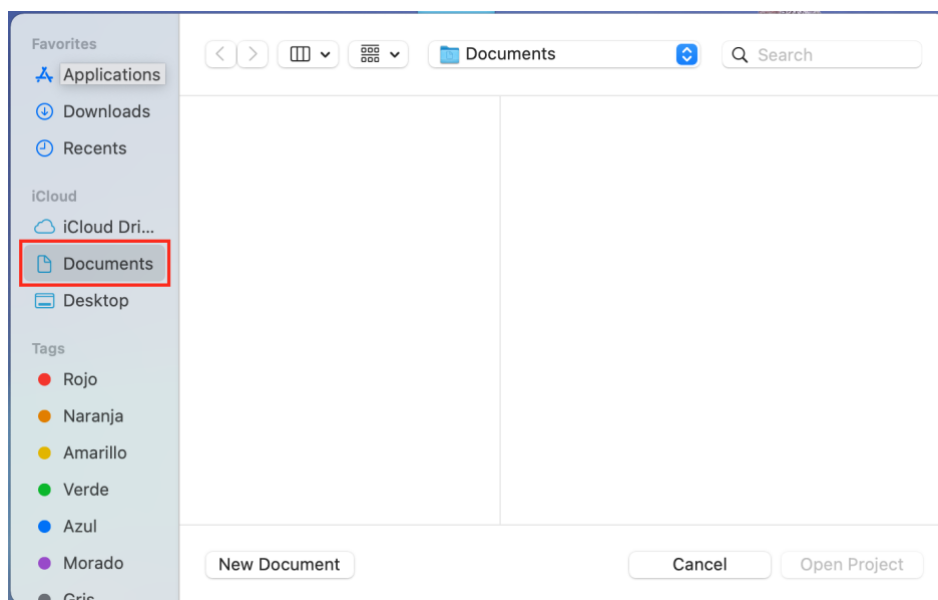


Figura 4.3.2

- Posteriormente nos mandará a la pantalla mostrada en la figura 4.3.3, así podremos elegir de qué tipo de modelo será según sea las necesidades del proyecto al cual vamos a implementar el modelo de machine learning, de lado izquierdo nos muestra las distintas agrupaciones de modelos que existen, si le damos en la opción “all” nos muestra todos los modelos de los cuales podemos hacer uso, existen diversos modelos como clasificación de imágenes, la detección de objetos, transferencia de estilos, clasificación de acciones, clasificación de actividades, clasificación de sonido, clasificación de texto, etiquetado de palabras, clasificación tabular, regresión tabular y recomendación. Para nuestra aplicación ocuparemos el modelo de “Image Classification”, por lo cual se selecciona y presionamos next.

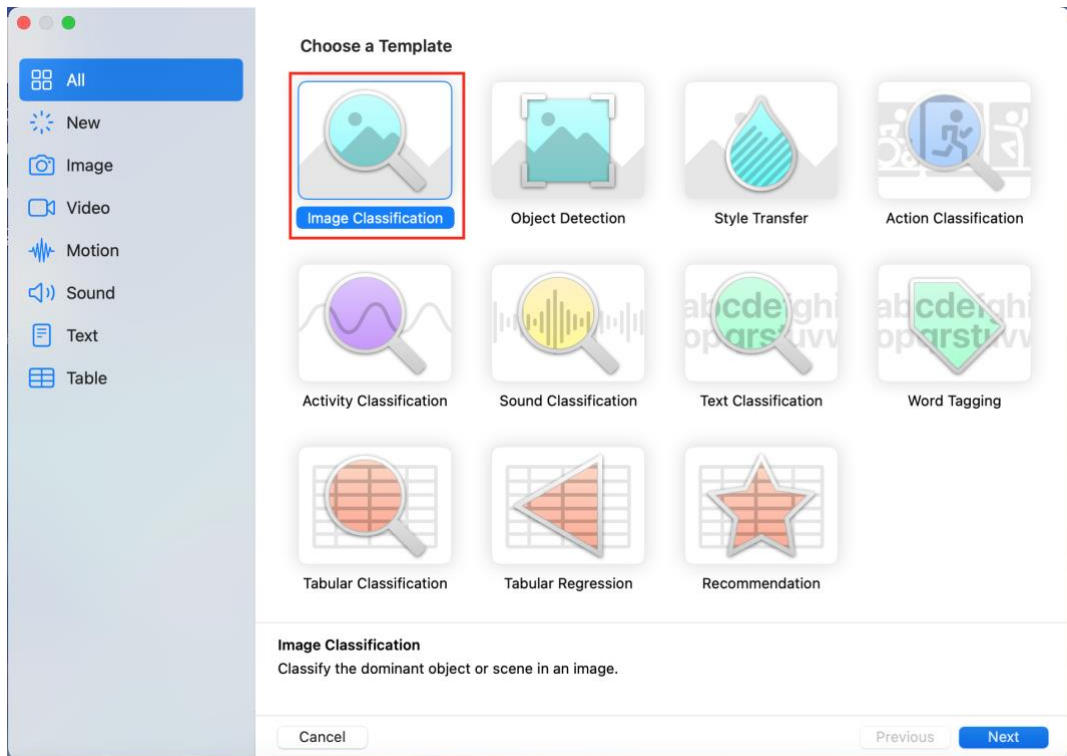


Figura 4.3.3

- Sucesivamente nos mostrará la figura 4.3.4 donde tenemos que asignar un nombre relacionado al proyecto, autor, licencia y descripción. En nuestro caso, el modelo que nosotros creamos lleva como nombre “Inceptionv3”. Una vez llenados los datos que nos piden seleccionamos la opción next.

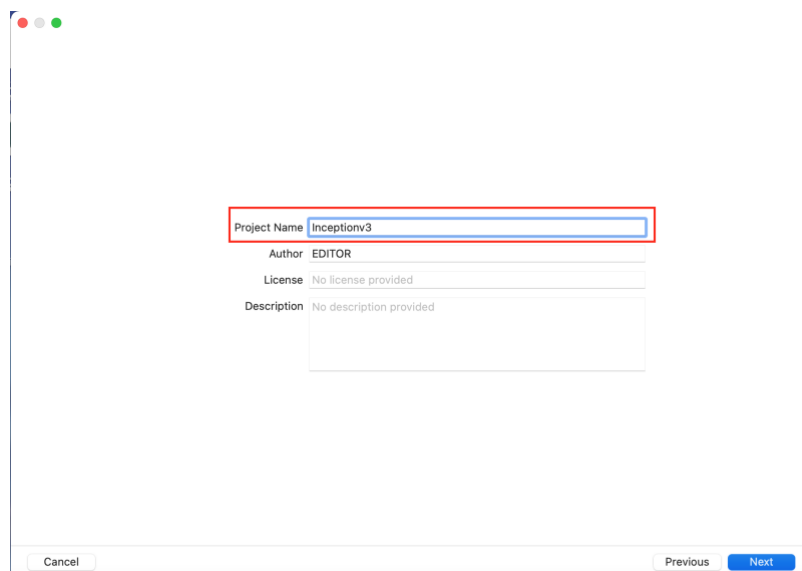


Figura 4.3.4

- Para concluir la creación del modelo, nos manda a la figura 4.3.5 donde se tiene que volver a asignar la ubicación donde se quiere guardar el modelo, después seleccionamos el botón “Create”.

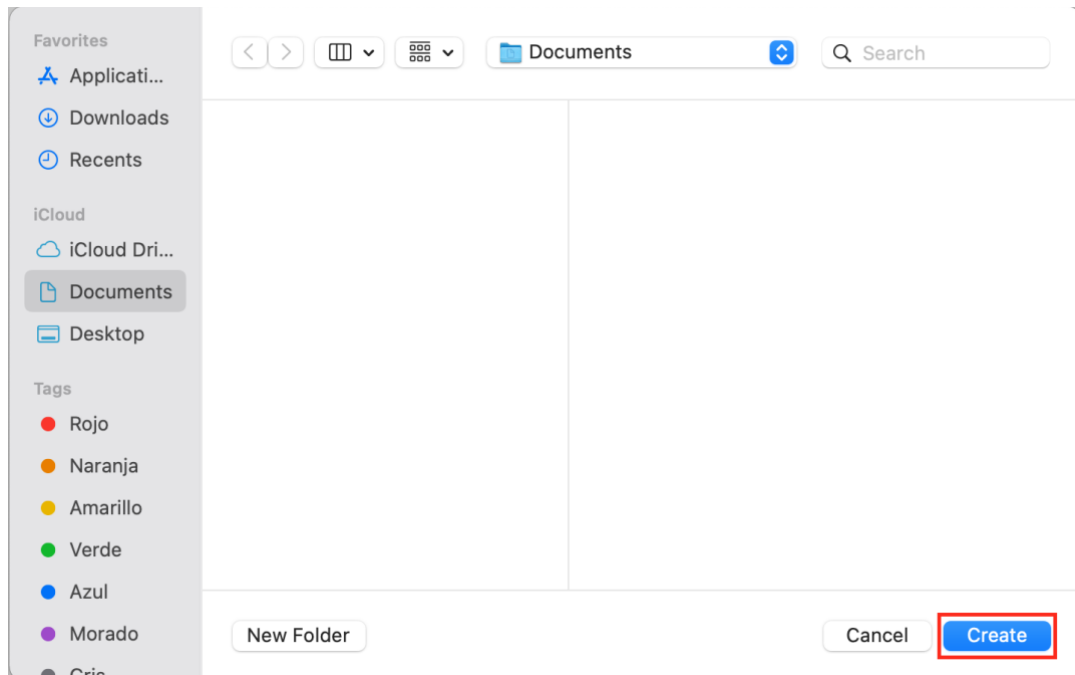


Figura 4.3.5

- Una vez concluido este proceso y previamente guardados los cambios nos genera un archivo con extensión “.mlproject” con el nombre que nosotros previamente asignamos como se puede ver en la figura 4.3.6. Con este proyecto trabajaremos en nuestra aplicación.

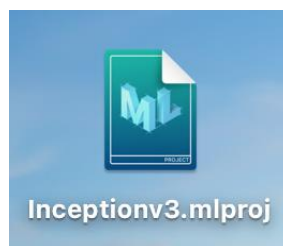


Figura 4.3.6

- Una vez creado el modelo con las características necesarias, al momento de abrirlo el sistema nos mostrará la siguiente pantalla como se muestra en la imagen 4.3.7, donde en la parte de arriba hay un encabezado de opciones que se pueden realizar, de lado izquierdo podemos visualizar en el proyecto que estamos trabajando.

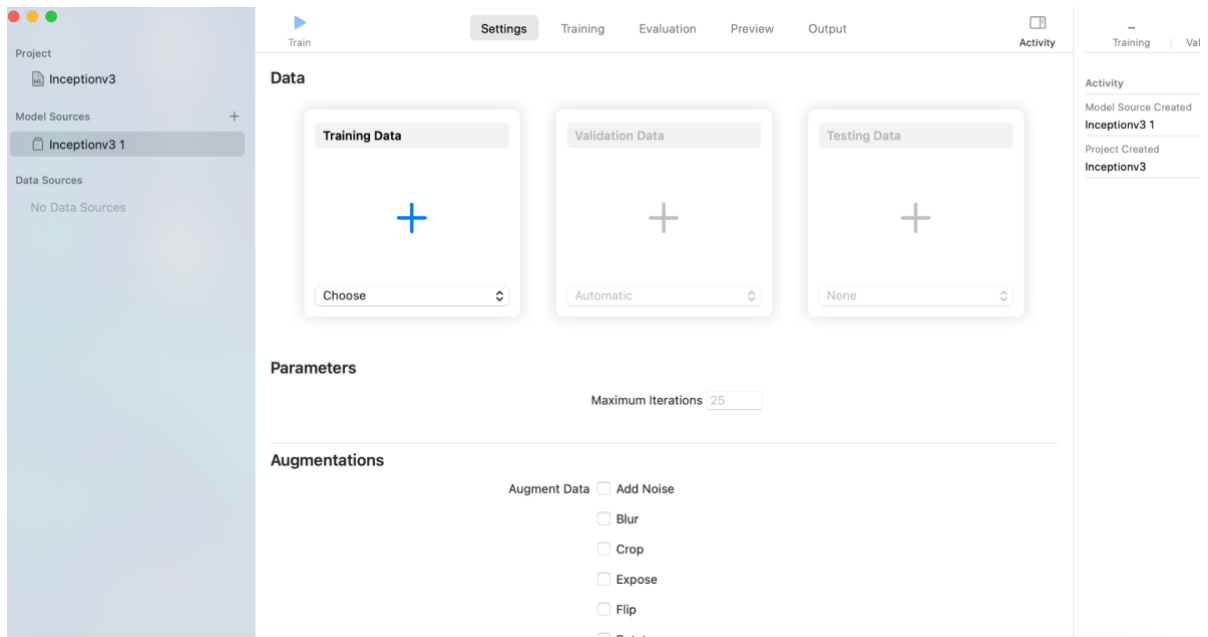


Figura 4.3.7

Como ya lo mencionamos previamente, CoreML cuenta con una gran variedad de modelos de aprendizaje automático la cual incluye redes neuronales, modelos de árboles, etc. [39].

- Para poder realizar el entrenamiento del modelo, lo que debemos de tener es un conjunto de imágenes relacionadas y categorizadas por tipo de ejercicio previamente tomadas y seleccionadas como se muestra en la figura 4.3.8.

Nosotros lo tenemos agrupado en una carpeta que lleva como nombre ejercicio, a su vez tenemos agrupados según la zona que se quiere trabajar, por ejemplo, abdomen, brazo, glúteo y pierna.



Figura 4.3.8 Imágenes de los ejercicios del cuerpo.

Es importante mencionar que las imágenes deben de contemplar ciertas restricciones ya que se debe contar con un cierto número de imágenes para así poder alimentar el modelo de machine learning. La Aplicación asigna automáticamente el número de imágenes requerido para entrenamiento, pruebas según en estándar 80%-20%.



Figura 4.3.9 Conjunto dividido en entrenamiento, prueba y validación.

- Una vez que se tienen las imágenes, se debe entrenar el modelo, para esto es necesario arrastrar las imágenes de la figura 4.3.8 a la sección de “Training data”, en caso de que no se quieran arrastrar, en esa misma opción se encuentra un botón de choose y así mismo las de “Testing” en su sección correspondiente, así como se muestra 4.4.1, posteriormente seleccionamos la opción de train.

