**Identificación:** Anexo07-Modelos  
**Versión:** 1.0

Tabla de contenido

[Introducción 2](#_Toc63111498)

[Para empezar, creemos un modelo Eloquent. Los modelos normalmente se encuentran en el directorio app\Models, pero puede colocarlos en cualquier lugar que pueda cargarse automáticamente de acuerdo con su archivo composer.json. Todos los modelos de Eloquent amplían la clase Illuminate\Database\Eloquent\Model. 2](#_Toc63111499)

[Ejercicio 3](#_Toc63111500)

[Recuperar Datos 3](#_Toc63111501)

[Agregando Condiciones Adicionales 3](#_Toc63111502)

[El método Eloquent all devolverá todos los resultados en la tabla del modelo. Dado que cada modelo de Eloquent sirve como generador de consultas, también puede agregar restricciones a las consultas y luego usar el método get para recuperar los resultados: 3](#_Toc63111503)

[En el ejemplo anterior, desde el controlador creamos una variable $users y le asignamos un modelo Users y le pasamos varios metodos 3](#_Toc63111504)

[orderBy → Organizamos de forma descendente de acuerdo al campo name 3](#_Toc63111505)

[take →devuelve una nueva colección con el número especificado de elementos es decir 10 3](#_Toc63111506)

[get → Recupera los datos. 3](#_Toc63111507)

[Bibliografia 3](#_Toc63111508)

# Introducción

## Para empezar, creemos un modelo Eloquent. Los modelos normalmente se encuentran en el directorio **app\Models**, pero puede colocarlos en cualquier lugar que pueda cargarse automáticamente de acuerdo con su archivo composer.json. Todos los modelos de Eloquent amplían la clase **Illuminate\Database\Eloquent\Model**.

Crear el modelo de la migración students

php artisan make:model students

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class students extends Model

{

protected $table = ‘students’;

protected $primaryKey = 'id';

protected $fillable = [

//Los campos de la base datos

];

}

# **Ejercicio**

## Recuperar Datos

Algo que debemos hacer continuamente en cada una de nuestras aplicaciones es recuperar y tratar datos.

### Agregando Condiciones Adicionales

## El método Eloquent all devolverá todos los resultados en la tabla del modelo. Dado que cada modelo de Eloquent sirve como generador de consultas, también puede agregar restricciones a las consultas y luego usar el método get para recuperar los resultados:

$students= App\Models\students::all()

->orderBy('name', 'desc')

->take(10)

->get();

## En el ejemplo anterior, desde el controlador creamos una variable $users y le asignamos un modelo Users y le pasamos varios metodos

## orderBy → Organizamos de forma descendente de acuerdo al campo name

## take →devuelve una nueva colección con el número especificado de elementos es decir 10

## get → Recupera los datos.

# **Bibliografia**

https://laravel.com/docs/8.x