

Jennifer Paola Durango Monsalve

1128402240

Análisis de datos

Actividad 1

CODERHOUSE

Introducción

Actualmente la industria del fitness ha tomado fuerza de forma constante y vigorosa, dado que factores como el sedentarismo o el trabajo en casa, entre otros, han creado la cultura del autocuidado. Dentro de los elementos importantes de la industria del fitness podemos mencionar la ejecución de actividades físicas, el deporte de alto rendimiento y muy importante la calidad de la alimentación y/o nutrición. Dado el aumento de la demanda de los servicios y productos de la industria, se ha hecho necesario para los usuarios, ya sean entrenadores o asesorados/deportistas/atletas acceder a la información de manera ordenada y fácil para alcanzar sus objetivos.

Una de las herramientas más usadas para obtener y/o transferir la información son las APPs , las cuales permiten interacción coach/asesorado, y además permiten tener registros y conocer datos sobre el proceso.

El fondo de una APP implica tener un base de datos o un conjunto de datos, por lo cual se hace necesario integrar datos obtenidos de forma adecuada, para esto se propone en esta actividad tres DATASETS concernientes al fitness, dos de ellos son datos concernientes al entrenamiento físico y dos concernientes a la nutrición y suplementación deportiva.

De forma tentativa, uno de los objetivos del curso ANÁLISIS DE DATOS de CODERHOUSE, es realizar como proyecto final la base para la construcción de una APP que permita como coach/usuario tener un conjunto de datos asociados al entrenamiento físico y a la nutrición dentro del contexto del mismo.

Hipótesis

De manera general la hipótesis pensada al momento de crear una APP enfocada en el entrenamiento y el fitness en general es que durante la interacción entre el coach y asesorado/deportista/atleta, la presentación de la información en ocasiones es agobiante y no muy clara, además de desordenada para el usuario de la app, lo cual lleva en algunas ocasiones a que el asesorado desista de realizar su actividad dado la falta de entendimiento de ella. Otra de las finalidades y objetivos en la construcción de una APP con temática fitness es que se busca mejorar la comunicación entre coach y asesorado/deportista/atleta, además de la información disponible, sin tener que redundar constantemente en ella.

Metodología

La fuente a través de la cual se obtuvieron los DATASETS es kaggle.com, con su motor de búsqueda las palabras claves fueron FITNESS DATASETS y NUTRITION DATASETS. Se obtuvieron los siguientes DATASETS con nombres literales: *megaGymDataset*, *cardioActivities*, *Sports Supplements*, *nutrition*. Todos se visualizaron a través de la función DATOS de excel, con los parámetros origen de archivo: 1252:Europeo occidental (windows), Delimitador: coma, Detección del tipo de datos: Basado en las primeras 200 filas.

megaGymDataset/ DATASET MEGAGYM

Para este DATASET se tenía inicialmente nueva atributos o columnas y 2916 tuplas o filas, dado que no contenía la fecha se le insertó de forma arbitraria un campo de fecha, quedando así un DATASET de diez columnas y 2916 filas. Se considera un DATASET muy completo y adecuado a la hora de consultar un ejercicio, dado que entre otros contiene el nombre del ejercicio, la forma de ejecutarlo, el músculo o grupo muscular trabajado, el tipo de habilidad trabajada, los equipos utilizados y el nivel del usuario. Esta información puede dar bases adecuadas para realizar una *base de datos relacional*, con algunos de dichos atributos como tablas.

cardioActivities

Para este DATASET se tienen 14 columnas o atributos y 508 tuplas o filas. Entre otros, la información que provee dicho DATASET son el tipo de actividad, los promedios entre distancias recorridas, los promedios de velocidad, las calorías gastadas, lo cual permite posteriormente que el usuario conozca cuánta actividad y qué tipo debe realizar para lograr sus objetivos. El DATASET no fue modificado, se presenta tal cual se descargó. Se presume que posteriormente se requiere de un uso y análisis muy inteligente de los datos para poder obtener una información adecuada.

Sports Supplements

Para este DATASET se tiene 46 columnas o atributos y 96 tuplas o filas. Si bien la cantidad de tuplas es muy limitada, se tiene un DATASET muy completo dado que contiene una gran cantidad de información sobre muchos suplementos deportivos y/o moléculas químicas que realizan funciones a nivel bioquímico. Se tiene una redundancia de atributos, dado que se referencian muchas fuentes de consultas y se comparan estudios sobre la eficacia de dichos suplementos, se presume que posteriormente se realizará una limpieza de datos para evitar la redundancia o el exceso de datos innecesarios.

nutrition

Para este DATASET se tiene 77 columnas o atributos y 8788 tuplas o filas. Es un DATASET bastante grande, con una gran cantidad de datos, que permite conocer el tipo de alimentos, su composición química y/o bioquímica, las cantidades de este, entre otros; infiriendo que es una buena información a la hora de construir tablas nutricionales y relaciones con las necesidades nutricionales del asesorado/deportista/atleta.

