

A photograph of a young boy in a dark shirt and light shorts standing by a pond, holding a fishing rod. A man is crouching in the background. The scene is dimly lit, suggesting dusk or dawn.

**Las oportunidades son como los
amaneceres. Si esperas demasiado
tiempo, las echas de menos**

-William Arthur Ward

alanwalter45@gmail.com | <http://github.com/alanwalter45>



TS



NodeJS con Typescript

TSC JS NODEJS

DESARROLLO WEB

Autor : Alan Walter Machuca Durex

El objetivo de este codelab será mostrar el proceso de desarrollo de una aplicación, resumido en **7 pasos** básicos usando nodeJs y Typescript.

Nota: se pueden realizar los pasos en el orden que deseen

Tiempo : **60min**



HERRAMIENTAS



<https://www.google.com/chrome/>



<https://code.visualstudio.com/>



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



express



PASO1 : INSTALACIÓN DE NODEJS Y TYPESCRIPT

Windows

<https://medium.com/@adsonrocha/como-instalar-o-node-js-no-windows-10-cf2bd460b8a8>

Linux Ubuntu

<https://nodejs.org/en/download/package-manager/>

MacOS

<https://pplware.sapo.pt/tutoriais/node-js-aprenda-instalar-no-macos/>



PASO1 : INSTALACIÓN DE NODEJS Y TYPESCRIPT

Instalando typescript

```
$ npm i -g typescript
```

Instalando git como Sistema de Control de Versiones - cvs

<https://git-scm.com/downloads>

Configuración básica

<https://git-scm.com/book/es/v1/Empezando-Configurando-Git-por-primera-vez>



PASO2 : PROYECTO Y CONFIGURACIÓN

- CREAR U OBTENER APLICACIÓN HTML5

- Descargar template html5,
- Utilizar un framework o librería
- Crear tu propia aplicación, página ó sitio web.

```
// archivo : main.js
function saludar(){
    alert('hello world from nodejs')
}
```

```
/* archivo style.css */
body{
    background: orangered;
    color: yellow;
}
```



PASO2 : PROYECTO Y CONFIGURACIÓN

```
<!-- archivo : index.html -->
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>Page Title</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" media="screen" href="./assets/css/style.css" />
<script src="./assets/js/main.js"></script>
</head>
<body>
<h1>Bienvenido a NODEJ con TypeScript</h1>
<p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Corrupti eveniet maxime odio
ipsa nam, expedita voluptas error tempore, natus maiores minus fugit dolore eaque aliquid!
Dignissimos quam nam asperiores molestias.</p>
<button onclick="saludar()">pulsame!</button>
</body>
</html>
```



PASO2 : PROYECTO Y CONFIGURACIÓN

- Crear directorio de trabajo y establecer como directorio actual
`$ mkdir app-node && cd app-node`
- Inicializar sistema de control de versiones (cvs)
`$ git init`
- Establecer directorio como proyecto (package ó module)
`$ npm init`
- Establecer directorio uso de typescript
`$ tsc --init`



PASO3 : CREAR SERVIDOR WEB

- Archivo de arranque por defecto

- Crear archivo

```
$ touch index.js
```

- Editar archivo main.js

```
$ vim index.js
```

```
// archivo : index.js
```

```
require('http')
```

```
.createServer((req,res)=>res.end('hello nodejs'))
```

```
.listen(7000)
```

- Copia tu app para compartirla como endpoint

```
$ cp -R ../../demo ./app
```



PASO3 : CREAR SERVIDOR WEB

Cargando index.html

```
// archivo : main.js
require('fs').readFile(__dirname+'/index.html',(err,data)=>{
  if(err)throw err
  require('http')
    .createServer((req,res)=>{
      res.setHeader('Content-Type','text/html')
      res.end(data)
    }).listen(7000)
})
```



PASO3 : CREAR SERVIDOR WEB

Cargando index.html

```
// archivo : main.js
require('http')
  .createServer((req,res)=>{
    const data = require('fs').readFileSync(__dirname+'/index.html');
    res.setHeader('Content-Type', 'text/html')
    res.end(data)
  }).listen(7000)
```



PASO3 : CREAR SERVIDOR WEB

```
// archivo : main.js
require('http')
  .createServer((req,res)=>{
    const css = require('fs').readFileSync(__dirname+'/assets/style.css');
    const js = require('fs').readFileSync(__dirname+'/assets/main.js');
    const html = require('fs').readFileSync(__dirname+'/index.html');
    res.setHeader('Content-Type', 'text/html')
    res.write(css)
    res.write(js)
    res.end(html)
  })
  .listen(7000)
```



