

Fundación Universitaria Claretiana.

Facultad de ingeniería de sistemas

Kenneth Rances Arriaga Wnfried

4 de Febrero de 2025.

**Electiva de Profundizacion II: Desarrollo de Software (Programción de Sitio
Web)**

ReciclApp - ¿Qué es?

ReciclApp es una plataforma digital diseñada para fomentar el reciclaje responsable mediante una experiencia interactiva y educativa. Su objetivo principal es conectar a los ciudadanos con centros de reciclaje, brindar información sobre la correcta clasificación de residuos y promover una cultura de sostenibilidad en las comunidades.

La aplicación web permite:

- Consultar puntos de reciclaje cercanos según su ubicación.
- Aprender a clasificar correctamente distintos tipos de residuos.
- Participar en actividades y retos con sistema de puntos y recompensas.
- Generar estadísticas sobre el impacto ambiental personal.

El problema

En muchas ciudades los ciudadanos no tienen acceso fácil a información sobre reciclaje, lo que provoca una clasificación inadecuada de residuos y desinterés por prácticas sostenibles. La falta de incentivos y educación ambiental práctica también contribuye a esfuerzos de reciclaje ineficaces.

Principales problemas:

- Desconocimiento de puntos de reciclaje disponibles.
- Confusión sobre la clasificación de residuos.
- Falta de motivación para reciclar.
- Escasa visibilidad del impacto positivo generado por el reciclaje.

La solución

ReciclApp propone una plataforma web accesible desde cualquier dispositivo, con diseño intuitivo y enfoque educativo, para involucrar a cualquier persona en el reciclaje.

Funcionalidades principales:

- Mapa interactivo con centros de acopio cercanos.
- Guía dinámica para clasificar residuos correctamente.
- Sistema de puntos y recompensas por participación activa.
- Estadísticas personalizadas del impacto ambiental del usuario.
-

Actores y Casos de Uso

Actores: Ciudadano, Administrador, Centro de reciclaje.

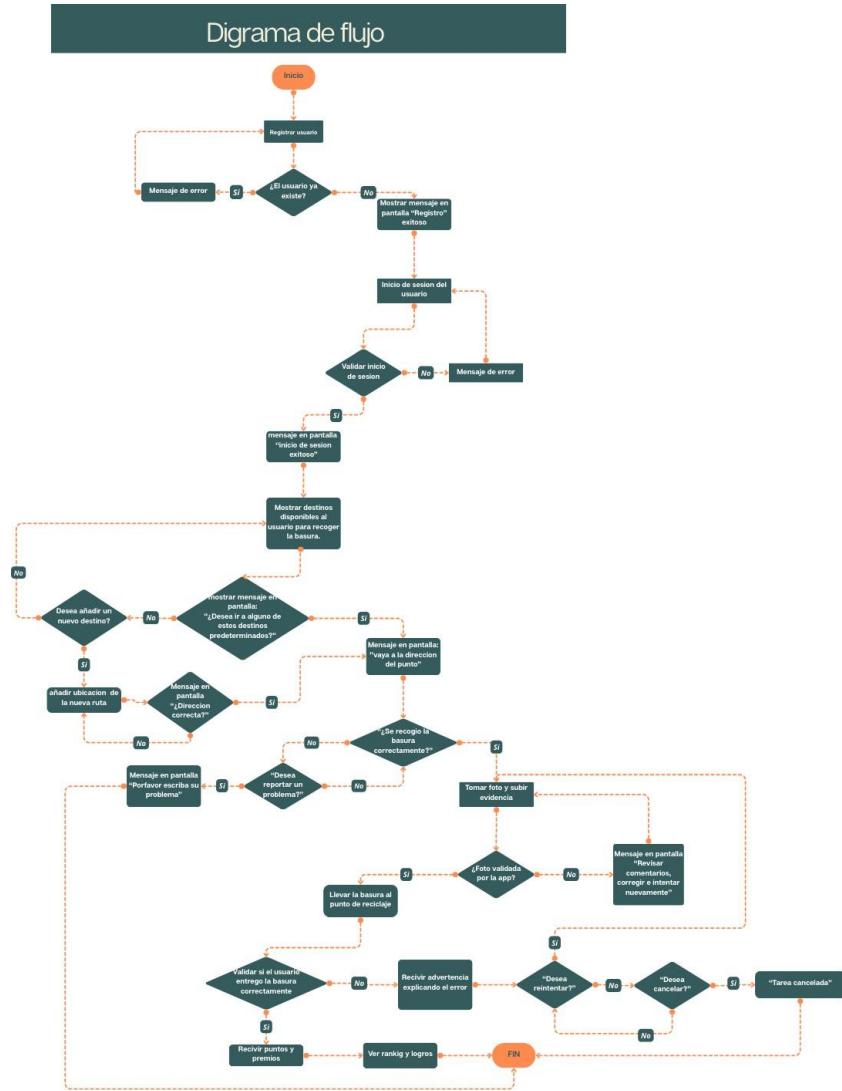
Casos de uso – Ciudadano

- Registro e inicio de sesión.
- Consultar puntos de reciclaje.
- Clasificar residuos.
- Ver estadísticas.

Casos de uso – Administrador

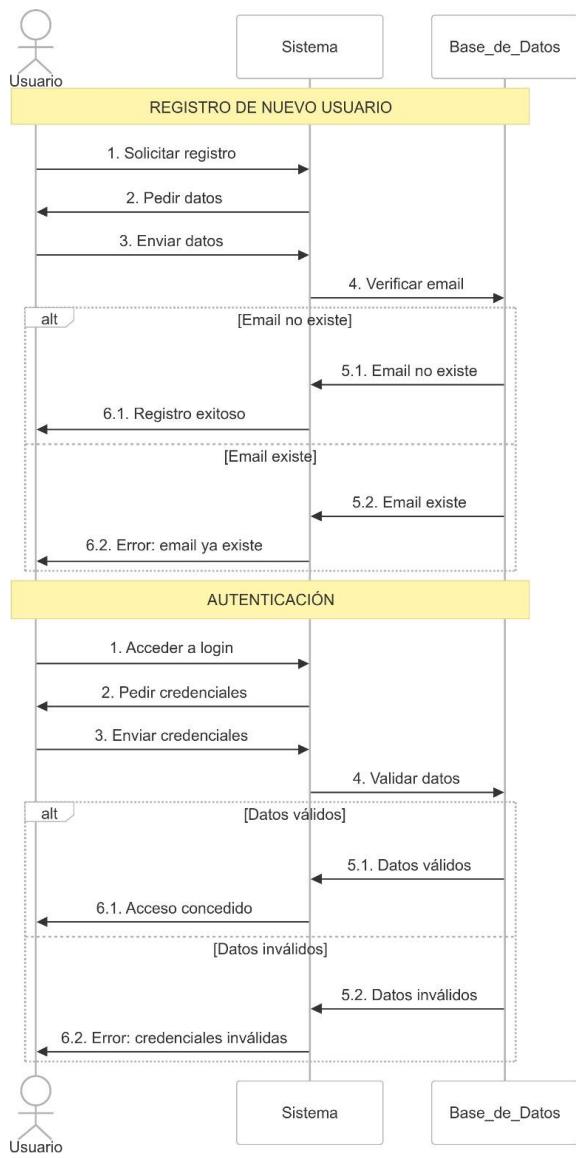
- Registro e inicio de sesión.
- Gestionar usuarios.
- Validar información.
- Crear actividades.

Diagrama de Flujo General

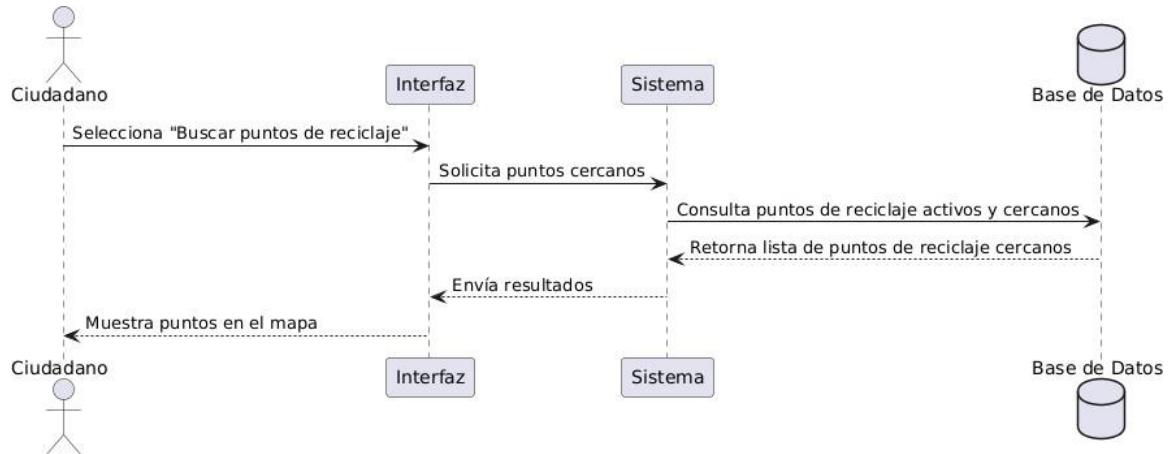


Diagramas de secuencia

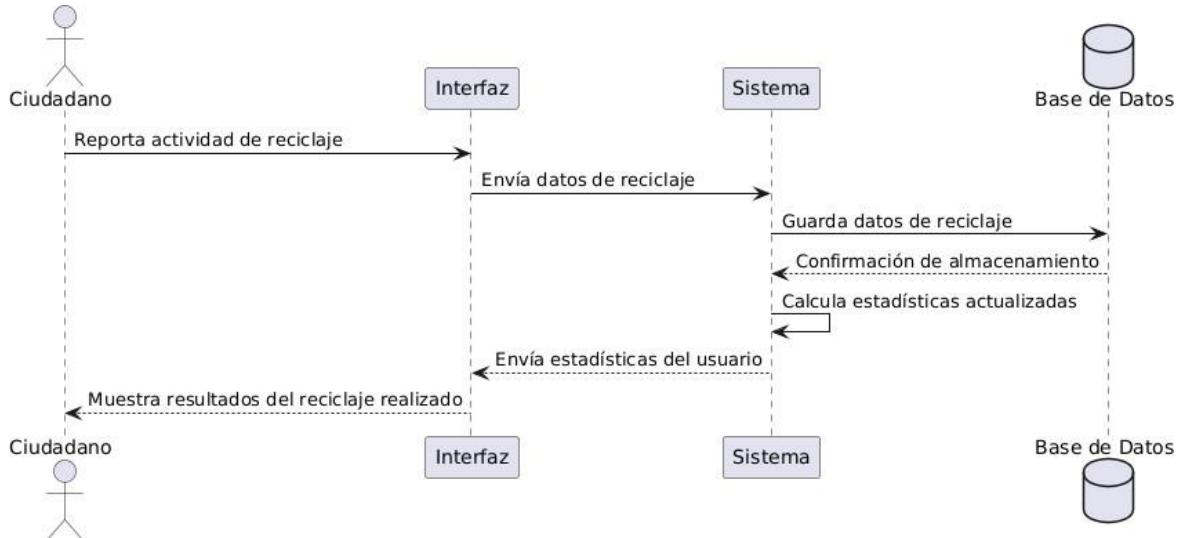
Registro e inicio de sesión



Consulta de punto de reciclaje



Carga de estadísticas



Página de inicio

RecicApp - Inicio

127.0.0.1:8000

Reiniciar para actualizar

RecicApp

Inicio Sobre Características Iniciar sesión Registrarse

Reciclaje inteligente para tu ciudad

Convierte cada residuo en una oportunidad para el planeta.

