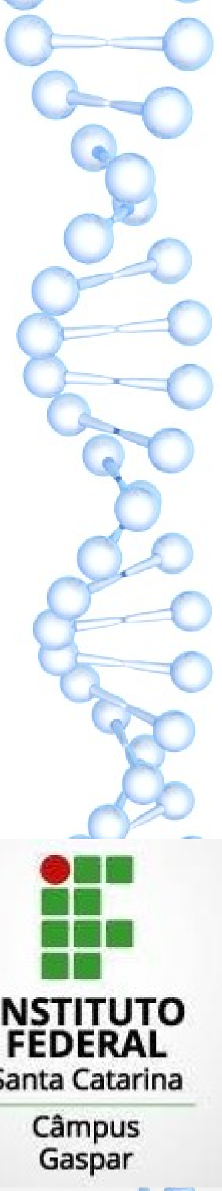


Marcos Rodrigo Momo, M.Sc.
e-mail: marcos.momo@ifsc.edu.br

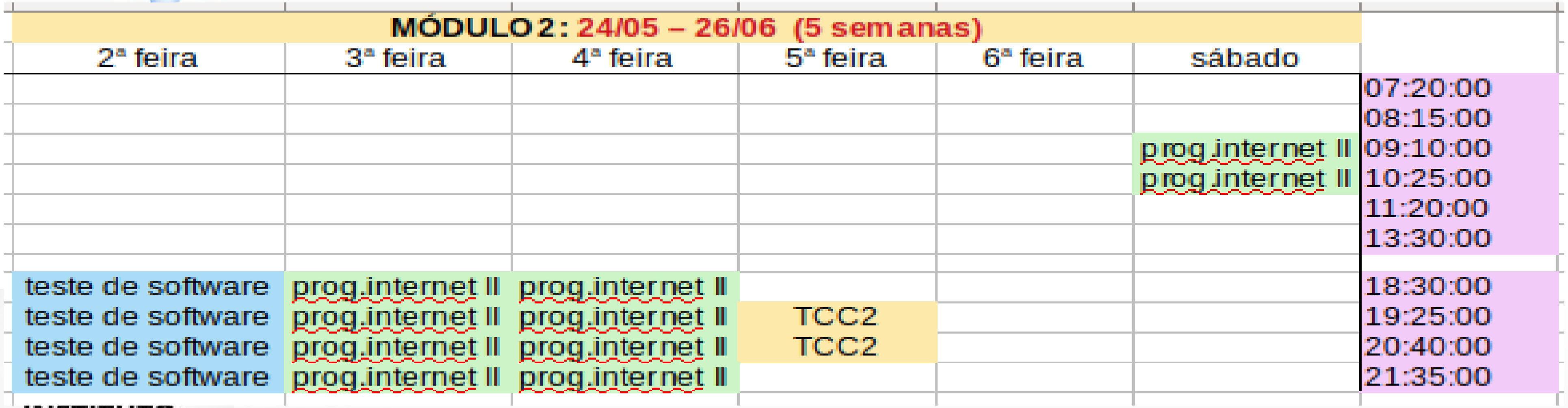
Programação para Internet II

INSTITUTO FEDERAL
Santa Catarina
Câmpus
Gaspar

Gaspar, maio 2021.

Cronograma das aulas módulo 2

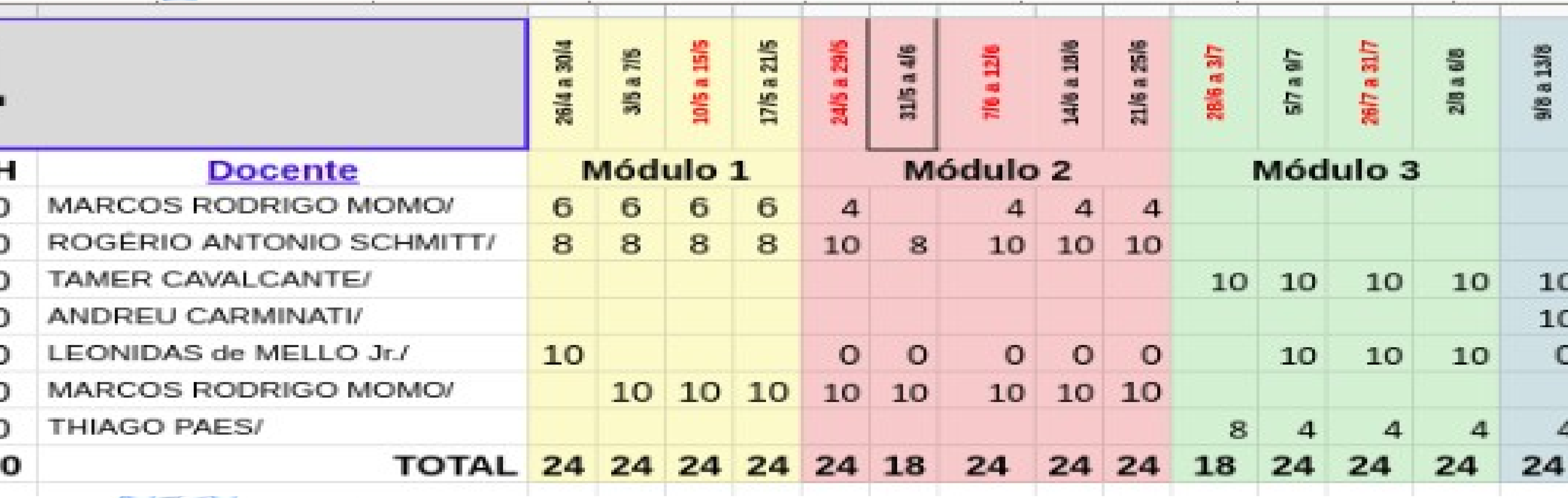
Observação: não termos mais aula no sábado
5 semanas 15 encontros



MÓDULO 2: 24/05 – 26/06 (5 semanas)						
2ª feira	3ª feira	4ª feira	5ª feira	6ª feira	sábado	
						07:20:00
						08:15:00
					prog.internet II	09:10:00
					prog.internet II	10:25:00
						11:20:00
						13:30:00
teste de software	prog.internet II	prog.internet II				18:30:00
teste de software	prog.internet II	prog.internet II	TCC2			19:25:00
teste de software	prog.internet II	prog.internet II	TCC2			20:40:00
teste de software	prog.internet II	prog.internet II				21:35:00

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Calendário ADS 4



		26/4 a 30/4	3/5 a 7/5	10/5 a 15/5	17/5 a 21/5	24/5 a 29/5	31/5 a 4/6	7/6 a 12/6	14/6 a 18/6	21/6 a 25/6	28/6 a 3/7	5/7 a 9/7	26/7 a 31/7	2/8 a 6/8	9/8 a 13/8
H	Docente	Módulo 1				Módulo 2					Módulo 3				
0	MARCOS RODRIGO MOMO/	6	6	6	6	4		4	4	4					
0	ROGÉRIO ANTONIO SCHMITT/	8	8	8	8	10	8	10	10	10					
0	TAMER CAVALCANTE/										10	10	10	10	10
0	ANDREU CARMINATI/														10
0	LEONIDAS de MELLO Jr./	10				0	0	0	0	0		10	10	10	0
0	MARCOS RODRIGO MOMO/		10	10	10	10	10	10	10	10					
0	THIAGO PAES/										8	4	4	4	4
0	TOTAL	24	24	24	24	24	18	24	24	24	18	24	24	24	24

Roteiro

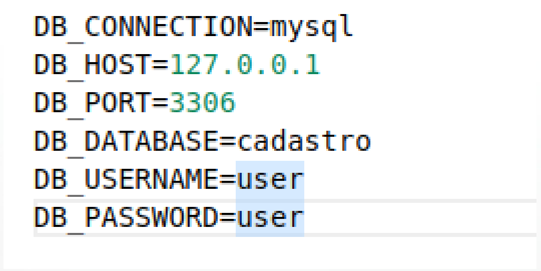
- Criar o projeto

  laravel new cadastro

- Criar uma base de dados MySQL

  create database cadastro

- Configurar o arquivo .env



```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=cadastro
DB_USERNAME=user
DB_PASSWORD=user
```

INSTITUTO FEDERAL
Santa Catarina
Câmpus
Gaspar

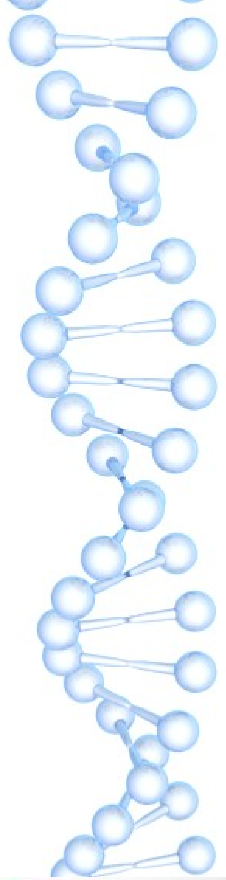
Roteiro

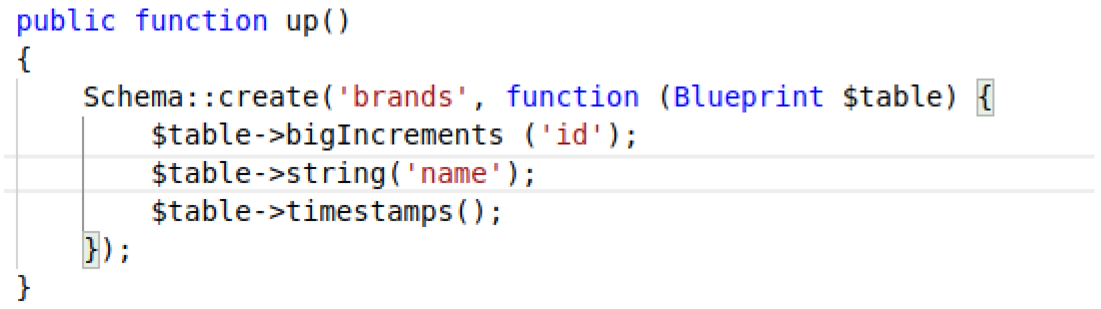
- Criar uma migrations para brands, products e departaments

```
php artisan make:migration create_brands
php artisan make:migration create_products
php artisan make:migration create_departments
```

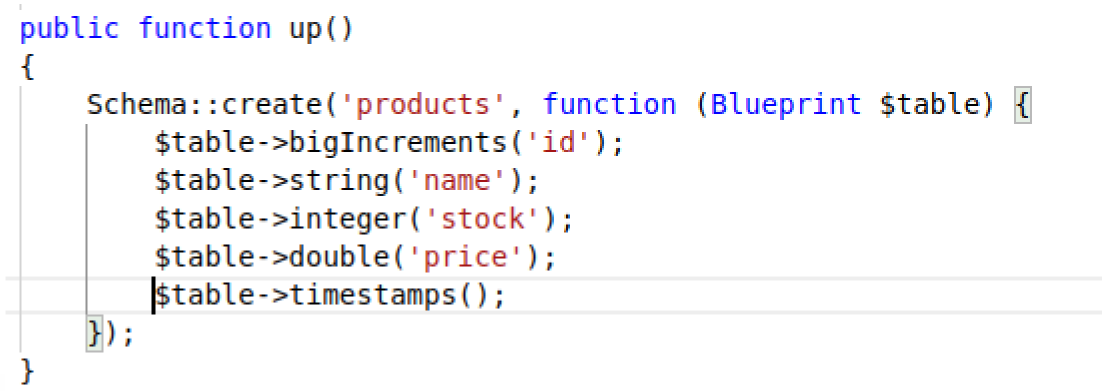
INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Configurar a funtion up() de criação de campos e tabelas no banco de dados

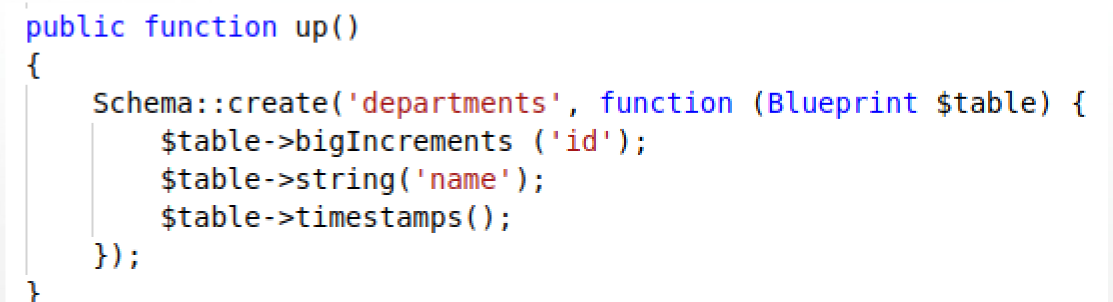




```
public function up()
{
    Schema::create('brands', function (Blueprint $table) {
        $table->bigIncrements ('id');
        $table->string('name');
        $table->timestamps();
    });
}
```



```
public function up()
{
    Schema::create('products', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->string('name');
        $table->integer('stock');
        $table->double('price');
        $table->timestamps();
    });
}
```

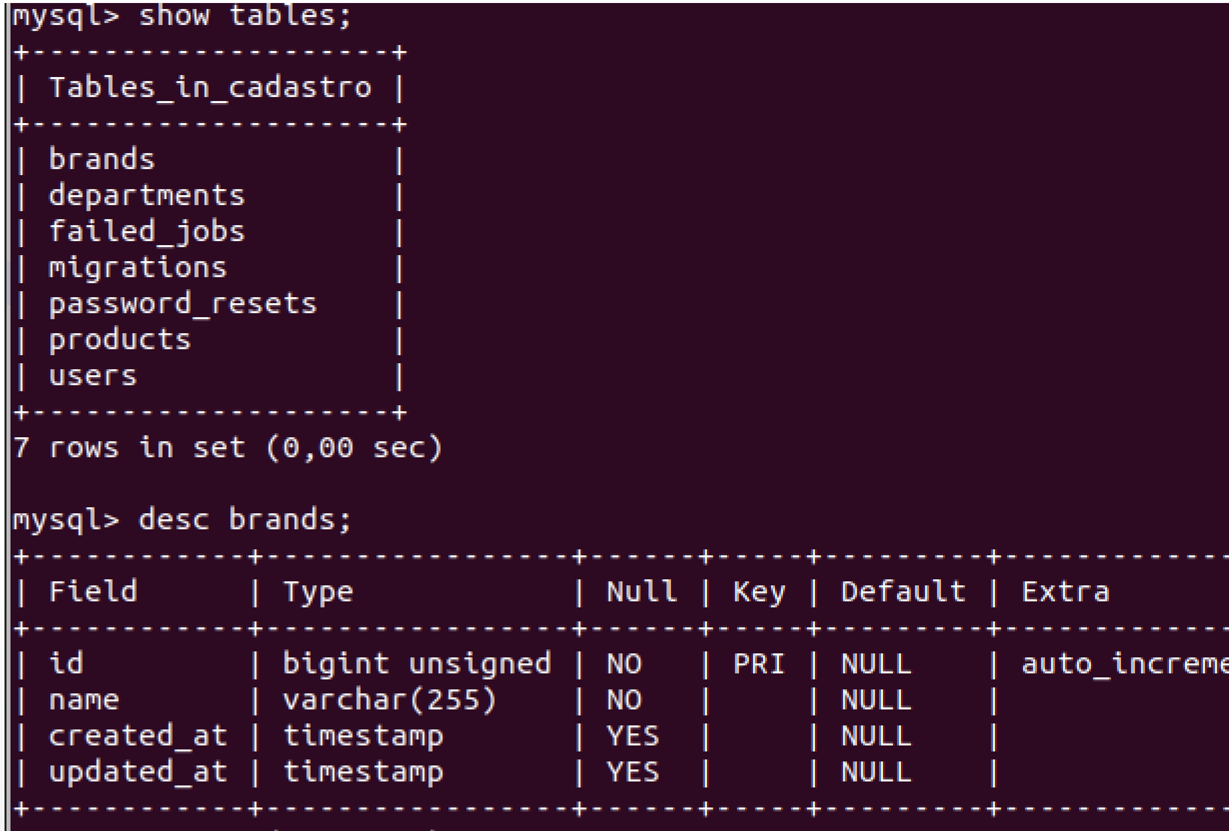


```
public function up()
{
    Schema::create('departments', function (Blueprint $table) {
        $table->bigIncrements ('id');
        $table->string('name');
        $table->timestamps();
    });
}
```

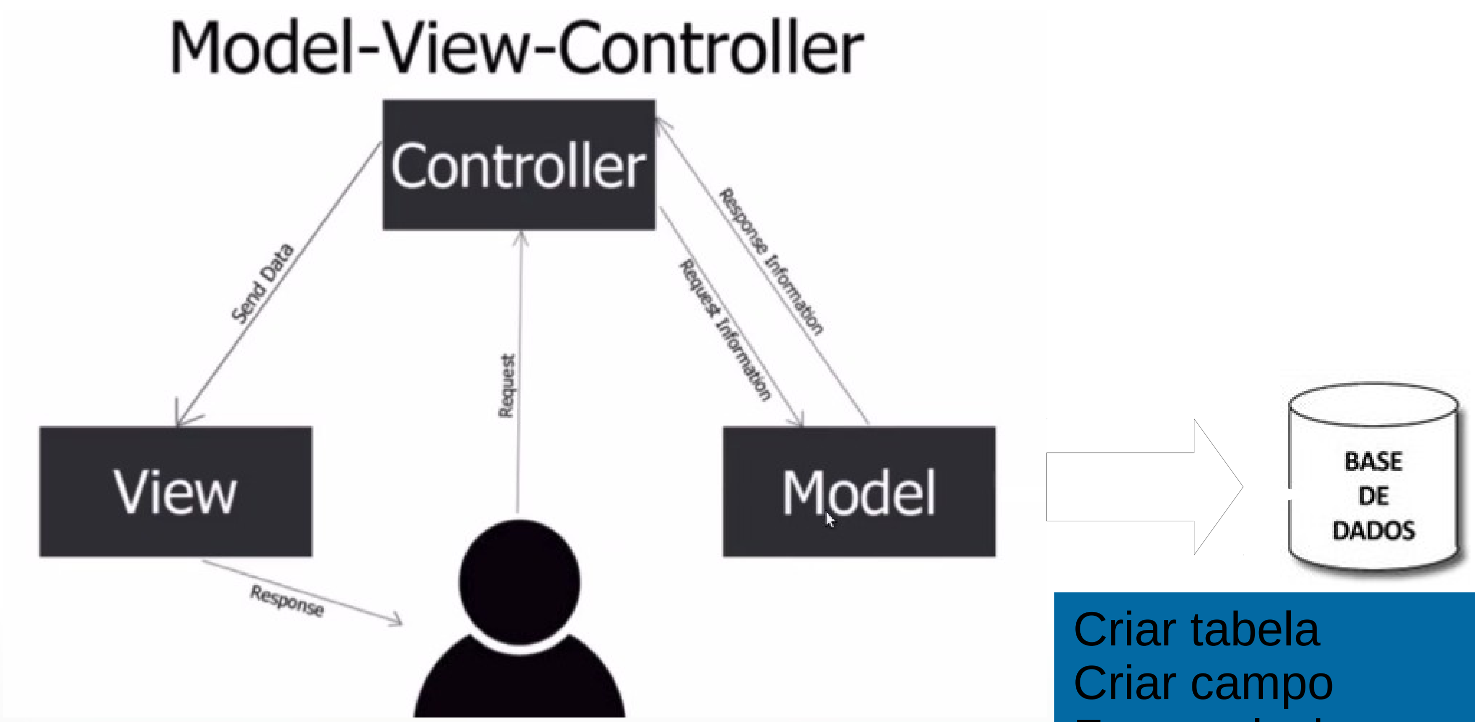
INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Executar o migration

php artisan migrate


```
mysql> show tables;
+--------------------+
| Tables_in_cadastro |
+--------------------+
| brands             |
| departments        |
| failed_jobs        |
| migrations         |
| password_resets    |
| products           |
| users              |
+--------------------+
7 rows in set (0,00 sec)

mysql> desc brands;
+------------+-----------------+------+-----+---------+------------
| Field      | Type            | Null | Key | Default | Extra
+------------+-----------------+------+-----+---------+------------
| id         | bigint unsigned | NO   | PRI | NULL    | auto_increme
| name       | varchar(255)    | NO   |     | NULL    |
| created_at | timestamp       | YES  |     | NULL    |
| updated_at | timestamp       | YES  |     | NULL    |
+------------+-----------------+------+-----+---------+------------
```

Roteiro
Modelo



Criar tabela
Criar campo
Fazer relacionamentos
Fazer consultas

Criando os modelos

- Vamos criar três modelos
  - Produto
  - Marca
  - Departamento
- A partir do comando php

```
php artisan:make model Product
php artisan:make model Brand
php artisan:make model Department
```

Classes

- Na pasta App, esse comando vai criar as três classes respectivas
- São classes vazias que herdam da classe Model

Inserindo registros com Tinker composer require laravel/tinker

- Através do Tinker
  - O Tinker é um console interativo do Laravel, um shell do PHP com acesso às classes do nosso projeto.
  - Através do Tinker eu posso realizar as tarefas de BD
- Para acessar o Tinker

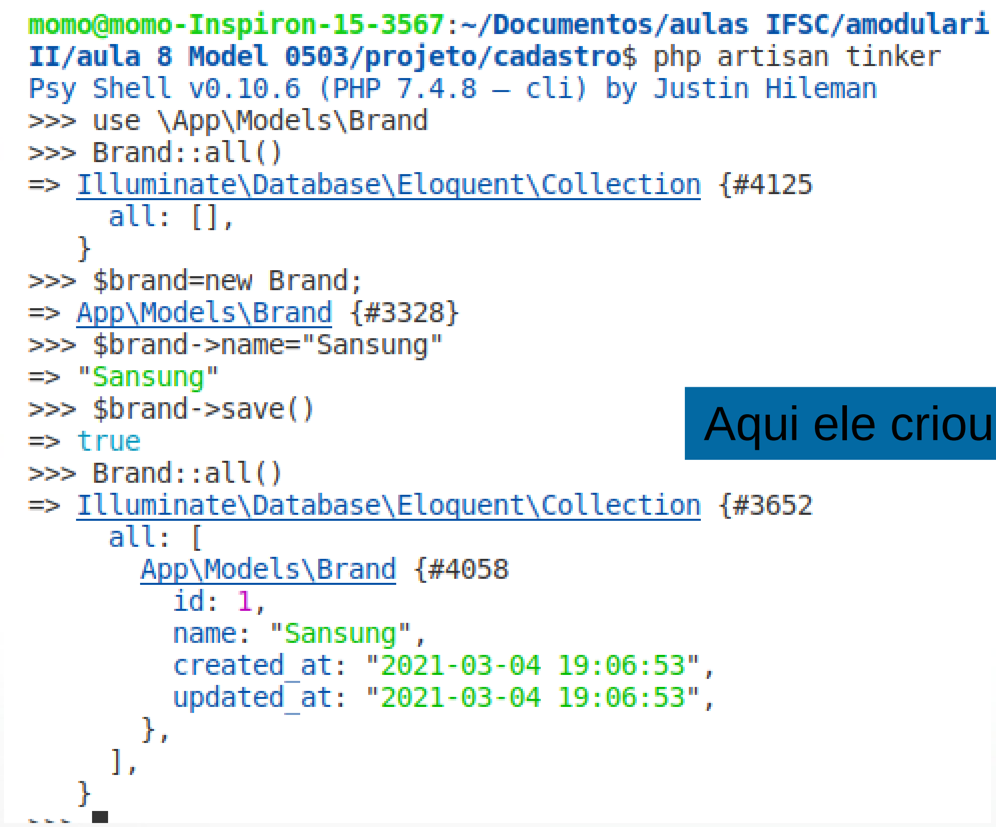
```
php artisan tinker
>>>Brand::all()
```

  - >>>erro
- O erro é porque a classe Brand está em App

Inserindo registros com Tinker

- Portanto temos que setar o caminho através do comando use
  - >>>use \App\Models\Brand;
  - >>>Brand::all()
  - Está vazio
  -

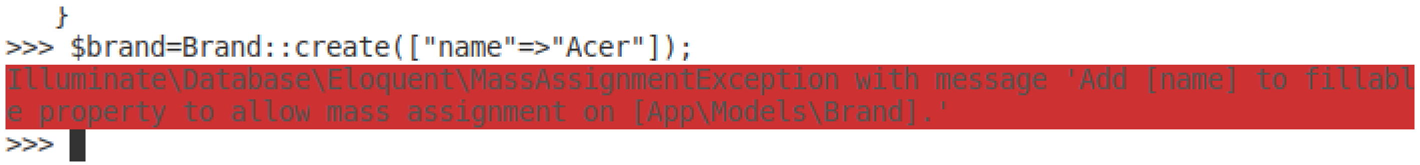
Inserindo registros com Tinker



```
momo@momo-Inspiron-15-3567:~/Documentos/aulas IFSC/amodulari
II/aula 8 Model 0503/projeto/cadastro$ php artisan tinker
Psy Shell v0.10.6 (PHP 7.4.8 — cli) by Justin Hileman
>>> use \App\Models\Brand
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4125
     all: [],
   }
>>> $brand=new Brand;
=> App\Models\Brand {#3328}
>>> $brand->name="Sansung"
=> "Sansung"
>>> $brand->save()
=> true
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#3652
     all: [
       App\Models\Brand {#4058
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
     ],
   }
```

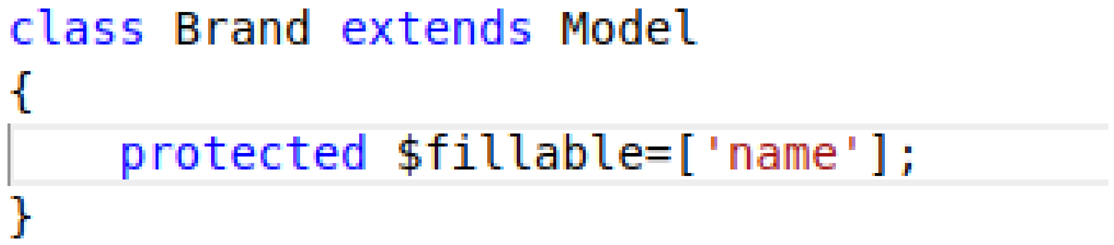
Aqui ele criou um registro

Inserindo vários registros com Tinker



```
    }
>>> $brand=Brand::create(["name"=>"Acer"]);
Illuminate\Database\Eloquent\MassAssignmentException with message 'Add [name] to fillabl
e property to allow mass assignment on [App\Models\Brand].'
>>> 
```

