

Universidad Peruana de Ciencias Aplicadas



**Ingeniería de Sistemas de Información**  
**Ciclo:04**

**Curso:** Diseño de Base de Datos

**Sección:** WX 41

**Profesor:** Jorge Luis Mayta Guillermo

**Informe de Trabajo Final**

**Nombre del startup:** ARGLOP

**Nombre del producto:** Power Nutrition

**Integrantes:**

<b>Apellido Nombre</b>	<b>Código</b>	<b>Carrera</b>
Acosta Santamaria, Kevin Omar	u20181g830	Sistema de Información
Aquino Gutiérrez, Joshep Giovani	u202022089	Sistemas de Información
Cachique Espinoza Rafael	u20211d147	Sistemas de Información
Ccorahua Madera Leonardo Jesús	u202216007	Sistemas de Información
Corman Cardozo, Gilmar Angel A.	u202222557	Sistemas de Información
Mori De La Cruz Roly Romel	u20211d593	Sistemas de Información

**2023-01**



**Kristen Stewart**

**Age**  
26 años

**Occupation**  
Ingeniera de Sistemas

**Status**  
Soltera

**Location**  
Lima - Peru

**Tier**  
Estudiante

**Archetype**  
Dueño de departamento.

**Motivations**

- Incentive
- Fear
- Achievement
- Growth
- Power
- Social

**Bio**

Kristen es una universitaria que vive sola desde hace tiempo, le gusta estudiar, viajar y como también cuidar su salud. Le encanta el mundo de las tecnologías, ya que le gusta hacer diseños para las computadoras y también optimizar su buen funcionamiento.

**Personality**

- Introvertido
- Emocional
- Planificador
- Sensible

**Goals**

- Estimar costos y ejecución de un proyecto.
- Mejorar sus hábitos de nutrición saludable.
- Diseñar arquitecturas virtuales y soluciones inteligentes.
- Especializarse en Seguridad Informática

**Frustrations**

- No poder controlar el tiempo para realizar sus actividades diarias.
- Tener malos hábitos de nutrición.
- Ansiedad por la comida chatarra.

**Skills**

- Cyber Security
- Diseño
- Base de datos

**Technology**

- Redes e Internet
- Cyber Security
- Programación

**Brands**

- Notion
- 8fit

## Capítulo III: Database Design

### 3.1. Entities

En una base de datos, las entidades son importantes, ya que son una representación de un objeto o concepto del mundo real descrito en la base de datos. Ejemplos de nombres de entidades: Estudiante, Empleado, Artículo, Noticias, etc. Las entidades se describen en una estructura de base de datos utilizando un modelo de datos.

De esta manera, con respecto a nuestro startup hemos considerado las siguientes tablas que formarán parte de la base de datos de nuestro proyecto.

- **Usuarios:** Entidad que almacena la información básica y de contacto de todos los usuarios de la plataforma, tanto clientes como especialistas. Es fundamental para la autenticación y comunicación dentro del sistema.

Usuarios

- **Nutricionistas:** Entidad que almacena la información personal de especialistas en nutrición que la app permite que atienda a los usuarios.

Nutricionistas

- **Planes nutricionales:** Contiene información sobre los productos disponibles en la plataforma, como nombre, descripción o precio.

<b>Planes_nutricionales</b>

- **Boleta:** Entidad que almacena las compras hechas por cada usuario dentro de la plataforma.

<b>Boleta</b>

- **Alimentos:** Entidad que almacena la información sobre los alimentos que pueden aparecer en cada uno de los planes\_nutricionales.

<b>Alimentos</b>

- **Calificaciones Aplicativo:** Entidad que registra las calificaciones detalladas por parte de los usuarios, como experiencias, recomendaciones y sugerencias.

<b>Calificaciones_Aplicativo</b>

- **Calificaciones del nutricionista:** Entidad que registra las calificaciones de los nutricionistas, como experiencias positivas y recomendaciones.

<b>Calificaciones_Nutricionista</b>

- **Controles:** Entidad que almacena los registros de peso y talla que el usuario puede guardar de manera periódica y así pueda reflejar su progreso.

<b>Controles</b>

- **Consultas\_nutricionistas:** Entidad que almacena una relación entre usuarios y nutricionistas en la que el nutricionista le da recomendaciones al usuario según la información brindada.

<b>Consultas_nutricionistas</b>

- **Alimentos\_en\_plan:** Entidad que almacena las relaciones entre los alimentos y los planes\_nutricionales que ofrece la app.

<b>Alimentos_en_Plan</b>
--------------------------

- **Alimentos\_no\_deseados:** Entidad que almacena los alimentos no deseados por los usuarios. Facilita la gestión y tendencia de los usuarios.

<b>Alimentos_no_Deseados</b>
------------------------------

- **Valores\_nutricionales:** Entidad que almacena todos los valores nutricionales de un plan.

<b>Valores_nutricionales</b>
------------------------------

- **Planes\_por\_consulta:** Entidad que almacena la relación entre los planes que se recomiendan en cada consulta.

<b>Planes_por_consulta</b>
----------------------------

- **Detalles\_boleta:** Entidad que almacena los detalles sobre la compra de un plan nutricional y lo asocia a una boleta.

<b>Detalles_boleta</b>
------------------------

### 3.2. Atributes

Usuarios	
<b>Código</b>	<b>Código del registro</b>
<b>Nombre</b>	<b>Nombre del usuario</b>
<b>Clave</b>	<b>Clave de la cuenta del registro</b>
<b>Apellido_Paterno</b>	<b>Apellido Paterno</b>
<b>Apellido_Materno</b>	<b>Apellido Materno</b>
<b>Teléfono</b>	<b>Número de teléfono del usuario</b>
<b>Correo_Electrónico</b>	<b>Correo personal del usuario</b>
<b>Fecha_Nacimiento</b>	<b>Fecha de nacimiento del usuario</b>

<b>Genero</b>	<b>Género del usuario</b>
<b>Altura_Inicial</b>	<b>Altura inicial del usuario</b>
<b>Peso_Inicial</b>	<b>Peso inicial del usuario</b>
<b>Dirección</b>	<b>Dirección del usuario</b>
<b>Objetivos</b>	<b>Los objetivos que tendrá el usuario</b>
<b>Restricciones</b>	<b>Restricciones que tendrá el usuario</b>

Nutricionistas	
<b>Codigo_Nutricionista</b>	<b>Código del Nutricionista</b>
<b>Nombre</b>	<b>Nombre del Nutricionista</b>
<b>Apellido_Paterno</b>	<b>Apellido paterno del Nutricionista</b>
<b>Apellido_Materno</b>	<b>Apellido materno del Nutricionista</b>
<b>Correo</b>	<b>Correo del nutricionista</b>
<b>Especialidad</b>	<b>Especialidad del Nutricionista</b>
<b>Puntuacion</b>	<b>Puntuación promedio del Nutricionista</b>

Planes_nutricionales	
<b>Codigo</b>	<b>Código del plan nutricional</b>
<b>Nombre</b>	<b>Nombre del plan</b>
<b>Valores_nutricionales_codigo</b>	<b>Código de los valores nutricionales</b>
<b>Descripción</b>	<b>Descripción del plan nutricional</b>
<b>Precio</b>	<b>Precio del plan nutricional</b>

Boleta	
<b>Usuario_código</b>	<b>Código del usuario</b>
<b>Código</b>	<b>Código de la boleta</b>
<b>Fecha</b>	<b>Fecha de la compra</b>
<b>Delivery</b>	<b>Determina si el usuario desea que se le lleve su orden a su domicilio</b>
<b>Recargo_delivery</b>	<b>Monto adicional por delivery</b>
<b>Lugar_recogida</b>	<b>Lugar en el que el usuario pueda recoger su orden</b>
<b>Monto_total</b>	<b>Monto total del plan</b>

Consultas_nutricionistas	
Usuarios_codigo	Código del usuario
Nutricionistas_codigo	Código del nutricionista
Código	Código de la consulta
Fecha	Fecha de los comentarios
Comentarios	Detalles y comentarios por el nutricionista

Calificaciones_nutricionista	
Usuarios_codigo	Código del usuario
Nutricionistas_codigo	Código del nutricionista
Calificación	Calificación al nutricionista

Alimentos_en_plan	
Alimentos_codigo	Código del alimento
Planes_recomendados_codigo	Código del plan recomendado
Cantidad_gramos	Cantidad de gramos del alimento dentro del plan

Controles	
Codigo	Código de los controles
Usuarios_codigo	Código de los usuarios
Fecha	Descripción de la fecha
Peso	Descripción del peso
Talla	Descripción de la talla

Alimentos	
Codigo	Código del alimento
Nombre	Nombre del alimento
Categoria	Categorías de los alimentos
Calorias	Calorías de cada alimento
Proteinas	Proteínas de un alimento
Carbohidratos	Carbohidratos de un alimento
Grasas	Grasas de un alimento

Calificaciones_aplicativo	
<b>Codigo</b>	<b>Código de la calificación a la app</b>
<b>Usuarios_codigo</b>	<b>Código del usuario</b>
<b>Calificacion</b>	<b>Calificación dada por el usuario</b>
<b>Comentario</b>	<b>Comentario hecho por el usuario</b>
<b>Fecha</b>	<b>Fecha en la que se hizo la calificación</b>

Alimentos_no_deseados	
<b>Alimentos_codi</b>	<b>Código de los alimentos</b>
<b>Usuarios_codig</b>	<b>Descripción del código de usuario</b>

Valores_nutricionales	
<b>Codigo</b>	<b>Código del valor nutricional</b>
<b>Calorias</b>	<b>Calorías del plan</b>
<b>Proteinas</b>	<b>Proteínas del plan</b>
<b>Carbohidratos</b>	<b>Carbohidratos del plan</b>
<b>Grasas</b>	<b>Grasas del plan</b>

Detalles_boleta	
<b>Planes_nutricionales_codigo</b>	<b>Código del plan nutricional</b>
<b>Boleta_codigo</b>	<b>Código de la boleta a la que pertenece esta parte de la compra</b>
<b>Cantidad</b>	<b>Cantidad del plan nutricional que el usuario desea llevar</b>

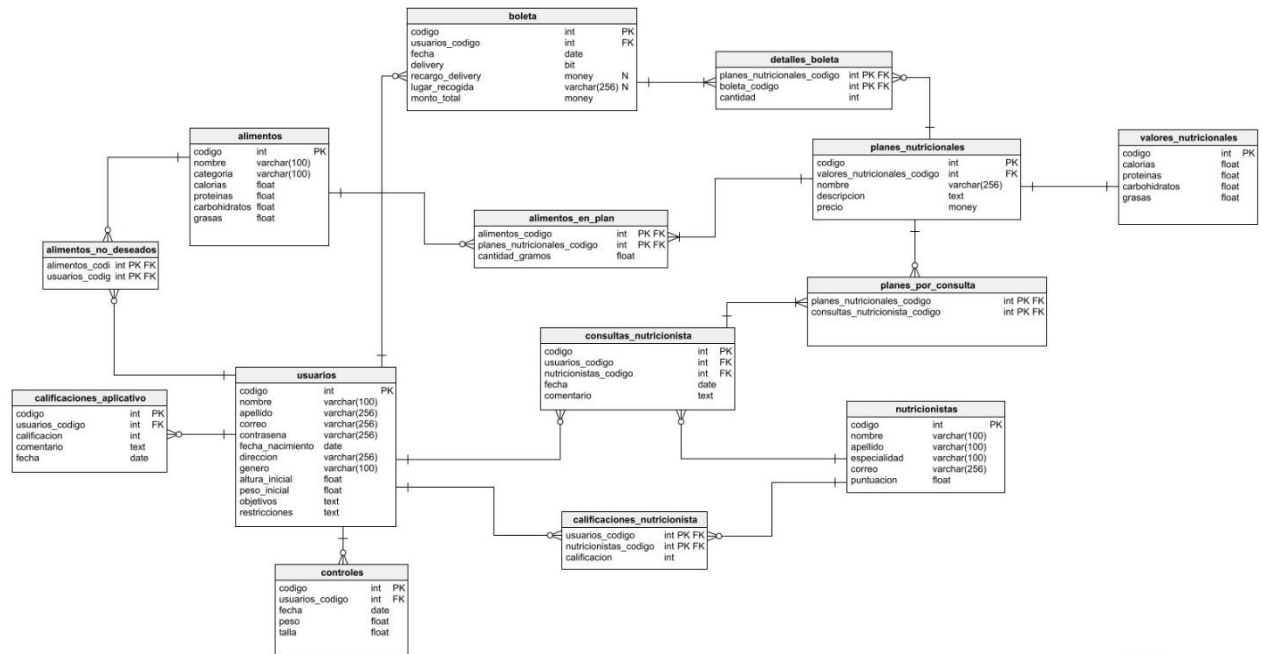
Planes_por_consulta	
<b>Planes_nutricionales_codigo</b>	<b>Código del plan nutricional</b>
<b>Consultas_nutricionista_codigo</b>	<b>Código de la consulta en la que se sugiere el plan</b>

### 3.3. Logical Data Model

A continuación, se procede a representar las entidades y atributos a implementar en la base de datos del proyecto **Arglor** a través de un modelo gráfico. En este sentido, el equipo de trabajo

alternó por seleccionar el modelo de datos lógico. Los motivos recaen en el establecimiento de las relaciones lógicas entre las diferentes entidades a partir de la cardinalidad.

<https://my.vertabelo.com/public-model-view/SVvEiJhLMTuW3oY5q1qKgkkn0XQCMDPA9ZnJM9day127EIq8WaVPGCtvrYW2WeK?x=2628&y=3413&zoom=0.4218>



### 3.4. Collections

“Usuarios”: La colección usuarios se usará para guardar la información de las personas que se registren en la app. Se usará la tabla usuarios, controles, calificaciones\_aplicativo, alimentos y alimentos\_no\_deseados. Seleccionamos estas tablas porque nos permite obtener toda la información sobre el usuario y así en el nutricionista pueda acceder a esta en las consultas.

“Compras\_plan”: La colección compras\_plan se usará para detallar la compra de un plan hecha por el usuario. Se usarán las tablas boleta, detalles\_boleta. Esta colección nos ayudará a tener toda la información de cada compra que haga cada usuario, incluyendo los planes que comprará, la fecha o el precio del delivery.

