

```

// [Authorize]
[Route("api")]
[ApiController]
1 referencia
public class EnvioFacturaController : Controller
{
    readonly IGeneratePdf _generatePdf;
    private readonly string _connectionString;

    0 referencias
    public EnvioFacturaController(IGeneratePdf generatePdf, IConfiguration _configuration)
    {
        _generatePdf = generatePdf;
        _connectionString = _configuration.GetConnectionString("OracleConnection");

        Globales.AdminUser = _configuration.GetValue<string>("correoFactura");
        Globales.AdminPassWord = _configuration.GetValue<string>("facturaPassWord");
        Globales.SMTPName = _configuration.GetValue<string>("SMTPName");
        Globales.SMTPPort = _configuration.GetValue<int>("SMTPPort");
    }

    OracleCommand cmd = new OracleCommand();

```

METODO PARA GENERAR FACTURA ELECTRONICA EN C#

```

//Metodo EnviarFacturaElectronica
[Route("EnviarFacturaElectronica")]
[HttpGet]
0 referencias
public async Task<Response> EnviarFacturaElectronica()
{
    FacturaElectronica factuPendientes = facturasPendientes();

    Factura factura = new Factura();
    factura.Encabezado = facturaEncabezado(factuPendientes);
    factura.Detalle = detalleFactura(factuPendientes);

    if (factuPendientes.NO_FACTU != null)
    {
        try
        {
            string nombrePdf = "", subjet = "", body = "", msg = "";
            byte[] pdf = null, array = null;

            if (factura.Encabezado.IND_ANULADA != "N")
            {
                subjet = $"Anulación de Factura Electronica No. {factuPendientes.NO_FACTU} {factura.Encabezado.NOMBRE_CORTO}";
                factura.Encabezado.TEXTO_ANULADO = "ANULADO";
                body = string.Format(@"<p>Estimado Proveedor</p>
                <p>Adjunto la Factura Electronica ANULADA de número <strong>{factuPendientes.NO_FACTU}</strong> </p>

                <p>NOTAS:</p>
                <p>**Por favor, NO responda a este mensaje, es un envío automático.** </p>");
            }
            else
            {
                //NOTA: unir todo en el de anulado y el de no anulado
                subjet = $"Envío Factura Electronica {factura.Encabezado.NOMBRE_CORTO} / {factura.Encabezado.NOMBRE_CLIENTE}";
                factura.Encabezado.TEXTO_ANULADO = "";
                string cssStyle = "border: 1px solid black; background: #008f39; color: white;";
                string style = "border-collapse: collapse; border: 1px solid black; text-align:center;";

```

```

API_Services.Controllers.EnvioFacturaController  EnviarFacturaElectronica()
body = string.Format(@"$"
<p>Adjunto la Factura Electronica número <strong>{factura.Encabezado.NO_FISICO}-{factura.Encabezado.SERIE_FISICO}</strong> </p>
<p>Buen día le saludamos de Hierro del Rayo, adjunto encuentra la siguiente factura:</p>

<table style='{style}'>
<thead style='{cssStyle}'>
<tr>
<th style='{style}' width='150'>No.Cliente</th>
<th style='{style}' width='300'>Nombre del Cliente</th>
<th style='{style}' width='100'>Fecha Factura</th>
<th style='{style}' width='150'>Correlativo</th>
<th style='{style}' width='200'>No.Factura</th>
<th style='{style}' width='200'>Serie Factura</th>
<th style='{style}' width='250'>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td style='{style}'>{factura.Encabezado.NO_CLIENTE_FA}</td>
<td style='{style}'>{factura.Encabezado.NOMBRE_CLIENTE}</td>
<td style='{style}'>{factura.Encabezado.FECHA}</td>
<td style='{style}'>{factura.Encabezado.CORRELATIVO_INTERNO}</td>
<td style='{style}'>{factura.Encabezado.NO_FISICO}</td>
<td style='{style}'>{factura.Encabezado.SERIE_FISICO}</td>
<td style='{style}'>{factura.Encabezado.S_MONEDA} &nbsp;{factura.Encabezado.TOTAL}</td>
</tr>
</tbody>
</table>

<p>Confirmación de recibido, envío de contraseña de pago digital, o algún inconveniente referente a esta factura contactar a: </p>
{factura.Encabezado.NOMBRE_ASESOR} <br>
<a href='https://api.whatsapp.com/send?phone=502{factura.Encabezado.TELEFONO_ASESOR}' target='_blank'>Chatear en WhatsApp</a><br><br>

<p>NOTAS:</p>
<p>**POR FAVOR NO RESPONDER ESTE CORREO** </p>
<p>Este mensaje de correo electrónico se originó de {factura.Encabezado.FIRMA_EMPRESA}, puede contener información confidencial o legalmente
protegida y está destinado únicamente para el uso del destinatario(s) previsto. Cualquier divulgación, difusión, distribución, copia o la toma de
cualquier acción basada en la información aquí contenida está prohibida. Si recibe este mensaje por error, favor de notificarlo al remitente de
inmediato y destruirlo de su sistema con sus adjuntos </p>