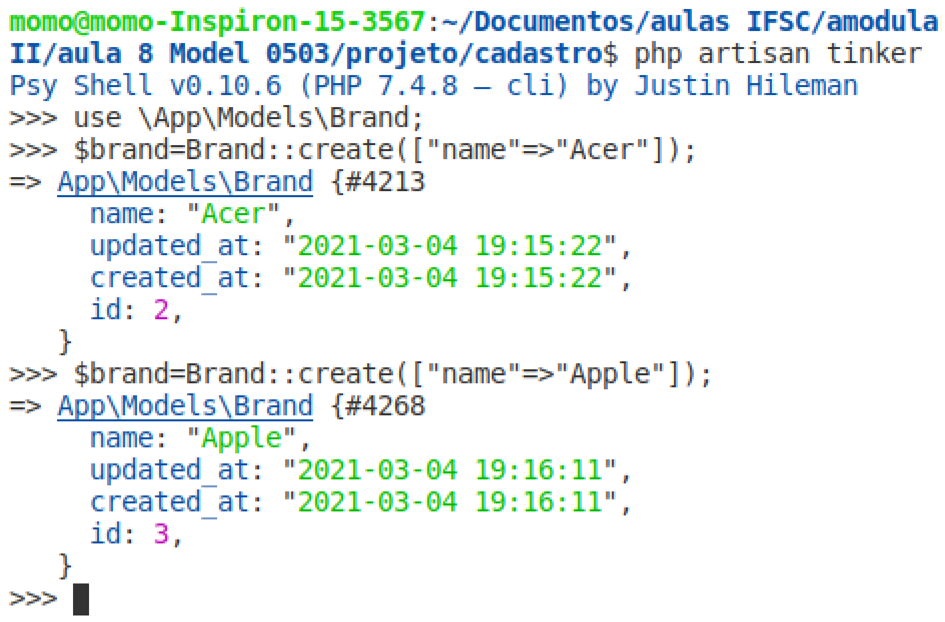
- Adicionar atributo fillable na classe Brand



```
class Brand extends Model
{
    protected $fillable=['name'];
}
```

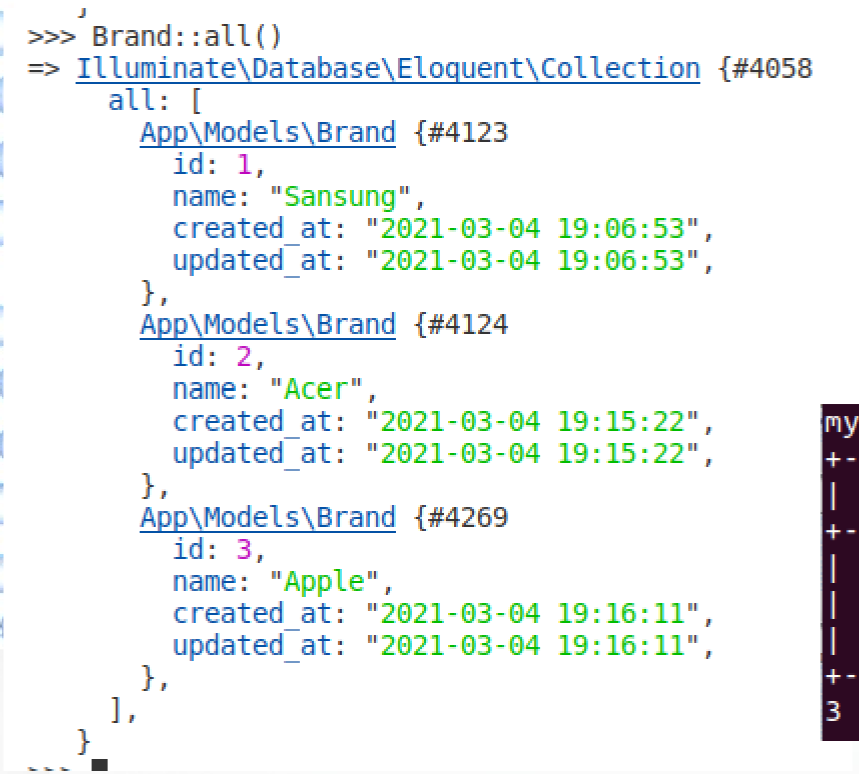
Inserindo registros com Tinker

- Sair do tinker
  - >>>quit
- Entrar novamente
  - >>>php artisan tinker
- Informar o path
  - >>>use \App\Models\Brand;

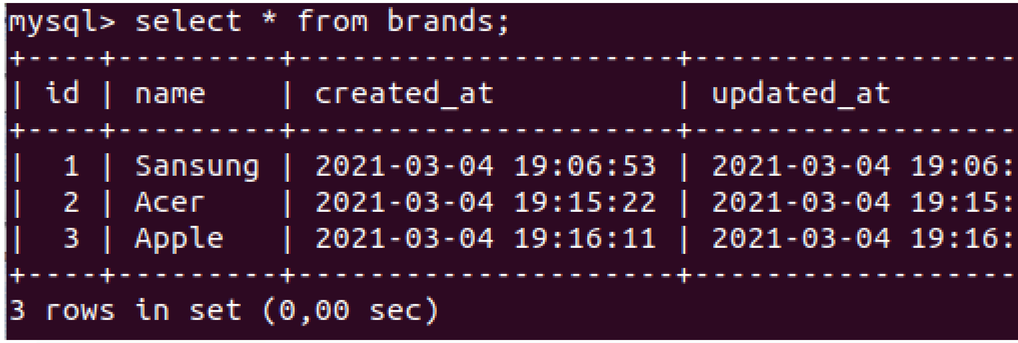


```
momo@momo-Inspiron-15-3567:~/Documentos/aulas IFSC/amodula
II/aula 8 Model 0503/projeto/cadastro$ php artisan tinker
Psy Shell v0.10.6 (PHP 7.4.8 — cli) by Justin Hileman
>>> use \App\Models\Brand;
>>> $brand=Brand::create(["name"=>"Acer"]);
=> App\Models\Brand {#4213
     name: "Acer",
     updated_at: "2021-03-04 19:15:22",
     created_at: "2021-03-04 19:15:22",
     id: 2,
   }
>>> $brand=Brand::create(["name"=>"Apple"]);
=> App\Models\Brand {#4268
     name: "Apple",
     updated_at: "2021-03-04 19:16:11",
     created_at: "2021-03-04 19:16:11",
     id: 3,
   }
>>> █
```

Verificar a inserção



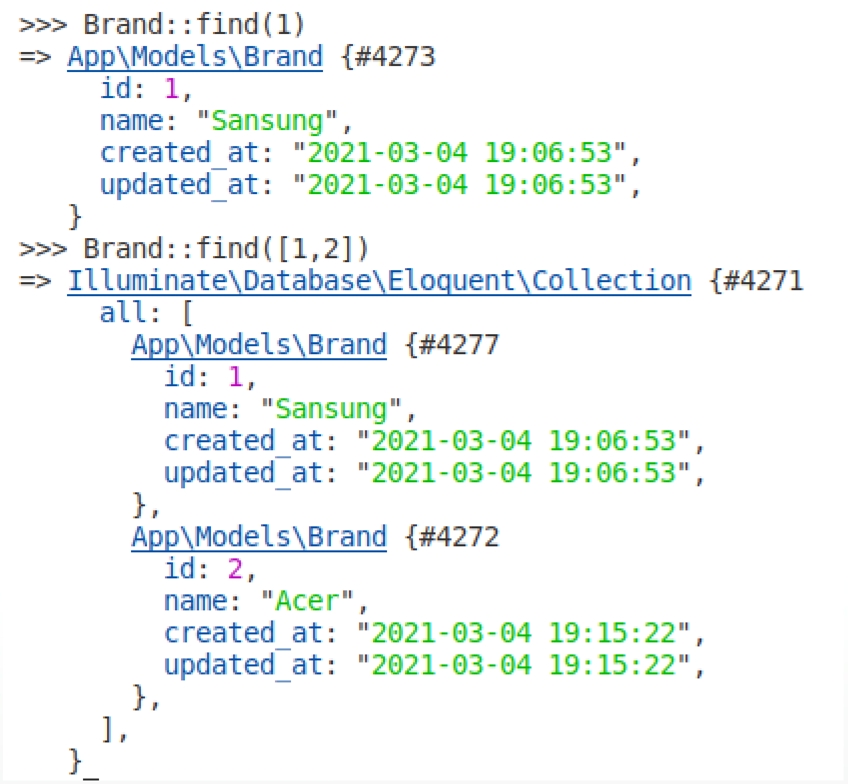
```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4058
     all: [
       App\Models\Brand {#4123
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4124
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4269
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }
```



```
mysql> select * from brands;
+----+----------+---------------------+------------------
| id | name     | created_at          | updated_at
+----+----------+---------------------+------------------
|  1 | Sansung  | 2021-03-04 19:06:53 | 2021-03-04 19:06:
|  2 | Acer     | 2021-03-04 19:15:22 | 2021-03-04 19:15:
|  3 | Apple    | 2021-03-04 19:16:11 | 2021-03-04 19:16:
+----+----------+---------------------+------------------
3 rows in set (0,00 sec)
```

Realizando consultas find

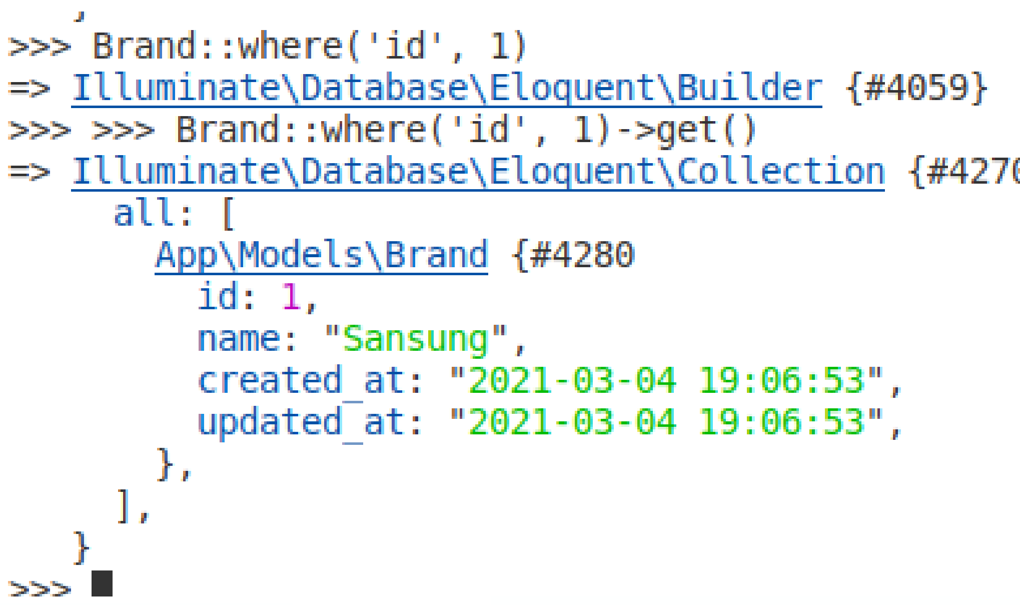
- Com find



```
>>> Brand::find(1)
=> App\Models\Brand {#4273
     id: 1,
     name: "Sansung",
     created_at: "2021-03-04 19:06:53",
     updated_at: "2021-03-04 19:06:53",
   }
>>> Brand::find([1,2])
=> Illuminate\Database\Eloquent\Collection {#4271
     all: [
       App\Models\Brand {#4277
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4272
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
     ],
   }_
```

Realizando consultas where

- Com where
- Retorna um collection
  - Posso criar uma consulta mais sofisticada



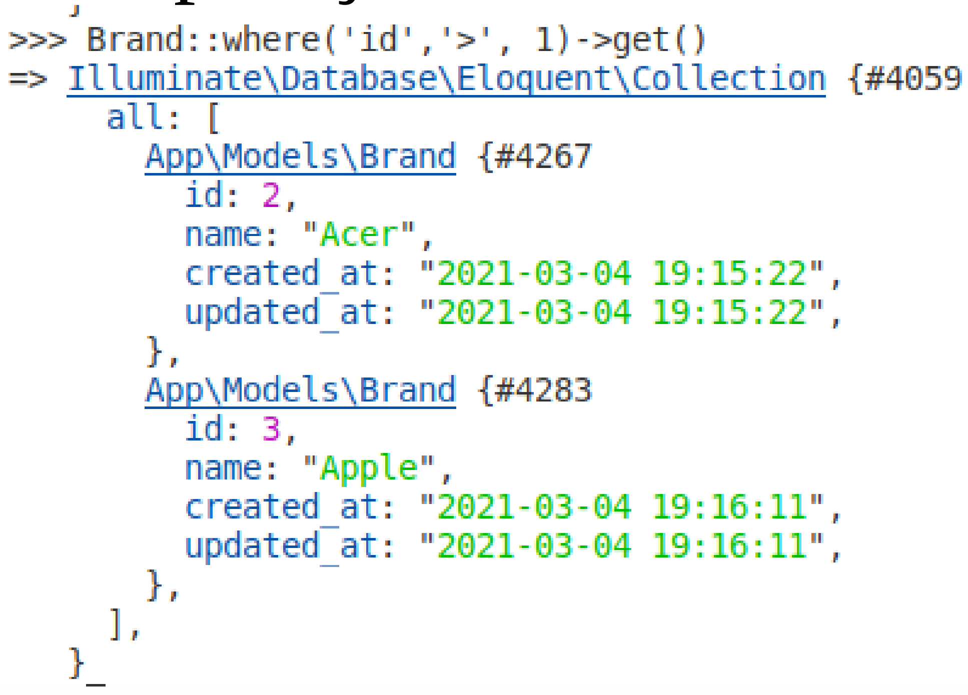
```
>>> Brand::where('id', 1)
=> Illuminate\Database\Eloquent\Builder {#4059}
>>> >>> Brand::where('id', 1)->get()
=> Illuminate\Database\Eloquent\Collection {#427
     all: [
       App\Models\Brand {#4280
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
     ],
   }
>>> 
```

Realizando consultas where com operação

- Com where com operação

Retorna um collection

- Posso criar uma consulta mais sofisticada



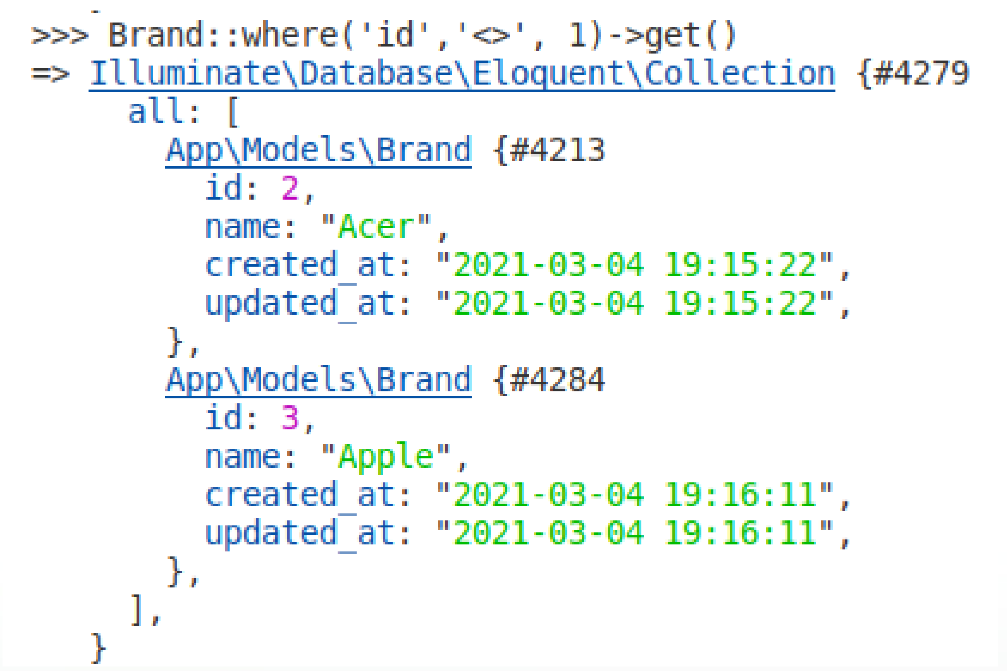
```
>>> Brand::where('id','>', 1)->get()
=> Illuminate\Database\Eloquent\Collection {#4059
     all: [
       App\Models\Brand {#4267
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4283
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }
```

Realizando consultas where

- Com where com operação

Retorna um collection

- Posso criar uma consulta mais sofisticada

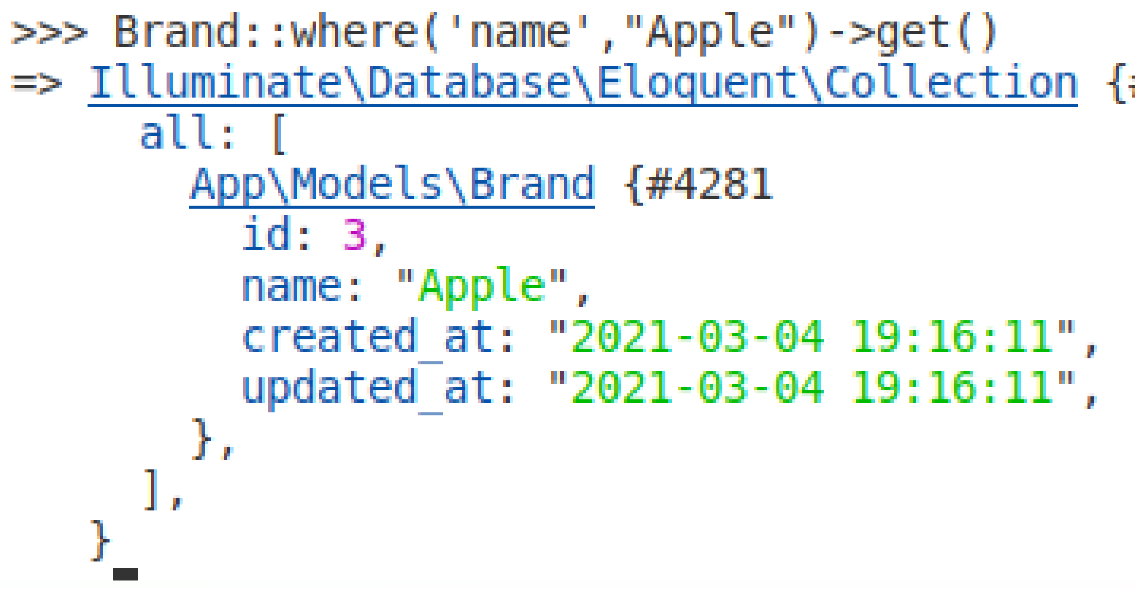


```
>>> Brand::where('id','<>', 1)->get()
=> Illuminate\Database\Eloquent\Collection {#4279
     all: [
       App\Models\Brand {#4213
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4284
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }
```

Realizando consultas where

- Com where com operação

Retorna um collection
- Posso criar uma consulta mais sofisticada



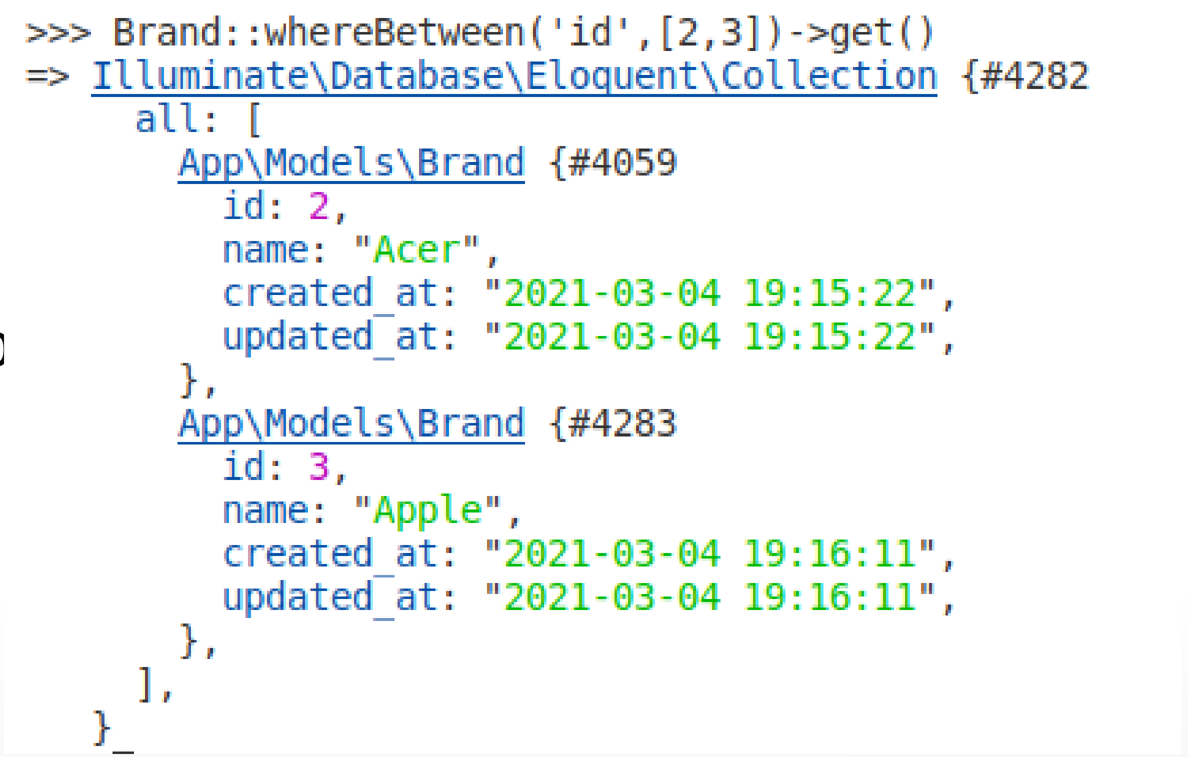
```
>>> Brand::where('name',"Apple")->get()
=> Illuminate\Database\Eloquent\Collection {
     all: [
       App\Models\Brand {#4281
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }
```

Realizando consultas whereBetween

- Com where com intervalos de ID

Retorna um collectio

- Posso criar uma consulta mais sofisticada

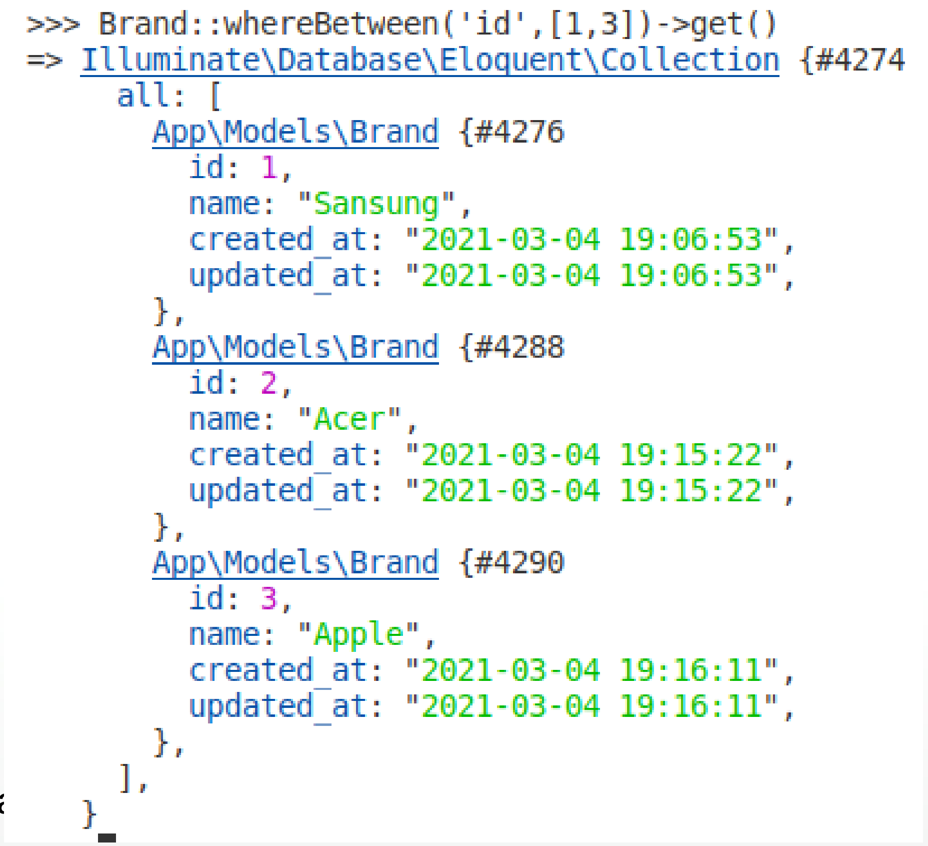

```
>>> Brand::whereBetween('id',[2,3])->get()
=> Illuminate\Database\Eloquent\Collection {#4282
     all: [
       App\Models\Brand {#4059
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4283
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }_
```

Realizando consultas whereBetween

- Com where com intervalos de ID

Retorna um collection

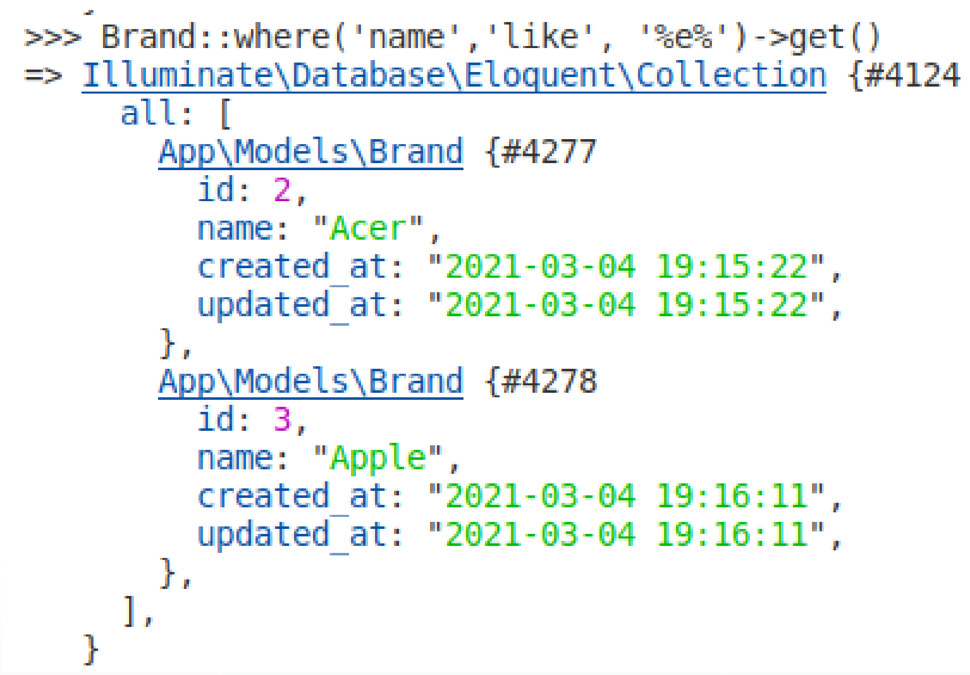
- Posso criar uma consulta mais sofisticada



```
>>> Brand::whereBetween('id',[1,3])->get()
=> Illuminate\Database\Eloquent\Collection {#4274
     all: [
       App\Models\Brand {#4276
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4288
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4290
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }
```

Realizando consultas where like

- Com like



```
>>> Brand::where('name','like', '%e%')->get()
=> Illuminate\Database\Eloquent\Collection {#4124
     all: [
       App\Models\Brand {#4277
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4278
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }
```

