Gabriel Molina

Manual técnico de la librería ZKSoftwareAPI

Gabriel Molina

1. Dispositivo ZKSoftware X628C.

El dispositivo X628C sirve para tomar el tiempo y asistencias del personal, las características técnicas del dispositivo se pueden observar en los siguientes links:

a) http://www.zk-software.com/628c.html
b) http://www.zksoftwares.com.mx/checador-zksoftware-x628c.html

En youtube se pueden apreciar varios videos acerca de su funcionamiento operativo, este manual está enfocado al desarrollo de aplicaciones utilizando el SDK del dispositivo y la librería ZKSoftwareAPI.dll

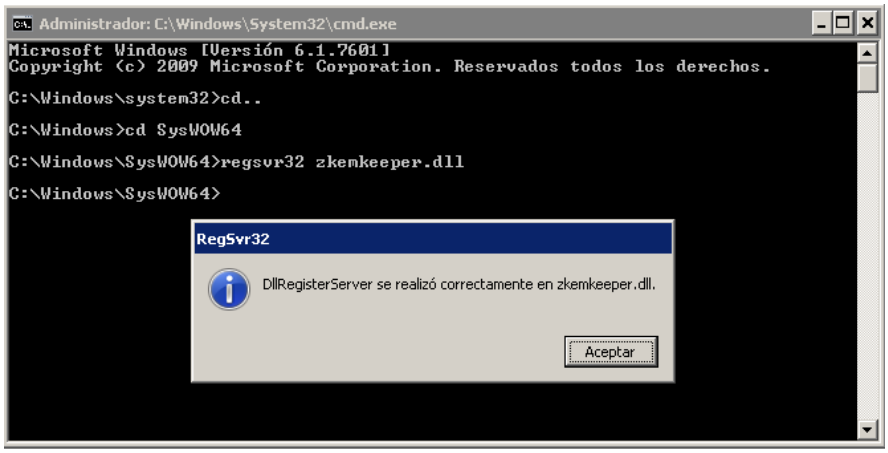
2. Instalación del SDK.

Los requisitos para la instalación del SDK son:

- SO Windows XP / 7 / 8 / 8.1 /Server 2003 / Server 2008/ Server 2008R2 (la librería y SDK fueron probados en cada uno de los SO mencionados y no se reportó error alguno).
- NET Framework 2.0 o superior

Para la instalación del SDK es necesario extraer los archivos que se encuentran dentro de ZKSoftwareAPI.rar, una vez ejecutada la acción es necesario **copiar todos los archivos que se encuentran dentro de la carpeta SDK** hacia la carpeta System32, **en caso de tener sistema operativo de 64 bits mover los archivos hacia la carpeta SysWOW64**, seguidamente abrimos la consola de Windows **en modo Administrador** y ejecutamos el comando **regsvr32 zkemkeeper.dll.**

