

Aficiones

Pepe Calderón, Fernando Sola, Daniel Ayala, Inma Hernández, Margarita Cruz, Carlos Arévalo, David Ruiz



Escuela Técnica Superior de
Ingeniería Informática

Índice

1. Requisitos	2
2. RI-1: Usuarios	3
3. RI-02: Aficiones	4
4. RF-03: Informes sobre aficiones	5
5. Modificación solicitada:	6
6. Modelo Conceptual	7
7. Diagrama de clases	8
8. Diagrama de clases	9
9. Variante estática (Aficiones como atributos en su relación)	10
9.1. Intensión	10
9.2. Extensión	10
9.3. Álgebra relacional	11
10. Variante dinámica (tabla intermedia usuario–afición)	12
10.1. Intensión	12
10.2. Extensión	12
10.3. Álgebra relacional	13
11. Modelo Tecnológico (MariaDB)	14
12. Versión Estática	15
12.1. Script SQL para crear la base de datos	15
12.2. Script SQL para la carga inicial de datos	15
12.3. Consultas	16
13. Versión Dinámica	17
13.1. Script SQL para crear la base de datos	17
13.2. Script SQL para la carga inicial de datos	17
13.3. Consultas	18
13.4. SQL Avanzado	18
14. Pruebas SQL	20
15. Transacciones	21

Requisitos

Catálogo de Requisitos Añada los siguientes requisitos al ejercicio Usuarios:

RI-1: Usuarios

- Mismo que ejercicio de Usuarios

RI-02: Aficiones

- Como: Profesor de la asignatura
- Quiero: Almacenar las aficiones de los usuarios, que pueden ser: literatura, cine, deporte o gastronomía. Los usuarios pueden varias aficiones o ninguna.
- Para: Que el estudiante tenga en cuenta este requisito en el modelo conceptual, relacional y tecnológico

RF-03: Informes sobre aficiones

- Como: Profesor de la asignatura
- Quiero: Que el sistema sea capaz de generar los siguientes informes:
 - Listado de usuarios con sus aficiones.
 - Usuarios a los que le gusta el cine.
 - Usuarios que no tienen aficiones.
 - Número de aficiones de cada usuario.
 - Máximo número de aficiones que tiene un usuario.
 - Usuarios con el máximo número de aficiones.
- Para: Que el alumno realice consultas en Álgebra Relacional y SQL.

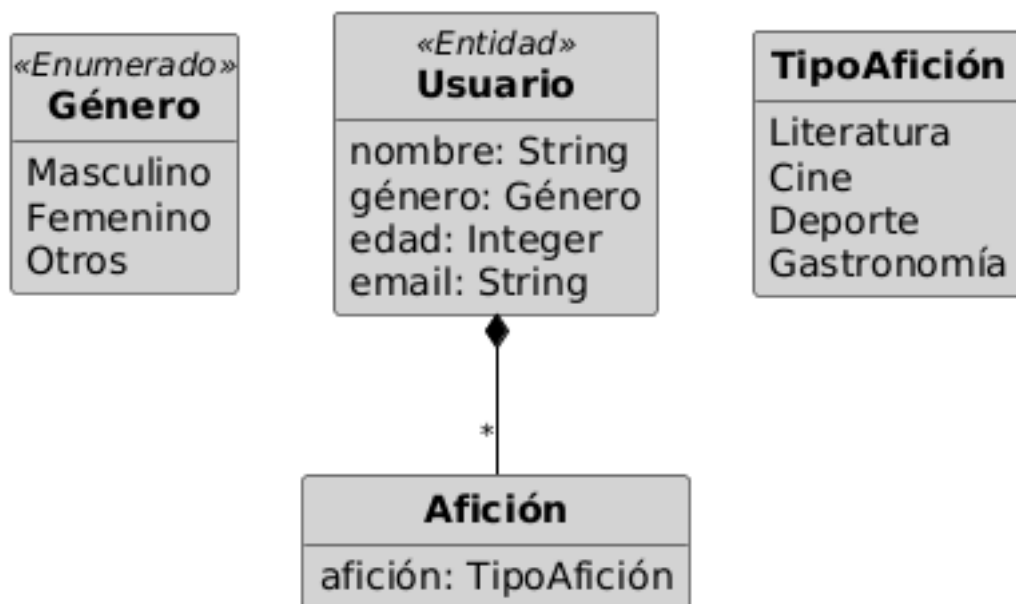
Modificación solicitada:

Modifique el RI-02 de la versión estática/cerrada del ejercicio de Aficiones para que las aficiones de los Usuarios sean dinámicas/abierta.

