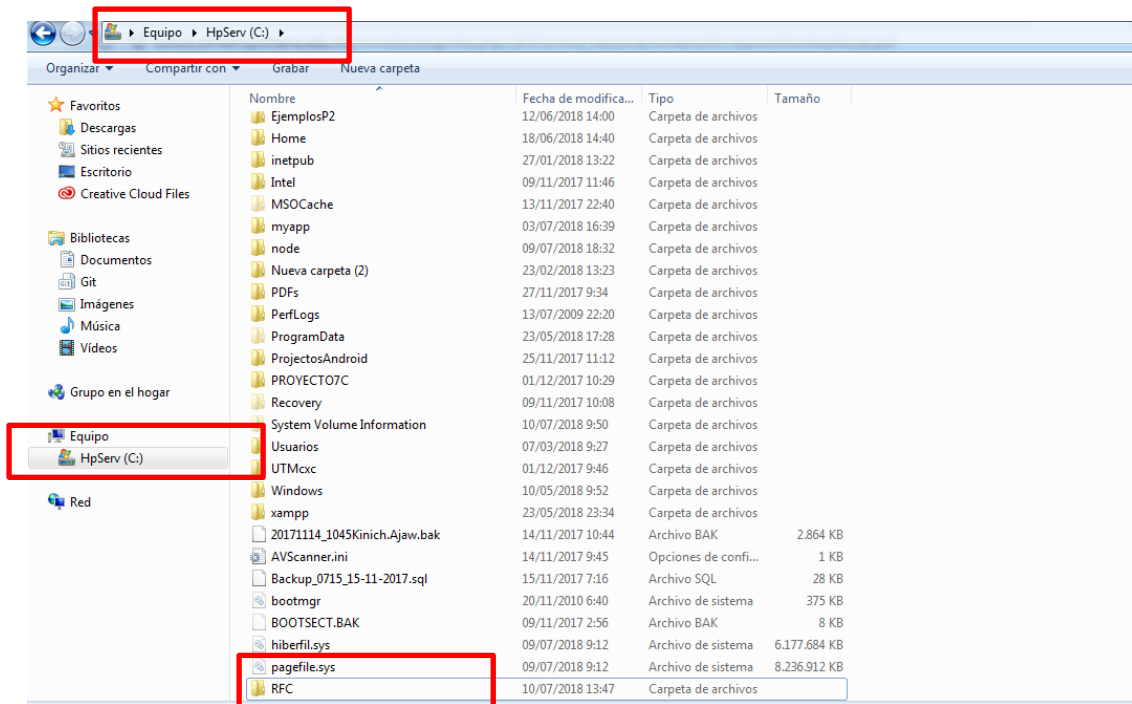


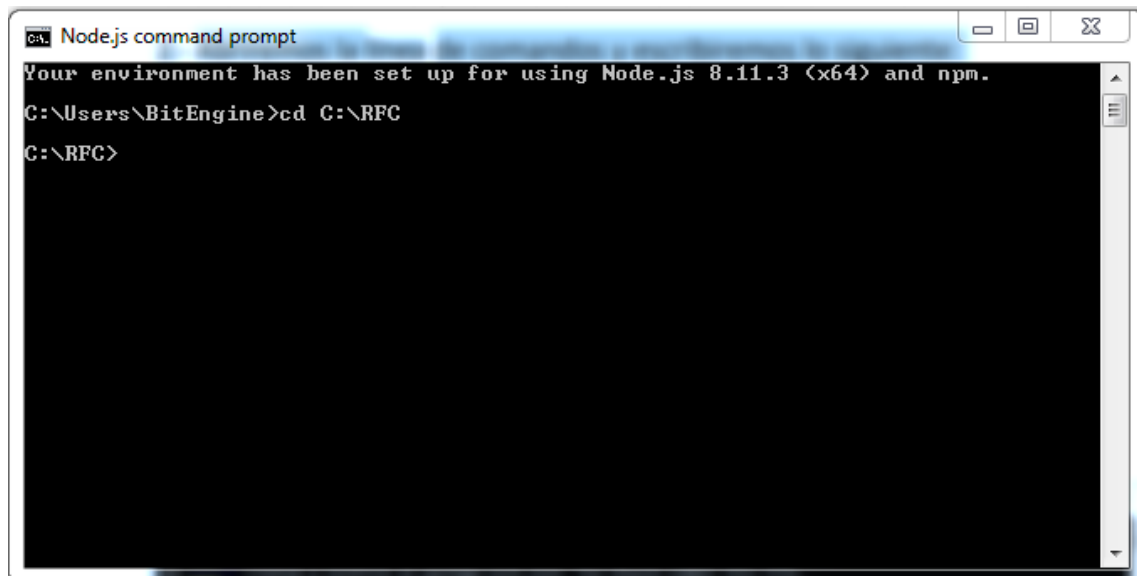
Manual para crear una aplicación RFC con (node.js y express).



