

Parte 1

Proyecto

Proyecto

Información general

ImpresionMobile	
Directorio	W:\RAMAS\INNIT\MAIN\ImpresionMobile
Primera ventana	WIN_PrintOnAndroid
Plantilla	250 Phoenix

Proyecto

Lista de elementos del proyecto

	Nombre	Título
	WIN_PrintOnAndroid	
	BaseCFETicketBuilder	
	CFE	
	Detalles	
	Emisor	
	ETicketBuilder	
	Receptor	
	Totales	
	Factory	
	W:\RAMAS\INNIT\MAIN\ImpresionMobile\core-3.5.3.jar	

Estadísticas	
Número de ventanas:	1
Número de clases:	7
Número de conjuntos de procedimientos:	1

Proyecto

Lista de Carpetas Personales

Carpeta personal		Nombre
CFE		CFE
		Detalles
		Emisor
		Receptor
		Totales

Proyecto

Código

Inicialización de ImpresionMobile

```
gs_CFE          is ANSI string
gs_MACAddress   is string      = "66:32:58:DC:2F:53"
```

Proyecto

Errores, advertencias y mensajes de información

	Detalles del error	Ubicación	Fila	Columna
	El procedimiento 'RepeatString' sobrescribe la función WLanguage del mismo nombre.	BaseCFETicketBuilder.RepeatString Procedimiento local	11	11

Proyecto

Estadísticas de código

	Líneas ¹	% comen. ¹	Lin./proc. ¹
Factory	42	76	42
WIN_PrintOnAndroid	68	1	22
ImpresionMobile	3	0	3
BaseCFETicketBuilder	860	59	19

	Líneas ¹	% comen. ¹	Lin./proc. ¹
CFE	45	7	45
Detalles	30	0	30
ETicketBuilder	82	57	20
Emisor	28	0	28
Receptor	26	0	26
Totales	31	0	31
	1120	53	19

¹ Líneas : Número total de líneas de código.

% comen. : Porcentaje de comentarios en el código.

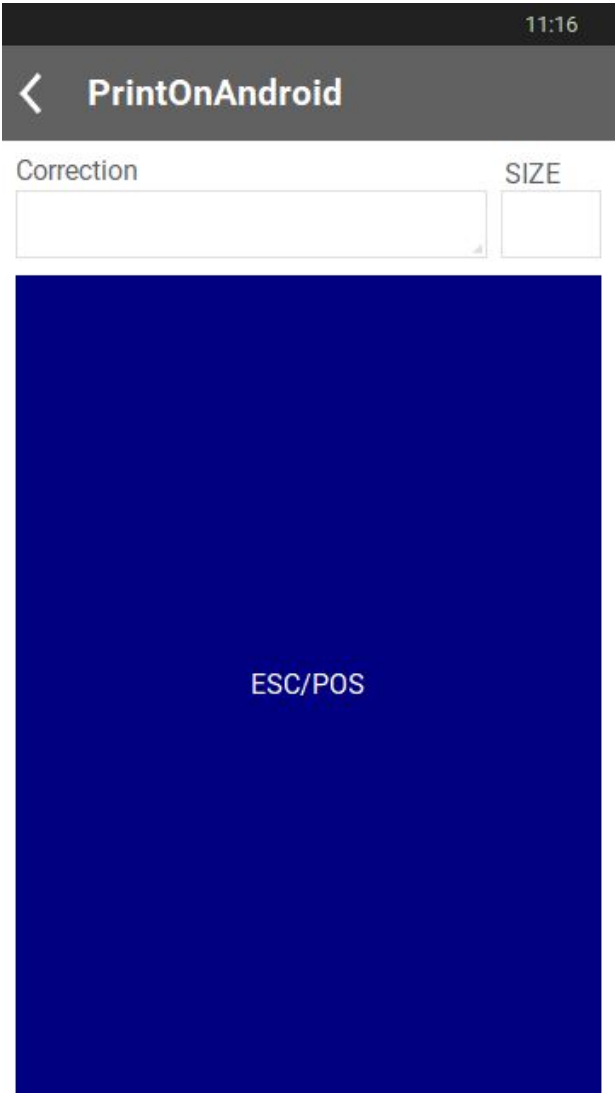
Lin./proc. : Número de líneas de código por proceso.

Parte 2

Ventana WINDEV

WIN_PrintOnAndroid

Imagen



WIN_PrintOnAndroid

Información general

WIN_PrintOnAndroid	
Archivo físico	W:\RAMAS\INNIT\MAIN\ImpresionMobile\WIN_PrintOnAndroid.wdw
Creado por	JBARBAT
Fecha de creación	4/4/2025 17:27:23
Última modificación por	JBARBAT





Última modificación	24/10/2025 10:21:02
Título	ES: PrintOnAndroid

Posición X	0	Posición Y	0	Ancho	360	Alto	569
------------	---	------------	---	-------	-----	------	-----

Visible	✓	Barra de título	✓	Barra de estado
Escala de 9 sectores		Redimensionable	✓	Desplazar/Fondo
Barra de desplazamiento auto.	✓			
Ctx. HFSQL indep.				

Estado inicial	Activo
Cursor sobre	0
Captura de pantalla	Sí
Nº Opciones menú principal	0
Unidad de tamaños	Plataforma única escritorio O móvil
Anclar al contenido	Vertical
Orientación ventana	Libre: la ventana sigue la orientación
Opacidad	100
Action Bar	Sí
Modo Hoja	No
Trans. barra de sistema	<Ninguno>
Desplazar para sensor	No
Tipo de bordes	Doble
Ignorar tema oscuro	No
Deslizar para actualizar	No
Modo de apertura (Móvil)	Maximizada al abrir
PopOver (iPad)	No
Modo widget	No
Actualización widget	0
Validación automática.	0
Tít. + Tiempo restante	Sí
Modo teclado virtual	No
Texto Markdown	No
Modo menú contex.	<Ninguno>
U.M.C.	No
Animación	Automático
Ancho ven. izquierda deslizando	0
Ancho ven. derecha deslizando	0
Deslizamiento ventana izquierda	Sí
Deslizamiento ventana derecha	Sí
Efecto Ventana deslizando	Cubrir: la ventana deslizando cubre la ventana actual
Ventana deslizando inferior: estilo	Sí
Ventana deslizando inferior: modal	No
Ventana deslizando inferior:	Sí

Estadísticas de la ventana

	Campo de entrada	1		Botón	1
	Combo Box	1		Action bar	1

Número de elementos en la ventana: 4

WIN_PrintOnAndroid

Información sobre los controles

Ventana : WIN_PrintOnAndroid

Campo de entrada

	Posición X	Posición Y	Ancho	Alto	Plano	Visible	Estado inicial	Tecla TAB	Tipo de entrada	Multilínea	Formato RTF	Contraseña	Entrada obligatoria	En blanco si es cero	NULL si está vacío	Sugerencias basadas en archivos	Borrar automáticamente	Tabulación automática	Desplazamiento horiz.	Desplazamiento vert.	Barra despl. horiz.	Barra despl. vert.
EDT_SIZE	293	8	59	63	0	✓	Activo	✓	Texto				No	✓			✓		✓			

Botón

	Posición X	Posición Y	Ancho	Alto	Plano	Visible	Estado inicial	Tecla TAB	Tipo botón
BTN_ESC_POS	8	79	344	482	0	✓	Activo	✓	Normal

Combo Box

	Posición X	Posición Y	Ancho	Alto	Plano	Visible	Estado inicial	Tecla TAB	Archivo de datos	Examinar	Ordenada	Con entrada	Entrada obligatoria	Sugerencias basadas en archivos	NULL si está vacío	Tooltip barra de desplazamiento	Desplazamiento horiz.
COMBO_Correction	8	8	277	63	0	✓	Activo	✓	Memoria	✓			No			✓	

