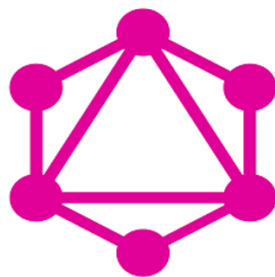




PRACTICA II

NODEJS - GRAPHQL



Curso: Desarrollo de App.Web Avanzado

Docente: Junior Usca Huacasi

Estudiante: Mariam Dalia Apaza Santillana

Especialidad: 5C24 B

2021



Desarrollo

Una empresa requiere una aplicación para manejar sus finanzas. Deberas realizar el backend del proyecto usando nodeJS, mongoDB.

Usando API- REST (Node JS)

- Implementa un CRUD para registrar entidades (personas o empresas juridicas) (2 puntos)

POST: <http://localhost:3000/api/persons>

Request

```
{
  "nombre": "Jimena Vargas Gonzales",
  "dni": "40542155",
  "telefono": "945127845",
  "email": "jvargas@test.com",
  "direccion": "Av.Peru 304"
}
```

Response

```
{
  "message": "Se registro la nueva Persona con DNI:40542155",
  "data": {
    "id": "60b72cca3e92ab453c10bbaa",
    "nombre": "Jimena Vargas Gonzales",
    "dni": "40542155",
    "telefono": "945127845",
    "email": "jvargas@test.com",
    "direccion": "Av.Peru 304"
  }
}
```

- Implementa un CRUD para registrar cuentas cupo (similar a una cuenta de banco).

POST: <http://localhost:3000/api/accounts>

Request

```
{
  "dni": "40542155",
  "nro_cuenta": "1240542155",
  "saldo": "0.00"
}
```

Response

```
{
  "message": "Se regitstro la Cuenta con id:60b72cca3e92ab453c10bbab",
  "data": {
    "id": "60b72e643e92ab453c10bbab",
    "id_person": "60b72cca3e92ab453c10bbaa",
    "nro_cuenta": "1240542155",
    "saldo": 0,
    "tipo": "Cuenta de Ahorros",
    "fecha_creacion": "2021-06-02T07:08:20.370Z"
  }
}
```

- Cada entidad puede abrir multiples cuenta cupo (2 puntos)

POST: <http://localhost:3000/api/persons>

Request

```
{
  "dni": "40542155",
  "nro_cuenta": "1240542156",
  "saldo": "0.00"
}
```

Response

```
{
  "message": "Se regitstro la Cuenta con id:60b72cca3e92ab453c10bbaa",
  "data": {
    "id": "60b72eae3e92ab453c10bbac",
    "id_person": "60b72cca3e92ab453c10bbaa",
    "nro_cuenta": "1240542156",
    "saldo": 0,
    "tipo": "Cuenta de Ahorros",
    "fecha_creacion": "2021-06-02T07:09:34.669Z"
  }
}
```



- Implementa un api para abonar y desembolsar dinero en una cuenta cupo (2 puntos)
POST: <http://localhost:3000/api/transactions/deposit>

Request

```
{
  "dni": "40542155",
  "nro_cuenta": "1240542156",
  "cantidad": "1000",
  "description": "Deposito Mayo"
}
```

Response

```
{
  "message": "Se deposito :1000 a la cuenta N°1240542156",
  "data": {
    "id": "60b72fc13e92ab453c10bbad",
    "id_account": "60b72eae3e92ab453c10bbac",
    "token": "823b32c2a8c127b3573e9048dd1f3ea3a11e83f6",
    "cantidad": 1000,
    "description": "Deposito Mayo",
    "tipo_transaction": "deposito",
    "fecha_trans": "2021-06-02T07:14:09.416Z"
  }
}
```

POST: <http://localhost:3000/api/transactions/retire>

Request

```
{
  "dni": "40542155",
  "nro_cuenta": "1240542156",
  "cantidad": "500",
  "description": "Retiro Prestamo"
}
```

Response

```
{
  "message": "Se retiro :500 de la cuenta N°1240542156",
  "data": {
    "id": "60b730913e92ab453c10bbae",
    "id_account": "60b72eae3e92ab453c10bbac",
    "token": "f41c0b6b8019466698581ec3dcc9b635ec22cf69",
    "cantidad": 500,
    "description": "Retiro Prestamo",
    "tipo_transaction": "retiro",
    "fecha_trans": "2021-06-02T07:17:37.528Z"
  }
}
```

