Universidad Católica Andrés Bello
Facultad de Ingeniería - Escuela de Ingeniería Informática
Asignatura: Sistemas Distribuidos
Semestre 202315
Prof. Carlos Acosta
Fecha de asignación: 24 de Noviembre de 2022, 9:00pm.

Proyecto Final de Sistemas Distribuidos
(Valor: 4,0 puntos de la Nota Definitiva de la Asignatura)
(Fecha de entrega: Lunes 12 de Diciembre de 2022, a las 5:00pm)

Indicaciones previas:

A. El proyecto final puede ser realizado en su casa o en los laboratorios de la UCAB. Usted debe tener conocimientos básicos de comandos y shell del Sistema Operativo Linux, distro Devian o Ubuntu preferiblemente, asi como de programación en lenguaje Java, Phyton, C/C++ o C#, JavaScript, PHP, etc.

B. Los estudiantes pueden conformar equipos de proyecto de hasta 4 estudiantes a lo sumo.

C. La entrega comprende un informe digital claro, concreto y conciso. Debe cuidar la redacción y la ortografia. Este informe NO debe ser mayor a 10 páginas.

D. Debe subir los códigos fuente y ejecutables del cliente, del proxy y del servidor, el archivo de datos de entrada, el archivo de resultados de salida, ambos en formato .txt y el informe en formato .pdf. Comprima todo lo anterior en un archivo .zip y súbalo a M7.

E. Además, para poder hacer portable y fácil la prueba de ejecución del sistema distribuido completo que desarrolló utilice alguna herramienta de virtualización como Docker, por ejemplo.

F. Puede elegir el lenguaje de programación de su preferencia, pero debe proveer todas las indicaciones asociadas a la ejecución de la aplicación y demás dependencias. Se recomienda use virtulización para facilitar la prueba de ejecución del software distribuido por parte del profesor. Si el proyecto no corre o no cumple con las funcionalidades requeridas, será reprobado.

G. Recuerde que está prohibido copiarse entre Ustedes por ningún medio, o de cualquier otra fuente no autorizada por el profesor. Esto aplica a cualquier evaluación de la asignatura, en cuyo caso será evaluada con cero (0) puntos y habrá consecuencias académicas.

1. ENUNCIADO DEL PROYECTO:

La Empresa SUSUERTE ha encargado a los estudiantes de la asignatura Sistemas Distribuidos de la Escuela de Ingeniería Informática de la Universidad Católica Andrés Bello el diseño e implementación de un Sistema Distribuido de Firma Electrónica Básico (Ver Figura No. 1a).

El objetivo principal de este proyecto final de la materia es desarrollar una aplicación distribuida que simule la gestión de firmas electrónicas de mensajes de datos usando el modelo cliente/servidor. Todo esto mediante el control de la generación de claves de cifrado/descifrado, su uso para firmar documentos digitales, la verificación de integridad del mensaje y la autenticación de las identidades de los firmantes como usuarios o propietarios de dichas claves.

En general, la aplicación realiza tres (3) operaciones básicas: a) **FIRMAR** la cual implica hacer una solicitud al **Servidor A** de Claves para obtener una clave para firmar electrónicamente, b) **AUTENTICAR** la cual implica consultar la existencia de la identidad del usuario de una clave al **Servidor B** de autenticación para obtener de él una respuesta positiva (VÁLIDA) o negativa (INVÁLIDA) sobre la identidad del supuesto usuario o propietario de la clave consultada, y c) **INTEGRIDAD** donde el cliente realiza localmente el cálculo de la integridad del contenido de un Mensaje de Datos para lo cual descifra la firma del Mensaje para obtener el Hash de éste y compararlo con el Hash obtenido directamente del Mensaje de Datos y determinar si el Mensaje de Datos ha sido alterado (NO INTEGRO) o no (INTEGRO).

