IMPLEMENTAÇÃO DE SERVIÇOS RESTFUL

Bernardo Cirne Severo

Fabricio Felipe de Carvalho

Luiz Alexandre Costa

[**Definições**](#_zfdggj7n9oal) **2**

[**Modelo Lógico:**](#_lx5a51el9rke) **2**

[**Scripts do banco de dados:**](#_mq1x779hzrln) **2**

[**Criação das tabelas**](#_4vj956mufyd2) **2**

[**Script de preenchimento das tabelas:**](#_m45iqndm32jp) **5**

[**Serviços**](#_9pwjrbirvtqw) **6**

[**get\_autores()**](#_4syryroe4vo) **6**

[**get\_autores\_id(id)**](#_tafu5sjykio) **6**

[**post\_autores\_id(id, nome, cpf, citacao)**](#_1evz862v6pwy) **7**

[**update\_autores\_id(id, nome, cpf, citacao)**](#_9wcdnbbvd50h) **7**

[**delete\_autor(id)**](#_ony3ioxmdyzy) **7**

[**get\_artigos()**](#_md7bmvx57uqe) **7**

[**get\_artigos\_id(id)**](#_edqoepx35fhe) **8**

[**post\_artigo\_id(id, nome, tipo)**](#_l27ylu15sidk) **8**

[**update\_artigo\_id(id, nome, tipo)**](#_x1o11udstroj) **8**

[**delete\_artigo(id)**](#_i5r7y0z8088h) **9**

[**get\_artigos\_citacao()**](#_thfycti2zmy8) **9**

[**get\_qualis(id)**](#_5fayysbbvwzl) **9**

[**get\_soma\_qualis(id)**](#_qyocwnvky3yv) **10**

[**get\_pontuacao\_artigo\_qualis(id)**](#_qi1ckjtc2h9k) **10**

# 

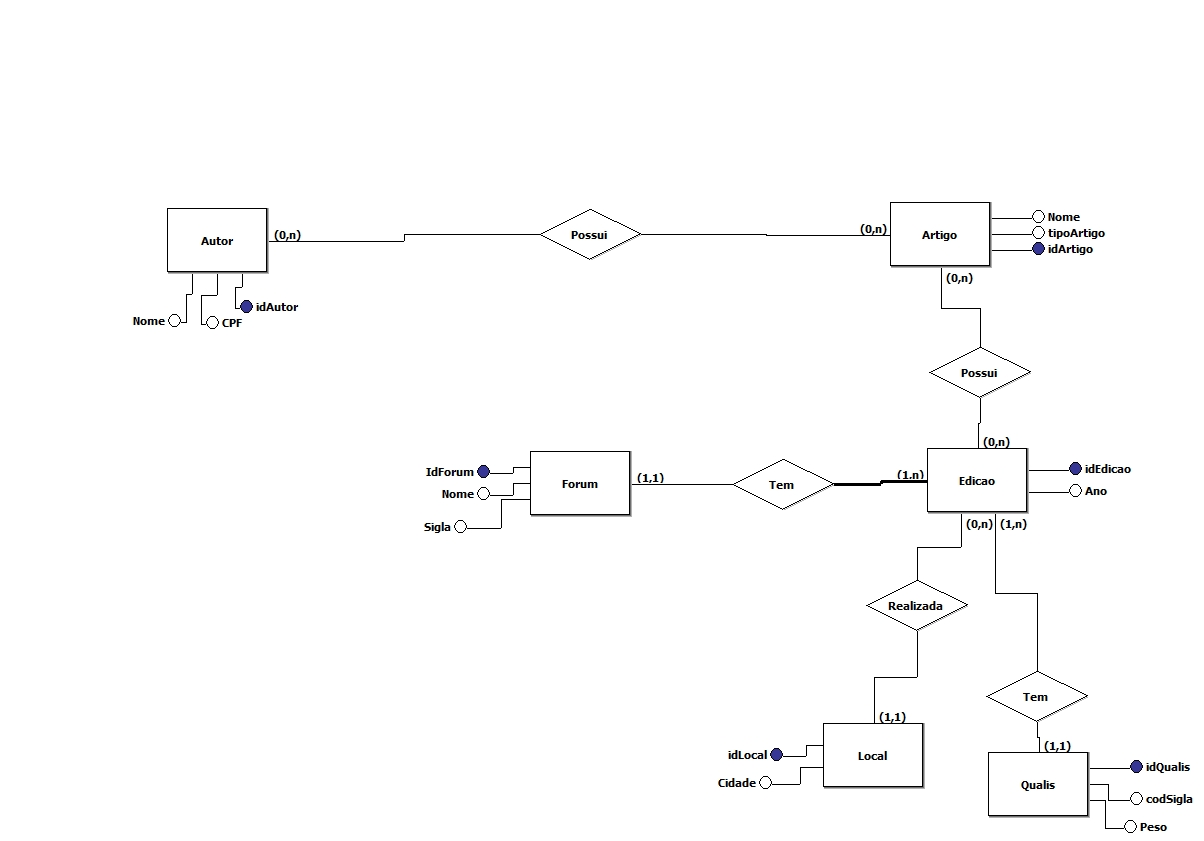
# Definições

Serviços de dados básicos possuem a tarefa de armazenar, modificar e ler dados de múltiplas fontes . Eles costumam ser parte do primeiro conjunto de serviços que escondem detalhes de implantação de um backend específico. Eles são a base para serviços compostos e serviços de processos. São serviços básicos de dados e lógica. Por exemplo: criar um novo usuário, alterar um usuário, retornar a idade de uma pessoa cadastrada, dizer se um ano é bissexto ou não, etc. Pertencem a apenas um tipo de backend.