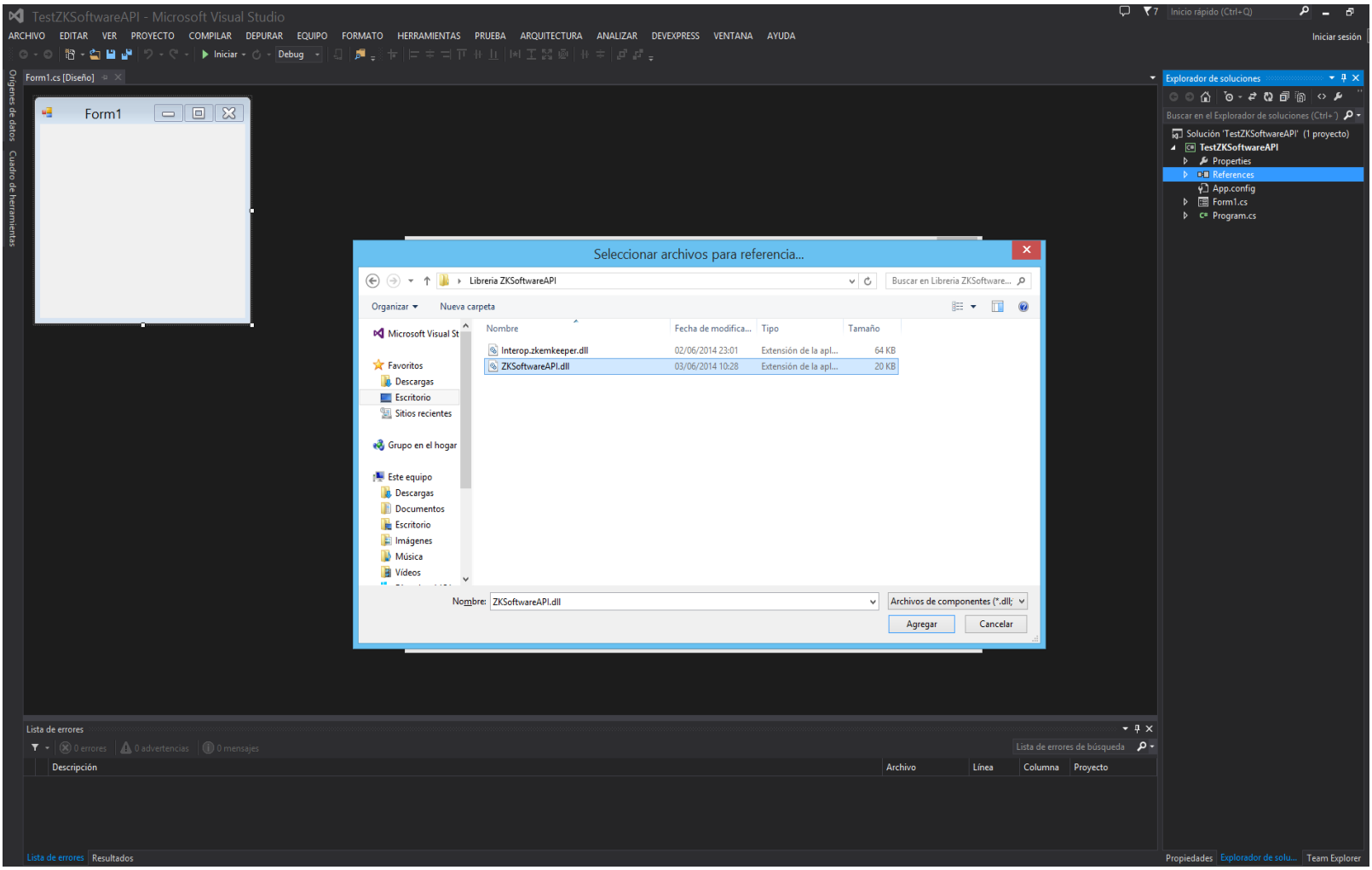
En la imagen que aparece debajo se muestra los pasos a seguir en caso de tener SO de 64 BITS y el mensaje que debe aparecer después de ejecutar el comando **regsvr32 zkemkeeper.dll**



