

Práctica 3

Autenticación de Usuarios y Gestión de Préstamos

Índice

Índice.....	2
1. Autenticación: Laravel Jetstream proporciona una implementación robusta de autenticación y manejo de usuarios. Para instalar Jetstream, utiliza composer require laravel/jetstream. Luego, utiliza php artisan jetstream:install livewire para generar la estructura necesaria.	3
2. Modelo de Usuarios: Laravel viene con un modelo de Usuarios por defecto. Necesitarás añadir una relación entre los usuarios y los préstamos (uno a muchos), donde un usuario puede tener múltiples préstamos.....	4
3. Controlador de Usuarios: Añade funciones en el controlador de Usuarios para que los usuarios puedan ver sus propios préstamos. No olvides proteger estas rutas para que solo el usuario autenticado pueda acceder a ellas.	7
4. Controlador de Préstamos: Modifica las funciones de creación y finalización de préstamos para que sólo los usuarios autenticados puedan crear o finalizar préstamos. Asegúrate de que el préstamo se asocie con el usuario autenticado.	10
5. Vistas de Usuarios y Préstamos: Crea vistas para que los usuarios puedan ver sus préstamos. Modifica las vistas de préstamos para que los usuarios puedan solicitar préstamos de libros una vez que estén autenticados.	10
6. Rutas: Asegúrate de que todas las rutas que permiten la creación, visualización o finalización de préstamos estén protegidas para que sólo los usuarios autenticados puedan acceder a ellas.....	11

1. Autenticación: Laravel Jetstream proporciona una implementación robusta de autenticación y manejo de usuarios. Para instalar Jetstream, utiliza `composer require laravel/jetstream`. Luego, utiliza `php artisan jetstream:install livewire` para generar la estructura necesaria.

Se ejecuta en terminal: **composer require laravel/jetstream**

Para instalar Livewire con Teams:

```
php artisan jetstream:install livewire --teams
```

Y para el Dark Mode:

```
php artisan jetstream:install livewire --dark
```

Se finaliza con:

npm install

npm run build

php artisan migrate

El **Log in** y **Register**, por defecto, vuelca en la vista de `welcome.blade.php`, pero para el caso de esta práctica, toda la información de los demás botones e interacciones se encuentran en el `home.blade.php`, por lo que se ha movido esa parte del código:

```
@if (Route::has('login'))
    <div class="sm:fixed sm:top-0 sm:right-0 p-6 text-right z-10">
        @auth
            <a href="{{ url('/dashboard') }}" class="font-semibold text-gray-600 hover:text-gray-900
            focus:outline focus:outline-2 focus:rounded-sm focus:outline-red-500">Dashboard</a>
        @else
            <a href="{{ route('login') }}" class="font-semibold text-gray-600 hover:text-gray-900
            focus:outline focus:outline-2 focus:rounded-sm focus:outline-red-500">Log in</a>

            @if (Route::has('register'))
                <a href="{{ route('register') }}" class="ml-4 font-semibold text-gray-600
                hover:text-gray-900 focus:outline focus:outline-2 focus:rounded-sm
                focus:outline-red-500">Register</a>
            @endif
        @endauth
    </div>
@endif
```

Quedando la vista de la url así:



2. Modelo de Usuarios: Laravel viene con un modelo de Usuarios por defecto. Necesitarás añadir una relación entre los usuarios y los préstamos (uno a muchos), donde un usuario puede tener múltiples préstamos.

```
//Un usuario tiene muchos prestamos
0 references | 0 overrides
public function prestamo(){
    return $this->hasMany(Prestamo::class);
}
```

```
//Un prestamo tiene un usuario
0 references | 0 overrides
public function user(){
    return $this->belongsTo(User::class);
}
```

Tengo que añadir, para poder hacer la relación, en la tabla de Préstamo el campo Usuario:

```
php artisan make:migration alter_prestamos_table
```

```
public function up(): void
{
    Schema::table('prestamos', function (Blueprint $table) {
        $table->unsignedBigInteger('user_id')->nullable();
        $table->foreign('user_id')->references('id')->on('users');
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::table('prestamos', function (Blueprint $table) {
        $table->dropForeign(['user_id']);
        $table->dropColumn('user_id');
    });
}
```

Se actualiza la función showPrestamos:

```
1 reference | 0 overrides
public function showPrestamos()
{
    // $allPrestamos = $this->prestamo->obtenerPrestamos();
    // return view('mostrarPrestamo', ['prestamos' => $allPrestamos]);
    $prestamos = Prestamo::with('user')->get();
    return view('mostrarPrestamo', ['prestamos' => $prestamos]);
}
```

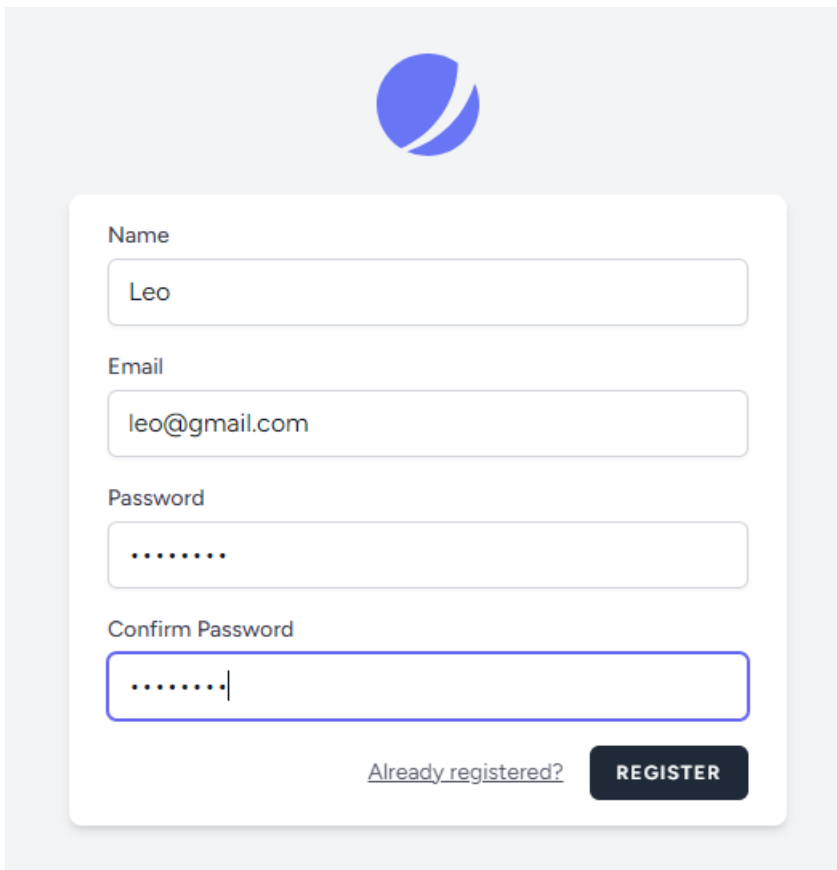
Y así quedaría la vista blade:

```
@extends('home')

@section('title', 'Mostrar Prestamos')

@section('content')
    @if (count ($prestamos) > 0)
        @foreach ($prestamos as $prestamo )
            <p> Usuario {{$prestamo->user->name}}</p>
            <p> Titulo {{$prestamo->titulo}}</p>
            <p> Fecha Préstamo {{$prestamo->fecha_prestamo}} </p>
            <p> Fecha Devolución {{$prestamo->fecha_devolucion}} </p>
            <p> Disponible {{$prestamo->disponible}} </p>
        @endforeach
    @endif
@endsection
```

Ahora se crea un usuario:



Registration form with the following fields:

- Name: Leo
- Email: leo@gmail.com
- Password: [masked]
- Confirm Password: [masked]

Buttons: [Already registered?](#) **REGISTER**

Y se nos vuelca:

SELECT * FROM `users`

Perfilando

Editar en línea

Editar

Explicar SQL

Crear código PHP

Actualizar

Mostrar todo

Número de filas:

25

Filtrar filas:

Buscar en esta tabla

Ordenar según la clave:

Ninguna

Opciones extra

			id	name	email	email_verified_at	password
<div><div></div><div>Editar</div><div>Copiar</div><div>Borrar</div></div>		1	Nerea	nerea@gmail.com	NULL	\$2y\$12\$OEJhaYiDUQvAkOnlIE2qOaQSmj6etWCOemdMW/et1t..	
<div><div></div><div>Editar</div><div>Copiar</div><div>Borrar</div></div>		2	Leo	leo@gmail.com	NULL	\$2y\$12\$qtXzc8pHu/2fMbit5nXSzu0MbAKZGRITDXZDABC5/LI...	

Si ahora se crea un préstamo, al mismo usuario le hemos podido asignar dos préstamos.