- Implementa un api para transferir dinero de una cuenta a otra. (3 puntos)
POST: <http://localhost:3000/api/transactions/transfer>

Request

```
{
  "dni": "40542155",
  "nro_cuenta_origen": "1240542156",
  "nro_cuenta_destino": "1240542155",
  "cantidad": "90",
  "description": "Transferencia Pago Curso"
}
```

Response

```
{
  "message": "Se hizo la transferencia a la cuenta N°1240542155",
  "data": {
    "id": "60b731863e92ab453c10bbb0",
    "id_account": "60b72e643e92ab453c10bbab",
    "nro_cuenta_origen": "1240542156",
    "nro_cuenta_destino": "1240542155",
    "token": "baf00375b7e96128cebdbf27fdecf4767089f835",
    "cantidad": 90,
    "description": "Transferencia Pago Curso",
    "tipo_transaction": "transferencia",
    "fecha_trans": "2021-06-02T07:21:42.494Z"
  }
}
```

- Cada entidad solo puede mover dinero de alguna de sus propias cuentas cupo a otra cuenta cupo.

```
1 {
2   "dni": "40542155",
3   "nro_cuenta_origen": "1240542156",
4   "nro_cuenta_destino": "2234567869",
5   "cantidad": "20",
6   "description": "Transferencia Invalida"
7 }
```

Body Cookies Headers (16) Test Results Status: 400 Bad Request

Pretty Raw Preview Visualize JSON ↺

```
1 {
2   "message": "Las cuentas no pertenecen al mismo usuario. Intentelo Nuevamente"
3 }
```

- Si la cuenta no tiene saldo suficiente el movimiento de dinero no debe realizarse

```
1 {
2   "dni": "40542155",
3   "nro_cuenta_origen": "1240542156",
4   "nro_cuenta_destino": "1240542155",
5   "cantidad": "90000",
6   "description": "Transferencia Pago Curso"
7 }
```

Body Cookies Headers (16) Test Results Status: 400 Bad Request

Pretty Raw Preview Visualize JSON ↺

```
1 {
2   "message": "El saldo es insuficiente. Solo tiene $410. Le falta $89590"
3 }
```

- Implementa un api para que cada entidad pueda ver todos los movimientos que fueron realizados en sus cuentas cupo (2 puntos)

GET: <http://localhost:3000/api/transactions/40542155>

```
← → ↺ 🏠 localhost:3000/api/transactions/40542155
{
  message: "Jimena Vargas Gonzales tiene 3 cuentas y sus transferencias son:",
  cuentas: [
    {
      id_account: "60b72e643e92ab453c10bbab"
    },
    {
      id_account: "60b72eae3e92ab453c10bbac"
    },
    {
      id_account: "60b731573e92ab453c10bbaf"
    }
  ]
},
```

```

    'data': [
      {
        _id: "60b731863e92ab453c10bbb0",
        id_account: "60b72e643e92ab453c10bbab",
        nro_cuenta_origen: "1240542156",
        nro_cuenta_destino: "1240542155",
        token: "baf00375b7e96128cebdbf27fdecf4767089f835",
        cantidad: 90,
        description: "Transferencia Pago Curso",
        tipo_transaction: "transferencia",
        fecha_trans: "2021-06-02T07:21:42.494z"
      },
      {
        _id: "60b731df3e92ab453c10bbb1",
        id_account: "60b72e643e92ab453c10bbab",
        nro_cuenta_origen: "1240542156",
        nro_cuenta_destino: "1240542155",
        token: "dc6e3f08a4fbbcae36c26ce5f0af527d776efc66",
        cantidad: 90000,
        description: "Transferencia Pago Curso",
        tipo_transaction: "transferencia",
        fecha_trans: "2021-06-02T07:23:11.759z"
      }
    ],
    'data': [
      {
        _id: "60b72fc13e92ab453c10bbad",
        id_account: "60b72eae3e92ab453c10bbac",
        token: "823b32c2a8c127b3573e9048dd1f3ea3a11e83f6",
        cantidad: 1000,
        description: "Deposito Mayo",
        tipo_transaction: "deposito",
        fecha_trans: "2021-06-02T07:14:09.416z"
      }
    ]
  }
}

```

- Implementa un api para que cada entidad pueda obtener el saldo de cada cuenta cupo (2 puntos)