A continuación el link de la entrada Github con el presente pdf y los DATASETS referenciados en el trabajo <https://github.com/jpdurangom/FITNESS-DATASETS>

PARTE II

Diagrama Entidad/Relación

Para la elaboración de una APP de Fitness que busca una correcta comunicación y un flujo de información entre Asesorado/Deportista/Atleta y COACH, además de elementos de valor, se considera conveniente el uso de las siguientes temáticas: Ejercicios, Nutrición, Suplementación. Por lo anterior se proponen las siguientes nueve tablas, con su nombre literal para la composición de una Base de Datos relacional: Usuarios, Entrenamientos, Ejercicios, Detalles_de_Entrenamiento Alimentos, Planes_de_Nutricion, Suplementos, Registro_de_Consumo_de_Alimentos, Registro_de_Consumo_de_Suplementos.

A continuación se presentan dos esquemas, Figura 1. Esquema Entity Relationship Diagram, elaborado en la app online miro.com y Figura 2. Diagrama Entidad Relacion , elaborado como resultado de los query realizados a través de SQL Server Management Studio, el cual se añade como documento anexo en el presente trabajo

APP FITNESS

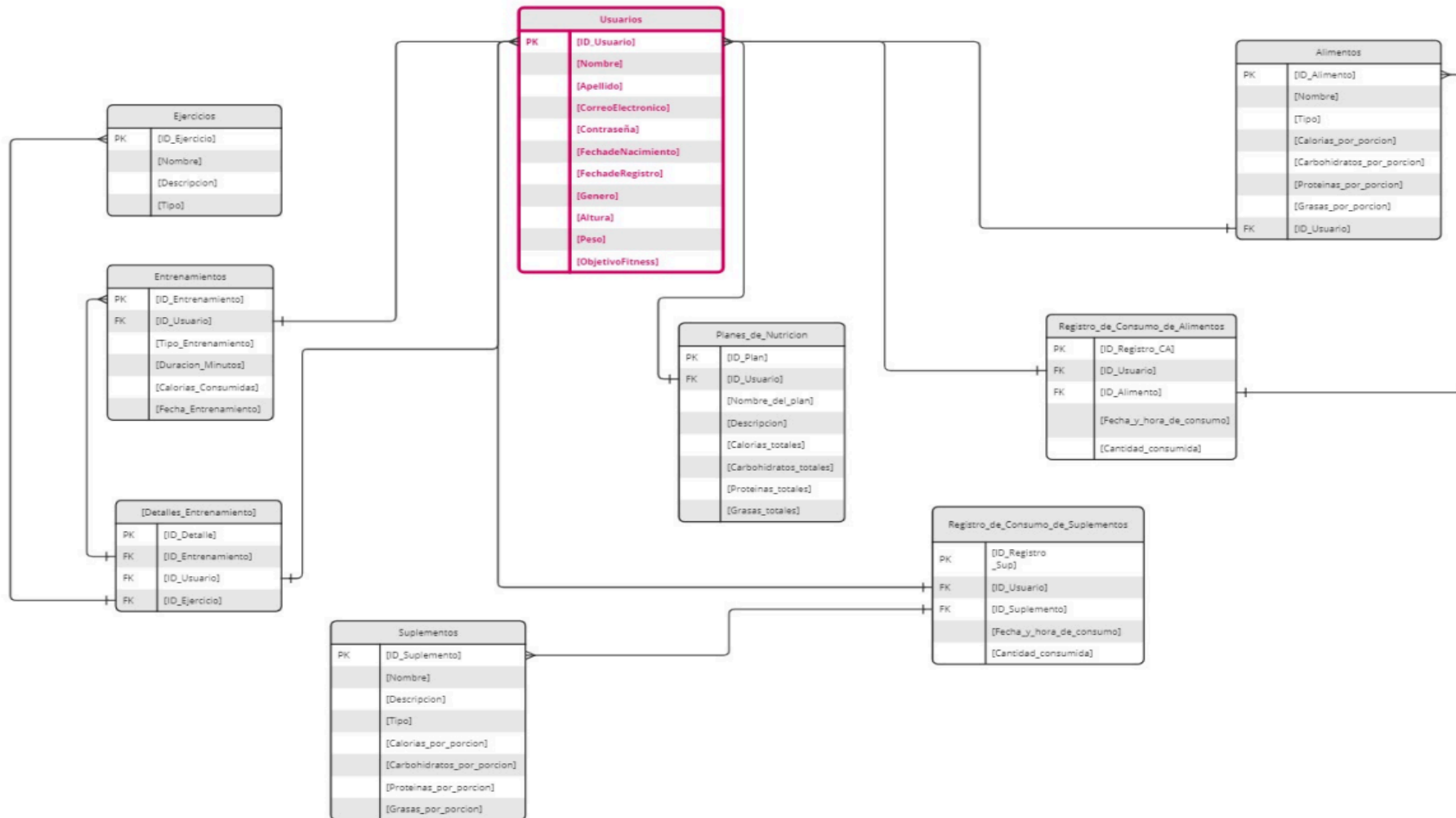


Figura 1. Esquema Entity Relationship Diagram

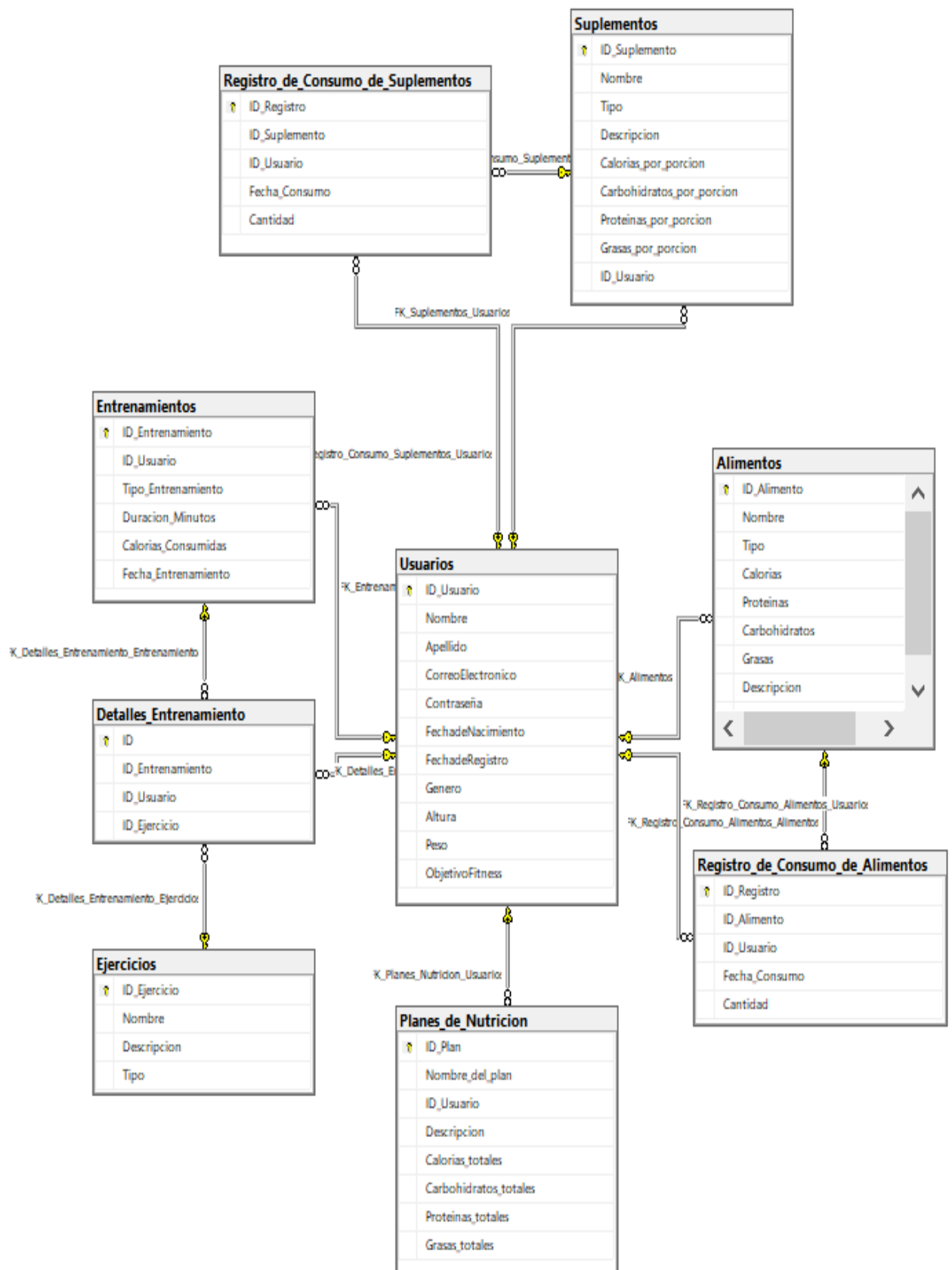


Figura 2. Diagrama Entidad Relación

Definición de Tablas

Tabla 1. Usuarios

Tipo de clave	Campo	Tipo de campo
PK	[ID_Usuario]	INT IDENTITY(1,1) PRIMARY KEY
-	[Nombre]	VARCHAR(50) NULL
-	[Apellido]	VARCHAR(50) NULL
-	[CorreoElectronico]	VARCHAR(50) NULL,
-	[Contraseña]	VARCHAR(100) NULL
-	[FechadeNacimiento]	DATE NULL
-	[FechadeRegistro]	DATETIME NULL
-	[Genero]	VARCHAR(10) NULL
-	[Altura]	DECIMAL(5,2)
-	[Peso]	DECIMAL(5,2) NULL
-	[ObjetivoFitness]	VARCHAR(300) NULL

La Tabla 1 correspondiente a los usuarios aguarda registros importantes para reconocer los objetivos a nivel físico del Usuario de la APP, parámetros como género, edad, peso, altura son de suma importancia para obtener algunas conclusiones científicas o información sobre el proceso y la evolución del mismo. La PK es el ID del usuario, parametrizado como INT IDENTITY autoincremental con la finalidad de evitar coincidencias y duplicidades en las ID.