Retorna um collection

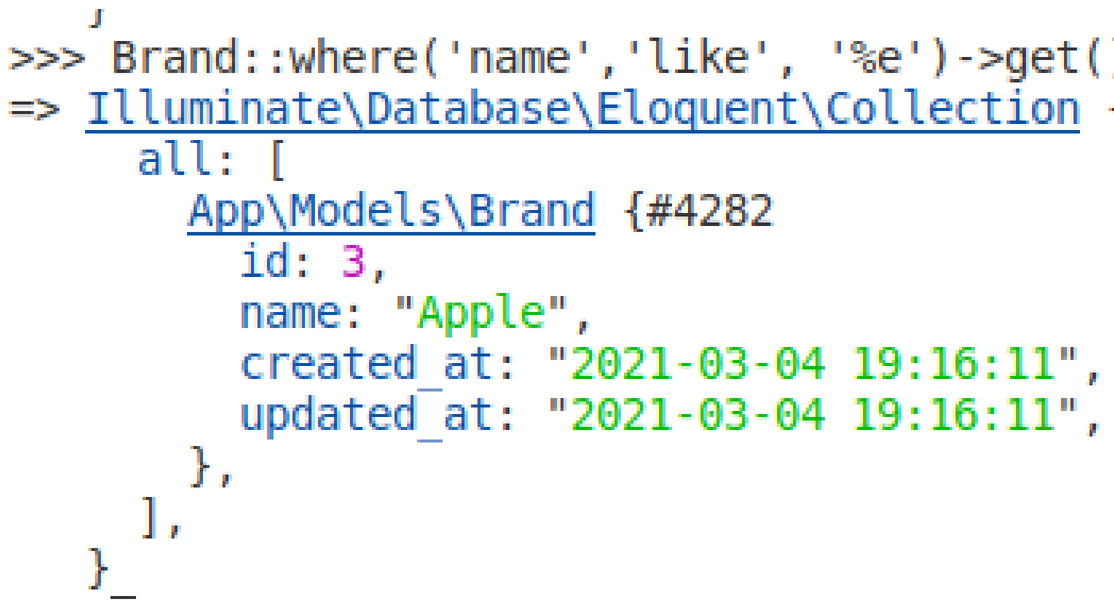
- Por exemplo, criar consulta buscando por produtos com a letra “%e%”

Realizando consultas

- Com like

Retorna um collection

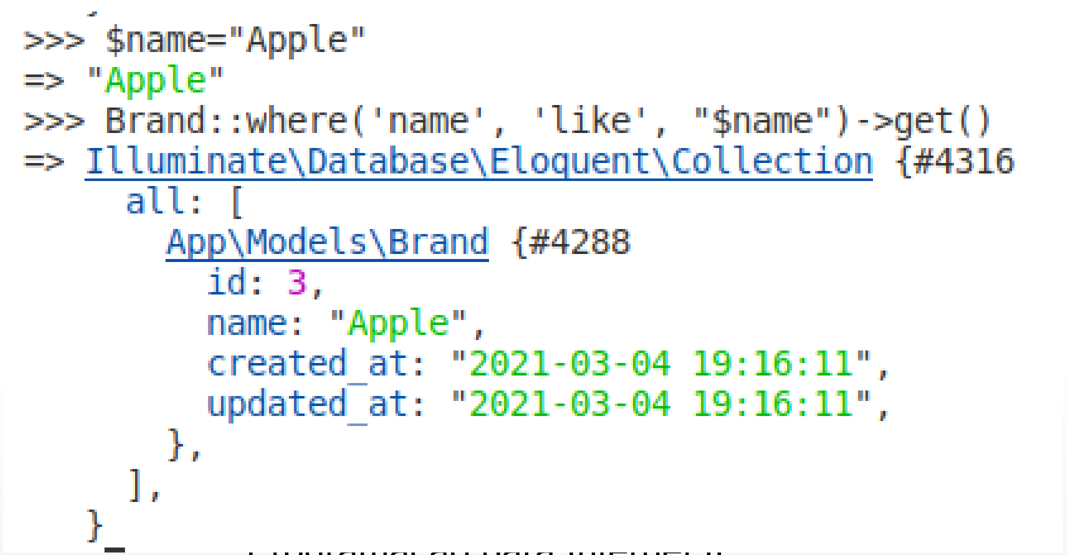
- Por exemplo, criar consulta buscando por produtos com a letra “%e”



```
>>> Brand::where('name','like', '%e')->get(
=> Illuminate\Database\Eloquent\Collection
     all: [
       App\Models\Brand {#4282
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }_
```

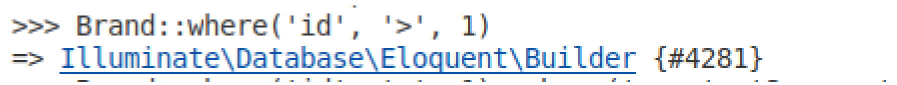
Realizando consultas com like

- Exemplo de uso
  - Implementar uma consulta buscando por uma variável vindo de um formulário $name=”e”

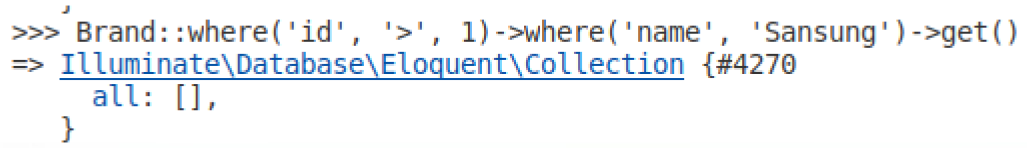


```
>>> $name="Apple"
=> "Apple"
>>> Brand::where('name', 'like', "$name")->get()
=> Illuminate\Database\Eloquent\Collection {#4316
     all: [
       App\Models\Brand {#4288
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }
```

Encadeando queries

- Por exemplo, comparando mais de um campo
  - Aqui o where, retorna um builder, posso ir encadeando a busca
  - 

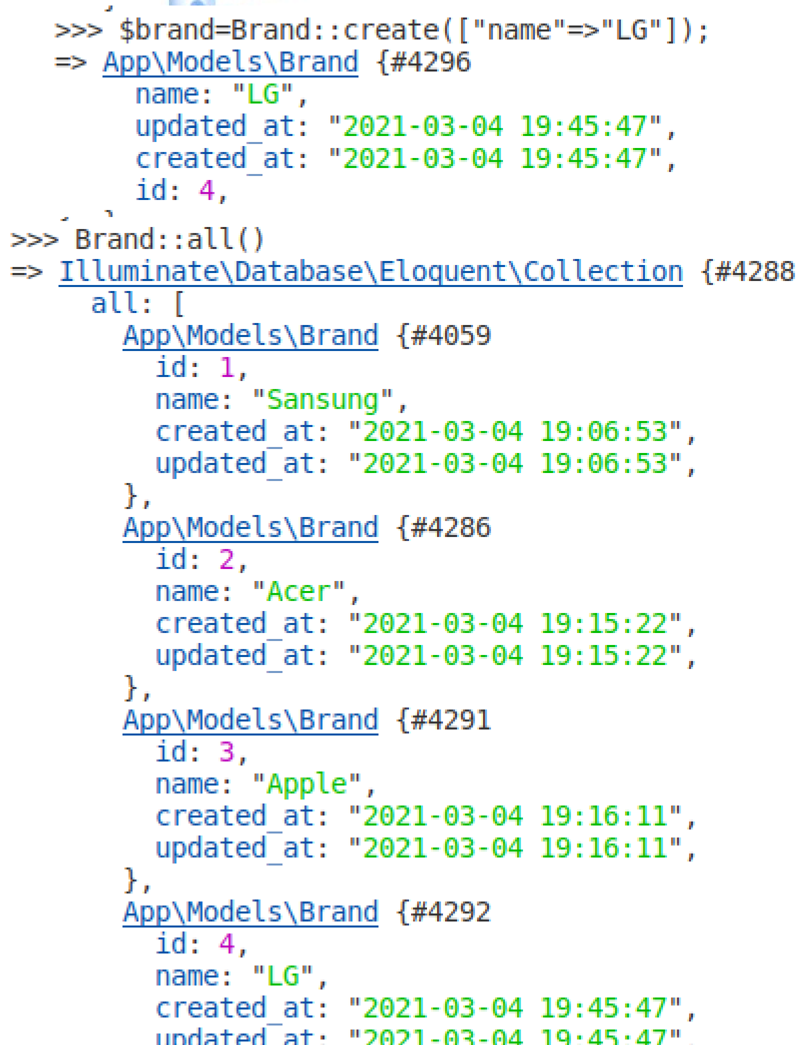
```
>>> Brand::where('id', '>', 1)
=> Illuminate\Database\Eloquent\Builder {#4281}
```

  - Aqui o where, retorna um collection
  - 

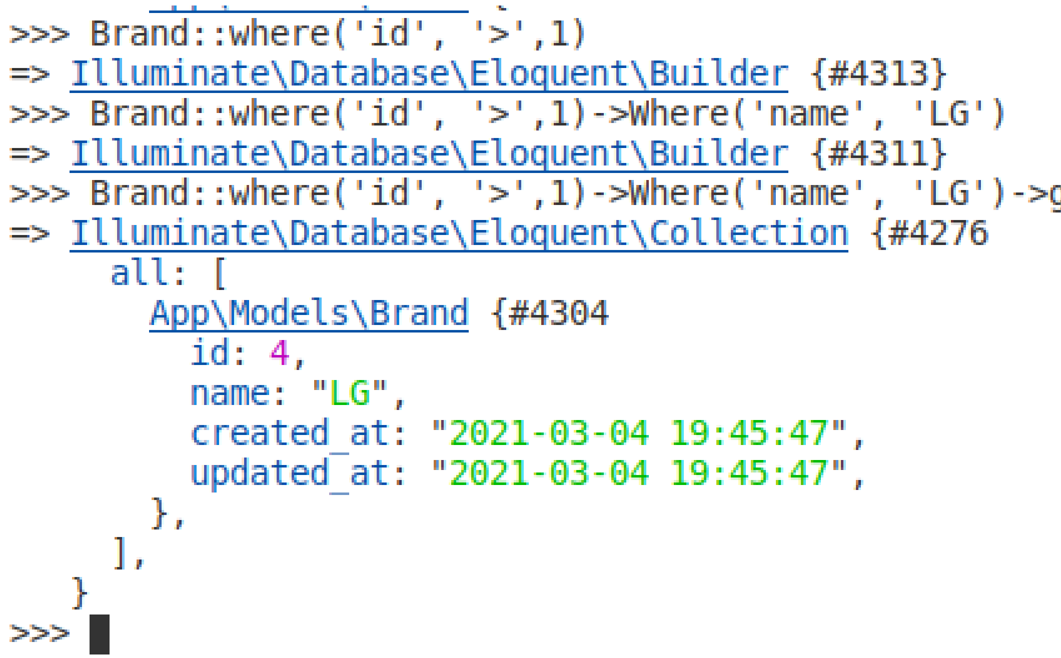
```
>>> Brand::where('id', '>', 1)->where('name', 'Sansung')->get()
=> Illuminate\Database\Eloquent\Collection {#4270
     all: [],
   }
```

Encadeando queries “And”

- Por exemplo:



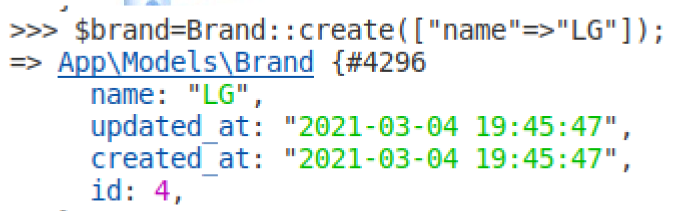
```
>>> $brand=Brand::create(["name"=>"LG"]);
=> App\Models\Brand {#4296
     name: "LG",
     updated_at: "2021-03-04 19:45:47",
     created_at: "2021-03-04 19:45:47",
     id: 4,
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4288
     all: [
       App\Models\Brand {#4059
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4286
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4291
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
       App\Models\Brand {#4292
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
```



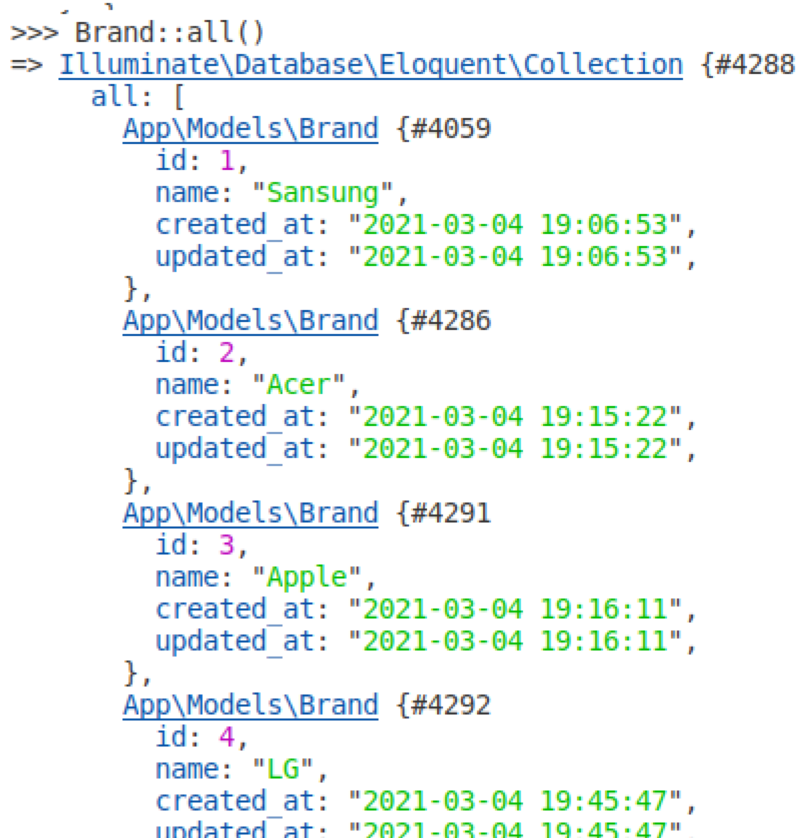
```
>>> Brand::where('id', '>',1)
=> Illuminate\Database\Eloquent\Builder {#4313}
>>> Brand::where('id', '>',1)->Where('name', 'LG')
=> Illuminate\Database\Eloquent\Builder {#4311}
>>> Brand::where('id', '>',1)->Where('name', 'LG')->g
=> Illuminate\Database\Eloquent\Collection {#4276
     all: [
       App\Models\Brand {#4304
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
       },
     ],
   }
>>>
```

Encadeando queries “Or”

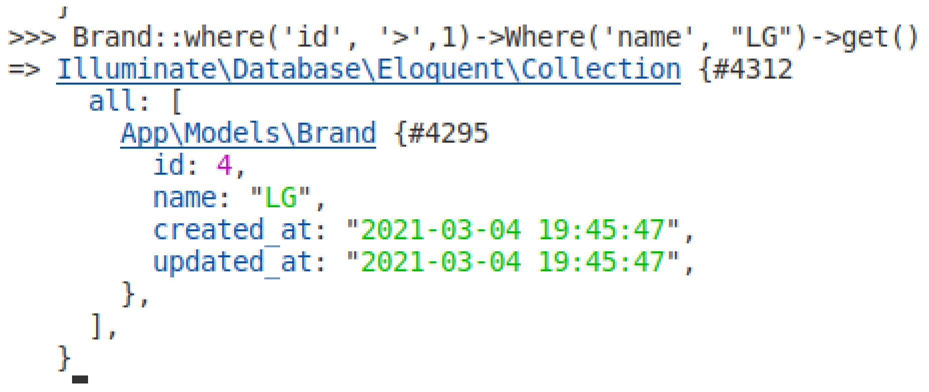
- Por exemplo:



```
>>> $brand=Brand::create(["name"=>"LG"]);
=> App\Models\Brand {#4296
     name: "LG",
     updated_at: "2021-03-04 19:45:47",
     created_at: "2021-03-04 19:45:47",
     id: 4,
```



```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4288
     all: [
       App\Models\Brand {#4059
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4286
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4291
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
       App\Models\Brand {#4292
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
```



```
>>> Brand::where('id', '>',1)->Where('name', "LG")->get()
=> Illuminate\Database\Eloquent\Collection {#4312
     all: [
       App\Models\Brand {#4295
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
       },
     ],
   }
```

Agrupando queries

- Aqui podemos fazer consulta sofisticadas
- Por exemplo:
  - (id > 1 and id<4) or (name='apple' or name='sansung')
  - parte 1: (id > 1 and id<4)
    - where('id, '>', 1)->where('id', '<', 4)-> get()

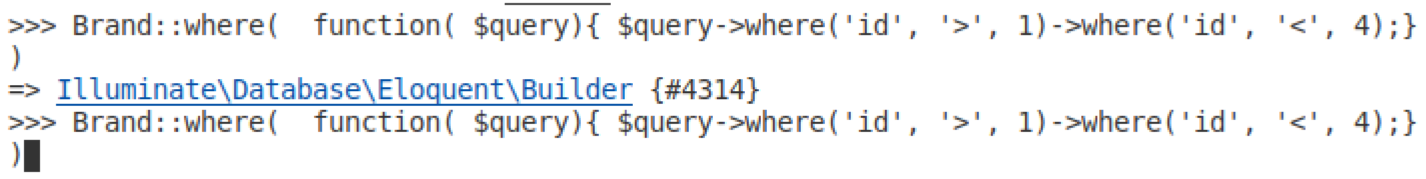
  Parte 2: (name='apple' or name='sansung')

  where('name', 'apple') → orwhere('name', 'sansung')-> get()

-

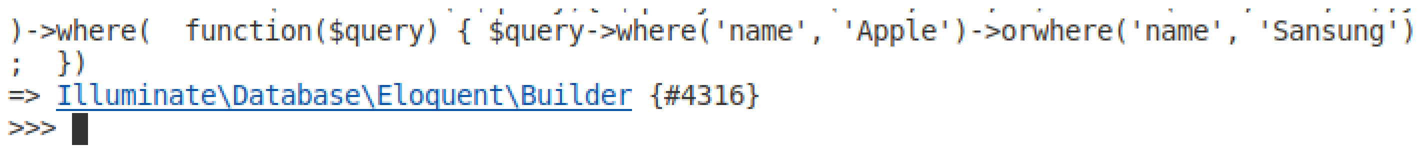
Agrupando queries om função

- Agrupando: (id > 1 and id<4) or (name=’Apple’ or name=’Sansung’)
  - (id > 1 and id<4)



```
>>> Brand::where(  function( $query){ $query->where('id', '>', 1)->where('id', '<', 4);}
)
=> Illuminate\Database\Eloquent\Builder {#4314}
>>> Brand::where(  function( $query){ $query->where('id', '>', 1)->where('id', '<', 4);}
)
```

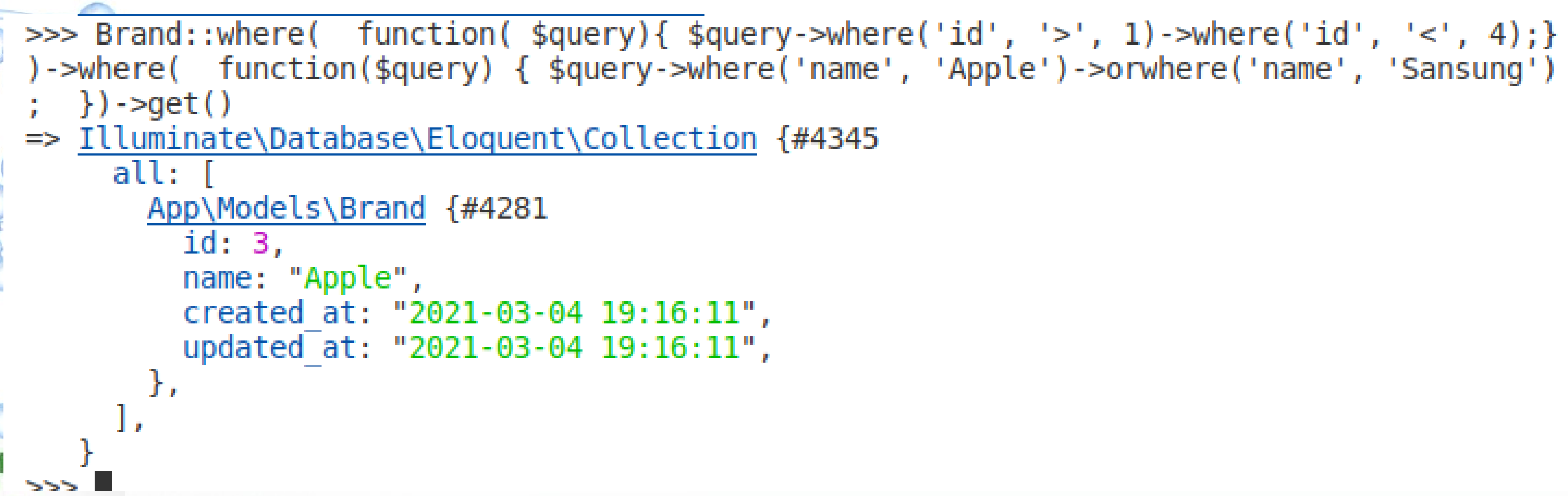
  - (name=’Apple’ or name=’Sansung’)



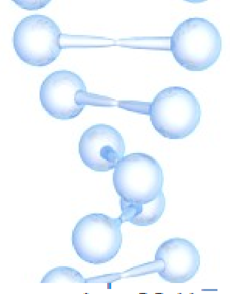
```
)->where(  function($query) { $query->where('name', 'Apple')->orwhere('name', 'Sansung')
;  })
=> Illuminate\Database\Eloquent\Builder {#4316}
>>> 
```

Agrupando queries om função

- Agrupando: (id > 1 and id<4) or (name='Apple' or name='Sansung')

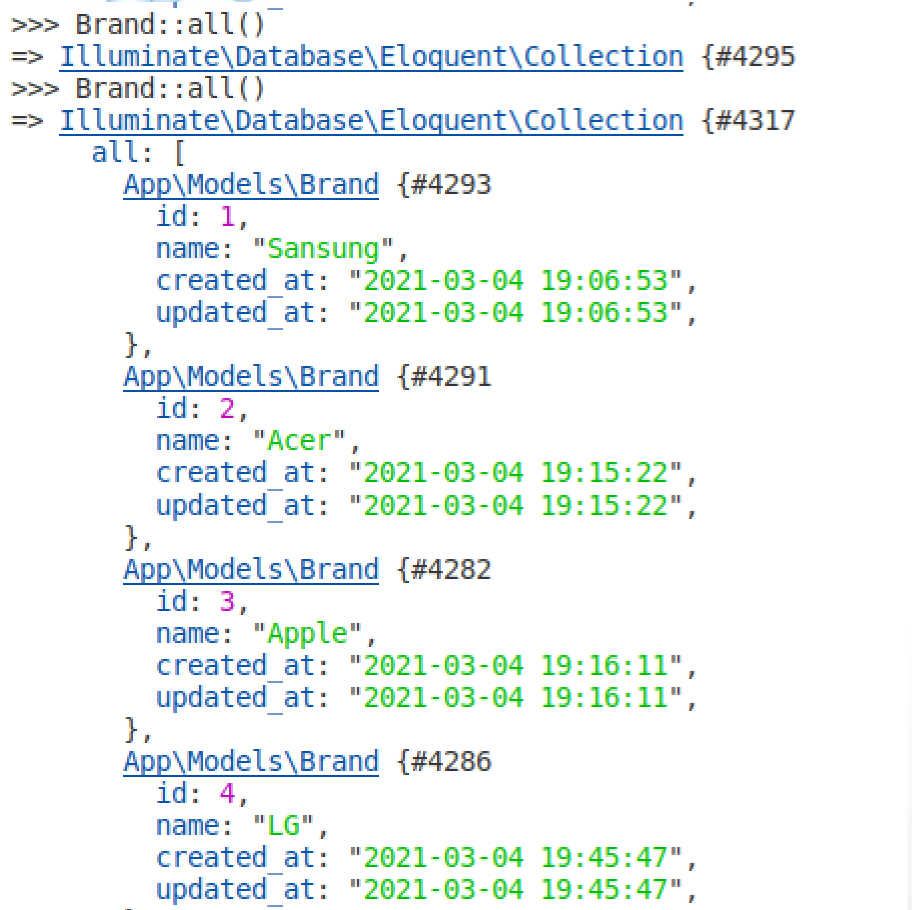


```
>>> Brand::where(  function( $query){ $query->where('id', '>', 1)->where('id', '<', 4);}
)->where(  function($query) { $query->where('name', 'Apple')->orwhere('name', 'Sansung')
;  })->get()
=> Illuminate\Database\Eloquent\Collection {#4345
     all: [
       App\Models\Brand {#4281
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }
>>> 
```

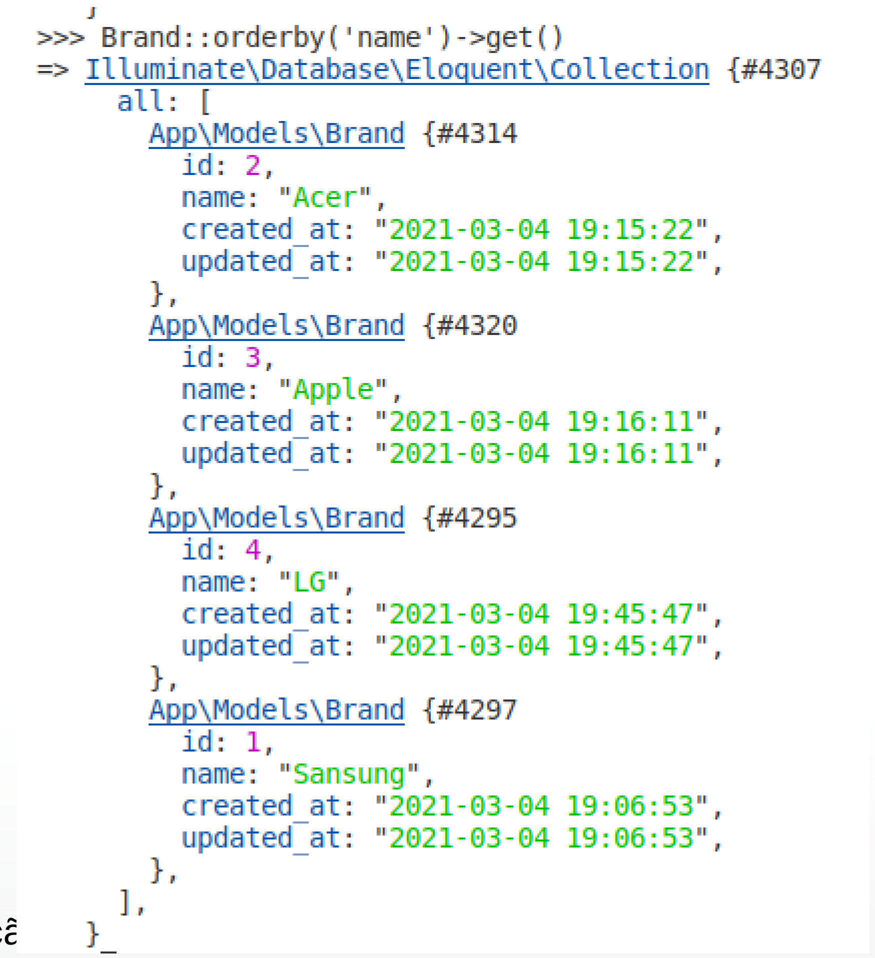


Ordenando

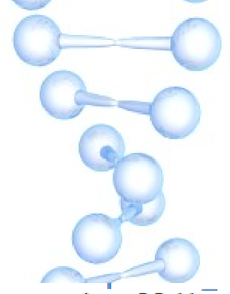
- Posso ordenar por campos
- Por exemplo, por nome



```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4295
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4317
     all: [
       App\Models\Brand {#4293
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4291
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4282
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
       App\Models\Brand {#4286
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
```

