



# Alison

## SDK Javascript

Alison SDK es una de las piezas fundamentales para la integración de la funcionalidad de firma en navegadores.

La librería cuenta con diversos objetos que pueden ser utilizados por el programador para indicar qué tipo de documentos se pueden firmar y qué tipo de firmas se desean generar.

Los certificados a ser utilizados pueden encontrarse instalados en el equipo del usuario o en un servidor. La librería permite mostrar todos los certificados de manera similar para que el usuario seleccione el apropiado para el uso al que lo va a destinar.

Por medio de un archivo de configuración, que es alojado en el servidor, es posible configurar todo el comportamiento de la librería, especialmente cuando opera en modalidad gráfica.

Al encontrarse desarrollada 100% en lenguaje Javascript la misma es compatible con las versiones actuales de todos los navegadores. La misma ha sido homologada en Internet Explorer, Microsoft Edge, Mozilla Firefox, Google Chrome, Opera y Safari.

Para realizar la generación de la firma la librería se comunica con el Alison Provider correspondiente.

Los Alison Providers disponibles o requeridos son definidos en el archivo de configuración.

Si el certificado ha sido provisto desde un servidor (Alison Server), el usuario deberá cumplir con el challenge requerido para garantizar su titularidad antes que pueda ejecutar la función de firma.



Cuando se trate de certificados instalados en su propia computadora es necesario que el usuario tenga instalado Alison Desktop<sup>1</sup>.

Alison Desktop y Alison Server generan firmas en los diversos formatos más utilizados tales como CAAdES, XAdES, XML-DSign y PAdES.

## Componentes y Estándares

Alison SDK (Javascript) es una librería con un conjunto de objetos destinados a realizar la firma y recuperar su resultado.

El resultado de las firmas, los datos del firmante, los códigos de error y todo tipo de estructuras se encuentran definidos en JSON para que puedan ser fácilmente transmitidas y utilizadas por el programador.

La librería puede ser operada concurrentemente con interfaz de usuario o de manera transparente.

En el primero de los casos el usuario cuenta con una interfaz para seleccionar el certificado a utilizar y puede visualizar el texto a firmar si el mismo se encuentra en formato HTML o Texto.

El estilo de las ventanas, basadas en bootstrap, puede ser personalizado con colores y logos propios de una manera sumamente sencilla.

Sin embargo, si el programador quiere hacer uso de la librería sin la interfaz gráfica deberá proveer información al momento de generar la firma, tal como el identificador del certificado a utilizar.

Un ejemplo de la ventana de selección de un certificado para realizar una validación del certificado se puede observar en el gráfico anterior.



<sup>1</sup> Alison Desktop se encuentra disponible para plataformas Windows, MacOS y Linux. Un asistente es utilizado para realizar su instalación e integración con el navegador.



## Funciones principales

Alison SDK cuenta con los siguientes tipos de funciones:

- Inicialización

Durante esta etapa se indican las características principales, tal como el listado de los proveedores de certificados que se desean acceder y cómo comportarse ante la ausencia de alguno de ellos.

Por ejemplo, en el caso de que el proveedor Alison Desktop sea requerido de manera obligatoria, el usuario recibe una notificación para que sea direccionado al asistente de instalación.

También se indican los logos o repositorio de mensajes para cada uno de los idiomas<sup>2</sup>.

```
var ali_sdk = null;
var configFile = "../../alison/config/alison-config.desktop_d.json";

/*
 * Funcion ejemplo de firma.
 */
function alisonSignGui() {
  try {
    // Cargamos archivo de configuracion
    ali_sdk = new AlisonWebGuiSDK(configFile, function(opResult0) {
      if (!ali_sdk.isLoaded()) {
        // Si no se pudo cargar exitosamente, entonces salir.
        alert("Alison not loaded!!!!: " + JSON.stringify(opResult0));
        return;
      }
      // Armamos un contexto de usuario con el userID 12345678
      var uContext = new UserContext("12345678");

      // Inicializamos la librería.
      ali_sdk.initialize(uContext, null, function(opResult1) {
        if (opResult1.isStatus(AlisonConst.OP_ERROR)) {
          // Si se produjo un error sale
          alert("Alison Initalize Error: "+JSON.stringify(opResult1)); return;
        } else {
          // Creamos una operacion de firma y agregamos un texto a firmar
          var certSelector = null;
          var sign = new AlisonSignatureOperation(AlisonConst.OPERATION_SIGN, null);
          var dataS = new AlisonSignedData("<strong>This</strong> is an example <i>of signature</i>.",
            AlisonConst.SGN_TYPE_CADES, AlisonConst.FMT_TEXT_HTML);
          sign.addSignedData(dataS);
          // Procesamos la operacion de firma y mostramos el resultado.
          ali_sdk.processSignature(sign, function (opResult2, values) {
            alert("Alison Signature Result: "+JSON.stringify(opResult2));
            return;
          });
        }
      });
    });
  } catch (e) {
    alert("Alison Library not detected!: "+e); return;
  }
}
```