RecicApp te ayuda a encontrar puntos de reciclaje, registrar tus entregas y visualizar el impacto real de tus acciones de una forma sencilla y motivadora.

Comenzar ahora → Ya tengo cuenta

Tu impacto hoy En crecimiento

El mejor residuo es el que no se genera: reutiliza bolsas, botellas y frascos siempre que sea posible.

Cada registro cuenta para acercarnos a una ciudad más limpia.

¿POR QUÉ RECICLAPP?



127.0.0.1:8000

Reiniciar para actualizar

¿POR QUÉ RECICLAPP?

Una plataforma pensada para personas, comunidades y ciudades.

RecicApp conecta a ciudadanos, puntos de reciclaje y administradores en un mismo lugar para que el proceso de reciclar sea claro, trazable y motivador.



Encuentra dónde reciclar

Explora un mapa de puntos de reciclaje verificados y descubre qué materiales acepta cada uno.

Registra tus entregas

Lleva el historial de tus entregas y consulta cuánto has reciclado en cualquier momento.

Gana puntos y logros

Obtén puntos, logros y recompensas por mantener el hábito de reciclar de forma constante.

Mide tu impacto

Visualiza estadísticas sobre tu contribución ambiental y compártelas con tu comunidad.

¿CÓMO FUNCIONA?

1 Crea tu cuenta y completa tu perfil

127.0.0.1:8000

Reiniciar para actualizar

¿CÓMO FUNCIONA?

Empieza a reciclar con RecicApp en solo tres pasos.

- 1 Crea tu cuenta y completa tu perfil
- 2 Encuentra un punto y registra tu entrega
- 3 Sigue tu progreso y canjea recompensas

1 Crea tu cuenta y completa tu perfil

Regístrate en minutos, indica tu ciudad y empiezas a ver los puntos de reciclaje cercanos.

2 Encuentra un punto y registra tu entrega

Entrega tus materiales en un punto allado, registra lo que reciclaste y genera puntos.

3 Sigue tu progreso y canjea recompensas

Consulta tu impacto acumulado y canjea beneficios definidos por tu institución o comunidad.

¿Listo para llevar tu reciclaje al siguiente nivel?

Regístrate gratis, empieza a registrar tus acciones y muestra el impacto real que generas.

Crear mi cuenta → Ya tengo cuenta

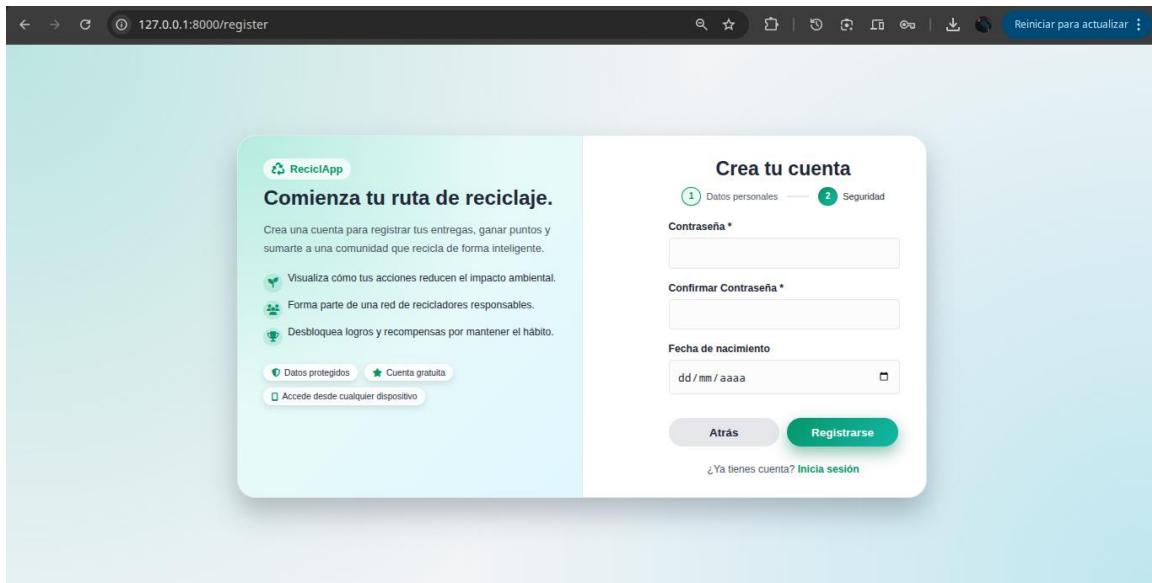
© 2025 RecicApp. Todos los derechos reservados.

Login

The screenshot shows a web browser window with the URL 127.0.0.1:8000/login. The page has a light blue background. On the left, there's a sidebar with the ReciciApp logo and the text "Vuelve a conectar con tu impacto." It lists three items: "Accede a tu panel de reciclaje personalizado.", "Consulta puntos de reciclaje y propuestas.", and "Sigue de cerca tus recompensas y logros." On the right, there's a main form titled "Iniciar sesión" with the sub-instruction "Ingresa tus credenciales para acceder a tu panel." It contains fields for "Correo Electrónico" (with placeholder "tu@email.com") and "Contraseña" (with placeholder "*****"). Below these is a green "Iniciar Sesión" button. At the bottom of the form, there's a link "¿Aún no tienes cuenta? Crear una cuenta". The browser toolbar at the top includes icons for back, forward, search, and refresh.

Register

The screenshot shows a web browser window with the URL 127.0.0.1:8000/register. The page has a light blue background. On the left, there's a sidebar with the ReciciApp logo and the text "Comienza tu ruta de reciclaje." It lists three items: "Visualiza cómo tus acciones reducen el impacto ambiental.", "Forma parte de una red de recicladore responsables.", and "Desbloquea logros y recompensas por mantener el hábito." It also includes links for "Datos protegidos", "Cuenta gratuita", and "Accede desde cualquier dispositivo". On the right, there's a main form titled "Crea tu cuenta" with two tabs: "1 Datos personales" (selected) and "2 Seguridad". The "Datos personales" tab contains fields for "ID / Cédula *", "Nombre *", "Apellido *", and "Correo *". Below these is a green "Siguiente" button. At the bottom of the form, there's a link "¿Ya tienes cuenta? Inicia sesión". The browser toolbar at the top includes icons for back, forward, search, and refresh.



Panel del usuario

The screenshot shows a web browser window at the URL 127.0.0.1:8000/panel. The header is green with the RecicApp logo and navigation links: Inicio, Sobre, Características, and Cerrar sesión.

Left Sidebar (Teal):

- Section Title:** Resumen
- Links:** Gestionar reciclaje, Canjear premios, Puntos de reciclaje, Actividades y eventos.
- User Info:** Kenneth, ken123@gmail.com

Main Content Area (White):

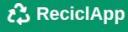
- Section Title:** Gestión de reciclaje
- Statistics:** Puntos acumulados: 15, Entregas aprobadas: 1, Canjes realizados: 0.
- Section Title:** Impacto de tu reciclaje
- Text:** Una vista rápida de cómo tus acciones podrían reflejarse en el impacto ambiental.
- Figure:** Bar chart titled "Kg reciclados por mes" showing monthly recycling volume: Ene (12 kg), Feb (19 kg), Mar (24 kg), Abr (29 kg).
- Section Title:** Beneficios estimados
- List:**
 - Cada 10 kg de plástico reciclado pueden evitar la emisión de varios kg de CO₂.
 - Tus entregas aprobadas ayudan a mantener más limpios los puntos de reciclaje.
 - Estos datos permiten planear campañas y mejoras en tu comunidad.

Gestionar reciclaje

The screenshot shows the RecicApp dashboard at the URL 127.0.0.1:8000/panel/reciclaje. The header features the RecicApp logo and navigation links for Inicio, Sobre, Características, and Cerrar sesión. A green sidebar on the left contains a back button labeled 'Volver al panel' and a link 'Gestionar mi reciclaje'. The main content area has a title 'Elige cómo quieres aportar hoy' with a sub-instruction: 'Desde este espacio puedes proponer nuevos puntos, reportar problemas y registrar tus entregas para que tu impacto quede registrado y llegue al equipo administrador.' Below this, there's a section titled 'ACCIONES RÁPIDAS' with three items: 'Gestiona tus acciones de reciclaje', 'Guía rápida', and 'Proponer nuevo punto'. The 'Gestiona tus acciones de reciclaje' section includes a note about updating point and delivery information and a 'Proporcionar punto' button. The 'Guía rápida' section provides a quick guide for reporting new points, reporting problems, and registering deliveries. The 'Proponer nuevo punto' section details how to propose new collection points.

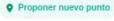
This screenshot focuses on the 'Proporcionar nuevo punto' section from the previous dashboard. It includes a note: 'Si conoces un lugar nuevo para reciclar, propónlo. Si ya existe en el mapa, simplemente utilízalo para tus entregas.' Below this are three steps: 1. 'Reporta novedades del punto': 'Si encuentras un contenedor lleno, dañado o fuera de servicio, reportalo para que el equipo pueda actuar.' 2. 'Registra tus entregas': 'Indica qué tipo de material entregaste y en qué cantidad. Tus aportes se enviarán al administrador para validación.' 3. '¿Qué hace el administrador con tu información?': 'Todo lo que registras en este panel se usa para mejorar la red de reciclaje y reconocer tu esfuerzo.' The list under this question includes: • Las propuestas de puntos se revisan antes de publicarlas en el mapa público. • Los reportes de problemas ayudan a priorizar mantenimientos y correcciones. • Las entregas registradas se validan y se convierten en puntos para tu cuenta.

Proponer nuevo punto

 ReciclApp

Inicio Sobre Características Cerrar sesión

[← Volver](#)

 Propone nuevo punto

Ayuda a ampliar la red de reciclaje

Comparte un lugar donde otras personas puedan llevar materiales reciclables. Puede ser un centro formal, un contenedor de tu barrio o un espacio temporal después de un evento.

Datos del punto de reciclaje

Completa la información básica del lugar. Mientras más claro sea el detalle, más fácil será ubicarlo y validar.

