Seguridad y autenticación

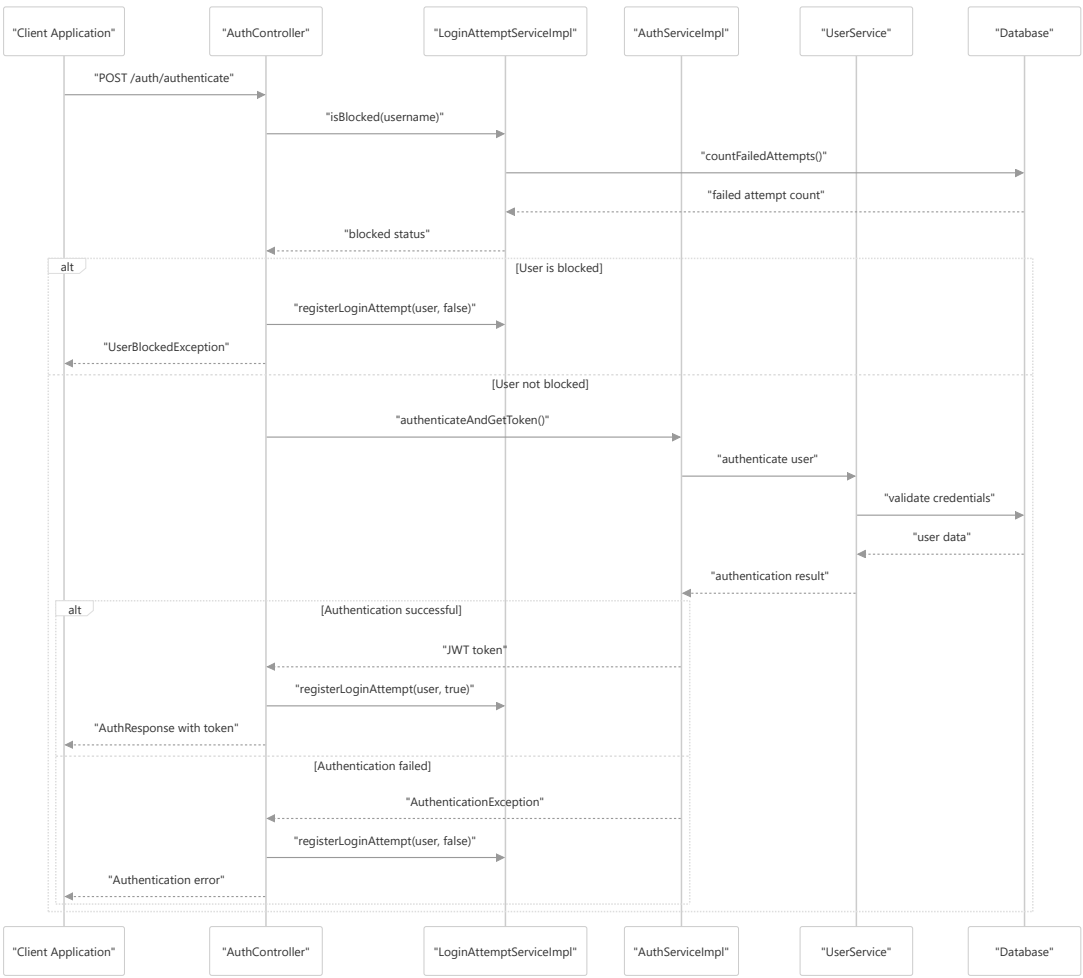
Este documento describe el sistema integral de seguridad y autenticación implementado en la aplicación ProyectoGestorGastos. El sistema proporciona múltiples mecanismos de autenticación, protección contra ataques de fuerza bruta y control de acceso basado en roles para garantizar el acceso seguro a los datos financieros y las funciones administrativas.

Para obtener información detallada sobre la configuración de Spring Security y las políticas CORS, consulte Configuración de seguridad web .

Descripción general de la autenticación

El sistema implementa un enfoque de autenticación multicapa compatible con la autenticación tradicional mediante nombre de usuario y contraseña, así como con proveedores OAuth2 (Google y GitHub). Todas las solicitudes autenticadas utilizan tokens JWT para la gestión de sesiones sin estado.

