

# /tsc --init

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

D:\marcelomos\laboratorios>npm install -g typescript
C:\Users\marce\AppData\Roaming\npm\tsc -> C:\Users\marce\AppData\Roaming\npm\node_modules\typescript\bin\tsc
C:\Users\marce\AppData\Roaming\npm\tsserver -> C:\Users\marce\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
+ typescript@3.4.2
added 1 package from 1 contributor in 11.128s

D:\marcelomos\laboratorios>
```

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

D:\marcelomos\laboratorios>mkdir api-ts-express-mongo

D:\marcelomos\laboratorios>cd api-ts-express-mongo

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

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

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

D:\marcelomos\laboratorios\api-ts-express-mongo>tsc --init
message TS6071: Successfully created a tsconfig.json file.

D:\marcelomos\laboratorios\api-ts-express-mongo>
```

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

D:\marcelomos\laboratorios\api-ts-express-mongo>npm i express morgan cors compression
helmet mongoose morgan
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN api-ts-express-mongo@1.0.0 No description
npm WARN api-ts-express-mongo@1.0.0 No repository field.

+ compression@1.7.4
+ cors@2.8.5
+ morgan@1.9.1
+ morgan@1.9.1
+ express@4.16.4
+ mongoose@5.5.0
+ helmet@3.16.0
added 98 packages from 58 contributors and audited 206 packages in 19.995s
found 0 vulnerabilities

D:\marcelomos\laboratorios\api-ts-express-mongo>
```

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

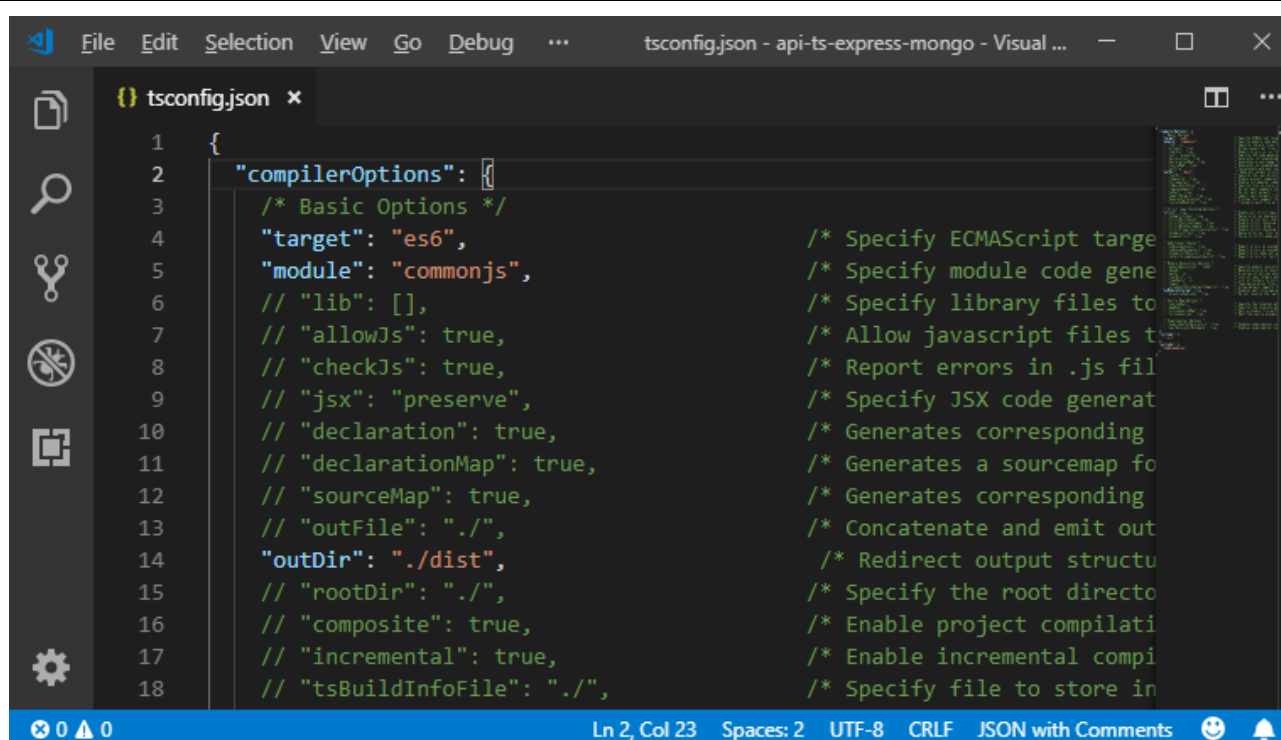
D:\marcelomos\laboratorios\api-ts-express-mongo>npm i @types/express @types/mongoose
@types/node @types/cors @types/morgan @types/helmet @types/compression -D
npm notice save @types/compression is being moved from dependencies to devDependencies
npm notice save @types/cors is being moved from dependencies to devDependencies
npm notice save @types/helmet is being moved from dependencies to devDependencies
npm notice save @types/morgan is being moved from dependencies to devDependencies
npm WARN api-ts-express-mongo@1.0.0 No description
npm WARN api-ts-express-mongo@1.0.0 No repository field.

+ @types/express@4.16.1
+ @types/cors@2.8.4
+ @types/mongoose@5.3.25
+ @types/compression@0.0.36
+ @types/node@11.13.2
+ @types/helmet@0.0.43
+ @types/morgan@1.7.35
updated 7 packages and audited 282 packages in 2.259s
found 0 vulnerabilities
```

