

Práctica 1: Instalación de Laravel y Creación de una Tabla de Libros

En esta práctica, se ha desarrollado un Sistema de Gestión de Biblioteca utilizando el framework Laravel de PHP. El sistema permite a los usuarios realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre libros y préstamos.

Descripción del código fuente:

- Se ha creado un modelo `Libro` para representar los libros en la biblioteca, con métodos para realizar operaciones CRUD.
- Se ha creado un modelo `Prestamo` para representar los préstamos de libros a usuarios, con relaciones definidas con los modelos `User` y `Libro`.

Ambos han sido creados por medio de los comandos `php artisan make:model <NombreControlador> -m` y completados posteriormente.

```

class Libro extends Model
{
    use HasFactory;
    protected $table = 'libros';
    protected $fillable = ['titulo', 'autor'];

    public static function getAll(){
        return Libro::all();
    }

    public static function getById($id){
        return Libro::find($id);
    }

    public static function getByTitle($titulo){
        return Libro::where('titulo', '=', '$titulo');
    }

    public static function saveLibro($titulo, $autor, $fecha_publicacion){
        $libro = new Libro;
        $libro->titulo = $titulo;
        $libro->autor = $autor;
        $libro->fecha_publicacion = $fecha_publicacion;
        $libro->save();
        return $libro->id;
    }

    public static function updateLibro($titulo, $autor, $fecha_publicacion){
        $libro = Libro::find($id);
        if( isset($libro) ){
            $libro->titulo = $titulo;
            $libro->autor = $autor;
            $libro->fecha_publicacion = $fecha_publicacion;
            $libro->save();
            return $libro->id;
        }
        return "Libro no encontrado";
    }

    public static function deleteLibro($id){
        $libro = Libro::find($id);
        if( isset($libro) ){
            $libro->delete();
        }
    }
}

```

```

7
8 class Prestamo extends Model
9 {
10     use HasFactory;
11
12     protected $table = 'prestamos';
13     protected $fillable = ['user', 'book_id', 'fecha_prestamo', 'fecha_devolucion'];
14
15     public function user()
16     {
17         return $this->belongsTo(User::class);
18     }
19
20     public function libro()
21     {
22         return $this->belongsTo(Libro::class, 'book_id');
23     }
24
25     public static function getAll(){
26         return Prestamo::all();
27     }
28
29     public static function getById($id){
30         return Prestamo::find($id);
31     }
32
33     public static function getByTitle($titulo){
34         return Prestamo::where('titulo', '=', '$titulo');
35     }
36
37     public static function savePrestamo($user, $book_id , $fecha_prestamo , $fecha_devolucion ){

```

Se han creado migraciones para las tablas `libros` y `prestamos`, definiendo su estructura y relaciones. En el caso de los préstamos, se tiene en cuenta la relación entre el libro y el préstamo.

```

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up()
    {
        Schema::create('libros', function (Blueprint $table) {
            $table->id();
            $table->string('titulo');
            $table->string('autor');
            $table->date('fecha_publicacion');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void

```

```

7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up()
13     {
14         Schema::create('prestamos', function (Blueprint $table) {
15             $table->id();
16             $table->string('user');
17             $table->unsignedBigInteger('book_id');
18             $table->date('fecha_prestamo');
19             $table->date('fecha_devolucion');
20             $table->timestamps();
21
22             $table->foreign('book_id')->references('id')->on('libros');
23         });
24     }
25     /**
26      * Reverse the migrations.
27      */

```

Por último, se ha usado el comando `php artisan migrate` para mantener la integridad y consistencia en la base de datos.

```

PS C:\Users\jorge\PROJECTS\masterFullStack\Laravel\p1\libreria> php artisan migrate

[WARN] The database 'laravel' does not exist on the 'mysql' connection.

Would you like to create it? (yes/no) [no]
> y

[INFO] Preparing database.

Creating migration table ..... 9ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 28ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 6ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 19ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 26ms DONE
2024_02_23_195115_create_libros_table ..... 6ms DONE

```

Para poder comprobar el funcionamiento, se han creado las vistas de añadir libro y añadir préstamo, como se puede ver el código fuente y en las siguientes capturas:

The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/addLibro`. The page title is "Añadir Libro". Below the title, there is a form with three input fields: "Título" (containing "Harry Potter y la piedra filosofal"), "Autor" (containing "J.K. Rowling"), and "Fecha publicacion" (containing "26/06/1997"). There is a calendar icon next to the date field. To the right of the date field is a "Guardar" (Save) button.

Opciones extra

			id	título	autor	fecha_publicacion	created_at	updated_at
<input type="checkbox"/>				3	Harry Potter y la piedra filosofal	J.K. Rowling	1997-06-26	2024-03-25 20:53:18
<input type="checkbox"/>				4	El Señor de los Anillos: La Comunidad del Anillo	J.R.R. Tolkien	1997-06-26	2024-03-25 20:54:09
<input type="checkbox"/>				5	Orgullo y prejuicio	Jane Austen	2024-03-07	2024-03-25 20:54:23
<input type="checkbox"/>				6	El principito	Antoine de Saint-Exupéry	2024-02-29	2024-03-25 20:54:43
<input type="checkbox"/>				7	Cien años de soledad	Gabriel García Márquez	2024-02-29	2024-03-25 20:54:55

☐ Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

← → ↺ 🏠 ⌚ 127.0.0.1:8000/addPrestamo

Añadir Prestamo

Usuario ID del Libro Fecha de Préstamo Fecha de Devolución

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	user	book_id	fecha_prestamo	fecha_devolucion	created_at	updated_at
<input type="checkbox"/>				2	Jorge Moreno	3	2024-03-28	2024-03-06	2024-03-25 20:55:34	2024-03-25 20:55:34
<input type="checkbox"/>				3	Antonio Martin	5	2024-03-28	2024-03-06	2024-03-25 20:56:13	2024-03-25 20:56:13
<input type="checkbox"/>				4	Cristiano Ronaldo	3	2024-03-28	2024-03-06	2024-03-25 20:56:26	2024-03-25 20:56:26

☐ Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar