

# Instituto Tecnológico de Costa Rica



## **Escuela de Ingeniería en Computación Introducción al Desarrollo de Páginas Web**

Apunte de Node.js, Express.js

### **Profesora:**

Ericka Solano Fernández

### **Estudiantes:**

Arlem Gabriel Brenes Aguilar

**Verano, 2020**

## Experiencia de Instalación

Para efectos de la instalación, el instructivo fue lo suficientemente claro para realizar cada uno de los pasos correspondientes, además de que la tecnología no ameritaba realizar exhaustivas configuraciones por tanto la experiencia fue excelente.

## Aspectos Relevantes de la Tecnología

### Node.js y Express.js

La diferencia node js es un entorno de trabajo, es un servidor y express js es un framework. Node Js fue fundado por Ryan Dahi en 2009, express js fue desarrollado por TJ Holowaychuk en 2010. Npm es una herramienta para administrar paquetes, se creo en 2009. Compatibilidad con Windows en el 2001, ahora bien Deno es un runtime para javascript.

Soportado por: Linux, mac, Windows, Smart os, free bsd, open bsd, ibm AIX

**Ventajas:** es enhebrado crea bucles de procesos, crea muchas conexiones con mejor rendimiento, v8 motor que compila código hecho por Google, mejor gestión de paquetes mediante npm, se puede conectar a un navegador, base de datos, MVC. Escalabilidad sencilla, ordenamiento porque puede tener desde el front end hasta el backend se puede usar node js

### Desventajas:

Debe tenerse app compatible con bash, además módulo de compilación incompatibles con windows

## Descripción del Ejercicio

Para efectos del taller el ejercicio consistió en realizar una aplicación sobre administración de empleados, en el cual se pueden agregar empleados y visualizarlos

## Evidencias Visuales

### Código:

- Routes employee.js:

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

```
module.exports = {
```

```

addEmployeePage: (req, res) => {
  res.render('add-employee.ejs',
    { title: "Bienvenido al taller | Agregar un empleado", message: ''
  });
},
addEmployee: (req, res) => {
  if (!req.files) {
    return res.status(400).send("No files were uploaded.");
  }
  let message = '';
  let first_name = req.body.first_name;
  let last_name = req.body.last_name;
  let position = req.body.position;
  let number = req.body.number; let uploadedFile = req.files.image; let
image_name = uploadedFile.name; let fileExtension =
uploadedFile.mimetype.split('/')[1]; let cc = req.body.cc; image_name = cc +
'.' + fileExtension; let userQuery = "SELECT * FROM `empleados` WHERE cc = '"
+ cc + "'"; db.query(userQuery, (err, result) => {
  if (err) { return res.status(500).send(err); } if (result.length >
0) { message = 'El empleado ya existe'; res.render('add-employee.ejs', {
message, title: "Bienvenido al taller | Agregar un empleado" }); } else { //
check the filetype before uploading it
    if (uploadedFile.mimetype === 'image/png' ||
uploadedFile.mimetype === 'image/jpeg' || uploadedFile.mimetype ===
'image/gif') { // upload the file to the /public/assets/img directory

uploadedFile.mv(`C://Users//Alejandro//Downloads//taller-node//public//assets//
img//${image_name}`, (err) => {
      if (err) { return res.status(500).send(err); } // send
the player's details to the database
      let query = "INSERT INTO `empleados` (first_name,
last_name, position, number, image, cc) VALUES ('" + first_name + "', '" +
last_name + "', '" + position + "', '" + number + "', '" + image_name + "', '"
+ cc + "')"; db.query(query, (err, result) => { if (err) { return
res.status(500).send(err); } res.redirect('/'); });
    });
  } else { message = "Invalid File format. Only 'gif', 'jpeg'
and 'png' images are allowed."; res.render('add-employee.ejs', { message,
title: "Bienvenido al taller | Agregar un empleado" }); }
}

```

```

    }
  });
},
editEmployeePage: (req, res) => {
  let employeeId = req.params.id;
  let query = "SELECT * FROM `empleados` WHERE id = '" + employeeId + "'";
  db.query(query, (err, result) => {
    if (err) {
      return res.status(500).send(err);
    }
    res.render('edit-employee.ejs', {
      title: "Edit Employee",
      employee: result[0], message: ''
    });
  });
},
editEmployee: (req, res) => {
  let employeeId = req.params.id;
  let first_name = req.body.first_name;
  let last_name = req.body.last_name;
  let position = req.body.position;
  let number = req.body.number;
  let query = "UPDATE `empleados` SET `first_name` = '" + first_name + "', `last_name` = '" + last_name + "', `position` = '" + position + "', `number` = '" + number + "' WHERE `empleados`.`id` = '" + employeeId + "'";
  db.query(query, (err, result) => {
    if (err) {
      return res.status(500).send(err);
    }
    res.redirect('/');
  });
},
deleteEmployee: (req, res) => {
  let employeeId = req.params.id;
  let getImageQuery = 'SELECT image from `empleados` WHERE id = ' + employeeId + ''';
  let deleteUserQuery = 'DELETE FROM empleados WHERE id = ' + employeeId + ''';
  db.query(getImageQuery, (err, result) => {

