

Manual TOTAL POS SDK NET

Para dispositivos Android

TOTAL POS SDK NET

Lima, 2026

Tabla de contenido

Objetivo del Documento.....	1
Definiciones.....	1
Diagrama de interacción.....	2
Fases para liberación.....	2
Transacciones Soportadas.....	3
Requerimientos de Hardware y Software.....	3
Dispositivo Pin Pad Android.....	4
Instalación de drivers.....	4
Configuración de dispositivo Android Serial (COM).....	7
Configuración de dispositivo Android Inalámbrico (Wifi).....	10
Pin Pad Pax A35.....	10
Implementación.....	15
Objeto Configuración e inicialización.....	15
Parámetros.....	15
Ejemplo de configuración e inicialización en Windows (Pinpad Android USB).....	16
Ejemplo de configuración e inicialización en Windows (Pinpad Android Wifi).....	17
Operaciones Retail.....	17
Parámetros de operación.....	18
Venta.....	19
Evento cuando se selecciona pago con QR.....	20
Venta QR.....	21
Venta PSI.....	23
Venta con cuotas.....	24
Anulación de Venta QR.....	25
Anulación de Venta.....	26
Consulta De Transacción.....	27
Operaciones Pre-autorización.....	27
Parámetros de operación.....	28
Pre-autorización.....	29
Confirmación Pre-autorización.....	30
Anulación Pre-autorización.....	31
Consulta de Pre-autorizaciones.....	32
Operaciones Cierre de Turno.....	32
Parámetros de operación.....	32

Cierre de Turno (Ejemplo).....	33
Respuesta de la operación.....	33
Manejo de errores.....	34
Consideraciones.....	34
Carga de llaves.....	34
Carga de llaves manual.....	35
Reversos.....	35
Reportes.....	35
Reporte Firma Física.....	36
Reporte Firma Totalizado.....	36
Reporte Firma Detallado.....	37
Reporte Cierre Resumen.....	38
Reporte Cierre Operaciones.....	39
Respuesta.....	40
Catálogos.....	42
Moneda.....	42
Firma.....	42
Códigos de Respuesta.....	42
Descripción de Clases.....	44
Clase Tarjeta.....	44
Clase Interfaz.....	45
Clase Configuración.....	45
Clase Petición.....	46
Clase Respuesta.....	47
Clase Reporteria.....	49
Clase RespuestaReporteFirma.....	49
Clase RespuestaReporteTotalizado.....	50
Clase RespuestaReporteDetallado.....	51
Clase RespuestaReporteResumen.....	52
Clase RespuestaReporteOperaciones.....	53
Clase Txn.....	54
Clase PeticionException.....	55
Clase PinPadKeysException.....	55
ANEXO Ejemplos Diagramación De Vouchers.....	56
1) PAGO CON TARJETA FISICA PIN.....	56
2) PAGO CON TARJETA FISICA CON FIRMA.....	57
3) PAGO CON TARJETA FISICA VENTA CON CUOTAS.....	58

4) PAGO CON QR.....	59
5) ANULACION PAGO CON TARJETA FISICA.....	60
6) ANULACION PAGO CON QR.....	61
7) VOUCHER CON CAMPAÑA CSI DINERS.....	62
8) VOUCHER CON CAMPAÑA PSI BBVA.....	63
Obtención de datos.....	63

Objetivo del Documento

Describir a detalle las funciones que integran la plataforma transaccional de TOTAL POS SDK para su implementación por parte del comercio en sus puntos de venta, para el servicio de autorización de tarjetas bancarias.

Definiciones

TOTAL POS SDK: Es la aplicación diseñada para que sea centralizada, escalable, segura y orientada a la alta disponibilidad a través de la división de procesos en API's utilizando la tecnología REST, estándares de tecnología (Como mensajería JSON) y buenas prácticas de programación e implementación basadas en las normas de PA-DSS y PCI.

Punto de venta: Controla todos los dispositivos para realizar una venta como son la interfaz con el operador, la báscula, la torreta, el lector de barras, impresión del pagaré, etc. Además, se encarga de interactuar con el sistema TOTAL POS SDK para la realización de operaciones financieras con tarjeta.

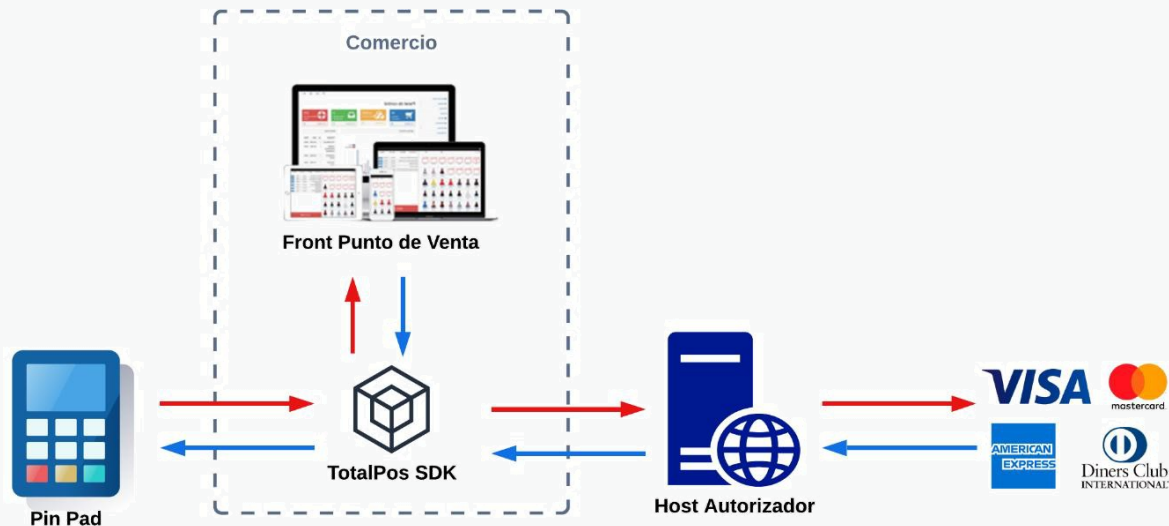
Afiliación: Es proporcionada por BBVA, está ligada con las cuentas en donde le deposita al comercio, además de ser el identificador asignado a los comercios.

Retail: Es el término utilizado para hacer referencia al conjunto de transacciones estándar de ventas, anulación de venta, venta QR y promociones.

Pago rápido: Esquema de pagos rápidos permite a los comercios emitir boletas sin firma para transacciones de compra que califiquen para el servicio.

Carga de Llaves: Es una operación que se debe realizar para que el dispositivo Pin Pad pueda cifrar los datos que obtiene al leer las tarjetas, es de carácter obligatoria. De esta forma se protege la integridad de los datos, si no se realiza la carga de llaves, el Pin Pad no funcionará.

Diagrama de interacción



1. **TOTAL POS SDK** al estar integrado con el punto de venta del comercio es quien manejará el envío y la recepción de la mensajería (JSON) con el host autorizador, al igual que maneja la mensajería con el Pin Pad para el uso de tarjetas de crédito y débito.
2. La conexión del punto de venta es por internet en donde la transacción viajará cifrada. Esta conexión puede ser de internet local o empresarial.
3. El Host Autorizador es quien recibe los requerimientos (transacción) para enviar estos datos a las marcas y solicitar una autorización. Una vez obtenida una respuesta de las marcas se devolverá la respuesta correspondiente al punto de venta.
4. Las marcas son las encargadas de dar una autorización a la transacción correspondiente o declinarla según sea el caso.

Fases para liberación

Para poder liberar a producción, es necesario pasar por las siguientes etapas:

➤ Pruebas Unitarias e Integrales

- Se requiere que el medio principal de comunicación esté instalado y configurado.
- Se deben programar ventanas de prueba con BBVA.
- Tiempo máximo de 10 días en sesiones de 2 horas diarias.

➤ Certificación

- En conferencia se procesa la matriz de pruebas proporcionada por BBVA.
- Se deben programar ventanas de prueba.
- Tiempo máximo de dos días en sesiones de 2 horas diarias.
- En caso de ser exitosas las pruebas se programará la entrada en producción.

➤ **Salida a producción**

- Entrega por parte de E-Global el usuario y contraseña para el acceso al portal de generación de credenciales de autenticación.
- Concluida la etapa del PILOTO se procederá a la expansión de sucursales o terminales, según sea el caso.
- Entrega de matriz de escalamiento para la atención de reportes a problemas.

Tiempo promedio para salir a producción desde el inicio del desarrollo: 1 mes (El tiempo puede variar dependiendo de diferentes circunstancias)

Transacciones Soportadas

Funcionalidad Retail

- Venta
- Venta con cuotas
- Anulación de venta
- Venta QR
- Reversos
- Carga de llaves
- Cierre de Turno

Requerimientos de Hardware y Software

- Sistema operativo Windows en sus versiones: 11 en adelante, a 32 y 64 bits.
- CPU Pentium recomendable i3 o superior.
- NET Framework 4.8
- Aplicación **TOTAL POS SDK.net**.
- Configuración de direccionamientos.
- Desarrollo del comercio para interactuar con la aplicación **TOTAL POS SDK.net**.
- Puerto serial o USB para conexión del Pin Pad.
- Tener permisos a nivel comunicación para alcanzar al Router o acceso al internet corporativo.
- Tener permisos de escritura y lectura sobre la ruta raíz del proyecto.
- Contar con impresora para la emisión de los pagarés, incluyendo los drivers del proveedor (para Pin Pad seriales).

Estos requerimientos se deben considerar para cualquier equipo que el comercio asigne como punto de venta.

Dispositivo Pin Pad Android

Un Pin Pad o dispositivo de entrada de Pin es un dispositivo electrónico cuya finalidad es obtener el número de identificación personal (PIN) del tarjetahabiente. En la actualidad también puede ser usado para obtener la firma electrónica de algunas tarjetas de crédito, así como el código de seguridad.

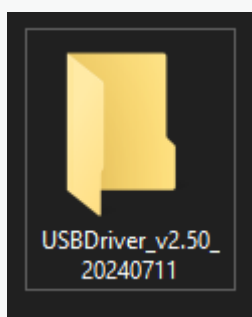
Antes de iniciar la instalación de algún dispositivo Android proporcionado por BBVA, se debe de tener en consideración lo siguiente:

- Utilice únicamente dispositivos procedentes de BBVA.
- No use controladores de dispositivos que no hayan sido proporcionados por BBVA.
- No se garantiza el funcionamiento adecuado con controladores genéricos alternativos.
- Al inicio de la instalación, NO conecte el dispositivo al equipo de cómputo, solo conéctelo al finalizar la ejecución del instalador

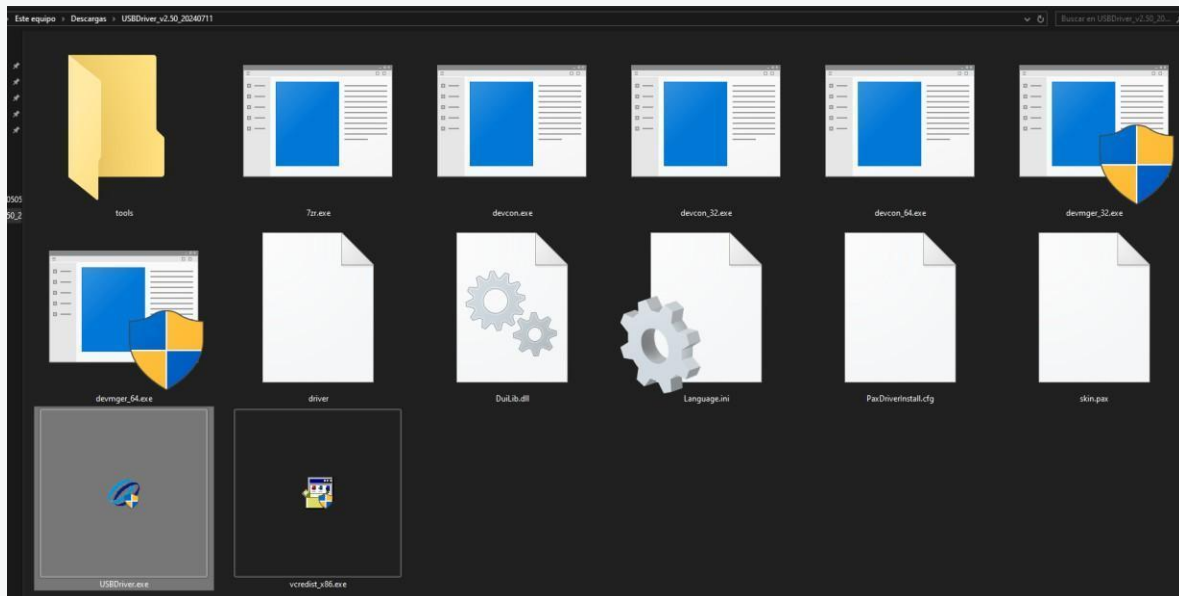
Instalación de drivers

Driver: Software esencial que permite que el sistema operativo (como Windows o macOS) se comunique correctamente con un componente de hardware específico (como una impresora, terminal pos o un dispositivo USB).

