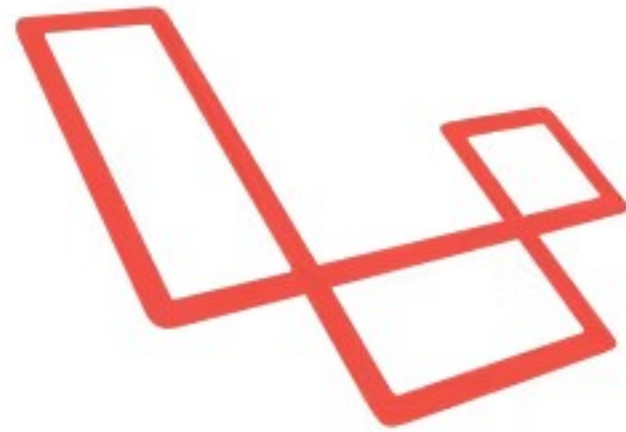


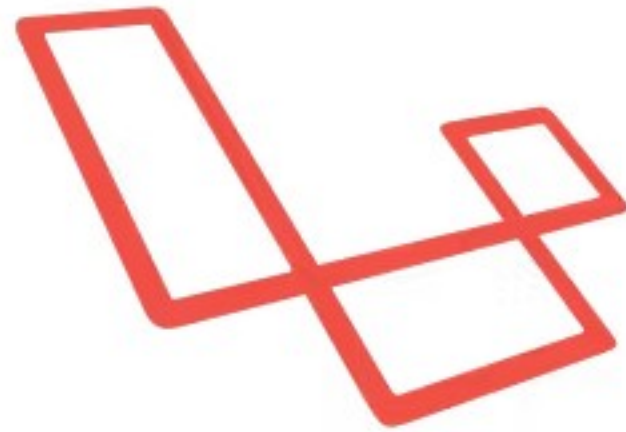
UD 3.2

Modelos en Laravel



Modelos en Laravel

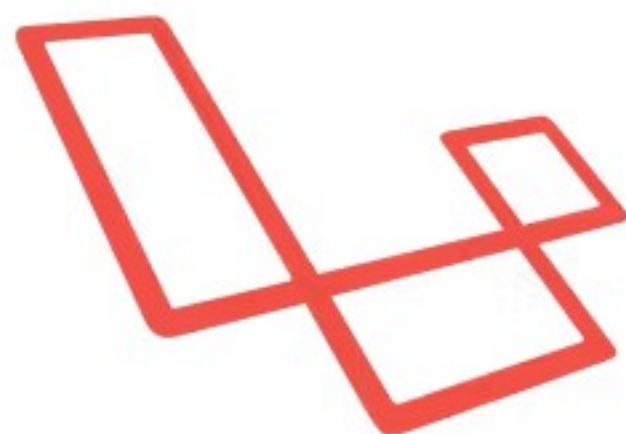
Un **modelo** en Laravel es una representación de una tabla en la base de datos. Cada modelo se asocia con una tabla y permite interactuar con sus registros mediante el ORM Eloquent. Los modelos facilitan realizar operaciones CRUD sin necesidad de escribir SQL manualmente.



1. ¿Qué es un modelo en Laravel?

- Es una clase PHP que representa una tabla en la base de datos.
- Define las operaciones que se pueden realizar en los datos de esa tabla (CRUD: Crear, Leer, Actualizar, Eliminar).
- Por defecto, cada modelo está asociado con una tabla con el nombre plural del modelo.

Ejemplo: Un modelo `Post` está vinculado a una tabla `posts` en la base de datos.



