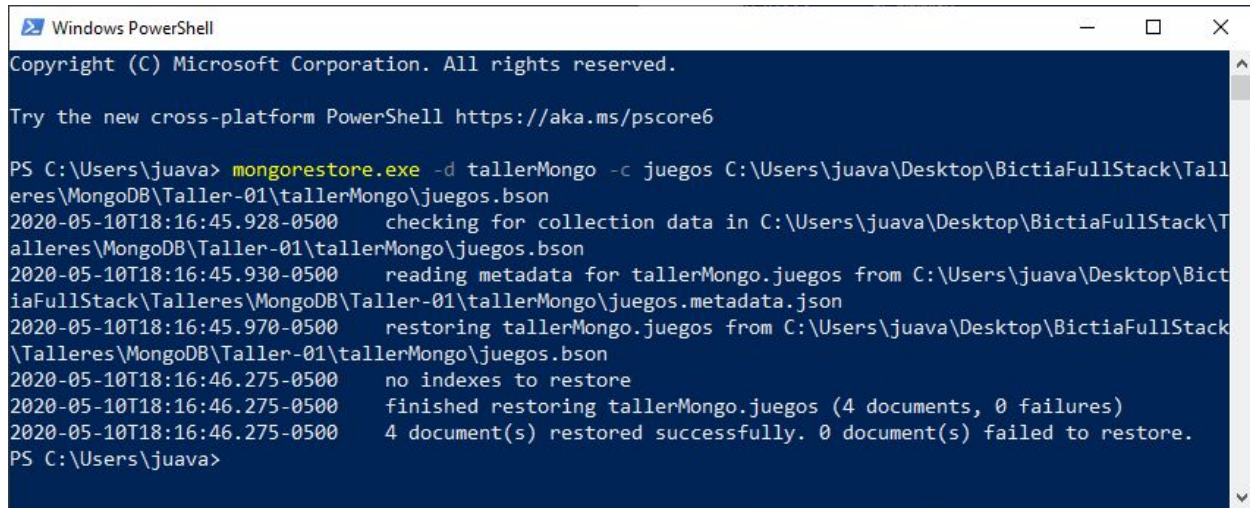


Taller #1 MongoDB

Realizado por: Juan Castañeda

05-2020

1. Importar Base de Datos



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\juava> mongorestore.exe -d tallerMongo -c juegos C:\Users\juava\Desktop\BictiaFullStack\Talleres\MongoDB\Taller-01\tallerMongo\juegos.bson
2020-05-10T18:16:45.928-0500   checking for collection data in C:\Users\juava\Desktop\BictiaFullStack\Talleres\MongoDB\Taller-01\tallerMongo\juegos.bson
2020-05-10T18:16:45.930-0500   reading metadata for tallerMongo.juegos from C:\Users\juava\Desktop\BictiaFullStack\Talleres\MongoDB\Taller-01\tallerMongo\juegos.metadata.json
2020-05-10T18:16:45.970-0500   restoring tallerMongo.juegos from C:\Users\juava\Desktop\BictiaFullStack\Talleres\MongoDB\Taller-01\tallerMongo\juegos.bson
2020-05-10T18:16:46.275-0500   no indexes to restore
2020-05-10T18:16:46.275-0500   finished restoring tallerMongo.juegos (4 documents, 0 failures)
2020-05-10T18:16:46.275-0500   4 document(s) restored successfully. 0 document(s) failed to restore.
PS C:\Users\juava>
```

1.a Agregar a cada uno de los juegos un campo llamado plataforma cuyo valor será un arreglo.

Konami

