

INSTRUCCIONES PARA PUSH Y CREACIÓN DE PR

IMPORTANTE: Permisos de GitHub

El token de GitHub actual no tiene permisos suficientes para hacer push directamente. Necesitas dar permisos adicionales a la GitHub App de Abacus.AI.

Pasos para Dar Permisos

1. **Visita la configuración de la GitHub App:**

👉 https://github.com/apps/abacusai/installations/select_target (https://github.com/apps/abacusai/installations/select_target)

2. **Selecciona el repositorio EcoResiduosMaster**

3. **Asegúrate de que la app tenga permisos de:**

- ☒ Read and Write access to code
- ☒ Read and Write access to pull requests
- ☒ Read and Write access to contents

OPCIÓN 1: Push Manual desde tu Máquina Local

Si prefieres hacer el push desde tu computadora local:

```
# 1. Clonar el repositorio (si no lo tienes)
git clone https://github.com/johnmebao/EcoResiduosMaster.git
cd EcoResiduosMaster

# 2. Agregar el remote si es necesario
git remote add origin https://github.com/johnmebao/EcoResiduosMaster.git

# 3. Fetch todas las ramas
git fetch origin

# 4. Checkout a la rama feat/fase2-prioridad
git checkout feat/fase2-prioridad

# 5. Pull los cambios más recientes (si los hay)
git pull origin feat/fase2-prioridad

# 6. Push la rama
git push origin feat/fase2-prioridad
```

OPCIÓN 2: Crear PR Manualmente en GitHub

Una vez que la rama esté en GitHub (ya sea por push manual o después de dar permisos):

1. **Ve a GitHub:**

👉 <https://github.com/johnmebao/EcoResiduosMaster> (<https://github.com/johnmebao/EcoResiduosMaster>)

2. **Haz clic en “Pull requests”**

3. **Haz clic en “New pull request”**

4. **Configura el PR:**

- **Base:** feat/fase1-notificaciones-localidad-ruta
- **Compare:** feat/fase2-prioridad

5. **Título del PR:**

FASE 2: Mejoras de Alta Prioridad - Validaciones, Orgánicos, Peligrosos y Puntos

6. **Descripción del PR:**

Copia el contenido del archivo `PR_DESCRIPTION.md` (ver abajo)

DESCRIPCIÓN DEL PR (PR_DESCRIPTION.md)

FASE 2 - MEJORAS DE ALTA PRIORIDAD

📋 Resumen de Cambios

Esta PR implementa las mejoras de alta prioridad para el sistema EcoResiduos, incluyendo validaciones de negocio, módulos de recolección específicos por tipo de residuo, y mejoras en la lógica de asignación de puntos.

✅ Funcionalidades Implementadas

1. Validaciones de Negocio

- ✅ Request classes para validación de cada tipo de recolección
- ✅ Validación de frecuencias según tipo de residuo
- ✅ Validación de duplicados y límites temporales
- ✅ Métodos de validación en modelos

2. Módulo Recolecciones Orgánicos

- ✅ Programación automática semanal por localidad
- ✅ Asignación automática de fecha según ruta
- ✅ Comando artisan para generación masiva
- ✅ Scheduler configurado (Lunes 6:00 AM)
- ✅ No permite selección manual de fecha

3. Módulo Recolecciones Inorgánicos

- ✅ Dos modalidades: programada (máx 2/semana) y por demanda
- ✅ Validación de frecuencia semanal
- ✅ Permite selección de fecha
- ✅ Validación de duplicados

4. Módulo Recolecciones Peligrosos

- ✅ Sistema completo de solicitudes
- ✅ Flujo: solicitado → aprobado → programado → completado
- ✅ Panel de administración para aprobar/rechazar
- ✅ Límite de una solicitud por mes por usuario
- ✅ Tracking completo con fechas y usuario aprobador

5. Lógica de Puntos Mejorada

- ✅ Campo requisitos_separacion en collection_details
- ✅ PuntosService con cálculo automático
- ✅ Solo inorgánicos reciclables generan puntos
- ✅ Solo si requisitos_separacion = true
- ✅ Eventos automáticos en modelo
- ✅ Fórmula: 1 kg = 1 punto (preparado para Strategy Pattern)

📊 Estadísticas

- ****Archivos nuevos:**** 13
- ****Archivos modificados:**** 6
- ****Commits:**** 5
- ****Líneas agregadas:**** ~1,600+

🧪 Testing

Migraciones Verificadas:

```
```bash
php artisan migrate --pretend
```

✅ Todas las migraciones tienen sintaxis correcta

## Comandos Disponibles:

```
Dry run para probar
php artisan recolecciones:generar-organicos-semanales --dry-run

Generar recolecciones reales
php artisan recolecciones:generar-organicos-semanales --semanas=1
```



## Seguridad

- ☒ Todas las rutas protegidas con middleware `auth`
- ☒ Rutas de administración con `role:Administrador`
- ☒ Validaciones en Request classes
- ☒ Validaciones en modelos
- ☒ Logging de operaciones críticas



## Commits

1. feat: agregar validaciones de negocio para tipos de recolección
2. feat: implementar módulo de recolecciones orgánicos con programación automática
3. feat: implementar módulo de solicitudes de recolecciones peligrosos
4. feat: mejorar lógica de asignación de puntos con requisitos de separación
5. docs: agregar documentación completa de FASE 2



## Próximos Pasos

Después de merge:

1. Ejecutar migraciones en producción: `php artisan migrate`
2. Configurar cron job para scheduler
3. Probar flujos completos en staging
4. Capacitar usuarios sobre nuevas funcionalidades



## Documentación

Ver archivo completo: `FASE2_RESUMEN_CAMBIOS.md`



**IMPORTANTE:** Las migraciones están creadas pero NO ejecutadas. Ejecutar `php artisan migrate` después del merge.

**Nota:** El scheduler requiere configurar cron job en el servidor para ejecutar

`php artisan schedule:run` cada minuto.

...

## ESTADO ACTUAL

---

### Completado:

- [x] Todos los archivos creados y modificados
- [x] 5 commits realizados con mensajes descriptivos
- [x] Migraciones verificadas (sintaxis correcta)
- [x] Documentación completa creada
- [x] Código listo para revisión

### Pendiente:

- [ ] Push de la rama `feat/fase2-prioridad` a GitHub
- [ ] Creación del Pull Request
- [ ] Revisión de código
- [ ] Merge a `feat/fase1-notificaciones-localidad-ruta`

---

## SOPORTE

Si tienes problemas con el push o la creación del PR, contacta al equipo de desarrollo o revisa la documentación de GitHub sobre permisos de aplicaciones.

**Repositorio:** `https://github.com/johnmebao/EcoResiduosMaster`

**Rama actual:** `feat/fase2-prioridad`

**Rama base:** `feat/fase1-notificaciones-localidad-ruta`