

# Lección 5: Carga de datos desde archivos .json

```
3 require File.expand_path("../support/spec_helper.rb", __FILE__)
4 # Prevent database truncation if the environment is production
5 abort("The Rails environment is running in production")
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'capybara/rspec'
10 require 'capybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create(:name => "Categoría")
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |with|
16     with.test_framework :rspec
17     with.library :rails
18   end
19 end
20
21 # Add additional requires below this line
22 # Requires supporting ruby files with spec/
23 # spec/support/ and its subdirectories
24 # in _spec.rb will both be required
25 # run twice. It is recommended to
26 # end with _spec.rb. You can control
27 # this behavior with the --no-color
28 # option on the command line.
29
30 # Add additional requires below this line
31 # Requires supporting ruby files with spec/
32 # spec/support/ and its subdirectories
33 # in _spec.rb will both be required
34 # run twice. It is recommended to
35 # end with _spec.rb. You can control
36 # this behavior with the --no-color
37 # option on the command line.
```

**Tiempo de ejecución: 4 horas**  
**Planteamiento de la sesión**



## Materiales

- [multer - npm](#)
- [https://github.com/daduke/innovador\\_project](https://github.com/daduke/innovador_project)

Vamos a aplicar todo lo aprendido para crear un nuevo libro. Lo que haremos es agregar al final del archivo books.json un nuevo libro con los mismos campos de los ya creados, el reto está en varios puntos, el primero es cargar la portada del libro, ya que necesitamos de una herramienta como multer que permite configurar el directorio de carga de archivos, el segundo está en sólo permitir la creación de nuevos libros a los usuarios con roles admin y editor, el tercero está en cómo agregar ese nuevo libro al final del archivo teniendo en cuenta el formato del json.

Vamos a trabajar uno por uno para dividir el problema en pequeñas partes ya que cada uno de los puntos listados va a requerir la creación de vistas, rutas y métodos en los controladores.

Vamos a comenzar entonces configurando multer y probándolo en un formulario cualquiera que permita registrar una descripción y una imagen, con esto comprobaremos que podemos procesar la petición correctamente.

El siguiente paso es crear una vista protegida dentro de la ruta /admin que liste los libros y que cuente con un botón para agregar uno nuevo, este botón nos debe desplegar una nueva vista con un formulario para la creación del libro y si todo es correcto entonces debe aparecer en el listado de libros el nuevo agregado al final.

La solución de esta lección está disponible en la rama step6

## }Instalación y configuración de Multer

Multer es un paquete de npm que se utiliza para procesar archivos del lado del servidor. Este paquete es usado generalmente como un middleware y no solo nos va a permitir subir archivos al servidor, sino que también nos permite elegir en qué ubicación del servidor queremos que se almacenen estos archivos y el nombre con el que se guardará cada uno de ellos.

Para instalar este paquete la documentación oficial de npm nos da una guía [multer - npm](#)

```
npm install --save multer
```

Y justo después de la instalación, la documentación indica que para poder agregar la carga de archivos a los formularios, es necesario incluir el enctype:

```
<form action="/profile" method="post" enctype="multipart/form-data">
  <input type="file" name="avatar" />
</form>
```

Algo que utilizaremos más adelante.

Ahora viene la configuración en la ruta en la que necesitemos procesar los archivos, por ahora vamos a crear una ruta dentro de /admin por ejemplo /admin/books para listar los libros y /admin/books/create para crear un nuevo libro, es allí entonces, en el enrutador de admin, donde vamos a implementar multer.

Tengamos en cuenta que el multer es un middleware, podríamos perfectamente generar un middleware llamado multerBooks.js para el manejo de la carga de archivos.

Vamos primero a crear la ruta que permita listar los libros es decir /admin/books, para ello vamos a organizar un poco las vistas, vamos a crear una carpeta para los partials, componentes que compartirán las vistas y una carpeta de admin para agregar allí las vistas que correspondan al administrador, recuerden que al modificar la ruta, debo modificar los métodos render para que todo siga funcionando bien.

Con las carpetas y las vistas ajustadas vamos a crear la vista de /admin/books, recuerden que como tenemos un array de objetos, vamos a necesitar los métodos `forEach()`, `Object.values()` y `Object.keys()`, con lo que podremos recorrer objetos iterables, acceder a los valores de objetos literales y acceder a las llaves de los objetos literales respectivamente.

