

https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction

npm install express --save

Web-APP



Api-Rest



t

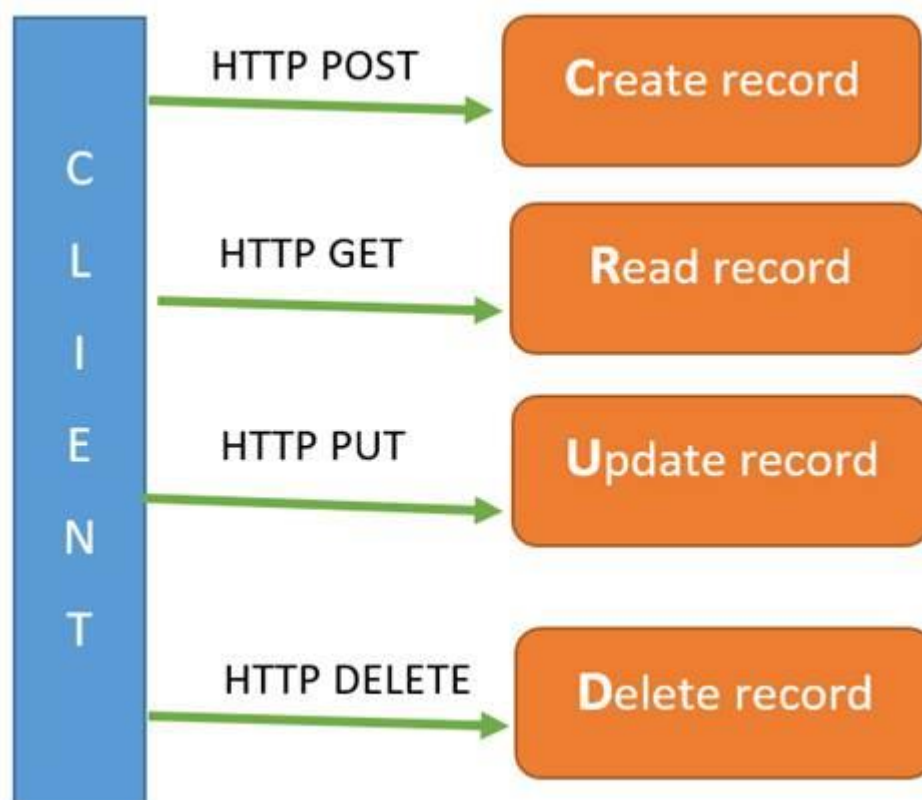
API Rest : express.js node.js

Api (Application programming interface)

Comunicación sobre http

Rest (Representation State Transfer)

Reglas de distribución de datos al cliente



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

C:\Users\NODE JS\laboratorio>mkdir express-morgan-node

C:\Users\NODE JS\laboratorio>npm init --yes
Wrote to C:\Users\NODE JS\laboratorio\package.json:

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

C:\Users\NODE JS\laboratorio>mkdir src
```

/src/modulo-http.js

```
const http = require('http');

const servidor = http.createServer((req, resp) => {

  resp.writeHead(200, { 'Content-Type': 'text/html' });
  resp.write(`
    <!doctype html>
    <html>
      <head></head>
      <body>
        <h1>Sitio en desarrollo</h1>
      </body>
    </html>`);
  resp.end();
});

servidor.listen(3000);
console.log('Servidor web iniciado en el puerto 3000');
```

#npm install express --save

/src/express-app.js

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

app.get('/', (req, res) => {
  res.send('Hola Mundo!');
});

app.listen(3000, () => {
  console.log('Aplicación ejemplo, escuchando el puerto 3000!');
});
```

express.Router()

/src/index.js

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

//rutas
app.get('/', (req, res) => {
  res.send('Hola Mundo!');
});

app.use('/', require('./rutas/pagina.ruta'));

app.use('/api-cognos', require('./rutas/api.rutas'));

