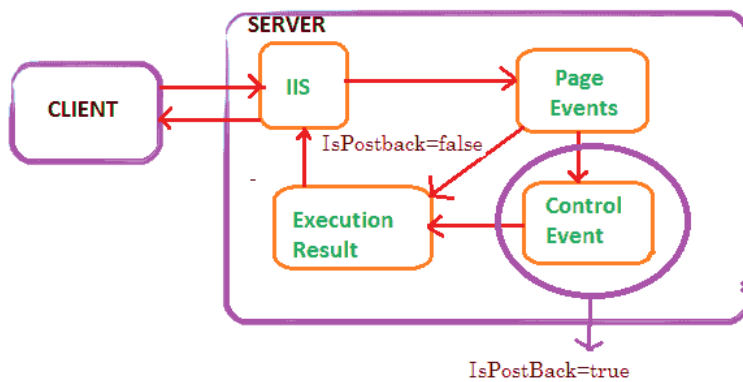


```
if (!Page.IsPostBack)
{
    llenarGv();
}
```

```
if (!Page.IsPostBack)
{
    nAlumno nalumno = new nAlumno();
    List<Alumno> ListaAlumnos = nalumno.Consultar();
    gvAlumnos.DataSource = ListaAlumnos;
    gvAlumnos.DataBind();
}
```



Is Postback se utiliza normalmente en la página `_load` evento para detectar si la página web se está generando debido a la devolución solicitada por un control de la página o si la página se está cargando por primera vez.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        // generate form;
    }
    else
    {
        //process submitted data;
    }
}
```

Enviar Id mediante el Url

```
Response.Redirect($"Details.aspx?id={id}");
```

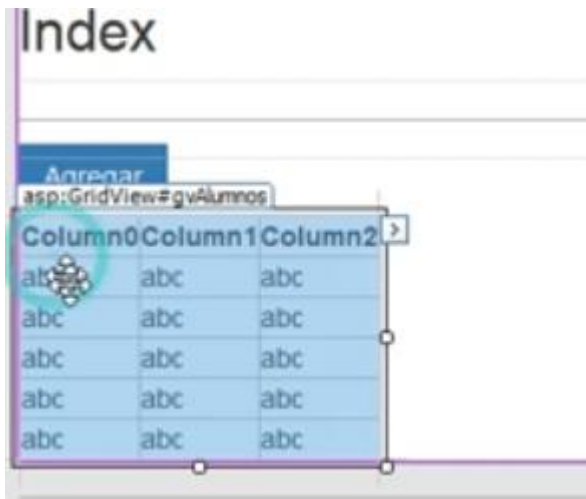
Recibir Id en Details