Este software debe ser desarrollado como un sistema distribuido, de forma no interactiva con el usuario **(**Ver Figura No. 1b), y poseer una arquitectura cliente/servidor donde los clientes pueden requerir dos tipos de servicios **(**Ver Figura No. 1a):

a) Tipo 1: Solicitar una clave criptográfica a un **servidor A** de claves, y

b) Tipo 2: Solicitar la autenticación de la identidad del usuario de una clave a un **servidor B**.

En este escenario, los clientes pueden realizar localmente dos tareas, las cuales NO realizan los servidores:

a) Usar la clave criptográfica recibida del **servidor A** para proceder a firmar electrónicamente uno o varios Mensajes de Datos. Para generar esta firma electrónica, el cliente debe primero calcular el bloque Hash del Mensaje de Datos que se desea firmar usando los algoritmos MD5 o Sha256, y luego proceder a cifrar este bloque Hash con la clave criptográfica del firmante o usuario (Ver Figura 2a).

b) También, el cliente puede usar la clave para **descifrar** la firma electrónica asociada a un Mensaje de Datos y extraer su bloque Hash para determinar la integridad o no alteración del Mensaje (Ver Figura 2b). Esto se realiza comparando el bloque Hash extraído de la firma electrónica con el bloque Hash calculado del Mensaje de Datos firmado.

Por otro lado, se tienen dos **servidores A y B** que hacen lo siguiente:

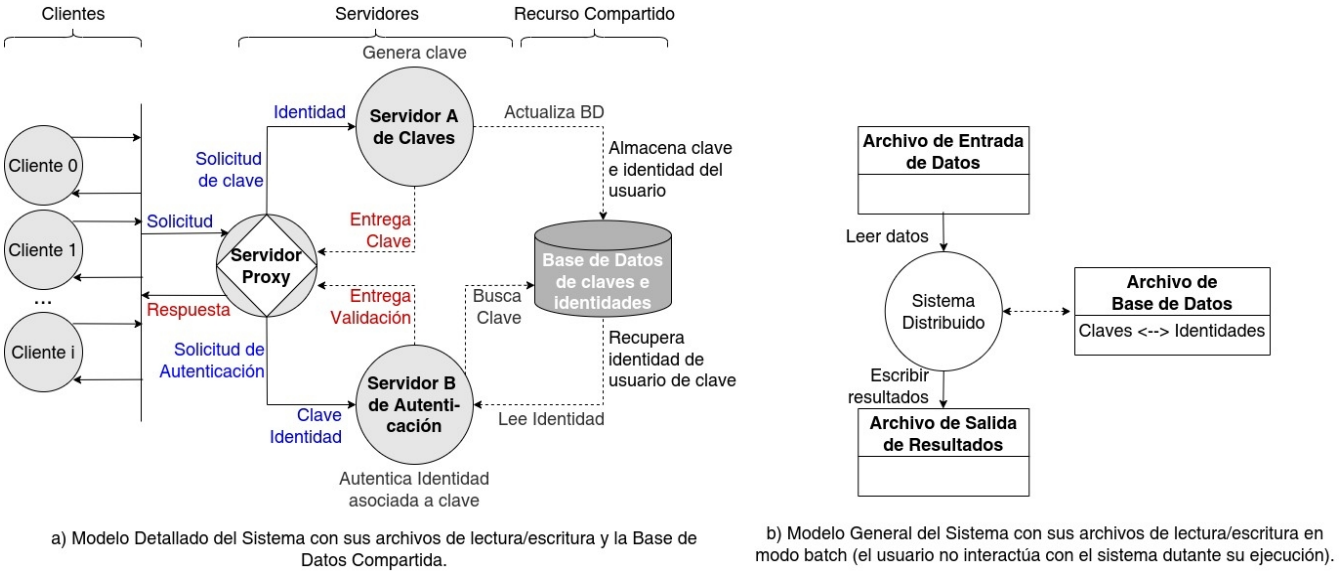
a) Un **servidor A** de claves criptográficas que genera aleatoriamente una clave numérica de 8 dígitos solicitada por un cliente y la asocia a la identidad del usuario solicitante almacenándola en una tabla o una base de datos compartida entre el **servidor A** y el **servidor B**.

b) Otro **servidor B** se encarga de autenticar la identidad del usuario de una clave recibida de un cliente consultando su existencia o no en la tabla o base datos compartida.