GET: <http://localhost:3000/api/accounts/40542159>

```

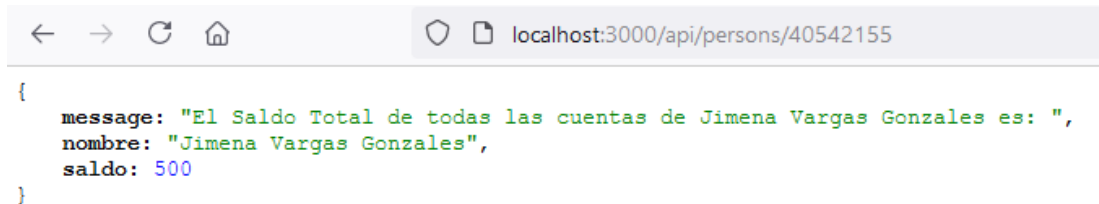
localhost:3000/api/accounts/40542159

{
  message: "Juan Zeballos Vizcarra tiene 2 cuentas:",
  data: [
    {
      _id: "60b7205b6095700644b4843f",
      id_person: "60b720456095700644b4843e",
      nro_cuenta: "2234567869",
      saldo: 550,
      tipo: "Cuenta de Ahorros",
      fecha_creacion: "2021-06-02T06:08:27.243z"
    },
    {
      _id: "60b735203e92ab453c10bbb3",
      id_person: "60b720456095700644b4843e",
      nro_cuenta: "1240542157",
      saldo: 10,
      tipo: "Cuenta de Ahorros",
      fecha_creacion: "2021-06-02T07:37:04.363z"
    }
  ]
}

```

- Implementa un api para que cada entidad pueda obtener el saldo total de todas sus cuentas (2 puntos)

GET: <http://localhost:3000/api/persons/40542155>



```
{
  message: "El Saldo Total de todas las cuentas de Jimena Vargas Gonzales es: ",
  nombre: "Jimena Vargas Gonzales",
  saldo: 500
}
```

- Llena la base de datos con algunas entidades, cuentas y abónales dinero. (2 puntos)

DAWA-PRACTICA2

DATABASE SIZE: 3.76KB INDEX SIZE: 108KB TOTAL COLLECTIONS: 3 [CREATE COLLECTION](#)

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
accounts	7	1.07KB	156B	1	36KB	36KB
persons	3	476B	159B	1	36KB	36KB
transactions	9	2.23KB	254B	1	36KB	36KB

GRAPH QL (Proxy)

Implementa un proyecto en graphQL que sirve de proxy para las consultas:

Schema

```
type Query {
  getPersons: [Person!]!
  getAccounts(dni: String!): [Account!]!
  getAccountsTotal(dni: String!): Int!
  getAccountsbyID(dni: String!, nroAccount: String!): [TransactionsbyAc
}

type Person {
  _id: String!
  nombre: String!
  dni: String!
  telefono: Int!
  email: String!
  direccion: String!
}
```

```

type Transaction {
  id: String!
  id_account: String!
  nro_cuenta_origen: String
  nro_cuenta_destino: String
  token: String!
  cantidad: Int!
  description: String!
  tipo_transaction: String!
  fecha_trans: String!
}

type Account {
  _id: String!
  id_person: String!
  nro_cuenta: String!
  saldo: Int!
  tipo: String!
  fecha_creacion: String!
}

```

```

type Mutation {
  addTransaction(
    dni: String!
    nro_cuenta_origen: String!
    nro_cuenta_destino: String!
    cantidad: String!
    description: String!
  ): Transaction
}

enum CacheControlScope {
  PUBLIC
  PRIVATE
}

# The `Upload` scalar type represents a
scalar Upload

```

- Las queries que se deben poder hacer son:
- Obtener saldo de una entidad (la suma del saldo de todas las cuenta cupo de esa entidad) (2 puntos)

```

1 {
2   getAccountsTotal(dni:"40542159")
3 }

