

esto es un documento de ayuda para poder facilitar la creacion de proyecto que utilizan **crystal report**, en este documento se explicara:

- librerias que tienes que importar para que el proyecto funcione correctamente
- archivos que tienes que modificar para que se puedan visualizar las pestañas
- manejo simple de informes y errores comunes al importar los elementos en la vista grafica

si al abrir la aplicacion te da un error de que no se encuentra el paquete para manejar .NET4.8, esto se debe a lo siguiente:

- no tienes instalado el paquete de desarrollador de .NET4.8, puedes instalarlo [pinchando aqui](#).

1. CRYSTAL REPORT | LIBRERIAS NECESARIAS

tienes que tener en cuenta que antes tienes que tener instalada la herramienta de crystal report, si no la tienes [pincha aqui](#). los paquetes que son necesarios para una implementacion de crystal report con una base de datos mysql son los siguientes:

- **crystaldecisions.crystalreport.engine** → 1.0.0
- **crystaldecisions.shared** → 1.0.0
- **crystaldecisions.windows.forms** → 1.0.0
- **crystalreports.engine** → 13.0.4003
- **crystalreports.shared** → 13.0.4003
- **crystalreports.wpf.viewer** → 13.0.4003
- **microsoft.codeanalysis.csharp** → 4.12.0
- **microsoft.codeanalysis.netanalyzers** → 9.0.0
- **mysql.data** → 9.1.0
- **system.diagnostics.diagnosticsource** → 9.0.1

disclaimer:

- recuerda que si ya tienes instalado algun paquete con la misma version no hace falta que lo instales.

2. CRYSTAL REPORT | ARCHIVOS MODIFICADOS

antes de modificar nada asegurate de que el proyecto funciona correctamente, si funciona bien no tendras que hacer este paso ya que se habra configurado automaticamente el proyecto; si no es asi, deberas dar doble clic en el nombre del proyecto dentro del explorador de soluciones. ejemplo:

- solucion "proyectoPersonas"/**proyectoPersonas** → tendremos que dar doble clic en el proyecto *proyectoPersonas* .

a continuacion tendras que ver las siguientes etiquetas ya que utiliza el formato xml y sustituirlo por las siguientes opciones

<!-- valores por defecto del proyecto -->

```

<PropertyGroup>

  <OutputType>WinExe</OutputType>

  <TargetFramework>net8.0-windows</TargetFramework>

  <Nullable>enable</Nullable>

  <ImplicitUsings>enable</ImplicitUsings>

  <UseWPF>true</UseWPF>

</PropertyGroup>

```

<!-- valores sustituidos -->

```

<PropertyGroup>

  <OutputType>WinExe</OutputType>

  <TargetFramework>net48-windows8.0</TargetFramework>

  <Nullable>enable</Nullable>

  <ImplicitUsings>enable</ImplicitUsings>

  <UseWPF>true</UseWPF>

  <PlatformTarget>AnyCPU</PlatformTarget>

<LangVersion>latest</LangVersion>

<ReferencePath>C:\\\\Program Files (x86)\\\\SAP BusinessObjects\\\\Crystal Reports for .NET Framework
4.0\\\\Common\\\\SAP BusinessObjects Enterprise XI 4.0\\\\win64_x64\\\\dotnet\\\\\\</ReferencePath>

</PropertyGroup>

```

- una vez cambiado los atributos del proyecto, limpia la solución, recompila, cierra el proyecto y vuélvelo a abrir para asegurarte de que todo se ha quedado configurado y se ha vuelto a cargar con los ajustes nuevos.