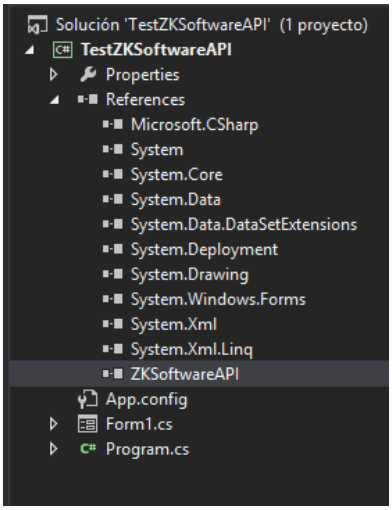
Gabriel Molina

3. Establecer la referencia en el proyecto.

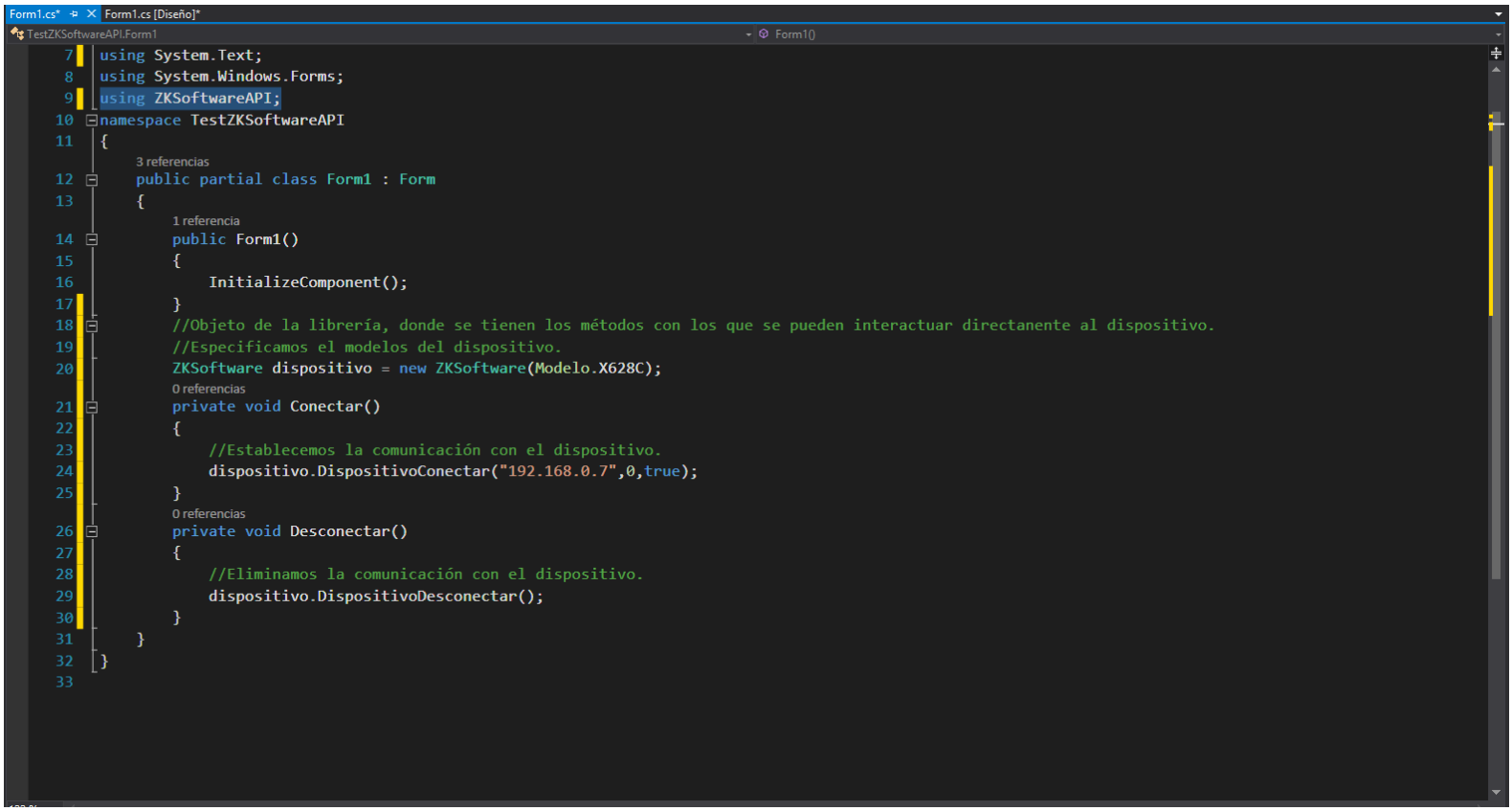
Creamos nuestro proyecto en VB o C# con el Framework 2.0 o superior, seguidamente damos clic derecho a nuestro proyecto seleccionamos Agregar Referencia y luego ubicamos la librería tal como se muestra en la imagen.



Después de agregar la referencia la podremos observar en nuestro explorador de soluciones.



Una vez establecida la referencia ya es posible trabajar con la librería, a continuación se muestra un ejemplo para la conexión y desconexión del dispositivo con el uso de la librería ZKSoftwareAPI.



4. Clases.

4.1. MarcajeOperativo

La clase MarcajeOperativo es un complemento de la clase UsuarioMarcaje, en la cual se guardan las diferentes operaciones que un usuario administrador puede realizar.

- NumeroDispositivo.- Número que tiene configurado el dispositivo, por default es 1, debido a que la librería solo soporta conexión a través de Ethernet el número de dispositivo siempre será 1, para conexiones de tipo COM o USB el número de dispositivo puede ser diferente a 1.
- Operacion.- Es la operación que se realizó.
- Parametro1, Parametro2, Parametro3, Parametro4.- Los valores de estos parámetros varía dependiendo de la propiedad Operacion. En la tabla se explican las operaciones y sus parámetros.

Operacion	Significado de Operacion	Parametro1	Parametro2	Parametro3	Parametro4
0	Encendido				
1	Apagado				
3	Alarma	Tipo de Alarma. 58: Falsa alarma. 54: Puerta alarma de entrada, 53: Botón de salida, 55: Se alteró el valor de la alarma, 65535: Alarma desactivada			
4	Entrada al Menu				
5	Cambio de configuración	Número de la configuración cambiada			
6	Registro de huella	Index del usuario el cuál registró la huella (el index no es el número de credencial del usuario, es el índice de la posición que ocupa en el dispositivo).	Resultado de la operación. 0: Correcto. Otros valores: Incorrecto.	Index de la huella registrada	Longitud de la cadena base64 de la huella registrada
7	Registro de contraseña				
14	Se creo una tarjeta MF	Index del usuario que lo realize		Número de huellas escritas en la tarjeta MF	Tamaño utilizado para las huellas en la tarjeta MF
20	Se copiaron datos de la tarjeta MF hacia el dispositivo			Número de huellas leídas de la tarjeta MF	
22	Se restauró la configuración de fabrica				
30	Registro de nuevo usuario	Index del usuario que lo realize	Resultado de la operación.		

