

# Etapa 1 del Proyecto Integrador - Bases de Datos I

Tomás Rando - 14004

Marzo - 2024

## 1 Índice

<b>1 Índice</b>	<b>1</b>
<b>2 Introducción</b>	<b>1</b>
<b>3 Consignas</b>	<b>2</b>
3.1 Definir el nombre de la empresa, por ejemplo "Nexo Turismo": . . . . .	2
3.2 Describir las actividades que realiza la empresa. Ejemplos: Nexo Turismo realiza excursiones en el interior de Mendoza, organiza salidas para empresas, etc: . . . . .	3
3.3 Describir la estructura de la empresa describiendo las áreas que la conforman, descripción de cada una de ellas y objetivos: . . . . .	3
3.4 Definir un organigrama de acuerdo al punto anterior. . . . .	5
3.5 Brindar otra información que considere relevante para el proyecto. . . . .	5
3.5.1 Producto de la empresa: . . . . .	5
3.5.2 Público Objetivo: . . . . .	6
3.5.3 Ubicación: . . . . .	6
3.5.4 Estructura Jerárquica: . . . . .	6
3.5.5 Veganismo o Vegetarianismo: . . . . .	6
3.5.6 Canales de distribución: . . . . .	6
3.5.7 Política de precios: . . . . .	7
3.5.8 Perspectiva del futuro: . . . . .	7
<b>4 Conclusión</b>	<b>7</b>

## 2 Introducción

En el presente documento se realiza la primer parte o etapa del proyecto integrador de la materia "Teoría de bases de datos I" de la carrera Ciencias de la Computación del año 2024. En ella, se pide realizar básicamente el resumen del funcionamiento de la empresa. En mi caso, mi tema designado fue "Aplicación de gestión de dietas y nutrición para jóvenes deportistas" por lo que el proyecto comienza a ser elaborado tomando en cuenta ese enfoque. A partir de la sección 3 se enumeran las diferentes consignas a realizar.

Además, el presente documento se elabora utilizando el software en línea Overleaf, respetando los formatos estipulados en la guía de actividades de la cátedra.

Cabe aclarar que la primer parte del proyecto lleva asociado una parte importante de investigación de la importancia de la buena nutrición y el deporte en los jóvenes, para al final poder llegar a elaborar un producto o proyecto con un objetivo bien claro. Para ello, se utilizaron diferentes fuentes, entre las que se incluyen chatGPT, Wikipedia y una médica profesional de la provincia de Mendoza. Por último, se buscaron diferentes aplicaciones que siguen temáticas parecidas para observar los enfoques que estas realizan del tema en cuestión. Algunos ejemplos de éstas últimas son: Google Fit, Lose It o My Fitness Pal, sin embargo, éstas están más enfocadas a la parte de las calorías que a brindar una dieta en sí, pero a pesar de esto se nota el compromiso por remarcar la importancia de la actividad física para lograr una mejor calidad de vida.

### 3 Consignas

#### 3.1 Definir el nombre de la empresa, por ejemplo "Nexo Turismo":

El nombre de la empresa elegido fue "Young and Healthy Life", la cual en español significa "Vida joven y saludable". Se eligió dicho nombre ya que se encuentra estrechamente relacionado con el tema designado del proyecto. Es decir, se trata de una aplicación de gestión de dietas y nutrición para jóvenes deportistas, por lo que se eligió un nombre que resumiera de alguna manera el objetivo o razón de existir de la empresa en específico.

El objetivo de la empresa sería intentar ayudar a los chicos y chicas jóvenes que realicen deportes o estén empezando a realizar, para que tengan hábitos de consumo de alimentos y bebida saludables que les permita potenciar su rendimiento en el área de su deporte en específico. Esto es ya que es sumamente importante promover los valores de la salud en la población para terminar con muchas de las afecciones producidas por la falta del deporte o actividad física, por ejemplo:

- **Obesidad:** La falta de actividad física puede conducir al aumento de peso, llegando un punto en el que se le podría considerar obesidad. Si a esto se le suma una mala alimentación, se podría ocasionar un desequilibrio energético, ya que la ingesta de calorías superaría el gasto de ellas, lo que podría conducir a un aumento enorme de peso y a problemas de la salud relacionadas con la obesidad.
- **Enfermedades Cardiovasculares:** La falta de ejercicio también puede aumentar el riesgo de enfermedades cardiovasculares, es decir, enfermedades relacionadas a nuestro corazón, algunas de estas pueden ser la hipertensión arterial, enfermedad coronaria, accidentes cerebrovasculares y otras más. Sin embargo, con una buena alimentación y la realización regular de ejercicio se logra mantener un corazón saludable, mejorando la circulación sanguínea, reduciendo la presión y el colesterol malo, y aumentando el bueno.
- **Problemas musculoesqueléticos:** La falta de actividad física puede debilitar los músculos y huesos, lo que a su vez aumenta el riesgo de lesiones y enfermedades musculoesqueléticas, como podrían llegar a ser la osteoporosis, artritis, dolor lumbar y problemas de espalda.
- **Disminución de la función pulmonar:** La falta de ejercicio podría llegar a dar como resultado la disminución de la función pulmonar y la capacidad aeróbica de nuestro cuerpo, lo que podría provocar dificultad para respirar, fatiga y menor resistencia física.
- **Problemas relacionados a la salud mental:** Hacer ejercicio regularmente promueve una mejor salud mental y bienestar emocional. La falta de actividad física podría aumentar el riesgo de contraer enfermedades emocionales como lo son la depresión y la ansiedad, así como disminuir la autoestima y la calidad de vida en general. Realizando actividad física se puede lograr un aumento del bienestar emocional, por ejemplo logrando una disminución del sentimiento de soledad, en el caso de que el deporte sea realizado en un equipo, ya que esto puede derivar en lograr un sentimiento de pertenencia de la persona al equipo. Por último, se puede lograr una reducción importante de la ansiedad o estrés, al gastar la energía acumulada diariamente.
- **Mayor riesgo de enfermedades crónicas:** La falta de ejercicio está relacionada con un mayor riesgo de desarrollar una serie de enfermedades crónicas como la diabetes tipo 2, algunos tipos de cáncer, enfermedades del hígado y muchas otras.

Con el propósito de reducir las enfermedades o afecciones previamente descriptas, la empresa asumirá el compromiso de promover la salud y el bienestar de los jóvenes que quieran ser deportistas desde una perspectiva dietética y nutricional. La empresa se compromete a ofrecer una aplicación que apoye y ofrezca recursos para fomentar el interés de los jóvenes en participar en actividades deportivas, además de facilitar el proceso de obtención de una dieta saludable y disfrutable. Creemos en la evidencia científica que demuestra la cantidad de beneficios que tiene la práctica regular de actividad física deportiva junto a una correcta nutrición en la salud física, mental y emocional de las personas más jóvenes.

La estrategia de la empresa "Young and Healthy Life" se centrará en ofrecer asesoramiento nutricional específicamente diseñado para apoyar y potenciar el rendimiento deportivo de los jóvenes, para así promover hábitos alimenticios saludables y equilibrados que puedan permanecer en la vida de la o el joven durante mucho tiempo. Se busca informar a las personas de este rango etario con el

conocimiento y las herramientas necesarias para tomar decisiones informadas sobre su alimentación y su estilo de vida.

Además de esto, obviamente nuestro objetivo es brindar apoyo a los jóvenes que ya se encuentren involucrados en el desarrollo de actividad física, sea cual sea, con el objetivo de fomentar la persistencia y informar acerca de los beneficios para que el o la joven pueda mantenerse saludable y feliz. Esto es ya que siempre pueden existir dificultades en el camino del deporte y la correcta nutrición, sin embargo, el plan es brindar nuestro apoyo y ayuda para mitigar estos problemas y lograr que los jóvenes puedan mantener los hábitos saludables para el resto de sus vidas.

### **3.2 Describir las actividades que realiza la empresa. Ejemplos: Nexo Turismo realiza excursiones en el interior de Mendoza, organiza salidas para empresas, etc:**

En el caso de "Young and Healthy Life" se realizan diferentes actividades inicialmente en la zona de Mendoza también. Algunas de las actividades que realiza sería el seguimiento de chicos jóvenes deportistas que estén siguiendo una dieta específica. Es decir, por ejemplo, que se recolectan y almacenan algunos de sus datos, como su nombre completo, su peso actual, la dieta actualmente seguida, deporte que realizan, horas que duermen por día, agua que deberían tomar por día, medida en vasos, el índice de masa corporal, horas por semana que se realiza el deporte, altura, edad, sexo, entre otros.

Además, dependiendo del deporte que se realice se podría asignar una dieta específica a seguir, podría almacenarse una dieta específica para un cierto tipo de deporte y diferenciarlas dependiendo si la persona necesitase o no aumentar el peso, capaz pudiendo asignarla o relacionarla con el peso actual del usuario y las horas por semana que realizan el ejercicio. También, pudiendo efectuar la actualización del mismo en el caso de que estos datos sean actualizados por un cambio en los hábitos o un cambio positivo impulsado por la dieta asignada previamente, o, en el caso de que no se haya seguido correctamente la dieta otorgada, un cambio negativo, por ejemplo, el aumento de peso a niveles no saludables.

Otra actividad que se podría llegar a realizar sería calificar el peso de la persona, teniendo en cuenta su edad, sexo y altura en una escala que podría ser : "Bajo", "Adecuado", "Excesivo". Esto se realizaría para tener un indicador más "comprensible" o cercano al usuario, para que pueda entender de manera más clara en que estado se encuentra.

### **3.3 Describir la estructura de la empresa describiendo las áreas que la conforman, descripción de cada una de ellas y objetivos:**

La empresa en cuestión, "Young and Healthy Life", posee diferentes áreas que se encargarán de diferentes tareas sumamente importantes dentro de la misma. Estas son:

1. Área de desarrollo de la aplicación: Es un área que se enfoca en el diseño, la programación y todo lo relacionado con el desarrollo de la aplicación como tal. Implica lo que tiene que ver con la interfaz del usuario, la experiencia del usuario, la funcionalidad de la aplicación y la integración de características específicas relacionadas con la dieta y la nutrición para los jóvenes hombres y mujeres deportistas. Ésta será el área en la que nos enfocaremos más que nada durante el desarrollo de esta consigna, ya que será la que se intentará desarrollar a lo largo del proyecto, con la ayuda de pequeñas partes de otras áreas.
2. Dietas y nutrición: Es el área que contaría con la presencia de profesionales en el área de nutrición y dietética. Se irán a desarrollar planes de alimentación o dietas, junto con posible contenido educativo e informativo, utilizando los conocimientos profesionalmente adquiridos para lograr satisfacer las necesidades específicas de los jóvenes deportistas. Se podría incluir gente especializada en dietas, nutricionistas con orientación en el área de deportes y médicos especializados en nutrición. Es la que brindará información al área "Desarrollo de la aplicación" para así poder lograr una aplicación funcional y precisa.
3. Investigación y desarrollo: Es un área de la empresa que estaría fuertemente relacionada con el área anterior, pues es la que se encargaría de mantenerse actualizada con respecto a los avances en la ciencia de la nutrición orientada al deporte. Además, debería encargarse de investigar nuevas formas o estrategias para lograr mejorar el rendimiento deportivo a través de la nutrición, además de la calidad de vida del usuario en cuestión. Esto podría llevar a elaborar

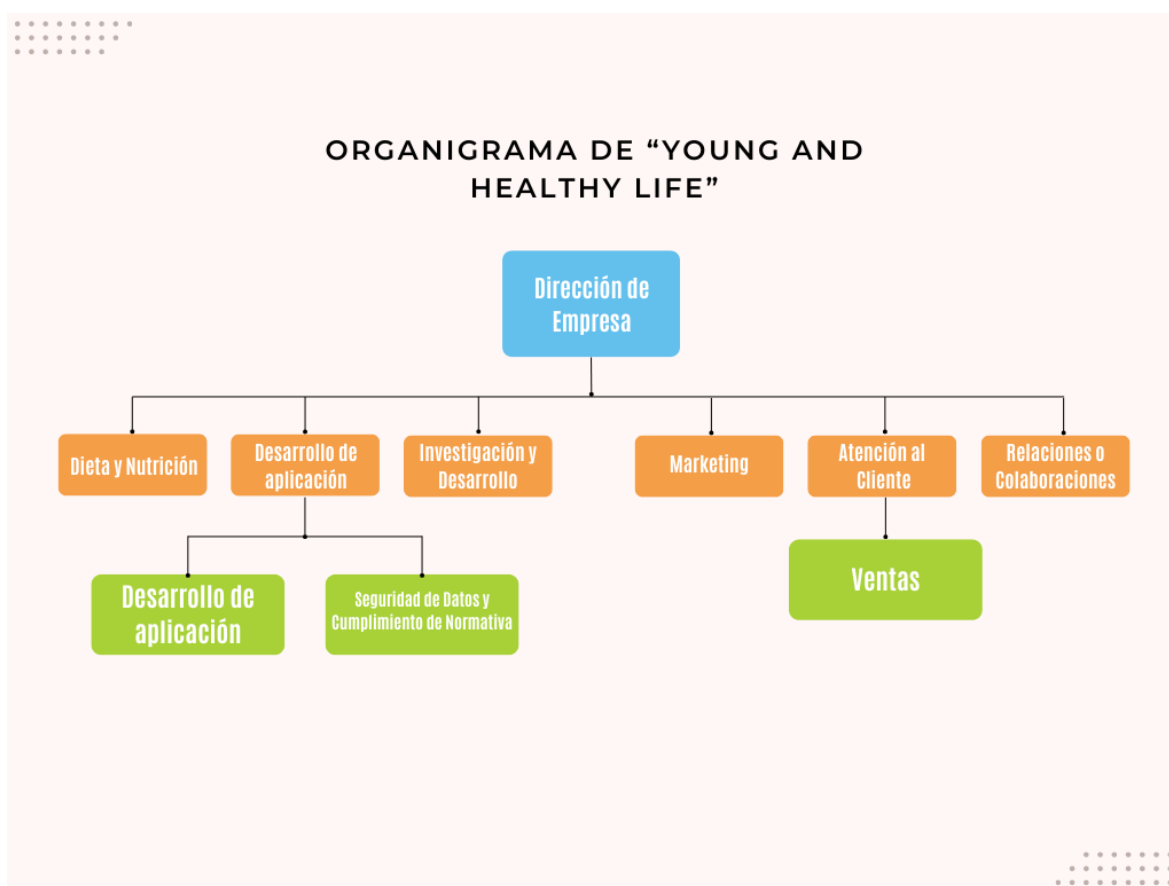
algunos estudios clínicos con los datos recopilados de los usuarios y se podría colaborar con diversas instituciones académicas o instituciones ligadas al mundo del deporte.