```

```

{
  "data": {
    "getAccountsTotal": 560
  }
}

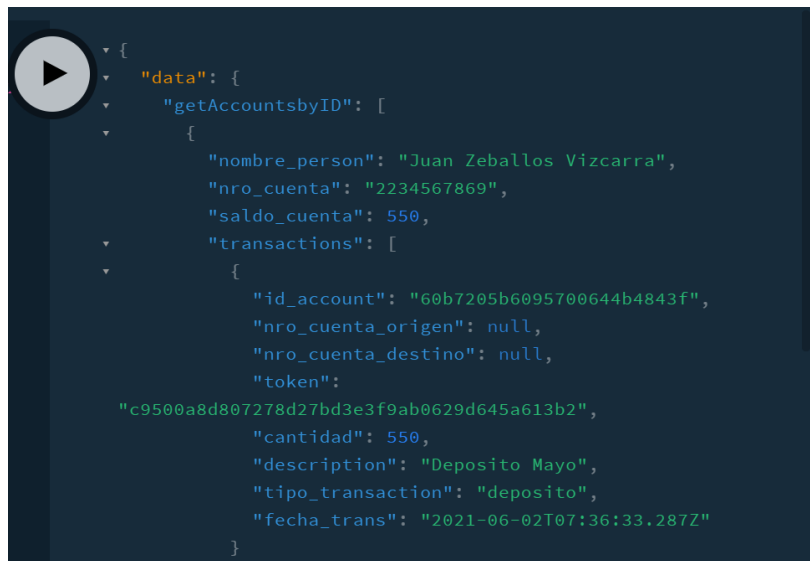
```

- Obtener cuentas cupo. En esta query por cada cuenta cupo debo poder traer por lo menos: (3 puntos)
 - - el nombre
 - - a quien le pertenece
 - - los movimientos realizados (origen, destino, monto)
 - - el saldo de la cuenta

```

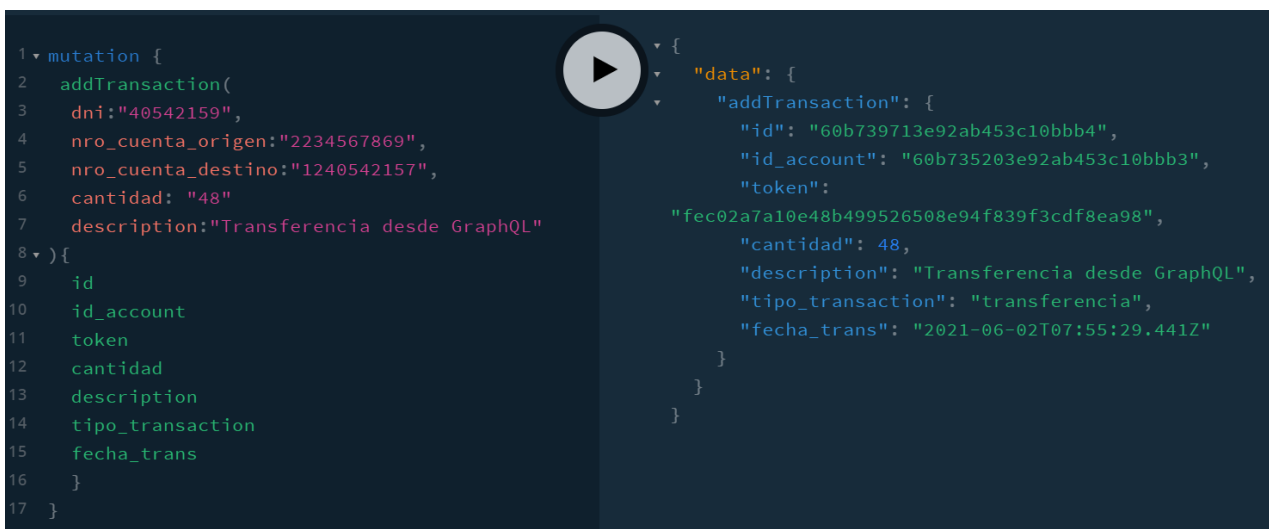
{
  getAccountsbyID(dni:"40542159",nroAccount:"2234567869"){
    nombre_person
    nro_cuenta
    saldo_cuenta
    transactions{
      id_account
      nro_cuenta_origen
      nro_cuenta_destino
      token
      cantidad
      description
      tipo_transaction
      fecha_trans
    }
  }
}

```



```
1 {
2   "data": {
3     "getAccountsbyID": [
4       {
5         "nombre_person": "Juan Zeballos Vizcarra",
6         "nro_cuenta": "2234567869",
7         "saldo_cuenta": 550,
8         "transactions": [
9           {
10            "id_account": "60b7205b6095700644b4843f",
11            "nro_cuenta_origen": null,
12            "nro_cuenta_destino": null,
13            "token":
14              "c9500a8d807278d27bd3e3f9ab0629d645a613b2",
15            "cantidad": 550,
16            "description": "Deposito Mayo",
17            "tipo_transaction": "deposito",
18            "fecha_trans": "2021-06-02T07:36:33.287Z"
19          }
20        ]
21      }
22    ]
23  }
24 }
```