4.2 UsuarioHuella.

La clase UsuarioHuella es un complemento para la clase UsuarioInformacion, por cada instancia de la clase UsuarioHuella se almacenan las propiedades descritas en la tabla.

Nombre propiedad	Tipo de dato	Descripción
B64Huella	string	Huella dactilar en base64
IndexHuella	int	Índice de la huella
LongitudHuella	int	Longitud de la propiedad B64Huella

4.3 UsuarioInformacion.

La clase UsuarioInformacion contiene la estructura de la información relacionada con un usuario.

Nombre propiedad	Tipo de dato	Descripción
Activo	Bool	Indica si el usuario está activo, true = Activo; false = Desactivado.
Contrasenia	string	Contraseña del usuario.
Huellas	List<UsuarioHuella>	Lista de las huellas que el usuario tiene registrado (ver punto 4.2).
Nombre	string	Nombre del usuario.
NumeroCredencial	string	Número de credencial del usuario.
Permiso	Enum.Permiso	Permiso del usuario (ver punto 6.4).

4.4 UsuarioMarcaje.

La clase UsuarioMarcaje almacena la información relacionada con el marcaje de los usuarios, cada instancia de la clase equivale a un marcaje, si el marcaje es de tipo operativo entonces la propiedad MarcajeOperativo es llenada con los datos que corresponden (para ver la información que se almacena ver el punto 4.1).

Nombre propiedad	Tipo de dato	Descripción
Dia	Int	Número de día.
Mes	Int	Número de mes.
Anio	Int	Número de año.
Hora	Int	Número de hora.
Minuto	Int	Número de minuto.
Segundo	Int	Número de segundo.
NumeroCrendencial	Int	Número de credencial del usuario.
MarcajeOperativo	MarcajeOperativo	Es null cuando el marcaje no es de tipo operativo, en caso contrario ver punto 4.1.

5. Propiedades.

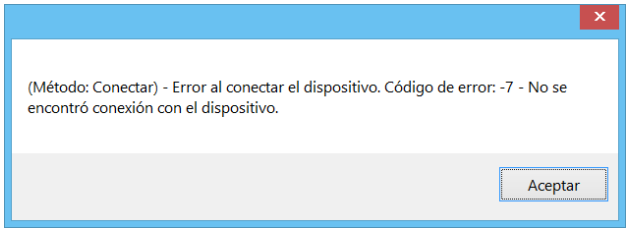
Las propiedades de la clase ZKSoftware se utilizan para almacenar los resultados de algunos métodos, a continuación se describen cada uno de ellos.

5.1.ERROR.

La propiedad ERROR almacena el error por los cuales un método no se ejecuta, todos los métodos devuelven true o false, cuando un método retorna false es necesario leer la propiedad ERROR para saber el motivo de la falla, ejemplo en el código:

```
21          private void Conectar()
22          {
23              //Establecemos la comunicación con el dispositivo.
24              if (!dispositivo.DispositivoConectar("192.168.0.221", 0, true))
25              {
26                  //Si el método retorna false, entonces mostramos un mensaje de error.
27                  MessageBox.Show(dispositivo.ERROR); //Aquí se lee la propiedad ERROR.
28              }
29          }
```

A continuación un ejemplo del error cuando no encuentre conexión:



5.2.ListaUsuarios.

La propiedad ListaUsuarios almacena una lista con objetos de tipo UsuarioInformacion. Esta propiedad se resetea y se actualiza cada vez que un método relacionado es llamado para ejecutarse. Tabla con los métodos relacionados para esta propiedad.

Métodos relacionados	Punto de referencia
UsuarioBuscarTodos	10.4
Clases relacionadas	**Punto de referencia**
UsuarioInformacion	4.3

5.3.ListaMarcajes.

La propiedad ListaMarcajes almacena una lista con objetos de tipo UsuarioMarcaje. Esta propiedad se resetea y se actualiza cada vez que un método relacionado es llamado para ejecutarse.

Métodos relacionados	Punto de referencia
DispositivoObtenerRegistrosAsistencias	11.3
DispositivoObtenerRegistrosOperativos	11.4
Clases relacionadas	**Punto de referencia**
UsuarioMarcaje	4.4

5.4.ResultadoConsulta.

La propiedad ResultadoConsulta almacena el resultado de los métodos relacionados con tipo de dato int.

Métodos relacionados	Punto de referencia
DispositivoConsultar	8.3

5.5.Usuario.

La propiedad Usuario almacena el resultado de los métodos relacionados, el dato es de tipo UsuarioInformacion.

Métodos relacionados	Punto de referencia
UsuarioBuscar	10.3
Clase relacionada	**Punto de referencia**
UsuarioInformación	4.3

6. Enumeradores.

6.1.Estatus

Valor	Descripción
Habilitar	Opción que se pasa como parámetro para habilitar el dispositivo.
Deshabilitar	Opción que se envía como parámetro para deshabilitar el dispositivo.