//server
app.listen(3000, () => {
  console.log('Aplicación ejemplo, escuchando el puerto 3000!');
});
```

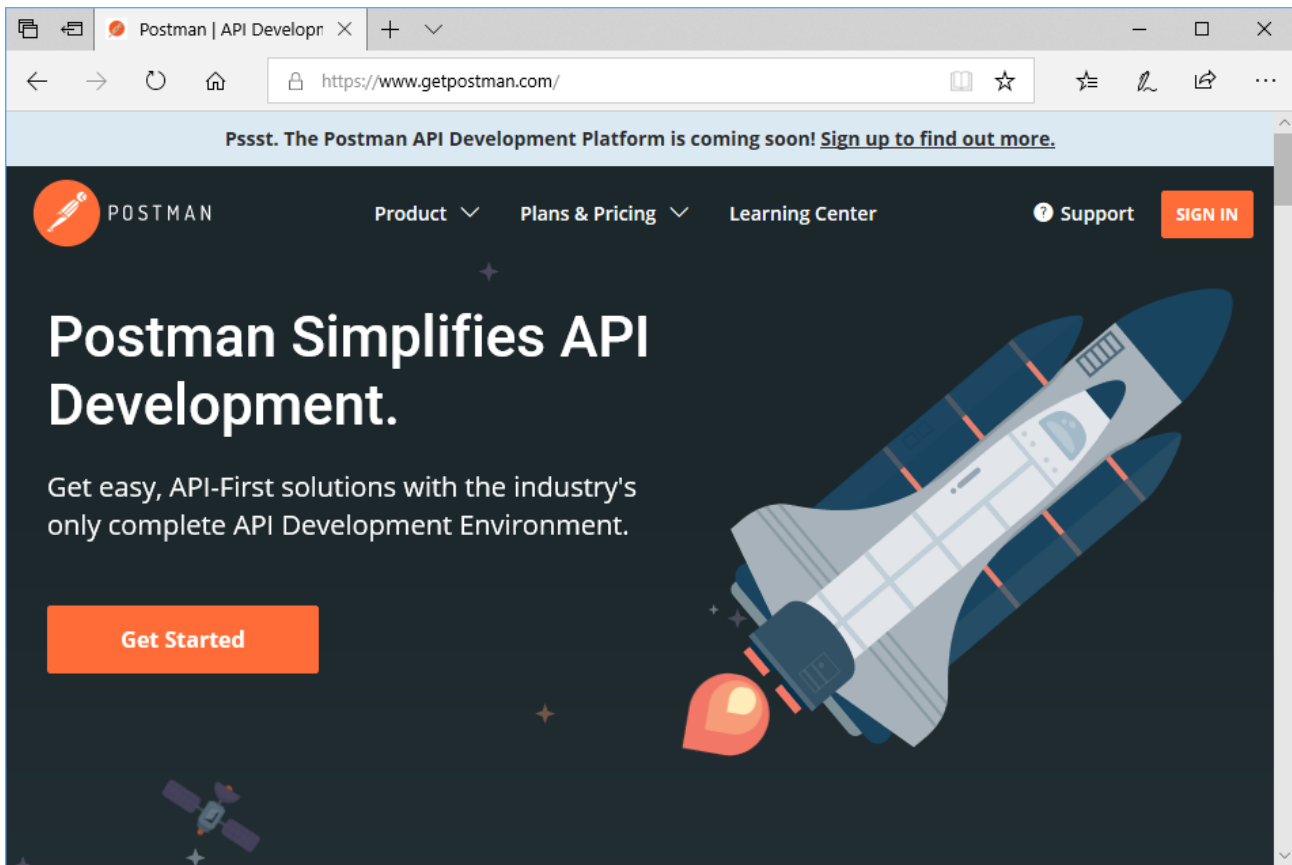
/src/rutas/pagina.ruta.js