Ventana : WIN_PrintOnAndroid

Botón : BTN_ESC_POS

Título	ES: ESC/POS
Orden de tabulación	3
Cursor sobre	0
Cambios de UI del usuario	Según ventana
On/Off	No
Accesibilidad: incluido	Sí
Tasa Paral. Alto	0
Tasa Paral. Y	0
Fijo	No
Ocultar durante el desplazamiento	No
Anclar al contenido	<Ninguno>
Sangría izquierda	0
Opacidad	100
Texto en movimiento	No
Animación predeterminada	Sí
Modo elipsis	<Ninguno>
Alineación de un contenedor	definido por el control Flexbox (auto)
Factor de contracción (flex-shrink)	0
Factor de crecimiento (flex-grow)	0
Texto Markdown	No
Modo menú contex.	<Ninguno>
Botón Apple	Desactivado
Botón Iniciar sesión con Apple -	Blanco
Botón de localización de Apple -	Ícono de flecha con relleno
Botón de localización de Apple -	"Ubicación actual"
Nº estados imagen	1
Nº estados fondo	5
Expandir imagen de fondo	Escala de 9 sectores personalizada
Nº. etapas animación	1
Anclaje	Ancho y alto
Tasa ancl. derecha	1000
Tasa anc. abajo	1000
Tasa ancl. horiz.	1000
Tasa ancl. vertical	1000

Combo Box : COMBO_Correction

Título	ES: Correction
Orden de tabulación	1
Cursor sobre	0
Cambios de UI del usuario	Según ventana
Utilizar el valor de gStoredValue	No
Accesibilidad: incluido	Sí
Tasa Paral. Alto	0
Tasa Paral. Y	0
Fijo	No
Ocultar durante el desplazamiento	No
Alineación horiz.	Izquierda
Alineación vert.	Centro
Orientación vertical	No
Alto List Box	0
Ancho List Box	277
Ancho expandido perso.	No
Alto fila	48
Sangría izquierda	2
Posición X interna	8
Ancho interno	277
Opacidad	100
Tamaño de entrada	0
Tipo de entrada	Texto
Modo elipsis	<Ninguno>
Alineación de un contenedor	definido por el control Flexbox (auto)
Factor de contracción (flex-shrink)	0
Factor de crecimiento (flex-grow)	0
Entrada no válida	Sí, bloqueante
Entrada obligatoria (error): Estilo	Sí
Entrada obligatoria (error): Texto	No
Entrada obligatoria (error): Icono	No
Entrada obligatoria (indicación):	No
Entrada obligatoria (indicación):	No
Entrada obligatoria (indicación):	Sí
Entrada no válida: Icono	No
Entrada no válida: Texto	No
Entrada no válida: Estilo	Sí
Borrar automáticamente	No
Tabulación automática	No
Vacío si es cero	No
Con foco	Sí
Historial de entradas	No
Historial de selección	No
gLinkActivate	Sí
Activa el teclado	Sí
Modo Smartphone	No
Valor persistente	No
Texto Markdown	No
Modo menú contex.	<Ninguno>

Control nativo		No					
Nº Máx. Reg		0					
1ª fila personalizada.		No					
Anclaje		<Ninguno>					
Tasa ancl. derecha	1000	Tasa anc. abajo	1000	Tasa ancl. horiz.	1000	Tasa ancl. vertical	1000

Campo de entrada : EDT_SIZE

Título	ES: SIZE
Orden de tabulación	2
Cursor sobre	0
Cambios de UI del usuario	Según ventana
Botón Calendario	No
Botón Eliminar	<Ninguno>
Tecla TAB	Acceder al control con TAB
Accesibilidad: incluido	Sí
Tasa Paral. Alto	0
Tasa Paral. Y	0
Fijo	No
Ocultar durante el desplazamiento	No
Anclar al contenido	<Ninguno>
Alineación horiz.	Izquierda
Posición X interna	293
Ancho interno	59
Opacidad	100
Formato HTML	0
Seguridad HTML	Sí
Markdown de solo lectura	No
Tamaño de entrada	1
Máscara de entrada	ES: 5
Autocompletar	<Automático>
Límite en el tooltip	No
Límite en el texto de sugerencia	Sí
Selección múltiple	No
Modo elipsis	<Ninguno>
Revisar ortografía	Sí
Alineación de un contenedor	definido por el control Flexbox (auto)
Factor de contracción (flex-shrink)	0
Factor de crecimiento (flex-grow)	0
Barra despl. horiz.	Nunca
Barra despl. vert.	Nunca
Entrada no válida	Sí, no bloqueante
Entrada obligatoria (error): Estilo	Sí
Entrada obligatoria (error): Texto	No
Entrada obligatoria (error): Icono	No
Entrada obligatoria (indicación):	No
Entrada obligatoria (indicación):	No
Entrada obligatoria (indicación):	Sí

Entrada no válida: Icono	No						
Entrada no válida: Texto	No						
Entrada no válida: Estilo	Sí						
Con foco	Sí						
Historial de entradas	No						
Activa el teclado	Sí						
Botón teclado	<Automático>						
Modo Smartphone	No						
Valores múltiples (tokens)	No						
Permitir duplicados	No						
Token eliminable	Sí						
Valor persistente	No						
Texto Markdown	No						
Modo menú contex.	<Ninguno>						
Gestión Euro	No						
Botón izquierdo: Número de estados	1						
Botón derecho: Número de estados	1						
Botón izquierdo: Ancho	0						
Botón derecho: Ancho	0						
Botón izquierdo: Alto	0						
Botón derecho: Alto	0						
Botón izquierdo: Monocromo	Color definido por el estilo						
Botón derecho: Monocromo	Color definido por el estilo						
Anclaje	<Ninguno>						
Tasa ancl. derecha	1000	Tasa anc. abajo	1000	Tasa ancl. horiz.	1000	Tasa ancl. vertical	1000

WIN_PrintOnAndroid

Código

Declaraciones globales de WIN_PrintOnAndroid

```
PROCEDURE MyWindow()
```

```
COMBO_Correction.DeleteAll()
COMBO_Correction.Add("QR_CORRECTION_L"+gLink(48))
COMBO_Correction.Add("QR_CORRECTION_M"+gLink(49))
COMBO_Correction.Add("QR_CORRECTION_Q"+gLink(50))
COMBO_Correction.Add("QR_CORRECTION_H"+gLink(51))
```

```
COMBO_Correction = 51
EDT_SIZE = 4
```

WIN_PrintOnAndroid

Código de los controles

Clic en BTN_ESC_POS

printCFE()

Inicialización de COMBO_Correction

```
// Version 1
// Description
// Simple list combo box
```

WIN_PrintOnAndroid

Procedimientos

Procedimiento local printCFE

```
/**
 * @brief <indique la acción que realiza el procedimiento>
 *
 * @note Descripción de los parámetros de entrada/salida de 'printCFE':
 * Explique el modo de funcionamiento.
 * Indique los casos particulares y los límites.
 * El desarrollador (JBARBAT) no ha introducido ninguna documentación
 */
PROCEDURE printCFE()
    cl_CFE          is CFE
    cl_Detalles      is Detalles

    s_SocketName is string    = SysDateTime()

    cl_CFE.tipoCFE          = 101
    cl_CFE.serieCFE         = "X"
    cl_CFE.numeroCFE        = 1234
    cl_CFE.codSeg           = 123456
    cl_CFE.total            = 1830
    cl_CFE.fecha            = SysDateTime()
    cl_CFE.qr               = "https://holawindev.com/public/qr.png"
    cl_CFE.cae              = 123456789
    cl_CFE.caeDesde         = 1
    cl_CFE.caeHasta         = 99999
    cl_CFE.resolucionDgi    = 1234
    cl_CFE.formaPago        = 2
    cl_CFE.sucursal         = 1
    cl_CFE.fechaGeneracion  = SysDateTime()
    cl_CFE.vencimiento      = SysDateTime()
    cl_CFE.tipoMoneda       = "USD"
    cl_CFE.adenda           = "Esta es la adenda"

    cl_CFE.Receptor.tipoDoc = 3
    cl_CFE.Receptor.documento = 123456789
    cl_CFE.Receptor.nombre   = "Empresa Testing"
    cl_CFE.Receptor.direccion = "18 de Julio 1234"
    cl_CFE.Receptor.ciudad   = "Montevideo"
    cl_CFE.Receptor.departamento = "Montevideo"
    cl_CFE.Receptor.pais     = "Uruguay"
    cl_CFE.Receptor.codPais  = "UY"
```

```

cl_Detalles.indFact          = 3
cl_Detalles.nomItem         = "PRODUCTO DE PRUEBAS"
cl_Detalles.cantidad        = 1
cl_Detalles.uniMedida        = "uni"
cl_Detalles.precio          = 1830
cl_Detalles.total           = 1830

cl_CFE.Detalles.add(cl_Detalles)

cl_CFE.Emisor.rut           = 123456789
cl_CFE.Emisor.razonSocial   = "EMISOR TESTING"
cl_CFE.Emisor.nombreComercial = "EMISOR TESTING"
cl_CFE.Emisor.direccion     = "Gral. Luna 1234"
cl_CFE.Emisor.ciudad        = "MONTEVIDEO"
cl_CFE.Emisor.razonSocial    = "EMISOR TESTING"
cl_CFE.Emisor.razonSocial    = "EMISOR TESTING"

cl_CFE.totales.netoIvaBasico = 1500
cl_CFE.Totales.ivaBasico     = 330

qr_Correction    is int          = Val(COMBO_Correction.StoredValue)
qr_Size          is int          = EDT_SIZE

builder          is object ETicketBuilder = GetCFETicketBuilder(101, BaseCFETicketBuilder::
PAPERWIDTH_50MM)
builder.SetPrinterWidth = 384

buffer           is Buffer
buffer += builder.InitPrinter()
buffer += builder.BuildTicket(cl_CFE,qr_Correction,qr_Size)
buffer += builder.ClosePrinter()

IF SocketConnectBluetooth(s_SocketName, "SerialPortServiceClass_UUID", gs_MACAddress) THEN
    SocketWrite(s_SocketName, buffer)
    SocketClose(s_SocketName)
ELSE
    Error(ErrorInfo(errFullDetails))
END

