

Espectáculos

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. Entrevista	3
3. Diagrama de clases	5
4. Intensión	6
5. Extensión (fragmento)	7
6. Álgebra relacional	8
7. Modelo Tecnológico (MariaDB)	10
8. Script SQL para crear la base de datos	11
9. Script SQL para la carga inicial de datos	12
10. Consultas	13
11. SQL Avanzado	14

Requisitos

Catálogo de Requisitos La transcripción que aparece a continuación corresponde a una entrevista realizada al gerente de un teatro para determinar los objetivos y requisitos de una aplicación web que gestione la programación anual de espectáculos y venta de entradas a dichos espectáculos.

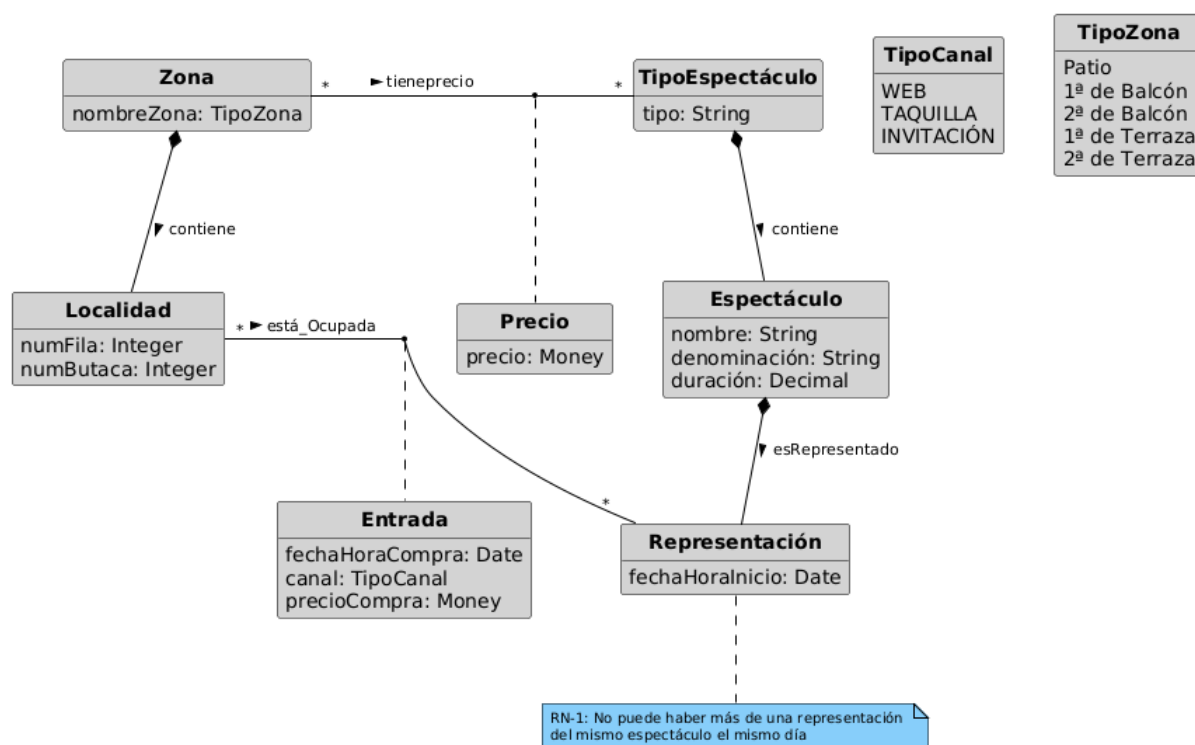
Entrevista

- Cuestión: Para empezar, hableme de los espectáculos. Me interesa conocer la información que debe gestionarse por parte de la aplicación web.
 - Respuesta: De acuerdo, los espectáculos que se programan en nuestro teatro pueden ser de distinto tipo (p.e. “Concierto”, “Opera”, ..). Cada espectáculo es de un tipo.
- Cuestión: ¿Qué información desea recoger de los espectáculos?
 - Respuesta: Básicamente, su denominación, tipo, duración y fecha y hora en las que se representan. Un espectáculo puede tener varias representaciones.
- Cuestión: ¿Puede haber más de una representación el mismo día?
 - Respuesta: Sí, hay días con más de una representación pero nunca del mismo espectáculo.
- Cuestión: ¿Algo más acerca de los espectáculos?
 - Respuesta: Se me olvidaba, también quiero guardar el precio. Cada tipo de espectáculo tiene un precio de la entrada.
- Cuestión: ¿El precio varía según tipo de espectáculo?
 - Respuesta: Sí, pero también depende de la zona del teatro donde esté situada la localidad.
- Cuestión: Acláreme esto, por favor.
 - Respuesta: El teatro está dividido en zonas. Las localidades más caras corresponden a la zona mejor situada, es decir la zona de “Patio”. Otras zonas son “1 de Balcón”, “1 de Terraza” etc. Cada zona tiene distinto precio.
- Cuestión: En definitiva ¿el precio depende del tipo de espectáculo y zona del teatro?
 - Respuesta: Así es. También quiero conocer las localidades vendidas para cada representación.
- Cuestión: ¿Hablamos de las entradas?