```

```

var Options = new ConvertOptions
{
    PageOrientation = Orientation.Portrait,
    PageSize = Size.Letter
};
_generatePdf.SetConvertOptions(Options);

if (factura.Encabezado.NO_CIA == "10")
{
    if (factura.Encabezado.IND_EXPORTACION == "S")
    {
        array = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronicaExportacion.cshtml", factura);
    }
    else
    {
        array = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronica.cshtml", factura);
    }
}
else if (factura.Encabezado.NO_CIA == "04")
{
    array = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronica.cshtml", factura);
}
else if (factura.Encabezado.NO_CIA == "02")
{
    array = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronica02.cshtml", factura);
}
else if (factura.Encabezado.NO_CIA == "20")
{
    array = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronica20.cshtml", factura);
}

if (factura.Encabezado.IND_ANULADA == "N")
{
    System.IO.File.WriteAllBytes("Factura Electronica " + factura.Encabezado.CORRELATIVO_INTERNO + "-" + factura.Encabezado.NO_FACTU + "-" +
factura.Encabezado.SERIE_FISICO + ".pdf", array);
    nombrePdf = "Factura Electronica " + factura.Encabezado.CORRELATIVO_INTERNO + "-" + factura.Encabezado.NO_FACTU + "-" +
factura.Encabezado.SERIE_FISICO + ".pdf";
}

```

```

        System.IO.File.WriteAllBytes("Factura Electronica " + factura.Encabezado.CORRELATIVO_INTERNO + "-" + factura.Encabezado.NO_FACTU + "-" +
            factura.Encabezado.SERIE_FISICO + ".pdf", array);
        nombrePdf = "Factura Electronica " + factura.Encabezado.CORRELATIVO_INTERNO + "-" + factura.Encabezado.NO_FACTU + "-" +
            factura.Encabezado.SERIE_FISICO + ".pdf";
    }
    else
    {
        System.IO.File.WriteAllBytes("Factura Electronica ANULADA " + factura.Encabezado.CORRELATIVO_INTERNO + "-" + factura.Encabezado.NO_FACTU + "-" +
            factura.Encabezado.SERIE_FISICO + ".pdf", array);
        nombrePdf = "Factura Electronica ANULADA " + factura.Encabezado.CORRELATIVO_INTERNO + "-" + factura.Encabezado.NO_FACTU + "-" +
            factura.Encabezado.SERIE_FISICO + ".pdf";
    }

    MemoryStream memoryStream = new MemoryStream();

    if (factura.Encabezado.NO_CIA == "10")
    {
        if (factura.Encabezado.IND_EXPORTACION == "S")
        {
            pdf = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronicaExportacion.cshtml", factura);
        }
        else
        {
            pdf = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronica.cshtml", factura);
        }
    }
    else if (factura.Encabezado.NO_CIA == "04")
    {
        pdf = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronica.cshtml", factura);
    }
    else if (factura.Encabezado.NO_CIA == "02")
    {
        pdf = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronica02.cshtml", factura);
    }
    else if (factura.Encabezado.NO_CIA == "20")
    {
        pdf = await _generatePdf.GetByteArray("Views/Employee/FacturaElectronica20.cshtml", factura);
    }

