

Fundación Universitaria Claretiana.

Facultad de ingeniería de sistemas

Kenneth Ranses 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.

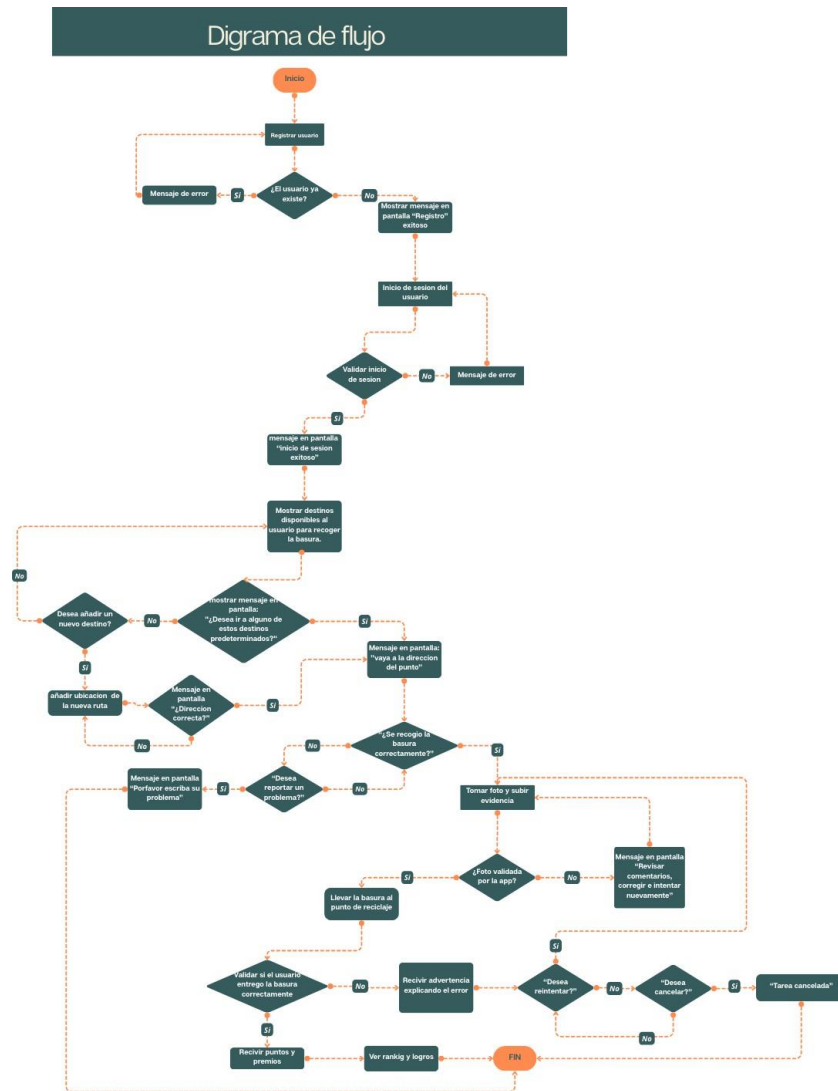
Casos de uso – Ciudadano

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

Casos de uso – Administrador

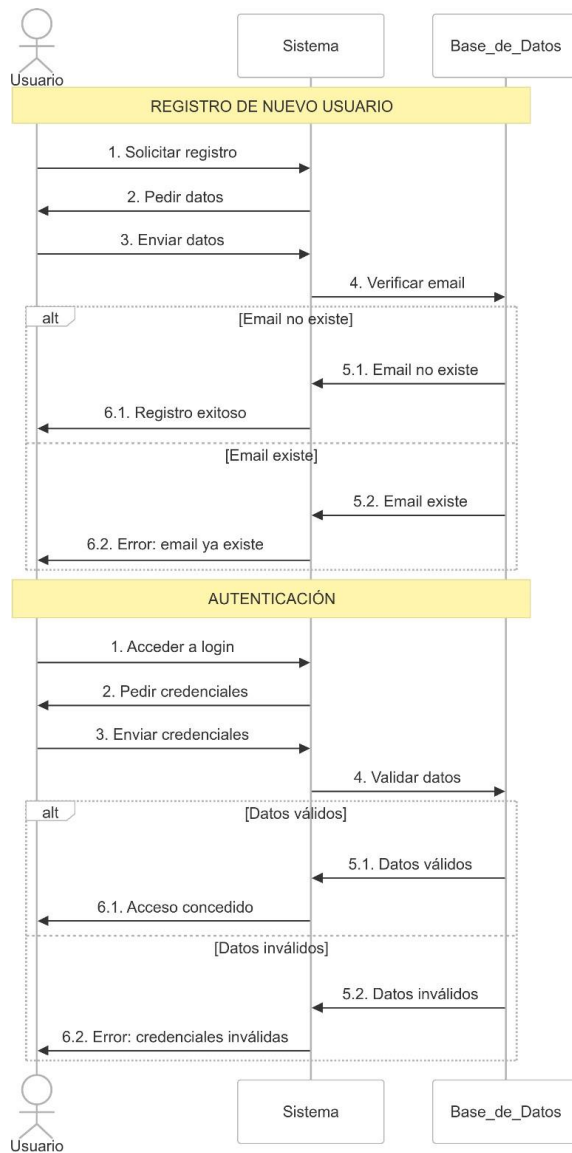
- Registro e inicio de sesión.
- Gestionar usuarios.
- Validar información.
- Crear actividades.

