

# actividad2

February 2, 2023

**Curso: Herramientas de Bases de Datos**

**Estudiante: Santiago Castro Moncada**

## 1 Indice

1. Análisis de la base de datos relacional.
2. Análisis de los datos de la base de datos relacional.
3. Migración de los datos de la base de datos relacional.
4. Migración de la base de datos relacional a no relacional (MongoDB).
  - 4.1 Conexión a la bbdd no relacional
  - 4.2 Creación de la bbdd no relacional
  - 4.3 Insertar los documentos a las colecciones
  - 4.4 Inserción de todos los datos
5. Consultas a base de datos no relacional.
  - 5.1 Consultas generales
  - 5.2 Consultas simples/básicas
  - 5.3 Consultas avanzadas
6. Actualizaciones a base de datos no relacional.
  - 6.1 Actualizar el valor de un campo de una tabla de la bbdd
  - 6.2 Cambiar el tipo de datos de un campo de una tabla de la bbdd
7. Borrados a base de datos no relacional.
  - 7.1 Borrar un registro de la bbdd
  - 7.2 Eliminar un campo de una tabla de la bbdd
  - 7.3 Borrar una tabla de la bbdd
8. Conclusiones.

## 2 1. Análisis de la base de datos relacional

La base de datos de las estadísticas del deporte beisbol se encuentra conformada por las siguientes tablas y sus respectivos campos.

### 2.0.1 Tabla Jugador

**Campos:** (jugadorID, nombre, apellidos, nombreDeportivo, anyoNacimiento, mesNacimiento, diaNacimiento, paisNacimiento, estadoNacimiento, ciudadNacimiento, anyoFallecimiento, mesFallecimiento, diaFallecimiento, paisFallecimiento, estadoFallecimiento, ciudadFallecimiento, peso, altura, batear, lanzar, fechaDebut, fechaRetirada)

PK: jugadorID

### 2.0.2 Tabla Estadio

**Campos:** (estadioID, nombre, alias, ciudad, estado, pais)

PK: estadioID

### 2.0.3 Tabla Equipo

**Campos:** (anyo, liga, equipoId, equipoNombre, estadio, publicoEstadio, equipoDivision, clasificacionFinal, G, Ghome, W, L, DivWin, WCWin, LgWin, WSWin, R, AB, H, B2, B3, HR, BB, SO, SB, CS, HBP, SF, RA, ER, ERA, CG, SHO, SV, IPouts, HA, HRA, BBA, SOA, E, DP, FP, BPF, PPF)

PK: equipoId

### 2.0.4 Tabla Universidad

**Campos:** (universidadID, nombre, ciudad, estado, pais)

PK: universidadID

### 2.0.5 Tabla BatearEstadistica

**Campos:** (id, jugador, anyo, equipo, liga, G, AB, R, H, B2, B3, HR, RBI, SB, CS, BB, SO, IBB, HBP, SH, SF, GIDP)

PK: id

FK1: BatearEstadistica.jugador -> Jugador.jugadorID

FK2: BatearEstadistica.equipo -> Equipo.equipoId

### 2.0.6 Tabla LanzarEstadistica

**Campos:** (id, jugador, anyo, equipo, liga, W, L, G, GS, CG, SHO, SV, IPouts, H, ER, HR, BB, SO, BAOpp, ERA, IBB, WP, HBP, BK, BFP, GF, R, SH, SF, GIDP)

PK: id

FK1: LanzarEstadistica.jugador -> Jugador.jugadorID

FK2: LanzarEstadistica.equipo -> Equipo.equipoId

### 2.0.7 Tabla CampoEstadistica

**Campos:** (id, jugador, anyo, equipo, liga, POS, G, GS, InnOuts, PO, A, E, DP, PB, WP, SB, CS, ZR)

PK: id

FK1: CampoEstadistica.jugador -> Jugador.jugadorID

FK2: CampoEstadistica.equipo -> Equipo.equipoId

### 2.0.8 Tabla Sueldo

**Campos:** (id, anyo, equipo, liga, jugador, sueldo)

PK: id

FK1: Sueldo.jugador -> Jugador.jugadorID

FK2: Sueldo.equipo -> Equipo.equipoId

### 2.0.9 Tabla JugadorPremio

**Campos:** (id, jugador, premio, anyo, liga)

PK: id

FK1: JugadorPremio.jugador -> Jugador.jugadorID

### 2.0.10 Tabla JugadorPosicionCampo

**Campos:** (id, jugador, anyo, Glf, Gcf, Grf)

PK: id

FK1: JugadorPosicionCampo.jugador -> Jugador.jugadorID ### **Tabla PremioVotacion**  
**Campos:**

(id, premio, anyo, liga, jugador, puntosGanados, puntosPremio, votosPrimerPuesto)

PK: id

FK1: PremioVotacion.jugador -> Jugador.jugadorID

### 2.0.11 Tabla JugadorPosicionDetalle

**Campos:** (id, anyo, equipo, liga, jugador, G\_all, GS, G\_batting, G\_defense, G\_p, G\_c, G\_1b, G\_2b, G\_3b, G\_ss, G\_lf, G\_cf, G\_rf, G\_of, G\_dh, G\_ph, G\_pr)

PK: id

FK1: JugadorPosicionDetalle.jugador -> Jugador.jugadorID

FK2: JugadorPosicionDetalle.equipo -> Equipo.equipoId

### 2.0.12 Tabla JugadorUniversidad

**Campos:** (id, jugador, universidad, anyo)

PK: id

FK1: JugadorUniversidad.jugador -> Jugador.jugadorID

FK2: JugadorUniversidad.universidad -> Universidad.universidadID

### 2.0.13 Tabla EstadioPartidos

**Campos:** (id, anyo, liga, equipo, estadio, primerPartido, ultimoPartido, partidosJugados, publico)

PK: id

FK1: EstadioPartidos.equipo -> Equipo.EquipoID

FK2: EstadioPartidos.estadio -> Estadio.estadioID

## 3 2. Análisis de los datos de la base de datos relacional.

### 3.0.1 Tabla jugador

- jugadorID se transforma en ObjectID
- nombre, apellidos, nombreDeportivo, paisNacimiento, estadoNacimiento, ciudadNacimiento, paisFallecimiento, estadoFallecimiento, ciudadFallecimiento, batear, lanzar se transforman en un dato String.
- anyoNacimiento, mesNacimiento, diaNacimiento, anyoFallecimiento, mesFallecimiento, diaFallecimiento, peso, altura se transforman en dato int.
- fechaDebut, fechaRetirada continúan siendo Date

### 3.0.2 Tabla equipo

- equipoID se transforma en ObjectID
- anyo, clasificacionFinal, G, Ghom, W, L R, AB, H, B2, B3, HR, BB, SO, SB, CS, HBP, SF, RA, ER, ERA, CG, SHO, SV, Ipouts, HA, HRA, BBA, SOA, E, DP, FP, BPF, PFF se transforman en integer
- liga, equipoNombre, estadio se transforman en String
- publicoEstadio INTEGER se transforma en integer
- equipoDivision, DivWin, WCWin, LgWin, WSWin se transforman en String

### 3.0.3 Tabla estadio

- estadioID se transforma en ObjectID
- nombre, alias, ciudad, estado, pais se transforman en String

### 3.0.4 Tabla universidad

- universidadID se transforma en ObjectID
- nombre, ciudad, estado, pais se transforman en String.

### **3.0.5 Tabla sueldo:**

- anyo se transforma en integer
- liga se transforma en String
- jugador, equipo se transforman en String
- sueldo se transforma en integer

### **3.0.6 Tabla jugadorPosicionCampo:**

- jugador se transforma en String
- anyo, Glf, Gcf, Grf se transforman en integer

### **3.0.7 Tabla jugadorPremio:**

- jugador, premio se transforman en String
- anyo se transforma en integer
- liga se transforma String

### **3.0.8 Tabla premioVotacion:**

- jugador, premio se transforman en String
- liga se transforma String
- anyo, puntosGanados, puntosPremio, votosPrimerPuesto se transforman en integer

### **3.0.9 Tabla batearEstadistica:**

- jugador, equipo se transforman en String
- liga se transforma en String
- anyo, G, AB, R, H, B2, B3, HR, RBI, SB, CS, BB, SO, IBB, HBP, SH, SF, GIDP se transforman en integer

### **3.0.10 Tabla lanzarEstadistica**

- jugador, equipo se transforman en String
- liga se transforma en String
- BAOpp se transforma en Double
- anyo, W, L, G, GS, CG, SHO, SV, IPouts, H, ER, HR, BB, SO, ERA, IBB, WP, HBP, BK, BFP, GF, R, SH, SF, GIDP se transforman en integer

### **3.0.11 Tabla campoEstadistica:**

- jugador, equipo se transforman en String
- liga, POS se transforman en String
- anyo, G, GS, InnOuts, PO, A, E, DP, PB, WP, SB, CS, ZR se transforman en integer

### **3.0.12 Tabla estadioPartidos:**

- liga se transforma String
- equipo, estadio, primerPartido, ultimoPartido se transforman en String
- anyo se transforma en integer
- publico se transforma en integer

### 3.0.13 Tabla jugadorPosicionDetalle:

- jugador, equipo se transforman en String
- liga se transforma String
- anyo, G\_all, GS, G\_batting, G\_defense, G\_p, G\_c, G\_1b, G\_2b, G\_3b, G\_ss, G\_lf, G\_cf, G\_rf, G\_of, G\_dh, G\_ph, G\_pr se transforman en integer

### 3.0.14 Tabla jugadorUniversidad:

- jugador, universidad se convierten en String
- anyo se transforma en integer

## 4 3. Migración a la base de datos No relacional

Para migrar la base de datos relacional (Postgres) a una base de datos no relacional (MongoDB) es importante tener en cuenta que MongoDB maneja de forma diferente las relaciones, por lo que se replantearon las relaciones existentes en la base de datos relacional para adaptarlas a MongoDB, ya que MongoDB maneja las relaciones de forma denormalizada, mediante la incorporación de la información en un solo documento. De este análisis se optó por crear 4 colecciones (jugador, equipo, estadio, universidad) dentro de la base de datos en MongoDB.

### 4.1 Esquema de datos

La razón por la cual se crean colecciones separadas para la tabla jugadores, equipos, estadios y universidades es debido a que cada uno de estos representa una entidad o un objeto de negocio diferente, y almacenarlos en colecciones separadas permite una mayor flexibilidad en la estructura de los datos, facilitando la consulta y actualización de la información. Además, al tener estas entidades en colecciones separadas, se pueden crear índices específicos para cada una de ellas y agregar las relaciones adjuntas dentro de la colección, lo que mejoraría el rendimiento de las consultas.

#### 4.1.1 1. Colección Jugador

En esta colección de datos se tendrá agrupada la información completa del usuario que se encuentra relacionada con sueldos, premios, votaciones, estadísticas, posiciones de juego y universidades, lo que ayudará a optimizar las consultas a la hora de acceder a la información relacionada con el jugador.

esquema de datos:

```
{
  "_id": "ObjectID",
  "jugadorID": "String",
  "nombre": "String",
  "apellido": "String",
  ...
  ...
  ...
  "fechaDebut": "Date",
  "fechaRetirada": "Date",
  "posiciones": [
```

```

    {
        "anyo": "Integer",
        "equipo": "String",
        "liga": "String",
        "G_all": "Integer",
        "GS": "Integer",
        "G_batting": "Integer",
        "G_defense": "Integer",
        ...
        "G_ph": "Integer",
        "G_pr": "Integer",
    },
    ...
],
"sueldos":[
    {
        "anyo": "Integer",
        "equipo": "String",
        "liga": "String",
        "sueldo": "Integer",
    },
    ...
],
"premios": [
    {
        "premio": "String",
        "anyo": "Integer",
        "liga": "String"
    },
    ...
],
"votaciones": [
    {
        "premio": "String",
        "anyo": "Integer",
        "liga": "String",
        "puntosGanados": "Integer",
        "puntosPremio": "Integer",
        "votosPrimerPuesto": "Integer"
    },
    ...
],
"estadisticas":{
    "bateo": [
        {
            "anyo": "Integr",
            "equipo": "String",
            "liga": "String",

```

```

        "G": "Integer",
        "AB": "Integer",
        ...
        ...
        "SF": "Integer",
        "GIDP": "Integer",
    },
    ...
],
"lanzar": [
    {
        "anyo": "Integer",
        "equipo": "String",
        "liga": "String",
        "W": "Integer",
        "L": "Integer",
        ...
        ...
        "SF": "Integer",
        "GIDP": "Integer",
    },
    ...
],
"campo": [
    {
        "anyo": "Integer",
        "equipo": "String",
        "liga": "String",
        "POS": "Integer",
        "G": "Integer",
        ...
        "CS": "Integer",
        "ZR": "Integer",
    },
    ...
]
},
"universidades": [
    {
        "universidad": "String",
        "anyo": "Integer",
    },
    ...
]
}

```



#### 4.1.2 2. Colección Equipo

La tabla de equipo se puede representar mediante una colección como tal, ya que esta no comprende relaciones de alto nivel que impliquen tener que crear listas adjuntas.

esquema de datos:

```
{
  "_id": "ObjectID",
  "equipoId": "String",
  "anyo": "Integer",
  "liga": "String",
  "equipoNombre": "String",
  "estadio": "String",
  "publicoEstadio": "Integer",
  "equipoDivision": "String",
  ...
  ...
  "BPF": "Integer",
  "PPF": "Integer"
}
```

#### 4.1.3 3. Colección Estadio

La colección de estadio representa la información de la tabla de estadio y adicional a esto se tiene una lista adjunta de los partidos que se han disputado en este estadio.

esquema de datos:

```
{
  "_id": "ObjectID",
  "estadioID": "String",
  "nombre": "String",
  "alias": "String",
  "ciudad": "String",
  "estado": "String",
  "paisnombre": "String",
  "partidos": [
    {
      "anyo": "Integer",
      "liga": "String",
      "equipo": "String",
      "primerPartido": "Date",
      "ultimoPartido": "Date",
      "partidosJugados": "Integer",
      "publico": "Integer"
    },
    ...
  ]
}
```

#### 4.1.4 4. Colección Universidad

La tabla de universidad se puede representar mediante una colección como tal, ya que esta no comprende relaciones de alto nivel que impliquen tener que crear listas adjuntas.

esquema de datos:

```
{
  "_id": "ObjectID",
  "universidadID": "String",
  "nombre": "String",
  "ciudad": "String",
  "estado": "String",
  "pais": "String"
}
```

## 5 4. Migración de la base de datos relacional a no relacional (MongoDB)

### 5.1 4.1 Conexión a la bbdd no relacional

```
[ ]: # Conexión con MongoDB
import pymongo
import csv
import datetime

USER='usuario'
PASSWORD='ivo47qOLkyTGtJAh'

try:
    client = pymongo.MongoClient(f"mongodb+srv://{USER}:{PASSWORD}@cluster0.
    ↳kgpyong.mongodb.net/?retryWrites=true&w=majority")
except pymongo.errors.OperationFailure as error:
    print(error)

# Listado de las bases de datos existentes
print(client.list_database_names())
```

/usr/lib/python3/dist-packages/requests/\_\_init\_\_.py:89:

RequestsDependencyWarning: urllib3 (1.26.8) or chardet (3.0.4) doesn't match a supported version!

warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported "

['beisbol', 'sample\_guides', 'sample\_mflix', 'sample\_supplies',  
'sample\_training', 'sample\_weatherdata', 'admin', 'local']

## 5.2 4.2 Creación de la bbdd no relacional

```
[ ]: db = client['beisbol']
db.create_collection("jugador")
db.create_collection("equipo")
db.create_collection("estadio")
db.create_collection("universidad")
```

### 5.2.1 4.3. Insertar los documentos a las colecciones

#### Colección Jugador

```
[ ]: #Colección jugadores
collection_jugador = db["jugador"]
jugador1={
    "jugadorID": "mtplance12",
    "nombre": "Martin",
    "apellidos": "Plance",
    "nombreDeportivo": "Martin Plance",
    "anyoNacimiento": 1934,
    "mesNacimiento": 2,
    "diaNacimiento": 5,
    "paisNacimiento": "USA",
    "estadoNacimiento": "AL",
    "ciudadNacimiento": "Mobile",
    "anyoFallecimiento": 2021,
    "mesFallecimiento": 1,
    "diaFallecimiento": 22,
    "paisFallecimiento": "USA",
    "estadoFallecimiento": "GA",
    "ciudadFallecimiento": "Georgia",
    "peso": 180,
    "altura": 72,
    "batear": "R",
    "lanzar": "R",
    "fechaDebut": datetime.datetime.strptime("1974-07-13", "%Y-%m-%d"),
    "fechaRetirada": datetime.datetime.strptime("1986-11-03", "%Y-%m-%d"),
    "sueldos": [
        {
            "anyo": 1978,
            "equipo": "Dallas FC",
            "liga": "NC",
            "sueldo": 56000
        },
        {
            "anyo": 1990,
            "equipo": "NY soccer",
            "liga": "NC",
```

```

        "sueldo": 78000
    }
],
"premios": [],
"votaciones": [
    {
        "premio": "MVP",
        "anyo": 1955,
        "liga": "NL",
        "puntosGanados": 36,
        "puntosPremio": 36,
        "votosPrimerPuesto": 0
    },
    {
        "premio": "MVP",
        "anyo": 1956,
        "liga": "NL",
        "puntosGanados": 144,
        "puntosPremio": 336,
        "votosPrimerPuesto": 0
    }
],
"estadisticas": {
    "bateo": [
        {
            "anyo": 1984,
            "equipo": "TL1",
            "liga": "NL",
            "G": 122,
            "AB": 468,
            "R": 58,
            "H": 131,
            "B2": 27,
            "B3": 6,
            "HR": 13,
            "RBI": 69,
            "SB": 2,
            "CS": 2,
            "BB": 28,
            "SO": 39,
            "IBB": 0,
            "HBP": 3,
            "SH": 6,
            "SF": 4,
            "GIDP": 13
        },
        {

```

```

    "anyo": 1955,
    "equipo": "ML1",
    "liga": "NL",
    "G": 153,
    "AB": 602,
    "R": 105,
    "H": 189,
    "B2": 37,
    "B3": 9,
    "HR": 27,
    "RBI": 106,
    "SB": 3,
    "CS": 1,
    "BB": 49,
    "SO": 61,
    "IBB": 5,
    "HBP": 3,
    "SH": 7,
    "SF": 4,
    "GIDP": 20
  },
  {
    "anyo": 1976,
    "equipo": "ML4",
    "liga": "AL",
    "G": 85,
    "AB": 271,
    "R": 22,
    "H": 62,
    "B2": 8,
    "B3": 0,
    "HR": 10,
    "RBI": 35,
    "SB": 0,
    "CS": 1,
    "BB": 35,
    "SO": 38,
    "IBB": 1,
    "HBP": 0,
    "SH": 0,
    "SF": 2,
    "GIDP": 8
  }
],
"lanzar": [],
"campo": [
  {

```

```

        "anyo": 1954,
        "equipo": "ML1",
        "liga": "NL",
        "POS": "OF",
        "G": 116,
        "GS": 113,
        "InnOuts": 3093,
        "PO": 223,
        "A": 5,
        "E": 7,
        "DP": 0,
        "PB": 0,
        "WP": 0,
        "SB": 0,
        "CS": 0,
        "ZR": 0
    },
    {
        "anyo": 1955,
        "equipo": "ML1",
        "liga": "NL",
        "POS": "2B",
        "G": 27,
        "GS": 26,
        "InnOuts": 709,
        "PO": 86,
        "A": 84,
        "E": 6,
        "DP": 23,
        "PB": 0,
        "WP": 0,
        "SB": 0,
        "CS": 0,
        "ZR": 0
    }
]
},
"posiciones": [
    {
        "anyo": 1954,
        "Glf": 105,
        "Gcf": 0,
        "Grf": 11
    },
    {
        "anyo": 1955,
        "Glf": 26,

```

```

        "Gcf": 0,
        "Grf": 105
    }
],
"posicionDetalle": [
    {
        "anyo": 1954,
        "equipo": "ML1",
        "liga": "NL",
        "G_all": 122,
        "GS": 113,
        "G_batting": 122,
        "G_defense": 116,
        "G_p": 0,
        "G_c": 0,
        "G_1b": 0,
        "G_2b": 0,
        "G_3b": 0,
        "G_ss": 0,
        "G_lf": 105,
        "G_cf": 0,
        "G_rf": 11,
        "G_of": 116,
        "G_dh": 0,
        "G_ph": 7,
        "G_pr": 1
    },
    {
        "anyo": 1955,
        "equipo": "ML1",
        "liga": "NL",
        "G_all": 153,
        "GS": 151,
        "G_batting": 153,
        "G_defense": 151,
        "G_p": 0,
        "G_c": 0,
        "G_1b": 0,
        "G_2b": 27,
        "G_3b": 0,
        "G_ss": 0,
        "G_lf": 30,
        "G_cf": 0,
        "G_rf": 104,
        "G_of": 126,
        "G_dh": 0,
        "G_ph": 2,
    }
]

```

```

        "G_pr": 0
    }
],
"universidades": [
    {
        "universidad": "pennst",
        "anyo": 2003
    },
    {
        "universidad": "vermont",
        "anyo": 1998
    }
]
}

jugador2={
    "jugadorID": "rdgache34",
    "nombre": "Ronald",
    "apellidos": "Gache",
    "nombreDeportivo": "Ronald Gache",
    "anyoNacimiento": 1934,
    "mesNacimiento": 2,
    "diaNacimiento": 5,
    "paisNacimiento": "USA",
    "estadoNacimiento": "AL",
    "ciudadNacimiento": "Mobile",
    "anyoFallecimiento": 2021,
    "mesFallecimiento": 1,
    "diaFallecimiento": 22,
    "paisFallecimiento": "USA",
    "estadoFallecimiento": "GA",
    "ciudadFallecimiento": "Georgia",
    "peso": 180,
    "altura": 72,
    "batear": "R",
    "lanzar": "R",
    "fechaDebut": datetime.datetime.strptime("1974-07-13", "%Y-%m-%d"),
    "fechaRetirada": datetime.datetime.strptime("1986-11-03", "%Y-%m-%d"),
    "sueldos": [],
    "premios": [
        {
            "premio": "Most Valuable Player",
            "anyo": 1937,
            "liga": "NL"
        },
        {
            "premio": "TSN All-Star",

```



```

        "anyo": 1981,
        "liga": "NL"
    }
],
"votaciones": [
    {
        "premio": "MVP",
        "anyo": 1955,
        "liga": "NL",
        "puntosGanados": 36,
        "puntosPremio": 36,
        "votosPrimerPuesto": 0
    },
    {
        "premio": "MVP",
        "anyo": 1956,
        "liga": "NL",
        "puntosGanados": 144,
        "puntosPremio": 336,
        "votosPrimerPuesto": 0
    }
],
"estadisticas": {
    "bateo": [
        {
            "anyo": 1984,
            "equipo": "TL1",
            "liga": "NL",
            "G": 122,
            "AB": 468,
            "R": 58,
            "H": 131,
            "B2": 27,
            "B3": 6,
            "HR": 13,
            "RBI": 69,
            "SB": 2,
            "CS": 2,
            "BB": 28,
            "SO": 39,
            "IBB": 0,
            "HBP": 3,
            "SH": 6,
            "SF": 4,
            "GIDP": 13
        },
        {

```

```

    "anyo": 1955,
    "equipo": "ML1",
    "liga": "NL",
    "G": 153,
    "AB": 602,
    "R": 105,
    "H": 189,
    "B2": 37,
    "B3": 9,
    "HR": 27,
    "RBI": 106,
    "SB": 3,
    "CS": 1,
    "BB": 49,
    "SO": 61,
    "IBB": 5,
    "HBP": 3,
    "SH": 7,
    "SF": 4,
    "GIDP": 20
  },
  {
    "anyo": 1976,
    "equipo": "ML4",
    "liga": "AL",
    "G": 85,
    "AB": 271,
    "R": 22,
    "H": 62,
    "B2": 8,
    "B3": 0,
    "HR": 10,
    "RBI": 35,
    "SB": 0,
    "CS": 1,
    "BB": 35,
    "SO": 38,
    "IBB": 1,
    "HBP": 0,
    "SH": 0,
    "SF": 2,
    "GIDP": 8
  }
],
"lanzar": [],
"campo": [
  {

```

```

        "anyo": 1954,
        "equipo": "ML1",
        "liga": "NL",
        "POS": "OF",
        "G": 116,
        "GS": 113,
        "InnOuts": 3093,
        "PO": 223,
        "A": 5,
        "E": 7,
        "DP": 0,
        "PB": 0,
        "WP": 0,
        "SB": 0,
        "CS": 0,
        "ZR": 0
    },
    {
        "anyo": 1955,
        "equipo": "ML1",
        "liga": "NL",
        "POS": "2B",
        "G": 27,
        "GS": 26,
        "InnOuts": 709,
        "PO": 86,
        "A": 84,
        "E": 6,
        "DP": 23,
        "PB": 0,
        "WP": 0,
        "SB": 0,
        "CS": 0,
        "ZR": 0
    }
]
},
"posiciones": [
    {
        "anyo": 1954,
        "Glf": 105,
        "Gcf": 0,
        "Grf": 11
    },
    {
        "anyo": 1955,
        "Glf": 26,

```

```

        "Gcf": 0,
        "Grf": 105
    }
],
"posicionDetalle": [
    {
        "anyo": 1954,
        "equipo": "ML1",
        "liga": "NL",
        "G_all": 122,
        "GS": 113,
        "G_batting": 122,
        "G_defense": 116,
        "G_p": 0,
        "G_c": 0,
        "G_1b": 0,
        "G_2b": 0,
        "G_3b": 0,
        "G_ss": 0,
        "G_lf": 105,
        "G_cf": 0,
        "G_rf": 11,
        "G_of": 116,
        "G_dh": 0,
        "G_ph": 7,
        "G_pr": 1
    },
    {
        "anyo": 1955,
        "equipo": "ML1",
        "liga": "NL",
        "G_all": 153,
        "GS": 151,
        "G_batting": 153,
        "G_defense": 151,
        "G_p": 0,
        "G_c": 0,
        "G_1b": 0,
        "G_2b": 27,
        "G_3b": 0,
        "G_ss": 0,
        "G_lf": 30,
        "G_cf": 0,
        "G_rf": 104,
        "G_of": 126,
        "G_dh": 0,
        "G_ph": 2,
    }
]

```

```

        "G_pr": 0
    }
],
"universidades": [
    {
        "universidad": "pennst",
        "anyo": 2003
    },
    {
        "universidad": "vermont",
        "anyo": 1998
    }
]
}

collection_jugador.insert_many([jugador1, jugador2])

```

[ ]: <pymongo.results.InsertManyResult at 0x7fa3bc1c4550>

## Colección Equipo

```

[ ]: #Colección equipos
collection_equipo=db["equipo"]
equipo1={
    "equipoId": "PAL",
    "anyo": 2002,
    "liga": "NL",
    "equipoNombre": "Mender Dacotass",
    "estadio": "Lukis Field",
    "publicoEstadio": 3498977,
    "equipoDivision": "W",
    "clasificacionFinal": 1,
    "G": 162,
    "Ghome": 81,
    "W": 98,
    "L": 64,
    "DivWin": "Y",
    "WCWin": "N",
    "LgWin": "N",
    "WSWin": "N",
    "R": 679,
    "AB": 5489,
    "H": 1297,
    "B2": 308,
    "B3": 31,
    "HR": 144,
    "BB": 537,

```

```

"SO": 1465,
"SB": 43,
"SF": 28,
"RA": 893,
"ER": 804,
"ERA": 511,
"CG": 3,
"SHO": 4,
"SV": 22,
"IPouts": 4252,
"HA": 1480,
"HRA": 232,
"BBA": 555,
"SOA": 1238,
"E": 100,
"DP": 113,
"FP": 983,
"BPF": 99,
"PPF": 101
}

equipo2={
  "equipoId": "PPT",
  "anyo": 2001,
  "liga": "NL",
  "equipoNombre": "Londer Braves",
  "estadio": "Mexico Arena",
  "publicoEstadio": 1020314,
  "equipoDivision": "E",
  "clasificacionFinal": 5,
  "G": 161,
  "Ghome": 81,
  "W": 68,
  "L": 93,
  "DivWin": "N",
  "WCWin": "N",
  "LgWin": "N",
  "WSWin": "N",
  "R": 649,
  "AB": 5514,
  "H": 1404,
  "B2": 295,
  "B3": 27,
  "HR": 122,
  "BB": 502,
  "SO": 1240,
  "SB": 75,

```

```

    "CS": 34,
    "HBP": 59,
    "SF": 52,
    "RA": 779,
    "ER": 725,
    "ERA": 451,
    "CG": 1,
    "SHO": 9,
    "HA": 1414,
    "HRA": 177,
    "BBA": 547,
    "SOA": 1227,
    "E": 101,
    "DP": 134,
    "FP": 983,
    "BPF": 102,
    "PPF": 104
}
collection_equipo.insert_many([equipo1,equipo2])

```

```
[ ]: <pymongo.results.InsertManyResult at 0x7fa3cc1b09d0>
```

## Colección Estadio

```

[ ]: #Colección estadios
collection_estadio=db["estadio"]

estadio1={
    "estadioID": "tAL04",
    "nombre": "Talouis Bail",
    "alias": "",
    "ciudad": "Toulouse",
    "estado": "MD",
    "paisnombre": "US",
    "partidos": [
        {
            "anyo": 1884,
            "liga": "UA",
            "equipo": "BLU",
            "primerPartido": datetime.datetime.strptime("1884-04-17", "%Y-%m-%d"),
            "ultimoPartido": datetime.datetime.strptime("1884-06-17", "%Y-%m-%d"),
            "partidosJugados": 49,
            "publico": 0
        },
        {
            "anyo": 1884,
            "liga": "UA",

```

```

        "equipo": "CHU",
        "primerPartido": datetime.datetime.strptime("1876-04-17", "%Y-%m-%d"),
        "ultimoPartido": datetime.datetime.strptime("1877-04-17", "%Y-%m-%d"),
        "partidosJugados": 1,
        "publico": 0
    }
]
}
estadio2={
    "estadioID": "LAL12",
    "nombre": "Oriole Park IV",
    "alias": "American League Park",
    "ciudad": "Baltimore",
    "estado": "MD",
    "paisnombre": "US",
    "historialpartidos": [
        {
            "anyo": 1901,
            "liga": "AL",
            "equipo": "BLA",
            "primerPartido": datetime.datetime.strptime("1888-04-13", "%Y-%m-%d"),
            "ultimoPartido": datetime.datetime.strptime("1889-01-17", "%Y-%m-%d"),
            "partidosJugados": 66,
            "publico": 137193
        },
        {
            "anyo": 1902,
            "liga": "AL",
            "equipo": "BLA",
            "primerPartido": datetime.datetime.strptime("1890-04-17", "%Y-%m-%d"),
            "ultimoPartido": datetime.datetime.strptime("1899-05-11", "%Y-%m-%d"),
            "partidosJugados": 64,
            "publico": 161307
        }
    ]
}
collection_estadio.insert_many([estadio1,estadio2])

```

```
[ ]: <pymongo.results.InsertManyResult at 0x7fa3bc257040>
```

## Colección Universidad

```

[ ]: #Colección universidad
collection_universidad=db["universidad"]
universidad1={
    "universidadID":"phristmast",
    "nombre":"Ristmas Pensilvane University",

```



```

        "ciudad": "Alabama",
        "estado": "AL",
        "pais": "USA"
    }
    universidad2={
        "universidadID": "poldelphi",
        "nombre": "New Delphi University",
        "ciudad": "Dormund City",
        "estado": "NY",
        "pais": "USA"
    }
    collection_universidad.insert_many([universidad1, universidad2])

```

```
[ ]: <pymongo.results.InsertManyResult at 0x7fa3bc257d60>
```

## 5.2.2 4.4 Inserción de todos los datos

```

[ ]: # Inserción de la coleccion de jugadores
import csv
import datetime
from itertools import groupby
RUTA_LOCAL='/mnt/c/Users/santiago/Downloads/ficheros/'

def convert_date(date:str) -> datetime:
    try:
        date = datetime.datetime.strptime(date, "%Y-%m-%d")
    except Exception as e:
        try:
            date = datetime.datetime.strptime(date, "%d/%m/%Y")
        except Exception as e:
            date = None
    return date

values = {}
with open(f'{RUTA_LOCAL}1-Jugador.csv', newline='', encoding='utf-8-sig') as f:
    reader = csv.DictReader(f, delimiter=';')
    for index, row in enumerate(reader):
        for value in row:
            if value in ('fechaDebut', 'fechaRetirada'):
                row[value] = convert_date(row[value])
            else:
                row[value] = int(row[value]) if row[value].isdigit() else None
        values[row['jugadorID']] = row

# Agregar estadísticas de bateo

```

```

    with open(f'{RUTA_LOCAL}2-BatearEstadistica.csv', newline='',
encoding='utf-8-sig') as File:
        reader = csv.DictReader(File, delimiter=';')
        for key, group in groupby(reader, key=lambda x: x['jugador']):
            group = list(group)
            for value in group:
                for i in value:
                    value[i] = int(value[i]) if value[i].isdigit() else
value[i] if value[i] else None
            values[key]['estadisticas'] = {}
            values[key]['estadisticas']['bateo'] = group

    # Agregar estadisticas de lanzamiento
    with open(f'{RUTA_LOCAL}3-LanzarEstadistica.csv', newline='',
encoding='utf-8-sig') as File:
        reader = csv.DictReader(File, delimiter=';')
        for key, group in groupby(reader, key=lambda x: x['jugador']):
            group = list(group)
            for value in group:
                for i in value:
                    value[i] = int(value[i]) if value[i].isdigit() else
value[i] if value[i] else None
            values[key]['estadisticas']['lanzar'] = group

    # # Agregar estadisticas de campo
    with open(f'{RUTA_LOCAL}4-CampoEstadistica.csv', newline='',
encoding='utf-8-sig') as File:
        reader = csv.DictReader(File, delimiter=';')
        for key, group in groupby(reader, key=lambda x: x['jugador']):
            group = list(group)
            for value in group:
                for i in value:
                    value[i] = int(value[i]) if value[i].isdigit() else
value[i] if value[i] else None
            values[key]['estadisticas']['campo'] = group

    # # # Agregar sueldos
    with open(f'{RUTA_LOCAL}6-Sueldo.csv', newline='', encoding='utf-8-sig') as
File:
        reader = csv.DictReader(File, delimiter=';')
        for key, group in groupby(reader, key=lambda x: x['jugador']):
            group = list(group)
            for value in group:
                for i in value:
                    value[i] = int(value[i]) if value[i].isdigit() else
value[i] if value[i] else None

```

```

        values[key]['sueldos'] = group

    ### Agregar Premios
    with open(f'{RUTA_LOCAL}8-JugadorPremio.csv', newline='',
encoding='utf-8-sig') as File:
        reader = csv.DictReader(File, delimiter=';')
        for key, group in groupby(reader, key=lambda x: x['jugador']):
            group = list(group)
            for value in group:
                for i in value:
                    value[i] = int(value[i]) if value[i].isdigit() else
value[i] if value[i] else None
            values[key]['premios'] = group

    ### Agregar Posiciones
    with open(f'{RUTA_LOCAL}7-JugadorPosicionCampo.csv', newline='',
encoding='utf-8-sig') as File:
        reader = csv.DictReader(File, delimiter=';')
        for key, group in groupby(reader, key=lambda x: x['jugador']):
            group = list(group)
            for value in group:
                for i in value:
                    value[i] = int(value[i]) if value[i].isdigit() else
value[i] if value[i] else None
            values[key]['posiciones'] = group

    ### Agregar Votaciones
    with open(f'{RUTA_LOCAL}9-PremioVotacion.csv', newline='',
encoding='utf-8-sig') as File:
        reader = csv.DictReader(File, delimiter=';')
        for key, group in groupby(reader, key=lambda x: x['jugador']):
            group = list(group)
            for value in group:
                for i in value:
                    value[i] = int(value[i]) if value[i].isdigit() else
value[i] if value[i] else None
            values[key]['votaciones'] = group

    ### Agregar posicionesDetalle
    with open(f'{RUTA_LOCAL}11-JugadorPosicionDetalle.csv', newline='',
encoding='utf-8-sig') as File:
        reader = csv.DictReader(File, delimiter=';')
        for key, group in groupby(reader, key=lambda x: x['jugador']):
            group = list(group)
            for value in group:
                for i in value:

```

```

        value[i] = int(value[i]) if value[i].isdigit() else
↪value[i] if value[i] else None
        values[key]['posicionDetalle'] = group

    ### Agregar posicionesDetalle
    with open(f'{RUTA_LOCAL}13-JugadorUniversidad.csv', newline='',
↪encoding='utf-8-sig') as File:
        reader = csv.DictReader(File, delimiter=';')
        for key, group in groupby(reader, key=lambda x: x['jugador']):
            group = list(group)
            for value in group:
                for i in value:
                    value[i] = int(value[i]) if value[i].isdigit() else
↪value[i] if value[i] else None
            values[key]['universidades'] = group
collection = db["jugador"]
collection.insert_many(list(values.values()))
print('CREACIÓN EXITOSA')

```

CREACIÓN EXITOSA

```

[ ]: # Inserción de la coleccion de equipos
import csv
import datetime
from itertools import groupby
RUTA_LOCAL='/mnt/c/Users/santiago/Downloads/ficheros/'

values = {}
with open(f'{RUTA_LOCAL}5-Equipo.csv', newline='', encoding='utf-8-sig') as
↪File:
    reader = csv.DictReader(File, delimiter=';')
    for row in reader:
        for value in row:
            row[value] = int(row[value]) if row[value].isdigit() else
↪row[value] if row[value] else None
        values[row['equipoId']] = **row

collection = db["equipo"]
collection.insert_many(list(values.values()))
print('CREACIÓN EXITOSA DE LA COLECCIÓN DE EQUIPO')

```

CREACIÓN EXITOSA DE LA COLECCIÓN DE EQUIPO

```

[ ]: # Inserción de la coleccion de universidades
import csv
import datetime
from itertools import groupby

```

```

RUTA_LOCAL='/mnt/c/Users/santiago/Downloads/ficheros/'

values = {}
with open(f'{RUTA_LOCAL}12-Universidad.csv', newline='', encoding='utf-8-sig') as File:
    reader = csv.DictReader(File, delimiter=';')
    for row in reader:
        for value in row:
            row[value] = int(row[value]) if row[value].isdigit() else None
        values[row['universidadID']] = {**row}

collection = db["universidad"]
collection.insert_many(list(values.values()))
print('CREACIÓN EXITOSA DE LA COLECCIÓN DE UNIVERSIDAD')

```

CREACIÓN EXITOSA DE LA COLECCIÓN DE UNIVERSIDAD

```

[ ]: # Inserción de la coleccion de estadio
import csv
import datetime
from itertools import groupby
RUTA_LOCAL='/mnt/c/Users/santiago/Downloads/ficheros/'

def convert_date(date:str) -> datetime:
    try:
        date = datetime.datetime.strptime(date, "%Y-%m-%d")
    except Exception as e:
        try:
            date = datetime.datetime.strptime(date, "%d/%m/%Y")
        except Exception as e:
            date = None
    return date

values = {}
with open(f'{RUTA_LOCAL}10-Estadio.csv', newline='', encoding='utf-8-sig') as File:
    reader = csv.DictReader(File, delimiter=';')
    for row in reader:
        for value in row:
            row[value] = int(row[value]) if row[value].isdigit() else None
        values[row['estadioID']] = {**row}

    # Agregar partidos
    with open(f'{RUTA_LOCAL}14-EstadioPartidos.csv', newline='', encoding='utf-8-sig') as File:

```

```

reader = csv.DictReader(File, delimiter=';')
for key, group in groupby(reader, key=lambda x: x['estadio']):
    group = list(group)
    for value in group:
        for i in value:
            if i in ('primerPartido', 'ultimoPartido'):
                value[i] = convert_date(value[i])
            else:
                value[i] = int(value[i]) if value[i].isdigit() else
↪value[i] if value[i] else None
        values[key]['partidos'] = group

collection = db["estadio"]
collection.insert_many(list(values.values()))
print('CREACIÓN EXITOSA DE LA COLECCIÓN DE ESTADIO')

```

CREACIÓN EXITOSA DE LA COLECCIÓN DE ESTADIO

## 6 5. Consultas a base de datos no relacional

## 5.1. Consultas generales

```

[ ]: ## Listar colecciones
collections = db.list_collection_names()
print(collections)

```

['universidad', 'jugador', 'estadio', 'equipo']

```

[ ]: #Listar los campos de las colecciones

## Jugador
collection = db['jugador'].find({})[0]
print('Campos colección Jugador')
print(list(collection.keys()))

## equipo
collection = db['equipo'].find({})[0]
print('Campos colección Equipo')
print(list(collection.keys()))

## estadio
collection = db['estadio'].find({})[0]
print('Campos colección Estadio')
print(list(collection.keys()))

## universidad
collection = db['universidad'].find({})[0]

```

```
print('Campos colección Universidad')
print(list(collection.keys()))
```

Campos colección Jugador

```
['_id', 'jugadorID', 'nombre', 'apellidos', 'nombreDeportivo', 'anyoNacimiento',
'mesNacimiento', 'diaNacimiento', 'paisNacimiento', 'estadoNacimiento',
'ciudadNacimiento', 'anyoFallecimiento', 'mesFallecimiento', 'diaFallecimiento',
'paisFallecimiento', 'estadoFallecimiento', 'ciudadFallecimiento', 'peso',
'altura', 'batear', 'lanzar', 'fechaDebut', 'fechaRetirada', 'sueldos',
'premios', 'votaciones', 'estadisticas', 'posiciones', 'posicionDetalle',
'universidades']
```

Campos colección Equipo

```
['_id', 'equipoId', 'anyo', 'liga', 'equipoNombre', 'estadio', 'publicoEstadio',
'equipoDivision', 'clasificacionFinal', 'G', 'Ghome', 'W', 'L', 'DivWin',
'WCWin', 'LgWin', 'WSWin', 'R', 'AB', 'H', 'B2', 'B3', 'HR', 'BB', 'SO', 'SB',
'CS', 'HBP', 'SF', 'RA', 'ER', 'ERA', 'CG', 'SHO', 'HA', 'HRA', 'BBA', 'SOA',
'E', 'DP', 'FP', 'BPF', 'PPF']
```

Campos colección Estadio

```
['_id', 'estadioID', 'nombre', 'alias', 'ciudad', 'estado', 'paisnombre',
'historialpartidos']
```

Campos colección Universidad

```
['_id', 'universidadID', 'nombre', 'ciudad', 'estado', 'pais']
```

[ ]: *#Listar los primeros 3 documentos de las colecciones*

```
## Jugador
collection = db['jugador'].find({})[:3]
print('Documentos de la colección Jugador')
print(list(collection))

## equipo
collection = db['equipo'].find({})[:3]
print('Documentos de la colección Jugador')
print(list(collection))

## estadio
collection = db['estadio'].find({})[:3]
print('Documentos de la colección Estadio')
print(list(collection))

## universidad
collection = db['universidad'].find({})[:3]
print('Documentos de la colección Universidad')
print(list(collection))
```

Documentos de la colección Jugador

```
{['_id': ObjectId('63d3410a94df42c5c832de3e'), 'jugadorID': 'mtplance12',
'nombre': 'Martin', 'apellidos': 'Plance', 'nombreDeportivo': 'Martin Plance',
```

```

'anyoNacimiento': 1934, 'mesNacimiento': 2, 'diaNacimiento': 5,
'paisNacimiento': 'USA', 'estadoNacimiento': 'AL', 'ciudadNacimiento': 'Mobile',
'anyoFallecimiento': 2021, 'mesFallecimiento': 1, 'diaFallecimiento': 22,
'paisFallecimiento': 'USA', 'estadoFallecimiento': 'GA', 'ciudadFallecimiento':
'Georgia', 'peso': 180, 'altura': 72, 'batear': 'R', 'lanzar': 'R',
'fechaDebut': datetime.datetime(1974, 7, 13, 0, 0), 'fechaRetirada':
datetime.datetime(1986, 11, 3, 0, 0), 'sueldos': [{ 'anyo': 1978, 'equipo':
'Dallas FC', 'liga': 'NC', 'sueldo': 56000}, { 'anyo': 1990, 'equipo': 'NY
soccer', 'liga': 'NC', 'sueldo': 78000}], 'premios': [], 'votaciones':
[{ 'premio': 'MVP', 'anyo': 1955, 'liga': 'NL', 'puntosGanados': 36,
'puntosPremio': 36, 'votosPrimerPuesto': 0}, { 'premio': 'MVP', 'anyo': 1956,
'liga': 'NL', 'puntosGanados': 144, 'puntosPremio': 336, 'votosPrimerPuesto':
0}], 'estadisticas': { 'bateo': [{ 'anyo': 1984, 'equipo': 'TL1', 'liga': 'NL',
'G': 122, 'AB': 468, 'R': 58, 'H': 131, 'B2': 27, 'B3': 6, 'HR': 13, 'RBI': 69,
'SB': 2, 'CS': 2, 'BB': 28, 'SO': 39, 'IBB': 0, 'HBP': 3, 'SH': 6, 'SF': 4,
'GIDP': 13}, { 'anyo': 1955, 'equipo': 'ML1', 'liga': 'NL', 'G': 153, 'AB': 602,
'R': 105, 'H': 189, 'B2': 37, 'B3': 9, 'HR': 27, 'RBI': 106, 'SB': 3, 'CS': 1,
'BB': 49, 'SO': 61, 'IBB': 5, 'HBP': 3, 'SH': 7, 'SF': 4, 'GIDP': 20}, { 'anyo':
1976, 'equipo': 'ML4', 'liga': 'AL', 'G': 85, 'AB': 271, 'R': 22, 'H': 62, 'B2':
8, 'B3': 0, 'HR': 10, 'RBI': 35, 'SB': 0, 'CS': 1, 'BB': 35, 'SO': 38, 'IBB': 1,
'HBP': 0, 'SH': 0, 'SF': 2, 'GIDP': 8}], 'lanzar': [], 'campo': [{ 'anyo': 1954,
'equipo': 'ML1', 'liga': 'NL', 'POS': 'OF', 'G': 116, 'GS': 113, 'InnOuts':
3093, 'PO': 223, 'A': 5, 'E': 7, 'DP': 0, 'PB': 0, 'WP': 0, 'SB': 0, 'CS': 0,
'ZR': 0}, { 'anyo': 1955, 'equipo': 'ML1', 'liga': 'NL', 'POS': '2B', 'G': 27,
'GS': 26, 'InnOuts': 709, 'PO': 86, 'A': 84, 'E': 6, 'DP': 23, 'PB': 0, 'WP': 0,
'SB': 0, 'CS': 0, 'ZR': 0}], 'posiciones': [{ 'anyo': 1954, 'Glf': 105, 'Gcf':
0, 'Grf': 11}, { 'anyo': 1955, 'Glf': 26, 'Gcf': 0, 'Grf': 105}],
'posicionDetalle': [{ 'anyo': 1954, 'equipo': 'ML1', 'liga': 'NL', 'G_all': 122,
'GS': 113, 'G_batting': 122, 'G_defense': 116, 'G_p': 0, 'G_c': 0, 'G_1b': 0,
'G_2b': 0, 'G_3b': 0, 'G_ss': 0, 'G_lf': 105, 'G_cf': 0, 'G_rf': 11, 'G_of':
116, 'G_dh': 0, 'G_ph': 7, 'G_pr': 1}, { 'anyo': 1955, 'equipo': 'ML1', 'liga':
'NL', 'G_all': 153, 'GS': 151, 'G_batting': 153, 'G_defense': 151, 'G_p': 0,
'G_c': 0, 'G_1b': 0, 'G_2b': 27, 'G_3b': 0, 'G_ss': 0, 'G_lf': 30, 'G_cf': 0,
'G_rf': 104, 'G_of': 126, 'G_dh': 0, 'G_ph': 2, 'G_pr': 0}], 'universidades':
[{ 'universidad': 'pennst', 'anyo': 2003}, { 'universidad': 'vermont', 'anyo':
1998}], { '_id': ObjectId('63d3410a94df42c5c832de3f'), 'jugadorID': 'rdgache34',
'nombre': 'Ronald', 'apellidos': 'Gache', 'nombreDeportivo': 'Ronald Gache',
'anyoNacimiento': 1934, 'mesNacimiento': 2, 'diaNacimiento': 5,
'paisNacimiento': 'USA', 'estadoNacimiento': 'AL', 'ciudadNacimiento': 'Mobile',
'anyoFallecimiento': 2021, 'mesFallecimiento': 1, 'diaFallecimiento': 22,
'paisFallecimiento': 'USA', 'estadoFallecimiento': 'GA', 'ciudadFallecimiento':
'Georgia', 'peso': 180, 'altura': 72, 'batear': 'R', 'lanzar': 'R',
'fechaDebut': datetime.datetime(1971, 2, 23, 0, 0), 'fechaRetirada':
datetime.datetime(1975, 1, 3, 0, 0), 'sueldos': [], 'premios': [{ 'premio': 'Most
Valuable Player', 'anyo': 1937, 'liga': 'NL'}, { 'premio': 'TSN All-Star',
'anyo': 1981, 'liga': 'NL'}], 'votaciones': [{ 'premio': 'MVP', 'anyo': 1955,
'liga': 'NL', 'puntosGanados': 36, 'puntosPremio': 36, 'votosPrimerPuesto': 0},
{ 'premio': 'MVP', 'anyo': 1956, 'liga': 'NL', 'puntosGanados': 144,