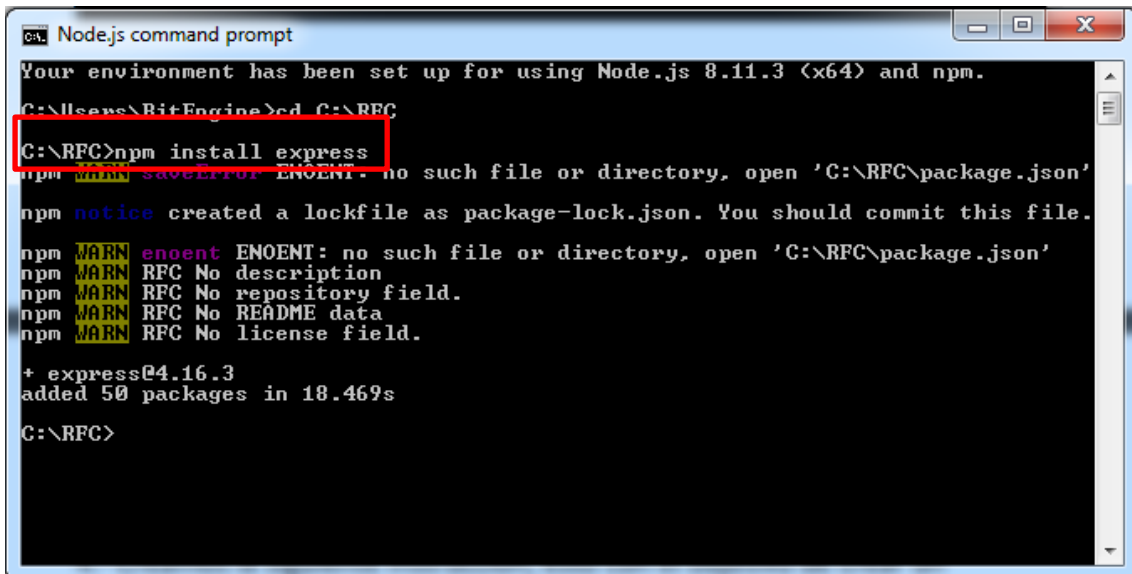
1.- Crearemos una carpeta llamada RFC



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



3.- Instalaremos express con la siguiente instrucción:

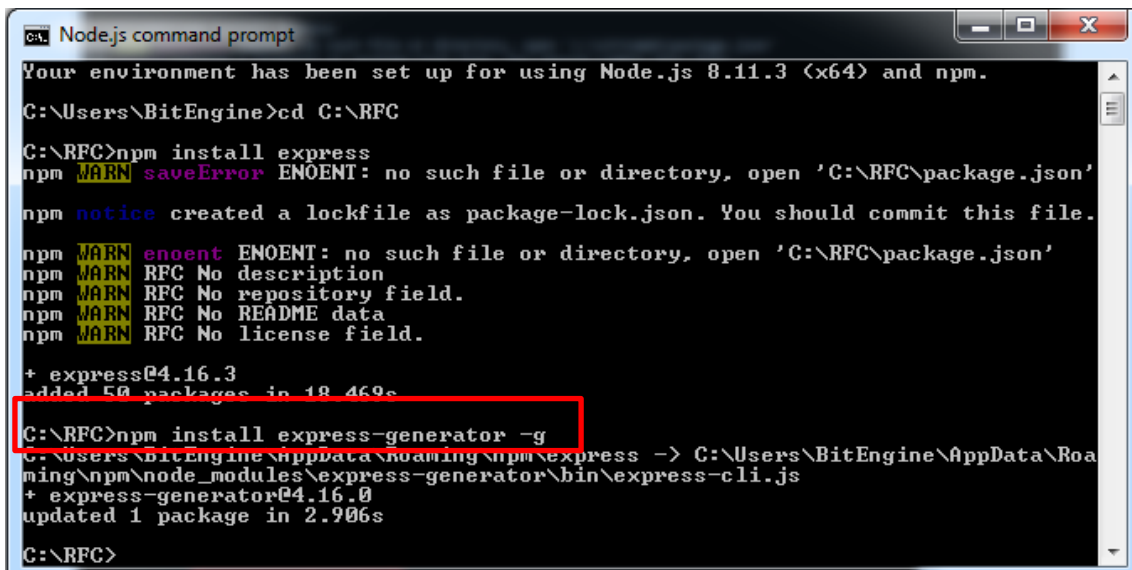


```

Node.js command prompt
Your environment has been set up for using Node.js 8.11.3 (x64) and npm.
C:\Users\BitEngine>cd C:\RFC
C:\RFC>npm install express
npm WARN saveError ENOENT: no such file or directory, open 'C:\RFC\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:\RFC\package.json'
npm WARN RFC No description
npm WARN RFC No repository field.
npm WARN RFC No README data
npm WARN RFC No license field.
+ express@4.16.3
added 50 packages in 18.469s
C:\RFC>

```

4.- Creamos la siguiente instrucción, esto con el objetivo de crear un proyecto con express:

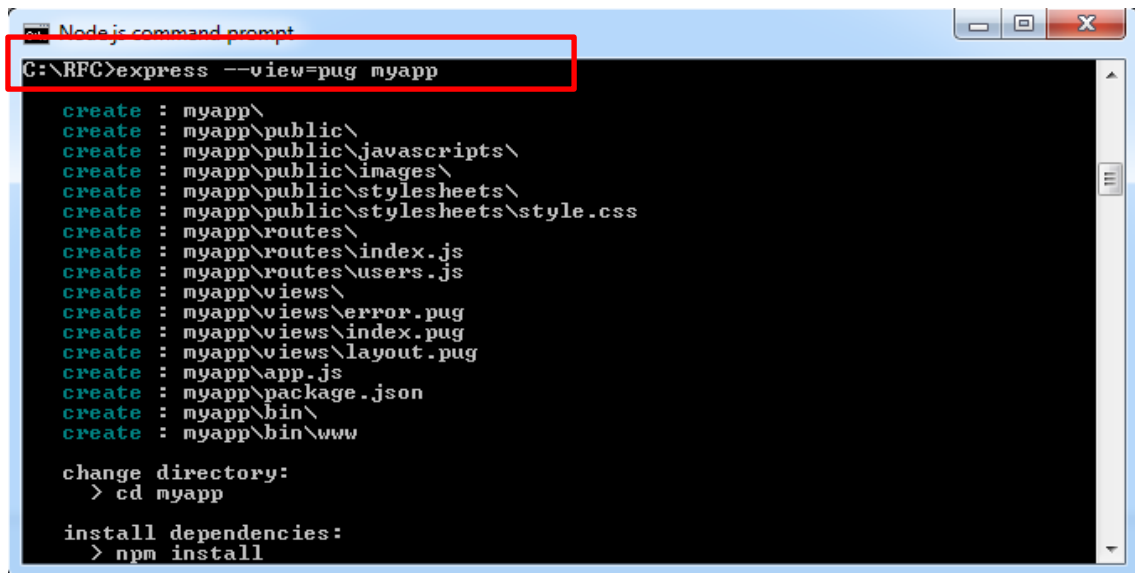


```

Node.js command prompt
Your environment has been set up for using Node.js 8.11.3 (x64) and npm.
C:\Users\BitEngine>cd C:\RFC
C:\RFC>npm install express
npm WARN saveError ENOENT: no such file or directory, open 'C:\RFC\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:\RFC\package.json'
npm WARN RFC No description
npm WARN RFC No repository field.
npm WARN RFC No README data
npm WARN RFC No license field.
+ express@4.16.3
added 50 packages in 18.469s
C:\RFC>npm install express-generator -g
C:\Users\BitEngine\AppData\Roaming\npm\express -> C:\Users\BitEngine\AppData\Roaming\npm\node_modules\express-generator\bin\express-cli.js
+ express-generator@4.16.0
updated 1 package in 2.906s
C:\RFC>