Diagrama de Flujo General

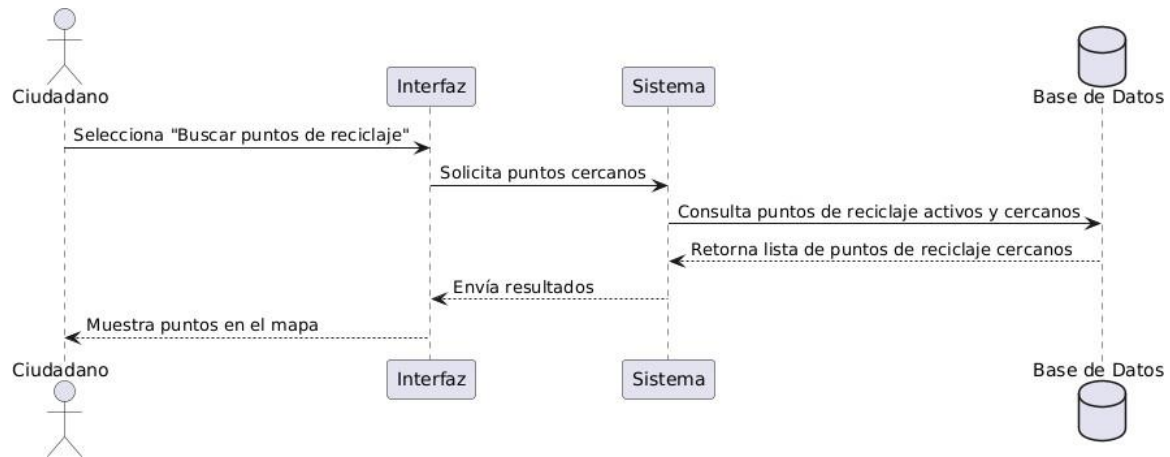


Diagramas de secuencia

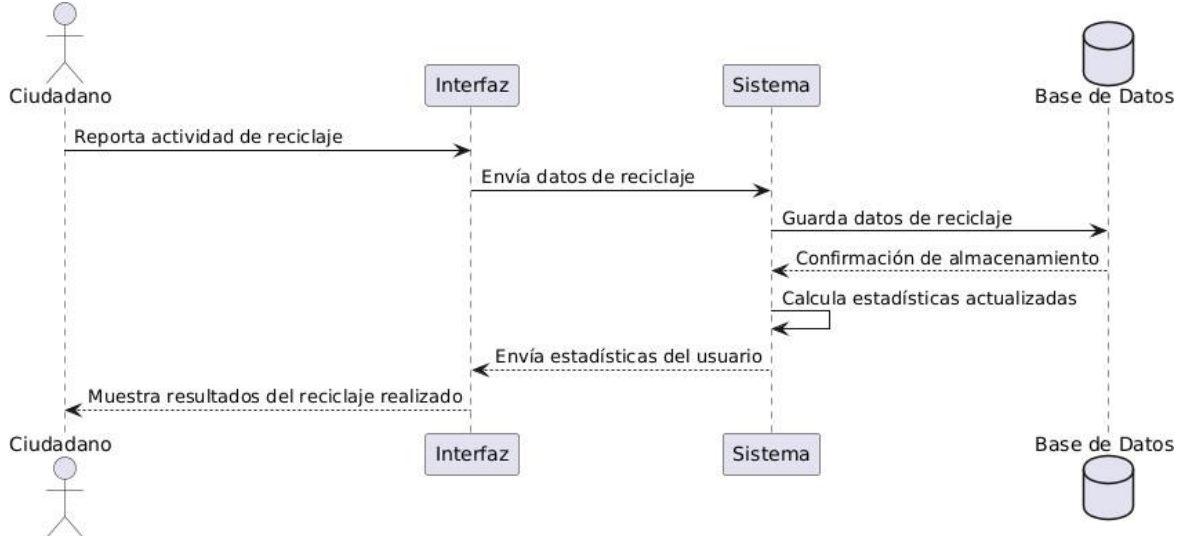
Registro e inicio de sesión