Dado que el **servidor A** y el **servidor B** pueden acceder de forma concurrente a esta tabla o base de datos es necesario garantizar la exclusión mútua entre ellos para mantener la consistencia y coherencia de la base de datos.

Las claves son numéricas de 8 dígitos y la firma electrónica es un patrón alfanumérico producto de cifrar el Hash de un mensaje de datos.

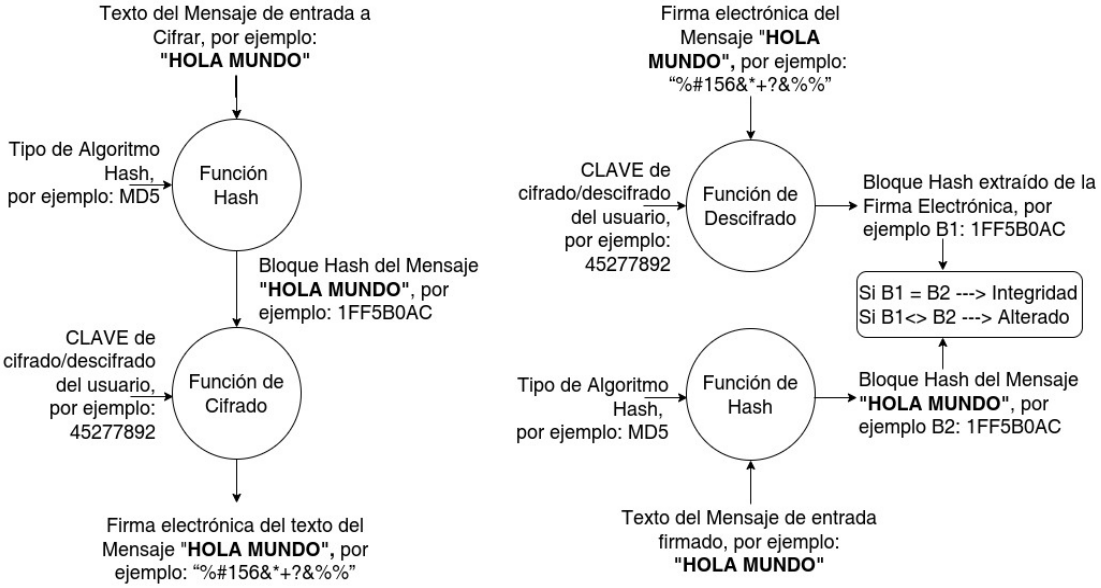
Nota: Para las funcionalidades asociadas al algoritmo criptográfico de cifrado/descifrado, el de generacion aleatorias de claves y el algoritmo de generación de bloque Hash se pueden usar librerias o programas externos ya hechos e integrarlos como *backend* de su aplicación. Todos los archivos de entrada, de salida y la base de datos son archivos en formato .txt.



a) Modelo Detallado del Sistema con sus archivos de lectura/escritura y la Base de Datos Compartida.

b) Modelo General del Sistema con sus archivos de lectura/escritura en modo batch (el usuario no interactúa con el sistema dutante su ejecución).

Figura 1. Diagrama Lógico del Modelo Cliente/Servidor del Sistema Básico de Firma Electrónica.



a) Algoritmo del proceso de cifrado del Mensaje para producir la Firma Electrónica.

b) Algoritmo del proceso de descifrado de la Firma Electrónica para Verificar Integridad del Mensaje.

Figura 2. Tareas del Cliente: a) Proceso de cifrado para firma electrónica, b) Proceso de descifrado para verificar integridad.