```
const express = require('express');
const router = express.Router();

router.get('/inicio', (req, res) => {
  res.send('pagina de inicio');
});

router.get('/acercade', (req, res) => {
  res.send('Acerca de esta pagina');
});

module.exports = router;
```



/src//rutas/api.rutas.js

```
const express = require('express');
const router = express.Router();

router.get('/', (req, res) => {
  res.json({mensaje: 'api-cognos'})
});

router.get('/:id', (req, res) => {
  const id = parseInt(req.params.id);
  res.json({id, mensaje: 'api-cognos'});
});

router.post('/', (req, res) => {
  res.json({mensaje: 'api-cognos', datos: req.body});
});

router.put('/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const {nombres, apellidos} = req.body;
  res.json({mensaje: 'api-cognos', datos: {id, nombres, apellidos}});
});

router.delete('/:id', (req, res) => {
  const id = parseInt(req.params.id);
  res.json({mensaje: 'api-cognos', datos: {id}});
});

module.exports = router;
```


The screenshot displays a REST client interface with two panels. The top panel shows a POST request to `http://localhost:3000/api-cognos/`. The request body is set to 'raw' and 'JSON (application/json)', containing the following JSON:

```
{
  "nombres": "Ana",
  "apellidos": "Gomez"
}
```

The bottom panel shows the response to the GET request `http://localhost:3000/api-cognos/1`. The response is in JSON format and contains:

```
{
  "id": 1,
  "mensaje": "api-cognos"
}
```

Annotations in the image include red boxes and numbers: 1 points to the POST method, 2 to the Body tab, 3 to the raw radio button, 4 to the JSON (application/json) dropdown, and 5 to the request body JSON.

Middleware

/src/index.js

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

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

npm install Morgan

/src/index.js

```
const express = require('express');
const logger = require('morgan');
const app = express();

app.use(logger('dev'));
```

npm install cors

/src/index.js

```
const express = require('express');
const app = express();
const cors = require('cors');

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

express.static('public')

/src/index.js

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

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

app.use('/api-cognos', require('./rutas/api.rutas'));

// static files
app.use(express.static(path.join(__dirname, 'public')));

//server
app.listen(3000, () => {
  console.log('Aplicación ejemplo, escuchando el puerto 3000!');
});
```

/src/public/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <h1>public web cognos</h1>
</body>
</html>
```

Views

npm install hbs

/src/index.js

```
const express = require('express');
const app = express();
const cors = require('cors');
const path = require('path');
const alumnos = require('./data/alumnos.json');

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

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

//views
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'hbs');

app.use('/', (req,res)=>{
  res.render('index', { titulo:'plantilla hbs',datos:alumnos });
});

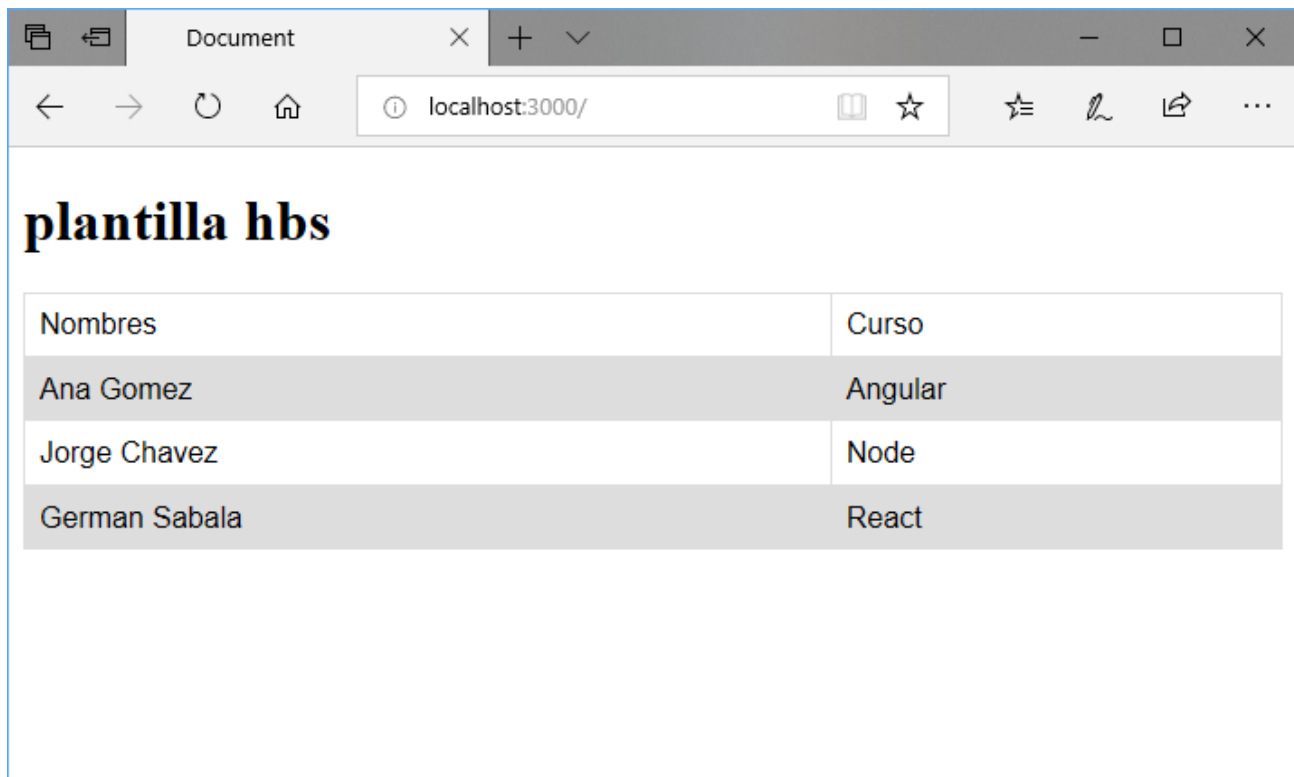
//server
app.listen(app.get('port'), () => {
  console.log(` ${new Date()} : server on port ${app.get('port')}`);
});
```

