SISTEMA DE ROLES - DOCUMENTACIÓN COMPLETA

& RESUMEN EJECUTIVO

Se ha implementado exitosamente un **sistema completo de roles y permisos** para el MoodTracker, incluyendo migraciones, seeders, triggers de validación y corrección de problemas de integridad de datos.

FECHA DE IMPLEMENTACIÓN

24 de Octubre de 2025

E COMPONENTES IMPLEMENTADOS

1 MIGRACIONES CREADAS

A) Tabla de Roles

- Archivo: database/migrations/2025_10_24_135951_create_roles_table.php
- Propósito: Crear tabla para almacenar roles del sistema
- Estructura:

```
id (Primary Key)name (Unique) - Nombre del roldescription (Nullable) - Descripción del roltimestamps (created_at, updated_at)
```

B) Tabla Pivot Role-User

- Archivo: database/migrations/2025_10_24_140145_create_role_user_table.php
- Propósito: Relación many-to-many entre usuarios y roles
- Estructura:

```
user_id (Foreign Key → users.id) CASCADE DELETE
role_id (Foreign Key → roles.id) CASCADE DELETE
company_id (Foreign Key → companies.id) NULL ON DELETE
PRIMARY KEY (user_id, role_id, company_id)
INDEX (user_id, company_id)
INDEX (role_id)
```

C) Corrección de Triggers

- Archivo: database/migrations/2025_10_24_142820_fix_trigger_validation_logic.php
- Propósito: Corregir lógica de validación de triggers para manejar correctamente tipos de preguntas
- Mejoras:
 - Validación mejorada para preguntas tipo scale, bool, select
 - o Manejo correcto de valores NULL
 - o Mensajes de error más descriptivos

2 SEEDERS IMPLEMENTADOS

A) RolesTableSeeder

- Archivo: database/seeders/RolesTableSeeder.php
- **Propósito:** Crear roles base del sistema
- Roles creados:

```
super_admin: 'Acceso total al sistema'
hr_admin: 'Panel RRHH (empresa/área)'
manager: 'Gestión de equipo/segmento'
employee: 'Usuario estándar (autogestión)'
```

B) AttachAdminRoleSeeder

- Archivo: database/seeders/AttachAdminRoleSeeder.php
- **Propósito:** Asignar rol de administrador a usuario específico
- Configuración:
 - Email del admin configurable via ADMIN_EMAIL en .env
 - Asigna rol hr_admin al usuario admin
 - Maneja multiempresa con company_id

C) UserSeeder (Modificado)

- Archivo: database/seeders/UserSeeder.php
- Modificación: Añadido usuario admin con email evablancomart@gmail.com
- **Propósito:** Crear usuario administrador para el sistema

D) DemoDataSeeder (Corregido)

- Archivo: database/seeders/DemoDataSeeder.php
- Problema resuelto: Lógica de inserción de respuestas según tipo de pregunta
- Mejoras implementadas:

```
    Para preguntas 'scale': answer_numeric con valor aleatorio
    Para preguntas 'bool': answer_bool con true/false aleatorio
    Para preguntas 'select': answer_option_key con opción aleatoria
```

E) DatabaseSeeder (Actualizado)

- Archivo: database/seeders/DatabaseSeeder.php
- Modificaciones:
 - Añadido RolesTableSeeder::class
 - Añadido AttachAdminRoleSeeder::class
 - o Orden optimizado de ejecución
 - Import statements corregidos

PROBLEMAS RESUELTOS

1 Error de Namespace

- Problema: RolesTableSeeder sin namespace causaba conflicto de clases
- Solución: Añadido namespace Database\Seeders;

2 Error de Company ID

- Problema: AttachAdminRoleSeeder intentaba insertar company_id = null
- **Solución:** Obtener ID real de la empresa democorp

3 Error de Validación de Triggers

- Problema: Trigger rechazaba datos válidos del DemoDataSeeder
- Solución: Corregir lógica de validación en trigger y adaptar seeder

4 Error de Tipos de Respuesta

- Problema: DemoDataSeeder insertaba siempre answer numeric sin importar tipo de pregunta
- Solución: Lógica condicional según tipo de pregunta

III RESULTADOS FINALES

☑ Datos Creados:

- 4 usuarios (incluyendo admin)
- 4 roles implementados
- **1 asignación** de rol (admin → hr_admin)
- 500 entradas de mood generadas
- Sistema de validación funcionando

✓ Funcionalidades:

- Sistema de roles completo
- Asignación automática de roles
- Validación de integridad de datos
- Datos de prueba generados
- Triggers de validación operativos

© CONFIGURACIÓN DEL SISTEMA

Variables de Entorno:

```
ADMIN_EMAIL=evablancomart@gmail.com
```

Comandos de Ejecución:

```
# Ejecutar migraciones y seeders
php artisan migrate:fresh --seed

# Verificar datos
php artisan tinker
>>> DB::table('role_user')->get();
```

Q VERIFICACIÓN DEL SISTEMA

Comandos de Verificación:

```
# Verificar usuarios
php artisan tinker --execute="echo 'Usuarios: ' . DB::table('users')->count();"

# Verificar roles
php artisan tinker --execute="echo 'Roles: ' . DB::table('roles')->count();"

# Verificar asignaciones
php artisan tinker --execute="echo 'Asignaciones: ' . DB::table('role_user')->count();"

# Verificar entradas mood
php artisan tinker --execute="echo 'Entradas mood: ' . DB::table('mood_entries')->count();"
```

& PRÓXIMOS PASOS

Para Desarrollo del Dashboard:

- 1. Autenticación: Implementar login con roles
- 2. Middleware: Crear middleware de verificación de roles
- 3. Vistas: Desarrollar vistas específicas por rol
- 4. API: Crear endpoints protegidos por roles

Para Producción:

1. **Seguridad:** Revisar permisos de base de datos

2. Backup: Implementar estrategia de respaldo

3. Monitoreo: Añadir logging de accesos

4. Documentación: Crear manual de usuario

ARCHIVOS MODIFICADOS

Migraciones:

- 2025_10_24_135951_create_roles_table.php (NUEVO)
- 2025_10_24_140145_create_role_user_table.php (NUEVO)
- 2025_10_24_142820_fix_trigger_validation_logic.php (NUEVO)

Seeders:

- RolesTableSeeder.php (NUEVO)
- AttachAdminRoleSeeder.php (NUEVO)
- UserSeeder.php (MODIFICADO)
- DemoDataSeeder.php (MODIFICADO)
- DatabaseSeeder.php (MODIFICADO)

Configuración:

• .env (MODIFICADO - añadido ADMIN_EMAIL)

☆ LOGROS TÉCNICOS

- 1. Sistema de roles completo implementado
- 2. Triggers de validación funcionando correctamente
- 3. Seeders optimizados para diferentes tipos de preguntas
- 4. Integridad de datos garantizada
- 5. Configuración flexible via variables de entorno
- 6. Documentación completa del proceso

CONTACTO TÉCNICO

Desarrollado por: Asistente Al **Fecha:** 24 de Octubre de 2025

Proyecto: MoodTracker - Sistema de Roles **Estado:** ✓ COMPLETADO Y FUNCIONAL