3. CRYSTAL REPORT | TRUCOS DE CREACION

esto es un paso opcional que puedes omitir perfectamente ya que no influye en la creación del proyecto pero si es de ayuda para evitar algunos errores comunes al estar trabajando en un proyecto que utiliza informes. los trucos son los siguientes:

- el **<Viewer:CrystalReportsViewer x:Name="reportViewer"/>** se debe crear en la ventana y no en un subelemento como puede ser un **<TabItem/>**: nos daría el error de que el tabitem no se ha inicializado de manera correcta.
- **crear un atributo y un método aparte para introducir los datos dentro del informe:** esto facilitaría la comprensión del código y evitaría errores de inicialización de componentes. el siguiente código es un ejemplo:

```

namespace proyectoPersonas {

    public partial class MainWindow : Window {

        DataTable tablaInforme; // -> creamos un elemento DataTable

```

```
// aqui ira el codigo adicional que necesites
```

```
public MainWindow () {  
    InitializeComponent();  
    //.... -> aqui ira el codigo que tu necesites para cargar los objetos  
    tablaInforme = new DataTable("DataTable1"); // asignamos al elemento tablaInforme el  
DataTable1 y sus datos  
    this.Loaded += (s, e) => cargarInforme(); // comprueba si todo es correcto y ejecuta la funcion  
cargarInforme  
}
```

```
private void cargarInforme () {  
    // reportViewer es el nombre de nuestro CrystalReportsViewer, que es necesario para  
visualizar el report/informe  
    if (reportViewer == null) {  
        MessageBox(" -> El visor de informes no esta inicializado");  
        return;  
    }
```

```
    // asignamos los valores de nuestro objeto como columnas del DataTable  
    tablaInforme.Columns.Add("Nombre");  
    tablaInforme.Columns.Add("Apellidos");  
    tablaInforme.Columns.Add("Edad");
```

```
    // crearemos una fila con los datos cada persona que haya en la lista personas  
    foreach (Persona auxPersona : listaPersonas) {  
        DataRow row = tablaInforme.NewRow();  
        row["Nombre"] = persona.Nombre;  
        row["Apellidos"] = persona.Apellidos;  
        row["Edad"] = persona.Edad;  
        tablaInforme.Rows.Add(row);  
    }
```

```
    // creamos el report de CrystalReport y le asignamos los valores de nuestra DataTable  
tablaInforme  
    CrystalReport1 report = new CrystalReport1();
```

```

        if (report.Database.Tables["DataTable1"] == null) {
            MessageBox.Show("Error: La tabla 'DataTable1' no existe en el informe.");
            return;
        }

        report.Database.Tables["DataTable1"].SetDataSource(tablaInforme);

        // comprobamos si el viewcore es null antes de asignar el reporte
        if (reportViewer.ViewerCore != null) {
            reportViewer.ViewerCore.ReportSource = report;
        } else {}
    }
}
}

```

- con estos simples pasos ya podras modificar este metodo para que te cree informes con cualquier lista de datos que necesites.
- **para poder utilizar <Viewer:CrystalReportsViewer x:Name="reportViewer"/>**, tendras que especificar en el proyecto a que te refieres con el <Viewer/>, a continuacion tienes la linea que tienes que agregar en tu MainWindow.xaml:

<!-- lienas que tienes que tener de normal -->

<Window x:Class="gestproJuanGarcia.MainWindow"

xmlns="<http://schemas.microsoft.com/winfx/2006/xaml/presentation>"

xmlns:x="<http://schemas.microsoft.com/winfx/2006/xaml>"

xmlns:d="<http://schemas.microsoft.com/expression/blend/2008>"

xmlns:mc="<http://schemas.openxmlformats.org/markup-compatibility/2006>"

xmlns:local="clr-namespace:gestproJuanGarcia" xmlns:xctk="<http://schemas.xceed.com/wpf/xaml/toolkit>"

mc:Ignorable="d"

Title="GestPro Juan Garcia" Height="450" Width="800"

Background="#FFEBECDE" UseLayoutRounding="True">

<!------->

<!-- la siguiente linea la tienes que agregar de la siguiente manera -->

<Window x:Class="gestproJuanGarcia.MainWindow"

xmlns:Viewer="clr-

namespace:SAPBusinessObjects.WPF.Viewer;assembly=SAPBusinessObjects.WPF.Viewer"

....>

- con estos pasos ya podras tener un `<Viewer:CrystalReportsViewer x:Name="reportViewer"/>` dentro de un `tabitem`
-