4. Marketing: En el caso de que la aplicación se desarrollara a un nivel en el que se pudiese comercializar, se necesitaría de un adecuado equipo de marketing y ventas que pueda lograr llegar a la audiencia objetivo de la empresa y promover la aplicación utilizando diferentes técnicas. Esto podría incluir por ejemplo la gestión de redes sociales, la creación de contenido promocional para estas, la publicidad digital, las relaciones públicas e incluso la posible colaboración con deportistas y entrenadores profesionales que sean influyentes en el mundo del ámbito deportivo. Esta será un área que no será desarrollada exhaustivamente en el desarrollo del proyecto, ya que el objetivo principal del proyecto en la materia no sería la comercialización de la aplicación, sin embargo, es importante nombrarla como parte importante de la empresa ya que sin marketing, no hay usuarios y sin usuarios, el producto no sirve.
5. Atención al cliente: Es el área que se encargaría de ofrecer el servicio de atención al cliente. Debería ser un servicio de alta calidad para responder a las preguntas y preocupaciones que los usuarios pudiesen llegar a tener. Además, el equipo debería ser capaz de proveer o proporcionar asistencia técnica para así poder garantizar una experiencia cómoda y positiva con la aplicación. Esto implicaría tener un equipo dedicado al soporte y al desarrollo de recursos de ayuda y tutoriales. Las personas propias de este área deberían ser capaces de resolver los problemas del usuario correctamente y saber sobrellevar situaciones típicas de la relación cliente - empresa de una manera amable y atenta. Nuevamente, en el desarrollo del presente proyecto no será un área especialmente abordada por no ser el objetivo principal de la materia.
6. Seguridad de datos y cumplimiento de la normativa: Será el área encargada de que la aplicación, al involucrar datos personales y de salud de los usuarios, garantice la seguridad y privacidad de la información. Implicaría seguir las regulaciones establecidas referidas a la protección de datos y la incorporación de medidas de seguridad contra los posibles ataques para robar información que se pudiesen sufrir. Además, sería el área encargada de realizar investigaciones periódicas para la detección de posibles vulnerabilidades en el sistema. La función de este área, en el presente proyecto, será realizada por el área de "Desarrollo de la aplicación". Sin embargo, con un equipo más grande, podría estar separada, por lo que se nombra como un área aparte en la empresa.
7. Relaciones o colaboraciones: Es el área de la empresa que se encargaría de buscar posibles colaboraciones o asociaciones con organizaciones deportivas, gimnasios, equipos, federaciones, marcas de alimentos o suplementos, para ampliar el alcance de la aplicación y ofrecer mayor cantidad de servicios a los usuarios. Generalmente, ciertas instituciones especializadas en ciertos temas tienen un mayor grado de conocimiento con respecto a las necesidades de los usuarios en cuestión, por lo que sería de gran utilidad la escucha atenta hacia estas asociaciones con el fin de lograr una mejor y correcta resolución de las posibles necesidades de los jóvenes deportistas.
8. Dirección de la empresa: Sería el área encargada de "dirigir" la empresa. Podría realizar diferentes actividades, como por ejemplo la designación de líderes para cada área. Nuevamente, al no ser el objetivo principal de la materia, este área tampoco será uno de los más abordados en el proyecto. Sin embargo, en una empresa "real", sería una de las áreas más importantes que existen, ya que sin dirección es muy difícil que la empresa logre seguir un rumbo específico sin desviaciones entre medio.

Habiendo nombrado las áreas de la empresa "Young and Healthy Life", nos enfocaremos en la más importante en la realización de este proyecto: El área de desarrollo de la aplicación. Al proyecto tratarse de la realización de una aplicación, sería correcto interpretarla como la principal, solo en el contexto del proyecto. Como fue mencionado, será el área que tratará el desarrollo de la aplicación propiamente dicha. Es decir, se desarrollará la parte referida a la interfaz del usuario, la experiencia del usuario, la funcionalidades de la aplicación, la integración de toda la información referida a la dieta y la nutrición para cada deporte y característica de los usuarios en conjunto. Además, será el área que se encargue del desarrollo de la base de datos de la aplicación y, posiblemente, el desarrollo de una pequeña interfaz visual que ayude a la correcta interpretación de la información. Este área utilizará ciertas tecnologías que fueron propuestas por la cátedra para el desarrollo del proyecto. Principalmente, se utilizará Overleaf para realizar toda la documentación escrita de la aplicación en cuestión. Además, se utilizará PostgreSQL y PGAdmin/Dbeaver para la realización de la base de datos de la empresa. A su vez, también se utilizará el lenguaje de programación Java para el desarrollo de la aplicación, junto a su IDE (El entorno de desarrollo integrado correspondiente).

### 3.4 Definir un organigrama de acuerdo al punto anterior.

El organigrama que se presenta a continuación se realizó utilizando el editor en línea Canva. Este organigrama se diseñó teniendo en cuentas las áreas (o departamentos) descriptos previamente en el punto de la consigna anterior. Se elaboró con el objetivo de representar visualmente de manera clara y comprensible la estructura organizativa de la empresa. Básicamente, el organigrama aquí presentado nos muestra la distribución y la relación entre las diversas áreas funcionales de la empresa, lo que facilita la comprensión del funcionamiento que tiene internamente y la jerarquía de los departamentos o áreas. A través de esta representación más gráfica se ofrece una visión general de como se organizan y interactúan entre sí las distintas áreas y responsabilidades dentro de la empresa "Young and Healthy Life", lo que contribuye a una mejor comprensión de la dinámica. Respecto al orden jerárquico, el único área que estaría en un nivel más alto sería la dirección de la empresa, esto es ya que sería la encargada de designar a los líderes de las demás áreas. Los otros departamentos, entonces, compartirían el mismo nivel jerárquico, con la excepción de la posible existencia de un jefe o líder de área que sea el encargado de mantener el rumbo y objetivo del desarrollo de la aplicación.



### 3.5 Brindar otra información que considere relevante para el proyecto.

En esta sección se brinda contexto de la empresa, para poder lograr un mejor entendimiento del producto al aumentar el grado de contexto de la misma.

#### 3.5.1 Producto de la empresa:

El producto que la empresa está desarrollando, como ya fue mencionado, será una aplicación de gestión de dietas y nutrición para jóvenes deportistas. La aplicación en cuestión será nombrada "HealthieR", ya que, si bien es un nombre simple, resume y canaliza perfectamente el objetivo actual de la empresa. Además, es un nombre corto y bastante atractivo a la vista, por lo que, al menos por ahora, ese será el nombre del producto que se irá a desarrollar.

### **3.5.2 Público Objetivo:**

La aplicación que está siendo desarrollada por la empresa está destinada jóvenes deportistas, como ya fue mencionado, sin embargo, nunca mencionamos el rango etario al que está destinado. Después de un largo análisis se llegó a la conclusión de que la aplicación sería desarrollada para personas de entre 14 y 26 años. Y, específicamente, los que estén realizando algún deporte o estén interesados en iniciar su vida deportiva para lograr cambiar sus hábitos y lograr una calidad de vida mejor.

### **3.5.3 Ubicación:**

La empresa, al menos inicialmente, estará ubicada en la provincia de Mendoza, en el país Argentina. Esta decisión de establecerse inicialmente en la provincia es el resultado de consideraciones más personales del fundador de la empresa.

Como fundador, mi nacionalidad es Argentina y mi actual residencia es la gran provincia de Mendoza, por lo cuál esto ha impactado de manera sustancial en la decisión. La provincia, al estar ubicada en Argentina, tiene acceso a una cantidad de alimentos de gran calidad, el país es una excelente fuente de carnes y productos vegetales que brindarán la posibilidad de seguir las dietas ofrecidas por la aplicación.

A pesar de todo esto, si bien la primera aproximación al mercado de la aplicación sería en Mendoza, no sería incorrecto pensar en las posibilidades de expansión futuras. Se podría llegar al resto de Argentina e incluso podría expandirse internacionalmente al resto de países. Con el mundo globalizado de este momento de la historia, no sería incoherente pensar en ésta posibilidad.

### **3.5.4 Estructura Jerárquica:**

La estructura de la empresa es jerárquica, en el sentido de que habrá una dirección que comande a todas las áreas (O departamentos) de ésta. Sin embargo, todas las demás áreas tendrán el mismo nivel jerárquico dentro de la misma, con la excepción de que pudiera haber un líder o jefe de división, con el único objetivo de que no se pierda el rumbo dentro del departamento específico.

### **3.5.5 Veganismo o Vegetarianismo:**

En la etapa inicial del desarrollo de la aplicación se tomó la decisión de no incluir dietas completamente veganas o vegetarianas. En lugar de eso, nos enfocaremos en satisfacer las necesidades nutricionales de las personas que tengan una dieta omnívora actualmente. Esta decisión se basa en una serie de prioridades que consideramos como resultado del establecimiento de los objetivos iniciales de la aplicación.

Es decir, nuestro objetivo principal en la primer etapa es intentar proporcionar la mejor experiencia para la mayor cantidad de usuarios posibles, y, estos serían los usuarios que posean una dieta de carácter omnívoro.

Por lo tanto, si bien las dietas veganas y vegetarianas no estarían contempladas en la primera versión de la aplicación desarrollada por la empresa, en el futuro estamos abiertos a la posibilidad de incluirlas en futuras actualizaciones de la misma.

### **3.5.6 Canales de distribución:**

En el caso hipotético de que se llegase a desarrollar una versión completamente funcional y agradable para el usuario, la empresa podría considerar que la misma fuese distribuida inicialmente en plataformas menos automatizadas, lo que permitiría una mayor interacción con los usuarios por parte de la empresa. Esto podría incluir la distribución a través de páginas web propias o redes sociales, donde la empresa podría proporcionar asistencia técnica y recopilar retroalimentación de los usuarios de una manera más directa y cercana a ellos.

Sin embargo, si la aplicación fuese lanzada de manera exitosa, recibiendo una valoración positiva por parte de los usuarios, se podría llegar a introducir en nuevas tiendas de aplicaciones más reconocidas mundialmente como lo pueden ser Google Play Store, para teléfonos Android, o Apple Store, para teléfonos iOS.

### **3.5.7 Política de precios:**

En el hipotético escenario en el que la aplicación desarrollada por nuestra empresa finalmente llegue al mercado, nos comprometeríamos a ofrecerla de manera completamente gratuita, es decir, con costo igual a 0 para todos los usuarios que deseen iniciarse en una vida más sana. Esta decisión se basa en nuestra firme convicción y defensa de la idea de que el acceso al software debería ser un derecho universal, sin importar el nivel socio económico de los usuarios. Además, creemos necesario no olvidar que lo más importante serían los usuarios y la necesidad de ayudarlos a mantenerse y potenciarse en el mundo de la actividad física deportiva.

También creemos que el software libre y de código abierto no solo libera el acceso a la tecnología, sino que también ayuda a la innovación y al progreso en la sociedad. Al eliminar las transacciones para acceder a nuestra aplicación, se esperará no solo ayudar a los jóvenes de la provincia, si no también contribuir a la innovación en el ámbito de las aplicaciones relacionadas a la nutrición en todo el mundo.

Si bien nuestra aplicación se ofrecería de forma gratuita si saliese al mercado, desde la empresa se reconoce la importancia de generar ingresos para el mantenimiento y el futuro crecimiento de la empresa. Para lograr esto de manera ética y sostenible, sin que afecte a los usuarios, planeamos implementar estrategias que no comprometan la experiencia de los jóvenes usuarios de la aplicación. Una de estas estrategias podría ser la inclusión de anuncios no invasivos dentro de la aplicación. Estos anuncios serían cuidadosamente seleccionados y presentados de manera discreta, sin interrumpir la función principal de la aplicación ni afectar de manera negativa la experiencia del usuario en cuestión.

Además, en el futuro se explorarán otras maneras de generación de ingresos, como la posibilidad de ofertar funciones premium opcionales o la colaboración con asociaciones deportivas y nutricionales que compartan nuestros propósitos y objetivos. Igualmente, estas funcionalidades serían de carácter complementario, no formando parte de las principales de la aplicación, ya que serían solo una manera de recibir indirectamente donaciones de los usuarios que estén interesados en ayudarnos a mantener nuestra aplicación en funcionamiento y en desarrollo. Siempre se realizaría todo teniendo en cuenta lo antes estipulado acerca de ofrecer el software de forma gratuita, es decir, con costo 0.

En resumen, nuestra empresa se compromete a ofrecer nuestra aplicación de forma gratuita para todos los jóvenes usuarios que quieran utilizar y dar una oportunidad a la aplicación, creyendo firmemente en el poder que podría llegar a tener el software que crearemos y en nuestra capacidad para encontrar modelos de negocio sostenibles que apoyen nuestra misión sin afectar la experiencia del usuario.

### **3.5.8 Perspectiva del futuro:**

A corto plazo se quiere lograr llegar al desarrollo final de la aplicación de gestión de dietas y nutrición para jóvenes y adolescentes. Sin embargo, en el caso de que el desarrollo fuese mejor de lo esperado, pudiendo lograr una base de datos correctamente desarrollada, útil y ordenada, se podría proyectar en el futuro un avance mayor en la producción de la aplicación. Es decir que, por ejemplo, se podría mejorar la parte de la interfaz gráfica, que inicialmente será reducida. Además, se podría pensar en incluir nuevas funcionalidades que no serán incorporadas en la primera versión de la aplicación. Sin embargo, todo depende de como se desarrolle la primera versión del proyecto en cuestión y el entusiasmo por el tema trabajado.

## **4 Conclusión**

Se finaliza la etapa 1 del proyecto de la materia "Teoría de Bases de Datos" de la carrera Licenciatura en Ciencias de la Computación. Básicamente se realizó una descripción de la empresa, de su funcionamiento y de las actividades que realizará la aplicación en cuestión, de acuerdo a las consignas solicitadas. Además, en la consigna número 5 se brindaron diferentes datos referidos al contexto de la empresa, con el objetivo de lograr una mejor conexión con las ideas de ésta.

# Etapa 2 - Proyecto Bases de Datos

Tomás Rando Muñoz - 14004

Marzo 2024

<b>1 Introducción.</b>	<b>1</b>
<b>2 Consignas.</b>	<b>1</b>
2.1 Enumerar y describir los motivos para el desarrollo del sistema. En el caso de la empresa Nexo, un ejemplo es el siguiente "se necesita organizar las excursiones de manera eficiente y comunicar la información de las mismas a los guías y turistas." . . .	1
2.2 Identificar los usuarios del sistema y sus características. . . . .	2
2.2.1 Usuarios del sistema. . . . .	2
2.2.2 Características de los usuarios: . . . . .	2
2.3 Describir los requisitos funcionales y no funcionales del sistema. . . . .	3
2.3.1 Requisitos funcionales: . . . . .	3
2.3.2 Requisitos no funcionales: . . . . .	3
2.4 Definir el alcance del sistema, es decir cuales funciones se incluirán y cuales no. Justificar. . . . .	3
2.4.1 Funciones incluidas: . . . . .	4
2.4.2 Funciones no incluidas: . . . . .	4
<b>3 Conclusión.</b>	<b>4</b>

## 1 Introducción.

Se realiza la etapa 2 del proyecto de la materia "Teoría de Bases de Datos I". En la misma, se trata contenido de la unidad 1 de la materia. Nuevamente, como en la primer etapa, cada consigna estará representada con una subsección de la sección 2 del documento. Para la realización del informe, se utiliza el editor online Overleaf y las especificaciones solicitadas por el profesor.