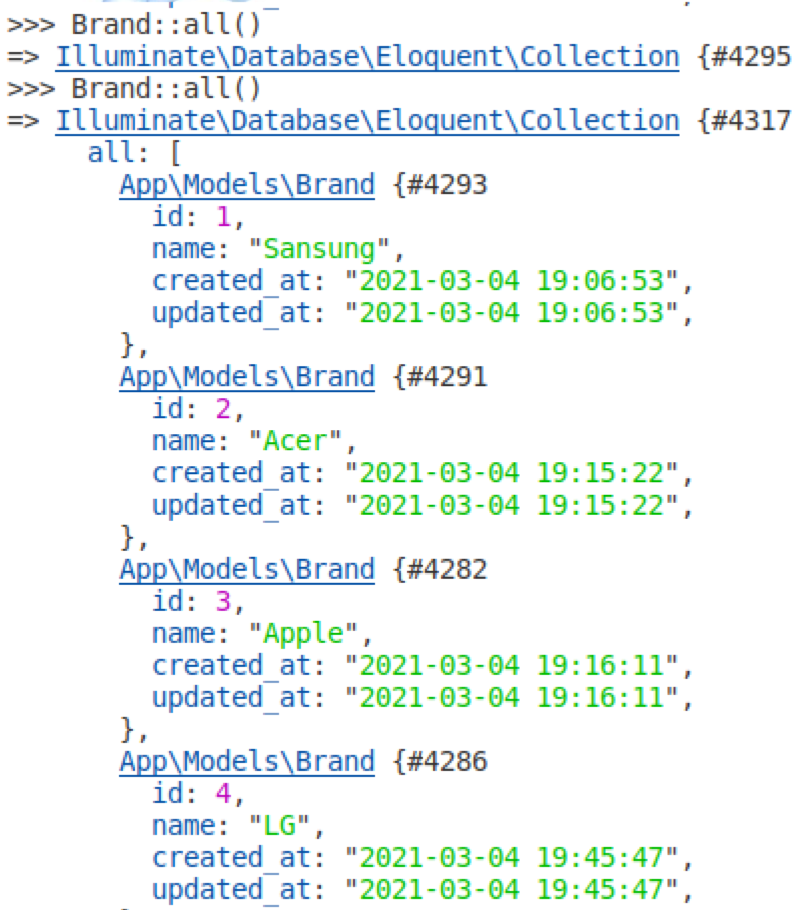


```
>>> Brand::orderby('name')->get()
=> Illuminate\Database\Eloquent\Collection {#4307
     all: [
       App\Models\Brand {#4314
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4320
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
       App\Models\Brand {#4295
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
       },
       App\Models\Brand {#4297
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
     ],
   }_
```

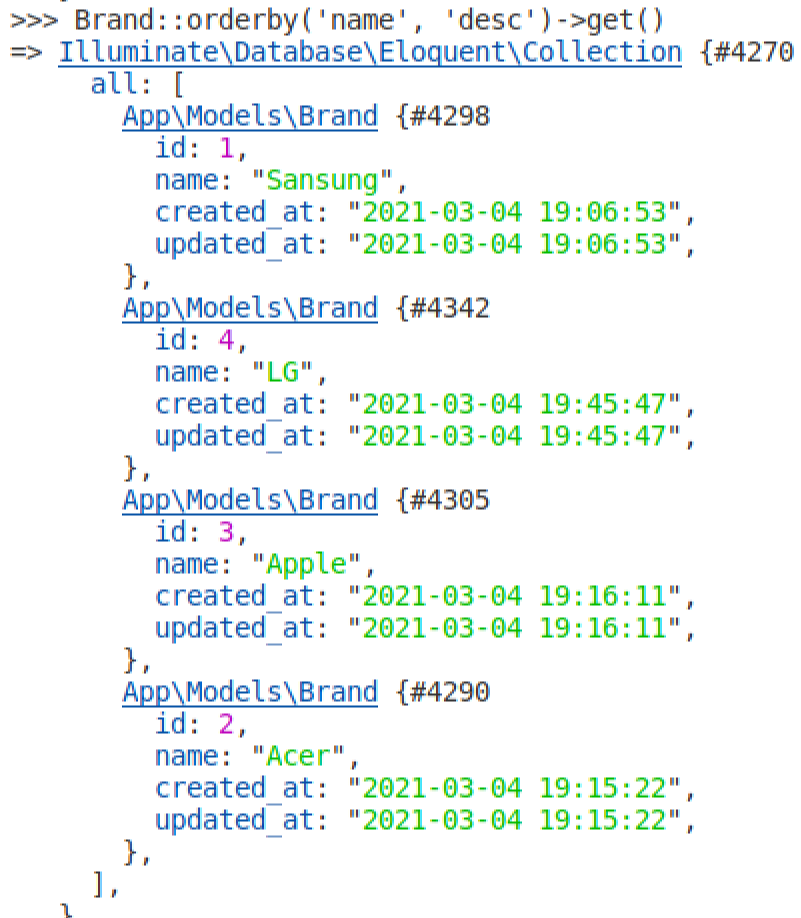


Ordenando

- Posso ordenar por campos
- Por exemplo, por nome descendente



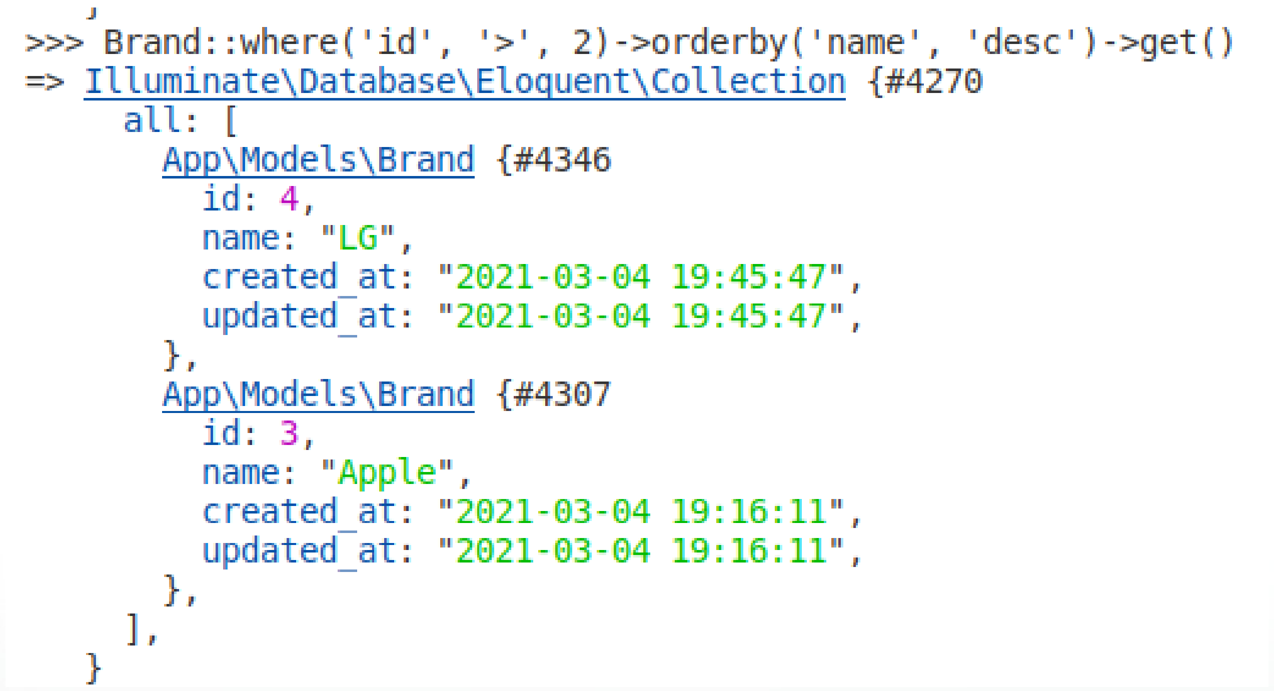
```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4295
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4317
     all: [
       App\Models\Brand {#4293
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4291
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4282
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
       App\Models\Brand {#4286
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
```



```
>>> Brand::orderby('name', 'desc')->get()
=> Illuminate\Database\Eloquent\Collection {#4270
     all: [
       App\Models\Brand {#4298
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4342
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
       },
       App\Models\Brand {#4305
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
       App\Models\Brand {#4290
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
     ],
   }
```

Ordenando

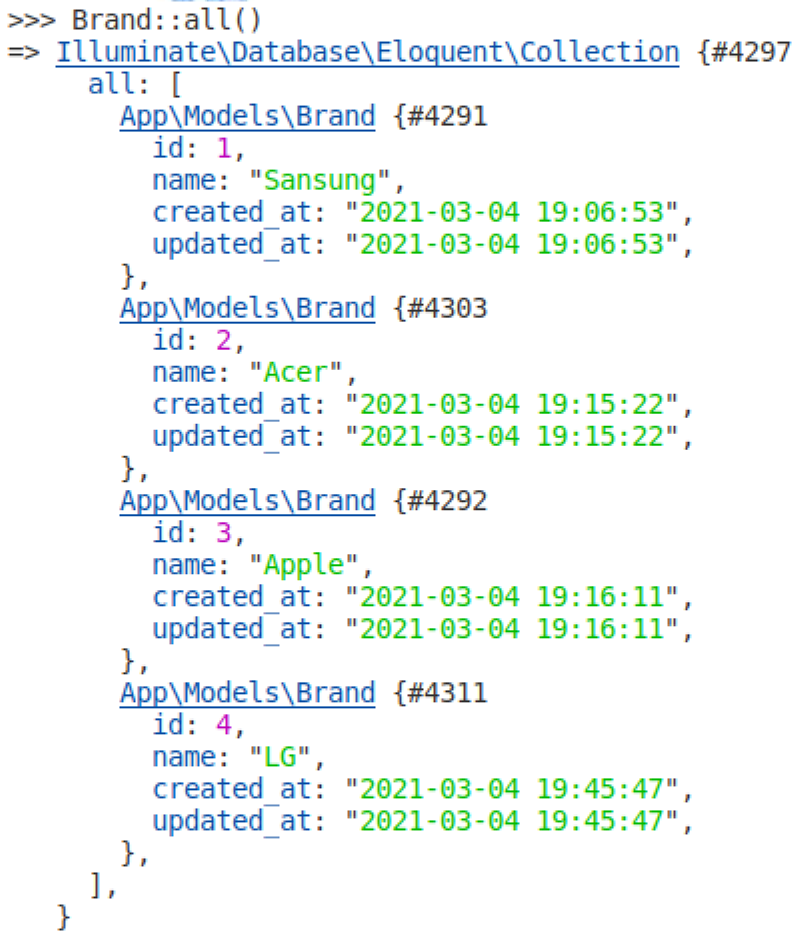
- Definindo limiar com where



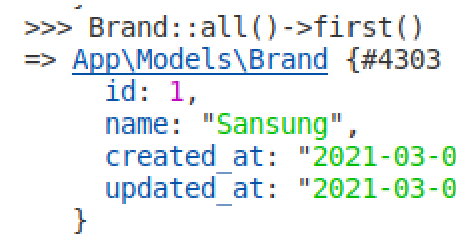
```
   ,
>>> Brand::where('id', '>', 2)->orderby('name', 'desc')->get()
=> Illuminate\Database\Eloquent\Collection {#4270
     all: [
       App\Models\Brand {#4346
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
       },
       App\Models\Brand {#4307
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
     ],
   }
```

Trabalhando com Collections

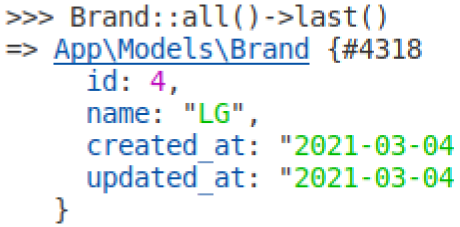
- Um collections tem funções que podemos utilizar



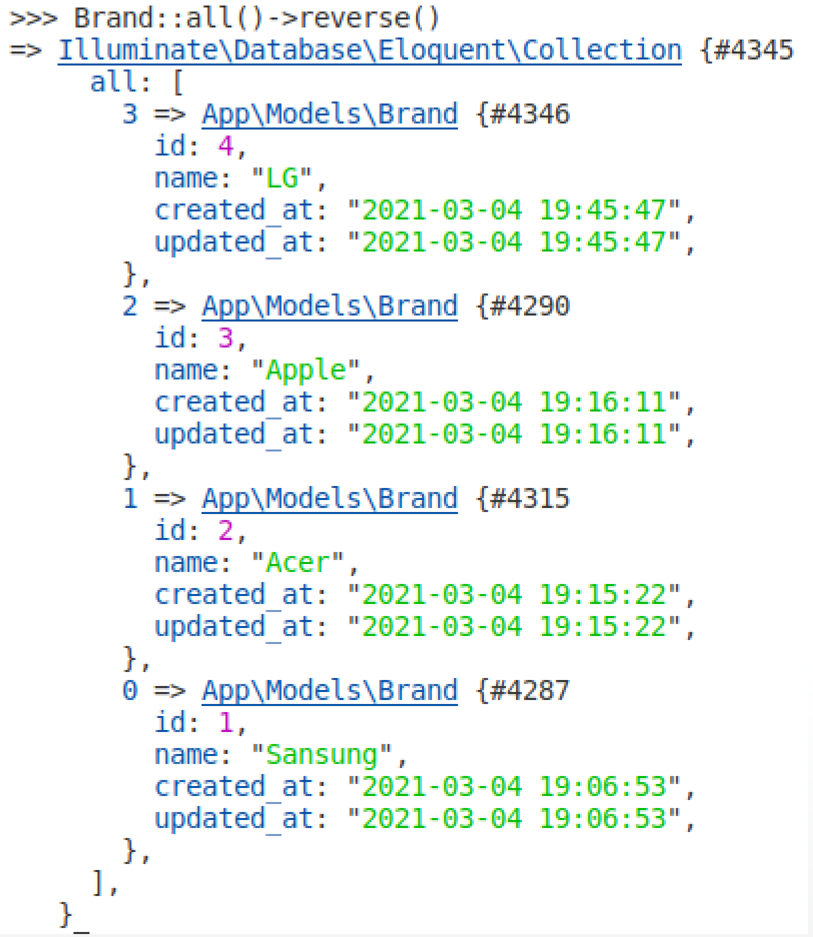
```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4297
     all: [
       App\Models\Brand {#4291
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4303
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4292
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
       App\Models\Brand {#4311
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
       },
     ],
   }
```



```
>>> Brand::all()->first()
=> App\Models\Brand {#4303
     id: 1,
     name: "Sansung",
     created_at: "2021-03-0
     updated_at: "2021-03-0
   }
```



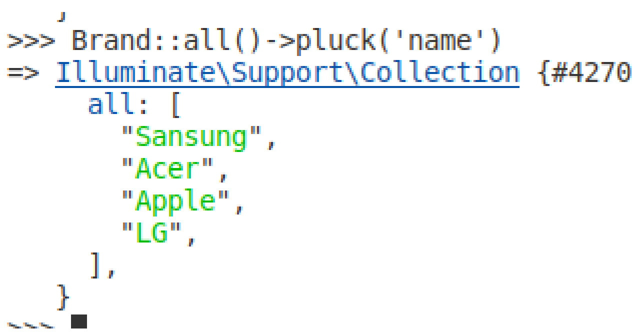
```
>>> Brand::all()->last()
=> App\Models\Brand {#4318
     id: 4,
     name: "LG",
     created_at: "2021-03-04
     updated_at: "2021-03-04
   }
```



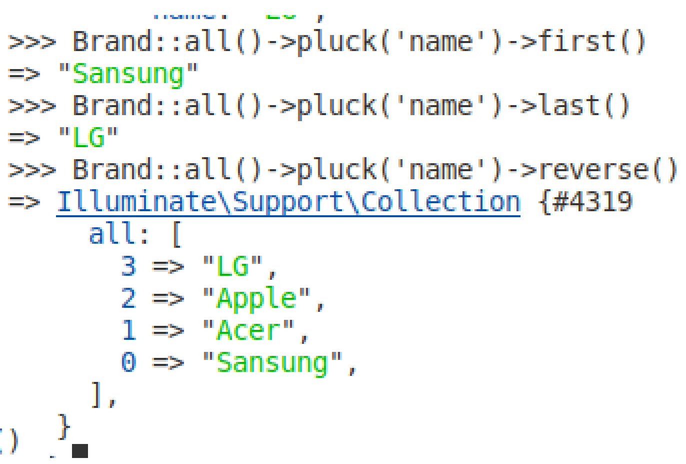
```
>>> Brand::all()->reverse()
=> Illuminate\Database\Eloquent\Collection {#4345
     all: [
       3 => App\Models\Brand {#4346
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 19:45:47",
       },
       2 => App\Models\Brand {#4290
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
       1 => App\Models\Brand {#4315
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       0 => App\Models\Brand {#4287
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
     ],
   }_
```

Trabalhando com Collection

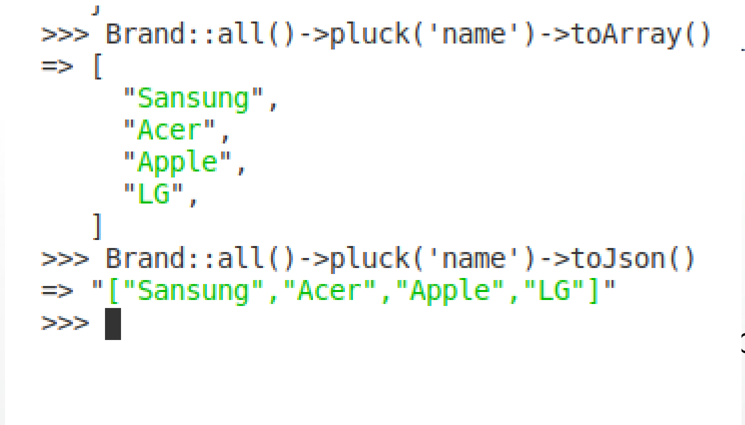
- Funções especiais de collections pluck ()



```
>>> Brand::all()->pluck('name')
=> Illuminate\Support\Collection {#4270
     all: [
       "Sansung",
       "Acer",
       "Apple",
       "LG",
     ],
   }
```



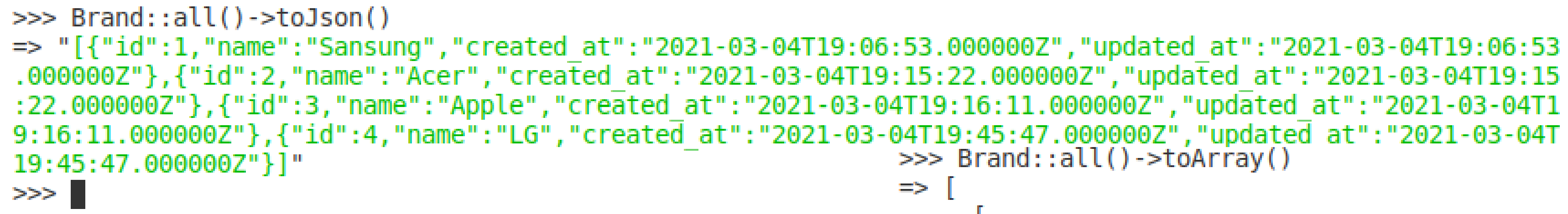
```
>>> Brand::all()->pluck('name')->first()
=> "Sansung"
>>> Brand::all()->pluck('name')->last()
=> "LG"
>>> Brand::all()->pluck('name')->reverse()
=> Illuminate\Support\Collection {#4319
     all: [
       3 => "LG",
       2 => "Apple",
       1 => "Acer",
       0 => "Sansung",
     ],
   }
```



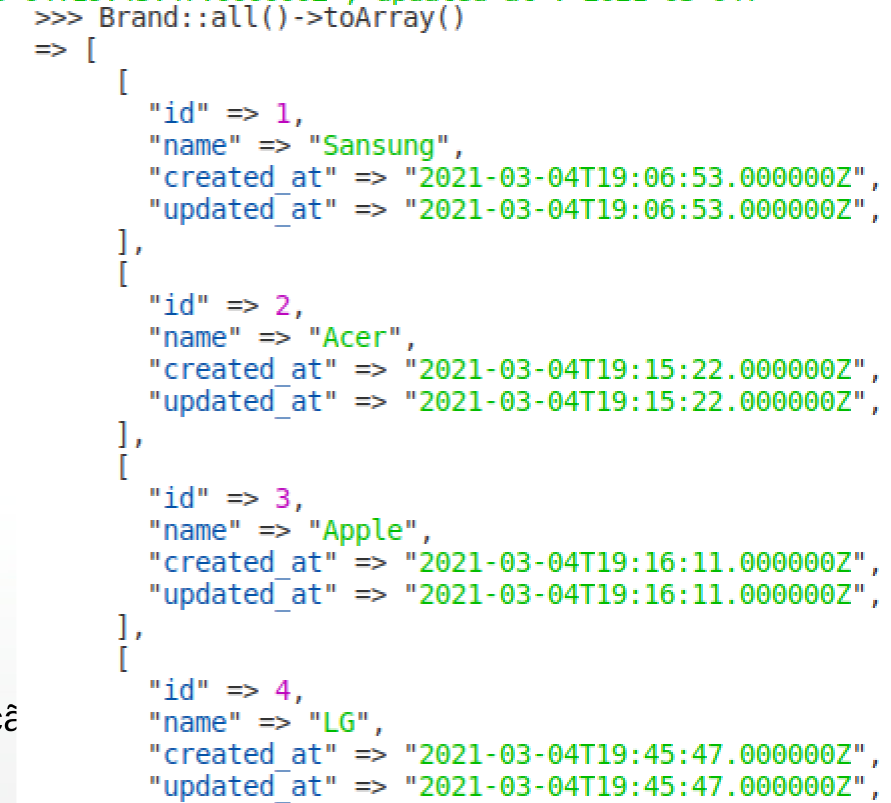
```
>>> Brand::all()->pluck('name')->toArray()
=> [
     "Sansung",
     "Acer",
     "Apple",
     "LG",
   ]
>>> Brand::all()->pluck('name')->toJson()
=> "["Sansung","Acer","Apple","LG"]"
>>>
```

Trabalhando com Collection

- Funções especiais de collections



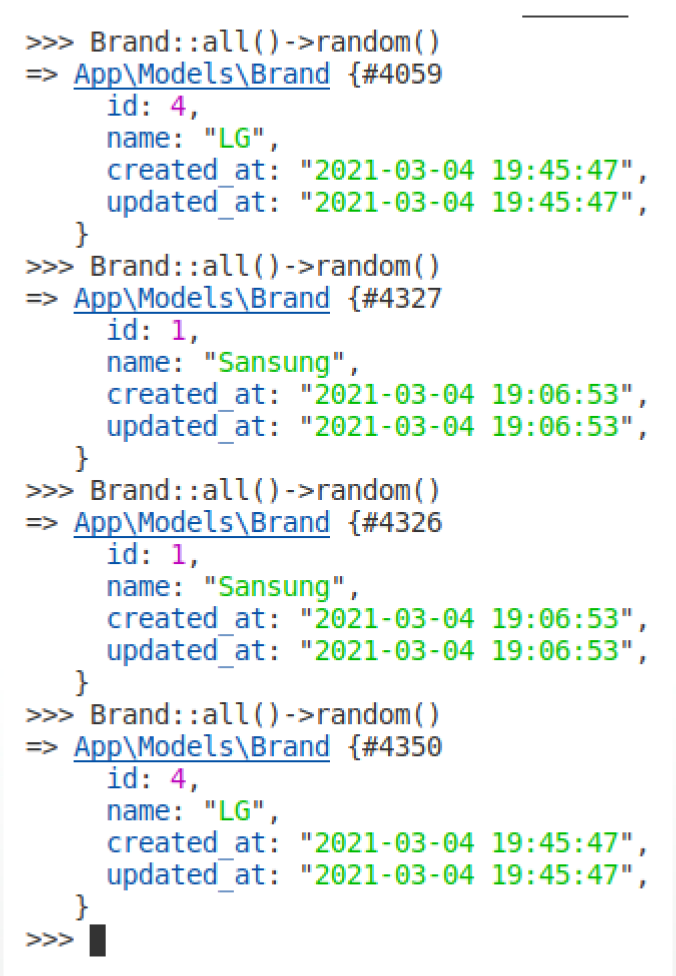
```
>>> Brand::all()->toJson()
=> "[{"id":1,"name":"Sansung","created_at":"2021-03-04T19:06:53.000000Z","updated_at":"2021-03-04T19:06:53
.000000Z"},{"id":2,"name":"Acer","created_at":"2021-03-04T19:15:22.000000Z","updated_at":"2021-03-04T19:15
:22.000000Z"},{"id":3,"name":"Apple","created_at":"2021-03-04T19:16:11.000000Z","updated_at":"2021-03-04T1
9:16:11.000000Z"},{"id":4,"name":"LG","created_at":"2021-03-04T19:45:47.000000Z","updated_at":"2021-03-04T
19:45:47.000000Z"}]"
>>> 
```



```
>>> Brand::all()->toArray()
=> [
     [
       "id" => 1,
       "name" => "Sansung",
       "created_at" => "2021-03-04T19:06:53.000000Z",
       "updated_at" => "2021-03-04T19:06:53.000000Z",
     ],
     [
       "id" => 2,
       "name" => "Acer",
       "created_at" => "2021-03-04T19:15:22.000000Z",
       "updated_at" => "2021-03-04T19:15:22.000000Z",
     ],
     [
       "id" => 3,
       "name" => "Apple",
       "created_at" => "2021-03-04T19:16:11.000000Z",
       "updated_at" => "2021-03-04T19:16:11.000000Z",
     ],
     [
       "id" => 4,
       "name" => "LG",
       "created_at" => "2021-03-04T19:45:47.000000Z",
       "updated_at" => "2021-03-04T19:45:47.000000Z",
```

