

1. Liste 5 documentos de cada coleção para validar a importação.

```
dsm_geo_Erlon> db.inscricoes.find().limit(5)
[
  {
    _id: ObjectId('68d57db242b4d04c338443db'),
    alunoId: '232ed842',
    nome: 'Aluno 04',
    curso: 'DSM3',
    turno: 'NOITE',
    bairro: 'Bloco B',
    meioTransporte: 'ONIBUS',
    localizacao: { type: 'Point', coordinates: [ -47.837251, -24.471078 ] }
  },
  {
    _id: ObjectId('68d57db242b4d04c338443dc'),
    alunoId: '55a87ee5',
    nome: 'Aluno 12',
    curso: 'DSM3',
    bairro: 'Agrochá',
    meioTransporte: 'MOTO',
    localizacao: { type: 'Point', coordinates: [ -47.879459, -24.524963 ] }
  },
  {
    _id: ObjectId('68d57db242b4d04c338443dd'),
    alunoId: 'f2885c6f',
    nome: 'Aluno 11',
    curso: 'DSM2',
    turno: 'MANHA',
    bairro: 'Paulistano',
    meioTransporte: 'BICICLETA',
    localizacao: { type: 'Point', coordinates: [ -47.858985, -24.502591 ] }
  },
  {
    _id: ObjectId('68d57db242b4d04c338443de'),
    alunoId: 'f12532e3',
    nome: 'Aluno 13',
    curso: 'DSM1',
    turno: 'MANHA',
    bairro: 'Jardim São Paulo',
    meioTransporte: 'A_PE',
    localizacao: { type: 'Point', coordinates: [ -47.828403, -24.492903 ] }
  },
  {
    _id: ObjectId('68d57db242b4d04c338443df'),
    alunoId: 'bf85ba6b',
    nome: 'Aluno 14',
    curso: 'DSM3',
    turno: 'NOITE',
    bairro: 'Beira Rio',
    meioTransporte: 'BICICLETA',
    localizacao: { type: 'Point', coordinates: [ -47.828403, -24.492903 ] }
  }
]
```

2. Recupere o documento do **CAMPUS** em pontos e guarde em uma variável `campus`.
3. Usando `$geoNear`, liste os **4 alunos mais próximos** do campus (retorne nome, bairro e a `distancia_m`).

```
dsm_geo_Erlon> const campus = db.pontos.findOne({tipo:"CAMPUS"})

dsm_geo_Erlon> db.inscricoes.aggregate([
...   { $geoNear: {
...     near: campus.localizacao,
...     distanceField: "distancia_m",
...     spherical: true
...   }},
...   { $limit: 4 }
... ])
[
  {
    _id: ObjectId('68d57db242b4d04c338443f2'),
    alunoId: 'a809edeb',
    nome: 'Aluno 27',
    curso: 'DSM1',
    turno: 'TARDE',
    bairro: 'Vila Nova',
    meioTransporte: 'ONIBUS',
    localizacao: { type: 'Point', coordinates: [ -47.832727, -24.498367 ] },
    distancia_m: 1212.296269576149
  },
  {
    _id: ObjectId('68d57db242b4d04c338443dd'),
    alunoId: 'f2885c6f',
    nome: 'Aluno 11',
    curso: 'DSM2',
    turno: 'MANHA',
    bairro: 'Paulistano',
    meioTransporte: 'BICICLETA',
    localizacao: { type: 'Point', coordinates: [ -47.858985, -24.502591 ] },
    distancia_m: 1584.495859076122
  },
  {
    _id: ObjectId('68d57db242b4d04c338443de'),
    alunoId: 'f12532e3',
    nome: 'Aluno 13',
    curso: 'DSM1',
    turno: 'MANHA',
    bairro: 'Jardim São Paulo',
    meioTransporte: 'A_PE',
    localizacao: { type: 'Point', coordinates: [ -47.828403, -24.492903 ] },
    distancia_m: 1702.9616498162625
  },
  {
    _id: ObjectId('68d57db242b4d04c338443e7'),
    alunoId: '2c7cc9d4',
    nome: 'Aluno 21',
    curso: 'DSM3',
    turno: 'TARDE',
  }
]
```

