Firma Digital INTI

Vitale Luciano Nahuel[1]
40389087

Director de beca: Ing. Gustavo Escudero[1]

[1]INTI,
Departamento de Validación de Dispositivos y Sistemas Electrónicos,
Av. General Paz Nº 5445 - San Martín, Buenos Aires, Argentina

Resumen. Software multiplataforma para la creación de claves privadas, claves públicas, digestos, firma de archivos y verificación de la firma

Palabras claves: Clave privada, Clave pública, Digesto, Firma, Hash, Cifrado.

1 Introducción

La **firma digital** es el resultado de aplicar a un documento digital un procedimiento matemático que requiere información de exclusivo conocimiento del firmante, encontrándose ésta bajo su absoluto control. La firma digital debe ser susceptible de verificación por terceras partes, tal que dicha verificación simultáneamente permita identificar al firmante y detectar cualquier alteración del documento digital posterior a su firma. [1]

La **firma electrónica** es el conjunto de datos electrónicos integrados, ligados o asociados de manera lógica a otros datos electrónicos, utilizado por el signatario como su medio de identificación, que carezca de alguno de los requisitos legales para ser considerada firma digital. En caso de ser desconocida la firma electrónica corresponde a quien la invoca acreditar su validez. [1]

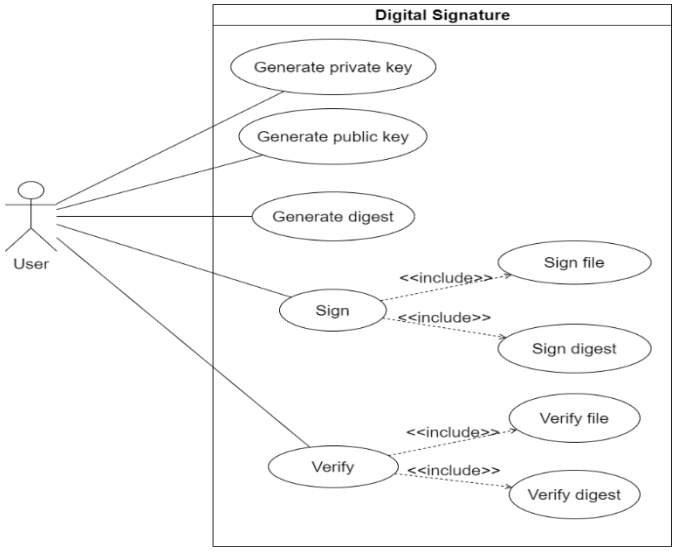
- La funcionalidad del software es permitir la generación, con distintos tipos de cifrado, de claves privadas y claves públicas para firmar archivos. A su vez, se permite generar un archivo de digesto con los archivos originales aplicando un hash definido. Además, a través de cualquier clave privada indicada en un archivo de extensión “pem” es posible crear la firma de un archivo o digesto y hacer su verificación.

2 Desarrollo

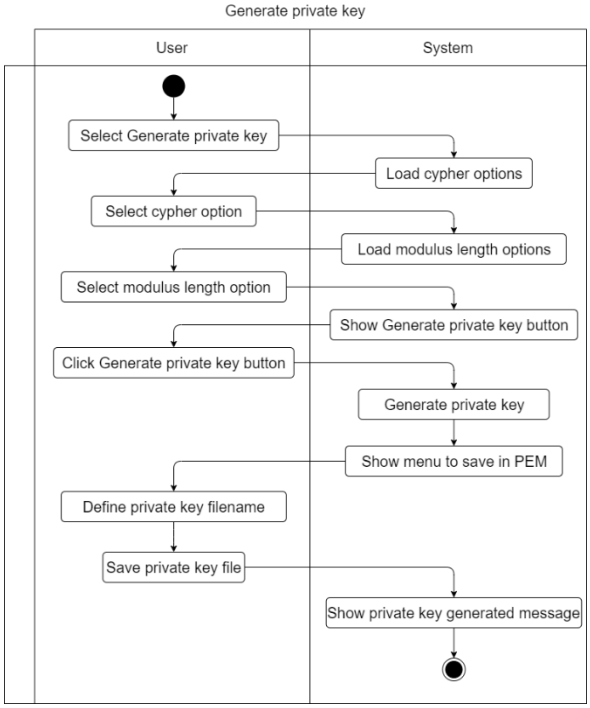
- **Dirección web del repositorio GitHub:**