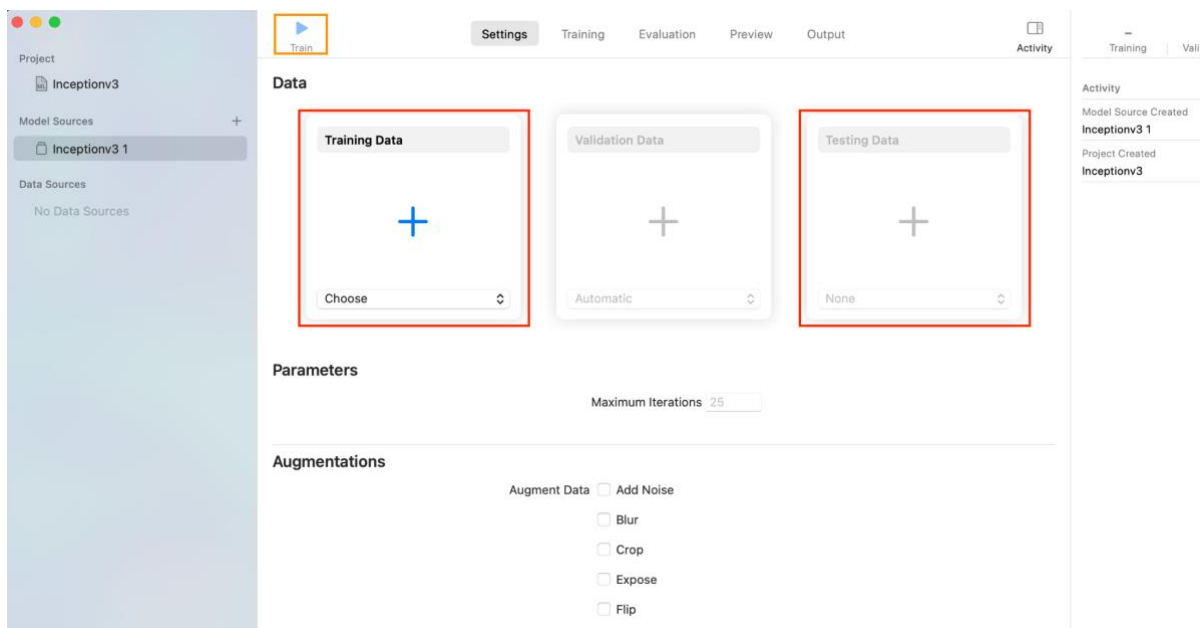


Figura 4.4.1 Nombre

- Una vez que se ha creado el modelo, podemos observar en el training data de la figura 4.4.2 que efectivamente se están mostrando las 4 clases correspondientes a las partes del cuerpo respectivamente, junto con su respectivo “Testing” donde al igual que el “Training Data” corresponde a 4 clases con menor numero imágenes (Es decir como sabemos, el conjunto de prueba siempre será menor) ; así mismo, se puede observar que se tiene un máximo de 5 épocas, las cuales están mostradas en la sección de parámetros que como lo comentamos esto no sirve para poder tener una mejor precisión.

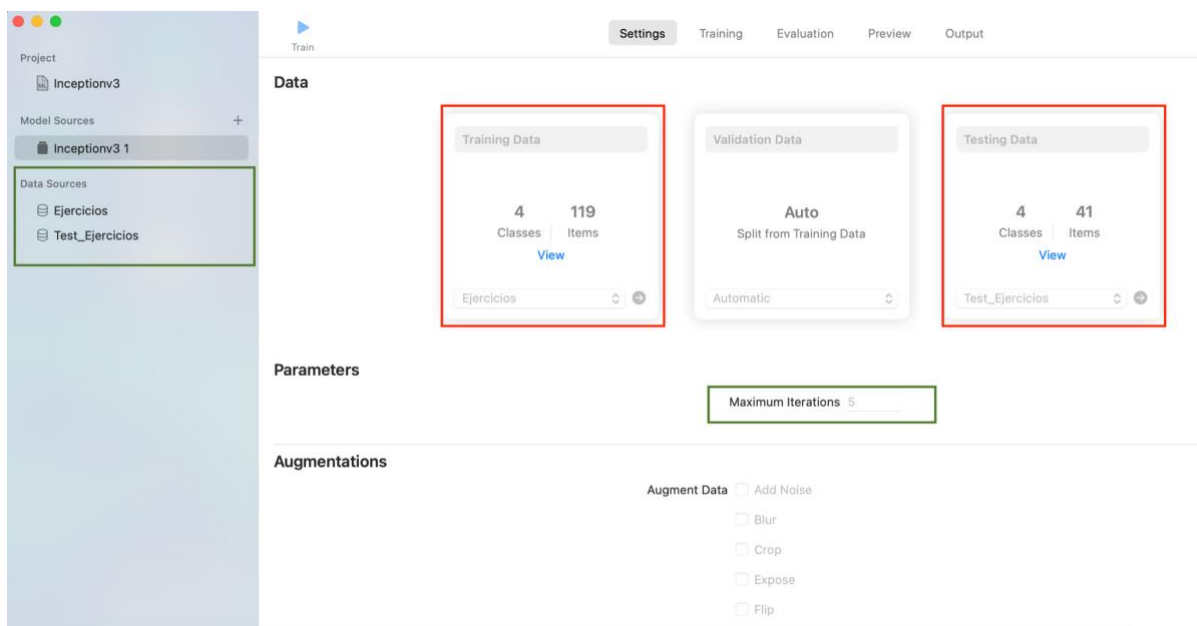


Figura 4.4.2 Categorías en CoreML

- Posteriormente, para poder exportar este modelo tenemos que dirigirnos a la pestaña llamada “Output”, la cual se encuentra seleccionada por un rectángulo rojo en la figura 4.4.3, después seleccionamos el botón “Get”, el cual está marcado con un rectángulo verde.

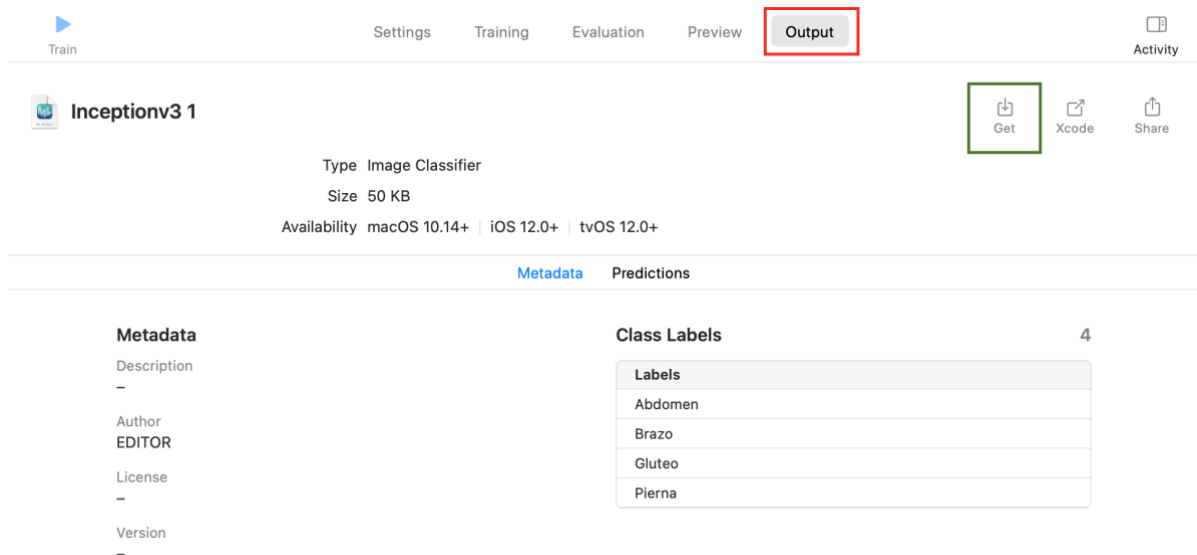


Figura 4.4.3 Exportación el modelo

- Una vez realizada la exportación nos pedirá guardar los cambios y nos genera un archivo con la extensión “.mlmodel” con el nombre que previamente asignamos, así como lo muestra la figura 4.4.4. Y con este archivo es con el que podremos trabajar con nuestra aplicación durante todo el proceso.



Figura 4.4.4

- Dado que ya se creó el proyecto en “Xcode”, procedemos a integrar el modelo de CoreML, para esto es necesario solicitar al sistema que otorgue los permisos para poder acceder en su galería a través de su archivo “Info.plist” como se muestra en el rectángulo naranja de la figura 4.4.5. Así mismo, para poder integrar el modelo ya creado, simplemente se arrastra el modelo “Inceptionv3” al proyecto de “X Code” previamente creado, como se muestra en la siguiente imagen:

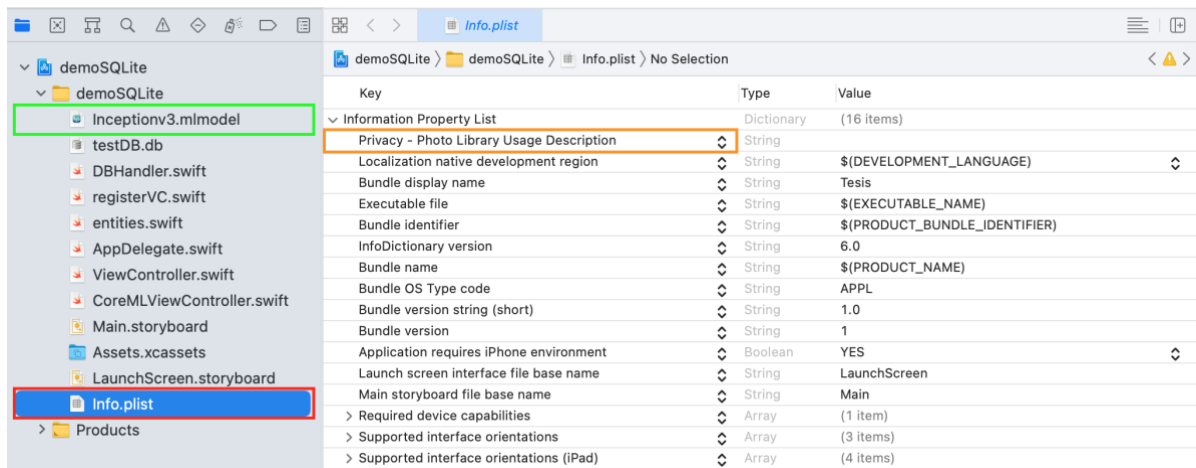


Figura 4.4.5

- Una vez que se tiene el modelo integrado en el proyecto, se debe de crear una clase de tipo Cocoa Touch Class, para esto solo es necesario dar “clic” derecho en el proyecto de “XCode”, particularmente en este caso es la carpeta “demoSQLite”, por lo que nos mostrará:

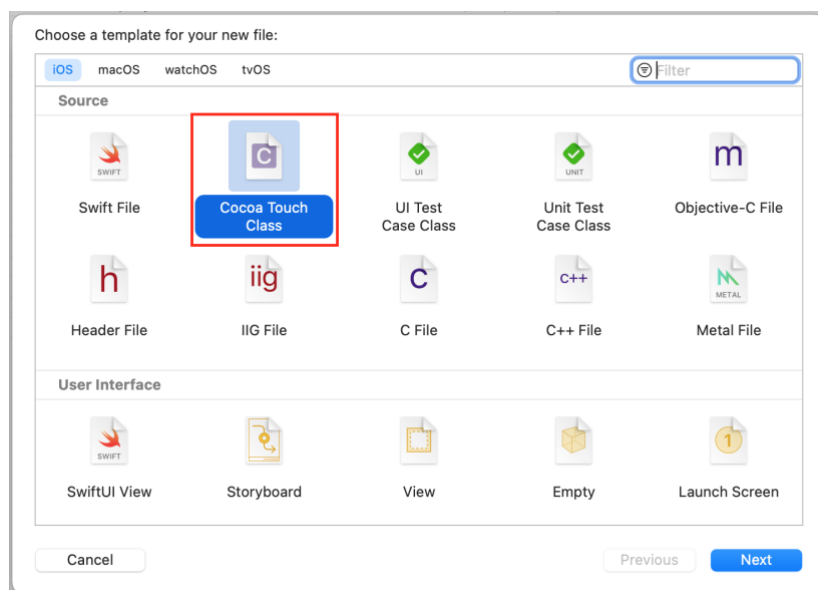


Figura 4.4.6

- Posteriormente nos desplegará una pantalla como se muestra en la figura 4.4.7, donde nos pide ingresar la clase, la subclase y el lenguaje en el que se va a desarrollar, para nuestro proyecto generamos una clase con el nombre de “CoreMLViewController” y después seleccionamos la opción de next.

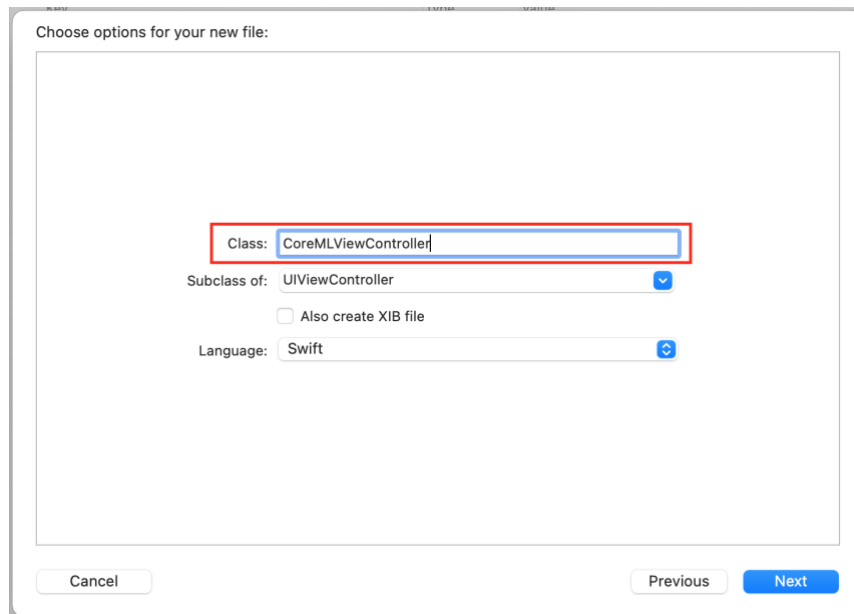


Figura 4.4.7

- Después de haber creado la clase observamos a la izquierda de la figura 4.4.8 en el rectángulo rojo que se ha añadido nuestra clase de manera correcta. Una vez que se han creado los recursos en "Main.storyboard" en el proyecto que estamos ocupando, se implementa la lógica de programación, se observa que se han creado dos objetos uno de tipo "UIImageView" llamado img y otro de tipo "UILabel" llamado ClasificaciónLb el cual está resaltado en un rectángulo naranja, integrando eso objetos simplemente arrastrandolos como se puede observar con las líneas verdes, también podemos observar que se ha creado un botón llamado "CamaraTap" que es donde se solicita el acceso de cámara.