Flujo de autenticación



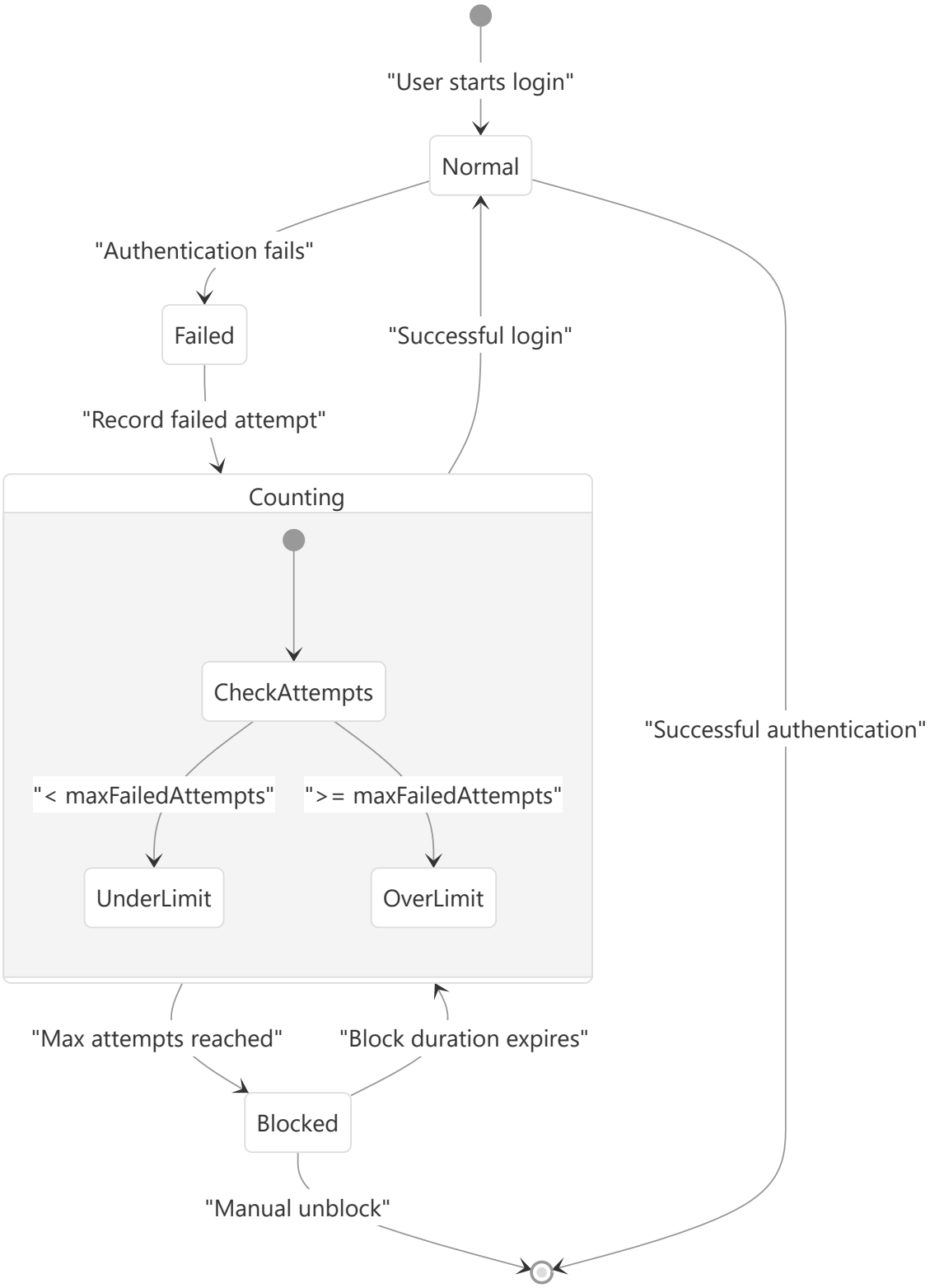
Seguimiento de intentos de inicio de sesión y protección contra ataques de fuerza bruta

El sistema implementa una sofisticada protección contra fuerza bruta a través de la `LoginAttemptServiceImpl` clase, que rastrea todos los intentos de inicio de sesión y bloquea temporalmente a los usuarios después de múltiples intentos fallidos.

Configuración de protección contra fuerza bruta

Propiedad de configuración	Propósito predeterminado	Implementación
`attemp.login.max.failed`	Máximo de intentos fallidos antes del bloqueo	Utilizado en `isBlocked()` el método
`attemp.login.block.duration`	Duración del bloque en minutos	Aplicado en cálculos de tiempo
`attemp.login.delete.log`	Si se deben limpiar las entradas de registro antiguas	Controla la retención de registros

Ciclo de vida del intento de inicio de sesión



Proporciona `LoginAttemptServiceImpl` estos métodos clave:

- `registerLoginAttempt(String username, boolean success)` - Registra cada intento de inicio de sesión con marca de tiempo
- `isBlocked(String username)` - Comprueba si el usuario ha superado el umbral de intentos fallidos
- `timeUntilUnlock(String username)` - Calcula el tiempo de bloque restante

Puntos finales de registro y autenticación de usuarios

Proporciona `AuthController` dos puntos finales principales para la autenticación y el registro de usuarios:

Punto final de autenticación

- **Ruta** : `POST /auth/authenticate`
- **Propósito** : Valida las credenciales del usuario y devuelve el token JWT
- **Solicitud** : `LoginRequest` DTO que contiene nombre de usuario/correo electrónico y contraseña
- **Respuesta** : `AuthResponse` contiene el token JWT

Punto final de registro

- **Ruta** : `POST /auth/signup`
- **Propósito** : Crea una nueva cuenta de usuario y devuelve el token JWT
- **Solicitud** : `SignUpRequest` DTO con datos de registro de usuario
- **Respuesta** : `AuthResponse` contiene el token JWT
- **Validación** : evita nombres de usuario y direcciones de correo electrónico duplicados

Administración de intentos de inicio de sesión

Los administradores pueden supervisar la seguridad de la autenticación a través de `LoginAttempController` , que proporciona acceso a los registros de intentos de inicio de sesión.

Monitoreo del inicio de sesión del administrador

- **Ruta** : `GET /admin/loginAttemps/`
- **Acceso** : Se requiere rol de administrador con autenticación de token de portador
- **Filtrado** : parámetro opcional de nombre de usuario/correo electrónico para filtrar los resultados
- **Respuesta** : Lista de `LoginAttempt` entidades con marcas de tiempo y estado de éxito

El punto final admite tanto la supervisión exhaustiva (todos los intentos) como la investigación específica (intentos de usuarios específicos).

Capa de acceso a datos para seguridad

El sistema de seguridad utiliza interfaces de repositorio dedicadas para la persistencia de datos:

Métodos del repositorio de intentos de inicio de sesión

- `countFailedAttempts(String username, Instant since)` - Cuenta los intentos fallidos dentro de la ventana de tiempo