- PhpMyAdmin

```
SELECT * FROM `prestamos`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

```
UPDATE `prestamos` SET `user_id` = '1' WHERE `prestamos`.`id` = 2;
```

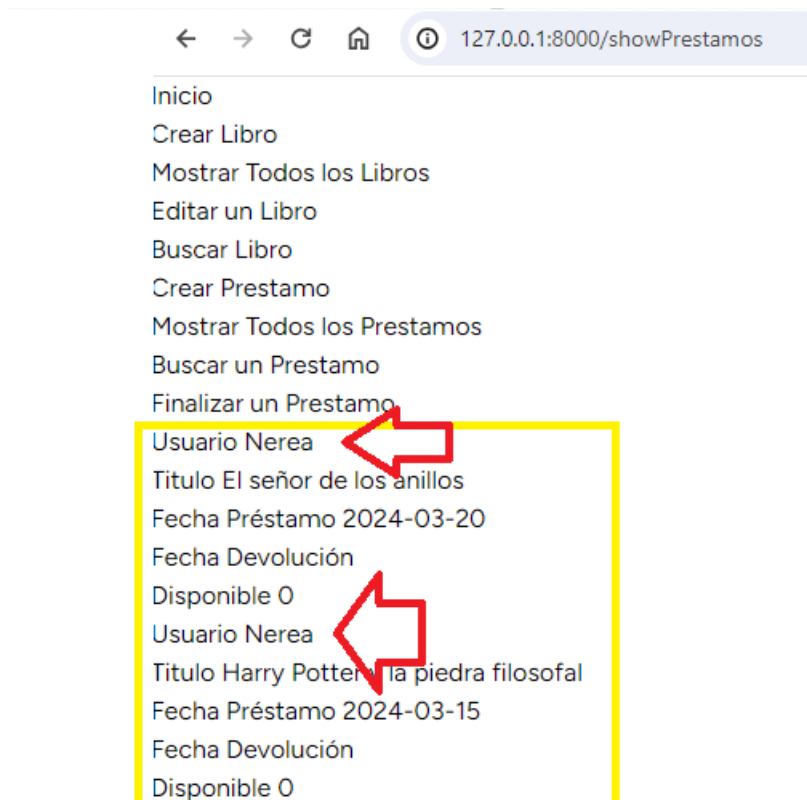
[Editar en línea] [Editar] [Crear código PHP]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

		id	titulo	fecha_prestamo	fecha_devolucion	disponible	created_at	updated_at	user_id
<input type="checkbox"/>	Editar		1	El señor de los anillos	2024-03-20	NULL	0 2024-03-31 16:27:31	2024-03-31 16:27:31	1
<input type="checkbox"/>	Editar		2	Harry Potter y la piedra filosofal	2024-03-15	NULL	0 2024-03-31 16:38:43	2024-03-31 16:38:43	1

- Navegador

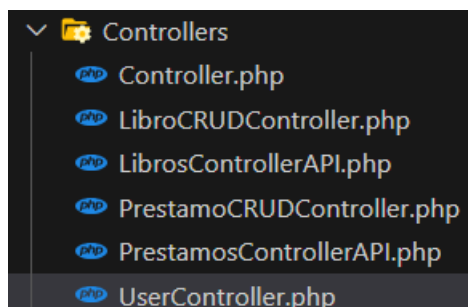


3. Controlador de Usuarios: Añade funciones en el controlador de Usuarios para que los usuarios puedan ver sus propios préstamos. No olvides proteger estas rutas para que solo el usuario autenticado pueda acceder a ellas.

Se comienza añadiendo la ruta en web.php:

```
Route::middleware(['auth'])->group(function () {  
    Route::get('/misPrestamos', [PrestamoWebController::class, 'misPrestamos'])->name('misPrestamos');  
});
```

Se crea el Controlador UserController:



0 references | 0 implementations

```
class UserController extends Controller
```

```
{
```

0 references | 0 overrides

```
public function mostrarMisPrestamos()
```

```
{
```

```
    // Obtener el usuario autenticado
```

```
    $user = Auth()->user();
```

```
    // Obtener los préstamos del usuario
```

```
    $prestamos = $user->prestamo;
```

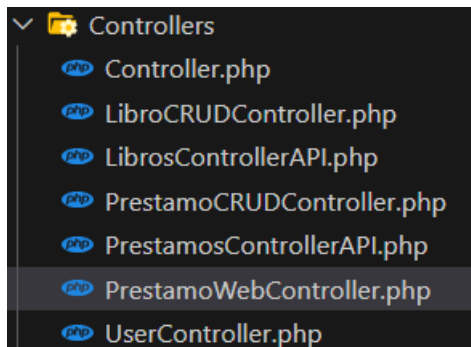
```
    // Devolver los préstamos del usuario a la vista
```

