

SeguriData Privada, S.A. de C.V.

Av. Insurgentes Sur #2375, 3er. piso,
Col. Tizapán, Del. Alvaro Obregón,
C.P. 01000, México, D.F.

Tel. +52 (55) 3098-0700

Fax. +52 (55) 3098-0702

<http://www.seguridata.com>

Derechos Reservados © SeguriData IP S.A. de C.V., Av. Insurgentes Sur #2375, 3er. piso, Col. Tizapán, Del. Alvaro Obregón, C.P. 01000, México, D.F. Este producto constituye una obra intelectual protegida por las leyes nacionales y tratados internacionales en materia de derechos de autor, y queda prohibida su reproducción o uso total o parcial, que no sean autorizadas por su titular.

Número de Parte: SeguriNotary_Funciones_6.2

Contenido

Capítulo 1. ¿Cómo utilizar este manual?

1.1 Simbología y convenciones	1 - 1
1.1.1. Recomendaciones y Advertencias	1 - 1
1.2 Objetivo de este Manual	1 - 1

Capítulo 2. Biblioteca de Funciones C

2.1 En qué consiste	2 - 1
2.2 Funciones	2 - 1

Capítulo 3. Ejemplos de uso de la Biblioteca de Funciones C

3.1 Ejemplo memoria.c	3 - 1
3.2 Ejemplo archivo.c	3 - 5

Capítulo 4. Biblioteca de Funciones Java Native Interface (JNI)

4.1 Métodos	4 - 1
4.2 Propiedades	4 - 6

Capítulo 5. Ejemplos de uso de la Biblioteca de Funciones Java Native Interface (JNI)

5.1 Ejemplo de Archivo	5 - 1
5.2 Ejemplo de Memoria	5 - 3
5.3 Ejemplo de Memoria ACBase64	5 - 4

CAPÍTULO 1

¿Cómo utilizar este manual?

1.1 Simbología y convenciones

En todo el manual se hace uso de una simbología específica para hacer más sencilla la identificación del tipo de información que se expone, así como de convenciones tipográficas, para hacer más clara la documentación.

1.1.1 Recomendaciones y Advertencias

En los lugares que resulte más oportuno, se insertarán comentarios sobre el contenido del texto.



Importante

Este tipo de anotaciones contienen sugerencias y aclaraciones que facilitan el uso de la aplicación.



Precaución

Este tipo de anotaciones advierten sobre posibles riesgos en las operaciones descritas en el texto y que pueden causar pérdida de funcionalidad o datos.

1.2 Objetivo de este Manual

Este manual sirve como referencia en el uso de la biblioteca de funciones de SeguriNotary (C y Java).

CAPÍTULO 2

Biblioteca de Funciones C

2.1 En qué consiste

Dentro de las capacidades de [SeguriNotary](#), se encuentra la de proporcionar un medio de acceso a ciertas funciones, que pueden invocarse desde cualquier aplicación propietaria. Las rutinas tienen la funcionalidad necesaria para:

- Solicitar una estampa de tiempo
- Decodificar una estampa de tiempo

2.2 Funciones

Solicita_Estampa_Tiempo_Memoria

Descripción. Solicita una estampa de tiempo y la deposita en un buffer.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] Solicita_Estampa_Tiempo_Memoria (char * cURL_Servidor,  
[2] unsigned char * cBuffer_Contenido_Archivo_Fuente,  
[3] unsigned int uiLongitud_Buffer_Contenido_Archivo_Fuente,  
[4] unsigned char * ucBuffer_Contenido_Archivo_AC,  
[5] unsigned int uiLongitud_Buffer_Contenido_Archivo_AC,  
[6] int iFormato_Estampa_Tiempo,  
[7] char * cBuffer_Estampa_Tiempo,  
[8] int * iLongitud_Buffer_Estampa_Tiempo,  
[9] char * cMensaje_Error);
```

Parámetros.

Parámetros de Entrada

El parámetro...	Es...
cURL_Servidor	la URL en donde se encuentra el servidor de SeguriNotary.
cBuffer_Contenido_Archivo_Fuente	es el buffer con la información a la que se le emitirá la estampa de tiempo.
uiLongitud_Buffer_Contenido_Archivo_Fuente	la longitud de la información.
ucBuffer_Contenido_Archivo_AC	el buffer con el contenido del certificado de la Autoridad Certificadora.
uiLongitud_Buffer_Contenido_Archivo_AC	la longitud del certificado de la Autoridad Certificadora.
iFormato_Estampa_Tiempo	el formato en el que se desea obtener la estampa de tiempo: 0- binario y 1-Base 64.

Parámetros de Salida

La variable...	De tipo...	Es...
cBuffer_Estampa_tiempo	char[8192]	el buffer en donde se devolverá la estampa de tiempo.
iLongitud_Buffer_Estampa_Tiempo	int *	la longitud de la estampa de tiempo.
cMensaje_Error	char[512]	el buffer en donde se devolverá el mensaje de error, en caso de que lo haya.

Solicita_Estampa_Tiempo_Archivo

Descripción. Solicita una estampa de tiempo y la deposita en un archivo.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] Solicita_Estampa_Tiempo_Archivo(char * cURL_Servidor,  
[2] char * cRuta_Archivo_Fuente,  
[3] char * cRuta_Archivo_AC,  
[4] int iFormato_Estampa_Tiempo,  
[5] char * cRuta_Archivo_Estampa_Tiempo,  
[6] char * cMensaje_Error);
```

Parámetros.