```
> db.juegos.updateMany( {"nombre": "Konami"}, {$set: {"juegos.0.plataforma": ["PSX", "PS3", "Sega Saturn", "Xbox 360", "Android"], "juegos.1.plataforma": ["SNES", "Nintendo Wii", "Nintendo Wii U", "Nintendo 3DS"], "juegos.2.plataforma": ["PS2", "Nintendo 3DS", "PS Vita", "PS3"] } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

```

> db.juegos.find( {"nombre": "Konami"}, {_id: 0} ).pretty()
{
  "nombre" : "Konami",
  "juegos" : [
    {
      "nombre" : "Castlevania: Symphony Of The Night",
      "anio" : 1997,
      "plataforma" : [
        "PSX",
        "PS3",
        "Sega Saturn",
        "Xbox 360",
        "Android"
      ]
    },
    {
      "nombre" : "Contra III",
      "anio" : 1992,
      "plataforma" : [
        "SNES",
        "Nintendo Wii",
        "Nintendo Wii U",
        "Nintendo 3DS"
      ]
    },
    {
      "nombre" : "Metal Gear Solid III: Sneake Eater",
      "anio" : 2004,
      "plataforma" : [
        "PS2",
        "Nintendo 3DS",
        "PS Vita",
        "PS3"
      ]
    }
  ]
}

```

Capcom

```

> db.juegos.updateMany( {"nombre": "Capcom"}, {$set: {"juegos.0.plataforma": ["PSX", "PS3", "Windows", "Sega Dreamcast", "Nintendo GameCub", "PSP"], "juegos.1.plataforma": ["Arcade", "Nintendo Wii", "Nintendo Wii U", "Nintendo 3DS", "iOS", "Android", "Switch", "NES"], "juegos.2.plataforma": ["Arcade"] } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

```

```

> db.juegos.find( {"nombre": "Capcom"}, {_id: 0} ).pretty()
{
  "nombre" : "Capcom",
  "juegos" : [
    {
      "nombre" : "Resident Evil 3: Nemesis",
      "anio" : 1999,
      "plataforma" : [
        "PSX",
        "PS3",
        "Windows",
        "Sega Dreamcast",
        "Nintendo GameCub",
        "PSP"
      ]
    },
    {
      "nombre" : "Ghosts 'n Goblins",
      "anio" : 1985,
      "plataforma" : [
        "Arcade",
        "Nintendo Wii",
        "Nintendo Wii U",
        "Nintendo 3DS",
        "iOS",
        "Android",
        "Switch",
        "NES"
      ]
    },
    {
      "nombre" : "Cadillacs and Dinosaurs",
      "anio" : 1993,
      "plataforma" : [
        "Arcade"
      ]
    }
  ]
}

```

Ubisoft

```

> db.juegos.updateMany( {"nombre": "Ubisoft"}, {$set: {"juegos.0.plataforma": ["Xbox 360", "PS3", "Windows", "Macintosh"], "juegos.1.plataforma": ["PS4", "Xbox One", "Windows"], "juegos.2.plataforma": ["PS4", "Xbox One", "Windows"] } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

```

```

> db.juegos.find( {"nombre": "Ubisoft"}, {_id: 0} ).pretty()
{
  "nombre" : "Ubisoft",
  "juegos" : [
    {
      "nombre" : "Assassin's Creed II",
      "anio" : 2009,
      "plataforma" : [
        "Xbox 360",
        "PS3",
        "Windows",
        "Macintosh"
      ]
    },
    {
      "nombre" : "Watchdogs 2",
      "anio" : 2016,
      "plataforma" : [
        "PS4",
        "Xbox One",
        "Windows"
      ]
    },
    {
      "nombre" : "FarCry 5",
      "anio" : 2018,
      "plataforma" : [
        "PS4",
        "Xbox One",
        "Windows"
      ]
    }
  ]
}

```

Naughty Dog

```

> db.juegos.updateMany( {"nombre": "Naughty Dog"}, {$set: {"juegos.0.plataforma": ["PSX", "PS3", "PSP"], "juegos.1.plataforma": ["PS4"], "juegos.2.plataforma": ["PS4", "PS3"] } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