Métodos relacionados	Punto de referencia
`DispositivoCambiarEstatus`	`8.2`
`UsuarioBuscarTodos`	`10.4`

6.2.Modelo

Valor	Descripción
X628C	Modelo del dispositivo el cual se estará utilizando.
Métodos relacionados	**Punto de referencia**
`ZKSoftware (Constructor)`	`Consultar los ejemplos del punto 3.`

6.3.NumeroDe

Valor	Descripción
AdministradoresRegistrados	Opción que se pasa como parámetro para obtener el número de administradores registrados.
CapacidadActualAsistencias	Opción que se pasa como parámetro para obtener el número actual de registros de tipo de asistencia.
CapacidadActualHuellas	Opción que se pasa como parámetro para obtener la capacidad actual de huellas que todavía se pueden registrar.
CapacidadActualUsuarios	Opción que se pasa como parámetro para obtener la capacidad actual de Usuarios que todavía se pueden registrar.
CapacidadTotalHuellas	Opción que se pasa como parámetro para obtener la capacidad total de huellas.
CapacidadTotalAsistencias	Opción que se pasa como parámetro para obtener la capacidad total de asistencias.
CapacidadTotalFotos	Opción que se pasa como parámetro para obtener la capacidad total de fotos.
CapacidadTotalUsuarios	Opción que se pasa como parámetro para obtener la capacidad total de usuarios.
ContraseniasRegistradas	Opción que se pasa como parámetro para obtener el número de contraseñas que se encuentran registradas.
FotosRegistradas	Opción que se pasa como parámetro para obtener el número de fotos que se encuentran registradas.
HuellasRegistradas	Opción que se pasa como parámetro para obtener el número de huellas que se encuentran registradas.
RegistrosDeAsistencias	Opción que se pasa como parámetro para obtener el número de registros de asistencia que se encuentran en el dispositivo.
RegistrosDeOperativos	Opción que se pasa como parámetro para obtener el número de registros operativos que se encuentran en el dispositivo.
UsuariosRegistrados	Opción que se pasa como parámetro para obtener el número de usuarios registrados en el dispositivo.
Métodos relacionados	**Punto de referencia**
DispositivoConsultar	`8.3`

6.4.Permiso

Valor	Descripción
UsuarioNormal	Indica que el usuario solo puede realizar marcajes de entrada y salida.
UsuarioEnrolador	Indica que el usuario puede enrolar usuarios nuevos.
UsuarioSupervisor	Sin documentar
UsuarioAdministrador	Indica que el usuario puede realizar configuraciones en el dispositivo.
Métodos relacionados	**Punto de referencia**
UsuarioAgregar	10.1
UsuarioBuscar	10.3
UsuarioBuscarTodos	10.4

7. Comunicación con el dispositivo.

7.1. DispositivoConectar

Establece la comunicación con el dispositivo, para ver un ejemplo de funcionamiento consultar el punto 3.

Parámetro	Tipo de dato	Descripción
IP	string	IP del dispositivo al que se desea conectar.
IntentosConexion	int	Número de intentos de conexión en caso de fallar. 0 (CERO) equivale a un intento.
bBeep	bool	Indica si el dispositivo emitirá un sonido en caso de establecer la comunicación.
Respuesta	**Tipo de dato**	**Descripción**
true/false	bool	true = operación correcta, false = falló la operación
Métodos relacionados		
DispositivoDesconectar		
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.

7.2. DispositivoDesconectar

Finaliza la comunicación con el dispositivo, para ver un ejemplo de funcionamiento consulta el punto 3.

Métodos relacionados		Descripción
DispositivoConectar		Ver punto 7.1

8. Métodos generales.

8.1. DispositivoBeep

Envía una petición al dispositivo para que emita un sonido.

Métodos relacionados		Descripción
DispositivoConectar		Ver punto 7.1

8.2. DispositivoCambiarEstatus

Envía una petición al dispositivo para habilitar / deshabilitar el mismo.

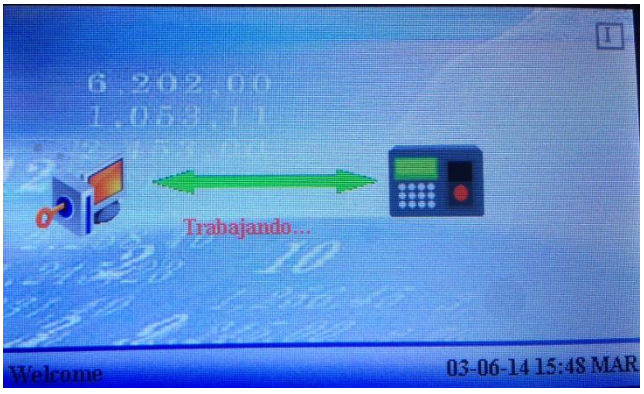
Parámetro	Tipo de dato	Descripción
Estatus	Estatus	Ver punto 6.1.
Respuesta	**Tipo de dato**	**Descripción**
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.