Trabalhando com Collection

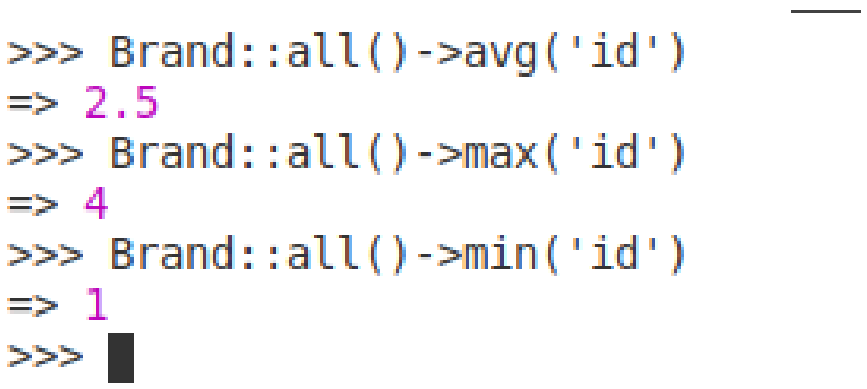
- Funções especiais de collections



```
>>> Brand::all()->random()
=> App\Models\Brand {#4059
     id: 4,
     name: "LG",
     created_at: "2021-03-04 19:45:47",
     updated_at: "2021-03-04 19:45:47",
   }
>>> Brand::all()->random()
=> App\Models\Brand {#4327
     id: 1,
     name: "Sansung",
     created_at: "2021-03-04 19:06:53",
     updated_at: "2021-03-04 19:06:53",
   }
>>> Brand::all()->random()
=> App\Models\Brand {#4326
     id: 1,
     name: "Sansung",
     created_at: "2021-03-04 19:06:53",
     updated_at: "2021-03-04 19:06:53",
   }
>>> Brand::all()->random()
=> App\Models\Brand {#4350
     id: 4,
     name: "LG",
     created_at: "2021-03-04 19:45:47",
     updated_at: "2021-03-04 19:45:47",
   }
>>> 
```

Trabalhando com Collection

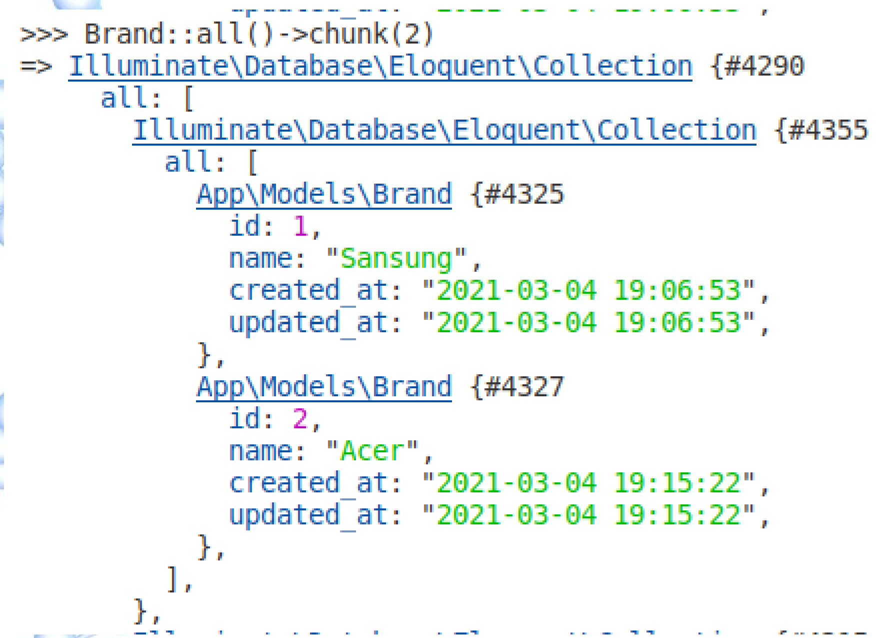
- Funções especiais de collections para cálculos



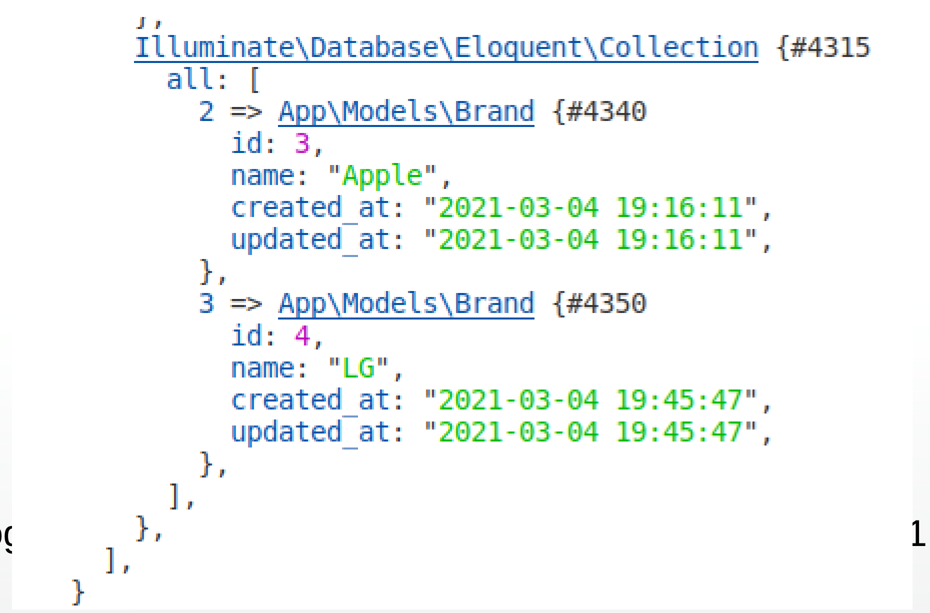
```
>>> Brand::all()->avg('id')
=> 2.5
>>> Brand::all()->max('id')
=> 4
>>> Brand::all()->min('id')
=> 1
>>> 
```

Trabalhando com Collection

Funções especiais de collections chunk()



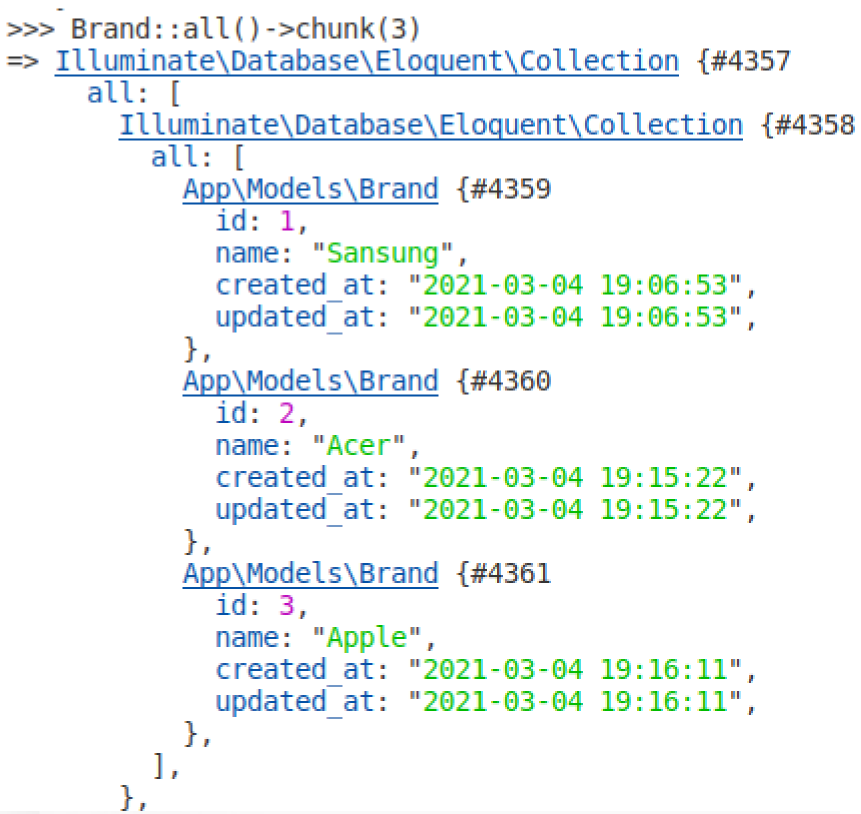
```
>>> Brand::all()->chunk(2)
=> Illuminate\Database\Eloquent\Collection {#4290
     all: [
       Illuminate\Database\Eloquent\Collection {#4355
         all: [
           App\Models\Brand {#4325
             id: 1,
             name: "Sansung",
             created_at: "2021-03-04 19:06:53",
             updated_at: "2021-03-04 19:06:53",
           },
           App\Models\Brand {#4327
             id: 2,
             name: "Acer",
             created_at: "2021-03-04 19:15:22",
             updated_at: "2021-03-04 19:15:22",
           },
         ],
       },
```



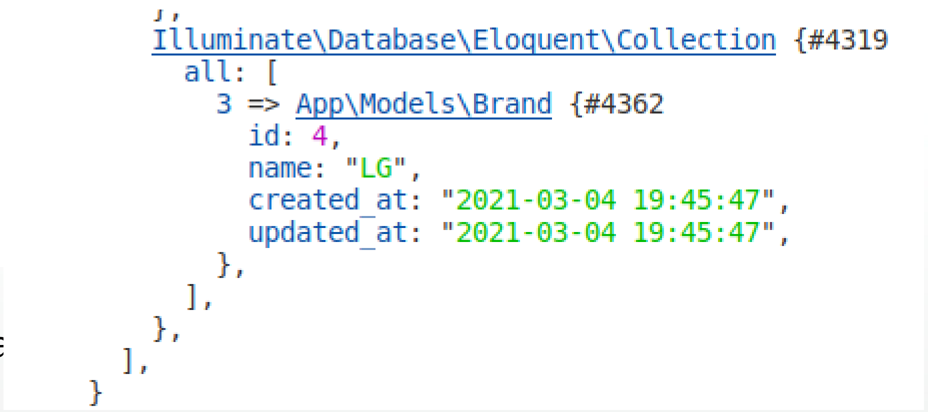
```
       Illuminate\Database\Eloquent\Collection {#4315
         all: [
           2 => App\Models\Brand {#4340
             id: 3,
             name: "Apple",
             created_at: "2021-03-04 19:16:11",
             updated_at: "2021-03-04 19:16:11",
           },
           3 => App\Models\Brand {#4350
             id: 4,
             name: "LG",
             created_at: "2021-03-04 19:45:47",
             updated_at: "2021-03-04 19:45:47",
           },
         ],
       },
     ],
   }
```

Trabalhando com Collection

Funções especiais de collections chunk()



```
>>> Brand::all()->chunk(3)
=> Illuminate\Database\Eloquent\Collection {#4357
     all: [
       Illuminate\Database\Eloquent\Collection {#4358
         all: [
           App\Models\Brand {#4359
             id: 1,
             name: "Sansung",
             created_at: "2021-03-04 19:06:53",
             updated_at: "2021-03-04 19:06:53",
           },
           App\Models\Brand {#4360
             id: 2,
             name: "Acer",
             created_at: "2021-03-04 19:15:22",
             updated_at: "2021-03-04 19:15:22",
           },
           App\Models\Brand {#4361
             id: 3,
             name: "Apple",
             created_at: "2021-03-04 19:16:11",
             updated_at: "2021-03-04 19:16:11",
           },
         ],
       },
```



```
       },
       Illuminate\Database\Eloquent\Collection {#4319
         all: [
           3 => App\Models\Brand {#4362
             id: 4,
             name: "LG",
             created_at: "2021-03-04 19:45:47",
             updated_at: "2021-03-04 19:45:47",
           },
         ],
       },
     ],
   }
```

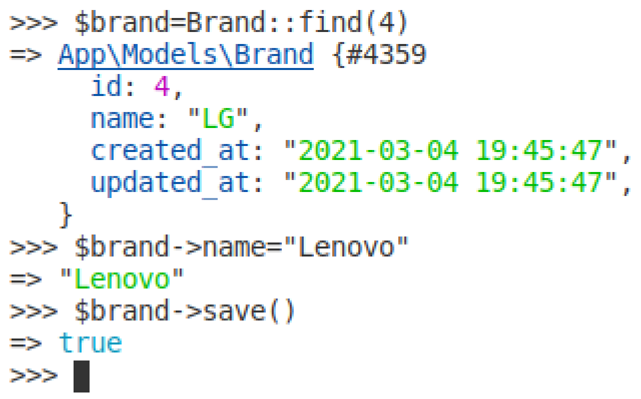
Collections

- Classe collection no laravel é classe que fornece diversas funções para trabalhar com o Laravel e banco de dados
- Ver: https://laravel.com/docs/8.x/collections
-

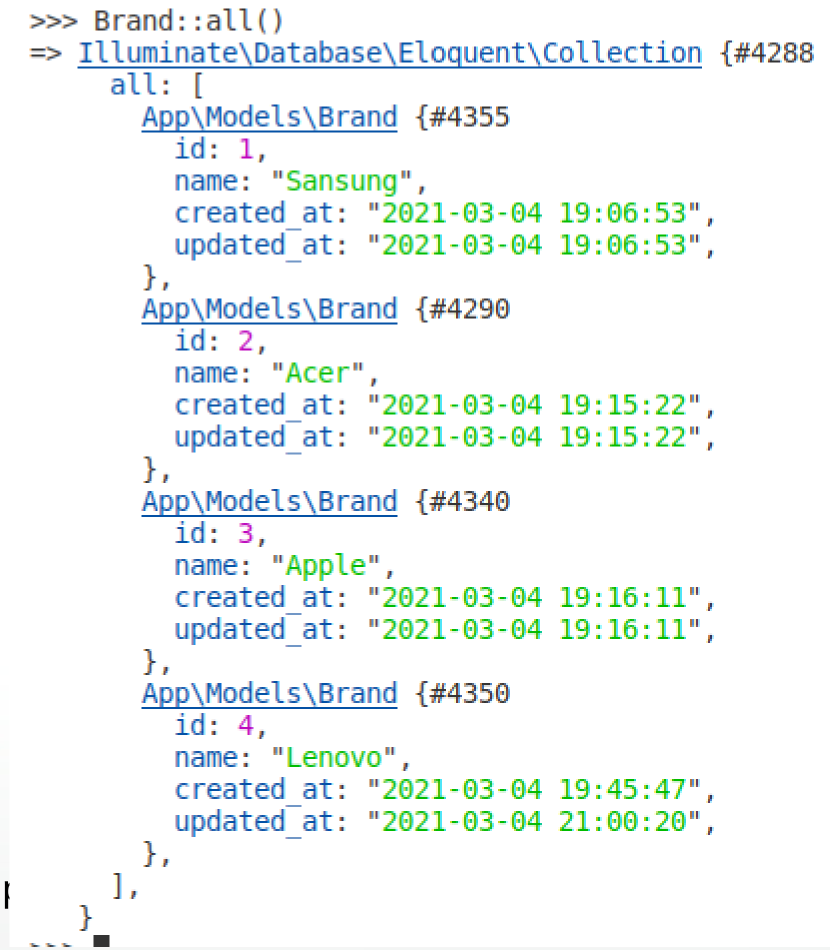
INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Atualizando registros

- Vamos atualizar id=4 p/ Lenovo



```
>>> $brand=Brand::find(4)
=> App\Models\Brand {#4359
     id: 4,
     name: "LG",
     created_at: "2021-03-04 19:45:47",
     updated_at: "2021-03-04 19:45:47",
   }
>>> $brand->name="Lenovo"
=> "Lenovo"
>>> $brand->save()
=> true
>>> 
```



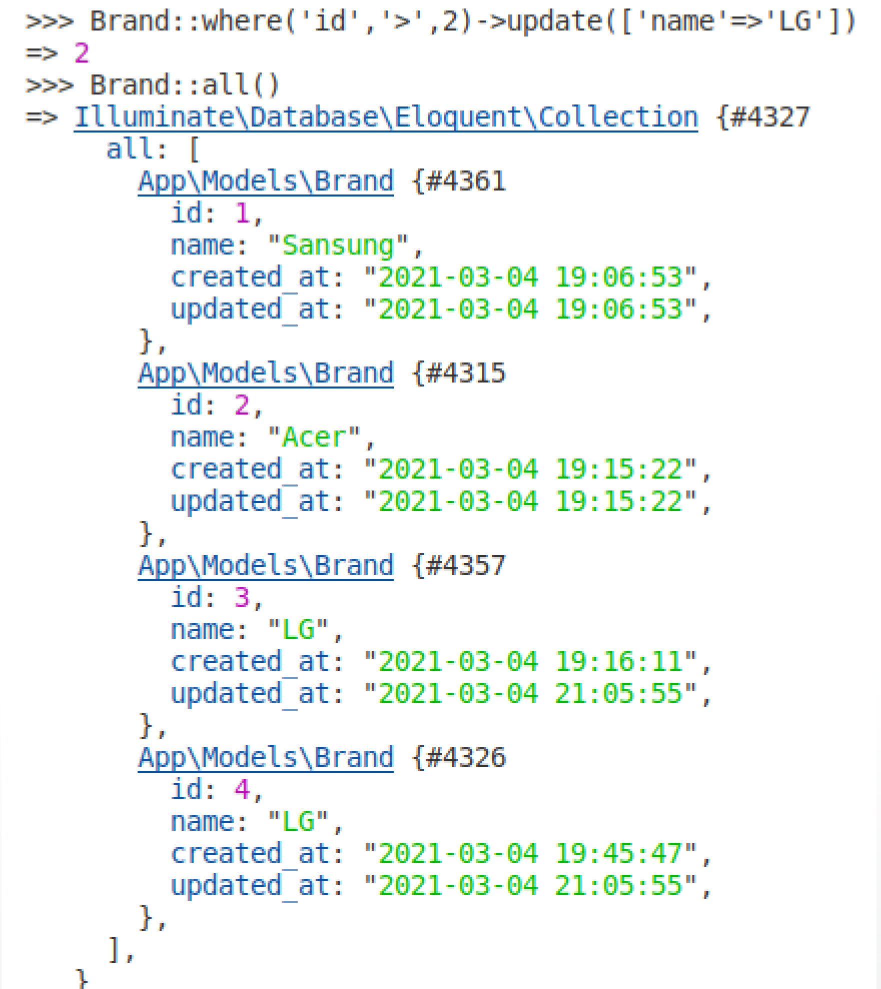
```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4288
     all: [
       App\Models\Brand {#4355
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4290
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4340
         id: 3,
         name: "Apple",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 19:16:11",
       },
       App\Models\Brand {#4350
         id: 4,
         name: "Lenovo",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 21:00:20",
       },
     ],
   }
```

Atualizando registros

- Com array associativo
- Função fill
- Muito útil, para fazer update com request -> all()
- Vamos alterar novamente
-

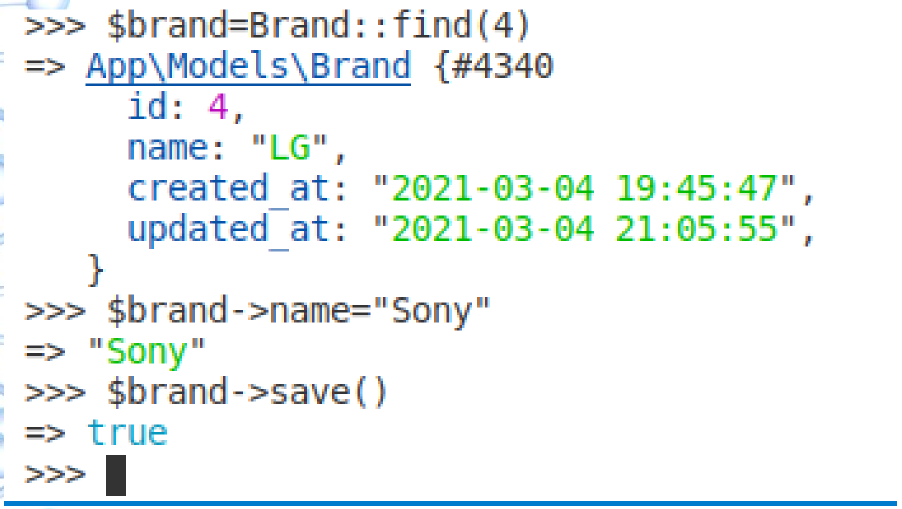
Atualizando registros simultaneamente

- Utilizando o where podemos alterar registros simultanemente

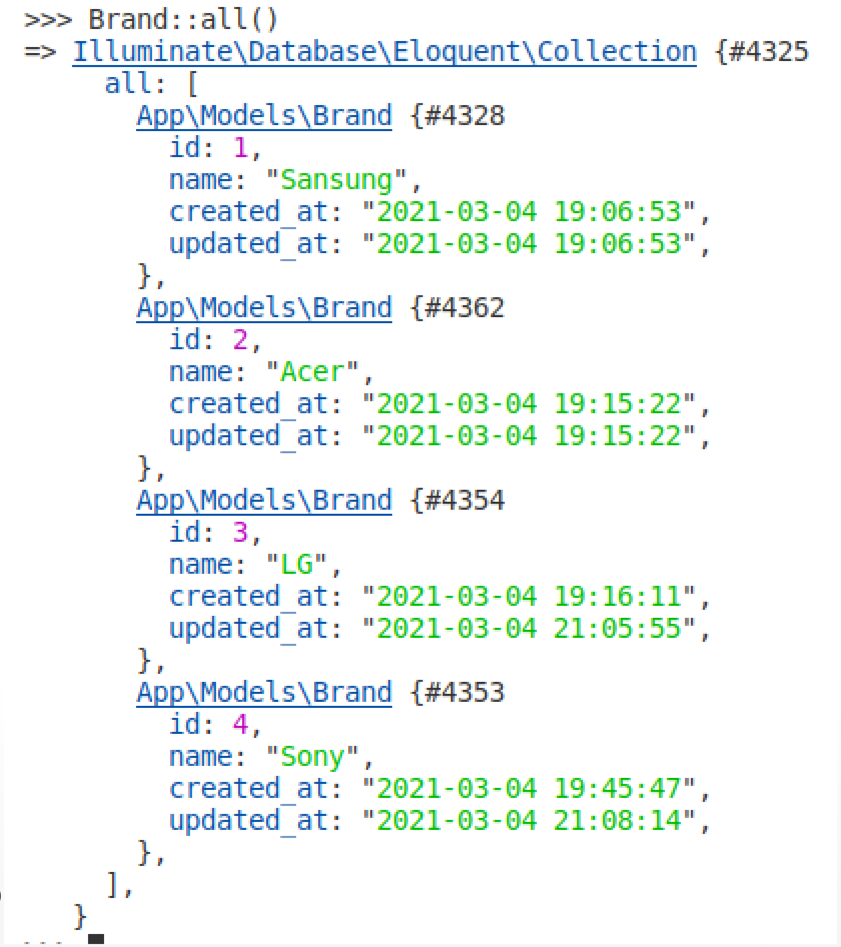


```
>>> Brand::where('id','>',2)->update(['name'=>'LG'])
=> 2
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4327
     all: [
       App\Models\Brand {#4361
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4315
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4357
         id: 3,
         name: "LG",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 21:05:55",
       },
       App\Models\Brand {#4326
         id: 4,
         name: "LG",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 21:05:55",
       },
     ],
   }
```

Atualizando novamente o ID 4 para Sony



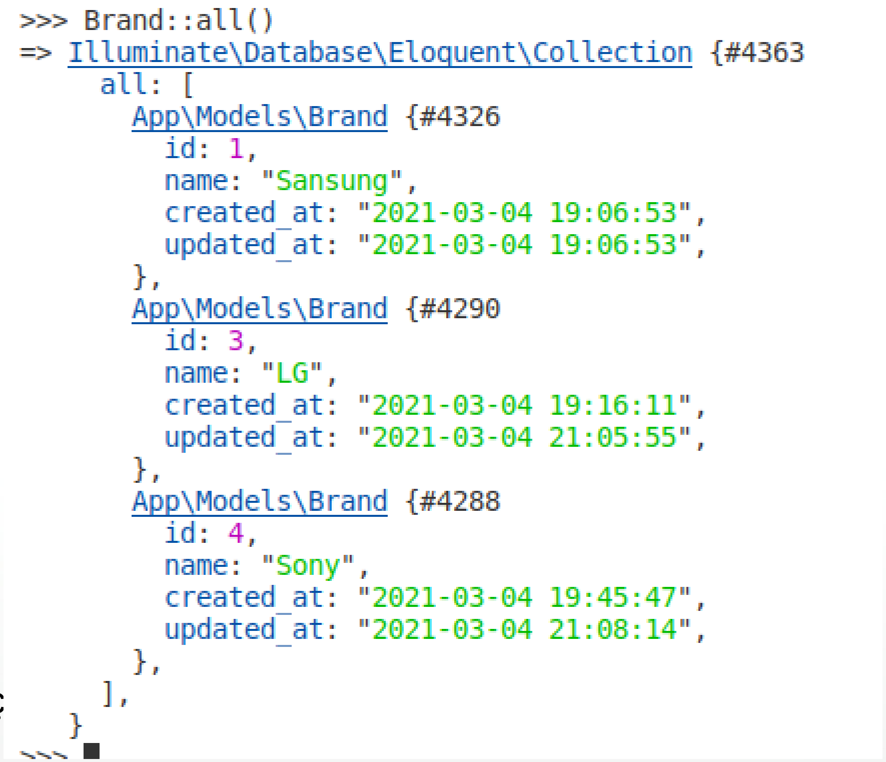
```
>>> $brand=Brand::find(4)
=> App\Models\Brand {#4340
     id: 4,
     name: "LG",
     created_at: "2021-03-04 19:45:47",
     updated_at: "2021-03-04 21:05:55",
   }
>>> $brand->name="Sony"
=> "Sony"
>>> $brand->save()
=> true
>>> 
```



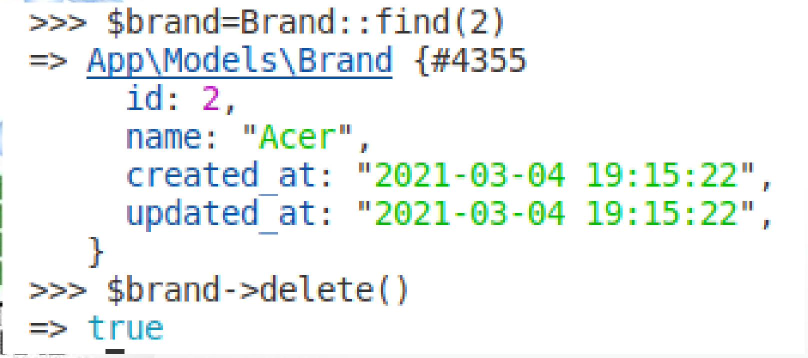
```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4325
     all: [
       App\Models\Brand {#4328
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4362
         id: 2,
         name: "Acer",
         created_at: "2021-03-04 19:15:22",
         updated_at: "2021-03-04 19:15:22",
       },
       App\Models\Brand {#4354
         id: 3,
         name: "LG",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 21:05:55",
       },
       App\Models\Brand {#4353
         id: 4,
         name: "Sony",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 21:08:14",
       },
     ],
   }
```