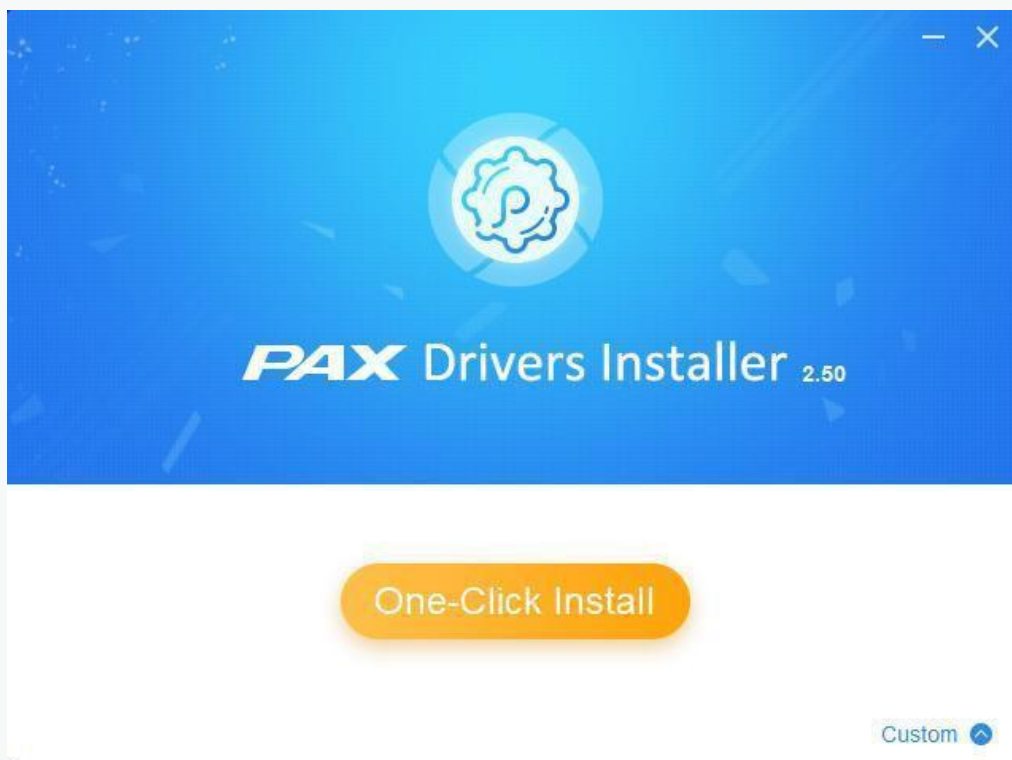
Para iniciar, es fundamental contar con el software USBDriver, ya que este programa será la herramienta principal que guiará y permitirá la correcta aplicación de este manual.



Localiza y ejecuta el archivo USBDriver.exe (doble clic) dentro de la carpeta del software. Cuando se requieran, otorga los permisos de administrador. Esto asegurará la instalación de los componentes esenciales y el reconocimiento de los dispositivos.

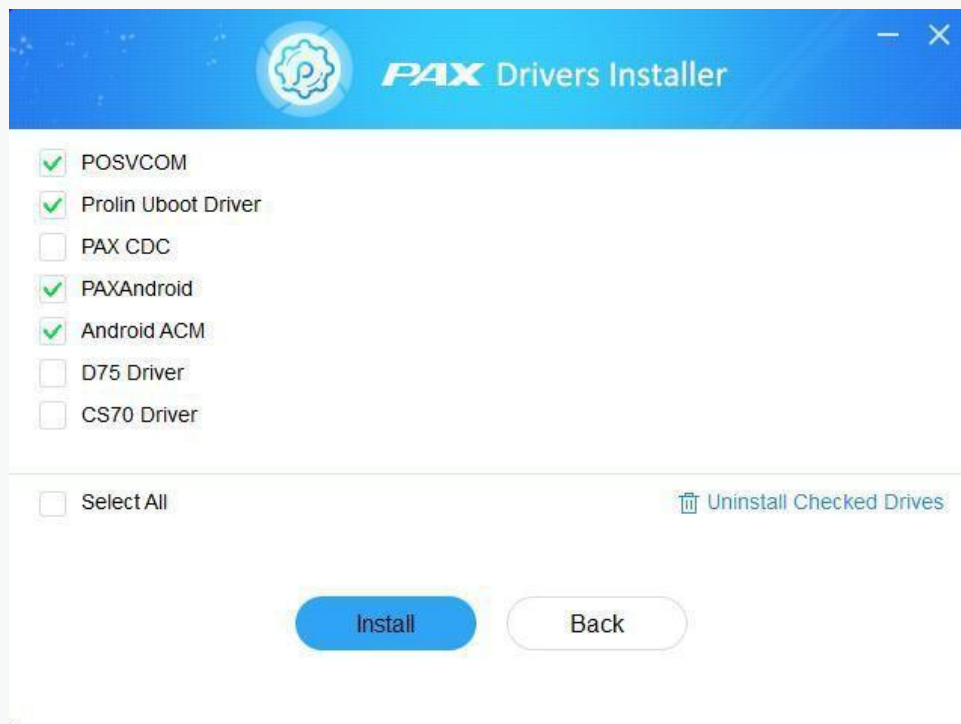


Una vez abierto el PAX Drivers Installer, visualizarás dos opciones de instalación, representadas por dos botones. El primero, de color naranja y etiquetado como "One-Click Install", realizará una instalación estándar de los drivers esenciales para la compatibilidad con los terminales PAX. El segundo botón, llamado "Custom", te brinda la posibilidad de llevar a cabo una instalación avanzada, donde podrás elegir individualmente los drivers que se instalarán en el sistema.

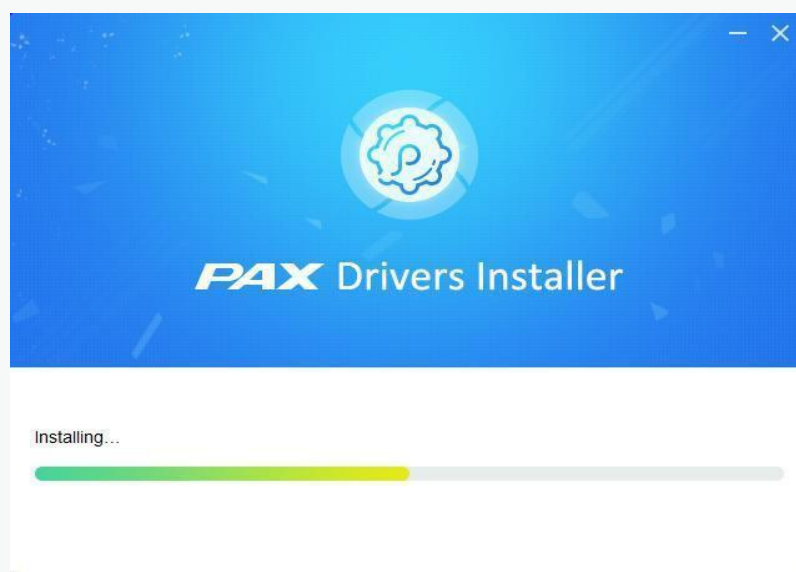


Al optar por la instalación "Custom", se mostrará una lista de drivers con selecciones predeterminadas, las cuales puedes ajustar. En la parte inferior, el botón "Install" ejecuta la

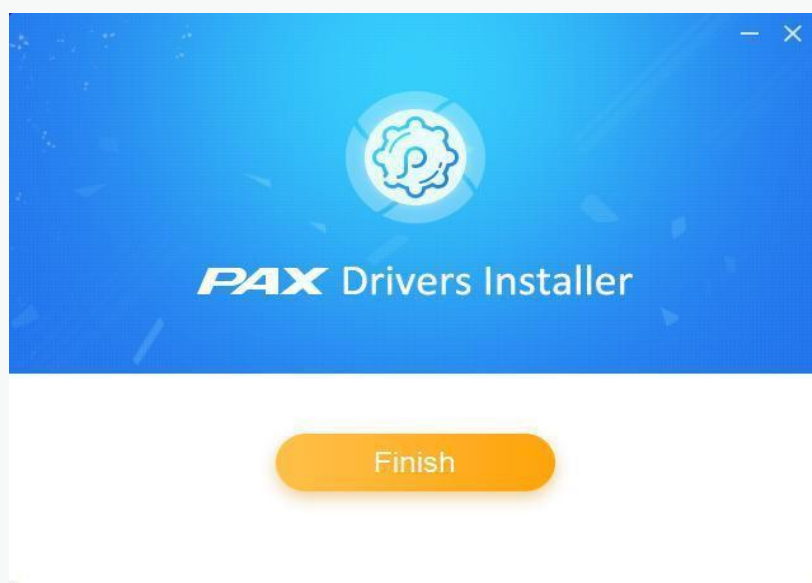
instalación de los drivers elegidos, mientras que el otro botón te devuelve a la pantalla principal. A la derecha, el botón "Uninstall Checked Drivers" realiza la desinstalación de los drivers que hayas marcado en la lista y que se encuentren instalados en el equipo.



Al presionar "Install" en cualquier modo de instalación, el software comenzará a instalar los elementos necesarios. Una barra de progreso en la parte inferior indicará el avance. Simplemente espera a que termine.



Cuando la barra de progreso indique que la instalación está completa, un botón naranja con el texto "Finish" se hará visible. Haz clic en él para dar por concluido el proceso y cerrar la ventana del software de instalación.



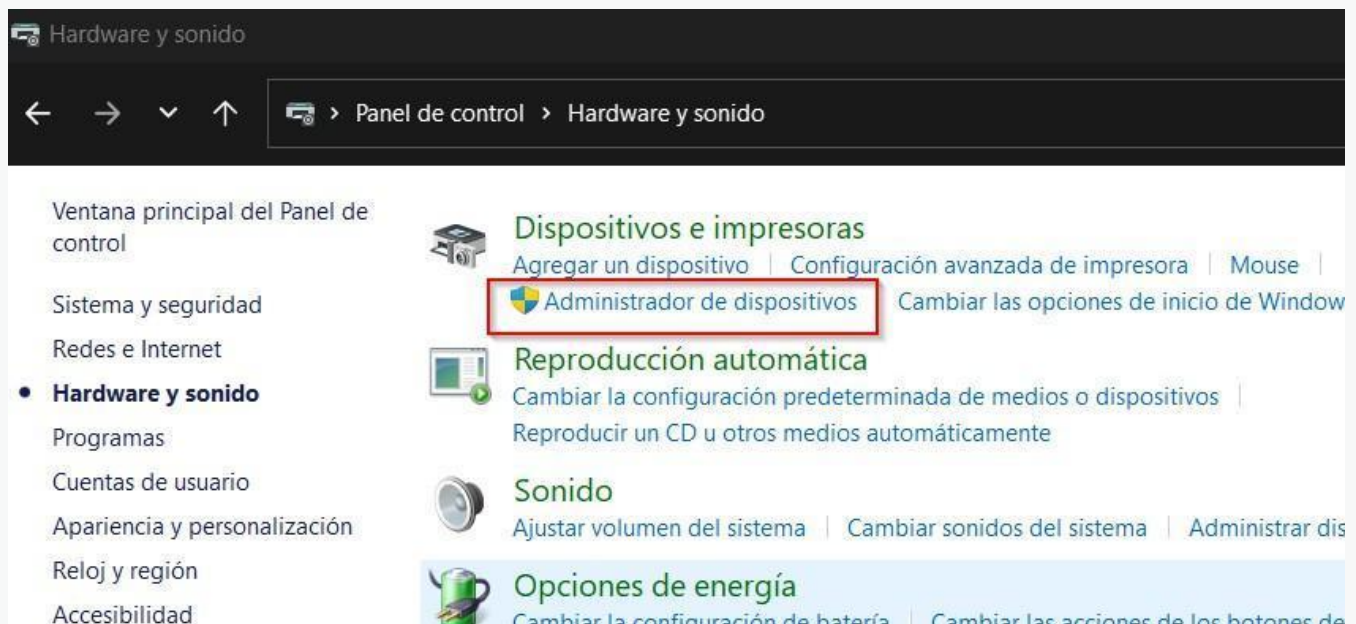
Configuración de dispositivo Android Serial (COM)