Tabla 2. Entrenamientos

Tipo de clave	Campo	Tipo de campo
PK	[ID_Entrenamiento]	INT IDENTITY(1,1) PRIMARY KEY
FK	[ID_Usuario]	INT
-	[Tipo_Entrenamiento]	NVARCHAR(50)
-	[Duracion_Minutos]	INT
-	[Calorias_Consumidas]	DECIMAL(8,2)

-	[Fecha_Entrenamiento]	DATE
---	-----------------------	------

La Tabla 2 aguarda información sobre las sesiones de Entrenamiento del usuario, esto es de suma importancia en el fitness pues de dicha manera se mide el progreso del usuario, por ellos se incluyen los tiempos de entrenamiento y se puede tener calorías consumidas como dato de entrada o salida, ya que propone relaciones matemáticas directas con la intensidad y efectividad de los entrenamientos. La FK es la ID_Usuario dado que de esta manera vamos a relacionar las sesiones de entrenamiento con un usuario específico de la APP.

Tabla 3. Ejercicios

Tipo de clave	Campo	Tipo de campo
PK	[ID_Ejercicio]	INT IDENTITY(1,1) PRIMARY KEY
-	[Nombre]	NVARCHAR(100)
-	[Descripcion]	NVARCHAR(MAX)
-	[Tipo]	NVARCHAR(50)

La Tabla 3, reúne información sobre ejercicios, lo cual es de suma importancia en la adjudicación de las rutinas, dado que se trabajan grupos musculares y se puede discriminar algunos ejercicios que pueden no ser adecuados en determinadas situaciones. LaPK es ID_Ejercicio para reconocer los diferentes ejercicios que hay.

Tabla 4. Detalle_Entrenamiento

Tipo de clave	Campo	Tipo de campo
PK	[ID_Detalles_Ent]	INT IDENTITY(1,1) PRIMARY KEY
FK	[ID_Entrenamiento]	INT
FK	[ID_Usuario]	INT
FK	[ID_Ejercicio]	INT

La Tabla 4, Detalle_Entrenamiento, es una tabla que busca reunir información sobre los entrenamientos, los ejercicios y el usuario, con la finalidad de establecer relaciones más claras dentro de la BD.

Tabla 5. Alimentos

Tipo de clave	Campo	Tipo de campo
PF	[ID_Alimento]	INT IDENTITY(1,1) PRIMARY KEY
-	[Nombre]	NVARCHAR(100)
-	[Tipo]	NVARCHAR(50)
-	[Calorias]	DECIMAL(8,2)
-	[Proteinas]	DECIMAL(8,2)
-	[Carbohidratos]	DECIMAL(8,2)
-	[Grasas]	DECIMAL(8,2)
-	[Descripcion]	NVARCHAR(MAX)
FK	[ID_Usuario]	INT

La Tabla 5, entre otras reúne información sobre alimentos y su valor nutricional, viéndose esto como un *plus* de suprema importancia en un proceso de entrenamiento físico o preparación deportiva, relaciona a los usuarios a través de la FK ID_Usuario.

Tabla 6. Registro_de_Consumo_de_Aliementos

Tipo de clave	Campo	Tipo de campo
PK	[ID_Registro]	INT IDENTITY(1,1) PRIMARY KEY
FK	[ID_Alimento]	INT
FK	[ID_Usuario]	INT
-	[Fecha_Consumo]	DATE
-	[Cantidad]	DECIMAL(8,2)

La Tabla 6, busca de igual manera relacionar a los usuarios con los alimentos, permitiendo que estos lleven un control de su consumo nutricional.

Tabla 7. Planes_de_Nutricion

Tipo de clave	Campo	Tipo de campo
PK	[ID_Plan]	INT IDENTITY(1,1) PRIMARY KEY
-	[Nombre_del_plan]	NVARCHAR(100)
FK	[ID_Usuario]	INT
-	[Descripcion]	NVARCHAR(MAX)
-	[Calorias_totales]	NVARCHAR(10)
-	[Carbohidratos_totales]	NVARCHAR(10)
-	[Proteinas_totales]	NVARCHAR(10)
-	[Grasas_totales]	NVARCHAR(10)

La Tabla 7, permite la generación de un plan nutricional para un usuario específico, por esto su FK es ID_Usuario, pues dicho plan está enfocado en el objetivo específico del usuario.

Tabla 8. Suplementos

Tipo de clave	Campo	Tipo de campo
PK	[ID_Suplemento]	INT IDENTITY(1,1) PRIMARY KEY
-	[Nombre]	NVARCHAR(100)
-	[Tipo]	NVARCHAR(50)
-	[Descripcion]	NVARCHAR(MAX)
-	[Calorias_por_porcion]	NVARCHAR(50)
-	[Carbohidratos_por_porcion]	NVARCHAR(50)
-	[Proteinas_por_porcion]	NVARCHAR(50)
-	[Grasas_por_porcion]	NVARCHAR(50)
FK	[ID_Usuario]	INT

La tabla 8, reúne información sobre la suplementación, la cual como es sabido es un ítem fundamental en el área deportiva o fitness y además su descripción es detallada, dado que los suplementos se consumen de acuerdo a los objetivos del usuario, por ello el ID_Usuario es la FK.