/src/views/index.hbs

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    table {font-family: arial, sans-serif;
           border-collapse: collapse; width: 100%;}
    td,
    th {border: 1px solid #dddddd;text-align: left;padding: 8px;}
    tr:nth-child(even) {background-color: #dddddd;}
  </style>
</head>
<body>
  <h1>{{titulo}}</h1>
  <table border="1">
    <tr>
      <td>Nombres</td>
      <td>Curso</td>
    </tr>
    {{#each datos }}
    <tr>
      <td>{{this.nombres}}</td>
      <td>{{this.curso}}</td>
    </tr>
    {{/each}}
  </table>
</body>
</html>
```

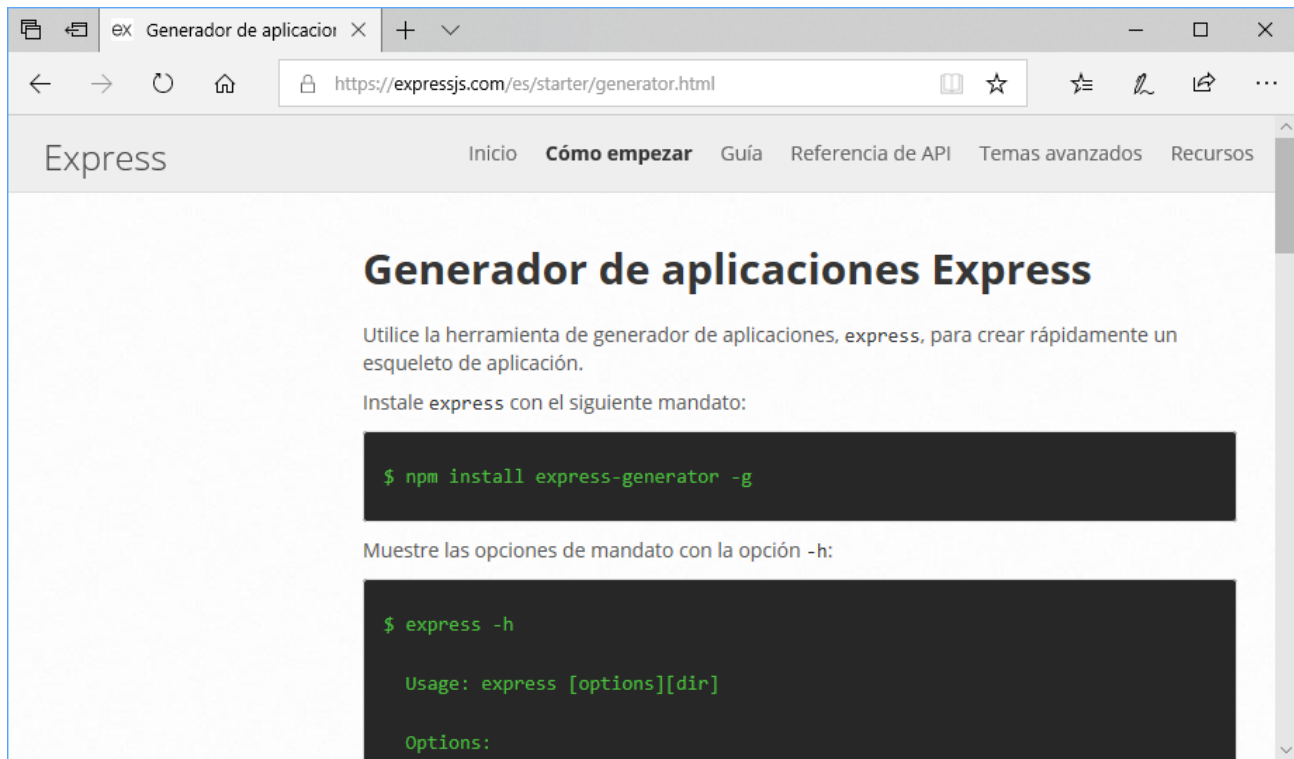
/src/data/alumnos.js

