

Projeto Base de Dados

Relatório - Parte 1

1º semestre

2023/2024

Informática e Gestão de Empresas

Grupo 44

Alexandre Carvoeiro nº110905

Francisco Lopes nº111358

Francisco Rainho nº106312

27 outubro de 2023

1.Índice

2.Introdução	2
3.Desenvolvimento	3
3.1.Diagrama de classes	4
3.2.Eschema relacional	6
3.3.Dicionário de Dados	7
4.Conclusão	10

2.Introdução

O presente trabalho aborda um projeto sobre a UC Bases de Dados, com objetivo da criação de uma base de dados relacional, e o desenvolvimento de um protótipo de uma aplicação web para a manusear. Nesta parte 1, que é o presente trabalho, irá ser abordado uma conceção da base de dados “MUSISYS”, onde será apresentado um diagrama de classes de objetos, um modelo relacional, isto é, um esquema relacional, um dicionário de dados e a sua justificação da estratégia de gestão da integridade referencial e por fim, uma implementação da base de dados em MariaDB.

3.Desenvolvimento

3.1.Diagrama de classes

3.1.1 Análise

Em primeiro lugar, cada **FESTIVAL**, tem um nome, uma localidade, um local, uma data de início, uma data fim, e uma lotação total prevista, sendo que cada um tem uma ou várias **EDIÇÕES** com o seu nº de edição.

Cada **FESTIVAL** poderá ter 1 a 7 **PALCOS**, identificados por um código (primary key) e um nome.

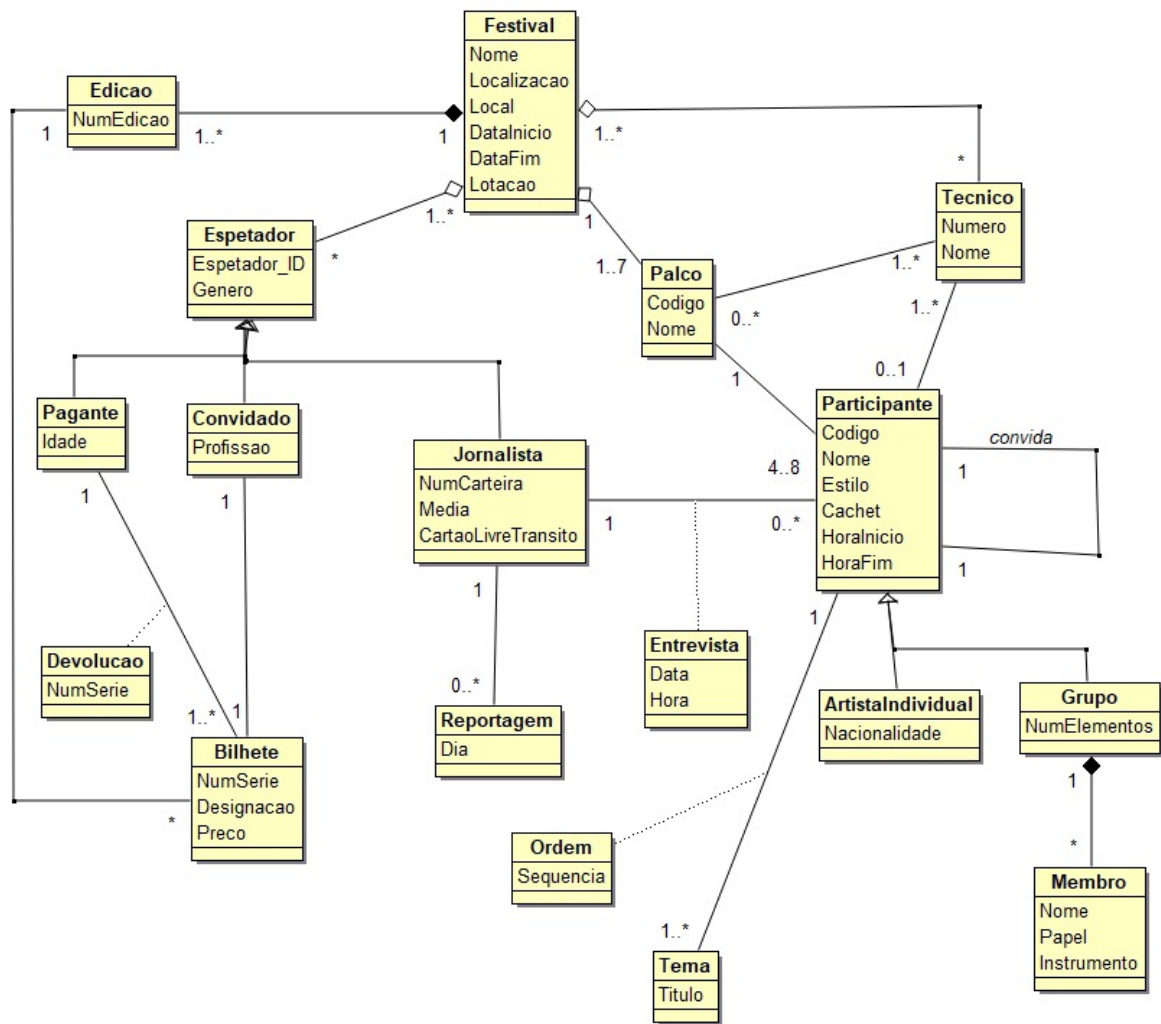
Para a sua realização, o festival deverá ter **TÉCNICOS** identificados por um número (primary key) e por nome. Um **TÉCNICO** poderá estar associado a vários **PALCOS** (Técnico de palco), ou estar associado a um **PARTICIPANTE** (roadies).