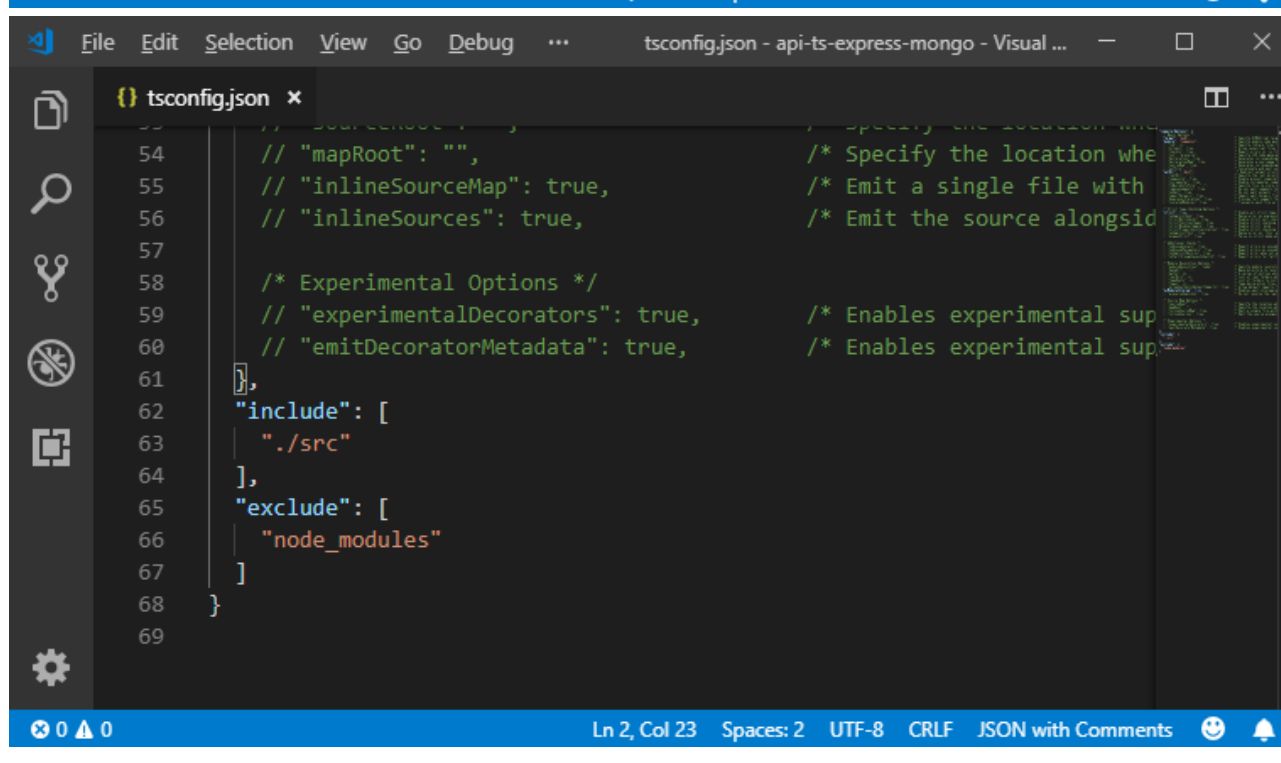
/package.json

```
{
  "name": "api-ts-express-mongo",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node dist/."
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "compression": "^1.7.4",
    "cors": "^2.8.5",
    "express": "^4.16.4",
    "helmet": "^3.16.0",
    "mongoose": "^5.5.0",
    "morgan": "^1.9.1"
  },
  "devDependencies": {
    "@types/compression": "0.0.36",
    "@types/cors": "^2.8.4",
    "@types/express": "^4.16.1",
    "@types/helmet": "0.0.43",
    "@types/mongoose": "^5.3.25",
    "@types/morgan": "^1.7.35",
    "@types/node": "^11.13.2"
  }
}
```

