

DEPARTAMENTO:	CIENCIAS DE LA COMPUTACIÓN	CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN		
ASIGNATURA:	Desarrollo Web para Integración	NIVEL:	6	FECHA:	16/05/2025
DOCENTE:	Ing. Pablo Puente	EXAMEN N°:	1	CALIFICACIÓN:	

Examen

Mejia Gaibor Anthony Geovanny

Creacion del composer y proyecto en Laravel

```
PS C:\Users\USUARIO\Anthony\Desktop\Examen_Desarrollo> composer create-project laravel/laravel gestion-computadores
Creating a "laravel/laravel" project at "../gestion-computadores"
Installing laravel/laravel (v12.0.9)
- Downloading laravel/laravel (v12.0.9)
- Installing laravel/laravel (v12.0.9): Extracting archive
Created project in C:\Users\USUARIO\Anthony\Desktop\Examen_Desarrollo\gestion-computadores
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 110 installs, 0 updates, 0 removals
- Locking brick/math (0.12.3)
- Locking carbonphp/carbon-doctrine-types (3.2.0)
- Locking dflydev/dot-access-data (v3.0.3)
- Locking doctrine/inflector (2.0.10)
- Locking doctrine/lexer (3.0.1)
- Locking dragonmantank/cron-expression (v3.4.0)
- Locking egulias/email-validator (4.0.4)
- Locking fakerphp/faker (v1.24.1)
- Locking filp/whoops (2.18.0)
- Locking fruitcake/php-cors (v1.3.0)
- Locking graham-campbell/result-type (v1.1.3)
```

Este método muestra la lista de computadores, permitiendo filtrar los resultados. Obtiene un término de búsqueda (query) desde la solicitud y usa `Computador::query()` para buscar coincidencias parciales en los campos `codigo_tienda` o `procesador`. Finalmente, devuelve la vista `computadores.index` con la lista de computadores.

```
public function index(Request $request)
{
    $query = $request->input('query');
    $computadores = Computador::query()
        ->when($query, function ($q) use ($query) {
            return $q->where('codigo_tienda', 'like', "%{$query}%")
                ->orWhere('procesador', 'like', "%{$query}%");
        })
        ->get();
    return view('computadores.index', compact('computadores'));
}
```

Este método devuelve la vista `computadores.create`, que contiene un formulario para crear un nuevo computador. Es la página que los usuarios ven cuando desean agregar un nuevo registro.

```
public function create()
{
    return view('computadores.create');
}
```

Este método procesa el formulario de creación de un computador. Valida los datos enviados (como `codigo_tienda`, `almacenamiento`, `ram`, etc.) y, si son válidos, crea un nuevo registro en la base de datos usando `Computador::create()`. Redirige al listado con un mensaje de éxito.

```
public function store(Request $request)
{
    $validated = $request->validate([
        'codigo_tienda' => 'required|string|max:255',
        'almacenamiento' => 'required|in:256GB,512GB,1TB',
        'ram' => 'required|in:8GB,16GB,32GB',
        'tarjeta_grafica' => 'required|in:NVIDIA GTX 1650,AMD Radeon RX 580,Intel UHD Graphics',
        'precio' => 'required|numeric|min:0',
        'descripcion' => 'required|string',
        'imagen' => 'required|url',
        'procesador' => 'required|in:Intel i5,Intel i7,AMD Ryzen 5,AMD Ryzen 7',
    ]);

    Computador::create($validated);

    return redirect()->route('computadores.index')->with('success', 'Computador creado exitosamente.');
```

Este método busca un computador por su ID con `Computador::findOrFail($id)`.

