

Atividade Prática 6.2 - Sistema de Empréstimo (simplificado)

Introdução

Nesta atividade, você implementará o sistema de empréstimos no projeto de biblioteca. Serão criadas rotas, um controlador dedicado (`BorrowingController`), e as funcionalidades de registrar empréstimos, devoluções e exibir históricos de empréstimos em livros e usuários.

Objetivo

O objetivo é implementar as operações necessárias para gerenciar os empréstimos no sistema.

- Criar rotas específicas para empréstimos e devoluções.
- Implementar métodos no controlador para gerenciar as operações de empréstimo e devolução.
- Atualizar as views de `Book` e `User` para exibir históricos de empréstimos.

Passo 1: Configurar as Rotas para Empréstimos e Devoluções

Adicione as rotas necessárias no arquivo `routes/web.php` para gerenciar os empréstimos de livros e devoluções, seguindo o padrão das atividades anteriores:

Rotas para a Funcionalidade de Empréstimos

```
use App\Http\Controllers\BorrowingController;

// Rota para registrar um empréstimo
Route::post('/books/{book}/borrow', [BorrowingController::class,
'store'])->name('books.borrow');

// Rota para listar o histórico de empréstimos de um usuário
Route::get('/users/{user}/borrowings', [BorrowingController::class,
'userBorrowings'])->name('users.borrowings');

// Rota para registrar a devolução
Route::patch('/borrowings/{borrowing}/return', [BorrowingController::class,
'returnBook'])->name('borrowings.return');
```

Passo 2: Criar o Controlador `BorrowingController`

Com as rotas definidas, o próximo passo será criar o controlador `BorrowingController` e implementar as funções necessárias para registrar empréstimos e devoluções. Além disso, ajustaremos as views de `Book` e `User` para incluir o histórico de empréstimos.

Crie o controlador utilizando o Artisan:

```
php artisan make:controller BorrowingController
```

No arquivo `app/Http/Controllers/BorrowingController.php` implemente os métodos necessários conforme descrito a seguir.

Primeiro inclua as classes que serão utilizadas no controller:

```
use App\Models\User;
use App\Models\Book;
use App\Models\Borrowing;
```

1. Método `store` para Registrar Empréstimos

Este método registra um novo empréstimo para o livro especificado:

```
public function store(Request $request, Book $book)
{
    $request->validate([
        'user_id' => 'required|exists:users,id',
    ]);

    Borrowing::create([
        'user_id' => $request->user_id,
        'book_id' => $book->id,
        'borrowed_at' => now(),
    ]);

    return redirect()->route('books.show', $book)->with('success', 'Empréstimo registrado com sucesso.');
}
```

2. Método `returnBook` para Registrar Devoluções

Este método atualiza a data de devolução de um empréstimo existente:

```
public function returnBook(Borrowing $borrowing)
{
    $borrowing->update([
```

```
    'returned_at' => now(),
]);

return redirect()->route('books.show', $borrowing->book_id)->with('success',
'Devolução registrada com sucesso.');
}
```

3. Método `userBorrowings` para Listar Empréstimos de um Usuário

Este método exibe o histórico de empréstimos de um usuário:

```
public function userBorrowings(User $user)
{
    $borrowings = $user->books()->withPivot('borrowed_at', 'returned_at')->get();

    return view('users.borrowings', compact('user', 'borrowings'));
}
```

Passo 3: Atualização da action show e books para trazer a lista de usuários

Como vamos fazer o empréstimo de livros na no books.show então precisamos pré-carregar a lista de usuários para o formulário de empréstimo na action show do BooksController.

Controller BookController

Atualize o método `show` no controller `BookController` para carregar todos usuários:

```
use App\Models\User; // incluir

public function show(Book $book)
{
    // Carregando autor, editora e categoria do Livro com eager Loading
    $book->load(['author', 'publisher', 'category']);

    // Carregar todos os usuários para o formulário de empréstimo
    $users = User::all();

    return view('books.show', compact('book', 'users'));
}
```

Passo 4: Atualização nos Modelos

Será necessária uma adaptação nos models de book e users, para que alguns links funcionem corretamente em eager loading.