Ejemplo de uso:

```
ZKSoftware dispositivo = new ZKSoftware(Modelo.X628C);
1 referencia
private void CambiarEstado(Estatus estatus)
{
    if (!dispositivo.DispositivoCambiarEstatus(estatus))
    {
        MessageBox.Show(dispositivo.ERROR);
    }
}
1 referencia
private void btnConectar_Click(object sender, EventArgs e)
{
    Conectar();
    CambiarEstado(Estatus.Habilitar);
    dispositivo.DispositivoDesconectar();
}
```

NOTA: Cuando se modifica el estatus a Deshabilitado impide a los usuarios que puedan realizar marcajes, es muy importante volver a colocar el estatus Habilitado al dispositivo.

Mientras el estatus sea deshabilitado aparece la siguiente imagen en el dispositivo.



8.3. DispositivoConsultar

Realiza una consulta del dispositivo y dependiendo del parámetro enviado será la respuesta que se obtenga en la variable ResultadoConsulta.

Parámetro	Tipo de dato	Descripción
DatoConsultar	NumeroDe	Ver punto 6.3.
Respuesta	**Tipo de dato**	**Descripción**
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.
ResultadoConsulta		Ver punto 5.4.

Ejemplo de uso:

```
ZKSoftware dispositivo = new ZKSoftware(Modelo.X628C);
1 referencia
private void Consultar(NumeroDe numeroDe)
{
    if (dispositivo.DispositivoConsultar(numeroDe))
    {
        MessageBox.Show(dispositivo.ResultadoConsulta.ToString());
    }
    else
    {
        MessageBox.Show(dispositivo.ERROR);
    }
}
1 referencia
private void btnConectar_Click(object sender, EventArgs e)
{
    Conectar();
    Consultar(NumeroDe.UsuariosRegistrados);
    Desconectar();
}
```

9. Configuración de fecha y hora.

9.1. DispositivoCambiarHoraAutomatico

Realiza una petición al dispositivo para actualizar la hora, por default toma los valores de la computadora donde se realiza el llamado del método.

Respuesta	Tipo de dato	Descripción
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.

Ejemplo de uso:

```
ZKSoftware dispositivo = new ZKSoftware(Modelo.X628C);
1 referencia
private void CambiarHoraAutomatica()
{
    if(!dispositivo.DispositivoCambiarHoraAutomatico())
    {
        MessageBox.Show(dispositivo.ERROR);
    }
}
1 referencia
private void btnConectar_Click(object sender, EventArgs e)
{
    Conectar();
    CambiarHoraAutomatica();
    Desconectar();
}
```

9.2. DispositivoCambiarHoraManual

Realiza una petición al dispositivo para actualizar la hora.

Parámetro	Tipo de dato	Descripción
iDia	int	Número de día.
iMes	int	Número de mes.
iAnio	int	Número de año.
iHora	int	Número de hora.
iMinuto	int	Número de minuto.
iSegundo	int	Número de segundo.
Respuesta	**Tipo de dato**	**Descripción**
true/false	bool	true = operación correcta, false = falló la operación
Métodos relacionados		**Descripción**
CambiarEstatus		Internamente se ejecuta este método para que los usuarios no puedan realizar marcajes hasta que el dispositivo actualice la hora correctamente.
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.

Ejemplo de uso:

```
ZKSoftware dispositivo = new ZKSoftware(Modelo.X628C);
1 referencia
private void CambiarHoraManual(int Dia,int Mes,int Anio, int Hora, int Minuto, int Segundo)
{
    if (!dispositivo.DispositivoCambiarHoraManual(Dia,Mes,Anio,Hora,Minuto,Segundo))
    {
        MessageBox.Show(dispositivo.ERROR);
    }
}
1 referencia
private void btnConectar_Click(object sender, EventArgs e)
{
    Conectar();
    // Fecha y hora: 03/06/2014 17:00:00
    CambiarHoraManual(3, 6, 2014, 17, 0, 0);
    Desconectar();
}
```

10. Gestión de usuarios.

10.1. UsuarioAgregar

Agrega un nuevo usuario al dispositivo.

Parámetro	Tipo de dato	Descripción
NumeroCredencial	int	Número identificador del usuario al que se va a agregar.
UsuarioNombre	string	Nombre del usuario, máximo de caracteres 80.
TipoPermiso	Permiso	Ver punto 6.4.
indexHuella	int	Número de hora.
b64Huella	string	Número de minuto.
Respuesta	**Tipo de dato**	**Descripción**
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.