“Planes\_nutricionales”: La colección planes\_nutricionales se usará para detallar un plan hecho para el usuario. Se usarán las tablas planes\_recomendados, valores\_nutricionales, alimentos y alimentos\_en\_plan. Esta colección permitirá obtener la información de los alimentos y de los valores nutricionales contenidos en cada uno de los planes para recomendar y vender.

“Nutricionistas”: La colección nutricionistas se usará para guardar la información de los nutricionistas que sugerirán los planes nutricionales. Se usarán las tablas nutricionistas,



calificaciones\_nutricionista, consultas\_nutricionista, planes\_por\_consulta. Esta colección servirá para que los usuarios puedan ver la información básica de los nutricionistas y se puedan almacenar las consultas entre nutricionistas y usuarios.

### **3.5. Schema Validation**

A continuación, se muestra la validación de las colecciones en nuestra base de datos no relacional utilizando JSON Schema. Esto se realiza con el objetivo de que MongoDB genere errores o rechace documentos que no cumplan con las reglas de validación al ingresar datos a la base de datos. Es importante tener en cuenta que cuando se agrega validación a una colección, los datos existentes no se validarán automáticamente a menos que se realice una actualización o modificación en ellos. En nuestro caso, las colecciones se crearon junto con la validación, lo que significa que los documentos ingresados ya están validados.

## Collection

usuarios:

```

db.createCollection("usuarios", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["_id", "nombre", "apellido", "correo", "contrasena", "fecha_nacimiento", "direccion", "genero",
        "altura_inicial", "peso_inicial", "objetivos", "restricciones"],
      properties: {
        _id: {
          bsonType: "string"
        },
        nombre: {
          bsonType: "string"
        },
        apellido: {
          bsonType: "string"
        },
        correo: {
          bsonType: "string",
          pattern: "^[^\\s@]+@[^\\s@]+\\.([^\\s@]+)$"
        },
        contrasena: {
          bsonType: "string"
        },
        fecha_nacimiento: {
          bsonType: "date"
        },
        direccion: {
          bsonType: "string"
        },
        genero: {
          bsonType: "string",
          enum: ["Masculino", "Femenino"]
        },
        altura_inicial: {
          bsonType: ["double", "int"],
          minimum: 0
        },
        peso_inicial: {
          bsonType: ["double", "int"],
          minimum: 0
        },
        objetivos: {
          bsonType: "string"
        },
        restricciones: {
          bsonType: "string"
        },
        alimentos_no_deseados: {
          bsonType: "array",
          uniqueItems: true,
          items: {
            bsonType: "string"
          }
        },
        controles: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: ["fecha", "peso", "talla"],
            properties: {
              fecha: {
                bsonType: "date"
              },
              peso: {
                bsonType: ["double", "int"],
                minimum: 0
              },
              talla: {
                bsonType: ["double", "int"],
                minimum: 0
              }
            }
          }
        }
      }
    }
  }
})

```

```

    calificaciones_app: {
      bsonType: "array",
      items: {
        bsonType: "object",
        required: ["calificacion", "comentario", "fecha"],
        properties: {
          calificacion: {
            bsonType: "int",
            minimum: 1,
            maximum: 5
          },
          comentario: {
            bsonType: "string"
          },
          fecha: {
            bsonType: "date"
          }
        }
      }
    }
  }
}
})

```

## Collection “Planes\_Nutricionales”

```

db.createCollection("planes_nutricionales",{
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["_id", "nombre", "alimentos", "descripcion", "precio"],
      properties: {
        _id: {
          bsonType: "string"
        },
        nombre: {
          bsonType: "string"
        },
        alimentos: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: ["nombre", "categoria", "calorias", "proteinas", "carbohidratos", "grasas", "cantidad"],
            properties: {
              nombre: {
                bsonType: "string"
              },
              categoria: {
                bsonType: "string"
              },
              calorias: {
                bsonType: ["double", "int"],
                minimum: 0.0
              },
              proteinas: {
                bsonType: ["double", "int"],
                minimum: 0.0
              },
              carbohidratos: {
                bsonType: ["double", "int"],
                minimum: 0.0
              },
              grasas: {

```

```

        bsonType: ["double", "int"],
        minimum: 0.0
    },
    cantidad: {
        bsonType: ["double", "int"]
    }
}
},
uniqueItems: true
},
descripcion: {
    bsonType: "string"
},
precio: {
    bsonType: ["double", "int"],
    minimum: 0.0
},
valor_nutricional: {
    bsonType: "array",
    items: {
        bsonType: "object",
        required: ["calorias", "proteinas", "carbohidratos", "grasas"],
        properties: {
            calorias: {
                bsonType: ["double", "int"],
                minimum: 0.0
            },
            proteinas: {
                bsonType: ["double", "int"],
                minimum: 0.0
            },
            carbohidratos: {
                bsonType: ["double", "int"],
                minimum: 0.0
            },
            grasas: {
                bsonType: ["double", "int"],
                minimum: 0.0
            }
        }
    }
}
}
}
})
```

### Collection “Compras\_plan”

```

db.createCollection("compras_plan", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["_id", "id_usuario", "detalles_compra", "fecha", "delivery"],
      properties: {
        _id: {
          bsonType: "string"
        },
        id_usuario: {
          bsonType: "string"
        },
        detalles_compra: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: ["id_plan", "cantidad"],
            properties: {
              id_plan: {
                bsonType: "string"
              },
              cantidad: {
                bsonType: "int",
                minimum: 1
              }
            }
          }
        },
        fecha: {
          bsonType: "date"
        },
        delivery: {
          bsonType: "bool"
        },
        recargo_delivery: {
          bsonType: ["double", "int"],
          minimum: 0.0
        },
        monto_total: {
          bsonType: ["double", "int"],
          minimum: 0.0
        },
        lugar_recogida: {
          bsonType: "string"
        }
      }
    }
  }
});

```

## Collection “Nutricionistas”

```

db.createCollection("nutricionistas", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["_id", "nombre", "apellido", "especialidad", "correo", "puntuacion"],
      properties: {
        _id: {
          bsonType: "string"
        },
        nombre: {
          bsonType: "string"
        },
        apellido: {
          bsonType: "string"
        },
        especialidad: {
          bsonType: "string"
        },
        correo: {
          bsonType: "string",
          pattern: "^[^\\s@]+@[^\\s@]+\\.\\.[^\\s@]+$"
        },
        puntuaciones_usuarios: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: ["usuarios_codigo", "calificacion"],
            properties: {
              usuarios_codigo: {
                bsonType: "string"
              },
              calificacion: {
                bsonType: "int",
                minimum: 1,
                maximum: 5
              }
            }
          }
        }
      }
    }
  }
})

```

```

    }
  }
},
puntuacion: {
  bsonType: ["double", "int"],
  minimum: 0,
  maximum: 5
},
consultas: {
  bsonType: "array",
  items: {
    bsonType: "object",
    required: ["usuarios_codigo", "fecha", "comentarios", "planes_recomendados"],
    properties: {
      usuarios_codigo: {
        bsonType: "string"
      },
      fecha: {
        bsonType: "date"
      },
      comentarios: {
        bsonType: "string"
      },
      planes_recomendados: {
        bsonType: "array",
        items: {
          bsonType: "string"
        }
      }
    }
  }
}
}
}
}
})

```

### 3.6. Data Model Patterns

Para relacionar las colecciones dentro de nuestra base de datos en Mongo db, hemos utilizado la categoría de patrón "Embedded document pattern". Las vistas con los datos embebidos de nuestro proyecto basadas en el Logical Data model son las siguientes:

#### Colección usuarios:

```
{
  _id: "id_usuario_1",
  nombre: "Bernardo",
  apellido: "Silva",
  correo: "b_silva123@gmail.com",
  contrasena: "citizen777",
  fecha_nacimiento: ISODate("1990-01-01"),
  direccion: "Av. Central 457 - Chorrillos",
  genero: "Masculino",
  altura_inicial: 170.0,
  peso_inicial: 70.0,
  objetivos: "Quiero obtener mayor masa muscular con el consumo de fibra saludable",
  restricciones: "Soy intolerante a la lactosa",
  alimentos_no_deseados: ["Zanahoria", "Queso"],
  controles: [
    {fecha: ISODate("2023-06-04"), peso: 70.0, talla: 1.70}
  ],
  calificaciones_app: [
    {calificacion: 5, comentario: "Excelente aplicación", fecha: ISODate("2023-03-04")},
    {calificacion: 5, comentario: "Recomendada, me ayudo a aumentar musculo", fecha:
      ISODate("2023-11-04")}
  ]
}
```

#### Colección

#### nutricionistas:

```
{
  _id: "id_nutricionista_1",
  nombre: "Sergio",
  apellido: "Andrade",
}
```

```

especialidad: "Nutricionista deportivo",
correo: "sergi_a11@gmail.com",
puntuaciones_usuarios: [{usuarios_codigo: "id_usuario_1", calificacion: 5}],
puntuacion: 0.0,
consultas: [{usuarios_codigo: "id_usuario_1", fecha: ISODate("2023-06-04"),
comentarios: "Como veo que deseas ganar masa muscular, te recomendaría incluir la
palta en tu dieta porque la palta es rica en grasas monoinsaturadas, que son consideradas
grasas saludables. Estas grasas son importantes para la salud del corazón y también
ayudan a reducir la inflamación en el cuerpo, lo cual es beneficioso para el proceso de
recuperación muscular después del entrenamiento.", planes_recomendados:
["id_plan_1", "id_plan_2"]}], {usuarios_codigo: "id_usuario_4", fecha: ISODate("2023-
06-14"),
comentarios: "Si deseas tener un cuerpo más flexible, te recomendaría incorporar el
consumo de frutas en tu dieta diaria. Las frutas son una excelente fuente de vitaminas,
minerales y antioxidantes que ayudan a fortalecer tus músculos y mejorar la flexibilidad
de tu cuerpo. Además, su contenido de agua y fibra te mantendrá hidratado y contribuirá
a una mejor digestión.", planes_recomendados: ["id_plan_3", "id_plan_5"]}]]
}

```

Colección planes\_nutricionales:

```

{
  _id: "id_plan_1",
  nombre: "Ensalada de quinua y palta",
  alimentos: [
    {nombre: "Palta", categoria: "Frutas", calorías: 160.0, proteínas: 2.0, carbohidratos:
    9.0, grasas: 14.7, cantidad: 100},
    {nombre: "Quinua", categoria: "Granos Integrales", calorías: 370.0, proteínas: 14.0,
    carbohidratos: 64.0, grasas: 6.7, cantidad: 50}],
  descripcion: "La ensalada de quinua y palta es una opción deliciosa y nutritiva que
  combina la textura suave y cremosa de la palta con el grano rico en proteínas y fibra de
  la quinua. La ensalada de quinua y palta puede servirse como plato principal para una
  comida ligera o como acompañamiento de otros platos. Combina bien con proteínas a la
  parrilla, pescado, pollo, o como guarnición en barbacoas y comidas al aire libre.",
  precio: 14.0
}

```

Colección compras\_plan:

```

{
  _id: "id_compras_plan_1",
  id_usuario: "id_usuario_1",
  detalles_compra: [{id_plan: "id_plan_1", cantidad: 3}, {id_plan: "id_plan_2", cantidad:
  2}],
  fecha: ISODate("2023-01-01"),
  delivery: true,
}

```



```

recargo_delivery: 8.5,
monto_total: 80.5,
}

```

## Capítulo IV: Database Implementation

### 4.1 Relational Database Management System

Según Computer Weekly un sistema de gestión de bases de datos relacionales (RDBMS) es una colección de programas y capacidades que permiten a los equipos de TI ya otros crear, actualizar, administrar e interactuar con una base de datos relacional. La mayoría de los RDBMS comerciales utilizan Structured Query Language (SQL) para acceder a la base de datos, aunque SQL fue inventado después del desarrollo inicial del modelo relacional y no es necesario para su uso. En un RDBMS, la estructura de tabla basada en filas conecta elementos de datos relacionados y garantiza la seguridad, precisión, integridad y consistencia de los datos. Las funciones básicas incluyen las operaciones CRUD (crear, leer, actualizar y eliminar), y el RDBMS también proporciona diccionarios de datos, gestión del almacenamiento de datos, acceso mediante SQL, gestión de seguridad, soporte para administradores de bases de datos (DBA) y cumplimiento de ACID (atomicidad, consistencia, aislamiento y durabilidad). Los RDBMS son fundamentales en diversas aplicaciones empresariales, como contabilidad, reservas de viajes y ventas minoristas en línea, así como en sistemas de informes, análisis y almacenamiento de datos.

#### **-Lenguajes en un sistema gestor de bases de datos relacional**

A través de un sistema gestor de bases de datos, se pueden administrar las bases de datos utilizando cuatro lenguajes principales: DML (Data Manipulation Language), DDL (Data Definition Language), DCL (Data Control Language) y TCL (Transaction Control Idioma). A continuación, se proporciona una breve explicación de cada uno de estos lenguajes:

- DML (Data Manipulation Language): El lenguaje de manipulación de datos se utiliza para manipular los datos dentro de una base de datos. Permite realizar como operaciones la inserción (INSERT), actualización (UPDATE) y eliminación (DELETE) de registros en las tablas de la base de datos. Además, el DML también proporciona la capacidad de recuperar datos mediante consultas SELECT.

- DDL (Data Definition Language): El lenguaje de definición de datos se utiliza para definir y modificar la estructura de la base de datos. Permite crear tablas (CREATE TABLE), definir columnas, tipos de datos y restricciones, así como modificar la estructura existente mediante instrucciones como ALTER TABLE o DROP TABLE. El DDL también se utiliza para definir índices, vistas, secuencias y otros objetos relacionados con la estructura de la base de datos.

•**DCL (Data Control Language):** El lenguaje de control de datos se utiliza para administrar los permisos y la seguridad en la base de datos. Permite conceder privilegios de acceso (GRANT), revocar privilegios (REVOKE) y controlar el acceso a los objetos de la base de datos. Con el DCL, se pueden gestionar los permisos de los usuarios y roles, presentar políticas de seguridad para proteger los datos y garantizar la integridad de la base de datos.