```



'puntosPremio': 336, 'votosPrimerPuesto': 0}], 'estadisticas': {'bateo':  
 [{'anyo': 1984, 'equipo': 'TL1', 'liga': 'NL', 'G': 122, 'AB': 468, 'R': 58,  
 'H': 131, 'B2': 27, 'B3': 6, 'HR': 13, 'RBI': 69, 'SB': 2, 'CS': 2, 'BB': 28,  
 'SO': 39, 'IBB': 0, 'HBP': 3, 'SH': 6, 'SF': 4, 'GIDP': 13}, {'anyo': 1955,  
 'equipo': 'ML1', 'liga': 'NL', 'G': 153, 'AB': 602, 'R': 105, 'H': 189, 'B2':  
 37, 'B3': 9, 'HR': 27, 'RBI': 106, 'SB': 3, 'CS': 1, 'BB': 49, 'SO': 61, 'IBB':  
 5, 'HBP': 3, 'SH': 7, 'SF': 4, 'GIDP': 20}, {'anyo': 1976, 'equipo': 'ML4',  
 'liga': 'AL', 'G': 85, 'AB': 271, 'R': 22, 'H': 62, 'B2': 8, 'B3': 0, 'HR': 10,  
 'RBI': 35, 'SB': 0, 'CS': 1, 'BB': 35, 'SO': 38, 'IBB': 1, 'HBP': 0, 'SH': 0,  
 'SF': 2, 'GIDP': 8}], 'lanzar': [], 'campo': [{'anyo': 1954, 'equipo': 'ML1',  
 'liga': 'NL', 'POS': 'OF', 'G': 116, 'GS': 113, 'InnOuts': 3093, 'PO': 223, 'A':  
 5, 'E': 7, 'DP': 0, 'PB': 0, 'WP': 0, 'SB': 0, 'CS': 0, 'ZR': 0}, {'anyo': 1955,  
 'equipo': 'ML1', 'liga': 'NL', 'POS': '2B', 'G': 27, 'GS': 26, 'InnOuts': 709,  
 'PO': 86, 'A': 84, 'E': 6, 'DP': 23, 'PB': 0, 'WP': 0, 'SB': 0, 'CS': 0, 'ZR':  
 0}], 'posiciones': [{'anyo': 1954, 'Glf': 105, 'Gcf': 0, 'Grf': 11}, {'anyo':  
 1955, 'Glf': 26, 'Gcf': 0, 'Grf': 105}], 'jugadorposiciondetalle': [{'anyo':  
 1954, 'equipo': 'ML1', 'liga': 'NL', 'G\_all': 122, 'GS': 113, 'G\_batting': 122,  
 'G\_defense': 116, 'G\_p': 0, 'G\_c': 0, 'G\_1b': 0, 'G\_2b': 0, 'G\_3b': 0, 'G\_ss':  
 0, 'G\_lf': 105, 'G\_cf': 0, 'G\_rf': 11, 'G\_of': 116, 'G\_dh': 0, 'G\_ph': 7,  
 'G\_pr': 1}, {'anyo': 1955, 'equipo': 'ML1', 'liga': 'NL', 'G\_all': 153, 'GS':  
 151, 'G\_batting': 153, 'G\_defense': 151, 'G\_p': 0, 'G\_c': 0, 'G\_1b': 0, 'G\_2b':  
 27, 'G\_3b': 0, 'G\_ss': 0, 'G\_lf': 30, 'G\_cf': 0, 'G\_rf': 104, 'G\_of': 126,  
 'G\_dh': 0, 'G\_ph': 2, 'G\_pr': 0}], 'universidades': [{'universidad': 'pennst',  
 'anyo': 2003}, {'universidad': 'vermont', 'anyo': 1998}], {'\_id':  
 ObjectId('63d57b1cffbe9ca91b44b98d'), 'jugadorID': 'aardsda01', 'nombre':  
 'David', 'apellidos': 'Aardsma', 'nombreDeportivo': 'David Allan',  
 'anyoNacimiento': 1981, 'mesNacimiento': 12, 'diaNacimiento': 27,  
 'paisNacimiento': 'USA', 'estadoNacimiento': 'CO', 'ciudadNacimiento': 'Denver',  
 'anyoFallecimiento': None, 'mesFallecimiento': None, 'diaFallecimiento': None,  
 'paisFallecimiento': None, 'estadoFallecimiento': None, 'ciudadFallecimiento':  
 None, 'peso': 215, 'altura': 75, 'batear': 'R', 'lanzar': 'R', 'fechaDebut':  
 datetime.datetime(2004, 4, 6, 0, 0), 'fechaRetirada': datetime.datetime(2015, 8,  
 23, 0, 0), 'estadisticas': {'bateo': [{'jugador': 'aardsda01', 'anyo': 2015,  
 'equipo': 'ATL', 'liga': 'NL', 'G': 33, 'AB': 1, 'R': 0, 'H': 0, '2B': 0, '3B':  
 0, 'HR': 0, 'RBI': 0, 'SB': 0, 'CS': 0, 'BB': 0, 'SO': 1, 'IBB': 0, 'HBP': 0,  
 'SH': 0, 'SF': 0, 'GIDP': 0}], 'lanzar': [{'jugador': 'aardsda01', 'anyo': 2015,  
 'equipo': 'ATL', 'liga': 'NL', 'W': 1, 'L': 1, 'G': 33, 'GS': 0, 'CG': 0, 'SHO':  
 0, 'SV': 0, 'IPouts': 92, 'H': 25, 'ER': 16, 'HR': 6, 'BB': 14, 'SO': 35,  
 'BAOpp': '0.223', 'ERA': 470, 'IBB': 3, 'WP': 1, 'HBP': 1, 'BK': 0, 'BFP': 129,  
 'GF': 9, 'R': 17, 'SH': 0, 'SF': 1, 'GIDP': 4}], 'campo': [{'jugador':  
 'aardsda01', 'anyo': 2015, 'equipo': 'ATL', 'liga': 'NL', 'POS': 'P', 'G': 33,  
 'GS': 0, 'InnOuts': 92, 'PO': 0, 'A': 1, 'E': 1, 'DP': 0, 'PB': None, 'WP':  
 None, 'SB': None, 'CS': None, 'ZR': None}], 'sueldos': [{'anyo': 2012,  
 'equipo': 'NYA', 'liga': 'AL', 'jugador': 'aardsda01', 'sueldo': 500000}],  
 'posicionDetalle': [{'anyo': 2015, 'equipo': 'ATL', 'liga': 'NL', 'jugador':  
 'aardsda01', 'G\_all': 33, 'GS': 0, 'G\_batting': 30, 'G\_defense': 33, 'G\_p': 33,  
 'G\_c': 0, 'G\_1b': 0, 'G\_2b': 0, 'G\_3b': 0, 'G\_ss': 0, 'G\_lf': 0, 'G\_cf': 0,  
 'G\_rf': 0, 'G\_of': 0, 'G\_dh': 0, 'G\_ph': 0, 'G\_pr': 0}], 'universidades':

```
[{'jugador': 'aardsda01', 'universidad': 'pennst', 'anyo': 2001}, {'jugador': 'aardsda01', 'universidad': 'rice', 'anyo': 2002}, {'jugador': 'aardsda01', 'universidad': 'rice', 'anyo': 2003}]]]

Documentos de la colección Jugador
[{'_id': ObjectId('63d343c294df42c5c832de41'), 'equipoId': 'PPT', 'anyo': 2001, 'liga': 'NL', 'equipoNombre': 'Londer Braves', 'estadio': 'Mexico Arena', 'publicoEstadio': 1020314, 'equipoDivision': 'E', 'clasificacionFinal': 5, 'G': 161, 'Ghome': 81, 'W': 68, 'L': 93, 'DivWin': 'N', 'WCWin': 'N', 'LgWin': 'N', 'WSWin': 'N', 'R': 649, 'AB': 5514, 'H': 1404, 'B2': 295, 'B3': 27, 'HR': 122, 'BB': 502, 'SO': 1240, 'SB': 75, 'CS': 34, 'HBP': 59, 'SF': 52, 'RA': 779, 'ER': 725, 'ERA': 451, 'CG': 1, 'SHO': 9, 'HA': 1414, 'HRA': 177, 'BBA': 547, 'SOA': 1227, 'E': 101, 'DP': 134, 'FP': 983, 'BPF': 102, 'PPF': 104}, {'_id': ObjectId('63d343c294df42c5c832de40'), 'equipoId': 'PAL', 'anyo': 2002, 'liga': 'NL', 'equipoNombre': 'Mender Dacotass', 'estadio': 'Lukis Field', 'publicoEstadio': 3498977, 'equipoDivision': 'W', 'clasificacionFinal': 1, 'G': 162, 'Ghome': 81, 'W': 98, 'L': 64, 'DivWin': 'Y', 'WCWin': 'N', 'LgWin': 'N', 'WSWin': 'N', 'R': 679, 'AB': 5489, 'H': 1297, 'B2': 308, 'B3': 31, 'HR': 144, 'BB': 537, 'SO': 1465, 'SB': 43, 'SF': 28, 'RA': 893, 'ER': 804, 'ERA': 511, 'CG': 3, 'SHO': 4, 'SV': 22, 'IPouts': 4252, 'HA': 1480, 'HRA': 232, 'BBA': 555, 'SOA': 1238, 'E': 100, 'DP': 113, 'FP': 983, 'BPF': 99, 'PPF': 101}, {'_id': ObjectId('63d587f7ffbe9ca91b45091f'), 'anyo': 1884, 'liga': 'UA', 'equipoId': 'ALT', 'equipoNombre': 'Altoona Mountain City', 'estadio': None, 'publicoEstadio': None, 'equipoDivision': None, 'clasificacionFinal': 10, 'G': 25, 'Ghome': None, 'W': 6, 'L': 19, 'DivWin': None, 'WCWin': None, 'LgWin': 'N', 'WSWin': None, 'R': 90, 'AB': 899, 'H': 223, 'B2': 30, 'B3': 6, 'HR': 2, 'BB': 22, 'SO': 130, 'SB': None, 'CS': None, 'HBP': None, 'SF': None, 'RA': 216, 'ER': 114, 'ERA': 467, 'CG': 20, 'SHO': 0, 'SV': 0, 'IPouts': 659, 'HA': 292, 'HRA': 3, 'BBA': 52, 'SOA': 93, 'E': 156, 'DP': 4, 'FP': 862, 'BPF': 101, 'PPF': 109, 'partidos': [{'anyo': 1884, 'liga': 'UA', 'equipo': 'ALT', 'estadio': 'ALT01', 'primerPartido': datetime.datetime(1884, 4, 30, 0, 0), 'ultimoPartido': datetime.datetime(1884, 5, 31, 0, 0), 'partidosJugados': 18, 'publico': 0}]]]

Documentos de la colección Estadio
[{'_id': ObjectId('63d3454094df42c5c832de43'), 'estadioID': 'LAL12', 'nombre': 'Oriole Park IV', 'alias': 'American League Park', 'ciudad': 'Baltimore', 'estado': 'MD', 'paisnombre': 'US', 'historialpartidos': [{'anyo': 1901, 'liga': 'AL', 'equipo': 'BLA', 'primerPartido': datetime.datetime(1888, 4, 13, 0, 0), 'ultimoPartido': datetime.datetime(1889, 1, 17, 0, 0), 'partidosJugados': 66, 'publico': 137193}, {'anyo': 1902, 'liga': 'AL', 'equipo': 'BLA', 'primerPartido': datetime.datetime(1890, 4, 17, 0, 0), 'ultimoPartido': datetime.datetime(1899, 5, 11, 0, 0), 'partidosJugados': 64, 'publico': 161307}]], {'_id': ObjectId('63d3454094df42c5c832de42'), 'estadioID': 'tAL04', 'nombre': 'Talouis Bail', 'alias': '', 'ciudad': 'Tolouse', 'estado': 'MD', 'paisnombre': 'US', 'partidos': [{'anyo': 1884, 'liga': 'UA', 'equipo': 'BLU', 'primerPartido': datetime.datetime(1884, 4, 17, 0, 0), 'ultimoPartido': datetime.datetime(1884, 6, 17, 0, 0), 'partidosJugados': 49, 'publico': 0}, {'anyo': 1884, 'liga': 'UA', 'equipo': 'CHU', 'primerPartido': datetime.datetime(1876, 4, 17, 0, 0), 'ultimoPartido': datetime.datetime(1877, 4, 17, 0, 0), 'partidosJugados': 1, 'publico': 0}]], {'_id':
```

```
ObjectId('63d589c5ffbe9ca91b450e6f'), 'estadioID': 'ALB01', 'nombre': 'Riverside Park', 'alias': None, 'ciudad': 'Albany', 'estado': 'NY', 'pais': 'US']}]
```

Documentos de la colección Universidad

```
[{'_id': ObjectId('63d3465094df42c5c832de45'), 'universidadID': 'poldelphi', 'nombre': 'New Delphi University', 'ciudad': 'Dormund City', 'estado': 'NY', 'pais': 'USA'}, {'_id': ObjectId('63d3465094df42c5c832de44'), 'universidadID': 'phristmast', 'nombre': 'Ristmas Pensilvane University', 'ciudad': 'Alabama', 'estado': 'AL', 'pais': 'USA'}, {'_id': ObjectId('63d58996ffbe9ca91b4509b4'), 'universidadID': 'abilchrist', 'nombre': 'Abilene Christian University', 'ciudad': 'Abilene', 'estado': 'TX', 'pais': 'USA'}]
```