```
[  
  {  
    "nombres": "Ana Gomez",  
    "curso": "Angular"  
  },  
  {  
    "nombres": "Jorge Chavez",  
    "curso": "Node"  
  },  
  {  
    "nombres": "German Sabala",  
    "curso": "React"  
  }  
]
```



The screenshot shows a web browser window with the address bar at localhost:3000/. The page title is "plantilla hbs". Below the title is a table with two columns: "Nombres" and "Curso". The table contains three rows of data, alternating between light and dark gray background colors for each row.

Nombres	Curso
Ana Gomez	Angular
Jorge Chavez	Node
German Sabala	React



<https://expressjs.com/es/starter/generator.html>

#npm install express-generator -g

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

D:\marcelomos\laboratorios> npm install express-generator -g
C:\Users\marce\AppData\Roaming\npm\express -> C:\Users\marce\AppData\Roaming\npm\node_modules\express-generator\bin\express-cli.js
+ express-generator@4.16.0
added 10 packages from 13 contributors in 4.359s

D:\marcelomos\laboratorios> express --view=hbs hbs-generator-express

  create : hbs-generator-express\
  create : hbs-generator-express\public\
  create : hbs-generator-express\public\javascripts\
  create : hbs-generator-express\public\images\
  create : hbs-generator-express\public\stylesheets\
  create : hbs-generator-express\public\stylesheets\style.css
  create : hbs-generator-express\routes\
  create : hbs-generator-express\routes\index.js
  create : hbs-generator-express\routes\users.js
  create : hbs-generator-express\views\
  create : hbs-generator-express\views\error.hbs
  create : hbs-generator-express\views\index.hbs
  create : hbs-generator-express\views\layout.hbs
  create : hbs-generator-express\app.js
  create : hbs-generator-express\package.json
  create : hbs-generator-express\bin\
  create : hbs-generator-express\bin\www

  change directory:
    > cd hbs-generator-express

  install dependencies:
    > npm install

  run the app:
    > SET DEBUG=hbs-generator-express:* & npm start

D:\marcelomos\laboratorios>
```

#express --view=hbs <nombre-app>


```
npm
D:\marcelomos\laboratorios>cd hbs-generator-express
D:\marcelomos\laboratorios\hbs-generator-express>npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
added 64 packages from 81 contributors and audited 153 packages in 8.897s
found 0 vulnerabilities

D:\marcelomos\laboratorios\hbs-generator-express>code .
D:\marcelomos\laboratorios\hbs-generator-express>SET DEBUG=hbs-generator-express:* & npm start
> hbs-generator-express@0.0.0 start D:\marcelomos\laboratorios\hbs-generator-express
> node ./bin/www

hbs-generator-express:server Listening on port 3000 +0ms
GET / 200 85.546 ms - 204
GET /stylesheets/style.css 200 6.283 ms - 111
```