Tabla 9. Registro_de_Consumo_de_Suplementos

Tipo de clave	Campo	Tipo de campo
PK	[ID_Registro]	INT IDENTITY(1,1) PRIMARY KEY
FK	[ID_Suplemento]	INT
FK	[ID_Usuario]	INT
-	[Fecha_Consumo]	DATE
-	[Cantidad]	INT

Finalmente la Tabla 9, muestra los registros de consumo de suplemento y los usuarios, la información ingresada es de suma importancia para que el usuario reconozca como impactan los suplementos consumidos y cómo estos pueden ayudar en el alcance de sus objetivos.

ANEXOS

Anexo1. Query de SQL Server Management Studio nombrado en <https://github.com/jpdurangom/FITNESS-DATASETS> como [SQLQueryAPPFIT2.sql](#)
En este Query se creó la BD APP FITNESS 2 con las nueve tablas mencionadas en el presente trabajo.

Anexo 2. Código SQL Server Management Studio

```
USE [APP FITNESS 2]
GO
```

```
--Creamos tabla para Usuarios
```

```
CREATE TABLE [dbo].[Usuarios](
    [ID_Usuario] INT IDENTITY(1,1) PRIMARY KEY,
    [Nombre] VARCHAR(50) NULL,
    [Apellido] VARCHAR(50) NULL,
    [CorreoElectronico] VARCHAR(50) NULL,
    [Contraseña] VARCHAR(100) NULL,
    [FechadeNacimiento] DATE NULL,
    [FechadeRegistro] DATETIME NULL,
    [Genero] VARCHAR(10) NULL,
    [Altura] DECIMAL(5,2) NULL,
    [Peso] DECIMAL(5,2) NULL,
    [ObjetivoFitness] VARCHAR(300) NULL
);
```

```
-- Insertar registros de ejemplo en la tabla Usuarios
```

```
INSERT INTO [dbo].[Usuarios] ([Nombre], [Apellido], [CorreoElectronico],
[Contraseña], [FechadeNacimiento], [FechadeRegistro], [Genero], [Altura], [Peso],
[ObjetivoFitness])
VALUES
('Juan', 'Pérez', 'juan@example.com', 'contraseña123', '1990-05-15', GETDATE(),
'Masculino', 175, 75, 'Aumento de masa muscular'),
('María', 'García', 'maria@example.com', 'contraseña456', '1988-10-20', GETDATE(),
'Femenino', 160, 60, 'Pérdida de peso'),
('Luis', 'Rodríguez', 'luis@example.com', 'contraseña789', '1995-03-08', GETDATE(),
'Masculino', 180, 80, 'Mantenimiento'),
('Ana', 'López', 'ana@example.com', 'contraseñaabc', '1992-07-12', GETDATE(),
'Femenino', 165, 55, 'Aumento de masa muscular'),
('Carlos', 'Martínez', 'carlos@example.com', 'contraseñadef', '1987-09-25',
GETDATE(), 'Masculino', 170, 70, 'Pérdida de peso'),
```

```
('Elena', 'Sánchez', 'elena@example.com', 'contraseñaghi', '1993-12-03',  
GETDATE(), 'Femenino', 155, 50, 'Mantenimiento'),  
( 'Miguel', 'Torres', 'miguel@example.com', 'contraseñaijk', '1989-04-18', GETDATE(),  
'Masculino', 178, 82, 'Aumento de masa muscular'),  
( 'Laura', 'Gómez', 'laura@example.com', 'contraseñalmn', '1991-08-22', GETDATE(),  
'Femenino', 162, 58, 'Pérdida de peso'),  
( 'Pedro', 'Ruiz', 'pedro@example.com', 'contraseñaopq', '1994-01-30', GETDATE(),  
'Masculino', 175, 75, 'Mantenimiento'),  
( 'Sofía', 'Díaz', 'sofia@example.com', 'contraseñarst', '1996-06-14', GETDATE(),  
'Femenino', 168, 63, 'Aumento de masa muscular');
```

```
--SELECT *FROM [dbo].[Usuarios]
```

```
--Creamos una tabla de Entrenamientos
```

```
CREATE TABLE [dbo].[Entrenamientos](  
    [ID_Entrenamiento] INT IDENTITY(1,1) PRIMARY KEY,  
    [ID_Usuario] INT,  
    [Tipo_Entrenamiento] NVARCHAR(50),  
    [Duracion_Minutos] INT,  
    [Calorias_Consumidas] DECIMAL(8,2),  
    [Fecha_Entrenamiento] DATE,  
    CONSTRAINT [FK_Entrenamientos_Usuarios] FOREIGN KEY ([ID_Usuario])  
REFERENCES [dbo].[Usuarios] ([ID_Usuario])  
);
```

```
-- Insertar registros de ejemplo en la tabla Entrenamientos
```

```
INSERT INTO [dbo].[Entrenamientos] ([ID_Usuario], [Tipo_Entrenamiento],  
[Duracion_Minutos], [Calorias_Consumidas], [Fecha_Entrenamiento])  
VALUES  
(1, 'Pesas', 60, 400, '2024-03-10'),  
(2, 'Cardio', 45, 350, '2024-03-10'),  
(3, 'Flexibilidad', 30, 200, '2024-03-10'),  
(4, 'CrossFit', 90, 600, '2024-03-10'),  
(5, 'Ciclismo', 75, 500, '2024-03-10'),  
(6, 'Yoga', 60, 250, '2024-03-10'),  
(7, 'Natación', 45, 400, '2024-03-10'),  
(8, 'Entrenamiento HIIT', 30, 350, '2024-03-10'),  
(9, 'Carrera', 90, 700, '2024-03-10'),  
(10, 'Entrenamiento funcional', 60, 450, '2024-03-10');
```

```
--SELECT *FROM [dbo].[Entrenamientos]
```