Parámetros de Entrada

El Parámetro...	Es...
cURL_Servidor	la URL en donde se encuentra el servidor de SeguriNotary.
cRuta_Archivo_Fuente	la ruta en donde se encuentra el archivo que contiene la información a la que se le emitirá la estampa de tiempo. Se deberá especificar la ubicación y nombre del archivo.
cRuta_Archivo_AC	la ruta en donde se encuentra el certificado de la Autoridad Certificadora. Se deberá especificar la ubicación y nombre del archivo.
iFormato_Estampa_Tiempo	el formato en el que se desea obtener la estampa de tiempo: 0- binario y 1-Base 64.

Parámetros de Salida

La variable...	De tipo...	Es...
cRuta_Archivo_Estampa_tiempo	char[255]	la ruta en donde se almacenará la estampa de tiempo. Se deberá especificar la ubicación y nombre del archivo.
cMensaje_Error	char[512]	el buffer en donde se devolverá el mensaje de error, en caso de que lo haya.

Decodifica_Estampa_Tiempo_Memoria

Descripción. Verifica que el formato de una estampa de tiempo almacenada en un buffer, sea válido.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] Decodifica_Estampa_Tiempo_Memoria(char * cBuffer_Estampa_Tiempo,  
[2] int iLongitud_Buffer_Estampa_Tiempo,  
[3] int iFormato_Estampa_Tiempo,  
[4] char * cDatos_Estampa_Tiempo,  
[5] char * cMensaje_Error);
```

Parámetros.

Parámetros de Entrada

El parámetro...	Es...
cBuffer_Estampa_Tiempo	el buffer con el contenido de la estampa de tiempo a verificar.
iLongitud_Buffer_Estampa_Tiempo	la longitud de la estampa de tiempo.
iFormato_Estampa_Tiempo	el formato en que se encuentra la estampa de tiempo: 0- binario y 1-Base 64.

Parámetros de Salida

La variable...	De tipo...	Es...
cDatos_Estampa_Tiempo	char[1024]	el buffer en donde se devolverán los datos de la estampa de tiempo.
cMensaje_Error	char[512]	el buffer en donde se devolverá el mensaje de error, en caso de que lo haya.

Decodifica_Estampa_Tiempo_Archivo

Descripción. Verifica que el formato de una estampa de tiempo almacenada en archivo, sea válido.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] Decodifica_Estampa_Tiempo_Archivo(char * cRuta_Archivo_Estampa_Tiempo,  
[2] int iFormato_Estampa_Tiempo,  
[3] char * cDatos_Estampa_Tiempo,  
[4] char * cMensaje_Error);
```

Parámetros.

Parámetros de Entrada

El parámetro...	Es...
cRuta_Archivo_Estampa_Tiempo	la ruta en donde se encuentra el archivo que contiene la estampa de tiempo a verificar. Se deberá especificar la ubicación y nombre del archivo.
iFormato_Estampa_Tiempo	El formato en que se encuentra la estampa de tiempo: 0- binario y 1-Base 64.

Parámetros de Salida

La variable...	De tipo...	Es...
cDatos_Estampa_Tiempo	char[1024]	el buffer en donde se devolverán los datos de la estampa de tiempo.
cMensaje_Error	char[512]	el buffer en donde se devolverá el mensaje de error, en caso de que lo haya.

Extrae_Certificado_Estampa_Tiempo_Memoria

Descripción. Verifica que el formato de una estampa de tiempo almacenada en un buffer, sea válido y extrae el certificado del firmante.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] Extrae_Certificado_Estampa_Tiempo_Memoria(char *cBuffer_Estampa_Tiempo,  
[2]      int iLongitud_Buffer_Estampa_Tiempo,  
[3]      int iFormato_Estampa_Tiempo,  
[4]      unsigned char *ucBuffer_Certificado_Firmante,  
[5]      int *iLongitud_Buffer_Certificado_Firmante,  
[6]      char *cDatos_Estampa_Tiempo,  
[7]      char *cMensaje_Error);
```

Parámetros de Entrada

El parámetro...	Es...
cBuffer_Estampa_Tiempo	el buffer con el contenido de la estampa de tiempo a verificar.
iLongitud_Buffer_Estampa_Tiempo	la longitud de la estampa de tiempo.
iFormato_Estampa_Tiempo	El formato en que se encuentra la estampa de tiempo: 0- binario y 1-Base 64.

Parámetros de Salida

La variable...	De tipo...	Es...
ucBuffer_Certificado_Firmante	unsigned char [6144]	el buffer en donde se devolverá el certificado del firmante de la estampa de tiempo.
iLongitud_Buffer_Certificado_Firmante	int *	la longitud del certificado del firmante.
cDatos_Estampa_Tiempo	char[1024]	el buffer en donde se devolverán los datos de la estampa de tiempo.
cMensaje_Error	char[512]	el buffer en donde se devolverá el mensaje de error, en caso de que lo haya.

Extrae_Certificado_Estampa_Tiempo_Archivo

Descripción. Verifica que el formato de una estampa de tiempo almacenada en archivo, sea válido y extrae el certificado del firmante.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[8] Extrae_Certificado_Estampa_Tiempo_Archivo(char *cRuta_Archivo_Estampa_Tiempo,
[9] int iFormato_Estampa_Tiempo,
[10] unsigned char *ucBuffer_Certificado_Firmante,
[11] int *iLongitud_Buffer_Certificado_Firmante,
[12] char *cDatos_Estampa_Tiempo,
[13] char *cMensaje_Error);
```

Parámetros de Entrada

El parámetro...	Es...
cRuta_Archivo_Estampa_Tiempo	la ruta en donde se encuentra el archivo que contiene la estampa de tiempo a verificar. Se deberá especificar la ubicación y nombre del archivo.
iFormato_Estampa_Tiempo	el formato en que se encuentra la estampa de tiempo: 0- binario y 1-Base 64.

Parámetros de Salida

La variable...	De tipo...	Es...
ucBuffer_Certificado_Firman te	unsigned char [6144]	el buffer en donde se devolverá el certificado del firmante de la estampa de tiempo.
iLongitud_Buffer_Certificad o_Firmante	int *	la longitud del certificado del firmante.
cDatos_Estampa_Tiempo	char[1024]	el buffer en donde se devolverán los datos de la estampa de tiempo.
cMensaje_Error	char[512]	el buffer en donde se devolverá el mensaje de error, en caso de que lo haya.

CAPÍTULO 3

Ejemplos de uso de la Biblioteca de Funciones C

3.1 Ejemplo memoria.c

```
[1]  /*****  
[2]  #include <stdio.h>  
[3]  #include <stdlib.h>  
[4]  #include <string.h>  
[5]  #include "snotary.h"  
[6]  *****/  
[7]  long Obtiene_Longitud_Archivo(FILE * Apuntador_Archivo)  
[8]  {  
[9]      long Posicion_Actual;  
[10]     long Longitud;  
[11]  
[12]     if((Posicion_Actual = ftell(Apuntador_Archivo)) < 0)  
[13]         return -1;  
[14]     if(fseek(Apuntador_Archivo, 0L, SEEK_END))  
[15]         return -1;  
[16]     if((Longitud = ftell(Apuntador_Archivo)) < 0)  
[17]         return -1;  
[18]     if(fseek(Apuntador_Archivo, Posicion_Actual, SEEK_SET))  
[19]         return -1;  
[20]  
[21]     return Longitud;  
[22] }  
[23] /*****  
[24] int Guarda_Datos_Archivo(unsigned char * Datos,  
[25]                          unsigned int Longitud_Datos,  
[26]                          char * Nombre_Archivo)  
[27] {  
[28]     FILE * fp;  
[29]
```

```

[30]     if((fp = fopen(Nombre_Archivo, "wb")) == NULL)
[31]     {
[32]         return 0;
[33]     }
[34]
[35]     fwrite(Datos, 1, Longitud_Datos, fp);
[36]     fclose(fp);
[37]
[38]     return 1;
[39] }
[40] /*****
[41] int main(int argc, char* argv[])
[42] {
[43]     char URL_Servidor[255];
[44]     char Archivo_Fuente[255];
[45]     char Archivo_Destino[255];
[46]     char Archivo_AC[255];
[47]     int Numero_Estampas_Tiempo;
[48]     FILE * fp;
[49]     unsigned int Longitud_Buffer_Contenido_Archivo_Fuente;
[50]     unsigned char * Buffer_Contenido_Archivo_Fuente;
[51]     unsigned int Longitud_Buffer_Contenido_Archivo_AC;
[52]     unsigned char * Buffer_Contenido_Archivo_AC;
[53]     int i;
[54]     char Buffer_Estampa_Tiempo[1024*8];
[55]     int Longitud_Buffer_Estampa_Tiempo;
[56]     char Mensaje_Error[512];
[57]     char Datos_Estampa_Tiempo[1024];
[58]     int Formato_Estampa_Tiempo;
[59]
[60]     if(argc != 7)
[61]     {
[62]         printf("\n\nUso: memoria URL_Servidor Archivo_Fuente Formato_Estampa Archivo_Destino Certificado_AC
Numero_Estampas\n\n");
[63]         printf("donde:\n\n");
[64]         printf("URL_Servidor - URL del servidor TSP\n");
[65]         printf("Archivo_Fuente - Ruta y nombre del archivo del que se quiere obtener la estampa de tiempo\n");
[66]         printf("Formato_Estampa - Formato de la estampa de tiempo, 0 (Binario) 1 (Base64)\n");
[67]         printf("Archivo_Destino - Ruta y nombre del archivo donde se almacenara la estampa de tiempo\n");
[68]         printf("Certificado_AC - Ruta y nombre del archivo de la AC que emitio el certificado del servidor TSP\n");
[69]         printf("Numero_Estampas - Numero de estampas de tiempo a solicitar\n\n");
[70]         return 0;
[71]     }
[72]
[73]     strcpy(URL_Servidor, argv[1]);
[74]     strcpy(Archivo_Fuente, argv[2]);
[75]     Formato_Estampa_Tiempo = atoi(argv[3]);
[76]     strcpy(Archivo_Destino, argv[4]);
[77]     strcpy(Archivo_AC, argv[5]);
[78]     Numero_Estampas_Tiempo = atoi(argv[6]);

