Examen 2ª Evaluación

4 de marzo de 2024

Instrucciones:

- Entra en tu cuenta de GitHub y haz un *fork* del repositorio https://github.com/DamFleming/PSP20240304
- Cuando se haya completado el *fork*, clona desde Eclipse tu nuevo repositorio e importa el proyecto.
- Renombra el proyecto con tu nombre usando el formato siguiente: *apellidos, nombre*.
- Deshabilita cualquier conexión a Internet en el ordenador donde realizas el examen.
- Cuando finalices el examen:
  - Exporta el proyecto a un archivo comprimido.
  - Pide permiso para habilitar de nuevo la conexión de Internet.
  - Entrega el archivo comprimido con el proyecto del examen en la tarea de Teams.
  - Ejecuta un *commit & push* con el repositorio.

El proyecto incluye un conjunto de pruebas JUnit 5 definidas en la clase *dam.psp.ServidorUnitTest*, dentro de la carpeta *test* del proyecto. La calificación del examen se obtendrá del resultado de la ejecución de los 24 casos de prueba que contiene. Durante la ejecución del test, se muestra el nombre de cada caso de prueba y su puntuación en caso de éxito.

Ejercicio 1.

Usando el lenguaje Java y técnicas de programación orientada a objetos, completa la clase *dam.psp.Servidor* de la carpeta *src* para crear un servidor de cifrado que acepte peticiones con el formato que se describe a continuación:

- Petición para obtener el hash de una secuencia de bytes.

  Formato:

  La cadena `"hash"` seguida de otra cadena que contenga el nombre de un algoritmo de resumen seguido de una secuencia de bytes de longitud arbitraria.

  Respuesta si la petición es correcta:

  `"OK:`*`resumen_codificado_en_base64`*`"`

  Respuestas de error:

  `"ERROR:Se esperaba un algoritmo"` si no se recibe una cadena con el nombre del algoritmo.

  `"ERROR:Se esperaban datos"` si no se recibe la secuencia de bytes.

- Petición para almacenar en un objeto *KeyStore* un certificado.

  Formato:

  El *String* `"cert",` seguido de un *String* que contenga un alias para el certificado, seguido de un *String* que contenga una codificación en Base64 de una secuencia de bytes que el

cliente habrá obtenido como resultado de invocar al método *getEncoded* de una instancia de la clase *Certificate*.

El servidor deberá crear el almacén de claves si no existe. No se puede usar el que se incluye en el proyecto.

Respuesta si la petición es correcta:

"OK:*hash_sha-256_del_tercer_string_de_la_petición_codificado_en_base64*"

Respuestas de error:

"ERROR:Se esperaba un alias" si no se recibe una cadena con el alias.

"ERROR:Se esperaba un certificado" si no se recibe una cadena con el certificado.

"ERROR:Se esperaba base64" si no el certificado recibido no está codificado en Base64.

- Petición para cifrar una secuencia de bytes con el algoritmo *"RSA/ECB/PKCS1Padding"*.

  Formato:

  El *String* "cifrar", seguido de un *String* que contenga una alias de un certificado almacenado en el objeto *KeyStore* del servidor, seguido de la secuencia de bytes de longitud arbitraria.

  Respuesta si la petición es correcta:

  El cifrado se realizará leyendo la secuencia recibida en bloques de 256 bytes y cifrándolos por separado. Cada vez que se cifre un bloque se enviará al cliente una cadena con el formato siguiente:

  "OK:*bloque_cifrado_codificado_en_base64*"

  Cuando se hayan cifrado todos los bloques se enviará una última cadena con el formato:

  "FIN:CIFRADO"

  Respuestas de error:

  "ERROR:Se esperaba un alias" si no se recibe una cadena con el alias.

  "ERROR:'*alias*' no es un certificado" si el almacén de claves no contiene el alias que ha enviado el cliente.

  "ERROR:'*alias*' no contiene una clave RSA" si la clave del certificado identificado con el alias que ha enviado el cliente no es una clave RSA.

  "ERROR:Se esperaban datos" si no se recibe la secuencia de bytes.

El servidor también enviará las respuestas de error que se describen a continuación

"ERROR:Read timed out" si una petición provoca un *time out* , que en el servidor estará definido para un valor de cinco segundos.

"ERROR:Se esperaba una petición" si el cliente envía una petición vacía.

"ERROR:'*petición*' no se reconoce como una petición válida" si el primer *String* recibido no se reconoce como una petición válida.

Ejercicio 2.

Completa el caso de prueba llamado test17 para que realice una prueba de una petición válida para cifrar un texto de una longitud superior a 256 bytes que no sea múltiplo de 256.