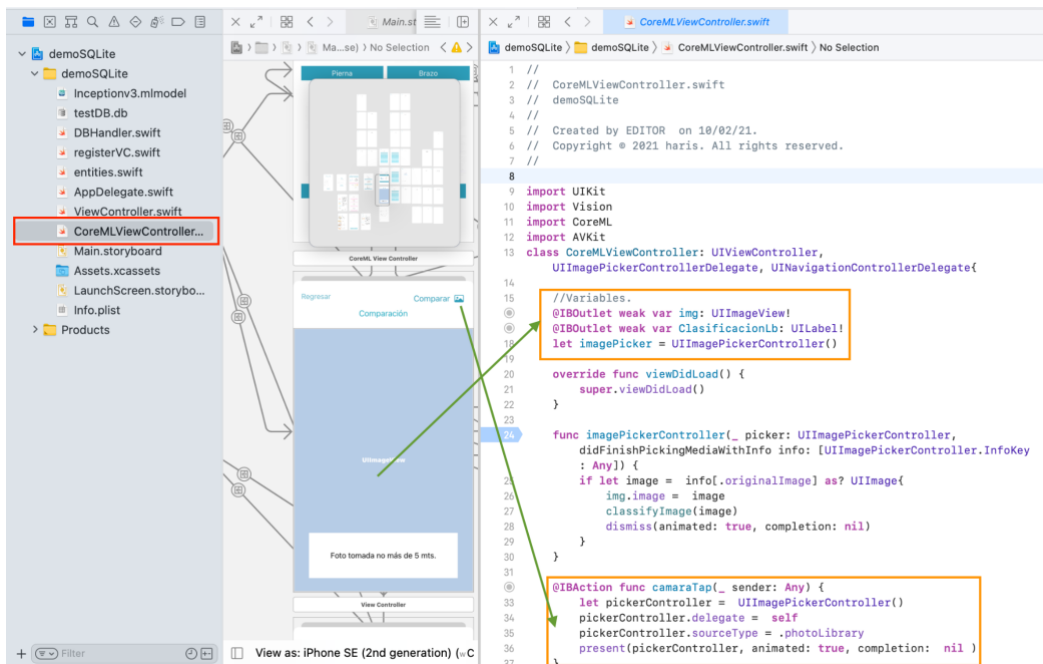


Figura 4.4.8

4.4 Implementación de Core Motion

En la actualidad Core Motion proporciona un sin fin de opciones en cuestiones de desarrollo, por ejemplo, se puede realizar notificaciones programadas donde la persona pasa de estar parada a correr, caminar, etc., usando como núcleo el mismo Core Motion. Así mismo, con esta extensión del framework “XCode” puede usar los datos brutos recopilados del dispositivo y así mismo también se puede usar ciertas propiedades desde las clases ya procesadas.

En esta ocasión para nuestro sistema Core Motion se ocupa de una manera que monitoriza los ejercicios realizados, mostrando así el porcentaje que se está realizando, como se muestra a continuación:

- Se puede observar un bloque de código que tiene como función clasificar las clases antes generadas en el modelo de CoreML para así poder compararlas contra la foto tomada por la persona, así mismo muestra la cantidad de fatiga muscular por cada ejercicio realizado mostrada en porcentaje. Por otro lado, el rectángulo verde contempla una clasificación diferente a los ejercicios preestablecidos en el modelo “Inceptionv3”, trabajando así sobre el mismo archivo “CoreMLViewController” antes realizado en la sección de CoreML.

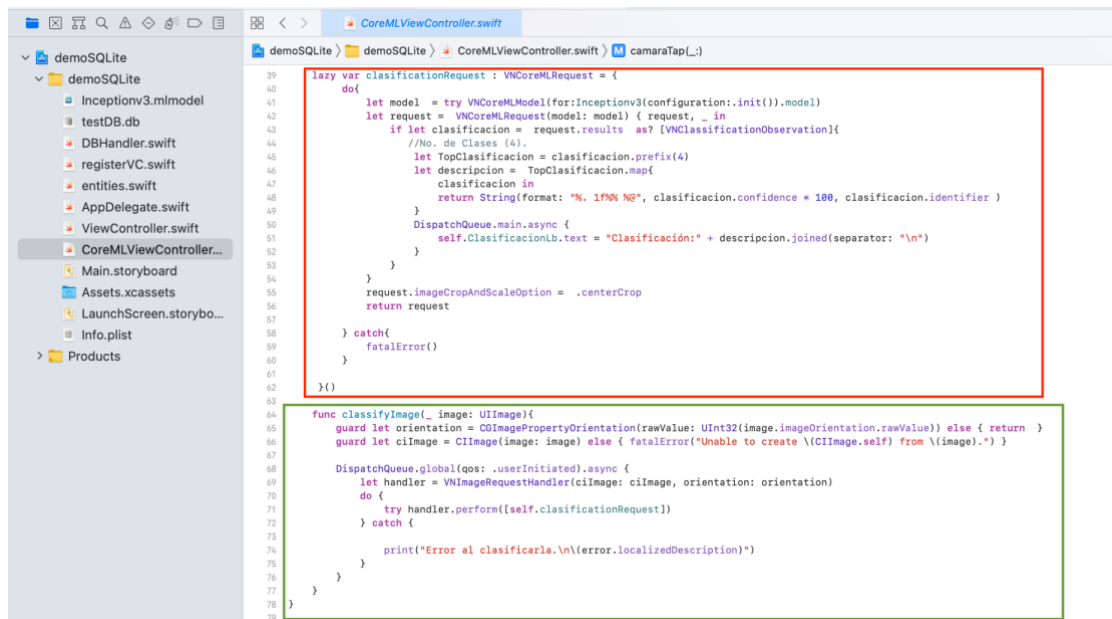


Figura 4.4 Implementación de Core Motion.

4.5 Integración

Es muy importante la integración de un sistema, ya que es el momento en el que todos los pequeños sistemas se interconectan para así poder formar un ambiente de trabajo con el cual cada componente realiza su respectiva función. En esta sección se presenta el sistema integrado² y las diferentes vistas (Frontend) basadas en el diseño del anterior capítulo, así mismo la base de datos (backend) y el cómo están relacionados.

Para poder llevar a cabo esta integración se llevaron a cabo los siguientes pasos:

- Dado que se ha trabajado sobre un mismo proyecto, en la siguiente imagen se puede observar la integración de las vistas implementadas (centro de la imagen); así mismo, en el rectángulo rojo se tiene implementado tanto la parte de “CoreML” con el archivo “Inceptionv3” y la base de datos hecha en “SQLite” con el archivo llamado “testDB”, por lo que, dando “clic” en el botón de “play” (rectángulo amarillo) podemos observar la ejecución del aplicativo:

² Este sistema se puede obtener en el anexo C, ocupando la herramienta de versionamiento llamada “Git”.

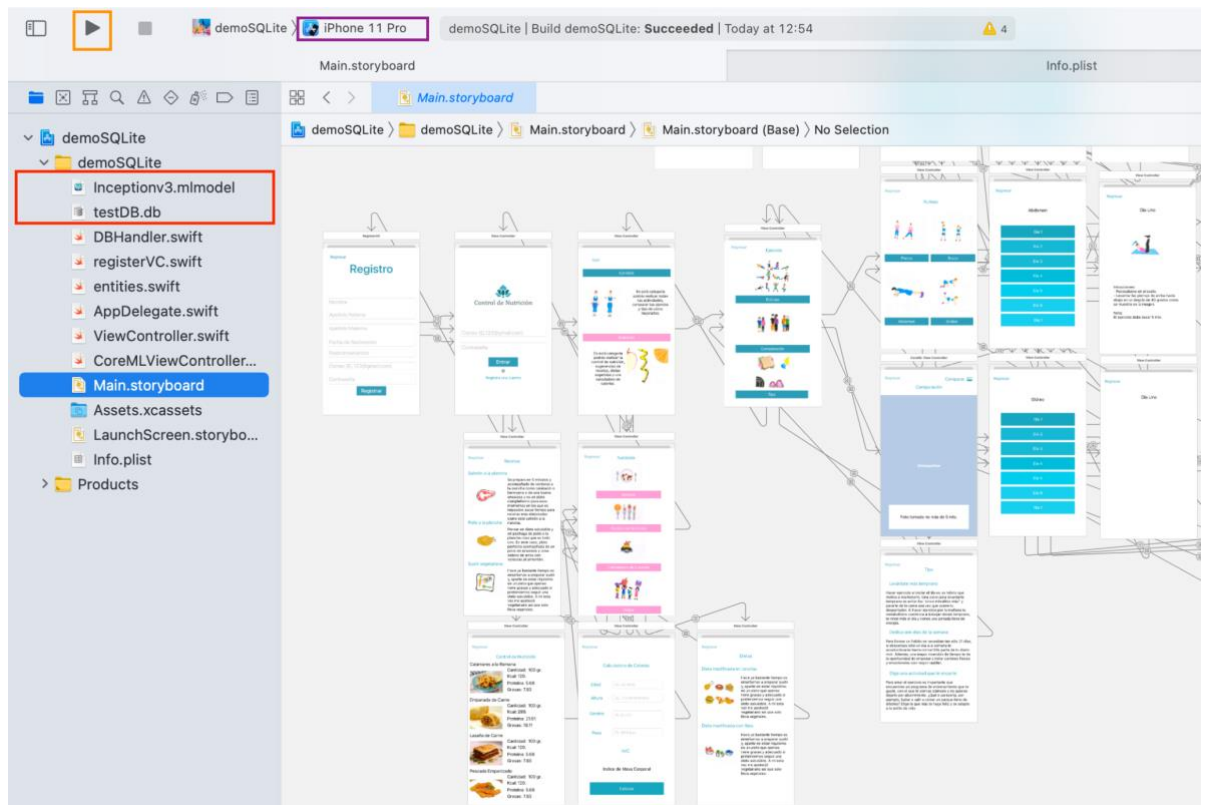


Figura 4.5.1 Integración del Aplicativo

Una vez desplegado el sistema, nos mostrará el emulador previamente seleccionado, por lo que, nos aparecerá el icono de la aplicación simulando el menú de un teléfono como se puede ver en la Figura 4.5.2 consecuentemente nos mostrará un “splash” representativo de la aplicación haciendo énfasis a un mejor control de nutrición (Figura 4.5.3):



Figura 4.5.2 y Figura 4.5.3

Principalmente se tiene la ventana de inicio, donde nos solicita entrar a través de un correo y una contraseña, si no se cuenta con un registro previo, la opción a escoger será “Registra una cuenta”, dónde nos solicitará varios campos a llenar para que así el sistema tenga un registro de nuestros datos, cumpliendo así con su respectivo caso de uso.

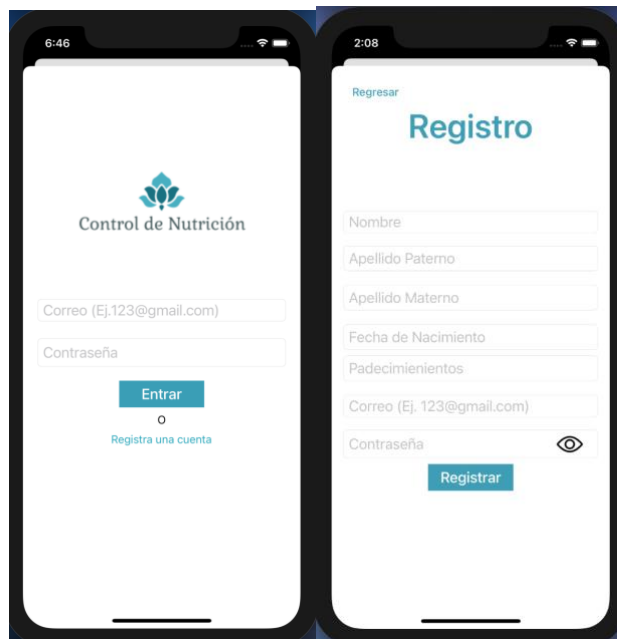


Figura 4.5.4 y 4.5.5

Una vez que la persona ha ingresado correctamente, se tienen dos opciones a elegir, una es la de ejercicio si desea entrenar ocupando una rutina específica o si desea comparar el ejercicio a entrenar o tips precargados sobre diferentes ejercicios. Así mismo, si se desea llevar un mejor control de nutrición, se tienen como opciones diferentes recetas, control de nutrición, calculadora de calorías y diferentes dietas.



Figura 4.5.6

Si se selecciona la opción de ejercicio, se presenta un menú donde dependiendo lo que se desee se escoge la opción. Se muestran tres opciones que son: rutinas, comparación y tips (Figura 4.5.7)



Figura 4.5.7

Por otra parte, si lo que se desea es entrenar alguna parte del cuerpo, la sección de “Rutinas” es la mejor opción, ya que así se podrá entrenar una sección del cuerpo en particular, decidiendo así entre el ejercicio de pierna, brazo, abdomen y glúteo (Figura 4.5.8). Dado que ya se ha elegido el ejercicio que se va a trabajar, se tiene una lista particular para ese ejercicio, donde dependiendo del día de la semana es el ejercicio que le tocará realizar (Figura 4.5.9). Es decir, el sistema tiene la capacidad para monitorear el ejercicio y sugerir el más apropiado.