•**TCL (Transaction Control Language):** El lenguaje de control de transacciones se utiliza para gestionar las transacciones en la base de datos. Permite definir el inicio (BEGIN), la finalización (COMMIT) y la cancelación (ROLLBACK) de transacciones. Las transacciones son unidades lógicas de trabajo que agrupan varias operaciones y aseguran la consistencia y la integridad de los datos. El TCL permite controlar el estado y la durabilidad de las transacciones en la base de datos.

Estos cuatro lenguajes principales (DML, DDL, DCL y TCL) son fundamentales en un sistema gestor de bases de datos, ya que pueden manipular y administrar los datos, definir la estructura de la base de datos, controlar los permisos y la seguridad, y gestionar las transacciones. Proporcione las herramientas necesarias para interactuar con eficacia con una base de datos relacional y aproveche al máximo sus capacidades.

### **-Nuestra elección como RDBMS: SQL Server Management Studio**

SQL Server Management Studio (SSMS) es una herramienta de administración gráfica y de desarrollo utilizada para administrar y trabajar con bases de datos de Microsoft SQL Server. Proporciona una interfaz de usuario intuitiva y un conjunto completo de características para realizar tareas de administración, desarrollo y depuración en entornos de bases de datos SQL Server.



### **¿Por qué optamos por usar el SQL Server?**

SQL Server es una opción popular como sistema gestor de bases de datos debido a sus ventajas clave:

- **Potencia y rendimiento:** Diseñado para manejar grandes volúmenes de datos y cargas de trabajo exigentes, con técnicas avanzadas de optimización de consultas.
- **Escalabilidad:** Permite crecer junto con las necesidades de la organización mediante la adición de servidores, división de tablas y técnicas de clustering.
- **Seguridad:** Ofrece funciones de autenticación, autorización, cifrado y auditoría para proteger los datos y cumplir con estándares de seguridad.
- **Herramientas de administración:** SQL Server Management Studio (SSMS) facilita la administración, monitoreo y configuración de la base de datos.

- Integración con el ecosistema de Microsoft: Se integra de manera fluida con otros productos y tecnologías de Microsoft, simplificando el desarrollo y administración de aplicaciones.

En resumen, SQL Server proporciona potencia, escalabilidad, seguridad y herramientas robustas, junto con una estrecha integración con el ecosistema de Microsoft, lo que lo convierte en una opción atractiva para la administración de bases de datos en organizaciones.

### **-Otras opciones de sistemas de base de datos:**

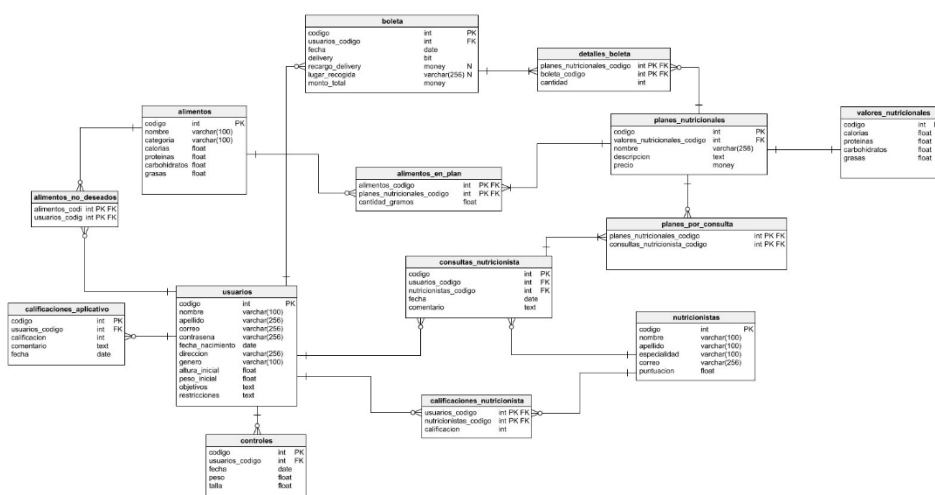
Además de SQL Server, existen otras opciones populares como RDBMS. Algunas de ellas son:

**Oracle Database:** Es uno de los sistemas gestores de bases de datos más utilizados en el mundo empresarial. Ofrece un alto rendimiento, escalabilidad y seguridad avanzada. Oracle Database es conocido por su capacidad de manejar grandes volúmenes de datos y su amplia compatibilidad con diferentes plataformas.

**MySQL:** Es un sistema gestor de bases de datos de código abierto muy popular, especialmente en entornos web. MySQL es conocido por su velocidad, escalabilidad y facilidad de uso. Es ampliamente utilizado en aplicaciones web y sitios de comercio electrónico.

Estas son solo algunas de las opciones disponibles en el mercado. La elección del sistema gestor de bases de datos obtenidos de los requisitos específicos de la organización, el entorno tecnológico, el tamaño de los datos y las necesidades de rendimiento y seguridad. Es importante evaluar cada opción en función de los requisitos y objetivos del proyecto.

## **4.2. Physical Data Model**



Vertabelo

**Link para mejor visualización:**

<https://my.vertabelo.com/public-model-view/SVvEiJhLMTuW3oY5q1qKgkkn0XQCMDPA9ZnJM9day127Elfq8WaVPGCtvrYW2WeK?x=2628&y=3413&zoom=0.4218>

### SCRIPT:

-- tables

-- Table: alimentos

```
CREATE TABLE alimentos (
  codigo int NOT NULL,
  nombre varchar(100) NOT NULL,
  categoria varchar(100) NOT NULL,
  calorias float NOT NULL,
  proteinas float NOT NULL,
  carbohidratos float NOT NULL,
  grasas float NOT NULL,
  CONSTRAINT alimentos_pk PRIMARY KEY (codigo)
);
```

-- Table: alimentos\_en\_plan

```
CREATE TABLE alimentos_en_plan (
  alimentos_codigo int NOT NULL,
  planes_nutricionales_codigo int NOT NULL,
  cantidad_gramos float NOT NULL,
  CONSTRAINT alimentos_en_plan_pk PRIMARY KEY
(alimentos_codigo,planes_nutricionales_codigo)
);
```

-- Table: alimentos\_no\_deseados

```
CREATE TABLE alimentos_no_deseados (
  alimentos_codigo int NOT NULL,
  usuarios_codigo int NOT NULL,
  CONSTRAINT alimentos_no_deseados_pk PRIMARY KEY
(alimentos_codigo,usuarios_codigo)
);
```

-- Table: boleta

```
CREATE TABLE boleta (
  codigo int NOT NULL,
  usuarios_codigo int NOT NULL,
  fecha date NOT NULL,
  delivery bit NOT NULL,
  recargo_delivery money NULL,
  lugar_recogida varchar(256) NULL,
```

```

    monto_total money NOT NULL,
    CONSTRAINT boleta_pk PRIMARY KEY (codigo)
);

```

-- Table: calificaciones\_aplicativo

```

CREATE TABLE calificaciones_aplicativo (
    codigo int NOT NULL DEFAULT null,
    usuarios_codigo int NOT NULL,
    calificacion int NOT NULL DEFAULT null,
    comentario text NOT NULL DEFAULT null,
    fecha date NOT NULL DEFAULT null,
    CONSTRAINT calificacion_aplicativo_pk PRIMARY KEY (codigo)
);

```

-- Table: calificaciones\_nutricionista

```

CREATE TABLE calificaciones_nutricionista (
    usuarios_codigo int NOT NULL,
    nutricionistas_codigo int NOT NULL,
    calificacion int NOT NULL,
    CONSTRAINT calificaciones_nutricionista_pk PRIMARY KEY
    (usuarios_codigo,nutricionistas_codigo)
);

```

-- Table: consultas\_nutricionista

```

CREATE TABLE consultas_nutricionista (
    codigo int NOT NULL,
    usuarios_codigo int NOT NULL,
    nutricionistas_codigo int NOT NULL,
    fecha date NOT NULL DEFAULT null,
    comentario text NOT NULL DEFAULT null,
    CONSTRAINT cometarios_nutricionista_pk PRIMARY KEY (codigo)
);

```

-- Table: controles

```

CREATE TABLE controles (
    codigo int NOT NULL DEFAULT null,
    usuarios_codigo int NOT NULL,
    fecha date NOT NULL DEFAULT null,
    peso float NOT NULL DEFAULT null,
    talla float NOT NULL,
    CONSTRAINT historial_peso_pk PRIMARY KEY (codigo)
);

```

-- Table: detalles\_boleta

```
CREATE TABLE detalles_boleta (
  planes_nutricionales_codigo int NOT NULL,
  cantidad int NOT NULL,
  boleta_codigo int NOT NULL,
  CONSTRAINT detalles_boleta_pk PRIMARY KEY
  (planes_nutricionales_codigo,boleta_codigo)
);
```

-- Table: nutricionistas

```
CREATE TABLE nutricionistas (
  codigo int NOT NULL DEFAULT null,
  nombre varchar(100) NOT NULL DEFAULT null,
  apellido varchar(100) NOT NULL DEFAULT null,
  especialidad varchar(100) NOT NULL DEFAULT null,
  correo varchar(256) NOT NULL DEFAULT null,
  puntuacion float NOT NULL,
  CONSTRAINT nutricionista_pk PRIMARY KEY (codigo)
);
```

-- Table: planes\_nutricionales

```
CREATE TABLE planes_nutricionales (
  codigo int NOT NULL,
  valores_nutricionales_codigo int NOT NULL,
  nombre varchar(256) NOT NULL,
  descripcion text NOT NULL,
  precio money NOT NULL,
  CONSTRAINT planes_nutricionales_pk PRIMARY KEY (codigo)
);
```

-- Table: planes\_por\_consulta

```
CREATE TABLE planes_por_consulta (
  planes_nutricionales_codigo int NOT NULL,
  consultas_nutricionista_codigo int NOT NULL,
  CONSTRAINT planes_por_consulta_pk PRIMARY KEY
  (planes_nutricionales_codigo,consultas_nutricionista_codigo)
);
```

-- Table: usuarios

```
CREATE TABLE usuarios (
  codigo int NOT NULL DEFAULT null,
  nombre varchar(100) NOT NULL DEFAULT null,
  apellido varchar(256) NOT NULL DEFAULT null,
  correo varchar(256) NOT NULL DEFAULT null,
  contrasena varchar(256) NOT NULL DEFAULT null,
```

```

    fecha_nacimiento date NOT NULL DEFAULT null,
    direccion varchar(256) NOT NULL,
    genero varchar(100) NOT NULL DEFAULT null,
    altura_inicial float NOT NULL DEFAULT null,
    peso_inicial float NOT NULL DEFAULT null,
    objetivos text NOT NULL,
    restricciones text NOT NULL,
    CONSTRAINT usuario_pk PRIMARY KEY (codigo)
);

-- Table: valores_nutricionales
CREATE TABLE valores_nutricionales (
    codigo int NOT NULL DEFAULT null,
    calorias float NOT NULL DEFAULT null,
    proteinas float NOT NULL DEFAULT null,
    carbohidratos float NOT NULL DEFAULT null,
    grasas float NOT NULL DEFAULT null,
    CONSTRAINT valores_nutricionales_pk PRIMARY KEY (codigo)
);

-- foreign keys
-- Reference: alimento_en_plan_alimentos (table: alimentos_en_plan)
ALTER TABLE alimentos_en_plan ADD CONSTRAINT alimento_en_plan_alimentos
    FOREIGN KEY (alimentos_codigo)
    REFERENCES alimentos (codigo);

-- Reference: alimentos_en_plan_planes_nutricionales (table: alimentos_en_plan)
ALTER TABLE alimentos_en_plan ADD CONSTRAINT
alimentos_en_plan_planes_nutricionales
    FOREIGN KEY (planes_nutricionales_codigo)
    REFERENCES planes_nutricionales (codigo);

-- Reference: alimentos_no_deseados_alimentos (table: alimentos_no_deseados)
ALTER TABLE alimentos_no_deseados ADD CONSTRAINT
alimentos_no_deseados_alimentos
    FOREIGN KEY (alimentos_codigo)
    REFERENCES alimentos (codigo);

-- Reference: alimentos_no_deseados_usuarios (table: alimentos_no_deseados)
ALTER TABLE alimentos_no_deseados ADD CONSTRAINT
alimentos_no_deseados_usuarios
    FOREIGN KEY (usuarios_codigo)
    REFERENCES usuarios (codigo);

```

```

-- Reference: calificaciones_aplicativo_usuarios (table: calificaciones_aplicativo)
ALTER      TABLE      calificaciones_aplicativo      ADD      CONSTRAINT
calificaciones_aplicativo_usuarios
    FOREIGN KEY (usuarios_codigo)
    REFERENCES usuarios (codigo);

-- Reference: calificaciones_nutricionista_nutricionistas (table: calificaciones_nutricionista)
ALTER      TABLE      calificaciones_nutricionista      ADD      CONSTRAINT
calificaciones_nutricionista_nutricionistas
    FOREIGN KEY (nutricionistas_codigo)
    REFERENCES nutricionistas (codigo);

-- Reference: calificaciones_nutricionista_usuarios (table: calificaciones_nutricionista)
ALTER      TABLE      calificaciones_nutricionista      ADD      CONSTRAINT
calificaciones_nutricionista_usuarios
    FOREIGN KEY (usuarios_codigo)
    REFERENCES usuarios (codigo);

-- Reference: cometarios_nutricionista_nutricionistas (table: consultas_nutricionista)
ALTER      TABLE      consultas_nutricionista      ADD      CONSTRAINT
cometarios_nutricionista_nutricionistas
    FOREIGN KEY (nutricionistas_codigo)
    REFERENCES nutricionistas (codigo);

-- Reference: cometarios_nutricionista_usuarios (table: consultas_nutricionista)
ALTER      TABLE      consultas_nutricionista      ADD      CONSTRAINT
cometarios_nutricionista_usuarios
    FOREIGN KEY (usuarios_codigo)
    REFERENCES usuarios (codigo);

-- Reference: compra_plan_usuarios (table: boleta)
ALTER TABLE boleta ADD CONSTRAINT compra_plan_usuarios
    FOREIGN KEY (usuarios_codigo)
    REFERENCES usuarios (codigo);

-- Reference: detalles_boleta_boleta (table: detalles_boleta)
ALTER TABLE detalles_boleta ADD CONSTRAINT detalles_boleta_boleta
    FOREIGN KEY (boleta_codigo)
    REFERENCES boleta (codigo);

-- Reference: detalles_boleta_planes_nutricionales (table: detalles_boleta)
ALTER TABLE detalles_boleta ADD CONSTRAINT detalles_boleta_planes_nutricionales
    FOREIGN KEY (planes_nutricionales_codigo)
    REFERENCES planes_nutricionales (codigo);

```



```

-- Reference: historiales_usuarios (table: controles)
ALTER TABLE controles ADD CONSTRAINT historiales_usuarios
    FOREIGN KEY (usuarios_codigo)
    REFERENCES usuarios (codigo);

-- Reference: planes_por_consulta_consultas_nutricionista (table: planes_por_consulta)
ALTER TABLE planes_por_consulta ADD CONSTRAINT
planes_por_consulta_consultas_nutricionista
    FOREIGN KEY (consultas_nutricionista_codigo)
    REFERENCES consultas_nutricionista (codigo);

-- Reference: planes_por_consulta_planes_nutricionales (table: planes_por_consulta)
ALTER TABLE planes_por_consulta ADD CONSTRAINT
planes_por_consulta_planes_nutricionales
    FOREIGN KEY (planes_nutricionales_codigo)
    REFERENCES planes_nutricionales (codigo);

-- Reference: planes_recomendados_valores_nutricionales (table: planes_nutricionales)
ALTER TABLE planes_nutricionales ADD CONSTRAINT
planes_recomendados_valores_nutricionales
    FOREIGN KEY (valores_nutricionales_codigo)
    REFERENCES valores_nutricionales (codigo);

-- End of file.