Actualmente, incluye un `dd($computador)` para depuración, mostrando los datos del computador. Luego, devuelve la vista `computadores.edit` con los datos del computador para su edición.

```
public function edit($id)
{
    $computador = Computador::findOrFail($id);
    dd($computador);
    return view('computadores.edit', compact('computador'));
}
```

Este método procesa el formulario de edición de un computador. Valida los datos enviados, busca el computador por su ID con `Computador::findOrFail($id)`, actualiza el registro con los datos validados usando `update()`, y redirige al listado con un mensaje de éxito.

```
public function update(Request $request, $id)
{
    $validated = $request->validate([
        'codigo_tienda' => 'required|string|max:255',
        'almacenamiento' => 'required|in:256GB,512GB,1TB',
        'ram' => 'required|in:8GB,16GB,32GB',
        'tarjeta_grafica' => 'required|in:NVIDIA GTX 1650,AMD Radeon RX 580,Intel UHD Graphics',
        'precio' => 'required|numeric|min:0',
        'descripcion' => 'required|string',
        'imagen' => 'required|url',
        'procesador' => 'required|in:Intel i5,Intel i7,AMD Ryzen 5,AMD Ryzen 7',
    ]);

    $computador = Computador::findOrFail($id);
    $computador->update($validated);

    return redirect()->route('computadores.index')->with('success', 'Computador actualizado exitosamente.');
```

Este método elimina un computador de la base de datos. Busca el computador por su ID con `Computador::findOrFail($id)`, lo elimina con `delete()`, y redirige al listado con un mensaje de éxito.

```
public function destroy($id)
{
    $computador = Computador::findOrFail($id);
    $computador->delete();

    return redirect()->route('computadores.index')->with('success', 'Computador eliminado exitosamente.');
```

protected \$table = 'computadores': Especifica que el modelo está vinculado a la tabla computadores en la base de datos y protected \$fillable = [...]: Define los campos que pueden ser asignados masivamente (mass assignable) al crear o actualizar registros, incluyendo codigo_tienda, almacenamiento, ram, tarjeta_grafica, precio, descripcion, imagen y procesador.

```
app > Models > Computador.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Computador extends Model {
9      use HasFactory;
10
11      protected $fillable = [
12          'codigo_tienda',
13          'almacenamiento',
14          'ram',
15          'tarjeta_grafica',
16          'precio',
17          'descripcion',
18          'imagen',
19          'procesador',
20      ];
21  }
22
```

Este método define la creación de la tabla computadores en la base de datos. Utiliza Schema::create para crear una tabla con columnas: id (autoincremental), codigo_tienda, almacenamiento, ram, tarjeta_grafica (cadenas de texto), precio (decimal con 2 decimales), descripcion (texto largo), imagen, procesador (cadenas de texto), y timestamps (campos created_at y updated_at).

```
public function up(): void {
    Schema::create('computadores', function (Blueprint $table) {
        $table->id(); // Campo ID automático
        $table->string('codigo_tienda'); // Código de tienda
        $table->string('almacenamiento'); // Almacenamiento (ej. "512GB")
        $table->string('ram'); // RAM (ej. "16GB")
        $table->string('tarjeta_grafica'); // Tarjeta gráfica (ej. "NVIDIA GTX 1650")
        $table->decimal('precio', 8, 2); // Precio (ej. 999.99, con 2 decimales)
        $table->text('descripcion'); // Descripción (texto largo)
        $table->string('imagen'); // Imagen (ruta o URL de la imagen)
        $table->string('procesador'); // Procesador (ej. "Intel i7")
        $table->timestamps(); // Campos created_at y updated_at
    });
}
```

