

Sistema de Autenticación con Google OAuth y Consentimiento

Resumen Ejecutivo

Este documento describe la implementación completa del sistema de autenticación con Google OAuth y el flujo de consentimiento informado para usuarios empleados en MoodTracker. El sistema permite dos métodos de autenticación:

1. **Login tradicional** con email y contraseña
2. **Login con Google OAuth** usando Laravel Socialite

Ambos métodos incluyen un sistema de consentimiento obligatorio para empleados y redirección basada en roles.

Objetivos Implementados

- Autenticación con Google OAuth mediante Laravel Socialite
 - Login tradicional con email/contraseña
 - Sistema de consentimiento obligatorio para empleados
 - Redirección automática según rol (employee → formulario, admin/rrhh → dashboard)
 - Middleware de protección para rutas que requieren consentimiento
 - Manejo de errores OAuth (InvalidStateException)
-

Archivos Creados/Modificados

Nuevos Archivos Creados

1. [app/Http/Controllers/Admin/GoogleController.php](#)
 - Controlador para manejar autenticación OAuth con Google
 - Métodos: `redirect()` y `callback()`
2. [app/Http/Controllers/Auth/ConsentController.php](#)
 - Controlador para manejar el flujo de consentimiento
 - Métodos: `show()` y `store()`
3. [app/Http/Middleware/EnsureUserConsented.php](#)
 - Middleware que verifica si el usuario ha dado consentimiento
 - Solo aplica a empleados, admins pueden pasar sin consentimiento
4. [resources/views/auth/consent.blade.php](#)
 - Vista del formulario de consentimiento informado

5. database/migrations/2025_11_03_102537_add_consent_at_and_role_to_users_table.php

- Migración que añade `consent_at` y `role` a la tabla `users`

Archivos Modificados

1. routes/web.php

- Añadidas rutas para Google OAuth (`/auth/redirect/google`, `/auth/callback/google`)
- Añadidas rutas para consentimiento (`/consent` GET y POST)
- Modificada ruta POST `/login` para incluir lógica de consentimiento y redirección por rol
- Añadido formulario de login tradicional en la vista de login

2. resources/views/auth/login.blade.php

- Añadido formulario de login tradicional (email/password)
- Mantenido botón de Google OAuth
- Separador visual entre ambas opciones

3. config/services.php

- Configuración de Google OAuth con credenciales desde `.env`

4. bootstrap/app.php

- Registrado middleware `EnsureUserConsented` con alias `consented`
- Configurado `SetLocale` para omitir rutas OAuth

5. database/seeders/UserSeeder.php

- Añadido soporte para campo `role` en usuarios
- Actualizado para crear usuarios con roles específicos

6. app/Http/Middleware/SetLocale.php

- Modificado para saltar middleware en rutas OAuth (`auth/redirect/*`, `auth/callback/*`)

⚙️ Configuración

1. Variables de Entorno (.env)

Añade las siguientes variables a tu archivo `.env`:

```
# Google OAuth Credentials
GOOGLE_CLIENT_ID=tu_client_id_de_google
GOOGLE_CLIENT_SECRET=tu_client_secret_de_google
GOOGLE_REDIRECT_URI=http://localhost:8000/auth/callback/google

# Email del usuario admin (opcional)
ADMIN_EMAIL=evablancomart@gmail.com
```

2. Crear Credenciales en Google Cloud Console

1. Ve a [Google Cloud Console](#)
2. Crea un nuevo proyecto o selecciona uno existente
3. Habilita la **Google+ API** o **Google Identity API**
4. Ve a **Credenciales** → **Crear credenciales** → **ID de cliente OAuth 2.0**
5. Configura:
 - **Tipo de aplicación:** Aplicación web
 - **URI de redirección autorizados:** <http://localhost:8000/auth/callback/google>
 - **Orígenes JavaScript autorizados:** <http://localhost:8000>
6. Copia el **Client ID** y **Client Secret** a tu [.env](#)

3. Instalar Laravel Socialite

```
composer require laravel/socialite
```

4. Ejecutar Migraciones

```
php artisan migrate
```

Esto creará las columnas `consent_at` y `role` en la tabla `users`.