```

```

        if (err){
            return res.status(500).send(err);
        }
        let image = result[0].image;

fs.unlink(`C://Users//Alejandro//Downloads//taller-node//public//assets//img//
${image}`, (err) => {
    if (err) {
        return res.status(500).send(err);
    }
    db.query(deleteUserQuery, (err, result) => {
        if (err) {
            return res.status(500).send(err);
        }
        res.redirect('/');
    });
});
});
}
}

```

- **Routes index.js:**

```

module.exports = {
    getHomePage: (req, res) => {
        let query = "SELECT * FROM `empleados` ORDER BY id ASC"; // query
        // execute query
        db.query(query, (err, result) => {
            if (err) {
                res.redirect('/');
            }
            res.render('index.ejs', {
                title: "Bienvenido al taller | Agregar un empleado" ,
                employees: result
            });
        });
    }
};

```

- **Views add-employee.ejs:**

```

<%- include ('partials/header.ejs'); -%>
<div class="container">
  <% if (message != '') { %>
    <p class="text-center text-danger"><%= message %></p>
  <% } %>
  <form class="add-employee-form" action="" method="post"
enctype="multipart/form-data">
    <div class="form-row">
      <div class="form-group col-md-4">
        <input type="text" class="form-control" name="first_name"
id="first-name" placeholder="First Name" required>
      </div>
      <div class="form-group col-md-4">
        <input type="text" class="form-control" name="last_name"
id="last-name" placeholder="Last Name" required>
      </div>
      <div class="form-group col-md-4">
        <input type="text" class="form-control" name="cc" id="cc"
placeholder="cc" required>
      </div>
    </div>
    <div class="form-row">
      <div class="form-group col-md-6">
        <input type="number" class="form-control" name="number"
id="number" placeholder="Number" required>
      </div>
      <div class="form-group col-md-6">
        <select id="position" name="position" class="form-control"
required>
          <option selected disabled>Seleccione el
puesto</option>
          <option>Gerente</option>
          <option>Supervisor</option>
          <option>Programador</option>
          <option>Disenador</option>
          <option>Disenador</option>
        </select>
      </div>
    </div>
    <div class="col-md-12">

```

```

        <label for="employee-img"><b>Employee
Image</b></label><br>
        <input type="file" name="image" id="employee-img" class=""
required>
    </div>
</div>
    <button type="submit" class="btn btn-primary float-right">Agregar
empleado</button>
</form>
</div>
</div>
</body>
</html>

```

- Views edit-employee.ejs:

```

<%- include ('partials/header.ejs'); -%>
<div class="container">
    <% if (message) { %>
        <p class="text-center text-danger"><%= message %></p>
    <% } %>

    <% if (employee) { %>
        <form class="add-employee-form" action="" method="post"
enctype="multipart/form-data">
            <div class="form-row">
                <div class="form-group col-md-4">
                    <label for="first-name">First Name</label>
                    <input type="text" class="form-control" name="first_name"
id="first-name" value="<%= employee.first_name %>" required>
                </div>
                <div class="form-group col-md-4">
                    <label for="last-name">Last Name</label>
                    <input type="text" class="form-control" name="last_name"
id="last-name" value="<%= employee.last_name %>" required>
                </div>
                <div class="form-group col-md-4">
                    <label for="cc">CC</label>
                    <input type="text" class="form-control" name="cc" id="cc"
value="<%= employee.cc %>" required disabled title="Username cannot be
edited.">

```

```

        </div>
    </div>
    <div class="form-row">
        <div class="form-group col-md-6">
            <label for="number">Number</label>
            <input type="number" class="form-control" name="number"
id="number" placeholder="Number" value="<%= employee.number %>" required>
        </div>
        <div class="form-group col-md-6">
            <label for="position">Position</label>
            <select id="position" name="position" class="form-control"
required>
                <option selected disabled>Seleccione el
puesto</option>
                <option>Gerente</option>
                <option>Supervisor</option>
                <option>Programador</option>
                <option>Disenador</option>
                <option>Disenador</option>
            </select>
        </div>
    </div>
    <button type="submit" class="btn btn-success float-right">Update
employee</button>
</form>
<% } else { %>
    <p class="text-center">Employee Not Found. Go <a href="/add">here</a>
to add employees.</p>
<% } %>
</div>
</div>
</body>
</html>