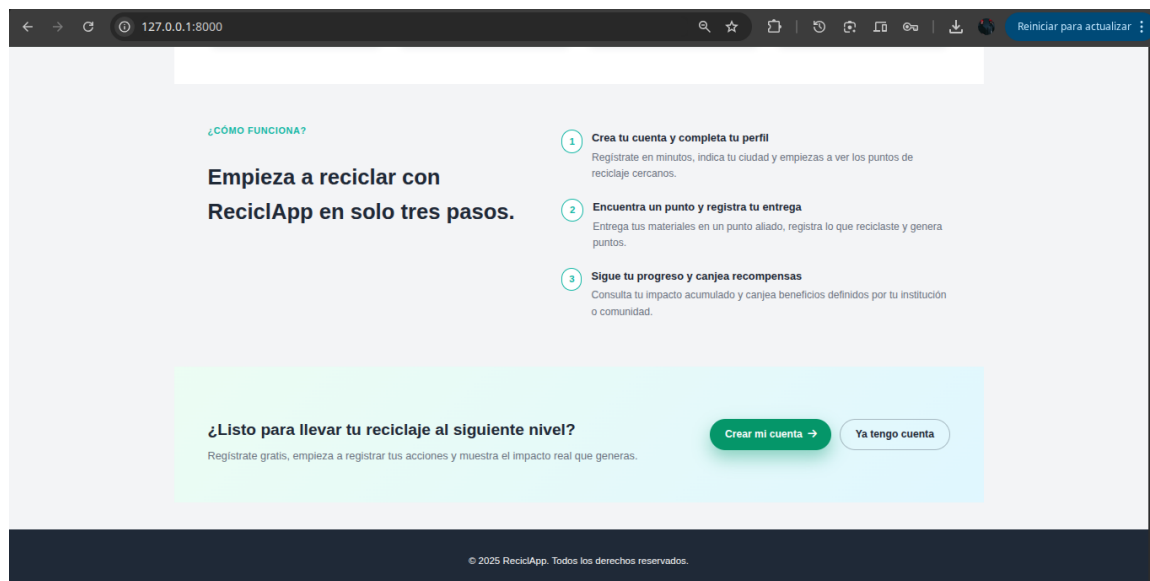
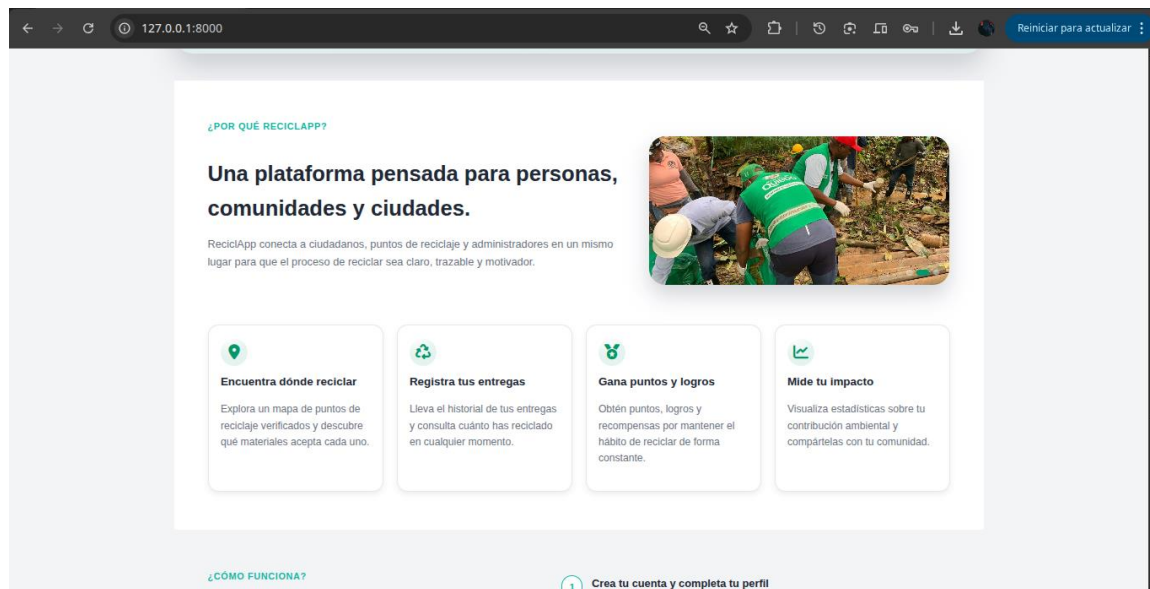
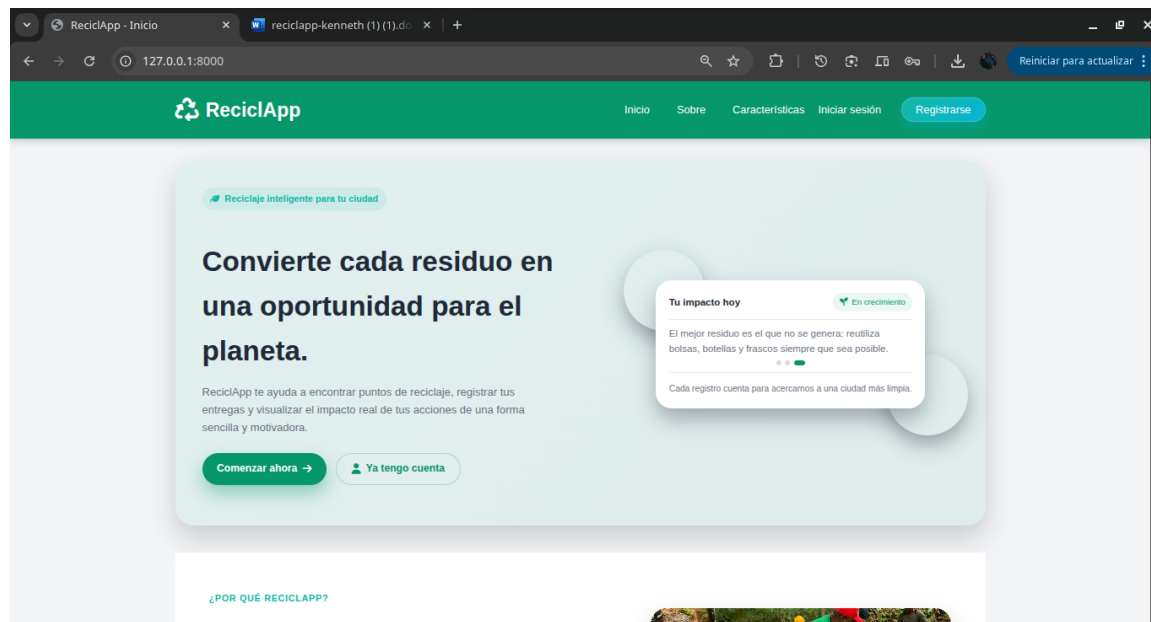
Consulta de punto de reciclaje