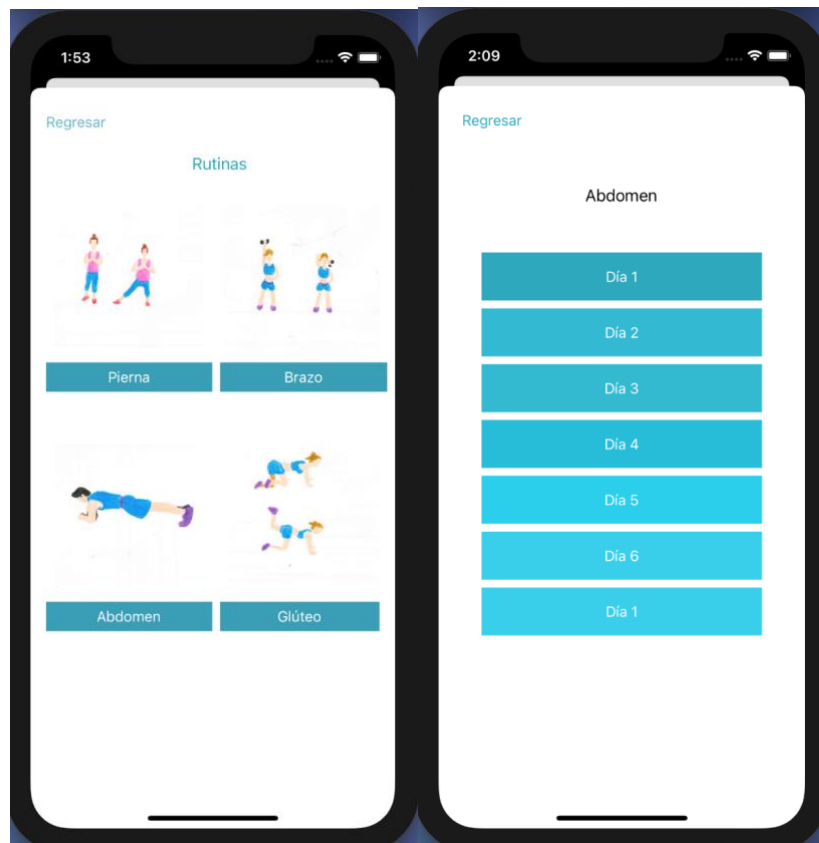


Figura 4.5.8 y Figura 4.5.9

Por último, la Figura 4.5.1.1 nos mostrará el ejercicio que corresponde al día que se haya elegido previamente, por ejemplo:



Figura 4.5.1.1

Ocupando la comparación de los ejercicios seleccionados, se le mostrará al usuario el porcentaje del ejercicio que ha realizado, esto para animar a que se siga esforzando.

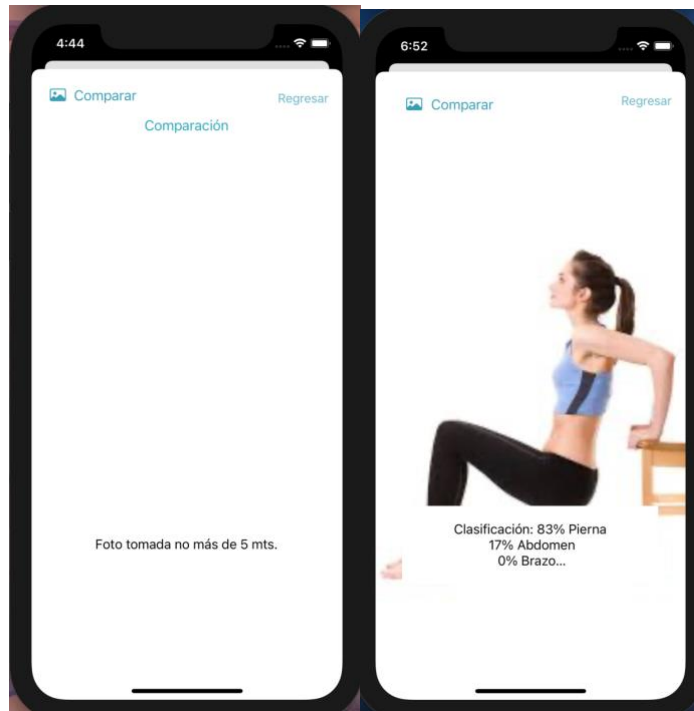


Figura 4.5.1.2 y Figura 4.5.1.3

Para esta sección de la aplicación, se muestran diferentes “tips”, para así poder ayudar a las personas con diferentes problemas que la comunidad se ha enfrentado y poder así resolverlas a través de estos conceptos:

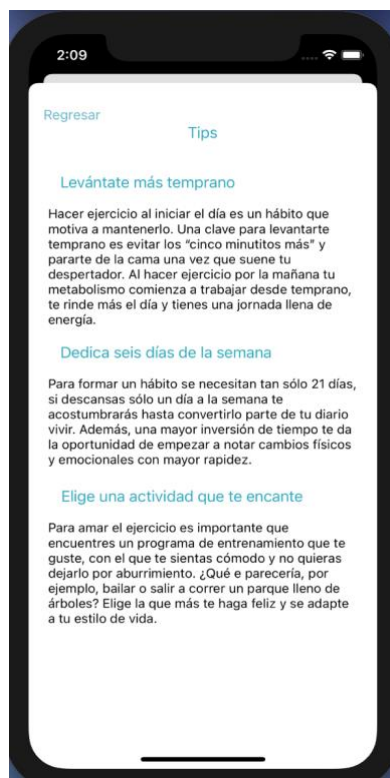


Figura 4.5.3.3

Posteriormente si lo que se desea es llevar un mejor control de nutrición, esta sección de la aplicación es lo ideal ya que cuenta con diferentes opciones, se tienen diferentes recetas, un control de nutrición, una calculadora de calorías y diferentes dietas enfocadas para la persona que utiliza la aplicación, cada sección del aplicativo se detalla más adelante.



Figura 4.5.3.4

Para la opción de recetas, se cuenta con una pequeña imagen y una breve descripción del platillo que se quiere realizar, así mismo, cuenta con un link de cada receta para poder realizar la receta.



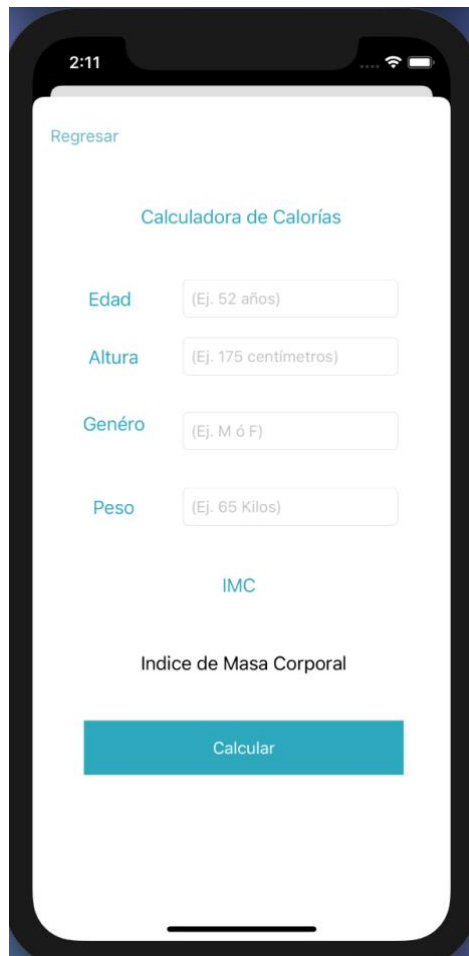
Figura 4.5.3.5

Así mismo, en la Figura 4.5.3.6 se tiene el “Control de nutrición”, donde se tiene una lista de diferentes platillos que normalmente las personas consumen para así tener una mejor conciencia de saber que están consumiendo por porción saludable y adecuada a sus requerimientos calóricos.



Figura 4.5.3.6

Así mismo, se tiene una calculadora de calorías para saber el estado en el que se encuentra.



The image shows a mobile application interface for a BMI calculator. At the top left, there is a "Regresar" link. The main title is "Calculadora de Calorías". Below this, there are four input fields: "Edad" (Age) with a placeholder "(Ej. 52 años)", "Altura" (Height) with a placeholder "(Ej. 175 centímetros)", "Género" (Gender) with a placeholder "(Ej. M ó F)", and "Peso" (Weight) with a placeholder "(Ej. 65 Kilos)". Below the input fields, the text "IMC" is displayed, followed by "Indice de Masa Corporal". At the bottom, there is a large teal button labeled "Calcular". The status bar at the top shows the time "2:11" and signal strength indicators.

Figura 4.5.3.7

Por último, se tienen diferentes dietas como se puede observar en la Figura 4.5.3.8, así las personas que utilicen el sistema pueden escoger cual dieta seguir según lo que quieran trabajar, por ejemplo, si se desea bajar de peso se debe seguir la dieta llamada “Dieta modificadora de calorías”.



Figura 4.5.3.8

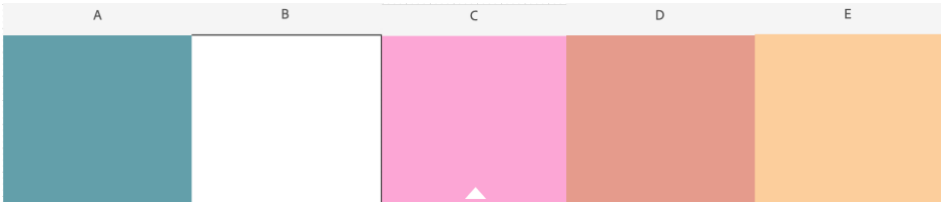
Capítulo V. Pruebas y resultados

En este capítulo hablaremos sobre las pruebas del sistema planeadas y realizadas sobre nuestra aplicación móvil. Es importante mencionar que las pruebas son muy importantes para cada sistema, esto debido a que nos ayuda a verificar que el sistema en efecto haga lo requerido por el usuario, que no tenga ningún problema y que todo funcione correctamente como se tiene previsto previamente en los casos de usos e historias de usuarios, ya que con esto también podemos medir y determinar la calidad del software que se está desarrollando. Existen diversos tipos de pruebas, en nuestro caso se realizarán pruebas funcionales y no funcionales con las cuales en unas verificaremos el funcionamiento del sistema de manera interna, y por otro lado verificaremos el diseño considerando la experiencia del usuario.

5.1 Pruebas de la aplicación

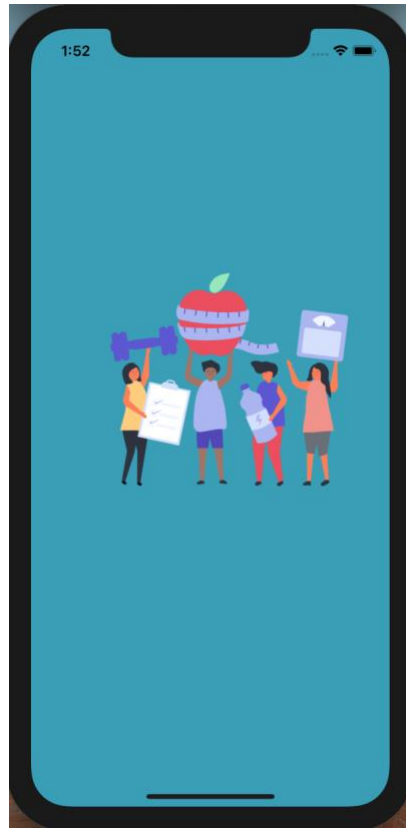
Las pruebas de la aplicación hacen referencia a la funcionalidad del sistema, que cumpla las reglas de negocio y que no arroje ningún problema al momento de utilizarlo, es importante mencionar que es necesario verificar cada uno de los flujos y, en cuanto a manera más interna, en caso de existir bases de datos, microservicios, etc., es importante también realizar esas validaciones. En nuestro caso, para el sistema solo utilizamos el tema de la base de datos, por lo tanto, también es importante validar que en efecto, la lectura de los datos y uso de registros se realicen correctamente. Así mismo, también los componentes que conforman nuestra aplicación como son *los colores, tipografía, etc.* Cabe mencionar que hay elementos que son considerados también parte de UX, pero se pueden integrar en cualquiera de las dos pruebas que se realizan.

A continuación, veremos el caso de prueba que se adaptó de una tabla en producción (Anexo D):

Caso de Uso asociado	CU-MDAIOS-A
Paso a ejecutar	Validación de colores
Rol que Ejecuta	QA
Fácil de navegar	Sí
Comentarios	<p>Durante el proceso de las pruebas, nos dimos a la tarea de validar los colores que tenemos dentro de nuestro aplicativo, con esto podemos validar que los colores no sean ni muy exagerados ni muy débiles para la vista de nuestros usuarios. En este caso los colores que se utilizaron fueron los siguientes para las vistas y para botones:</p> <ul style="list-style-type: none"> • Color azul, con RGB igual a (96, 159, 170, 67). • Color blanco, con RGB igual a (255, 255, 255, 100). • Color rosa, con RGB igual a (253, 164, 213, 99). • Color crema, con RGB igual a (230, 155, 138, 90). • Color crema, con RGB igual a (253, 206, 152, 99).
Evidencia	 <p>The image shows a horizontal row of five color swatches labeled A through E. Swatch A is a teal color. Swatch B is white. Swatch C is a light pink color. Swatch D is a light red or coral color. Swatch E is a light orange or peach color. A mouse cursor is visible over the bottom center of swatch C.</p>

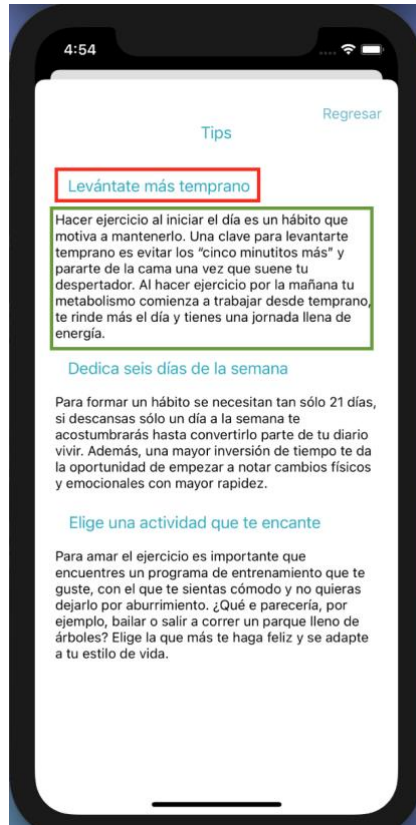
Caso de Uso asociado	CU-MDAIOS-B
Paso a ejecutar	Validar imágenes
Rol que Ejecuta	QA
Fácil de navegar	Sí
Comentarios	<p>En estas pruebas lo que se validó es lo relacionado a las imágenes que tenemos en el aplicativo como las características de cada una como puede ser el tamaño, calidad, el formato, así como todas estas sean representativas de lo necesario.</p> <ul style="list-style-type: none">• Las imágenes cuentan con un tamaño de 530x464 píxeles.• Las imágenes son representativas por cada ventana.• Las imágenes deben tener formato .png• Las imágenes deben de tener buena calidad esto quiere decir que no deben de ser borrosas.