```

```
> db.juegos.find( {"nombre": "Naughty Dog"}, {_id: 0} ).pretty()
{
  "nombre" : "Naughty Dog",
  "juegos" : [
    {
      "nombre" : "Crash Bandicoot",
      "anio" : 1996,
      "plataforma" : [
        "PSX",
        "PS3",
        "PSP"
      ]
    },
    {
      "nombre" : "Uncharted 4: A Thief's End",
      "anio" : 2016,
      "plataforma" : [
        "PS4"
      ]
    },
    {
      "nombre" : "The Last of Us",
      "anio" : 2013,
      "plataforma" : [
        "PS4",
        "PS3"
      ]
    }
  ]
}
```

1.b Obtener los documentos cuya existencia en plataforma sea de PlayStation 4 o PS4

```
> db.juegos.find( {"juegos.plataforma": "PS4"}, {_id: 0} ).pretty()
{
  "nombre" : "Ubisoft",
  "juegos" : [
    {
      "nombre" : "Assassin's Creed II",
      "anio" : 2009,
      "plataforma" : [
        "Xbox 360",
        "PS3",
        "Windows",
        "Macintosh"
      ]
    },
    {
      "nombre" : "Watchdogs 2",
      "anio" : 2016,
      "plataforma" : [
        "PS4",
        "Xbox One",
        "Windows"
      ]
    },
    {
      "nombre" : "FarCry 5",
      "anio" : 2018,
      "plataforma" : [
        "PS4",
        "Xbox One",
        "Windows"
      ]
    }
  ]
}
```

```
{
  "nombre" : "Naughty Dog",
  "juegos" : [
    {
      "nombre" : "Crash Bandicoot",
      "anio" : 1996,
      "plataforma" : [
        "PSX",
        "PS3",
        "PSP"
      ]
    },
    {
      "nombre" : "Uncharted 4: A Thief's End",
      "anio" : 2016,
      "plataforma" : [
        "PS4"
      ]
    },
    {
      "nombre" : "The Last of Us",
      "anio" : 2013,
      "plataforma" : [
        "PS4",
        "PS3"
      ]
    }
  ]
}
```

1.c Actualizar el subdocumento Metal Gear Solid 3: Snake Eater, modificando únicamente su nombre por Metal Gear Solid 3: Snake Eater

```
> db.juegos.updateOne( {"juegos.2.nombre": "Metal Gear Solid III: Snake Eater"}, {$set: {"juegos.2.nombre": "Metal Gear Solid III: Snake Eater" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

```
> db.juegos.find( {"juegos.nombre": "Metal Gear Solid III: Snake Eater"}, {_id: 0, "juegos.plataforma": 0} ).pretty()
{
  "nombre" : "Konami",
  "juegos" : [
    {
      "nombre" : "Castlevania: Symphony Of The Night",
      "anio" : 1997
    },
    {
      "nombre" : "Contra III",
      "anio" : 1992
    },
    {
      "nombre" : "Metal Gear Solid III: Snake Eater",
      "anio" : 2004
    }
  ]
}
```

1.d Obtener todos los videojuegos cuyo año de lanzamiento este entre 1980 y 2000

```
> db.juegos.find( {"juegos.anio": {$gte: 1980, $lte: 2000 } }, { "_id": 0, "juegos.plataforma": 0 } ).pretty()
{
  "nombre" : "Capcom",
  "juegos" : [
    {
      "nombre" : "Resident Evil 3: Nemesis",
      "anio" : 1999
    },
    {
      "nombre" : "Ghosts 'n Goblins",
      "anio" : 1985
    },
    {
      "nombre" : "Cadillacs and Dinosaurs",
      "anio" : 1993
    }
  ]
}
{
  "nombre" : "Konami",
  "juegos" : [
    {
      "nombre" : "Castlevania: Symphony Of The Night",
      "anio" : 1997
    },
    {
      "nombre" : "Contra III",
      "anio" : 1992
    },
    {
      "nombre" : "Metal Gear Solid III: Snake Eater",
      "anio" : 2004
    }
  ]
}
{
  "nombre" : "Naughty Dog",
  "juegos" : [
    {
      "nombre" : "Crash Bandicoot",
      "anio" : 1996
    },
    {
      "nombre" : "Uncharted 4: A Thief's End",
      "anio" : 2016
    },
    {
      "nombre" : "The Last of Us",
      "anio" : 2013
    }
  ]
}
```


2. En la base de datos tallerMongo, crear la colección películas y agregar los documentos