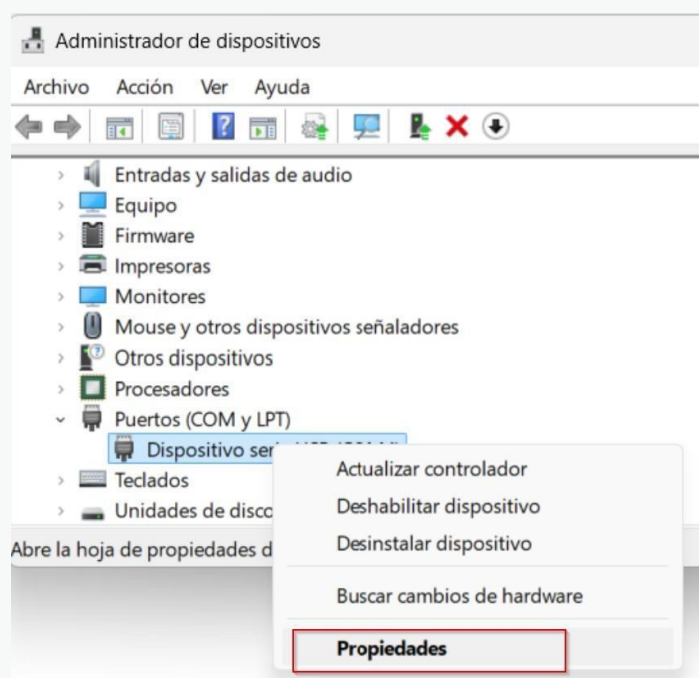
PAX A35

Una vez concluida la instalación conecte el dispositivo Pin Pad a un puerto USB disponible, es necesario que valide el puerto COM asignado por el sistema operativo en el Administrador de dispositivos. Esta consideración es debido a que en algunas circunstancias el puerto COM mostrado en la pantalla principal del administrador de dispositivos no muestra el correcto puerto COM asignado. Verifique el puerto de la siguiente manera:

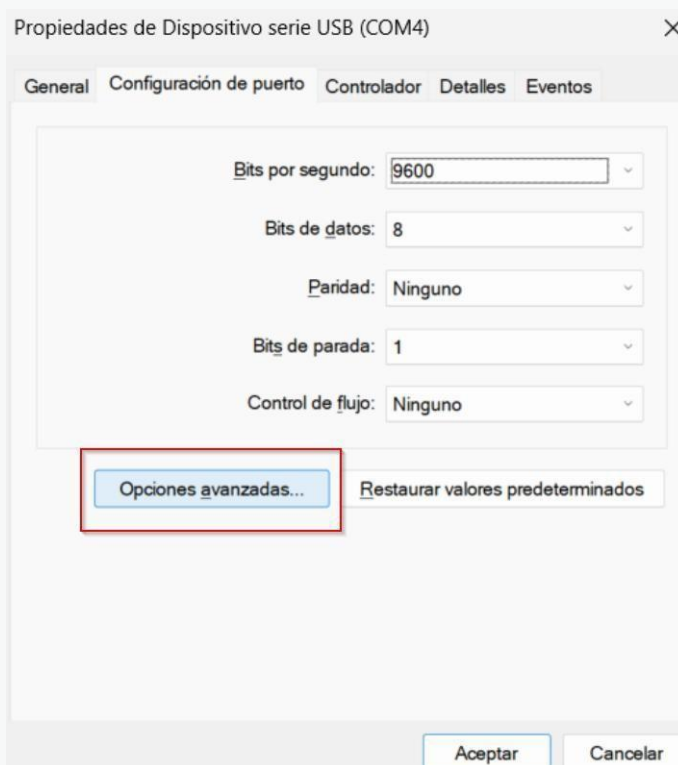
Desde el panel de control seleccione la opción de Administrador de dispositivos (en Windows)



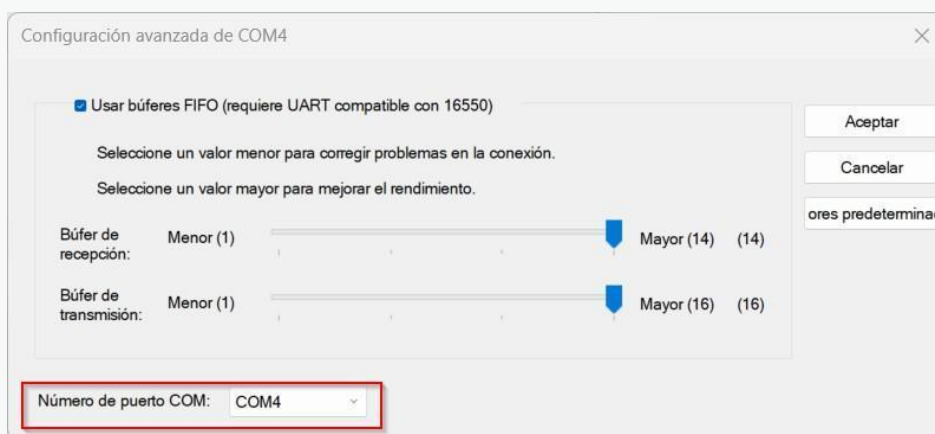
En el administrador de dispositivos señale con el botón secundario del mouse el dispositivo Pin Pad recién conectado, en la opción “puertos COM” y seleccione la opción propiedades del menú emergente.



Seleccione la pestaña configuración de puerto y opciones avanzadas...



En la ventana Configuración avanzada de COM vemos el puerto real que el sistema operativo esta asignando al dispositivo Pin Pad.



Su personal de TI (Sistemas) puede modificar estos valores de acuerdo con las necesidades o disponibilidad de los puertos de sus equipos. Lo más importante es que debe tener conocimiento del puerto real interno ya que este dato será necesario en la configuración del componente SDK.net [Conectividad por puerto COM].

Configuración de dispositivo Android Inalámbrico (Wifi)



PAX A35

Existen algunos Pin Pad's que pueden operar mediante conectividad WIFI, es decir, se comunicarán con el desarrollo del comercio a través de su red inalámbrica en lugar del cable USB.

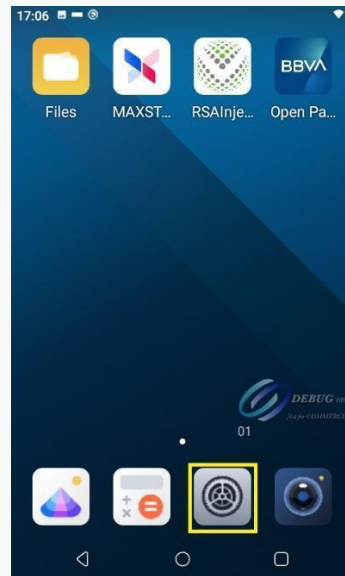
Nota 1: Solo para los dispositivos que entregue BBVA que soporten la conectividad Wifi deberán ser configurados mediante los pasos citados más adelante, en caso de que la conectividad sea mediante USB no podrá conectar los dispositivos mediante Wifi.

Nota 2: Debe conectar los dispositivos a la misma red que se encuentren conectados los puntos de venta o garantizar que se puede realizar una comunicación entre la caja y el Pin Pad a través de la red del comercio.

Nota 3: Sólo debe asociar un Punto de Venta a un Pin Pad inalámbrico, de lo contrario no se podrá garantizar el correcto funcionamiento del dispositivo lector de tarjetas.

Pin Pad Pax A35

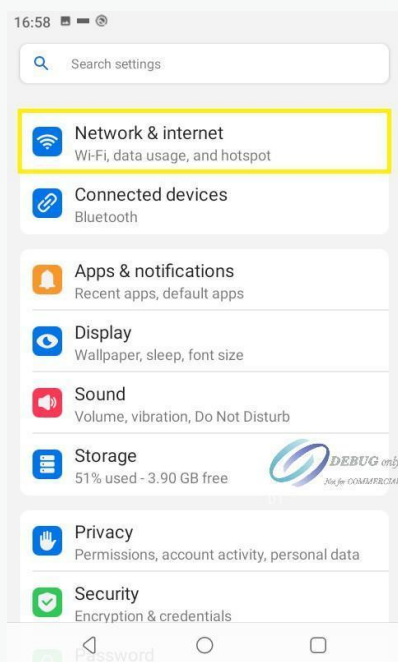
1. Para esta marca el modelo que se utiliza para transaccionar con la aplicación es el A35, para iniciar el pinpad conecte el dispositivo al puerto USB de su PC.
2. Una vez que termine de cargar observará la pantalla principal, en ella dar clic en el icono del engrane.



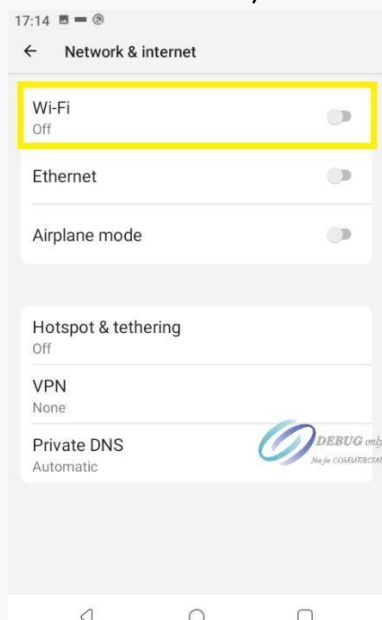
3. Introducir la clave (9876 o pax9876@@) y presionar el botón verde en el dispositivo.



4. Seleccionar la opción “Network & Internet”



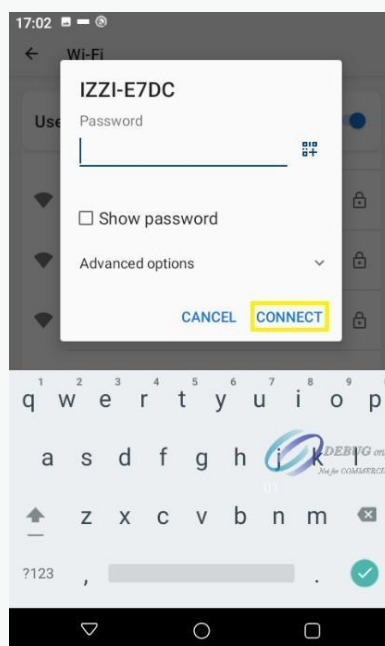
5. Seleccionar la opción “Wi-Fi” (en caso de que no esté habilitada dar clic en el botón switch del que se encuentra del lado derecho).



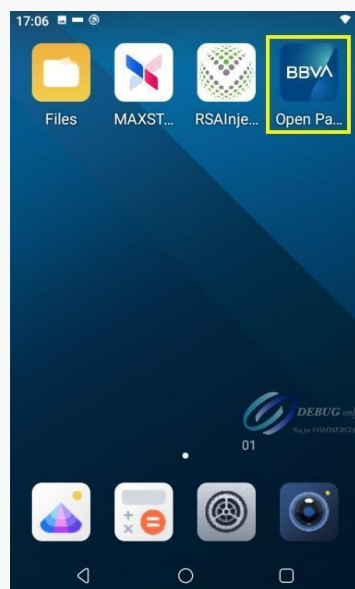
6. Seleccionar la red a la que se desea conectar.



7. Ingresar los datos de conexión y presionar la opción “CONNECT”.



8. Regresar a la pantalla principal e iniciar la aplicación dando clic en el icono con la imagen de BBVA.



9. Si se realizó correctamente la configuración, observará, en la esquina inferior derecha, la IP y puerto asignados al dispositivo, mismos que deberá indicar en la configuración del SDK.



Una vez conectado el Pin Pad a la red del comercio y haber obtenido la IP asignada a éste, deberá introducir los siguientes parámetros en el objeto Configuración que se describe más adelante:

- **Configuracion.PinPadConexion:** Deberá introducir el valor de la IP asignada para indicar que se utilizará un Pin Pad Wifi.
- **Configuracion.PinPadPuerto:** Generalmente es el 5000 pero puede variar si el puerto ya está siendo utilizado por el dispositivo.