Evidencia



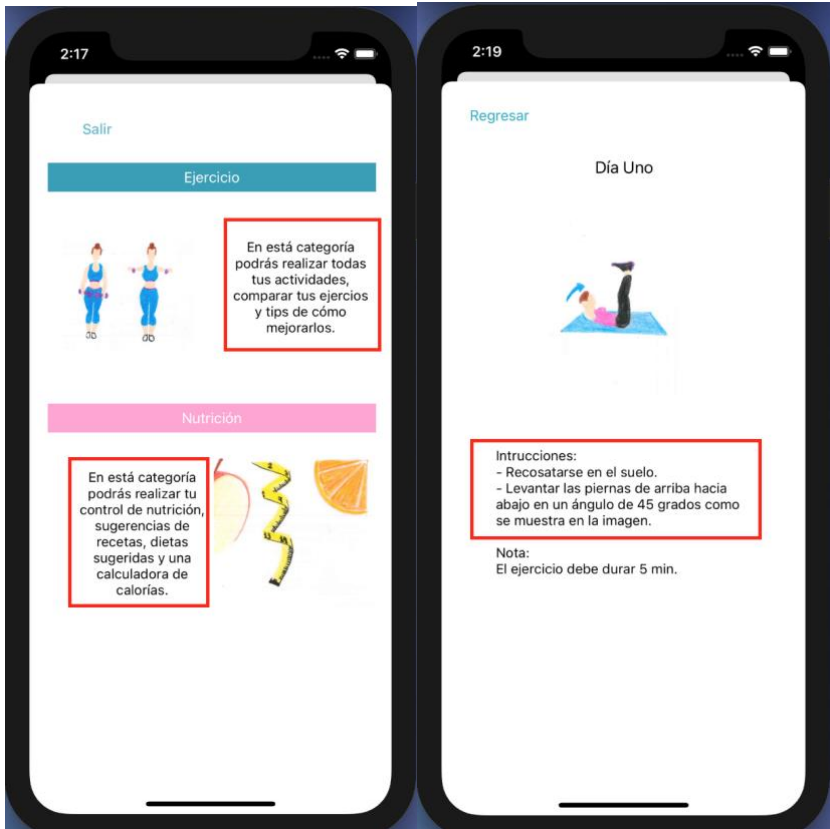
Caso de Uso asociado	CU-MDAIOS-C
Paso a ejecutar	Validar los estilos
Rol que Ejecuta	QA
Fácil de navegar	Sí
Comentarios	<p>En cualquier sistema u aplicación los estilos forman una parte muy importante en cuanto a la presentación, por lo que es necesario la homologación de estilos a lo largo de todo el sistema. En este caso se definió que los estilos fueran los siguientes:</p> <ul style="list-style-type: none">• Tipo de letra: Arial.• Tamaño de los títulos: 12 pixeles.• Tamaño de los subtítulos: 10 pixeles.• Tamaño de las descripciones: 12 píxeles.• Color de la letra: color azul, en RGB es (96, 159, 170, 67).

Evidencia




Caso de Uso asociado	CU-MDAIOS-D
Paso a ejecutar	Validar lenguaje con el usuario.
Rol que Ejecuta	QA
Fácil de navegar	Sí


<p>Comentarios</p>	<p>Algo que es muy relevante es la manera en la que te quieras comunicar con los usuarios, no hay problema si es de tu o de usted, pero debe de estar homologado, lo que es incorrecto es que en una pantalla le hables de tu y en la siguiente de usted. Por esto a lo largo de las pruebas realizadas fue importante validar que en efecto se cumpliera con esta homologación en nuestro sistema se seleccionó la siguiente:</p> <ul style="list-style-type: none"> • Se puede comunicar “de usted” al usuario.
---------------------------	--

<p>Evidencia</p>	 <p>The image shows two smartphone screens from an application. The left screen, taken at 2:17, has a 'Salir' button at the top. Below it is a blue bar labeled 'Ejercicio' with two female icons. A red-bordered text box explains that users can track activities and tips. Below that is a pink bar labeled 'Nutrición' with icons of a water bottle, a measuring tape, and an orange. A red-bordered text box explains that users can manage nutrition, get recipes, and use a calorie calculator. The right screen, taken at 2:19, has a 'Regresar' button at the top. Below it is the text 'Día Uno' and an illustration of a person performing a leg raise. A red-bordered text box lists instructions: 'Recosarse en el suelo' and 'Levantar las piernas de arriba hacia abajo en un ángulo de 45 grados como se muestra en la imagen.' Below that, a 'Nota:' section states 'El ejercicio debe durar 5 min.'</p>
-------------------------	--


<p>Caso de Uso asociado</p>	<p>CU-MDAIOS-E</p>
<p>Paso a ejecutar</p>	<p>Validar alineación</p>

Rol que Ejecuta	QA
Fácil de navegar	Sí
Comentarios	Las alineaciones en estos casos deben de ser de todo tipo, a nivel de imágenes, botones, textos, elementos que tengamos.
Evidencia	


Caso de Uso asociado	CU-MDAIOS-F
Paso a ejecutar	Validar fechas

Rol que Ejecuta	QA
Fácil de navegar	Sí
Comentarios	En nuestro aplicativo no solicitamos muchas fechas, pero es necesario Validar que la fecha de nacimiento sea la correcta y que no se puedan poner fechas incongruentes.
Evidencia	 <p>The screenshot shows a mobile application interface for registration. The screen is titled 'Registro' and has a 'Regresar' link at the top left. Below the title are several input fields: 'Nombre', 'Apellido Paterno', 'Apellido Materno', 'Fecha de Nacimiento', 'País', 'Correo (Ej. 123@gmail.com)', and 'Contraseña'. A blue 'Registrar' button is at the bottom. An alert dialog box is overlaid on the 'Fecha de Nacimiento' field, with the title 'Alerta' and the message 'Verifica que la fecha este bien escrita'. The dialog has an 'Aceptar' button.</p>

Caso de Uso asociado	CU-MDAIOS-G
Paso a ejecutar	Validar que sea intuitivo.

Rol que Ejecuta	QA
Fácil de navegar	Sí
Comentarios	<p>El sistema debe de ser intuitivo para todos los usuarios, que no sea necesario un manual o una explicación previa al cómo usarlo considerando que es muy fácil de utilizar.</p>
Evidencia	 <p>The image shows two smartphone screens displaying a fitness application. The left screen, timestamped 2:26, features a 'Regresar' button at the top left and a 'Ejercicio' header. Below the header is a grid of colorful exercise icons. Three blue buttons with white text are highlighted with red boxes: 'Rutinas' (highlighted first), 'Comparación', and 'Tips'. The right screen, timestamped 2:27, shows a 'Regresar' button at the top left and an 'Abdomen' header. Below the header is a vertical list of seven blue buttons with white text, also highlighted with a red box: 'Día 1', 'Día 2', 'Día 3', 'Día 4', 'Día 5', 'Día 6', and 'Día 1'.</p>

Caso de Uso asociado	CU-MDAIOS-H
-----------------------------	-------------

Paso a ejecutar	Validar tamaños.
Rol que Ejecuta	QA
Fácil de navegar	Sí
Comentarios	En cuanto a la parte de validación de tamaños puede ser referente a elementos de aplicativos como son imágenes, texto, imágenes, botones, etc. Como lo mencionamos previamente es necesario estandarizar en este caso los tamaños.
Evidencia	

5.2 Pruebas de la funcionalidad (UX)

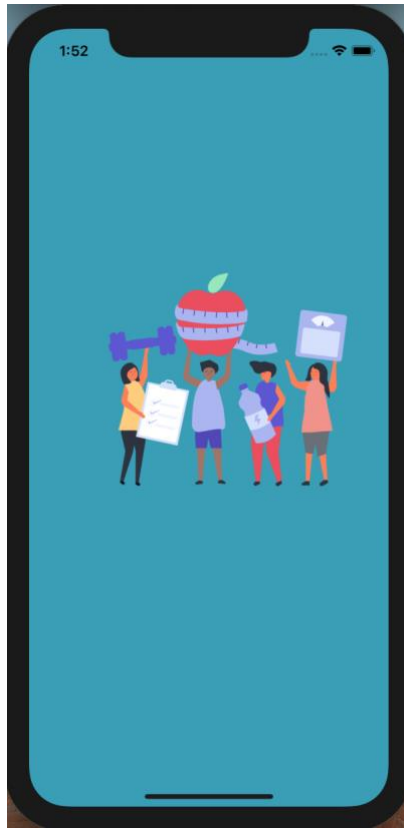
En cuanto a la experiencia de usuario, es aquello que una persona percibe al interactuar con un producto o servicio, se dice que logramos un buen UX al momento

de enfocarnos en diseñar productos útiles, usables y deseables, lo cual influye en que los usuarios se sientan satisfechos en el manejo del aplicativo.

Las pruebas de funcionalidad se engloban en cuanto al diseño y también en cuanto al cumplimiento de las reglas de negocio, así mismo también en cuanto al prototipo y los flujos. En nuestro caso nos basamos en los casos de uso y en el prototipo que previamente ya se había creado.

Caso de Uso asociado	CU-MDAIOS-001
Paso a ejecutar	Inicio de la aplicación
Rol que Ejecuta	Usuario
Prueba Exitosa	Sí
Comentarios	El sistema muestra la pantalla: <ul style="list-style-type: none">• Un "Splash" correspondiente.

Evidencia



Caso de Uso asociado	CU-MDAIOS-001.1
Paso a ejecutar	Pantalla de inicio
Rol que Ejecuta	Usuario
Prueba Exitosa	Sí

Comentarios

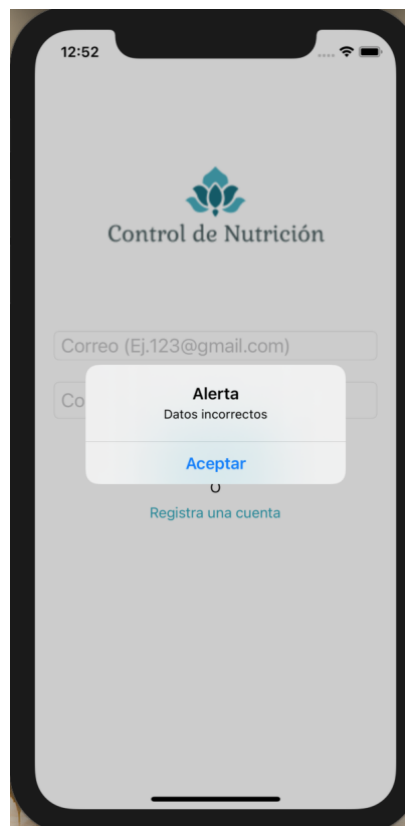
El sistema muestra la pantalla de inicio y solicita los siguientes datos:

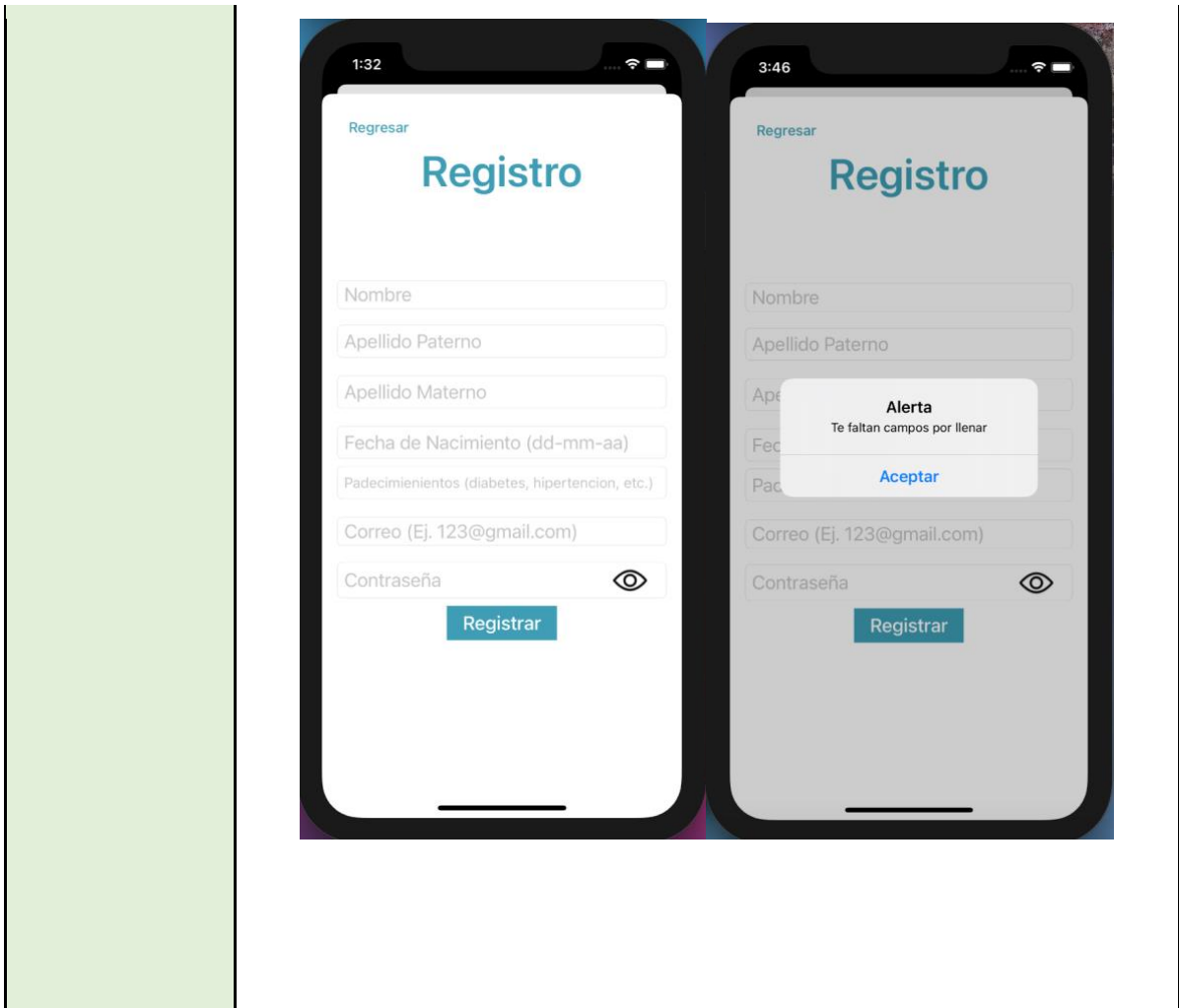
- Correo.
- Contraseña.

El sistema muestra la pantalla de registro y solicita los siguientes datos para el registro:

- Nombre.
- Apellido Paterno.
- Apellido Materno.
- Fecha de nacimiento en formato: dd-mm-aaaa.
- Padecimientos.
- Correo.
- Contraseña.