Carga de estadísticas




Pagina de inicio



Login

← → ↻ 127.0.0.1:8000/login 🔍 ☆ 📄 🕒 🔄 📱 🖨️ ⌵ 🧑 Reiniciar para actualizar ⋮



Vuelve a conectar con tu impacto.

Ingresa a tu cuenta para registrar nuevas entregas, revisar tus puntos acumulados y descubrir nuevas formas de reciclar en tu comunidad.

- Accede a tu panel de reciclaje personalizado.
- Consulta puntos de reciclaje y propuestas.
- Sigue de cerca tus recompensas y logros.

Iniciar sesión

Ingresa tus credenciales para acceder a tu panel.

Correo Electrónico


Contraseña

Iniciar Sesión

¿Aún no tienes cuenta? [Crear una cuenta](#)

Register



← → ↻ 127.0.0.1:8000/register 🔍 ☆ 📄 🕒 🔄 📱 🖨️ ⌵ 🧑 Reiniciar para actualizar ⋮




Comienza tu ruta de reciclaje.

Crea una cuenta para registrar tus entregas, ganar puntos y sumarte a una comunidad que recicla de forma inteligente.

- Visualiza cómo tus acciones reducen el impacto ambiental.
- Forma parte de una red de recicladores responsables.
- Desbloquea logros y recompensas por mantener el hábito.

 Datos protegidos  Cuenta gratuita

 Accede desde cualquier dispositivo

Crea tu cuenta

1 Datos personales 2 Seguridad

ID / Cédula *

Nombre *

Apellido *

Correo *

Siguiendo

¿Ya tienes cuenta? [Inicia sesión](#)

ReciclApp

Comienza tu ruta de reciclaje.

Crea una cuenta para registrar tus entregas, ganar puntos y sumarte a una comunidad que recicla de forma inteligente.

Visualiza cómo tus acciones reducen el impacto ambiental.

Forma parte de una red de recicladores responsables.

Desbloquea logros y recompensas por mantener el hábito.

Datos protegidos

Cuenta gratuita

Accede desde cualquier dispositivo

Crea tu cuenta

1 Datos personales

2 Seguridad

Contraseña *

Confirmar Contraseña *

Fecha de nacimiento

dd / mm / aaaa

Atrás

Registrarse

¿Ya tienes cuenta? Inicia sesión

Panel del usuario

ReciclApp

Inicio

Sobre

Características

Cerrar sesión

ReciclApp

Resumen

Gestionar reciclaje

Canjear premios

Puntos de reciclaje

Actividades y eventos

Kenneth

ken123@gmail.com

Gestionar reciclaje

Revisa tu progreso y accede rápidamente a las principales acciones de la plataforma.

Puntos acumulados

15

Entregas aprobadas

1

Canjes realizados

0

Impacto de tu reciclaje

Una vista rápida de cómo tus acciones podrían reflejarse en el impacto ambiental.