```
    return view('prestamos.index', ['prestamos' => $prestamos]);
```

```
}
```

```
}
```

Se crea un PrestamoWebController.php:



2 references | 0 implementations

```
class PrestamoWebController extends Controller
```

```
{
```

1 reference | 0 overrides

```
public function misPrestamos()
```

```
{
```

```
    // Obtener el usuario autenticado
```

```
    $user = Auth::user();
```

```
    // Obtener los préstamos asociados al usuario autenticado
```

```
    $prestamos = $user->prestamos;
```

```
    return view('misPrestamos', ['prestamos' => $prestamos]);
```

```
}
```

```
}
```


A continuación, se crea la vista misPrestamos.blade.php:

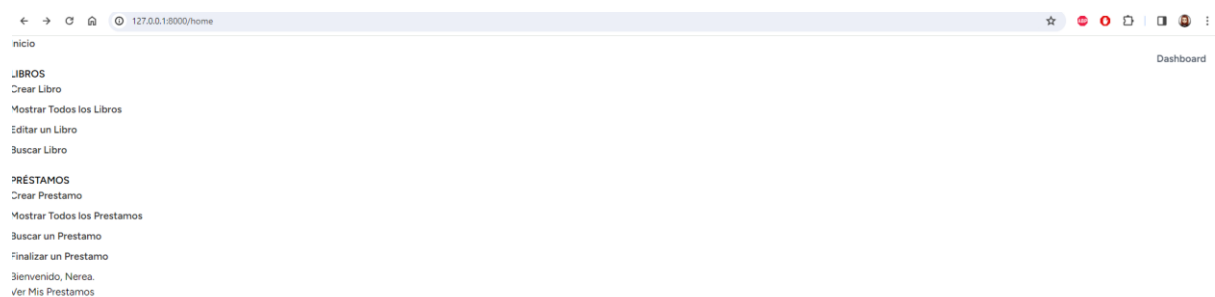
```
<body>
  <h1>Mis Préstamos</h1>
  <ul>
    {{-- Verifica si $prestamos está vacío --}}
    @if ($prestamos->isEmpty())
      <li>Este usuario no tiene préstamos</li>
    @else
      {{-- Itera sobre los préstamos --}}
      @foreach ($prestamos as $prestamo)
        <li>Titulo {{ $prestamo->titulo }}</li>
        <li>Fecha prestamo{{ $prestamo->fecha_prestamo }}</li>
      @endforeach
    @endif
  </ul>
</body>

</html>
```

Y la ruta en el home:

```
<div class="card-body">
  Bienvenido, {{ Auth::user()->name }}.
  <br>
  <a href="{{ route('misPrestamos') }}" class="btn btn-primary">Ver Mis Prestamos</a>
</div>
```

Quedando así el navegador, logueado con el usuario Nerea:



Cuando se hace click en el botón Ver Mis Prestamos:

PRÉSTAMOS

Crear Prestamo

Mostrar Todos los Prestamos

Buscar un Prestamo

Finalizar un Prestamo

Bienvenido, Nerea.

Ver Mis Prestamos



























Se obtiene:

Mis Préstamos

- Titulo El señor de los anillos
- Fecha prestamo 2024-03-20
- Titulo Harry Potter y la piedra filosofal
- Fecha prestamo 2024-03-15
- Titulo Alicia en el país de las maravillas
- Fecha prestamo 2024-03-03

Que se corresponde con lo que está en phpMyAdmin, ya que el user_id 1 de la tabla Prestamos corresponde con el id de Nerea en User:

			id	titulo	fecha_prestamo	fecha_devolucion	disponible	created_at	updated_at	user_id				
<input type="checkbox"/>		Editar		Copiar		Borrar	1	El señor de los anillos	2024-03-20	NULL	0	2024-03-31 16:27:31	2024-03-31 16:27:31	1
<input type="checkbox"/>		Editar		Copiar		Borrar	2	Harry Potter y la piedra filosofal	2024-03-15	NULL	0	2024-03-31 16:38:43	2024-03-31 16:38:43	1
<input type="checkbox"/>		Editar		Copiar		Borrar	3	El señor de los anillos	2024-04-01	NULL	0	2024-04-06 06:33:29	2024-04-06 06:33:29	2
<input type="checkbox"/>		Editar		Copiar		Borrar	4	Alicia en el país de las maravillas	2024-03-03	NULL	0	2024-04-06 06:40:22	2024-04-06 06:40:22	1
<input type="checkbox"/>		Editar		Copiar		Borrar	5	Don Quijote de la Mancha	2024-07-01	NULL	0	2024-04-07 09:53:16	2024-04-07 09:53:16	2

			id	name	email				
<input type="checkbox"/>		Editar		Copiar		Borrar	1	Nerea	nerea@gmail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	2	Leo	leo@gmail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	3	Sergio	sergio@gmail.com

4. Controlador de Préstamos: Modifica las funciones de creación y finalización de préstamos para que sólo los usuarios autenticados puedan crear o finalizar préstamos. Asegúrate de que el préstamo se asocie con el usuario autenticado.
5. Vistas de Usuarios y Préstamos: Crea vistas para que los usuarios puedan ver sus préstamos. Modifica las vistas de préstamos para que los usuarios puedan solicitar préstamos de libros una vez que estén autenticados.

6. Rutas: Asegúrate de que todas las rutas que permiten la creación, visualización o finalización de préstamos estén protegidas para que sólo los usuarios autenticados puedan acceder a ellas.

Los apartados 4,5 y 6 se van a realizar de forma conjunta:

- Creación de préstamos

Primero se ha modificado la home porque estaba diseñada que si no estabas autenticado no te dejaba entrar:

```
<div class="card-body">
  @auth
    Bienvenido, {{ Auth::user()->name }}.
    <br>
    <a href="{{ route('misPréstamos') }}" class="btn btn-primary">Ver Mis Préstamos</a>
  @else
    Usuario sin autenticar
  @endauth
</div>
```

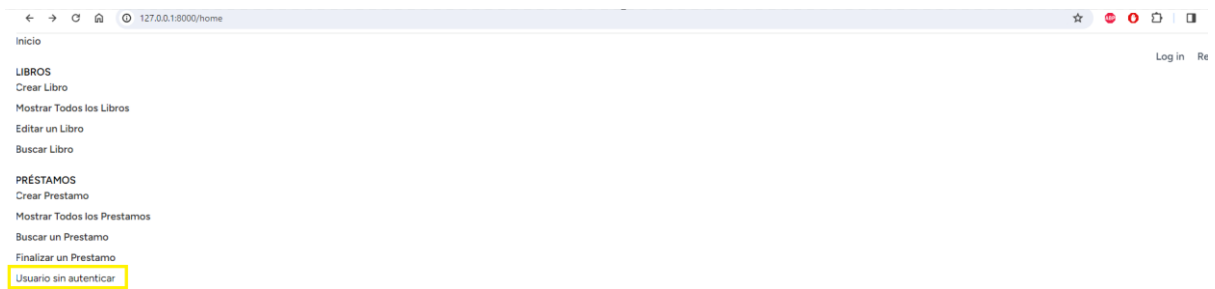
Luego se ha modificado PrestamoCRUDController:

```
0 references | 0 overrides
public function __construct(Prestamo $prestamo)
{
    $this->prestamo = $prestamo;
    $this->middleware('auth'); // Agrega autenticación a todas las funciones del controlador
}
```

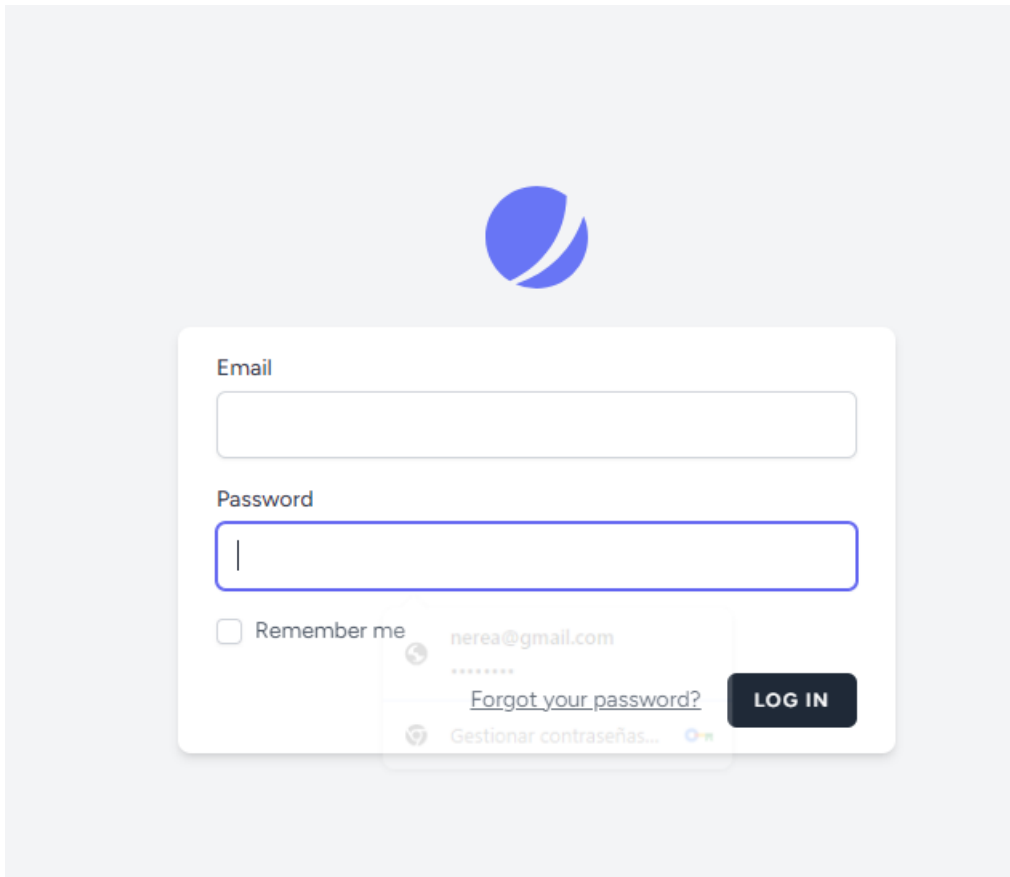
```
1 reference | 0 overrides
public function addPrestamo(Request $datosEnviados)
{
    $user = Auth::user(); // Obtener el usuario autenticado
    $prestamo = new Prestamo();
    $prestamo->titulo = $datosEnviados->input('titulo');
    $prestamo->fecha_prestamo = $datosEnviados->input('fechaPrestamo');
    $prestamo->user_id = $user->id; // Asociar el préstamo con el usuario autenticado
    $prestamo->save();

    return "Préstamo guardado con éxito";
}
```

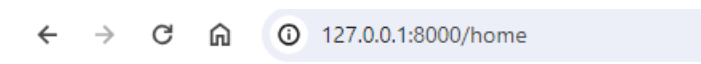
Quedando nuestro navegador:



Si estando sin autenticar, intentamos dar en el botón de “Crear Préstamo” nos manda directamente a que nos autenticemos:



Una vez que lo hemos realizado:



Inicio

LIBROS

Crear Libro

Mostrar Todos los Libros

Editar un Libro

Buscar Libro

PRÉSTAMOS

Crear Prestamo

Mostrar Todos los Prestamos

Buscar un Prestamo

Finalizar un Prestamo

Bienvenido, Nerea.

Ver Mis Prestamos

Ya nos deja crear el Prestamo:

Título Orgullo y prejuicio Fecha de Prestamo 07/04/2024 Guardar Prestamo



Prestamo guardado con éxito

Y se guarda automáticamente con el user_id que es Nerea:

SELECT * FROM `prestamos`

Perfilando

Editar en línea

Editar

Explicar SQL

Crear código PHP

Actualizar

Mostrar todo

Número de filas:

25

Filtrar filas:

Buscar en esta tabla

Ordenar según la clave:

Ninguna

Opciones extra

				id	titulo	fecha_prestamo	fecha_devolucion	disponible	created_at	updated_at	user_id
<input type="checkbox"/>				1	El señor de los anillos	2024-03-20	NULL	0	2024-03-31 16:27:31	2024-03-31 16:27:31	1
<input type="checkbox"/>				2	Harry Potter y la piedra filosofal	2024-03-15	NULL	0	2024-03-31 16:38:43	2024-03-31 16:38:43	1
<input type="checkbox"/>				3	El señor de los anillos	2024-04-01	NULL	0	2024-04-06 06:33:29	2024-04-06 06:33:29	2
<input type="checkbox"/>				4	Alicia en el país de las maravillas	2024-03-03	NULL	0	2024-04-06 06:40:22	2024-04-06 06:40:22	1
<input type="checkbox"/>				5	Don Quijote de la Mancha	2024-07-01	NULL	0	2024-04-07 09:53:16	2024-04-07 09:53:16	2
<input type="checkbox"/>				6	El Principito	2024-04-17	NULL	1	2024-04-07 15:42:06	2024-04-07 15:42:06	1
<input type="checkbox"/>				14	Señales y Sistemas	2024-04-17	NULL	1	2024-04-07 15:48:40	2024-04-07 15:48:40	1
<input type="checkbox"/>				15	Orgullo y prejuicio	2024-04-07	NULL	1	2024-04-07 16:02:00	2024-04-07 16:02:00	1

Y si se le da a "Ver mis Prestamos"

Mis Préstamos

- Titulo El señor de los anillos
- Fecha prestamo 2024-03-20
- Titulo Harry Potter y la piedra filosofal
- Fecha prestamo 2024-03-15
- Titulo Alicia en el país de las maravillas
- Fecha prestamo 2024-03-03
- Titulo El Principito
- Fecha prestamo 2024-04-17
- Titulo Señales y Sistemas
- Fecha prestamo 2024-04-17
- Titulo Orgullo y prejuicio
- Fecha prestamo 2024-04-07

- Finalización de préstamos

Se modifica la función en PrestamoCRUDController.php:

```
1 reference | 0 overrides
public function finalizarPrestamo(Request $datosEnviados)
{
    $prestamo = Prestamo::findOrFail($datosEnviados->input('id'));
    // Verificar si el préstamo pertenece al usuario autenticado
    if ($prestamo->user_id === Auth::id()) {
        $prestamo->fecha_devolucion = $datosEnviados->input('fechaDevolucion');
        $prestamo->save();
        return "Préstamo finalizado con éxito";
    } else {
        return "Error";
    }
}
```

La vista del navegador es igual que con la creación, si no está autenticado pasa lo mismo.

Ahora vamos a verificar que la función se comporta de forma correcta en los siguientes casos.

Esta es la tabla de Prestamos:

SELECT * FROM `prestamos`

Perfilando

[Editar en línea]

[Editar]

[Explicar SQL]

[Crear código PHP]

[Actualizar]

Mostrar todo

Número de filas:

25

Filtrar filas:

Buscar en esta tabla

Ordenar según la clave:

Ninguna

Opciones extra

				id	titulo	fecha_prestamo	fecha_devolucion	disponible	created_at	updated_at	user_id
<input type="checkbox"/>				1	El señor de los anillos	2024-03-20	2024-04-07	0	2024-03-31 16:27:31	2024-04-07 16:18:08	1
<input type="checkbox"/>				2	Harry Potter y la piedra filosofal	2024-03-15	NULL	0	2024-03-31 16:38:43	2024-03-31 16:38:43	1
<input type="checkbox"/>				4	Alicia en el país de las maravillas	2024-03-03	NULL	0	2024-04-06 06:40:22	2024-04-06 06:40:22	1
<input type="checkbox"/>				5	Don Quijote de la Mancha	2024-07-01	NULL	0	2024-04-07 09:53:16	2024-04-07 09:53:16	2
<input type="checkbox"/>				6	El Principito	2024-04-17	NULL	1	2024-04-07 15:42:06	2024-04-07 15:42:06	1
<input type="checkbox"/>				14	Señales y Sistemas	2024-04-17	NULL	1	2024-04-07 15:48:40	2024-04-07 15:48:40	1
<input type="checkbox"/>				15	Orgullo y prejuicio	2024-04-07	2024-04-08	1	2024-04-07 16:02:00	2024-04-07 16:13:03	1

Y esta la de usuarios:

`SELECT * FROM `users``

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: B

Opciones extra

	id	name	email
<input type="checkbox"/> Editar Copiar Borrar	1	Nerea	nerea@gmail.com
<input type="checkbox"/> Editar Copiar Borrar	2	Leo	leo@gmail.com
<input type="checkbox"/> Editar Copiar Borrar	3	Sergio	sergio@gmail.com

Se continúa trabajando con el usuario Nerea, con user_id: 1

- Supuesto de finalizar un préstamo del usuario que esta autenticado, en este caso Nerea, vamos a intentar finalizar el libro El Señor de los Anillos con ID 1:

ID 1 Título El señor de los anillos Fecha de Devolucion 07/04/2024 Guardar Devolucion

← → ↻ 🏠 ⓘ 127.0.0.1:8000/finalizarPrestamoPost

Prestamo finalizado con éxito

`SELECT * FROM `prestamos``

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

	id	titulo	fecha_prestamo	fecha_devolucion	disponible	created_at	updated_at	user_id
<input type="checkbox"/> Editar Copiar Borrar	1	El señor de los anillos	2024-03-20	2024-04-07	0	2024-03-31 16:27:31	2024-04-07 16:18:08	1
<input type="checkbox"/> Editar Copiar Borrar	2	Harry Potter y la piedra filosofal	2024-03-15	NULL	0	2024-03-31 16:38:43	2024-03-31 16:38:43	1