Ejemplo de uso:

```
private void AgregarUsuario(int numeroCredencial,string Nombre,Permiso permiso,int IndexHuella,string b64Huella)
{
    if (!dispositivo.UsuarioAgregar(numeroCredencial, Nombre, permiso, IndexHuella, b64Huella))
    {
        MessageBox.Show(dispositivo.ERROR);
    }
}
1 referencia
private void btnConectar_Click(object sender, EventArgs e)
{
    Conectar();
    AgregarUsuario(123456, "Gabriel Molina", Permiso.UsuarioAdministrador, 0, "AQUÍ_VA_LA_HUELLA_EN_BASE_64");
    Desconectar();
}
```

10.2. UsuarioBorrar

Elimina un usuario y todas sus huellas que se encuentren en el dispositivo.

Parámetro	Tipo de dato	Descripción
NumeroCredencial	int	Número identificador del usuario al que se va a eliminar.
Respuesta	**Tipo de dato**	**Descripción**
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.

Ejemplo de uso:

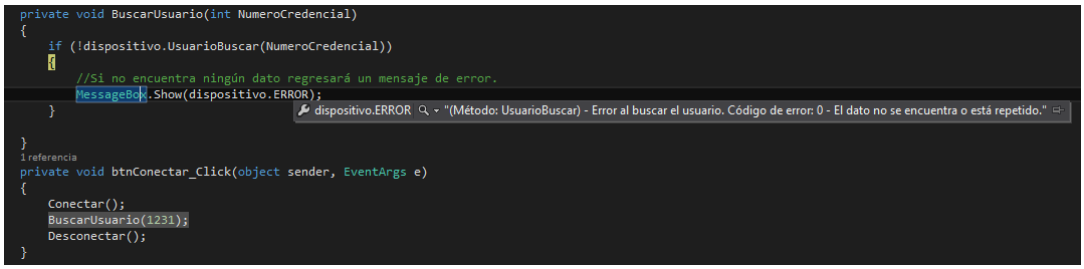
```
private void EliminarUsuario(int NumeroCredencial)
{
    if (!dispositivo.UsuarioBorrar(NumeroCredencial))
    {
        MessageBox.Show(dispositivo.ERROR);
    }
}
1 referencia
private void btnConectar_Click(object sender, EventArgs e)
{
    Conectar();
    EliminarUsuario(123456);
    Desconectar();
}
```

10.3. UsuarioBuscar

Obtiene un usuario del dispositivo y el resultado lo almacena en la propiedad Usuario, en caso de no encontrar ningún dato entonces envía un mensaje de error.

Parámetro	Tipo de dato	Descripción
NumeroCredencial	int	Número identificador del usuario que se quiere obtener.
Respuesta	**Tipo de dato**	**Descripción**
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.
Usuario		Ver punto 5.5.

Ejemplo de uso:

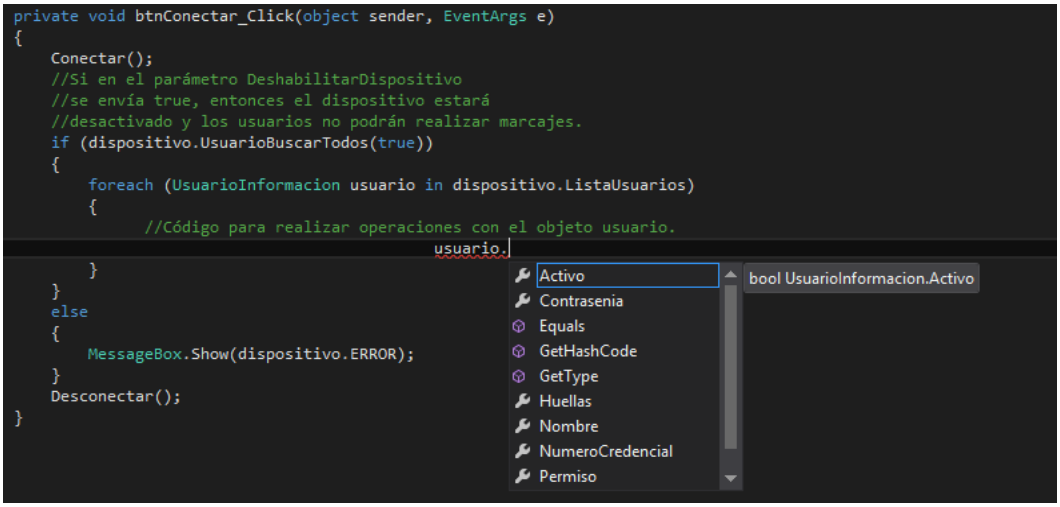


10.4. UsuarioBuscarTodos

Obtiene todos los usuarios que se encuentren registrados en el dispositivo y almacena el resultado en la propiedad ListaUsuarios.