Kg reciclados por mes

Ene

Feb

Mar

Abr

12 kg

18 kg

24 kg

28 kg

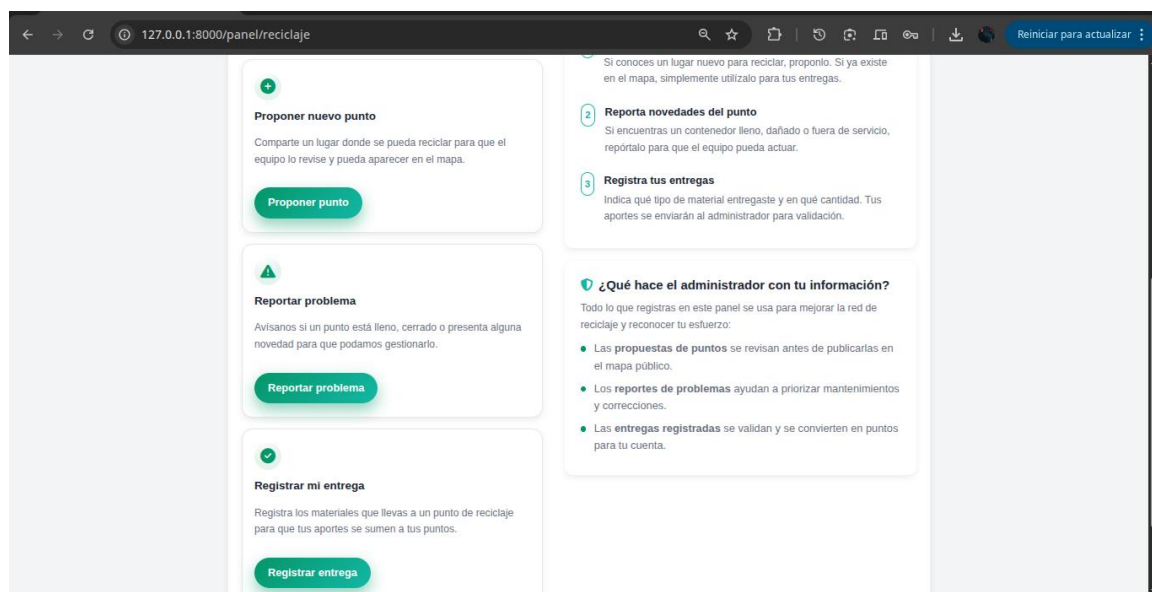
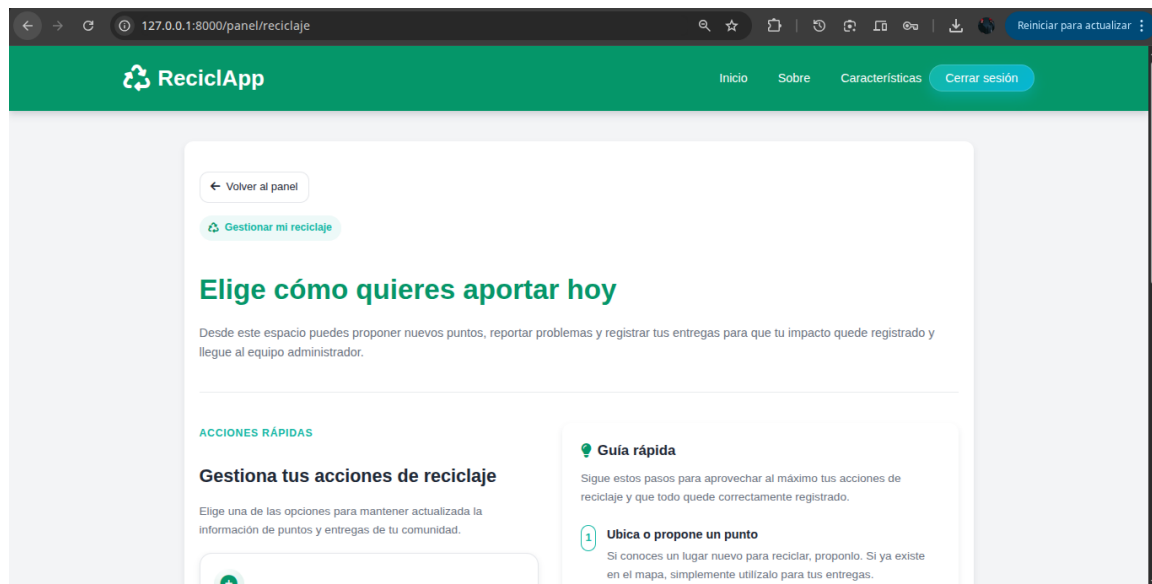
Beneficios estimados

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



Proponer nuevo punto

ReciclApp

Inicio

Sobre

Características

Cerrar sesión

← Volver

Proponer 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 validarlo.

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.

Enviar propuesta

← Cancelar y volver



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.

© 2025 ReciclApp. Todos los derechos reservados.

Reportar problema

ReciclApp

Inicio

Sobre

Características

Cerrar sesión

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

Selecciona un punto

Tipo de problema

Dirección incorrecta

Comentarios adicionales

Detalles adicionales, si los hay...

Enviar reporte

Volver

¿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.

© 2025 ReciclApp. Todos los derechos reservados.

Registrar tu entrega

ReciclApp

[Inicio](#)[Sobre](#)[Características](#)[Cerrar sesión](#)

Registrar mi entrega

Registra tu aporte al planeta

Ingresa 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

dd / mm / aaaa

Lugar de la recolección (opcional)


Ej: Punto limpio del parque central, supermercado X...

Descripción general de los materiales

Ej: Botellas de plástico, cartón, vidrio limpio...

Registrar entrega

Volver



Imagina que cada bolsa que entregas 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.).

© 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 Quidbo. 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 mas lugares a medida que la comunidad crece.

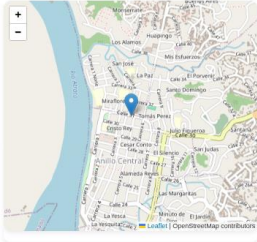
Parque Central de Quidbo

Punto fijo de reciclaje en el parque central de la ciudad.
Tipo: plastic, paper 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.