Creación de un Modelo

1. Comando para crear un modelo:

```
php artisan make:model Post
```

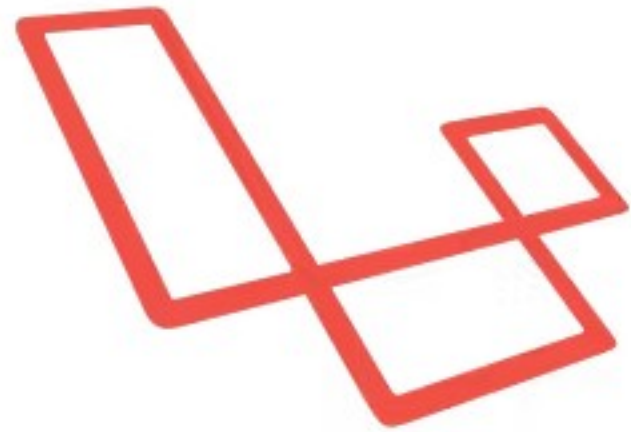
Esto genera el archivo del modelo en `app/Models/Post.php`.

2. **Crear un modelo con migración:** Añade el flag `-m` para generar el modelo junto con una migración:

```
php artisan make:model Post -m
```

Esto genera:

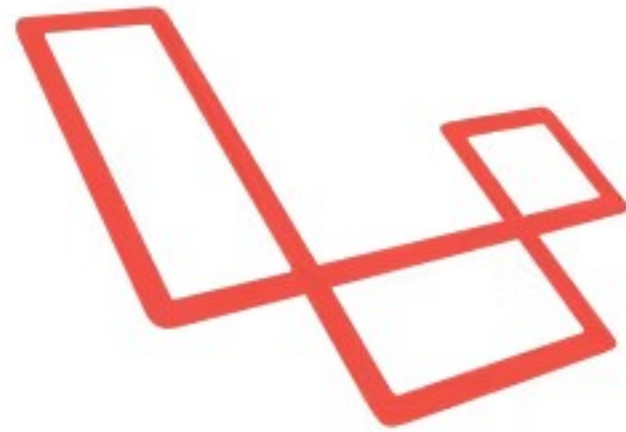
- Un modelo en `app/Models/Post.php`.
- Una migración en `database/migrations/XXXX_create_posts_table.php`.



Estructura de un Modelo

Un modelo básico tiene la siguiente estructura:

```
class Post extends Model {  
    protected $table = 'posts';  
}
```



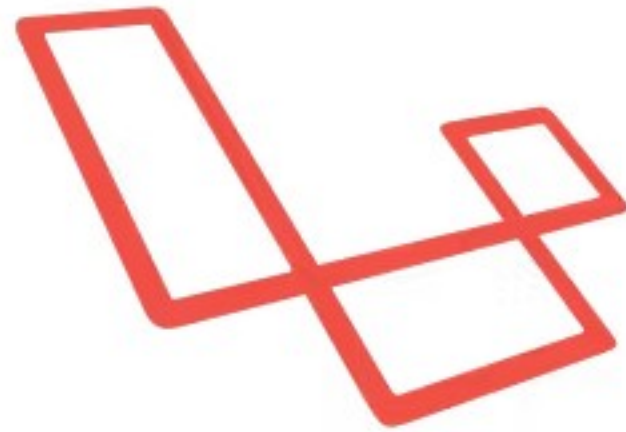
Campos importantes en los Modelos

1. \$fillable

Define qué columnas pueden ser rellenas masivamente al usar `Model::create()` o `$model->fill()`.

Ejemplo:

```
protected $fillable = ['titulo', 'contenido', 'user_id'];
```



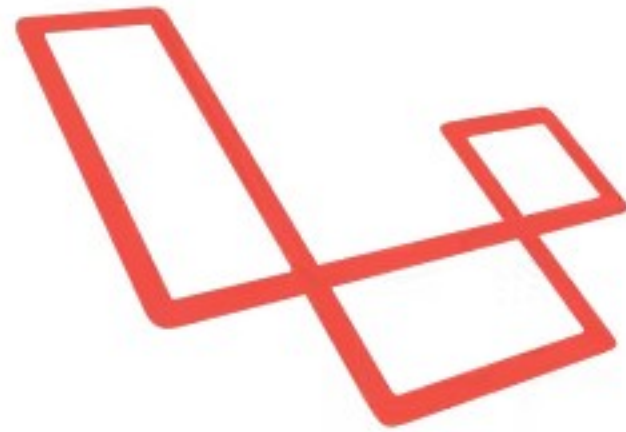
Campos importantes en los Modelos

2. \$guarded

Es lo contrario de `$fillable`. Indica qué columnas no pueden ser asignadas masivamente.