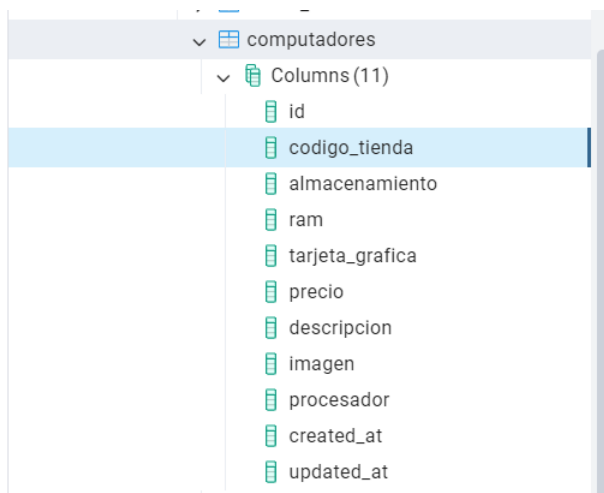
Este método genera 10 registros de prueba para la tabla computadores usando el paquete Faker. Crea instancias de Computador con datos aleatorios: codigo_tienda (una palabra), almacenamiento (256GB, 512GB, o 1TB), ram (8GB, 16GB, o 32GB), tarjeta_grafica (NVIDIA GTX 1650, AMD Radeon RX 580, o Intel UHD Graphics), precio (entre 500 y 3000 con 2 decimales), descripcion (una oración), imagen (URL de imagen), y procesador (Intel i5, Intel i7, AMD Ryzen 5, o AMD Ryzen 7).

```
public function run(): void {
    $faker = Faker::create();
    for ($i = 0; $i < 10; $i++) {
        Computador::create([
            'codigo_tienda' => $faker->word,
            'almacenamiento' => $faker->randomElement(['256GB', '512GB', '1TB']),
            'ram' => $faker->randomElement(['8GB', '16GB', '32GB']),
            'tarjeta_grafica' => $faker->randomElement(['NVIDIA GTX 1650', 'AMD Radeon RX 580', 'Intel UHD Graphics']),
            'precio' => $faker->randomFloat(2, 500, 3000),
            'descripcion' => $faker->sentence,
            'imagen' => $faker->imageUrl(),
            'procesador' => $faker->randomElement(['Intel i5', 'Intel i7', 'AMD Ryzen 5', 'AMD Ryzen 7']),
        ]);
    }
}
```

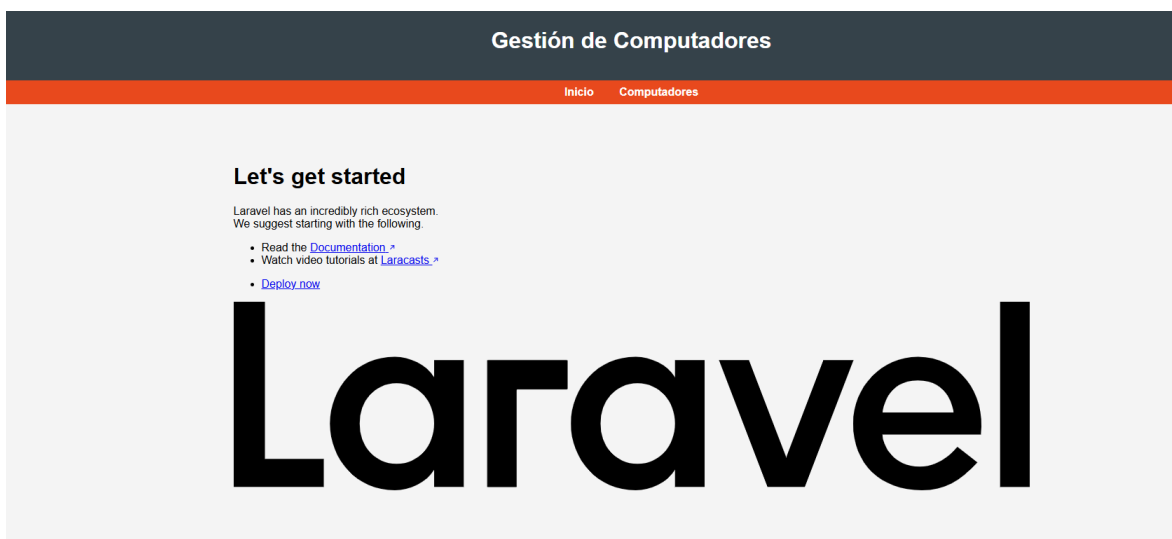
Este comando se utiliza en Laravel, un framework de PHP, para llenar la base de datos con datos de prueba usando una clase "seeder".

```
PS C:\Users\USUARIO\Anthony\Desktop\Examen_Desarrollo\gestion-computadores> php artisan db:seed --class=ComputadorSeeder  
INFO Seeding database.  
PS C:\Users\USUARIO\Anthony\Desktop\Examen_Desarrollo\gestion-computadores>
```