```

Inserción de datos para el modelado relacional:  
 INSERT INTO alimentos (codigo, nombre, categoria, calorías, proteínas, carbohidratos,  
 grasas) VALUES

```

(1, 'Palta', 'Frutas', 160.0, 2.0, 9.0, 14.7),
(2, 'Quinoa', 'Granos Integrales', 370.0, 14.0, 64.0, 6.7),
(3, 'Atun', 'Pescados', 137.0, 23.0, 0.0, 5.4),
(4, 'Platano', 'Frutas', 96.0, 1.2, 22.0, 0.2),
(5, 'Fresa', 'Frutas', 32.0, 0.7, 7.7, 0.3),
(6, 'Leche de almendras', 'Lacteos Alternativos', 17.0, 0.4, 0.6, 1.5),
(7, 'Pollo', 'Carnes', 165.0, 31.0, 0.0, 3.6),
(8, 'Espinaca', 'Hortalizas', 23.0, 2.9, 3.6, 0.4),
(9, 'Yogur natural', 'Lacteos', 61.0, 5.4, 7.04, 1.55),
(10, 'Granola', 'Cereales', 471.0, 11.9, 68.6, 17.0),
(11, 'Arandanos', 'Frutas', 57.0, 0.7, 14.49, 0.3),
(12, 'Huevo', 'Otros derivados animales', 143.0, 12.6, 1.1, 9.5),
(13, 'Champañon', 'Setas', 22.0, 3.1, 3.3, 0.3),
(14, 'Sardina', 'Pescados', 140.0, 18.1, 0.0, 7.5),

```

(15, 'Pepino', 'Hortalizas', 11.73, 0.7, 1.9, 0.2),  
 (16, 'Queso de cabra', 'Lacteos alternativos', 360.0, 23.5, 0.8, 28.9),  
 (17, 'Papa', 'Hortalizas', 77.0, 2.0, 17.5, 0.1),  
 (18, 'Coliflor', 'Hortalizas', 11.73, 0.7, 1.9, 0.2),  
 (19, 'Salmon', 'Pescados', 175.0, 20.1, 0.0, 9.9),  
 (20, 'Esparrago', 'Hortalizas', 11.73, 0.7, 1.9, 0.2),  
 (21, 'Jamon', 'Embutidos', 121.0, 18.1, 2.0, 4.0),  
 (22, 'Zanahoria', 'Hortalizas', 41.0, 0.63, 9.6, 0.15),  
 (23, 'Pimiento', 'Hortalizas', 20.0, 0.9, 4.0, 0.2),  
 (24, 'Langostino', 'Mariscos', 85.0, 17.0, 0.0, 1.5),  
 (25, 'Brócoli', 'Hortalizas', 34.0, 2.8, 6.6, 0.4),  
 (26, 'Carne de res', 'Carnes', 250.0, 26.0, 0.0, 17.0),  
 (27, 'Calamar', 'Mariscos', 97.0, 15.6, 3.1, 1.4),  
 (28, 'Queso', 'Lacteos', 356.0, 22.9, 2.8, 28.1);

INSERT INTO valores\_nutricionales (codigo, calorías, proteínas, carbohidratos, grasas)  
VALUES

(1, 345, 9, 41, 18.05),  
 (2, 414.5, 33.5, 48, 10.425),  
 (3, 103.4, 1.705, 20.675, 1.855),  
 (4, 376, 67.8, 7.2, 8),  
 (5, 532.5, 23.55, 84.4, 13.75),  
 (6, 399, 40.1, 19.6, 20.8),  
 (7, 411.46, 44.65, 4.04, 24.07),  
 (8, 138.884, 8.59, 19.02, 3.23),  
 (9, 115.495, 10.42, 3.71, 6.4);

INSERT INTO usuarios (codigo, nombre, apellido, correo, contrasena, fecha\_nacimiento,  
direccion, genero, altura\_inicial, peso\_inicial, objetivos, restricciones) VALUES

(1, 'Bernardo', 'Silva', 'b\_silva123@gmail.com', 'citizen777', '1990-01-01', 'Av. Central 457 -  
Chorrillos', 'Masculino', 170.0, 70.0, 'Quiero obtener mayor masa muscular con el consumo de  
fibra saludable', 'Soy intolerante a la lactosa'),  
 (2, 'Sofía', 'López', 'sofialopez@gmail.com', 'lukita578', '1992-03-15', 'Calle San Sebastian 756  
- Miraflores', 'Femenino', 165.0, 60.0, 'Quiero mejorar mi resistencia física', 'Ninguna'),  
 (3, 'Carlos', 'Gomez', 'c\_gomez123@gmail.com', 'amongusvivamessi', '1985-07-03', 'Av.  
Simon Bolivar 488 - Pueblo libre', 'Masculino', 182.0, 90.0, 'Quiero perder peso y mejorar mi  
salud', 'Soy intolerante a los mariscos'),  
 (4, 'María', 'López', 'maria.lopez@outlook.com', 'elpepetesech', '1998-12-02', 'Calle Principal  
456 – Jesus Maria', 'Femenino', 165.0, 60.0, 'Quiero ganar flexibilidad y mejorar mi postura',  
'Ninguna'),  
 (5, 'Ana', 'García', 'ana.garcia@yahoo.com', 'esotilin123', '1995-08-15', 'Calle del Sol 123 – Los  
Olivos', 'Femenino', 160.0, 55.0, 'Quiero mantener un estilo de vida saludable', 'Ninguna'),

(6, 'Juan', 'Ramirez', 'juan.ramirez@gmail.com', 'coronelleoncioprado777', '1999-11-12', 'Avenida Formacion 987 - Barranco', 'Masculino', 182.0, 89.0, 'Quiero perder peso y mejorar mi condición cardiovascular', 'Soy vegano');

```
INSERT INTO alimentos_no_deseados (alimentos_codigo, usuarios_codigo) VALUES
(22, 1),
(16, 1),
(23, 2),
(24, 3),
(27, 3),
(3, 3),
(25, 4),
(17, 5),
(26, 6),
(7, 6);
```

```
INSERT INTO calificaciones_aplicativo (codigo, usuarios_codigo, calificacion, comentario,
fecha) VALUES
(1, 1, 5, 'Excelente aplicación', '2023-03-04'),
(2, 1, 5, 'Recomendada, me ayudó a aumentar músculo', '2023-11-04'),
(3, 2, 3, 'Buena aplicación pero podrían considerar más variedad de alimentos', '2023-06-04'),
(4, 4, 5, '¡Excelente aplicación!', '2023-09-04'),
(5, 6, 5, 'Excelente aplicación para llevar una alimentación saludable', '2023-09-14');
```

```
INSERT INTO controles (codigo, usuarios_codigo, fecha, peso, talla) VALUES
(1, 2, '2023-05-20', 60, 1.65),
(2, 2, '2023-06-18', 57.5, 1.65),
(3, 5, '2023-06-12', 55, 1.6),
(4, 6, '2023-06-04', 87, 1.82),
(5, 6, '2023-09-04', 81, 1.84);
```

```
INSERT INTO planes_nutricionales (codigo, descripcion, nombre, precio,
valores_nutricionales_codigo)
VALUES
```

```
(1, 'La ensalada de quinua y palta es una opción deliciosa y nutritiva que combina la textura suave y cremosa de la palta con el grano rico en proteínas y fibra de la quinua. La ensalada de quinua y palta puede servirse como plato principal para una comida ligera o como acompañamiento de otros platos. Combina bien con proteínas a la parrilla, pescado, pollo, o como guarnición en barbacoas y comidas al aire libre.', 'Ensalada de quinua y palta', 14, 1),
(2, 'El bowl de atún y quinua es una opción saludable y equilibrada que combina los beneficios del atún y la quinua. El atún es una excelente fuente de proteínas magras y ácidos grasos omega-3, que son beneficiosos para la salud del corazón y el sistema nervioso. La quinua, por su parte, es un grano rico en proteínas de alta calidad, fibra y varios nutrientes esenciales como hierro, magnesio y vitaminas del complejo B. Puedes añadir vegetales frescos y aderezos saludables
```

para completar tu bowl. Es ideal como almuerzo o cena nutritiva.', 'Bowl de atun y quinua', 15, 2),

(3, 'El batido de frutas es una opción refrescante y nutritiva para disfrutar como desayuno o merienda. Combina plátano, fresas y leche de almendras para obtener una mezcla equilibrada de carbohidratos, proteínas y grasas saludables. Puedes agregar hielo y endulzantes naturales si lo deseas. ¡Ideal para empezar el día con energía!', 'Batido de frutas', 10, 3),

(4, 'La ensalada de pollo y espinacas es una opción rica en proteínas y nutrientes. Combina pechuga de pollo a la parrilla con espinacas frescas, ofreciendo una mezcla de sabores y texturas. Puedes agregar ingredientes adicionales como tomate, pepino y aderezo ligero. Perfecta como plato principal para una comida completa y saludable.', 'Ensalada de pollo y espinacas', 28, 4),

(5, 'El tazón de yogur y granola es una opción deliciosa y nutritiva para el desayuno o la merienda. Combina yogur natural cremoso con granola crujiente y arándanos frescos. Este tazón es rico en proteínas, fibra y antioxidantes. Puedes agregar miel o nueces para darle un toque extra. ¡Una opción saludable y satisfactoria!', 'Tazon de yogur y granola', 12.5, 5),

(6, 'La tortilla de espinacas y champiñones es una opción saludable y sabrosa para cualquier comida. Combina huevos batidos con espinacas y champiñones salteados para obtener una tortilla rica en proteínas y vegetales. Puedes añadir especias o condimentos de tu elección para darle sabor. ¡Perfecta para un desayuno o una cena ligera!', 'Tortilla de espinacas y champiñones', 18, 6),

(7, 'El Carpaccio de pepino con sardinas es una deliciosa y nutritiva opción que combina la frescura del pepino con el sabor único de las sardinas. El pepino, rico en agua y bajo en calorías, aporta hidratación y fibra, mientras que las sardinas son una excelente fuente de proteínas, ácidos grasos omega-3 y diversos micronutrientes. Puedes disfrutar esta preparación como entrada o como plato principal, acompañada de ensaladas frescas o pan crujiente.', 'Carpaccio de pepino con sardinas', 14.5, 7),

(8, 'Las papas son una fuente de carbohidratos y proporcionan energía, mientras que la coliflor agrega nutrientes adicionales y una textura suave al pure. El salmón, por su parte, aporta proteínas de alta calidad, ácidos grasos omega-3 y diversos nutrientes esenciales. Este plato ofrece beneficios para la salud, ya que el salmón es conocido por su contenido de ácidos grasos omega-3, que tienen propiedades antiinflamatorias y beneficiosas para la salud cardiovascular. La coliflor también es rica en vitamina C, vitamina K y fibra, lo que promueve la salud general y el funcionamiento adecuado del sistema digestivo.', 'Pure de coliflor', 11, 8),

(9, 'Los huevos son una excelente fuente de proteínas, que son fundamentales para la reparación y construcción de tejidos en nuestro cuerpo. Además, contienen importantes nutrientes como la vitamina B12, que es esencial para el funcionamiento adecuado del sistema nervioso. Los espárragos verdes son bajos en calorías y ricos en fibra, lo que los convierte en una excelente opción para una alimentación equilibrada. También son una fuente de antioxidantes y contienen vitaminas y minerales como la vitamina K, vitamina C y folato. La combinación de huevos y espárragos verdes crea un plato sabroso y nutritivo. Los huevos aportan proteínas y grasas saludables, mientras que los espárragos agregan fibra, vitaminas y minerales.', 'Huevos revueltos con espárragos verdes', 13.5, 9);

```

INSERT INTO alimentos_en_plan (planes_nutricionales_codigo, alimentos_codigo,
cantidad_gramos)
VALUES
(1, 1, 100),
(1, 2, 50),
(2, 3, 100),
(2, 2, 75),
(3, 4, 65),
(3, 5, 75),
(3, 6, 100),
(4, 7, 200),
(4, 8, 200),
(5, 9, 300),
(5, 10, 50),
(5, 11, 200),
(6, 12, 200),
(6, 8, 300),
(6, 13, 200),
(7, 14, 200),
(7, 15, 200),
(7, 16, 30),
(8, 17, 100),
(8, 18, 80),
(8, 19, 30),
(9, 12, 60),
(9, 20, 150),
(9, 21, 10);