```
> db.peliculas.insertMany([dos, tres, cuatro, cinco, seis, siete, ocho, nueve, diez])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5eb98b72243b2ff5d4f202e1"),
    ObjectId("5eb98b72243b2ff5d4f202e2"),
    ObjectId("5eb98b72243b2ff5d4f202e3"),
    ObjectId("5eb98b72243b2ff5d4f202e4"),
    ObjectId("5eb98b72243b2ff5d4f202e5"),
    ObjectId("5eb98b72243b2ff5d4f202e6"),
    ObjectId("5eb98b72243b2ff5d4f202e7"),
    ObjectId("5eb98b72243b2ff5d4f202e8"),
    ObjectId("5eb98b72243b2ff5d4f202e9")
  ]
}
```

```
> db.peliculas.find()
{ "_id" : ObjectId("5eb98943243b2ff5d4f202e0"), "titulo" : "Assassin's Creed", "director" : "Justin Kurzel", "anio" : 2016 }
{ "_id" : ObjectId("5eb98b72243b2ff5d4f202e1"), "titulo" : "El cadáver de la novia", "director" : "Tim Burton", "anio" : 2005, "actores" : [ "Johnny Depp", "Helena Bonham-Carter" ] }
{ "_id" : ObjectId("5eb98b72243b2ff5d4f202e2"), "titulo" : "Avatar", "director" : "James Cameron", "anio" : 2005, "actores" : [ "Sam Worthington", "Zoe Saldana", "Sigourney Weaver" ] }
{ "_id" : ObjectId("5eb98b72243b2ff5d4f202e3"), "titulo" : "Tomb Raider", "director" : "Roar Uthaug", "anio" : 2018, "actores" : [ "Alicia Vikander", "Dominic West" ] }
{ "_id" : ObjectId("5eb98b72243b2ff5d4f202e4"), "titulo" : "Deadpool", "director" : "Tim Miller", "anio" : 2016, "actores" : [ "Ryan Reynolds", "Morena Bacarin" ] }
{ "_id" : ObjectId("5eb98b72243b2ff5d4f202e5"), "titulo" : "El joven manos de Tijera", "director" : "Tim Burton", "anio" : 1990, "actores" : [ "Johnny Depp", "Winona Ryder" ] }
{ "_id" : ObjectId("5eb98b72243b2ff5d4f202e6"), "titulo" : "Sin City", "director" : "Quentin Tarantino", "anio" : 2005, "actores" : [ "Bruce Willis", "Jessica Alba" ] }
{ "_id" : ObjectId("5eb98b72243b2ff5d4f202e7"), "titulo" : "Piratas del Caribe: El cofre de la Muerte", "director" : "Gore Verbinski", "anio" : 2006, "actores" : [ "Johnny Depp", "Orlando Bloom", "Keira Knightley" ] }
{ "_id" : ObjectId("5eb98b72243b2ff5d4f202e8"), "titulo" : "El señor de los Anillos: Las Dos Torres", "director" : "Peter Jackson", "anio" : 2005, "actores" : [ "Orlando Bloom", "Elijah Wood", "Ian McKellen" ] }
{ "_id" : ObjectId("5eb98b72243b2ff5d4f202e9"), "titulo" : "X-Men Dias del Futuro Pasado", "director" : "Brian Singer", "anio" : 2014, "actores" : [ "Ian McKellen", "Hugh Jackman" ] }
```

Consultas / Buscar documentos

2.a Obtener documentos con director igual a "Tim Burton"