## 6.1 5.2 Consultas simples/básicas

### 6.1.1 1. Descripción:

Esta consulta devuelve la información del equipo pasado como parámetro

```
[ ]: # este parametro es variable, insertar acá el Id del equipo a consultar
equipo='BRF'
consulta = db['equipo'].find({'equipoId': equipo})[0]
print(f'Esta es la información que se tiene del equipo {equipo}:')
print(consulta)
```

Esta es la información que se tiene del equipo BRF:

```
{'_id': ObjectId('63d587f7ffbe9ca91b450933'), 'anyo': 1915, 'liga': 'FL', 'equipoId': 'BRF', 'equipoNombre': 'Brooklyn Tip-Tops', 'estadio': None, 'publicoEstadio': None, 'equipoDivision': None, 'clasificacionFinal': 7, 'G': 153, 'Ghome': None, 'W': 70, 'L': 82, 'DivWin': None, 'WCWin': None, 'LgWin': 'N', 'WSWin': None, 'R': 647, 'AB': 5035, 'H': 1348, 'B2': 205, 'B3': 75, 'HR': 36, 'BB': 473, 'SO': 654, 'SB': 249, 'CS': None, 'HBP': None, 'SF': None, 'RA': 673, 'ER': 507, 'ERA': 337, 'CG': 78, 'SHO': 10, 'SV': 16, 'IPouts': 4065, 'HA': 1299, 'HRA': 27, 'BBA': 536, 'SOA': 467, 'E': 285, 'DP': 103, 'FP': 955, 'BPF': 99, 'PPF': 100, 'partidos': [{'anyo': 1915, 'liga': 'FL', 'equipo': 'BRF', 'estadio': 'NYC19', 'primerPartido': datetime.datetime(1915, 4, 10, 0, 0), 'ultimoPartido': datetime.datetime(1915, 9, 30, 0, 0), 'partidosJugados': 74, 'publico': 27641}]]}
```

### 6.1.2 2. Descripción:

Esta consulta devuelve las universidades en las que ha participado un jugador.

```
[ ]: jugador = 'abbotky01'
consulta=db['jugador'].find_one({'jugadorID': jugador})
print(f'El jugador {jugador} ha participado en {len(set(list(map(lambda x: x["universidad"], consulta.get("universidades", []))))}) universidades')
if consulta.get("universidades", []):
    print(set(list(map(lambda x: x['universidad'], consulta.get("universidades", []))))))
```

El jugador abbotky01 ha participado en 2 universidades  
{'ucsd', 'longbeach'}

## 6.2 5.3 Consultas Avanzadas

### 6.2.1 1. Descripción:

Consultar cual fue el estadio que albergó mas público durante un lapso de tiempo

```
[ ]: # formato de fecha YYYY-MM-DD

from datetime import datetime

fecha_inicio='1950-10-01' # parámetros variables
fecha_final='1970-07-01'

def convert_date(date:str):
    return datetime.strptime(date, '%Y-%m-%d')

consulta=db['estadio'].find({"$and": [
    {'partidos.
    ↪primerPartido': {'$gte': convert_date(fecha_inicio)}},
    {'partidos.
    ↪ultimoPartido': {'$lte': convert_date(fecha_final)}},
    ]}).sort("partidos.publico",
    ↪-1).limit(1)[0]
print(f"El estadio que albergó mas público para este rango de fechas fue el
    ↪estadio {consulta['nombre']} "
      f"con un total de {consulta['partidos'][0]['publico']}
    ↪personas")
```

El estadio que albergó mas público para este rango de fechas fue el estadio Los Angeles Memorial Coliseum con un total de 1804865 personas

### 6.2.2 2. Descripción

Seleccionar el top 5 de los jugadores con mayor número de jonrones en sus estadísticas de bateo

```
[ ]: from bson.son import SON
pipeline = [
    {'$unwind': '$estadisticas.bateo'},
    {'$group': {'_id': '_id', 'count': {'$sum': '$estadisticas.bateo.HR'}}},
    {'$sort': SON([('count', -1)])},
    {'$limit': 5},
]

consulta=db['jugador'].aggregate(pipeline)
for i in consulta:
    sub_consulta = db['jugador'].find_one({"_id": i['_id']},
```

```

        {'jugadorID': True, 'nombreDeportivo':
↪True})
    print(f"El jugador conocido como {sub_consulta['nombreDeportivo']} tiene un_
↪registro de "
        f"{i['count']} jonrones")

```

El jugador conocido como Martin Plance tiene un registro de 50 jonrones  
 El jugador conocido como Ronald Gache tiene un registro de 50 jonrones  
 El jugador conocido como Salvador Johan tiene un registro de 48 jonrones  
 El jugador conocido como Vladimir tiene un registro de 48 jonrones  
 El jugador conocido como Shohei tiene un registro de 46 jonrones

## 7 6. Actualizaciones a base de datos no relacional

### 7.0.1 6.1 Actualizar el valor de un campo de una tabla de la bbdd

En este ejemplo se actualizará el nombre deportivo de un jugador

```

[ ]: jugadorID="abbotku01"
    nombre_deportivo_nuevo="Kiliam Boutar"
    db['jugador'].update_one({"jugadorID": jugadorID},
        {"$set": {"nombreDeportivo":
↪nombre_deportivo_nuevo}})
    print('La información del jugador fue actualizada exitosamente')

# Confirmación de la actualización del registros
    confirmar = db['jugador'].find_one({"jugadorID": jugadorID},
        {"nombreDeportivo": 1})
    print(confirmar['nombreDeportivo'])

```

La información del jugador fue actualizada exitosamente  
 Kiliam Boutar

### 7.0.2 6.2 Cambiar el tipo de datos de un campo de una tabla de la bbdd.

En este ejemplo se cambiará el dato **altura** y **peso** del jugador a un campo de tipo String

```

[ ]: from pprint import pprint

    db['jugador'].update_many(
        {"altura": {"$type": "int"}, "peso": {"$type": "int"}},
        [{"$set": {"altura": {"$toString": "$altura"}},
        {"$set": {"peso": {"$toString": "$peso"}}}
    ]
    )
    print('Se actualizó correctamente las claves altura y peso de la colección de_
↪jugador')

```

```
# Confirmación de la actualización
confirmar = db['jugador'].find({}, {"altura": 1, 'peso': 1})[0]
pprint(confirmar)
```

Se actualizó correctamente las claves altura y peso de la colección de jugador  
{'\_id': ObjectId('63d3410a94df42c5c832de3e'), 'altura': '72', 'peso': '180'}

## 8 7. Borrados a base de datos no relacional

### 8.0.1 7.1 Borrar un registro de la bbdd

En este ejemplo se eliminará la información de un estadio y con ello la información de los partidos jugados allí

```
[ ]: estadioID="CLE05"

db["estadio"].delete_one({"estadioID": estadioID})

print(f'Se borró correctamente la información del estadio {estadioID}')
```

Se borró correctamente la información del estadio CLE05

### 8.0.2 7.2 Eliminar un campo de una tabla de la bbdd

En este ejemplo se eliminará el campo **pais** de la colección de estadio

```
[ ]: db["estadio"].update_many({}, {"$unset": {"pais": ""}})

print(f'Se Eliminó exitosamente el campo pais de la coleccion estadio')
```

Se Eliminó exitosamente el campo pais de la coleccion estadio

### 8.0.3 7.3 Borrar una tabla de la bbdd

En este ejemplo se eliminará la colección de universidades

```
[ ]: db["universidad"].drop()
print('Se Eliminó la colección universidad exitosamente')

# comprobación de la eliminación de la colección:
print(db.list_collection_names())
```

Se Eliminó la colección universidad exitosamente  
['jugador', 'equipo', 'estadio']

## 9 Conclusiones

1. Las bases de datos no relaciones nos ofrecen mayores beneficios en comparación a una base de datos SQL cuando de manejar grandes volúmenes de datos se tratan (Lo que se puede

comprobar inspeccionando los tiempos de ejecución de las consultas), entre sus principales beneficios encontramos la **escalabilidad**, **flexibilidad** y **agilidad**, las cuales nos permiten acceder a los datos con mayor rapidez.

2. Durante la migración de la base de datos de las estadísticas del deporte beisbol de una base de datos SQL (Postgres) a una base de datos NoSQL (MongoDB) fue muy importante reconocer aquellas tablas principales de las cuales dependían la mayoría de registros, ya que desde allí podríamos identificar las colecciones *pivot* de las cuales partirían las demás relaciones que se podrían adjuntar a esta mediante un diccionario o listas adjuntas. De este análisis se identificó que la colección de jugador era la principal y de esta dependían muchas otras, lo que nos llevo a plantear una estructura de datos que nos permitiera poder actualizar y posteriormente acceder a estos datos mediante consultas, es por esto que se radica en la importancia de plantear una estructura de datos dinámica y robusta que nos permita comprender la relación que existe entre los datos de la colección.