2. FORMATO DEL ARCHIVO DE ENTRADA (Ver Figura 3a)

2.1 Primera linea del archivo: FIRMAR=Consultar al Servidor de Clave
AUTENTICAR=Consultar al Servidor de Autenticación
INTEGRIDAD=Cliente descifra la firma electrónica

2.2 Segunda linea del archivo: En caso de FIRMAR=<Identidad de usuario de la clave>
En caso de AUTENTICAR=<CLAVE numérica>
En caso de INTEGRIDAD=<CLAVE numérica>

2.3 Tercera linea del archivo: En caso de FIRMAR=<Texto del Mensaje>
En caso de AUTENTICAR=<Identidad de usuario de clave>
En caso de INTEGRIDAD=<Texto del Mensaje>

2.4 Cuarta linea del archivo: En caso de FIRMAR=<>
En caso de AUTENTICAR=<>
En caso de INTEGRIDAD=<Texto de la Firma del Mensaje>

2.5 Última linea del archvio: 0

3. FORMATO DEL ARCHIVO DE SALIDA (Ver Figura 3b)

3.1 Primera linea del archivo: En caso de FIRMAR=<CLAVE numérica>
En caso de AUTENTICAR=<Clave VALIDA o INVALIDA>
En caso de INTEGRIDAD=<Mensaje INTEGRO o NO INTEGRO>

3.2 Segunda linea del archivo: En caso de FIRMAR=<Texto de la Firma del Mensaje>
En caso de AUTENTICAR=<>
En caso de INTEGRIDAD=<>

3.3 Última linea del archvio: 0

4. FORMATO DEL ARCHIVO DE CLAVES E IDENTIDADES (BASE DE DATOS, Ver Figura 3c)

4.1 Primera linea del archivo: En caso de FIRMAR=<CLAVE numérica>

4.2 Segunda linea del archivo: En caso de FIRMAR=<Identidad de usuario de la clave>,

4.3 Última linea del archvio: 0

5. EJEMPLOS DE FORMATOS DE ARCHIVOS:

FIRMAR <Identidad de usuario de clave> <Texto del Mensaje>	<Clave numérica de 8 dígitos> <Texto de la firma del mensaje>	<Clave numérica de 8 dígitos> <Identidad de usuario de clave>,
a) Archivo de Entrada	**b) Archivo de Salida**	**c) Archivo de Claves e Identidades**

FIRMAR <Identidad de usuario de clave> <Texto del Mensaje> 0	<Clave numérica de 8 dígitos> <Texto de Firma del Mensaje>	<Clave numérica de 8 dígitos> <Identidad de usuario de clave>,
a) Archivo de Entrada	**b) Archivo de Salida**	**c) Archivo de Claves e Identidades**

INTEGRIDAD CLAVE <Texto del Mensaje> <Texto de la Firma del Mensaje> 0	<INTEGRO / NO INTEGRO>	<Clave numérica de 8 dígitos> <Identidad de usuario de clave>,
a) Archivo de Entrada	**b) Archivo de Salida**	**c) Archivo de Claves e Identidades**

AUTENTICAR CLAVE <Identidad de usuario de clave> 0	<Clave VÁLIDA / INVÁLIDA>	<Clave numérica de 8 dígitos> <Identidad de usuario de clave>,
a) Archivo de Entrada	**b) Archivo de Salida**	**c) Archivo de Claves e Identidades**

Instancia de ejemplo:

FIRMAR Pedro Pérez “Hola Mundo!” 0	45277892 “%#156&*+?&%%” 0	45277892 Pedro Pérez, 0
a) Archivo de Entrada	**b) Archivo de Salida**	**c) Archivo de Claves e Identidades**

Figura 3. Ejemplos de formato de los archivos utilizados: a) Archivo donde el programa cliente lee sus parámetros de entrada, b) Archivo donde el cliente escribe sus resultados, c) Archivo donde el Servidor A escribe las claves generadas junto a la identidad de sus propietarios.
Nota: Las comillas no son parte del texto del mensaje ni de del texto de la firma electrónica.

