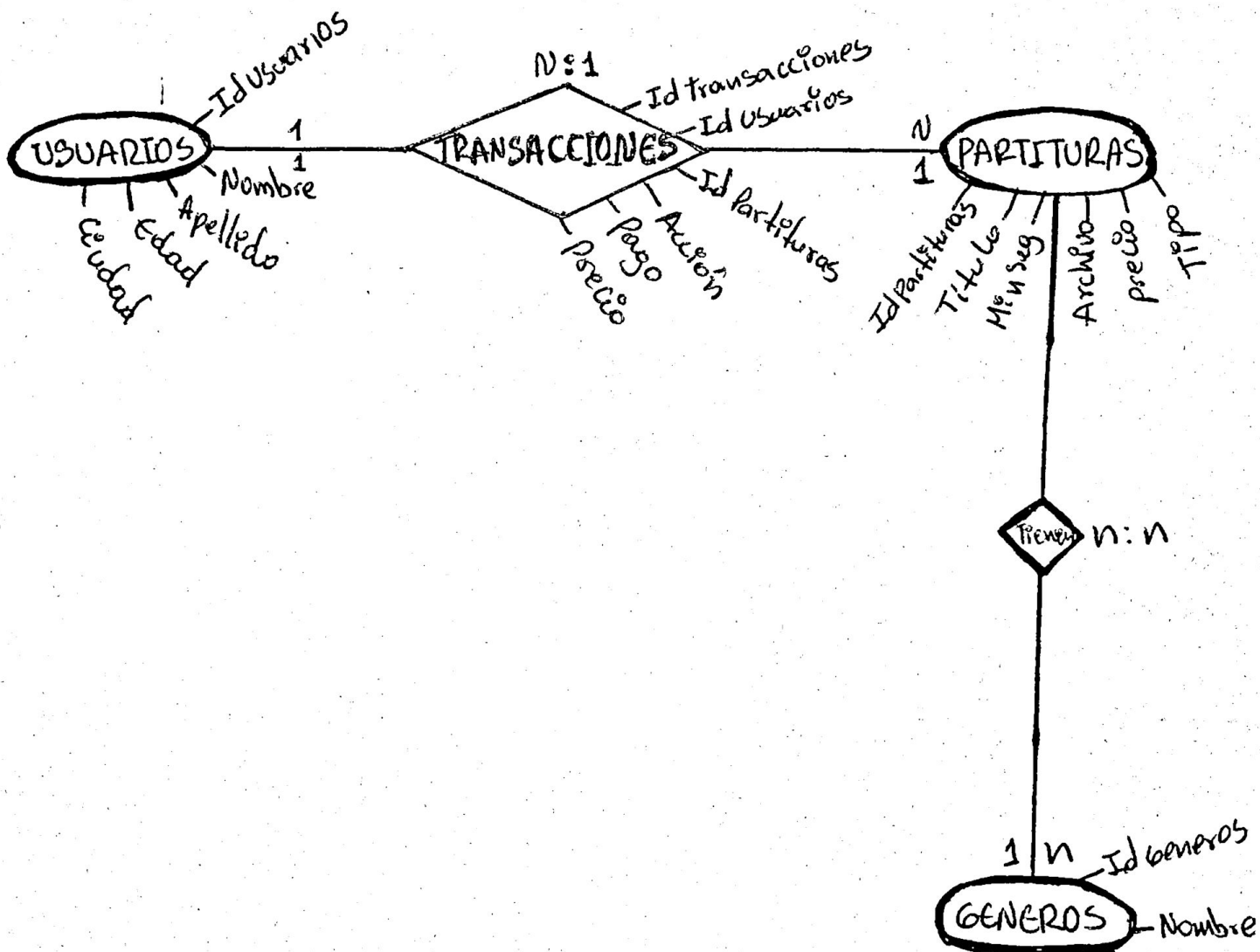


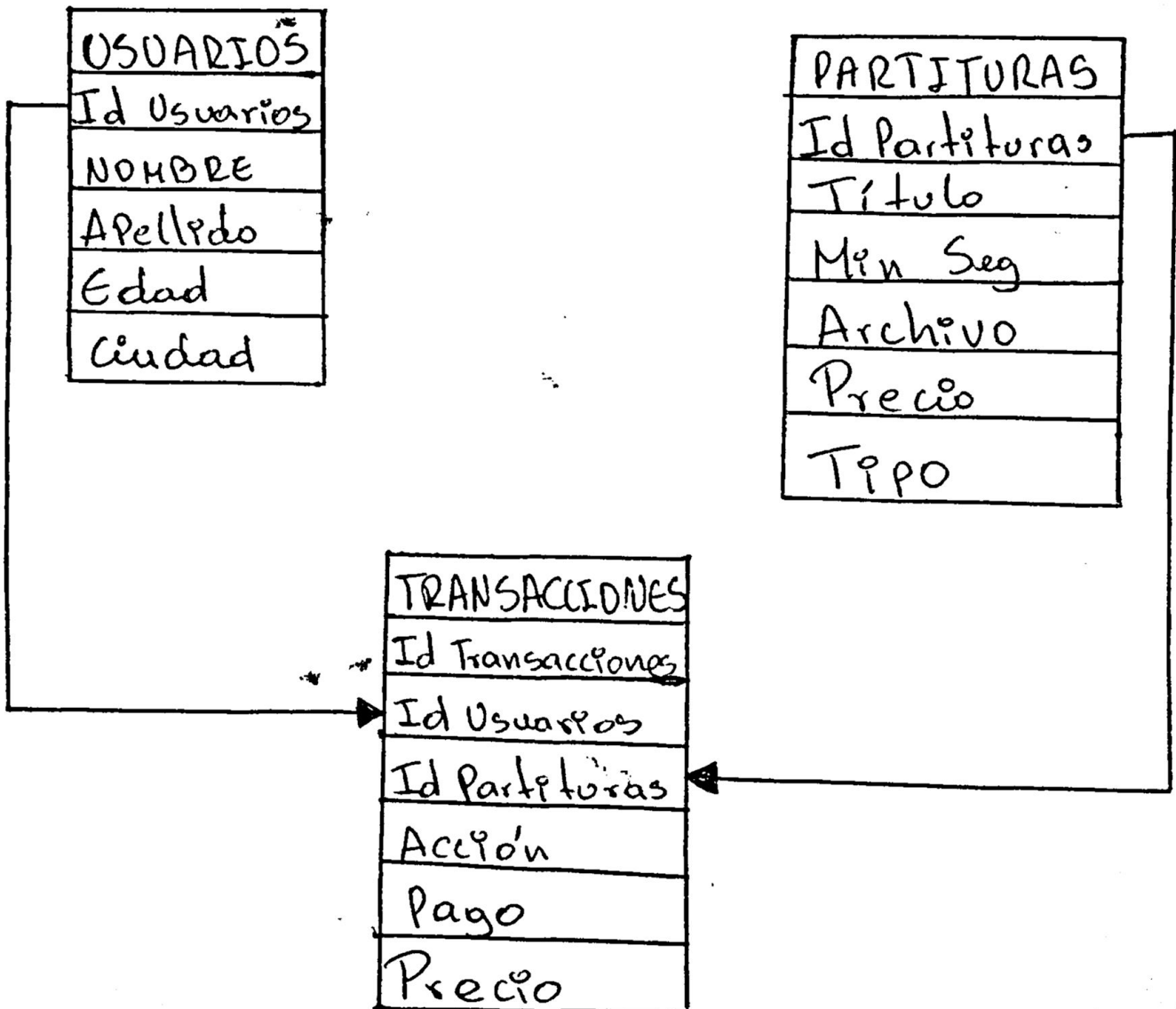
PROPUESTA

La aplicación Coscheapp se dedicará principalmente a la compra y venta de partituras, para ello será necesario un login, cada usuario se registrará con los datos correspondientes. El usuario asume el papel de comprador o vendedor, según requiera el caso, por otro lado cada partitura se mostrará en una lista, junto con 15 seg de audio de la misma, quien este interesado en ella podrá hacer clic y escucharla. Las partituras tendrán datos como el título, duración, tipo, precio, la clasificación tendrán lugar en los comentarios de la página. La app tendrá una sección de preguntas frecuentes donde se expondrán inquietudes y soluciones o dudas correspondientes a la temática musical.

MODELO ENTIDAD RELACION



MODELO RELACIONAL



Consultas

```
Select usuario.nombre, usuario.Apellido, partitura.Titulo, Partitura.Precio from usuario  
INNER JOIN Partitura ON id-partitura = (select  
id-partitura from transaccion where  
id-usuario = usuario.id)  
Where Partitura.Precio > 1.000.000
```

Nombre consulta:

Retorna los nombres de los usuarios cuya ciudad comienza con M

Consulta:

```
Select usuario.nombre from usuario where  
