



HACK ACADEMY

Coding Bootcamp Sprint 2



Temario

Temario



- File Uploads.
- Variables de Entorno.
- Dotenv.



File Uploads

Subir y guardar archivos



File Uploads (1/4)

En una aplicación web es usual que los usuarios suban una foto de perfil, un video, un CV en PDF, etc.

Los archivos que se suben a un sitio se pueden guardar en:

- Filesystem (sistema de archivos del servidor).
- [Amazon S3](#).
- [Rackspace Cloud Files](#).
- [Google Cloud Storage](#).
- [Microsoft Azure Storage](#).
- Etc.



File Uploads (2/4) – Formulario

Para poder adjuntar un archivo en un formulario HTML simplemente se utiliza un `<input>` con `type="file"`. También se debe agregar `enctype="multipart/form-data"`:

```
<form action="..." method="post" enctype="multipart/form-data">
    ...
    <div class="custom-file mb-3">
        <input id="image" name="image" type="file" class="custom-file-input">
        <label class="custom-file-label" for="image">Seleccionar una imagen</label>
    </div>
    ...
</form>
```

Nota: este campo es poco personalizable en cuanto a su aspecto. Cada browser / sistema operativo lo muestra de forma diferente. Pero hay soluciones, por ejemplo: <https://getbootstrap.com/docs/4.4/components/forms/#file-browser>.



File Uploads (3/4) – Formidable

Para agilizar el trabajo de procesar un archivo recibido, utilizaremos un módulo llamado [formidable](#). Para instalarlo, ejecutar el comando:

```
npm i formidable
```

Luego importarlo con:

```
const formidable = require("formidable");
```



File Uploads (4/4) – Ejemplo de uso

```
app.post("/articulos", (req, res) => {  
  
  const form = formidable({  
    multiples: true,  
    uploadDir: __dirname + "/public/img",  
    keepExtensions: true,  
  });  
  
  form.parse(req, (err, fields, files) => {  
    res.redirect("articulos");  
  });  
});
```




Variables de Entorno



Variables de entorno (1/3)

Las aplicaciones que creamos suelen correr en distintos **entornos** (*environments*).
Por ejemplo:

- Nuestra computadora.
- La computadora de los desarrolladores miembros del equipo.
- Un servidor de testing.
- Un servidor de producción.
- Etc.



Variables de entorno (2/3)

Para cada uno de los distintos entornos, existen variables de configuración distintas (relativas a cada entorno) como, por ejemplo:

- Puerto donde correrá la aplicación.
- Ubicación de la base de datos.
- Usuario / Contraseña de la base de datos.
- Puerto de la base de datos.
- API Keys de servicios externos.
- Etc.

Variables de entorno



Variables de entorno (3/3)

Dado que los datos mencionados dependen de cada entorno y dado que en algunos casos se trata de datos extremadamente confidenciales, no es buena idea que los mismos formen parte del código fuente de la aplicación, que al formar parte del repositorio (Git) queda compartido con todo el equipo de desarrolladores.

👉 Los variables de entorno jamás se deben compartir ni quedar públicas.

Node.js tiene un objeto global llamado `process.env` que justamente tiene el propósito de guardar variables de entorno. ¿Pero cómo agregamos datos a dicho objeto sin que queden formando parte del código fuente?



Dotenv



Dotenv (1/4)

Existe un módulo llamado [dotenv](#) que permite levantar variables de entorno desde un archivo llamado `.env`. Para instalar el módulo se utiliza siguiente comando:

```
npm i dotenv
```

Luego, se crea el archivo `.env` en la raíz del proyecto y finalmente se levanta desde la aplicación:

```
require("dotenv").config();
```



Dotenv (2/4)

Un archivo `.env` es un archivo de texto muy simple, con la siguiente estructura:

```
APP_PORT=3000
APP_DOMAIN=ha.edu.uy
APP_NAME="Hack Academy Blog"

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=ha_blog
DB_USERNAME=root
DB_PASSWORD=root
```

Cada entorno tendrá su propio archivo `.env`, el cual nunca deberá incluirse en el repositorio.



Dotenv (3/4)

Finalmente, para hacer uso de las variables de entorno dentro de la aplicación se llama al objeto global llamado `process.env`

```
process.env.APP_PORT
```

👉 Dotenv se encarga de levantar las variables definidas en el archivo `.env` y colocarlas dentro de `process.env`.



Dotenv (4/4)

Comentarios adicionales:

- El archivo `.env` jamás se debe compartir en el repositorio (dado que contiene datos que dependen de cada entorno y algunos pueden ser confidenciales). Por eso, lo aconsejable es colocar `“.env”` el archivo en el `.gitignore`.
- Cuando se trabaja en equipo, una buena práctica es compartir un archivo `.env.example` con valores de ejemplo de forma tal que otros desarrolladores puedan conocer cuáles son las variables que se deben configurar, pero sin exponer datos confidenciales. Este archivo sí puede formar parte del repositorio.