

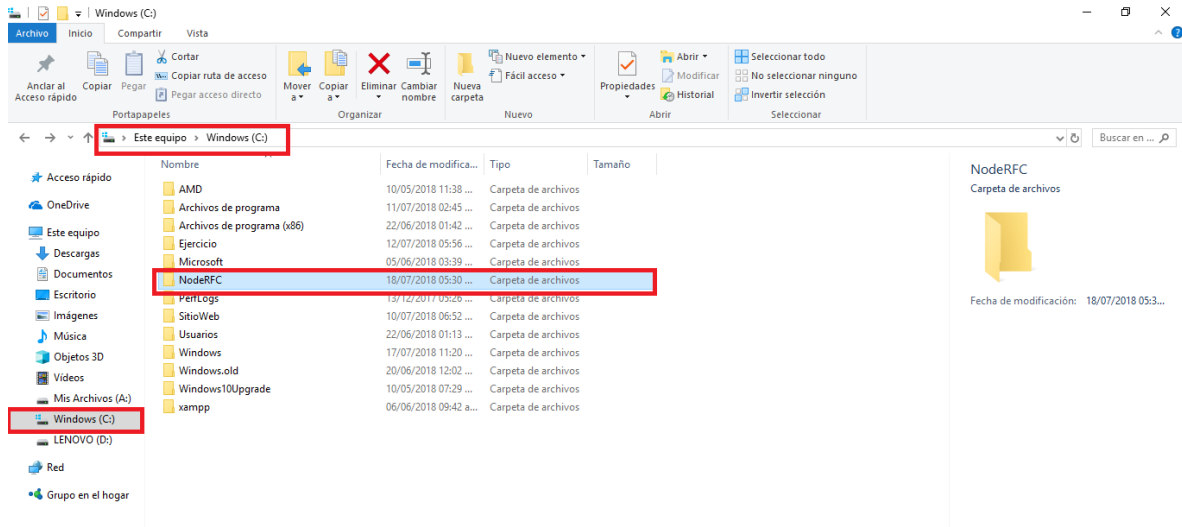


# CREAR UNA APLICACIÓN RFC CON NODE.JS Y EXPRESS

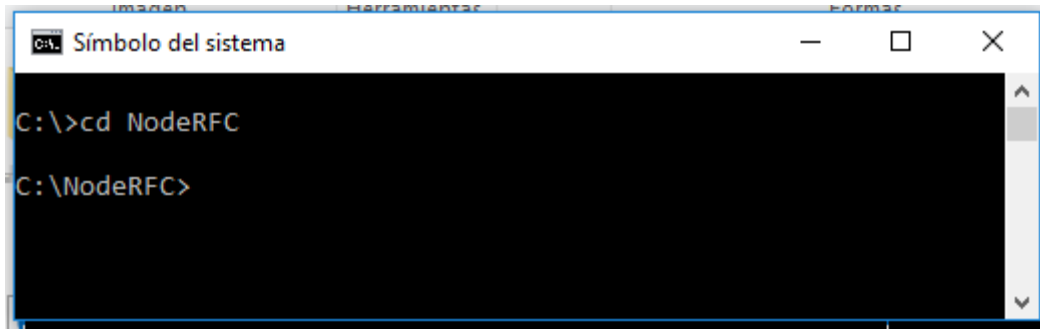
Carrillo Bejerano, Naomi.

2018

1. Crearemos una carpeta en el disco local que utilice (para este ejemplo, en el disco local con el nombre del proyecto (en este caso, NodeRFC).



2. Abriremos la línea de comandos y escribiremos lo siguiente:



3. Instalaremos Express con la siguiente instrucción:

**npm install express**

```
Símbolo del sistema
C:\NodeRFC>npm install express
npm WARN registry Using stale data from https://registry.npmjs.org/ because the host is inaccessible -- are you offline?
npm WARN registry Using stale package data from https://registry.npmjs.org/ due to a request error during revalidation.
npm WARN saveError ENOENT: no such file or directory, open 'C:\NodeRFC\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\NodeRFC\package.json'
npm WARN NodeRFC No description
npm WARN NodeRFC No repository field.
npm WARN NodeRFC No README data
npm WARN NodeRFC No license field.

+ express@4.16.3
added 50 packages in 61.818s
C:\NodeRFC>
```

4. Creamos la siguiente instrucción, esto con el objetivo de crear un proyecto con Express

**npm install express-generator -g**

```
Símbolo del sistema
C:\NodeRFC>npm install express-generator -g
C:\Users\naomi\AppData\Roaming\npm\express -> C:\Users\naomi\AppData\Roaming\npm\node_modules\express-generator\bin\express-cli.js
+ express-generator@4.16.0
updated 1 package in 1.972s
C:\NodeRFC>
```

5. Como siguiente paso crearemos toda la estructura del sitio web con la siguiente instrucción:

**express --view=ejs**

```
Símbolo del sistema
C:\NodeRFC>express --view=ejs
destination is not empty, continue? [y/N] y

create : public\
create : public\javascripts\
create : public\images\
create : public\stylesheets\
create : public\stylesheets\style.css
create : routes\
create : routes\index.js
create : routes\users.js
create : views\
create : views\error.ejs
create : views\index.ejs
create : app.js
create : package.json
create : bin\
create : bin\www

install dependencies:
> npm install

run the app:
> SET DEBUG=noderfc:* & npm start

C:\NodeRFC>
```