```

Parte 3

Clase

Totales

Código

Declaración de Totales

```
Totales is a Class
    noGravado          is numeric
    exportacion        is numeric
    impuestoPercibido is numeric
    ivaSuspendo        is numeric
    netoIvaMinimo      is numeric
    netoIvaBasico      is numeric
    netoIvaOtra        is numeric
    ivaMinimo          is numeric
    ivaBasico          is numeric
    ivaOtra            is numeric
    retenido           is numeric
    creditosFiscales  is numeric
    noFacturable      is numeric
END
```

Constructor

```
PROCEDURE Constructor()
```

Destructor

```
PROCEDURE Destructor()
```


Emisor

Código

Declaración de Emisor

```
Emisor is a Class
    rut                is numeric
    razonSocial        is string
    nombreComercial    is string
    logo               is string
    direccion          is string
    telephone1         is string
    telephone2         is string
    email              is string
    web                is string
    ciudad             is string
END
```

Constructor

```
PROCEDURE Constructor()
```

Destructor

```
PROCEDURE Destructor()
```

Detalles

Código

Declaración de Detalles

```
Detalles is a Class
    indFact          is numeric
    nomItem           is string
    descripcion       is string
    cantidad          is numeric
    uniMedida         is string
    precio            is numeric
    descRec           is numeric
    descRecItem       is numeric
    descRecItemTipo   is numeric
    codItem           is string
    retenidoPercibido is numeric
    total             is numeric
END
```

Constructor

```
PROCEDURE Constructor()
```

Destructor

```
PROCEDURE Destructor()
```

Receptor

Código

Declaración de Receptor

```
Receptor is a Class
  tipoDoc      is string
  documento    is string
  nombre       is string
  direccion    is string
  ciudad       is string
  departamento is string
  pais         is string
  codPais      is string
END
```

Constructor

```
PROCEDURE Constructor()
```

Destructor

```
PROCEDURE Destructor()
```

ETicketBuilder

Código

Declaración de ETicketBuilder

```
ETicketBuilder is a Class
    inherits from BaseCFETicketBuilder
END
```

Método BuildTicket

```
#!/
#!/ @brief Construye un ticket de tipo eTicket (consumo final) completo en formato ESC/POS
#!/
#!/ @param cl_CFE Objeto CFE con todos los datos del comprobante fiscal electrónico
#!/ @param correctionLevel Nivel de corrección de errores para el código QR (valores
    válidos: QR_CORRECTION_L, QR_CORRECTION_M, QR_CORRECTION_Q, QR_CORRECTION_H). Por defecto
    QR_CORRECTION_M (15% de corrección)
#!/ @param moduleSize Tamaño del módulo del código QR en puntos. Usar -1 para tamaño automático según
    el ancho del papel (5 para 50mm, 8 para 80mm)
#!/ @return Buffer con todos los comandos ESC/POS para imprimir el ticket completo
#!/
#!/ @note Sobrescribe el método abstracto de la clase base BaseCFETicketBuilder.
#!/ Implementación específica para tickets de tipo 101 (eTicket - Consumo Final).
#!/
#!/ Estructura del ticket generado:
#!/ 1. Header: Datos del emisor (nombre comercial, RUT, dirección, etc.)
#!/ 2. Título CFE: Tipo, serie, número, fecha y moneda
#!/ 3. Separador de puntos
#!/ 4. Texto "CONSUMO FINAL" centrado (identificación del tipo de comprobante)
#!/ 5. Separador de puntos
#!/ 6. Receptor: Datos del cliente (si existen)
#!/ 7. Detalles: Líneas del comprobante con productos/servicios
#!/ 8. Totales: Desglose de IVA, retenciones y total general
#!/ 9. Información adicional: Si existe en el CFE
#!/ 10. Adenda: Texto adicional si existe
#!/ 11. Footer: Código QR, leyenda y datos del CAE
#!/
#!/ El método utiliza todos los métodos heredados de BaseCFETicketBuilder para
#!/ construir cada sección, garantizando formato consistente y compatible con
#!/ impresoras térmicas ESC/POS estándar.
#!/
#!/ @code
#!/ // Ejemplo de uso:
#!/ builder is ETicketBuilder
#!/ builder.SetPrinterWidth = BaseCFETicketBuilder::PAPERWIDTH_50MM
#!/
#!/ bufferTicket is Buffer
#!/ bufferTicket = builder.BuildTicket(objetoCFE, BaseCFETicketBuilder::QR_CORRECTION_M, 6)
#!/
#!/ // Enviar a impresora
#!/ SocketWrite(socketName, bufferTicket)
#!/ @endcode
#!/
#!/ @warning Este método es específico para eTickets (tipo 101). Para otros tipos
#!/ de CFE se deben crear clases derivadas con su propia implementación.
#!/
#!/ @see BaseCFETicketBuilder::BuildTicket() Método base abstracto
```

```

///! @see AddHeader() Para personalizar el encabezado
///! @see AddFooter() Para personalizar el pie con QR
///!
PROCEDURE BuildTicket(cl_CFE is CFE, correctionLevel is int<useful> = QR_CORRECTION_M, moduleSize is
int <useful>= -1) : Buffer

b is Buffer

b += AddHeader(cl_CFE.Emission)
b += AddCFETitleSection(cl_CFE)
b += AddDotSeparatorLine()
b += Charact(10) + AddCenteredText("CONSUMO FINAL")
b += AddDotSeparatorLine()
b += AddReceiver(cl_CFE.Receptor)
b += AddDetails(cl_CFE.Detalles)
b += AddTotals(cl_CFE)
b += AddInfoAdicional(cl_CFE)
b += AddAdenda(cl_CFE)
b += AddFooter(cl_CFE, correctionLevel, moduleSize)

RESULT b

```

Constructor

```
PROCEDURE Constructor()
```

Destructor

```
PROCEDURE Destructor()
```

CFE

Código

Declaración de CFE

```
CFE is a Class
  tipoCFE      is numeric
  serieCFE     is string
  numeroCFE    is numeric
  codSeg       is string
  total        is numeric
  fecha        is DateTime
  qr           is string
  cae          is string
  caeDesde     is numeric
  caeHasta     is numeric
  caeVencimiento is DateTime
  resolucioDgi is string
  tipoTraslado is numeric
  montosBrutos is numeric
  formaPago    is numeric
  sucursal     is numeric
  infoAdicional is string
  isAnulado    is numeric
  isCobranza   is numeric
  fechaGeneracion is DateTime
  vencimiento  is DateTime
  Receptor     is Receptor <serialize="receptor">
  tipoMoneda   is string
  Detalles     is array <serialize="detalles"> of Detalles
  adenda       is string
  Emisor       is Emisor <serialize="emisor">
  Totales      is Totales <serialize="totales">
  descRecs     is array of strings // Empty array, undefined type
  referencias  is array of strings // Empty array, undefined type
  mediosPago   is array of strings // Empty array, undefined type
END
```