Modelo Conceptual

Modelo conceptual (Versión estática)

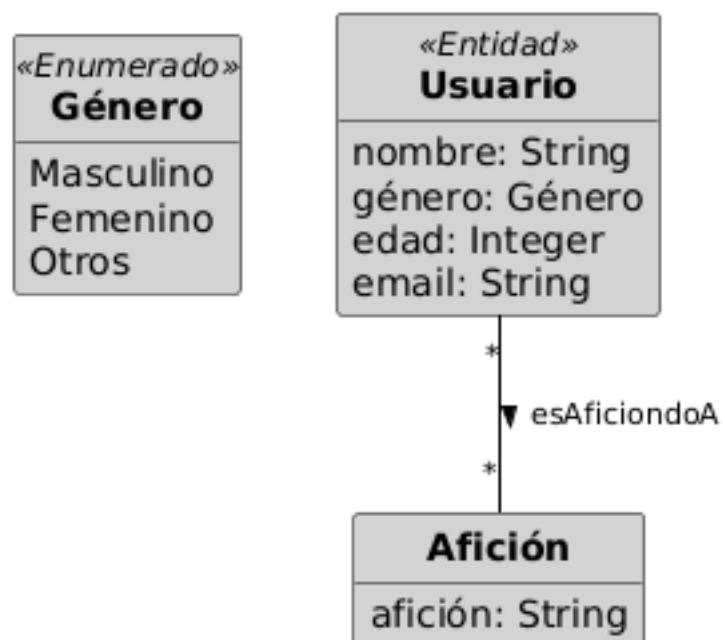
Diagrama de clases



Modelo conceptual (Versión dinámica)

Diagrama de clases

En la versión dinámica, las aficiones se convierten en entidad propia para permitir un catálogo abierto y gestionable. Normalmente aparecen las entidades Usuario y Afición, y una asociación Usuario–Afición para resolver la relación *.*.



Modelo relacional

Variante estática (Aficiones como atributos en su relación)

9.1. Intensión

```
Usuarios(usuarioId, nombre, edad, género, email)
  PK(usuarioId)
  AK(email)
Aficiones(aficionId, usuarioId, afición)
  PK(aficionId)
  FK(usuarioId) / Usuarios
  AK(usuarioId, afición)
```

9.2. Extensión

```
Usuarios = { u1..u11 como en el ejercicio de Usuarios }
Aficiones = {
  (a1, u1, "Deporte"),
  (a2, u1, "Gastronomía"),
  (a3, u2, "Deporte"),
  (a4, u2, "Literatura"),
  (a5, u2, "Cine"),
```

```
(a6, u4, "Gastronomía"),
(a7, u4, "Cine"),
(a8, u4, "Literatura"),
(a9, u5, "Deporte"),
(a10, u6, "Cine"),
(a11, u8, "Deporte"),
(a12, u8, "Gastronomía"),
(a13, u8, "Literatura"),
(a14, u8, "Cine"),
(a15, u9, "Deporte"),
(a16, u9, "Literatura"),
(a17, u9, "Cine"),
(a18, u10, "Gastronomía")
}
```

9.3. Álgebra relacional

- Renombrado:

$$\rho_{U(uid,nu,ed,g,em)}(Usuarios)$$

$$\rho_{A(aid,uid,af)}(Aficiones)$$

- Usuarios con sus aficiones:

$$UA \leftarrow U \bowtie A$$

- Usuarios a las que les gusta el cine:

$$UCine \leftarrow \sigma_{af=CINE}(UA)$$

- Usuarios sin aficiones:

$$UsuSinAfi \leftarrow \Pi_{uid,nu}(U \bowtie (\Pi_{uid}(U) \setminus \Pi_{uid}(A)))$$

- Número de aficiones por usuario:

$$NumAfiUsu \leftarrow \gamma_{COUNT(*) uid}(UA)$$

Variante dinámica (tabla intermedia usuario–afición)

10.1. Intensión

```
Usuarios(usuarioId, nombre, edad, género, email)
  PK(usuarioId)
  AK(email)
Aficiones(aficionId, afición)
  PK(aficionId)
UsuariosAficiones(usuarioAficionId, usuarioId, aficiónId)
  PK(usuarioAficionId)
  FK(usuarioId) / Usuarios
  FK(aficiónId) / Aficiones
  AK(usuarioId, aficiónId)
```

10.2. Extensión

```
Usuarios = { u1..u11 como en el ejercicio de Usuarios }

Aficiones = {
  (a1, "Cine"), (a2, "Deporte"), (a3, "Jugar al fútbol"),
```

```

    (a4, "Hacer senderismo"), (a5, "Montar a caballo")
}
UsuariosAficiones = {
    (u1,a3), (u1,a4), (u2,a1), (u2,a2), (u2,a3), (u4,a4),
    (u4,a2), (u4,a1), (u5,a3), (u6,a2), (u8,a3), (u8,a4),
    (u8,a1), (u8,a2), (u9,a3), (u9,a1), (u9,a2), (u10,a4),
    (u11,a5)
}

```

10.3. Álgebra relacional

- Renombrado:

$$\begin{aligned}
 & \rho_{U(uid, nu, ed, g, em)} (Usuarios) \\
 & \rho_{A(aid, af)} (Aficiones) \\
 & \rho_{UA(uid, aid)} (UsuariosAficiones)
 \end{aligned}$$