La ruta se vería algo así:

```
router.get('/books', adminMiddleware, AdminController.books);
```

El método en el controlador es el mismo del utilizado en el index, se vería así:

```
books: (req, res) => {
  //get the books.json file path
  const booksFilePath = path.join(__dirname, '../data/books.json');
  //read the books.json file
  let data = fs.readFileSync(path.join('data', 'books.json'), 'utf8');
  //parse the books data
  const books = JSON.parse(data);
  //render the books view and pass the books data to the view
  res.render('admin/admin-books', { title: 'Admin Books Page', books,
active: 'books'});
}
```

La vista creada admin-books, ahora se carga desde la carpeta de vistas admin y se ve así:

```
<table class="table table-striped">
  <thead>
    <tr>
      <th scope="col">#</th>
      <% Object.keys(books[0]).forEach(header => { %>
        <th scope="col"><%= header %></th>
      <% }) %>
    </tr>
  </thead>
```

```

<tbody>
  <% books.forEach((book, index) => { %>
    <tr>
      <th scope="row"><%= index + 1 %></th>
      <% Object.values(book).forEach(value => { %>
        <td><%= value %></td>
      <% }) %>
    </tr>
  <% }) %>

</tbody>
</table>

```

Agregando un poco de estilos y un botón para crear libro, la página se renderiza así:

#	title	author	year	cover	published	quantity	genre	price	synopsis
1	The Silent Patient	Alex Michaelides	2019	silent_patient_cover.webp	Orion Publishing Co	10	Psychological Thriller	15.99	Alicia Berenson's life is seemingly perfect. That is until she shoots her husband in the face and never speaks another word.
2	Where the Crawdads Sing	Delia Owens	2018	where_crawdads_sing_cover.webp	G.P. Putnam's Sons	10	Coming-of-Age Fiction	12.99	Kya Clark, known as the Marsh Girl, has survived alone in the marshes of North Carolina for years. But when a murder occurs, she becomes a suspect.
3	Becoming	Michelle Obama	2018	becoming_cover.webp	Crown Publishing Group	10	Autobiography	13.99	Michelle Obama invites readers into her world, chronicling the experiences that have shaped her, from her childhood in Chicago to her time as First Lady of the United States.
4	Educated	Tara Westover	2018	educated_cover.webp	Random House	10	Memoir	14.99	Tara Westover recounts her journey from being raised in a strict and abusive household in rural Idaho to earning a PhD from Cambridge University.
5	The Testaments	Margaret Atwood	2019	testaments_cover.webp	Nan A. Talese	10	Dystopian Fiction	17.99	Set 15 years after the events of The Handmaid's Tale, this novel explores the inner workings of Gilead and the lives of three women.
6	The Institute	Stephen King	2019	institute_cover.webp	Scribner	10	Horror Fiction	16.99	A group of children with special abilities are held captive in an institution where they are subjected to sinister experiments.
7	Normal People	Sally Rooney	2018	normal_people_cover.webp	Hogarth Press	10	Literary Fiction	13.99	The complex relationship between Marianne and Connell unfolds over the course of their school years in a small Irish town.
8	Circe	Madeline Miller	2018	circe_cover.webp	Little, Brown and Company	10	Mythological Fiction	11.99	The story of the mythical witch Circe, who is banished to a deserted island and must find her place in the world of gods and mortals.
9	The Girls of Stars	Inji Meyer	2019	girls_of_stars_cover.webp	Damla Doman Books	10	Historical Fiction	15.99	Set in Depression-era America, five women join a traveling library and embark on a life-changing journey.

Ahora vamos a configurar Multer como middleware e incluirlo en la ruta de admin

```
const multer = require('multer');
const path = require('path');

const storage = multer.diskStorage({
  destination: (req, file, cb)=>{
    cb(null, path.join(__dirname + '/../public/images/books'));
  },

  filename: (req, file, cb)=>{
    cb(null, req.body.title + '_' + 'cover' + path.extname(file.originalname));
  }
})

const upload = multer({storage : storage});

module.exports = upload;
```

Destination es la ruta donde se almacenarán los archivos y filename es el nombre que le asignaremos.

## Creando un elemento

Queremos crear un nuevo libro con los datos del siguiente objeto

```
{
  "title": "Dune",
  "author": "Frank Herbert",
  "year": "1965",
  "cover": "dune_cover",
  "published": "Ace Books",
  "quantity": "15",
  "genre": "Science Fiction",
  "price": "10.99",
  "synopsis": "A complex tale of politics, religion, and interstellar travel."
},
```

Para ello entonces en la ruta `/admin/books/create` debemos cargar un formulario de tal forma que el atributo `name` en los `inputs` correspondan con los campos que utilizamos para procesar el objeto, también revisemos el tipo de objeto. Las ruta para cargar el formulario entonces es:

```
router.get('/books/create', adminMiddleware, AdminController.createBook);
```

El método `create book` en controlador solo hace la renderización de la vista que contiene el formulario para la creación del libro:

```
createBook: (req, res) => {
  //render the create book view
  res.render('admin/create-book', { title: 'Create Book' });
},
```