Nombre o referencia del lugar
Ej: Parque Central - Feria de libros

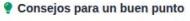
Tipo de materiales disponibles
Selecciona el tipo

Dirección o ubicación aproximada
Calle, barrio, ciudad o punto de referencia

Horario recomendado
Ej: Disponible desde las 10:00 hasta las 18:00

Detalles adicionales (opcional)
Ej: tras evento deportivo, festival o feria, materiales separados, etc.



 Consejos para un buen punto

Ten en cuenta estos detalles al proponer un nuevo lugar de reciclaje:

- Verifica que el lugar sea seguro y accesible para otras personas.
- Indica referencias claras (parques, esquinas, locales cercanos, etc.).
- Describe qué tipo de materiales suelen acumularse allí.
- Si es un evento temporal, menciona la fecha o duración aproximada.

 ¿Qué pasa después de enviar tu propuesta?

3. Tus propuestas ayudan a que la red de reciclaje crezca y sea más útil para tu comunidad.

Puedes seguir proponiendo nuevos puntos siempre que detectes lugares donde el reciclaje pueda tener impacto.

[Enviar propuesta](#) [Cancelar y volver](#)

© 2025 ReciclApp. Todos los derechos reservados.

Reportar problema

▲ Reportar problema en un punto**Ayúdanos a mantener la red de reciclaje en buen estado**

Selecciona el punto y describe lo que sucede para que podamos tomar acciones y que otros usuarios tengan información confiable.

Punto de reciclaje**Tipo de problema****Comentarios adicionales**

Detalles adicionales, si los hay...

💡 ¿Por qué es importante reportar?

Cada reporte ayuda a detectar puntos con problemas antes de que afecten a más personas. Así podemos priorizar limpiezas, cambios de ubicación o correcciones en la información.



3 pts sin reportes 7 pts con reportes 12 pts atendidos

En este periodo, 7 puntos reportados permitieron que 12 ubicaciones fueran atendidas y mejoradas frente a solo 3 sin incidentes registrados.

💡 Tips para un buen reporte

- Indica si el punto está inaccesible, sucio, dañado o si la dirección parece incorrecta.
- Si es seguro, puedes mencionar referencias cercanas (parques, esquinas, locales).
- Usa el campo "Otros" solo cuando el problema no encaje en las categorías anteriores.

Registrar tu entrega

ReciclApp

Inicio Sobre Características Cerrar sesión

[Registrar mi entrega](#)

Registra tu aporte al planeta

Ingresá los datos de los materiales que has llevado a reciclar para que podamos sumar tu impacto.

Tipo de residuo

Cantidad (kg)

Fecha de la recolección

Lugar de la recolección (opcional)

Descripción general de los materiales

Imagina que cada bolsa que entregás aquí se transforma en árboles, parques limpios y menos residuos en tu ciudad.

Registrar tu entrega es el primer paso.

Tips para registrar mejor tus materiales

- Indica el tipo de residuo de forma clara (plástico, cartón, vidrio, etc.).
- Si no sabes el peso exacto, aproxima la cantidad en kilogramos.
- Describe brevemente el estado de los materiales (limpios, separados, mezclados, etc.).

[Registrar entrega](#) [← Volver](#)

© 2025 ReciclApp. Todos los derechos reservados.

Encontrar puntos de reciclaje

ReciclApp

Inicio Sobre Características Cerrar sesión

[Volver al panel](#)

[Puntos de reciclaje](#)

Encuentra puntos de reciclaje cerca de ti

Usa el mapa para ubicar puntos oficiales de reciclaje en Quibdo. El marcador del Parque Central te muestra un punto clave en el centro de la ciudad.

Resumen de puntos

Estos son algunos puntos de reciclaje destacados. Seguiremos agregando más lugares a medida que la comunidad crece:

- Parque Central de Quibdo

Punto fijo de reciclaje en el parque central de la ciudad.
Tipo: plástico, papel y vidrio

Como usar este mapa

Mueve el mapa, acerca o aleja con el zoom y pulsa sobre los marcadores para ver detalles del punto de reciclaje.
Si conoces un nuevo lugar donde podría instalarse un punto, puedes usar la opción Proporponer nuevo punto en tu panel.

© 2025 ReciclApp. Todos los derechos reservados.

Actividades y eventos

Panel del administrador

Hola, ADMINISTRADOR

Desde aquí puedes monitorear la actividad de RecicApp, gestionar usuarios, puntos de reciclaje, reportes y recompensas.

USUARIOS	PUNTOS DE RECICLAJE	REPORTES	PREMIOS
17 Registrados en la plataforma	2 Centros registrados	1 Reportes registrados	5 Disponibles para canje

Atajos rápidos

- Gestionar usuarios
- Revisar propuestas
- Gestionar reportes
- Configurar premios

Resumen rápido

Un vistazo general al estado actual de RecicApp.

Categoría	Datos
Usuarios activos	17
Puntos de reciclaje	2
Reportes registrados	1
Premios activos	5

Usuarios

Gestión de usuarios

Administra roles y estado de los usuarios registrados en RecicApp.

Nombre	Email	Role	Estado	Acciones
ADMINISTRADOR AD	administrador@gmail.com	Administrador	Activo	Pasar a usuario Eliminar
Admin Sl	admin2@gmail.com	Administrador	Activo	Pasar a usuario Eliminar
Admin Principal	admin@gmail.com	Administrador	Activo	Pasar a usuario Eliminar
Kenneth Wnfried	admin0@gmail.com	Usuario	Activo	Hacer admin Eliminar
Kenneth		Usuario	Activo	Hacer admin Eliminar
Kenneth Wnfried	usuario1@gmail.com	Usuario	Activo	Hacer admin Eliminar
Kenneth Wnfried	rances@gmail.com	Usuario	Activo	Hacer admin Eliminar
Kenneth Wnfried	ken123@gmail.com	Usuario	Activo	Hacer admin Eliminar
Ramzes Wnfried	admin1@gmail.com	Administrador	Activo	Pasar a usuario Eliminar
admin adminn	admin3@gmail.com	Usuario	Activo	Hacer admin Eliminar
juan rada	jrama7674@gmail.com	Usuario	Activo	Hacer admin Eliminar
ke ra	ad@gmail.com	Administrador	Activo	Pasar a usuario Eliminar
kene eñe	root@gmail.com	Administrador	Activo	Pasar a usuario Eliminar
kenneth arraga	kenneth1002@gmail.com	Usuario	Activo	Hacer admin Eliminar
prueba prueba 2	ass@gmail.com	Usuario	Activo	Hacer admin Eliminar

Showing 1 to 15 of 17 results

Gestión de premios

 RecicApp Admin

[Dashboard](#)

[Usuarios](#)

[Premios](#)

[Propuestas](#)

[Reportes](#)

[Entregas](#)

[Auditoría](#)

[Actividades](#)

[Logros](#)

[Cerrar sesión](#)

Gestión de Premios

Administra el catálogo de premios y los canjes realizados por los usuarios.

Premio eliminado correctamente

Premios configurados

Premio	Puntos	Stock	Estado	Imagen	Acciones
lilito	100	50	Inactivo	Sin imagen	Borrar Guardar
lilito	100	50	Inactivo	Sin imagen	Borrar Guardar

Nuevo premio

Nombre:

Descripción:

Puntos necesarios: stock (opcional) Vacío = ilimitado

URL / ruta de imagen: img/premios/ejemplo.png

Estado: Activo

[Guardar premio](#)

Canjes de premios

Usuario	Premio	Puntos usados	Código	Fecha canje	Estado	Observaciones	Acciones
No hay canjes registrados.							

Gestion de propuestas

 RecicApp Admin

[Dashboard](#)

[Usuarios](#)

[Premios](#)

[Propuestas](#)

[Reportes](#)

[Entregas](#)

[Auditoría](#)

[Actividades](#)

[Logros](#)

[Cerrar sesión](#)

Propuestas de nuevos puntos

Revisa y decide qué puntos de reciclaje sugeridos por usuarios se activan en el sistema.

Propuestas pendientes de revisión

Nombre	Tipo	Dirección	Horario	Descripción	Propuesto por	Fecha propuesta	Acciones
buque	Misto	nño jesus	12:00 PM		Usuario desconocido	21/11/2025 00:24	Acepta Rechaza

Gestion de reportes

The screenshot shows the ReciclApp Admin interface. On the left, there's a sidebar with navigation links: Dashboard, Usuarios, Premios, Propuestas, Reportes (selected), Entradas, Auditoria, Actividades, and Logros. At the bottom of the sidebar is a 'Cerrar sesión' button. The main content area has a title 'Reportes de recolección' and a subtitle 'Listado de reportes enviados por los usuarios para seguimiento y resolución.' Below this is a table with columns: ID, Usuario, Punto, Fecha, Estado, Observaciones, and Acciones. One row is visible: ID 691fb154e2aa4acada00bd66, Usuario N/D, Punto 691fb154e2aa4acada00bd66, Fecha 21/11/2025 00:25, Estado En_progreso, Observaciones 'Tipo de problema: zona_peligro | Descripción: peligro, rob...', and an 'Acciones' button labeled 'Ver'.

Creación de modelos

- modelo de auditoria.

The screenshot shows a code editor with a dark theme. The file path is 'app > Models > Auditoria.php'. The code defines a Model named 'Auditoria' extending 'Jenssegers\Mongodb\Eloquent\Model'. It includes protected properties for connection ('mongodb'), collection ('auditoria'), and fillable fields ('accion', 'usuario', 'fecha', 'descripcion', 'modulo_afectado', 'resultado').

```
app > Models > Auditoria.php
1  <?php
2
3  namespace App\Models;
4
5  use Jenssegers\Mongodb\Eloquent\Model;
6
7  class Auditoria extends Model
8  {
9      protected $connection = 'mongodb';
10     protected $collection = 'auditoria';
11
12     protected $fillable = [
13         'accion',
14         'usuario',
15         'fecha',
16         'descripcion',
17         'modulo_afectado',
18         'resultado',
19     ];
20 }
21
```

- modelo canjeo de premios del usuario

```
1 <?php
2
3 namespace App\Models;
4
5 use Jenssegers\Mongodb\Eloquent\Model;
6
7 class CanjeUsuario extends Model
8 {
9     protected $connection = 'mongodb';
10    protected $collection = 'canjes_usuario';
11
12    protected $fillable = [
13        'id_usuario',
14        'id_premio',
15        'fecha_canje',
16        'estado',
17    ];
18 }
19
```

- Modelo reporte de recoleccion

```
app > Models >  DetalleReporteRecolección.php
1  <?php
2
3  namespace App\Models;
4
5  use Jenssegers\Mongodb\Eloquent\Model;
6
7  class DetalleReporteRecolección extends Model
8  {
9      protected $connection = 'mongodb';
10     protected $collection = 'detalle_reporte_recolección';
11
12    protected $fillable = [
13        'id_reporte',
14        'tipo_residuo',
15        'peso',
16        'unidad',
17    ];
18 }
19
```

- modelo logros

```
app > Models > Logro.php
1  <?php
2
3  namespace App\Models;
4
5  use Jenssegers\Mongodb\Eloquent\Model;
6
7  class Logro extends Model
8  {
9      protected $connection = 'mongodb';
10     protected $collection = 'logros';
11
12    protected $fillable = [
13        'nombre',
14        'descripcion',
15        'puntos_requeridos',
16        'icono',
17    ];
18 }
19
```

- modelo Premio

```
app > Models > Premio.php
1  <?php
2
3  namespace App\Models;
4
5  use MongoDB\Laravel\Eloquent\Model;
6
7
8  class Premio extends Model
9  {
10     protected $connection = 'mongodb';
11     protected $collection = 'premios';
12
13     protected $fillable = [
14         'nombre',
15         'descripcion',
16         'puntos_necesarios',
17         'stock',
18         'estado',
19     ];
20 }
21
```

-Modelo punto de reciclaje

```
app > Models > PuntoReciclaje.php
```

- Modelo reporte de recoleccion

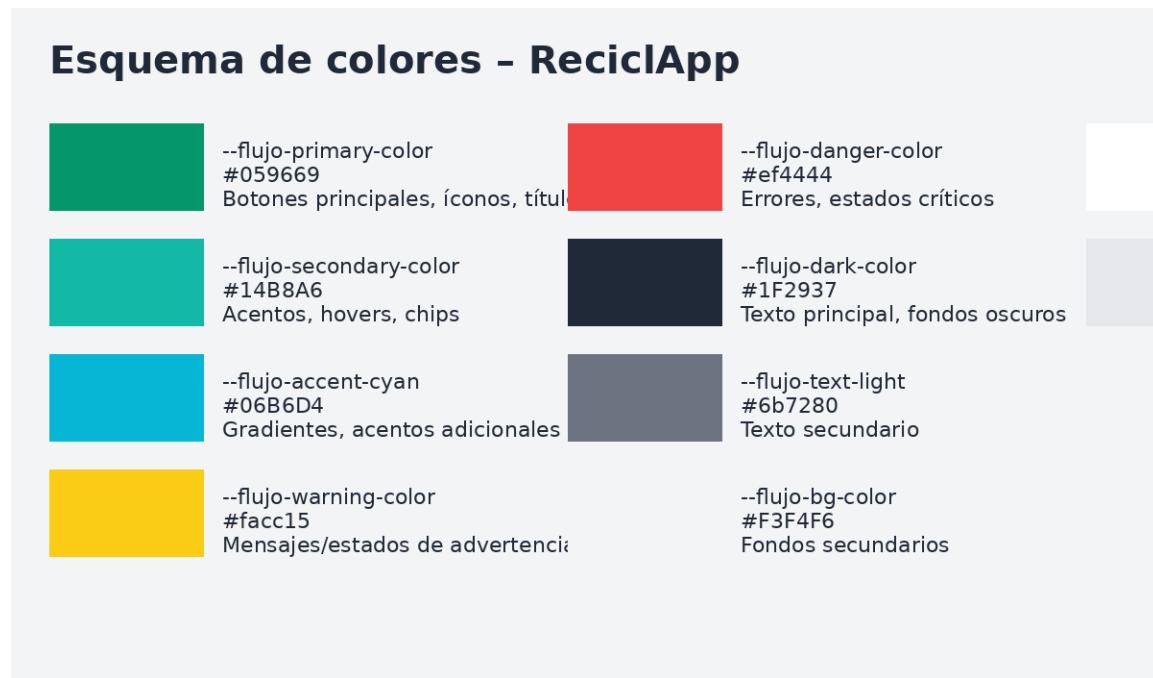
```
app > Models > ReporteRecolección.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use MongoDB\Laravel\Eloquent\Model as Eloquent;
7
8  class ReporteRecolección extends Eloquent
9  {
10      use HasFactory;
11
12      protected $connection = 'mongodb';
13      protected $collection = 'reportes_recolección_usuario';
14
15      protected $fillable = [
16          'id_usuario',
17          'id_punto',
18          'fecha_reporte',
19          'estado',
20          'observaciones',
21      ];
22  }
```

- modelos tipo de residuos

- modelo logro usuario

```
app > Models > UsuarioLogro.php
1  <?php
2
3  namespace App\Models;
4
5  use Jenssegers\Mongodb\Eloquent\Model;
6
7  class UsuarioLogro extends Model
8  {
9      protected $connection = 'mongodb';
10     protected $collection = 'usuario_logros_obtenidos';
11
12    protected $fillable = [
13        'id_usuario',
14        'id_logro',
15        'fecha_obtencion',
16    ];
17 }
18 }
```

Esquema de colores



Variable	Color / Código	Uso
--flujo-primary-color	#059669	Botones principales, íconos, títulos
--flujo-secondary-color	#14B8A6	Acentos, hovers, chips
--flujo-accent-cyan	#06B6D4	Gradientes, acentos adicionales
--flujo-warning-color	#facc15	Mensajes/estados de advertencia
--flujo-danger-color	#ef4444	Errores, estados críticos
--flujo-dark-color	#1F2937	Texto principal, fondos oscuros
--flujo-text-light	#6b7280	Texto secundario
--flujo-bg-color	#F3F4F6	Fondos secundarios
--flujo-white	#FFFFFF	Fondos de tarjetas/paneles
--flujo-border-color	#e5e7eb	Bordes suaves de tarjetas, inputs

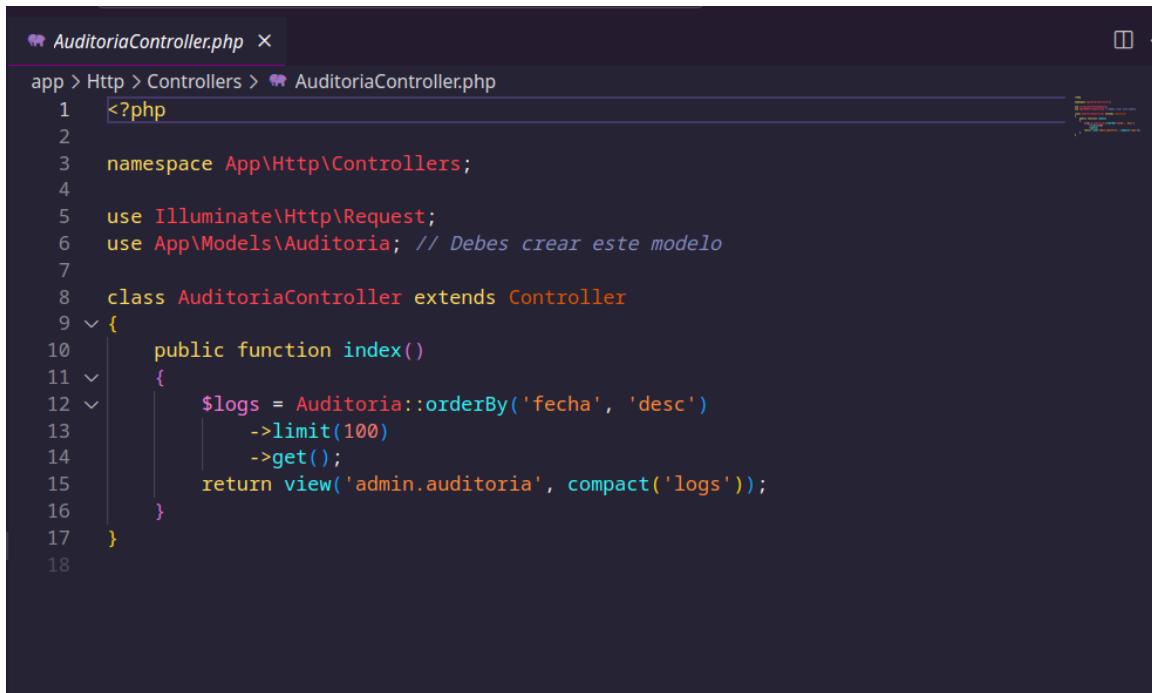
Controladores (MVC)

ActividadController

```
app > Http > Controllers > ActividadController.php
4
5  use Illuminate\Http\Request;
6  use App\Models\Actividad;
7
8 class ActividadController extends Controller
9 {
10    public function index()
11    {
12        $actividades = Actividad::orderBy('fecha', 'desc')->get();
13        return view('admin.actividades', compact('actividades'));
14    }
15
16    public function store(Request $request)
17    {
18        $data = $request->validate([
19            'titulo'      => 'required|string|max:255',
20            'descripcion' => 'nullable|string|max:2000',
21            'fecha'       => 'required|date',
22            'lugar'       => 'nullable|string|max:255',
23            'estado'      => 'required|string|in:activa,finalizada,cancelada',
24        ]);
25
26        Actividad::create($data);
27
28        return redirect()->route('admin.actividades')->with('success', 'Actividad creada correctamente');
29    }
30
31    public function update(Request $request, $id)
32    {
33        $actividad = Actividad::find($id);
34
35        if (!$actividad) {
36            return redirect()->route('admin.actividades')->with('error', 'La actividad no existe.');
37        }
38
39        $data = $request->validate([
40            'titulo'      => 'required|string|max:255',
41            'descripcion' => 'nullable|string|max:2000',
42            'fecha'       => 'required|date',
43            'lugar'       => 'nullable|string|max:255',
44            'estado'      => 'required|string|in:activa,finalizada,cancelada',
45        ]);
46
47        $actividad->update($data);
48
49        return redirect()->route('admin.actividades')->with('success', 'Actividad actualizada correctamente');
50    }
51
52    public function destroy($id)
```

Explicacion ActividadController Este controlador administra las actividades ambientales del sistema en la parte de administración y en el panel de usuario. Carga la lista de actividades desde el modelo Actividad, permite crear nuevas actividades, actualizarlas y eliminarlas, validando primero los datos del formulario. También tiene un método público (indexPublic) que muestra solo las actividades activas o próximas para que el usuario final las vea en su panel.

AuditoriaController



The screenshot shows a code editor window with the file 'AuditoriaController.php' open. The file is located in the 'app > Http > Controllers' directory. The code defines a controller class 'AuditoriaController' that extends the 'Controller' class. It contains a single method 'index()' which retrieves the last 100 logs from the 'Auditoria' model, orders them by date in descending order, and returns them to the 'admin.auditoria' view.

```
<?php  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
use App\Models\Auditoria; // Debes crear este modelo  
  
class AuditoriaController extends Controller  
{  
    public function index()  
    {  
        $logs = Auditoria::orderBy('fecha', 'desc')  
            ->limit(100)  
            ->get();  
        return view('admin.auditoria', compact('logs'));  
    }  
}
```

explicacion: Muestra el historial de auditoría del sistema (acciones importantes hechas por usuarios y administradores).

Sirve tambien para Consultar los últimos registros en la colección de auditoría, los ordena por fecha descendente y los pasa a la vista de administración para que el admin pueda revisar qué acciones se hicieron y cuándo.

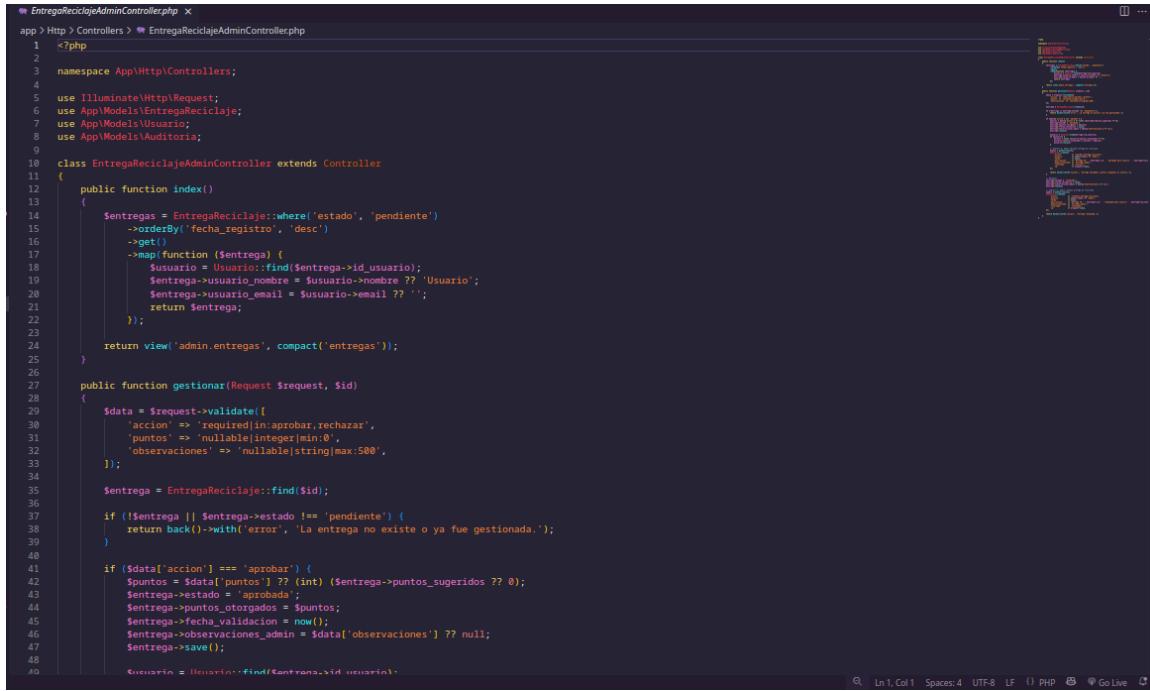
AuthController

```
AuthController.php
app > Http > Controllers > AuthController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Auth;
7  use Illuminate\Support\Facades\Hash;
8  use App\Models\Usuario;
9  use App\Models\Auditoria;
10
11 class AuthController extends Controller
12 {
13     /**
14      * Mostrar vista de login
15     */
16     public function showLogin()
17     {
18         return view('login');
19     }
20
21     /**
22      * Procesar login
23     */
24     public function login(Request $request)
25     {
26         $credentials = $request->validate([
27             'email' => 'required|email',
28             'password' => 'required',
29         ]);
30
31         if (Auth::attempt([
32             'email' => $credentials['email'],
33             'password' => $credentials['password']
34         )))
35         {
36             $request->session()->regenerate();
37
38             $rol = trim(strtolower(Auth::user()->rol ?? ''));
39
40             Auditoria::create([
41                 'accion' => 'login',
42                 'usuario' => $request->email,
43                 'fecha' => now(),
44                 'descripcion' => 'Inicio de sesión correcto',
45                 'modulo_afectado' => 'auth',
46                 'resultado' => 'exitoso',
47                 'ip' => $request->ip(),
48             ]);
49         }
50     }
51 }
```

```
AuthController.php
app > Http > Controllers > AuthController.php
12
13     {
14         $request->validate([
15             'email' => 'required|string|email|max:255|unique:usuarios',
16             'password' => 'required|string|min:6|confirmed',
17             'fecha_nacimiento' => 'nullable|date',
18         ]);
19
20         // + Crear usuario
21         $usuario = Usuario::create([
22             'id_usuario' => $request->id_usuario,
23             'nombre' => $request->nombre,
24             'apellido' => $request->apellido,
25             'email' => $request->email,
26             'contraseña_hash' => Hash::make($request->password),
27             'rol' => 'usuario',
28             'fecha_nacimiento' => $request->fecha_nacimiento,
29             'estado' => 'activo',
30             'puntos_acumulados' => 0,
31         ]);
32
33         // Registrar en auditoría el registro de un nuevo usuario
34         Auditoria::create([
35             'accion' => 'registro_usuario',
36             'usuario' => $usuario->email,
37             'fecha' => now(),
38             'descripcion' => 'Nuevo usuario registrado en el sistema',
39             'modulo_afectado' => 'auth',
40             'resultado' => 'exitoso',
41             'ip' => $request->ip(),
42         ]);
43
44         return redirect()
45             ->route('login')
46             ->with('success', 'Registro exitoso. Ahora puedes iniciar sesión con tus credenciales.');
47     }
48
49     /**
50      * Cerrar sesión
51     */
52     public function logout(Request $request)
53     {
54         Auth::logout();
55
56         $request->session()->invalidate();
57         $request->session()->regenerateToken();
58
59         return redirect()->route('login'); // volver al login
60     }
61 }
```

Explicacion: Maneja todo el proceso de autenticación: iniciar sesión, registrar nuevos usuarios y cerrar sesión. También muestra las vistas de login y registro, valida los datos del formulario, intenta autenticar usando el modelo de Usuario, redirige según el rol (admin o usuario normal) y registra en la auditoría los inicios de sesión y registros. También se encarga de cerrar la sesión del usuario.

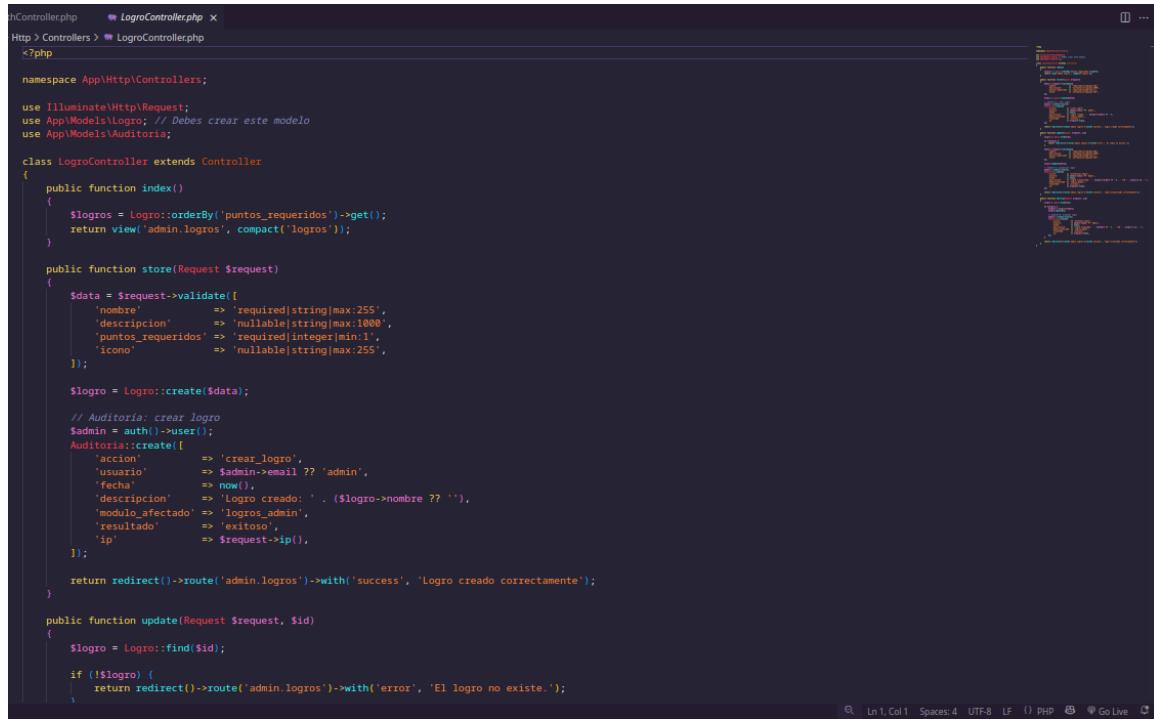
EntregaReciclajeAdminController



```
EntregaReciclajeAdminController.php X
app > Http > Controllers > EntregaReciclajeAdminController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\EntregaReciclaje;
7  use App\Models\Usuario;
8  use App\Models\Auditoria;
9
10 class EntregaReciclajeAdminController extends Controller
11 {
12     public function index()
13     {
14         $entregas = EntregaReciclaje::where('estado', 'pendiente')
15             ->orderBy('fecha_registro', 'desc')
16             ->get()
17             ->map(function ($entrega) {
18                 $usuario = Usuario::find($entrega->id_usuario);
19                 $entrega->usuario_nombre = $usuario->nombre ?? 'Usuario';
20                 $entrega->usuario_email = $usuario->email ?? '';
21                 return $entrega;
22             });
23
24         return view('admin.entregas', compact('entregas'));
25     }
26
27     public function gestionar(Request $request, $id)
28     {
29         $data = $request->validate([
30             'accion' => 'required|in:aprobar,rechazar',
31             'puntos' => 'nullable|integer|min:0',
32             'observaciones' => 'nullable|string|max:500',
33         ]);
34
35         $entrega = EntregaReciclaje::find($id);
36
37         if (!$entrega || $entrega->estado !== 'pendiente') {
38             return back()->with('error', 'La entrega no existe o ya fue gestionada.');
39         }
40
41         if ($data['accion'] === 'aprobar') {
42             $puntos = $data['puntos'] ?? (int) ($entrega->puntos_sugeridos ?? 0);
43             $entrega->estado = 'aprobada';
44             $entrega->puntos_otorgados = $puntos;
45             $entrega->fecha_validacion = now();
46             $entrega->observaciones_admin = $data['observaciones'] ?? null;
47             $entrega->save();
48
49             Auditoria::create([
50                 'accion' => 'EntregaReciclajeController.gestionar',
51                 'id_entrega' => $entrega->id,
52                 'id_usuario' => auth()->user()->id,
53                 'observaciones' => $data['observaciones'],
54             ]);
55         }
56     }
57 }
```

Explicacion: Permite al administrador revisar y gestionar las entregas de reciclaje realizadas por los usuarios. Tambien Lista las entregas pendientes, mostrando información básica del usuario; al gestionar, el admin puede aprobar o rechazar la entrega. Si aprueba, asigna puntos al usuario y actualiza el estado. Cada acción queda registrada en la tabla de auditoría.

LogroController



```
LogroController.php  LogroController.php x
Http > Controllers > LogroController.php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Logro; // Debes crear este modelo
use App\Models\Auditoria;

class LogroController extends Controller
{
    public function index()
    {
        $logros = Logro::orderBy('puntos_requeridos')->get();
        return view('admin.logros', compact('logros'));
    }

    public function store(Request $request)
    {
        $data = $request->validate([
            'nombre' => 'required|string|max:255',
            'descripcion' => 'nullable|string|max:1000',
            'puntos_requeridos' => 'required|integer|min:1',
            'icono' => 'nullable|string|max:255',
        ]);

        $logro = Logro::create($data);

        // Auditoria: crear logro
        $admin = auth()->user();
        Auditoria::create([
            'accion' => 'crear_logro',
            'usuario' => $admin->email ?? 'admin',
            'fecha' => now(),
            'descripcion' => 'Logro creado: ' . ($logro->nombre ?? ''),
            'modulo_afectado' => 'logros_admin',
            'resultado' => 'exitoso',
            'ip' => $request->ip(),
        ]);

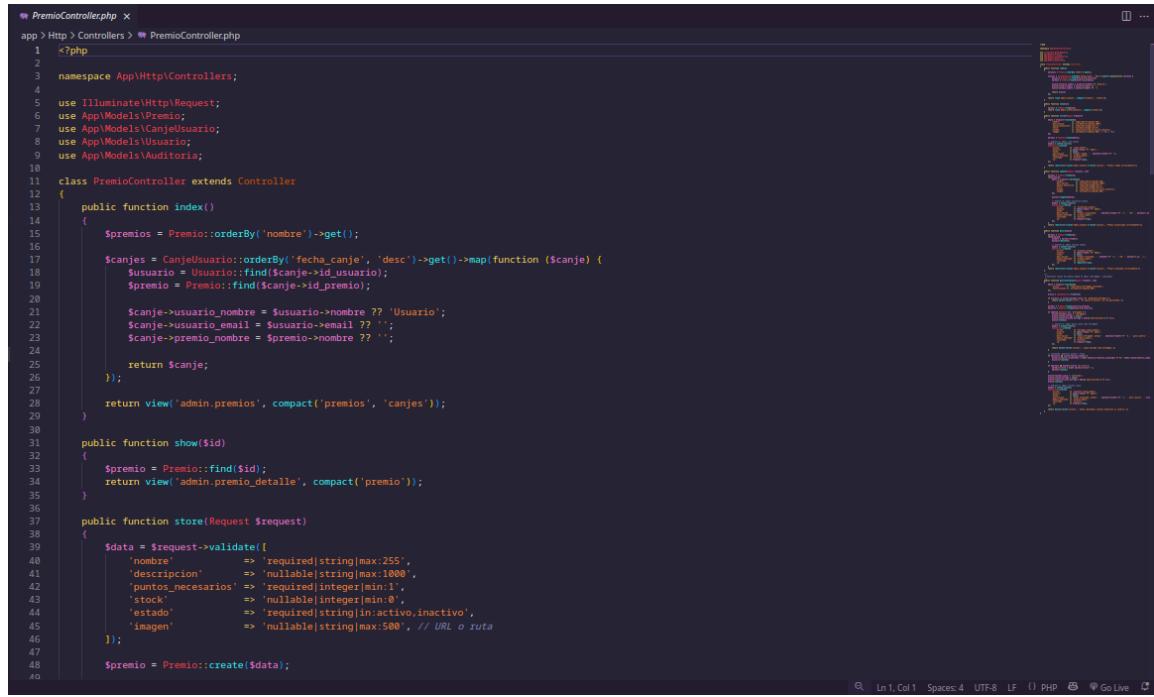
        return redirect()->route('admin.logros')->with('success', 'Logro creado correctamente');
    }

    public function update(Request $request, $id)
    {
        $logro = Logro::find($id);

        if (!$logro) {
            return redirect()->route('admin.logros')->with('error', 'El logro no existe.');
        }
    }
}
```

Explicacion: Administra los logros o “achievements” que los usuarios pueden alcanzar dentro de la plataforma. Tambien permite listar, crear, actualizar y eliminar logros desde el panel de administración. Cada vez que se crea, modifica o borra un logro, se registra la acción en la auditoría con la información del administrador que la hizo.

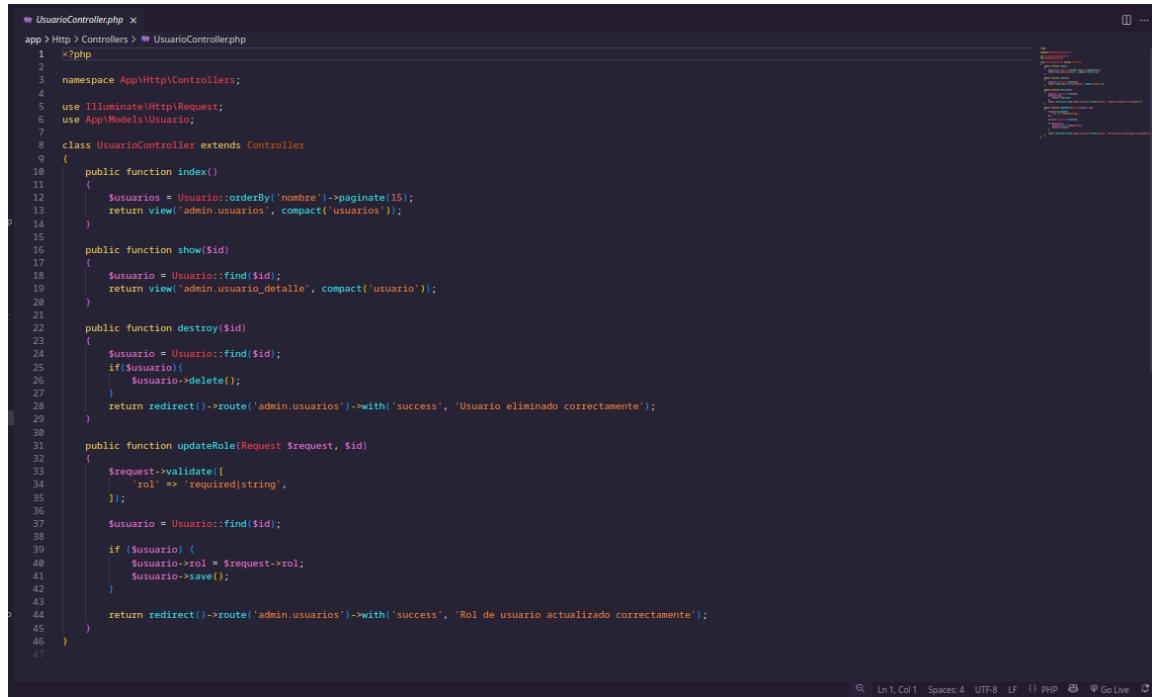
PremioController



```
  PremioController.php
app > Http > Controllers > PremioController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Premio;
7  use App\Models\CanjeDeUsuario;
8  use App\Models\Usuario;
9  use App\Models\Auditoria;
10
11 class PremioController extends Controller
12 {
13     public function index()
14     {
15         $premios = Premio::orderBy('nombre')->get();
16
17         $cambios = CanjeDeUsuario::orderBy('fecha_cambio', 'desc')->get()->map(function ($cambio) {
18             $usuario = Usuario::find($cambio->id_usuario);
19             $premio = Premio::find($cambio->id_premio);
20
21             $cambio->usuario_nombre = $usuario->nombre ?? 'Usuario';
22             $cambio->usuario_email = $usuario->email ?? '';
23             $cambio->premio_nombre = $premio->nombre ?? '';
24
25             return $cambio;
26         });
27
28         return view('admin.premios', compact('premios', 'cambios'));
29     }
30
31     public function show($id)
32     {
33         $premio = Premio::find($id);
34         return view('admin.premio_detalle', compact('premio'));
35     }
36
37     public function store(Request $request)
38     {
39         $data = $request->validate([
40             'nombre' => 'required|string|max:255',
41             'descripcion' => 'nullable|string|max:1000',
42             'puntos_necesarios' => 'required|integer|min:1',
43             'stock' => 'nullable|integer|min:0',
44             'estado' => 'required|string|in:activo,inactivo',
45             'imagen' => 'nullable|string|max:500', // URL o ruta
46         ]);
47
48         $premio = Premio::create($data);
49     }
50 }
```

Explicacion: Gestiona la parte de premios vista desde el usuario final: ver qué premios hay y canjearlos usando sus puntos acumulados. Tambien Muestra todos los premios disponibles en el panel. Cuando el usuario canjea uno, verifica que tenga suficientes puntos y que haya stock, crea un registro de canje, descuenta los puntos al usuario, actualiza el stock del premio y registra la operación en la auditoría