5. Configurar Middleware

El middleware ya está configurado en `bootstrap/app.php`. Verifica que esté registrado:

```
$middleware->alias([  
    'consented' => \App\Http\Middleware\EnsureUserConsented::class,  
]);
```

🔗 Flujo de Autenticación

Flujo 1: Login con Google OAuth

1. Usuario hace clic en "Continuar con Google"
↓
2. Redirect a Google (`GoogleController@redirect`)
↓
3. Usuario autoriza en Google
↓
4. Google redirige a `/auth/callback/google`
↓
5. `GoogleController@callback`:

- Obtiene datos del usuario de Google
- Busca o crea usuario en BD
- Inicia sesión
- ↓
- 6. Verifica rol y consentimiento:
 - Si es employee SIN consentimiento → /consent
 - Si es employee CON consentimiento → /moods/create
 - Si es admin/rrhh → /dashboard

Flujo 2: Login Tradicional (Email/Password)

1. Usuario ingresa email y contraseña
- ↓
2. POST /login (routes/web.php)
- ↓
3. Auth::attempt() valida credenciales
- ↓
4. Si es válido:
 - Regenera sesión
 - Verifica rol y consentimiento:
 - * Si es employee SIN consentimiento → /consent
 - * Si es employee CON consentimiento → /moods/create
 - * Si es admin/rrhh → /dashboard

Flujo 3: Sistema de Consentimiento (Solo Empleados)

1. Usuario employee accede sin consentimiento
- ↓
2. Middleware EnsureUserConsented intercepta
- ↓
3. Redirige a /consent
- ↓
4. Usuario ve formulario de consentimiento
- ↓
5. POST /consent (ConsentController@store)
- ↓
6. Actualiza consent_at en BD
- ↓
7. Redirige según rol:
 - employee → /moods/create
 - admin/rrhh → /dashboard

Roles y Permisos

Roles Disponibles

- **employee**: Empleado regular (requiere consentimiento)
- **hr_admin**: Administrador de RRHH (no requiere consentimiento)
- **admin**: Administrador del sistema (no requiere consentimiento)
- **manager**: Gerente (no requiere consentimiento)

Redirección por Rol

Rol	Sin Consentimiento	Con Consentimiento
employee	→ /consent	→ /moods/create
hr_admin	→ /dashboard	→ /dashboard
admin	→ /dashboard	→ /dashboard
manager	→ /dashboard	→ /dashboard

📝 Código Clave

GoogleController

```
public function callback()
{
    try {
        $googleUser = Socialite::driver('google')->user();
    } catch (\Laravel\Socialite\Two\InvalidStateException $e) {
        // Fallback si se pierde la sesión
        $googleUser = Socialite::driver('google')->stateless()->user();
    }

    $email = $googleUser->getEmail();
    $company = Company::firstOrFail();

    $user = User::firstOrCreate(
        ['email' => $email, 'company_id' => $company->id],
        [
            'name' => $googleUser->getName() ?: 'Usuario Google',
            'email_verified_at' => now(),
            'role' => 'employee',
            'company_id' => $company->id,
        ]
    );
}

Auth::login($user);

// Lógica de redirección por rol
$role = $user->role ?? 'employee';
if ($role === 'employee' && is_null($user->consent_at)) {
    return redirect()->to('/consent');
}
if ($role === 'employee') {
    return redirect()->to('/moods/create');
```

```

    }
    return redirect()->to('/dashboard');
}

```

EnsureUserConsented Middleware

```

public function handle(Request $request, Closure $next)
{
    $user = $request->user();

    // Si no está logueado, ya consintió, o está en la ruta de consentimiento →
    // pasa
    if (! $user || $user->consent_at || $request->is('consent*')) {
        return $next($request);
    }

    // Solo fuerzo a empleados. Admin/manager pueden entrar sin consentimiento
    if (($user->role ?? 'employee') === 'employee') {
        return redirect('/consent');
    }

    return $next($request);
}

```

ConsentController

```

public function store(Request $request)
{
    $request->validate(['accept_terms' => 'required|accepted']);

    $user = $request->user();
    $user->update(['consent_at' => now()]);
    $user->refresh();

    // Redirigir según el rol
    if ($user->role === 'employee') {
        return redirect('/moods/create')
            ->with('success', 'Has aceptado los términos y condiciones
correctamente.');
    }

    return redirect('/dashboard')
        ->with('success', 'Has aceptado los términos y condiciones
correctamente.');
}

```

```

// Google OAuth
Route::get('/auth/redirect/google', [GoogleController::class, 'redirect'])
    ->name('google.redirect');
Route::get('/auth/callback/google', [GoogleController::class, 'callback'])
    ->name('google.callback');

// Consentimiento
Route::get('/consent', [ConsentController::class, 'show'])
    ->name('consent.show')
    ->middleware('auth');
Route::post('/consent', [ConsentController::class, 'store'])
    ->name('consent.store')
    ->middleware('auth');

// Login tradicional
Route::get('/login', fn() => view('auth.login'))->name('login');
Route::post('/login', function() {
    // Lógica de autenticación y redirección
})->name('login.post');

// Rutas protegidas con consentimiento
Route::get('/dashboard', [DashboardController::class, 'overview'])
    ->middleware('auth', 'consented')
    ->name('user.dashboard');

Route::prefix('moods')->middleware('auth', 'consented')->group(function() {
    Route::get('/create', [MoodEmotionController::class, 'create']);
    Route::post('/', [MoodEmotionController::class, 'store']);
});