Constructor

```
PROCEDURE Constructor()
```

Destructor

```
PROCEDURE Destructor()
```

BaseCFETicketBuilder

Código

Declaración de BaseCFETicketBuilder

```

///
///! @class BaseCFETicketBuilder
///! @brief Clase base para construir tickets de Comprobantes Fiscales Electrónicos (CFE) en formato
///! ESC/POS
///!
///! @note Esta clase proporciona la funcionalidad base para generar comandos de impresión térmica.
///! Es una clase abstracta que debe ser heredada para implementar tipos específicos de CFE.
///! Soporta impresoras de 50mm y 80mm con comandos ESC/POS estándar.
///!
///! @author JBARBAT
///! @date 17/10/2025
///!
BaseCFETicketBuilder is a Class
    // Ancho en puntos (píxeles térmicos horizontales): 384 (48mm), 576 (72mm)
    printerWidth is int = 576 // Default: 72mm (80mm de papel)

    CONSTANT
        PAPERWIDTH_50MM = 384
        PAPERWIDTH_80MM = 576

        QR_CORRECTION_L = 48
        QR_CORRECTION_M = 49
        QR_CORRECTION_Q = 50
        QR_CORRECTION_H = 51
    END
END

```

Método AddAdenda

```

///
///! @brief Agrega la sección de adenda al ticket
///!
///! @param cl_CFE Objeto CFE que contiene los datos del comprobante
///! @return String con la adenda formateada para impresión
///!
///! @note Procesa la adenda del CFE si existe, decodificando URL y convirtiendo HTML a texto plano.
///! Solo genera contenido si cl_CFE.adenda no está vacío.
///! Elimina tags HTML y caracteres especiales para compatibilidad con impresoras térmicas.
///!
PROCEDURE AddAdenda(cl_CFE is CFE) : string
res is string = ""
// Solo si cfe.adenda no es vacío
IF cl_CFE.adenda <> "" THEN
    res += AddSeparatorLine()
    res += AddBoldText("Adenda:") + Charact(10)
    // Decodificamos la URL
    decodedAdenda is string = URLDecode(cl_CFE.adenda)
    // Quitar tags HTML (si viene con <p>, <span>, etc.)
    localAdendaText is string = HTMLToText(decodedAdenda)
    // Muestra en el ticket

```

```

    res += localAdendaText + Charact(10)
    res += AddSeparatorLine()
END
RESULT res

```

Método AddBoldText

```

//!
//! @brief Aplica formato de negrita al texto usando comandos ESC/POS
//!
//! @param text Texto al que se aplicará formato negrita
//! @return String con comandos ESC/POS para texto en negrita
//!
//! @note Utiliza el comando ESC E 1 para activar negrita y ESC E 0 para desactivarla.
//! Compatible con la mayoría de impresoras ESC/POS estándar.
//!
PROCEDURE AddBoldText(text is string) : string

RETURN Charact(27) + Charact(69) + Charact(1) + text + Charact(27) + Charact(69) + Charact(0)

```

Método AddCAESection

```

//!
//! @brief Genera la sección de información del CAE (Código de Autorización Electrónico)
//!
//! @param cl_CFE Objeto CFE con datos del comprobante
//! @return Buffer con comandos ESC/POS para la sección CAE
//!
//! @note Genera una tabla con el CAE y su rango de numeración.
//! Incluye la fecha de vencimiento del CAE centrada.
//! El formato se ajusta automáticamente al ancho del papel configurado.
//!
PROCEDURE AddCAESection(cl_CFE is CFE) : Buffer
buf is Buffer
maxChars is int = GetMaxCharsPerLine()
// Primera línea: títulos
line1 is string = AlignText("CAE", "Inicio", "Fin", maxChars)
// Segunda línea: valores
line2 is string = AlignText(cl_CFE.cae, cl_CFE.serieCFE + cl_CFE.caeDesde, cl_CFE.serieCFE +
cl_CFE.caeHasta, maxChars)
// Separador
separator is string = RepeatString("-", maxChars)
// Fecha de vencimiento CAE (centrado)
line3 is string = CenteredLine("Fecha de vencimiento CAE")
line4 is string = CenteredLine(DateToString(cl_CFE.caeVencimiento, "DD/MM/YYYY"))
// Armar el buffer con saltos de línea
buf += line1 + Charact(10)
buf += line2 + Charact(10)
buf += separator + Charact(10)
buf += line3 + Charact(10)
buf += line4 + Charact(10)
RESULT buf

```

Método AddCenteredQR

```

//!
//! @brief Genera un código QR centrado horizontalmente

```



```

//!
//! @param qr Contenido del código QR
//! @param paperWidth Ancho del papel en puntos (384 o 576)
//! @param correctionLevel Nivel de corrección de errores (QR_CORRECTION_L/M/Q/H)
//! @param moduleSize Tamaño del módulo QR (-1 para automático)
//! @return Buffer con comandos ESC/POS para QR centrado
//!
//! @note Centra el QR usando ESC a 1, lo imprime y restaura alineación izquierda.
//! Incluye reset de impresora para salir del modo gráfico.
//!
PROCEDURE AddCenteredQR(qr is string, paperWidth is int = 576, correctionLevel is int = QR_CORRECTION_M
, moduleSize is int = -1) : Buffer
res is Buffer
// 1) Centrar
res += Charact(27) + Charact(97) + Charact(1) // ESC a 1
// 2) Agregar el QR propiamente dicho usando tu método actual
res += AddQR(qr, paperWidth, correctionLevel, moduleSize)
// Reset de la impresora (salir de modo gráfico)
res += Charact(27) + Charact(64)
// 3) Volver a la alineación izquierda
res += Charact(27) + Charact(97) + Charact(0)
RESULT res
    
```

Método AddCenteredQRAsImage

```

//!
//! @brief Genera un código QR como imagen centrada
//!
//! @param qrText Texto a codificar en el QR
//! @return Buffer con la imagen del QR centrada
//!
//! @note Utiliza la función Java GenerateQRCodeBase64 para crear el QR como bitmap.
//! La imagen se centra horizontalmente y se restaura la alineación después.
//!
PROCEDURE AddCenteredQRAsImage(qrText is string)<zombie>: Buffer
res is Buffer
// Centrado
res += Charact(27) + Charact(97) + Charact(1) // ESC a 1
// QR como imagen
res += AddQRAsImage(qrText)
// Reset y alineación izquierda
res += Charact(27) + Charact(64) // ESC @
res += Charact(27) + Charact(97) + Charact(0) // ESC a 0
RESULT res
    
```

Método AddCenteredText

```

//!
//! @brief Centra texto horizontalmente en la impresora
//!
//! @param text Texto a centrar
//! @return String con comandos ESC/POS para texto centrado
//!
//! @note Usa ESC a 1 para centrar y ESC a 0 para restaurar alineación.
//! Incluye salto de línea automático después del texto.
//!
PROCEDURE AddCenteredText(text is string) : string

RETURN Charact(27) + Charact(97) + Charact(1) + text + Charact(10) + Charact(27) + Charact(97) +
Charact(0)
    
```

Método AddCenteredTextWidth

```

///!
///! @brief Centra texto considerando el ancho específico del papel
///!
///! @param text Texto a centrar
///! @param paperWidth Ancho del papel (no usado directamente, para compatibilidad)
///! @return String con el texto centrado usando espacios
///!
///! @note Calcula el padding necesario basándose en GetMaxCharsPerLine().
///! Útil cuando se necesita centrado manual sin comandos ESC.
///!
PROCEDURE AddCenteredTextWidth(text is string, paperWidth is int <useful>) : string
maxChars is int = GetMaxCharsPerLine()
RETURN Charact(27) + Charact(97) + Charact(1) + PadCenter(text, maxChars) + Charact(10) + Charact(27) +
    Charact(97) + Charact(0)
    
```

Método AddCFETitleSection

```

///!
///! @brief Genera la sección de título del CFE
///!
///! @param cl_cfe Objeto CFE con datos del comprobante
///! @return String con el título formateado
///!
///! @note Incluye tipo de ticket (eTicket), serie y número.
///! Muestra fecha/hora y tipo de moneda alineados en la misma línea.
///! El formato se ajusta automáticamente al ancho del papel.
///!
PROCEDURE AddCFETitleSection(cl_cfe is CFE) : string
cfeTitle is string = ""
// Línea centrada con tipo de comprobante
cfeTitle += Charact(10) + Charact(10) + "eTicket " + cl_cfe.serieCFE + NumToString(cl_cfe.numeroCFE) +
    Charact(10)
// Fecha y moneda alineados
fechaStr is string = DateToString(cl_cfe.fecha, "DD/MM/YYYY HH:mm")
monedaStr is string = "Moneda: " + cl_cfe.tipoMoneda
charWidth is int = GetCharWidthForPrinter()
espacios is int = charWidth - Length(fechaStr) - Length(monedaStr)
fechaYMoneda is string = fechaStr
IF espacios > 0 THEN
    fechaYMoneda += RepeatString(" ", espacios)
END
fechaYMoneda += monedaStr + Charact(10)
cfeTitle += fechaYMoneda
RESULT cfeTitle
    
```