--Creamos la tabla para Ejercicios

```
CREATE TABLE [dbo].[Ejercicios](  
    [ID_Ejercicio] INT IDENTITY(1,1) PRIMARY KEY,  
    [Nombre] NVARCHAR(100),  
    [Descripcion] NVARCHAR(MAX),  
    [Tipo] NVARCHAR(50)  
);
```

-- Insertar registros de ejemplo en la tabla Ejercicios

```
INSERT INTO [dbo].[Ejercicios] ([Nombre], [Descripcion], [Tipo])  
VALUES  
(  
'Sentadillas', 'Las sentadillas son un ejercicio compuesto que involucra varios grupos musculares, como los cuádriceps, los glúteos y los músculos isquiotibiales.', 'Fuerza'),  
(  
'Carrera', 'La carrera es una actividad aeróbica que implica correr a diferentes velocidades y distancias.', 'Cardiovascular'),  
(  
'Flexiones de brazos', 'Las flexiones de brazos son un ejercicio de peso corporal que fortalece los músculos del pecho, los hombros y los tríceps.', 'Fuerza'),  
(  
'Plancha', 'La plancha es un ejercicio de fortalecimiento del core que implica mantener una posición de tabla en el suelo.', 'Core'),  
(  
'Peso muerto', 'El peso muerto es un ejercicio de levantamiento de pesas que trabaja varios grupos musculares, incluidos los glúteos, los isquiotibiales y la espalda baja.', 'Fuerza'),  
(  
'Burpees', 'Los burpees son un ejercicio de cuerpo completo que combina flexiones, saltos y sentadillas.', 'Cardiovascular'),  
(  
'Press de banca', 'El press de banca es un ejercicio de levantamiento de pesas que se enfoca en el pecho, los hombros y los tríceps.', 'Fuerza'),  
(  
'Saltos de cuerda', 'Los saltos de cuerda son un ejercicio cardiovascular que mejora la resistencia y la coordinación.', 'Cardiovascular'),  
(  
'Desplantes', 'Los desplantes son un ejercicio que fortalece las piernas y los glúteos.', 'Fuerza'),  
(  
'Mountain climbers', 'Los mountain climbers son un ejercicio de cardio que también trabaja el core y los brazos.', 'Cardiovascular');
```

--SELECT *FROM [dbo].[Ejercicios]

--Creamos una tabla Detalles de entrenamiento, buscando establecer mejores relaciones entre entrenamientos, ejercicios y usuario

```
CREATE TABLE [dbo].[Detalles_Entrenamiento](  
    [ID] INT IDENTITY(1,1) PRIMARY KEY,  
    [ID_Entrenamiento] INT,
```

```

        [ID_Usuario] INT,
        [ID_Ejercicio] INT,
        CONSTRAINT [FK_Detalles_Entrenamiento_Entrenamientos] FOREIGN KEY
        ([ID_Entrenamiento]) REFERENCES [dbo].[Entrenamientos] ([ID_Entrenamiento]),
        CONSTRAINT [FK_Detalles_Entrenamiento_Usuarios] FOREIGN KEY
        ([ID_Usuario]) REFERENCES [dbo].[Usuarios] ([ID_Usuario]),
        CONSTRAINT [FK_Detalles_Entrenamiento_Ejercicios] FOREIGN KEY
        ([ID_Ejercicio]) REFERENCES [dbo].[Ejercicios] ([ID_Ejercicio])
    );

```

-- Insertar registros de ejemplo en la tabla Detalles de entrenamiento

```

INSERT INTO [dbo].[Detalles_Entrenamiento] ([ID_Entrenamiento], [ID_Usuario],
[ID_Ejercicio])
VALUES
(1, 1, 1),
(2, 2, 2),
(3, 3, 3),
(4, 4, 4),
(5, 5, 5),
(6, 6, 6),
(7, 7, 7),
(8, 8, 8),
(9, 9, 9),
(10, 10, 10);

```

--SELECT *FROM [dbo].[Detalles_Entrenamiento]

--Creamos tabla para Alimentos

```

CREATE TABLE [dbo].[Alimentos](
    [ID_Alimento] INT IDENTITY(1,1) PRIMARY KEY,
    [Nombre] NVARCHAR(100),
    [Tipo] NVARCHAR(50),
    [Calorias] DECIMAL(8,2),
    [Proteinas] DECIMAL(8,2),
    [Carbohidratos] DECIMAL(8,2),
    [Grasas] DECIMAL(8,2),
    [Descripcion] NVARCHAR(MAX),
    [ID_Usuario] INT,
    CONSTRAINT [FK_Alimentos] FOREIGN KEY ([ID_Usuario]) REFERENCES
    [dbo].[Usuarios] ([ID_Usuario]),
);