```

```
[79]
[80] if(fp = fopen(Archivo_Fuente, "rb")) == NULL)
[81] {
[82]     printf("\n\nError al leer el archivo %s\n\n", Archivo_Fuente);
[83]     return 0;
[84] }
[85]
[86] Longitud_Buffer_Contenido_Archivo_Fuente = (unsigned int)Obtiene_Longitud_Archivo(fp);
[87]
[88] if(Longitud_Buffer_Contenido_Archivo_Fuente <= 0)
[89] {
[90]     fclose(fp);
[91]     printf("\n\nEl archivo %s esta vacio\n\n", Archivo_Fuente);
[92]     return 0;
[93] }
[94]
[95] Buffer_Contenido_Archivo_Fuente = (unsigned char *)malloc(Longitud_Buffer_Contenido_Archivo_Fuente);
[96] if(Buffer_Contenido_Archivo_Fuente == NULL)
[97] {
[98]     fclose(fp);
[99]     printf("\n\nError al obtener memoria dinamica para el archivo %s\n\n", Archivo_Fuente);
[100]    return 0;
[101] }
[102] fread(Buffer_Contenido_Archivo_Fuente, 1, Longitud_Buffer_Contenido_Archivo_Fuente, fp);
[103] fclose(fp);
[104]
[105] if(fp = fopen(Archivo_AC, "rb")) == NULL)
[106] {
[107]     free(Buffer_Contenido_Archivo_Fuente);
[108]     printf("\n\nError al leer el archivo %s\n\n", Archivo_AC);
[109]     return 0;
[110] }
[111]
[112] Longitud_Buffer_Contenido_Archivo_AC = (unsigned int)Obtiene_Longitud_Archivo(fp);
[113]
[114] if(Longitud_Buffer_Contenido_Archivo_AC <= 0)
[115] {
[116]     fclose(fp);
[117]     free(Buffer_Contenido_Archivo_Fuente);
[118]     printf("\n\nEl archivo %s esta vacio\n\n", Archivo_AC);
[119]     return 0;
[120] }
[121]
[122] Buffer_Contenido_Archivo_AC = (unsigned char *)malloc(Longitud_Buffer_Contenido_Archivo_AC);
[123] if(Buffer_Contenido_Archivo_AC == NULL)
[124] {
[125]     fclose(fp);
[126]     free(Buffer_Contenido_Archivo_Fuente);
[127]     printf("\n\nError al obtener memoria dinamica para el archivo %s\n\n", Archivo_AC);
[128]    return 0;
```

```

[129]     }
[130]
[131]     fread(Buffer_Contenido_Archivo_AC, 1, Longitud_Buffer_Contenido_Archivo_AC, fp);
[132]     fclose(fp);
[133]
[134]     for(i = 1; i <= Numero_Estampas_Tiempo; i++)
[135]     {
[136]         if(!Solicita_Estampa_Tiempo_Memoria(URL_Servidor,
[137]             Buffer_Contenido_Archivo_Fuente,
[138]             Longitud_Buffer_Contenido_Archivo_Fuente,
[139]             Buffer_Contenido_Archivo_AC,
[140]             Longitud_Buffer_Contenido_Archivo_AC,
[141]             Formato_Estampa_Tiempo,
[142]             Buffer_Estampa_Tiempo,
[143]             &Longitud_Buffer_Estampa_Tiempo,
[144]             Mensaje_Error))
[145]         {
[146]             free(Buffer_Contenido_Archivo_AC);
[147]             free(Buffer_Contenido_Archivo_Fuente);
[148]             printf("\n\n%s\n\n", Mensaje_Error);
[149]             return 0;
[150]         }
[151]
[152]         printf("\nEstampa de tiempo %i/%i", i, Numero_Estampas_Tiempo);
[153]     }
[154]
[155]     if(!Decodifica_Estampa_Tiempo_Memoria(Buffer_Estampa_Tiempo,
[156]         Longitud_Buffer_Estampa_Tiempo,
[157]         Formato_Estampa_Tiempo,
[158]         Datos_Estampa_Tiempo,
[159]         Mensaje_Error))
[160]     {
[161]         free(Buffer_Contenido_Archivo_AC);
[162]         free(Buffer_Contenido_Archivo_Fuente);
[163]         printf("\n\n%s\n\n", Mensaje_Error);
[164]         return 0;
[165]     }
[166]
[167]     free(Buffer_Contenido_Archivo_AC);
[168]     free(Buffer_Contenido_Archivo_Fuente);
[169]
[170]     printf("\n\n%s\n\n", Datos_Estampa_Tiempo);
[171]
[172]     if(!Guarda_Datos_Archivo((unsigned char *)Buffer_Estampa_Tiempo,
[173]         (unsigned int)Longitud_Buffer_Estampa_Tiempo,
[174]         Archivo_Destino))
[175]     {
[176]         printf("\n\nError al almacenar la estampa de tiempo en el archivo %s\n\n", Archivo_Destino);
[177]         return 0;
[178]     }