6. REQUERIMIENTOS FUNCIONALES (Ver Figura 1)

6.1 Cliente: *Realiza la firma electrónica de un mensaje de datos usando una clave clave criptográfica de cifrado obtenida previamente del* ***servidor A****. También, puede realizar la comprobación de integridad de un mensaje de datos firmado electrónicamente.*

6.1.1 Solicita al **servidor A** la generación y entrega de una CLAVE criptográfica.

6.1.2 Solicita al **servidor B** la autenticación de la identidad del usuario de una CLAVE criptográfica.

6.2 Servidor Proxy: *Redirige las consultas de los clientes al servidor correspondiente (de claves o de autenticación), asimismo las respuestas hacia los clientes.*

6.2.1 Redirige del cliente la solicitud de CLAVE criptográfica y de validación al servidor apropiado.

6.3.2 Recibe la respuesta del servidor y la redirige al cliente apropiado.

6.3 Servidor de claves: *Genera claves y las almacena junto con la Identidad de los usuarios.*

6.3.1 Genera aleatoriamente una CLAVE criptográfica a solicitud del cliente.

6.3.2 Cada CLAVE criptográfica generada se almacena en una tabla o en una base de datos junto con la identidad del cliente (usuario o signatario) a quien se le asignó.

6.4 Servidor de validación de identidad: *Autentica la identidad de los usarios que tienen clave.*

6.4.1 Recibe la solicitud de autenticación de una identidad de una CLAVE criptográfica.

6.4.2 Responde al cliente con identidad VÁLIDA o INVÁLIDA.

7. ARCHIVOS DE ENTRADA DE PRUEBA: *Use estos archivos para probar el funcionamiento de su aplicación. El profesor podrá usar otros archivos de prueba para verificar el software desarrollado.*

6.1 Instancia de prueba No. 1 de Solicitud de Clave de Usuario para Firmar:

FIRMAR Julio Chaykovsky “Saludos desde Rusia” 0	<Aqui la Clave de Chaykovsky> <Aqui va el texto de la firma> 0	45277892 Pedro Pérez, <Aqui la Clave de Chaykovsky> Julio Chaykovsky, 0
a) Archivo de Entrada	**b) Archivo de Salida**	**c) Archivo de Claves e Identidades**

6.2 Instancia de prueba No. 2 de Solicitud de Verificación de Integridad del Mensaje:

INTEGRIDAD <Aqui la Clave de Chaykovsky> “Saludos desde Rusia” <Aqui va el texto de la firma> 0	<INTEGRO / NO INTEGRO>	45277892 Pedro Pérez, <Aqui la Clave de Chaykovsky> Julio Chaykovsky, 0
a) Archivo de Entrada	**b) Archivo de Salida**	**c) Archivo de Claves e Identidades**

6.3 Instancia de prueba No. 3 de Solicitud de Autenticación de Identidad del Firmante:

AUTENTICAR 45277892 Pedro Pérez, 0	<VÁLIDO>	45277892 Pedro Pérez, <Aqui la Clave de Chaykovsky> Julio Chaykovsky, 0
a) Archivo de Entrada	**b) Archivo de Salida**	**c) Archivo de Claves e Identidades**

7. FORMATO DEL INFORME A ENTREGAR

7.1 Portada: Encabezado, Titulo del Informe, nombres/apellidos/cédulas de los integrantes del equipo y fecha de entrega.

7.2 Introducción: Antecedente del tema de de desarrollo del proyecto, definiciones, problema, objetivo del proyecto, etc.

7.3 Diseño del Sistema: Diagrama que muestre el diseño de los componentes lógicos que interactúan y conforman el Sistema Distribuido de Firma Electrónica.

7.4 Implementación del código del cliente: Código fuente intradocumentado.

7.5 Implementación del código del proxy: Código fuente intradocumentado

7.6 Implementación del código del servidor: Código fuente intradocumentado

7.7 Resultados: Pantallas comentadas que muestren la traza de ejecución con los resultados.

7.8 Conclusiones: Reflexiones sobre lo desarrolado como proyecto, sobre el tema involucrado, qué se logró con el proyecto y comentarios sobre los resultados del sistema distribuido.

Éxito a todos!
Caracas, 24 de Noviembre de 2022
CAAL