

Universidad De Puerto Rico

Recinto De Mayagüez

## Proyecto Final DB- Fitness Tracker

Marco A. Yu Cordero

Javier A. Flores Ayala

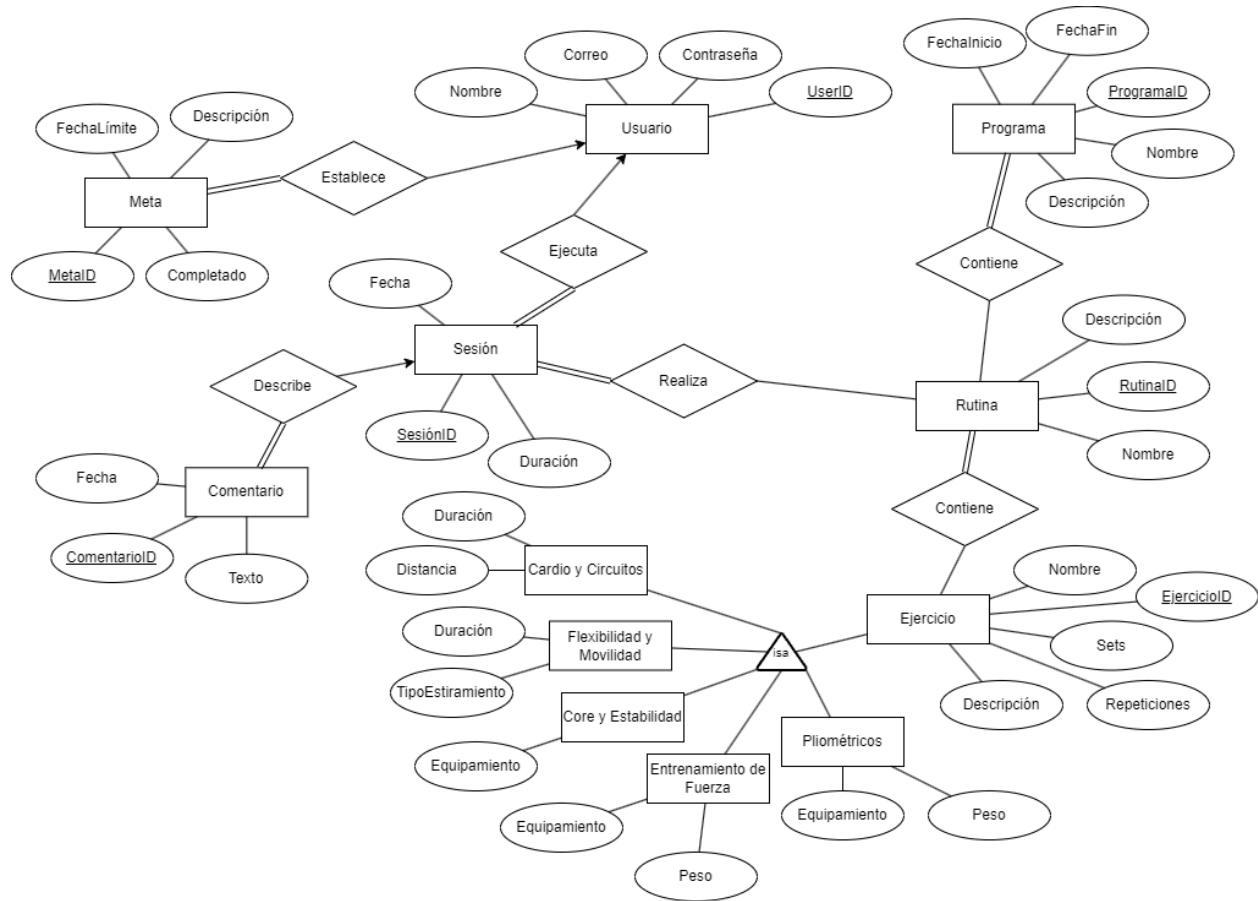
Departamento de Matemáticas

Ciencias en Computadoras

## Indice

<b>Modelo ER.....</b>	<b>3</b>
Entidades.....	3
Relaciones.....	6
<b>Dependencias funcionales y restricciones.....</b>	<b>8</b>
Dependencias Funcionales.....	8
Restricciones.....	9
<b>Convertir modelo ER en Tablas.....</b>	<b>11</b>
Normalización hasta 3NF.....	12
Normalización hasta BCNF.....	12

## Modelo ER



## Entidades

### 1. Usuario

- Atributos: UserID (PK), Nombre, Email, Contraseña.
- Tipo de Entidad: Fuerte, ya que tiene una clave primaria propia y es independiente.