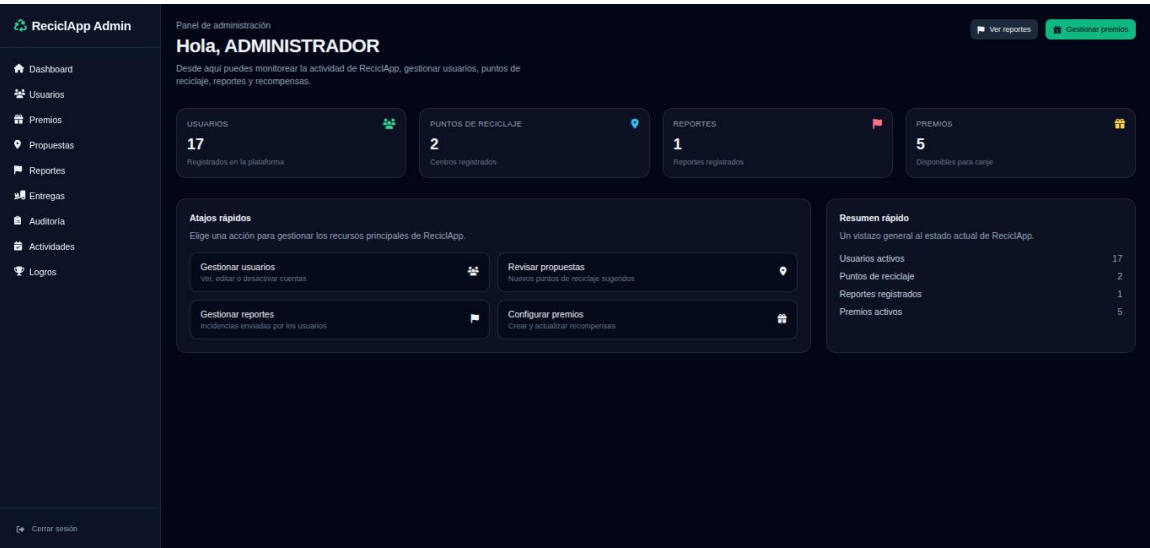
Si conoces un nuevo lugar donde podria instalarse un punto, puedes usar la opción **Proponer nuevo punto** en tu panel.



© 2025 ReciclApp. Todos los derechos reservados.

Actividades y eventos

Panel del administrador



Usuarios

ReciclApp Admin		Gestión de usuarios				
		Administra roles y estado de los usuarios registrados en ReciclApp.				
		Nombre	Email	Rol	Estado	Acciones
		ADMINISTRADOR AD ID: 0000000	administrador@gmail.com	Administrador	Activo	<button>Passar a usuario</button> <button>Eliminar</button>
		Admin SI ID: 00000002	admin2@gmail.com	Administrador	Activo	<button>Passar a usuario</button> <button>Eliminar</button>
		Admin Principal ID: admin001	admin@gmail.com	Administrador	Activo	<button>Passar a usuario</button> <button>Eliminar</button>
		Kenneth Wnfried ID: 000000011	admin0@gmail.com	Usuario	Activo	<button>Hacer admin</button> <button>Eliminar</button>
		Kenneth ID:		Usuario	Activo	<button>Hacer admin</button> <button>Eliminar</button>
		Kenneth Wnfried ID: 107742777	usuario1@gmail.com	Usuario	Activo	<button>Hacer admin</button> <button>Eliminar</button>
		Kenneth Wnfried ID: 00000001	ranses@gmail.com	Usuario	Activo	<button>Hacer admin</button> <button>Eliminar</button>
		Kenneth Wnfried ID: 1077427629	ken123@gmail.com	Usuario	Activo	<button>Hacer admin</button> <button>Eliminar</button>
		Ramses Wnfried ID: 1077427621	admin1@gmail.com	Administrador	Activo	<button>Passar a usuario</button> <button>Eliminar</button>
		admin admin ID: 10101010	admin3@gmail.com	Usuario	Activo	<button>Hacer admin</button> <button>Eliminar</button>
		Juan rada ID: admin121470	jrada7674@gmail.com	Usuario	Activo	<button>Hacer admin</button> <button>Eliminar</button>
		ke ra ID: 123	ad@gmail.com	Administrador	Activo	<button>Passar a usuario</button> <button>Eliminar</button>
		kene ene ID: 123456789	root@gmail.com	Administrador	Activo	<button>Passar a usuario</button> <button>Eliminar</button>
		kenneth arriaga ID: 1077427620	kenneth1002@gmail.com	Usuario	Activo	<button>Hacer admin</button> <button>Eliminar</button>
		prueba prueba 2 ID: 10122	ass@gmail.com	Usuario	Activo	<button>Hacer admin</button> <button>Eliminar</button>
		Showing 1 to 15 of 17 results				
		<div>< 1 2 ></div>				

Gestion de premios

ReciclApp Admin

Dashboard

Usuarios

Premios

Propuestas

Reportes

Entregas

Auditoria

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
	limitado	inactivo	Sin imagen	<div><div>Nombre</div><div>Puntos</div><div>Stock (vacío = limitado)</div><div>Ruta imagen (opcional)</div><div>Activo</div><div>Guardar</div></div>	<div>Eliminar</div>
	limitado	inactivo	Sin imagen	<div><div>Nombre</div><div>Puntos</div><div>Stock (vacío = limitado)</div><div>Ruta imagen (opcional)</div><div>Activo</div><div>Guardar</div></div>	<div>Eliminar</div>

