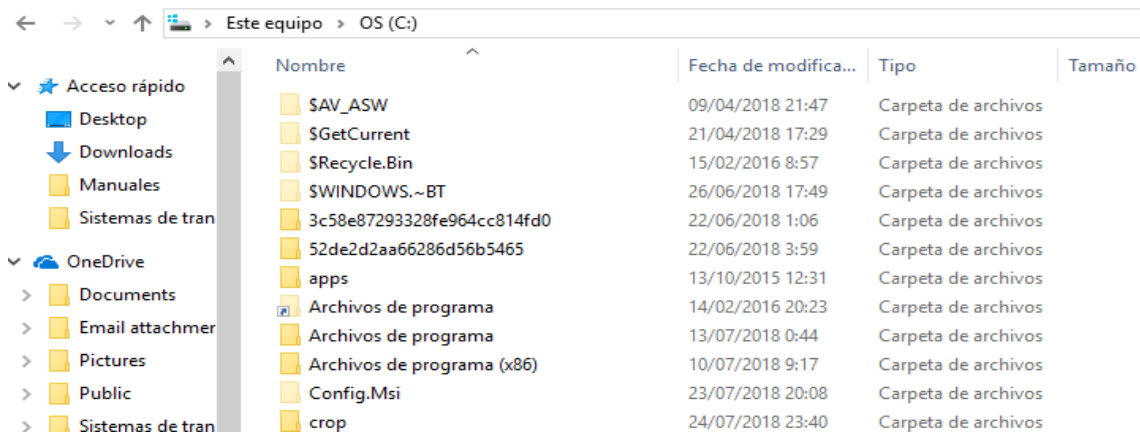


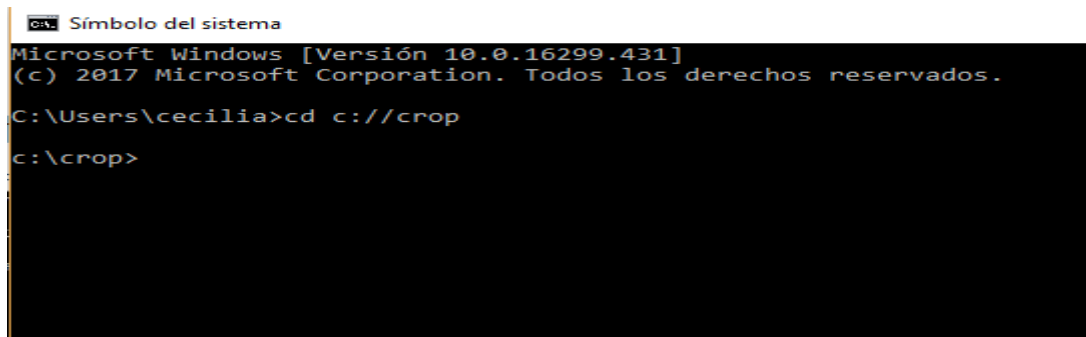
-

# Manual para realizar un crop de imágenes

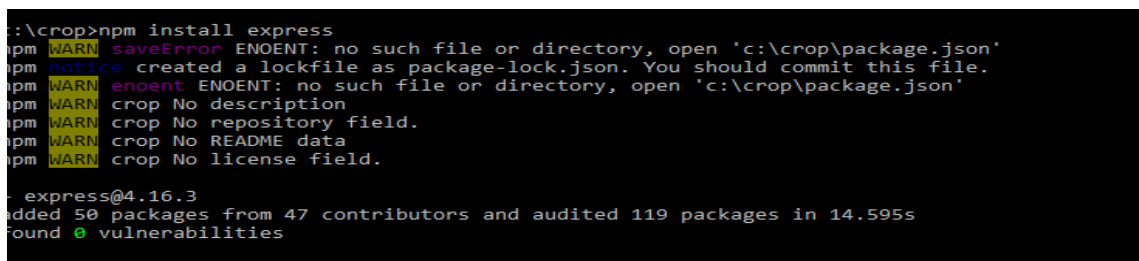
1-creamos una carpeta llamada crop



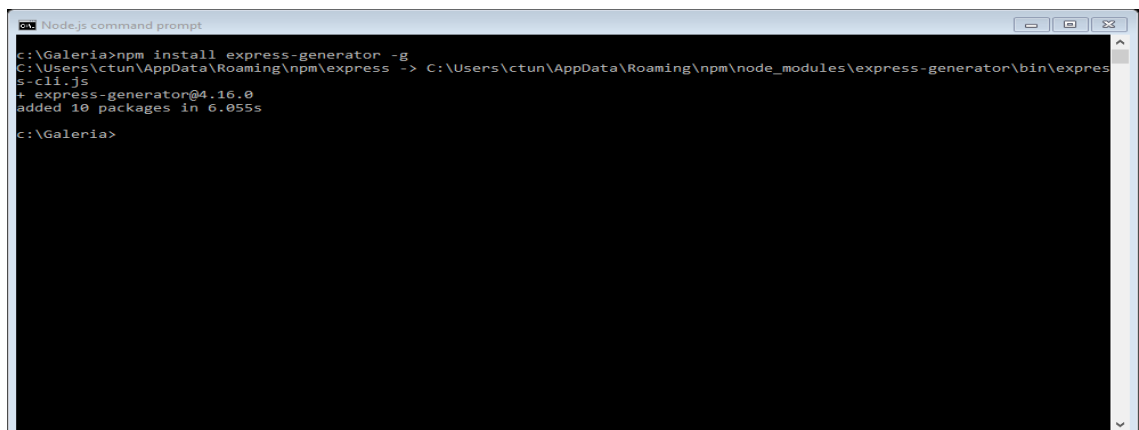
2-Abrimos la línea de comandos node.js y escribimos lo siguiente:



3-instalaremos Express con el siguiente comando:



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



5-Con el siguiente comando creamos la estructura del crop

```
c:\crop>npm install express-generator -g
c:\Users\cecilia\AppData\Roaming\npm\express -> C:\Users\cecilia\AppData\Roaming\npm\node_modules\express-generator
express-cli.js
+ express-generator@4.16.0
updated 1 package in 3.484s

c:\crop>express --views-ejs

Usage: express [options] [dir]

Options:

  --version      output the version number
  -e, --ejs      add ejs engine support
  --pug          add pug engine support
  --hbs          add handlebars engine support
  -H, --hogan     add hogan.js engine support
  -v, --view <engine> add view <engine> support (dust|ejs|hbs|hjs|jade|pug|twig|vash) (defaults to jade)
  --no-view      use static html instead of view engine
  -c, --css <engine> add stylesheet <engine> support (less|stylus|compass|sass) (defaults to plain css)
  --git          add .gitignore
  -f, --force    force on non-empty directory
  -h, --help     output usage information

error: unknown option '--views-ejs'
```

6-Corremos la aplicación con el siguiente comando

```
c:\crop>npm start
npm ERR! path c:\crop\package.json
npm ERR! code ENOENT
npm ERR! errno -4058
npm ERR! syscall open
npm ERR! enoent ENOENT: no such file or directory, open 'c:\crop\package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\cecilia\AppData\Roaming\npm-cache\_logs\2018-07-25T05_11_01_436Z-debug.log

c:\crop>npm install http-errors
```

7-se puede observar que al momento de correr la aplicación surgieron errores esto requiere de una dependencia para que funcione correctamente

```
c:\crop>npm install http-errors
pm WARN saveError ENOENT: no such file or directory, open 'c:\crop\package.json'
pm WARN enoent ENOENT: no such file or directory, open 'c:\crop\package.json'
pm WARN crop No description
pm WARN crop No repository field.
pm WARN crop No README data
pm WARN crop No license field.

+ http-errors@1.6.3
updated 1 package and audited 266 packages in 2.684s
found 0 vulnerabilities
```