Apagando registros

- Temos várias formas de apagar registros da tabela
- Usando uma variável



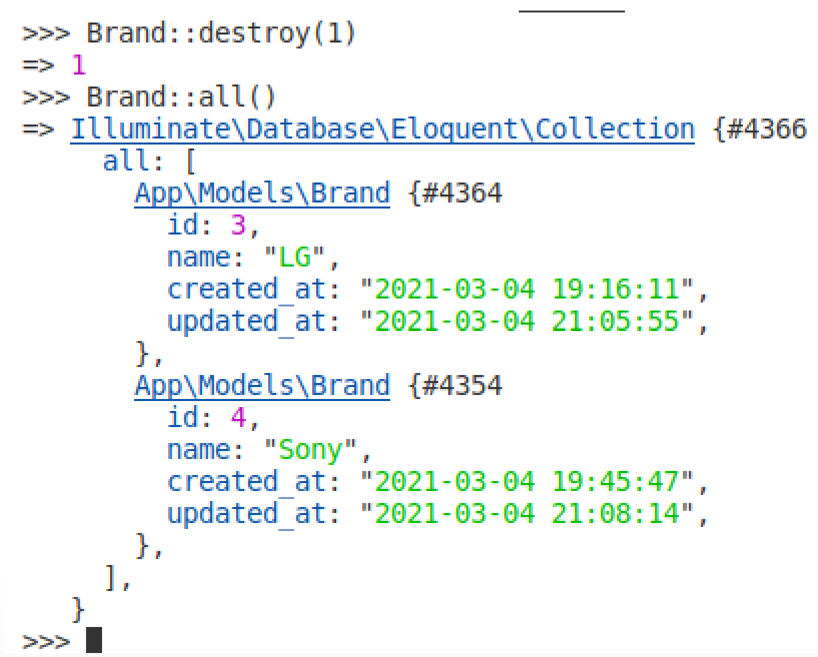
```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4363
     all: [
       App\Models\Brand {#4326
         id: 1,
         name: "Sansung",
         created_at: "2021-03-04 19:06:53",
         updated_at: "2021-03-04 19:06:53",
       },
       App\Models\Brand {#4290
         id: 3,
         name: "LG",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 21:05:55",
       },
       App\Models\Brand {#4288
         id: 4,
         name: "Sony",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 21:08:14",
       },
     ],
   }
>>> 
```



```
>>> $brand=Brand::find(2)
=> App\Models\Brand {#4355
     id: 2,
     name: "Acer",
     created_at: "2021-03-04 19:15:22",
     updated_at: "2021-03-04 19:15:22",
   }
>>> $brand->delete()
=> true
```

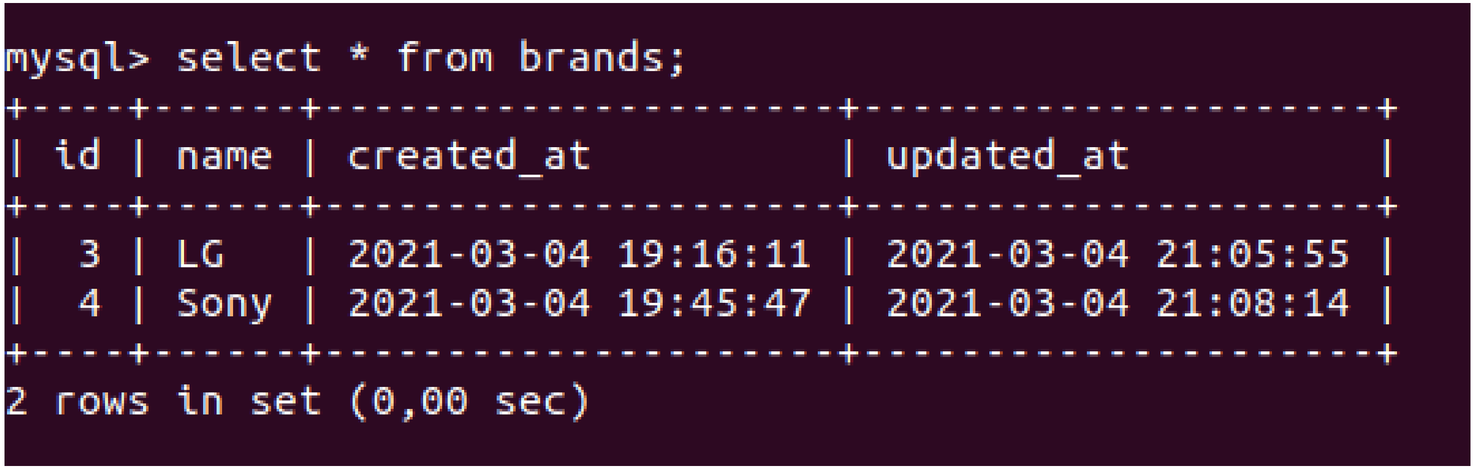
Apagando registros

- Temos várias formas de apagar registros da tabela
- Usando id, exemplo id=1



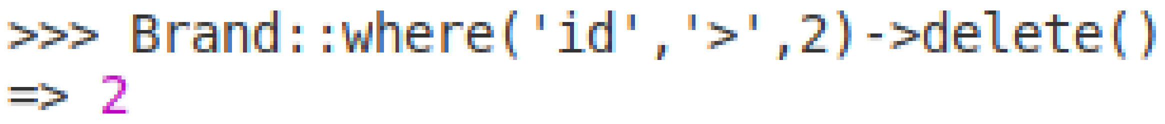
```
>>> Brand::destroy(1)
=> 1
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4366
     all: [
       App\Models\Brand {#4364
         id: 3,
         name: "LG",
         created_at: "2021-03-04 19:16:11",
         updated_at: "2021-03-04 21:05:55",
       },
       App\Models\Brand {#4354
         id: 4,
         name: "Sony",
         created_at: "2021-03-04 19:45:47",
         updated_at: "2021-03-04 21:08:14",
       },
     ],
   }
>>> 
```

Verificando a tabela no banco

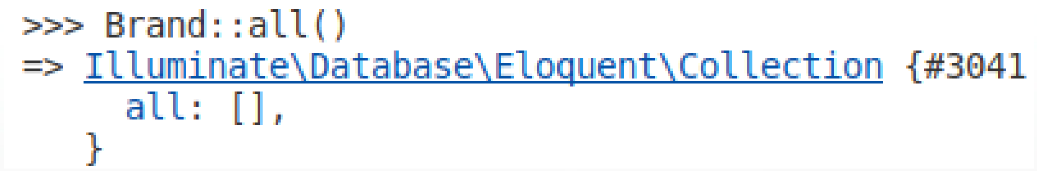

```
mysql> select * from brands;
+----+------+---------------------+---------------------+
| id | name | created_at          | updated_at          |
+----+------+---------------------+---------------------+
|  3 | LG   | 2021-03-04 19:16:11 | 2021-03-04 21:05:55 |
|  4 | Sony | 2021-03-04 19:45:47 | 2021-03-04 21:08:14 |
+----+------+---------------------+---------------------+
2 rows in set (0,00 sec)
```

Apagando registros

- Temos várias formas de apagar registros da tabela
- Usando where, numa query



```
>>> Brand::where('id','>',2)->delete()
=> 2
```



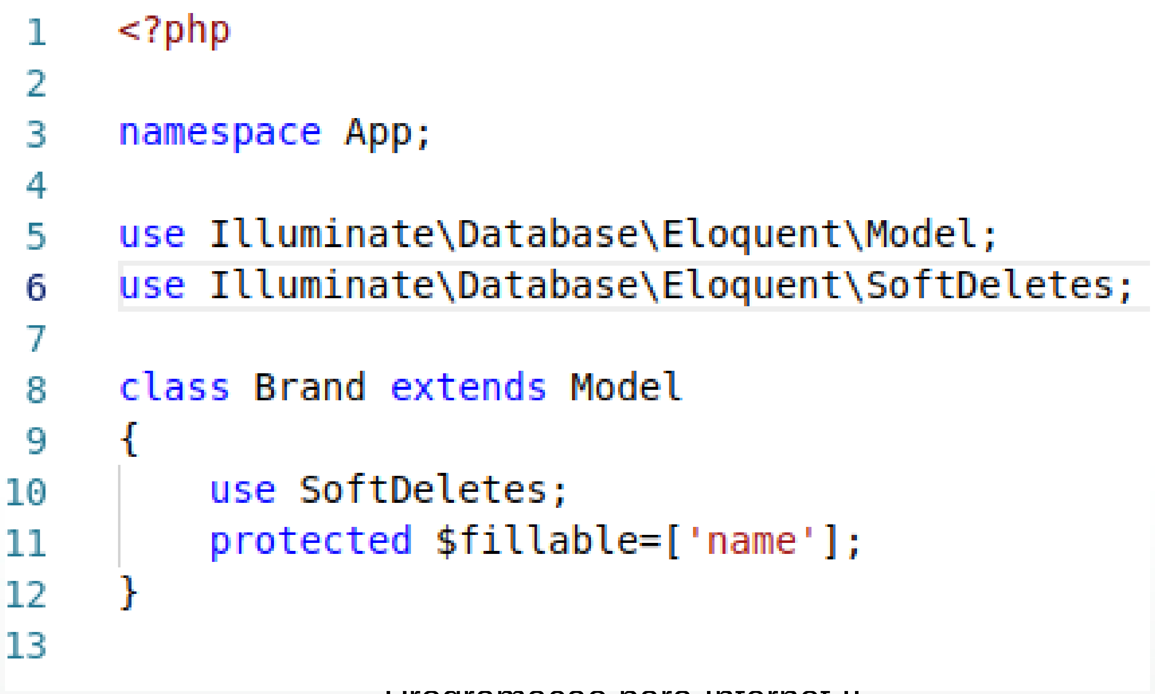
```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#3041
     all: [],
   }
```

Soft delete

- É o conceito de apagar um registro da tabela, sem que seja apagado, efetivamente da base de dados
- O Laravel permite implementar o soft delete
- Para implementar, devemos alterar:
  - Model
  - Migration

Model

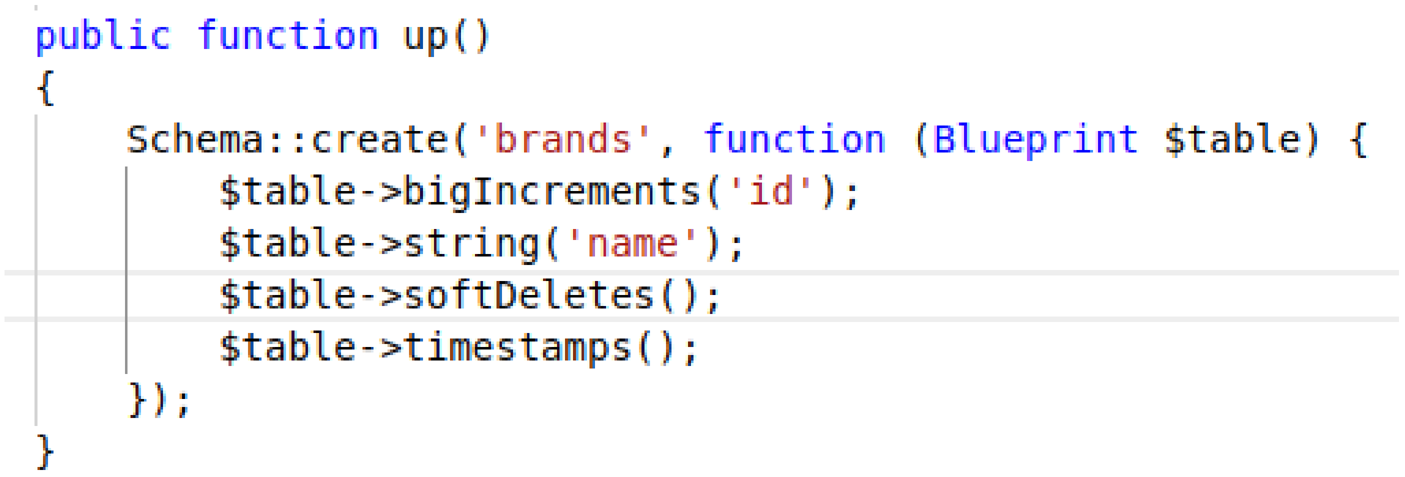
Informar que vai usar
Usar o SoftDelete



```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Brand extends Model
{
    use SoftDeletes;
    protected $fillable=['name'];
}

```

Migration

Implementar o campo de SoftDelete na tabela



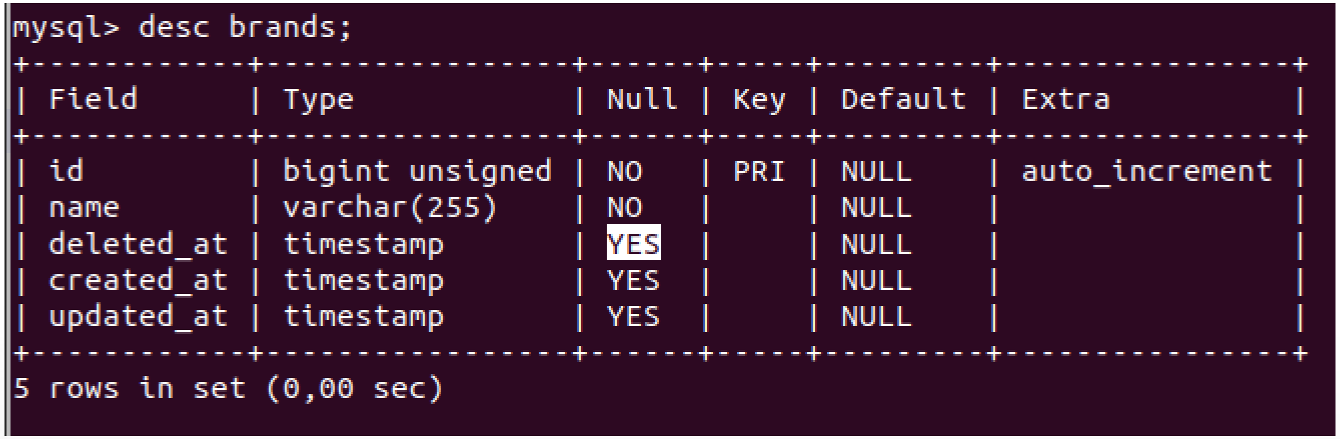
```php
public function up()
{
    Schema::create('brands', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->string('name');
        $table->softDeletes();
        $table->timestamps();
    });
}
```

Efetivar o migration

- Como alteramos a classe migration, devemos efetivar essas alterações
- O comando fresh, vai apagar todas as tabelas e vai criar tudo novamente

```
php artisan migrate:fresh
```

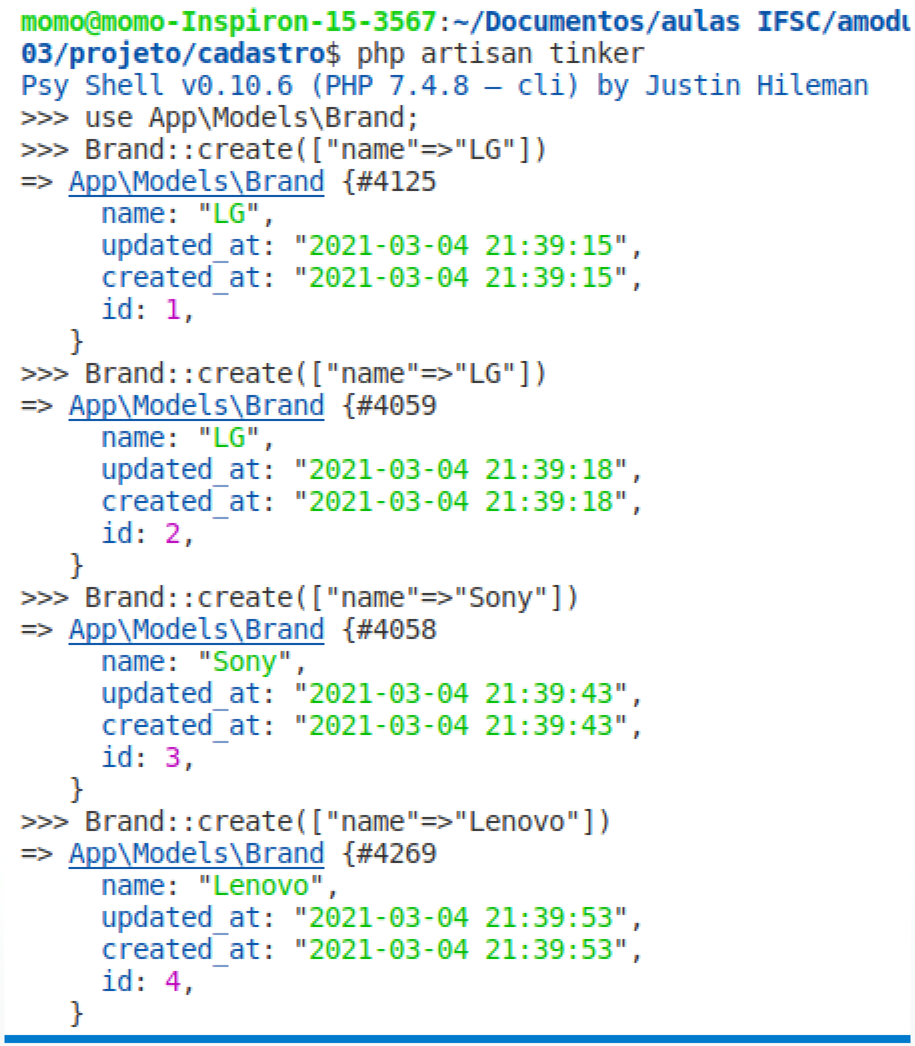
Verificar no SQL campo criado

- O campo delete_at
- Guardará um registro da hora em que foi apagado
- 

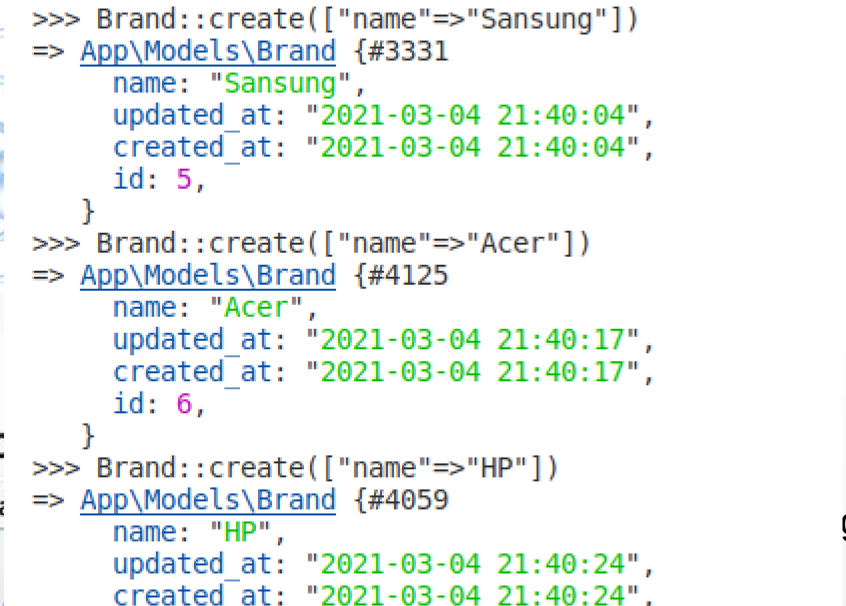
```
mysql> desc brands;
+------------+-----------------+------+-----+---------+----------------+
| Field      | Type            | Null | Key | Default | Extra          |
+------------+-----------------+------+-----+---------+----------------+
| id         | bigint unsigned | NO   | PRI | NULL    | auto_increment |
| name       | varchar(255)    | NO   |     | NULL    |                |
| deleted_at | timestamp       | YES  |     | NULL    |                |
| created_at | timestamp       | YES  |     | NULL    |                |
| updated_at | timestamp       | YES  |     | NULL    |                |
+------------+-----------------+------+-----+---------+----------------+
5 rows in set (0,00 sec)
```

Criar os brands

- Pelo tinker
  - php artisan tinker
  - use \App\Models\Brand

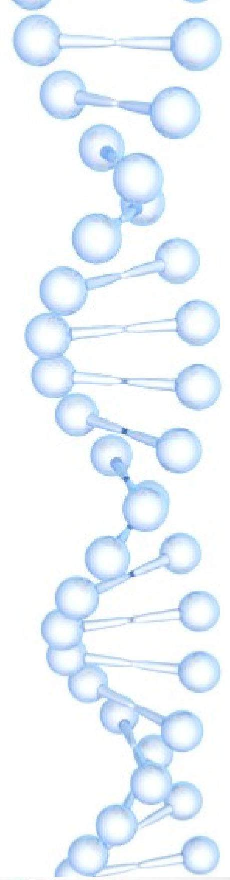


```
momo@momo-Inspiron-15-3567:~/Documentos/aulas IFSC/amodu
03/projeto/cadastro$ php artisan tinker
Psy Shell v0.10.6 (PHP 7.4.8 — cli) by Justin Hileman
>>> use App\Models\Brand;
>>> Brand::create(["name"=>"LG"])
=> App\Models\Brand {#4125
     name: "LG",
     updated_at: "2021-03-04 21:39:15",
     created_at: "2021-03-04 21:39:15",
     id: 1,
   }
>>> Brand::create(["name"=>"LG"])
=> App\Models\Brand {#4059
     name: "LG",
     updated_at: "2021-03-04 21:39:18",
     created_at: "2021-03-04 21:39:18",
     id: 2,
   }
>>> Brand::create(["name"=>"Sony"])
=> App\Models\Brand {#4058
     name: "Sony",
     updated_at: "2021-03-04 21:39:43",
     created_at: "2021-03-04 21:39:43",
     id: 3,
   }
>>> Brand::create(["name"=>"Lenovo"])
=> App\Models\Brand {#4269
     name: "Lenovo",
     updated_at: "2021-03-04 21:39:53",
     created_at: "2021-03-04 21:39:53",
     id: 4,
   }
```

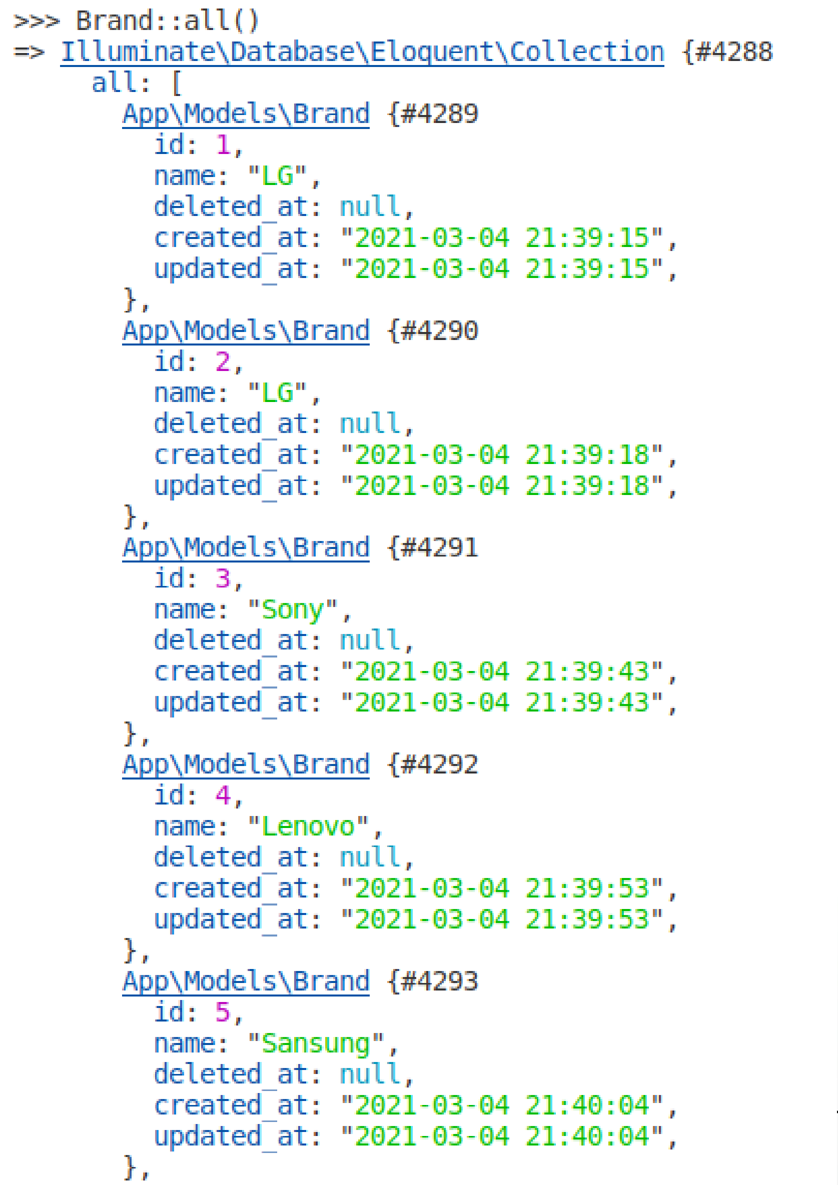


```
>>> Brand::create(["name"=>"Sansung"])
=> App\Models\Brand {#3331
     name: "Sansung",
     updated_at: "2021-03-04 21:40:04",
     created_at: "2021-03-04 21:40:04",
     id: 5,
   }
>>> Brand::create(["name"=>"Acer"])
=> App\Models\Brand {#4125
     name: "Acer",
     updated_at: "2021-03-04 21:40:17",
     created_at: "2021-03-04 21:40:17",
     id: 6,
   }
>>> Brand::create(["name"=>"HP"])
=> App\Models\Brand {#4059
     name: "HP",
     updated_at: "2021-03-04 21:40:24",
     created_at: "2021-03-04 21:40:24",
```

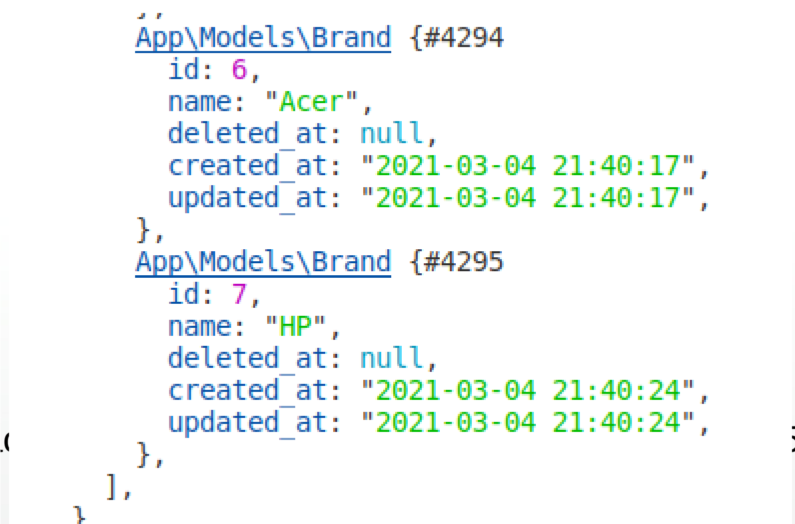
INSTITUTO FEDERAL Santa Catarina Câmpus Gaspar



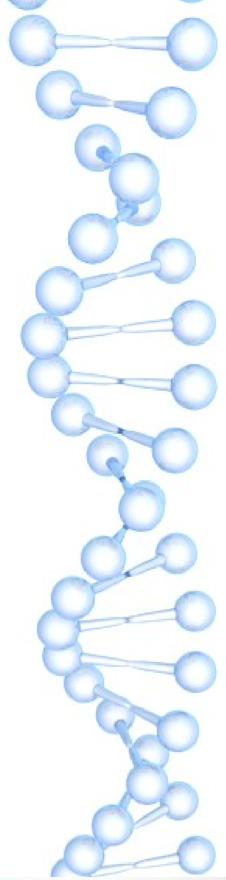
Listar brands



```
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4288
     all: [
       App\Models\Brand {#4289
         id: 1,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:15",
         updated_at: "2021-03-04 21:39:15",
       },
       App\Models\Brand {#4290
         id: 2,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:18",
         updated_at: "2021-03-04 21:39:18",
       },
       App\Models\Brand {#4291
         id: 3,
         name: "Sony",
         deleted_at: null,
         created_at: "2021-03-04 21:39:43",
         updated_at: "2021-03-04 21:39:43",
       },
       App\Models\Brand {#4292
         id: 4,
         name: "Lenovo",
         deleted_at: null,
         created_at: "2021-03-04 21:39:53",
         updated_at: "2021-03-04 21:39:53",
       },
       App\Models\Brand {#4293
         id: 5,
         name: "Sansung",
         deleted_at: null,
         created_at: "2021-03-04 21:40:04",
         updated_at: "2021-03-04 21:40:04",
       },
```