UsuariosController

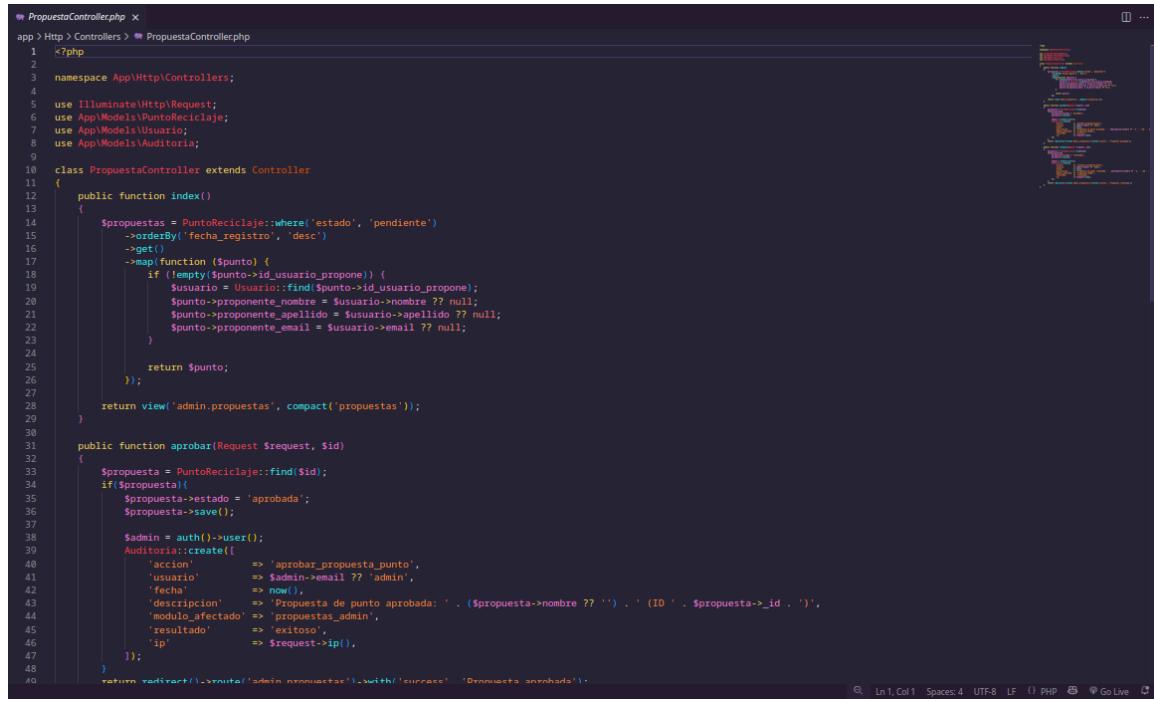


The screenshot shows a code editor with the file `UsuarioController.php` open. The code is written in PHP and defines a controller for managing users. It includes methods for listing users, viewing a user's details, deleting a user, and updating a user's role. The code uses Laravel's Eloquent ORM and blade templating.

```
 1  <?php
 2
 3  namespace App\Http\Controllers;
 4
 5  use Illuminate\Http\Request;
 6  use App\Models\Usuario;
 7
 8  class UsuarioController extends Controller
 9  {
10      public function index()
11      {
12          $usuarios = Usuario::orderBy('nombre')->paginate(15);
13          return view('admin.usuarios', compact('usuarios'));
14      }
15
16      public function show($id)
17      {
18          $usuario = Usuario::find($id);
19          return view('admin.usuario_detalle', compact('usuario'));
20      }
21
22      public function destroy($id)
23      {
24          $usuario = Usuario::find($id);
25          if ($usuario) {
26              $usuario->delete();
27          }
28          return redirect()->route('admin.usuarios')->with('success', 'Usuario eliminado correctamente');
29      }
30
31      public function updateRole(Request $request, $id)
32      {
33          $request->validate([
34              'rol' => 'required|string',
35          ]);
36
37          $usuario = Usuario::find($id);
38
39          if ($usuario) {
40              $usuario->rol = $request->rol;
41              $usuario->save();
42          }
43
44          return redirect()->route('admin.usuarios')->with('success', 'Rol de usuario actualizado correctamente');
45      }
46  }
```

Explicacion: Controla la vista principal del panel de usuario, donde se muestra un resumen de su actividad. Tambien Verifica que el usuario esté logueado y luego calcula sus puntos acumulados, cuántas entregas de reciclaje aprobadas tiene y cuántos canjes realizó. Con esa información construye un resumen que se muestra en la vista principal del panel.

PropuestaController



```
PropuestaController.php
app > Http > Controllers > PropuestaController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\PuntoReciclaje;
7 use App\Models\Usuario;
8 use App\Models\Auditoria;
9
10 class PropuestaController extends Controller
11 {
12     public function index()
13     {
14         $propuestas = PuntoReciclaje::where('estado', 'pendiente')
15             ->orderBy('fecha_registro', 'desc')
16             ->get()
17             ->map(function ($punto) {
18                 if (empty($punto->id_usuario_propone)) {
19                     $usuario = Usuario::find($punto->id_usuario_propone);
20                     $punto->proponente_nombre = $usuario->nombre ?? null;
21                     $punto->proponente_apellido = $usuario->apellido ?? null;
22                     $punto->proponente_email = $usuario->email ?? null;
23                 }
24
25                 return $punto;
26             });
27
28         return view('admin.propuestas', compact('propuestas'));
29     }
30
31     public function aprobar(Request $request, $id)
32     {
33         $propuesta = PuntoReciclaje::find($id);
34         if ($propuesta) {
35             $propuesta->estado = 'aprobada';
36             $propuesta->save();
37
38             $admin = auth()->user();
39             Auditoria::create([
40                 'accion' => 'aprobar_propuesta_punto',
41                 'usuario' => $admin->email ?? 'admin',
42                 'fecha' => now(),
43                 'descripcion' => "Propuesta de punto aprobada: " . ($propuesta->nombre ?? '') . ' (' . $ID . ' ' . $propuesta->_id . ')',
44                 'modulo_afectado' => 'propuestas_admin',
45                 'resultado' => 'exitoso',
46                 'ip' => $request->ip(),
47             ]);
48         }
49         return redirect('/admin.propuestas')->with('success', 'Propuesta aprobada');
50     }
}
```

Explicación: Permite al administrador revisar y decidir sobre las propuestas de nuevos puntos de reciclaje enviadas por los usuarios. También Carga los puntos de reciclaje con estado pendiente, agrega información del usuario que hizo la propuesta y los muestra en el panel admin. Desde aquí se puede aprobar o rechazar una propuesta, cambiando su estado y registrando la acción en la auditoría.

ReciclajeController

```
ReciclajeController.php ×
app > Http > Controllers > ReciclajeController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\PuntoReciclaje;
7  use App\Models\EntregaReciclaje;
8  use App\Models\ReporteRecolección;
9  use Illuminate\Support\Facades\Auth;
10 use App\Models\Auditoria;
11
12 class ReciclajeController extends Controller
13 {
14     // Mostrar la vista principal de "Gestionar mi reciclaje"
15     public function index()
16     {
17         return view('panel.reciclaje', [
18             'active' => 'reciclaje',
19         ]);
20     }
21
22     // Mostrar el formulario para proponer un nuevo punto
23     public function proponer()
24     {
25         return view('reciclaje.proponer');
26     }
27
28     // Guardar la propuesta de un nuevo punto
29     public function guardarPropuesta(Request $request)
30     {
31         $request->validate([
32             'nombre' => 'required|string|max:255',
33             'tipo' => 'required|string|max:100',
34             'direccion' => 'required|string|max:255',
35             'horario' => 'nullable|string|max:100',
36             'descripcion' => 'nullable|string|max:500',
37         ]);
38
39         PuntoReciclaje::registrarPropuesta($request->all());
40
41         // Auditoria: usuario propone un nuevo punto de reciclaje
42         $usuario = Auth::user();
43         Auditoria::create([
44             'accion' => 'crear_propuesta_punto',
45             'usuario' => $usuario->email ?? 'invitado',
46             'fecha' => now(),
47             'descripcion' => 'Propuesta de punto: ' . ($request->nombre ?? '') . ' - ' . ($request->direccion ?? ''),
48             'modulo_afectado' => 'panel_reciclaje',
49             'resultado' => 'aviso'
50         ]);
51     }
52 }
```

```
ReciclajeController.php ×
app > Http > Controllers > ReciclajeController.php
1
2
3     // Alias genérico para la ruta /reciclaje/guardar (por ahora usado para registrar entrega)
4     public function guardar(Request $request)
5     {
6         return $this->guardarEntrega($request);
7     }
8
9     // Mostrar el formulario para registrar una entrega de reciclaje
10    public function registrar()
11    {
12        return view('reciclaje.registrar');
13    }
14
15    // Guardar los datos de una entrega de reciclaje
16    public function guardarEntrega(Request $request)
17    {
18        $request->validate([
19            'tipo_residuo' => 'required|string|max:100',
20            'cantidad' => 'required|numeric|min:0.1',
21            'descripcion_general_materiales' => 'nullable|string|max:500',
22            'fecha_recolección_usuario' => 'nullable|date',
23            'ubicacion_recolección' => 'nullable|string|max:255',
24        ]);
25
26        $userId = Auth::id();
27
28        if (!$userId) {
29            return redirect()->route('login');
30        }
31
32        $cantidadKg = ($float) $request->cantidad; // el formulario trabaja en KG
33        $tipo = $request->tipo_residuo;
34        $puntosSugeridos = $this->calcularPuntosPorKg($tipo, $cantidadKg);
35
36        $entrega = EntregaReciclaje::create([
37            'id_usuario'          => (string) $userId,
38            'tipo_residuo'        => $tipo,
39            'cantidad_kg'         => $cantidadKg,
40            'puntos_sugeridos'   => $puntosSugeridos,
41            'puntos_otorgados'   => 0,
42            'estado'              => 'pendiente',
43            'fecha_registro'     => now(),
44            'descripcion_general_materiales' => $request->descripcion_general_materiales,
45            'fecha_recolección_usuario' => $request->fecha_recolección_usuario,
46            'ip_reporte'          => $request->ip(),
47            'ubicacion_recolección' => $request->ubicacion_recolección,
48        ]);
49    }
50 }
```

```

  ReciclajeController.php ×
app > Http > Controllers > ReciclajeController.php
13
127 {
171     return back()->with('success', 'Entrega registrada y enviada al administrador para validación.');
173 }
174 /**
175 * Calcular puntos sugeridos según tipo de residuo y cantidad en KG.
176 */
177
178 protected function calcularPuntosPorKg(string $tipoResiduo, float $kg): int
179 {
180     $tipo = mb_strtolower($tipoResiduo);
181
182     // Tabla simple de puntos por kg según material
183     $tabla = [
184         'plástico' => 10,
185         'plástico' => 10,
186         'pet' => 12,
187         'papel' => 8,
188         'cartón' => 8,
189         'cartón' => 8,
190         'vidrio' => 6,
191         'metal' => 12,
192         'aluminio' => 12,
193         'orgánico' => 4,
194         'orgánico' => 4,
195     ];
196
197     $puntosPorKg = 5; // valor por defecto
198
199     foreach ($tabla as $clave => $valor) {
200         if (str_contains($tipo, $clave)) {
201             $puntosPorKg = $valor;
202             break;
203         }
204     }
205
206     return (int) round($kg * $puntosPorKg);
207 }
208 }
209

```

Explicacion: Maneja el módulo de “Gestionar mi reciclaje” en el panel de usuario: proponer puntos, reportar problemas y registrar entregas de material recicitable. Tambien Muestra la pantalla principal de reciclaje, el formulario para proponer nuevos puntos y el formulario para reportar problemas en un punto existente. También permite registrar entregas de reciclaje, calcular los puntos sugeridos según el tipo de residuo y la cantidad, crear el registro de la entrega y dejar constancia de todas estas acciones en la auditoría.

ReporteController

```
ReporteController.php ×
app > Http > Controllers > ReporteController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\ReporteRecoleccion;
7
8  class ReporteController extends Controller
9  {
10     public function index()
11     {
12         $reportes = ReporteRecoleccion::all();
13         return view('admin.reportes', compact('reportes'));
14     }
15
16     public function show($id)
17     {
18         $reporte = ReporteRecoleccion::find($id);
19         return view('admin.reporte_detalle', compact('reporte'));
20     }
21
22     public function resolver(Request $request, $id)
23     {
24         $reporte = ReporteRecoleccion::find($id);
25         if($reporte){
26             $reporte->estado = $request->estado ?? 'resuelto';
27             $reporte->save();
28         }
29         return redirect()->route('admin.reportes')->with('success', 'Reporte actualizado');
30     }
31 }
32 }
```

Explicacion: Administra los reportes de recolección o incidencias que se generan sobre los puntos de reciclaje. Tambien Lista todos los reportes en el panel de administrador, permite ver el detalle de un reporte específico y cambiar su estado (por ejemplo a resuelto), guardando el cambio y redirigiendo a la lista de reportes.

