

# Inserindo dados

Para inserirmos dados às tabelas criadas nos passos anteriores, necessitamos implementar estruturas com o objetivo de transportar esses dados à API. Essas estruturas corresponderão aos tipos de informação que utilizamos na linguagem Swift. Crie um arquivo Swift com o nome Model e implemente o código a seguir:

```
struct Banda {  
    var idBanda: Int  
    var nome: String  
    var colunas: [Any] { [ idBanda, nome ] }  
}  
  
struct Estilo {  
    var idEstilo: Int  
    var nome: String  
    var colunas: [Any] { [ idEstilo, nome ] }  
}
```

```
struct Album {
    var idAlbum: Int
    var nome: String
    var ano: Int
    var banda: Banda
    var estilo: Estilo
    var colunas: [Any] {
        [ idAlbum, nome, ano, banda.idBanda,
estilo.idEstilo ] }
}

struct Musica {
    var idMusica: Int
    var nome: String
    var estilo: Estilo
    var banda: Banda
    var album: Album
    var colunas: [Any] {
        [ banda.idBanda, idMusica, nome,
estilo.idEstilo ] }
    var join: [Any] { [ banda.idBanda, idMusica,
album.idAlbum ] }
}
```

As estruturas implementadas contêm as propriedades correspondentes aos atributos das tabelas acrescidas de uma propriedade que representa as colunas da tabela. A estrutura **Musica**, além de todas as propriedades citadas, contém a propriedade que representa a junção das tabelas Banda e Álbum. Essas propriedades extras servirão de apoio ao processamento para o método da API que efetuará a gravação dos dados contidos nessas estruturas para as tabelas correspondentes.

Com as estruturas criadas, implementaremos o método na classe `CommonUtils`, que as receberá e efetuará a gravação no banco de dados:

```
func executaQuery(_ query: Query) {
    let thread = DispatchGroup()
    thread.enter()
    if let connection = getConnection() {
        connection.execute(query: query) { result in
            var nomeQuery = String(describing:
type(of:query))
            if nomeQuery == "Raw" {
                nomeQuery = String(describing: query.self)
                    .split(separator:
"\\"") [1]
                    .split(separator: "
") [0]
                    .capitalized
            }

            if let erro = result.asError {
                print("\(nomeQuery), Falha na execução:
\erro)")
            }
            thread.leave()
        }
    } else {
        print("Sem conexão")
        thread.leave()
    }
    thread.wait()
}
```

Esse método obtém a conexão ao banco de dados e executa a transação de maneira assíncrona. Na ocorrência de falha, uma mensagem será apresentada. A parte central desse método identifica o nome da execução SQL para identifica-lo na mensagem de erro.

Vamos iniciar a construção da função responsável por criar as listas que conterão as informações e chamar o método criado no passo anterior para gravar esses dados no banco de dados. Crie um arquivo Swift com o nome `InserirDados` e implemente o código a seguir:

```
import Foundation
import SwiftKuery

func insereDados() {
    let utils = CommonUtils.sharedInstance
}
```

A função `insereDados` conterá toda a lógica necessária para inserir os dados. Foi declarada uma constante `utils`, que contém a referência à instância da classe que implementa a nossa API para o acesso ao MySQL. Vamos construir a lista de bandas para inseri-las na tabela correspondente, vejamos:

```
func insereDados() {
    let utils = CommonUtils.sharedInstance

    let bandas = [
        Banda(idBanda: 1, nome: "Tecnimatic"),
        Banda(idBanda: 2, nome: "Trap"),
        Banda(idBanda: 3, nome: "Alix Perez"),
        Banda(idBanda: 4, nome: "Maduk"),
        Banda(idBanda: 5, nome: "AudioSketch"),
        Banda(idBanda: 6, nome: "The Ghost Inside")
    ]

    utils.executaQuery(Insert(into: Bandas(),
                               rows: bandas.map{
$0.colunas })))
    print("Os registros de Bandas foram inseridos")
}
```