Implementación

Objeto Configuración e inicialización

Es necesario realizar una configuración e inicialización del SDK previo al envío de transacciones, preferentemente al iniciar el punto de venta. Para ello se instancia un objeto de la clase **Configuracion** y se brindan los parámetros necesarios para la inicialización los cuales se enlistan a continuación:

Parámetros

- **Logs:** Nos indica si se tiene acceso a la descarga de un archivo de texto que permite conocer la comunicación y las acciones realizadas entre el Pin Pad y el Host autorizador.
- **PinPadConexion:** Es el puerto de conexión con el dispositivo Pin Pad:
 - Ejemplo: COM3, para dispositivos seriales
 - Ejemplo: 192.168.1.1 para dispositivos Wifi.
- **PinPadPuerto:** Indica el puerto en que el dispositivo Pin Pad escuchará los comandos (cuando la conexión sea por Wifi, si la conexión es serial se ignorará este valor).
- **PinpadTimeout:** Es el tiempo en segundos que el Pin Pad espera para realizar la lectura de una tarjeta antes de declinar la transacción.
- **PinpadAndroid:** Indica si el dispositivo Pin Pad cuenta con el sistema operativo Android.
- **RutaBase:** Indica la ruta donde se crearán los archivos que necesita el SDK para su operación. Por default la propiedad tiene el valor de vacío ("") y los archivos se generarán en la misma ruta del desarrollo que implemente el componente.
- **HostUrl:** Es la dirección URL del Host Autorizador donde se conectará TotalPOS SDK.
- **HostTimeout:** Es el tiempo máximo en segundos que se espera para recibir la respuesta de una transacción por parte del Host Autorizador. Por default se recomiendan 40 segundos.
- **UrlProxy:** Url del servidor proxy que se utilizará para la comunicación.
- **PuertoProxy:** Puerto del servidor proxy que se utilizará para la comunicación.
- **Afiliacion:** Este es el número afiliación default utilizado para enviar transacciones.
- **MonedaAfiliacion:** Moneda de la afiliación.

- **IdAplicacion:** Client-id para realizar la autenticación con el Host Autorizador.
- **ClaveSecreta:** Client-secret para realizar la autenticación con el Host Autorizador.
- **NumeroTerminal:** identificador lógico de la terminal del comercio (1-99, solo números) utilizado para identificar la terminal que realiza la operación.

Una vez establecidos los parámetros se asigna el objeto configuración a la instancia de Interfaz de la siguiente forma:

Interfaz.Instance.Configuracion = configuracion;

Finalmente se inicializa el SDK mediante la siguiente instrucción:

Interfaz.Instance.Inicializar();

Es importante que se inicialice el SDK debido a que se inician los procesos en donde se recibe información acerca de la carga de llaves y dispara los reversos.

Ejemplo de configuración e inicialización en Windows (Pinpad Android USB)

```
Configuracion configuracion;

try
{
    configuracion = new Configuracion
    {
        Logs = false,
        PinPadConexion = "COM4",
        PinPadPuerto = "",
        PinPadTimeout = "40",
        PinPadMensaje = "",
        PinPadAndroid = true,
        RutaBase = "",
        HostUrl = "http://localhost:8080",
        HostTimeout = "30",

        // Configuración adicional de Proxy
        UrlProxy = "192.0.0.103",
        PuertoProxy = "779",

        Afiliacion = "79791285",
        MonedaAfiliacion = Moneda.Soles,
        NumeroTerminal = "1"
        IdAplicacion = "8c678668c897df37c2ebe29d03cc",
        ClaveSecreta = "5921-e11c-4107-9813-4a9e72c0"
    };

    Interfaz.Instance.Configuracion = configuracion;
    Interfaz.Instance.Inicializar();
}
catch (Exception)
{
    throw;
```



Por default tendrán valor **false** las siguientes propiedades, en caso de no asignar un valor:

- Logs

Ejemplo de configuración e inicialización en Windows (Pinpad Android Wifi)

```
Configuracion configuracion;

try
{
    configuracion = new Configuracion
    {
        Logs = false,
        PinPadConexion = "192.168.0.1",
        PinPadPuerto = "5000",
        PinPadTimeOut = "40",
        PinPadMensaje = "",
        PinPadAndroid = true,
        RutaBase = "",
        HostUrl = "http://localhost:8080",
        HostTimeOut = "30",

        // Configuración adicional de Proxy
        UrlProxy = "192.0.0.103",
        PuertoProxy = "779",

        Afiliacion = "79791285",
        MonedaAfiliacion = Moneda.Soles,
        NumeroTerminal = "1",
        IdAplicacion = "8c678668c897df37c2ebe29d03cc",
        ClaveSecreta = "5921-e11c-4107-9813-4a9e72c0",
    };

    Interfaz.Instance.Configuracion = configuracion;
    Interfaz.Instance.Inicializar();
}
catch (Exception)
{
    throw;
}
```



Por default tendrán valor **false** las siguientes propiedades, en caso de no asignar un valor:

- Logs

Operaciones Retail

A continuación, se enlistan los métodos necesarios para realizar las transacciones de la funcionalidad Retail, los cuales están definidos dentro de una clase especial del tipo enum. Los

tipos enum sirven para restringir el contenido de una variable a una serie de valores predefinidos.

- **OPERACION:** Objeto de tipo enum que contendrá las variables definidas relacionadas a los tipos de operación, los cuales se describen a continuación.
 - ✓ **Venta:** Es la petición que realiza el comercio al host solicitando la autorización por el importe de la mercancía o servicio solicitando la tarjeta para su autorización.
 - ✓ **VentaQR:** Es la petición que realiza el comercio al host solicitando la autorización por el importe de la mercancía o servicio solicitando la lectura de un código QR para su autorización.
 - ✓ **AnulacionVentaQR:** Es la petición que realiza el comercio al host autorizador para que se cancele el movimiento QR y se le restituya el importe.
 - ✓ **AnulacionVentaTarjeta:** Es la petición que realiza el comercio al host autorizador para que se cancele el movimiento y se le restituya el importe. A diferencia de la transacción anterior se solicita la tarjeta para realizar la operación.
 - ✓ **CargaLlaves:** Esta operación es obligatoria ya que es de configuración para el Pin Pad, se debe realizar cada vez que el Pin Pad es nuevo y antes de enviar cualquier transacción, tiene tres modalidades las cuales se explican en el apartado de carga de llaves.

Los parámetros establecidos en cada operación son manejados de forma interna, de modo que ya están preestablecidos por el host autorizador.

Parámetros de operación

- **ParametroOperacion:** Objeto de tipo enum que contendrá las variables definidas de manera interna con relación a los parámetros que tendrán las operaciones, dependiendo también el tipo de acción.
 - ✓ Importe
 - ✓ Propina
 - ✓ ReferenciaFinanciera
- **Importe:** Importe total de la venta.
- **Propina:** Importe de propina para una venta.
- **ReferenciaFinanciera:** Identificador único para una transacción realizada.

Venta

```
Peticion p;
Respuesta r;
Dictionary<ParametroOperacion, object> parametros;

try
{
    parametros = new Dictionary<ParametroOperacion, object>
    {
        { ParametroOperacion.Importe, "100.00" },
        { ParametroOperacion.Propina, "10.00" } //Opcional
    };

    p = new Peticion();
    p.Operador = "OPE001";
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");
    p.SetOperacion(Operacion.Venta, parametros);
    (1) p.CambioMetodoPago += Venta_CambioMetodoPago;

    r = p.Autorizar();

    foreach (PropertyInfo pi in r.GetType().GetProperties())
    {
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));
    }

    p.CambioMetodoPago -= Venta_CambioMetodoPago;
}
catch (PinPadKeysException clEx)
{
    Console.WriteLine("Error: " + clEx.Message);
}
catch (PeticionException pEx)
{
    Console.WriteLine("Error: " + pEx.Message);
}
```



(1) La propiedad CambioMetodoPago permite agregar un evento para saber cuándo la operativa cambia a Operacion.VentaQR después de seleccionar la opción desde la pantalla del PinPad. En este evento se deberá implementar la VentaQR.

Evento cuando se selecciona pago con QR

```
private Respuesta Venta_CambioMetodoPago()
{
    Peticion p;
    Dictionary<ParametroOperacion, object> parametros;
    Respuesta r;

    try
    {
        parametros = new Dictionary<ParametroOperacion, object>
        {
            { ParametroOperacion.Importe, "53.00" },
            { ParametroOperacion.Propina, "0.00" } //Opcional
        };

        p = new Peticion();
        p.Operador = "OPE001";
        p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");
        p.SetOperacion(Operacion.VentaQR, parametros);
        p.GeneraQR();
        r = p.FinalizarVentaQR();

        return r;
    }
    catch (PinPadKeysException)
    {
        throw;
    }
    catch (PeticionException)
    {
        throw;
    }
}
```

Venta QR

```
Peticion p;  
Dictionary<ParametroOperacion, object> parametros;  
Respuesta r;  
  
try  
{  
    parametros = new Dictionary<ParametroOperacion, object>  
    {  
        { ParametroOperacion.Importe, "53.00" },  
        { ParametroOperacion.Propina, "0.00" } //Opcional  
    };  
  
    p = new Peticion();  
    p.Operador = "OPE001";  
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");  
    p.SetOperacion(Operacion.VentaQR, parametros);  
    p.GeneraQR();  
    r = p.FinalizarVentaQR();  
  
    foreach (PropertyInfo pi in r.GetType().GetProperties())  
    {  
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));  
    }  
}  
catch (PinPadKeysException clEx)  
{  
    Console.WriteLine("Error: " + clEx.Message);  
}  
catch (PeticionException pEx)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

NOTA: La llamada SDK para esperar la confirmación de pago QR no debe bloquear el hilo de la interfaz de usuario (UI Thread), por lo que se requiere estrictamente el uso de procesos asíncronos en .NET (como Task.Run() o async/await).

Flujo de Cancelación y Expiración en Pagos con QR

Para garantizar una experiencia de usuario fluida, el sistema debe contemplar dos escenarios de interrupción de la venta:

- **Expiración de Tiempo en pantalla:** El proceso de espera en el POS cuenta con una vigencia de 120 segundos, que inicia en el momento en que el código se despliega en pantalla. Si el pago no se confirma en ese intervalo, la sesión de espera en la terminal finalizará y retornará un código de respuesta “99” indicando “ESTADO INDETERMINADO, CONSULTAR ESTADO”.
- **Cancelación Manual (Interfaz de Usuario):** Es responsabilidad del comercio implementar un botón de “*Cancelar*” visible en la interfaz del punto de venta (POS) mientras se muestra el código QR. Si el cliente decide cambiar su método de pago o desistir de la compra, el operador debe poder activar esta función. Al hacerlo, el sistema debe invocar el método correspondiente del SDK para liberar la transacción y permitir que el POS regrese a su estado inicial de forma inmediata.

Ejemplo de implementación:

```
private void BtnCancel_Click(object sender, EventArgs e) {  
    p.CancelarVentaQr();  
}
```

Venta PSI

```

Peticion p;
Tarjeta t;
Respuesta r;
Dictionary<ParametroOperacion, object> parametros;

try
{
    parametros = new Dictionary<ParametroOperacion, object>
    {
        { ParametroOperacion.Importe, "120.20" }
    };

    p = new Peticion();
    p.Operador = "OPE001";
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");
    p.SetOperacion(Operacion.Venta, parametros);
    (1) p.CambioMetodoPago += Venta_CambioMetodoPago;

    t = p.LeerTarjeta();

    (2)(3) if (t.Producto == "C" && t.MaxCuotasPsi > 0)
    {
        p.SetPromocionMeses(Promocion.MesesSinIntereses, 12);
    }

    r = p.Autorizar();

    foreach (PropertyInfo pi in r.GetType().GetProperties())
    {
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));
    }

    p.CambioMetodoPago -= Venta_CambioMetodoPago;
}
catch (PinPadKeysException clEx)
{
    Console.WriteLine("Error: " + clEx.Message);
}
catch (PeticionException pEx)
{
    Console.WriteLine("Error: " + pEx.Message);
}

```



- (1) La propiedad CambioMetodoPago permite agregar un evento para saber cuándo la operativa cambia a Operacion.VentaQR después de seleccionar la opción desde la pantalla del PinPad. En este evento se deberá implementar la VentaQR.
- (2) Si el método **MaxCuotasPsi** es mayor a cero significa que se le puede ofrecer al tarjetahabiente la promoción de pagos sin intereses.
- (3) Para dispositivos Android, la validación de cuotas se hace internamente y la selección de estas se realiza directamente en el dispositivo.

Venta con cuotas

```
Tarjeta t;
Respuesta r;
Dictionary<ParametroOperacion, object> parametros;

try
{
    parametros = new Dictionary<ParametroOperacion, object>
    {
        { ParametroOperacion.Importe, "120.20" }
    };

    Peticion p = new Peticion();
    p.Operador = "OPE001";
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");
    p.SetOperacion(Operacion.Venta, parametros);
    (1) p.CambioMetodoPago += Venta_CambioMetodoPago;

    t = p.LeerTarjeta();

    (2) if (t.CuotasPci)
    {
        p.SetPromocionMeses(Promocion.MesesConIntereses, 12);
    }

    r = p.Autorizar();

    foreach (PropertyInfo pi in r.GetType().GetProperties())
    {
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));
    }

    p.CambioMetodoPago -= Venta_CambioMetodoPago;
}
catch (PinPadKeysException clEx)
{
    Console.WriteLine("Error: " + clEx.Message);
}
catch (PeticionException pEx)
{
    Console.WriteLine("Error: " + pEx.Message);
}
```



- (1) La propiedad CambioMetodoPago permite agregar un evento para saber cuándo la operativa cambia a Operacion.VentaQR después de seleccionar la opción desde la pantalla del PinPad. En este evento se deberá implementar la VentaQR.
- (2) Para dispositivos Android, la validación de cuotas se hace internamente y la selección de estas se realiza directamente en el dispositivo.

Anulación de Venta QR