4. Encontre **todos os alunos a até 2 km** do campus com **\$geoWithin/\$centerSphere**.

```
dsm_geo_Erlon> const campus2 = db.pontos.findOne({tipo:"CAMPUS"})
  bairro: 'Agrochã',
dsm_geo_Erlon> db.inscricoes.find({
...   localizacao: {
...     $geoWithin: {
...       $centerSphere: [ campus2.localizacao.coordinates, 2000/6378137 ]
...     }
...   }
... })
[
  {
    _id: ObjectId('68d57db242b4d04c338443dd'),
    alunoId: 'f2885c6f',
    nome: 'Aluno 11',
    curso: 'DSM2',
    turno: 'MANHA',
    bairro: 'Paulistano',
    meioTransporte: 'BICICLETA',
    localizacao: { type: 'Point', coordinates: [ -47.858985, -24.502591 ] }
  },
  {
    _id: ObjectId('68d57db242b4d04c338443de'),
    alunoId: 'f12532e3',
    nome: 'Aluno 13',
    curso: 'DSM1',
    turno: 'MANHA',
    bairro: 'Jardim São Paulo',
    meioTransporte: 'A_PE',
    localizacao: { type: 'Point', coordinates: [ -47.828403, -24.492903 ] }
  },
  {
    _id: ObjectId('68d57db242b4d04c338443f2'),
    alunoId: 'a809edeb',
    nome: 'Aluno 27',
    curso: 'DSM1',
    turno: 'TARDE',
    bairro: 'Vila Nova',
    meioTransporte: 'ONIBUS',
    localizacao: { type: 'Point', coordinates: [ -47.832727, -24.498367 ] }
  },
  {
    _id: ObjectId('68d57db242b4d04c338443e7'),
    alunoId: '2c7cc9d4',
    nome: 'Aluno 21',
    curso: 'DSM3',
    turno: 'TARDE',
    bairro: 'Vila Cabral',
    meioTransporte: 'BICICLETA',
    localizacao: { type: 'Point', coordinates: [ -47.833031, -24.510828 ] }
  }
]
```

5. Entre os alunos até **3 km** do campus, gere um **ranking por meioTransporte** (contagem decrescente).

```
dsm_geo_Erlon> db.inscricoes.aggregate([
...   { $match: {
...     localizacao: {
...       $geoWithin: { $centerSphere: [ campus3.localizacao.coordinates, 3000/6378137 ] }
...     }
...   }},
...   { $group: { _id: "$meioTransporte", qtd: { $sum: 1 } } },
...   { $sort: { qtd: -1 } }
... ])
[
  { _id: 'BICICLETA', qtd: 3 },
  { _id: 'ONIBUS', qtd: 2 },
  { _id: 'A_PE', qtd: 1 }
]
```

6. Pegue o documento **“Ponto de Encontro Norte”** e liste os **5 alunos mais próximos** com **\$geoNear**.

```

dsm_geo_Erlon> db.inscricoes.aggregate([
...   { $geoNear: {
...     near: pontoNorte.localizacao,
...     distanceField: "distancia_m",
...     spherical: true
...   }},
...   { $limit: 5 }
... ])
[
  {
    _id: ObjectId('68d57db242b4d04c338443f2'),
    alunoId: 'a809edeb',
    nome: 'Aluno 27',
    curso: 'DSM1',
    turno: 'TARDE',
    bairro: 'Vila Nova',
    meioTransporte: 'ONIBUS',
    localizacao: { type: 'Point', coordinates: [ -47.832727, -24.498367 ] },
    distancia_m: 705.3837480371411
  },
  {
    _id: ObjectId('68d57db242b4d04c338443e2'),
    alunoId: '60b2c235',
    nome: 'Aluno 17',
    curso: 'DSM3',
    turno: 'TARDE',
    bairro: 'Arpoador',
    meioTransporte: 'BICICLETA',
    localizacao: { type: 'Point', coordinates: [ -47.825947, -24.509524 ] },
    distancia_m: 715.9064874148371
  },
  {
    _id: ObjectId('68d57db242b4d04c338443e7'),
    alunoId: '2c7cc9d4',
    nome: 'Aluno 21',
    curso: 'DSM3',
    turno: 'TARDE',
    bairro: 'Vila Cabral',
    meioTransporte: 'BICICLETA',
    localizacao: { type: 'Point', coordinates: [ -47.833031, -24.510828 ] },
    distancia_m: 824.6355176294518
  },
  {
    _id: ObjectId('68d57db242b4d04c338443de'),
    alunoId: 'f12532e3',
    nome: 'Aluno 13',
    curso: 'DSM1',
    turno: 'MANHA',
    bairro: 'Jardim São Paulo',
    meioTransporte: 'A_PE',
  }
]

