

3.4 INYECCIÓN DE **DEPENDENCIAS**

Para realizar la inyección de dependencias, debemos incluir la clase a inyectar desde el propio constructor del controlador:

```
public function __construct(
    PostRepository $postRepository
)
{
    $this->postRepository = $postRepository;
}
```

3.4 INYECCIÓN DE DEPENDENCIAS

La clase `PostRepository` es la implementación del patrón repositorio con la que accederemos a los datos de la base de datos.

Su contenido estará dentro de `app\Repositories` con el nombre de archivo `PostRepository.php`

Explicaremos en el siguiente capítulo el acceso a datos. Por ahora simplemente devolveremos un array.

3.4 INYECCIÓN DE **DEPENDENCIAS**

```
namespace App\Repositories;
class PostRepository
{
    public function find($id)
    {
        $data = [
            '1' => 'Post number 1',
            '2' => 'Post number 2',
        ];
        return isset($data[$id]) ? $data[$id] : null;
    }
}
```

3.4 INYECCIÓN DE DEPENDENCIAS

El método show de `PostController` ya podrá hacer la llamada al repositorio ya que la dependencia está resuelta en el constructor:

```
public function show($id)
{
    $post = $this->postRepository->find($id);
    return "Showing post " . $post;
}
```