```

-- Insertar registros de ejemplo en la tabla Alimentos

```
INSERT INTO [dbo].[Alimentos] ([Nombre], [Tipo], [Calorias], [Proteinas],  
[Carbohidratos], [Grasas], [Descripcion], [ID_Usuario])  
VALUES  
( 'Pollo a la parrilla', 'Proteína', 250, 30, 0, 10, 'Pollo a la parrilla sin piel', 1),  
( 'Arroz integral', 'Carbohidrato', 200, 5, 45, 1, 'Arroz integral cocido', 2),  
( 'Espinacas', 'Verdura', 20, 3, 1, 0, 'Espinacas crudas', 3),  
( 'Salmón', 'Proteína', 300, 25, 0, 20, 'Filete de salmón al horno', 4),  
( 'Avena', 'Carbohidrato', 150, 6, 30, 2, 'Avena en hojuelas cocida', 5),  
( 'Huevo', 'Proteína', 80, 6, 1, 5, 'Huevo cocido', 6),  
( 'Brócoli', 'Verdura', 30, 2, 5, 0, 'Brócoli cocido al vapor', 7),  
( 'Atún en lata', 'Proteína', 120, 25, 0, 3, 'Atún en lata en agua', 8),  
( 'Quinoa', 'Carbohidrato', 180, 8, 35, 3, 'Quinoa cocida', 9),  
( 'Yogur griego', 'Proteína', 150, 10, 8, 5, 'Yogur griego natural', 10);
```

--SELECT *FROM [dbo].[Alimentos]

--Creamos una tabla para Registro de consumo de alimentos, lo cual nos permite establecer relaciones entre alimentos y usuarios

```
CREATE TABLE [dbo].[Registro_de_Consumo_de_Alimentos](  
    [ID_Registro] INT IDENTITY(1,1) PRIMARY KEY,  
    [ID_Alimento] INT,  
    [ID_Usuario] INT,  
    [Fecha_Consumo] DATE,  
    [Cantidad] DECIMAL(8,2),  
    CONSTRAINT [FK_Registro_Consumo_Alimentos_Alimentos] FOREIGN KEY  
    ([ID_Alimento]) REFERENCES [dbo].[Alimentos] ([ID_Alimento]),  
    CONSTRAINT [FK_Registro_Consumo_Alimentos_Usuarios] FOREIGN KEY  
    ([ID_Usuario]) REFERENCES [dbo].[Usuarios] ([ID_Usuario])  
);
```

-- Insertar registros de ejemplo en la tabla consumo de Alimentos

```
INSERT INTO [dbo].[Registro_de_Consumo_de_Alimentos] ([ID_Alimento],  
[ID_Usuario], [Fecha_Consumo], [Cantidad])  
VALUES  
(1, 1, '2024-03-10', 200),  
(2, 2, '2024-03-10', 150),  
(3, 3, '2024-03-10', 100),  
(4, 4, '2024-03-10', 180),  
(5, 5, '2024-03-10', 120),  
(6, 6, '2024-03-10', 100),
```

```
(7, 7, '2024-03-10', 80),  
(8, 8, '2024-03-10', 130),  
(9, 9, '2024-03-10', 150),  
(10, 10, '2024-03-10', 200);
```

```
--SELECT *FROM [dbo].[Registro_de_Consumo_de_Alimentos]
```

```
--Creamos tabla para Planes de Nutricion
```

```
CREATE TABLE [dbo].[Planes_de_Nutricion](  
    [ID_Plan] INT IDENTITY(1,1) PRIMARY KEY,  
    [Nombre_del_plan] NVARCHAR(100),  
    [ID_Usuario] INT,  
    [Descripcion] NVARCHAR(MAX),  
    [Calorias_totales] NVARCHAR(10),  
    [Carbohidratos_totales] NVARCHAR(10),  
    [Proteinas_totales] NVARCHAR(10),  
    [Grasas_totales] NVARCHAR(10),  
  
    CONSTRAINT [FK_Planes_Nutricion_Usuarios] FOREIGN KEY ([ID_Usuario])  
REFERENCES [dbo].[Usuarios] ([ID_Usuario])  
);
```

```
--Creamos algunos valores para la tabla Planes de nutricion
```

```
INSERT INTO [dbo].[Planes_de_Nutricion] ([Nombre_del_plan], [ID_Usuario],  
[Descripcion], [Calorias_totales], [Carbohidratos_totales], [Proteinas_totales],  
[Grasas_totales])  
VALUES  
(  
'Plan 1', 1, 'Plan de alimentación para aumentar masa muscular', '2500', '300', '200',  
'70'),  
(  
'Plan 2', 2, 'Plan de alimentación para pérdida de peso', '1800', '150', '120', '50'),  
(  
'Plan 3', 3, 'Plan de alimentación para mantenimiento', '2000', '200', '150', '60'),  
(  
'Plan 4', 4, 'Plan de alimentación para mejorar rendimiento deportivo', '2800', '350',  
'220', '80'),  
(  
'Plan 5', 5, 'Plan de alimentación para aumento de energía', '2300', '250', '180', '65'),  
(  
'Plan 6', 6, 'Plan de alimentación para control de glucosa', '1800', '120', '100', '40'),  
(  
'Plan 7', 7, 'Plan de alimentación para mejorar la salud cardiovascular', '2000', '180',  
'150', '55'),  
(  
'Plan 8', 8, 'Plan de alimentación para entrenamientos intensos', '2700', '320', '210',  
'75'),  
(  
'Plan 9', 9, 'Plan de alimentación para vegetarianos', '2200', '270', '170', '60'),  
(  
'Plan 10', 10, 'Plan de alimentación para aumento de masa muscular magra', '2600',  
'300', '220', '70');
```



```
--SELECT *FROM [dbo].[Planes_de_Nutricion]
```