Ejemplo:

```
protected $guarded = ['id'];
```

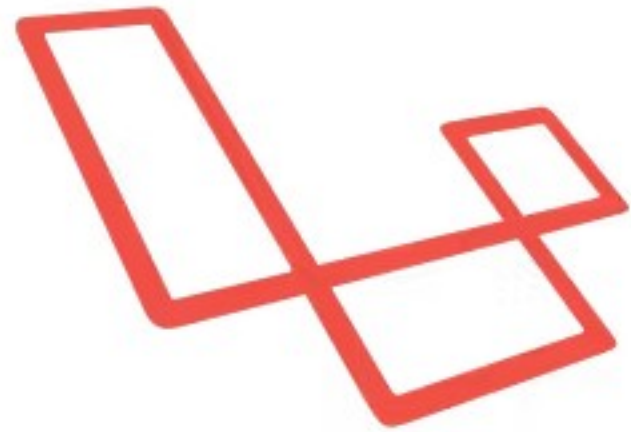


Campos importantes en los Modelos

3. \$table

Por defecto, Laravel asume que el nombre de la tabla es el plural del modelo. Si el nombre de la tabla es diferente, puedes definirlo:

```
protected $table = 'articulos';
```

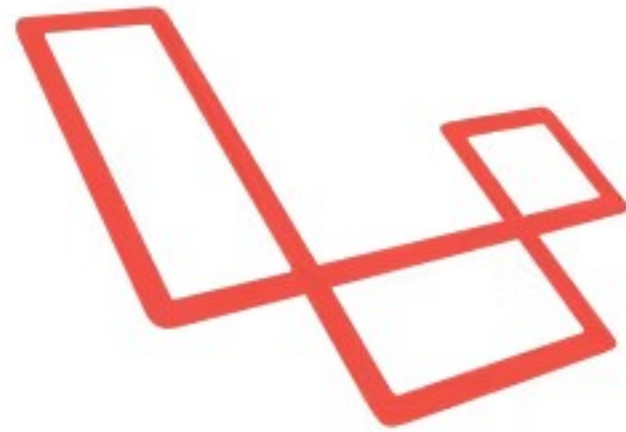



Campos importantes en los Modelos

4. \$primaryKey

Define la llave primaria de la tabla. Por defecto, es `id`.

```
protected $primaryKey = 'post_id';
```

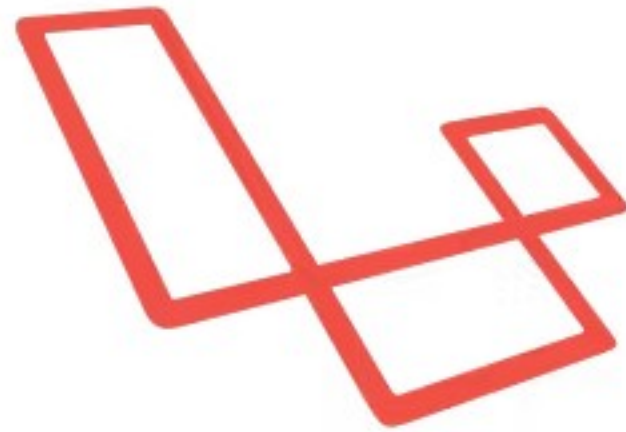


Campos importantes en los Modelos

5. \$timestamps

Indica si la tabla tiene las columnas `created_at` y `updated_at`. Por defecto, está habilitado.

```
public $timestamps = true; (por defecto) public  
$timestamps = false;
```