```

```

[179]
[180]     return 1;
[181] }
[182] /*****
[183]

```

3.2 Ejemplo archivo.c

```

[1] #include <stdio.h>
[2] #include <stdlib.h>
[3] #include <string.h>
[4] #include "snotary.h"
[5] /*****
[6] int main(int argc, char* argv[])
[7] {
[8]     char URL_Servidor[255];
[9]     char Archivo_Fuente[255];
[10]    char Archivo_Destino[255];
[11]    char Archivo_AC[255];
[12]    int Numero_Estampas_Tiempo;
[13]    int i;
[14]    char Mensaje_Error[1024];
[15]    char Datos_Estampa_Tiempo[1024];
[16]    int Formato_Estampa_Tiempo;
[17]
[18]    if(argc != 7)
[19]    {
[20]        printf("\n\nUso: archivo URL_Servidor Archivo_Fuente Formato_Estampa Archivo_Destino Certificado_AC
Numero_Estampas\n\n");
[21]        printf("donde:\n\n");
[22]        printf("URL_Servidor - URL del servidor TSP\n");
[23]        printf("Archivo_Fuente - Ruta y nombre del archivo del que se quiere obtener la estampa de tiempo\n");
[24]        printf("Formato_Estampa - Formato de la estampa de tiempo, 0 (Binario) 1 (Base64)\n");
[25]        printf("Archivo_Destino - Ruta y nombre del archivo donde se almacenara la estampa de tiempo\n");
[26]        printf("Certificado_AC - Ruta y nombre del archivo de la AC que emitio el certificado del servidor TSP\n");
[27]        printf("Numero_Estampas - Numero de estampas de tiempo a solicitar\n\n");
[28]        return 0;
[29]    }
[30]
[31]    strcpy(URL_Servidor, argv[1]);
[32]    strcpy(Archivo_Fuente, argv[2]);
[33]    Formato_Estampa_Tiempo = atoi(argv[3]);
[34]    strcpy(Archivo_Destino, argv[4]);
[35]    strcpy(Archivo_AC, argv[5]);
[36]    Numero_Estampas_Tiempo = atoi(argv[6]);
[37]
[38]    for(i = 1; i <= Numero_Estampas_Tiempo; i++)
[39]    {
[40]        if(!Solicita_Estampa_Tiempo_Archivo(URL_Servidor,
[41]            Archivo_Fuente,

```



```
[42]             Archivo_AC,  
[43]             Formato_Estampa_Tiempo,  
[44]             Archivo_Destino,  
[45]             Mensaje_Error))  
[46]     {  
[47]         printf("\n\n%s\n\n", Mensaje_Error);  
[48]         return 0;  
[49]     }  
[50]  
[51]     printf("\nEstampa de tiempo %i/%i", i, Numero_Estampas_Tiempo);  
[52] }  
[53]  
[54] if(!Decodifica_Estampa_Tiempo_Archivo(Archivo_Destino,  
[55]             Formato_Estampa_Tiempo,  
[56]             Datos_Estampa_Tiempo,  
[57]             Mensaje_Error))  
[58] {  
[59]     printf("\n\n%s\n\n", Mensaje_Error);  
[60]     return 0;  
[61] }  
[62]  
[63] printf("\n\n%s\n\n", Datos_Estampa_Tiempo);  
[64]  
[65] return 1;  
[66] }  
[67] /*****  
[68]
```

CAPÍTULO 4

Biblioteca de Funciones Java
Native Interface (JNI)

A continuación se describen los métodos y propiedades de la biblioteca de funciones.

4.1 Métodos

Solicita_Estampa_Tiempo_Archivo

Descripción. Solicita una estampa de tiempo y la deposita en un archivo.
Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] public native int Solicita_Estampa_Tiempo_Archivo(String URL,  
[2]           String Archivo_Fuente,  
[3]           String Archivo_AC,  
[4]           String Archivo_Estampa);
```

Parámetros.

Parámetros de Entrada

El parámetro...	es...
URL	la URL en donde se encuentra el servidor de SeguriNotary.
Archivo_Fuente	el archivo con la información a la que se le emitirá la estampilla de tiempo. Se deberá especificar la ubicación y nombre del archivo.

Parámetros de Entrada (Continuación)

El parámetro...	es...
Archivo_AC	el archivo con el certificado de la Autoridad Certificadora. Se deberá especificar la ubicación y nombre del archivo.

Parámetros de Salida

La variable...	De tipo...	Es...
Archivo_Estampa	string	el archivo donde se almacenará la estampilla de tiempo. Se deberá especificar la ubicación y nombre del archivo.

Solicita_Estampa_Tiempo_Memoria

Descripción. Solicita una estampa de tiempo y la deposita en un string.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] public native int Solicita_Estampa_Tiempo_Memoria(String URL,  
[2]                                     String Fuente,  
[3]                                     String Archivo_AC);
```

Parámetros.