UsuarioController

```
必不可  
 UserController.php x  
 app > ~/Documentos/repo-github/reciclapp/controller.php  
 1 laravel/app/Http/Controllers/UsuarioController.php (preview ⓘ)  
 2  
 3 namespace App\Http\Controllers;  
 4  
 5 use Illuminate\Http\Request;  
 6 use App\Models\Usuario;  
 7  
 8 class UsuarioController extends Controller  
 9 {  
10     public function index()  
11     {  
12         $usuarios = Usuario::orderBy('nombre')->paginate(15);  
13         return view('admin.usuarios', compact('usuarios'));  
14     }  
15  
16     public function show($id)  
17     {  
18         $usuario = Usuario::find($id);  
19         return view('admin.usuario_detalle', compact('usuario'));  
20     }  
21  
22     public function destroy($id)  
23     {  
24         $usuario = Usuario::find($id);  
25         if($usuario){  
26             $usuario->delete();  
27         }  
28         return redirect()->route('admin.usuarios')->with('success', 'Usuario eliminado correctamente');  
29     }  
30  
31     public function updateRole(Request $request, $id)  
32     {  
33         $request->validate([  
34             'rol' => 'required|string',  
35         ]);  
36  
37         $usuario = Usuario::find($id);  
38  
39         if ($usuario) {  
40             $usuario->rol = $request->rol;  
41             $usuario->save();  
42         }  
43  
44         return redirect()->route('admin.usuarios')->with('success', 'Rol de usuario actualizado correctamente');  
45     }  
46 }  
47 }
```

Explicacion: Gestiona los usuarios del sistema desde el panel de administración. Tambien Permite listar usuarios con paginación, ver el detalle de cada usuario, eliminar usuarios y cambiar su rol (por ejemplo de usuario normal a administrador). Todas estas acciones se reflejan en la base de datos de usuarios

Diccionario de Datos

Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id_detalle_reporte	int(11)		Identificador único.
Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id	bigint(20)		Identificador único.
Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id	bigint(20)		Identificador único.
Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id	varchar(255)		Identificador único.
Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id_logro	int(11)		Identificador único.

Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id	int(10)		Identificador único.
Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
email	varchar(255)		Correo electrónico del usuario.
Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id_premio	int(11)		Identificador único.

Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id_punto	int(11)		Identificador único.
id_reporte_problema_punto	int(11)		Identificador único.
id_reporte_recolección	int(11)		Identificador único.
id	varchar(255)		Identificador único.
id_tipo_residuo	int(11)		Identificador único.
id_usuario	varchar(20)		Identificador único.

Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id_usuario_logro	int(11)		Identificador único.

Estructura de Datos – MongoDB

El proyecto ReciclApp utiliza **MongoDB** como base de datos NoSQL debido a su flexibilidad, esquema dinámico y facilidad para manejar documentos complejos sin requerir estructuras rígidas. Esto permite registrar actividades, reportes, puntos de reciclaje, usuarios, auditorías y todo el ecosistema del sistema de manera eficiente.

A continuación, se presenta la **estructura real de datos** del proyecto, organizada por colecciones, ejemplos de documentos y una explicación clara del propósito de cada una.

Colección: actividades

Esta colección almacena los eventos o actividades relacionadas con el reciclaje que se ofrecen a los usuarios.

Sirve para mostrar en la aplicación actividades como jornadas, talleres o campañas ambientales.

Ejemplo de documento:

```
{
  "_id": "691fb24de2aa4ac9ad08bd6f",
  "titulo": "Prueba",
  "descripcion": "123A",
  "fecha": "2025-11-30T12:00",
  "lugar": "Centro comercial el Caraño",
  "estado": "activa",
  "updated_at": "2025-11-21T00:29:01.819Z",
  "created_at": "2025-11-21T00:29:01.819Z"
}
```

Colección: auditoria

Registra acciones administrativas importantes dentro del sistema (por ejemplo: creación de usuarios, cambios críticos, modificaciones en tablas).

Sirve para mantener trazabilidad y control interno de cambios realizados por administradores o procesos automáticos.

Ejemplo de documento:

```
{  
  "_id": "68fb0d3990cd161799059389",  
  "id_registro_auditoria": 1,  
  "id_usuario_afectado": "admin001",  
  "accion_realizada": "NUEVO USUARIO REGISTRADO",  
  "tabla_modificada": "usuarios",  
  "id_registro_modificado": "admin001",  
  "detalles_accion": "Usuario registrado: admin@gmail.com, Rol: administrador",  
  "ip_origen": "SYSTEM_TRIGGER",  
  "fecha_hora_auditoria": "2025-06-04 06:56:18"  
}
```

Colección: auditorias

Registra los accesos al sistema, especialmente eventos de inicio de sesión (login) y logout.

Se usa para monitorear seguridad, intentos fallidos y accesos sospechosos.

Ejemplo de documento:

```
{  
  "_id": "69170a283fbf75e0de0abab6",  
  "accion": "login",  
  "usuario": "usuario1@gmail.com",  
  "fecha": "2025-11-14T10:53:28.134Z",  
  "descripcion": "Inicio de sesión correcto",  
  "modulo_afectado": "auth",  
  "resultado": "exitoso",  
  "ip": "127.0.0.1",  
  "updated_at": "2025-11-14T10:53:28.135Z",  
  "created_at": "2025-11-14T10:53:28.135Z"  
}
```

Colección: canjes_usuario

Guarda el historial de canjes hechos por los usuarios, incluyendo puntos gastados, fecha del canje y estado.

Sirve para controlar la entrega de premios y verificar cuántos puntos se usan.

Ejemplo de documento:

```
{  
  "_id": "68fb0d3890cd161799059370",  
  "id_canje": 1,  
  "id_usuario": "1077427620",  
  "id_premio": 1,  
  "fecha_canje": "2025-06-04 16:08:19",  
  "puntos_usados": 500,  
  "codigo_canje_fisico": "CANJE_6840b5c36705f",  
  "estado_canje": "pendiente_entrega",  
  "fecha_entrega": null,  
  "observaciones_entrega": null  
}
```

Colección: detalle_reporte_recolección

Desglosa los residuos individuales dentro de un reporte de recolección.

Sirve para calcular la cantidad exacta de kilos por tipo de residuo y los puntos ganados por cada uno.

Ejemplo de documento:

```
{  
  "_id": "68fb0d3990cd16179905937b",  
  "id_detalle_reporte": 1,  
  "id_reporte_recolección": 1,  
  "id_tipo_residuo": 1,  
  "cantidad_kg_específica": "500.00",  
  "puntos_calculados_detalle": 7500  
}
```

Colección: entrega_reciclajes

Almacena las entregas de reciclaje que realizan los usuarios junto con los puntos otorgados.

Representa el núcleo del sistema porque registra las recolecciones verificadas.

Ejemplo de documento:

```
{  
  "_id": "691fb1c6e2aa4ac9ad08bd6b",  
  "id_usuario": "68fb0d3890cd161799059367",  
  "tipo_residuo": "Botellas de ron",  
  "cantidad_kg": 3,  
  "puntos_sugeridos": 15,  
  "puntos_otorgados": "15",  
  "estado": "aprobada",  
  "fecha_registro": "2025-11-21T00:26:46.752Z",  
  "descripcion_general_materiales": "d",  
  "fecha_recolección_usuario": "2020-11-19",  
  "ip_reporte": "127.0.0.1"  
}
```

Colección: logros

Define los logros que los usuarios pueden desbloquear dependiendo de sus actividades (primera recolección, cantidad de kilos, etc.).

Sirve para gamificar el sistema y motivar a los usuarios a reciclar más.

Ejemplo de documento:

```
{  
  "_id": "68fb0d3990cd161799059386",  
  "id_logro": 1,  
  "nombre_logro": "Reciclador Novato",  
  "descripcion_logro": "Completa tu primera recolección aprobada.",  

```

Colección: premios

Contiene los premios disponibles para canjear con puntos, incluyendo nombre, descripción, imagen y puntos requeridos.

Es usada para el módulo de canje de recompensas.

Ejemplo de documento:

```
{  
  "_id": "68fb0d3890cd16179905936d",  
  "id_premio": 3,  
  "nombre_premio": "Recarga Celular $10.000",  
  "descripcion_premio": "Recarga de saldo para operadores seleccionados.",  

```

```
"puntos_requeridos": 1000,  
"ruta_imagen_premio": "media/recarga.png",  
"activo": 1  
}
```

Colección: puntos_reciclaje

Registra los puntos físicos donde se pueden realizar entregas de reciclaje.
Incluye dirección, horarios, tipo de residuos aceptados y una descripción.
Se usa en el mapa y en el buscador de puntos.

Ejemplo de documento:

```
{  
  "_id": "68fbf5b62ef51fc2840d3e92",  
  "nombre": "Parque Central",  
  "tipo": "Basura reciclable",  
  "direccion": "Calle 123 #45-67",  
  "horario": "08:00 - 18:00",  
  "descripcion": "Punto de reciclaje cerca de la plaza central.",  
  "fecha_registro": "2025-10-24T21:55:02.746Z"  
}
```

Colección: reporte_recolección

Almacena reportes enviados por los usuarios respecto a problemas en puntos de reciclaje (zonas peligrosas, puntos llenos, daños, etc.).
Permite a los administradores gestionar incidencias.

Ejemplo de documento:

```
{  
  "_id": "691fb16fe2aa4ac9ad08bd68",  
  "id_usuario": null,  
  "id_punto": "691fb154e2aa4ac9ad08bd66",  
  "fecha_reporte": "2025-11-21T00:25:19.641Z",  
  "estado": "en_progreso",  
  "observaciones": "Tipo de problema: zona_peligrosa | Descripción: peligro, roban."  
}
```

Colección: sessions

Es usada por Laravel para manejar sesiones cuando estás usando MongoDB como storage.

Guarda datos temporales de login, IP, actividad y autenticación.

Ejemplo de documento:

```
{  
  "_id": "f4W8H3hropX6XVENBy67m687A1ujWAQnlxpNv9xv",  
  "last_activity": 1763696034,  
  "ip_address": "127.0.0.1"  
}
```

Colección: tipos_residuos

Define los tipos de residuos aceptados (plástico, cartón, vidrio, etc.) y los puntos que otorgan por kilo.

Sirve para calcular puntajes y clasificar materiales.

Ejemplo de documento:

```
{  
  "_id": "68fb0d3990cd16179905937c",  
  "id_tipo_residuo": 1,  
  "nombre": "Plástico",  
  "descripcion": "Botellas PET, envases, tapas, etc.",  
  "puntos_por_kg": "15.00"  
}
```

Colección: usuarios

Guarda toda la información de los usuarios registrados: nombre, correo, contraseña en hash, rol, puntos acumulados, estado y fecha de registro.

Es la colección principal del sistema de autenticación y puntajes.

Ejemplo de documento:

```
{  
  "_id": "68fb0d3890cd161799059362",  
  "id_usuario": "00000001",  
  "nombre": "Kenneth",  
  "apellido": "Wnfried",  
  "email": "rances@gmail.com",  
  "rol": "usuario",  
  "puntos_acumulados": 0,  
  "fecha_registro": "2025-09-18 15:04:04",  
  "estado": "activo"  
}
```