## 2 Consignas.

### 2.1 Enumerar y describir los motivos para el desarrollo del sistema. En el caso de la empresa Nexo, un ejemplo es el siguiente "se necesita organizar las excursiones de manera eficiente y comunicar la información de las mismas a los guías y turistas."

El motivo del desarrollo del sistema tiene que ver con el objetivo de la empresa. La empresa "Young and Healthy Life" tiene la misión de ayudar a jóvenes deportistas desde el punto de vista de la nutrición y la dieta, entonces, para ello necesita un medio por el cual se logre interactuar con los jóvenes para lograr cumplir con el objetivo de la compañía, por ello se pensó en la creación del sistema de gestión de dietas y nutrición.

El sistema permitirá resolver el problema principal de la empresa, lograr brindar el servicio para el cual fue pensada. Es decir, a través de él se podrá conectar con los usuarios para lograr una correcta interpretación de los datos del usuario y poder asignarle una correcta dieta, además de poder brindarle información relacionada a su situación actual, como lo pueden ser su IMC, el indicador de peso del usuario o un indicador que nos indique si el comportamiento actual del usuario es el correcto o necesitaría mejorarse para funcionar correctamente.



El sistema podría ser implementado tranquilamente por la empresa de manera "tradicional" o "no tecnológica", es decir, sin la creación de una aplicación propiamente dicha. Sin embargo, el motivo principal de la creación de un sistema es que nos trae la posibilidad de agilizar todos los procesos, es decir, es más fácil cargar datos, gestionar los datos, interpretarlos, encontrarlos y nos ahorra muchísimo tiempo que podremos destinar a hacer mejor nuestro sistema y de mejor calidad. Además, es una manera de acercarnos al usuario, ya que incluso podrían llegar a ser ellos mismos quienes realicen la carga de datos, ahorrando el tiempo que se necesitaría de no ser así y que fuese necesario que una persona de la empresa se comuniqué con el usuario para que ella misma le cargue los datos.

## **2.2 Identificar los usuarios del sistema y sus características.**

La empresa fue creada con el objetivo de ayudar más que nada a los jóvenes deportistas, teniendo esto en cuenta, se pueden destacar dos tipos de usuarios principales.

### **2.2.1 Usuarios del sistema.**

**Jóvenes deportistas:** Son jóvenes hombres y mujeres que tienen entre 14 y 26 años que actualmente se encuentran realizando algún tipo de deporte.

**Jóvenes con la intención de ser deportistas:** Son jóvenes hombres y mujeres, nuevamente que tienen edades entre 14 y 26 años que actualmente no se encuentran realizando algún tipo de deporte, pero que estén en el proceso de inserción a alguno.

Por el objetivo de la empresa, en principio las personas que no se encuentren dentro del rango etario mencionado o que no tengan intención de realizar deporte no serán tenidos en cuenta en este momento del desarrollo del sistema.

### **2.2.2 Características de los usuarios:**

Se solicitarán múltiples características de los usuarios, muchas de ellas mencionadas en la anterior etapa del proyecto.

- Nombre: Primer nombre del usuario
- Apellido: Primer apellido del usuario
- DNI: DNI Del usuario
- Peso actual: Peso que tiene el usuario al momento de ingresar el dato
- Deporte: Deporte que se realiza actualmente
- Dieta actual: En el caso de que se tenga una dieta asignada, poder identificar cual es
- Horas que se duerme por día: Horas que el usuario duerme en promedio por día
- Agua tomada por día: En vasos, cuanta cantidad se toma en promedio por día
- Horas por semana que se realiza el deporte seleccionado
- Altura: Altura del usuario
- Edad: Edad del usuario
- Sexo: Sexo del usuario
- Género: Género del usuario
- Número de teléfono: Número de teléfono del usuario para posibles notificaciones o comunicaciones por parte de la empresa
- Email: Email del usuario con los mismos fines que el número de teléfono
- (Posibilidad) Una meta del usuario: Es decir, el objetivo que quiere seguir el usuario con ayuda de la dieta. Por ejemplo, pérdida de peso, aumento de masa muscular, aumento de rendimiento deportivo, entre otros.
- (Posibilidad) Fumador: Para indicar si el usuario fuma regularmente

## **2.3 Describir los requisitos funcionales y no funcionales del sistema.**

### **2.3.1 Requisitos funcionales:**

- Alta de usuarios en el sistema: El sistema debería ser capaz de dar de alta a los usuarios con sus respectivos datos en el sistema.
- Baja de usuarios en el sistema: El sistema debería ser capaz también de dar de baja a los usuarios con sus respectivos datos en el sistema.
- Modificación de usuarios en el sistema: El sistema también debería ser capaz de manejar correctamente la modificación de los datos de los usuarios que estén dados de alta en el sistema.
- Asignación de dieta correctamente: El sistema debería ser capaz de asignar correctamente una dieta específica a un usuario teniendo en cuenta la información de este (Como el peso, altura, deporte realizado, objetivo, entre otros).
- Identificar correctamente información obtenida a partir de datos del usuario: Es decir, que se calcule y manejen bien la información calculada a partir de los datos ingresados por el usuario, es decir, por ejemplo, el IMC o el indicador de peso del usuario.
- Adaptarse a las modificaciones: Es decir, el sistema debería ser capaz de actualizar correctamente la información (Como la dieta recomendada, el IMC, el indicador de peso, entre otros) a partir de posibles modificaciones en los datos del usuario. Por ejemplo, si el usuario cambiase su objetivo de "Aumento de rendimiento deportivo" a "Pérdida de peso", sería lógico que la dieta seguida se actualice a una que le pueda garantizar un déficit calórico que le permita alcanzar la meta proporcionada.

### **2.3.2 Requisitos no funcionales:**

- Rendimiento: El sistema debería lograr tiempos de respuesta rápidos o medianamente aceptables para las consultas y altas/bajas/modificaciones de los datos.
- Escalabilidad: El sistema debería permitir o ser capaz de manejar la carga de bastantes usuarios y datos
- Mantenibilidad: El sistema debería facilitar la modificación y actualización de datos, intentando lograr una estructura ordenada y bien lograda.
- Integridad de los datos: El sistema debería ser capaz de lograr evitar la pérdida de datos y tratar de garantizar que sean correctos
- Usabilidad: En el caso de que se logre llegar a hacer una pequeña interfaz, intentar que sea lo más intuitiva posible y fácil de utilizar.

## **2.4 Definir el alcance del sistema, es decir cuales funciones se incluirán y cuales no. Justificar.**

Estas funciones son las elegidas para ser incluidas ya que creo que son las que más tienen que ver con la aplicación de gestión de dietas y nutrición para jóvenes deportistas que se solicitó. Las no incluidas considero que podrían ser buenas para otra instancia del sistema, pero no considero que sean las más relacionadas con el tema que se me asignó, sin embargo, en la primera etapa del proyecto estuve pensando en ellas porque me parecieron interesantes para una aplicación "Del mundo real".

Además, quiero aclarar que las funciones aquí presentadas son las que se piensan en esta etapa del proyecto, sin embargo, creo que en el futuro con la adquisición de nuevo conocimiento proveniente de las clases teóricas o prácticas de bases de datos estas podrían aumentar o modificarse en el caso de que se vea necesario. Creo que en el futuro está la posibilidad de que vea con otro panorama el proyecto y las funciones a incluir aumenten o se realicen de otra manera. Por el momento, están pensadas de ésta manera:

### **2.4.1 Funciones incluidas:**

- **ABM de usuarios:** Es decir, el alta, baja y modificación de usuarios en el sistema. Implicaría crear nuevos usuarios (darlos de alta) en el sistema, capturando los datos necesarios. Además, se debería poder dar de baja a estos usuarios o eliminarlos del sistema, en el caso de que se elija no seguir formando parte del sistema o que existan errores en el usuario en cuestión. Por último, se debe poder actualizar la información de un usuario en el sistema.
- **Almacenamiento y modificación de dietas:** Se debería poder almacenar la información relacionada a las diferentes dietas y a la nutrición. Además, se debe poder agregar más, eliminar o modificar en caso de que se lo necesite.
- **Calcular datos:** Debe permitir calcular correctamente todos los datos que se calculen en base a otros datos, como por ejemplo el IMC o el indicador de peso.
- **Correcta asignación de dietas:** Debe asignar o intentar asignar correctamente las dietas a los usuarios según el perfil de cada uno. Y, en el caso de que se modifique algún dato del usuario que deba llevar consigo la modificación de la dieta, que esa actualización se realice correctamente teniendo en cuenta todos los datos nuevos del usuario
- **Actualizar los datos del usuario:** Se debe lograr la correcta actualización de los datos de los usuarios correctamente en caso de que ocurra un cambio en los hábitos de este o alguna modificación en sus datos.

### **2.4.2 Funciones no incluidas:**

- **Seguimiento de consumo de calorías:** Creo que el cálculo de calorías es muy interesante para este tipo de sistemas, ya que para una persona que busque perder peso, considero que el seguimiento de las calorías consumidas es muy importante. Sin embargo, creo que el mundo de las calorías es demasiado amplio para ser abordado por una sola persona que no tiene los conocimientos necesarios para hacerlos, además de que no es el objetivo principal de la aplicación.
- **Planificación de dietas específicas:** Las dietas no se elaboraran para una persona en específico, si no para un grupo de personas que tengan unas características en común. Por ejemplo, dos personas que tengan pesos y alturas similares, que hagan ambos tenis y que tengan el objetivo de bajar de peso es muy probable que tengan asignada la misma dieta.
- **Planificación de ejercitación:** El sistema no asignará planes de entrenamiento para los usuarios, si no que simplemente se centrará en la parte de nutrición. Esto es ya que la ejercitación se alejaría de la consigna principal que fue brindada.
- **Asesoramiento médico:** El sistema no ofrecerá asesoramiento médico de ningún tipo, nuevamente ya que se centrará en la nutrición para no alejarse del objetivo principal.

## **3 Conclusión.**

Se realizó la etapa 2 del proyecto, en ella se habló más que nada de las funciones o requisitos que tendrá el sistema. Además, se visualizó quienes serán los usuarios objetivo de la aplicación. Por último se mencionaron los motivos de desarrollo del sistema.

# Etapa 3 - Proyecto Bases de Datos

Tomás Rando Muñoz - 14004

Marzo 2024

<b>1</b>	<b>Introducción.</b>	<b>1</b>
<b>2</b>	<b>1. Definir el diagrama entidad relación del sistema.</b>	<b>1</b>
<b>3</b>	<b>2. Describir las entidades, tipos, atributos, claves, tipos de relaciones, roles y restricciones estructurales.</b>	<b>3</b>
3.1	Entidades y tipos: . . . . .	3
3.2	Atributos y tipos: . . . . .	3
3.3	Claves: . . . . .	4
3.4	Tipos de relaciones y restricciones: . . . . .	5
<b>4</b>	<b>3. Definir las restricciones de integridad referencial y otras restricciones necesarias.</b>	<b>5</b>
<b>5</b>	<b>4. Realizar el mapeo del diagrama entidad relación al modelo relacional</b>	<b>5</b>

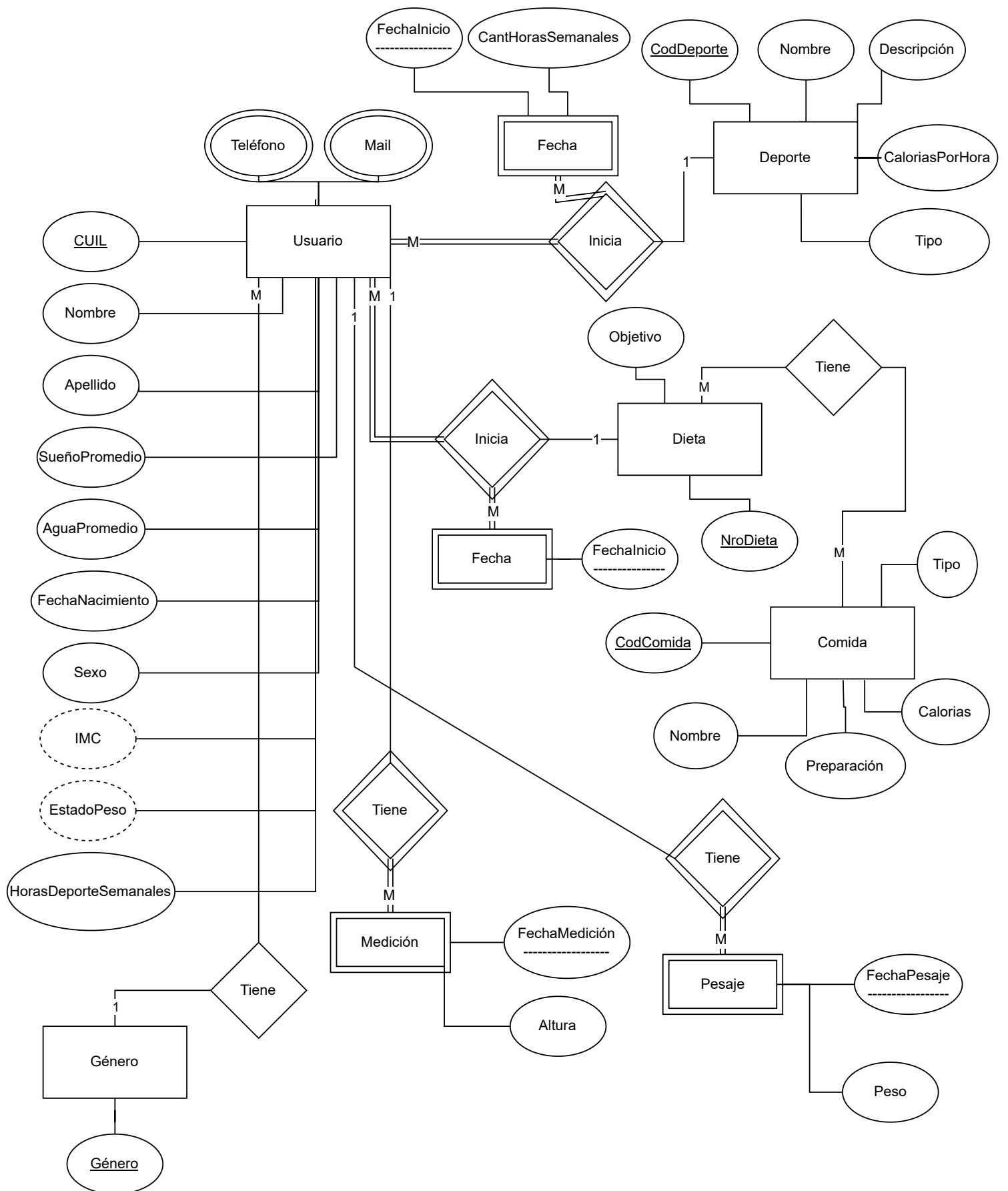
## 1 Introducción.

Se realiza la etapa 3 del proyecto de la materia "Teoría De Bases de Datos I", esta trató mayormente de la realización del diagrama de entidad-relación. Además, en la consigna número 4 se hizo el traspaso al modelo relacional sin su correspondiente normalización.

Para la realización de esta etapa nuevamente se tuvieron en cuenta los formatos establecidos por el profesor al principio del proyecto y se realizó en el editor online Overleaf.

## 2 1. Definir el diagrama entidad relación del sistema.

Se realizó el diagrama entidad relación teniendo en cuenta lo realizado en la etapa 1 y 2 del proyecto. El esquema fue realizado con la página web "draw.io".



### **3 2. Describir las entidades, tipos, atributos, claves, tipos de relaciones, roles y restricciones estructurales.**

#### **3.1 Entidades y tipos:**

- Usuarios: Son los usuarios que serán dados de alta en el sistema. Estos poseerán la mayor parte de atributos, los cuales incluyen: Apellido, Nombre, Teléfono, Mail, Cuil, SueñoPromedio, AguaPromedio, FechaNacimiento, Sexo, HorasDeporteSemanal, IMC y EstadoPeso. Además, la clave primaria estará definida por el Cuil.
- Deporte: Es la entidad correspondiente a los deportes que es posible realizar. Los atributos son CodDeporte (Clave primaria), Nombre, Descripción, CaloriasPorHora, Tipo (Aeróbico, anaeróbico, mixto).
- Dieta: Es la entidad correspondiente a las diferentes dietas que estarán disponibles. Su atributo es NroDieta y será la identificación de la dieta correspondiente, además, es también la clave primaria. Por último, posee otro atributo "Objetivo" que es el objetivo principal de la dieta, por ejemplo "Pérdida de peso".
- Pesaje: Es la entidad correspondiente al pesaje de la persona, tendrá como atributos peso y fechaPesaje. Además, será una entidad débil ya que depende de Usuario para subsistir. Por último, su discriminante será fechaPesaje.
- Medición: Es la entidad correspondiente a la medición de la altura del usuario, tendrá como atributos altura y fechaMedición. Será nuevamente una entidad débil ya que depende del usuario para subsistir. Su discriminante será fechaMedición.
- Fecha: Son dos entidades débiles que están ahí para identificar la fecha de inicio del deporte y de la dieta específica. Su atributo fechaInicio será también su discriminante.
- Género: Es la entidad que representa al género de las personas. Fue realizado como entidad ya que en este momento los géneros son muy variables. Su atributo es género que es el género de la persona propiamente dicho y su clave primaria también es ese.
- Comida: Es la entidad que representa las distintas comidas. Está asociada a las dietas y sus atributos son CodComida (Clave primaria), Nombre, Preparación, Calorías y Tipo (Colación, Plato Principal).

#### **3.2 Atributos y tipos:**

- Apellido: Es el apellido que tendrá el usuario. Es monovalorado.
- Usuario.Nombre: Es el nombre del usuario en cuestión. Es monovalorado.
- Teléfono: Es el teléfono del usuario. En este caso, será multivalor porque el usuario puede tener más de un teléfono asociado. Es multivalorado.
- Email: Es el email asociado al usuario. En este caso, el usuario puede tener más de un mail, por lo que será un atributo como teléfono, es decir, es multivaluado.
- Cuil: Es el CUIL del usuario. Es lo que identificará a cada usuario. Anteriormente se pensó en sustituirlo por el DNI, sin embargo, se vió que, aunque es un caso extraño, los DNI se pueden repetir entre personas de un mismo país. Es monovalorado.
- SueñoPromedio: Son las horas que se duerme en promedio por día. Es monovalorado.
- AguaPromedio: Es la cantidad de agua que el usuario toma por día, en vasos. Es monovalorado.
- FechaNacimiento: Es la fecha en la que nació el usuario en cuestión, se utiliza en vez de edad ya que esta no cambia, al contrario que la otra. Es monovalorado.
- Sexo: Es el sexo del usuario, definido al nacer cada uno. Es monovalorado.
- Género: Es el género del usuario, es decir, el género con el que cada usuario se identifica, no definido al nacer. Es atributo de la entidad también denominada Género. Es monovalorado.

- IMC: Es el índice de masa corporal, es un atributo que se calcula utilizando otros atributos, en este caso el peso y la altura. Es derivado.
- EstadoPeso: Es un índice que nos dice si el peso del usuario es "Malo/Regular/Bueno". Se calcula nuevamente teniendo en cuenta el peso y la altura. Es derivado.
- altura: Es la altura del usuario, medida en metros, es un atributo de la entidad altura. Es monovalorado.
- fechaMedicion: Es la fecha en la que se realizó la medición específica. Es monovalorado.
- fechaPesaje: Es la fecha en la que se tomó el peso especificado. Es monovalorado.
- peso: Es la cantidad de kilos que pesa el usuario en la fecha determinada. Es monovalorado.
- NroDieta: Es un número que nos identifica a la dieta específica. Es monovalorado.
- Objetivo: Es una cadena que nos identifica el objetivo de la dieta específica. Por ejemplo "Pérdida de peso". Es monovalorado.
- fechaInicio: Es la fecha en la que se empieza la dieta especificada. Es monovalorado.
- Deporte.Nombre: Es el nombre del deporte que se realiza. Es monovalorado.
- fechaInicio: Es la fecha en la que se comienza a realizar el deporte especificado. Es monovalorado.
- HorasDeporteSemanales: Es la cantidad de horas promedio que se realiza el deporte especificado. Es un atributo de la relación "Usuario". Es monovalorado.
- Comida.Nombre: Es cada comida que puede existir en una dieta. Es un atributo de la entidad denominada "Comida". Es monovalorado.
- CodComida: Es el código o id de cada comida. Es monovalorado.
- Preparación: Es la preparación o forma de preparar cada comida. Es monovalorado.
- Calorías: Es la cantidad de calorías que posee cada comida cada 100 gramos. Es monovalorado.
- Comida.Tipo: Nos indica si es una "Colación" o "Plato Principal". Es monovalorado.
- Deporte.Tipo: Indica si es aeróbico, anaeróbico o mixto. Es monovalorado.
- CaloriasPorHora: Nos indica cuantas calorías por hora se gastan en el deporte indicado. Es monovalorado.
- Deporte.Descripción: Es una breve descripción del deporte especificado. Es monovalorado.
- CodDeporte: Es un número que identifica cada deporte específico. Es monovalorado.
- Fecha.CantHorasSemanales: Número que indica la cantidad de horas que se realiza el deporte especificado cuando se comenzó en la fecha indicada. Es monovalorado.

### 3.3 Claves:

Se utilizan diferentes claves en el diagrama.

- Cuil: Clave primaria de la entidad usuario. Identifica unívocamente a cada usuario.
- NroDieta: Es la clave primaria de la entidad Dieta.
- CodDeporte: Es la clave primaria de la entidad Deporte. Cada deporte tendrá un número en específico.
- Cuil y fechaMedición: Es la clave de la entidad Medición. Al ser una entidad débil estará compuesta por la clave de la entidad identificadora y por el discriminante.
- Cuil y fechaPesaje: Es la clave de la entidad Pesaje. Al ser una entidad débil estará compuesta por la clave de la entidad identificadora y por el discriminante.

- CodComida: Es la clave primaria de la entidad comida. Cada comida será diferente por lo que se identifica unívocamente cada una.
- género: Es la clave primaria de la entidad género de mismo nombre. Cada género será identificado unívocamente.

Además existirán otras claves de las entidades débiles "Fecha" que serán formadas por las claves de las entidades identificadoras y sus discriminantes, que en ambos casos son "fechaInicio".

### 3.4 Tipos de relaciones y restricciones:

Hay diversas relaciones que aparecieron en el diagrama entidad relación. Por ejemplo, la relación "Inicia" entre Usuarios y Deporte, será una relación de tipo "1 a muchos", ya que un usuario iniciará un solo deporte específico en la fecha determinada, sin embargo, muchos usuarios pueden iniciar un deporte en una fecha específica. Hay también relaciones "Muchos a muchos", como la es la relación "Tiene" entre Comida y Dieta, por lo que una comida estará presente en muchas dietas y una dieta tendrá muchas comidas. Por último, hay muchas relaciones ternarias "M a M a 1", es decir, por ejemplo, la relación "Inicia" entre Dieta, Usuario y Fecha, un usuario iniciará una dieta determinada en muchas fechas, además, una dieta en una fecha específica será de varios usuarios, y por último, un usuario en una fecha determinada tendrá una única dieta.

En total quedaron 6 relaciones: 3 "1 a M", 2 "M a M a 1" y 1 "M a M".

## 4 3. Definir las restricciones de integridad referencial y otras restricciones necesarias.

Existen algunas restricciones de integridad del modelo. Por ejemplo, los atributos altura, peso, AguaPromedio, SueñoPromedio, HorasDeporteSemanales no pueden ser valores negativos. Además, todas las fechas deben ser fechas correctas, es decir, por ejemplo, que si el mes es febrero, los días no puedan ser más de 29. También, los teléfonos deberán tener 10 o 7 números; los mails deberán poseer por lo menos un punto y una arroba; y los CUIL tendrán 11 números.

Además, también se tomó en cuenta la regla de integridad de unicidad de la clave primaria y la regla de integridad de entidad de la clave primaria, las cuales básicamente establecen que cada clave primaria debe ser única y que las claves primarias no pueden tener valores nulos. Estas dos reglas ya fueron contempladas en el desarrollo del diagrama de entidad relación, por lo que se eligieron las claves primarias teniendo esto en cuenta. Si nos vamos al modelo, vemos que ninguna aparentemente puede ser nula y que son únicas, por lo que cumplen perfectamente con ambas reglas.

Con respecto a la regla de integridad referencial, para el modelo relacional se tiene en cuenta específicamente que los valores de las claves foráneas deben ser o nulos o que las claves primarias referenciadas deben existir. Esto será tomado en cuenta especialmente en el punto 4 en el que se realiza el mapeo al modelo relacional, sin embargo, está pensado para que esto se cumpla.

## 5 4. Realizar el mapeo del diagrama entidad relación al modelo relacional

Se realiza el mapeo del diagrama anteriormente presentado al modelo relacional. Se realizó tomando en cuenta los métodos vistos en teoría, además de algunas consideraciones que vimos en las clases posteriores. Es importante mencionar que el siguiente modelo no fue normalizado ya que esto será realizado en la etapa 4 del presente proyecto. Como aclaración adicional, las claves primarias de las relaciones se encuentran subrayadas con línea continua y las claves foráneas se encuentran subrayadas con línea punteada.

—  
 Usuario(Cuil, Nombre, Apellido, SueñoPromedio, AguaPromedio, FechaNacimiento, Sexo, Género, IMC, EstadoPeso, HorasDeporteSemanales)  
 Género(Género)  
 TelefonoUsuario(Cuil, Teléfono)  
 MailUsuario(Cuil, Mail)  
 Dieta(NroDieta, Objetivo)  
 Deporte(CodDeporte, Nombre, Descripción, Tipo, CaloriasPorHora)



Comida(CodComida, Nombre, Preparación, Calorías, Tipo)  
DietaTieneComida(NroDieta, NroComida)  
IniciaDeporte(Cuil, FechaInicioDeporte, CodDeporte, CantHorasSemanales)  
Medicion(Cuil, fechaMedicion, Altura)  
Pesaje(Cuil, FechaPesaje, Peso)  
RealizaDieta(Cuil, FechaInicio, NroDieta)

# Etapa 4 - Proyecto Bases de Datos

Tomás Rando Muñoz - 14004

Abril 2024

<b>1</b>	<b>Introducción:</b>	<b>2</b>
<b>2</b>	<b>Describir las dependencias funcionales del modelo elaborado previamente.</b>	<b>2</b>
2.1	Usuario: . . . . .	2
2.2	Género: . . . . .	2
2.3	TelefonoUsuario. . . . .	2
2.4	MailUsuario. . . . .	3
2.5	Dieta. . . . .	3
2.6	Deporte. . . . .	3
2.7	Comida. . . . .	3
2.8	DietaTieneComida. . . . .	3
2.9	IniciaDeporte. . . . .	3
2.10	Medición. . . . .	3
2.11	Pesaje. . . . .	3
2.12	RealizaDieta. . . . .	3
<b>3</b>	<b>Definir el modelo entidad relación en primera forma normal.</b>	<b>3</b>
3.1	Usuario: . . . . .	3
3.2	Género: . . . . .	3
3.3	TelefonoUsuario. . . . .	4
3.4	MailUsuario. . . . .	4
3.5	Dieta. . . . .	4
3.6	Deporte. . . . .	4
3.7	Comida. . . . .	4
3.8	DietaTieneComida. . . . .	4
3.9	IniciaDeporte. . . . .	4
3.10	Medición. . . . .	4
3.11	Pesaje. . . . .	4
3.12	RealizaDieta. . . . .	4
<b>4</b>	<b>Definir el modelo entidad relación en segunda forma normal.</b>	<b>5</b>
4.1	Usuario: . . . . .	5
4.2	Género: . . . . .	5
4.3	TelefonoUsuario. . . . .	5
4.4	MailUsuario. . . . .	5
4.5	Dieta. . . . .	5
4.6	Deporte. . . . .	5
4.7	Comida. . . . .	5
4.8	DietaTieneComida. . . . .	5
4.9	IniciaDeporte. . . . .	5
4.10	Medición. . . . .	6
4.11	Pesaje. . . . .	6
4.12	RealizaDieta. . . . .	6

<b>5 Definir el modelo entidad relación en tercera forma normal.</b>	<b>6</b>
5.1 Usuario: . . . . .	6
5.2 Género: . . . . .	6
5.3 TelefonoUsuario. . . . .	6
5.4 MailUsuario. . . . .	6
5.5 Dieta. . . . .	6
5.6 Deporte. . . . .	7
5.7 Comida. . . . .	7
5.8 DietaTieneComida. . . . .	7
5.9 IniciaDeporte. . . . .	7
5.10 Medición. . . . .	7
5.11 Pesaje. . . . .	7
5.12 RealizaDieta. . . . .	7
<b>6 Conclusión:</b>	<b>7</b>

## 1 Introducción:

En la etapa 4 del proyecto que estamos realizando se analizan las dependencias funcionales y las tres formas normales del modelo que se realizó en la etapa 3. Se utiliza el editor online Overleaf y las especificaciones dadas por el profesor referidas a la presentación del documento.