Nuevo premio

Nombre

Descripción

Puntos necesarios

Stock (opcional)

Vacio = limitado

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

ReciclApp Admin

Dashboard

Usuarios

Premios

Propuestas

Reportes

Entregas

Auditoria

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
bugue	Mixto	niño Jesus	12:00 PM		Usuario desconocido	21/11/2025 00:14	<div><div>✓ Aceptar</div><div>✗ Rechazar</div></div>

Gestion de reportes

ReciclApp Admin							
Reportes de recolección							
Listado de reportes enviados por los usuarios para seguimiento y resolución.							
ID	Usuario	Punto	Fecha	Estado	Observaciones	Acciones	
691b10f2aa4c9ad0b068	N/D	691b154e2aa4c9ad0b066	23/11/2025 00:25	En progreso	Tipo de problema: zona_peligrosa Descripción: peligro, reb...	Ver	

Creación de modelos

- modelo de auditoria.

-

```
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 canje 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 > DetalleReporteRecoleccion.php
1  <?php
2
3  namespace App\Models;
4
5  use Jenssegers\Mongodb\Eloquent\Model;
6
7  class DetalleReporteRecoleccion extends Model
8  {
9      protected $connection = 'mongodb';
10     protected $collection = 'detalle_reporte_recoleccion';
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 > ReporteRecoleccion.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 ReporteRecoleccion extends Eloquent
9  {
10     use HasFactory;
11
12     protected $connection = 'mongodb';
13     protected $collection = 'reportes_recoleccion_usuario';
14
15     protected $fillable = [
16         'id_usuario',
17         'id_punto',
18         'fecha_reporte',
19         'estado',
20         'observaciones',
21     ];
22 }
23
```










- 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

Esquema de colores - ReciclApp

	<code>--flujo-primary-color</code> <code>#059669</code> Botones principales, íconos, títulos		<code>--flujo-danger-color</code> <code>#ef4444</code> Errores, estados críticos	
	<code>--flujo-secondary-color</code> <code>#14B8A6</code> Acentos, hovers, chips		<code>--flujo-dark-color</code> <code>#1F2937</code> Texto principal, fondos oscuros	
	<code>--flujo-accent-cyan</code> <code>#06B6D4</code> Gradientes, acentos adicionales		<code>--flujo-text-light</code> <code>#6b7280</code> Texto secundario	
	<code>--flujo-warning-color</code> <code>#facc15</code> Mensajes/estados de advertencia		<code>--flujo-bg-color</code> <code>#F3F4F6</code> Fondos secundarios	

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

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

```
AuditoriaController.php X
app > Http > Controllers > AuditoriaController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Auditoria; // Debes crear este modelo
7
8  class AuditoriaController extends Controller
9  {
10     public function index()
11     {
12         $logs = Auditoria::orderBy('fecha', 'desc')
13             ->limit(100)
14             ->get();
15         return view('admin.auditoria', compact('logs'));
16     }
17 }
18
```

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             $request->session()->regenerate();
36
37             $rol = trim(strtolower(Auth::user()->rol ?? ''));
38
39             Auditoria::create([
40                 'accion' => 'login',
41                 'usuario' => $credentials['email'],
42                 'fecha' => now(),
43                 'descripcion' => 'Inicio de sesión correcto',
44                 'modulo_afectado' => 'auth',
45                 'resultado' => 'exitoso',
46                 'ip' => $request->ip(),
47             ]);
48         }
49     }
50 }
```

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

Explicacion: Maneja todo el proceso de autenticación: iniciar sesión, registrar nuevos usuarios y cerrar sesión. Tambien 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.');
```

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

```
LnController.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 X
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\CanjeUsuario;
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         $canjes = CanjeUsuario::orderBy('fecha_canje', 'desc')->get()->map(function ($canje) {
18             $usuario = Usuario::find($canje->id_usuario);
19             $premio = Premio::find($canje->id_premio);
20
21             $canje->usuario_nombre = $usuario->nombre ?? 'Usuario';
22             $canje->usuario_email = $usuario->email ?? '';
23             $canje->premio_nombre = $premio->nombre ?? '';
24
25             return $canje;
26         });
27
28         return view('admin.premios', compact('premios', 'canjes'));
29     }
30
31     public function show($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

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\User;
7
8 class UsersController extends Controller
9 {
10     public function index()
11     {
12         $users = User::orderBy('name')->paginate(15);
13         return view('admin.users', compact('users'));
14     }
15
16     public function show($id)
17     {
18         $user = User::find($id);
19         return view('admin.user_detail', compact('user'));
20     }
21
22     public function destroy($id)
23     {
24         $user = User::find($id);
25         if($user){
26             $user->delete();
27         }
28         return redirect()->route('admin.users')->with('success', 'User eliminated correctly');
29     }
30
31     public function updateRole(Request $request, $id)
32     {
33         $request->validate([
34             'role' => 'required|string',
35         ]);
36
37         $user = User::find($id);
38
39         if ($user) {
40             $user->role = $request->role;
41             $user->save();
42         }
43
44         return redirect()->route('admin.users')->with('success', 'Role of user updated correctly');
45     }
46 }
47
```