```
Peticion p;  
Dictionary<ParametroOperacion, object> parametros;  
Respuesta r;  
  
try  
{  
    parametros = new Dictionary<ParametroOperacion, object>  
    {  
        { ParametroOperacion.Importe, "53.00" }  
        { ParametroOperacion.ReferenciaFinanciera, "123456789012" }  
    };  
  
    Peticion p = new Peticion();  
    p.Operador = "OPE001";  
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");  
    p.SetOperacion(Operacion.AnulacionVentaQR, parametros);  
  
    r = p.Autorizar();  
  
    foreach (PropertyInfo pi in r.GetType().GetProperties())  
    {  
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));  
    }  
}  
catch (PinPadKeysException clEx)  
{  
    Console.WriteLine("Error: " + clEx.Message);  
}  
catch (PeticionException pEx)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Anulación de Venta

```
Peticion p;  
Dictionary<ParametroOperacion, object> parametros;  
Respuesta r;  
  
try  
{  
    parametros = new Dictionary<ParametroOperacion, object>  
    {  
        { ParametroOperacion.Importe, "100.00" },  
        { ParametroOperacion.ReferenciaFinanciera, "123456789012" }  
    };  
  
    p = new Peticion();  
    p.Operador = "OPE001";  
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");  
    p.SetOperacion(Operacion.AnulacionVentaTarjeta, parametros);  
  
    r = p.Autorizar();  
  
    foreach (PropertyInfo pi in r.GetType().GetProperties())  
    {  
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));  
    }  
}  
catch (PinPadKeysException clEx)  
{  
    Console.WriteLine("Error: " + clEx.Message);  
}  
catch (PeticionException pEx)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Consulta De Transacción

```
Peticion p;  
Respuesta r;  
  
try  
{  
    p = new Peticion();  
    r = p.ConsultarTransaccion("f3b29b92-25b2-4527-9b5d-d99ed19b138f");  
  
    foreach (PropertyInfo pi in r.GetType().GetProperties())  
    {  
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));  
    }  
}  
catch (PeticionException pEx)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Operaciones Pre-autorización

A continuación, se enlistan los métodos necesarios para realizar las transacciones de la funcionalidad Pre-autorización, los cuales están definidos dentro de una clase especial del tipo enum. Los tipos enum sirven para restringir el contenido de una variable a una serie de valores predefinidos.

- **OPERACION:** Objeto de tipo enum que contendrá las variables definidas relacionadas a los tipos de operación, los cuales se describen a continuación:
 - ✓ **Preautorizacion:** Es la petición que realiza el comercio al host para apartar una cantidad del saldo disponible de su cliente para asegurar que la tarjeta cuenta con lo necesario para cubrir con los productos o servicios que desea adquirir. Esta operación es utilizada para garantizar las reservaciones; no es una operación contable, no se hace el cobro solo se retiene la cantidad solicitada.
 - ✓ **ConfirmacionPreautorizacion:** Es la petición que realiza el comercio al host autorizador para cerrar o "liberar" la pre-autorización, ya sea enviando el mismo importe, importe menor o mayor al de la pre-autorización.
 - ✓ **AnulacionPreautorizacion:** Es la petición que realiza el comercio al host autorizador para que se cancele el movimiento y se le restituya el importe.

- ✓ **ConsultarPreautorizaciones:** Es la petición que realiza el comercio al host autorizador para que obtenga el listado de pre-autorizaciones pendientes de confirmación.

Los parámetros establecidos en cada operación son manejados de forma interna, de modo que ya están preestablecidos por el host autorizador.

Parámetros de operación

- **ParametroOperacion:** Objeto de tipo enum que contendrá las variables definidas de manera interna con relación a los parámetros que tendrán las operaciones, dependiendo también el tipo de acción.
 - ✓ **Importe**
 - ✓ **ConfirmacionImporte**
 - ✓ **Folio**
 - ✓ **ReferenciaFinanciera**
- **Importe:** Importe total de la pre-autorización.
- **ConfirmacionImporte:** Importe total por el cual se hará la confirmación de Pre-autorización.
- **Folio:** Dato alfanumérico de siete posiciones con el cual se asociará la tarjeta del tarjetahabiente mientras dure la retención. Por lo general se utiliza para el número de reservación.
- **ReferenciaFinanciera:** Identificador único para una transacción realizada.

Pre-autorización

```
Peticion p;  
Respuesta r;  
Dictionary<ParametroOperacion, object> parametros;  
  
try  
{  
    parametros = new Dictionary<ParametroOperacion, object>  
    {  
        { ParametroOperacion.Importe, "135.00" },  
        { ParametroOperacion.Folio, "0000000" }  
    };  
  
    p = new Peticion();  
    p.Operador = "OPE001";  
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");  
    p.SetOperacion(Operacion.Preautorizacion, parametros);  
  
    r = p.Autorizar();  
  
    foreach (PropertyInfo pi in r.GetType().GetProperties())  
    {  
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));  
    }  
}  
catch (PinPadKeysException clEx)  
{  
    Console.WriteLine("Error: " + clEx.Message);  
}  
catch (PeticionException pEx)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Confirmación Pre-autorización

```
Peticion p;  
Respuesta r;  
Dictionary<ParametroOperacion, object> parametros;  
  
try  
{  
    parametros = new Dictionary<ParametroOperacion, object>  
    {  
        { ParametroOperacion.Importe, "135.00" },  
        { ParametroOperacion.ConfirmacionImporte, "135.00" },  
        { ParametroOperacion.Folio, "0000000" }  
    };  
  
    p = new Peticion();  
    p.Operador = "OPE001";  
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");  
    p.SetOperacion(Operacion.ConfirmacionPreautorizacion, parametros);  
  
    r = p.Autorizar();  
  
    foreach (PropertyInfo pi in r.GetType().GetProperties())  
    {  
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));  
    }  
}  
catch (PinPadKeysException clEx)  
{  
    Console.WriteLine("Error: " + clEx.Message);  
}  
catch (PeticionException pEx)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Anulación Pre-autorización

```
Peticion p;  
Dictionary<ParametroOperacion, object> parametros;  
Respuesta r;  
  
try  
{  
    parametros = new Dictionary<ParametroOperacion, object>  
    {  
        { ParametroOperacion.Importe, "100.00" },  
        { ParametroOperacion.ReferenciaFinanciera, "123456789012" }  
    };  
  
    p = new Peticion();  
    p.Operador = "OPE001";  
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");  
    p.SetOperacion(Operacion.AnulacionPreautorizacion, parametros);  
  
    r = p.Autorizar();  
  
    foreach (PropertyInfo pi in r.GetType().GetProperties())  
    {  
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));  
    }  
}  
catch (PinPadKeysException clEx)  
{  
    Console.WriteLine("Error: " + clEx.Message);  
}  
catch (PeticionException pEx)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Consulta de Pre-autorizaciones

```
Peticion p;  
RespuestaPreautorizacion rp;  
  
try  
{  
    p = new Peticion();  
    rp = p. ConsultarPreautorizaciones ("20250912000000", "20250912235959");  
  
    if (rp.Preautorizaciones.Count > 0)  
    {  
        foreach (Respuesta r in rp.Preautorizaciones)  
        {  
            Console.WriteLine("Fecha: {0}\\tTipo Operación: {1}\\tTarjeta:  
{2}\\tFolio: {3}\\tImporte: {4}", r.FechaHora, r.NombreTransaccion,  
r.NumeroTarjeta, r.Folio, r.Importe);  
        }  
    }  
}  
catch (PeticionException pEx)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Operaciones Cierre de Turno

A continuación, se enlistan los métodos necesarios para ejecutar la operación Cierre de Turno en el SDK. Esta Operación permite cerrar el turno actual del comercio.

- **OPERACION:** Objeto de tipo enum que contiene los tipos de operación disponibles en el SDK
- **Cierre de turno:** Ejecuta el cierre de turno actual en el Host

Parámetros de operación

- **ParametroOperacion:** Para la operacion **CierreTurno** no se requieren parámetros en el diccionario (Dictionary<ParametroOperacion>, object). Los datos necesarios se obtienen/configuran de forma interna (por ejemplo: terminal lógica configurada, serie física del pinpad, etc)

Cierre de Turno (Ejemplo)

```
Peticion p;  
Respuesta r;  
  
try  
{  
    p = new Peticion();  
    p.Operador = "OPE001";  
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");  
  
    // No requiere parámetros  
    p.SetOperacion(Operacion.CierreTurno, null);  
  
    r = p.Autorizar();  
  
    foreach (PropertyInfo pi in r.GetType().GetProperties())  
    {  
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));  
    }  
  
    // Campos típicos de salida  
    Console.WriteLine("CodigoRespuesta: " + r.CodigoRespuesta);  
    Console.WriteLine("Leyenda: " + r.Leyenda);  
    Console.WriteLine("IdTurno: " + r.IdTurno);  
    Console.WriteLine("NumeroTerminal: " + r.NumeroTerminal);  
}  
catch (PinPadKeysException ex)  
{  
    Console.WriteLine("Error: " + ex.Message);  
}  
catch (PeticionException ex)  
{  
    Console.WriteLine("Error: " + ex.Message);  
}  
}
```

Respuesta de la operación

La operación **CierreTurno** devuelve un objeto **Respuesta** con información como:

- **CodigoRespuesta:** (por ejemplo "00" cuando fue aprobada)
- **Leyenda:** (mensaje del host)
- **IdTurno:** (identificador del turno cerrado)
- **NumeroTerminal:** (terminal lógica reportada)

Manejo de errores

- Si no hay turno abierto, el Host puede responder con un error de negocio (ej. “No puedes hacer el cierre”).
- Errores de llaves/pinpad se reportan como PinpadKeysException.
- Errores generales del SDK/Host se reportan como PeticionException.

Consideraciones

- El cierre de turno solo aplica si existe un turno abierto.
- La terminal lógica (NumeroTerminal) debe estar configurada correctamente en la configuración del SDK.
- El pinpad debe estar disponible para obtener datos internos requeridos por el flujo (según configuración y entorno).

Carga de llaves

Existen 3 escenarios a considerar para generar la carga de llaves los cuales se detallan a continuación, se deben desarrollar los tres escenarios.

1. **Carga de Llave Manual:** Esta transacción es para cargar la llave que el Pin Pad utiliza para cifrar la información sensible de las tarjetas y sin la cual el sistema no puede hacer cobros con tarjetas, esta carga de llave será generada por el comercio mediante un mensaje que más adelante se detalla al igual que su flujo. Esta operación sólo se deberá realizar cuando se configura el equipo por primera vez y cuando la interfaz lo indique.
2. **Carga de Llaves al inicializar:** Esta operación se realiza cuando el host autorizador determina que es necesario mandar una carga de llaves automática la próxima vez que se inicialice el SDK de Total-POS. Por lo que se almacena una bandera de forma interna y cuando se manda a llamar el método inicializar de la instancia Interfaz de forma automática se realiza la carga de llaves.
3. **Carga de Llaves en Automático:** Esta operación la realiza la interfaz en automático cuando el host autorizador indica que hay que realizar carga de llaves, la interfaz regresa la respuesta de la transacción que solicitó para que el punto de venta imprima el voucher y de forma simultánea la interfaz también está realizando la carga de llaves, el resultado de esta operación se refleja en la pantalla del Pin Pad.

Carga de llaves manual

```
Respuesta r;  
  
try  
{  
    Peticion p = new Peticion();  
    p.Operador = "OPE001";  
    p.Fecha = DateTime.Now.ToString("yyyyMMddHHmmss");  
    p.SetOperacion(Operacion.CargaLlaves, null);  
  
    r = p.Autorizar();  
  
    foreach (PropertyInfo pi in r.GetType().GetProperties())  
    {  
        Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(r, null));  
    }  
}  
catch (PinPadKeysException clEx)  
{  
    Console.WriteLine("Error: " + clEx.Message);  
}  
catch (PeticionException pEx)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Reversos

Cuando se envía una transacción y no se logra recuperar la respuesta del servidor, la interfaz genera un reverso y lo guarda en una base de datos local de forma cifrada, dicho reverso es enviado al servidor para no generar un doble cargo por parte de la interfaz de forma automática, Esto se logra únicamente llamando al método de inicialización de TOTAL POS SDK.

Interfaz.Instance.Inicializar();

Con esta instrucción se permite disparar los reversos tal y como se especifica en el apartado “Objeto Configuración”.

Reportes

Es necesario realizar una configuración e inicialización del SDK previo a la generación de los reportes, preferentemente al iniciar el punto de venta. Para ello se instancia un objeto de la

clase “**Configuracion**” y se brindan los parámetros necesarios para la inicialización (ver apartado “**Objeto Configuración e inicialización**”).

Reporte Firma Física

Para generar el reporte de firma física se deberán proporcionar los siguientes parámetros a la función:

- ✓ **Moneda:** Indica la moneda de la cual se desea obtener el reporte (Soles o Dólares).