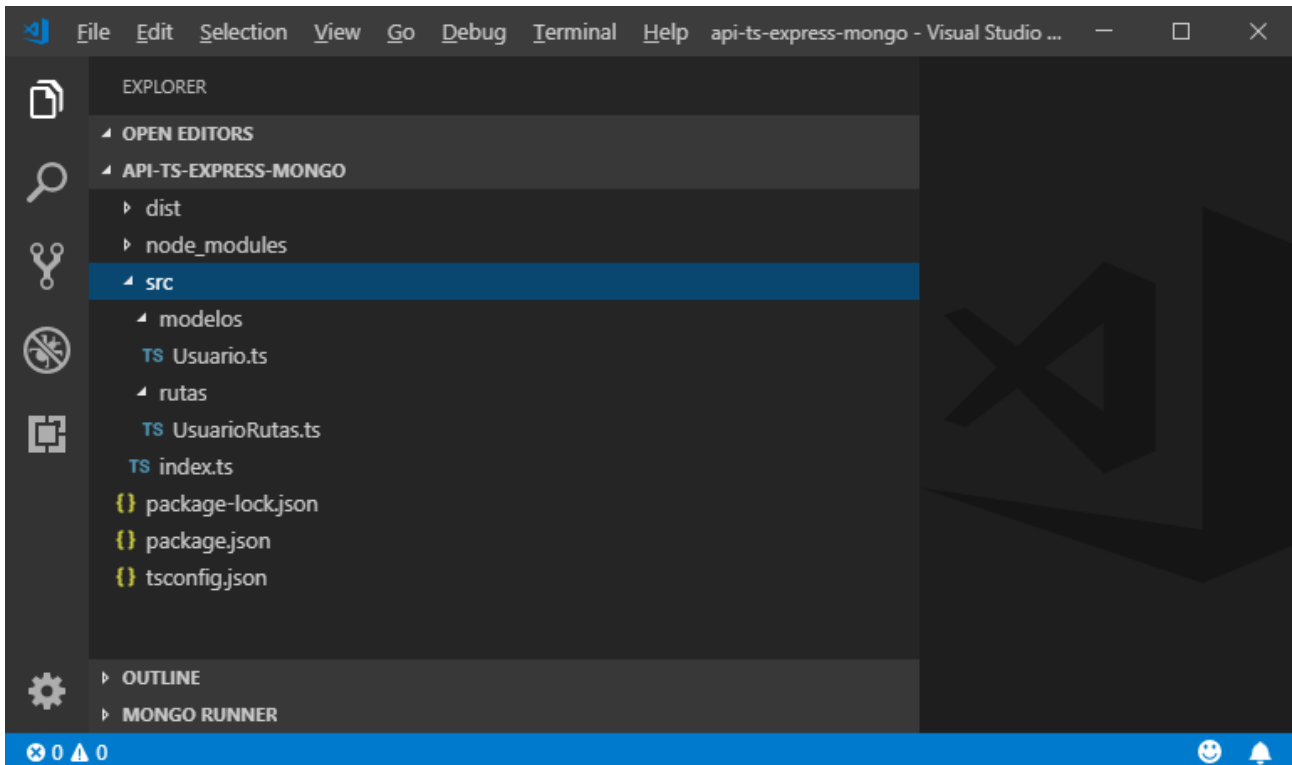
/tsconfig.json



```
1 {
2   "compilerOptions": {
3     /* Basic Options */
4     "target": "es6", /* Specify ECMAScript target
5     "module": "commonjs", /* Specify module code gene
6     // "lib": [], /* Specify library files to
7     // "allowJs": true, /* Allow javascript files t
8     // "checkJs": true, /* Report errors in .js fil
9     // "jsx": "preserve", /* Specify JSX code generat
10    // "declaration": true, /* Generates corresponding
11    // "declarationMap": true, /* Generates a sourcemap fo
12    // "sourceMap": true, /* Generates corresponding
13    // "outFile": "./", /* Concatenate and emit out
14    "outDir": "./dist", /* Redirect output structu
15    // "rootDir": "./", /* Specify the root directo
16    // "composite": true, /* Enable project compilati
17    // "incremental": true, /* Enable incremental compi
18    // "tsBuildInfoFile": "./", /* Specify file to store in
```



```
54    // "mapRoot": "", /* Specify the location whe
55    // "inlineSourceMap": true, /* Emit a single file with
56    // "inlineSources": true, /* Emit the source alongsid
57
58    /* Experimental Options */
59    // "experimentalDecorators": true, /* Enables experimental sup
60    // "emitDecoratorMetadata": true, /* Enables experimental sup
61  },
62  "include": [
63    "./src"
64  ],
65  "exclude": [
66    "node_modules"
67  ]
68 }
69
```