### 2. Sesión

- Atributos: SesiónID (PK), Fecha, Duración.
- Tipo de Entidad: Fuerte, aunque depende del usuario (relación).

### 3. Comentario

- Atributos: ComentarioID (PK), Texto, Fecha.
- Tipo de Entidad: Fuerte, pero dependiente de la relación con el usuario.

### 4. Rutina

- Atributos: RutinaID (PK), Nombre, Descripción.
- Tipo de Entidad: Fuerte, con clave primaria propia.

### 5. Ejercicio

- Atributos: EjercicioID (PK), Nombre, Descripción, Sets, Repeticiones.
- Tipo de Entidad: Fuerte, con clave primaria propia.

### 6. Programa

- Atributos: ProgramaID (PK), Nombre, Descripción, FechaInicio, FechaFin.
- Tipo de Entidad: Fuerte, con clave primaria propia.

### 7. Meta

- Atributos: MetaID (PK), Descripción, FechaLimite, Completado.
- Tipo de Entidad: Fuerte, aunque vinculada a un usuario.

### 8. Cardio y Circuitos

- Atributos: Duración, Distancia
- Tipo de Entidad: Subentidad de ejercicio

## 9. Entrenamiento de Fuerza

- Atributos: Peso, Equipamiento
- Tipo de Entidad: Subentidad de ejercicio

## 10. Flexibilidad y Movilidad

- Atributos: Duración, TipoEstiramiento
- Tipo de Entidad: Subentidad de ejercicio

## 11. Core y Estabilidad

- Atributos: Equipamiento
- Tipo de Entidad: Subentidad de ejercicio

## 12. Pliométricos

- Atributos: Equipamiento, Peso
- Tipo de Entidad: Subentidad de ejercicio

## Relaciones

### 1. Ejecuta (Usuario -> Sesión)

- Tipo: Uno a muchos (1:N).
- Tipo de Participación: Total del lado de sesión, parcial para usuario.
- Descripción: Muchas sesiones pueden estar asociadas a un usuario, pero un usuario sólo puede estar asociada a una sesión. No puede haber una sesión que no esté relacionada con al menos un usuario, pero un usuario puede estar sin relacionarse a una sesión.

### 2. Realiza (Rutina -> Sesión)

- Tipo: Muchosa muchos (M:N).
- Tipo de Participación: Total del lado de sesión, parcial para rutina.
- Descripción: Muchas rutinas pueden estar asociadas a muchas sesiones. No puede haber una sesión que no esté relacionada con al menos una rutina, pero puede existir una rutina que no esté asociada a una sesión.

### 3. Comenta (Sesión -> Comentario)

- Tipo: Uno a muchos (1:N)
- Tipo de Participación: Total del lado de comentario y parcial del lado de sesión.
- Descripción: Una sesión puede tener muchos comentarios, pero un comentario sólo puede estar asociada a una sesión. Una sesión puede no tener comentarios, pero los comentarios deben estar asociados a una sesión.

#### 4. Contiene (Rutina -> Ejercicio)

- Tipo: Muchos a muchos (M:N)
- Tipo de Participación: Total del lado de rutina y parcial del lado de ejercicio.
- Descripción: Una rutina puede contener varios ejercicios y un ejercicio puede estar en varias rutinas. Una rutina debe tener al menos un ejercicio. No todos los ejercicios tienen que estar en una rutina.

#### 5. Contiene (Programa -> Rutina)

- Tipo: Muchos a muchos (M:N)
- Tipo de Participación: Total del lado de programa y parcial para rutina.
- Descripción: Un programa puede contener varias rutinas y una rutina puede estar en varios programas. Un programa está diseñado para contener al menos una rutina. Algunas rutinas están diseñadas inicialmente sin asignarlas a un programa específico, permitiendo flexibilidad en su asignación futura.

#### 6. Establece (Usuario -> Meta)

- Tipo: Uno a muchos (1:N)
- Tipo de Participación: Parcial del lado de usuario y total para el lado de meta.
- Descripción: Un usuario puede establecer varias metas, pero una meta tiene que estar asociada a un solo usuario. No puede existir una meta sin ser asociada a un usuario, pero un usuario puede existir sin establecer metas.