6. A continuación, vamos a intentar correr la aplicación escribiendo el siguiente comando:

**npm start**

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

run the app:
> SET DEBUG=sitioweb:* & npm start

C:\NodeRFC>npm start
> sitioweb@0.0.0 start C:\NodeRFC>
> node ./bin/www

module.js:540
    throw err;
    ^

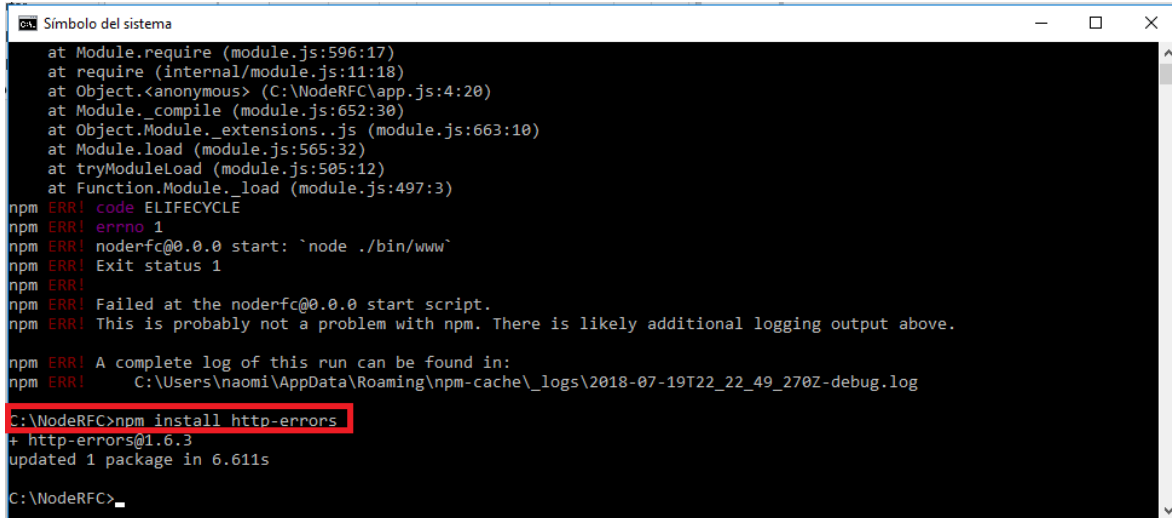
Error: Cannot find module 'http-errors'
    at Function.Module._resolveFilename (module.js:538:15)
    at Function.Module._load (module.js:468:25)
    at Module.require (module.js:587:17)
    at require (internal/module.js:11:18)
    at Object.<anonymous> (c:\sitioweb\app.js:1:81)
    at Module._compile (module.js:643:30)
    at Object.Module._extensions..js (module.js:654:10)
    at Module.load (module.js:556:32)
    at tryModuleLoad (module.js:499:12)
    at Function.Module._load (module.js:491:3)
npm ERR! code ELIFECYCLE
npm ERR! errno 1
npm ERR! sitioweb@0.0.0 start: `node ./bin/www`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the sitioweb@0.0.0 start script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.
npm WARN Local package.json exists, but node_modules missing, did you mean to install?

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\PC\AppData\Roaming\npm-cache\_logs\2018-07-04T05_00_00_329Z-debug.log

C:\NodeRFC>
```

7. Notaras que esto podría provocar un error del tipo: **Cannot find module 'http-errors'**, esto es un aviso que nuestra aplicación requiere una dependencia para poder funcionar así que proseguiremos a instalar http-errors con la siguiente instrucción:

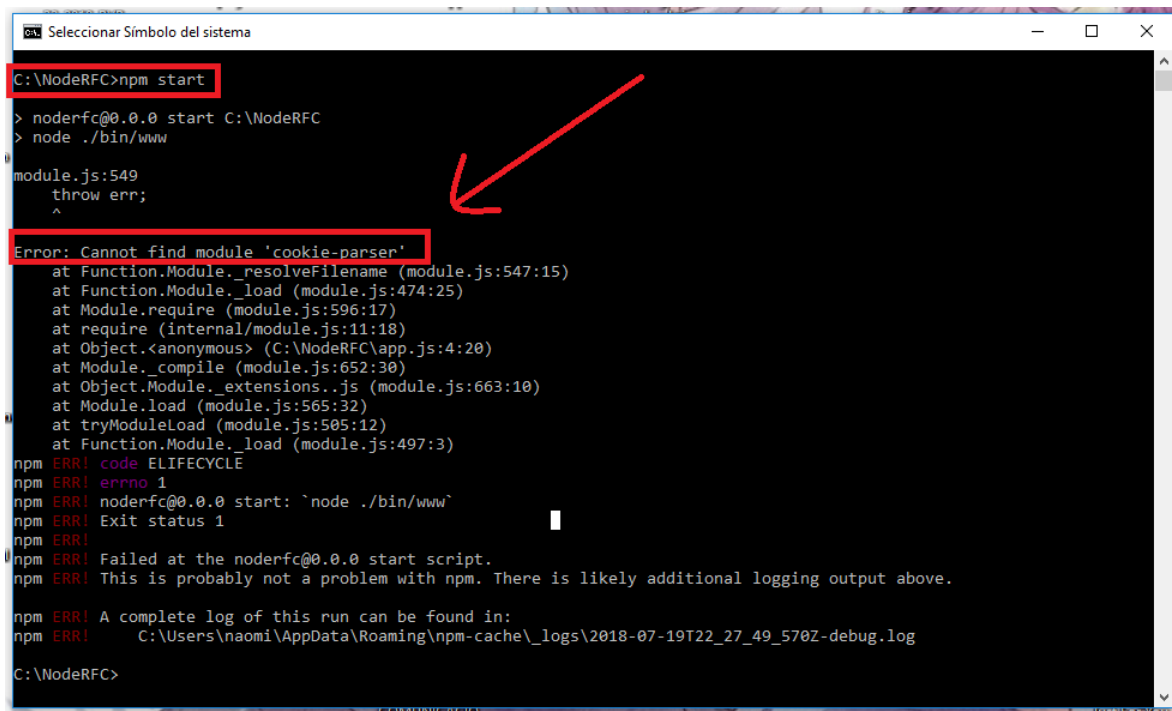
**npm install http-errors**



```
Símbolo del sistema
at Module.require (module.js:596:17)
at require (internal/module.js:11:18)
at Object.<anonymous> (C:\NodeRFC\app.js:4:20)
at Module._compile (module.js:652:30)
at Object.Module._extensions..js (module.js:663:10)
at Module.load (module.js:565:32)
at tryModuleLoad (module.js:505:12)
at Function.Module._load (module.js:497:3)
npm ERR! code ELIFECYCLE
npm ERR! errno 1
npm ERR! noderfc@0.0.0 start: `node ./bin/www`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the noderfc@0.0.0 start script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\naomi\AppData\Roaming\npm-cache\_logs\2018-07-19T22_22_49_270Z-debug.log
C:\NodeRFC>npm install http-errors
+ http-errors@1.6.3
updated 1 package in 6.611s
C:\NodeRFC>
```

8. En caso de no marcar un error anterior, podría marcar el siguiente error:



```
Seleccionar Símbolo del sistema
C:\NodeRFC>npm start
> noderfc@0.0.0 start C:\NodeRFC
> node ./bin/www

module.js:549
  throw err;
  ^
Error: Cannot find module 'cookie-parser'
at Function.Module._resolveFilename (module.js:547:15)
at Function.Module._load (module.js:474:25)
at Module.require (module.js:596:17)
at require (internal/module.js:11:18)
at Object.<anonymous> (C:\NodeRFC\app.js:4:20)
at Module._compile (module.js:652:30)
at Object.Module._extensions..js (module.js:663:10)
at Module.load (module.js:565:32)
at tryModuleLoad (module.js:505:12)
at Function.Module._load (module.js:497:3)
npm ERR! code ELIFECYCLE
npm ERR! errno 1
npm ERR! noderfc@0.0.0 start: `node ./bin/www`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the noderfc@0.0.0 start script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\naomi\AppData\Roaming\npm-cache\_logs\2018-07-19T22_27_49_570Z-debug.log
C:\NodeRFC>
```

9. De igual forma instalaremos la dependencia utilizando npm install nombreDependencia:

**npm install cookie-parser**