```

```

        var pdfStream = new System.IO.MemoryStream();
        pdfStream.Write(pdf, 0, pdf.Length);
        pdfStream.Position = 0;
        memoryStream.Position = 0;

        string tipo = $"Factura Electronica {factura.Encabezado.NOMBRE_CORTO}";
        List<string> listaCorreos2 = new List<string>();
        List<string> listaCorreos3 = new List<string>();
        List<string> listaEmailsDistribucion = new List<string>();
        ListEmailsDistribucion = listaCorreosDistribucion(factuPendientes);
        await MailHelper.SendMail(correoCliente(factuPendientes), listaCorreos2, listaCorreos3, listaEmailsDistribucion, sujet, body, tipo, pdfStream,
            nombrePdf);

        if (factura.Encabezado.IND_ANULADA != "N")
        {
            msg = $"Factura {factura.Encabezado.NOMBRE_CORTO} ANULADA No. {factuPendientes.NO_CIA} enviado con éxito";
        }
        else
        {
            msg = $"Factura {factura.Encabezado.NOMBRE_CORTO} No. {factuPendientes.NO_CIA} enviado con éxito";
        }

        ActualizarEstado(factuPendientes);

        return new Response
        {
            IsSuccess = true,
            Message = msg
        };
    }
    catch (Exception ERROR)
    {
        return new Response
        {
            IsSuccess = false,
            Message = "ERROR: " + ERROR.Message,
        };
    }

```

Este código es una función asincrónica llamada **"EnviarFacturaElectronica"**, que devuelve un objeto **"Response"**. La función hace uso de una clase llamada **"FacturaElectronica"**, que tiene un método llamado **"facturasPendientes"**, el cual retorna las facturas pendientes a enviar. La función crea una nueva instancia de la clase **"Factura"**, estableciendo sus propiedades **"Encabezado"** y **"Detalle"** a partir de los resultados de los métodos **"facturaEncabezado"** y **"detalleFactura"**, respectivamente.

Después, la función verifica si la propiedad **"NO_FACTU"** de la factura pendiente es diferente de **"null"**. Si es así, se establecen ciertas variables para construir un correo electrónico con la factura

anulada adjunta. Si la factura no está anulada, se establecen diferentes variables para construir un correo electrónico con la factura adjunta.

La función luego construye el cuerpo del correo electrónico utilizando un formato de cadena, utilizando las propiedades de la factura "Encabezado" para llenar la información. También se construye una tabla en HTML para mostrar los detalles de la factura. Al final, se agregan algunas notas y una descarga de responsabilidad al correo electrónico.

En general, esta función envía facturas electrónicas pendientes a los clientes a través de correo electrónico. Si la factura está anulada, se envía un correo electrónico diferente para informar a los clientes sobre la anulación.