```

7) Gere uma “**pré-rota de van**”: pegue os **8 alunos** mais próximos do campus, ordenados por distância, mostrando nome, bairro e distancia_m.

```
dsms_geo_Erlon> db.inscricoes.aggregate([
...   { $geoNear: {
...     near: campus4.localizacao,
...     distanceField: "distancia_m",
...     spherical: true
...   }},
...   { $limit: 8 },
...   { $project: { nome: 1, bairro: 1, distancia_m: 1, _id: 0 } }
... ])
[
  {
    nome: 'Aluno 27',
    bairro: 'Vila Nova',
    distancia_m: 1212.296269576149
  },
  {
    nome: 'Aluno 11',
    bairro: 'Paulistano',
    distancia_m: 1584.495859076122
  },
  {
    nome: 'Aluno 13',
    bairro: 'Jardim São Paulo',
    distancia_m: 1702.9616498162625
  },
  {
    nome: 'Aluno 21',
    bairro: 'Vila Cabral',
    distancia_m: 1934.616515957337
  },
  {
    nome: 'Aluno 17',
    bairro: 'Arpoador',
    distancia_m: 2348.1036367492256
  },
  {
    nome: 'Aluno 04',
    bairro: 'Bloco B',
    distancia_m: 2980.10650207966
  },
  {
    nome: 'Aluno 05',
    bairro: 'Bloco B',
    distancia_m: 3026.603881783604
  },
  {
    nome: 'Aluno 25',
    bairro: 'Vila Nova',
    distancia_m: 3185.975936552661
  }
]
```