```
Símbolo del sistema
> noderfc@0.0.0 start C:\NodeRFC
> node ./bin/www

module.js:549
  throw err;
  ^

Error: Cannot find module 'cookie-parser'
    at Function.Module._resolveFilename (module.js:547:15)
    at Function.Module._load (module.js:474:25)
    at Module.require (module.js:596:17)
    at require (internal/module.js:11:18)
    at Object.<anonymous> (C:\NodeRFC\app.js:4:20)
    at Module._compile (module.js:652:30)
    at Object.Module._extensions..js (module.js:663:10)
    at Module.load (module.js:565:32)
    at tryModuleLoad (module.js:505:12)
    at Function.Module._load (module.js:497:3)
npm ERR! code ELIFECYCLE
npm ERR! errno 1
npm ERR! noderfc@0.0.0 start: `node ./bin/www`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the noderfc@0.0.0 start script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!   C:\Users\naomi\AppData\Roaming\npm-cache\_logs\2018-07-19T22_30_20_132Z-debug.log

C:\NodeRFC>npm install cookie-parser
+ cookie-parser@1.4.3
added 1 package in 34.959s

C:\NodeRFC>
```

10. Vamos a realizar el mismo procedimiento, pero con la dependencia morgan, así que también lo instalaremos de la siguiente forma:

**npm install morgan**

```
Símbolo del sistema
  throw err;
  ^

Error: Cannot find module 'cookie-parser'
    at Function.Module._resolveFilename (module.js:547:15)
    at Function.Module._load (module.js:474:25)
    at Module.require (module.js:596:17)
    at require (internal/module.js:11:18)
    at Object.<anonymous> (C:\NodeRFC\app.js:4:20)
    at Module._compile (module.js:652:30)
    at Object.Module._extensions..js (module.js:663:10)
    at Module.load (module.js:565:32)
    at tryModuleLoad (module.js:505:12)
    at Function.Module._load (module.js:497:3)
npm ERR! code ELIFECYCLE
npm ERR! errno 1
npm ERR! noderfc@0.0.0 start: `node ./bin/www`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the noderfc@0.0.0 start script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!   C:\Users\naomi\AppData\Roaming\npm-cache\_logs\2018-07-19T22_30_20_132Z-debug.log

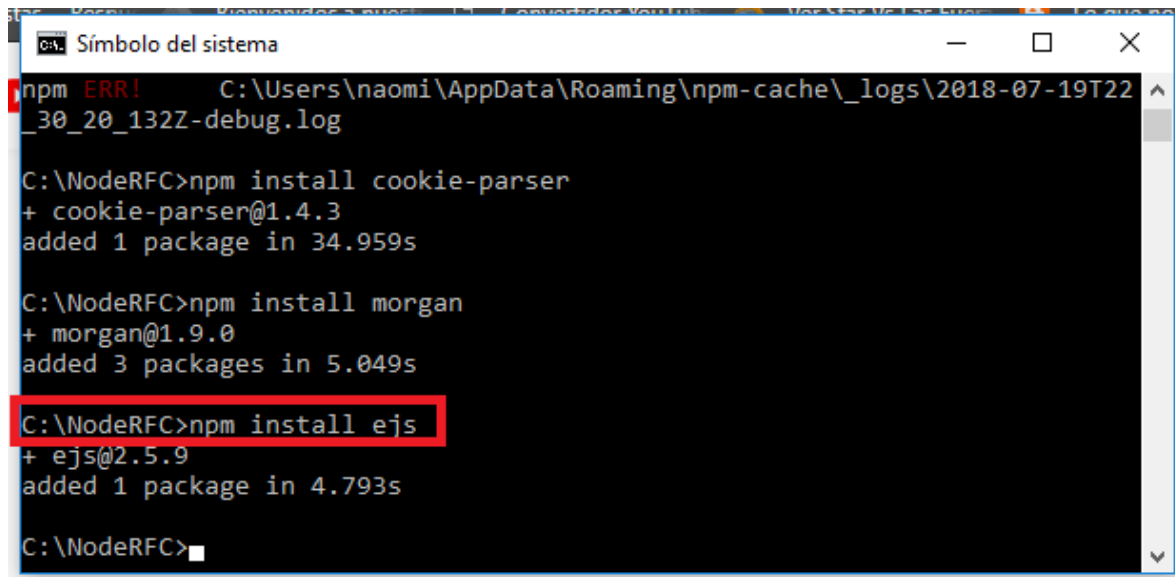
C:\NodeRFC>npm install cookie-parser
+ cookie-parser@1.4.3
added 1 package in 34.959s

C:\NodeRFC>npm install morgan
+ morgan@1.9.0
added 3 packages in 5.049s

C:\NodeRFC>
```

11. Como último punto instalaremos el motor de plantillas ejs para trabajar con las views así que escribiremos lo siguiente:

### npm install ejs



```
C:\Users\naomi\AppData\Roaming\npm-cache\_logs\2018-07-19T22_30_20_132Z-debug.log

C:\NodeRFC>npm install cookie-parser
+ cookie-parser@1.4.3
added 1 package in 34.959s

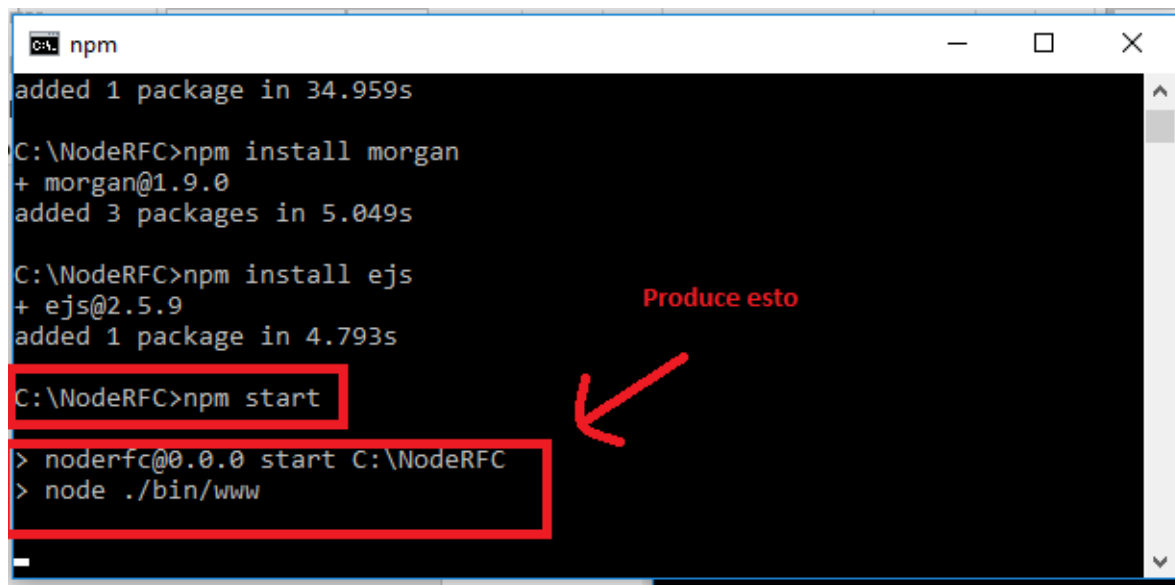
C:\NodeRFC>npm install morgan
+ morgan@1.9.0
added 3 packages in 5.049s