METODO PARA EL ENCABEZADO DE LAS FACTURAS

```
1 referencia
public FacturaEletronica facturaEncabezado(FacturaEletronica fe)
{
    string ppath_img = "", img_hiero = "", FIRMA_EMPRESA = "";
    string HoraPdf = DateTime.Now.ToString("dd-MM-yyyy");
    string TimePdf = DateTime.Now.ToString("HH:mm:ss");
    Double TotalDocumento = 0;
    var connection = new OracleConnection(_connectionString);
    FacturaEletronica obj = new FacturaEletronica();
    connection.Close();
    connection.Open();
    cmd.CommandType = CommandType.Text;
    cmd.Connection = connection;
    string queryEncabezado = @"$select
                                no_cia,
                                centro,
                                tipo_doc,
                                no_factu,
                                fisico,
                                no_fisico,
                                serie_fisico,
                                grupo_fa,
                                no_cliente_fa,
                                nombre_cliente,
                                s_moneda,
                                dia,
                                mes,
                                anio,
                                fecha,
                                total,
                                concepto,
                                moneda,
                                direccion,
                                observ1,
                                nit,
                                no_pedido,
                                dias_credito,
                                no_vendedor,
```

```

        nombre_asesor,
        telefono_asesor
        from vw_enviofacturas where no_cia = '{fe.NO_CIA}' and no_factu = '{fe.NO_FACTU}'
        --from vw_enviofacturas where no_cia = '10' and no_factu = '1604600115';

cmd.CommandText = queryEncabezado;
OracleDataReader dtr = cmd.ExecuteReader();

while (dtr.Read())
{
    TotalDocumento += Double.Parse(dtr["TOTAL"].ToString());

    if (dtr["NO_CIA"].ToString() == "10")
    {
        ppath_img = "http://hierrodelrayo.net/img/LogoCORPACAM.png";
        img_hiero = "http://hierrodelrayo.net/img/hierrodelrayo58x170.png";
        FIRMA_EMPRESA = "Corporación Acerera Centroamericana, S.A.";
    }
    else if (dtr["NO_CIA"].ToString() == "04")
    {
        ppath_img = "http://hierrodelrayo.net/img/LogoBODEASSA.png";
        img_hiero = "";
        FIRMA_EMPRESA = "BODEASSA, S.A.";
    }
    else if (dtr["NO_CIA"].ToString() == "20")
    {
        ppath_img = "http://hierrodelrayo.net/img/LogoDIAHRSA.png";
        img_hiero = "";
        FIRMA_EMPRESA = "DISTRIBUIDORA ACERERA HR, S.A. DE CV";
    }
    else if (dtr["NO_CIA"].ToString() == "02")
    {
        ppath_img = "http://hierrodelrayo.net/img/logoTranstemasa.png";
        img_hiero = "";
        FIRMA_EMPRESA = "Transtemasa, S.A.";
    }
}