Um **PARTICIPANTE**, contém um código identificador(primary key), um nome, um estilo, um cachet que este recebe, hora início e hora fim que este atua. Este poderá ser um **ARTISTA INDIVIDUAL**, onde é necessário registar a sua nacionalidade, ou um **GRUPO**, onde se regista o nº de elementos. Cada elemento do grupo, é representado por uma classe **MEMBRO** que tem como atributos, nome, o seu papel na banda e o instrumento que toca. Para além disso, um **PARTICIPANTE** poderá convidar outro **PARTICIPANTE** e cada um deles apresenta um ou vários **TEMAS**, representados por um título, tendo uma **ORDEM** definida por uma sequência.

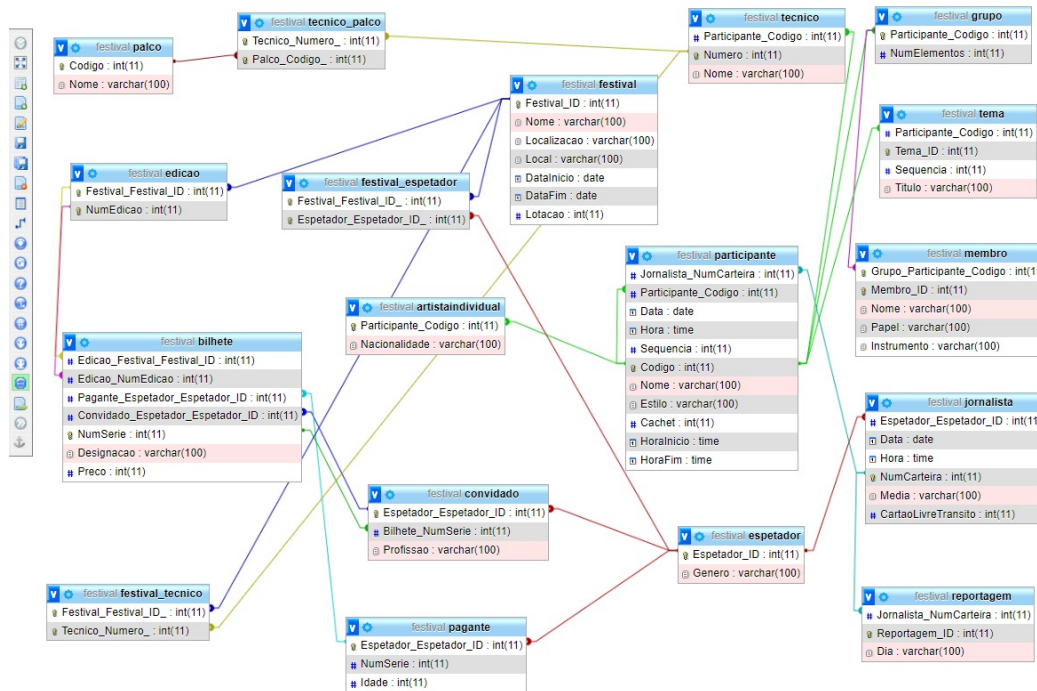
Um **FESTIVAL** contém vários **ESPECTADORES**, caracterizados por um ID de espectador (primary key) e um género. Um **ESPECTADOR** é um **PAGANTE**, ou um **CONVIDADO** ou um **JORNALISTA**. Em primeiro lugar, no **PAGANTE** é necessário registar a idade. Em segundo lugar, no **CONVIDADO** é necessário registar a profissão. Em terceiro lugar, no caso de **JORNALISTA** é necessário registar o seu número de carteira (primary key), o media, e um cartão de livre trânsito. O **JORNALISTA** pode entrevistar vários **PARTICIPANTES** onde se regista, a data e a hora da **ENTREVISTA**. Por outro lado, este também pode fazer várias **REPORTAGENS**, registando-se o dia.