## Dependencias funcionales y restricciones

### Dependencias Funcionales

1. Usuario:
  - $\text{UserID} \rightarrow \text{Nombre, Correo, Contraseña}$
  - $\text{Correo} \rightarrow \text{UserID, Nombre, Contraseña}$
2. Sesión:
  - $\text{SesiónID} \rightarrow \text{UserID, Fecha, Duración}$
3. Comentario:
  - $\text{ComentarioID} \rightarrow \text{SesiónID, Texto, Fecha}$
4. Programa:
  - $\text{ProgramaID} \rightarrow \text{Nombre, Descripción, FechaInicio, FechaFin}$
5. Rutina:
  - $\text{RutinaID} \rightarrow \text{SesiónID, Nombre, Descripción}$
6. Ejercicio:
  - $\text{EjercicioID} \rightarrow \text{Nombre, Descripción, Sets, Repeticiones, Peso, Equipamiento, Duración, Distancia, TipoEstiramiento, isEntrenamientoDeFuerza, isCardio\_Circuitos, isCore\_Estabilidad, isPliométricos, isFlexibilidad\_Movilidad}$
7. Meta:
  - $\text{MetaID} \rightarrow \text{UserID, FechaLímite, Completado, Descripción}$



## Restricciones

### 1. Restricciones de Integridad Referencial

Estas restricciones aseguran que las relaciones entre tablas permanezcan consistentes. Por ejemplo:

- Restricciones de Clave Foránea:
  - En la tabla Sesión, UserID debe ser una clave foránea que referencia a UserID en la tabla Usuario. Esto asegura que no se pueda crear una sesión para un usuario que no existe.
  - En Comentario, SesiónID debe ser una clave foránea que referencia SesiónID en la tabla Sesión, garantizando que cada comentario esté vinculado a una sesión existente.
  - En las tablas de relación como PlanificarRutina (con UserID y RutinaID), ambas deben ser claves foráneas que referencian a sus respectivas tablas primarias. Esto valida que las rutinas y usuarios asociados realmente existan.

### 2. Restricciones de Dominio

Estas restricciones definen los valores permitidos para los atributos en la base de datos. O sea, se definen al crear cada tabla:

- Tipos de Datos: Asegurar que cada atributo tenga el tipo de datos más apropiado. Por ejemplo, 'Duración' podría ser un entero que representa minutos, mientras que 'Correo' debe ser un tipo de datos de texto formateado adecuadamente.
- Valores de Atributos:

- 'FechaInicio' y 'FechaFin' en la tabla Programa deben ser fechas válidas, y 'FechaFin' debe ser igual o posterior a 'FechaInicio'.
- 'Sets' y 'Repeticiones' en Ejercicio deben ser enteros positivos.
- Restricciones como 'isEntrenamientoDeFuerza', 'isCardio\_Circuito's, etc., deberían ser booleanos (TRUE o FALSE).
- 'Correo' debe ser único, o sea, con una restricción de unicidad.

### 3. Restricciones de Participación

Estas restricciones determinan si la presencia de un dato en una tabla implica la necesidad de datos relacionados en otra tabla:

- Participación Total:
  - En Sesión, cada sesión debe estar asociada a un Usuario, reflejando una participación total del lado de la sesión en la relación con Usuario.
  - En Comentario, cada comentario debe estar vinculado a una sesión específica, etc.

## Convertir modelo ER en Tablas

1. Usuario(UserID, Nombre, Correo, Contraseña)
2. Sesión(SesiónID, UserID, Fecha, Duración)
3. Comentario(ComentarioID, SesiónID, Texto, Fecha)
4. Programa(ProgramaID, Nombre, Descripción, FechaInicio, FechaFin)
5. Rutina(RutinaID, Nombre, Descripción)
6. Ejercicio(EjercicioID, Nombre, Descripción, Sets, Repeticiones, Peso, Equipamiento, Duración, Distancia, TipoEstiramiento, isEntrenamientoDeFuerza, isCardio\_Circuitos, isCore\_Estabilidad, isPliométricos, isFlexibilidad\_Movilidad)
7. Meta(MetaID, UserID, FechaLímite, Completado, Descripción)
8. SesionContieneRutina(SesiónID, RutinaID)
9. RutinaContieneEjercicio(RutinaID, EjercicioID)
10. ProgramaContieneRutina(ProgramaID, RutinaID)

PK

FK

Booleano

**Normalización hasta 3NF**

Todas las tablas listadas están en 3NF porque no hay atributos no clave que dependen de otras claves no primarias.

**Normalización hasta BCNF**

Las tablas están en BCNF ya que no hay dependencias funcionales que violen la condición de BCNF. Es decir, todas las dependencias funcionales tienen superclaves como determinantes.