```

- Views index.ejs:

```

<%- include ('partials/header.ejs'); -%>
<div class="table-wrapper">
    <% if (employees.length > 0) {%>
        <table class="table table-hovered">
            <thead class="thead-dark">

```



```

        <tr>
            <th scope="col">ID</th>
            <th scope="col">Image</th>
            <th scope="col">First Name</th>
            <th scope="col">Last Name</th>
            <th scope="col">Position</th>
            <th scope="col">Number</th>
            <th scope="col">CC</th>
            <th scope="col">Action</th>
        </tr>
    </thead>
    <tbody>
        <% employees.forEach((employee, index) => { %>
            <tr>
                <th scope="row"><%= employee.id %></th>
                <td></td>
                <td><%= employee.first_name %></td>
                <td><%= employee.last_name %></td>
                <td><%= employee.position %></td>
                <td><%= employee.number %></td>
                <td><%= employee.cc %></td>
                <td>
                    <a href="/edit/<%= employee.id %>"
target="_blank" rel="noopener" class="btn btn-sm btn-success">Edit</a>
                    <a href="/delete/<%= employee.id %>"
class="btn btn-sm btn-danger">Delete</a>
                </td>
            </tr>
        <% }) %>
    </tbody>
</table>
<% } else { %>
    <p class="text-center">No employees found. Go <a href="/add"
>here</a> to add employees.</p>
    <% } %>
</div>
</div>
</body>

```

</html>

- **ejercicio.js:**

```
const express = require('express');
const fileUpload = require('express-fileupload');
const bodyParser = require('body-parser');
const mysql = require('mysql');
const path = require('path');
const app = express();
const { getHomePage } = require('./routes/index');
const { addEmployeePage, addEmployee, deleteEmployee, editEmployee,
editEmployeePage } = require('./routes/employee');
const port = 5000;
// create connection to database
const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'test'
});
// connect to database
db.connect((err) => {
  if (err) {
    throw err;
  }
  console.log('Connected to database');
});
global.db = db;
// configure middleware
app.set('port', process.env.port || port); // set express to use this port
app.set('views', __dirname + '/views'); // set express to look in this folder
to render our view
app.set('view engine', 'ejs'); // configure template engine
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json()); // parse form data client
app.use(express.static(path.join(__dirname, 'public'))); // configure express
to use public folder
app.use(fileUpload()); // configure fileupload
// routes for the app
app.get('/', getHomePage);
```

```
app.get('/add', addEmployeePage);
app.get('/edit/:id', editEmployeePage);
app.get('/delete/:id', deleteEmployee);
app.post('/add', addEmployee);
app.post('/edit/:id', editEmployee);
// set the app to listen on the port
app.listen(port, () => {
  console.log(`Server running on port: ${port}`);
});
```

**Resultado Obtenido:**

WhatsAppTelegram WebGuide - inPaaS API Docs985.pdfApuntes TypescriptEvaluacionesApuntes Node y Expresslocalhost / 127.0.0.1Bienvenido al taller | Agregarv4-728px-Che-Geo

localhost:5000

EmpleadosAdd a Employee

ID	Image	First Name	Last Name	Position	Number	CC	Action
1		Alejandro	Schmidt	Programador	12345678	115650346	<a href="#">Edit</a> <a href="#">Delete</a>

WhatsAppTelegram WebGuide - inPaaS API Docs985.pdfApuntes Typescriptv4-728px-Che-GeoEvaluacionesApuntes Node y Expresslocalhost / 127.0.0.1/v4-728px-Che-Geo-1Bienvenido al taller | Agregar

localhost:5000/add

EmpleadosAdd a Employee

Alejandro

Schmidt

115650346

12345678

Programador

Employee Image

Browse...

v4-728px-Che-Geo-Images-Step-1.jpg

Agregar empleado

Empleados [Add a Employee](#)

First Name	Last Name	CC
<input type="text" value="Alejandro"/>	<input type="text" value="Schmidt"/>	<input type="text" value="115650346"/>
Number	Position	
<input type="text" value="12345678"/>	<input type="text" value="Seleccione el puesto"/>	

[Update employee](#)