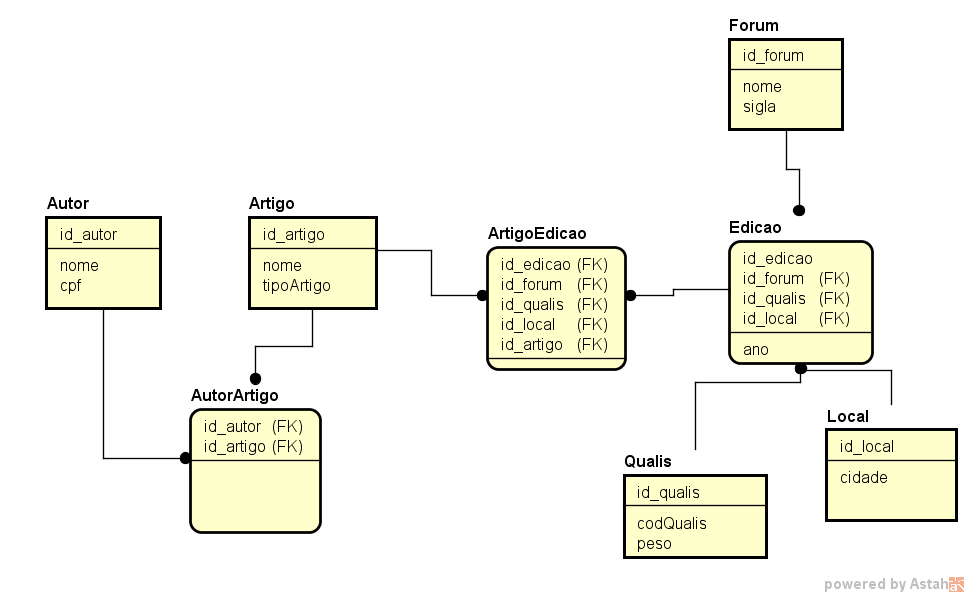
Serviços compostos são serviços criados com base em serviços básicos ou outros serviços compostos.

Serviços de processo são serviços que representam um workflow ou um processo de negócios. Eles representam um fluxo macro que executa um fluxo de atividades (serviços) que não podem ser interrompidos por intervenção humana.

# Modelo Conceitual:



# Modelo Lógico:



# Scripts do banco de dados:

# Criação das tabelas

CREATE TABLE Artigo (

id\_artigo INT NOT NULL,

nome VARCHAR(200),

tipoArtigo VARCHAR(1)

);

ALTER TABLE Artigo ADD CONSTRAINT PK\_Artigo PRIMARY KEY (id\_artigo);

drop table AutorArtigo;

CREATE TABLE Autor (

id\_autor INT NOT NULL,

nome VARCHAR(200),

cpf VARCHAR(11) not null,

citacao VARCHAR(50) NULL

);

ALTER TABLE Autor ADD CONSTRAINT PK\_Autor PRIMARY KEY (id\_autor);

CREATE TABLE AutorArtigo (

id\_autor INT NOT NULL,

id\_artigo INT NOT NULL

);

ALTER TABLE AutorArtigo ADD CONSTRAINT PK\_AutorArtigo PRIMARY KEY (id\_autor,id\_artigo);

CREATE TABLE Forum (

id\_forum INT NOT NULL,

nome VARCHAR(200),

sigla VARCHAR(10)

);

ALTER TABLE Forum ADD CONSTRAINT PK\_Forum PRIMARY KEY (id\_forum);

CREATE TABLE Local (

id\_local INT NOT NULL,

cidade VARCHAR(50)

);

ALTER TABLE Local ADD CONSTRAINT PK\_Local PRIMARY KEY (id\_local);

CREATE TABLE Qualis (

id\_qualis INT NOT NULL,

codQualis VARCHAR(2),

peso INT

);

