



<https://nodejs.org>

# Qué es?

Es un framework que permite desarrollar aplicaciones web utilizando JavaScript del lado del servidor

# Por qué?

- Rapidez del Motor v8
- Asíncrono
- Base de datos no relacionales
- El mismo lenguaje en todo el stack

# Simple WebServer Demo

```
var http = require("http");

var handler = function(request, response)
{
    console.log("Holaaa!!!");
    response.end();
};

var server = http.createServer(handler);

server.listen(8080);
```

# Hola Mundo Demo

```
var http = require("http")

var html = "<html><head></head><body>Hola Mundo cruel</body></html>"

http.createServer(function(req, res) {
  res.writeHead(200, { "Content-Type": "text/html" });
  res.write(html);
  res.end();
}).listen(8080);
```

# Routes

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

http.createServer(function (req, res) {
  // set up some routes
  switch(req.url) {
    case '/':
      console.log("[501] " + req.method + " to " + req.url);
      res.writeHead(501, "Not implemented", {'Content-Type': 'text/html'});
      res.end('<html><head><title>501 - Not implemented</title></head><body><h1>Not
implemented!</h1></body></html>');
      break;

    case '/formhandler':
      console.log("[501] " + req.method + " to " + req.url);
      res.writeHead(501, "Not implemented", {'Content-Type': 'text/html'});
      res.end('<html><head><title>501 - Not implemented</title></head><body><h1>Not
implemented!</h1></body></html>');
      break;

    default:
      res.writeHead(404, "Not found", {'Content-Type': 'text/html'});
      res.end('<html><head><title>404 - Not found</title></head><body><h1>Not found.</h1></
body></html>');
      console.log("[404] " + req.method + " to " + req.url);
  };
}).listen(8080);
```

# Express Demo

<http://expressjs.com/>

```
npm install --save express
```

```
var express = require("express");
```

```
var app = express();
```

```
app.get("/", function(req, res) {  
  res.send("Hola mundo");  
});
```

```
app.listen(8070);
```

# Jade

<http://jade-lang.com/>

```
npm install jade --save
```

```
var express = require("express");
```

```
var app = express();
```

```
app.set("view engine", "jade");
```

```
app.get("/", function(req, res) {  
    // should exists 'index.jade' in a folder named  
    'views'  
    res.render("index", { name: "Juan" });  
});
```

```
app.listen(8070);
```



# POST Demo

```
npm install body-parser --save
```

```
var express = require("express");
var bodyParser = require("body-parser");
var app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

app.post('/login', function(request, response) {
  console.log("User: " + request.body.username);
  console.log("Password: " + request.body.password);
  response.send("Logged in :)");
});

app.listen(8070);
```

```
curl -H "Content-Type: application/json" -X POST -d '{"username":"xyz","password":"xyz"}' http://localhost:8070/login
```

# MongoDB

<http://mongoosejs.com/>

```
npm install mongoose --save
```

```
var express = require("express");
var bodyParser = require("body-parser");
var app = express();
var mongoose = require("mongoose");
var Schema = mongoose.Schema;

mongoose.connect("mongodb://localhost/truco");

var playerSchemaJSON = {
  email: String,
  password: String
}

var playerSchema = new Schema(playerSchemaJSON);
var Player = mongoose.model("User", playerSchema);

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

app.get("/players", function(req, res){
  Player.find(function(err, doc){
    console.log(doc);
    res.end("player's list");
  });
});

app.post('/login', function(req, res){
  var player = new Player({email: req.body.email, password: req.body.password});

  player.save(function(){
    res.send("Created User");
  });
});

app.listen(8070);
```

<https://www.mongodb.com/>

```
$ mongod
```