```

```

TIPO_CAMBIO = dtr["TIPO_CAMBIO"].ToString(),
DESCRIPCION = dtr["DESCRIPCION"].ToString(),
OBSERVACIONES = dtr["OBSERVACIONES"].ToString(),
DESCUENTO = dtr["DESCUENTO"].ToString(),
TOT_LIN = dtr["TOT_LIN"].ToString(),
CORRELATIVO_INTERNO = dtr["CORRELATIVO_INTERNO"].ToString(),
TOTAL_BRUTO = dtr["TOTAL_BRUTO"].ToString(),
NIT_VENDEDOR = dtr["NIT_VENDEDOR"].ToString(),
RAZON_SOCIAL = dtr["RAZON_SOCIAL"].ToString(),
NOMBRE_VENDEDOR = dtr["NOMBRE_VENDEDOR"].ToString(),
DIRECCION_VENDEDOR = dtr["DIRECCION_VENDEDOR"].ToString(),
NUMERO_AUTORIZACION = dtr.IsDBNull("NUMERO_AUTORIZACION") ? "." : dtr["NUMERO_AUTORIZACION"].ToString(),
FECHAHORA_CERTIFICACION = dtr.IsDBNull("FECHAHORA_CERTIFICACION") ? "." : dtr["FECHAHORA_CERTIFICACION"].ToString(),
REGION = dtr["REGION"].ToString(),
REGIMEN = dtr["REGIMEN"].ToString(),
AGENTE_RETENEDOR = dtr["AGENTE_RETENEDOR"].ToString(),
IND_ANULADA = dtr["IND_ANULADA"].ToString(),
DIRECCION_ENTREGA = dtr.IsDBNull("DIRECCION_ENTREGA") ? "." : dtr["DIRECCION_ENTREGA"].ToString(),
NOMBRE_CORTO = dtr["NOMBRE_CORTO"].ToString(),
CAI = dtr["CAI"].ToString(),
NO_FISICO_LARGO = dtr["NO_FISICO_LARGO"].ToString(),
//FECHA_RESOLUCION = dtr["FECHA_RESOLUCION"].ToString(),
FECHA_RESOLUCION = dtr.IsDBNull("FECHA_RESOLUCION") ? "" : DateTime.Parse(dtr["FECHA_RESOLUCION"].ToString()).ToString("dd/MM/yyyy"),
TotalLetra = GetNumberLetter.NumeroALetras(Convert.ToDecimal(Math.Abs(TotalDocumento)), dtr["S_MONEDA"].ToString()),
img_empresa = ppath_img,
IMG_HIERORAYO = img_hiero,
FIRMA_EMPRESA = FIRMA_EMPRESA,
IND_EXPORTACION = dtr["IND_EXPORTACION"].ToString(),
NOMBRE_ASESOR = dtr["NOMBRE_ASESOR"].ToString(),
TELEFONO_ASESOR = dtr["TELEFONO_ASESOR"].ToString().Replace("-", "");
};

obj = objeto;
}
dtr.Close();
connection.Close();
connection.Dispose();
return obj;

```

Este método llamado "**facturaEncabezado**" devuelve un objeto "**FacturaElectronica**" y utiliza una consulta SQL para recuperar datos del encabezado de una factura electrónica de una base de datos Oracle.

Primero, se inicializan algunas variables para almacenar información como la imagen del logotipo de la empresa, la firma de la empresa, etc.

Luego, se abre una conexión a la base de datos Oracle utilizando la cadena de conexión proporcionada en el constructor de la clase.

A continuación, se construye una consulta SQL utilizando los parámetros NO_CIA y NO_FACTU del objeto FacturaElectronica proporcionado como parámetro de entrada.

Después, se ejecuta la consulta SQL utilizando un objeto OracleDataReader para recuperar los resultados de la consulta. Se utiliza un bucle while para iterar a través de los resultados devueltos por el objeto OracleDataReader y se utiliza cada registro para construir un nuevo objeto FacturaElectronica.

Finalmente, se cierra la conexión a la base de datos y se devuelve el objeto FacturaElectronica construido.

