



# http://localhost:3000/api-cognos/alumnos

Postman interface showing a GET request to `http://localhost:3000/api-cognos/alumnos`. The response is a JSON array of two student objects.

Request Method: GET  
URL: `http://localhost:3000/api-cognos/alumnos`

Response Status: 200 OK  
Time: 204 ms  
Size: 650 B

Response Body (JSON):

```
[
  {
    "id_alumno": 1,
    "nombre": "jorge",
    "apellido": "ruiz",
    "email": "jorge@cognos.com",
    "doc_identidad": 123456,
    "fecha_creacion": "2019-02-26T13:18:10.466Z"
  },
  {
    "id_alumno": 2,
    "nombre": "maria",
    "apellido": "castro",
    "email": "maria@cognos.com",
    "doc_identidad": 112233,
    "fecha_creacion": "2019-02-26T13:18:10.466Z"
  }
]
```

The screenshot displays the Postman application interface. At the top, the title bar reads "Postman". Below it is a menu bar with "File", "Edit", "View", and "Help". A toolbar contains buttons for "New", "Import", "Runner", "My Workspace", "Invite", and "Upgrade". The main workspace shows a collection of requests, with the selected one being a POST request to "http://localhost:3000/api-cognos/alumnos". The request body is a JSON object: 

```
{  "nombre": "Roberto",  "apellido": "Zanbrana",  "email": "rogertoe@cognos.com",  "doc_identidad": 123456}
```

. The response tab is active, showing a status of "200 OK", a time of "174 ms", and a size of "225 B". The response body is a JSON object: 

```
{  "response": 4}
```

. The bottom status bar includes icons for "Learn", "Build", and "Browse".

```
C:\WINDOWS\system32\cmd.exe

D:\marcelomos\laboratorios>mkdir api-postgres-node-express

D:\marcelomos\laboratorios>cd api-postgres-node-express

D:\marcelomos\laboratorios\api-postgres-node-express>npm init --yes
Wrote to D:\marcelomos\laboratorios\api-postgres-node-express\package.json:

{
  "name": "api-postgres-node-express",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

D:\marcelomos\laboratorios\api-postgres-node-express>npm i express pg
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN api-postgres-node-express@1.0.0 No description
npm WARN api-postgres-node-express@1.0.0 No repository field.

+ express@4.16.4
+ pg@7.8.1
added 64 packages from 44 contributors and audited 137 packages in 20.479s
found 0 vulnerabilities

D:\marcelomos\laboratorios\api-postgres-node-express>
```

```
CREATE TABLE public.alumno (  
  id_alumno serial NOT NULL,  
  nombre varchar(50) NULL,  
  apellido varchar(50) NULL,  
  email varchar(50) NULL,  
  doc_identidad int4 NULL,  
  fecha_creacion timestamp NULL DEFAULT now(),  
  CONSTRAINT alumno_pkey PRIMARY KEY (id_alumno)  
);
```

/src/db.js

```
const Pool = require("pg").Pool;  
const pool = new Pool({  
  user: "postgres",  
  host: "192.168.4.22",  
  database: "cognosdb",  
  password: "123456",  
  port: 5432  
});  
  
module.exports = pool;
```

/package.json

```
{
  "name": "api-postgres-node-express",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node src/."
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.16.4",
    "pg": "^7.8.1"
  }
}
```

/src/index.js

```
const express = require('express');
const path = require('path');

// initializations
const app = express();

// settings
app.set('port', process.env.PORT || 3000);

// middlewares
app.use(express.urlencoded({extended: false}));
app.use(express.json());

// routes
app.use(require('./rutas/alumnos-ruta.js'));

// Start the server
app.listen(app.get('port'), () => {
  console.log("Server escuchando ", app.get('port'))
});
```

/src/rutas/alumnos-rutas.js

```
const { Router } = require('express');
const pool = require('../db');

const router = new Router();

router.get('/api-cognos/alumnos', (req, res) => {
  pool.query(
    "SELECT * FROM alumno ORDER BY id_alumno ASC",
    (error, results) => {
      if (error) {
        throw error;
      }
      res.status(200).json(results.rows);
    }
  );
});

router.post('/api-cognos/alumnos', (req, res) => {
  const { nombre, apellido, email, doc_identidad } = req.body;
  pool.query(
    "INSERT INTO alumno (nombre, apellido, email, doc_identidad) VALUES ($1, $2, $3, $4) RETURNING id_alumno",
    [nombre, apellido, email, doc_identidad],
    (error, results) => {
      if (error) {
        throw error;
      }
      res.status(200).json({ response: results.rows[0].id_alumno });
    }
  );
});
```

```
router.get('/api-cognos/alumnos/:id', (req, res) => {
  const id = parseInt(req.params.id);
  pool.query(
    "SELECT * FROM alumno WHERE id_alumno = $1",
    [id],
    (error, results) => {
      if (error) {
        throw error;
      }
      res.status(200).json(results.rows);
    }
  )
});

router.put('/api-cognos/alumnos/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const { nombre, apellido, email, doc_identidad } = req.body;
  pool.query(
    "UPDATE alumno SET nombre = $1, apellido = $2, email = $3, doc_identidad = $4
    WHERE id_alumno = $5",
    [nombre, apellido, email, doc_identidad, id],
    (error, results) => {
      if (error) {
        throw error;
      }
      res.status(200).send(`User modified with ID: ${id}`);
    }
  )
});
```



```
router.delete('/api-cognos/alumnos/:id', (req, res) => {
  const id = parseInt(req.params.id);
  pool.query(
    "DELETE FROM alumno WHERE id_alumno = $1",
    [id],
    (error, results) => {
      if (error) {
        throw error;
      }
      res.status(200).send(`User deleted with ID: ${id}`);
    }
  );
});

module.exports = router;
```