Parámetros de Entrada

El parámetro...	es...
URL	la URL en donde se encuentra el servidor de SeguriNotary.
Fuente	el string con la información a la que se le emitirá la estampilla de tiempo.
Archivo_AC	el archivo con el certificado de la Autoridad Certificadora. Se deberá especificar la ubicación y nombre del archivo.

Solicita_Estampa_Tiempo_Memoria_ACB64

Descripción. Solicita una estampa de tiempo y la deposita en un string.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] public native int Solicita_Estampa_Tiempo_Memoria_ACB64(String URL,  
[2]                               String Fuente,  
[3]                               String Certificado_AC);
```

Parámetros.

Parámetros de Entrada

El parámetro...	es...
URL	la URL en donde se encuentra el servidor de SeguriNotary.
Fuente	el string con la información a la que se le emitirá la estampilla de tiempo.
Certificado_AC	el string con el contenido del certificado de la Autoridad Certificadora en Base64.

Decodifica_Estampa_Tiempo_Archivo

Descripción. Verifica que el formato de una estampa de tiempo almacenada en un archivo, sea válido.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] public native int Decodifica_Estampa_Tiempo_Archivo(String Archivo_Estampa);
```

Parámetros.

Parámetros de Entrada

El parámetro...	es...
Archivo_Estampa	el archivo con la estampilla de tiempo a decodificar.

Decodifica_Estampa_Tiempo_Memoria

Descripción. Verifica que el formato de una estampa de tiempo almacenada en un string, sea válido.

Esta función devuelve un valor de tipo INT, indicando con 1 (uno) que la operación fue exitosa y 0 (cero) en caso contrario.

Sintaxis.

```
[1] public native int Decodifica_Estampa_Tiempo_Memoria(String Estampa);
```

Parámetros

Parámetros de Entrada

El parámetro...	es...
Estampa	el string con la estampilla de tiempo a decodificar.

4.2 Propiedades

Propiedades del JNI

La variable...	de tipo...	es...
Get_Mensaje_Error	string	el string en donde se devolverá el mensaje de error, en caso de que lo haya.
Get_Datos_Estampa_Tiempo	string	el string en donde se devolverán los datos de la estampilla de tiempo.
Get_Nombre_TSP	string	el string en donde se devolverá el nombre de la Autoridad de Time Stamping que emitió la estampilla de tiempo.
Get_Nombre_AC	string	el string en donde se devolverá el nombre de la Autoridad Certificadora que emitió el certificado de la Autoridad de Time Stamping.
Get_Fecha_Emision	string	el string en donde se devolverá la fecha (Tiempo Universal Coordinado) en la que fue emitida la estampilla de tiempo.
Get_Secuencia	string	el string en donde se devolverá el número de secuencia de la estampilla de tiempo.

Propiedades del JNI (Continuación)

La variable...	de tipo...	es...
Get_Digestion	string	el string en donde se devolverá la digestión SHA1 de la información a la que le fue emitida la estampilla de tiempo.
Get_Estampa_Tiempo	string	el string en donde se devolverá la estampilla de tiempo.

CAPÍTULO 5

Ejemplos de uso de la Biblioteca de Funciones Java Native Interface (JNI)