Operaciones con Modelos

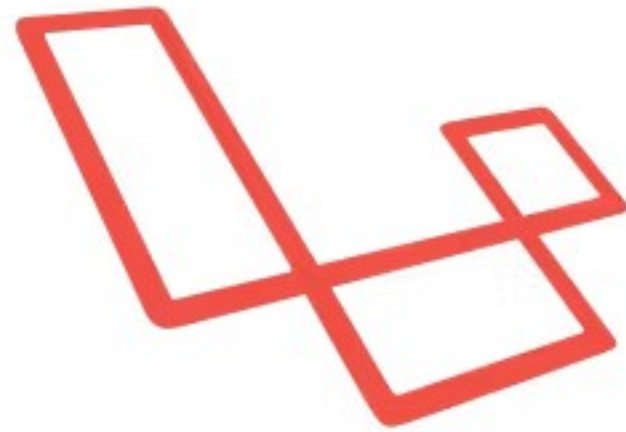
1. Crear un registro

- Usando el método `create()` (requiere `$fillable` configurado):

```
Post::create([  
    'titulo' => 'Mi primer post',  
    'contenido' => 'Este es el contenido del post.',  
    'user_id' => 1,  
]);
```

- Usando una instancia del modelo:

```
$post = new Post();  
$post->titulo = 'Mi segundo post';  
$post->contenido = 'Contenido del segundo post.';  
$post->user_id = 2;  
$post->save();
```

Operaciones con Modelos

2. Leer registros

- Obtener todos los registros:

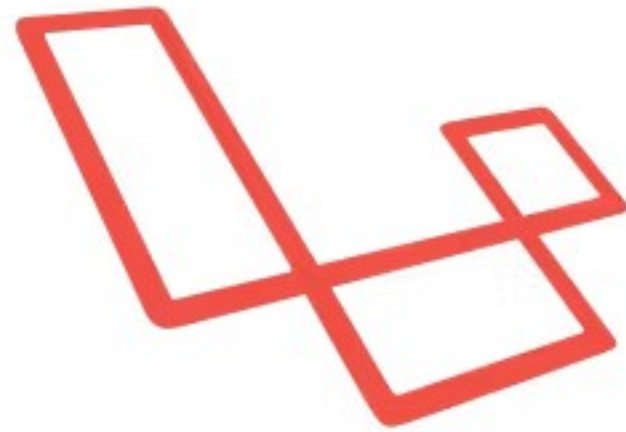
```
$posts = Post::all();
```

- Obtener un registro por ID:

```
$post = Post::find(1);
```

- Filtrar registros con where :

```
$posts = Post::where('user_id', 1)->get();
```

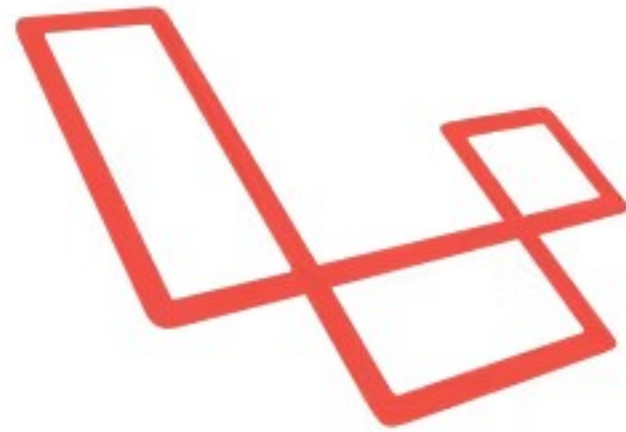



Operaciones con Modelos

3. Actualizar un registro

- Usando una instancia del modelo:

```
$post = Post::find(1);  
$post->contenido = 'Nuevo contenido  
actualizado.';  
$post->save();
```



Operaciones con Modelos

4. Eliminar un registro

- Usando una instancia del modelo:

```
$post = Post::find(1); $post->delete();
```

- Eliminación directa:

```
Post::destroy(1);
```

```
Post::where('user_id', 2)->delete();
```