```

```

INSERT INTO nutricionistas (codigo, apellido, correo, especialidad, nombre, puntuacion)
VALUES
(1, 'Andrade', 'sergi_a11@gmail.com', 'Nutricionista deportivo', 'Sergio', 5),
(2, 'Gómez', 'laura_gomez@gmail.com', 'Nutricionista clínico', 'Laura',
4.333333333333333),
(3, 'Rojas', 'carlos_rojas@gmail.com', 'Nutricionista clínico', 'Carlos', 3),
(4, 'López', 'maria_lopez@gmail.com', 'Nutricionista en alimentación vegetariana', 'María',
4),
(5, 'Torres', 'javier_torres@gmail.com', 'Nutricionista clínico', 'Javier', 4),
(6, 'Martínez', 'ana_martinez@gmail.com', 'Nutricionista deportivo', 'Ana', 0);

```

```

INSERT INTO calificaciones_nutricionista (nutricionistas_codigo, usuarios_codigo,
calificacion)
VALUES
(1, 1, 5),
(2, 2, 4),

```

(2, 3, 5),  
 (2, 4, 4),  
 (3, 5, 3),  
 (4, 2, 5),  
 (4, 4, 3),  
 (5, 3, 5),  
 (5, 5, 4),  
 (5, 6, 3);

INSERT INTO consultas\_nutricionista (codigo,nutricionistas\_codigo, usuarios\_codigo, fecha, comentario)

VALUES

(1, 1, 1, '2023-06-04', 'Como veo que deseas ganar masa muscular, te recomendaría incluir la palta en tu dieta porque la palta es rica en grasas monoinsaturadas, que son consideradas grasas saludables. Estas grasas son importantes para la salud del corazón y también ayudan a reducir la inflamación en el cuerpo, lo cual es beneficioso para el proceso de recuperación muscular después del entrenamiento.'),

(2, 1, 4, '2023-06-14', 'Si deseas tener un cuerpo más flexible, te recomendaría incorporar el consumo de frutas en tu dieta diaria. Las frutas son una excelente fuente de vitaminas, minerales y antioxidantes que ayudan a fortalecer tus músculos y mejorar la flexibilidad de tu cuerpo. Además, su contenido de agua y fibra te mantendrá hidratado y contribuirá a una mejor digestión.'),

(3, 2, 2, '2023-08-12', 'Si estás buscando mejorar tu resistencia física, te recomendaría incluir en tu dieta granos integrales y consumir pollo de manera saludable. Los granos integrales son excelentes fuentes de carbohidratos complejos. Estos carbohidratos se descomponen lentamente en tu organismo, proporcionándote energía de forma constante y sostenida. En cuanto al pollo, es una excelente fuente de proteínas magras. Las proteínas son fundamentales para la reparación y construcción de tejidos musculares, lo que favorece la recuperación y desarrollo de tu resistencia física.'),

(4, 3, 3, '2023-06-25', 'Si tienes como objetivo perder peso, te recomendaría adoptar una alimentación e incluir una amplia variedad de verduras en tu dieta diaria. Las verduras son bajas en calorías y ricas en nutrientes esenciales como vitaminas, minerales y fibra. Su alto contenido de fibra te ayudará a sentirte más lleno/a por más tiempo, lo que puede contribuir a reducir la ingesta de calorías. Además de las verduras, te sugiero limitar el consumo de alimentos procesados, altos en grasas saturadas, azúcares y sodio. Opta por fuentes de proteína magra como pollo, pescado, tofu o legumbres, que te ayudarán a sentirte saciado/a y a mantener la masa muscular mientras pierdes peso.'),

(5, 4, 6, '2023-07-25', 'Si buscas bajar de peso y mejorar tu condición cardiovascular, te recomiendo incluir alimentos saludables y equilibrados en tu dieta. Opta por una variedad de frutas y verduras frescas, que son ricas en vitaminas, minerales y antioxidantes. También elige granos integrales, como arroz integral y quinoa, que te proporcionarán energía y fibra. Incorpora fuentes de proteína magra, como legumbres, tofu y yogur bajo en grasa. Limita el consumo de alimentos procesados, altos en azúcares y grasas saturadas.'),

(6, 5, 5, '2023-08-30', 'Si estás buscando llevar una vida saludable en general, incluir verduras verdes en tu dieta es una excelente elección. Las verduras verdes, como espinacas, brócoli, kale y acelgas, son una fuente nutritiva y deliciosa de beneficios para tu salud. Estas verduras están cargadas de vitaminas, minerales y antioxidantes que ayudan a fortalecer tu sistema inmunológico, mantener una piel radiante y promover un funcionamiento óptimo del cuerpo. Su alto contenido de fibra también contribuye a la salud digestiva y puede ayudar a mantener un peso saludable.');

```
INSERT INTO planes_por_consulta (consultas_nutricionista_codigo,
planes_nutricionales_codigo)
VALUES
(1, 1),
(1, 2),
(2, 3),
(2, 5),
(3, 2),
(3, 4),
(4, 4),
(4, 6),
(5, 1),
(5, 4),
(6, 6),
(6, 7),
(6, 9);
```

```
INSERT INTO boleta (codigo, usuarios_codigo, fecha, delivery, recargo_delivery,
lugar_recogida, monto_total) VALUES
(1, 1, '2023-01-01', 1, 8.5, null, 37.5),
(2, 2, '2023-02-02', 0, null, 'Local 1 - Av. Carmen del Sol 234 Callao', 38),
(3, 3, '2023-03-03', 1, 6.5, null, 19),
(4, 4, '2023-04-04', 0, null, 'Local 2 - Av. Caminos del Sol 987 Chorrillos', 43.5),
(5, 5, '2023-04-05', 1, 3.8, null, 17.3),
(6, 6, '2023-04-06', 0, null, 'Local 3 - Av. Lineas de Nazca 1444 Rimac', 24),
(7, 1, '2023-07-07', 1, 9.5, null, 37.5),
(8, 2, '2023-02-08', 0, null, 'Local 1 - Av. Carmen del Sol 234 Callao', 33),
(9, 3, '2023-02-09', 1, 7.0, null, 31.5),
(10, 4, '2023-10-10', 1, 3.5, null, 26)
```

```
INSERT INTO detalles_boleta (planes_nutricionales_codigo, boleta_codigo, cantidad)
VALUES
(1, 1, 3),
(2, 1, 2),
(3, 2, 1),
(4, 2, 2),
```

(5, 3, 1),  
 (6, 4, 1),  
 (7, 4, 2),  
 (8, 4, 1),  
 (9, 5, 2),  
 (1, 6, 1),  
 (3, 6, 1),  
 (4, 7, 3),  
 (2, 8, 2),  
 (6, 8, 1),  
 (8, 9, 1),  
 (9, 9, 1),  
 (3, 10, 1),  
 (5, 10, 1)

#### **4.3. NonRelational Database.**

Según Ayudaley, una base de datos no relacional, también conocida como NoSQL, es un sistema de almacenamiento de información que no utiliza el lenguaje SQL para consultas. A diferencia de las bases de datos relacionales, las bases de datos no relacionales no almacenan los datos en tablas ni los organizan en registros. Además, se destacan por su escalabilidad y capacidad para gestionar grandes volúmenes de datos.

##### **Existen diferentes tipos de bases de datos no relacionales, entre ellos:**

**Clave-valor:** Almacenan la información en pares de clave-valor, donde cada clave es un identificador único y se le asigna un valor.

**Documentos:** Representan la información como documentos JSON. Estos documentos son flexibles, semiestructurados y jerárquicos, lo que facilita su almacenamiento, gestión y consulta.

**Gráficos:** Diseñados para crear y navegar relaciones entre entidades. Las entidades se almacenan como nodos y las relaciones se establecen mediante bordes.

**En memoria:** Diseñados para ofrecer respuestas rápidas en milisegundos.

##### **Las características principales de una base de datos no relacional son las siguientes:**

**Almacenamiento de datos en documentos:** En lugar de utilizar tablas, los datos se almacenan en documentos. Cada documento puede contener información estructurada o semiestructurada.



**Gestión de información no estructurada:** Las bases de datos no relacionales son especialmente útiles para organizar y gestionar información no estructurada, como documentos, texto libre o datos de tipo multimedia.

**Consultas sin lenguaje SQL:** A diferencia de las bases de datos relacionales, las bases de datos no relacionales no utilizan el lenguaje SQL como estándar para realizar consultas. Sin embargo, algunas bases de datos no relacionales pueden ofrecer soporte para consultas SQL como una herramienta adicional.

**Ausencia de garantía ACID:** Las bases de datos no relacionales no garantizan cumplir con las cualidades ACID (Atomicidad, Consistencia, Integridad y Durabilidad) presentes en las bases de datos relacionales. Esto implica que pueden ofrecer una flexibilidad mayor en ciertos escenarios, pero a costa de una menor rigurosidad en términos de garantías transaccionales.

**Escalabilidad:** Las bases de datos no relacionales están diseñadas para manejar grandes volúmenes de datos y ofrecer una alta escalabilidad. Pueden adaptarse y crecer de forma eficiente a medida que aumenta la cantidad de información almacenada.

### **Nuestra elección como NoSQL: MongoDB.**

Hemos decidido elegir MongoDB como nuestra opción de base de datos NoSQL. Aunque a menudo pensamos en las bases de datos relacionales y SQL, existen alternativas como las bases de datos no relacionales, y MongoDB es uno de los ejemplos más destacados en este campo.

MongoDB es un sistema de base de datos NoSQL orientado a documentos. Está escrito en C++ y es de código abierto. A diferencia de las bases de datos relacionales que almacenan datos en tablas, MongoDB utiliza estructuras de datos BSON, similares a JSON, para almacenar la información. Una base de datos de MongoDB puede contener una o más colecciones de datos y administrar diversos documentos con múltiples campos de datos.

Una de las ventajas de MongoDB es que es de código abierto, lo que significa que sus binarios están disponibles para sistemas operativos como Windows y Linux. Además, MongoDB ha sido adoptado por varias empresas reconocidas, como MTV Network, Foursquare y Craigslist, lo que demuestra su amplio uso en múltiples implementaciones.

### **Comparación de dos modelos de base de datos.**

MySQL y MongoDB son dos modelos de bases de datos con enfoques diferentes en cuanto al almacenamiento de datos. Aunque presentan similitudes en algunas estructuras básicas, también tienen diferencias fundamentales.

En MySQL, los datos se almacenan en tablas, mientras que en MongoDB se almacenan en colecciones, que podríamos considerar como "carpetas" que contienen documentos. En lugar

de utilizar filas y columnas como en las tablas de MySQL, MongoDB utiliza documentos en formato BSON, que definen campos similares a las columnas de las tablas SQL.

En MySQL, un registro correspondería a una fila en una tabla, mientras que, en MongoDB, un documento es una colección de pares clave-valor, similar a una fila en una tabla de MySQL. Estas similitudes en la estructura básica permiten cierta familiaridad entre ambos modelos.

Sin embargo, una diferencia fundamental radica en el enfoque de recuperación de datos. Como base de datos NoSQL, MongoDB no utiliza SQL como lenguaje de consulta. En su lugar, MongoDB dispone de su propio lenguaje de consulta y utiliza métodos específicos del lenguaje de programación del cliente para el procesamiento de los datos. Esto implica que la forma de interactuar con los datos y realizar consultas difiere significativamente entre MySQL y MongoDB.

### **¿Para qué sirve el MongoDB?**

MongoDB es versátil y se utiliza en una variedad de casos de uso, como la validación de documentos, la gestión de contenido, las tiendas online y el procesamiento de datos en tiempo real. Proporciona un almacenamiento eficiente, una escalabilidad horizontal y un rendimiento rápido, lo que lo convierte en una opción popular para aplicaciones que manejan grandes volúmenes de datos y requieren flexibilidad en su estructura.

### **¿Por qué optamos por usar el MongoDB?**

Hemos elegido MongoDB como nuestra opción de base de datos por varias razones:

**Velocidad y flexibilidad:** MongoDB ofrece un rendimiento rápido y la capacidad de agregar registros con diferentes campos en una misma colección de manera flexible. Esto es especialmente beneficioso para nuestra startup, ya que nos permite adaptarnos y evolucionar rápidamente a medida que crecemos y cambiamos nuestros requisitos de datos.

**Escalabilidad:** MongoDB es altamente escalable, lo que significa que puede crecer y adaptarse a medida que nuestra base de datos y la cantidad de información que almacenamos aumenten. Esto nos brinda la tranquilidad de que podemos manejar un mayor volumen de datos sin comprometer el rendimiento.

**Alta disponibilidad:** MongoDB está diseñado para ofrecer una alta disponibilidad, lo que significa que puede manejar un gran volumen de acceso de aplicaciones y usuarios. A medida que nuestra startup crezca y tengamos más usuarios interactuando con nuestra plataforma, podemos confiar en que MongoDB seguirá brindando un rendimiento sólido y consistente.

**Soporte de transacciones:** A diferencia de algunas bases de datos NoSQL, MongoDB ha introducido soporte para transacciones a partir de la versión 4.2. Esto nos permite realizar

operaciones atómicas y garantizar la consistencia de los datos, lo que era una limitación en versiones anteriores.

**Integración con proyectos de Big Data:** MongoDB se integra bien en proyectos de Big Data, ya que puede manejar datos estructurados de manera flexible. Esto es especialmente valioso, ya que nos permite procesar y analizar grandes volúmenes de datos complejos que las grandes empresas a menudo tienen que manejar.

**Documentación oficial completa:** MongoDB cuenta con una documentación oficial muy completa y de calidad. Esto nos brinda un recurso confiable para mejorar nuestras operaciones y aprovechar al máximo todas las características y funcionalidades que MongoDB ofrece.

**Código abierto:** MongoDB es un sistema de base de datos de código abierto, lo que significa que podemos acceder y utilizar su código fuente de forma gratuita. Esto nos brinda la libertad de personalizar y adaptar MongoDB según nuestras necesidades específicas, así como aprovechar las contribuciones y mejoras de la comunidad de desarrolladores.