Creación de las tablas computadores en pgAdmin 4



Página de inicio



Vista general de la sección de computadoras

Lista de Computadores

ID	Código de Tienda	Almacenamiento	RAM	Tarjeta Gráfica	Precio	Descripción	Imagen	Procesador	Acciones
1	est	256GB	32GB	AMD Radeon RX 580	793.83	Adipisci est neque odio commodi commodi necessitatibus.	Ver Imagen	AMD Ryzen 7	Editar <input type="button" value="Eliminar"/>
2	pariatur	512GB	16GB	AMD Radeon RX 580	2061.27	Ullam quis minus consectetur et id non accusantium in.	Ver Imagen	Intel i7	Editar <input type="button" value="Eliminar"/>
3	voluptatem	256GB	16GB	NVIDIA GTX 1650	921.51	Dolorem nam expedita beatae velit dignissimos vitae ratione.	Ver Imagen	Intel i5	Editar <input type="button" value="Eliminar"/>
4	minima	256GB	8GB	Intel UHD Graphics	1939.78	Enim totam officis culpa velit et doloribus.	Ver Imagen	Intel i5	Editar <input type="button" value="Eliminar"/>
5	ducimus	512GB	16GB	Intel UHD Graphics	882.01	Ipsam cupiditate atque corporis enim.	Ver Imagen	AMD Ryzen 5	Editar <input type="button" value="Eliminar"/>

Sección de crear un nuevo producto

Crear Nuevo Computador

Código de Tienda:

Almacenamiento:

256GB

RAM:

8GB

Tarjeta Gráfica:

NVIDIA GTX 1650

Precio:

Descripción:

URL de Imagen:

Procesador:

Intel i5

Opción de eliminar cualquier producto

127.0.0.1:8000 dice
¿Estás seguro de eliminar este computador?

Aceptar

Cancelar

Computadores

Buscar por código de tienda o procesador...

Buscar

Crear Nuevo Computador

Código de Tienda	Almacenamiento	RAM	Tarjeta Gráfica	Precio	Descripción	Imagen	Procesador	Acciones
	256GB	32GB	AMD Radeon RX 580	793.83	Adipisci est neque odio commodi commodi necessitatibus.	Ver Imagen	AMD Ryzen 7	Editar
	512GB	16GB	AMD Radeon RX 580	2061.27	Ullam quis minus consectetur et id non accusantium in.	Ver Imagen	Intel i7	

Opción de buscar cualquier producto por medio de su código de tienda

Lista de Computadores

Buscar

Crear Nuevo Computador

ID	Código de Tienda	Almacenamiento	RAM	Tarjeta Gráfica	Precio	Descripción	Imagen	Procesador	Acciones
4	minima	256GB	8GB	Intel UHD Graphics	1939.78	Enim totam officis culpa velit et doloribus.	Ver Imagen	Intel i5	Editar Eliminar

Tabla con los datos generado por IA

Data Output								
Showing rows: 1 to 10								
Page No: 1 of 1								
id	codigo_tienda	almacenamiento	ram	tarjeta_grafica	precio	descripcion	imagen	
1	1	est	256GB	32GB	AMD Radeon RX 580	793.83	Adipisci est neque odio commodi commodi necessitatibus.	https://via.placeholder.com/640x480
2	2	pariatum	512GB	16GB	AMD Radeon RX 580	2061.27	Ullam quis minus consectetur et id non accusantium in.	https://via.placeholder.com/640x480
3	3	voluptatem	256GB	16GB	NVIDIA GTX 1650	921.51	Dolorem nam expedita beatae velit dignissimos vitae ratione.	https://via.placeholder.com/640x480
4	4	minima	256GB	8GB	Intel UHD Graphics	1939.78	Enim totam officis culpa velit et doloribus.	https://via.placeholder.com/640x480
5	5	ducimus	512GB	16GB	Intel UHD Graphics	882.01	Ipsam cupiditate atque corporis enim.	https://via.placeholder.com/640x480
6	6	ut	512GB	32GB	AMD Radeon RX 580	2231.23	Dolores ipsam voluptas recusandae qui ipsam molestiae maxime.	https://via.placeholder.com/640x480
7	7	deserunt	256GB	16GB	NVIDIA GTX 1650	1325.04	Veritatis illo dolores incidunt possimus dolore consequatur libero culpa.	https://via.placeholder.com/640x480
8	8	ea	512GB	32GB	NVIDIA GTX 1650	529.17	Aut dolores id itaque pariatum fugit harum blanditiis aliquid.	https://via.placeholder.com/640x480