ALTER TABLE Qualis ADD CONSTRAINT PK\_Qualis PRIMARY KEY (id\_qualis);

CREATE TABLE Edicao (

id\_edicao INT NOT NULL,

id\_forum INT NOT NULL,

id\_qualis INT NOT NULL,

id\_local INT NOT NULL,

ano INT

);

ALTER TABLE Edicao ADD CONSTRAINT PK\_Edicao PRIMARY KEY (id\_edicao,id\_forum,id\_qualis,id\_local);

CREATE TABLE ArtigoEdicao (

id\_edicao INT NOT NULL,

id\_forum INT NOT NULL,

id\_qualis INT NOT NULL,

id\_local INT NOT NULL,

id\_artigo INT NOT NULL

);

ALTER TABLE ArtigoEdicao ADD CONSTRAINT PK\_ArtigoEdicao PRIMARY KEY (id\_edicao,id\_forum,id\_qualis,id\_local,id\_artigo);

ALTER TABLE AutorArtigo ADD CONSTRAINT FK\_AutorArtigo\_0 FOREIGN KEY (id\_autor) REFERENCES Autor (id\_autor);

ALTER TABLE AutorArtigo ADD CONSTRAINT FK\_AutorArtigo\_1 FOREIGN KEY (id\_artigo) REFERENCES Artigo (id\_artigo);

ALTER TABLE Edicao ADD CONSTRAINT FK\_Edicao\_0 FOREIGN KEY (id\_forum) REFERENCES Forum (id\_forum);

ALTER TABLE Edicao ADD CONSTRAINT FK\_Edicao\_1 FOREIGN KEY (id\_qualis) REFERENCES Qualis (id\_qualis);

ALTER TABLE Edicao ADD CONSTRAINT FK\_Edicao\_2 FOREIGN KEY (id\_local) REFERENCES Local (id\_local);

ALTER TABLE ArtigoEdicao ADD CONSTRAINT FK\_ArtigoEdicao\_0 FOREIGN KEY (id\_edicao,id\_forum,id\_qualis,id\_local) REFERENCES Edicao (id\_edicao,id\_forum,id\_qualis,id\_local);

ALTER TABLE ArtigoEdicao ADD CONSTRAINT FK\_ArtigoEdicao\_1 FOREIGN KEY (id\_artigo) REFERENCES Artigo (id\_artigo);

# Script de preenchimento das tabelas:

INSERT INTO Artigo VALUES (1, 'A Cloud-based Architecture for the Internet of Things Targeting Industrial Engine Remote Monitoring', 'C');

INSERT INTO Artigo VALUES (2, 'ResearchOps: The case for DevOps in scientific applications', 'C');

INSERT INTO Artigo VALUES (3, 'Architecting Cloud Tools using Software Product Line Techniques', 'R');

INSERT INTO Autor VALUES (1, 'Leonardo Azevedo', '12345678901','AZEVEDO,L');

INSERT INTO Autor VALUES (2, 'Luiz Alexandre', '04303590711', 'COSTA,L,A');

INSERT INTO Autor VALUES (3, 'Fabricio Felipe', '10987654321', 'CARVALHO,F');

INSERT INTO AutorArtigo VALUES (1, 1);

INSERT INTO AutorArtigo VALUES (1, 2);

INSERT INTO AutorArtigo VALUES (2, 3);

INSERT INTO AutorArtigo VALUES (3, 3);

INSERT INTO Forum VALUES (1, 'Sensors to Cloud Architectures Workshop', 'SCAW');

INSERT INTO Forum VALUES (2, 'IFIP/IEEE International Symposium', 'IM');

INSERT INTO Forum VALUES (3, '30th ACM/SIGAPP Symposium', 'SIGAPP');

INSERT INTO Local VALUES (1, 'San Francisco');

INSERT INTO Local VALUES (2, 'Ottawa');

INSERT INTO Local VALUES (3, 'Salamanca');

INSERT INTO Qualis VALUES (1, 'A1', 30);

INSERT INTO Qualis VALUES (2, 'A2', 26);

INSERT INTO Qualis VALUES (3, 'B1', 20);

INSERT INTO Qualis VALUES (4, 'B2', 16);

INSERT INTO Qualis VALUES (5, 'B3', 14);

INSERT INTO Edicao VALUES (1, 1, 1, 1, 2015);

INSERT INTO Edicao VALUES (2, 2, 2, 2, 2014);

INSERT INTO Edicao VALUES (3, 3, 3, 3, 2016);

INSERT INTO ArtigoEdicao VALUES (1, 1, 1, 1, 1);

INSERT INTO ArtigoEdicao VALUES (2, 2, 2, 2, 2);

INSERT INTO ArtigoEdicao VALUES (3, 3, 3, 3, 3);

# Serviços

# get\_autores()

Endpoint: /trab05/autores

Função: Obter as informações de todos os autores armazenados no banco de dados.