--Creamos una Tabla para Suplementos

```
CREATE TABLE [dbo].[Suplementos](
  [ID_Suplemento] INT IDENTITY(1,1) PRIMARY KEY,
  [Nombre] NVARCHAR(100),
  [Tipo] NVARCHAR(50),
  [Descripcion] NVARCHAR(MAX),
  [Calorias_por_porcion] NVARCHAR(50),
  [Carbohidratos_por_porcion] NVARCHAR(50),
  [Proteinas_por_porcion] NVARCHAR(50),
  [Grasas_por_porcion] NVARCHAR(50),
  [ID_Usuario] INT,
  CONSTRAINT [FK_Suplementos_Usuarios] FOREIGN KEY ([ID_Usuario])
REFERENCES [dbo].[Usuarios] ([ID_Usuario])
);
```

--Creamos algunas entradas para la tabla suplemento

```
INSERT INTO [dbo].[Suplementos] ([Nombre], [Tipo], [Descripcion],
[Calorias_por_porcion], [Carbohidratos_por_porcion], [Proteinas_por_porcion],
[Grasas_por_porcion], [ID_Usuario])
VALUES
('Proteína en polvo', 'Proteína', 'Proteína en polvo sabor chocolate', '120', '2g', '25g',
'1g', 1),
('Creatina', 'Energético', 'Suplemento de creatina monohidratada', '5', '0g', '0g', '0g',
2),
('BCAA', 'Aminoácido', 'Suplemento de aminoácidos de cadena ramificada', '10', '0g',
'5g', '0g', 3),
('Multivitamínico', 'Vitaminas', 'Suplemento multivitamínico', '15', '3g', '0g', '0g', 4),
('Omega 3', 'Ácidos grasos', 'Suplemento de ácidos grasos omega 3', '20', '0g', '0g',
'2g', 5),
('Glutamina', 'Aminoácido', 'Suplemento de glutamina en polvo', '5', '0g', '5g', '0g', 6),
('Cafeína', 'Energético', 'Suplemento de cafeína en tabletas', '0', '0g', '0g', '0g', 7),
('ZMA', 'Minerales', 'Suplemento de zinc, magnesio y vitamina B6', '10', '1g', '0g', '0g',
8),
('Caseína', 'Proteína', 'Proteína de liberación lenta sabor vainilla', '100', '2g', '20g',
'1g', 9),
('Pre-entrenamiento', 'Energético', 'Suplemento pre-entrenamiento', '50', '10g', '5g',
'0g', 10);
```

```
--SELECT *FROM [dbo].[Suplementos]
```

--Creamos una tabla de Registro de consumo de suplementos para establecer relaciones entre suplementos y usuarios

```
CREATE TABLE [dbo].[Registro_de_Consumo_de_Suplementos](
    [ID_Registro] INT IDENTITY(1,1) PRIMARY KEY,
    [ID_Suplemento] INT,
    [ID_Usuario] INT,
    [Fecha_Consumo] DATE,
    [Cantidad] INT,
    CONSTRAINT [FK_Registro_Consumo_Suplementos_Suplementos] FOREIGN
KEY ([ID_Suplemento]) REFERENCES [dbo].[Suplementos] ([ID_Suplemento]),
    CONSTRAINT [FK_Registro_Consumo_Suplementos_Usuarios] FOREIGN KEY
([ID_Usuario]) REFERENCES [dbo].[Usuarios] ([ID_Usuario])
);
```

--Creamos algunas entradas para la tabla Registro_consumo_suplementos

```
INSERT INTO [dbo].[Registro_de_Consumo_de_Suplementos] ([ID_Suplemento],
[ID_Usuario], [Fecha_Consumo], [Cantidad])
VALUES
(1, 1, '2024-03-10', 1),
(2, 2, '2024-03-10', 2),
(3, 3, '2024-03-10', 1),
(4, 4, '2024-03-10', 1),
(5, 5, '2024-03-10', 1),
(6, 6, '2024-03-10', 2),
(7, 7, '2024-03-10', 1),
(8, 8, '2024-03-10', 1),
(9, 9, '2024-03-10', 1),
(10, 10, '2024-03-10', 2);
```

--SELECT *FROM [dbo].[Registro_de_Consumo_de_Suplementos]