Logotipo

El contenido generado por IA puede ser incorrecto.

**TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE**

**CICLO:**

SEXTO CICLO

**MATERIA:**

APLICACIONES SEGURAS

**ESTUDIANTE:**

OSCAR ZURITA

**PROFESOR:**

ING. DIEGO CALE

**PERIODO:**

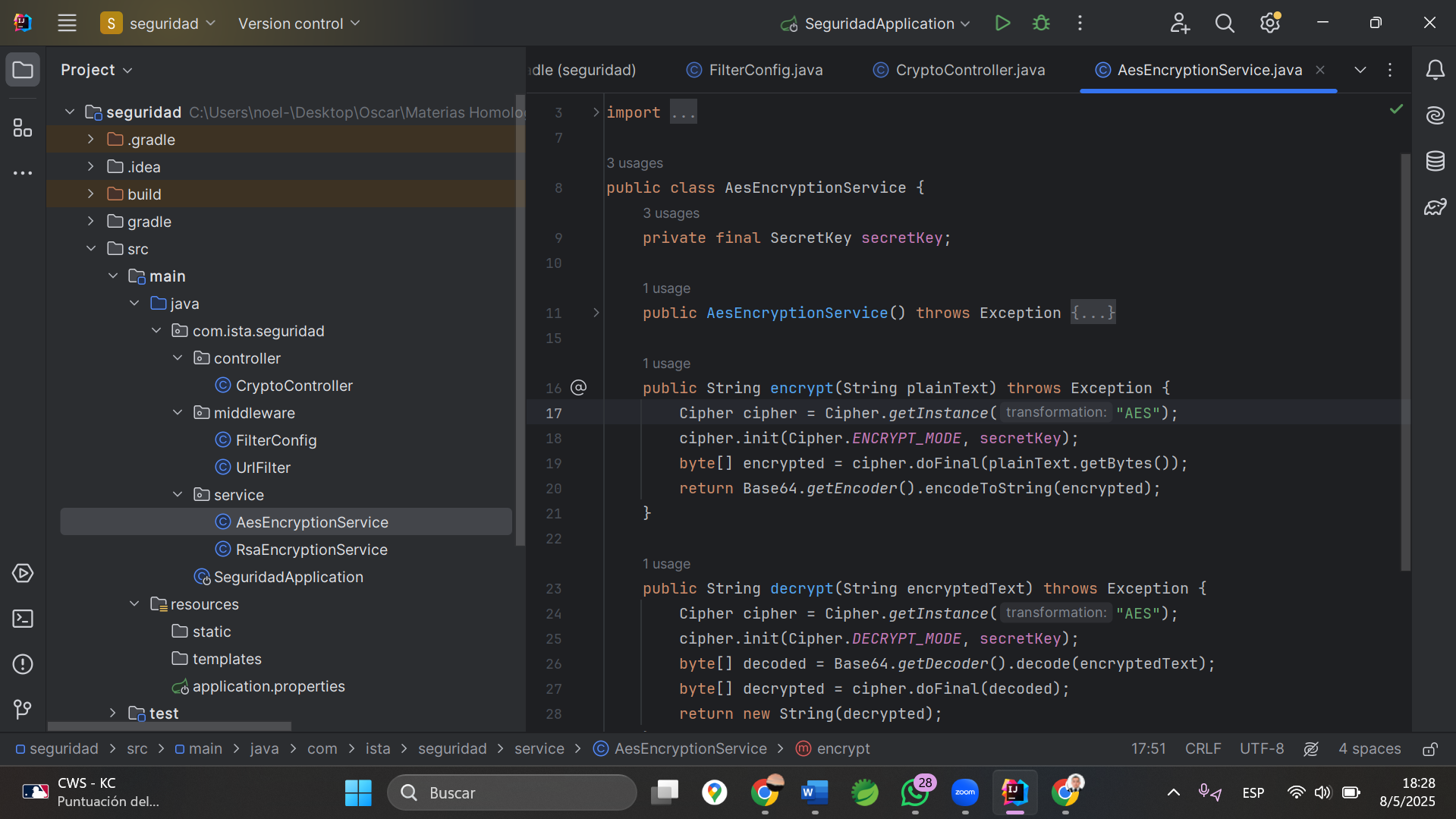
MARZO 2025– AGOSTO 2025

**GUIA PRÁCTICA 1**

**Paso 1: Cifrado Simétrico con AES**

En este paso se implementa el cifrado simétrico utilizando el algoritmo AES (Advanced Encryption Standard), el cual emplea una única clave secreta para cifrar y descifrar datos. Se crea una clase en Java encargada de generar esta clave automáticamente y de ofrecer métodos para convertir textos planos en datos cifrados y viceversa. Este tipo de cifrado es ideal cuando ambas partes comparten una clave de forma segura, y se requiere rapidez para proteger datos sensibles en reposo o en sistemas internos.

* 1. **Crear la clase de utilidad AES:**

****

**Paso 2: Cifrado Asimétrico con RSA**

Aquí se desarrolla una clase para aplicar cifrado asimétrico utilizando RSA, un algoritmo ampliamente usado en la transmisión segura de datos. A diferencia del cifrado simétrico, RSA utiliza un par de claves: una pública para cifrar la información y una privada para descifrarla. Esta separación de claves permite el intercambio seguro de datos sin necesidad de que las partes compartan una clave secreta previamente. En la práctica, se genera dinámicamente el par de claves y se crean métodos que cifran y descifran cadenas de texto, lo que permite entender cómo funcionan conceptos como firma digital o protección de datos en redes públicas.