```

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



```

Node.js command prompt
C:\RFC>express --view=pug myapp

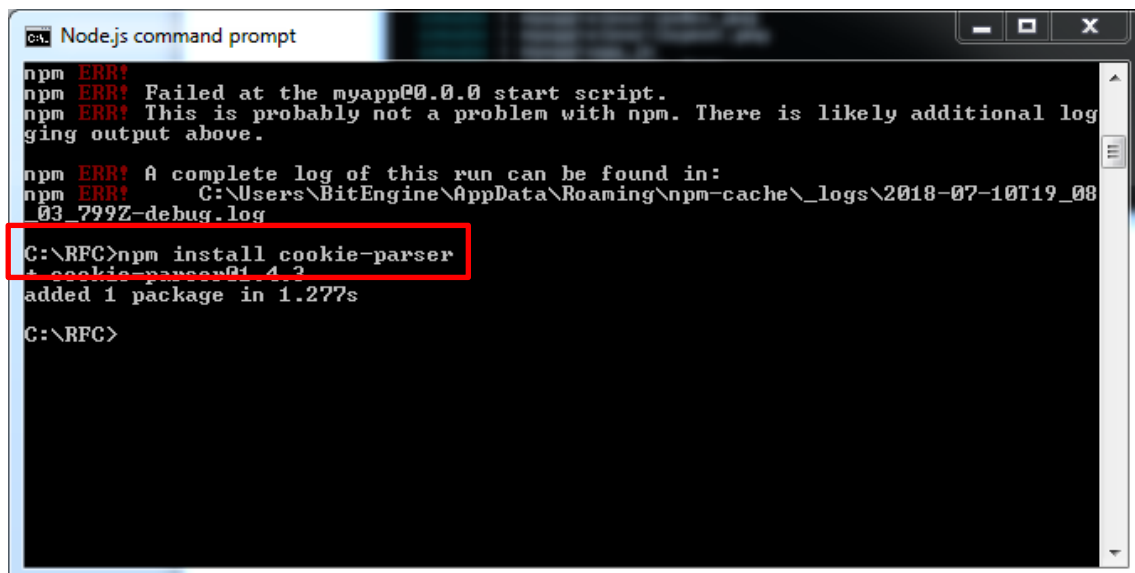
create : myapp\
create : myapp\public\
create : myapp\public\javascripts\
create : myapp\public\images\
create : myapp\public\stylesheets\
create : myapp\public\stylesheets\style.css
create : myapp\routes\
create : myapp\routes\index.js
create : myapp\routes\users.js
create : myapp\views\
create : myapp\views\error.pug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

change directory:
> cd myapp

install dependencies:
> npm install