Json:

[  
 [  
 1,   
 "Leonardo Azevedo",   
 "12345678901",   
 "AZEVEDO,L"  
 ],   
 [  
 2,   
 "Luiz Alexandre",   
 "04303590711",   
 "COSTA,L,A"  
 ],   
 [  
 3,   
 "Fabricio Felipe",   
 "10987654321",   
 "CARVALHO,F"  
 ]  
]

# get\_autores\_id(id)

Endpoint: /trab05/autores/<int:id>

Função: Obter as informações armazenadas no banco de dados de determinado autor.

Json:

[  
 [  
 1,   
 "Leonardo Azevedo",   
 "12345678901",   
 "AZEVEDO,L"  
 ]  
]

# post\_autores\_id(id, nome, cpf, citacao)

Endpoint: /trab05/autores/<int:id>/<string:nome>/<string:cpf>/<string:citacao>

Função: Inserir um novo autor no banco de dados.

Json:

{

"result": true

}

# update\_autores\_id(id, nome, cpf, citacao)

Endpoint: /trab05/autores/<int:id>/<string:nome>/<string:cpf>/<string:citacao>

Função: Atualizar as informações de determinado autor.

Json:

{

"result": true

}

# delete\_autor(id)

Endpoint: /trab05/autores/<int:id>

Função: Excluir determinado autor do banco de dados.

Json:

{

"result": true

}

# get\_artigos()

Endpoint: /trab05/artigos

Função: Obter as informações armazenadas no banco de dados de todos os artigos.

Json:

[

[

1,

"A Cloud-based Architecture for the Internet of Things Targeting Industrial Engine Remote Monitoring",

"C"

],

[

2,

"ResearchOps: The case for DevOps in scientific applications",

"C"

],

[

3,

"Architecting Cloud Tools using Software Product Line Techniques",

"R"

]

]

# get\_artigos\_id(id)

Endpoint: /trab05/artigos/<int:id>

Função: Obter as informações armazenadas no banco de dados de um artigo.

Json:

[

[

1,

"A Cloud-based Architecture for the Internet of Things Targeting Industrial Engine Remote Monitoring",

"C"

]

]

# post\_artigo\_id(id, nome, tipo)

Endpoint: /trab05/artigos/<int:id>/<string:nome>/<string:tipo>

Função:Cadastra um novo artigo no banco de dados.

Json:

{

"result": true

}

# update\_artigo\_id(id, nome, tipo)

Endpoint: /trab05/artigos/<int:id>/<string:nome>/<string:tipo>

Função: Autualiza as informações de um artigo cadastrado no banco de dados.

Json:

{

"result": true

}

# delete\_artigo(id)

Endpoint: /trab05/artigos/<int:id>

Função: Exclui um artigo do banco de dados.

Json:

{

"result": true

}

# get\_artigos\_citacao()

Endpoint: trab05/artigos/citacao

Função: Retorna as citações de todos os artigos.

Json:

[

[

"AZEVEDO,L",

"A Cloud-based Architecture for the Internet of Things Targeting Industrial Engine Remote Monitoring",

"Sensors to Cloud Architectures Workshop (SCAW)",

2015

],

[

"AZEVEDO,L",

"ResearchOps: The case for DevOps in scientific applications",

"IFIP/IEEE International Symposium (IM)",

2014

],

[

"CARVALHO,F",

"Architecting Cloud Tools using Software Product Line Techniques",

"30th ACM/SIGAPP Symposium (SIGAPP)",

2016

],

[

"COSTA,L,A",

"Architecting Cloud Tools using Software Product Line Techniques",

"30th ACM/SIGAPP Symposium (SIGAPP)",

2016

]

]

# get\_qualis(id)

Endpoint: /trab05/qualis/<int:id>

Função: Retorna os artigos com as qualis.

Json:

[

[

"A Cloud-based Architecture for the Internet of Things Targeting Industrial Engine Remote Monitoring",

"A1"

],

[

"ResearchOps: The case for DevOps in scientific applications",

"A2"

]

]

# get\_soma\_qualis(id)

Endpoint: /trab05/pontuacao/autor/<int:id>

Função: Retorna a soma dos valores da qualis do autor selecionado.

Json:

[

"Leonardo Azevedo",

56

]

# get\_pontuacao\_artigo\_qualis(id)

Endpoint: /trab05/pontuacaoartigo/autor/<int:id>

Função: Retorna a soma dos valores da qualis utilizando os serviços: retorna\_pontuacao\_autor(id) e consulta\_autor\_id(id).

Json:

[

[

[

1,

"Leonardo Azevedo",

"12345678901",

"AZEVEDO,L"

]

],

[

"Leonardo Azevedo",

56

]

]