### **Inserción de datos para el modelado no relacional**

```
db.usuarios.insertMany([
{
  _id: "id_usuario_1",
  nombre: "Bernardo",
  apellido: "Silva",
  correo: "b_silva123@gmail.com",
  contraseña: "citizen777",
  fecha_nacimiento: ISODate("1990-01-01"),
  direccion: "Av. Central 457 - Chorrillos",
  genero: "Masculino",
  altura_inicial: 170.0,
  peso_inicial: 70.0,
  objetivos: "Quiero obtener mayor masa muscular con el consumo de fibra saludable",
  restricciones: "Soy intolerante a la lactosa",
  alimentos_no_deseados: ["Zanahoria", "Queso"],
  controles: [
    {fecha: ISODate("2023-06-04"), peso: 70.0, talla: 1.70}
  ],
  calificaciones_app: [
    {calificacion: 5, comentario: "Excelente aplicación", fecha: ISODate("2023-03-04")},
    {calificacion: 5, comentario: "Recomendada, me ayudo a aumentar musculo", fecha:
ISODate("2023-11-04")}
  ]
},
{

```

```

_id: "id_usuario_2",
nombre: "Sofía",
apellido: "López",
correo: "sofialopez@gmail.com",
contrasena: "lukita578",
fecha_nacimiento: ISODate("1992-03-15"),
direccion: "Calle San Sebastian 756 - Miraflores",
genero: "Femenino",
altura_inicial: 165.0,
peso_inicial: 60.0,
objetivos: "Quiero mejorar mi resistencia física",
restricciones: "Ninguna",
alimentos_no_deseados: ["Pimiento"],
controles: [
{fecha: ISODate("2023-05-20"), peso: 60.0, talla: 1.65 },
{fecha: ISODate("2023-06-18"), peso: 59.5, talla: 1.65 }
],
calificaciones_app: [
{calificacion: 3, comentario: "Buena aplicación pero podrían considerar mas variedad de alimentos", fecha: ISODate("2023-06-04")}
],

```

```

{
_id: "id_usuario_3",
nombre: "Carlos",
apellido: "Gomez",
correo: "c_gomez123@gmail.com",
contrasena: "amongusvivamessi",
fecha_nacimiento: ISODate("1985-07-03"),
direccion: "Av. Simon Bolivar 488 - Pueblo libre",
genero: "Masculino",
altura_inicial: 182.0,
peso_inicial: 90.0,
objetivos: "Quiero perder peso y mejorar mi salud",
restricciones: "Soy intolerante a los mariscos",
alimentos_no_deseados: ["Langostino", "Calamar", "Atun"]
},

```

```

{
_id: "id_usuario_4",
nombre: "María",

```

```

apellido: "López",
correo: "maria.lopez@outlook.com",
contrasena: "elpepetesech",
fecha_nacimiento: ISODate("1998-12-02"),
direccion: "Calle Principal 456 – Jesus Maria",
genero: "Femenino",
altura_inicial: 165.0,
peso_inicial: 60.0,
objetivos: "Quiero ganar flexibilidad y mejorar mi postura",
restricciones: "Ninguna",
alimentos_no_deseados: ["Brócoli"],
calificaciones_app: [
{ calificacion: 5, comentario: "¡Excelente aplicación!", fecha: ISODate("2023-09-04") }
],

```

```

{
_id: "id_usuario_5",
nombre: "Ana",
apellido: "García",
correo: "ana.garcia@yahoo.com",
contrasena: "esotilin123",
fecha_nacimiento: ISODate("1995-08-15"),
direccion: "Calle del Sol 123 – Los Olivos",
genero: "Femenino",
altura_inicial: 160.0,
peso_inicial: 55.0,
objetivos: "Quiero mantener un estilo de vida saludable",
restricciones: "Ninguna",
alimentos_no_deseados: ["Papa"],
controles: [
{ fecha: ISODate("2023-06-12"), peso: 55.0, talla: 1.60 }
],

```

```

{
_id: "id_usuario_6",
nombre: "Juan",
apellido: "Ramirez",
correo: "juan.ramirez@gmail.com",
contrasena: "coronelleoncioprado777",
fecha_nacimiento: ISODate("1999-11-12"),
direccion: "Avenida Formacion 987 - Barranco",
genero: "Masculino",

```

```

altura_inicial: 182.0,
peso_inicial: 89.0,
objetivos: "Quiero perder peso y mejorar mi condición cardiovascular",
restricciones: "Soy vegano",
alimentos_no_deseados: ["Carne de res", "Pollo"],
calificaciones_app: [
{ calificacion: 5, comentario: "Excelente aplicación para llevar una alimentación saludable", fecha: ISODate("2023-09-14") }
]
]

```

```

db.planes_nutricionales.insertMany([
{
_id: "id_plan_1",
nombre: "Ensalada de quinua y palta",
alimentos: [
{nombre: "Palta", categoria: "Frutas", calorías: 160.0, proteínas: 2.0, carbohidratos: 9.0, grasas: 14.7, cantidad: 100},
{nombre: "Quinua", categoria: "Granos Integrales", calorías: 370.0, proteínas: 14.0, carbohidratos: 64.0, grasas: 6.7, cantidad: 50}],
descripcion: "La ensalada de quinua y palta es una opción deliciosa y nutritiva que combina la textura suave y cremosa de la palta con el grano rico en proteínas y fibra de la quinua. La ensalada de quinua y palta puede servirse como plato principal para una comida ligera o como acompañamiento de otros platos. Combina bien con proteínas a la parrilla, pescado, pollo, o como guarnición en barbacoas y comidas al aire libre.",
precio: 14.0
},
{
_id: "id_plan_2",
nombre: "Bowl de atun y quinua",
alimentos: [
{nombre: "Atun", categoria: "Pescados", calorías: 137.0, proteínas: 23.0, carbohidratos: 0.0, grasas: 5.4, cantidad: 100},
{nombre: "Quinua", categoria: "Granos Integrales", calorías: 370.0, proteínas: 14.0, carbohidratos: 64.0, grasas: 6.7, cantidad: 75}
],
descripcion: "El bowl de atún y quinua es una opción saludable y equilibrada que combina los beneficios del atún y la quinua. El atún es una excelente fuente de proteínas magras y ácidos grasos omega-3, que son beneficiosos para la salud del corazón y el sistema nervioso. La quinua, por su parte, es un grano rico en proteínas de alta calidad, fibra y varios nutrientes esenciales como hierro, magnesio y vitaminas del complejo B. Puedes añadir vegetales frescos y aderezos saludables para completar tu bowl. Es ideal como almuerzo o cena nutritiva.",
precio: 15.0
}
]

```

```

},
{
  _id: "id_plan_3",
  nombre: "Batido de frutas",
  alimentos: [
    {nombre: "Platano", categoria: "Frutas", calorías: 96.0, proteínas: 1.2, carbohidratos: 22.0, grasas: 0.2, cantidad: 65},
    {nombre: "Fresas", categoria: "Frutas", calorías: 32.0, proteínas: 0.7, carbohidratos: 7.7, grasas: 0.3, cantidad: 75},
    {nombre: "Leche de almendras", categoria: "Lacteos Alternativos", calorías: 17.0, proteínas: 0.4, carbohidratos: 0.6, grasas: 1.5, cantidad: 100}
  ],
  descripcion: "El batido de frutas es una opción refrescante y nutritiva para disfrutar como desayuno o merienda. Combina plátano, fresas y leche de almendras para obtener una mezcla equilibrada de carbohidratos, proteínas y grasas saludables. Puedes agregar hielo y endulzantes naturales si lo deseas. ¡Ideal para empezar el día con energía!",
  precio: 10.0
},
{
  _id: "id_plan_4",
  nombre: "Ensalada de pollo y espinacas",
  alimentos: [
    { nombre: "Pollo", categoria: "Carnes", calorías: 165.0, proteínas: 31.0, carbohidratos: 0.0, grasas: 3.6, cantidad: 200 },
    { nombre: "Espinacas", categoria: "Hortalizas", calorías: 23.0, proteínas: 2.9, carbohidratos: 3.6, grasas: 0.4, cantidad: 200 }
  ],
  descripcion: "La ensalada de pollo y espinacas es una opción rica en proteínas y nutrientes. Combina pechuga de pollo a la parrilla con espinacas frescas, ofreciendo una mezcla de sabores y texturas. Puedes agregar ingredientes adicionales como tomate, pepino y aderezo ligero. Perfecta como plato principal para una comida completa y saludable.",
  precio: 28.0
},
{
  _id: "id_plan_5",
  nombre: "Tazon de yogur y granola",
  alimentos: [
    {nombre: "Yogur natural", categoria: "Lacteos", calorías: 61.0, proteínas: 5.4, carbohidratos: 7.04, grasas: 1.55, cantidad: 300},
    {nombre: "Granola", categoria: "Cereales", calorías: 471.0, proteínas: 11.9, carbohidratos: 68.6, grasas: 17.0, cantidad: 50},
    { nombre: "Arandanos", categoria: "Frutas", calorías: 57.0, proteínas: 0.7, carbohidratos: 14.49, grasas: 0.3, cantidad: 200 }
  ]
}

```

],  
 descripcion: "El tazón de yogur y granola es una opción deliciosa y nutritiva para el desayuno o la merienda. Combina yogur natural cremoso con granola crujiente y arándanos frescos. Este tazón es rico en proteínas, fibra y antioxidantes. Puedes agregar miel o nueces para darle un toque extra. ¡Una opción saludable y satisfactoria!",  
 precio: 12.5  
 },  
 {  
 \_id: "id\_plan\_6",  
 nombre: "Tortilla de espinacas y champiñones",  
 alimentos: [  
 {nombre: "Huevo", categoria: "Otros derivados animales", calorías: 143.0, proteínas: 12.6, carbohidratos: 1.1, grasas: 9.5, cantidad: 200},  
 {nombre: "Espinacas", categoria: "Hortalizas", calorías: 23.0, proteínas: 2.9, carbohidratos: 3.6, grasas: 0.4, cantidad: 300},  
 {nombre: "Champiñones", categoria: "Setas", calorías: 22.0, proteínas: 3.1, carbohidratos: 3.3, grasas: 0.3, cantidad: 200}  
 ],  
 descripcion: "La tortilla de espinacas y champiñones es una opción saludable y sabrosa para cualquier comida. Combina huevos batidos con espinacas y champiñones salteados para obtener una tortilla rica en proteínas y vegetales. Puedes añadir especias o condimentos de tu elección para darle sabor. ¡Perfecta para un desayuno o una cena ligera!",  
 precio: 18.0  
 },  
 {  
 \_id: "id\_plan\_7",  
 nombre: "Carpaccio de pepino con sardinas",  
 alimentos: [  
 {nombre: "Sardina", categoria: "Pescados", calorías: 140.0, proteínas: 18.1, carbohidratos: 0.0, grasas: 7.5, cantidad: 200},  
 {nombre: "Pepino", categoria: "Hortalizas", calorías: 11.73, proteínas: 0.7, carbohidratos: 1.9, grasas: 0.2, cantidad: 200},  
 {nombre: "Queso de cabra", categoria: "Lacteos alternativos", calorías: 360.0, proteínas: 23.5, carbohidratos: 0.8, grasas: 28.9, cantidad: 30}  
 ],  
 descripcion: " El Carpaccio de pepino con sardinas es una deliciosa y nutritiva opción que combina la frescura del pepino con el sabor único de las sardinas. El pepino, rico en agua y bajo en calorías, aporta hidratación y fibra, mientras que las sardinas son una excelente fuente de proteínas, ácidos grasos omega-3 y diversos micronutrientes. Puedes disfrutar esta preparación como entrada o como plato principal, acompañada de ensaladas frescas o pan crujiente.",  
 precio: 14.5  
 },



```

{
  _id: "id_plan_8",
  nombre: "Pure de coliflor",
  alimentos: [
    {nombre: "Papa", categoria: "Hortalizas", calorías: 77.0, proteínas: 2.0, carbohidratos: 17.5, grasas: 0.1, cantidad: 100},
    {nombre: "Coliflor", categoria: "Hortalizas", calorías: 11.73, proteínas: 0.7, carbohidratos: 1.9, grasas: 0.2, cantidad: 80},
    {nombre: "Salmon", categoria: "Pescados", calorías: 175.0, proteínas: 20.1, carbohidratos: 0.0, grasas: 9.9, cantidad: 30}
  ],
  descripcion: "Las papas son una fuente de carbohidratos y proporcionan energía, mientras que la coliflor agrega nutrientes adicionales y una textura suave al pure. El salmón, por su parte, aporta proteínas de alta calidad, ácidos grasos omega-3 y diversos nutrientes esenciales. Este plato ofrece beneficios para la salud, ya que el salmón es conocido por su contenido de ácidos grasos omega-3, que tienen propiedades antiinflamatorias y beneficiosas para la salud cardiovascular. La coliflor también es rica en vitamina C, vitamina K y fibra, lo que promueve la salud general y el funcionamiento adecuado del sistema digestivo.",
  precio: 11.0
},
{
  _id: "id_plan_9",
  nombre: "Huevos revueltos con espárragos verdes",
  alimentos: [
    {nombre: "Huevo", categoria: "Otros derivados animales", calorías: 143.0, proteínas: 12.6, carbohidratos: 1.1, grasas: 9.5, cantidad: 60},
    {nombre: "Espárragos", categoria: "Hortalizas", calorías: 11.73, proteínas: 0.7, carbohidratos: 1.9, grasas: 0.2, cantidad: 150},
    {nombre: "Jamon", categoria: "Embutidos", calorías: 121.0, proteínas: 18.1, carbohidratos: 2.0, grasas: 4.0, cantidad: 10}
  ],
  descripcion: "Los huevos son una excelente fuente de proteínas, que son fundamentales para la reparación y construcción de tejidos en nuestro cuerpo. Además, contienen importantes nutrientes como la vitamina B12, que es esencial para el funcionamiento adecuado del sistema nervioso. Los espárragos verdes son bajos en calorías y ricos en fibra, lo que los convierte en una excelente opción para una alimentación equilibrada. También son una fuente de antioxidantes y contienen vitaminas y minerales como la vitamina K, vitamina C y folato. La combinación de huevos y espárragos verdes crea un plato sabroso y nutritivo. Los huevos aportan proteínas y grasas saludables, mientras que los espárragos agregan fibra, vitaminas y minerales.",
  precio: 13.5
},
]