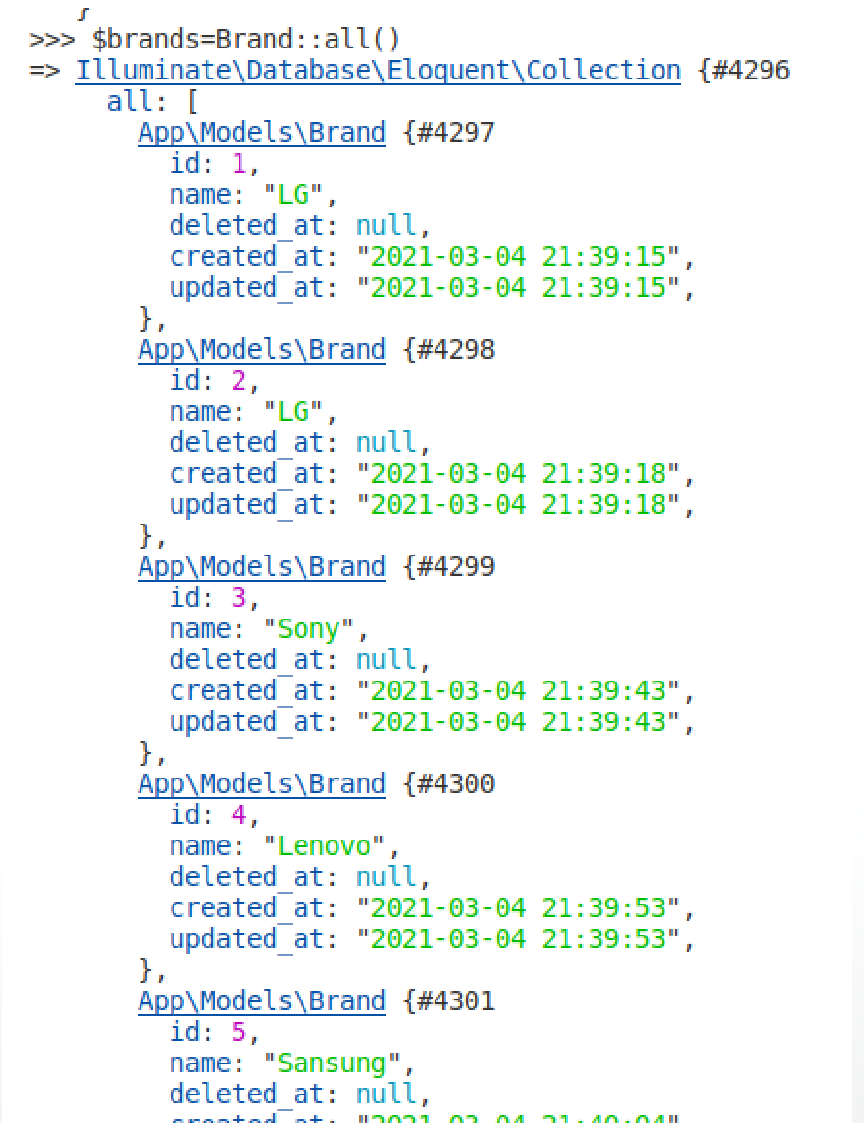


```
       },
       App\Models\Brand {#4294
         id: 6,
         name: "Acer",
         deleted_at: null,
         created_at: "2021-03-04 21:40:17",
         updated_at: "2021-03-04 21:40:17",
       },
       App\Models\Brand {#4295
         id: 7,
         name: "HP",
         deleted_at: null,
         created_at: "2021-03-04 21:40:24",
         updated_at: "2021-03-04 21:40:24",
       },
     ],
   }
```



Criar a var $brands

- Vamos criar uma variável e carregar o array de Brand

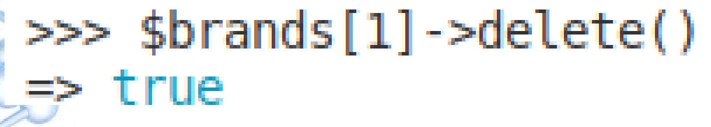


```
>>> $brands=Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4296
     all: [
       App\Models\Brand {#4297
         id: 1,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:15",
         updated_at: "2021-03-04 21:39:15",
       },
       App\Models\Brand {#4298
         id: 2,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:18",
         updated_at: "2021-03-04 21:39:18",
       },
       App\Models\Brand {#4299
         id: 3,
         name: "Sony",
         deleted_at: null,
         created_at: "2021-03-04 21:39:43",
         updated_at: "2021-03-04 21:39:43",
       },
       App\Models\Brand {#4300
         id: 4,
         name: "Lenovo",
         deleted_at: null,
         created_at: "2021-03-04 21:39:53",
         updated_at: "2021-03-04 21:39:53",
       },
       App\Models\Brand {#4301
         id: 5,
         name: "Sansung",
         deleted_at: null,
         created_at: "2021-03-04 21:40:04"
```

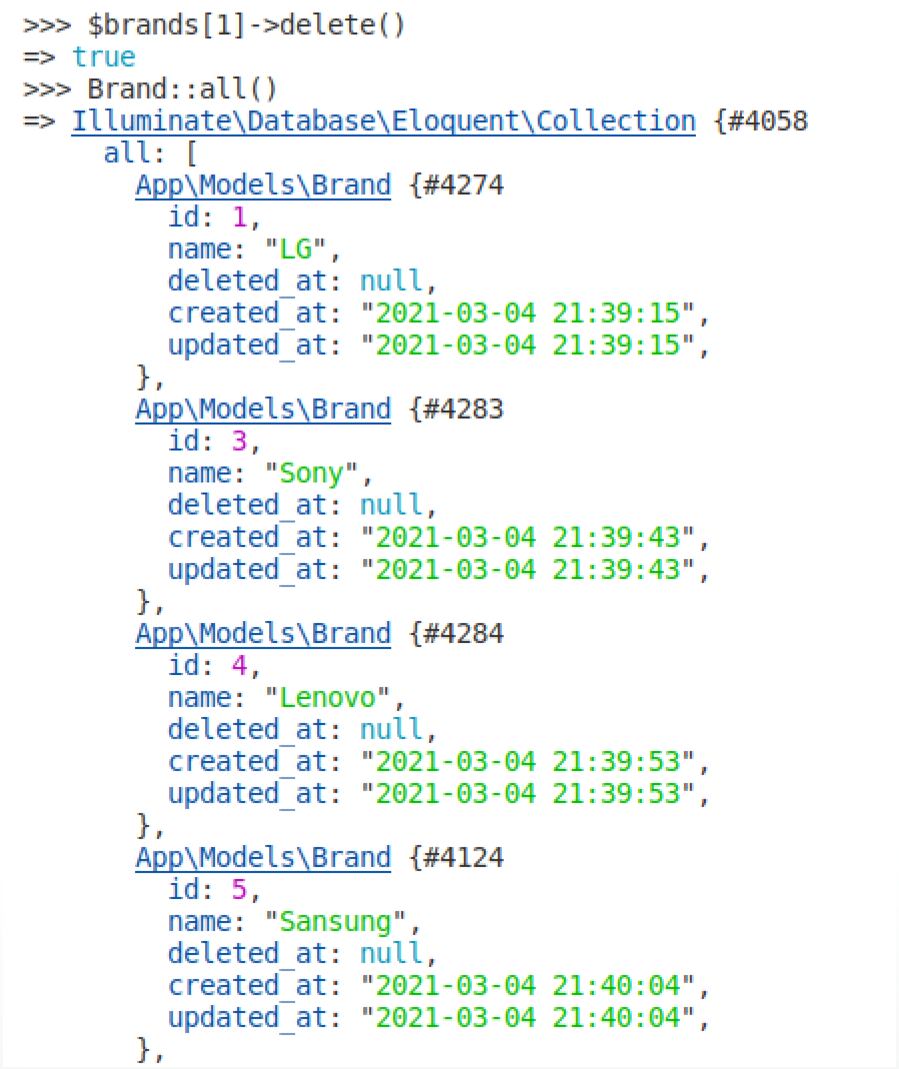
INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Apagar brands

- Uma vez que tenhamos o array de Brand passamos o índice e a função delete()



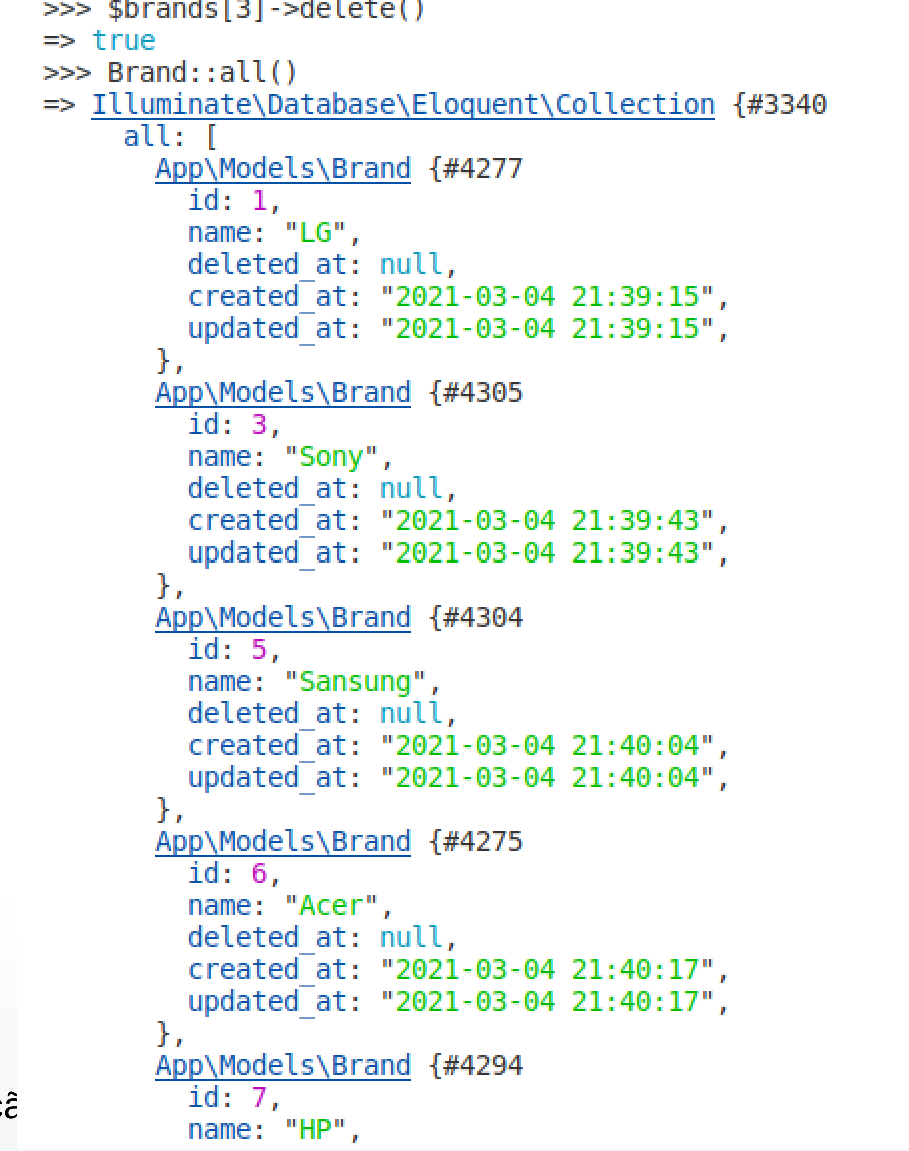
```
>>> $brands[1]->delete()
=> true
```



```
>>> $brands[1]->delete()
=> true
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4058
     all: [
       App\Models\Brand {#4274
         id: 1,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:15",
         updated_at: "2021-03-04 21:39:15",
       },
       App\Models\Brand {#4283
         id: 3,
         name: "Sony",
         deleted_at: null,
         created_at: "2021-03-04 21:39:43",
         updated_at: "2021-03-04 21:39:43",
       },
       App\Models\Brand {#4284
         id: 4,
         name: "Lenovo",
         deleted_at: null,
         created_at: "2021-03-04 21:39:53",
         updated_at: "2021-03-04 21:39:53",
       },
       App\Models\Brand {#4124
         id: 5,
         name: "Sansung",
         deleted_at: null,
         created_at: "2021-03-04 21:40:04",
         updated_at: "2021-03-04 21:40:04",
       },
```

Apagar brands

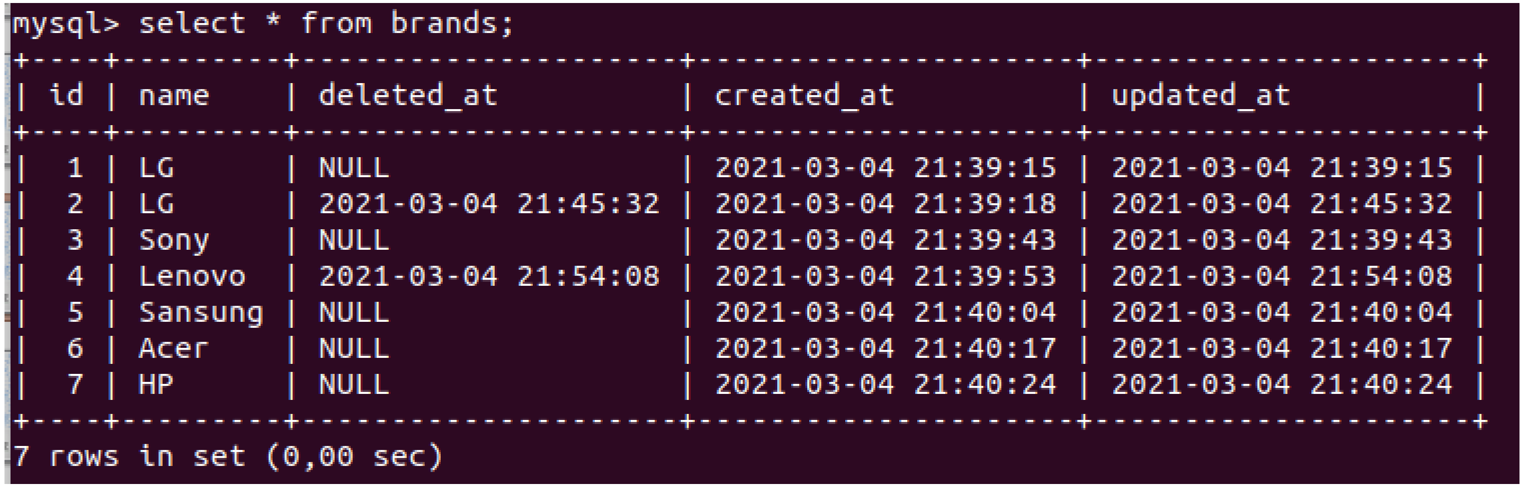
- Uma vez que tenhamos o array de Brand passamos o índice e a função delete()
- Apagando o índice 3
- LG e Lenovo com soft delete



```
>>> $brands[3]->delete()
=> true
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#3340
     all: [
       App\Models\Brand {#4277
         id: 1,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:15",
         updated_at: "2021-03-04 21:39:15",
       },
       App\Models\Brand {#4305
         id: 3,
         name: "Sony",
         deleted_at: null,
         created_at: "2021-03-04 21:39:43",
         updated_at: "2021-03-04 21:39:43",
       },
       App\Models\Brand {#4304
         id: 5,
         name: "Sansung",
         deleted_at: null,
         created_at: "2021-03-04 21:40:04",
         updated_at: "2021-03-04 21:40:04",
       },
       App\Models\Brand {#4275
         id: 6,
         name: "Acer",
         deleted_at: null,
         created_at: "2021-03-04 21:40:17",
         updated_at: "2021-03-04 21:40:17",
       },
       App\Models\Brand {#4294
         id: 7,
         name: "HP",
```

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Após a operação de soft delete

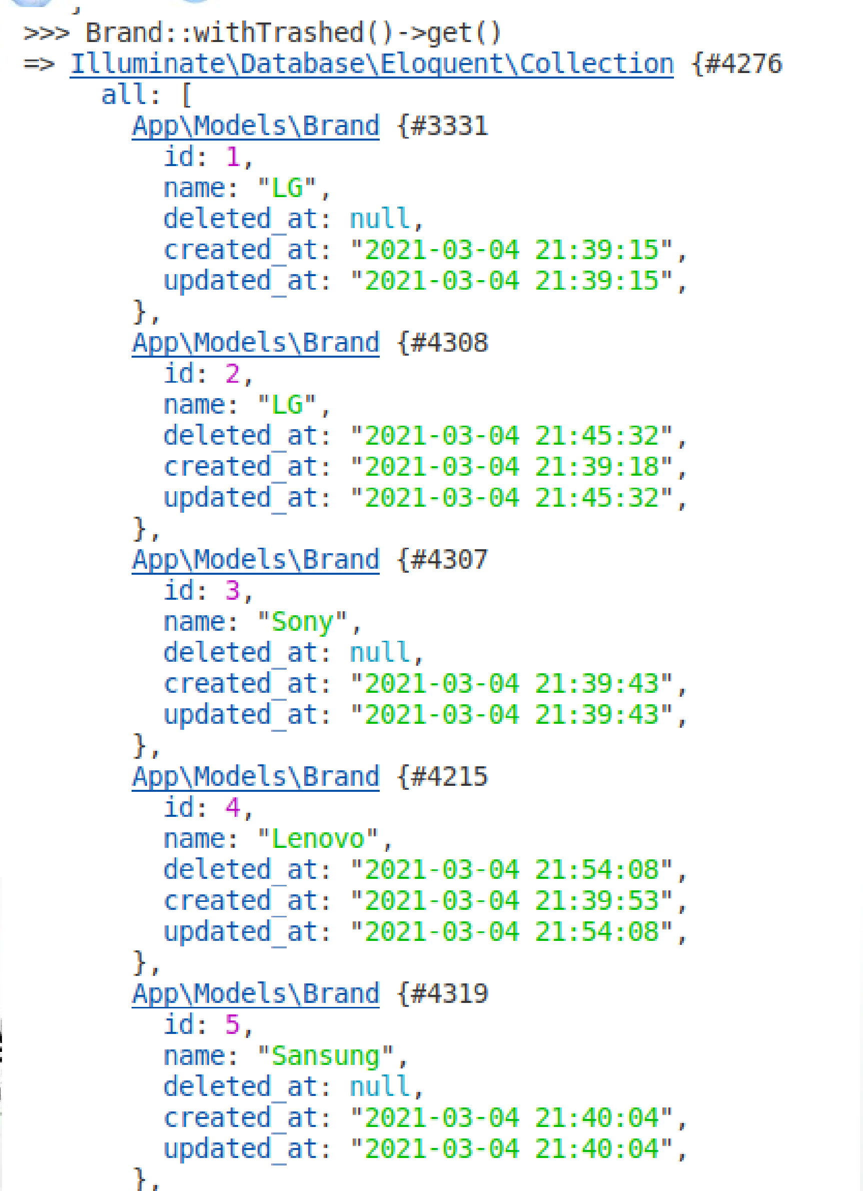


```
mysql> select * from brands;
+----+---------+---------------------+---------------------+---------------------+
| id | name    | deleted_at          | created_at          | updated_at          |
+----+---------+---------------------+---------------------+---------------------+
|  1 | LG      | NULL                | 2021-03-04 21:39:15 | 2021-03-04 21:39:15 |
|  2 | LG      | 2021-03-04 21:45:32 | 2021-03-04 21:39:18 | 2021-03-04 21:45:32 |
|  3 | Sony    | NULL                | 2021-03-04 21:39:43 | 2021-03-04 21:39:43 |
|  4 | Lenovo  | 2021-03-04 21:54:08 | 2021-03-04 21:39:53 | 2021-03-04 21:54:08 |
|  5 | Sansung | NULL                | 2021-03-04 21:40:04 | 2021-03-04 21:40:04 |
|  6 | Acer    | NULL                | 2021-03-04 21:40:17 | 2021-03-04 21:40:17 |
|  7 | HP      | NULL                | 2021-03-04 21:40:24 | 2021-03-04 21:40:24 |
+----+---------+---------------------+---------------------+---------------------+
7 rows in set (0,00 sec)
```

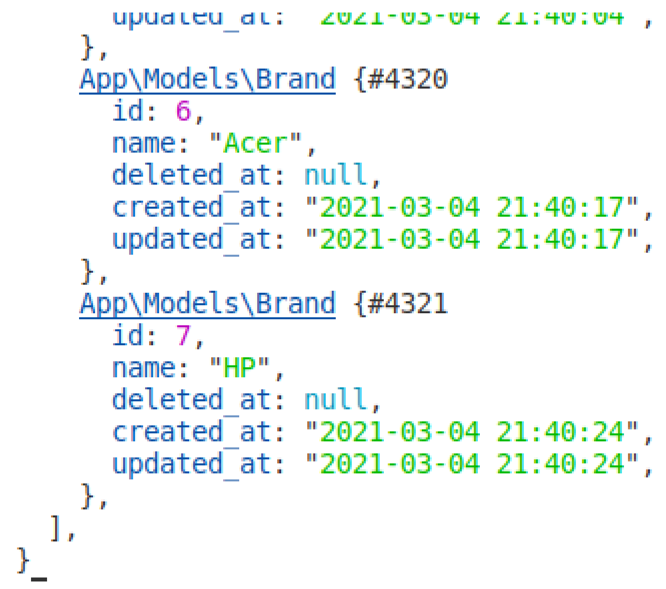
Mostrar novamente

- Através do comado mostra os registros apagados

```
Brand::withTrashed()->get()
```



```
>>> Brand::withTrashed()->get()
=> Illuminate\Database\Eloquent\Collection {#4276
     all: [
       App\Models\Brand {#3331
         id: 1,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:15",
         updated_at: "2021-03-04 21:39:15",
       },
       App\Models\Brand {#4308
         id: 2,
         name: "LG",
         deleted_at: "2021-03-04 21:45:32",
         created_at: "2021-03-04 21:39:18",
         updated_at: "2021-03-04 21:45:32",
       },
       App\Models\Brand {#4307
         id: 3,
         name: "Sony",
         deleted_at: null,
         created_at: "2021-03-04 21:39:43",
         updated_at: "2021-03-04 21:39:43",
       },
       App\Models\Brand {#4215
         id: 4,
         name: "Lenovo",
         deleted_at: "2021-03-04 21:54:08",
         created_at: "2021-03-04 21:39:53",
         updated_at: "2021-03-04 21:54:08",
       },
       App\Models\Brand {#4319
         id: 5,
         name: "Sansung",
         deleted_at: null,
         created_at: "2021-03-04 21:40:04",
         updated_at: "2021-03-04 21:40:04",
       },
```

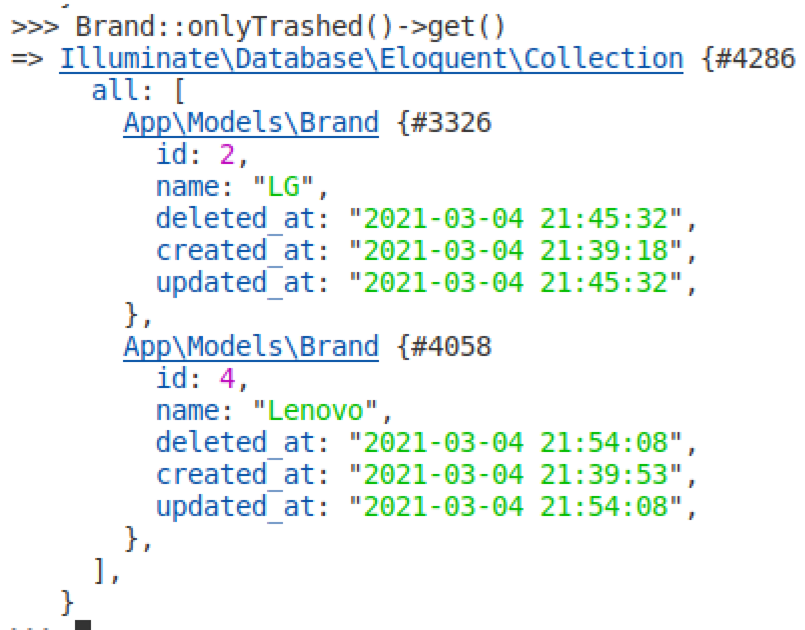


```
         updated_at: "2021-03-04 21:40:04",
       },
       App\Models\Brand {#4320
         id: 6,
         name: "Acer",
         deleted_at: null,
         created_at: "2021-03-04 21:40:17",
         updated_at: "2021-03-04 21:40:17",
       },
       App\Models\Brand {#4321
         id: 7,
         name: "HP",
         deleted_at: null,
         created_at: "2021-03-04 21:40:24",
         updated_at: "2021-03-04 21:40:24",
       },
     ],
   }
```

Mostrando somente os registros apagados com soft delete

- Para mostrar apenas os registros apagados com soft delete, temos a função onlyTrashed

Brand::onlyTrashed()->get()

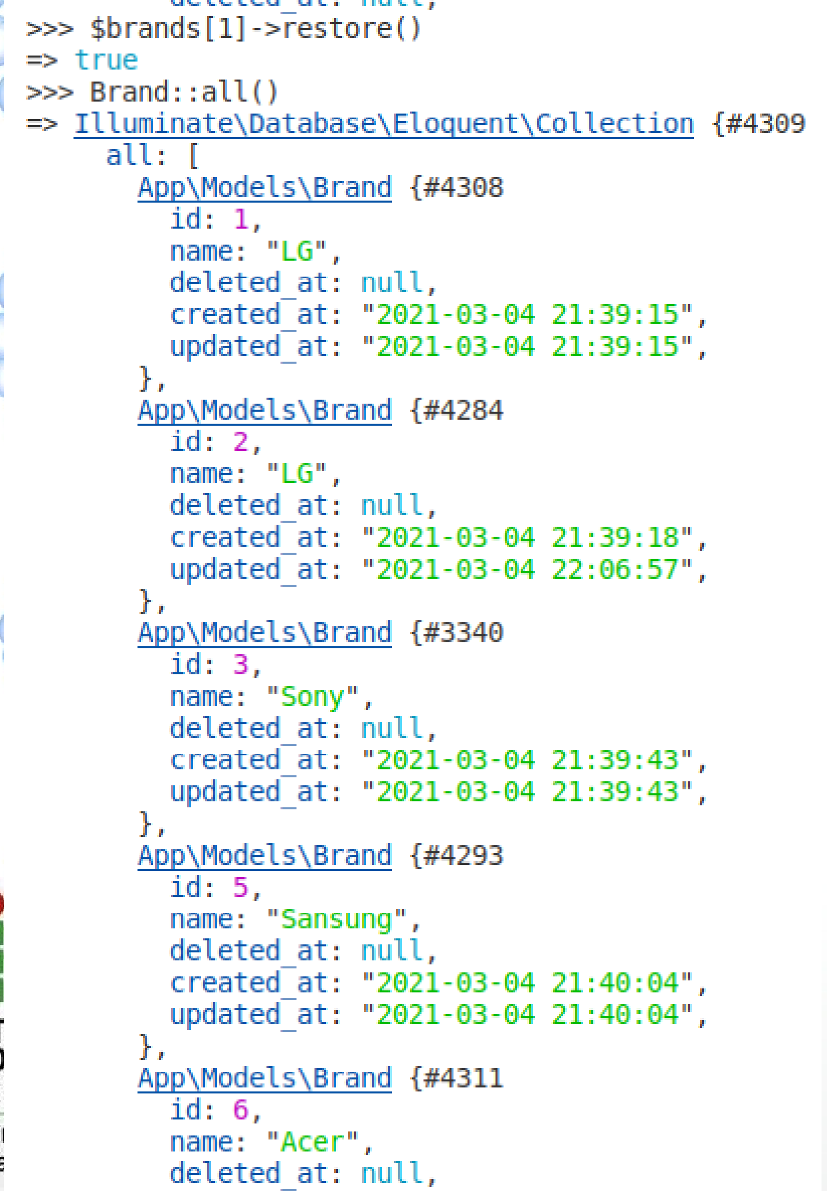


```
>>> Brand::onlyTrashed()->get()
=> Illuminate\Database\Eloquent\Collection {#4286
     all: [
       App\Models\Brand {#3326
         id: 2,
         name: "LG",
         deleted_at: "2021-03-04 21:45:32",
         created_at: "2021-03-04 21:39:18",
         updated_at: "2021-03-04 21:45:32",
       },
       App\Models\Brand {#4058
         id: 4,
         name: "Lenovo",
         deleted_at: "2021-03-04 21:54:08",
         created_at: "2021-03-04 21:39:53",
         updated_at: "2021-03-04 21:54:08",
       },
     ],
   }
```