Evidencia





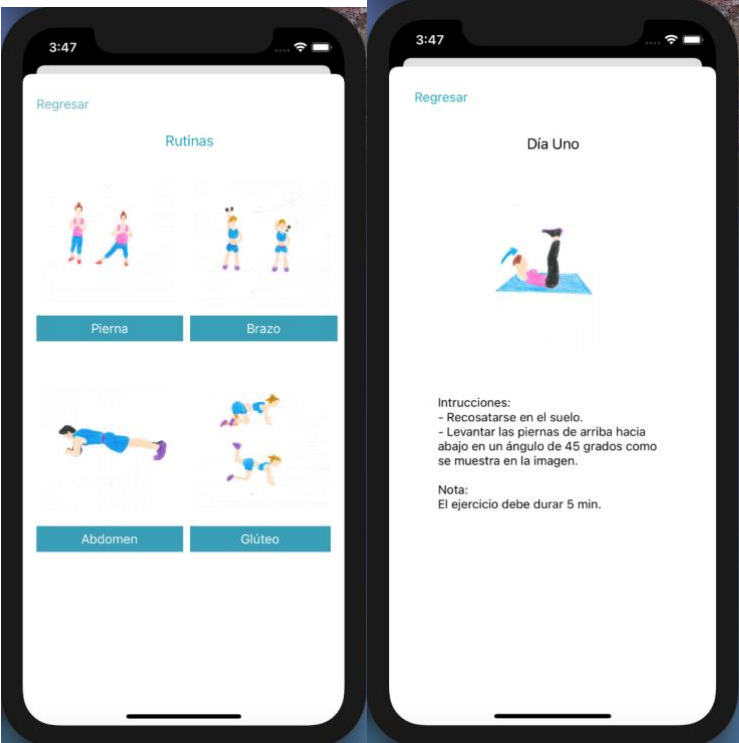
Caso de Uso asociado	CU-MDAIOS-001.1.1
Paso a ejecutar	Ejercicio
Rol que Ejecuta	Usuario
Prueba Exitosa	Sí

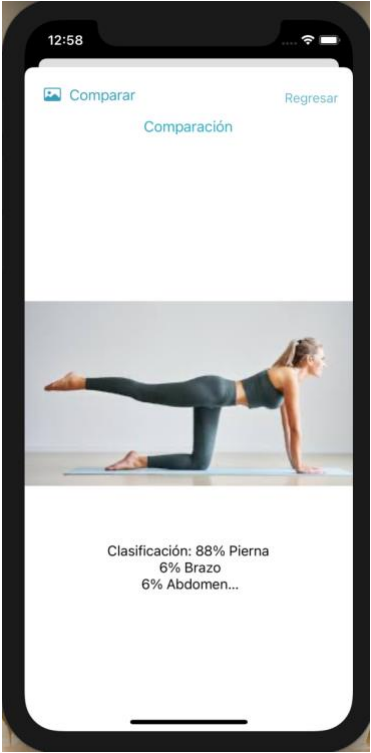
<p>Comentarios</p>	<p>El sistema muestra los siguientes datos e imágenes correspondientes en forma de lista:</p> <ul style="list-style-type: none"> • Opción de Rutinas. • Opción de Comparación. • Opción de Tips.
<p>Evidencia</p>	 <p>The screenshot shows a mobile application interface with a white background and a black border. At the top, there is a status bar with the time 3:46 and signal strength indicators. Below the status bar, there is a header with the word 'Ejercicio' in blue. To the left of the header is a link labeled 'Regresar'. The main content area features a grid of colorful icons representing various exercises. Below this grid are three blue buttons with white text: 'Rutinas', 'Comparación', and 'Tips'. Each button is accompanied by a small set of icons: 'Rutinas' has icons of people exercising, 'Comparación' has icons of a sandwich and a banana, and 'Tips' has icons of a book and two eggs.</p>

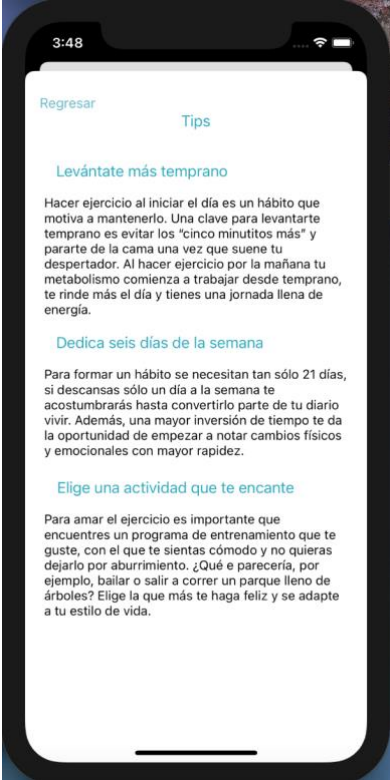
<p>Caso de Uso asociado</p>	<p>CU-MDAIOS-001.1.2</p>
<p>Paso a ejecutar</p>	<p>Nutrición</p>
<p>Rol que Ejecuta</p>	<p>Usuario</p>


Prueba Exitosa	Sí
Comentarios	<p>El sistema muestra los siguientes datos e imágenes correspondientes en forma de lista:</p> <ul style="list-style-type: none"> • Opción de Recetas. • Opción de Control de Nutrición. • Opción de Calculadora de Calorías. • Opción de Calculadora de Dietas.
Evidencia	

Caso de Uso asociado	CU-MDAIOS-001.1.1.1
Paso a ejecutar	Rutinas

Rol que Ejecuta	Usuario
Prueba Exitosa	Sí
Comentarios	<p>El sistema muestra la pantalla y solicita elegir cualquier opción:</p> <ul style="list-style-type: none">• Opción de Pierna.• Opción de Brazo.• Opción de Abdomen.• Opción de Glúteo. <p>El sistema muestra la pantalla y siguientes datos para realizar el ejercicio correspondiente:</p> <ul style="list-style-type: none">• Título del día.• Una imagen representativa.• Instrucciones correspondientes al ejercicio.
Evidencia	 <p>The image displays two screenshots of a mobile application interface. The left screenshot shows a 'Rutinas' screen with a 'Regresar' button at the top left. Below the title, there are four exercise options, each with a small illustration and a corresponding button: 'Pierna' (Legs), 'Brazo' (Arm), 'Abdomen' (Abdomen), and 'Glúteo' (Glute). The right screenshot shows a 'Día Uno' screen with a 'Regresar' button at the top left. Below the title, there is a large illustration of a person performing a leg lift exercise. Below the illustration, there are instructions and a note: 'Instrucciones: - Recostarse en el suelo. - Levantar las piernas de arriba hacia abajo en un ángulo de 45 grados como se muestra en la imagen. Nota: El ejercicio debe durar 5 min.'</p>

Caso de Uso asociado	CU-MDAIOS-001.1.1.2
Paso a ejecutar	Comparación
Rol que Ejecuta	Usuario
Prueba Exitosa	Sí
Comentarios	<p>El sistema muestra la pantalla, solicita y muestra los siguientes datos:</p> <ul style="list-style-type: none"> • Una imagen previamente tomada. • El porcentaje realizado por cada categoría de ejercicio.
Evidencia	 <p>The screenshot shows a mobile application interface for a 'Comparación' (Comparison) screen. At the top, there are two buttons: 'Comparar' (Compare) and 'Regresar' (Return). Below the buttons, the word 'Comparación' is centered. In the middle of the screen, there is a photograph of a woman performing a Bhujangasana (Cobra) yoga pose. Below the photograph, the classification results are displayed: 'Clasificación: 88% Pierna', '6% Brazo', and '6% Abdomen...'. The entire screenshot is framed within a black border representing a smartphone.</p>

Caso de Uso asociado	CU-MDAIOS-001.1.1.3
Paso a ejecutar	Consejos
Rol que Ejecuta	Usuario
Prueba Exitosa	Sí
Comentarios	<p>El sistema muestra la pantalla y los siguientes datos:</p> <ul style="list-style-type: none"> • Título de la vista. • Título de cada Tip. • Una breve descripción de cada Tip.
Evidencia	 <p>The screenshot shows a mobile application interface. At the top, there is a status bar with the time 3:48 and signal strength indicators. Below that, a white card contains the following content:</p> <ul style="list-style-type: none"> A blue link labeled "Regresar" in the top left corner. A blue title "Tips" centered at the top of the card. A section titled "Levántate más temprano" with a paragraph of text explaining the benefits of waking up earlier. A section titled "Dedica seis días de la semana" with a paragraph of text about habit formation. A section titled "Elige una actividad que te encante" with a paragraph of text about choosing enjoyable activities.

Caso de Uso asociado	CU-MDAIOS-001.1.2.1
Paso a ejecutar	Recetas
Rol que Ejecuta	Usuario
Prueba Exitosa	Sí
Comentarios	<p>El sistema muestra la pantalla y los siguientes datos:</p> <ul style="list-style-type: none"> • Título de la vista. • Título de cada receta. • Imagen de cada receta. • Una breve descripción de cada receta.
Evidencia	 <p>The screenshot shows a mobile application interface for a recipe book. At the top, there is a 'Regresar' button on the left and the title 'Recetas' on the right. Below the title, there are three recipe cards. The first card is titled 'Salmón a la plancha' and features a small image of a salmon fillet. The second card is titled 'Pollo a la plancha' and features a small image of a chicken breast. The third card is titled 'Sushi vegetariano' and features a small image of a sushi roll. Each card has a short paragraph of text describing the recipe.</p>

Caso de Uso asociado	CU-MDAIOS-001.1.2.2
Paso a ejecutar	Control de nutrición
Rol que Ejecuta	Usuario
Prueba Exitosa	Sí
Comentarios	<p>El sistema muestra la pantalla y los siguientes datos:</p> <ul style="list-style-type: none"> • Título de la vista. • Título de cada Control de Nutrición. • Imagen de cada receta a controlar. • Una breve descripción con los siguientes campos: <ul style="list-style-type: none"> ➤ Cantidad. ➤ Kilocalorías. ➤ Proteína. ➤ Grasas.

Evidencia



Caso de Uso asociado	CU-MDAIOS-001.1.2.3
Paso a ejecutar	Calculadora de calorías
Rol que Ejecuta	Usuario
Prueba Exitosa	Sí

Comentarios

El sistema muestra la pantalla y solicita los siguientes datos:

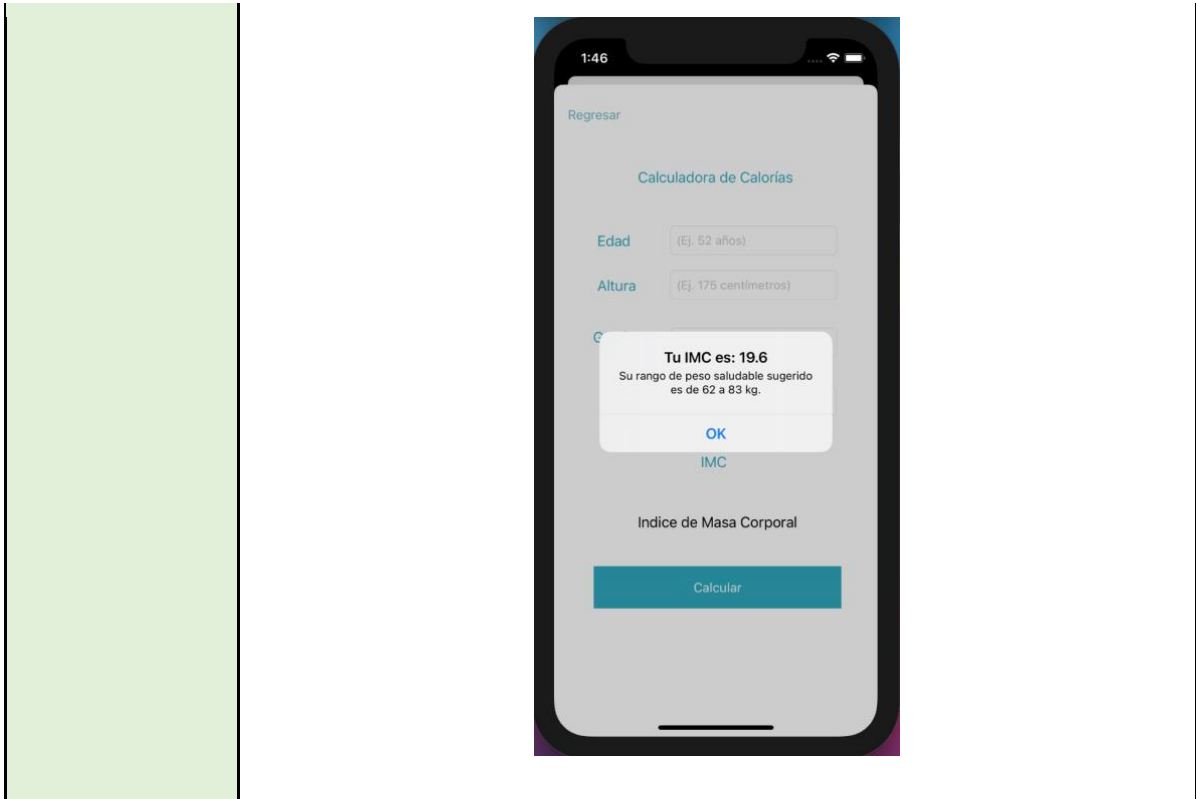
- Altura en centímetros.
- Peso en kilos.

Así, mostrando:

- Índice de masa corporal.

Evidencia

The image displays two side-by-side screenshots of a mobile application interface for a BMI calculator. Both screens show a 'Regresar' button at the top left and a 'Calculadora de Calorías' title. The left screenshot shows input fields for 'Edad' (52 años), 'Altura' (175 centímetros), 'Género' (M ó F), and 'Peso' (65 Kilos). The right screenshot shows input fields for 'Edad' (25), 'Altura' (175), 'Género' (M), and 'Peso' (60). Both screens feature a blue 'Calcular' button at the bottom and the text 'Índice de Masa Corporal' below the input fields.



Caso de Uso asociado	CU-MDAIOS-001.1.2.4
Paso a ejecutar	Dieta
Rol que Ejecuta	Usuario
Prueba Exitosa	Sí
Comentarios	<p>El sistema muestra la pantalla y los siguientes datos:</p> <ul style="list-style-type: none"> • Título de la vista. • Título de cada Dieta. • Imagen de cada Dieta. • Una breve descripción de cada Dieta.

Evidencia



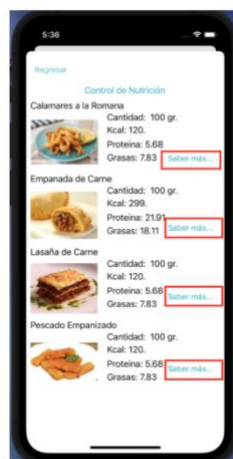
5.3 Siguientes pasos (Entrevista con usuarios)

La aplicación no ha sido publicada en la Apple Store, sin embargo, se les solicitó a dos usuarios que hicieran uso del aplicativo para que así pudieran compartir su experiencia y sus observaciones derivado del desarrollo.