```

```

db.compras_plan.insertMany([
  {
    _id: "id_compras_plan_1",
    id_usuario: "id_usuario_1",
    detalles_compra: [{id_plan: "id_plan_1", cantidad: 3}, {id_plan: "id_plan_2", cantidad:
2}],
    fecha: ISODate("2023-01-01"),
    delivery: true,
    recargo_delivery: 8.5,
    monto_total: 0.0,
  },
  {
    _id: "id_compras_plan_2",
    id_usuario: "id_usuario_2",
    detalles_compra: [{id_plan: "id_plan_3", cantidad: 1}, {id_plan: "id_plan_4", cantidad:
2}],
    fecha: ISODate("2023-02-02"),
    delivery: false,
    monto_total: 0.0,
    lugar_recogida: "Local 1 - Av. Carmen del Sol 234 Callao"
  },
  {
    _id: "id_compras_plan_3",
    id_usuario: "id_usuario_3",
    detalles_compra: [{id_plan: "id_plan_5", cantidad: 1}],
    fecha: ISODate("2023-03-03"),
    delivery: true,
    recargo_delivery: 6.5,
    monto_total: 0.0,
  },
  {
    _id: "id_compras_plan_4",
    id_usuario: "id_usuario_4",
    detalles_compra: [{id_plan: "id_plan_6", cantidad: 1}, {id_plan: "id_plan_7", cantidad:
2}, {id_plan: "id_plan_8", cantidad: 1}],
    fecha: ISODate("2023-04-04"),
    delivery: false,
    monto_total: 0.0,
    lugar_recogida: "Local 2 - Av. Caminos del Sol 987 Chorrillos"
  },
  {
    _id: "id_compras_plan_5",
    id_usuario: "id_usuario_5",

```

```

detalles_compra: [{id_plan: "id_plan_9", cantidad: 2}],
fecha: ISODate("2023-04-05"),
delivery: true,
recargo_delivery: 3.8,
monto_total: 0.0,
},
{
  _id: "id_compras_plan_6",
  id_usuario: "id_usuario_6",
  detalles_compra: [{id_plan: "id_plan_1", cantidad: 1}, {id_plan: "id_plan_3", cantidad:
1}],
  fecha: ISODate("2023-04-06"),
  delivery: false,
  monto_total: 0.0,
  lugar_recogida: "Local 3 - Av. Lineas de Nazca 1444 Rimac"
},
{
  _id: "id_compras_plan_7",
  id_usuario: "id_usuario_1",
  detalles_compra: [{id_plan: "id_plan_4", cantidad: 3}],
  fecha: ISODate("2023-07-07"),
  delivery: true,
  recargo_delivery: 9.5,
  monto_total: 0.0,
},
{
  _id: "id_compras_plan_8",
  id_usuario: "id_usuario_2",
  detalles_compra: [{id_plan: "id_plan_2", cantidad: 2}, {id_plan: "id_plan_6", cantidad:
1}],
  fecha: ISODate("2023-02-08"),
  delivery: false,
  monto_total: 0.0,
  lugar_recogida: "Local 1 - Av. Carmen del Sol 234 Callao"
},
{
  _id: "id_compras_plan_9",
  id_usuario: "id_usuario_3",
  detalles_compra: [{id_plan: "id_plan_8", cantidad: 1}, {id_plan: "id_plan_9", cantidad:
1}],
  fecha: ISODate("2023-02-09"),
  delivery: true,
  recargo_delivery: 7.0,
  monto_total: 0.0,

```

```

},
{
  _id: "id_compras_plan_10",
  id_usuario: "id_usuario_4",
  detalles_compra: [{id_plan: "id_plan_3", cantidad: 1}, {id_plan: "id_plan_5", cantidad: 1}],
  fecha: ISODate("2023-10-10"),
  delivery: true,
  recargo_delivery: 3.5,
  monto_total: 0.0,
}
])

```

```

db.nutricionistas.insertMany([
{
  _id: "id_nutricionista_1",
  nombre: "Sergio",
  apellido: "Andrade",
  especialidad: "Nutricionista deportivo",
  correo: "sergi_a11@gmail.com",
  puntuaciones_usuarios: [{usuarios_codigo: "id_usuario_1", calificacion: 5}],
  puntuacion: 0.0,
  consultas: [{usuarios_codigo: "id_usuario_1", fecha: ISODate("2023-06-04")},
  comentarios: "Como veo que deseas ganar masa muscular, te recomendaría incluir la
  palta en tu dieta porque la palta es rica en grasas monoinsaturadas, que son consideradas
  grasas saludables. Estas grasas son importantes para la salud del corazón y también
  ayudan a reducir la inflamación en el cuerpo, lo cual es beneficioso para el proceso de
  recuperación muscular después del entrenamiento.", planes_recomendados:
  ["id_plan_1", "id_plan_2"]], {usuarios_codigo: "id_usuario_4", fecha: ISODate("2023-
  06-14")},
  comentarios: "Si deseas tener un cuerpo más flexible, te recomendaría incorporar el
  consumo de frutas en tu dieta diaria. Las frutas son una excelente fuente de vitaminas,
  minerales y antioxidantes que ayudan a fortalecer tus músculos y mejorar la flexibilidad
  de tu cuerpo. Además, su contenido de agua y fibra te mantendrá hidratado y contribuirá
  a una mejor digestión.", planes_recomendados: ["id_plan_3", "id_plan_5"]}]
},
{
  _id: "id_nutricionista_2",
  nombre: "Laura",
  apellido: "Gómez",
  especialidad: "Nutricionista clínico",
  correo: "laura_gomez@gmail.com",
  puntuaciones_usuarios: [
  { usuarios_codigo: "id_usuario_2", calificacion: 4 },

```

```

{ usuarios_codigo: "id_usuario_3", calificacion: 5 },
{ usuarios_codigo: "id_usuario_4", calificacion: 4 }
],
puntuacion: 0.0,
consultas: [{usuarios_codigo: "id_usuario_2", fecha: ISODate("2023-08-12")},
comentarios: "Si estás buscando mejorar tu resistencia física, te recomendaría incluir en
tu dieta granos integrales y consumir pollo de manera saludable. Los granos integrales
son excelentes fuentes de carbohidratos complejos. Estos carbohidratos se descomponen
lentamente en tu organismo, proporcionándote energía de forma constante y sostenida.
En cuanto al pollo, es una excelente fuente de proteínas magras. Las proteínas son
fundamentales para la reparación y construcción de tejidos musculares, lo que favorece
la recuperación y desarrollo de tu resistencia física.", planes_recomendados:
["id_plan_2", "id_plan_4"]}]
},
{
_id: "id_nutricionista_3",
nombre: "Carlos",
apellido: "Rojas",
especialidad: "Nutricionista clínico",
correo: "carlos_rojas@gmail.com",
puntuaciones_usuarios: [
{ usuarios_codigo: "id_usuario_5", calificacion: 3 }
],
puntuacion: 0.0,
consultas: [{usuarios_codigo: "id_usuario_3", fecha: ISODate("2023-06-25")},
comentarios: "Si tienes como objetivo perder peso, te recomendaría adoptar una
alimentación e incluir una amplia variedad de verduras en tu dieta diaria. Las verduras
son bajas en calorías y ricas en nutrientes esenciales como vitaminas, minerales y fibra.
Su alto contenido de fibra te ayudará a sentirte más lleno/a por más tiempo, lo que puede
contribuir a reducir la ingesta de calorías. Además de las verduras, te sugiero limitar el
consumo de alimentos procesados, altos en grasas saturadas, azúcares y sodio. Opta por
fuentes de proteína magra como pollo, pescado, tofu o legumbres, que te ayudarán a
sentirte saciado/a y a mantener la masa muscular mientras pierdes peso.",
planes_recomendados: ["id_plan_4", "id_plan_6"]}]
},
{
_id: "id_nutricionista_4",
nombre: "María",
apellido: "López",
especialidad: "Nutricionista en alimentación vegetariana",
correo: "maria_lopez@gmail.com",
puntuaciones_usuarios: [
{ usuarios_codigo: "id_usuario_2", calificacion: 5 },
{ usuarios_codigo: "id_usuario_4", calificacion: 3 }

```

```

],
puntuacion: 0.0,
consultas: [{usuarios_codigo: "id_usuario_6", fecha: ISODate("2023-07-25"),
comentarios: "Si buscas bajar de peso y mejorar tu condición cardiovascular, te
recomiendo incluir alimentos saludables y equilibrados en tu dieta. Opta por una
variedad de frutas y verduras frescas, que son ricas en vitaminas, minerales y
antioxidantes. También elige granos integrales, como arroz integral y quinoa, que te
proporcionarán energía y fibra. Incorpora fuentes de proteína magra, como legumbres,
tofu y yogur bajo en grasa. Limita el consumo de alimentos procesados, altos en azúcares
y grasas saturadas.", planes_recomendados: ["id_plan_1", "id_plan_4"]}]
},
{
_id: "id_nutricionista_5",
nombre: "Javier",
apellido: "Torres",
especialidad: "Nutricionista clínico",
correo: "javier_torres@gmail.com",
puntuaciones_usuarios: [
{ usuarios_codigo: "id_usuario_3", calificacion: 5 },
{ usuarios_codigo: "id_usuario_5", calificacion: 4 },
{ usuarios_codigo: "id_usuario_6", calificacion: 3 }
],
puntuacion: 0.0,
consultas: [{usuarios_codigo: "id_usuario_5", fecha: ISODate("2023-08-30"),
comentarios: "Si estás buscando llevar una vida saludable en general, incluir verduras
verdes en tu dieta es una excelente elección. Las verduras verdes, como espinacas, brócoli,
kale y acelgas, son una fuente nutritiva y deliciosa de beneficios para tu salud. Estas
verduras están cargadas de vitaminas, minerales y antioxidantes que ayudan a fortalecer
tu sistema inmunológico, mantener una piel radiante y promover un funcionamiento
óptimo del cuerpo. Su alto contenido de fibra también contribuye a la salud digestiva y
puede ayudar a mantener un peso saludable.", planes_recomendados: ["id_plan_6",
"id_plan_7", "id_plan_9",]}]
},
{
_id: "id_nutricionista_6",
nombre: "Ana",
apellido: "Martínez",
especialidad: "Nutricionista deportivo",
correo: "ana_martinez@gmail.com",
puntuacion: 0.0,
}
]

```

#### 4.4. Queries

En el contexto de las bases de datos, una consulta (o query, en inglés) se refiere a una solicitud o instrucción específica que se realiza a un sistema de gestión de bases de datos (DBMS) para obtener información o realizar acciones en los datos almacenados. Las consultas son utilizadas para recuperar, insertar, actualizar o eliminar datos de una base de datos de acuerdo con los criterios establecidos por el usuario.

##### 4.4.1. Relational Database Queries

Query 1: Mostrar las consultas\_nutricionista en las que se le recomiende a un usuario un plan\_alimenticio en el que haya un alimento que el usuario lo haya señalado como alimento\_no\_deseado.

```
SELECT consultas_nutricionista_codigo, pn.nombre as plan_nutricional, u.nombre as
nombre_usuario,
    a2.nombre as alimento_no_deseado FROM consultas_nutricionista cn
JOIN planes_por_consulta ppc ON cn.codigo = ppc.consultas_nutricionista_codigo
JOIN usuarios u ON cn.usuarios_codigo = u.codigo
JOIN alimentos_no_deseados a ON u.codigo = a.usuarios_codigo
JOIN planes_nutricionales pn ON ppc.planes_nutricionales_codigo = pn.codigo
JOIN alimentos_en_plan aep ON pn.codigo = aep.planes_nutricionales_codigo
JOIN alimentos a2 ON aep.alimentos_codigo = a2.codigo
WHERE a2.codigo=a.alimentos_codigo
```

Query 2: Imprimir en pantalla la lista de usuarios indicando su nombre, apellido y codigo pero agregando tambien el IMC inicial de cada uno al momento de su registro ( $IMC = \text{peso} / \text{talla}^2$ ). Imprimir los datos como cadenas de texto y tener en cuenta pasar la talla a metros para el cálculo del IMC.

```
CREATE FUNCTION obtener_imc (@codigo int) RETURNS float
AS
BEGIN
    DECLARE @imc float
    SELECT @imc=peso_inicial/((altura_inicial/100)*(altura_inicial/100))
    FROM usuarios
    WHERE codigo=@codigo
    RETURN @imc
end
```

```
CREATE PROCEDURE USPMostrarIMC_usuarios
AS
```

```

BEGIN
DECLARE @codigo int, @nombre varchar(100), @apellido varchar(256), @imc float
DECLARE c_usuarios CURSOR FOR
SELECT codigo, nombre, apellido
FROM usuarios
OPEN c_usuarios
FETCH c_usuarios INTO @codigo, @nombre, @apellido

    WHILE ( @@FETCH_STATUS = 0 )
        BEGIN
            SET @imc = dbo.obtener_imc(@codigo)
            --imprimir
            PRINT CONCAT('El IMC de ', @nombre, ' ', @apellido, ' con codigo ', @codigo, ' es ',
@imc)
            FETCH c_usuarios INTO @codigo, @nombre, @apellido
        END
    CLOSE c_usuarios
    DEALLOCATE c_usuarios
end

EXEC USPMostrarIMC_usuarios

```

Query 3:

Crear un procedimiento almacenado que muestre la cantidad total de planes\_nutricionales comprados por cada usuario y solo muestre los usuarios cuya cantidad de compras sea mayor al promedio.

```

CREATE PROCEDURE USP_Usuarios_mas_compras_prom
AS
SELECT u.codigo as codigo_usuario, SUM(cantidad) as cant_total
FROM usuarios u
JOIN boleta b ON u.codigo = b.usuarios_codigo
JOIN detalles_boleta deb ON b.codigo = deb.boleta_codigo
GROUP BY u.codigo
HAVING SUM(cantidad) > (
    SELECT AVG(cant_total)
    FROM (
        SELECT SUM(cantidad) as cant_total
        FROM usuarios u
        JOIN boleta b ON u.codigo = b.usuarios_codigo
        JOIN detalles_boleta deb ON b.codigo = deb.boleta_codigo
        GROUP BY u.codigo
    ) as t
)