ciudad like '@M'
```

Nombre consulta:

Retorna las partituras con tipo vallenato

Consulta:

```
Select Partitura.Titulo from Partitura  
Where tipo = 'vallenato'
```

Nombre consulta:

Retornar todas las compras mayores o igual a 70 mil

Consulta:

```
Select usuario.nombre, usuario.apellido from
usuario
Where id-usuario (select id-usuario from transaccion
Where id-Partitura = (select id-partitura
from partitura
Where Precio >= 70.000))
```

Nombre Consulta:

Retorna las Partituras Pagadas

Consulta :

```
Select partitura.titulo, partitura.precio from
partitura
Where id-Partitura = (select id-Partitura
from transaccion
Where Accion = 'Pagada')
```

Nombre Consulta:

Retorna datos de compra donde la compra haya sido mayor a 1.000.000 millones

COLECCIONES

1. Usuarios.json

```
[  
  { Id 01,  
    nombre "Carlos",  
    apellido "López",  
    edad "25",  
    Ciudad "Cali"  
  }  
]
```

2. Partituras.json

```
[  
  { Id 002,  
    titulo "La escuela del bagre",  
    Min Seg "4:12",  
    archivo ".pdf",  
    precio "300.000",  
    tipo "partitura de estudio"  
  }  
]
```

3. Transacciones.json

```
[  
  { Id 014,  
    Id Usuarios "01",
```

```
Id Partituras "002",  
accion "comprar",  
pago "transaccion",  
precio "300000"  
}  
]
```

4. Generos.json

```
[  
  { Id 047,  
    nombre "porro"  
  }  
]
```

RUTAS

User Routes

Get/usuarios = todos los usuarios
Get/usuarios/:id = obtiene un único usuario
Post/usuarios/ = guarda un nuevo usuario
Post/UPDATE-usuario/ = actualiza un usuario
Post/Delete-usuario/ = elimina un usuario

Relación Routes

Post/Relación/ = guarda una nueva relación de partitura
con género
Get/Relación/ID-género = selecciona varias partituras por género

Transacciones Routes

Post/transaccion = obtiene registro de transacción
Post/transaccion/comentario = crea comentarios para una
partitura comprada
Get/sellings = obtiene el registro de compras
Get/transacciones = devuelve todos los registros de transaccio-
nes de la base de datos

Partituras Routes

Get/partitura/ = devuelve las partituras de la base de
datos
Get/files = devuelve los archivos de las partituras

Get/partitura/:id = obtiene una única partitura

Get/partitura/:usuario/:prize = devuelve un filtro de partituras por un usuario y un valor de costo

Post/partituras = añade una nueva partitura

Genero Route

Get/generos = obtiene todos los generos de las categorías