### /src/modelos/Usuario.ts

```
import { Schema, model } from 'mongoose';

const UsuarioSchema = new Schema({
  nombre: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  login: { type: String, required: true, lowercase: true },
  createdAt: { type: Date, default: Date.now() }
});

export default model('Usuario', UsuarioSchema);
```

**/src/rutas/UsuariosRutas.ts**

```
import { Request, Response, NextFunction, Router } from 'express';
import Usuario from '../modelos/Usuario';

class UserRouter {
  router: Router;

  constructor() {
    this.router = Router();
    this.rutas();
  }

  async obtenerUsuarios(req: Request, res: Response) {
    const usuarios = await Usuario.find();
    res.json(usuarios);
  }

  async obtenerUsuario(req: Request, res: Response) {
    const usuario = await Usuario.findById(req.params.id);
    res.json(usuario);
  }

  async crearUsuario(req: Request, res: Response) {
    const nuevoUsuario = new Usuario(req.body);
    await nuevoUsuario.save();
    res.json({ status: 200, nuevoUsuario });
  }

  async actualizarUsuario(req: Request, res: Response) {
    const { id } = req.params;
    const usuario = await Usuario.findByIdAndUpdate(id, req.body, {new: true});
    res.json(usuario);
  }

  async eliminarUsuario(req: Request, res: Response) {
    const { id } = req.params;
    const usuario = await Usuario.findByIdAndRemove(id);
    res.json(usuario);
  }
}
```

```
    rutas() {  
      this.router.get('/', this.obtenerUsuarios);  
      this.router.get('/:id', this.obtenerUsuario);  
      this.router.post('/', this.crearUsuario);  
      this.router.put('/:id', this.actualizarUsuario);  
      this.router.delete('/:id', this.eliminarUsuario);  
    }  
  }  
  
  const userRouter = new UserRouter();  
  export default userRouter.router;
```