PASO3 : CREAR SERVIDOR WEB

- Instalando express

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

- Instalando nodemon

```
$ npm i -g nodemon
```

```
// archivo : index.js
const express = require('express')
const app = express()
app.use(express.static(__dirname + '/app'))
app.get('/', (req, res, next) => res.sendFile(__dirname +
'/app/index.html'))
app.listen(7000)
```



PASO4 : PAQUETES Y CARACTERISTICAS EN NODE

- Timers en nodejs (simular carga de datos)

`SetTimeout` , `File System`, `Http`, `Callback` , `Promises` , `Async Await`

- Instalando node-fetch

```
$npm install node-fetch -S
```

- Utilizando Jsonplaceholder

<https://jsonplaceholder.typicode.com/>

```
// archivo : index.js
fetch('https://jsonplaceholder.typicode.com/todos/1')
  .then(response => response.json())
  .then(json => console.log(json))
```



PASO4 : PAQUETES Y CARACTERISTICAS EN NODE

```
// archivo : main.ts
enum Tipo{
  normal,
  especial
}
class Google{
  public constructor(private service:string){
  }
  public getData(tipo:Tipo){
    switch (tipo) {
      case Tipo.normal:
        console.log(`service ${this.service}`)
        break;
      case Tipo.especial:
        console.log(`Hola nuestro servicio es ${this.service}`)
        break;
      default:
        console.log(':( problema...')
        break;
    }
  }
}

new Google('cloud platform').getData(Tipo.especial)
```



PASO4 : PAQUETES Y CARACTERISTICAS EN NODE

```
function getData(){  
  setTimeout(()=>{  
    return {  
      id:1,  
      nombre:'mario moreno cantinflas',  
      estad_civil:'casado'  
    }  
  },2000)  
}  
  
function mostrarInformacion(){  
  let data = getData()  
  console.log(data)  
}  
  
mostrarInformacion()
```



PASO4 : PAQUETES Y CARACTERISTICAS EN NODE

```
function getData(callback:any){
    setTimeout(()=>{
        callback(null,{
            id:1,
            nombre:'mario moreno cantinflas',
            estad_civil:'casado'
        })
    },2000)
}

function mostrarInformacion(){
    getData((err:any,data:any)=>{
        if(err)throw err;
        console.log(data)
    })
}

mostrarInformacion()
```



PASO4 : PAQUETES Y CARACTERISTICAS EN NODE

```
declare var Promise:any

function getData(){
  return new Promise((resolve:any,reject:any)=>{
    setTimeout(()=>{
      resolve({
        id:1,
        nombre:'mario moreno cantinflas',
        estad_civil:'casado'
      })
    },2000)
  })
}

async function mostrarInformacion(){
  let data = await getData()
  console.log(data)
}

mostrarInformacion()
```



PASO5 : TASKS RUNNER

- Instalando gulp

```
$npm install -g gulp
```

```
// archivo : gulpfile.js
const gulp=require('gulp')
gulp.task('mi-tarea',function(){
    console.log('ejecutando mi-tarea')
})

gulp.task('default',['mi-tarea'],function(){
    console.log('ejecutando tarea por defecto')
})
```



PASO6 : PRUEBAS

- Debug VSCODE / Dev Tools Google

Instalando jasmine

```
$ npm install --save jasmine
```

```
$ npm install --global jasmine
```

Instalando karma

```
$ npm i -g karma
```

```
ó
```

```
$ npm I -g karma-cli
```



PASO6 : PRUEBAS

```
// archivo : script.spec.js
describe('mi unidad de testeo',function(){
    beforeEach(function(done){
        console.log('se ejecuto antes de cada caso spec')
        done()
    })
    it('deberia ser el numero verdadero',function(){
        expect(true).toBe(true)
    })
    afterEach(function(done){
        console.log('se ejecuto después de cada caso spec')
        done()
    })
})
```



PASO7 : DESARROLLO Y DEPLIEGUE

- Instalando nodemon

```
$ npm install --global nodemon
```

- Instalando pm2

```
$ npm i --global pm2
```

Conoce algo mas!

- <https://cloud.google.com/nodejs/>
- <https://www.docker.com/>
- <https://kubernetes.io/>



CONCLUSIONES

- Microservicios
- Backend
- Recursos en la plataforma
- Cobertura para javascript
- Trabajo en conjunto con las demás tecnologías.



GRACIAS.