Dentro dos **ESPECTADORES** o **PAGANTE** tem o direito de adquirir um ou mais **BILHETES** associados a uma **EDIÇÃO**. Porém, é importante referir que cada bilhete está associado exclusivamente a um **PAGANTE** ou um **CONVIDADO**, sendo que cada **CONVIDADO** só pode receber um **BILHETE**. Apenas o **PAGANTE** tem a opção de fazer a **DEVOLUÇÃO**, onde guarda o nº de série do bilhete devolvido. Cada **BILHETE** contém um nº de série (primary key), uma designação e um preço.

3.1.2 Implementação



3.2. Esquema relacional



3.3.Dicionário de Dados

3.3.1. Dicionário de Dados

Classe	Atributos	Tipo de Dados	PK	FK	NULL	Default
Participante	Jornalista_NumCarteira	int(11)			Não	Nenhum
Participante	Participante_Codigo	int(11)			Não	Nenhum
Participante	Data	date			Sim	NULL
Participante	Hora	time			Sim	NULL
Participante	Codigo	int(11)	x		Não	Nenhum
Participante	Nome	varchar(100)			Sim	NULL
Participante	Estilo	varchar(100)			Sim	NULL
Participante	Cachet	int(11)			Sim	NULL
Participante	HoraInicio	time			Sim	NULL
Participante	HoraFim	time			Sim	NULL
Participante	Sequencia	int(11)			Sim	NULL
Participante	Participante_noname_Jornalista	int(11)		x	Não	Nenhum
Participante	Participante_convinda_Participante	int(11)		x	Não	Nenhum
Festival	Festival_ID	int(11)	x		Não	Nenhum
Festival	Nome	varchar(100)			Sim	NULL
Festival	Localizacao	varchar(100)			Sim	NULL
Festival	Local	varchar(100)			Sim	NULL
Festival	DataFim	date			Sim	NULL
Festival	DataInicio	date			Sim	NULL
Festival	Lotacao	int(11)			Sim	NULL
Tecnico	Participante_Codigo	int(11)			Sim	NULL
Tecnico	Numero	int(11)	x		Não	Nenhum
Tecnico	Nome	varchar(100)			Sim	NULL
Tecnico	Tecnico_noname_participante	int(11)		x	Não	Nenhum
Grupo	Participante_Codigo	int(11)	x		Não	Nenhum
Grupo	NumElementos	int(11)			Sim	NULL
Tema	Participante_Codigo	int(11)			Não	Nenhum