- Respuesta: Una entrada es una localidad para una representación. Dentro de cada zona las localidades se identifican por fila y butaca (p.e. “fila 2, butaca 3 de Patio”, “fila 2, butaca 3 de 1 de Balcón”, ...)
- Cuestión: ¿Quiere que se puedan adquirir las entradas únicamente a través de la web?
 - Respuesta: No, seguiremos vendiendo entradas en taquilla. Además algunas entradas se regalan como invitación. Necesito saber el medio o canal por el que se han adquirido las entradas ya sea web, taquilla o invitación.
- Cuestión: ¿Qué información quiere almacenar de las entradas?
 - Respuesta: La fecha y hora de compra, el canal por el que se ha adquirido, representación, fila y butaca que le corresponde y el precio de compra al que se ha adquirido ya que puede variar con el paso del tiempo.
- Cuestión: ¿Algo más?
 - Respuesta: Creo que en principio es todo. Como ya le comenté, queremos que tanto la planificación de espectáculos como la venta de entradas estén gestionados por la aplicación y se puedan consultar a través de Internet.

Modelo conceptual

Diagrama de clases



Modelo relacional

Intensión

```
TiposEspectaculos(tipoEspectaculoId, tipo)
    PK(tipoEspectaculoId)
Zonas(zonaId, nombreZona)
    PK(zonaId)
Precios(precioId, zonaId, tipoEspectaculoId, precio)
    PK(precioId)
    FK(zonaId) / Zonas
    FK(tipoEspectaculoId) / TiposEspectaculos
Localidades(localidadId, zonaId, numFila, numButaca)
    PK(localidadId)
    FK(zonaId) / Zonas
    AK(zonaId, numFila, numButaca)
Espectaculos(espectaculoId, tipoEspectaculoId, nombre, denominacion,
duracion)
    PK(espectaculoId)
    FK(tipoEspectaculoId) / TiposEspectaculos
Representaciones(representacionId, espectaculoId, fechaHoraInicio)
    PK(representacionId)
    FK(espectaculoId) / Espectaculos
    AK(espectaculoId, fechaHoraInicio)
Entradas(entradaId, representacionId, localidadId, fHoraCompra, canal,
pCompra)
    PK(entradaId)
    FK(representacionId) / Representaciones
    FK(localidadId) / Localidades
    AK(representacionId, localidadId)
```

Extensión (fragmento)

```
TiposEspectaculos = { (tel, "Concierto") }
Zonas = { (z1, "Patio"), (z2, "Primera Balcón") }
Precios = { (p1, z1, tel, 50), (p2, z2, tel, 100) }
Localidades = { (l1, z1, 5, 12), (l2, z2, 1, 6) }
Espectaculos = { (e1, tel, "Concierto de ACDC", "Festival", 2.5) }
Representaciones = { (r1, e1, "2024-10-15 20:00"), (r2, e1, "2024-10-16
20:00") }
Entradas = { (en1, r1, l1, "2024-10-10 18:00", "Web", 50), (en2, r2, l2,
"2024-10-11 15:00", "Invitación", 0) }
```

Álgebra relacional

- Precios por zona y tipo:

$$PZT \leftarrow Precios \bowtie Zonas \bowtie TiposEspectaculos$$

- Localidades por zona/tipo/representación:

$$PZTRL \leftarrow PZT \bowtie Localidades$$

- Entradas por representación:

$$numER \leftarrow \gamma_{COUNT(*)} representacionId(Entradas \bowtie Representaciones)$$

- Representaciones con espectáculos:

$$RE \leftarrow Representaciones \bowtie Espectaculos$$

- Número de representaciones por espectáculo:

$$numRE \leftarrow \gamma_{COUNT(*)} espectaculoId(Representaciones \bowtie Espectaculos)$$

- Recaudación por representación:

$$Recaudaciones \leftarrow \gamma_{SUM(pCompra)} representacionId(Entradas)$$

- Localidades vendidas en r3:

$$LocR3 \leftarrow \gamma_{COUNT(localidadId)} representacionId(\sigma_{representacionId=3}(Entradas))$$

- Entradas por invitación:

$$EntradasInvitacion \leftarrow \sigma_{canal=Invitacion} (RE \bowtie Entradas)$$

Modelo Tecnológico (MariaDB)

Modelo tecnológico (MariaDB) Para crear el esquema de la base de datos en MariaDB se puede usar el siguiente script:

Script SQL para crear la base de datos

Espectaculos/createDB.sql

i Info

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

Script SQL para la carga inicial de datos

Para cargar los datos de prueba se puede usar el siguiente script:

Espectaculos/populateDB.sql

i Info

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

Consultas

Para crear las consultas SQL de las expresiones en Álgebra relacional se puede usar el siguiente script:

Espectaculos/queries.sql

i Info

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

SQL Avanzado

Para comprobar que la fecha de compra de una entrada es anterior a la fecha de la representación tenemos que usar triggers. El siguiente script muestra cómo se implementa un procedimiento almacenado que hace la comprobación y después se crean dos triggers, uno que hace la comprobación al insertar y otro al actualizar:

Espectaculos/tFechaCompra.sql

i Info

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

i Info

Versión PDF disponible