## 2 Describir las dependencias funcionales del modelo elaborado previamente.

Antes de comenzar con la resolución, se recuerda el modelo relacional que se elaboró en la etapa anterior del proyecto, el cual es el siguiente:

—  
 Usuario(Cuil, Nombre, Apellido, SueñoPromedio, AguaPromedio, FechaNacimiento, Sexo, Género, IMC, EstadoPeso, HorasDeporteSemanales)  
 Género(Género)  
 TelefonoUsuario(Cuil, Teléfono)  
 MailUsuario(Cuil, Mail)  
 Dieta(NroDieta, Objetivo)  
 Deporte(CodDeporte, Nombre, Descripción, Tipo, CaloriasPorHora)  
 Comida(CodComida, Nombre, Preparación, Calorías, Tipo)  
 DietaTieneComida(NroDieta, NroComida)  
 IniciaDeporte(Cuil, FechaInicioDeporte, CodDeporte, CantHorasSemanales)  
 Medicion(Cuil, fechaMedicion, Altura)  
 Pesaje(Cuil, FechaPesaje, Peso)  
 RealizaDieta(Cuil, FechaInicio, NroDieta)

Para mayor orden y claridad, se nombrarán las dependencias funcionales por cada relación. Como aclaración, cuando se dice que "X determina funcionalmente a X" se está mencionando implícitamente a la propiedad reflexiva.

### 2.1 Usuario:

Cuil determina funcionalmente a: Cuil, Nombre, Apellido, SueñoPromedio, AguaPromedio, FechaNacimiento, Sexo, Género, IMC, EstadoPeso, HorasDeporteSemanales

### 2.2 Género:

Género determina funcionalmente a Género, es decir, a sí misma.

### 2.3 TelefonoUsuario.

Cuil, Teléfono determina funcionalmente a Cuil y a Teléfono.

## **2.4 MailUsuario.**

Cuil, Mail determina funcionalmente a Cuil y a Mail

## **2.5 Dieta.**

NroDieta determina funcionalmente a NroDieta y a Objetivo.

## **2.6 Deporte.**

Nombre determina funcionalmente a Nombre

## **2.7 Comida.**

CodComida determina funcionalmente a CodComida, Nombre, Tipo, Calorías, Preparación

## **2.8 DietaTieneComida.**

NroDieta, NroComida determina funcionalmente a NroDieta y a NroComida

## **2.9 IniciaDeporte.**

Cuil, FechaInicioDeporte determina funcionalmente a CodDeporte, a Cuil, a FechaInicioDeporte y a CantHorasSemanales.

## **2.10 Medición.**

Cuil, fechaMedicion determina funcionalmente a Altura, a Cuil y a fechaMedicion.

## **2.11 Pesaje.**

Cuil, FechaPesaje determina funcionalmente a Peso, Cuil y a FechaPesaje.

## **2.12 RealizaDieta.**

Cuil, FechaInicio determina funcionalmente a NroDieta, FechaInicio y a Cuil.

## **3 Definir el modelo entidad relación en primera forma normal.**

Con respecto a esta consigna, surgió un problema, el modelo que se realizó en la anterior etapa ya está en la tercera forma normal. A causa de esto, lo que se hace en estas siguientes tres secciones es explicar y justificar por qué el modelo definido ya está en primera, segunda y tercera forma normal. Para ello, se analizará cada relación por separado

Recordamos que una relación está en primera forma normal si no tiene ningún grupo repetitivo en la relación. Es decir, para verificar la 1FN verificaremos esta condición.

### **3.1 Usuario:**

Vemos que está en primera forma normal ya que no posee ningún grupo repetitivo en ningún atributo, si nos remitimos al esquema, los atributos que formarían un grupo repetitivo serían teléfono y mail, pero fueron eliminados a la hora de hacer el traspaso al modelo relacional. Se encuentra en primera forma normal.

### **3.2 Género:**

Vemos que no tiene ningún grupo repetitivo al la clave primaria ser el único atributo de la relación. Se encuentra en primera forma normal.

### **3.3 TelefonoUsuario.**

Vemos que de es la relación utilizada para eliminar el grupo repetitivo de Usuario. No posee grupos repetitivos ya que todos los atributos forman parte de la clave primaria. Se encuentra en primera forma normal.

### **3.4 MailUsuario.**

Es la otra relación utilizada para eliminar el grupo repetitivo de Usuario. Tampoco posee grupos repetitivos ya que todos los atributos forman parte de la clave primaria. Se encuentra en primera forma normal.

### **3.5 Dieta.**

Nuevamente, no posee grupos repetitivos ya que cada dieta tendrá solamente un objetivo en específico. Se encuentra en primera forma normal.

### **3.6 Deporte.**

Como sucede en las anteriores, no posee grupo repetitivo ya que cada deporte tendrá atributos únicos. Se encuentra en primera forma normal.

### **3.7 Comida.**

Es el mismo caso que en las anteriores, no posee grupo repetitivo ya que cada comida tendrá atributos únicos. Se encuentra en primera forma normal.

### **3.8 DietaTieneComida.**

No posee grupo repetitivo ya que tienen dos atributos que forman parte en conjunto de la clave primaria. Se encuentra en primera forma normal.

### **3.9 IniciaDeporte.**

No posee grupo repetitivo ya que a cada clave primaria le corresponderá un solo NombreDeporte. Se encuentra en primera forma normal.

### **3.10 Medición.**

No posee grupo repetitivo ya que a cada Cuil y fechaMedicion le corresponderá una Altura específica. Se encuentra en primera forma normal.

### **3.11 Pesaje.**

No posee grupo repetitivo ya que a cada Cuil y FechaPesaje le corresponderá solo un Peso específico. Se encuentra en primera forma normal.

### **3.12 RealizaDieta.**

Nuevamente, no posee grupo repetitivo ya que a cada Cuil y FechaInicio le corresponderá un NroDieta específico. Se encuentra en primera forma normal.

—

Como podemos observar, el modelo se encuentra en primera forma normal.

## **4 Definir el modelo entidad relación en segunda forma normal.**

Ahora, verificaremos la segunda condición, es decir, la condición que será necesario cumplir para que el modelo se encuentre en segunda forma normal. Primero, el modelo debe estar en primera forma normal, luego, no deben haber dependencias parciales en ninguna relación, es decir, no existirán atributos que dependan funcionalmente de parte de la clave primaria. Para el análisis de este punto, nuevamente se distinguirá entre cada relación.

Se tiene en cuenta que en la anterior consigna se verificó que las relaciones estuvieran en primera forma normal.

### **4.1 Usuario:**

Vemos que no posee dependencias funcionales parciales ya que la relación posee una clave primaria. Entonces, se encuentra en segunda forma normal.

### **4.2 Género:**

En este caso, vemos que no se posee dependencias parciales ya que la relación solo tiene una clave primaria y no tiene atributos no clave. Entonces, se encuentra en segunda forma normal.

### **4.3 TelefonoUsuario.**

En este caso, vemos que si bien hay dos atributos que forman parte de la clave primaria, no existen dependencias funcionales parciales ya que no existen atributos no clave. Entonces, se encuentra en segunda forma normal.

### **4.4 MailUsuario.**

Nuevamente, vemos que hay dos atributos que forman parte de la clave primaria, sin embargo no hay dependencias parciales al no existir atributos no clave. Entonces, se encuentra en segunda forma normal.

### **4.5 Dieta.**

Como fue el caso de Usuario, Dieta no posee dependencias funcionales parciales ya que solo posee un atributo clave primaria. Entonces, se encuentra en segunda forma normal.

### **4.6 Deporte.**

Nuevamente, no se poseen dependencias funcionales parciales al existir solo un atributo en la clave primaria. Entonces, se encuentra en segunda forma normal.

### **4.7 Comida.**

Comida no posee dependencias funcionales parciales ya que solo existe un atributo que forme parte de la clave primaria. Entonces, se encuentra en segunda forma normal.

### **4.8 DietaTieneComida.**

En este caso, no existen dependencias funcionales parciales ya que no hay atributos no clave. Entonces, se encuentra en segunda forma normal.

### **4.9 IniciaDeporte.**

En este caso, no hay dependencias funcionales parciales ya que NombreDeporte depende funcionalmente tanto de Cuil como de FechaInicioDeporte, es decir, la dependencia funcional es total. Entonces, se encuentra en segunda forma normal.

#### **4.10 Medición.**

En este caso, nuevamente, no hay dependencias funcionales parciales ya que Altura depende tanto del Cuil como de la fechaMedicion, por lo que la dependencia funcional es total. Entonces, se encuentra en segunda forma normal.

#### **4.11 Pesaje.**

Nuevamente, no hay dependencias funcionales parciales ya que Peso es determinado funcionalmente mediante Cuil y FechaPesaje, es decir, con la totalidad de la clave primaria. Entonces, se encuentra en segunda forma normal.

#### **4.12 RealizaDieta.**

Finalmente, tampoco existen dependencias funcionales parciales ya que NroDieta depende de la totalidad de la clave primaria. Entonces, se encuentra en segunda forma normal.

—

Con ello, podemos afirmar que el modelo se encuentra en segunda forma normal.

### **5 Definir el modelo entidad relación en tercera forma normal.**

Habiendo definido el modelo en primera forma normal y segunda forma normal, nos queda verificar que el modelo se encuentre en la tercera forma normal. Para ello, el modelo debe cumplir tres condiciones específicas: El modelo debe encontrarse en primera forma normal, en segunda forma normal y no deben haber dependencias transitivas. Es decir, no deben existir atributos que dependan de la clave primaria gracias a que dependen funcionalmente de otro atributo que también dependa de la clave primaria.  $X \rightarrow Y \rightarrow Z$ ;  $X \rightarrow Z$  Nuevamente, se analizará cada relación por separado.

Se tiene en cuenta que ya se verificó en las anteriores consignas que el modelo se encontrase en primera y segunda forma normal, por lo que se omitirá esa parte de la verificación.

#### **5.1 Usuario:**

No existen dependencias transitivas ya que todos los atributos dependen funcionalmente de Cuil. En consecuencia, se encuentra en tercera forma normal.

#### **5.2 Género:**

No hay dependencias transitivas ya que no existen atributos no clave. En consecuencia, se encuentra en tercera forma normal.

#### **5.3 TelefonoUsuario.**

En esta relación tampoco existen atributos no clave, por lo que no existen dependencias transitivas. En consecuencia, se encuentra en tercera forma normal.

#### **5.4 MailUsuario.**

Por la misma razón no existen dependencias transitivas. En consecuencia, se encuentra en tercera forma normal.

#### **5.5 Dieta.**

En este caso, no existen dependencias transitivas ya que Objetivo depende funcionalmente de NroDieta. Además, se evidencia gracias al hecho de que solo existe un atributo que forma la clave primaria. En consecuencia, se encuentra en tercera forma normal.

## **5.6 Deporte.**

Los atributos dependen directamente de CodDeporte, por lo que no hay dependencias transitivas. En consecuencia, se encuentra en tercera forma normal.

## **5.7 Comida.**

Nuevamente, todos los atributos dependen de NroComida (o CodComida en el diagrama), por lo que no hay dependencias transitivas. En consecuencia, se encuentra en tercera forma normal.

## **5.8 DietaTieneComida.**

No existen atributos no clave, por lo que no existen dependencias transitivas. En consecuencia, se encuentra en tercera forma normal.

## **5.9 IniciaDeporte.**

No existen dependencias transitivas porque todos los atributos dependen única y directamente de la clave primaria. En consecuencia, se encuentra en tercera forma normal.

## **5.10 Medición.**

En este caso, tampoco existen más de un atributo no clave, por lo que tampoco hay dependencias transitivas. En consecuencia, se encuentra en tercera forma normal.

## **5.11 Pesaje.**

Nuevamente, solo hay un atributo no clave por lo que no hay dependencias transitivas. En consecuencia, se encuentra en tercera forma normal.

## **5.12 RealizaDieta.**

Por último, por la misma razón no existen dependencias transitivas, solo hay un atributo no clave en la relación, por lo que sería imposible que existiera una dependencia transitiva. En consecuencia, se encuentra en tercera forma normal.

—

Con ello, verificamos que el modelo se encuentra tanto en primera, segunda y tercera forma normal.

## **6 Conclusión:**

En esta etapa del proyecto se colocaron las dependencias funcionales y se verificó que el modelo desarrollado en la anterior etapa efectivamente estuviera en primera, segunda y tercera forma normal.



---

# LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

Teoría de Base de Datos

## **Proyecto Integrador**

Alumno: Tomás Rando

---

---

## Young and Healthy Life

---

## Contents

<b>1</b>	<b>Implementación de la Base de Datos</b>	<b>3</b>
1.1	Scripts de creación de la base de datos . . . . .	3
1.2	Scripts de creación de las tablas de la base de datos . . . . .	3
1.3	Scripts de creación de restricciones de la base de datos . . . . .	11
1.4	Scripts para el ingreso de datos en la base de datos . . . . .	11

# 1 Implementación de la Base de Datos

## 1.1 Scripts de creación de la base de datos

*--Crea el rol de administrador con todos los permisos para gestionar la base de datos*

```
CREATE DATABASE uncuyo;
```

```
CREATE ROLE administrador;
```

```
ALTER ROLE administrador
```

```
LOGIN
```

```
SUPERUSER
```

```
CREATEROLE
```

```
CREATEDB
```

```
REPLICATION
```

```
BYPASSRLS
```

```
PASSWORD '1234';
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO administrador;
```

*--Crea el rol de empleado con permisos para modificar datos de la base de datos*

```
CREATE ROLE empleado;
```

```
ALTER ROLE empleado
```

```
LOGIN
```

```
PASSWORD '1234';
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO empleado;
```

*--Crea el rol de cliente con permisos para consultar datos de algunas tablas de la base*

```
CREATE ROLE cliente;
```

```
ALTER ROLE cliente
```

```
LOGIN
```

```
PASSWORD '1234';
```

```
GRANT SELECT ON realizadieta TO cliente;
```

```
GRANT SELECT ON dietatienecomida TO cliente;
```

```
GRANT SELECT ON dieta TO cliente;
```

## 1.2 Scripts de creación de las tablas de la base de datos

Se inserta el script elaborado para crear las tablas dentro de la base de datos. Además, se añade un script para crear dos triggers que servirán para el calculo de datos.