/src/index.ts

```
import express from 'express';
import mongoose from 'mongoose';
import morgan from 'morgan';
import helmet from 'helmet';
import compression from 'compression';
import cors from 'cors';

// import routes
import usuarioRutas from './rutas/UsuarioRutas';

// Server Class
class Server {
  public app: express.Application;

  constructor() {
    this.app = express();
    this.config();
    this.routes();
  }

  public config(): void {
    const MONGO_URI = 'mongodb://localhost/cognosdb';
    mongoose.set('useFindAndModify', false);
    mongoose.connect(MONGO_URI || process.env.MONGODB_URL, {
      useNewUrlParser: true,
      useCreateIndex: true
    }).then(db=>{console.log('MongoDB esta conectado')});
    // Settings
    this.app.set('port', process.env.PORT || 3000);
    // middlewares
    this.app.use(morgan('dev'));
    this.app.use(express.json());
    this.app.use(express.urlencoded({extended: false}));
    this.app.use(helmet());
    this.app.use(compression());
    this.app.use(cors());
  }
}
```

```
public routes(): void {
    const router: express.Router = express.Router();

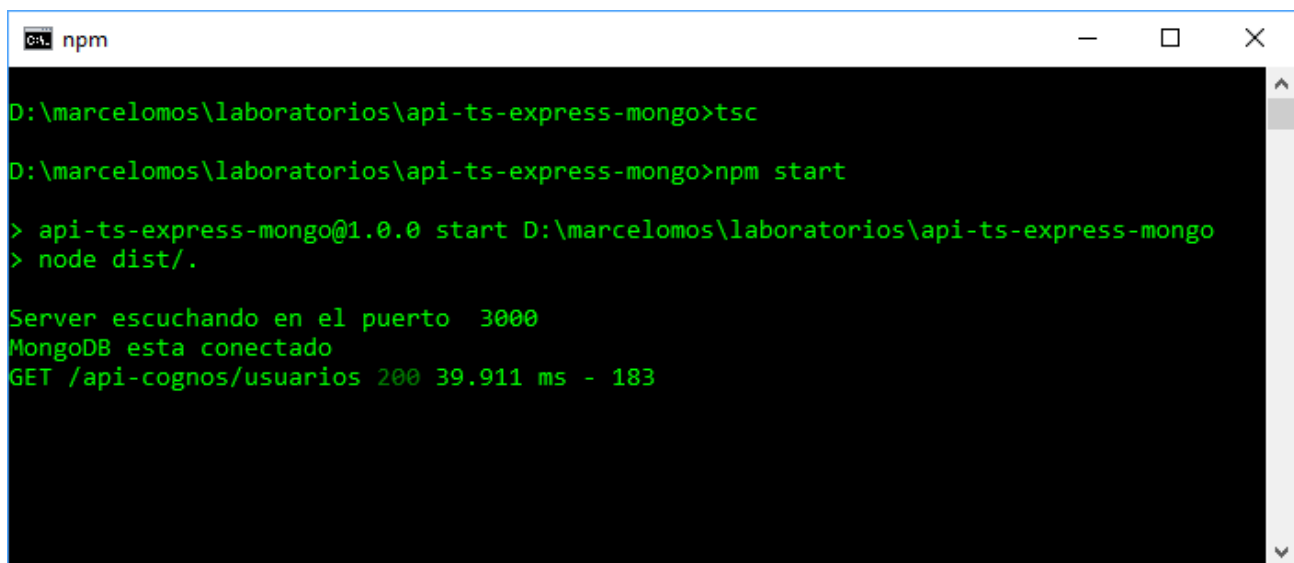
    this.app.use('/api-cognos/usuarios', usuarioRutas);
}

public start(): void {
    this.app.listen(this.app.get('port'), () => {
        console.log('Server escuchando en el puerto ', this.app.get('port'));
    });
}

const server = new Server();
server.start();
```