C:\NodeRFC>npm install ejs
+ ejs@2.5.9
added 1 package in 4.793s

C:\NodeRFC>
```

12. 11.- Una vez terminado todo el procedimiento de creación del proyecto escribimos lo siguiente:

### npm start



```
added 1 package in 34.959s

C:\NodeRFC>npm install morgan
+ morgan@1.9.0
added 3 packages in 5.049s

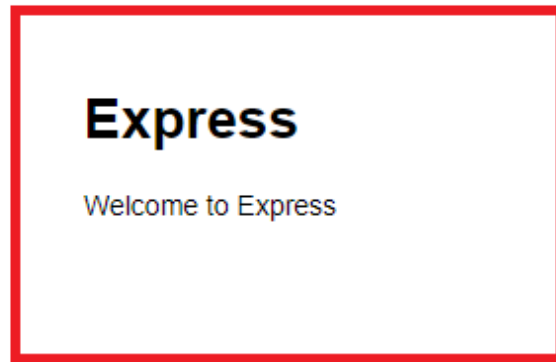
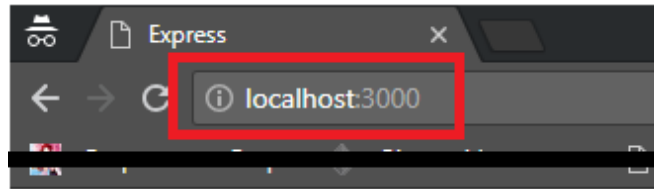
C:\NodeRFC>npm install ejs
+ ejs@2.5.9
added 1 package in 4.793s

C:\NodeRFC>npm start
> noderfc@0.0.0 start C:\NodeRFC
> node ./bin/www
```

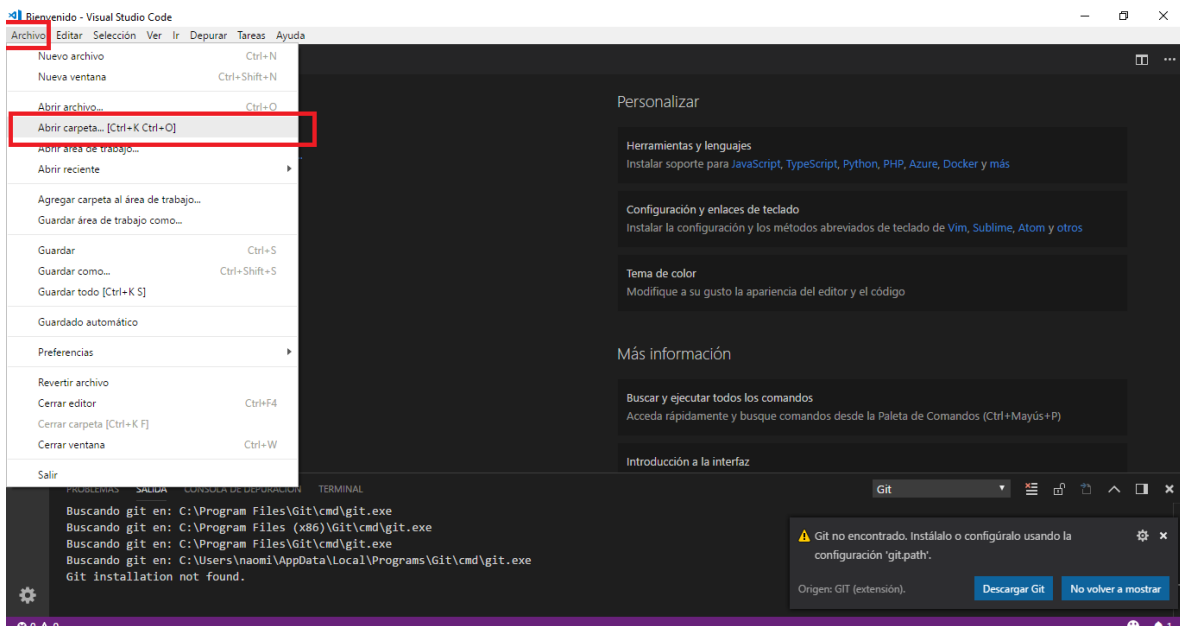
Produce esto

13. Abrimos el navegador y escribimos la siguiente ruta para cotejar la instalación de express, si te salió en el navegador de la siguiente forma ya estarás listo para empezar con express js:

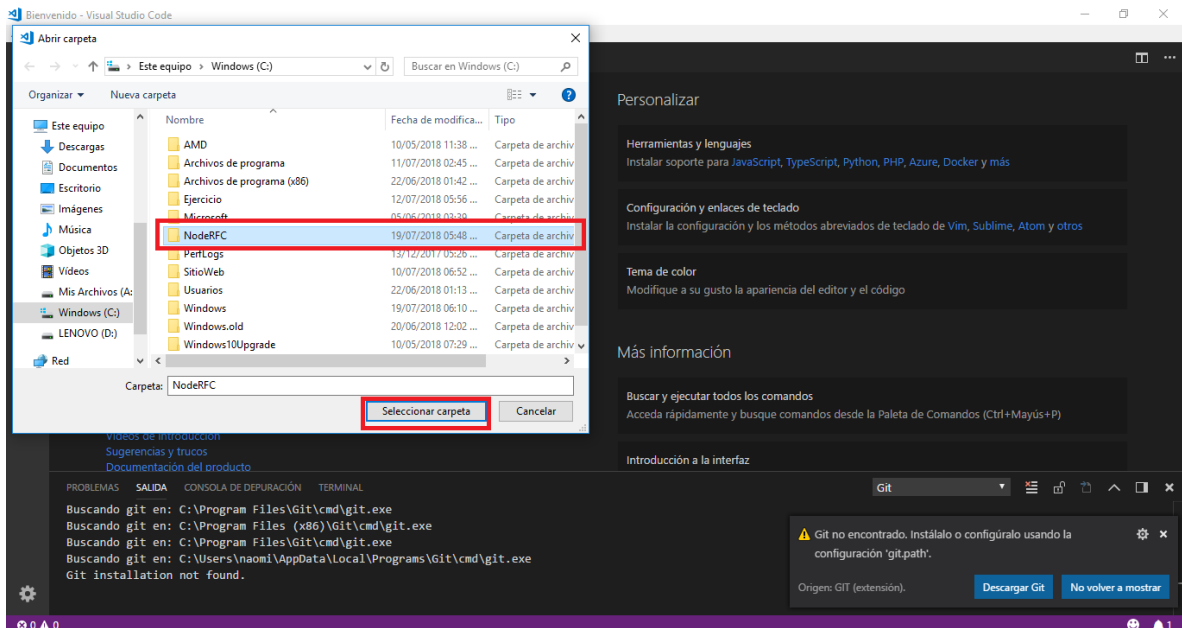
**localhost:3000**



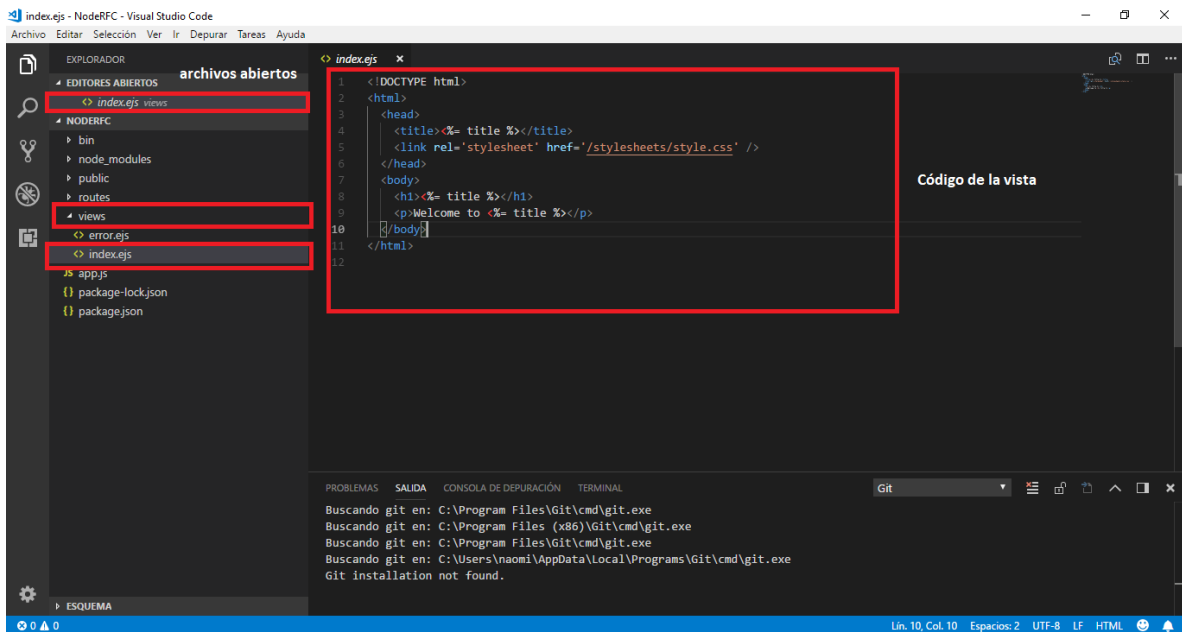
14. Abriremos la carpeta del proyecto con nuestro editor de código (por ejemplo, Visual Studio Code):







15. Una vez abierta la carpeta, abrimos **index.ejs** que se encuentra dentro de la carpeta **views**



16. Se borra el código dentro del body, se coloca el siguiente código para poder ingresar los datos necesarios para calcular el RFC:



```
<input type="text" name="nombre" />
</p>
<p>Apellido Paterno: </p>
<p>
  <input type="text" name="paterno" />
</p>
<p>Apellido Materno: </p>
<p>
  <input type="text" name="materno" />
</p>
<p>Fecha de Nacimiento: </p>
<p>
  <select name="dia">
    <option value="01">01</option>
    <option value="02">02</option>
    <option value="03">03</option>
    <option value="04">04</option>
    <option value="05">05</option>
    <option value="06">06</option>
    <option value="07">07</option>
    <option value="08">08</option>
    <option value="09">09</option>
    <option value="10">10</option>
    <option value="11">11</option>
    <option value="12">12</option>
    <option value="13">13</option>
    <option value="14">14</option>
    <option value="15">15</option>
    <option value="16">16</option>
    <option value="17">17</option>
    <option value="18">18</option>
    <option value="19">19</option>
    <option value="20">20</option>
    <option value="21">21</option>
    <option value="22">22</option>
    <option value="23">23</option>
    <option value="24">24</option>
    <option value="25">25</option>
    <option value="26">26</option>
    <option value="27">27</option>
    <option value="28">28</option>
    <option value="29">29</option>
    <option value="30">30</option>
    <option value="31">31</option>
  </select>
```

```
<select name="mes">
  <option value="01">01</option>
  <option value="02">02</option>
  <option value="03">03</option>
  <option value="04">04</option>
  <option value="05">05</option>
  <option value="06">06</option>
  <option value="07">07</option>
  <option value="08">08</option>
  <option value="09">09</option>
  <option value="10">10</option>
  <option value="11">11</option>
  <option value="12">12</option>
</select>
<select name="anio">
  <option value="2007">2007</option>
  <option value="2006">2006</option>
  <option value="2005">2005</option>
  <option value="2004">2004</option>
  <option value="2003">2003</option>
  <option value="2002">2002</option>
  <option value="2001">2001</option>
  <option value="2000">2000</option>
  <option value="1999">1999</option>
  <option value="1998">1998</option>
  <option value="1997">1997</option>
  <option value="1996">1996</option>
  <option value="1995">1995</option>
  <option value="1994">1994</option>
  <option value="1993">1993</option>
  <option value="1992">1992</option>
  <option value="1991">1991</option>
  <option value="1990">1990</option>
  <option value="1989">1989</option>
  <option value="1988">1988</option>
  <option value="1987">1987</option>
  <option value="1986">1986</option>
  <option value="1985">1985</option>
  <option value="1984">1984</option>
  <option value="1983">1983</option>
  <option value="1982">1982</option>
  <option value="1981">1981</option>
  <option value="1980">1980</option>
  <option value="1979">1979</option>
  <option value="1978">1978</option>
```

```
<option value="1977">1977</option>
<option value="1976">1976</option>
<option value="1975">1975</option>
<option value="1974">1974</option>
<option value="1973">1973</option>
<option value="1972">1972</option>
<option value="1971">1971</option>
<option value="1970">1970</option>
<option value="1969">1969</option>
<option value="1968">1968</option>
<option value="1967">1967</option>
<option value="1966">1966</option>
<option value="1965">1965</option>
<option value="1964">1964</option>
<option value="1963">1963</option>
<option value="1962">1962</option>
<option value="1961">1961</option>
<option value="1960">1960</option>
<option value="1959">1959</option>
<option value="1958">1958</option>
<option value="1957">1957</option>
<option value="1956">1956</option>
<option value="1955">1955</option>
<option value="1954">1954</option>
<option value="1953">1953</option>
<option value="1952">1952</option>
<option value="1951">1951</option>
<option value="1950">1950</option>
<option value="1949">1949</option>
<option value="1948">1948</option>
<option value="1947">1947</option>
<option value="1946">1946</option>
<option value="1945">1945</option>
<option value="1944">1944</option>
<option value="1943">1943</option>
<option value="1942">1942</option>
<option value="1941">1941</option>
<option value="1940">1940</option>
<option value="1939">1939</option>
<option value="1938">1938</option>
<option value="1937">1937</option>
<option value="1936">1936</option>
<option value="1935">1935</option>
<option value="1934">1934</option>
<option value="1933">1933</option>
```