```
> db.peliculas.find({director: "Tim Burton"}, {_id: 0}).pretty()
{
  "titulo" : "El cadáver de la novia",
  "director" : "Tim Burton",
  "anio" : 2005,
  "actores" : [
    "Johnny Depp",
    "Helena Bonham-Carter"
  ]
}
{
  "titulo" : "El joven manos de Tijera",
  "director" : "Tim Burton",
  "anio" : 1990,
  "actores" : [
    "Johnny Depp",
    "Winona Ryder"
  ]
}
```

2.b Obtener documentos con actores que incluyan a "Ian McKellen"

```
> db.peliculas.find({actores: "Ian McKellen"}, {_id: 0}).pretty()
{
  "titulo" : "El señor de los Anillos: Las Dos Torres",
  "director" : "Peter Jackson",
  "anio" : 2005,
  "actores" : [
    "Orlando Bloom",
    "Elijah Wood",
    "Ian McKellen"
  ]
}
{
  "titulo" : "X-Men Dias del Futuro Pasado",
  "director" : "Brian Singer",
  "anio" : 2014,
  "actores" : [
    "Ian McKellen",
    "Hugh Jackman"
  ]
}
```

2.c Obtener documentos con año mayor a "2013"

```
> db.peliculas.find({anio: {$gte: 2013} }, {_id: 0}).pretty()
{
  "titulo" : "Assassin's Creed",
  "director" : "Justin Kurzel",
  "anio" : 2016
}
{
  "titulo" : "Tomb Raider",
  "director" : "Roar Uthaug",
  "anio" : 2018,
  "actores" : [
    "Alicia Vikander",
    "Dominic West"
  ]
}
{
  "titulo" : "Deadpool",
  "director" : "Tim Miller",
  "anio" : 2016,
  "actores" : [
    "Ryan Reynolds",
    "Morena Bacarin"
  ]
}
{
  "titulo" : "X-Men Dias del Futuro Pasado",
  "director" : "Brian Singer",
  "anio" : 2014,
  "actores" : [
    "Ian McKellen",
    "Hugh Jackman"
  ]
}
```

2.d Obtener el actor que se encuentre en la posición 0 del arreglo actores

```
> db.peliculas.find( {}, { "actores": { $slice: 1 }, _id: 0 } ).pretty()
{
  "titulo" : "Assassin's Creed",
  "director" : "Justin Kurzel",
  "anio" : 2016
}
{
  "titulo" : "El cadáver de la novia",
  "director" : "Tim Burton",
  "anio" : 2005,
  "actores" : [
    "Johnny Depp"
  ]
}
{
  "titulo" : "Avatar",
  "director" : "James Cameron",
  "anio" : 2005,
  "actores" : [
    "Sam Worthington"
  ]
}
{
  "titulo" : "Tomb Raider",
  "director" : "Roar Uthaug",
  "anio" : 2018,
  "actores" : [
    "Alicia Vikander"
  ]
}
{
  "titulo" : "Deadpool",
  "director" : "Tim Miller",
  "anio" : 2016,
  "actores" : [
    "Ryan Reynolds"
  ]
}
{
  "titulo" : "El joven manos de Tijera",
  "director" : "Tim Burton",
  "anio" : 1990,
  "actores" : [
    "Johnny Depp"
  ]
}
{
  "titulo" : "Sin City",
  "director" : "Quentin Tarantino",
  "anio" : 2005,
  "actores" : [
    "Bruce Willis"
  ]
}
```

```
{
  "titulo" : "Piratas del Caribe: El cofre de la Muerte",
  "director" : "Gore Verbinski",
  "anio" : 2006,
  "actores" : [
    "Johnny Depp"
  ]
}
{
  "titulo" : "El señor de los Anillos: Las Dos Torres",
  "director" : "Peter Jackson",
  "anio" : 2005,
  "actores" : [
    "Orlando Bloom"
  ]
}
{
  "titulo" : "X-Men Dias del Futuro Pasado",
  "director" : "Brian Singer",
  "anio" : 2014,
  "actores" : [
    "Ian McKellen"
  ]
}
```

Actualizar Documentos

2.a Agregar actores a película con título: 'Assassin's Creed'