A constante `bandas` é um array que contém a lista das bandas que serão cadastradas na tabela `Banda`. Cada elemento da lista é uma instância da estrutura `Banda` e contém os dados relativos a cada registro a ser cadastrado. Na chamada ao método `executaQuery`, uma instância de `Insert` é passada como parâmetro, representando a ação que desejamos executar no banco de dados. Nessa instância, uma instancia da classe `Bandas` é passada como argumento para definir qual é a tabela e atributos a serem utilizados no processamento da informação. Além disso, a instância da lista que contém todas as instâncias também é entregue com os dados para cada registro.

```
func insereDados() {
    let utils = CommonUtils.sharedInstance

    let bandas = [
        Banda(idBanda: 1, nome: "Tecnimatic"),
        Banda(idBanda: 2, nome: "Trap"),
        Banda(idBanda: 3, nome: "Alix Perez"),
        Banda(idBanda: 4, nome: "Maduk"),
        Banda(idBanda: 5, nome: "AudioSketch"),
        Banda(idBanda: 6, nome: "The Ghost Inside")
    ]

    utils.executaQuery(Insert(into: Bandas(),
                             rows: bandas.map{ $0.colunas
}))
    print("Os registros das Bandas foram inseridos")

    let estilos = [
        Estilo(idEstilo: 1, nome: "Rock"),
        Estilo(idEstilo: 2, nome: "Dance"),
        Estilo(idEstilo: 3, nome: "Drum`n`bass"),
        Estilo(idEstilo: 4, nome: "Jungle")
    ]

    utils.executaQuery(Insert(into: Estilos(),
                             rows: estilos.map{
$0.colunas })))
    print("Os registros de Estilos foram inseridos")
}
```

```
let albuns = [  
    Album(idAlbum: 1, nome: "Empty words", ano: 2021,  
          banda: bandas[2], estilo: estilos[3]),  
    Album(idAlbum: 2, nome: "Simplicity", ano: 2021,  
          banda: bandas[4], estilo: estilos[1]),  
    Album(idAlbum: 3, nome: "Rise from the Ashe", ano:  
2021,  
          banda: bandas[5], estilo: estilos[0]),  
    Album(idAlbum: 4, nome: "Stay Like This", ano:  
2021,  
          banda: bandas[3], estilo: estilos[0]),  
    Album(idAlbum: 5, nome: "walk to you", ano: 2021,  
          banda: bandas[0], estilo: estilos[2]),  
    Album(idAlbum: 6, nome: "Above it All", ano: 2021,  
          banda: bandas[1], estilo: estilos[0])  
]  
  
utils.executaQuery(Insert(into: Albuns(),  
                          rows: albuns.map{ $0.colunas })))  
print("Os registros de \ (Albuns().nameInQuery) foram  
inseridos")
```

```
let musicas = [  
  Musica(idMusica: 1, nome: "Above it All",  
    estilo: estilos[0], banda: bandas[1],  
    album: albuns[5]),  
  Musica(idMusica: 2, nome: "Avalanche",  
    estilo: estilos[0], banda: bandas[5],  
    album: albuns[2]),  
  Musica(idMusica: 3, nome: "Between the Lines",  
    estilo: estilos[0], banda: bandas[5],  
    album: albuns[2]),  
  Musica(idMusica: 4, nome: "Burning Babylon",  
    estilo: estilos[3], banda: bandas[2],  
    album: albuns[0]),  
  Musica(idMusica: 5, nome: "Dear Youth",  
    estilo: estilos[0], banda: bandas[5],  
    album: albuns[2]),  
  Musica(idMusica: 6, nome: "Devotion",  
    estilo: estilos[1], banda: bandas[4],  
    album: albuns[1]),  
  Musica(idMusica: 7, nome: "Empty words",  
    estilo: estilos[3], banda: bandas[2],  
    album: albuns[0]),  
  Musica(idMusica: 8, nome: "Everlasting",  
    estilo: estilos[2], banda: bandas[0],  
    album: albuns[4]),  
  Musica(idMusica: 9, nome: "Greater Distance",  
    estilo: estilos[0], banda: bandas[5],  
    album: albuns[2])  
]  
  
utils.executaQuery(Insert(into: Musicas(),  
  rows: musicas.map{ $0.colunas })))  
print("Os registros de Musicas foram inseridos")  
}
```