<sup>2</sup> La librería tiene mensajes en idioma inglés y castellano. Otros idiomas, tal como portugués, serán incorporados en futuras versiones



- Listar certificados

El listado de certificados disponibles puede ser filtrado por el programador aunque principalmente puede ser hecho en el archivo de configuración.

Un conjunto de filtros predefinidos (para Listar, para Firmar, para Probar) puede ser definido y se aplican de manera automática a cada una de las funciones.

El listado incluye todos los certificados de cada uno de los proveedores de certificados definidos en el archivo de configuración y detectados durante la inicialización de la librería.

- Autenticar / Probar / Firmar

Cada una de estas funciones produce como resultado una firma, sin perjuicio de que el mensaje que el usuario reciba sea distinto, en función de la operación que ejecutó. El programador puede, además de los filtros propios de cada función, seleccionar el certificado que el usuario debe utilizar.

El resultado de una firma exitosa es devuelto en una estructura JSON junto con el certificado del firmante, tanto en formato Base64 como en una estructura WebCertificate<sup>3</sup>, que facilita su posterior procesamiento.

- Funciones de configuración

Un conjunto de funciones adicionales, tales como setear el idioma o configurar filtros ad-hoc, pueden ser invocados para lograr una personalización mayor o más dinámica.

Habitualmente y preferentemente todas estas características son detectadas de manera automática o pueden ser definidas en el archivo de configuración.

## Distribución y uso

---

La distribución de la librería contiene todos los componentes necesarios para su fácil integración y la documentación correspondiente en formato HTML.

<sup>3</sup> WebCertificate es una estructura JSON que contiene toda la información relevante de un certificado, tales como Titular, Emisor, periodo de validez, extensiones, etc, lo cual facilita el uso para un programador que desea mostrar o procesar datos de un certificado X509. También puede contener información adicional, tal como el Proveedor del Certificado que lo administra o el SSCD que lo contiene. Cuando el certificado ha sido validado por el servicio Debbie además puede contener información respecto de su estado.





Para facilitar su integración la librería cuenta con varios ejemplos de configuración, con proveedores de tipo Alison Desktop y Alison Server, así como ejemplos de integración sencillos como los que se encuentran en este documento.

Para probar cada una de las funcionalidades disponibles, se encuentra incluida una página de ejemplo donde es posible invocar a cada una de las funciones.

## Ventajas y Beneficios

---

- La librería Alison SDK se encuentra diseñada para que el programador no requiera conocimientos específicos en la materia, lo cual facilita su integración. Con solo tres pasos es posible realizar una operación de firma o autenticación.
- El filtro del tipo de certificado puede ser configurado posteriormente o realizar cambios en línea sin tener que afectar la aplicación desarrollada.
- Alison puede ser integrado fácilmente sobre cualquier circuito o sistema que requiera firma sobre un documento o su aprobación. Al encontrarse desarrollado 100% en Javascript, todas las versiones actuales de los navegadores pueden hacer uso del mismo.
- Los programadores no deben preocuparse por el tipo de navegador que empleará el usuario o cómo se verá la firma en cada uno de ellos, ya que estilo (basado en bootstrap) facilita la tarea y su integración. El estilo puede ser extendido o mejorado por el programador con un CSS propio.
- Los certificados, sin importar donde se encuentren instalados o cómo hayan sido gestionados, pueden ser accedidos desde cualquier navegador y con una experiencia uniforme para el usuario.
- Si un certificado se encuentra próximo a expirar, el usuario es alertado para que pueda tomar la acción de renovación correspondiente.
- La librería cuenta con funciones que permiten controlar si el usuario tiene instalado el proveedor de certificado apropiado, si existe una nueva versión y si requiere ser actualizado antes de continuar.