Método AddDetails

```

///!
///! @brief Genera el detalle de líneas del comprobante
///!
///! @param detailsList Array de objetos Detalles con las líneas del comprobante
///! @return String con todas las líneas de detalle formateadas
///!
///! @note Por cada ítem genera:
///! - Línea 1: Nombre del producto (negrita) y total alineado a derecha
///! - Línea 2: Descripción (si existe, comentada en el código)
///! - Línea 3: Cantidad x precio unitario y descuentos/recargos si aplican
///! Separa cada ítem con línea de puntos.
///!
PROCEDURE AddDetails(detailsList is array of Detalles) : string
resBuffer is string
    
```

```

resBuffer += AddDotSeparatorLine()
FOR EACH item OF detailsList
    // ===== LÍNEA 1: Nombre a la izq, total a la der =====
    leftPart is string = AddBoldText(item.nomItem)
    rightPart is string = FormatCurrency(item.total)
    resBuffer += AlignColumns(leftPart, rightPart)
    // ===== LÍNEA 2: Descripción (si existe) =====
    // Código comentado para descripción opcional
    // ===== LÍNEA 3: Cantidad, unidad, precio, dcto/recargo =====
    line3 is string = NumToString(item.cantidad) + " " + item.uniMedida + " x " + FormatCurrency(item.precio)
    IF item.descRecItem <> 0 THEN
        line3 += " "
        IF item.descRecItem < 0 THEN
            line3 += "- " + FormatCurrency(Abs(item.descRecItem)) + " dcto."
        ELSE
            line3 += "+ " + FormatCurrency(Abs(item.descRecItem)) + " rec."
        END
    END
    resBuffer += line3 + Charact(10)
    resBuffer += AddDotSeparatorLine()
END
RESULT resBuffer
    
```

Método AddDotSeparatorLine

```

//!
//! @brief Genera una línea separadora de puntos
//!
//! @return String con línea de puntos del ancho del papel
//!
//! @note El ancho se determina por GetCharWidthForPrinter().
//! Útil para separar secciones visualmente en el ticket.
//!
PROCEDURE AddDotSeparatorLine(): string

RETURN RepeatString(".", GetCharWidthForPrinter()) + Charact(10)
    
```

Método AddFooter

```

//!
//! @brief Genera el pie del ticket
//!
//! @param cl_CFE Objeto CFE con datos del comprobante
//! @param paperWidth Ancho del papel en puntos
//! @param correctionLevel Nivel de corrección QR
//! @param moduleSize Tamaño del módulo QR
//! @return Buffer con comandos para el pie del ticket
//!
//! @note Incluye código QR, leyenda del QR y sección CAE.
//! Finaliza con línea separadora de puntos.
//!
PROCEDURE AddFooter(cl_CFE is CFE, paperWidth is int <useful> = 576, correctionLevel is int <useful> =
QR_CORRECTION_M, moduleSize is int <useful> = -1) : Buffer
footer is Buffer
// --- Agregar QR ---
footer += AddQR(cl_CFE.qr)
// --- Leyenda debajo del QR ---
footer += Charact(10)
footer += AddQRLegendWrapByWords(cl_CFE)
// --- Información del CAE ---
    
```

```

footer += Charact(10)
footer += AddCAESection(cl_CFE)
// --- Línea final de separación ---
footer += Charact(10) + AddDotSeparatorLine()
RESULT footer

```

Método AddHeader

```

//!
//! @brief Genera el encabezado del ticket con datos del emisor
//!
//! @param emitter Objeto Emisor con datos de la empresa
//! @return String con el encabezado formateado
//!
//! @note Incluye nombre comercial centrado, razón social, RUT, dirección,
//! ciudad, teléfono y email (si existen).
//! Cada elemento en una línea separada.
//!
PROCEDURE AddHeader(emitter is Emisor) : string

header is string = ""

header += AddCenteredText(emitter.nombreComercial)
header += emitter.razonSocial + Charact(10)
header += "RUT " + emitter.rut + Charact(10)
header += emitter.direccion + Charact(10)
header += emitter.ciudad + Charact(10)
IF emitter.telephone1 <> "" THEN header += emitter.telephone1 + Charact(10)
IF emitter.email <> "" THEN header += emitter.email + Charact(10)

RESULT header

```

Método AddInfoAdicional

```

//!
//! @brief Agrega información adicional al ticket si existe
//!
//! @param cl_CFE Objeto CFE con los datos
//! @return String con la información adicional formateada
//!
//! @note Decodifica URL y elimina acentos para compatibilidad.
//! Solo genera contenido si infoAdicional no está vacía.
//! Se muestra entre líneas separadoras con título en negrita.
//!
PROCEDURE AddInfoAdicional(cl_CFE is CFE) : string
res is string = ""
// Solo si cfe.infoAdicional no es vacío
IF cl_CFE.infoAdicional <> "" THEN
  // Decodificamos por si viene con caracteres %xx
  localDecoded is string = URLDecode(cl_CFE.infoAdicional)
  res += AddSeparatorLine()
  res += AddBoldText("Informacion Adicional:") + Charact(10)
  res += NoAccent(localDecoded) + Charact(10)
  res += AddSeparatorLine()
END
RESULT res

```

Método AddQR

```

///!
///! @brief Genera comandos ESC/POS para imprimir un código QR
///!
///! @param qr Contenido del código QR
///! @param paperWidth Ancho del papel (384 para 50mm, 576 para 80mm)
///! @param correctionLevel Nivel de corrección (48=L, 49=M, 50=Q, 51=H)
///! @param moduleSize Tamaño del módulo (-1 para automático)
///! @return Buffer con comandos ESC/POS para el QR
///!
///! @note Tamaño automático: módulo 5 para papel 50mm, módulo 8 para 80mm.
///! Genera secuencia completa de comandos GS ( k para QR.
///! Incluye hack de "QR vacío" para compatibilidad con algunas impresoras.
///!
PROCEDURE AddQR(qr is string, paperWidth is int = 576, correctionLevel is int = QR_CORRECTION_M,
moduleSize is int = -1) : Buffer
res is Buffer
// Elegimos el moduleSize segun paperWidth
IF moduleSize = -1 THEN
    SWITCH paperWidth
        CASE 384:
            moduleSize = 5
        CASE 576:
            moduleSize = 8
        OTHER CASE:
            moduleSize = 8
    END
END
// Saltos de línea antes de imprimir
res += Charact(10) + Charact(10)
// Tamaño del QR (GS ( k 3 0 49 67 n)
res += Charact(29) + Charact(40) + Charact(107) + Charact(3) + Charact(0) + Charact(49) + Charact(67) +
Charact(moduleSize)
// Nivel de corrección de error (GS ( k 3 0 49 69 n)
res += Charact(29) + Charact(40) + Charact(107) + Charact(3) + Charact(0) + Charact(49) + Charact(69) +
Charact(correctionLevel)
// Enviar los datos del QR (GS ( k <Length> 0 49 80 48 + datos)
res += Charact(29) + Charact(40) + Charact(107) + Charact(Length(qr) + 3) + Charact(0) + Charact(49) +
Charact(80) + Charact(48) + qr
// Imprimir el QR (GS ( k 3 0 49 81 48)
res += Charact(29) + Charact(40) + Charact(107) + Charact(3) + Charact(0) + Charact(49) + Charact(81) +
Charact(48)
// Inmediatamente después, forzamos el "QR vacío"
res += Charact(29) + Charact(40) + Charact(107) // GS ( k
res += Charact(3) + Charact(0) // pL=3, pH=0
res += Charact(49) + Charact(80) + Charact(48) // cn=49, fn=80, subfn=48
RESULT res
    
```