**2.1 Crear clase RSA:**

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

**Paso 3: Controlador REST para pruebas**

Una vez implementadas las funcionalidades de cifrado, se expone una API REST usando Spring Boot, que permite probar los servicios de cifrado y descifrado. Este paso tiene como objetivo integrar las clases de lógica criptográfica con puntos finales accesibles mediante aplicaciones HTTP. De esta manera, se facilita la interacción con los métodos de cifrado desde el navegador o herramientas como Postman, permitiendo probar directamente el comportamiento del cifrado AES y RSA con diferentes entradas de texto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

**Paso 4: Implementar Middleware para controlar las URL**

En este paso se construye un middleware (filtro HTTP) que intercepta todas las solicitudes que llegan a la aplicación, con el objetivo de controlar el acceso a ciertas rutas. Por ejemplo, se puede permitir el acceso solo a usuarios provenientes de determinadas direcciones IP o cumplan ciertas condiciones. Este tipo de capa de seguridad permite agregar reglas personalizadas antes de que la petición llegue al controlador, lo que agrega una primera línea de defensa para proteger los recursos críticos de la aplicación.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

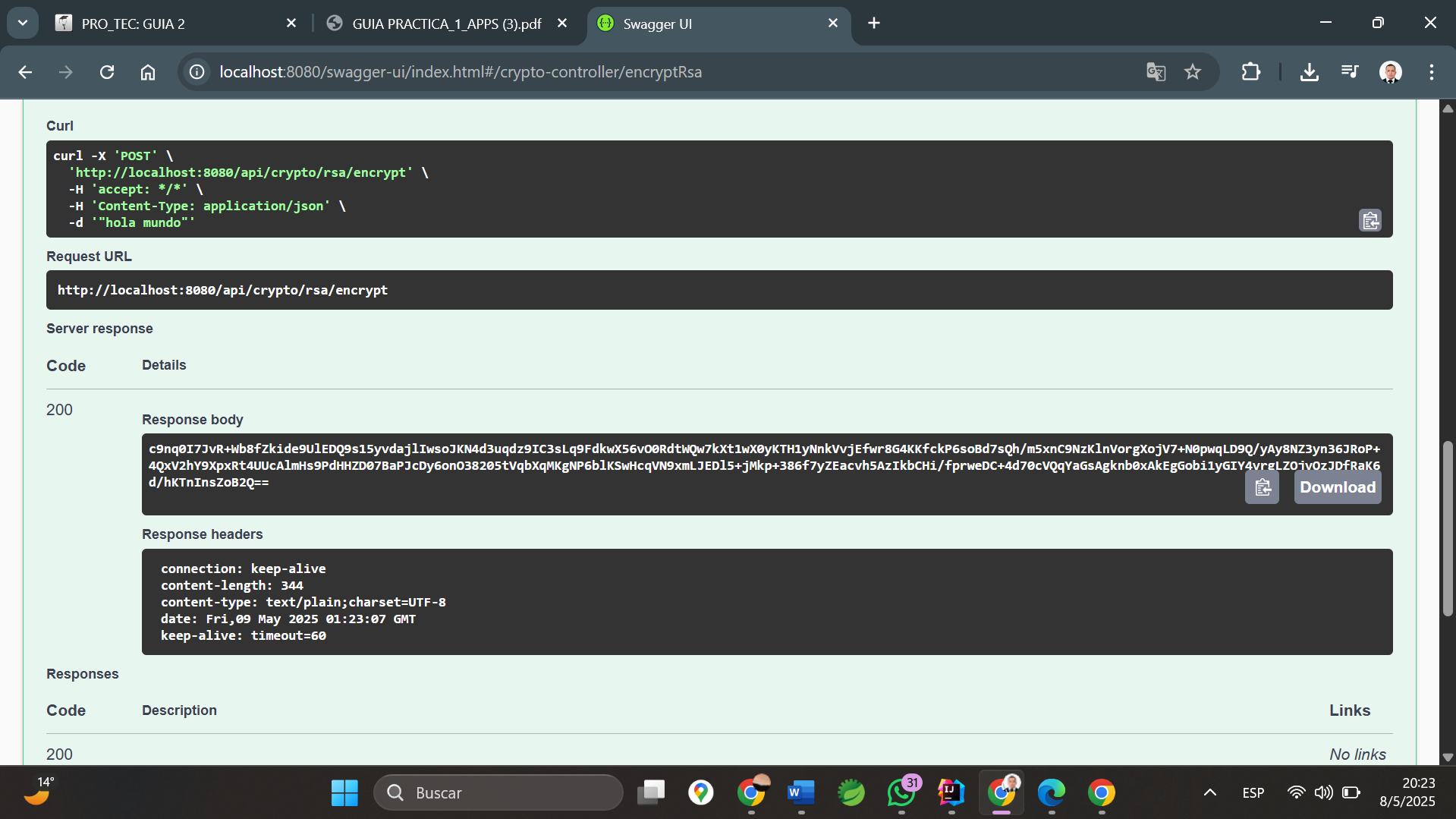
**4.2 Registrar el filtro:**

**Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.**

**Paso 5: Pruebas y Validación del Sistema**

Finalmente, se lleva a cabo una serie de pruebas funcionales para verificar que todas las partes de la aplicación funcionan correctamente. Se envían solicitudes HTTP a los endpoints definidos y se comprueba que los datos se cifran y descifran adecuadamente, tanto con AES como con RSA. También se valida que el middleware bloquea correctamente el acceso no autorizado a las URL protegidas. Este paso permite consolidar el aprendizaje, identificar posibles errores y confirmar la efectividad de las técnicas de cifrado y control de acceso implementadas.



Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.