  https://github.com/luvitale/inti-digital-signature

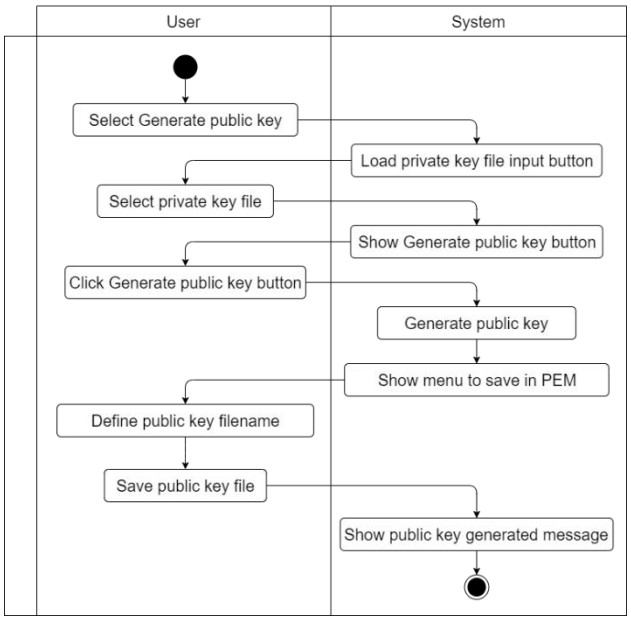
- **Casos de uso**



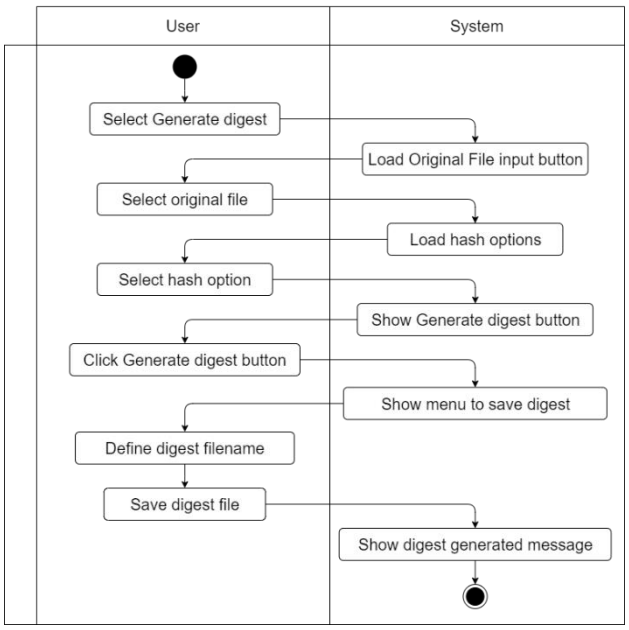
- **Diagramas de actividades**



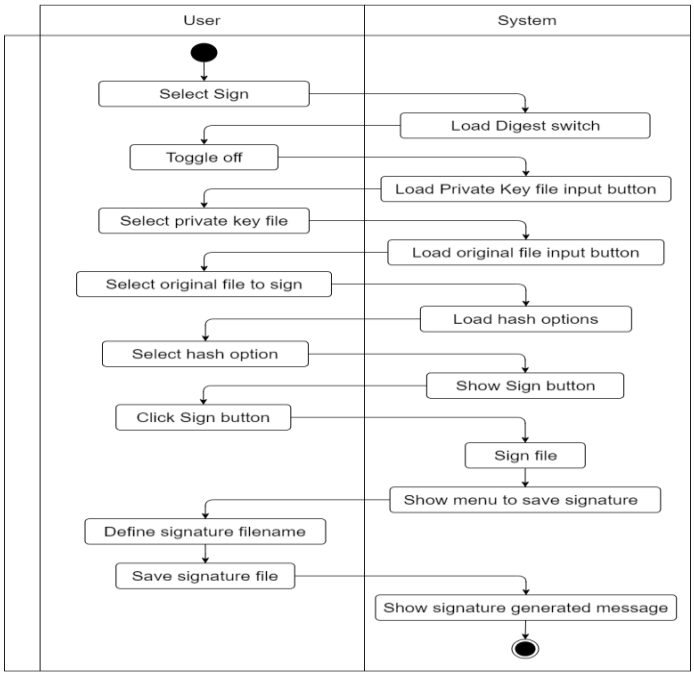
Generate public key



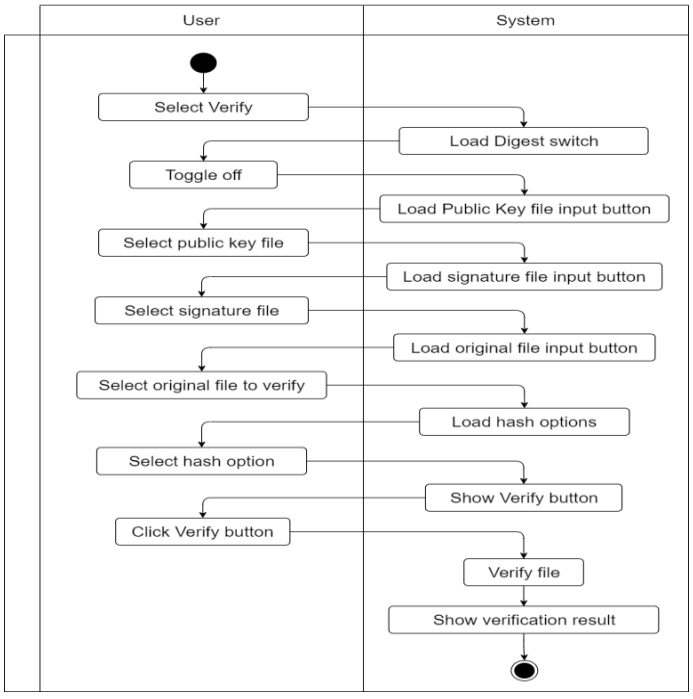
Generate digest



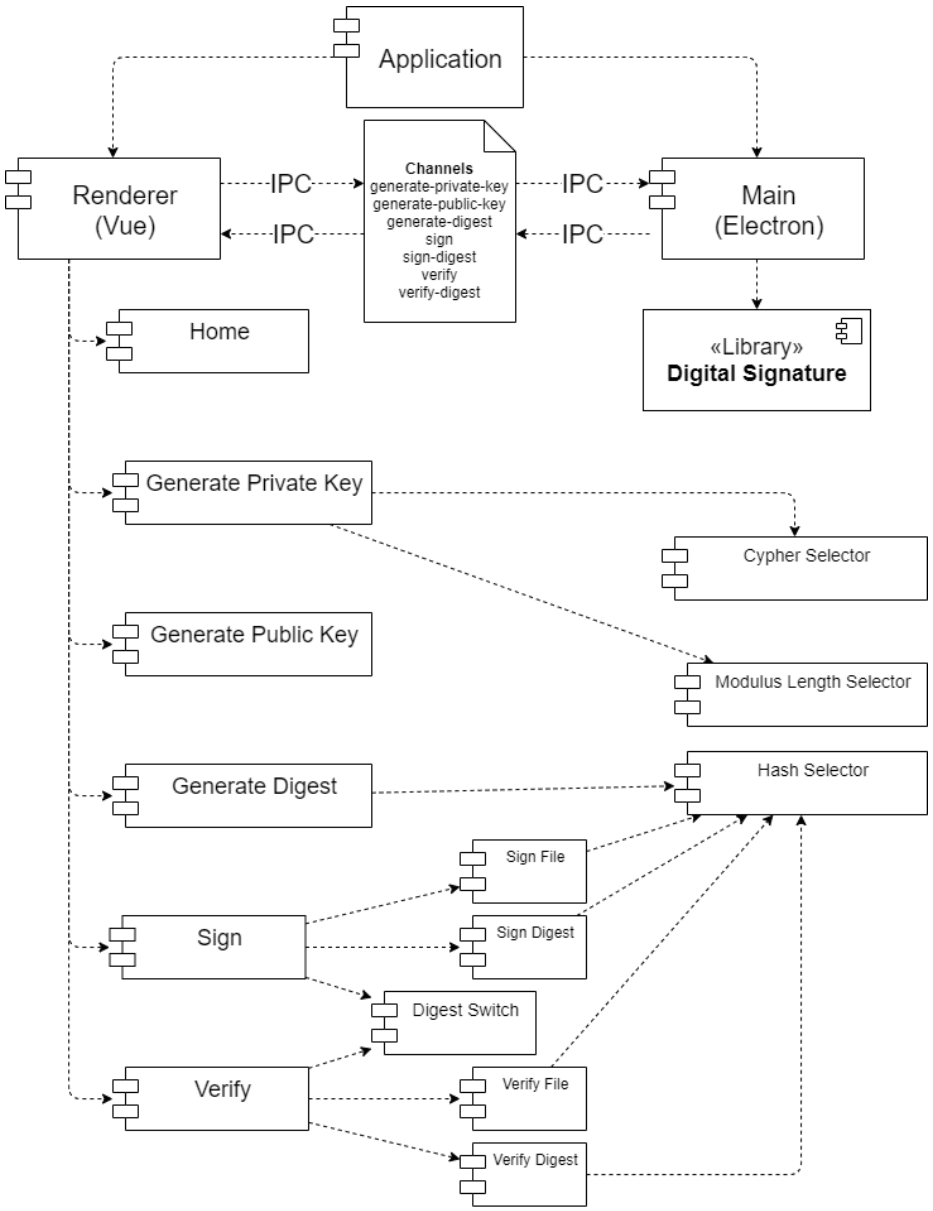
Sign file



Verify file



- **Diagrama de componentes**



3 Conclusiones

Para construir una solución para generar claves privadas, claves públicas, digestos, firmar y verificar las firmas se ha desarrollado un software con una vista implementada con Vue (Renderer) que se comunica con una biblioteca implementada con Electron (Main) a través de mensajes de IPC.