METODO PARA TRAER LOS DETALLES DE LAS FACTURAS.

```
1 referencia
public List<deta_envioFactura> detalleFactura(FacturaElectronica fe)
{
    var conexion = new OracleConnection(_connectionString);
    Decimal Subtotal = 0;
    Decimal totalDesc = 0;
    Decimal totalImpuesto = 0;

    conexion.Close();
    conexion.Open();
    cmd.CommandType = CommandType.Text;
    cmd.Connection = conexion;
    string comando = @"select
                        NO_TRANSA_MOV,
                        LINEA,
                        CODART,
                        CANTIDAD,
                        DESCRIPCION,
                        PRECIO,
                        TOTAL_DET,
                        IMP_INCLUIDO,
                        DESCUENTO_DET,
                        UNIDAD,
                        UNIDAD_MEDIDA,
                        BIEN_SERVICIO
                    from vw_detenviofacturas
                    where no_cia = '{fe.NO_CIA}'
                    and no_transa_mov = '{fe.NO_FACTU}'";

    cmd.CommandText = comando;
    OracleDataReader dtr = cmd.ExecuteReader();
    List<deta_envioFactura> listaDetalleFactua = new List<deta_envioFactura>();
```

```

OracleDataReader dtr = cmd.ExecuteReader();
List<deta_envioFactura> listaDetalleFactura = new List<deta_envioFactura>();

while (dtr.Read())
{
    var ob = new deta_envioFactura
    {
        NO_TRANSA_MOV = dtr["NO_TRANSA_MOV"].ToString(),
        LINEA = dtr["LINEA"].ToString(),
        CODART = dtr["CODART"].ToString(),
        CANTIDAD = dtr["CANTIDAD"].ToString(),
        DESCRIPCION = dtr["DESCRIPCION"].ToString(),
        TOTAL_DET = Convert.ToDecimal(dtr["TOTAL_DET"].ToString()).ToString("##,##0.00"),
        //TOTAL_DET = dtr["TOTAL_DET"].ToString(),
        IMP_INCLUIDO = Convert.ToDecimal(dtr["IMP_INCLUIDO"].ToString()).ToString("##,##0.00"),
        DESCUENTO_DET = Convert.ToDecimal(dtr["DESCUENTO_DET"].ToString()).ToString("##,##0.00"),
        UNIDAD = dtr["UNIDAD"].ToString(),
        PRECIO = fe.NO_CIA == "02" ? Convert.ToDecimal(dtr["PRECIO"].ToString()).ToString("##,##0.00") : Convert.ToDecimal(dtr["PRECIO"].ToString()).ToString("##,##0.00000"),
        //PRECIO = Convert.ToDecimal(dtr["PRECIO"].ToString()).ToString("##,##0.00000"),
        UNIDAD_MEDIDA = dtr["UNIDAD_MEDIDA"].ToString(),
        BIEN_SERVICIO = dtr["BIEN_SERVICIO"].ToString();
    };

    Subtotal += Convert.ToDecimal(dtr["TOTAL_DET"].ToString());
    totalDesc += Convert.ToDecimal(dtr["DESCUENTO_DET"].ToString());
    totalImpuesto += Convert.ToDecimal(dtr["IMP_INCLUIDO"].ToString());
    ob.SUB_TOTAL = Subtotal.ToString("##,##0.00");
    ob.Total_Descuento = totalDesc.ToString("##,##0.00");
    ob.TOTAL_IMPUESTO = totalImpuesto.ToString("##,##0.00");

    listaDetalleFactura.Add(ob);
}
dtr.Close();
conexion.Close();
return listaDetalleFactura;
}

```

El método **detalleFactura** realiza una consulta en la base de datos para obtener el detalle de una factura electrónica especificada en el parámetro **fe**. La consulta se realiza en la tabla **vw_detenviofacturas**, filtrando los registros por la compañía (**no_cia**) y el número de factura (**no_transa_mov**) proporcionados en el parámetro **fe**.

Luego, el método recorre el resultado de la consulta y crea objetos **deta_envioFactura** con la información obtenida de cada registro. También va sumando los valores de los campos **TOTAL_DET**, **DESCUENTO_DET** e **IMP_INCLUIDO** para calcular el subtotal, el total de descuentos y el total de impuestos de la factura.

el método recibe un objeto **FacturaElectronica** y consulta una base de datos Oracle para obtener los detalles de la factura correspondiente. Utiliza el mismo objeto **OracleConnection** y **OracleCommand** que el método anterior para conectarse a la base de datos y ejecutar la consulta.

Después de ejecutar la consulta y leer los resultados con un objeto **OracleDataReader**, el método itera sobre cada registro devuelto y crea un objeto **deta_envioFactura** para cada uno de ellos. Cada objeto creado se agrega a una lista de objetos **deta_envioFactura**, que es devuelta al final del método.

El método también calcula los totales de los campos **SUB_TOTAL**, **Total_Descuento** y **TOTAL_IMPUESTO**, sumando los valores correspondientes de cada registro y formateando los resultados como cadenas de texto.

Finalmente, el método devuelve una lista de objetos **deta_envioFactura** que contienen el detalle de la factura obtenido de la consulta.