El formulario debe contar con un `action` dirigido a `/admin/books/create` utilizando el `method POST`, también el `enctype` como se presentó al inicio de la documentación:

```
<form action="/admin/books/create" enctype="multipart/form-data"
method="post">
```

dentro del `form` vamos a crear todos los campos necesarios y el formulario se nos debería ver similar al siguiente:

The screenshot shows a web browser window with the URL `localhost:3000/admin/books/create`. The page is titled 'Bookstore Admin' and has a sidebar with navigation links: 'Inicio', 'Gestión de Libros', and 'Iniciar Bookstore'. A 'Logout' button is visible in the sidebar. The main content area contains a form with the following fields:

- Título**: Text input field.
- Autor**: Text input field.
- Año de publicación**: Text input field.
- Portada**: File upload field with a button labeled 'Seleccionar...' and a message 'No se ha seleccionado ningún archivo.'
- Cantidad**: Text input field.
- Genero**: Text input field.
- Precio**: Text input field.
- Sinopsis**: Text input field.

At the bottom of the form is a green button labeled 'Crear'.

Recuerde que el código completo lo puede encontrar en la rama step6.

Ahora ya tenemos un formulario listo para crear un nuevo libro, nos falta solo la ruta con el método post que reciba los datos para enviarlos al controlador correspondiente, procesar la info y agregar un nuevo elemento a la lista de libros. La ruta para crear requiere el multer y entonces se vería así:

```
const upload = require('../middlewares/multerBooks');
```

```
router.post('/books/create',      adminMiddleware,      upload.single('cover'),
AdminController.storeBook);
```

Y el método en el controlador quedaría de la siguiente manera:

```
storeBook: (req, res) => {
  //get the books.json file path
  const booksFilePath = path.join(__dirname, '../data/books.json');
  //read the books.json file
  let data = fs.readFileSync(path.join('data', 'books.json'), 'utf8');
  //parse the books data
  const books = JSON.parse(data);
  //get the book data from the form
  const book = req.body;
  //set the book cover file name
  book.cover = req.file.filename;

  //add the book to the books array
  books.push(book);
  //write the books data to the books.json file
  fs.writeFileSync(booksFilePath, JSON.stringify(books, null, 2));
  //redirect to the admin books page
  res.redirect('/admin/books');
}
```

Vamos entonces a probar si todo es correcto y se presenta un nuevo libro en la página principal.

