**Esquema Relacional**

**Cliente** (NIF, Nome);

{NIF} → {Nome}

**Bilhete** (id, NIF→Cliente);

{id} → {NIF→Cliente}

**BilheteGeral** (id→Bilhete, Preço);

{id→Bilhete} → {Preço}

**BilheteDiário** (id→Bilhete, Preço, Data→Dia);

{id→Bilhete} → {Preço, Data→Dia}

**Dia** (id, Data, HoraInício, HoraTérmino);

{id} → {Data, HoraInício, HoraTérmino}

**Banda** (Nome, Contacto, id→Membro);

{Nome} → {Contacto, id→Membro}

{Contacto} → {Nome}

**Membro** (id, Nome, Nome→Banda);

{id} → {Nome, Nome→Banda}

**Horário**(Dia→Data, Início, Fim, id→Palco, Nome→Banda);

{Dia→Data, id→Palco, Nome→Banda} → {Início, Fim}

**Género** (id, Nome, id→Palco, Nome→Banda);

{id} → {Nome, id→Palco, Nome→Banda}

{Nome→Banda, id→Palco} → {Nome}

**GéneroPalco** (id→Género, id→Palco);

**GéneroBanda** (id→Género, Nome→Banda);

**Palco** (id→Infraestrutura, Nome);

{id→Infraestrutura} → {Nome}

**Infraestrutura** (id, Disponibilidade, Capacidade, Localização);

{id} → {Disponibilidade, Capacidade, Localização}

**WC** (id→Infraestrutura, Chuveiros, Sanitas, Lavatórios);

{id→Infraestrutura} → {Chuveiros, Sanitas, Lavatórios}

**Restauração** (id→Infraestrutura, Restaurantes, Mesas, Cadeiras);

{id→Infraestrutura} → {Restaurantes, Mesas, Cadeiras}

**Campismo** (id → Infraestrutura, Zona, CaixoteLixo);

{id→Infraestrutura} → {Zona, CaixoteLixo}

**Staff** (id, Nome, Turno, Disponibilidade);

id → Nome

Nome → Turno, Disponibilidade

**Sub-Contratado** (id→Staff, Empresa);

{id→Staff} → {Empresa}

**Sponsor** (id → Staff, Empresa);

{id→Staff} → {Empresa}

**Restrições**

Para garantir uma base de dados consistente e acessível, foram usadas várias restrições como a restrição chave, de integridade referencial, NOT NULL e UNIQUE.

O uso de PRIMARY KEY é essencial para qualquer base de dados. Garante a identificação de cada instância da mesma classe.

* Em cada classe há um atributo id que é único e é chave dessa classe. Com o id é possível saber todos os restantes atributos.

A restrição NOT NULL é útil pois há classes que requerem o preenchimento de atributos para essa classe justificar a sua existência.

* Em subclasses como *GeneroPalco*, *GeneroBanda* e *StaffInfrastrutura* em que se os atributos forem nulos a existência destas subclasses deixa de fazer sentido.

Já a restrição UNIQUE é usado para atributos que não sendo chaves primárias são únicos para várias instâncias.

* Na classe *Cliente* o atributo NIF é diferente de qualquer outro cliente.