*--Crea tabla genero*

```
CREATE TABLE genero (  
genero VARCHAR(50) PRIMARY KEY  
);
```

*--Crea tabla usuario*

```
CREATE TABLE usuario (  
cuil BIGINT PRIMARY KEY,  
nombre VARCHAR(50),  
apellido VARCHAR(50),  
suenoPromedio REAL,  
aguaPromedio REAL,  
fechaNacimiento DATE,  
sexo CHAR(1),  
genero VARCHAR(50),  
imc NUMERIC,  
estadoPeso VARCHAR(30),  
horasDeporteSemanales REAL,  
FOREIGN KEY (genero) REFERENCES genero(genero)  
);
```

*--Crea tabla telefonousuario*

```
CREATE TABLE telefonoUsuario (  
telefono BIGINT,  
cuil BIGINT,  
PRIMARY KEY (cuil, telefono),  
FOREIGN KEY (cuil) REFERENCES usuario(cuil)  
);
```

*--Crea tabla mailUsuario*

```
CREATE TABLE mailUsuario (  
mail VARCHAR(50),  
cuil BIGINT,  
PRIMARY KEY (cuil, mail),  
FOREIGN KEY (cuil) REFERENCES usuario(cuil)  
);
```

*--Crea tabla dieta*

```
CREATE TABLE dieta (  
nrodieta SERIAL PRIMARY KEY,  
objetivo VARCHAR(50)  
);
```

*--Crea tabla comida*

```
CREATE TABLE comida (  
id SERIAL PRIMARY KEY,  
nombre VARCHAR(50),
```

```
calorias REAL,  
preparacion VARCHAR(300),  
tipo VARCHAR(40)  
);
```

```
--Crea tabla dietaTieneComida
```

```
CREATE TABLE dietaTieneComida (  
nroComida INT,  
nroDieta SERIAL,  
FOREIGN KEY (nroComida) REFERENCES comida(id),  
FOREIGN KEY (nroDieta) REFERENCES dieta(nrodieta),  
PRIMARY KEY (nroDieta, nroComida)  
);
```

```
--Crea tabla deporte
```

```
CREATE TABLE deporte (  
id SERIAL PRIMARY KEY,  
nombre VARCHAR(50),  
caloriasPorHora REAL,  
tipo VARCHAR(40),  
descripcion VARCHAR(250)  
);
```

```
--Crea tabla iniciaDeporte
```

```
CREATE TABLE iniciaDeporte (  
cuil BIGINT,  
fechaInicioDeporte DATE,  
nroDeporte INT,  
cantHorasSemanales REAL,  
PRIMARY KEY (cuil, fechaInicioDeporte),  
FOREIGN KEY (cuil) REFERENCES usuario(cuil),  
FOREIGN KEY (nroDeporte) REFERENCES deporte(id)  
);
```

```
--Crea tabla realizaDieta
```

```
CREATE TABLE realizaDieta (  
cuil BIGINT,  
fechaInicio DATE,  
nroDieta SERIAL,  
PRIMARY KEY (cuil, fechaInicio),  
FOREIGN KEY (cuil) REFERENCES usuario(cuil),  
FOREIGN KEY (nroDieta) REFERENCES dieta(nrodieta)
```

);

*--Crea tabla medicion*

```
CREATE TABLE medicion (  
    cuil BIGINT,  
    fechaMedicion DATE,  
    altura REAL,  
    PRIMARY KEY (cuil, fechaMedicion),  
    FOREIGN KEY (cuil) REFERENCES usuario(cuil)  
);
```

*--Crea tabla pesaje*

```
CREATE TABLE pesaje (  
    cuil BIGINT,  
    fechaPesaje DATE,  
    peso REAL,  
    PRIMARY KEY (cuil, fechaPesaje),  
    FOREIGN KEY (cuil) REFERENCES usuario(cuil)  
);
```

*--Crea función utilizada en el trigger de inserción a la tabla de medición*

```
CREATE OR REPLACE FUNCTION funcionMedicion()  
RETURNS TRIGGER AS $$  
DECLARE  
    p FLOAT;  
    imcc FLOAT;  
    a FLOAT;  
    f DATE;  
    fechaAct DATE;  
    nro INT;  
    nro2 INT;  
    objetivoc VARCHAR(50);  
    estadoPesoc VARCHAR(30);  
BEGIN  
    SELECT peso INTO p  
    FROM pesaje  
    WHERE cuil = NEW.cuil  
    ORDER BY fechapesaje DESC  
    LIMIT 1;  
  
    SELECT altura, fechaMedicion  
    INTO a, f
```

```
FROM medicion
WHERE cuil = NEW.cuil
ORDER BY fechaMedicion DESC
LIMIT 1;

IF f IS NOT NULL THEN
    IF f < NEW.fechamedicion THEN
        a := NEW.altura;
        fechaAct := NEW.fechaMedicion;
    ELSE
        fechaAct := f;
    END IF;
END IF;

IF p IS NOT NULL AND a > 0 THEN
    imcc := p / (a * a);
ELSE
    imcc := NULL;
END IF;

IF imcc < 18.50 THEN
    estadoPesoc := 'Delgadez';
    objetivoc := 'Control de Peso';
ELSIF imcc > 24.99 THEN
    estadoPesoc := 'Sobrepeso';
    objetivoc := 'Perdida De Peso';
ELSEIF imcc >= 18.50 AND imcc <= 24.99 THEN
    estadoPesoc := 'Peso Normal';
    objetivoc := 'Ganancia Muscular';
ELSE
    estadoPesoc := NULL;
    objetivoc := NULL;
END IF;

UPDATE usuario
SET imc = imcc,
    estadopeso = estadoPesoc
WHERE cuil = NEW.cuil;

SELECT nrodieta
INTO nro
FROM dieta
```

```
WHERE objetivo = objetivoc
LIMIT 1;
```

```
SELECT nrodieta
INTO nro2
FROM realizadieta
WHERE cuil = NEW.cuil
ORDER BY fechainicio DESC
LIMIT 1;
```

```
IF nro2 IS NULL THEN
    nro2 := 0;
END IF;
```

```
IF nro2 <> nro THEN
    IF objetivoc IS NOT NULL AND nro IS NOT NULL THEN
        INSERT INTO realizadieta (cuil, fechainicio, nrodieta) VALUES (NEW.cuil, fechaAct, nro);
    END IF;
END IF;
```

```
RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
--Crea trigger para la tabla medicion
CREATE TRIGGER calcularimcusuariomedicion
AFTER INSERT OR UPDATE ON medicion
FOR EACH ROW
EXECUTE FUNCTION funcionmedicion();
```

```
--Crea función utilizada en el trigger de inserción a la tabla de pesaje
CREATE OR REPLACE FUNCTION funcionpesaje()
RETURNS TRIGGER AS $$
DECLARE
p FLOAT;
imcc FLOAT;
a FLOAT;
f DATE;
fechaAct DATE;
nro INT;
nro2 INT;
estadoPesoc VARCHAR(30);
```



```
objetivoc VARCHAR(50);
BEGIN
SELECT altura INTO p
FROM medicion
WHERE cuil = NEW.cuil
ORDER BY fechamedicion DESC
LIMIT 1;

SELECT peso, fechaPesaje
INTO a, f
FROM pesaje
WHERE cuil = NEW.cuil
ORDER BY fechaPesaje DESC
LIMIT 1;

IF f IS NOT NULL THEN
    IF f < NEW.fechapesaje THEN
        a := NEW.peso;
        fechaAct := NEW.fechapesaje;
    ELSE
        fechaAct := f;
    END IF;
END IF;

IF p IS NOT NULL AND a > 0 THEN
    imcc := a / (p * p);
ELSE
    imcc := NULL;
END IF;

IF imcc < 18.50 THEN
    estadoPesoc := 'Delgadez';
    objetivoc := 'Control de Peso';
ELSIF imcc > 24.99 THEN
    estadoPesoc := 'Sobrepeso';
    objetivoc := 'Perdida De Peso';
ELSEIF imcc >= 18.50 AND imcc <= 24.99 THEN
    estadoPesoc := 'Peso Normal';
    objetivoc := 'Ganancia Muscular';
ELSE
    estadoPesoc := NULL;
```

```
objetivoc := NULL;
END IF;

UPDATE usuario
SET imc = imcc,
    estadopeso = estadoPesoc
WHERE cuil = NEW.cuil;

SELECT nrodieta
INTO nro
FROM dieta
WHERE objetivo = objetivoc
LIMIT 1;

SELECT nrodieta
INTO nro2
FROM realizadieta
WHERE cuil = NEW.cuil
ORDER BY fechainicio DESC
LIMIT 1;

IF nro2 IS NULL THEN
    nro2 := 0;
END IF;

IF nro2 <> nro THEN
    IF objetivoc IS NOT NULL AND nro IS NOT NULL THEN
        INSERT INTO realizadieta (cuil, fechainicio, nrodieta) VALUES (NEW.cuil, fechaAct, nro);
    END IF;
END IF;

RETURN NEW;
END;
$$ LANGUAGE plpgsql;

--Crea trigger para la tabla pesaje
CREATE TRIGGER calcularimcusuariopesaje
AFTER INSERT OR UPDATE ON pesaje
FOR EACH ROW
EXECUTE FUNCTION funcionpesaje();
```

### 1.3 Scripts de creación de restricciones de la base de datos

No hay scripts de restricciones de la base de datos ya que las restricciones referenciales se definieron en la creación de cada tabla en particular. Es decir, por ejemplo, en pesaje se colocaron las sentencias:

```
PRIMARY KEY (cuil, fechaPesaje),  
FOREIGN KEY (cuil) REFERENCES usuario(cuil)
```

Lo que permite definir las restricciones definiendo una clave foránea y una clave primaria.

### 1.4 Scripts para el ingreso de datos en la base de datos

Se inserta el script para el ingreso de datos dentro de la base.

```
--Se insertan datos en la tabla deporte  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Fútbol', 400,  
'Mixto', 'Deporte de equipo que se juega con una pelota entre dos equipos de once jugadores  
cada uno');  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Baloncesto', 300,  
'Mixto', 'Deporte de equipo que se juega con una pelota entre dos equipos de cinco jugadores  
cada uno');  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Tenis', 250,  
'Mixto', 'Deporte que se juega con una raqueta y una pelota entre dos jugadores');  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Voleibol', 200,  
'Mixto', 'Deporte de equipo que se juega con una pelota entre dos equipos de seis jugadores  
cada uno');  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Atletismo', 500,  
'Aerobico', 'Deporte que incluye diversas disciplinas como carreras, saltos y lanzamientos');  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Natación', 350,  
'Mixto', 'Deporte acuático que se practica en piscinas o aguas abiertas');  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Gimnasia', 200,  
'Mixto', 'Actividad física que combina movimientos corporales, fuerza y flexibilidad');  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Ciclismo', 300,  
'Aerobico', 'Deporte que se practica con una bicicleta en diferentes terrenos');  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Boxeo', 600,  
'Mixto', 'Deporte de combate que se practica con golpes de puño');  
INSERT INTO deporte (nombre, caloríasPorHora, tipo, descripcion) VALUES ('Hockey', 400,  
'Mixto', 'Deporte de equipo que se juega con un palo y una pelota entre dos equipos de once  
jugadores cada uno');  
  
--Se insertan datos en la tabla genero  
INSERT INTO genero (genero) VALUES ('Hombre');  
INSERT INTO genero (genero) VALUES ('Mujer');  
INSERT INTO genero (genero) VALUES ('No Binario');
```

--Se insertan datos en la tabla usuario

```
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (20451379632, 'Tomas', 'Rando',
6, 6, '2003-10-05', 'M', 'Hombre', NULL, NULL, 4);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (20403895033, 'Juan',
'González', 7, 4, '1985-05-15', 'M', 'Hombre', NULL, NULL, 2);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (27439583490, 'María', 'López',
6, 5, '1992-07-20', 'F', 'Mujer', NULL, NULL, 1);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (20175983008, 'Carlos',
'Martínez', 8, 6, '1988-03-10', 'M', 'Hombre', NULL, NULL, 5);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (27451374546, 'Laura',
'García', 7, 7, '1995-11-25', 'F', 'Mujer', NULL, NULL, 3);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (20205983849, 'Diego',
'Rodríguez', 6, 4, '1982-09-05', 'M', 'Hombre', NULL, NULL, 2);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (27334596848, 'Ana',
'Fernández', 8, 5, '1998-12-12', 'F', 'Mujer', NULL, NULL, 4);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (20451356438, 'Javier',
'Pérez', 7, 6, '1987-06-30', 'M', 'Hombre', NULL, NULL, 3);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (23464534954, 'Lucía', 'Gómez',
6, 7, '1993-04-18', 'F', 'Mujer', NULL, NULL, 2);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (20395434579, 'Pablo',
'Hernández', 8, 4, '1989-08-08', 'M', 'Hombre', NULL, NULL, 5);
INSERT INTO usuario (cuil, nombre, apellido, suenoPromedio, aguaPromedio, fechaNacimiento,
sexo, genero, imc, estadoPeso, horasDeporteSemanales) VALUES (27431294856, 'Marina', 'Torres',
7, 5, '1996-02-28', 'F', 'Mujer', NULL, NULL, 3);
```

--Se insertan datos en la tabla telefonoUsuario

```
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2613400481, 20451379632);
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2618910423, 20403895033);
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2614492875, 27439583490);
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2615829378, 20175983008);
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2613627459, 27451374546);
```

```
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2617038294, 20205983849);
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2612950386, 27334596848);
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2615038496, 20451356438);
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2617382940, 23464534954);
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2614059283, 20395434579);
INSERT INTO telefonoUsuario (telefono, cuil) VALUES (2619283650, 27431294856);
```

*--Se insertan datos en la tabla dieta*

```
INSERT INTO dieta (objetivo) VALUES ('Perdida De Peso');
INSERT INTO dieta (objetivo) VALUES ('Ganancia Muscular');
INSERT INTO dieta (objetivo) VALUES ('Control de Peso');
```

*--Se insertan datos en la tabla comida*

```
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Pavo', 170,
'Cocinar en horno con aceite de oliva', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Pechuga de pollo', 165,
'Cocinar a la plancha con especias', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Sopa de lentejas', 120,
'Cocinar con verduras y especias', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Verduras mixtas', 50,
'Saltear en aceite de oliva', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Sopa de verduras', 80,
'Cocinar con variedad de verduras', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Palta', 160,
'Consumir en rodajas o como guacamole', 'Colación');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Pollo a la parrilla', 180,
'Cocinar en parrilla con condimentos', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Salmón al horno', 220,
'Cocinar en horno con limón y especias', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Vegetales asados', 70,
'Asar en horno con aceite de oliva', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Tofu', 120,
'Cocinar a la plancha con salsa de soja', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Arroz integral', 130,
'Cocinar con agua y sal', 'Colación');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Carne magra', 200,
'Cocinar a la parrilla con especias', 'Plato Principal');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Huevos duros', 140,
'Cocinar en agua hirviendo', 'Colación');
INSERT INTO comida (nombre, calorías, preparacion, tipo) VALUES ('Ensalada de garbanzos', 180,
'Preparar con garbanzos, verduras y aderezo ligero', 'Plato Principal');
```