```



EXEC USP\_Usuarios\_mas\_compras\_prom

Query 4: Crear una funcion que reciba como parametro el codigo de un plan\_nutricional, y que retorne una tabla con los alimentos que tiene el plan.

```
create function obtener_alimentos (@cod_plan int) returns table
AS
return(
SELECT a.nombre
FROM planes_nutricionales pn
JOIN alimentos_en_plan aep on pn.codigo = aep.planes_nutricionales_codigo
JOIN alimentos a on aep.alimentos_codigo = a.codigo
WHERE pn.codigo=@cod_plan)
```

SELECT \* from obtener\_alimentos (1)

Query 5: Crear un procedimiento almacenado que muestre el nombre y la puntuación promedio de los nutricionistas junto con la cantidad de usuarios que los han calificado.

```
create procedure nutricionistas_promedio as
SELECT n.nombre, AVG(convert(float,cn.calificacion)) AS puntuacion_promedio,
COUNT(cn.usuarios_codigo) AS cantidad_usuarios_calificadores
FROM nutricionistas n
JOIN calificaciones_nutricionista cn ON n.codigo = cn.nutricionistas_codigo
GROUP BY n.nombre
```

EXEC nutricionistas\_promedio

Query 6: Crear un procedimiento almacenado que calcule los planes\_nutricionales con mas calorías y muestre el plan nutricional con mayor cantidad de calorías.

```
create procedure planes_nutricionales_con_mas_calorias
as
select nombre, calorías from planes_nutricionales
join valores_nutricionales vn on planes_nutricionales.codigo = vn.codigo
where calorías =(select max(calorías) from planes_nutricionales
join valores_nutricionales vn on planes_nutricionales.codigo = vn.codigo)
```

EXEC planes\_nutricionales\_con\_mas\_calorias

Query 7: Mostrar el total de calorías, proteínas, carbohidratos y grasas para cada categoría de alimentos.

```
SELECT a.categoria, SUM(a.calorias) AS total_calorias, SUM(a.proteinas)
AS total_proteinas,SUM(a.carbohidratos) AS total_carbohidratos,
```

```
SUM(a.grasas) AS total_grasas
FROM alimentos a
GROUP BY a.categoria
```

Query 8: Obtener el nombre y el total de calorías consumidas por cada usuario en cada boleta.

```
SELECT u.nombre, b.codigo as codigo_boleta, SUM(a.calorias * db.cantidad) AS
total_calorias
FROM                                usuarios                                u
JOIN      boleta      b      ON      u.codigo      =      b.usuarios_codigo
JOIN      detalles_boleta      db      ON      b.codigo      =      db.boleta_codigo
JOIN      planes_nutricionales      pn      ON      db.planes_nutricionales_codigo      =      pn.codigo
JOIN      alimentos_en_plan      ap      ON      pn.codigo      =      ap.planes_nutricionales_codigo
JOIN      alimentos      a      ON      ap.alimentos_codigo      =      a.codigo
GROUP BY u.codigo, u.nombre, b.codigo
```

Query 9: Crear un procedimiento almacenado que a partir de 2 valores indique el IMC y muestre la información de los usuarios cuyo IMC se encuentre en el rango de esas 2 cantidades.

Query 10: Crear una vista que muestre los usuarios con su respectivo plan nutricional que haya comprado.

```
CREATE VIEW vista_usuarios_planes AS
SELECT u.nombre, u.apellido, pn.nombre AS plan_nutricional
FROM usuarios u
JOIN boleta b ON u.codigo = b.usuarios_codigo
JOIN detalles_boleta db ON b.codigo = db.boleta_codigo
JOIN planes_nutricionales pn ON db.planes_nutricionales_codigo = pn.codigo
```

Query 11: Mostrar la suma de los pesos hechos por cada uno de los planes\_nutricionales

```
SELECT pn.nombre AS plan_nutricional, SUM(cantidad_gramos) as peso_total
FROM                                alimentos_en_plan                                ap
JOIN      alimentos      a      ON      ap.alimentos_codigo      =      a.codigo
JOIN      planes_nutricionales      pn      ON      ap.planes_nutricionales_codigo      =      pn.codigo
GROUP BY pn.nombre
```

Query 12: Crear una función que tome como parámetros dos fechas y devuelva el nombre, apellido y el total de consultas realizadas por cada nutricionista en ese rango de fechas.

```
CREATE FUNCTION obtener_total_consultas_por_fecha(@fecha_inicio DATE, @fecha_fin
DATE)
RETURNS                                TABLE
AS
```

```

RETURN
SELECT      n.nombre,      n.apellido,      COUNT(cn.codigo)      AS      total_consultas
FROM
FROM          nutricionistas          n
JOIN      consultas_nutricionista      cn      ON      n.codigo      =      cn.nutricionistas_codigo
WHERE      cn.fecha      BETWEEN      @fecha_inicio      AND      @fecha_fin
GROUP      BY      n.nombre,      n.apellido
)

```

```
SELECT * FROM obtener_total_consultas_por_fecha('2023-06-01', '2023-07-30')
```

#### 4.4.2. Non-Relational Database Queries

Los primeros 3 queries serán updates que nos permitan completar los campos para la base de datos.

Query 1:

En la coleccion planes\_nutricionales se tiene en el campo alimentos la información nutricional por cada 100g de cada alimento considerado y su respectiva cantidad.

Agregue un query que permita actualizar todos los documentos de planes\_nutricionales y agregue un nuevo campo valor\_nutricional, el cual es un documento que tiene los datos totales de calorías, proteínas, carbohidratos y grasas.

```

db.planes_nutricionales.updateMany({},
    [{
        $set: {
            valor_nutricional: {
                calorías: {
                    $reduce: {
                        input: "$alimentos",
                        initialValue: 0,
                        in: { $add: ["$$value", {$multiply: ["$$this.calorías", "$$this.cantidad",
0.01]]} }
                    }
                },
            },
            proteínas: {
                $reduce: {
                    input: "$alimentos",
                    initialValue: 0,
                    in: { $add: ["$$value", {$multiply: ["$$this.proteínas", "$$this.cantidad",
0.01]]} }
                }
            },
        }
    }],
    {
        multi: true
    }
)

```

```

        carbohidratos: {
          $reduce: {
            input: "$alimentos",
            initialValue: 0,
            in: { $add: ["$$value", {$multiply: ["$$this.carbohidratos",
"$$this.cantidad", 0.01]]} }
          }
        },
        grasas: {
          $reduce: {
            input: "$alimentos",
            initialValue: 0,
            in: { $add: ["$$value", {$multiply: ["$$this.grasas", "$$this.cantidad",
0.01]]} }
          }
        }
      }
    }
  }
}
})

```

Query 2: En la coleccion `compras_plan` existe un campo `monto_total` que se obtiene de la suma de los precios de los planes que elige el usuario y del monto de delivery en caso aplique. Por defecto el valor de este campo es 0. Así que se solicita un query que actualice este campo con el valor correcto para todos los documentos de la colección.

```

db.compras_plan.aggregate([
  {
    $lookup: {
      from: "planes_nutricionales",
      localField: "detalles_compra.id_plan",
      foreignField: "_id",
      as: "info_planes"
    }
  },
  {
    $unwind: "$info_planes"
  },
  {
    $unwind: "$detalles_compra"
  },
  {
    $match: {
      $expr: {

```

```

    $seq: ["$detalles_compra.id_plan", "$info_planes._id"]
  }
},
{
  $project:{
    precio: "$info_planes.precio",
    cantidad: "$detalles_compra.cantidad",
    recargo_delivery: "$recargo_delivery"
  }
},
{$group:{
  _id: "$_id",
  subtotal: {$sum:{$multiply: ["$cantidad", "$precio"]}},
  recargo_delivery: {$first: "$recargo_delivery"}
}},
{$group: {
  _id: "$_id",
  monto_total: {
    $sum: {
      $cond: {
        if: { $or: [ { $seq: [ "$recargo_delivery", null ] }, { $seq: [ "$recargo_delivery", 0 ] }
] },
        then: "$subtotal",
        else: { $add: [ "$subtotal", "$recargo_delivery" ] }
      }
    }
  }
}
}
}).forEach(function(doc) {
  db.compras_plan.updateOne(
    { _id: doc._id },
    { $set: { monto_total: doc.monto_total } }
  )
})

```

Query 3: En la colección nutricionistas, tenemos un array con las distintas calificaciones que pueden brindar los usuarios, además, también hay un campo puntuacion que esta por defecto en 0. Se solicita un query que permita actualizar puntuacion a el promedio aritmético de las calificaciones hechas por los usuarios.

```

db.nutricionistas.updateMany({},
[ {

```

```

    $set: {
      puntuacion: {
        $avg: {
          $cond: {
            if: { $not: { $isArray: "$puntuaciones_usuarios" } },
            then: [0],
            else: {
              $reduce: {
                input: "$puntuaciones_usuarios.calificacion",
                initialValue: [],
                in: { $concatArrays: ["$$value", ["$$this"]] }
              }
            }
          }
        }
      }
    }
  }
}
]]
)

```

Query 4: Mostrar todos los planes\_nutricionales cuyo precio sea mayor que 13 y contengan más de 2 alimentos.

```

db.planes_nutricionales.find({
  precio: { $gt: 13 },
  $expr: { $gt: [{ $size: "$alimentos" }, 2] }
})

```

Query 5: Obtener los nutricionistas con una calificación promedio mayor a 4, mostrar su ID y puntuación promedio.

```

db.nutricionistas.aggregate([
  {
    $match: {
      "puntuaciones_usuarios": { $exists: true, $ne: [] }
    }
  },
  {
    $addFields: {
      promedio_puntuaciones: { $avg: "$puntuaciones_usuarios.calificacion" }
    }
  },
  {
    $match: {
      promedio_puntuaciones: { $gt: 4 }
    }
  }
])

```

```

}
},
{
$project: {
_id: 1,
promedio_puntuaciones: 1
}
})

```

Query 6: Mostrar una lista de todos los usuarios con su respectiva cantidad de alimentos no deseados.

```

db.usuarios.aggregate([
  { $project: {
    _id: 0,
    usuario: "$nombre",
    cantidad_alimentos_no_deseados: { $size: "$alimentos_no_deseados" }
  }},
  { $match: { cantidad_alimentos_no_deseados: 2 } },
])

```

Query 7: Obtener los usuarios que han hecho al menos una compra por un monto total como mínimo de 30 y mostrar el monto máximo al que haya llegado cada usuario.

```

db.compras_plan.aggregate([
  { $match:
    { monto_total: { $gt: 30 } } },
  { $project:
    { _id: 0, id_usuario: 1, monto_total: 1 } },
  { $group:
    { _id: '$id_usuario',
    monto_maximo: { $max: '$monto_total' } } } ]])

```

Query 8: Obtener el promedio de todos los montos totales hechos en las compras de planes nutricionales.

```

db.compras_plan.aggregate([
  {
    $group:
      {
        _id:
          promedio_monto:
            {
              $avg:
                "$monto_total"
            }
      }
  })

```

Query 9: Obtener la cantidad de consultas atendidas por cada nutricionista.

```
db.nutricionistas.aggregate([
  {
    $project: {
      _id: 0,
      codigo_nutricionista: "$_id",
      cantidad_consultas_atendidas: { $cond: { if: { $not: { $isArray: "$consultas" } }, then: 0, else:
      { $size: "$consultas" } } },
    }
  })
```

Query 10: Obtener la calificación promedio de la aplicación a partir de cada una de las calificaciones de los usuarios.

```
db.usuarios.aggregate([
  { $unwind: "$calificaciones_app" },
  { $group: {
    _id: null,
    calificacion_promedio: { $avg: "$calificaciones_app.calificacion" },
  } },
])
```

Query 11: Obtener la información de cada control hecho por los usuarios.

```
db.usuarios.aggregate([
  { $unwind: "$controles" },
  { $project: {
    id_usuario: "$_id",
    talla_registrada: "$controles.talla",
    peso_registrado: "$controles.peso",
    fecha: "$controles.fecha"
  } }
])
```

Query 12: Obtener la cantidad de compras hechas por cada usuario.

```
db.compras_plan.aggregate([
  {
    $project: {
      _id: 1,
      id_usuario: "$id_usuario",
    }
  })
```



```

    },
    {
      $group: {
        _id: "$id_usuario",
        cantidad_compras: { $sum: 1 }
      }
    }
  ]
}

```

## Conclusiones:

Se concluyó que es muy importante realizar un adecuado análisis de mercado para identificar las debilidades y fortalezas de nuestros competidores directos a fin de garantizar una sólida estrategia comercial.

Las entrevistas fueron pieza clave para poder obtener información de nuestro segmento objetivo para así poder identificar mediante preguntas claves las necesidades, objetivos, pensamientos, conocimientos y más cualidades para poder entender su forma de pensar, actuar y vivir.

En conclusión, las entrevistas a los estudiantes nos ayudaron a comprender un poco más sobre lo importante que puede ser nuestro proyecto para los jóvenes de escasos recursos del país y además de cómo podría ayudarles a motivarse para poder concluir sus estudios de la manera más satisfactoria.

Realizar el análisis del todo Logical Data Mode pasando por los pasos previos de declaración de los Entities, sus atributos y diseñar el modelo de conexiones, nos ayudó a entender mejor cómo se desarrollará la Base de Datos de nuestro proyecto para poner.

A manera de conclusión, las entrevistas fueron pieza clave para poder obtener información de nuestro segmento objetivo para así poder identificar mediante preguntas claves las necesidades, objetivos, pensamientos, conocimientos y más cualidades para poder entender su forma de pensar, actuar y vivir. Se puede concluir que Microsoft SQL server es la herramienta más apropiada para este tipo de trabajo.

Después de utilizar la aplicación de Microsoft SQL Server, concluimos que es una herramienta muy poderosa y versátil para la gestión de bases de datos. A través de su interfaz intuitiva y completa, se puede realizar diversas tareas, como la creación de tablas, consultas, almacenamiento de procedimientos y funciones, así como la administración y configuración de la base de datos.

Microsoft SQL Server se destaca por su funcionalidad, rendimiento y seguridad, y la recomendamos como una solución sólida para la gestión de bases de datos en entornos tanto individuales como grupales.