5.1 Ejemplo de Archivo

```
[1] import SeguriNotary.*;
[2]
[3] public class SNotaryTSPDemoArchivo
[4] {
[5]
[6]     public static void main(String[] args)
[7]     {
[8]         try
[9]         {
[10]             if(args.length != 4)
[11]             {
[12]                 System.out.println("\n\nUso: java SNotaryTSPDemo [URL] [Fuente] [AC] [Estampa]\n");
[13]                 System.out.println("Donde:\n");
[14]                 System.out.println("URL      - URL:Puerto donde se encuentra el servicio de SeguriNotary");
[15]                 System.out.println("Fuente  - Archivo que contiene los datos de los que se solicita una estampa de tiempo");
[16]                 System.out.println("AC      - Ruta al archivo del certificado de la autoridad certificadora");
[17]                 System.out.println("Estampa - Ruta al archivo de salida donde se almacenara la estampa de tiempo\n\n");
[18]                 return;
[19]             }
[20]
[21]             int resultado;
[22]             SeguriNotaryTSP snotarytsp = new SeguriNotaryTSP();
[23]
[24]             /* Operaciones Archivo */
[25]             resultado = snotarytsp.Solicita_Estampa_Tiempo_Archivo(args[0], /* URL */
```

```

[26]         args[1], /* Archivo Fuente */
[27]         args[2], /* Archivo AC */
[28]         args[3]); /* Archivo Estampa Tiempo*/
[29]
[30]     if (resultado == 1)
[31]     {
[32]         System.out.println("\n\nObtencion de estampa exitoso");
[33]     }
[34]     else
[35]     {
[36]         System.out.println("\n\nOcurrio el siguiente error al solicitar la estampa:\n");
[37]         System.out.println(snotarytsp.Get_Mensaje_Error());
[38]         return;
[39]     }
[40]
[41]     resultado = snotarytsp.Decodifica_Estampa_Tiempo_Archivo(args[3]);
[42]
[43]     if (resultado == 1)
[44]     {
[45]         System.out.println("\n\nDecodificacion de estampa exitoso");
[46]         Muestra_Datos(snotarytsp);
[47]     }
[48]     else
[49]     {
[50]         System.out.println("\n\nOcurrio el siguiente error al decodificar la estampa:\n");
[51]         System.out.println(snotarytsp.Get_Mensaje_Error());
[52]         return;
[53]     }
[54] }
[55] catch(Exception e)
[56] {
[57]     e.printStackTrace();
[58] }
[59] }
[60]
[61] public static void Muestra_Datos(SeguriNotaryTSP s)
[62] {
[63]     System.out.println("\n\n*****\n\nPropiedades del objeto SeguriNotaryTSP:\n\n");
[64]     System.out.println("Datos de la Estampa de Tiempo:\n\n" + s.Get_Datos_Estampa_Tiempo() + "\n\n");
[65]     System.out.println("Nombre del Servidor TSP: " + s.Get_Nombre_TSP());
[66]     System.out.println("Nombre de la AC: " + s.Get_Nombre_AC());
[67]     System.out.println("Fecha de Emision: " + s.Get_Fecha_Emision());
[68]     System.out.println("Numero de Secuencia: " + s.Get_Secuencia());
[69]     System.out.println("Digestion: " + s.Get_Digestion());
[70]     System.out.println("Mensaje de Error: " + s.Get_Mensaje_Error());
[71]     System.out.println("\n\n*****\n\n");
[72] }
[73] }
[74]

```

5.2 Ejemplo de Memoria

```
[1] import SeguriNotary.*;
[2]
[3] public class SNotaryTSPDemoMemoria
[4] {
[5]
[6]     public static void main(String[] args)
[7]     {
[8]         try
[9]         {
[10]             if(args.length != 2)
[11]             {
[12]                 System.out.println("\n\nUso: java SNotaryTSPDemoMemoria [URL] [AC]\n");
[13]                 System.out.println("Donde:\n");
[14]                 System.out.println("URL    - URL:Puerto donde se encuentra el servicio de SeguriNotary");
[15]                 System.out.println("AC    - Ruta al archivo del certificado de la autoridad certificadora\n\n");
[16]                 return;
[17]             }
[18]
[19]             int resultado;
[20]             SeguriNotaryTSP snotarytsp = new SeguriNotaryTSP();
[21]
[22]             /* Operaciones Memoria */
[23]             resultado = snotarytsp.Solicita_Estampa_Tiempo_Memoria(args[0], /* URL */
[24]                             "Texto de Prueba",
[25]                             args[1]); /* Archivo AC */
[26]
[27]             if (resultado == 1)
[28]             {
[29]                 System.out.println("\n\nObtencion de estampa exitoso");
[30]             }
[31]             else
[32]             {
[33]                 System.out.println("\n\nOcurrio el siguiente error al solicitar la estampa:\n");
[34]                 System.out.println(snotarytsp.Get_Mensaje_Error());
[35]                 return;
[36]             }
[37]
[38]             System.out.println("\n\nEstampa Tiempo:\n\n" + snotarytsp.Get_Estampa_Tiempo());
[39]
[40]             resultado = snotarytsp.Decodifica_Estampa_Tiempo_Memoria(snotarytsp.Get_Estampa_Tiempo());
[41]
[42]             if (resultado == 1)
[43]             {
[44]                 System.out.println("\n\nDecodificacion de estampa exitoso");
[45]                 Muestra_Datos(snotarytsp);
[46]             }
[47]             else
[48]             {
[49]                 System.out.println("\n\nOcurrio el siguiente error al decodificar la estampa:\n");
[50]                 System.out.println(snotarytsp.Get_Mensaje_Error());
```

```

[51]         return;
[52]     }
[53] }
[54] catch(Exception e)
[55] {
[56]     e.printStackTrace();
[57] }
[58] }
[59]
[60] public static void Muestra_Datos(SeguriNotaryTSP s)
[61] {
[62]     System.out.println("\n\n*****\n\nPropiedades del objeto SeguriNotaryTSP:\n\n");
[63]     System.out.println("Datos de la Estampa de Tiempo:\n\n" + s.Get_Datos_Estampa_Tiempo() + "\n\n");
[64]     System.out.println("Nombre del Servidor TSP: " + s.Get_Nombre_TSP());
[65]     System.out.println("Nombre de la AC: " + s.Get_Nombre_AC());
[66]     System.out.println("Fecha de Emision: " + s.Get_Fecha_Emision());
[67]     System.out.println("Numero de Secuencia: " + s.Get_Secuencia());
[68]     System.out.println("Digestion: " + s.Get_Digestion());
[69]     System.out.println("Mensaje de Error: " + s.Get_Mensaje_Error());
[70]     System.out.println("\n\n*****\n\n");
[71] }
[72] }
[73]