```
> db.peliculas.updateOne( { "titulo": "Assassin's Creed" }, { $set: {"actores": ["Michael Fassbender", "Marion Cotillard", "Jeremy Irons"]} } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.peliculas.find( { "titulo": "Assassin's Creed" }, { _id: 0 } ).pretty()
{
  "titulo" : "Assassin's Creed",
  "director" : "Justin Kurzel",
  "anio" : 2016,
  "actores" : [
    "Michael Fassbender",
    "Marion Cotillard",
    "Jeremy Irons"
  ]
}
```

2.b Agregar sinopsis a "Deadpool"

```
> db.peliculas.updateOne( { "titulo": "Deadpool" }, { $set: {"sinopsis": "Basado en el anti-héroe menos convencional de la Marvel, Deadpool narra el origen de un ex-operativo de la fuerzas especiales llamado Wade Wilson, reconvertido a mercenario, y que tras ser sometido a un cruel experimento adquiere poderes de curación rápida, adoptando Wade entonces el alter ego de Deadpool."} } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.peliculas.find( { "titulo": "Deadpool" }, { _id: 0 } ).pretty()
{
  "titulo" : "Deadpool",
  "director" : "Tim Miller",
  "anio" : 2016,
  "actores" : [
    "Ryan Reynolds",
    "Morena Bacarin"
  ],
  "sinopsis" : "Basado en el anti-héroe menos convencional de la Marvel, Deadpool narra el origen de un ex-operativo de la fuerzas especiales llamado Wade Wilson, reconvertido a mercenario, y que tras ser sometido a un cruel experimento adquiere poderes de curación rápida, adoptando Wade entonces el alter ego de Deadpool."
}
```

2.c Agregar un actor llamado "Tim Owen" a la película "Sin City"

```
> db.peliculas.updateOne( { "titulo": "Sin City" }, { $push: {"actores": "Tim Owen" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.find( { "titulo": "Sin City" }, { _id: 0 } ).pretty()
{
  "titulo" : "Sin City",
  "director" : "Quentin Tarantino",
  "anio" : 2005,
  "actores" : [
    "Bruce Willis",
    "Jessica Alba",
    "Tim Owen"
  ]
}
```


2.d Agregar género a cada película

```
> db.peliculas.updateMany( {}, { $set: { "genero": [] } }, { "multi": true } )
{ "acknowledged" : true, "matchedCount" : 10, "modifiedCount" : 10 }
> db.peliculas.updateOne( { "titulo": "Assassin's Creed" }, { $push: { "genero": "Acción" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.updateOne( { "titulo": "El cadáver de la novia" }, { $push: { "genero": "Animación" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.updateOne( { "titulo": "Avatar" }, { $push: { "genero": "Acción" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.updateOne( { "titulo": "Tom Raider" }, { $push: { "genero": "Acción" } } )
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
>
> db.peliculas.updateOne( { "titulo": "Tomb Raider" }, { $push: { "genero": "Acción" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.updateOne( { "titulo": "Deadpool" }, { $push: { "genero": "Acción" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.updateOne( { "titulo": "El joven manos de Tijera" }, { $push: { "genero": "Drama" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.updateOne( { "titulo": "Sin City" }, { $push: { "genero": "Crimen" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.updateOne( { "titulo": "Piratas del Caribe: El cofre de la Muerte" }, { $push: { "genero": "Acción" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.updateOne( { "titulo": "El señor de los Anillos: Las Dos Torres" }, { $push: { "genero": "Adventura" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
> db.peliculas.updateOne( { "titulo": "X-Men Dias del Futuro Pasado" }, { $push: { "genero": "Acción" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