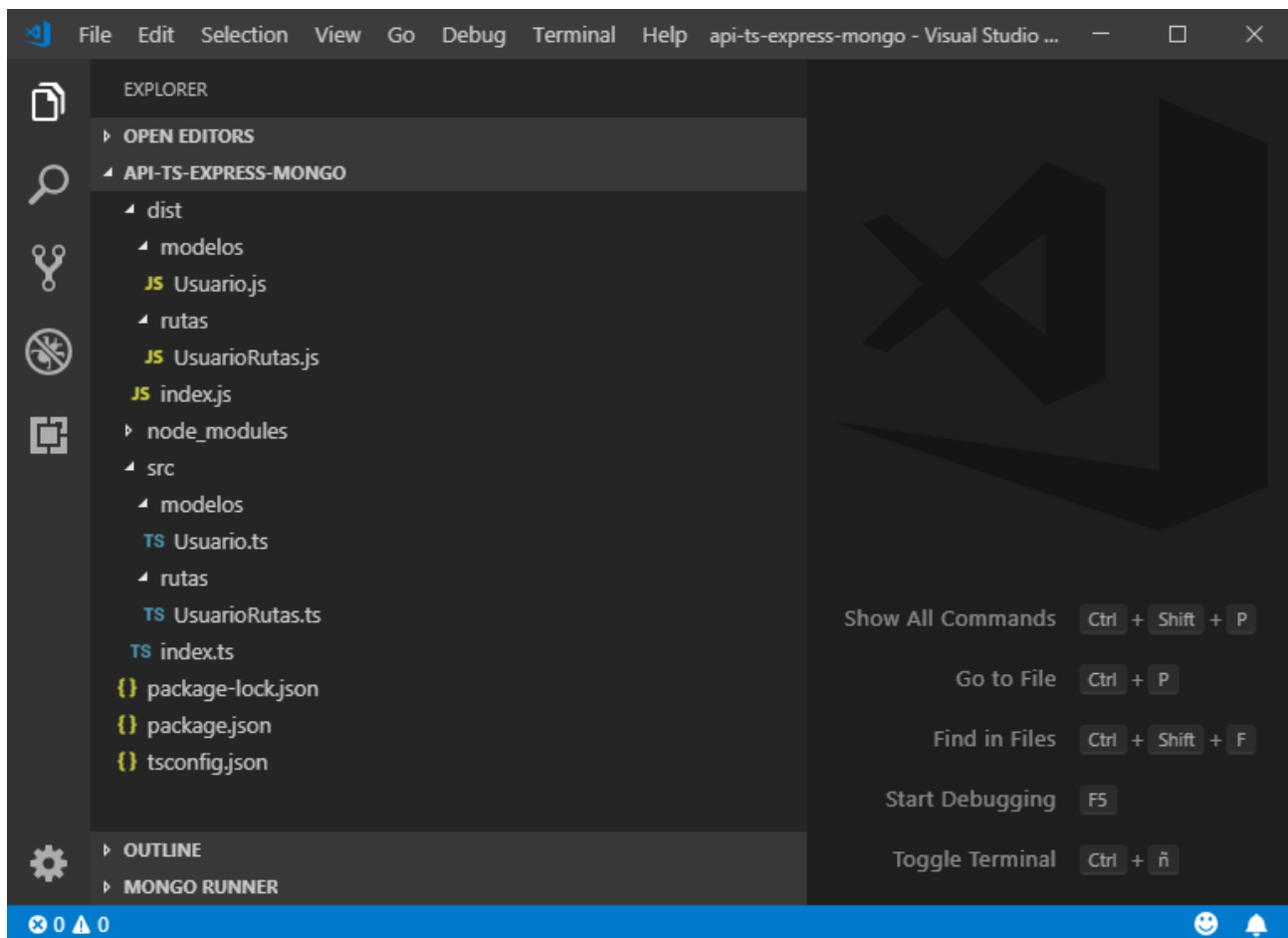
**/tsc**

**/npm start**



```
npm
D:\marcelomos\laboratorios\api-ts-express-mongo>tsc
D:\marcelomos\laboratorios\api-ts-express-mongo>npm start
> api-ts-express-mongo@1.0.0 start D:\marcelomos\laboratorios\api-ts-express-mongo
> node dist/.

Server escuchando en el puerto 3000
MongoDB esta conectado
GET /api-cognos/usuarios 200 39.911 ms - 183
```



The screenshot displays the Postman interface for a REST client. The top bar includes the Postman logo and standard window controls. Below the menu bar, there are buttons for 'New', 'Import', 'Runner', 'My Workspace', and 'Invite'. The main interface shows a selected 'POST' request to the endpoint `http://localhost:3000/api-cognos/usuarios`. The 'Body' tab is active, showing a JSON payload:

```
1 {
2   "nombre": "Marcelo Moscoso",
3   "email": "marcelo@test.com",
4   "password": "123456",
5   "login": "marcelo.moscoso"
6 }
```

The status bar at the bottom indicates a successful response: **Status: 200 OK**, **Time: 65 ms**, and **Size: 644 B**. The response body is shown in the 'Body' tab, displaying the following JSON:

```
1 [
2   {
3     "createdAt": "2019-04-09T14:17:18.109Z",
4     "_id": "5caca6ab84f694cc883bb79",
5     "nombre": "Marcelo Moscoso",
6     "email": "marcelo@test.com",
7     "password": "123456",
8     "login": "marcelo.moscoso",
9   }
10 ]
```

The bottom of the interface features a 'Bootcamp' button and 'Build' and 'Browse' buttons.

