

# Issue 7 del 27 de enero

Temas

Crud=acciones básicas para crear, leer, actualizar y eliminar datos en una base de datos.

-how to delete(crud)

-route model binding

-how to update(crud)

How to delete (crud)

Eliminaremos registros de la base de datos en este caso con un botón para eliminar comentarios

El diseño:

```
<button class="btn btn-danger btn-sm"> Delete </button>
```

Ruta:

```
Route::delete('/ideas/{id}', [IdeaController::class, 'store'] )->name('ideas.store');
```

Metodo en la clase principal:

```
<div>
  <form method="POST" action="{{ route('ideas.destroy', $idea->id) }}">
    @method('delete')
    <button class="btn btn-danger btn-sm"> X </button>
  </form>
</div>
```

Esto nos da como resultado un botón el cual borra los comentarios y al momento de querer eliminar uno que ya estaba eliminado manda error 404

metodo destroy en el controller.php:

```
public function destroy($id){
    $idea = Idea::where('id', $id)->firstOrFail();
    $idea->delete();

    return redirect()->route('dashboard')->with('success', 'Idea deleted successfully !');
}
```

# Issue 7 del 27 de enero

Route model binding:

permite inyectar automáticamente las instancias de modelos en las rutas

por ejemplo en el metodo destroy lo que hicimos fue que en la parte del argumento gracias a que estamos trabajando con laravel podemos dejarle idea \$idea y eliminar la línea que busca darle un valor a la variable

que es lo que sucede aquí lo que hace este pequeño cambio de idea \$idea es que laravel busca en los datos de idea automáticamente y se los asigna a \$idea haciendo mas compacto el código sin quitarle funcionalidad

antes:

```
public function destroy($id){  
  
    $idea = Idea::where('id',$id)->firstOrFail();  
    $idea->delete();  
  
    return redirect()->route('dashboard')->with('success','Idea deleted successfully !');  
}
```

Después:

```
public function destroy(Idea $idea){  
  
    $idea->delete();  
  
    return redirect()->route('dashboard')->with('success','Idea deleted successfully !')  
}
```

# Issue 7 del 27 de enero

how to update(crud):

añadiremos la opción de editar los comentarios en nuestra web

para comenzar añadimos el botón el cual accedera a la ruta

```
<a href="{{ route('ideas.show', $idea->id) }}"> Edit </a>
```

La ruta:

```
Route::get('/ideas/{idea}/edit', [IdeaController::class, 'edit'])->name('ideas.edit');
```

Ahora asignamos el if para que la app decida cuando estemos o no editando(nos mande o no a la pagina para actualizarlo)

```
@if ($editing ?? false)
    <form action="{{ route('ideas.update') }}" method="post">
        @csrf
        <div class="mb-3">
            <textarea name="content" class="form-control" id="content" rows="3">{{ $idea->content }}</textarea>
            @error('content')
                <span class="d-block fs-6 text-danger mt-2"> {{ $message }} </span>
            @enderror
        </div>
        <div class="">
            <button type="submit" class="btn btn-dark"> Share </button>
        </div>
    </form>
@else
    <p class="fs-6 fw-light text-muted">
        {{ $idea->content }}
    </p>
@endif
```

(1) Definimos la ruta de update al igual que (2) definir la variable \$idea será lo mismo que \$id

(1)

```
Route::put('/ideas/{idea}', [IdeaController::class, 'update'])->name('ideas.update');
```

(2)

```
@if ($editing ?? false)
    <form action="{{ route('ideas.update', $idea->id) }}" method="post">
```

Los respectivos métodos edit para confirmar que estamos editando en archivo respectivo al comentario y el update con las mismas limitaciones que al crear uno le asigna su nuevo valor al contenido para despues guardarlo y mandar de vuelta al usuario a la pagina principal

```
public function edit(Idea $idea){
    $editing = true;
    return view('ideas.show', compact('idea', 'editing'));
}

public function update(Idea $idea){
    request()->validate([
        'content' => 'required|min:3|max:240'
    ]);
    $idea->content = request()->get('content', '');
    $idea->save();
    return redirect()->route('ideas.show', $idea->id)->with('success', "Idea updated successfully!");
}
```