Bookstore Admin

Inicio  
Gestión de Libros  
Ir a la Bookstore

Logout

Título  
Dune

Autor  
Frank Herbert

Año de publicación  
1965

Portada  
dune.jpg

Casa editorial  
Ace Books

Cantidad  
15

Género  
Science Fiction

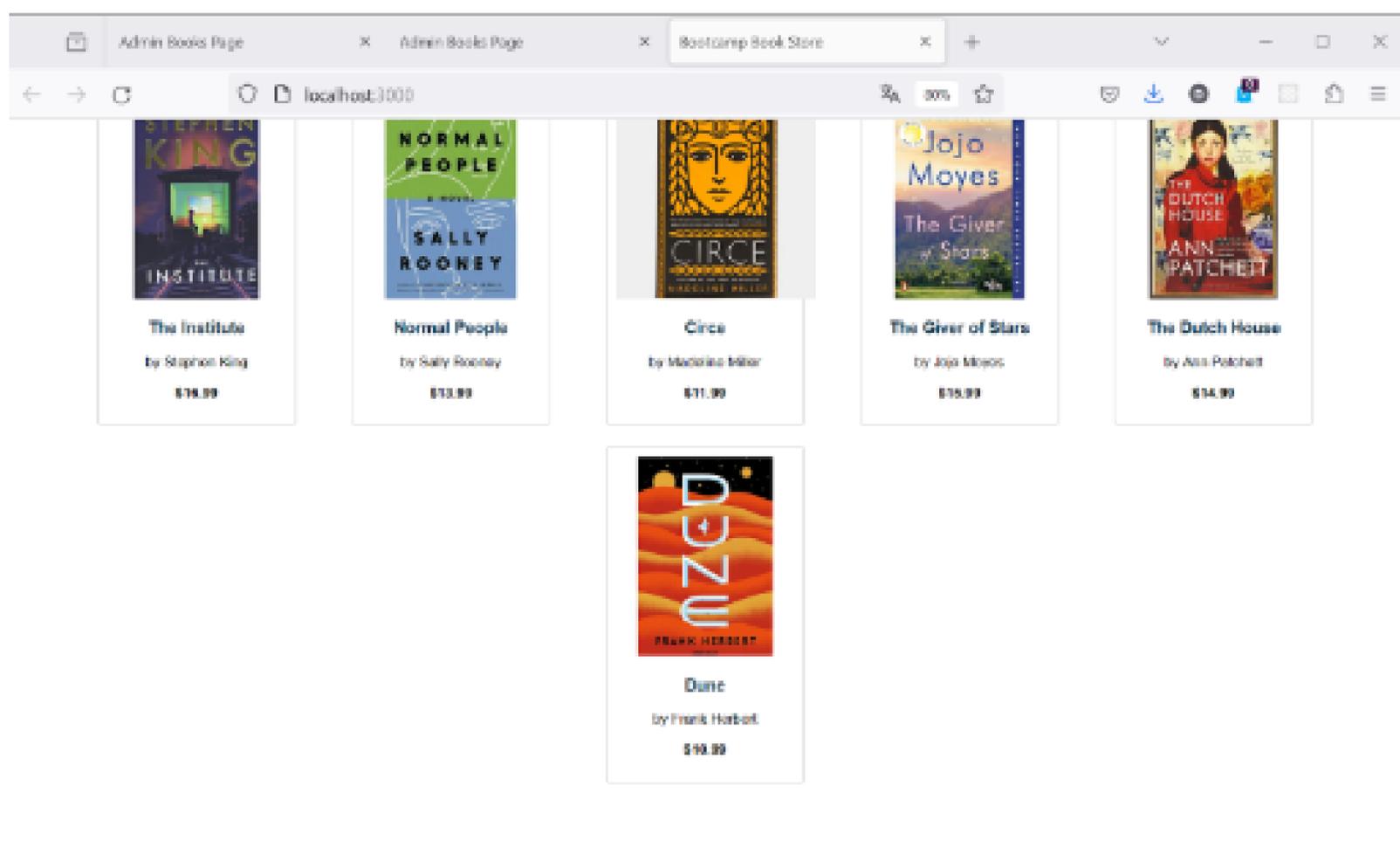
Precio  
10.99

Sinopsis  
A complex tale of politics, religion, and interstellar travel.

Crear

Si todo sale como esperamos deberíamos tener un nuevo libro en la tabla de admin y también en la vista principal

5	The Testaments	Margaret Atwood	2019	testaments_cover.webp	Ken A. Tolson	10	Dystopian Fiction	17.99	Set 15 years after the events of The Handmaid's Tale, this novel explores the inner workings of Gilead and the lives of three women.
6	The Institute	Stephen King	2019	institute_cover.webp	Scrivener	10	Horror Fiction	16.99	A group of children with special abilities are held captive in an institution where they are subjected to sinister experiments.
7	Normal People	Sally Rooney	2018	normal_people_cover.webp	Hogarth Press	10	Literary Fiction	13.99	The complex relationship between Marianne and Connell unfolds over the course of their school years in a small Irish town.
8	Circ	Madeline Miller	2018	circ_cover.webp	Little, Brown and Company	10	Mythological Fiction	11.99	The story of the mythical witch Circe, who is banished to a deserted island and must find her place in the world of gods and mortals.
9	The Glass of Stars	Jojo Moyes	2019	glass_of_stars_cover.webp	Parade Dragon Books	10	Historical Fiction	16.99	Set in Depression-era America, five women join a traveling library and embark on a life-changing journey.
10	The Dutch House	Aun Patchett	2019	dutch_house_cover.webp	Harper	10	Family Saga	14.99	The story of a brother and sister, their childhood home, and the bond that carries them through the decades.
11	Dune	Frank Herbert	1965	dune_cover.webp	Ace Books	10	Science Fiction	10.99	A complex tale of politics, religion, and interstellar travel.



Con esto hemos terminado la implementación de multer para la carga de un solo archivo, existe también la posibilidad de cargar varios archivos utilizando el atributo multiple en lo que debemos acceder al atributo files y no file como en el caso actual de un solo archivo.

Otro ajuste que se puede revisar es el de permitir solo unos tipos determinados de archivos.

Hasta aquí hemos logrado configurar multer y tener un proyecto que consume datos desde una fuente determinado, en este caso desde archivos JSON, este comportamiento se asemeja a lo que hacemos con una base de datos, en donde datos estructurados y no estructurados son almacenados para ser consumidos por una aplicación para determinado fin.

Es hora entonces de comenzar a revisar qué es eso de las bases de datos, la siguiente unidad, cubre lo que necesita saber de bases de datos y a utilizar sus comandos base, para luego adentrarnos en un ORM llamado sequelize que agregar una capa sobre el modelo para la gestión de la base de datos.

Recuerde que la solución hasta aquí se encuentra en la rama step6.