Método AddQRAsImage

```

///!
///! @brief Genera un QR como imagen bitmap usando Java/Android
///!
///! @param qrText Texto a codificar en el QR
///! @return Buffer con la imagen del QR en formato ESC/POS
///!
///! @note Determina el tamaño según printerWidth configurado.
///! Utiliza GenerateQRCodeBase64 (función Java) para crear el bitmap.
///! Retorna buffer vacío si hay error en generación.
///!
PROCEDURE AddQRAsImage(qrText is string)<zombie>: Buffer
res is Buffer
// 1. Definir tamaño del QR según ancho del papel
sizeQR is int
SWITCH printerWidth
    CASE PAPERWIDTH_50MM: sizeQR = 384 // 50% del ancho
    
```

```

CASE PAPERWIDTH_80MM: sizeQR = 384
OTHER CASE: sizeQR = 384
END
// 2. Ejecutar procedimiento Java y obtener imagen en base64
base64QR is string // = GenerateQRCodeBase64(qrText, sizeQR)
// 3. Validar y decodificar
IF base64QR <> "" THEN
  res = Decode(base64QR, encodeBASE64)
ELSE
  // En caso de error, mostrar mensaje opcional (debug)
  Info("Error al generar QR como imagen")
END
RESULT res

```

Método AddQRLegacy

```

//!
//! @brief Genera comandos QR legacy (versión antigua, no utilizada)
//!
//! @param qr Contenido del QR
//!
//! @note Método legacy mantenido por compatibilidad.
//! No retorna valor, posiblemente incompleto.
//! Se recomienda usar AddQR en su lugar.
//!
PROCEDURE AddQRLegacy(qr is string)
buffer is Buffer

// Código QR
buffer += Charact(10) + Charact(10)
buffer += Charact(29) + Charact(40) + Charact(107) + Charact(3) + Charact(0) + Charact(49) + Charact(67)
) + Charact(8)
buffer += Charact(29) + Charact(40) + Charact(107) + Charact(3) + Charact(0) + Charact(49) + Charact(69)
) + Charact(48)
buffer += Charact(29) + Charact(40) + Charact(107) + Charact(Length(qr) + 3) + Charact(0) + Charact(49)
+ Charact(80) + Charact(48) + qr
buffer += Charact(29) + Charact(40) + Charact(107) + Charact(3) + Charact(0) + Charact(49) + Charact(81)
) + Charact(48)

```

Método AddQRLegend

```

//!
//! @brief Genera leyenda del QR con cada elemento en línea separada
//!
//! @param cl_CFE Objeto CFE con datos del comprobante
//! @return String con la leyenda del QR
//!
//! @note Formato vertical con cada elemento alineado a la derecha.
//! Incluye fecha emisor, URL verificación, estado IVA, código seguridad.
//!
PROCEDURE AddQRLegend(cl_CFE is CFE) : string
legend is string = ""

fechaResolucion is string = cl_CFE.resolucionDgi

legend += AlignRight("Fecha emisor:")
legend += AlignRight(fechaResolucion)
legend += AlignRight("Puede verificar")
legend += AlignRight("comprobante en")
legend += AlignRight("www.efactura.com.uy")
legend += AlignRight("IVA al día")
legend += AlignRight("Cód. Seg. " + cl_CFE.codSeg)

```

```
legend += Charact(10)
```

```
RESULT legend
```

Método AddQRLegendSingleLine

```
///
///! @brief Genera leyenda del QR en una sola línea
///!
///! @param cl_CFE Objeto CFE con datos
///! @return String con leyenda en formato horizontal
///!
///! @note Todos los elementos separados por guiones en una línea.
///! Útil cuando el espacio vertical es limitado.
///!
PROCEDURE AddQRLegendSingleLine(cl_CFE is CFE) : string
fechaResolucion is string = cl_CFE.resolucionDgi

// Construimos todo en una sola línea
localLegend is string
localLegend += "Fecha emisor: " + fechaResolucion
localLegend += " - Puede verificar comprobante en www.efactura.com.uy"
localLegend += " - IVA al día"
localLegend += " - Cód. Seg. " + cl_CFE.codSeg

// Finalizamos con salto de línea
RESULT localLegend + Charact(10)
```

Método AddQRLegendWrapByWords

```
///
///! @brief Genera leyenda del QR con wrap inteligente por palabras
///!
///! @param cl_CFE Objeto CFE con datos
///! @return String con leyenda formateada con wrap
///!
///! @note Corta el texto en líneas sin partir palabras.
///! Se ajusta automáticamente al ancho del papel configurado.
///! Formato preferido para mejor legibilidad.
///!
PROCEDURE AddQRLegendWrapByWords(cl_CFE is CFE) : string
localLegend is string

// Armamos la leyenda en una sola línea
localLegend += "Fecha emisor:" + cl_CFE.resolucionDgi
localLegend += " - Puede verificar comprobante en www.efactura.com.uy"
localLegend += " - IVA al día"
localLegend += " - Cód. Seg. " + cl_CFE.codSeg

// Determinar el ancho en caracteres
maxWrap is int = GetCharWidthForPrinter() - 2
IF maxWrap < 1 THEN maxWrap = 1

// Cortar el texto en palabras sin partirlas
wrappedLegend is string = WrapTextByWords(localLegend, maxWrap)

// Retornar
RESULT wrappedLegend
```

Método AddReceiver

```

//!
//! @brief Genera la sección del receptor del comprobante
//!
//! @param receiver Objeto Receptor con datos del cliente
//! @return String con datos del receptor centrados
//!
//! @note Solo incluye campos no vacíos.
//! Todo el bloque se muestra centrado.
//! Incluye nombre, documento, dirección, ciudad y país si están presentes.
//!
PROCEDURE AddReceiver(receiver is Receptor) : string

data is string = ""
IF receiver.nombre <> "" THEN
  data += receiver.nombre + Charact(10)
  IF receiver.documento <> "" THEN data += "Doc: " + receiver.documento + Charact(10)
  IF receiver.direccion <> "" THEN data += receiver.direccion + Charact(10)
  IF receiver.ciudad <> "" THEN data += receiver.ciudad + Charact(10)
  IF receiver.pais <> "" THEN data += receiver.pais + Charact(10)
END

RESULT AddCenteredText(data)

```

Método AddSeparatorLine

```

//!
//! @brief Genera una línea separadora de guiones
//!
//! @return String con línea de guiones del ancho del papel
//!
//! @note El ancho se determina por GetCharWidthForPrinter().
//! Útil para separaciones más marcadas que los puntos.
//!
PROCEDURE AddSeparatorLine() : string

RETURN RepeatString("-", GetCharWidthForPrinter()) + Charact(10)

```

Método AddTotals

```

//!
//! @brief Genera la sección de totales del comprobante
//!
//! @param cl_CFE Objeto CFE con datos del comprobante
//! @return String con todos los totales formateados
//!
//! @note Incluye todos los tipos de IVA, retenciones, créditos fiscales, etc.
//! Solo muestra líneas con valores diferentes de cero.
//! Cada línea con descripción a la izquierda y monto a la derecha.
//! El TOTAL se muestra al final en mayúsculas.
//!
PROCEDURE AddTotals(cl_CFE is CFE) : string

res is string = ""
res += AddSeparatorLine()

// No Gravado
IF cl_CFE.Totales.noGravado <> 0 THEN
  res += AlignColumns("No gravado:", FormatCurrency(cl_CFE.Totales.noGravado))
END

// Exportación
IF cl_CFE.Totales.exportacion <> 0 THEN