```
RespuestaReporteFirma rrf;  
  
try  
{  
    rrf = Reporteria.GenerarReporteFirma(Moneda.Soles);  
  
    if (rrf != null)  
    {  
        foreach (PropertyInfo pi in rrf.GetType().GetProperties())  
        {  
            Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(respuesta, null));  
        }  
    }  
}  
catch (PeticiónException ex)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Reporte Firma Totalizado

Para generar el reporte de totalizado se deberán proporcionar los siguientes parámetros a la función:

- ✓ **Moneda:** Indica la moneda de la cual se desea obtener el reporte (Soles o Dólares).

```
RespuestaReporteTotalizado rrt;  
  
try  
{  
    rrt = Reporteria.GenerarReporteTotalizado(Moneda.Soles);  
  
    if (rrt != null)  
    {  
        foreach (PropertyInfo pi in rrt.GetType().GetProperties())  
        {  
            Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(respuesta, null));  
        }  
    }  
}  
catch (PeticiónException ex)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Reporte Firma Detallado

Para generar el reporte de detallado se deberán proporcionar los siguientes parámetros a la función:

- ✓ **Moneda:** Indica la moneda de la cual se desea obtener el reporte (Soles o Dólares).

```
RespuestaReporteDetallado rrd;  
  
try  
{  
    rrd = Reporteria.GenerarReporteDetallado(Moneda.Soles);  
  
    if (rrd != null)  
    {  
        foreach (PropertyInfo pi in rrd.GetType().GetProperties())  
        {  
            if (pi.Name == "DetalleConf" || pi.Name == "DetallePend" || pi.Name == "DetalleQr")  
            {  
                List<Txn> detalle = (List<Txn>)pi.GetValue(respuesta, null);  
  
                if (detalle.Count > 0)  
                {  
                    Console.WriteLine("##### {0} #####", pi.Name);  
  
                    foreach (Txn t in detalle)  
                    {  
                        Console.WriteLine("Ref: {0}\tTarj.: {1}\tMarca: {2}\tTipo:  
{3}\tImprt.: {4}", t.Referencia, t.Tarjeta, t.Marca, t.Tipo,  
t.Importe);  
                    }  
  
                    Console.WriteLine("#####");  
                }  
            }  
            else  
            {  
                Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(respuesta, null));  
            }  
        }  
    }  
}  
catch (PeticonException ex)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Reporte Cierre Resumen

Para generar el reporte de cierre resumen se deberán proporcionar los siguientes parámetros a la función:

- ✓ **idTurno:** Identificador del turno del cual se desea obtener el reporte (máximo 4 caracteres).

```
RespuestaReporteResumen rcr;  
  
try  
{  
    String idTurno = "88"  
  
    rcr = Reporteria.GenerarReporteCierreResumen(idTurno);  
  
    if (rcr != null)  
    {  
        foreach (PropertyInfo pi in rcr.GetType().GetProperties())  
        {  
            Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(rcr, null));  
        }  
    }  
    else  
    {  
        Console.WriteLine("Sin informacion para el idTurno: " + idTurno);  
    }  
}  
catch (PeticonException ex)  
{  
    Console.WriteLine("Error: " + pEx.Message);  
}
```

Reporte Cierre Operaciones

Para generar el reporte de cierre operaciones se deberán proporcionar los siguientes parámetros:

- ✓ **idTurno:** Identificador del turno del cual se desea obtener el reporte (máximo 4 caracteres).

```
RespuestaReporteOperaciones rco;

try
{
    String idTurno = "88"

    rco = Reporteria.GenerarReporteCierreOperaciones(idTurno);

    if (rco != null)
    {
        foreach (PropertyInfo pi in rcr.GetType().GetProperties())
        {
            if (pi.Name == "confTxns" || pi.Name == "pendTxns")
            {
                List<OperacionesCierre> lista =
                    (List<OperacionesCierre>)pi.GetValue(rco, null);

                Console.WriteLine("##### {0} #####", pi.Name);

                if (lista != null && lista.Count > 0)
                {
                    foreach (OperacionesCierre op in lista)
                    {
                        Console.WriteLine(
                            "Num: {0}\tRef: {1}\tTarj.: {2}\tMarca: {3}\tTipo: {4}\tImprt.: {5}",
                            op.Numero, op.Referencia, op.Tarjeta, op.Marca, op.Tipo, op.Importe
                        );
                    }
                }
                else
                {
                    Console.WriteLine("<Sin operaciones>");
                }
                Console.WriteLine("#####");
            }
            else
            {
                Console.WriteLine("{0}: {1}", pi.Name, pi.GetValue(rco, null));
            }
        }
    }
    else
    {
        Console.WriteLine("Sin información de operaciones para el idTurno: " + idTurno);
    }
}
catch (PeticonException ex)
{
    Console.WriteLine("Error: " + pEx.Message);
}
```

Respuesta

Una vez que una transacción fue ejecutada, vamos a poder obtener la información necesaria para realizar la impresión de los vouchers a través de los siguientes métodos que se proporcionan, los cuales están dentro de la clase **Respuesta**.

Los datos contenidos en la respuesta varían dependiendo de la transacción realizada. Si la respuesta no contiene alguna propiedad el valor por default devuelto será una cadena vacía "".

Para obtener el valor de la propiedad es necesario utilizar el nombre de la propiedad (Para más información ver la descripción de la clase Respuesta).

Propiedad	Voucher	Descripción
Afiliacion		Este número lo proporciona BBVA y está formado por 8 números.
Moneda	Mandatorio	Moneda con la que se realiza la operación.
RazonSocial		Razón social registrada para la afiliación.
Direccion		Dirección registrada para la afiliación.
SerialTerminal		Número de serie del equipo que envía la operación.
IdTransaccion		Identificador único generado para la transacción realizada.
CodigoTransaccion		Código de transacción del requerimiento, identifica el tipo de operación que será procesada.
NombreTransaccion		Indica el tipo de transaccion que se realizó.
Importe	Mandatorio	Indica el importe de la transacción.
Propina		Indica la propina de la transacción.
Folio		Identificador utilizado para hacer una confirmación o una anulación de pre-autorización.
CodigoRespuesta		Regresa el código de autorización (2 dígitos) proporcionado por el banco emisor. Ver Tabla 2 (Códigos de respuesta)
Autorizacion	Mandatorio	Regresa el número de autorización (6 dígitos) proporcionado por el banco emisor.
Leyenda	Mandatorio	Respuesta del emisor.
ReferenciaFinanciera		Identificador de la transacción utilizado para realizar una anulación de la venta.
SecuenciaTransaccion		Numero de secuencia asociado a la transacción.
NumeroTarjeta	Mandatorio	Contiene el número de la tarjeta
Tarjetahabiente		Nombre del tarjetahabiente.

Propiedad	Voucher	Descripción
ModoLectura		Contiene los valores: • 01 cuando la tarjeta fue digitada • 03 QR • 05 para tarjetas insertadas (Chip) • 07 para tarjetas contactless (Chip) • 80 cuando fue por error en chip (full back) • 90 cuando se deslizó la tarjeta (banda) • 91 para tarjetas contactless (banda)
IdAplicacionTarjeta		Esta propiedad debe de estar presente en el ticket cuando la transacción se realizó con una tarjeta con chip.
AplicacionTarjeta		Etiqueta asociada a la tarjeta.
CriptogramaTarjeta		Esta propiedad contiene información cuando la transacción se realizó con una tarjeta chip.
IdBilletera		Indica el Id de la billetera con la que se realizó un pago con QR.
Billetera		Nombre de la billetera con la que se realizó un pago con QR.
CodigoPromocion		Código del tipo de promoción, verificar en tabla correspondiente.
Financiamiento		Número de meses para comenzar el pago.
Cuotas		Número de meses para dividir el cargo.
Mensajes[]		Mensajes de texto devueltos por el emisor en una transacción de cuotas.
MontoCuota		En pagos sin intereses. Monto devuelto por el emisor correspondiente al monto de cada cuota.
Operador		Operador que envía la operación.
FechaHora		Fecha y hora de la transacción.
Firma	Mandatorio	Indica si la firma es autógrafa, electrónica o pago rápido.
idTurno		Identificador del turno asociado a la transacción / cierre (turno/caja).
NumeroTerminal		Id lógico de la terminal del comercio (1-99, solo números).

Catálogos

Moneda

Catálogo de códigos para identificar el tipo de moneda de la afiliación.

Constante
Dolares
Soles

Firma

Catálogo de códigos para identificar el tipo de firma.

Constante
Autografa
Electronica
SinFirma
SinLeyenda

Códigos de Respuesta

CÓDIGO	LEYENDA	DESCRIPCIÓN
00	Aprobada XXXXXX	Tarjeta autorizada por el emisor
01	Llame al adquiriente	Consulte al emisor de la tarjeta
03	Terminal inválida	Este mensaje se genera por que la afiliación no existe o no están bien definidos sus atributos.
04	Retenga y llame	Mensaje que indica que la tarjeta está reportada como extraviada

CÓDIGO	LEYENDA	DESCRIPCIÓN
05	Declinada	El emisor declina la solicitud de autorización por algún problema en la tarjeta.
10	Aprobación Parcial XXXXXXX	Aprobación parcial
12	Transacción inválida	Transacción inválida (Fallback) / PostAutorización sin una Pre-Autorización.
13	Cantidad inválida	Importe de una transacción que no cumple con lo esperado.
14	Tarjeta inválida	Tarjeta inválida
19	Reintente	Mensaje generado por problemas de comunicación
30	Error en formato	Error de formato
33	Tarjeta expirada	Tarjeta vencida o expirada
40	Servicio no disponible	Función no soportada
41	Retenga y llame	Recoger tarjeta
43	Retenga y llame	Recoger tarjeta
45	Promoción no permitida	La afiliación no está permitida para realizar operaciones de promociones.
47	Transacción no realizada acuda a sucursal	Transacción no realizada por haber excedido su límite permitido.
48	Ingrese cód. Seg.	Se requiere que sea ingresado el código de seguridad o cvv2 que aparece en la parte posterior de las tarjetas
49	Cod.de(seg.erroneo	Se ingresó un código de seguridad o cvv2 erróneo.
50	Ha superado el núm. de txns. rechazadas	Cuando se excedió el número de intentos en la parametría establecida para operaciones.

CÓDIGO	LEYENDA	DESCRIPCIÓN
51	Fondos insuficientes	No cuenta con el disponible suficiente para amparar la venta.
53	Cuenta inexistente	Cuenta inexistente
55	Pin incorrecto	Nip incorrecto
58	Operación invalida	La afiliación no tiene definidos los atributos para permitir este tipo de operación.
62	Tarjeta No Permitida	Tarjeta No Permitida.
65	Retiro cash excedido	Intentos De Retiros excedido.
71	Debe inicializar PIN PAD	El host determina que el dispositivo debe cargar llaves por primera vez o renovarlas
72	Reintente – Llamar a Soporte	Problema inicializando Llaves
73	Error en el Sistema – Llamar a Soporte	La caja o aplicación del comercio, alteraron o truncaron los datos cifrados, entregados por el dispositivo
75	PIN Incorrecto	Número de intentos de NIP excedidos
76	Cuenta bloqueada	Cuenta bloqueada

Descripción de Clases

Clase Tarjeta

Retorno	Método	Descripción
Pan	string ()	Número de tarjeta.
Producto	string ()	Si se tiene registrado retorna el producto de la tarjeta

MaxCuotasPsi	int()	Retorna el número máximo de cuotas sin intereses que se puede ofrecer al tarjetahabiente.
CuotasPci	bool()	Indica si se pueden ofrecer o no cuotas con intereses al tarjetahabiente.

Clase Interfaz

Método	Retorno	Descripción
Configuracion Configuracion = configuracion	Configuracion void	Instancia de configuración.
Instance	Interfaz	Obtiene instancia de Interfaz.
PinPadInfo()	Info	Recupera la versión, modelo, número de serie y actualización del Pin Pad.
Inicializar()	void	Inicializa el componente SDK.