- La mutation que se debe poder hacer es: (2 puntos)
- - transferir dinero

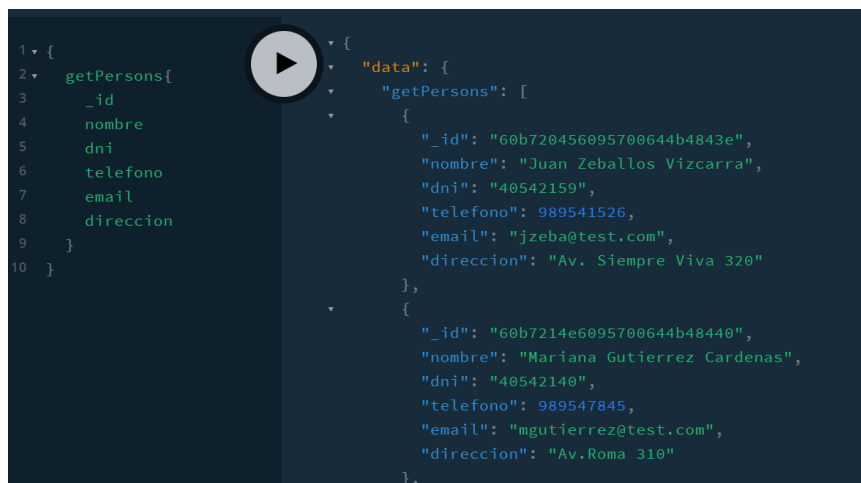


```
1 mutation {
2   addTransaction(
3     dni:"40542159",
4     nro_cuenta_origen:"2234567869",
5     nro_cuenta_destino:"1240542157",
6     cantidad: "48"
7     description:"Transferencia desde GraphQL"
8   ){
9     id
10    id_account
11    token
12    cantidad
13    description
14    tipo_transaction
15    fecha_trans
16  }
17 }
```

```
1 {
2   "data": {
3     "addTransaction": {
4       "id": "60b739713e92ab453c10bbb4",
5       "id_account": "60b735203e92ab453c10bbb3",
6       "token":
7         "fec02a7a10e48b499526508e94f839f3cdf8ea98",
8       "cantidad": 48,
9       "description": "Transferencia desde GraphQL",
10      "tipo_transaction": "transferencia",
11      "fecha_trans": "2021-06-02T07:55:29.441Z"
12    }
13  }
14 }
```

Otras queries:

- `getPersons()`



```
1 {
2   getPersons{
3     _id
4     nombre
5     dni
6     telefono
7     email
8     direccion
9   }
10 }
```

```
1 {
2   "data": {
3     "getPersons": [
4       {
5         "_id": "60b720456095700644b4843e",
6         "nombre": "Juan Zeballos Vizcarra",
7         "dni": "40542159",
8         "telefono": 989541526,
9         "email": "jzeba@test.com",
10        "direccion": "Av. Siempre Viva 320"
11      },
12      {
13        "_id": "60b7214e6095700644b48440",
14        "nombre": "Mariana Gutierrez Cardenas",
15        "dni": "40542140",
16        "telefono": 989547845,
17        "email": "mgutierrez@test.com",
18        "direccion": "Av. Roma 310"
19      }
20    ]
21  }
22 }
```




- `getAccounts()`

```
{
  getAccounts(dni:"40542159"){
    _id
    id_person
    nro_cuenta
    saldo
    tipo
    fecha_creacion
  }
}
```



```
{
  "data": {
    "getAccounts": [
      {
        "_id": "60b7205b6095700644b4843f",
        "id_person": "60b720456095700644b4843e",
        "nro_cuenta": "2234567869",
        "saldo": 502,
        "tipo": "Cuenta de Ahorros",
        "fecha_creacion": "2021-06-02T06:08:27.243Z"
      },
      {
        "_id": "60b735203e92ab453c10bbb3",
        "id_person": "60b720456095700644b4843e",
        "nro_cuenta": "1240542157",
        "saldo": 58,
        "tipo": "Cuenta de Ahorros",
        "fecha_creacion": "2021-06-02T07:37:04.363Z"
      }
    ]
  }
}
```