--Se insertan datos en la tabla *dietatieneComida*

```
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (1, 1);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (2, 1);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (3, 1);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (4, 1);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (5, 2);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (6, 2);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (7, 2);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (8, 2);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (9, 2);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (8, 3);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (10, 3);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (7, 3);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (11, 3);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (12, 3);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (13, 3);
INSERT INTO dietatieneComida (nroComida, nroDieta) VALUES (14, 3);
```

--Se insertan datos en la tabla *iniciaDeporte*

```
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(20451379632, '2024-01-05', 1, 4);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(20403895033, '2024-02-26', 1, 5);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(27439583490, '2024-01-02', 2, 3);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(20175983008, '2024-03-05', 3, 2);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(27451374546, '2024-05-05', 4, 6);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(20205983849, '2023-12-12', 5, 1);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(27334596848, '2023-10-05', 6, 7);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(20451356438, '2022-11-23', 7, 4);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(23464534954, '2024-01-24', 8, 5);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(20395434579, '2024-03-29', 9, 3);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
(27431294856, '2022-08-18', 10, 2);
INSERT INTO iniciaDeporte (cuil, fechaInicioDeporte, nroDeporte, cantHorasSemanales) VALUES
```

```
(20451356438, '2020-12-20', 7, 1);  
INSERT INTO iniciadeporte (cuil, fechainiciodeporte, nrodeporte, cantHorasSemanales) VALUES  
(20451379632, '2020-12-20', 7, 6);  
INSERT INTO iniciadeporte (cuil, fechainiciodeporte, nrodeporte, cantHorasSemanales) VALUES  
(20451379632, '2023-04-27', 2, 4);
```

*--Se insertan datos en la tabla mailusuario*

```
INSERT INTO mailusuario (mail, cuil) VALUES ('tomasrando@gmail.com', 20451379632);  
INSERT INTO mailusuario (mail, cuil) VALUES ('juangonzalez@gmail.com', 20403895033);  
INSERT INTO mailusuario (mail, cuil) VALUES ('marialopez@gmail.com', 27439583490);  
INSERT INTO mailusuario (mail, cuil) VALUES ('carlosmartinez@gmail.com', 20175983008);  
INSERT INTO mailusuario (mail, cuil) VALUES ('lauragarcia@gmail.com', 27451374546);  
INSERT INTO mailusuario (mail, cuil) VALUES ('diegorodriguez@gmail.com', 20205983849);  
INSERT INTO mailusuario (mail, cuil) VALUES ('anafernandez@gmail.com', 27334596848);  
INSERT INTO mailusuario (mail, cuil) VALUES ('javierperez@gmail.com', 20451356438);  
INSERT INTO mailusuario (mail, cuil) VALUES ('luciagomez@gmail.com', 23464534954);  
INSERT INTO mailusuario (mail, cuil) VALUES ('pablohernandez@gmail.com', 20395434579);  
INSERT INTO mailusuario (mail, cuil) VALUES ('marinatorres@gmail.com', 27431294856);
```

*--Se insertan datos en la tabla realizadieta*

*--Sustituido por un trigger*

*--Se insertan datos en la tabla medicion*

```
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (20451379632, '2024-05-05', 1.75);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (20403895033, '2024-03-20', 1.82);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (27439583490, '2023-11-10', 1.65);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (20175983008, '2023-12-05', 1.70);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (27451374546, '2024-01-22', 1.75);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (20205983849, '2023-09-28', 1.68);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (27334596848, '2024-03-10', 1.73);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (20451356438, '2023-10-15', 1.80);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (23464534954, '2024-02-08', 1.76);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (20395434579, '2023-08-20', 1.72);  
INSERT INTO medicion (cuil, fechamedicion, altura) VALUES (27431294856, '2023-12-30', 1.69);
```

*--Se insertan datos en la tabla pesaje*

```
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (20451379632, '2024-05-05', 80);  
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (20403895033, '2024-03-20', 85.0);  
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (27439583490, '2023-11-10', 67.5);  
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (20175983008, '2023-12-05', 70.0);  
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (27451374546, '2024-01-22', 75.0);  
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (20205983849, '2023-09-28', 68.0);
```



```
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (27334596848, '2024-03-10', 72.5);
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (20451356438, '2023-10-15', 79.0);
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (23464534954, '2024-02-08', 76.5);
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (20395434579, '2023-08-20', 74.0);
INSERT INTO pesaje (cuil, fechapesaje, peso) VALUES (27431294856, '2023-12-30', 53.0);
```



---

# LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

Teoría de Base de Datos

**Proyecto Integrador**

Alumno: Tomás Rando

---

---

## ETAPA 6 - HEALTHY LIFE

---

## Contents

<b>1</b>	<b>Implementación parcial del sistema</b>	<b>3</b>
1.1	Descripción del Sistema . . . . .	3
1.2	Explicación de la interfaz gráfica de usuario . . . . .	3
<b>2</b>	<b>Trabajos Futuros</b>	<b>10</b>
<b>3</b>	<b>Realizar backup de la base, exportar datos de la base en formato PDF y aplicar estrategias de seguridad</b>	<b>10</b>
<b>4</b>	<b>Guía para ejecutar la aplicación.</b>	<b>11</b>
4.1	Por ejecutable en consola. . . . .	11
4.2	Utilizando NetBeans. . . . .	12

# 1 Implementación parcial del sistema

## 1.1 Descripción del Sistema

El sistema gestiona las dietas de los diferentes usuarios cargados en el sistema. Luego de que cada usuario es cargado, es posible especificar deporte realizado, y mediciones de peso y altura, entre otros datos. A partir de lo cual, mediante un trigger programado con anterioridad, se calcula el índice de masa corporal y el estado del peso del usuario. Con esta nueva información, utilizando el mismo trigger, se realiza la asignación automática de una dieta a cada usuario. Además, si en el futuro las mediciones llegaran a cambiar, se iniciaría automáticamente una nueva dieta, en caso de ser necesario actualizar. Además, es posible agregar nuevas dietas al sistema, por lo que es posible ir agrandando el universo de dietas que es posible asignar a cada usuario. **En la última sección del documento (Sección 4) se incluye una guía para ejecutar la aplicación.**

## 1.2 Explicación de la interfaz gráfica de usuario

La interfaz gráfica implementa una parte del sistema. Primero, nos encontramos una pantalla de login que solicitará un rol y una contraseña. Esta pantalla dejará ingresar al sistema solo si la conexión se realiza correctamente, es decir, si el rol y la contraseña ingresados son correctos. Hasta este momento, se realizó la ventana de administración, por lo cual solo están habilitados para ingresar los usuarios "administrador" y "postgres", cuyas contraseñas son ambas "1234".

Posteriormente, se implementó la interfaz desde la cual es posible visualizar y manejar a los usuarios y las dietas principalmente. En total, se incorporaron 5 tablas, "usuario", "dieta", "pesaje", "medición" y "realizadieta". De ellas, se realizó el abm de las dos primeras, mientras que de las siguientes dos solo se desarrolló el alta y de la última solo se incluyó su visualización.

Por último, se incluyó la posibilidad de realizar un backup en formato .dump de la base de datos y restaurar un backup. Además, se desarrolló una funcionalidad para exportar los datos contenidos en las tablas, cuya diferencia con respecto a la funcionalidad se detallará más abajo durante el desarrollo de esta sección.

Se procede a explicar una por una las ventanas actuales del sistema:

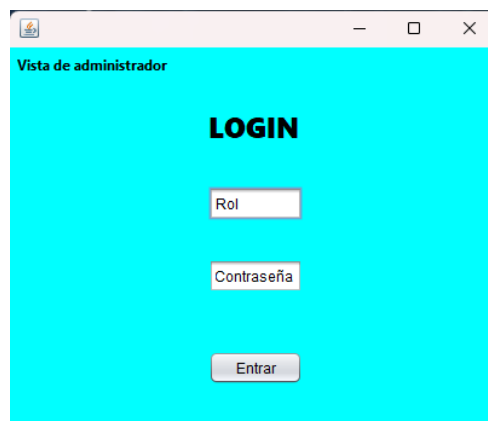
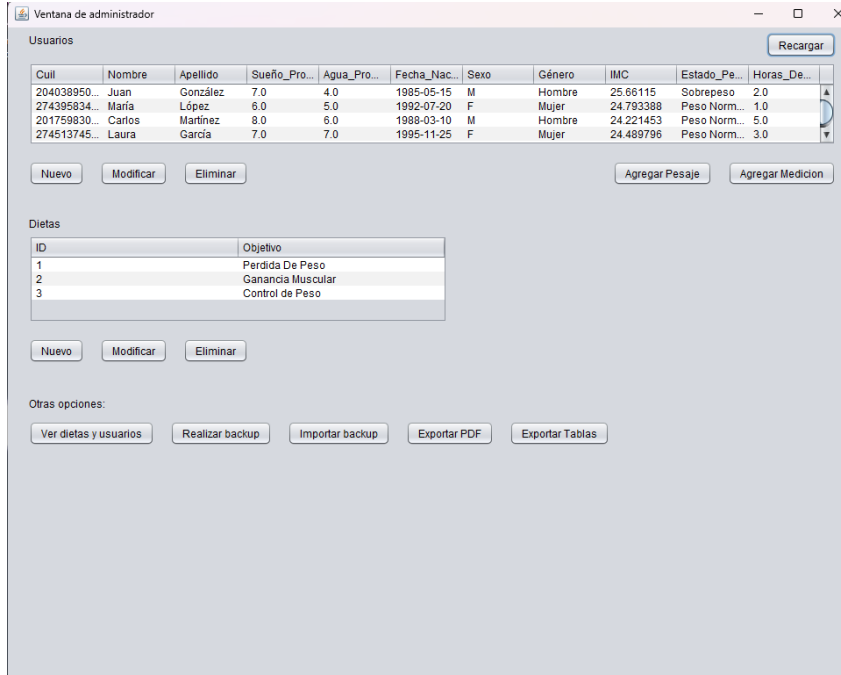


Figure 1: Ventana de Login. Se debe ingresar nombre del rol administrador y contraseña para ingresar



**Ventana de administrador**

Usuarios Recargar

Cuil	Nombre	Apellido	Sueño_Pro...	Agua_Pro...	Fecha_Nac...	Sexo	Género	IMC	Estado_Pe...	Horas_De...
204038950...	Juan	González	7.0	4.0	1985-05-15	M	Hombre	25.66115	Sobrepeso	2.0
274395834...	María	López	6.0	5.0	1992-07-20	F	Mujer	24.793388	Peso Norm...	1.0
201759830...	Carlos	Martínez	8.0	6.0	1988-03-10	M	Hombre	24.221453	Peso Norm...	5.0
274513745...	Laura	García	7.0	7.0	1995-11-25	F	Mujer	24.489796	Peso Norm...	3.0

Nuevo Modificar Eliminar Agregar Pesaje Agregar Medicion

**Diets**

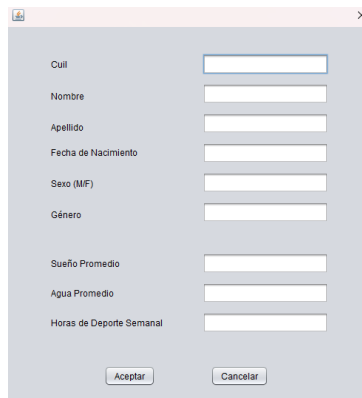
ID	Objetivo
1	Perdida De Peso
2	Ganancia Muscular
3	Control de Peso

Nuevo Modificar Eliminar

Otras opciones:

Ver dietas y usuarios Realizar backup Importar backup Exportar PDF Exportar Tablas

Figure 2: Ventana de administrador. Es la ventana principal del sistema. Se muestran las tablas "usuario" y "dieta". Además, se incluyen diferentes funcionalidades: realizar abm de usuario y dieta, realizar backup, importar backup, exportar datos como pdf, exportar tablas como pdf y realizar el alta de pesajes y mediciones. Además de refrescar la ventana.



**Nuevo**

Cuil   
 Nombre   
 Apellido   
 Fecha de Nacimiento   
 Sexo (MF)   
 Género   
 Sueño Promedio   
 Agua Promedio   
 Horas de Deporte Semanal

Aceptar Cancelar

Figure 3: "Nuevo" de usuario. Es la ventana que permite dar de alta a un usuario en el sistema. Nos solicita los diversos campos con los datos de este y se muestran dos botones, uno para aceptar y otro para cancelar. Si se acepta con datos incorrectos, saltará un aviso para informarlo.

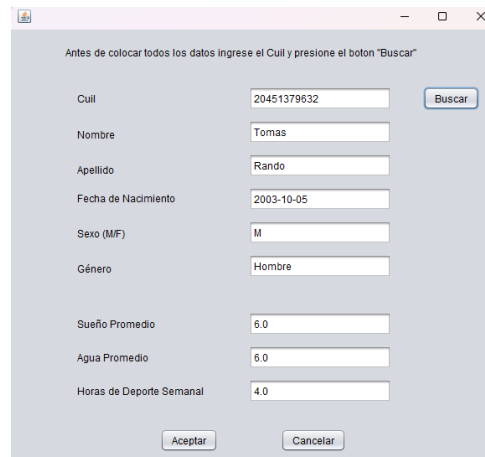


Figure 4: "Modificar" de usuario. Es la ventana que permite modificar a un usuario en el sistema. Similar al botón "nuevo", pero se incluye un botón "buscar" para autocompletar los campos luego de colocar el cui del usuario a modificar.