8-una vez instaladas las librerías abrimos visual studio code escribiremos el siguiente código

```
index.ejs - nodeuploads-master - Visual Studio Code
Archivo  Editor  Selección  Ver  Ir  Depurar  Tareas  Ayuda

EXPLORADOR
  EDITORES ABIERTOS
    index.ejs views
    README.md
  JS app.js
  NODEUPLOADS-MASTER
    views
      index.ejs
    .gitignore
  JS app.js
  package.json
  README.md

index.ejs
1 |<!DOCTYPE html>
2 |<html lang="en">
3 |<head>
4 |   <meta charset="UTF-8">
5 |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 |   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 |   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/css/materialize.min.css">
8 |
9 |   <title>Node File Uploads</title>
10 |</head>
11 |<body>
12 |   <div class="container">
13 |     <h1>File Upload</h1>
14 |     <%= typeof msg != 'undefined' ? msg : '' %>
15 |     <form action="/upload" method="POST" enctype="multipart/form-data">
16 |       <div class="file-field input-field">
17 |         <div class="btn grey">
18 |           <span>File</span>
19 |           <input name="myImage" type="file">
20 |         </div>
21 |         <div class="file-path-wrapper">
22 |           <input class="file-path validate" type="text">
23 |         </div>
24 |       </div>
25 |       <button type="submit" class="btn">Submit</button>
26 |     </form>
```

9-en el siguiente apartado de apps.js se escribirá el siguiente código

```
1  const express = require('express');
2  const multer = require('multer');
3  const ejs = require('ejs');
4  const path = require('path');
5
6  // Set The Storage Engine
7  const storage = multer.diskStorage({
8    destination: './public/uploads/',
9    filename: function(req, file, cb){
10      cb(null, file.fieldname + '-' + Date.now() + path.extname(file.originalname));
11    }
12  });
13
14  // Init Upload
15  const upload = multer({
16    storage: storage,
17    limits: { fileSize: 1000000 },
18    fileFilter: function(req, file, cb){
19      checkFileType(file, cb);
20    }
21  }).single('myImage');
22
23  // Check File Type
24  function checkFileType(file, cb){
25    // Allowed ext
26    const filetypes = /jpeg|jpg|png|gif/;
27    // Check ext
28    const extname = filetypes.test(path.extname(file.originalname).toLowerCase());
29    // Check mime
30    const mimetype = filetypes.test(file.mimetype);
31
32    if(mimetype && extname){
33      return cb(null, true);
34    }
35  }
```

Corremos la aplicación con el comando `npm start`. Nos aparecerá en la aplicación el siguiente formulario

**Cortar Imagen**

Seleccionar archivo

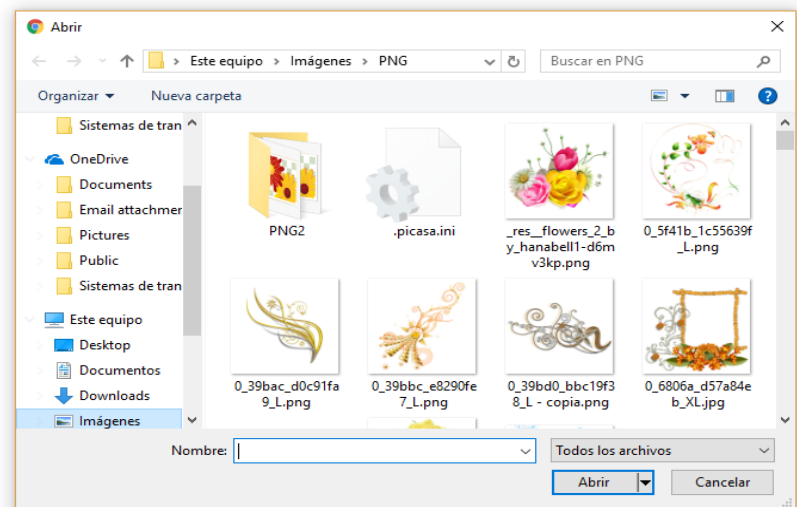
 Ningún archivo seleccionado

Cortar

Damos clic en seleccionar archivo, aparecerá el siguiente recuadro la cual podremos seleccionar la imagen que vamos a cortar

## Cortar Imagen

Seleccionar archivo Ningún archivo seleccionado



Al seleccionar la imagen ponemos cortar