Respuesta	Tipo de dato	Descripción
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.
ListaUsuarios		Ver punto 5.2.

Ejemplo de uso:



11. Manipulación de registros.

11.1. DispositivoBorrarRegistrosAsistencias

Elimina todos los registros de asistencia que se encuentren en el dispositivo.

Respuesta	Tipo de dato	Descripción
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.

Ejemplo de uso:

```
private void btnConectar_Click(object sender, EventArgs e)
{
    Conectar();
   if(!dispositivo.DispositivoBorrarRegistrosAsistencias())
   {
       MessageBox.Show(dispositivo.ERROR);
   }
    Desconectar();
}
```

11.2. DispositivoBorrarRegistrosOperativos

Elimina todos los registros de tipo operativo que se encuentren en el dispositivo.

Respuesta	Tipo de dato	Descripción
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.

Ejemplo de uso:

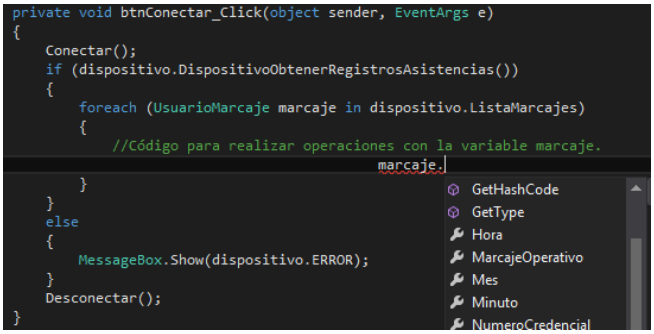
```
private void btnConectar_Click(object sender, EventArgs e)
{
    Conectar();
   if(!dispositivo.DispositivoBorrarRegistrosOperativos())
   {
       MessageBox.Show(dispositivo.ERROR);
   }
    Desconectar();
}
```

11.3. DispositivoObtenerRegistrosAsistencias

Obtiene todos los registros de asistencias y se almacena en la propiedad ListaMarcajes.

Respuesta	Tipo de dato	Descripción
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.
ListaMarcajes		Ver punto 5.3.

Ejemplo de uso:

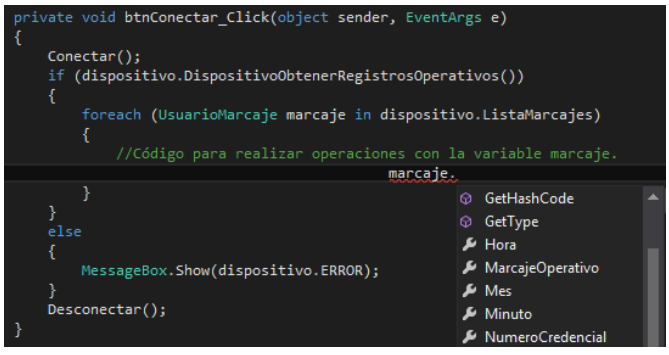


11.4. DispositivoObtenerRegistrosOperativos

Obtiene todos los registros de tipo operativo y se almacena en la propiedad ListaMarcajes.

Respuesta	Tipo de dato	Descripción
true/false	bool	true = operación correcta, false = falló la operación
Propiedades relacionadas		**Descripción**
ERROR		Ver punto 5.1.
ListaMarcajes		Ver punto 5.3.

Ejemplo de uso:



12. Bibliografía

A continuación la lista de todos los encales donde se obtuvo la información necesaria para elaborar la librería ZKSoftwareAPI.dll.

- http://www.zksoftwares.com.mx/checador-zksoftware-x628c.html
- http://usa.zkteco.com/view.do?id=43
- http://usa.zkteco.com/view.do?id=38
- http://www.zktechnology.com/Software.aspx
- http://www.zktechnology.eu/index.php/faq-category/sdk/registering-sdk
- https://www.youtube.com/watch?v=4jM8rDlP_ss
- https://www.youtube.com/watch?v=THx-HT3l4A8
- http://www.zktechnology.com/ProductDetail.aspx?cat=Biometric+Time+and+Attendance&series=X+Series&product=X628-C+
- http://www.reloj-checador.com/tienda/details/33/31/control-de-asistencia-biometrico/de-huella-digital/zk-x628c-reloj-checador-de-huella-redinternet-uso-rudousb.html

A continuación el video que se realizó como tutorial de la librería ZKSoftwareAPI.dll

- https://www.youtube.com/watch?v=f_CHbUWu69U&feature=youtu.be