Clase Configuración

Propiedad	Tipo de dato	Descripción
Logs	bool	Obtiene o establece si es necesario generar logs o no.
PinpadConexion	string	Obtiene o establece el tipo de conexión del Pin Pad.
PinPadPuerto	string	Obtiene o establece el puerto de escucha del dispositivo Android.
PinPadAndroid	bool	Obtiene o establece si el dispositivo que se utilizará es Android.
PinpadTimeOut	string	Obtiene o establece el tiempo en segundos que el Pin Pad espera una respuesta.
PinpadMensaje	string	Obtiene o establece el mensaje que se muestra en la pantalla del Pin Pad.
RutaBase	string	Obtiene o establece la ruta base de archivos.
HostUrl	string	Obtiene o establece la dirección IP o URL a la que el componente enviará las transacciones.
HostTimeOut	string	Obtiene o establece el tiempo que el componente espera una respuesta del Host.
UrlProxy	string	Url del servidor proxy que se desea utilizar.
PuertoProxy	string	Puerto del servidor proxy que se desea utilizar.
Afiliacion	string	Obtiene o establece la afiliación del comercio.
MonedaAfiliacion	Moneda	Obtiene o establece la moneda de la afiliación del comercio.
IdAplicacion	string	Obtiene o establece el identificador único del comercio que autoriza realizar transacciones.

Propiedad	Tipo de dato	Descripción
ClaveSecreta	string	Obtiene o establece el número secreto del comercio que autoriza realizar transacciones.
NumeroTerminal	string	Obtiene o establece el id lógico de la terminal del comercio.

Clase Petición

Propiedad	Tipo Dato	Descripción
Operador	string	Identifica al operador que realiza la operación
Fecha	string	Fecha local en la que se realiza la operación, con un formato de año, mes, día, hora, minutos y segundos. (AAAAMMDDHHMMSS).

Método	Retorno	Descripción
SetOperacion(Operacion codigoOperacion, Dictionary<ParametroOperacion, object> params)	void	Agrega la operación a realizar.
SetPromocionMeses(Promocion codigo, int parcializacion)	void	Indica el plan para diferir el pago.
LeerTarjeta()	Tarjeta	Lee tarjeta cargo del Pin Pad.
GenerarQR()	void	Genera el código QR en la pantalla del dispositivo.
FinalizarVentaQR()	Respuesta	Solicita el estatus de la venta QR.
CancelarVentaQR()	void	Solicita se cancele la transacción en curso.
Autorizar()	Respuesta	Solicita la autorización de la operación.
ConsultarTransaccion(string idTransaccion)	Respuesta	Consulta el estatus de una transacción.
SincronizacionFinal()	void	Cancelar transacción y finalizar comunicación con pinpad.
ConsultarPreautorizaciones(string fechaInicial, string fechaFinal)	RespuestaPreautorizacion	Contiene la lista de pre-autorizaciones según el rango de fechas proporcionado en formato yyyyMMddHHmmss.
CambioMetodoPago	Respuesta	Evento para realizar el cambio de método de pago entre tarjeta presente y QR.

Clase Respuesta

Propiedad	Tipo de Dato	Descripción
Afiliacion	string	Este número lo proporciona BBVA y está formado por 8 números.
AfiliacionAmex	string	Uso futuro
Moneda	string	Moneda de la solicitud.
RazonSocial	string	Razón social registrada para la afiliación.
Direccion	string	Dirección registrada para la afiliación.
SerieTerminal	string	Número de serie del equipo que envía la operación.
IdTransaccion	string	Identificador único generado para la transacción realizada.
CodigoTransaccion	string	Código de transacción del requerimiento, identifica el tipo de operación que será procesada.
NombreTransaccion	string	Indica el nombre de la transacción realizada.
Importe	string	Monto por el cual se realizará la solicitud.
Propina	string	Propina de la transacción
Folio	string	Indentificador utilizado para hacer una confirmación o una anulación de pre-autorización.
CodigoRespuesta	string	Regresa el código de respuesta (2 dígitos) proporcionado por el emisor
Autorizacion	string	Regresa el número de autorización (6 dígitos) proporcionado por el emisor.
Leyenda	string	Respuesta del emisor.
ReferenciaFinanciera	string	Durante el proceso de una operación de anulación, es necesario que éste dato concuerde con el enviado durante el proceso de venta, en caso de no coincidir con el de la venta la operación de cancelación y de post propina será denegada.
SecuenciaTransaccion	string	Numero de secuencia asociado a la transacción.
NumeroTarjeta	string	Contiene el número de la tarjeta.
Tarjetahabiente	string	Contiene el nombre del tarjetahabiente.
ModoLectura	Firma	Contiene los valores: • 01 cuando la tarjeta fue digitada • 03 QR • 05 para tarjetas insertadas (Chip) • 07 para tajteas contactless (Chip) • 80 cuando fue por error en chip (full back) • 90 cuando se delizó la tarjeta (banda) • 91 para tarjetas contacless (banda)
IdAplicacionTarjeta	string	Id de aplicación de la tarjeta

Propiedad	Tipo de Dato	Descripción
AplicacionTarjeta	string	Esta propiedad debe de estar presente en el ticket cuando la transacción se realizó con una tarjeta con chip.
CriptogramaTarjeta	string	Esta propiedad contiene información cuando la transacción se realizó con una tarjeta chip.
IdBilletera	string	Indica el Id de la billetera con la que se realizó un pago con QR.
Billetera	string	Nombre de la billetera con la que se realizó un pago con QR.
EmisorTarjeta	string	Banco emisor de la tarjeta.
ProductoTarjeta	string	Esta propiedad debe de estar presente en el ticket cuando la transacción se realizó con una tarjeta con chip.
CodigoPromocion	string	Código del tipo de promoción, verificar en tabla correspondiente.
Financiamiento	string	Número de meses para financiar el cargo.
Cuotas	string	Número de meses a dividir el cargo.
Mensajes	string[]	Mensajes de texto devueltos por el emisor en una transacción de cuotas
MontoCuota	string	En pagos sin intereses. Monto devuelto por el emisor correspondiente al monto de cada cuota.
Operador	string	Operador que envía la operación.
FechaHora	string	Fecha y hora de la transacción.
Firma	string	Indica si la firma es autógrafa, electrónica o pago rápido.

idTurno	string	Identificador del turno asociado a la transacción / cierre (turno/caja).
NumeroTerminal	string	Id lógico de la terminal del comercio (1-99, solo números).

Clase Reporteria

Método	Retorno	Descripción
GenerarReporteFirma (Moneda moneda)	RespuestaReporteFirma	Obtiene el reporte de firma física.
GenerarReporteTotalizado(Moneda moneda)	RespuestaReporteTotalizado	Obtiene el reporte totalizado.
GenerarReporteDetallado (Moneda moneda)	RespuestaReporteDetallado	Obtiene el reporte detallado
GenerarReporteResumen (string idTurno)	RespuestaReporteResumen	Obtiene el reporte resumen
GenerarReporteOperaciones (string idTurno)	RespuestaReporteOperaciones	Obtiene el reporte operaciones

Clase RespuestaReporteFirma

Propiedad	Tipo de Dato	Descripción
Operaciones	string	Número total de transacciones que requirieron firma autógrafa
TotalOperaciones	string	Importe totalizado de las transacciones que requirieron firma autógrafa
OperacionesVisa	string	Número total de transacciones pagadas con tarjeta Visa y que requirieron firma autógrafa
TotalOperacionesVisa	string	Importe totalizado de las transacciones pagadas con tarjeta Visa y que requirieron firma autógrafa
OperacionesMc	string	Número total de transacciones pagadas con tarjeta Master Card y que requirieron firma autógrafa
TotalOperacionesMc	string	Importe totalizado de las transacciones pagadas con tarjeta Master Card y que requirieron firma autógrafa
OperacionesAmex	string	Número total de transacciones pagadas con tarjeta Amex y que requirieron firma autógrafa
TotalOperacionesAmex	string	Importe totalizado de las transacciones pagadas con tarjeta Amex y que requirieron firma autógrafa

Propiedad	Tipo de Dato	Descripción
OperacionesDiners	string	Número total de transacciones pagadas con tarjeta Diners y que requirieron firma autógrafa
TotalOperacionesDiners	string	Importe totalizado de las transacciones pagadas con tarjeta Diners y que requirieron firma autógrafa

Clase RespuestaReporteTotalizado

Propiedad	Tipo de Dato	Descripción
OperacionesConf	string	Número total de transacciones confirmadas
TotalOperacionesConf	string	Importe totalizado de las transacciones confirmadas
Venta	string	Número total de ventas
TotalVenta	string	Importe totalizado de las ventas
PreautorizacionesConf	string	Número total de preautorizaciones confirmadas
TotalPreautorizacionesConf	string	Importe totalizado de las preautorizaciones confirmadas
Anulacion	string	Número total anulaciones
TotalAnulaciones	string	Importe totalizado de las anulaciones
Puntos	string	Número total de transacciones con puntos
TotalPuntos	string	Importe totalizado de las transacciones con puntos
TotalConfVisa	string	Importe totalizado de las transacciones confirmadas realizadas con tarjeta Visa
TotalConfMc	string	Importe totalizado de las transacciones confirmadas realizadas con tarjeta Master Card
TotalConfAmex	string	Importe totalizado de las transacciones confirmadas realizadas con tarjeta Amex
TotalConfDiners	string	Importe totalizado de las transacciones confirmadas realizadas con tarjeta Diners
PreautorizacionesPend	string	Número total de preautorizaciones pendientes
TotalPreautorizacionesPend	string	Importe totalizado de preautorizaciones pendientes
TotalPendVisa	string	Importe totalizado de las operaciones pendientes realizadas con tarjeta Visa
TotalPendMc	string	Importe totalizado de las operaciones pendientes realizadas con tarjeta Master Card
TotalPendAmex	string	Importe totalizado de las operaciones pendientes realizadas con tarjeta Amex
TotalPendDiners	string	Importe totalizado de las operaciones pendientes realizadas con tarjeta Diners
VentasQr	string	Número total de ventas QR
TotalVentasQr	string	Importe totalizado de las ventas QR
AnulacionesQr	string	Número total de anulaciones QR

Propiedad	Tipo de Dato	Descripción
TotalAnulacionesQr	string	Importe totalizado de anulaciones QR
TotalQrVisa	string	Importe totalizado de las operaciones QR realizadas con tarjeta Visa
TotalQrMc	string	Importe totalizado de las operaciones QR realizadas con tarjeta Master Card
TotalQrAmex	string	Importe totalizado de las operaciones QR realizadas con tarjeta Amex
TotalQrDiners	string	Importe totalizado de las operaciones QR realizadas con tarjeta Diners

Clase RespuestaReporteDetallado

Propiedad	Tipo de Dato	Descripción
Venta	string	Número total de ventas
TotalVenta	string	Importe totalizado de las ventas
Anulacion	string	Número total anulaciones
TotalAnulacion	string	Importe totalizado de las anulaciones
Puntos	string	Número total de transacciones con puntos
TotalPuntos	string	Importe totalizado de las transacciones con puntos
PreautorizacionConf	string	Número total de preautorizaciones confirmadas
TotalPreautorizacionConf	string	Importe totalizado de las preautorizaciones confirmadas
OperacionesConf	string	Número total de transacciones confirmadas
TotalConf	string	Importe totalizado de las transacciones confirmadas
DebitoConf	string	Número de transacciones confirmadas realizadas con tarjeta de débito
TotalDebitoConf	string	Importe totalizado de las transacciones confirmadas realizadas con tarjeta de débito
CreditoConf	string	Número de transacciones confirmadas realizadas con tarjeta de crédito
TotalCreditoConf	string	Importe totalizado de las transacciones confirmadas realizadas con tarjeta de crédito
DetalleConf	List<Txn>	Lista con el detalle de transacciones confirmadas
PreautorizacionPend	string	Número total de preautorizaciones pendientes
TotalPreautorizacionPend	string	Importe totalizado de las preautorizaciones pendientes
DebitoPend	string	Número de transacciones pendientes realizadas con tarjeta de débito
TotalDebitoPend	string	Importe totalizado de las transacciones pendientes realizadas con tarjeta de débito

Propiedad	Tipo de Dato	Descripción
CreditoPend	string	Número de transacciones pendientes realizadas con tarjeta de crédito
TotalCreditoPend	string	Importe totalizado de las transacciones pendientes realizadas con tarjeta de crédito
DetallePend	List<Txn>	Lista con el detalle de transacciones pendientes
OperacionQr	string	Número total de operaciones QR
TotalQr	string	Importe totalizado de operaciones QR
AnulacionesQR	string	Número total de anulaciones QR
TotalAnulacionesQR	string	Importe totalizado de las anulaciones QR realizadas
DebitoQr	string	Número de transacciones QR realizadas con tarjeta de débito
TotalDebitoQr	string	Importe totalizado de las transacciones QR realizadas con tarjeta de débito
BilleteraBBVA	string	Número de transacciones QR realizadas con billetera BBVA
TotalBilleteraBBVA	string	Importe totalizado de las transacciones QR realizadas con billetera BBVA
BilleteraYape	string	Número de transacciones QR realizadas con billetera Yape
TotalBilleteraYape	string	Importe totalizado de las transacciones QR realizadas con billetera Yape
DetalleQR	List<Txn>	Lista con el detalle de transacciones QR

Clase RespuestaReporteResumen

Propiedad	Tipo de Dato	Descripción
fechalnicio	string	String Fecha y hora de inicio del rango considerado en el cierre.
fechaCierre	string	String Fecha y hora de fin del rango considerado en el cierre.
marcasAmexOperaciones	string	Número total de operaciones realizadas con tarjeta American Express.
marcasAmexTotal	string	Importe totalizado de operaciones realizadas con tarjeta American Express.
marcasVisaOperaciones	string	Número total de operaciones realizadas con tarjeta Visa.
marcasVisaTotal	string	Importe totalizado de operaciones realizadas con tarjeta Visa.
marcasMcOperaciones	string	Número total de operaciones realizadas con tarjeta MasterCard.
marcasMcTotal	string	Importe totalizado de operaciones realizadas con tarjeta MasterCard.