Modelo Book

Atualize o método `users` no modelo `Book` para carregar os usuários associados com as informações do pivot:

```
public function users()
{
    return $this->belongsToMany(User::class, 'borrowings')
        ->withPivot('id', 'borrowed_at', 'returned_at')
        ->withTimestamps();
}
```

Modelo User

Atualize o método `books` no modelo `User` para carregar os livros associados com as informações do pivot:

```
public function books()
{
    return $this->belongsToMany(Book::class, 'borrowings')
        ->withPivot('id', 'borrowed_at', 'returned_at')
        ->withTimestamps();
}
```

Passo 5: Ajustar as Views

1. View `books.show`

Atualize a view para exibir o histórico de empréstimos de um livro e incluir a funcionalidade de registrar empréstimos. (adapte a versão anterior para acrescentar o formulário de empréstimo e o histórico de empréstimo)

```
<!-- Formulário para Empréstimos -->
<div class="card mb-4">
    <div class="card-header">Registrar Empréstimo</div>
    <div class="card-body">
        <form action="{{ route('books.borrow', $book) }}" method="POST">
            @csrf
            <div class="mb-3">
                <label for="user_id" class="form-label">Usuário</label>
                <select class="form-select" id="user_id" name="user_id" required>
                    <option value="" selected>Selecione um usuário</option>
                    @foreach($users as $user)
                        <option value="{{ $user->id }}>{{ $user->name }}</option>
                    @endforeach
                </select>
            </div>
            <button type="submit" class="btn btn-success">Registrar
                Empréstimo</button>
        </form>
    </div>
</div>

<!-- Histórico de Empréstimos -->
<div class="card">
    <div class="card-header">Histórico de Empréstimos</div>
    <div class="card-body">
        @if($book->users->isEmpty())
            <p>Nenhum empréstimo registrado.</p>
        @else
            <table class="table">
                <thead>
                    <tr>
                        <th>Usuário</th>
                        <th>Data de Empréstimo</th>
                        <th>Data de Devolução</th>
                        <th>Ações</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach($book->users as $user)
                        <tr>
                            <td>
                                <a href="{{ route('users.show', $user->id) }}">
```

```

        {{ $user->name }}
    </a>
</td>
<td>{{ $user->pivot->borrowed_at }}</td>
<td>{{ $user->pivot->returned_at ?? 'Em Aberto' }}</td>
<td>
    @if(is_null($user->pivot->returned_at))
        <form action="{{ route('borrowings.return', $user->pivot->id) }}" method="POST">
            @csrf
            @method('PATCH')
            <button class="btn btn-warning btn-sm">Devolver</button>
        </form>
    @endif
</td>
</tr>
@endforeach
</tbody>
</table>
@endif
</div>
</div>

```

2. View `users.show`

Atualize a view para exibir o histórico de empréstimos de um usuário:

```

<!-- Histórico de Empréstimos -->
<div class="card">
    <div class="card-header">Histórico de Empréstimos</div>
    <div class="card-body">
        @if($user->books->isEmpty())
            <p>Este usuário não possui empréstimos registrados.</p>
        @else
            <table class="table">
                <thead>
                    <tr>
                        <th>Livro</th>
                        <th>Data de Empréstimo</th>
                        <th>Data de Devolução</th>
                        <th>Ações</th>
                    </tr>
                </thead>
                <tbody>
@foreach($user->books as $book)
                    <tr>

```

```
<td>
    <a href="{{ route('books.show', $book->id) }}>
        {{ $book->title }}
    </a>
</td>
<td>{{ $book->pivot->borrowed_at }}</td>
<td>{{ $book->pivot->returned_at ?? 'Em Aberto' }}</td>
<td>
    @if(is_null($book->pivot->returned_at))
        <form action="{{ route('borrowings.return', $book->pivot->id) }}"
method="POST">
            @csrf
            @method('PATCH')
            <button class="btn btn-warning btn-sm">Devolver</button>
        </form>
    @endif
</td>
</tr>
@endforeach
</tbody>

</table>
@endif
</div>
</div>
```

Passo 6: Testar o Sistema

Certifique-se de que o projeto está rodando.

Teste as funcionalidades:

1. Acesse um livro e registre um empréstimo.
2. Verifique o histórico de empréstimos no `books.show` e `users.show`.
3. Registre devoluções e confirme as atualizações no sistema.