```
int id = Request.QueryString["id"] == null ? 1 : Convert.ToInt32(Request.QueryString["id"]);
```

1.- Agregar GridView Código

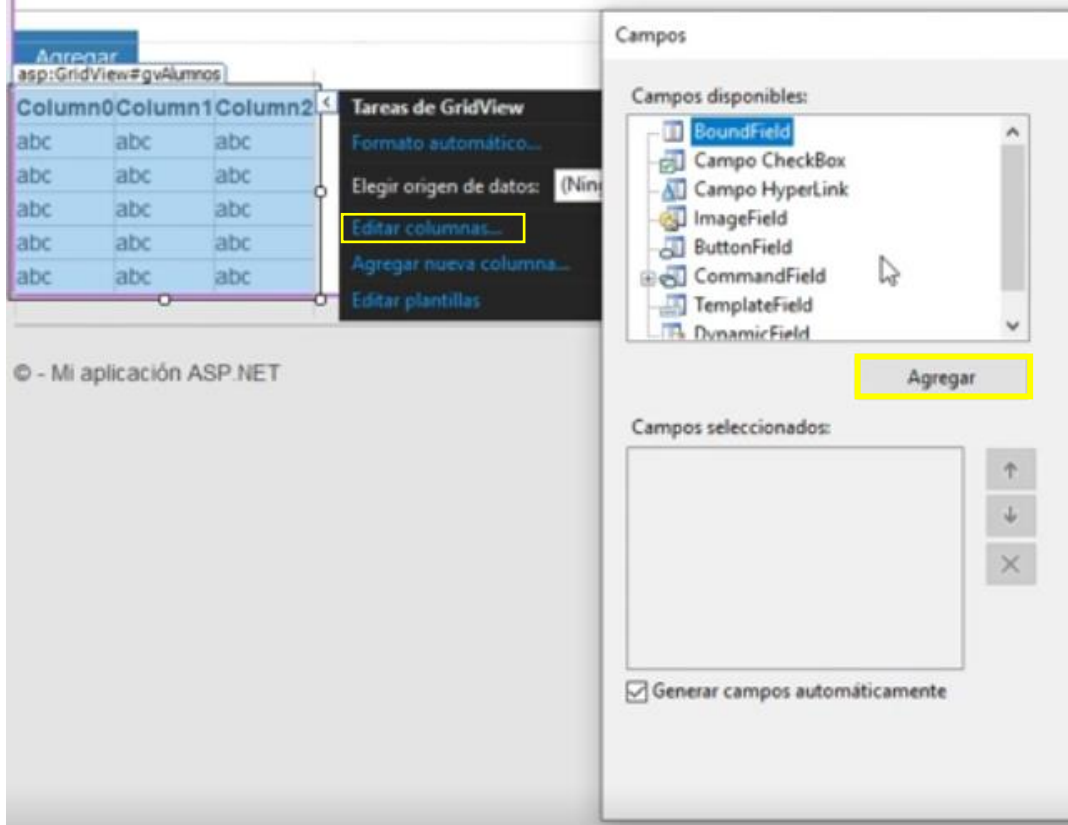
```
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <h2>Index</h2>
    <!-- Control asp Label para mensajes -->
    <hr />
    <p><a runat="server" href="Create.aspx" class="btn btn-primary">Agregar</a>
    <asp:GridView ID="gvAlumnos" runat="server"></asp:GridView>
</asp:Content>
```

GridView Diseño

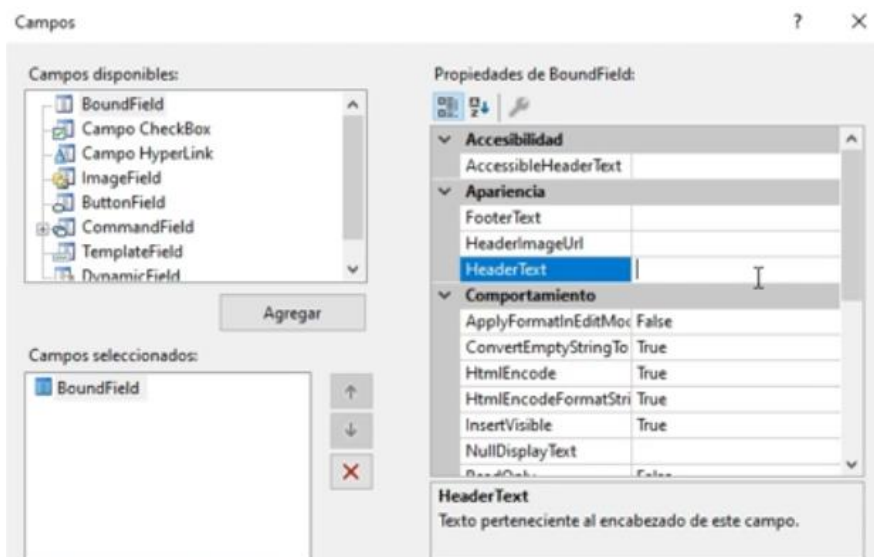


Configurar GridView

Index

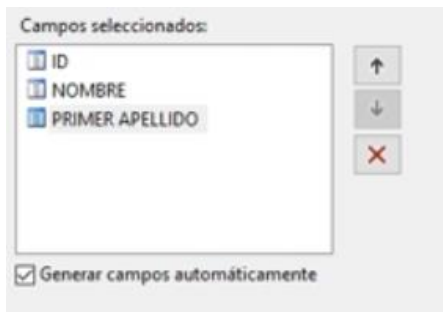


BoundField: campos Ligados a datos



Header Text: Establece el texto que se muestra en la sección de encabezado de la columna.

DataField: Establece el nombre del campo de datos que se va a enlazar al objeto BoundField. (cocar el nombre de la propiedad).



```
<asp:BoundField DataField="id" HeaderText="ID" />
<asp:BoundField DataField="nombre" HeaderText="NOMBRE" />
```

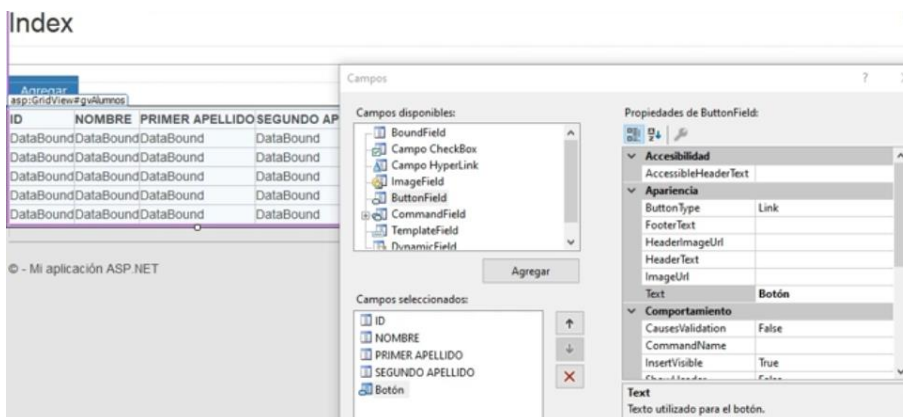
AutoGenerateColumns: Establece un valor que indica si las columnas se crean automáticamente.