```

5.3 Ejemplo de Memoria ACBase64

```

[1] import SeguriNotary.*;
[2] import java.io.*;
[3]
[4] public class SNotaryTSPDemoMemoriaACB64
[5] {
[6]
[7]     public static void main(String[] args)
[8]     {
[9]         try
[10]        {
[11]            if(args.length != 2)
[12]            {
[13]                System.out.println("\n\nUso: java SNotaryTSPDemoMemoriaACB64 [URL] [AC]\n\n");
[14]                System.out.println("Donde:\n\n");
[15]                System.out.println("URL      - URL:Puerto donde se encuentra el servicio de SeguriNotary");
[16]                System.out.println("AC      - Ruta al archivo del certificado de la autoridad certificadora\n\n");
[17]                return;
[18]            }
[19]
[20]            byte[] Arreglo_Bytes = new byte[6144];
[21]            int Bytes_Leidos;
[22]
[23]            /* Se carga el certificado de la Autoridad Certificadora en memoria y se pasa a BASE64 */
[24]            DataInputStream Certificado_AC = new DataInputStream(new FileInputStream(args[1]));

```

```

[25]     Bytes_Leidos = Certificado_AC.read(Arreglo_Bytes);
[26]     String B64_Cert_AC = ConvertirBytetoB64(Arreglo_Bytes, Bytes_Leidos);
[27]
[28]     int resultado;
[29]     SeguriNotaryTSP snotarytsp = new SeguriNotaryTSP();
[30]
[31]     /* Operaciones Memoria */
[32]     resultado = snotarytsp.Solicita_Estampa_Tiempo_Memoria_ACB64(args[0], /* URL */
[33] "Texto de Prueba",
[34] B64_Cert_AC); /* Certificado AC en Base 64 */
[35]
[36]     if (resultado == 1)
[37]     {
[38]         System.out.println("\n\nObtencion de estampa exitoso");
[39]     }
[40]     else
[41]     {
[42]         System.out.println("\n\nOcurrio el siguiente error al solicitar la estampa:\n");
[43]         System.out.println(snotarytsp.Get_Mensaje_Error());
[44]         return;
[45]     }
[46]
[47]     System.out.println("\n\nEstampa Tiempo:\n\n" + snotarytsp.Get_Estampa_Tiempo());
[48]
[49]     resultado = snotarytsp.Decodifica_Estampa_Tiempo_Memoria(snotarytsp.Get_Estampa_Tiempo());
[50]
[51]     if (resultado == 1)
[52]     {
[53]         System.out.println("\n\nDecodificacion de estampa exitoso");
[54]         Muestra_Datos(snotarytsp);
[55]     }
[56]     else
[57]     {
[58]         System.out.println("\n\nOcurrio el siguiente error al decodificar la estampa:\n");
[59]         System.out.println(snotarytsp.Get_Mensaje_Error());
[60]         return;
[61]     }
[62] }
[63] catch(Exception e)
[64] {
[65]     e.printStackTrace();
[66] }
[67] }
[68]
[69] /* Esta funcion convierte un arreglo de bytes a formato BASE64 */
[70] public static String ConvertirBytetoB64(byte[] buf, int leidos)
[71] {
[72]     String s = null;
[73]     byte[] aux = new byte[leidos];
[74]

```

```
[75]     try
[76]     {
[77]         for(int x = 0; x < leidos; x++)
[78]         {
[79]             aux[x] = buf[x];
[80]         }
[81]
[82]         s = new sun.misc.BASE64Encoder().encode(aux);
[83]     }
[84]     catch(Exception e)
[85]     {
[86]         System.out.println("Error al convertir el arreglo de bytes a BASE64: " + e.getMessage());
[87]     }
[88]
[89]     return s;
[90] }
[91]
[92] public static void Muestra_Datos(SeguriNotaryTSP s)
[93] {
[94]     System.out.println("\n\n*****\n\nPropiedades del objeto SeguriNotaryTSP:\n\n");
[95]     System.out.println("Datos de la Estampa de Tiempo:\n\n" + s.Get_Datos_Estampa_Tiempo() + "\n\n");
[96]     System.out.println("Nombre del Servidor TSP: " + s.Get_Nombre_TSP());
[97]     System.out.println("Nombre de la AC:      " + s.Get_Nombre_AC());
[98]     System.out.println("Fecha de Emision:      " + s.Get_Fecha_Emision());
[99]     System.out.println("Numero de Secuencia:   " + s.Get_Secuencia());
[100]    System.out.println("Digestion:             " + s.Get_Digestion());
[101]    System.out.println("Mensaje de Error:      " + s.Get_Mensaje_Error());
[102]    System.out.println("\n\n*****\n\n");
[103] }
[104] }
```