```



```

    res += AlignColumns("Exportacion:", FormatCurrency(cl_CFE.Totales.exportacion))
END

// Impuesto Percibido
IF cl_CFE.Totales.impuestoPercibido <> 0 THEN
    res += AlignColumns("Impuesto percibido:", FormatCurrency(cl_CFE.Totales.impuestoPercibido))
END

// IVA Suspenso
IF cl_CFE.Totales.ivaSuspenso <> 0 THEN
    res += AlignColumns("IVA suspenso:", FormatCurrency(cl_CFE.Totales.ivaSuspenso))
END

// Neto IVA mínimo
IF cl_CFE.Totales.netoIvaMinimo <> 0 THEN
    res += AlignColumns("Neto IVA minimo:", FormatCurrency(cl_CFE.Totales.netoIvaMinimo))
END

// Neto IVA básico
IF cl_CFE.Totales.netoIvaBasico <> 0 THEN
    res += AlignColumns("Neto IVA basico:", FormatCurrency(cl_CFE.Totales.netoIvaBasico))
END

// Neto IVA otra
IF cl_CFE.Totales.netoIvaOtra <> 0 THEN
    res += AlignColumns("Neto IVA otra:", FormatCurrency(cl_CFE.Totales.netoIvaOtra))
END

// IVA Mínimo
IF cl_CFE.Totales.ivaMinimo <> 0 THEN
    res += AlignColumns("IVA minimo:", FormatCurrency(cl_CFE.Totales.ivaMinimo))
END

// IVA Básico
IF cl_CFE.Totales.ivaBasico <> 0 THEN
    res += AlignColumns("IVA basico:", FormatCurrency(cl_CFE.Totales.ivaBasico))
END

// IVA Otra
IF cl_CFE.Totales.ivaOtra <> 0 THEN
    res += AlignColumns("IVA otra:", FormatCurrency(cl_CFE.Totales.ivaOtra))
END

// Retenido
IF cl_CFE.Totales.retenido <> 0 THEN
    res += AlignColumns("Retenido:", FormatCurrency(cl_CFE.Totales.retenido))
END

// Créditos Fiscales
IF cl_CFE.Totales.creditosFiscales <> 0 THEN
    res += AlignColumns("Creditos fiscales:", FormatCurrency(cl_CFE.Totales.creditosFiscales))
END

// No facturable
IF cl_CFE.Totales.noFacturable <> 0 THEN
    res += AlignColumns("No facturable:", FormatCurrency(cl_CFE.Totales.noFacturable))
END

// TOTAL
res += AlignColumns("TOTAL:", FormatCurrency(cl_CFE.total))

res += AddSeparatorLine()

RESULT res

```

Método AlignColumns

```

//!
//! @brief Alinea dos columnas (izquierda y derecha)
//!
//! @param leftText Texto para columna izquierda
//! @param rightText Texto para columna derecha
//! @return String con ambos textos alineados
//!
//! @note Calcula espacios necesarios considerando caracteres de control.
//! Útil para líneas con descripción y precio.
//! Garantiza al menos 1 espacio entre columnas.
//!
PROCEDURE AlignColumns(leftText is string, rightText is string) : string
lineWidth    is int    = GetCharWidthForPrinter()
spaces       is int    = lineWidth - Length(RemoveControlChars(leftText)) - Length(RemoveControlChars(
rightText))
IF spaces < 1 THEN spaces = 1
RESULT leftText + RepeatString(" ", spaces) + rightText + Charact(10)
    
```

Método AlignRight

```

//!
//! @brief Alinea texto a la derecha
//!
//! @param text Texto a alinear
//! @return String con padding izquierdo para alineación derecha
//!
//! @note Calcula padding basado en GetCharWidthForPrinter().
//! Si el texto es más largo que el ancho, no agrega padding.
//!
PROCEDURE AlignRight(text is string) : string
charWidth    is int    = GetCharWidthForPrinter()
padding       is int    = charWidth - Length(text)
IF padding < 0 THEN padding = 0

RESULT RepeatString(" ", padding) + text + Charact(10)
    
```

Método AlignText

```

//!
//! @brief Alinea tres columnas uniformemente
//!
//! @param col1 Texto columna izquierda
//! @param col2 Texto columna centro
//! @param col3 Texto columna derecha
//! @param maxLen Ancho total disponible
//! @return String con las tres columnas alineadas
//!
//! @note Divide el espacio en tres partes iguales.
//! col1 alineada izquierda, col2 centrada, col3 alineada derecha.
//!
PROCEDURE AlignText(col1 is string, col2 is string, col3 is string, maxLen is int) : string
colWidth is int = maxLen / 3
col1 = PadRight(col1, colWidth)
col2 = PadCenter(col2, colWidth)
col3 = PadLeft(col3, colWidth)
RESULT col1 + col2 + col3
    
```

Método BuildTicket

```

//!
//! @brief Método abstracto para construir el ticket completo
    
```

```

//!
//! @param cl_cfe Objeto CFE con datos del comprobante
//! @param correctionLevel Nivel de corrección QR (por defecto M)
//! @param moduleSize Tamaño del módulo QR (por defecto automático)
//! @return Buffer con el ticket completo
//!
//! @note DEBE ser sobrescrito en clases derivadas.
//! Genera error si se llama en la clase base.
//! Define la estructura y orden del ticket completo.
//!
PROCEDURE BuildTicket(cl_cfe is CFE<useful>, correctionLevel is int <useful> = QR_CORRECTION_M ,
moduleSize is int <useful> = -1 ): Buffer
// Método que cada subclase sobrescribirá
Error("BuildTicket must be implemented by a derived class.")
RESULT ""

```

Método CenteredLine

```

//!
//! @brief Centra una línea de texto usando espacios
//!
//! @param text Texto a centrar
//! @return String centrado con espacios
//!
//! @note No usa comandos ESC, solo padding con espacios.
//! El ancho se determina por GetMaxCharsPerLine().
//!
PROCEDURE CenteredLine(text is string) : string
maxChars is int = GetMaxCharsPerLine()
RETURN PadCenter(text, maxChars)

```

Método ClosePrinter

```

//!
//! @brief Finaliza la comunicación con la impresora
//!
//! @return Buffer con comandos de cierre
//!
//! @note Incluye 3 saltos de línea (~1.5cm papel).
//! Doble comando de corte para compatibilidad.
//! El segundo corte actúa como reset en algunas impresoras.
//!
PROCEDURE ClosePrinter() : Buffer
res is Buffer = ""

// Saltos de línea (equivalente a 3 líneas ~1.5cm)
res += Charact(10) + Charact(10) + Charact(10)

// Corte de papel (funciona como "reset" interno en algunas impresoras)
res += Charact(29) + Charact(86) + Charact(66) + Charact(0)

// Agregamos un *corte vacío* (hack): una línea en blanco y otro corte
res += Charact(10)
res += Charact(29) + Charact(86) + Charact(66) + Charact(0)

RESULT res

```

Constructor

```

//!
//! @brief Constructor de la clase

```

```

///!
///! @note Inicializa la instancia con valores por defecto.
///! printerWidth se establece en 576 (papel 80mm) por defecto.
///!
PROCEDURE Constructor()

```

Destructor

```

///!
///! @brief Destructor de la clase
///!
///! @note Libera recursos si los hubiera.
///! Actualmente vacío, disponible para extensiones futuras.
///!
PROCEDURE Destructor()

```

Método FormatCurrency

```

///!
///! @brief Formatea un valor numérico como moneda
///!
///! @param value Valor numérico a formatear
///! @return String con formato de moneda (2 decimales)
///!
///! @note Usa formato "0.2f" para garantizar 2 decimales.
///! No incluye símbolo de moneda, solo formato numérico.
///!
PROCEDURE FormatCurrency(LOCAL value is real) : string

RETURN NumToString(value, "0.2f")

```

Asignación de la propiedad SetPrinterWidth

```

///!
///! @brief Establece el ancho de la impresora
///!
///! @param width Ancho en puntos (384 o 576)
///!
///! @note Propiedad pública para configurar el ancho.
///! Afecta todos los cálculos de formato y alineación.
///!
PROCEDURE PUBLIC SetPrinterWidth(width is int)

printerWidth = width

```

Método GenerateQRLegend

```

///!
///! @brief Genera texto de leyenda del QR en formato simple
///!
///! @param cl_CFE Objeto CFE con datos
///! @return String con leyenda en una línea
///!
///! @note Formato compacto con todos los elementos separados por guiones.
///! Similar a AddQRLegendSingleLine pero más conciso.
///!
PROCEDURE GenerateQRLegend(cl_CFE is CFE) : string
RETURN "Fecha emisor: " + cl_CFE.resolucionDgi + " - " + ...