- Usuarios con sus aficiones:

$$UAA \leftarrow U \bowtie UA \bowtie A$$

- Usuarios a los que les gusta el cine:

$$UCine \leftarrow \Pi_{nu} \left(\sigma_{af = \text{CINE}} (UAA) \right)$$

- Usuarios sin aficiones:

$$USinAfi \leftarrow \Pi_{nu} \left(U \bowtie \left(\Pi_{uid} (U) \setminus \Pi_{uid} (UA) \right) \right)$$

- Usuarios con todas las aficiones:

$$UsuTodasAfi \leftarrow \Pi_{nu, uid} \left(\frac{UA}{\Pi_{aid} (A)} \bowtie U \right)$$

Modelo Tecnológico (MariaDB)

Modelo tecnológico (MariaDB)

Versión Estática

12.1. Script SQL para crear la base de datos

AficionesEst/createDB.sql

Info

Código SQL disponible en: <https://raw.githubusercontent.com/IISSE-US/silence-db/main/AficionesEst/sql/createDB.sql>

12.2. Script SQL para la carga inicial de datos

AficionesEst/populateDB.sql

Info

Código SQL disponible en: <https://raw.githubusercontent.com/IISSE-US/silence-db/main/AficionesEst/sql/populateDB.sql>

12.3. Consultas

AficionesEst/queries.sql

i Info

Código SQL disponible en: <https://raw.githubusercontent.com/ISSI-US/silence-db/main/AficionesEst/sql/queries.sql>

Versión Dinámica

13.1. Script SQL para crear la base de datos

AficionesDin/createDB.sql

Info

Código SQL disponible en: <https://raw.githubusercontent.com/IISSE-US/silence-db/main/AficionesDin/sql/createDB.sql>

13.2. Script SQL para la carga inicial de datos

AficionesDin/populateDB.sql

Info

Código SQL disponible en: <https://raw.githubusercontent.com/IISSE-US/silence-db/main/AficionesDin/sql/populateDB.sql>

13.3. Consultas

AficionesDin/queries.sql

Info

Código SQL disponible en: <https://raw.githubusercontent.com/IISSI-US/silence-db/main/AficionesDin/sql/queries.sql>

13.4. SQL Avanzado

Realice un procedimiento para insertar en la tabla de usuarios e implemente la siguiente prueba de aceptación:

Pruebas de aceptación **PA-001: Usuarios**

1. ☒ Insertar un usuario con datos correctos.
2. ☒ Insertar un usuario sin género.
3. ☐ Insertar un usuario con un email repetido.
4. ☐ Insertar un usuario menor de edad.

AficionesDin/pTestUsuario.sql

Info

Código SQL disponible en: <https://raw.githubusercontent.com/IISSI-US/silence-db/main/AficionesDin/sql/pTestUsuario.sql>

Transacciones Realice un procedimiento para insertar una afición a un usuario nuevo, es decir, que inserte en las tres tablas:

AficionesDin/pInsertarAficionUsuarioNuevo.sql

Info

Código SQL disponible en: <https://raw.githubusercontent.com/IISSI-US/silence-db/main/AficionesDin/sql/pInsertarAficionUsuarioNuevo.sql>

Realice el mismo procedimiento pero de forma transaccional:

AficionesDin/pInsertarAficionUsuarioNuevoTrans.sql

i Info

Código SQL disponible en: <https://raw.githubusercontent.com/ISSI-US/silence-db/main/AficionesDin/sql/pInsertarAficionUsuarioNuevoTrans.sql>

Pruebas SQL

Pruebas de aceptación SQL PA-001: Usuarios

1. ☑ Insertar un usuario con datos correctos.
2. ☑ Insertar un usuario sin género.
3. ✗ Insertar un usuario con un email repetido.
4. ✗ Insertar un usuario menor de edad.

Para hacer esta prueba crearemos un procedimiento para insertar un único Usuario, después usamos este procedimiento para insertar los datos de la prueba, teniendo en cuenta que antes hay que hacer la carga inicial de datos.

AficionesDin/pTestUsuario.sql

i Info

Código SQL disponible en: <https://raw.githubusercontent.com/IISSI-US/silence-db/main/AficionesDin/sql/pTestUsuario.sql>

Transacciones

Transacciones Realice un procedimiento para insertar una afición a un usuario nuevo, es decir, que inserte en las tres tablas:

AficionesDin/plInsertarAficionUsuarioNuevo.sql

i Info

Código SQL disponible en: <https://raw.githubusercontent.com/ISSI-US/silence-db/main/AficionesDin/sql/plInsertarAficionUsuarioNuevo.sql>

Realice el mismo procedimiento pero de forma transaccional:

AficionesDin/plInsertarAficionUsuarioNuevoTrans.sql

i Info

Código SQL disponible en: <https://raw.githubusercontent.com/ISSI-US/silence-db/main/AficionesDin/sql/plInsertarAficionUsuarioNuevoTrans.sql>

i Info

Versión PDF disponible