Tema	Participante_Codigo	int(11)			Não	Nenhum
Tema	Tema_ID	int(11)	x		Não	Nenhum
Tema	Sequencia	int(11)			Sim	NULL
Tema	Título	varchar(100)			Sim	NULL
Tema	Tema_noname_Participante	int(11)		x	Não	Nenhum
Membro	Grupo_Participante_Codigo	int(11)	x		Não	Nenhum
Membro	Membro_ID	int(11)	x		Não	Nenhum
Membro	Nome	varchar(100)			Sim	NULL
Membro	Papel	varchar(100)			Sim	NULL
Membro	Instrumento	varchar(100)			Sim	NULL
Jornalista	Espectador_Espectador_ID	int(11)			Não	Nenhum
Jornalista	NumCarteira	int(11)	x		Não	Nenhum
Jornalista	Media	varchar(100)			Sim	NULL
Jornalista	CartaoLivreTransito	int(11)			Sim	NULL
Jornalista	Data	date			Sim	NULL
Jornalista	Hora	time			Sim	NULL
Jornalista	Jornalista_Espectador	int(11)		x	Não	Nenhum
Reportagem	Jornalista_NumCarteira	int(11)			Não	Nenhum
Reportagem	Reportagem_ID	int(11)	x		Não	Nenhum
Reportagem	Dia	date			Sim	NULL
Reportagem	Reportagem_noname_Jornalista	int(11)		x	Não	Nenhum
Espectador	Espectador_ID	int(11)	x		Não	Nenhum
Espectador	Genero	varchar(100)			Sim	NULL
Convidado	Espectador_Espectador_ID	int(11)	x		Não	Nenhum
Convidado	Bilhete_NumSerie	int(11)			Não	Nenhum
Convidado	Profissao	varchar(100)			Sim	NULL
Convidado	Convidado_noname_Bilhete	int(11)		x	Não	Nenhum
Pagante	Espectador_Espectador_ID	int(11)	x		Não	Nenhum
Pagante	NumSerie	int(11)			Sim	NULL
Pagante	NumSerie	int(11)			Sim	NULL
Pagante	Idade	int(11)			Sim	NULL
Bilhete	Edicao_Festival_Festival_ID	int(11)			Não	Nenhum
Bilhete	Edicao_NumEdicao	int(11)			Não	Nenhum
Bilhete	Pagante_Espectador_Espectador_ID	int(11)			Não	Nenhum
Bilhete	Convidado_Espectador_Espectador_ID	int(11)			Não	Nenhum
Bilhete	NumSerie	int(11)	x		Não	Nenhum
Bilhete	Designacao	varchar(100)			Sim	NULL
Bilhete	Preco	int(11)			Sim	NULL
Bilhete	Bilhete_noname_Edicao	int(11)		x	Não	Nenhum
Bilhete	Bilhete_noname_pagante	int(11)		x	Não	Nenhum
Bilhete	Bilhete_noname_convidado	int(11)		x	Não	Nenhum
ArtistaIndividual	Participante_Codigo	int(11)	x		Não	Nenhum
ArtistaIndividual	Nacionalidade	varchar(100)			Sim	NULL
Edicao	Festival_Festival_ID	int(11)	x		Não	Nenhum
Edicao	NumEdicao	int(11)	x		Não	Nenhum
Festival_Espectador	Festival_Festival_ID	int(11)	x		Não	Nenhum
Festival_Espectador	Espectador_Espectador_ID_	int(11)	x		Não	Nenhum
Festival_Espectador	Espectador_Festival_Espectador_Festival	int(11)		x	Não	Nenhum
Tecnico_Palco	Tecnico_Numero	int(11)	x		Não	Nenhum
Tecnico_Palco	Palco_Codigo	int(11)	x		Não	Nenhum
Tecnico_Palco	Palco_Tecnico_Palco_Tecnico	int(11)		x	Não	Nenhum
Palco	Codigo	int(11)	x		Não	Nenhum
Palco	Nome	varchar(100)			Sim	NULL
Festival_Tecnico	Festival_Festival_ID	int(11)	x		Não	Nenhum
Festival_Tecnico	Tecnico_Numero_	int(11)	x		Não	Nenhum
Festival_Tecnico	Tecnico_Festival_Tecnico_Festival	int(11)		x	Não	Nenhum

3.3.2 Justificação da estratégia de gestão da integridade referencial

A estratégia de gestão da integridade referencial que decidimos adotar na nossa base de dados em relação às chaves estrangeiras foi uma opção de update de natureza “Cascade” e uma opção de delete de natureza “Restrict”.

A opção Update:Cascade permite que as alterações nos valores das chaves primárias sejam propagadas automaticamente para as chaves estrangeiras relacionadas. Esta abordagem é necessária para cenários em que a integridade referencial entre tabelas está interconectada e mudanças em uma tabela devem ser refletidas em outras.

A opção Delete:Restrict é uma restrição aplicada quando se tenta atualizar a chave primária de uma tabela referenciada por uma chave estrangeira, ou seja, impede que se atualize a chave primária caso haja registros de uma chave estrangeira de outra tabela que dependam dessa chave primária. Este procedimento é particularmente útil quando há dependências críticas entre tabelas, onde a remoção de um registro poderia comprometer a validade de outros dados relacionados. Esta opção fornece uma camada adicional de segurança, evitando a exclusão acidental de informações cruciais (como os identificadores dos técnicos, espectadores, participantes do festival) e mantendo a integridade referencial do sistema.

4.Conclusão

O trabalho abrangeu a concepção da base de dados, incluindo a criação de um diagrama de classes de objetos, um modelo relacional (esquema relacional), um dicionário de dados e a justificação da estratégia de gestão da integridade referencial. Além disso, realizou-se a implementação da base de dados no sistema MariaDB.

Esta pesquisa fornece uma sólida estrutura para a organização e gerenciamento de dados no contexto de um sistema de gestão de música, demonstrando a importância da integridade referencial na preservação da qualidade dos dados. No geral, o trabalho representa um passo significativo na direção da criação de uma aplicação web funcional e eficaz para gerenciar a base de dados "MUSISYS" e serve como uma base sólida para projetos futuros relacionados à gestão de festivais de música.