```

        <option value="1932">1932</option>
        <option value="1931">1931</option>
        <option value="1930">1930</option>
        <option value="1929">1929</option>
        <option value="1928">1928</option>
        <option value="1927">1927</option>
        <option value="1926">1926</option>
        <option value="1925">1925</option>
        <option value="1924">1924</option>
        <option value="1923">1923</option>
        <option value="1922">1922</option>
        <option value="1921">1921</option>
        <option value="1920">1920</option>
        <option value="1919">1919</option>
        <option value="1918">1918</option>
        <option value="1917">1917</option>
        <option value="1916">1916</option>
        <option value="1915">1915</option>
        <option value="1914">1914</option>
        <option value="1913">1913</option>
        <option value="1912">1912</option>
        <option value="1911">1911</option>
        <option value="1910">1910</option>
        <option value="1909">1909</option>
        <option value="1908">1908</option>
        <option value="1907">1907</option>
        <option value="1906">1906</option>
        <option value="1905">1905</option>
        <option value="1904">1904</option>
        <option value="1903">1903</option>
        <option value="1902">1902</option>
        <option value="1901">1901</option>
        <option value="1900">1900</option>
    </select>
</p>
<p>
    <input type="submit" name="btnGenerar" value="Generar RFC" />
</p>
<br/><br/>
<b><%= RFCres %></b>

```

Las etiquetas que se usan son las típicas de HTML para crear el formulario.

```
14 <body>
15 <div id="wrap"><center>
16 <p><h1>Generar RFC</h1></p>
17 <ul> </ul>
18 <form action="" method="post">
19 <p>Nombre: </p>
20 <p>
21 <input type="text" name="nombre" />
22 </p>
23 <p>Apellido Paterno: </p>
24 <p>
25 <input type="text" name="paterno" />
26 </p>
27 <p>Apellido Materno: </p>
28 <p>
29 <input type="text" name="materno" />
30 </p>
31 <p>Fecha de Nacimiento: </p>
32 <p>
33 <select name="dia">...
34 </select>
35 <select name="mes">...
36 </select>
37 <select name="anio">...
38 </select>
39 </p>
40 <p>
41 <input type="submit" name="btnGenerar" value="Generar RFC" />
42 </p>
43 <br/><br/>
44 <b><%= RFCres %></b>
45 </body>
46 </html>
```

17. Al correr el código con **npm start**, podría aparecer un error como el siguiente:

```
C:\NodeRFC\views\index.ejs:195:193| <p> 194| <br/><br/> >> 195| <b><%= RFCres %></b> 196|
</body> 197| </html> RFCres is not defined

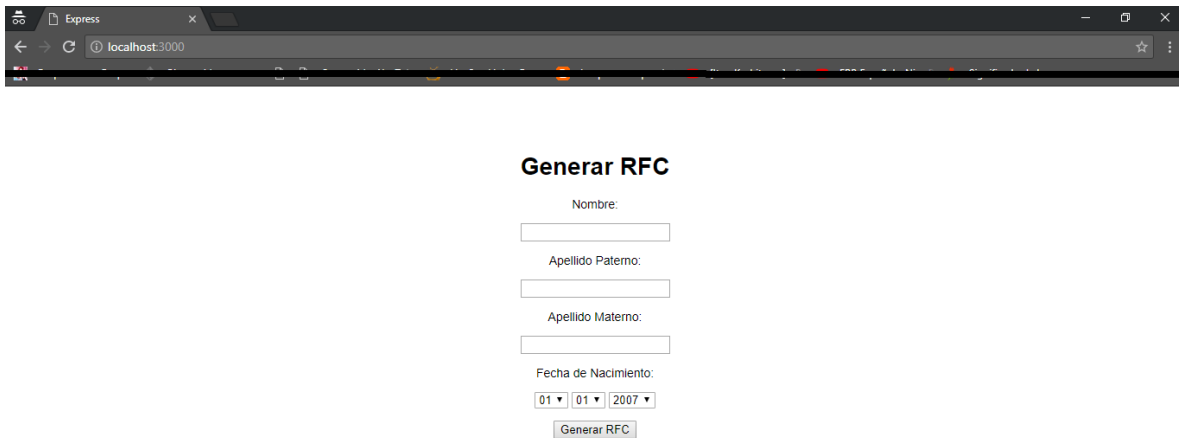
ReferenceError: C:\NodeRFC\views\index.ejs:195:193
193| <p>
194| <br/><br/>
>> 195| <b><%= RFCres %></b>
196| </body>
197| </html>

RFCres is not defined
at eval (eval at compile (C:\NodeRFC\node_modules\ejs\lib\ejs.js:592:12), <anonymous>:17:26)
at returnedFn (C:\NodeRFC\node_modules\ejs\lib\ejs.js:623:17)
at tryHandleCache (C:\NodeRFC\node_modules\ejs\lib\ejs.js:251:36)
at View.renderFile [as engine] (C:\NodeRFC\node_modules\ejs\lib\ejs.js:480:10)
at View.render (C:\NodeRFC\node_modules\express\lib\view.js:135:8)
at tryRender (C:\NodeRFC\node_modules\express\lib\application.js:640:10)
at Function.render (C:\NodeRFC\node_modules\express\lib\application.js:592:3)
at ServerResponse.render (C:\NodeRFC\node_modules\express\lib\response.js:1008:7)
at C:\NodeRFC\routes\index.js:6:7
at Layer.handle [as handle_request] (C:\NodeRFC\node_modules\express\lib\router\layer.js:95:5)
```

18. Esto es normal, y no hay necesidad de preocuparse, ya que la clave del error es **RFCres**, una variable que aún no ha sido definida, para arreglar este error, es necesario abrir **index.js** dentro de la carpeta **routes**, y agregar la variable, de la siguiente manera:

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function(req, res, next) {
6   res.render('index', { title: 'Express', RFCres: ' ' });
7 });
8
9 module.exports = router;
10
```

19. Vuelven a correr el código, y ya será visible el formulario:



**Generar RFC**

Nombre:

Apellido Paterno:

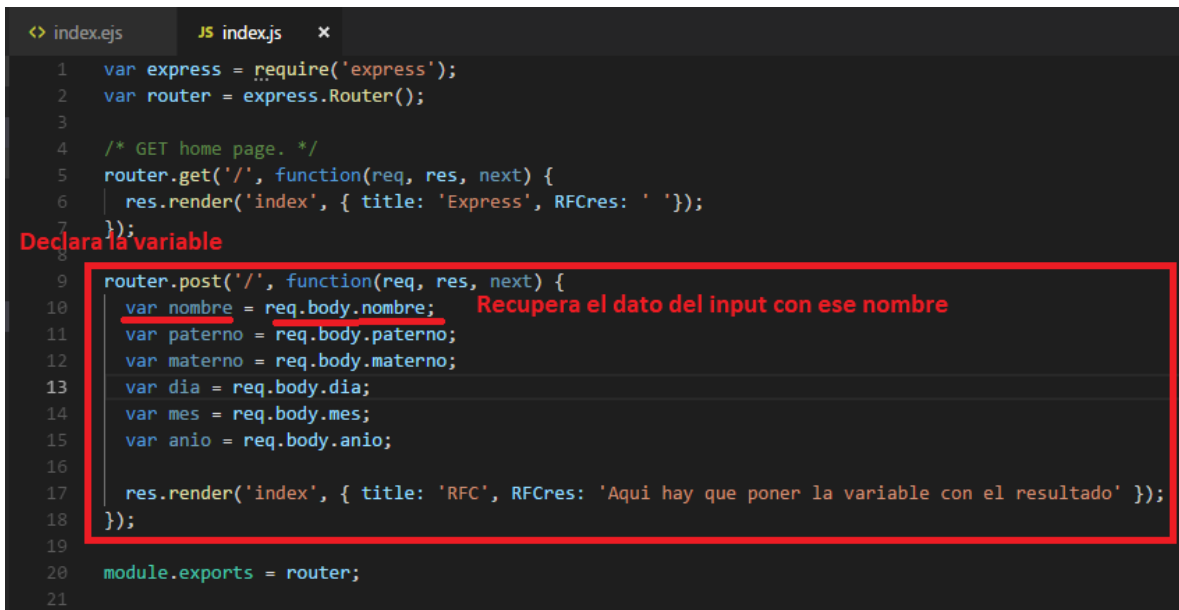
Apellido Materno:

Fecha de Nacimiento:

01 ▼ 01 ▼ 2007 ▼

Generar RFC

20. Para poder calcular el RFC, se debe crear el método POST en **index.js**, y se recuperaran los datos ingresados en la vista:



```
1  var express = require('express');
2  var router = express.Router();
3
4  /* GET home page. */
5  router.get('/', function(req, res, next) {
6    res.render('index', { title: 'Express', RFCres: ' ' });
7  });
8  Declara la variable
9  router.post('/', function(req, res, next) {
10     var nombre = req.body.nombre; Recupera el dato del input con ese nombre
11     var paterno = req.body.paterno;
12     var materno = req.body.materno;
13     var dia = req.body.dia;
14     var mes = req.body.mes;
15     var anio = req.body.anio;
16
17     res.render('index', { title: 'RFC', RFCres: 'Aqui hay que poner la variable con el resultado' });
18  });
19
20  module.exports = router;
21
```

Cuando se termine el código, se remplazará el texto por el valor de la variable que contenga el RFC.

21. Se procederá a agregar las funciones, empezando por las funciones **QuitarArticulos** y **EsVocal**, que se usaran después:

```
function QuitarArticulos(palabra)
{
  palabra=palabra.replace("DEL ", "", palabra);
  palabra=palabra.replace("LAS ", "", palabra);
  palabra=palabra.replace("DE ", "", palabra);
  palabra=palabra.replace("LA ", "", palabra);
}
```

```

    palabra=palabra.replace("Y ", "", palabra);
    palabra=palabra.replace("A ", "", palabra);
    return palabra;
}

function EsVocal(letra)
{
    if (letra == 'A' || letra == 'E' || letra == 'I' || letra == 'O' ||
letra == 'U' || letra == 'a' || letra == 'e' || letra == 'i' || letra == 'o'
|| letra == 'u'){
        return 1; }
    else {
        return 0;
    }
}

```

22. La siguiente función sirve para calcular la HomoClave:

```

function CalcularHomoclave(nombreCompleto, fecha, rfc)
{
    //Guardara el nombre en su correspondiente numérico
    //Agregamos un cero al inicio de la representación numérica del nombre
    var nombreEnNumero="0";
    //La suma de la secuencia de números de nombreEnNumero
    var valorSuma = 0;
    //region Tablas para calcular la homoclave
    var tablaRFC1 = [];
    tablaRFC1['&'] = '10';
    tablaRFC1['Ã'] = '10';
    tablaRFC1['A'] = '11';
    tablaRFC1['B'] = '12';
    tablaRFC1['C'] = '13';
    tablaRFC1['D'] = '14';
    tablaRFC1['E'] = '15';
    tablaRFC1['F'] = '16';
    tablaRFC1['G'] = '17';
    tablaRFC1['H'] = '18';
    tablaRFC1['I'] = '19';
    tablaRFC1['J'] = '21';
    tablaRFC1['K'] = '22';
    tablaRFC1['L'] = '23';
    tablaRFC1['M'] = '24';
    tablaRFC1['N'] = '25';
    tablaRFC1['O'] = '26';
    tablaRFC1['P'] = '27';
    tablaRFC1['Q'] = '28';
    tablaRFC1['R'] = '29';
}

```



```
tablaRFC1['S']='32';
tablaRFC1['T']='33';
tablaRFC1['U']='34';
tablaRFC1['V']='35';
tablaRFC1['W']='36';
tablaRFC1['X']='37';
tablaRFC1['Y']='38';
tablaRFC1['Z']='39';
tablaRFC1['0']='00';
tablaRFC1['1']='01';
tablaRFC1['2']='02';
tablaRFC1['3']='03';
tablaRFC1['4']='04';
tablaRFC1['5']='05';
tablaRFC1['6']='06';
tablaRFC1['7']='07';
tablaRFC1['8']='08';
tablaRFC1['9']='09';
```

```
var tablaRFC2 = [];
tablaRFC2[0]="1";
tablaRFC2[1]="2";
tablaRFC2[2]="3";
tablaRFC2[3]="4";
tablaRFC2[4]="5";
tablaRFC2[5]="6";
tablaRFC2[6]="7";
tablaRFC2[7]="8";
tablaRFC2[8]="9";
tablaRFC2[9]="A";
tablaRFC2[10]="B";
tablaRFC2[11]="C";
tablaRFC2[12]="D";
tablaRFC2[13]="E";
tablaRFC2[14]="F";
tablaRFC2[15]="G";
tablaRFC2[16]="H";
tablaRFC2[17]="I";
tablaRFC2[18]="J";
tablaRFC2[19]="K";
tablaRFC2[20]="L";
tablaRFC2[21]="M";
tablaRFC2[22]="N";
tablaRFC2[23]="P";
tablaRFC2[24]="Q";
```

```
tablaRFC2[25]="R";  
tablaRFC2[26]="S";  
tablaRFC2[27]="T";  
tablaRFC2[28]="U";  
tablaRFC2[29]="V";  
tablaRFC2[30]="W";  
tablaRFC2[31]="X";  
tablaRFC2[32]="Y";  
tablaRFC2[33]="Z";
```

```
var tablaRFC3 = [];  
tablaRFC3['A']=10;  
tablaRFC3['B']=11;  
tablaRFC3['C']=12;  
tablaRFC3['D']=13;  
tablaRFC3['E']=14;  
tablaRFC3['F']=15;  
tablaRFC3['G']=16;  
tablaRFC3['H']=17;  
tablaRFC3['I']=18;  
tablaRFC3['J']=19;  
tablaRFC3['K']=20;  
tablaRFC3['L']=21;  
tablaRFC3['M']=22;  
tablaRFC3['N']=23;  
tablaRFC3['O']=25;  
tablaRFC3['P']=26;  
tablaRFC3['Q']=27;  
tablaRFC3['R']=28;  
tablaRFC3['S']=29;  
tablaRFC3['T']=30;  
tablaRFC3['U']=31;  
tablaRFC3['V']=32;  
tablaRFC3['W']=33;  
tablaRFC3['X']=34;  
tablaRFC3['Y']=35;  
tablaRFC3['Z']=36;  
tablaRFC3['0']=0;  
tablaRFC3['1']=1;  
tablaRFC3['2']=2;  
tablaRFC3['3']=3;  
tablaRFC3['4']=4;  
tablaRFC3['5']=5;  
tablaRFC3['6']=6;  
tablaRFC3['7']=7;
```

```

tablaRFC3['8']=8;
tablaRFC3['9']=9;
tablaRFC3['']=24;
tablaRFC3[' ']=37;
//Recorremos el nombre y vamos convirtiendo las letras en
//su valor numerico
var len_nombreCompleto=nombreCompleto.length;
for(var x=0; x<len_nombreCompleto; x++)
{
    c=nombreCompleto.substring(x,x+1);
    if (tablaRFC1[c] !== undefined && tablaRFC1[c]!== null){
        nombreEnNumero += tablaRFC1[c]; }
    else
        nombreEnNumero += "00";
}
//Calculamos la suma de la secuencia de números
//calculados anteriormente
//la formula es:
//(el caracter actual multiplicado por diez)
//(mas el valor del caracter siguiente )
//(y lo anterior multiplicado por el valor del caracter siguiente)
var n = nombreEnNumero.length - 1;
for (var i = 0; i < n; i++)
{
    var prod1 = nombreEnNumero.substring(i, i + 2);
    var prod2 = nombreEnNumero.substring(i + 1, i + 2);
    valorSuma += prod1 * prod2;
}
//Los magic numbers que aparecen por ahí deben tener algún origen
matemático
//relacionado con el algoritmo al igual que el proceso mismo de calcular
el
//digito verificador.
var div = 0;
var mod = 0;
div = valorSuma % 1000;
mod = Math.floor(div / 34); //cociente
div = div - mod * 34; //residuo
var hc = tablaRFC2[mod];
hc += tablaRFC2[div];
rfc += hc;
//Aquí empieza el cálculo del digito verificador basado en lo que
tenemos del RFC
var sumaParcial = 0;
n=rfc.length;