8. (Desafio) Encontre **pares de colegas** que moram a ≤ 800 m um do outro (sugestões: Haversine via `$function` ou self-join com `$lookup` + geo pipeline).

```
2 | $lookup: {
3 |   from: "inscricoes",
4 |   let: { minhaLoc: "$localizacao", meuNome: "$nome" },
5 |   pipeline: [
6 |     {
7 |       $match: {
8 |
9 |       },
10 |     },
11 |     {
12 |       $lookup: {
13 |         from: "inscricoes",
14 |         let: { minhaLoc: "$localizacao", meuNome: "$nome" },
15 |         pipeline: [
16 |           {
17 |             $match: {
18 |               $expr: { $ne: ["$nome", "$$meuNome"] }
19 |             }
20 |           },
21 |           {
22 |             $addFields: {
23 |               distancia_m: {
24 |                 $function: {
25 |                   body: function(loc1, loc2) {
26 |                     const toRad = deg => (deg * Math.PI) / 180;
27 |                     const R = 6371000;
28 |                     const [lon1, lat1] = loc1.coordinates;
29 |                     const [lon2, lat2] = loc2.coordinates;
30 |
31 |                     const dLat = toRad(lat2 - lat1);
32 |                     const dLon = toRad(lon2 - lon1);
33 |                     const a =
34 |                       Math.sin(dLat / 2) * Math.sin(dLat / 2) +
35 |                       Math.cos(toRad(lat1)) *
36 |                       Math.cos(toRad(lat2)) *
37 |                       Math.sin(dLon / 2) * Math.sin(dLon / 2);
38 |                     const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
39 |                     return R * c;
40 |                   },
41 |                   args: ["$$minhaLoc", "$localizacao"],
42 |                   lang: "js"
43 |                 }
44 |               }
45 |             },
46 |             {
47 |               $match: { distancia_m: { $lte: 800 } }
48 |             }
49 |           ],
50 |           as: "colegasProximos"
51 |         }
52 |       }
53 |     }
54 |   ],
55 | }
```



```

...   }
... },
... { $match: { "colegasProximos.0": { $exists: true } } },
... {
...   $project: {
...     nome: 1,
...     colegasProximos: { nome: 1, distancia_m: 1 }
...   }
... }
... ])
[
  {
    _id: ObjectId('68d57db242b4d04c338443de'),
    nome: 'Aluno 13',
    colegasProximos: [ { nome: 'Aluno 27', distancia_m: 748.71449685646 } ]
  },
  {
    _id: ObjectId('68d57db242b4d04c338443e0'),
    nome: 'Aluno 16',
    colegasProximos: [ { nome: 'Aluno 06', distancia_m: 135.91998214489627 } ]
  },
  {
    _id: ObjectId('68d57db242b4d04c338443e2'),
    nome: 'Aluno 17',
    colegasProximos: [ { nome: 'Aluno 21', distancia_m: 731.2428958627884 } ]
  },
  {
    _id: ObjectId('68d57db242b4d04c338443e7'),
    nome: 'Aluno 21',
    colegasProximos: [ { nome: 'Aluno 17', distancia_m: 731.2428958627884 } ]
  },
  {
    _id: ObjectId('68d57db242b4d04c338443ee'),
    nome: 'Aluno 06',
    colegasProximos: [ { nome: 'Aluno 16', distancia_m: 135.91998214489627 } ]
  },
  {
    _id: ObjectId('68d57db242b4d04c338443f0'),
    nome: 'Aluno 07',
    colegasProximos: [ { nome: 'Aluno 28', distancia_m: 622.7522254460249 } ]
  },
  {
    _id: ObjectId('68d57db242b4d04c338443f2'),
    nome: 'Aluno 27',
    colegasProximos: [ { nome: 'Aluno 13', distancia_m: 748.71449685646 } ]
  },
  {

```

```

  {
    _id: ObjectId('68d57db242b4d04c338443f5'),
    nome: 'Aluno 28',
    colegasProximos: [ { nome: 'Aluno 07', distancia_m: 622.7522254460249 } ]
  }
]

```

9. (Desafio) Crie um **filtro combinado**: apenas alunos do **turno NOITE** e **curso DSM1** dentro de **1,2 km** do campus.

```
dsm_geo_Erlon> db.inscricoes.aggregate([
...   {
...     $geoNear: {
...       near: campus.localizacao,
...       distanceField: "distancia_m",
...       maxDistance: 1200,
...       spherical: true
...     }
...   },
...   {
...     $match: {
...       turno: "NOITE",
...       curso: "DSM1"
...     }
...   },
...   {
...     $project: {
...       nome: 1,
...       curso: 1,
...       turno: 1,
...       distancia_m: 1
...     }
...   }
... ])

dsm_geo_Erlon>
```

Obs: Não retornou ninguém pois os alunos cadastrados (4) moram em mais de 1,2km da unidade:

```
{
  _id: ObjectId('68d57db242b4d04c338443ea'),
  nome: 'Aluno 25',
  distancia_m: 3185.975936552661
},
{
  _id: ObjectId('68d57db242b4d04c338443f1'),
  nome: 'Aluno 26',
  distancia_m: 4822.8304246874895
},
{
  _id: ObjectId('68d57db242b4d04c338443e6'),
  nome: 'Aluno 22',
  distancia_m: 7485.228977601787
},
{
  _id: ObjectId('68d57db242b4d04c338443f6'),
  nome: 'Aluno 03',
  distancia_m: 7863.818975186834
}
]
```