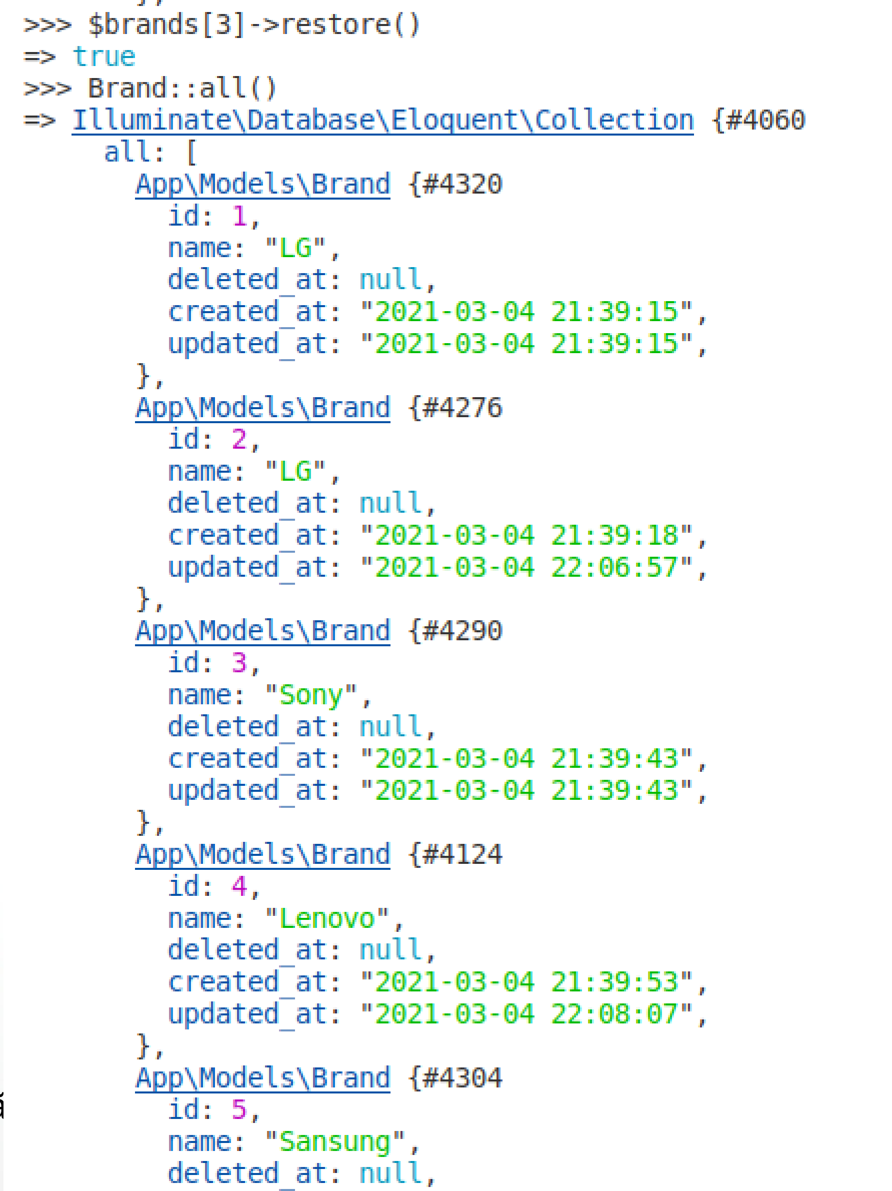
Restaurando registros do soft delete

- Para restaurar os registros soft delete, utilizamos uma função chamada restore()
  - $brands = Brand::onlyTrashed()-> get()

    $brands[0]-> restore()

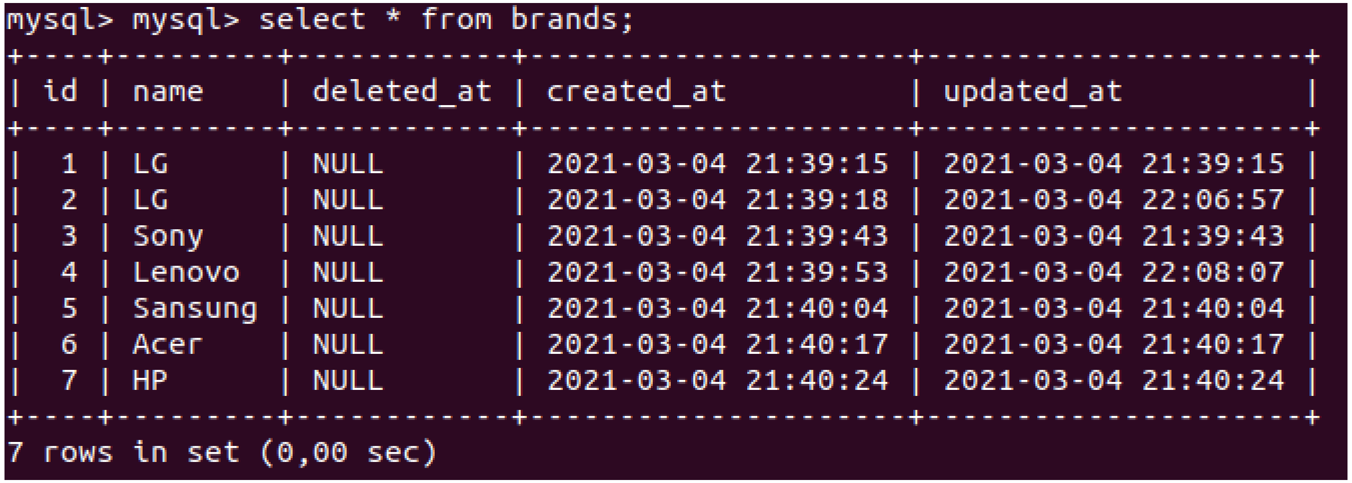


```
>>> $brands[1]->restore()
=> true
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4309
     all: [
       App\Models\Brand {#4308
         id: 1,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:15",
         updated_at: "2021-03-04 21:39:15",
       },
       App\Models\Brand {#4284
         id: 2,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:18",
         updated_at: "2021-03-04 22:06:57",
       },
       App\Models\Brand {#3340
         id: 3,
         name: "Sony",
         deleted_at: null,
         created_at: "2021-03-04 21:39:43",
         updated_at: "2021-03-04 21:39:43",
       },
       App\Models\Brand {#4293
         id: 5,
         name: "Sansung",
         deleted_at: null,
         created_at: "2021-03-04 21:40:04",
         updated_at: "2021-03-04 21:40:04",
       },
       App\Models\Brand {#4311
         id: 6,
         name: "Acer",
         deleted_at: null,
```



```
>>> $brands[3]->restore()
=> true
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4060
     all: [
       App\Models\Brand {#4320
         id: 1,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:15",
         updated_at: "2021-03-04 21:39:15",
       },
       App\Models\Brand {#4276
         id: 2,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:18",
         updated_at: "2021-03-04 22:06:57",
       },
       App\Models\Brand {#4290
         id: 3,
         name: "Sony",
         deleted_at: null,
         created_at: "2021-03-04 21:39:43",
         updated_at: "2021-03-04 21:39:43",
       },
       App\Models\Brand {#4124
         id: 4,
         name: "Lenovo",
         deleted_at: null,
         created_at: "2021-03-04 21:39:53",
         updated_at: "2021-03-04 22:08:07",
       },
       App\Models\Brand {#4304
         id: 5,
         name: "Sansung",
         deleted_at: null,
```

Após a operação de restore

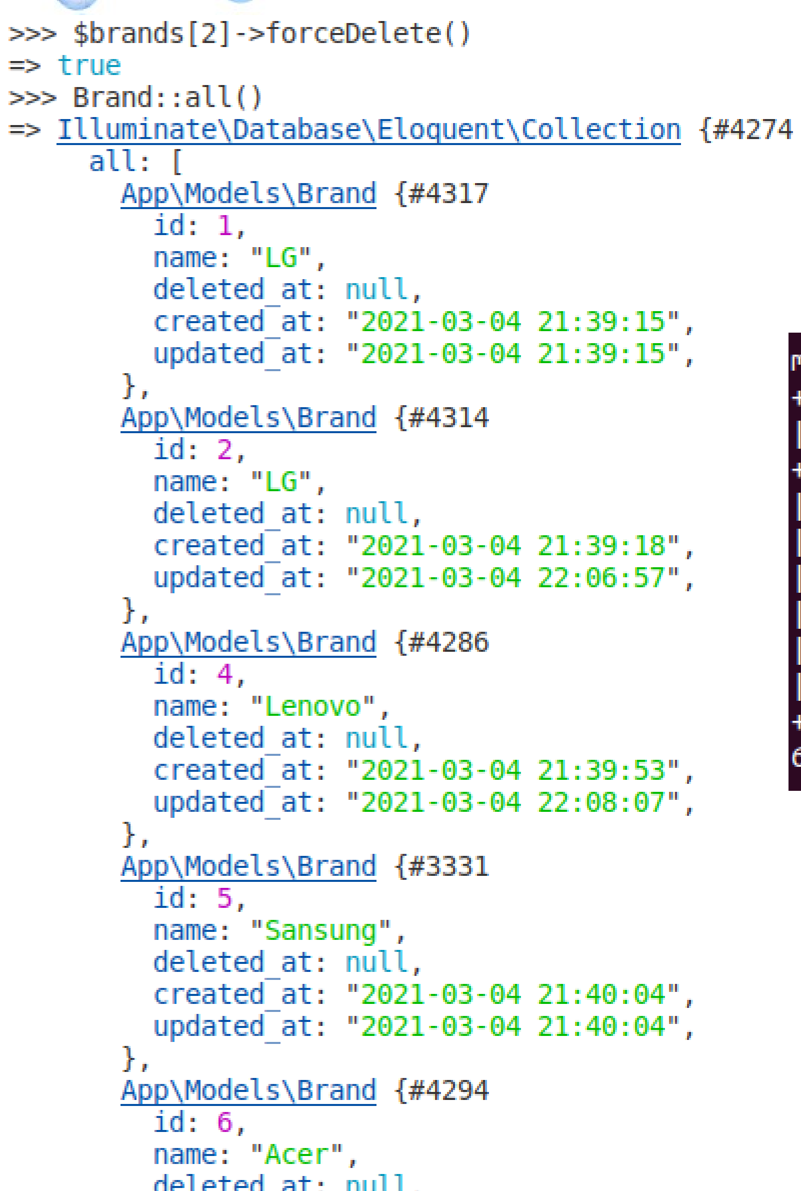


```
mysql> mysql> select * from brands;
+----+---------+------------+---------------------+---------------------+
| id | name    | deleted_at | created_at          | updated_at          |
+----+---------+------------+---------------------+---------------------+
|  1 | LG      | NULL       | 2021-03-04 21:39:15 | 2021-03-04 21:39:15 |
|  2 | LG      | NULL       | 2021-03-04 21:39:18 | 2021-03-04 22:06:57 |
|  3 | Sony    | NULL       | 2021-03-04 21:39:43 | 2021-03-04 21:39:43 |
|  4 | Lenovo  | NULL       | 2021-03-04 21:39:53 | 2021-03-04 22:08:07 |
|  5 | Sansung | NULL       | 2021-03-04 21:40:04 | 2021-03-04 21:40:04 |
|  6 | Acer    | NULL       | 2021-03-04 21:40:17 | 2021-03-04 21:40:17 |
|  7 | HP      | NULL       | 2021-03-04 21:40:24 | 2021-03-04 21:40:24 |
+----+---------+------------+---------------------+---------------------+
7 rows in set (0,00 sec)
```

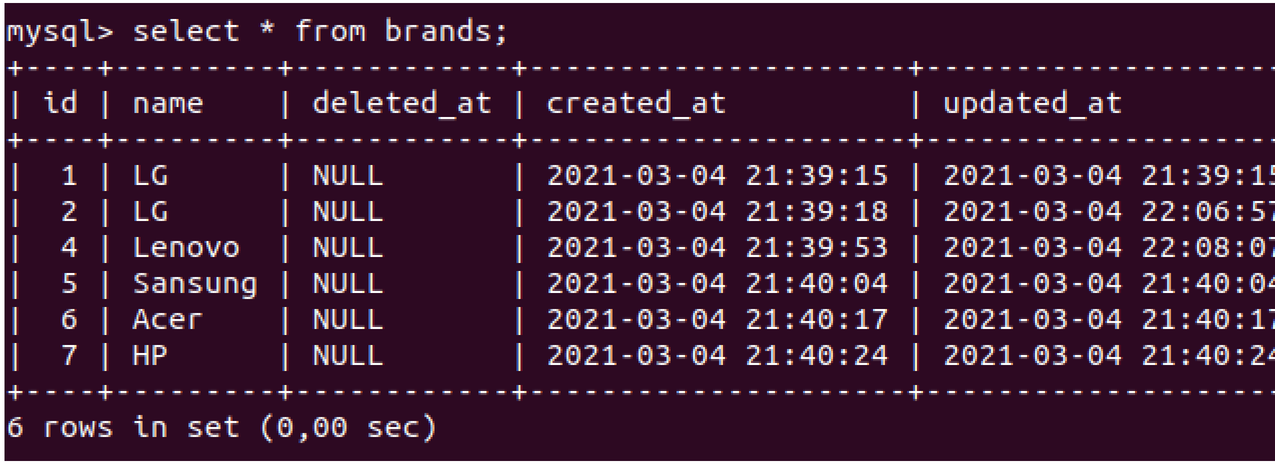
Apagar da base

- Para efetivar a exclusão do registro na base, utilizamos a função forceDelete()
  - $brands[0]-> forceDelete()
  - Ou
  - Brand::find(1)-> forceDelete()

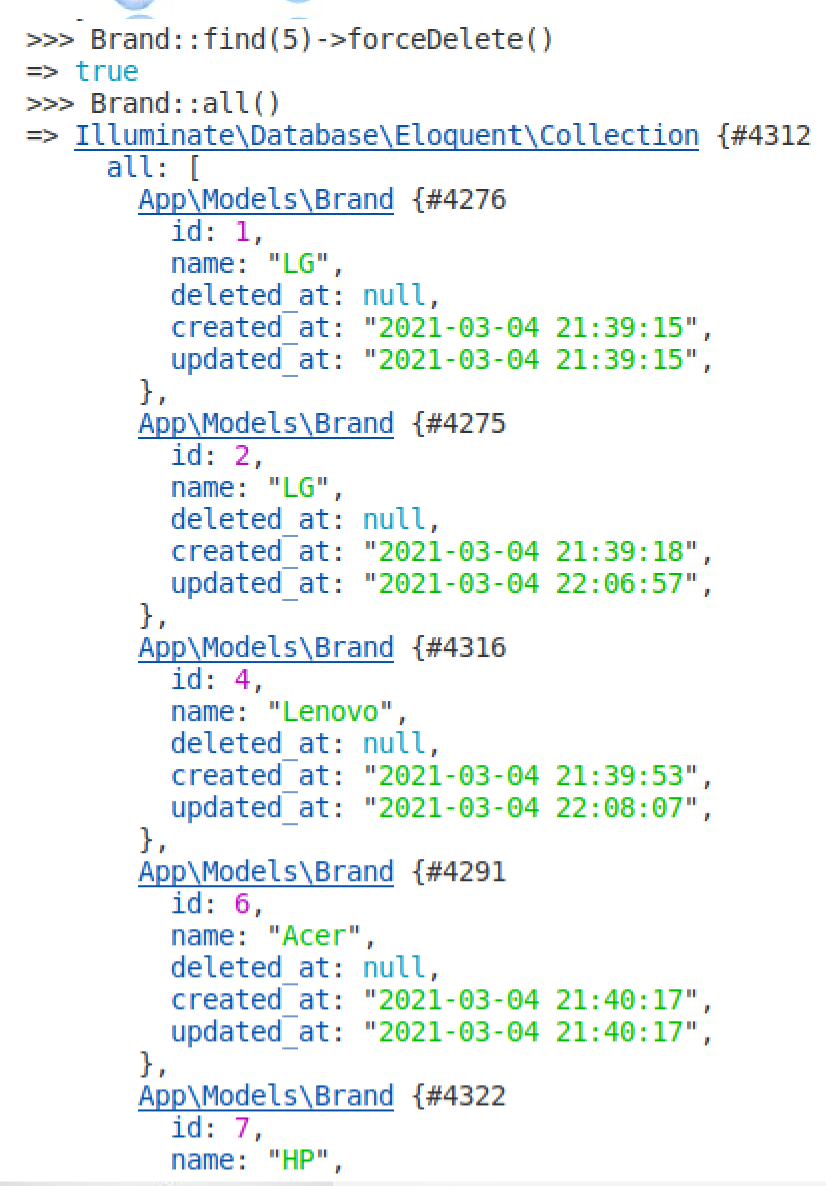
INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar



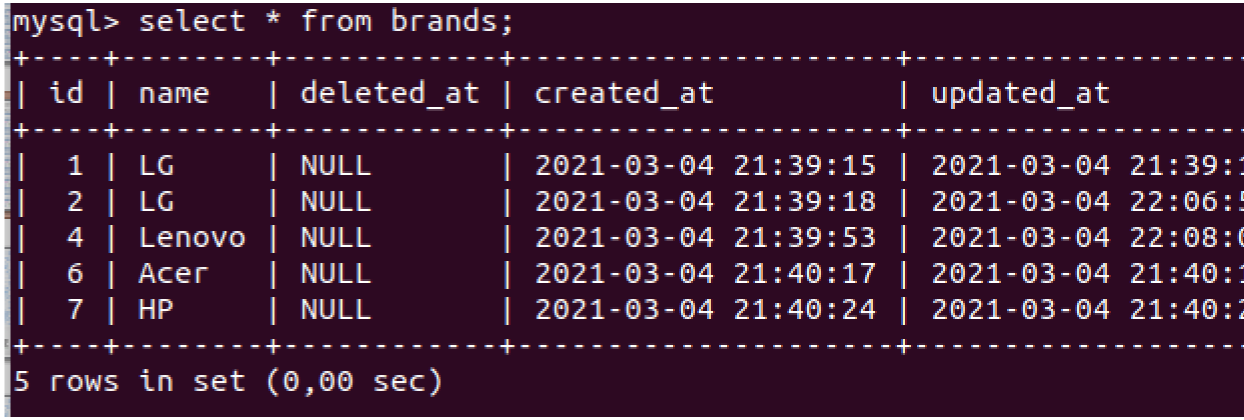
```
>>> $brands[2]->forceDelete()
=> true
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4274
     all: [
       App\Models\Brand {#4317
         id: 1,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:15",
         updated_at: "2021-03-04 21:39:15",
       },
       App\Models\Brand {#4314
         id: 2,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:18",
         updated_at: "2021-03-04 22:06:57",
       },
       App\Models\Brand {#4286
         id: 4,
         name: "Lenovo",
         deleted_at: null,
         created_at: "2021-03-04 21:39:53",
         updated_at: "2021-03-04 22:08:07",
       },
       App\Models\Brand {#3331
         id: 5,
         name: "Sansung",
         deleted_at: null,
         created_at: "2021-03-04 21:40:04",
         updated_at: "2021-03-04 21:40:04",
       },
       App\Models\Brand {#4294
         id: 6,
         name: "Acer",
         deleted_at: null,
```



```
mysql> select * from brands;
+----+---------+------------+---------------------+-------------------
| id | name    | deleted_at | created_at          | updated_at
+----+---------+------------+---------------------+-------------------
|  1 | LG      | NULL       | 2021-03-04 21:39:15 | 2021-03-04 21:39:1
|  2 | LG      | NULL       | 2021-03-04 21:39:18 | 2021-03-04 22:06:5
|  4 | Lenovo  | NULL       | 2021-03-04 21:39:53 | 2021-03-04 22:08:0
|  5 | Sansung | NULL       | 2021-03-04 21:40:04 | 2021-03-04 21:40:0
|  6 | Acer    | NULL       | 2021-03-04 21:40:17 | 2021-03-04 21:40:1
|  7 | HP      | NULL       | 2021-03-04 21:40:24 | 2021-03-04 21:40:2
+----+---------+------------+---------------------+-------------------
6 rows in set (0,00 sec)
```



```
>>> Brand::find(5)->forceDelete()
=> true
>>> Brand::all()
=> Illuminate\Database\Eloquent\Collection {#4312
     all: [
       App\Models\Brand {#4276
         id: 1,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:15",
         updated_at: "2021-03-04 21:39:15",
       },
       App\Models\Brand {#4275
         id: 2,
         name: "LG",
         deleted_at: null,
         created_at: "2021-03-04 21:39:18",
         updated_at: "2021-03-04 22:06:57",
       },
       App\Models\Brand {#4316
         id: 4,
         name: "Lenovo",
         deleted_at: null,
         created_at: "2021-03-04 21:39:53",
         updated_at: "2021-03-04 22:08:07",
       },
       App\Models\Brand {#4291
         id: 6,
         name: "Acer",
         deleted_at: null,
         created_at: "2021-03-04 21:40:17",
         updated_at: "2021-03-04 21:40:17",
       },
       App\Models\Brand {#4322
         id: 7,
         name: "HP",
```



```
mysql> select * from brands;
+----+--------+------------+---------------------+------------------
| id | name   | deleted_at | created_at          | updated_at
+----+--------+------------+---------------------+------------------
|  1 | LG     | NULL       | 2021-03-04 21:39:15 | 2021-03-04 21:39:1
|  2 | LG     | NULL       | 2021-03-04 21:39:18 | 2021-03-04 22:06:5
|  4 | Lenovo | NULL       | 2021-03-04 21:39:53 | 2021-03-04 22:08:0
|  6 | Acer   | NULL       | 2021-03-04 21:40:17 | 2021-03-04 21:40:1
|  7 | HP     | NULL       | 2021-03-04 21:40:24 | 2021-03-04 21:40:2
+----+--------+------------+---------------------+------------------
5 rows in set (0,00 sec)
```

Finalizando...

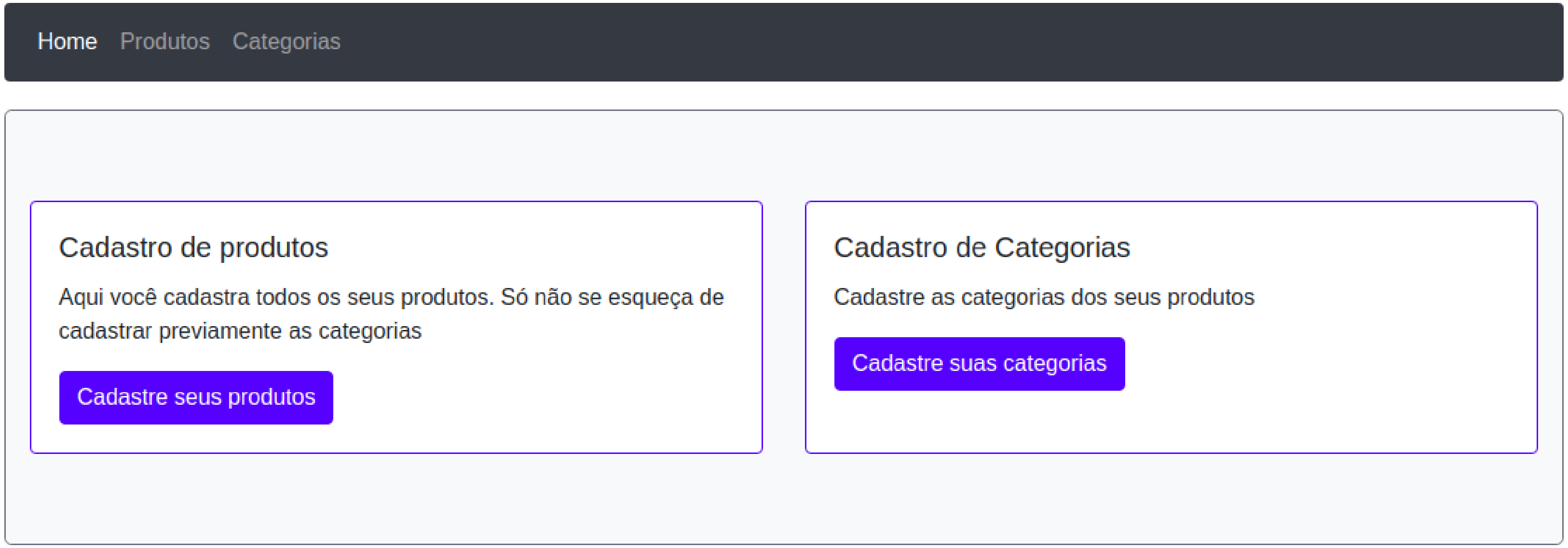
- Lembrando que o model é a interface com o BD
  - Aqui realizamos várias operações com BD utilizando o Tinker
- Portanto, tudo que realizamos aqui no tinker vamos fazer nos códigos do projeto dentro de models
- Agora vamos à prática...vamos realizar um estudo de caso

Roteiro
Projeto de Cadastro

- Rotas
- Views/Formulários
- Construtores
- Modelos
- Layout/Componentes/Estilização
- Persistência no banco

INSTITUTO FEDERAL
Santa Catarina
Câmpus
Gaspar

Roteiro Estudo de Caso



Cadastro de produtos

Aqui você cadastra todos os seus produtos. Só não se esqueça de cadastrar previamente as categorias

Cadastro de Categorias

Cadastre as categorias dos seus produtos

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Referêncicas

- Laravel documets. https://laravel.com/docs/8.x
- Laravel. Collection class. https://laravel.com/docs/8.x/collections
-