Los usuarios a los que se mostró quedaron satisfechos, aunque hicieron algunas recomendaciones, por ejemplo, un nutriólogo comentó que, se propone que a un largo plazo debería de considerarse que se lleve un control más estricto del control de nutrición en cuanto a cada componente que tenga cada alimento, un claro ejemplo que expuso fue sobre ensaladas, en este caso nos comenta que hay ensaladas, que llevan atún, lechuga, nueces, queso, aderezo, etc., y ya que en el módulo de control de nutrición se tiene a un alto nivel donde se engloban las calorías por receta, así que su recomendación es a un nivel más bajo, en el que sea por ingrediente que contenga cada una de las recetas o a un vínculo en línea (Véase siguiente Figura 5.3.2) que lleve la persona que está utilizando el sistema a una receta más específica para que al momento de que un usuario realice algún platillo de diferente manera sea un poco más exacto el cálculo de estas calorías tomando como referencia el video previamente seleccionado por nosotros.



Antes 5.3.1



Después 5.3.2

Consecuentemente, la siguiente persona a la que se le mostró el aplicativo fue un diseñador, el cual nos hizo algunas recomendaciones sobre los temas de los colores, estilos, UX en general, las cuales se aplicaron previamente en el sistema. Así mismo, hizo algunos comentarios sobre el cómo poder hablar con el usuario, sobre los tipos de error, los cuales, se hicieron cambios para que fuera un poco más amigable e intuitiva.



Antes 5.3.3



Después 5.3.4

Capítulo VI. Conclusiones y trabajos futuros

El presente trabajo muestra el desarrollo de una aplicación móvil que permite la mejora del control de la salud a través de la medicina deportiva utilizando un sistema basado en conocimiento y machine learning. Para el *diseño del sistema de control de la salud* se requirió la integración de los conceptos básicos de medicina del deporte ya que eran necesarios para el modelado de los datos, los cuales fueron: índice de masa corporal, peso, dieta, entre otros.

Se comienza destacando que, dentro de los primeros pasos fue el *diseñar un sistema de control de salud* vía móvil, para que así cualquier persona pudiera tener acceso a esta plataforma, por lo que se utilizaron herramientas y **tecnologías nativas** de desarrollo gratuitas tales como: “XCode” y “Swift” correspondientes al sistema operativo iOS, ya que con ellas permiten eficientar información en tiempo real y también permiten explorar nuevas ramas de la tecnología como lo es machine learning.

Por otra parte, el *modelo de base de datos* junto con la *integración del sistema* y, al *describir los elementos de medicina del deporte* no fue sencillo, ya que hubo que considerar puntos muy importantes como lo fueron el de qué tipo de base de datos se utilizaría, qué datos son relevantes a considerar con respecto a una persona que sufre de un padecimiento crónico degenerativo, cómo la base de datos se implementaría con respecto a la aplicación móvil, y la curva de aprendizaje con respecto a un lenguaje de programación relativamente nuevo, todos esos puntos fueron considerados e implementados exitosamente, ya que se cumplieron los *casos de uso* eficientemente descritos en el capítulo tres, tal es el caso que se creó el módulo de “comparación”, este componente clasificador del aplicativo se desarrolló para la mejora del deporte, fue utilizar machine learning como referente, la cual nos permite categorizar en diferentes grupos, descritos por: “Brazo, Pierna, Abdomen y Glúteo”.

La *implementación del módulo basado en conocimiento o machine learning* fue desafiante, ya que actualmente pocas aplicaciones móviles están enteramente desarrollando temas similares, por lo que se dedicó tiempo y esfuerzo para su análisis, entendimiento e implementación. Principalmente, se tuvo como objetivo crear un modelo de conocimiento a través de imágenes que propiamente se crearon para entrenar un modelo utilizando “CoreML”, generando un archivo con todas estas imágenes clasificadas, después, se utilizaron las categorías previamente definidas, así, el beneficio de este módulo es encargarse de categorizar y comparar el tipo de ejercicio que se está realizando en el momento, lo cuál, es un ‘hito’ y aporte en el mercado actualmente.

Respecto a la *implementación del sistema de control de salud utilizando “Core Motion”* se tuvo el mismo problema, ya que fue difícil de implementar en la aplicación, porque al igual que el modelo de machine learning hay pocas referencias bibliográficas, y las que existen se encuentran en idioma inglés. además de que hay una comunidad reducida, aunque creciente que actualmente está trabajando con este tipo de tecnología, lo cuál se soluciono investigando y estudiando sobre el tema, así mismo, el beneficio de este módulo es consultar y mostrar el porcentaje del ejercicio realizado previamente, como se describe en el capítulo cuatro.

Se puede concluir que, los objetivos planteados al principio de este documento fueron cumplidos ya que se realizó el diseño de la aplicación, pasando después por la etapa de implementación, hasta llegar a las pruebas exitosas, todo esto para poder ayudar a una población que no ha sido atendida de una manera correcta por mucho tiempo, es por eso que se piensa y crea este aplicativo, surge como una necesidad, por lo que se puede afirmar que es posible controlar la salud a través de un dispositivo ocupando una base de conocimiento y machine learning como referente. Así mismo, se espera que esta aplicación sea de gran utilidad para que los alumnos de la carrera de Matemáticas Aplicadas y Computación y cualquier persona ajena a la misma, pueda ocupar y aprovecharla para fines de entendimiento o en trabajos futuros propios [Apéndice C].

En cuanto a *trabajos futuros*, basándose en la experiencia que se ha adquirido a través del trabajo, se ha considerado que se podría transformar esta aplicación, ya que se trabajó de una manera lineal y, para poder cambiar a las tecnologías que se encuentran funcionando en el mercado, sería cambiarla a una manera modular en la cual se desarrollaría mediante microservicios, así mismo, con un servicio de la nube, para poder llevar esto a cabo, primero se generaría un *WBS*, después un diagrama de arquitectura (Figura 6.1) y posteriormente comenzar con el desarrollo de cada uno de los servicios y al final la integración con el *'front-end'*, también sería importante considerar la intervención de un diseñador para su punto de vista en cuanto a UX, así mismo nos apoye con la maquetación de un prototipo y posteriormente comenzar con el desarrollo del producto, al final también sería prudente contar con un tiempo de pruebas esto para que el producto salga sin errores o con el mínimo posible ya que se contaría como un nuevo desarrollo o proyecto.

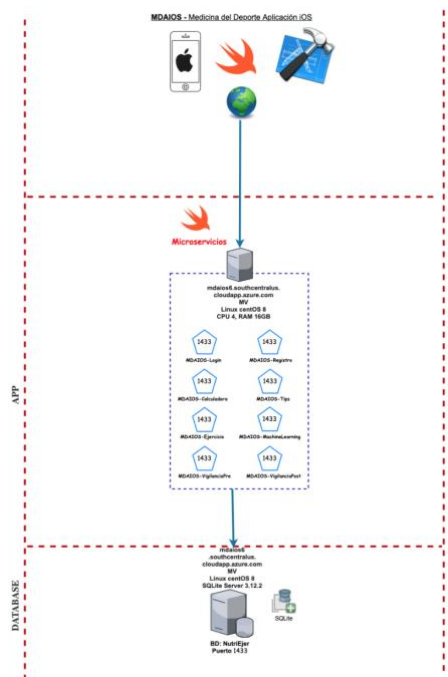


Diagrama de Arquitectura 6.1.

Referencias

- [1] Organización Mundial de la Salud. "Obesidad". [En línea]. Disponible en: <https://www.who.int/topics/obesity/es/>
- [2] Ballesteros Massó R (2001). Traumatología y Medicina del Deporte. Tomo I. 1ra Edic. Edit. Paraninfo.
- [3] imss.gob.mx, "Nutrición". [En línea]. Disponible en: <https://www.imss.gob.mx/salud-en-linea/nutricion>
- [4] Bello Gutiérrez, J. (2005). Calidad de vida, alimentos y salud humana. Edit. Díaz de Santos.
- [5] López Chicharro J., Fernández Vaquero A (2020). Fisiología del Ejercicio. Edit. Médica Panamericana.
- [6] Ben Coppin, "Artificial Intelligence Illuminated", 2004, pp 251-259.
- [7] Ben Coppin, "Artificial Intelligence Illuminated", 2004, pp 27-57.
- [8] ecured.cu, "Motor de búsqueda". [En línea]. Disponible en: https://www.ecured.cu/Motor_de_b%C3%BAsqueda
- [9] Stuart Rusell, Peter Norving (2004). Inteligencia Artificial, un enfoque moderno. Edit. Prentice Hall.
- [10] Willie Brand (IBM), "Machine learning for dummies". [En línea]. Disponible en: <https://www.ibm.com/downloads/cas/GB8ZMQZ3>
- [11] Eduardo Loza, Mayra Diaz. "Representación de problemas gráficos utilizando GeoGebra para los alumnos de Computación". UNAM, pp 4.
- [12] Eduardo Loza, Mayra Diaz. "Representación de problemas gráficos utilizando GeoGebra para los alumnos de Computación". UNAM, pp 2.

- [13] Eduardo Loza, Mayra Diaz. "Representación de problemas gráficos utilizando GeoGebra para los alumnos de Computación". UNAM, pp 2.
- [14] IBM, "El modelo de redes neuronales". [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html
- [15] UNAM. "Lenguajes de programación". [En línea]. Disponible en: http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf
- [16] Sergio Becerril, "*Swift 4. Aprende a crear Apps para iPhone y iPad*". Alfaomega, 2018, pp 93-94.
- [17] Apple, Inc. (2014). "Swift". [Figura]. Recuperado de <https://www.apple.com/mx/swift/>
- [18] amp.ipadizate.es, "X-code, ¿Qué es?, ¿Para qué sirve? Y ¿Cómo descargar?". [En línea]. Disponible en: <https://www.google.com/amp/s/amp.ipadizate.es/2014/07/20/xcode-93212>
- [19] Apple, Inc. (2003). "X code". [Figura]. Recuperado de <https://developer.apple.com/xcode/>
- [20] MathWorks (2021). "Machine Learning. Tres cosas que es necesario saber ". [En línea]. Disponible en: <https://es.mathworks.com/discovery/machine-learning.html>
- [21] Apple, Inc. (2018). "Core Motion - Apple Developer", Developer.apple.com, 2019. [En línea]. Disponible en: <https://developer.apple.com/documentation/coremotion>
- [22] Apple, Inc. (2018). "Core ML - Apple Developer", Developer.apple.com, 2019. [En línea]. Disponible en: <https://developer.apple.com/documentation/coreml>
- [23] Apple, Inc. (2018). "Core ML". [Figura]. Recuperado de <https://developer.apple.com/documentation/coreml>
- [24] Apple, Inc. (2019). "Core Motion - Apple Developer", Developer.apple.com, 2019. [En línea]. Disponible en: <https://developer.apple.com/documentation/coremotion>

- [25] Ricardo D. Ezquerro (2005). "Técnicas de respaldo y recuperación de bases de datos implementadas con el DBS Oracle". UNAM, pp 8-9.
- [26] Jan, L (2009). "Relational database design and implementation". Morgan Kaufmann.
- [27] Abraham S., Henry F., Surdashaan S. (2002). "Fundamentos de Base de Datos". Mc Graw Hill.
- [28] DBA Support (2019). "Características de MySQL". [En línea]. Disponible en: <https://www.dbasupport.com.mx/index.php/bases-de-datos/mysql/mysql-administracion/132-caracteristicas-de-mysql>
- [29] MySQL (2021). "MySQL". [En línea]. Disponible en: <https://www.mysql.com>
- [30] EcuRed (2021). "Oracle". [En línea]. Disponible en: <https://www.ecured.cu/Oracle>
- [31] Oracle Database (2021). "Oracle Database". [En línea]. Disponible en: <https://www.oracle.com/mx/database>
- [32] SQLite (2021). "DB Browser for SQLite". [En línea]. Disponible en: <https://www.sqlite.org/index.html>
- [33] EcuRed (2021). "Git". [En línea]. Disponible en: <https://www.ecured.cu/Git>
- [34] Git, Inc (2021). "Git". [En línea]. Disponible en: <https://git-scm.com>
- [35] Amazon (2021). "AWS Toolkit for Microsoft Azure DevOps Documentation". [En línea]. Disponible en: <https://docs.aws.amazon.com/vsts/>
- [36] Microsoft Azure (2021). "Azure DevOps Services". [En línea]. Disponible en: <https://azure.microsoft.com/es-mx/services/devops/>
- [37] Club de tecnología (2021). "GitHub vs. GitLab vs. Bitbucket: ¿Que repositorio elegir?". [En línea]. Disponible en: <https://www.clubdetecnologia.net/blog/2017/github-vs-gitlab-vs-bitbucket-que-repositorio-elegir/>
- [38] Bitbucket (2021). "Bitbucket". [En línea]. Disponible en: <https://bitbucket.org>

- [39] Gavin Allanwood, Peter Beare (2019). "User experience design, a practical. Introduction". Edit. Bloomsbury.
- [40] Diana MacDonald (2019). Practical UI patterns for design systems. Edit. Apress.
- [41] SQLite (2021). "DB Browser for SQLite". [En línea]. Disponible en: <https://www.sqlite.org/index.html>
- [42] Apple, Inc. (2021). "Xcode en Mac App Store". [En línea]. Disponible en: <https://apps.apple.com/es/app/xcode/id497799835?mt=12>
- [43] Sergio Becerril, "*Swift 4. Aprende a crear Apps para iPhone y iPad*". Alfaomega, 2018, pp 202-204.
- [44] iOS Development Lab, Facultad de Ingeniería (UNAM). "Core ML". [En línea]. Disponible en: <https://ioslabunam.github.io/wiki/posts/coreml.html>

Glosario

AI: Autoincremental.

API: Application Programming Language.

CU-MDAIOS: Caso de Uso Medicina del Deporte Aplicación iOS.

Database: Base de Datos.

FK: Foreign Key.

Frame: Marco.

ID: Identificador Único.

IDE: Integrated Development Environment.

Layout: Diseño.

PK: Primary Key.

Scroll: Desplazarse.

SGBD: Sistema de Gestión de Base de Datos.

Stakeholder: Interesado/a.

Testing Data: Datos de Prueba.

Training Data: Datos de Entrenamiento.

WBS: Servicio Web.