<input type="checkbox"/> Editar Copiar Borrar	4	Alicia en el país de las maravillas	2024-03-03	NULL	0	2024-04-06 06:40:22	2024-04-06 06:40:22	1
<input type="checkbox"/> Editar Copiar Borrar	5	Don Quijote de la Mancha	2024-07-01	NULL	0	2024-04-07 09:53:16	2024-04-07 09:53:16	2
<input type="checkbox"/> Editar Copiar Borrar	6	El Principito	2024-04-17	NULL	1	2024-04-07 15:42:06	2024-04-07 15:42:06	1
<input type="checkbox"/> Editar Copiar Borrar	14	Señales y Sistemas	2024-04-17	NULL	1	2024-04-07 15:48:40	2024-04-07 15:48:40	1
<input type="checkbox"/> Editar Copiar Borrar	15	Orgullo y prejuicio	2024-04-07	2024-04-08	1	2024-04-07 16:02:00	2024-04-07 16:13:03	1

- Supuesto que el usuario Nerea intenta finalizar el préstamo con id 5 El Quijote de la Mancha del user_id 2:

ID 5 Título Don Quijote de la Mancha Fecha de Devolucion 07/04/2024 Guardar Devolucion

← → ↻ 🏠 ⓘ 127.0.0.1:8000/finalizarPrestamoPost

Error

Y la tabla no se actualiza:

SELECT * FROM `prestamos`

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: | Ordenar según la clave: Ninguna

Opciones extra

				id	título	fecha_prestamo	fecha_devolucion	disponible	created_at	updated_at	user_id
<input type="checkbox"/>				1	El señor de los anillos	2024-03-20	2024-04-07	0	2024-03-31 16:27:31	2024-04-07 16:18:08	1
<input type="checkbox"/>				2	Harry Potter y la piedra filosofal	2024-03-15	NULL	0	2024-03-31 16:38:43	2024-03-31 16:38:43	1
<input type="checkbox"/>				4	Alicia en el país de las maravillas	2024-03-03	NULL	0	2024-04-06 06:40:22	2024-04-06 06:40:22	1
<input type="checkbox"/>				5	Don Quijote de la Mancha	2024-07-01	NULL	0	2024-04-07 09:53:16	2024-04-07 09:53:16	2
<input type="checkbox"/>				6	El Principito	2024-04-17	NULL	1	2024-04-07 15:42:06	2024-04-07 15:42:06	1
<input type="checkbox"/>				14	Señales y Sistemas	2024-04-17	NULL	1	2024-04-07 15:48:40	2024-04-07 15:48:40	1
<input type="checkbox"/>				15	Orgullo y prejuicio	2024-04-07	2024-04-08	1	2024-04-07 16:02:00	2024-04-07 16:13:03	1

NOTA

En esta práctica se han mejorado al final las vistas de las prácticas anteriores, ya que, anteriormente, se mostraba todo como extensión del menú home.

Adjunto capturas de las vistas mejoradas:

- Libros

← → ↻ 🏠 ⓘ 127.0.0.1:8000/addLibro

Crear libro

Título Autor Fecha de Publicacion 

← → ↻ 🏠 ⓘ 127.0.0.1:8000/showLibros

Libros disponibles

- Id 1
- Título El señor de los anillos
- Autor Tolkien
- Fecha Publicacion 1954-07-29
- Id 2
- Título Harry Potter y la piedra filosofal
- Autor J. K. Rowling
- Fecha Publicacion 1997-06-29

← → ↺ 🏠 ⓘ 127.0.0.1:8000/editLibro

Editar libro

ID Título Autor Fecha de Publicacion 📅

← → ↺ 🏠 ⓘ 127.0.0.1:8000/detailsLibro

Buscar un libro

Título

- Préstamos

← → ↺ 🏠 ⓘ 127.0.0.1:8000/addPrestamo

Crear Préstamo

Título Fecha de Prestamo 📅

← → ↺ 🏠 ⓘ 127.0.0.1:8000/detailsPrestamo

Buscar un préstamo

Título

← → ↺ 🏠 ⓘ 127.0.0.1:8000/finalizarPrestamo

Finalizar un préstamo

ID Título Fecha de Devolucion 📅

← → ↺ 🏠 ⓘ 127.0.0.1:8000/misPrestamos

Mis Préstamos

- Título El señor de los anillos
- Fecha prestamo2024-04-03
- Fecha devolución2024-04-13