Propiedad	Tipo de Dato	Descripción
marcasDinnersOperaciones	string	Número total de operaciones realizadas con tarjeta Dinners.
marcasDinnersTotal	string	Importe totalizado de operaciones realizadas con tarjeta Dinners.
productoDebitoOperaciones	string	Número total de operaciones realizadas con tarjeta Débito.
productoDebitoTotal	string	Importe totalizado de operaciones realizadas con tarjeta Débito.
productoCreditoOperaciones	string	Número total de operaciones realizadas con tarjeta Crédito.
productoCreditoTotal	string	Importe totalizado de operaciones realizadas con tarjeta Crédito.
tipoOperaciones	string	Número total de operaciones considerando todos los tipos de operación
tipoTotal	string	Importe total considerando todos los tipos de operación.
tipoVentasDccOperaciones	string	Número total de operaciones de Ventas DCC.
tipoVentasDccTotal	string	Importe totalizado de Ventas DCC.
tipoAnulacionesOperaciones	string	Número total de operaciones de Anulaciones.
tipoAnulacionesTotal	string	Importe totalizado de Anulaciones.
tipoConfirmacionesDccOperaciones	string	Número total de operaciones de Confirmaciones DCC.
tipoConfirmacionesDccTotal	string	Importe totalizado de Confirmaciones DCC.
tipoPropinasOperaciones	string	Número total de operaciones de Propinas.
tipoPropinasTotal	string	Importe totalizado de Propinas.
tipoPagoConPuntosOperaciones	string	Número total de operaciones de Pago con Puntos
tipoPagoConPuntosTotal	string	Importe totalizado de Pago con Puntos.
tipoVentasOperaciones	string	Número total de operaciones de Ventas.
tipoVentasTotal	string	Importe totalizado de Ventas.
tipoConfirmacionesOperaciones	string	Número total de operaciones de Confirmaciones.
tipoConfirmacionesTotal	string	Importe totalizado de Confirmaciones.

Clase RespuestaReporteOperaciones

Propiedad	Tipo de Dato	Descripción
fechaInicio	string	String Fecha y hora de inicio del rango considerado en el cierre.

Propiedad	Tipo de Dato	Descripción
fechaCierre	string	String Fecha y hora de fin del rango considerado en el cierre.
confTotal	string	Importe total de las operaciones confirmadas.
confOperaciones	string	Número total de operaciones confirmadas.
confVentasDccTotal	string	Importe total de ventas confirmadas.
confVentasDccOperaciones	string	Número total de operaciones de ventas DCC confirmadas.
confAnulacionesTotal	string	Importe total de anulaciones confirmadas.
confAnulacionesOperaciones	string	Número total de anulaciones confirmadas
confConfirmacionesDccTotal	string	Importe total de confirmaciones DCC confirmadas.
confConfirmacionesDccOperaciones	string	Número total de operaciones de confirmaciones DCC confirmadas.
confPagoPuntosTotal	string	Importe total de pago con puntos confirmadas.
confPagoPuntosOperaciones	string	Número total de operaciones de pago con puntos confirmadas
confVentasTotal	string	Importe total de ventas confirmadas.
confVentasOperaciones	string	Número total de operaciones de ventas confirmadas
confConfirmacionesTotal	string	Importe total de confirmaciones confirmadas.
confConfirmacionesOperaciones	string	Número total de operaciones de confirmaciones confirmadas.
confProductoDebitoTotal	string	Importe total de operaciones con tarjeta Débito en confirmadas.
confProductoDebitoOperaciones	string	Número total de operaciones con tarjeta Débito en confirmadas.
confProductoCreditoTotal	string	Importe total de operaciones con tarjeta Crédito en confirmadas.
confProductoCreditoOperaciones	string	Número total de operaciones con tarjeta Crédito en confirmadas.
confTxns	List<OperacionCierre>	Listado de transacciones confirmadas. Cada elemento se expone como objeto OperacionCierre.
pendTotal	string	Importe total de las operaciones pendientes.
pendOperaciones	String	Número total de operaciones pendientes.
pendTxns	List<OperacionCierre	Listado de transacciones pendientes. Cada elemento se expone como objeto OperacionCierre

Clase Txn

Propiedad	Tipo de Dato	Descripción
Referencia	string	Referencia financiera que identifica a la transacción
Tarjeta	string	Último 4 dígitos del número de la tarjeta con la que se realizó la transacción
Marca	string	Marca de la tarjeta con la que se realizó la transacción
Tipo	string	Identificador del tipo de transacción
Importe	string	Importe de la transacción

Clase PeticionException

La excepción es generada cuando genera cualquier error dentro del SDK.

Retorno	Método	Descripción
string	Message	Recupera el mensaje de error.

Clase PinPadKeysException

La excepción es generada cuando el Pin Pad requiere una carga de llaves.

Retorno	Método	Descripción
string	Message	Recupera el mensaje de error generado.

ANEXO Ejemplos Diagramación De Vouchers

1) PAGO CON TARJETA FISICA PIN

1) VENTA
2) OPENPAY BY BBVA
3) 10/04/2025 19:08
4) NOMBRE DE COMERCIO
5) REF: 1234
6) APP: 618234
9) TC: *****1234
10) TOTAL: S/ 8.00
11) PAGO RAPIDO, NO REQUIERE PIN NI FIRMA
12) AID: A000000000000031010
13) APP LABEL: VISA CREDITO
14) CRYPTO : 12 12 8D FE 23 3E 23 67

2) PAGO CON TARJETA FISICA CON FIRMA

- 1) VENTA
- 2) OPENPAY BY BBVA
- 3) 10/04/2025 19:08
- 4) NOMBRE DE COMERCIO
- 5) REF: 1234
- 6) APP: 618234
- 9) TC: *****1234
- 10) TOTAL: S/ 8.00
- 11) AUTOGRAFA

Firma

- 12) AID: A000000000000031010
- 13) APP LABEL: VISA CREDITO
- 14) CRYPTO : 12 12 8D FE 23 3E 23 67

3) PAGO CON TARJETA FISICA VENTA CON CUOTAS

```
1) VENTA
2) OPENPAY BY BBVA
3) 10/04/2025 19:08
4) NOMBRE DE COMERCIO
5) REF: 1234
6) APP: 618234
7) CUOTAS: 2
8) CUOTA APROX: 12.34
9) TC: *****1234
10) TOTAL: S/ 8.00
11) PAGO RAPIDO, NO REQUIERE PIN NI FIRMA
12) AID: A000000000000031010
13) APP LABEL: VISA CREDITO
14) CRYPTO : 12 12 8D FE 23 3E 23 67
```

4) PAGO CON QR

1) VENTA
2) OPENPAY BY BBVA
3) 10/04/2025 19:08
4) NOMBRE DE COMERCIO
5) REF: 1234
6) APP: 618234
9) TC: *****1234
10) TOTAL: S/ 8.00
11) PAGO CON QR
12) APP LABEL: VISA CREDITO

5) ANULACION PAGO CON TARJETA FISICA

1) ANULACION
2) OPENPAY BY BBVA
3) 10/04/2025 19:08
4) NOMBRE DE COMERCIO
5) REF: 1234
6) APP: 618234
9) TC: *****1234
10) TOTAL: S/ 8.00
11) PAGO RAPIDO, NO REQUIERE PIN NI FIRMA
12) AID: A000000000000031010
13) APP LABEL: VISA CREDITO
14) CRYPTO : 12 12 8D FE 23 3E 23 67

6) ANULACION PAGO CON QR

- 1) ANULACION
- 2) OPENPAY BY BBVA
- 3) 10/04/2025 19:08
- 4) NOMBRE DE COMERCIO
- 5) REF: 1234
- 6) APP: 618234
- 10) TOTAL: S/ 8.00
- 11) PAGO CON QR

7) VOUCHER CON CAMPAÑA CSI DINERS

- 1) VENTA
- 2) OPENPAY BY BBVA
- 3) 10/04/2025 19:08
- 4) NOMBRE DE COMERCIO
- 5) REF: 1234
- 6) APP: 618234
- 7) CUOTAS: 2
- 8) CUOTA APROX: S/ 12.34
- 9) TC: *****1234
- 10) TOTAL: S/ 8.00
- 11) PAGO RAPIDO, NO REQUIERE PIN NI FIRMA
- 12) AID: A000000000000031010
- 13) APP LABEL: VISA CREDITO
- 14) CRYPTO : 12 12 8D FE 23 3E 23 67
- 15) XXX ESTA COMPRA ESTA XXX
XXX DENTRO DEL PROGRAMA XXX
XXX CUOTAS SIN INTERESES XXX

8) VOUCHER CON CAMPAÑA PSI BBVA

1) VENTA
2) OPENPAY BY BBVA
3) 10/04/2025 19:08
4) NOMBRE DE COMERCIO
5) REF: 1234
6) APP: 618234
7) CUOTAS: 2
8) CUOTA APROX: S/ 12.34
9) TC: *****1234
10) TOTAL: S/ 8.00
16) PAGO SIN INTERESES
11) PAGO RAPIDO, NO REQUIERE PIN NI FIRMA
12) AID: A00000000000031010
13) APP LABEL: VISA CREDITO
14) CRYPTO : 12 12 8D FE 23 3E 23 67

Obtención de datos

1. Tipo de operación si es VENTA o ANULACION este valor es fijo
2. Nombre del adquirente OPENPAY BY BBVA este valor es fijo
3. Fecha y hora de la transacción se puede obtener del campo "fechaHora" de la respuesta de la transacción, formato DD/MM/AAAA HH:MM
4. Nombre del Comercio se puede obtener del campo "razonSocial" de la transacción
5. Se debe poner solo los ultimo 4 dígitos del número que viene en el campo "referenciaFinanciera"
6. Se debe mostrar el código de autorización que se obtiene del campo "autorización"

7. Se debe mostrar el número de cuotas que se obtiene del campo "cuotas"
8. Se debe mostrar el cálculo de cuota aproximada se obtiene del campo "montoCuota"
9. Se debe mostrar los últimos 4 dígitos de la tarjeta que se usó para el pago se obtiene del campo "numeroTarjeta"
10. Se debe mostrar el monto de la venta, se obtiene del campo uniendo el campo "moneda" + "importe"
11. Aquí se debe agregar el método de captura según la siguiente lógica:
 - a. Si en el campo firma viene:
 - i. ELECTRÓNICA – PIN VERIFICADO
 - ii. SIN FIRMA – PAGO RAPIDO, NO REQUIERE PIN NI FIRMA
 - iii. AUTÓGRAFA – Se debe agregar el campo de firma en el voucher para que el cliente firme de manera presencial
 - iv. Si se paga con QR en el campo "modoLectura" venta el código "03" – PAGO CON QR
12. Se debe mostrar el número que viene en el campo "IdAplicacionTarjeta"
13. Colocar la marca de la tarjeta si es Visa/Mastercard/Amex o Diners, esto vienen en el campo "aplicación tarjeta"
14. Colocar el número que se puede extraer del campo "CriptogramaTarjeta"
15. Colocar en el caso que venga poblado el campo "mensajes", por lo general viene poblado cuando el Comercio está afiliado a las campañas CSI de Diners = XXX ESTA COMPRA ESTA XXX XXX DENTRO DEL PROGRAMA XXXX XXX CUOTAS SIN INTERESES XXXX
16. Se debe mostrar el mensaje "PAGO SIN INTERESES", solo si el campo "codigoPromocion" venga poblado con el mensaje "PROMOCION.MESES_SIN_INTERESES", caso contrario no se debe mostrar ningún mensaje. Esto solo aplica si el Comercio está afiliado al programa de PAGOS SIN INTERESES con BBVA