```

"Puede verificar comprobante en www.efactura.com.uy - IVA al día - Cód. Seg. " + cl_CFE.codSeg

Método GetCharWidthForPrinter

```

///  

///  

///  

///  

///  

///  

///  

///  

///  

///  

PROCEDURE GetCharWidthForPrinter() : int

SWITCH printerWidth
    CASE 384: RETURN 32
    CASE 576: RETURN 56
    OTHER CASE: RETURN 56 // Valor por defecto
END

```

Método GetMaxCharsPerLine

```

///  

///  

///  

///  

///  

///  

///  

PROCEDURE GetMaxCharsPerLine() : int

n_Chars is int

IF printerWidth = 384 THEN
    n_Chars = 32
ELSE
    n_Chars = 56
END

RESULT n_Chars

```

Método InitPrinter

```

///  

///  

///  

///  

///  

///  

///  

PROCEDURE InitPrinter(): Buffer
RETURN Charact(27) + Charact("@") // ESC @ - Initialize printer

```

Método PadCenter

```

//!
//! @brief Centra texto agregando espacios a ambos lados
//!
//! @param txt Texto a centrar
//! @param size Ancho total deseado
//! @return String centrado con espacios
//!
//! @note Distribuye espacios equitativamente.
//! Si hay espacio impar, el extra va a la derecha.
//!
PROCEDURE PadCenter(txt is string, size is int) : string
leftSpace   is int  = (size - Length(txt)) / 2
rightSpace  is int  = size - Length(txt) - leftSpace
RETURN RepeatString(" ", leftSpace) + txt + RepeatString(" ", rightSpace)

```

Método PadLeft

```

//!
//! @brief Alinea texto a la izquierda con padding derecho
//!
//! @param txt Texto a alinear
//! @param size Ancho total deseado
//! @return String con padding a la izquierda
//!
//! @note Agrega espacios al inicio para alcanzar el tamaño.
//! Si el texto es más largo que size, no se modifica.
//!
PROCEDURE PadLeft(txt is string, size is int) : string
RETURN RepeatString(" ", size - Length(txt)) + txt

```

Método PadRight

```

//!
//! @brief Alinea texto a la derecha con padding izquierdo
//!
//! @param txt Texto a alinear
//! @param size Ancho total deseado
//! @return String con padding a la derecha
//!
//! @note Agrega espacios al final para alcanzar el tamaño.
//! Si el texto es más largo que size, no se modifica.
//!
PROCEDURE PadRight(txt is string, size is int) : string
RETURN txt + RepeatString(" ", size - Length(txt))

```

Método RemoveControlChars

```

//!
//! @brief Elimina caracteres de control del texto
//!
//! @param inputStr String a limpiar
//! @return String sin caracteres de control
//!
//! @note Mantiene solo ASCII 32-126 y tab/CR/LF.
//! Crítico para evitar problemas de impresión con caracteres especiales.
//! Usado en cálculos de alineación para ignorar comandos ESC.
//!
PROCEDURE RemoveControlChars(inputStr is string) : string

```

```

cleanStr is string = ""
FOR i = 1 _TO_ Length(inputStr)
    asciiVal is int = Asc(Middle(inputStr, i, 1))
    IF ( (asciiVal >= 32 AND asciiVal <= 126) OR asciiVal=9 OR asciiVal=10 OR asciiVal=13 ) THEN
        cleanStr += Middle(inputStr, i, 1)
    END
END
RESULT cleanStr

```

Método RepeatString

```

//!
//! @brief Repite una cadena un número determinado de veces
//!
//! @param c Cadena a repetir
//! @param times Número de repeticiones
//! @return String con la cadena repetida
//!
//! @note Sobrescribe función WLanguage del mismo nombre.
//! Optimización pendiente: usar sintaxis _TO_ en el bucle.
//!
PROCEDURE RepeatString(c is string, times is int) : string
res is string = ""
FOR i = 1 _TO_ times
    res += c
END
RESULT res

```

Método WrapText

```

//!
//! @brief Corta texto en líneas de longitud fija
//!
//! @param fullText Texto completo a cortar
//! @param maxChars Máximo de caracteres por línea
//! @return String con texto cortado en líneas
//!
//! @note Corta en posición exacta, puede partir palabras.
//! Cada línea termina con salto de línea.
//! Para no partir palabras, usar WrapTextByWords.
//!
PROCEDURE WrapText(fullText is string, maxChars is int) : string
localOutput is string = ""
localPos is int = 1
WHILE localPos <= Length(fullText)
    chunk is string = ExtractString(fullText, localPos, maxChars)
    localOutput += chunk + Charact(10)
    localPos += maxChars
END
RESULT localOutput

```

Método WrapTextByWords

```

//!
//! @brief Corta texto en líneas sin partir palabras
//!
//! @param fullText Texto completo a procesar
//! @param maxChars Máximo de caracteres por línea
//! @return String con texto formateado respetando palabras completas

```

```

//!
//! @note Algoritmo inteligente que no corta palabras.
//! Si una palabra no cabe, pasa a la siguiente línea.
//! Preserva la legibilidad del texto en el ticket.
//! Preferido sobre WrapText para textos largos.
//!
PROCEDURE WrapTextByWords(fullText is string, maxChars is int) : string

splitWords  is array of string  = StringSplit(fullText, " ")
localLine   is string           = ""
localOutput is string           = ""

FOR EACH word OF splitWords
  // Si la palabra + 1 espacio + lo que ya tenemos en línea sobrepasa el límite:
  IF (Length(localLine) + 1 + Length(word)) > maxChars THEN
    // Guardamos la línea actual y empezamos una nueva
    IF localLine <> "" THEN
      localOutput += localLine + Charact(10)
    END
    localLine = word
  ELSE
    // Agregamos la palabra a la línea actual
    IF localLine = "" THEN
      // Si está vacía, ponemos la palabra sin espacio inicial
      localLine = word
    ELSE
      // De lo contrario, agregamos un espacio y la palabra
      localLine += " " + word
    END
  END
END

// Al final, si quedó algo en la línea, lo agregamos
IF localLine <> "" THEN
  localOutput += localLine + Charact(10)
END

RESULT localOutput

```


Parte 4

Conjunto de procedimientos

Factory

Código

Procedimiento global GetCFETicketBuilder

```

///!
///! @brief Devuelve una instancia del constructor de ticket correspondiente al tipo de CFE solicitado.
///!
///! @details Este procedimiento actúa como **factory method**, devolviendo el objeto apropiado
///! de tipo derivado de `BaseCFETicketBuilder` según el tipo de comprobante fiscal electrónico (CFE).
///! En esta versión, solo se soporta el tipo 101 (eTicket - Consumo Final).
///!
///! @param tipoCFE Tipo de comprobante fiscal electrónico (por ejemplo: 101 = eTicket).
///! @param printerWidth Ancho del papel de impresión en milímetros (por ejemplo: 50 o 80).
///!
///! @return Objeto derivado de `BaseCFETicketBuilder` correspondiente al tipo de CFE solicitado.
///! Si el tipo no está soportado, retorna `Null`.
///!
///! @exception Muestra un mensaje de error si el tipo de CFE no está soportado.
///!
///! @note Este procedimiento simplifica la obtención del builder adecuado,
///! permitiendo centralizar la lógica de creación de tickets en un único punto.
///!
///! @code
///! // Ejemplo de uso:
///! builder is BaseCFETicketBuilder = GetCFETicketBuilder(101, 80)
///! IF builder <> Null THEN
///!     bufferTicket is Buffer = builder.BuildTicket(cfeObj)
///!     SocketWrite("PrinterSocket", bufferTicket)
///! ELSE
///!     Trace("Tipo de CFE no soportado")
///! END
///! @endcode
///!
///! @see ETicketBuilder Clase derivada para el tipo 101 (eTicket - Consumo Final).
///! @see BaseCFETicketBuilder Clase base abstracta que define la estructura general de un ticket.
///!
PROCEDURE GetCFETicketBuilder(tipoCFE is int, printerWidth is int) : BaseCFETicketBuilder

SWITCH tipoCFE
    CASE 101
        builder is object ETicketBuilder = new ETicketBuilder
        builder.SetPrinterWidth = printerWidth
        RESULT builder

    OTHER CASE
        Error("Tipo de CFE no soportado: " + tipoCFE)
        RESULT Null
END

```

Parte 5

Tabla de contenido

Proyecto ImpresionMobile

Parte 1	Proyecto	2
	Información general	2
	Lista de elementos del proyecto	2
	Lista de Carpetas Personales	2
	Código	3
	Errores, advertencias y mensajes de información	3
	Estadísticas de código	3
Parte 2	Ventana WINDEV	6
	WIN_PrintOnAndroid	6
	Imagen	6
	Información general	6
	Información sobre los controles	8
	Código	12
	Código de los controles	12
	Procedimientos	13
Parte 3	Clase	16
	Totales	16
	Código	16
	Emisor	17
	Código	17
	Detalles	18
	Código	18
	Receptor	19
	Código	19
	ETicketBuilder	20
	Código	20
	CFE	22
	Código	22
	BaseCFETicketBuilder	23
	Código	23
Parte 4	Conjunto de procedimientos	42
	Factory	42
	Código	42