```

6.- Instalaremos “cookie-parser” con la siguiente instrucción:



```

Node.js command prompt
npm ERR!
npm ERR! Failed at the myapp@0.0.0 start script.
npm ERR! This is probably not a problem with npm. There is likely additional log
ging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!     C:\Users\BitEngine\AppData\Roaming\npm-cache\_logs\2018-07-10T19_08
03_799Z-debug.log

C:\RFC>npm install cookie-parser
+ cookie-parser@0.4.3
added 1 package in 1.277s

C:\RFC>

```

7.- Instalaremos “morgan” con la siguiente instrucción:

```

at require (internal/module.js:11:18)
at Object.<anonymous> (C:\RFC\app.js:5:14)
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! myapp@0.0.0 start: `node ./bin/www`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the myapp@0.0.0 start script.
npm ERR! This is probably not a problem with npm. There is likely additional log
ging output above.

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\BitEngine\AppData\Roaming\npm-cache\_logs\2018-07-10T19_09
48_005Z-debug.log
C:\RFC>npm install morgan
+ morgan@1.9.0
added 3 packages in 1.591s
C:\RFC>

```

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

```

C:\RFC>npm install ejs
+ ejs@2.6.1
added 1 package in 1.37s
C:\RFC>

```

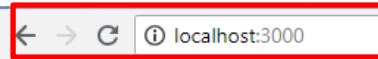
9.- Una vez terminado todo el procedimiento de creación del proyecto escribimos lo siguiente:

```

C:\RFC>npm install ejs
+ ejs@2.6.1
added 1 package in 1.37s
C:\RFC>npm start
> myapp@0.0.0 start C:\RFC
> node ./bin/www

```

10.- 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:



Express

Welcome to Express

11.- Después crearemos el diseño de nuestra aplicación con html desde la vista “index”:

 A screenshot of the Visual Studio Code editor. On the left, the Explorer sidebar shows a file tree with folders like 'bin', 'controllers', 'models', 'node_modules', 'public', 'routes', 'views', and 'app.js'. The 'views' folder is expanded, and 'index.js' is selected and highlighted with a red box. A red arrow points from this box to the main editor area. The main editor shows the content of 'index.js', which is an HTML file. The code includes a <form> with a <table> containing input fields for 'Nombre(s):' and 'Apellido Paterno:'. The 'Nombre(s):' field has attributes: id="NOMBRE_PERSONA", type="text", size="20", class="form-control", and maxlength="30". The 'Apellido Paterno:' field has attributes: id="APELLIDO_PATERNO", type="text", class="form-control", size="20", and maxlength="30". Both fields have an onchange event listener: "calcularRFC()".

 A screenshot of the Visual Studio Code editor, continuing from the previous one. The Explorer sidebar shows the same file tree, with 'index.js' still selected. The main editor shows the continuation of the HTML file. The code includes a <table> with a <tr> for 'Fecha Nacimiento:' and an input field with attributes: id="F_NACIMIENTO", name="F_NACIMIENTO", class="form-control", and type="text". The input field has an onchange event listener: "calcularRFC()". Below this is another <tr> for 'Sexo:' with two radio buttons. The first radio button has attributes: id="sexoFemenino", type="radio", name="SEXO", value="F", and class="simple_radio". The second radio button has attributes: id="sexoMasculino", type="radio", name="SEXO", value="M", and class="simple_radio". Below the radio buttons is a <label> for="SEXO" with class="error" and text: "Selección es requerida". The code ends with a </table> tag.

```

110      <span>Masculino</span>
111    </div>
112    <label for="SEXO" class="error">Selección es requerida</label>
113  </td>
114 </tr>
115 <tr>
116   <td>
117     RFC:
118   </td>
119   <td>
120     <input class="form-control" id="RFC" name="RFC" type="text" />
121   </td>
122 </tr>
123 </table>
124 </div>
125 </fieldset>
126 </body>
127 </html>

```

12.- Tendremos un diseño como este:

Cálculo de RFC online

Ingrese los datos correspondientes

Nombre(s):

Apellido Paterno:

Apellido Materno:

Fecha Nacimiento:

Sexo: ☐ Femenino ☒ Masculino

Selección es requerida

RFC:

@UTM - RFC Online

12.- Ahora crearemos el script que será el encargado de hacer el cálculo del RFC obteniendo los datos de los campos que agregamos:

```

8
9
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <meta http-equiv="X-UA-Compatible" content="ie=edge">
13 <title>Document</title>
14 <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js"></script>
15 <script>
16   function calcularRFC() {
17     function quitarArticulos(palabra) {
18       return palabra.replace("DEL ", "").replace("LAS ", "").replace("DE ",
19         "").replace("LA ", "").replace("Y ", "").replace("A ", "");
20     }
21     function esVocal(letra) {
22       if (letra == 'A' || letra == 'E' || letra == 'I' || letra == 'O'
23         || letra == 'U' || letra == 'a' || letra == 'e' || letra == 'i'
24         || letra == 'o' || letra == 'u')
25         return true;
26       else
27         return false;
28     }
29     nombre = $("#NOMBRE_PERSONA").val().toUpperCase();
30     apellidoPaterno = $("#APELLIDO_PATERNO").val().toUpperCase();
31     apellidoMaterno = $("#APELLIDO_MATERNO").val().toUpperCase();
32     fecha = $("#F_NACIMIENTO").val();
33     var rfc = "";
34     apellidoPaterno = quitarArticulos(apellidoPaterno);
35     apellidoMaterno = quitarArticulos(apellidoMaterno);
36     rfc += apellidoPaterno.substr(0, 1);
37     var l = apellidoPaterno.length;
38     var c;
39     for (i = 0; i < l; i++) {
40       c = apellidoPaterno.charAt(i);
41       if (esVocal(c)) {

```

```

28 nombre = $("#NOMBRE_PERSONA").val().toUpperCase();
29 apellidoPaterno = $("#APELLIDO_PATERNO").val().toUpperCase();
30 apellidoMaterno = $("#APELLIDO_MATERNO").val().toUpperCase();
31 fecha = $("#F_NACIMIENTO").val();
32 var rfc = "";
33 apellidoPaterno = quitaArticulos(apellidoPaterno);
34 apellidoMaterno = quitaArticulos(apellidoMaterno);
35 rfc += apellidoPaterno.substr(0, 1);
36 var l = apellidoPaterno.length;
37 var c;
38 for (i = 0; i < l; i++) {
39   c = apellidoPaterno.charAt(i);
40   if (esVocal(c)) {
41     rfc += c;
42     break;
43   }
44 }
45 rfc += apellidoMaterno.substr(0, 1);
46 rfc += nombre.substr(0, 1);
47 rfc += fecha.substr(0, 10);
48 rfc += fecha.substr(3, 5).substr(0, 2);
49 rfc += fecha.substr(0, 2);
50 // rfc += "-" + homclave;
51 $("#RFC").val(rfc);
52
53 </script>

```

12.- Por ultimo corremos el programa con este comando:

```

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

```

12.- Ahora nos dirigimos a nuestro navegador y corremos el programa con <http://localhost:3000/>, el resultado sería este:

Cálculo de RFC online

Ingrese los datos correspondientes

Nombre(s):

Apellido Paterno:

Apellido Materno:

Fecha Nacimiento:

Sexo:
☐ Femenino
☐ Masculino
 Selección es requerida

RFC:

@UTM - RFC Online

12.- Para finalizar llenamos los campos con nuestros datos y veremos que automáticamente se genera nuestro RFC:

Cálculo de RFC online

Ingrese los datos correspondientes

Nombre(s):

Apellido Paterno:

Apellido Materno:

Fecha Nacimiento:

Sexo:
☐ Femenino
☐ Masculino
Selección es requerida

RFC:

@UTM - RFC Online