Anexos

A. Complemento de las tablas de base de datos:

Clasificación:

Table

Clasificación

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Persona	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
id_Rutinas	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
id_Recetas	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
id_Dieta	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
id_Control_de_nutricion	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

```
1 CREATE TABLE "Clasificación" (  
2   "id_Persona" INTEGER,  
3   "id_Rutinas" INTEGER,  
4   "id_Recetas" INTEGER,  
5   "id_Dieta" INTEGER,  
6   "id_Control_de_nutricion" INTEGER,  
7   "id_Estado_de_Bienestar" INTEGER,  
8   FOREIGN KEY("id_Dieta") REFERENCES "Dieta"("id_Dieta"),  
9   FOREIGN KEY("id_Estado_de_Bienestar") REFERENCES "Estado_de_Bienestar"("id_Estado  
10  FOREIGN KEY("id_Control_de_nutricion") REFERENCES "Control de Nutrición"("id_Control_c  
11  FOREIGN KEY("id_Persona") REFERENCES "Persona"("id_Persona"),  
12  FOREIGN KEY("id_Recetas") REFERENCES "Recetas"("id_Recetas"),  
13  FOREIGN KEY("id_Rutinas") REFERENCES "Rutinas"("id_Rutinas")
```

Cancel OK

Control_de_nutricion:

Table

Control de Nutrición

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Control_de_nutricion	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
nombre_del_Alimento	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
imagen	BLOB	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
unidad_de_la_cantidad	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
sodio	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
azucares_totales	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
grasa_saturada	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

```
1 CREATE TABLE "Control de Nutrición" (  
2   "id_Control_de_nutricion" INTEGER PRIMARY KEY AUTOINCREMENT,  
3   "nombre_del_Alimento" TEXT,  
4   "imagen" BLOB,  
5   "unidad_de_la_cantidad" REAL,  
6   "sodio" REAL,  
7   "azucares_totales" REAL,  
8   "grasa_saturada" REAL,  
9   "energia" REAL,  
10  "calorias" REAL
```

Cancel OK

Datos_Medicos:

Table

Datos_Medicos

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Datos_Medicos	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
peso	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
talla	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
alergias	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
restricciones	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_Persona	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

1 CREATE TABLE "Datos_Medicos" (
2   "id_Datos_Medicos" INTEGER PRIMARY KEY AUTOINCREMENT,
3   "peso" REAL,
4   "talla" REAL,
5   "alergias" TEXT,
6   "restricciones" TEXT,
7   "id_Persona" INTEGER,
8   FOREIGN KEY("id_Persona") REFERENCES "Personas"("id_Persona")
9 );

```

Cancel OK

Dieta:

Table

Dieta

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Dieta	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
titulo	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
imagen	BLOB	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
descripcion	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
referencia	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

1 CREATE TABLE "Dieta" (
2   "id_Dieta" INTEGER PRIMARY KEY AUTOINCREMENT,
3   "titulo" TEXT,
4   "imagen" BLOB,
5   "descripcion" TEXT,
6   "referencia" TEXT
7 );

```

Cancel OK

Estado_del_Bienestar

Table

Estado_del_Bienestar

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Estado_del_Bienestar	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
estado_bienestar	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

1 CREATE TABLE "Estado_del_Bienestar" (
2   "id_Estado_del_Bienestar" INTEGER PRIMARY KEY AUTOINCREMENT,
3   "estado_bienestar" TEXT
4 );

```

Cancel OK

Recetas

Table

Recetas

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Recetas	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
titulo	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
imagen_de_la_receta	BLOB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ingredientes	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
breve_descripcion	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```
1 CREATE TABLE "Recetas" (  
2   "id_Recetas" INTEGER PRIMARY KEY AUTOINCREMENT,  
3   "titulo" TEXT,  
4   "imagen_de_la_receta" BLOB,  
5   "ingredientes" TEXT,  
6   "breve_descripcion" TEXT,  
7   "descripcion_general" TEXT,  
8   "web_link" TEXT  
9 );
```

Cancel OK

Rutinas:

Table

Rutinas

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Rutinas	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
id_Pierna	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_Brazo	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_Abdomen	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_Gluteo	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```
1 CREATE TABLE "Rutinas" (  
2   "id_Rutinas" INTEGER PRIMARY KEY AUTOINCREMENT,  
3   "id_Pierna" INTEGER,  
4   "id_Brazo" INTEGER,  
5   "id_Abdomen" INTEGER,  
6   "id_Gluteo" INTEGER,  
7   FOREIGN KEY("id_Brazo") REFERENCES "Subrutina_Brazo"("id_Subrutina_Brazo"),  
8   FOREIGN KEY("id_Gluteo") REFERENCES "Subrutina_Gluteo"("id_Subrutina_Gluteo"),  
9   FOREIGN KEY("id_Abdomen") REFERENCES "Subrutina_Abdomen"("id_Subrutina_Abdomen"),  
10  FOREIGN KEY("id_Pierna") REFERENCES "Subrutina_Pierna"("id_Pierna")  
11 );
```

Cancel OK

Subrutina_Abdomen:

Table

Subrutina_Abdomen

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Subrutina_Abdomen	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
dia	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ejercicio_del_Dia	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
descripcion	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

1 CREATE TABLE "Subrutina_Abdomen" (
2   "id_Subrutina_Abdomen" INTEGER PRIMARY KEY AUTOINCREMENT,
3   "dia" TEXT,
4   "ejercicio_del_Dia" TEXT,
5   "descripcion" TEXT
6 );

```

Cancel OK

Subrutina_Brazo:

Table

Subrutina_Brazo

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Subrutina_Brazo	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
dia	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ejercicio_del_Dia	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
descripcion	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

1 CREATE TABLE "Subrutina_Brazo" (
2   "id_Subrutina_Brazo" INTEGER PRIMARY KEY AUTOINCREMENT,
3   "dia" TEXT,
4   "ejercicio_del_Dia" TEXT,
5   "descripcion" TEXT
6 );

```

Cancel OK

Subrutina_Gluteo:

Table

Subrutina_Gluteo

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Subrutina_Gluteo	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
dia	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ejercicio_del_Dia	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
descripcion	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

1 CREATE TABLE "Subrutina_Gluteo" (
2   "id_Subrutina_Gluteo" INTEGER PRIMARY KEY AUTOINCREMENT,
3   "dia" TEXT,
4   "ejercicio_del_Dia" TEXT,
5   "descripcion" TEXT
6 );

```

Cancel OK

Subrutina_Pierna:

Table

Subrutina_Pierna

Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
id_Pierna	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
dia	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ejercicio_del_Dia	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
descripcion	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

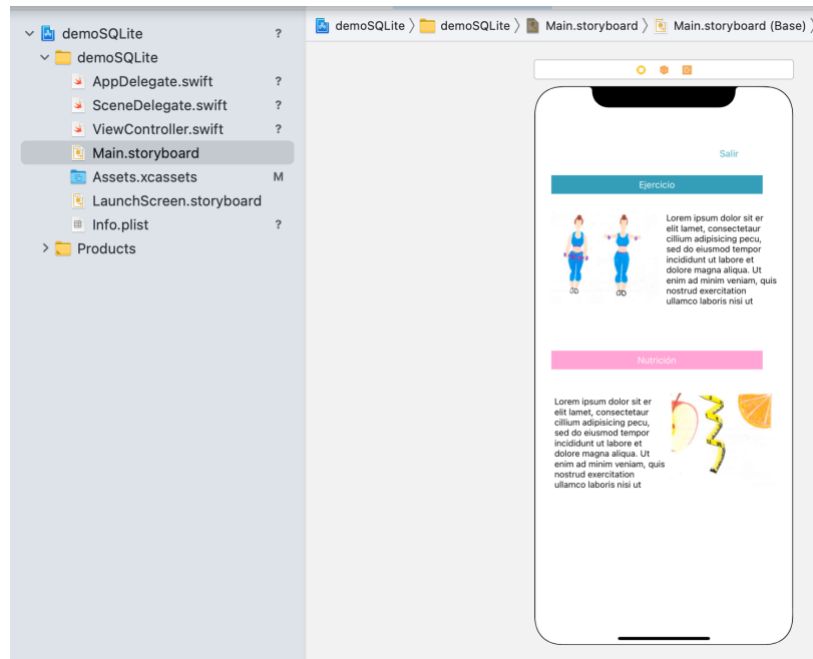
1 CREATE TABLE "Subrutina_Pierna" (
2   "id_Pierna" INTEGER PRIMARY KEY AUTOINCREMENT,
3   "dia" TEXT,
4   "ejercicio_del_Dia" TEXT,
5   "descripcion" TEXT
6 );

```

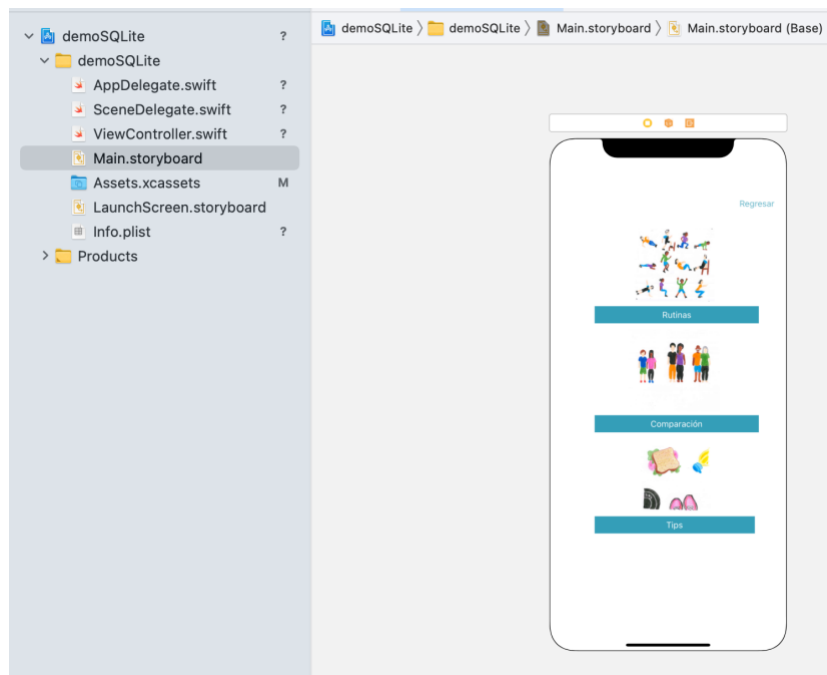
Cancel OK

B. Complemento de imágenes:

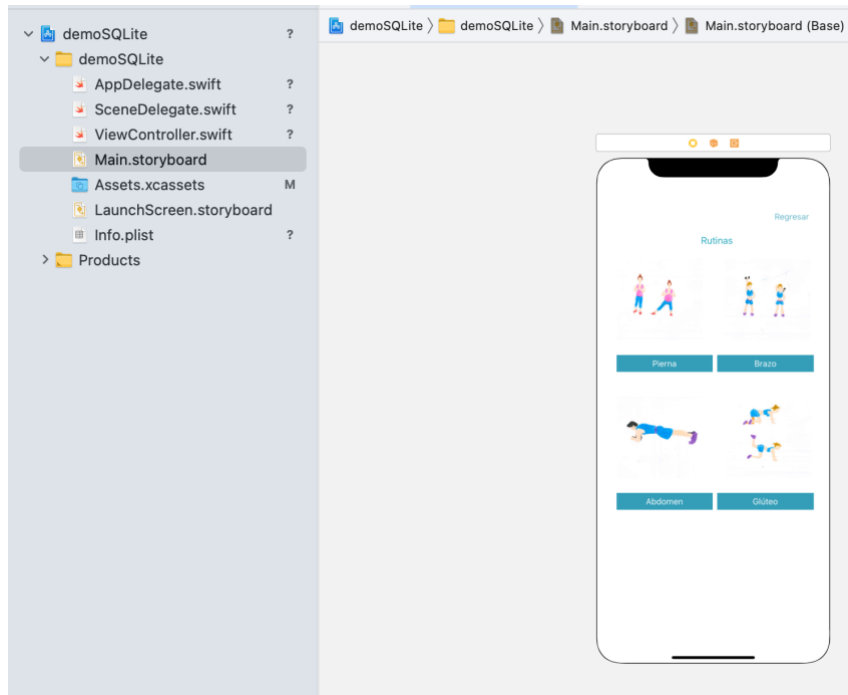
Ejercicio y Nutrición



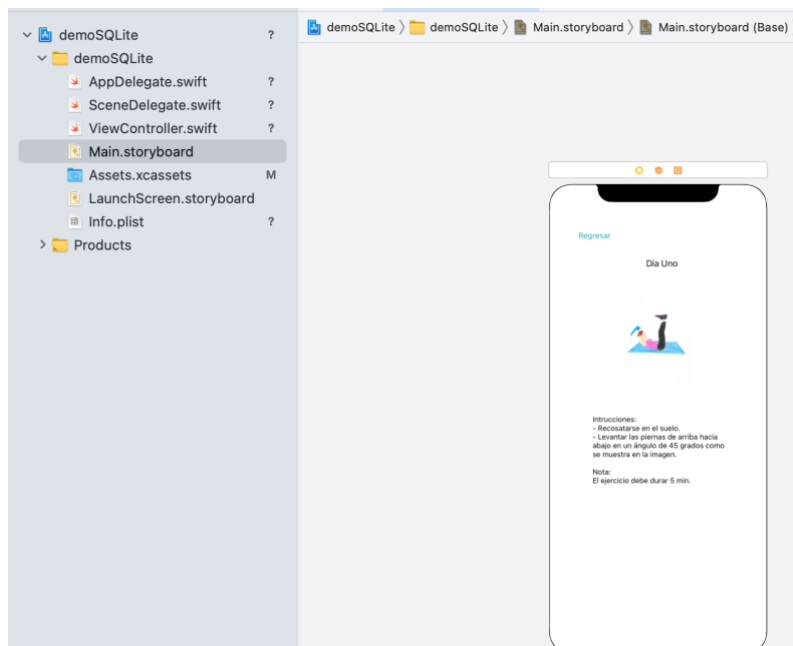
Menú Ejercicio



Rutinas



Rutina particular



C. El sistema se encuentra disponible en:

<https://github.com/eduardogarcia13/TesisPro.git>

D. Tabla de “testing” en una implementación real:

Caso de Uso asociado	Nombre de Caso de Uso correspondiente.
Paso a ejecutar	Características con las que debe contar el sistema.
Rol que Ejecuta	Persona que ejecuta el sistema, Usuario, QA, Líder de Proyecto, etc.
Prueba Exitosa	Valor para afirmar que la prueba fue exitosa o no.
Comentarios	Posibles comentarios para mejorar el sistema, en caso de que la prueba fuera no exitosa, los comentarios orientan al desarrollador a la mejora del sistema.
Evidencia	Captura de pantalla correspondiente.