- Se utilizó Vuex como administrador de estados para almacenar la información de los archivos ingresados y las opciones seleccionadas. A su vez se hizo uso de sus acciones para enviar los mensajes requeridos para ejecutar la funcionalidad correspondiente y recibir la respuesta a través de mecanismos IPC provistos por Electron. [2]
- Se realizó una validación de lo ingresado por el usuario utilizando los mecanismos de validación de los formularios de Vuetify informando los elementos que no fueron ingresados y son requeridos para permitir el funcionamiento de los botones de acción. [3]
- Fue necesario deshabilitar la integración para contenido remoto siguiendo las buenas prácticas y recomendaciones para reducir los riesgos de seguridad incorporando un script de precargado (preload) para definir los canales IPC permitidos. [4] [5]

4 Referencias

[1] Ministerio de Justicia y Derechos Humanos, Presidencia de la Nación, 2001. [En línea]. Available: http://servicios.infoleg.gob.ar/infolegInternet/anexos/70000-74999/70749/norma.htm.

[2] Vuetify, «Form Component» [En línea]. Available: https://vuetifyjs.com/en/components/forms/. [Último acceso: 2021].

[3] Vue, «What is Vuex?» [En línea]. Available: https://vuex.vuejs.org/. [Último acceso: 2021].

[4] N. Klayman, «Security | Vue CLI Plugin Electron Builder» 2021. [En línea]. Available: https://nklayman.github.io/vue-cli-plugin-electron-builder/guide/security.html#node-integration.

[5] Electron, «Security, Native Capabilities, and Your Responsibility» [En línea]. Available: https://www.electronjs.org/docs/latest/tutorial/security. [Último acceso: 2021].