O código destacado apresenta a construção das instâncias de listas para estilos, álbuns e músicas, cada qual com seus respectivos dados. Notamos também que nas instâncias `Album` e `Musica`, as referências a `Banda` e `Estilo` são passadas como parâmetros, em razão das associações existentes entre essas estruturas e que devem ser refletidas nas tabelas no banco de dados.

```
func insereDados() {
    let utils = CommonUtils.sharedInstance

    let bandas = [
        Banda(idBanda: 1, nome: "Tecnimatic"),
        Banda(idBanda: 2, nome: "Trap"),
        Banda(idBanda: 3, nome: "Alix Perez"),
        Banda(idBanda: 4, nome: "Maduk"),
        Banda(idBanda: 5, nome: "AudioSketch"),
        Banda(idBanda: 6, nome: "The Ghost Inside")
    ]

    utils.executaQuery(Insert(into: Bandas(),
                             rows: bandas.map{ $0.colunas
    }))
    print("Os registros de Bandas foram inseridos")

    let estilos = [
        Estilo(idEstilo: 1, nome: "Rock"),
        Estilo(idEstilo: 2, nome: "Dance"),
        Estilo(idEstilo: 3, nome: "Drum`n`bass"),
        Estilo(idEstilo: 4, nome: "Jungle")
    ]
}
```

```
utils.executaQuery(Insert(into: Estilos(),
                          rows: estilos.map{
$0.colunas })))
print("Os registros de Estilos foram inseridos")

let albuns = [
    Album(idAlbum: 1, nome: "Empty words", ano: 2021,
          banda: bandas[2], estilo: estilos[3]),
    Album(idAlbum: 2, nome: "Simplicity", ano: 2021,
          banda: bandas[4], estilo: estilos[1]),
    Album(idAlbum: 3, nome: "Rise from the Ashe",
ano: 2021,
          banda: bandas[5], estilo: estilos[0]),
    Album(idAlbum: 4, nome: "Stay Like This", ano:
2021,
          banda: bandas[3], estilo: estilos[0]),
    Album(idAlbum: 5, nome: "Walk to you", ano: 2021,
          banda: bandas[0], estilo: estilos[2]),
    Album(idAlbum: 6, nome: "Above it All", ano:
2021,
          banda: bandas[1], estilo: estilos[0])
]

utils.executaQuery(Insert(into: Albuns(),
                          rows: albuns.map{ $0.colunas })))
print("Os registros de \(Albuns().nameInQuery) foram
inseridos")
```

```
let musicas = [  
  Musica(idMusica: 1, nome: "Above it All",  
    estilo: estilos[0], banda: bandas[1],  
    album: albuns[5]),  
  Musica(idMusica: 2, nome: "Avalanche",  
    estilo: estilos[0], banda: bandas[5],  
    album: albuns[2]),  
  Musica(idMusica: 3, nome: "Between the Lines",  
    estilo: estilos[0], banda: bandas[5],  
    album: albuns[2]),  
  Musica(idMusica: 4, nome: "Burning Babylon",  
    estilo: estilos[3], banda: bandas[2],  
    album: albuns[0]),  
  Musica(idMusica: 5, nome: "Dear Youth",  
    estilo: estilos[0], banda: bandas[5],  
    album: albuns[2]),  
  Musica(idMusica: 6, nome: "Devotion",  
    estilo: estilos[1], banda: bandas[4],  
    album: albuns[1]),  
  Musica(idMusica: 7, nome: "Empty words",  
    estilo: estilos[3], banda: bandas[2],  
    album: albuns[0]),  
  Musica(idMusica: 8, nome: "Everlasting",  
    estilo: estilos[2], banda: bandas[0],  
    album: albuns[4]),  
  Musica(idMusica: 9, nome: "Greater Distance",  
    estilo: estilos[0], banda: bandas[5],  
    album: albuns[2])  
]
```

```
utils.executaQuery(Insert(into: Musicas(),
                        rows: musicas.map{ $0.colunas }))
print("Os registros de Musicas foram inseridos")

utils.executaQuery(Insert(into: MusicasDosAlbuns(),
                        rows: musicas.map{ $0.join }))
print("Os registros de Musicas dos Albuns foram
inseridos")
}
```

Por fim, a última execução permite inserir as associações N para N entre músicas e álbuns (dessa vez utilizando a instância da classe `MusicasDosAlbuns`), e a lista de instâncias de `Musica` que traz em sua estrutura as relações necessárias para atender aos requisitos de associação entre as duas entidades.

Agora vamos alterar o programa main para inserir os dados nas tabelas:

```
import Foundation

//criaTabelas()
insereDados()
```

Comentamos a chamada à função `criaTabelas` para que não ocorra tentativa de recriação das tabelas e adicionamos a chamada à função `insereDados`. Assim, a saída será:

```
Os registros de Bandas foram inseridos
Os registros de Estilos foram inseridos
Os registros de Album foram inseridos
Os registros de Musicas foram inseridos
Os registros de Musicas dos Albuns foram inseridos
```