```

█ Estructura de Base de Datos

Tabla **users** (modificada)

Se añadieron dos columnas:

```

Schema::table('users', function (Blueprint $table) {
    $table->timestamp('consent_at')->nullable()->after('email_verified_at');
    $table->string('role', 20)->default('employee')->after('consent_at');
});

```

- **consent_at**: Timestamp de cuando el usuario dio consentimiento (NULL si no ha consentido)
- **role**: Rol del usuario (employee, hr_admin, admin, manager)

█ Testing

Usuarios de Prueba

Se crearon usuarios de prueba mediante **UserSeeder**:

```
$people = [
    ['name' => 'Eva Blanco', 'email' => 'eva@democorp.test', 'role' =>
    'employee'],
    ['name' => 'Luis Pérez', 'email' => 'luis@democorp.test', 'role' =>
    'employee'],
    ['name' => 'Marta Ruiz', 'email' => 'marta@democorp.test', 'role' =>
    'employee'],
    ['name' => 'Eva Blanco Admin', 'email' => 'evablancomart@gmail.com', 'role' =>
    'hr_admin'],
];
];
```

Contraseña por defecto: **secret123**

Casos de Prueba

1. Login empleado sin consentimiento:

- Email: **eva@democorp.test** / Password: **secret123**
- Esperado: Redirección a **/consent** → aceptar → **/moods/create**

2. Login admin:

- Email: **evablancomart@gmail.com** / Password: **secret123**
- Esperado: Redirección directa a **/dashboard** (sin consentimiento)

3. Login con Google:

- Clic en "Continuar con Google"
- Esperado: Según el email de Google, redirección según rol

⚠ Solución de Problemas

Error: `InvalidStateException`

Causa: La sesión se pierde entre el redirect a Google y el callback.

Solución: El código ya incluye un fallback usando `stateless()`:

```
try {
    $googleUser = Socialite::driver('google')->user();
} catch (\Laravel\Socialite\Two\InvalidStateException $e) {
    $googleUser = Socialite::driver('google')->stateless()->user();
}
```

Si el problema persiste:

1. Verifica que `SESSION_DRIVER` esté configurado correctamente en `.env`
2. Limpia la caché: `php artisan config:clear && php artisan cache:clear`
3. Asegúrate de que las cookies de sesión estén habilitadas en el navegador

Error: Admin va al formulario en lugar del dashboard

Causa: El rol del usuario no está configurado correctamente o la lógica de redirección no está funcionando.

Solución:

1. Verifica el rol en la BD:

```
SELECT email, role, consent_at FROM users WHERE email = 'evablancomart@gmail.com';
```

2. Asegúrate de que el rol sea `hr_admin` o `admin`, no `employee`
3. Verifica que la lógica en `routes/web.php` y `GoogleController.php` esté correcta

Error: DriverMissingConfigurationException

Causa: Las credenciales de Google no están configuradas en `config/services.php`.

Solución:

1. Verifica que `config/services.php` tenga la configuración:

```
'google' => [  
    'client_id' => env('GOOGLE_CLIENT_ID'),  
    'client_secret' => env('GOOGLE_CLIENT_SECRET'),  
    'redirect' => env('GOOGLE_REDIRECT_URI'),  
,
```

2. Verifica que `.env` tenga las variables configuradas
3. Ejecuta `php artisan config:clear`

Error: Usuario no se crea en Google OAuth

Causa: Falta la compañía en la base de datos o hay un error en `firstOrCreate`.

Solución:

1. Asegúrate de tener al menos una compañía:

```
php artisan db:seed --class=CompanySeeder
```

2. Verifica que `Company::firstOrFail()` no lance excepción

Referencias

- [Laravel Socialite Documentation](#)
 - [Google OAuth 2.0 Documentation](#)
 - [Laravel Authentication Documentation](#)
-

Historial de Cambios

2025-11-03

- Implementación inicial de Google OAuth
 - Sistema de consentimiento obligatorio para empleados
 - Middleware de verificación de consentimiento
 - Redirección basada en roles
 - Manejo de errores OAuth (InvalidStateException)
 - Formulario de login tradicional añadido
 - Migración para `consent_at` y `role`
 - Actualización de seeders para incluir roles
-

Documento creado el: 2025-11-03

Última actualización: 2025-11-03

Versión: 1.0.0