Explicación: Controla la vista principal del panel de usuario, donde se muestra un resumen de su actividad. También 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 x
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             return $punto;
25         });
26     }
27     return view('admin.propuestas', compact('propuestas'));
28 }
29
30 public function aprobar(Request $request, $id)
31 {
32     $propuesta = PuntoReciclaje::find($id);
33     if ($propuesta) {
34         $propuesta->estado = 'aprobada';
35         $propuesta->save();
36
37         $admin = auth()->user();
38         Auditoria::create([
39             'accion' => 'aprobar_propuesta_punto',
40             'usuario' => $admin->email ?? 'admin',
41             'fecha' => now(),
42             'descripcion' => 'Propuesta de punto aprobada: ' . ($propuesta->nombre ?? '') . ' (ID: ' . $propuesta->id . ')',
43             'modulo_afectado' => 'propuestas_admin',
44             'resultado' => 'exitoso',
45             'ip' => $request->ip(),
46         ]);
47     }
48     return redirect()->route('admin.propuestas')->with('success' -> 'Propuesta aprobada');
49 }
```

Explicacion: Permite al administrador revisar y decidir sobre las propuestas de nuevos puntos de reciclaje enviadas por los usuarios. Tambien 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 x
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\ReporteRecoleccion;
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             'realizado' => 'exitoso'
50         ]);
51     }
52 }
```

```
ReciclajeController.php x
app > Http > Controllers > ReciclajeController.php
111 {
112     // Alias genérico para la ruta /reciclaje/guardar (por ahora usado para registrar entrega)
113     public function guardar(Request $request)
114     {
115         return $this->guardarEntrega($request);
116     }
117
118     // Mostrar el formulario para registrar una entrega de reciclaje
119     public function registrar()
120     {
121         return view('reciclaje.registrar');
122     }
123
124     // Guardar los datos de una entrega de reciclaje
125     public function guardarEntrega(Request $request)
126     {
127         $request->validate([
128             'tipo_residuo' => 'required|string|max:100',
129             'cantidad' => 'required|numeric|min:0.1',
130             'descripcion_general_materiales' => 'nullable|string|max:500',
131             'fecha_recoleccion_usuario' => 'nullable|date',
132             'ubicacion_recoleccion' => 'nullable|string|max:255',
133         ]);
134
135         $usuarioId = Auth::id();
136
137         if (!$usuarioId) {
138             return redirect()->route('login');
139         }
140
141         $cantidadKg = (float) $request->cantidad; // el formulario trabaja en KG
142         $tipo = $request->tipo_residuo;
143         $puntosSugeridos = $this->calcularPuntosPorKg($tipo, $cantidadKg);
144
145         $entrega = EntregaReciclaje::create([
146             'id_usuario' => (string) $usuarioId,
147             'tipo_residuo' => $tipo,
148             'cantidad_kg' => $cantidadKg,
149             'puntos_sugeridos' => $puntosSugeridos,
150             'puntos_otorgados' => 0,
151             'estado' => 'pendiente',
152             'fecha_registro' => now(),
153             'descripcion_general_materiales' => $request->descripcion_general_materiales,
154             'fecha_recoleccion_usuario' => $request->fecha_recoleccion_usuario,
155             'ip_reporte' => $request->ip(),
156             'ubicacion_recoleccion' => $request->ubicacion_recoleccion,
157         ]);
158     }
159 }
```



```

ReciclajeController.php x
app > Http > Controllers > ReciclajeController.php
13
127 {
171
172     return back()->with('success', 'Entrega registrada y enviada al administrador para validación.');
```

Tipo de Residuo	Puntos por Kg
plastico	10
plástico	10
pet	12
papel	8
carton	8
cartón	8
vidrio	6
metal	12
aluminio	12
organico	4
orgánico	4

```

173 }
174
175 /**
176  * Calcular puntos sugeridos según tipo de residuo y cantidad en KG.
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         'plastico' => 10,
185         'plástico' => 10,
186         'pet' => 12,
187         'papel' => 8,
188         'carton' => 8,
189         'cartón' => 8,
190         'vidrio' => 6,
191         'metal' => 12,
192         'aluminio' => 12,
193         'organico' => 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 reciclable. 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 x
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

```
UsuarioController.php x
app > ~/Documentos/repo-github/reciclapp-...
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.

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

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

Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
id	varchar(255)		Identificador único.

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

Campo	Tipo de Dato	Restricciones / Definición	Descripción sugerida
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_recoleccion

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_recoleccion": 1,
  "id_tipo_residuo": 1,
  "cantidad_kg_especifica": "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_recoleccion_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.",
  "condicion_tipo": "num_recoleccion",
  "condicion_valor_meta": "1.00",
  "puntos_recompensa_logro": 50,
  "icono_logro": "fa-leaf"
}
```

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_recoleccion

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": "ranses@gmail.com",
  "rol": "usuario",
  "puntos_acumulados": 0,
  "fecha_registro": "2025-09-18 15:04:04",
  "estado": "activo"
}
```