```

```

for (var i = 0; i < n; i++)
{
    var c=rfc.substring(i,i+1);
    if (tablaRFC3[c] !== undefined && tablaRFC3[c]!== null)
    {
        sumaParcial += (tablaRFC3[c] * (14 - (i + 1)));
    }
}

var moduloVerificador = sumaParcial % 11;
if (moduloVerificador == 0) {
}
else
{
    sumaParcial = 11 - moduloVerificador;
    if (sumaParcial == 10){
        rfc += "A"; }
    else{
        rfc += sumaParcial;
    }
}
return rfc;
}

```

23. Por último, se agrega la función para calcular el RFC, que hace uso de las funciones creadas anteriormente:

```

function CalcularRFC(nombre,apellidoPaterno,apellidoMaterno,fecha)
{
    /*Cambiamos todo a mayúsculas.
    Quitamos los espacios al principio y final del nombre y apellidos*/
    nombre = nombre.trim().toUpperCase();
    apellidoPaterno =apellidoPaterno.trim().toUpperCase();
    apellidoMaterno =apellidoMaterno.trim().toUpperCase();
    //RFC que se regresará
    var rfc = "";

    //Quitamos los articulos de los apellidos
    apellidoPaterno = QuitarArticulos(apellidoPaterno);
    apellidoMaterno = QuitarArticulos(apellidoMaterno);
    //Agregamos el primer caracter del apellido paterno
    rfc = apellidoPaterno.substring(0, 1);
    //Buscamos y agregamos al rfc la primera vocal del primer apellido
    var len_apellidoPaterno= apellidoPaterno.length;
    for(var x = 1; x < len_apellidoPaterno; x++)
    {

```

```

    var c = apellidoPaterno.substring(x,1);
    if (EsVocal(c))
    {
        rfc += c;
        break;
    }
}
//Agregamos el primer caracter del apellido materno
rfc += apellidoMaterno.substring(0, 1);

//Agregamos el primer caracter del primer nombre
rfc += nombre.substring(0, 1);

//agregamos la fecha ddmmYYYY
rfc += fecha.substring(6,8) + fecha.substring(2,4) +
fecha.substring(0,2);

//Le agregamos la homoclave al rfc
rfc = CalcularHomoclave(apellidoPaterno + " " + apellidoMaterno + " " +
nombre, fecha, rfc);
return rfc;
}

```

24. Por último, se crean las variables fecha y la variable que contendrá el resultado, al final, así debe quedar el código del POST:

```

7   });
8
9   router.post('/', function(req, res, next) {
10      var nombre = req.body.nombre;
11      var paterno = req.body.paterno;
12      var materno = req.body.materno;
13      var dia = req.body.dia;
14      var mes = req.body.mes;
15      var anio = req.body.anio;
16
17      function QuitarArticulos(palabra) ...
18      {
19      }
20
21      function EsVocal(letra) ...
22      {
23      }
24
25      function CalcularHomoclave(nombreCompleto, fecha, rfc) ...
26      {
27      }
28
29      function CalcularRFC(nombre,apellidoPaterno,apellidoMaterno,fecha) ...
30      {
31      }
32
33      var fecha = dia + mes +anio;
34      var resultat = "Tu RFC es " + CalcularRFC(nombre, paterno, materno, fecha);
35
36      res.render('index', { title: 'RFC', RFCres: resultat });
37   });
38
39   module.exports = router;
40
41

```

**Variables**

**Funciones**

**Ahora será visible el RFC**

25. Vuelve a cargar el código, ahora podrá calcular el RFC:

## Generar RFC

Nombre:

Juan

Apellido Paterno:

Lopez

Apellido Materno:

Perez

Fecha de Nacimiento:

04 ▼

06 ▼

1990 ▼

Generar RFC

Esto es un ejemplo

 ¿Necesitas conocer tu buro de crédito en forma gratuita? [Checar tu reporte de crédito](#)

### Consulta de RFC homoclave

Su RFC con homoclave es

**LOPJ9006042A2**

¿Deseas imprimir la Cédula de Identificación Fiscal como comprobante?

[Imprimir RFC](#)

Anuncios Google

[Imprimir RFC SAT](#)

[Citas por Internet](#)

### ¿Cuándo tiene sentido calcular mi RFC?

Para que tu RFC calculado tenga sentido debes haberlo tramitado utilizando tu CURP como explicación [cómo sacar el RFC](#). De no haberlo hecho, el RFC obtenido será meramente el resultado de un algoritmo que no tiene respaldo alguno en los registros del SAT.

## Generar RFC

Nombre:

Apellido Paterno:

Apellido Materno:

Fecha de Nacimiento:

01 ▼ 01 ▼ 2007 ▼

Generar RFC

Tu RFC es **LOPJ9006042A2**