2.e Agregar el actor Juanito Alimaña en todas las películas en la posición 1 del arreglo de actores.

```
> db.peliculas.updateMany( {}, { $push: { "actores": { $each: [ "Juanito Alimaña" ], $position: 1 } } }, { "multi": true } )
{ "acknowledged" : true, "matchedCount" : 10, "modifiedCount" : 10 }
```

```
> db.peliculas.find( {}, { _id: 0, titulo: 1, actores: 1 } ).pretty()
{
  "titulo" : "El cadáver de la novia",
  "actores" : [
    "Johnny Depp",
    "Juanito Alimaña",
    "Helena Bonham-Carter"
  ]
}
{
  "titulo" : "Avatar",
  "actores" : [
    "Sam Worthington",
    "Juanito Alimaña",
    "Zoe Saldana",
    "Sigourney Weaver"
  ]
}
{
  "titulo" : "Tomb Raider",
  "actores" : [
    "Alicia Vikander",
    "Juanito Alimaña",
    "Dominic West"
  ]
}
{
  "titulo" : "Deadpool",
  "actores" : [
    "Ryan Reynolds",
    "Juanito Alimaña",
    "Morena Bacarin"
  ]
}
{
  "titulo" : "El joven manos de Tijera",
  "actores" : [
    "Johnny Depp",
    "Juanito Alimaña",
    "Winona Ryder"
  ]
}
{
  "titulo" : "Sin City",
  "actores" : [
    "Bruce Willis",
    "Juanito Alimaña",
    "Jessica Alba",
    "Tim Owen"
  ]
}
```



```
{
  "titulo" : "Piratas del Caribe: El cofre de la Muerte",
  "actores" : [
    "Johnny Depp",
    "Juanito Alimaña",
    "Orlando Bloom",
    "Keira Knightley"
  ]
}
{
  "titulo" : "El señor de los Anillos: Las Dos Torres",
  "actores" : [
    "Orlando Bloom",
    "Juanito Alimaña",
    "Elijah Wood",
    "Ian McKellen"
  ]
}
{
  "titulo" : "X-Men Dias del Futuro Pasado",
  "actores" : [
    "Ian McKellen",
    "Juanito Alimaña",
    "Hugh Jackman"
  ]
}
{
  "titulo" : "Assassin's Creed",
  "actores" : [
    "Michael Fassbender",
    "Juanito Alimaña",
    "Marion Cotillard",
    "Jeremy Irons"
  ]
}
```

3. En la misma base de datos tallerMongo, crear una colección llamada Transmilenio

```
"ruta" : {
  "Portal 80 - Portal Usme" : {
    "H20 - D20" : {
      "paradas" : [
        {
          "Portal 80" : {
            "Alimentador" : {
              "1-1" : "Álamos",
              "1-2" : "Garcés Navas",
              "1-3" : "Villas de Granada",
              "1-4" : "Cortijo",
              "1-5" : "Colsubsidio",
              "1-6" : "Bolivia Oriental",
              "1-7" : "Quirigua",
              "1-8" : "Calle 80",
              "1-9" : "Villas del Dorado"
            }
          },
          "Avenida Cali",
          "Granja - Carrera 77",
          "Boyacá",
          "Ferias",
          "Avenida 68",
          "Polo",
          "Calle 45",
          "Calle 34",
          "Calle 26",
          "Calle 19",
          "Avenida Jiménez",
          "Calle 40 Sur",
          "Santa Lucía",
          "Molinos",
          {
            "Portal Usme" : {
              "Alimentador" : {
                "3-1" : "La Aurora",
                "3-2" : "Santa Librada",
                "3-3" : "Chiniza",
                "3-4" : "Alfonso López",
                "3-5" : "Usminia",
                "3-6" : "Danubio",
                "3-7" : "Avenida Caracas",
                "3-8" : "Virrey",
                "3-9" : "Marichuela",
                "3-10" : "Usme Centro",
                "3-11" : "La Fiscala",
                "3-12" : "Nebraska"
              }
            }
          }
        ]
      }
    }
  }
}
```

```
> db.transmilenio.insert(ruta)
WriteResult({ "nInserted" : 1 })
```