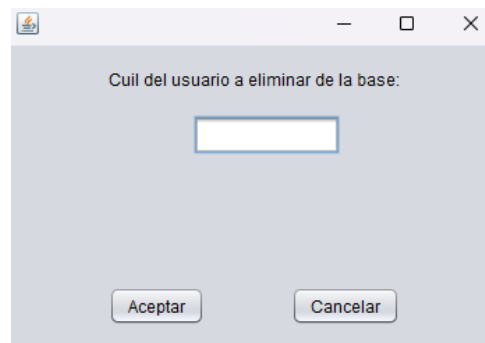


Figure 5: "Eliminar" de usuario. Es la ventana que permite borrar a un usuario del sistema. Se solicita solo el cui de este y se poseen los botones para aceptar o cancelar la operación

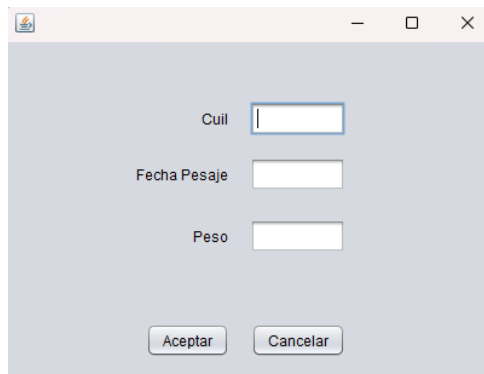


Figure 6: "Agregar pesaje". Es la ventana que permite agregar un pesaje de un usuario. Se solicita el cuil, la fecha del pesaje (en formato AAAA-MM-DD) y el peso en kg del usuario. Además se incluyen dos botones para aceptar o cancelar la operación

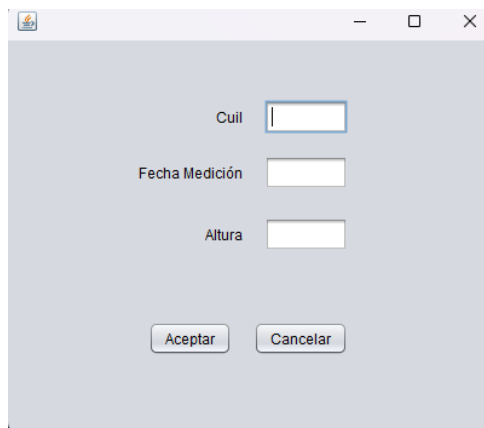


Figure 7: "Agregar medición". Es la ventana que permite agregar una medición de un usuario. Se solicita el cuil, la fecha de la medición (en formato AAAA-MM-DD) y la altura en metros (ej 1.70) del usuario. Además se incluyen dos botones para aceptar o cancelar la operación

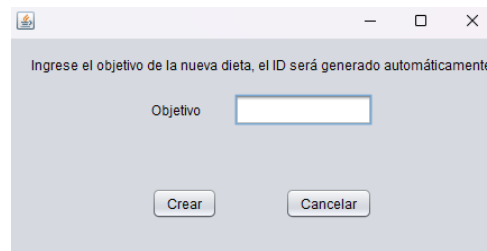


Figure 8: "Nuevo" de dieta. Nos permite dar de alta una dieta en el sistema. Solo se solicita el objetivo de la dieta puesto que el ID será generado automáticamente. Además, se incorporan los botones "Crear" y "Cancelar" para confirmar la operación

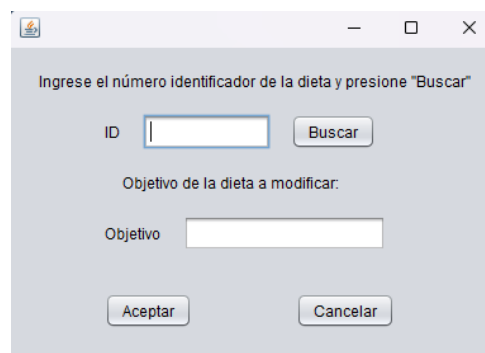


Figure 9: "Modificar" de dieta. En caso de error, se nos permite modificar el objetivo de la dieta. Al igual que en usuario, se incluye un campo ID y un botón "Buscar" con el que es posible autocompletar el Objetivo. Además, se incluyen nuevamente los botones "Aceptar" y "Cancelar" para operar

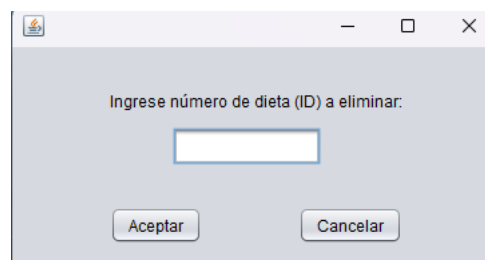
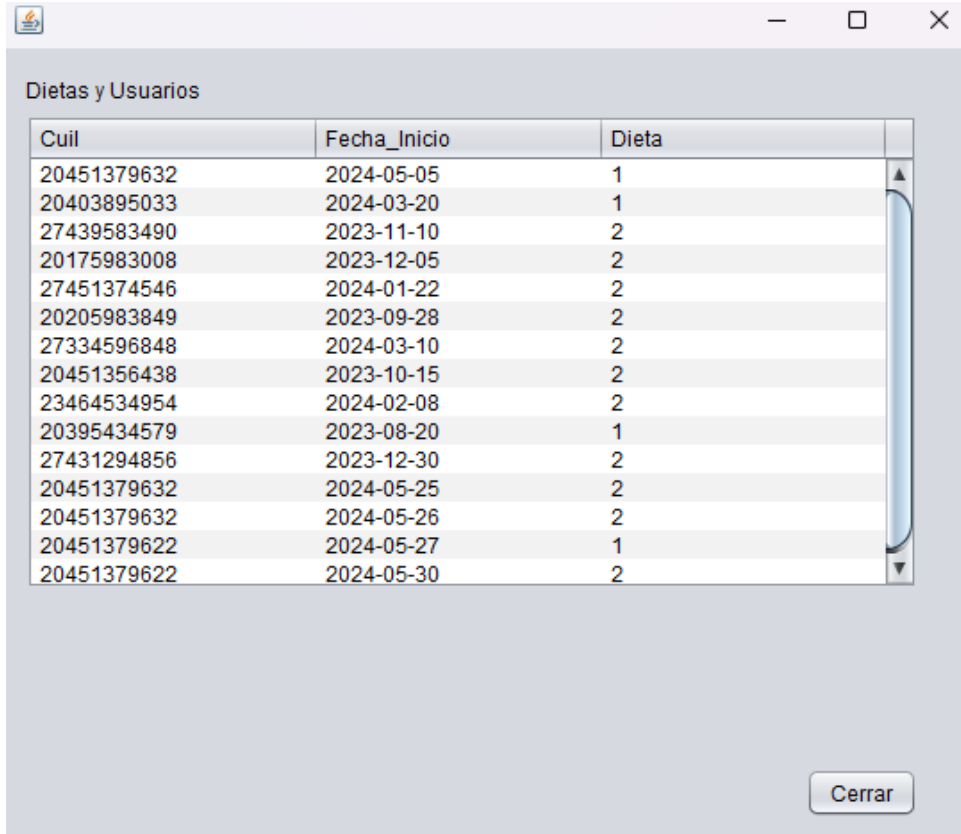


Figure 10: "Eliminar" de dieta. Se nos permite eliminar una dieta del sistema. Para ello, se nos solicita el ID (o número de dieta) y se nos brindan los botones "Aceptar" o "Cancelar" para confirmar la operación.



**Diets and Users**

Cuil	Fecha_Inicio	Dieta
20451379632	2024-05-05	1
20403895033	2024-03-20	1
27439583490	2023-11-10	2
20175983008	2023-12-05	2
27451374546	2024-01-22	2
20205983849	2023-09-28	2
27334596848	2024-03-10	2
20451356438	2023-10-15	2
23464534954	2024-02-08	2
20395434579	2023-08-20	1
27431294856	2023-12-30	2
20451379632	2024-05-25	2
20451379632	2024-05-26	2
20451379622	2024-05-27	1
20451379622	2024-05-30	2

**Cerrar**

Figure 11: "Ver dietas y usuarios". Nos muestra en forma de tabla la dieta asignada a cada usuario, el cuil del usuario y la fecha en la que se inició esa dieta. Es decir, nos permite visualizar el contenido de la tabla "realizadieta".

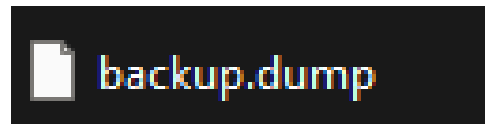


Figure 12: Archivo generado al clicar en el botón "Realizar backup". Además, es el archivo desde el que se importa el backup al presionar el botón "Importar backup". Debe estar en la misma carpeta que el programa y tener el nombre "backup.dump".



**Tabla: DEPORTE**

```

id: 1
nombre: Fútbol
-----
id: 2
nombre: Baloncesto
-----
id: 3
nombre: Tenis
-----
id: 4
nombre: Voleibol
-----
id: 5
nombre: Atletismo
-----

```

Figure 13: Fragmento del archivo pdf generado con la opción "Exportar PDF". Genera un archivo pdf con todos los datos de todas las tablas de la base de datos. Se coloca al principio el titulo de la tabla y posteriormente cada fila (con sus respectivas columnas) separadas por guiones.

**1. Usuarios**

**1.1. Tabla:**

Cuil	Nombre	Apellido	Sueño_Pr omedio	Agua_Pro medio	Fecha_Na cimiento	Sexo	Género	IMC	Estado_P eso	Horas_De porte_Se manales
20403895 033	Juan	González	7.0	4.0	1985-05-1 5	M	Hombre	25.66115	Sobrepeso	2.0
27439583 490	Maria	López	6.0	5.0	1992-07-2 0	F	Mujer	24.793388	Peso Normal	1.0
20175983 008	Carlos	Martínez	8.0	6.0	1988-03-1 0	M	Hombre	24.221453	Peso Normal	5.0
27451374 546	Laura	García	7.0	7.0	1995-11-2 5	F	Mujer	24.489796	Peso Normal	3.0
20205983 849	Diego	Rodríguez	6.0	4.0	1982-09-0 5	M	Hombre	24.092972	Peso Normal	2.0
27334596 848	Ana	Fernández	8.0	5.0	1998-12-1 2	F	Mujer	24.223997	Peso Normal	4.0
20451356 438	Javier	Pérez	7.0	6.0	1987-06-3 0	M	Hombre	24.382717	Peso Normal	3.0
23464534 954	Lucía	Gómez	6.0	7.0	1993-04-1 8	F	Mujer	24.696539	Peso Normal	2.0
20395434 579	Pablo	Hernández	8.0	4.0	1989-08-0 8	M	Hombre	25.01352	Sobrepeso	5.0
27431294 856	Marina	Torres	7.0	5.0	1996-02-2 8	F	Mujer	18.556772	Peso Normal	3.0
20451379 632	Tomas	Rando	7.0	6.0	2003-10-0 5	M	Hombre	26.1224	Sobrepeso	4.0

Figure 14: Primera parte del archivo generado al utilizar la opción "Exportar tablas". Se exportan las tablas "usuario" y "dieta", es decir, las tablas visualizadas en la ventana principal de la aplicación. Se exportan en formato pdf e incluyen todos los datos ahí mostrados.

### 1. Dietas

#### 1.1. Tabla:

ID	Objetivo
1	Perdida De Peso
2	Ganancia Muscular
3	Control de Peso

Figure 15: Segunda parte del archivo generado al utilizar la opción "Exportar tablas". La tabla mostrada corresponde a "dieta", visualizada en la ventana principal de la aplicación.

## 2 Trabajos Futuros

A continuación se detallan funciones que pueden implementarse en un futuro desarrollo del sistema.

1. Mejorar la interfaz gráfica. Luego de esta primera experiencia y de haber realizado esta primera versión de la aplicación considero que aprendí muchas cosas que son posibles de mejorar con más tiempo de desarrollo. Además, adquirí nuevas ideas que sería posible implementar en posteriores etapas.
2. Ampliar las funcionalidad ofrecidas en la interfaz gráfica. Si bien se implementó una parte del sistema, en el futuro sería posible ampliar las funcionalidades de la ventana de administrador y también incorporar una ventana para clientes y profesionales de la salud.
3. Mejorar el mecanismo de asignación de dietas. Actualmente, está implementado tomando en cuenta el IMC (Índice de masa corporal) de los usuarios. Sin embargo, no sería la manera óptima de realizar este proceso. Con más tiempo, creo que sería genial incorporar una lógica más compleja que tome en cuenta otros datos y no solo el IMC, algunos de estos podrían ser las horas semanales de deporte, la edad, entre otros. Además, sería óptimo poder conocer la impedancia bioeléctrica, sin embargo, la mayoría de los usuarios no conocen esto, por lo que sería difícil el uso del sistema. Otra opción a considerar sería que el usuario ingrese manualmente un objetivo personal y en relación a esto se le asigne una dieta.

## 3 Realizar backup de la base, exportar datos de la base en formato PDF y aplicar estrategias de seguridad

Se implementó la funcionalidad de backup con los dos botones que fueron explicados en la sección

1. La exportación de datos de la base en formato PDF fue explicada también en la sección 1. Con respecto a las estrategias de seguridad, se creó la pantalla del login vista en la sección 1 para poder ingresar a la aplicación. Además, se crearon ciertos roles que permiten ingresar a la base de datos con diferentes niveles de permisos.

```
--Crea el rol de administrador con todos los permisos para gestionar la base de datos  
CREATE ROLE administrador;  
ALTER ROLE administrador
```

```
LOGIN
SUPERUSER
CREATEROLE
CREATEDB
REPLICATION
BYPASSRLS
PASSWORD '1234';
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO administrador;

--Crea el rol de empleado con permisos para modificar datos de la base de datos
CREATE ROLE empleado;
ALTER ROLE empleado
LOGIN
PASSWORD '1234';
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO empleado;

--Crea el rol de cliente con permisos para consultar datos de algunas tablas de la base
CREATE ROLE cliente;
ALTER ROLE cliente
LOGIN
PASSWORD '1234';
GRANT SELECT ON realizadieta TO cliente;
GRANT SELECT ON dietatienecomida TO cliente;
GRANT SELECT ON dieta TO cliente;
```

## 4 Guía para ejecutar la aplicación.

Existen dos maneras para ejecutar el proyecto, puesto que fue incluido un .jar y el código fuente.

**Antes de ejecutar el proyecto:** Se debe crear la base de datos (nombre "uncuyo") con el script brindado en la etapa 5. Además, es importante crear los roles especificados. **Aclaración:** Se cambió el script de creación de la base de datos con respecto al presentado en la etapa 5, por lo que se incluye en el presente archivo todas las etapas ACTUALIZADAS.

### 4.1 Por ejecutable en consola.

Para poder ejecutar el archivo debemos poseer la versión **Java 21 o superior**. Se debe abrir la terminal o powershell dependiendo del sistema operativo. A partir de ahí, se debe navegar a la carpeta que contiene el archivo 2024dbapp.jar. Situados ahí, procedemos a escribir el siguiente comando:

```
java -jar 2024dbapp.jar
```

Así se nos ejecutará una ventana de login en el que podremos loguearnos utilizando las credenciales rol: "administrador" y contraseña: "1234"

## 4.2 Utilizando NetBeans.

Nuevamente, debemos poseer la versión **Java 21 o superior**.

Antes de ejecutar el archivo debemos verificar que la base de datos se encuentre creada con los roles definidos en la etapa 5 actualizada.

Se debe abrir el IDE NetBeans y abrir la carpeta que contiene el proyecto (2024dbapp). A partir de ahí, se debe ir al paquete **"com.uncuyo.dbapp.view"** y hacerle click derecho a la clase **"AdminLogin"** y seleccionar la opción **"Run file"**. A partir de ahí, se nos abrirá la ventana de login, para lo cual deberemos ingresar las credenciales rol: "administrador" y contraseña: "1234".