```
gvAlumnos.DataSource = ListaAlumnos;
gvAlumnos.DataBind();
```

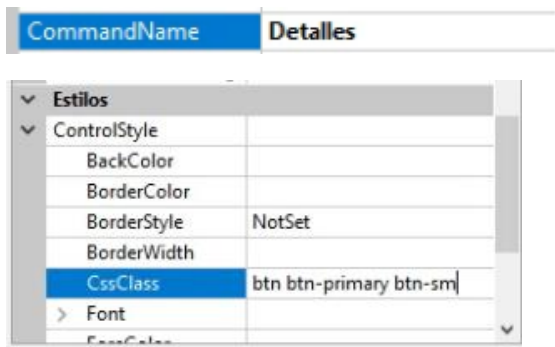
DataBind: Enlaza el origen de datos al control GridView.

Agregar botones



Text: nombre del botón.

CommandName: establece una cadena que representa la acción que se va a realizar cuando se hace clic en un botón.



ControlStyle: estilo del control.

```
<asp:ButtonField CommandName="Detalles" Text="Detalles">
<ControlStyle CssClass="btn btn-primary btn-sm" />
</asp:ButtonField>
```

Evento GridView



RowCommand (evento): Se produce cuando se hace clic en un botón en un control GridView.

```
gvAlumnos_RowCommand(object sender, GridViewCommandEventArgs e)
```

Sender: El que lo envía

GridViewCommandEventArgs: Argumentos (contiene datos de eventos).

Extraer el ID del alumno que se editara, consultara, eliminara. Desde el evento **RowCommand**.

CommandArgument: El valor que trae esta propiedad es el número del renglón que se hizo clic.

```
protected void gvAlumnos_RowCommand(object sender, GridView
{
    int NumRenglon = Convert.ToInt32(e.CommandArgument);
```

GridViewRow: Tipo de objeto del renglón.

Para extraer todo el renglón. (Para ubicar el número de renglón que se oprimió)

```
GridViewRow Renglon = gvAlumnos.Rows[NumRenglon];
```

Para extraer la Celda en la posición cero [0] (obtener el Id)

```
TableCell Celda = Renglon.Cells[0];
```

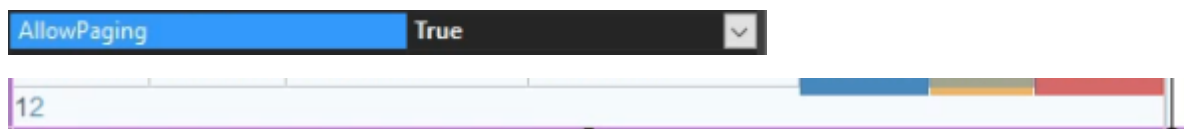
TableCell: tipo de variable celda.

```
protected void gvAlumnos_RowCommand(object sender, GridViewRowEventArgs e)
{
    int NumRenglon = Convert.ToInt32(e.CommandArgument);
    GridViewRow Renglon = gvAlumnos.Rows[NumRenglon];
    TableCell Celda = Renglon.Cells[0];
    int id = Convert.ToInt32(Celda.Text);
}
```

Para identificar el botón que se oprimió.

```
switch (e.CommandName)
{
    case "Detalles":
        Response.Redirect($"Details.aspx?id={id}");
        break;
    case "Editar":
        Response.Redirect($"Details.aspx?id={id}");
        break;
    case "Eliminar":
        Response.Redirect($"Delete.aspx?id={id}");
        break;
}
```

Para permitir la paginación



AllowPaging (propiedad): establece un valor que indica si la característica de paginación está habilitada.

Para definir el numero de datos a mostrar en el **GridView**

```
PageSize 1
```

PageSize (propiedad): establece el número de registros que se van a mostrar en una página de un control.

Para identificar cuando cambia de página mediante el evento.



PageIndexChanging (evento): Se produce cuando se hace clic en uno de los botones del localizador, pero antes de que el control GridView controle la operación de paginación.

Para llenar los elementos correspondientes a la página seleccionada.

```
protected void gvAlumnos_PageIndexChanging(object sender, GridViewPageEventArgs e)
{
    gvAlumnos.PageIndex = e.NewPageIndex;
}
```

Página que oprimieron a la cual se quieren ir.

NewPageIndex (propiedad): establece el índice de la nueva página que se mostrará en el control.

Para evitar que marque un error cuando trae un comando diferente.

```
switch (e.CommandName)
{
    case "Page":
```

```
if (e.CommandName == "Page")
{
    return;
}
```