https://github.com/j5anchez/DI04

https://github.com/j5anchez/DI05

https://github.com/j5anchez/DI-TE06

eliminar credenciales de usuario en windows:
ejecutar-> control keymgr.dll
buscar github

SUBIR
git init
git add .
git commit -m "initial commit"
git remote add origin https://github.com/j5anchez/repositorio.git
git push -u origin master
git push origin master

BAJAR
git clone https://github.com/j5anchez/repositorio.git

**Los 10 principios de usabilidad de Jakob Nielsen**

1- Visibilidad del estado del sistema
2- Adecuación entre el sistema y el mundo real.
3- Libertad y control por el usuario.
4- Consistencia y estándares.
5- Prevención de errores.
6- Reconocer mejor que recordar.
7- Flexiblidad y eficiencia de uso.
8- Estética y diseño minimalista.
9- Ayudar a los usuarios a reconocer, diagnosticar y solucionar los errores.
10- Ayuda y documentación.

## MODELO VISTA CONTROLADOR (MVC)

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

- El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La **Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

- El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

## El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

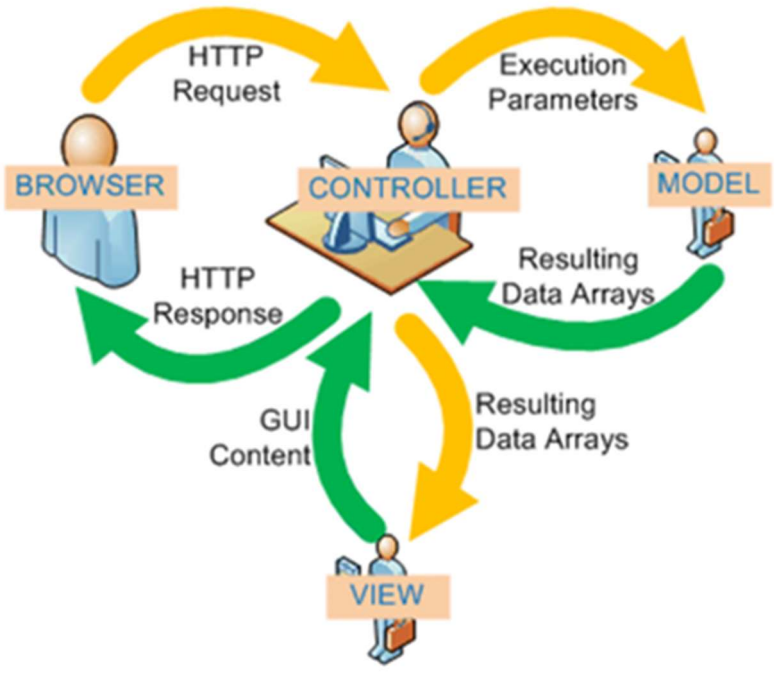
## El controlador es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega ( nueva_orden_de_venta )".

## Las vistas son responsables de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

El flujo que sigue el control generalmente es el siguiente:



1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

**Conectar con BD sin CS ni RV:**

Proyecto->Agregar origen de datos

Agregar conjunto de datos al proyecto (DataSet). Agregar TableAdapter. Nueva conexión y rellenar. En el form añadir DataGridView y seleccionar el origen de datos creado antes.

En properties->settings.settings se puede ver la cadena con el nombre del servidor:

DataSource=LAPTOP-QOH7Q921\MSSQLSERVER2; InitialCatalog=DASHBOARD; IntegratedSecurity=True

Data Source=LAPTOP-QOH7Q921\MSSQLSERVER2;Initial Catalog=DASHBOARD;User ID=DI;Password=1234

Para agregar mas elementos al cuadro de herramientas, boton derecho en windows forms y elegir elementos.

Para agregar instalador:

Agregar nuevo proyecto->setup project

Boton derecho en el directorio de application folder->add->resultados del proyecto

*Ejemplo para agregar control de usuario en la documentacion de microsoft*

Creación de un proyecto de Windows Forms

El primer paso es crear un proyecto de Windows forms.

En Visual Studio, en el menú Archivo, seleccione Nuevo > Proyecto.

Expanda Visual C# o Visual Basic en el panel izquierdo y, a continuación, seleccione Windows Desktop.

En el panel central, seleccione el tipo Windows proyecto Aplicación de Formularios.

Asigne al proyecto el nombre LookupControlWalkthrough y, a continuación, elija Aceptar.

El proyecto LookupControlWalkthrough se crea y se agrega al Explorador de soluciones.

Agregar un control de usuario al proyecto

Este tutorial crea un control de búsqueda a partir de un Control de usuario, por lo que debe agregar un elemento Control de usuario al proyecto LookupControlWalkthrough.

En el menú Proyecto, elija Agregar control de usuario.

Escriba LookupBox en el área Nombre y, a continuación, haga clic en Agregar.

El control LookupBox se agrega al Explorador de soluciones y se abre en el diseñador.

Diseño del control LookupBox

Para diseñar el control LookupBox, arrastre desde el Cuadro de herramientas hasta ComboBox la superficie de diseño del control de usuario.

Agregar el atributo de enlace de datos necesario

Para controles de búsqueda que admiten el enlace de datos, puede implementar LookupBindingPropertiesAttribute.

Cambie el control LookupBox a vista de código. (En el menú Ver, elija Código.)

Reemplace el código de LookupBox por lo siguiente:

```
using System.Windows.Forms;

namespace CS
{
   [System.ComponentModel.LookupBindingProperties("DataSource", "DisplayMember",
"ValueMember", "LookupMember")]
   public partial class LookupBox : UserControl
   {
      public object DataSource
      {
         get{ return comboBox1.DataSource; }
         set{ comboBox1.DataSource = value; }
      }

      public string DisplayMember
      {
         get{ return comboBox1.DisplayMember; }
         set{ comboBox1.DisplayMember = value; }
      }

      public string ValueMember
      {
         get{ return comboBox1.ValueMember; }
         set{ comboBox1.ValueMember = value; }
      }

      public string LookupMember
      {
```

```
        get{ return comboBox1.SelectedValue.ToString(); }
        set{ comboBox1.SelectedValue = value; }
    }



    public LookupBox()
    {
        InitializeComponent();
    }
  }
}
```

En el menú Compilar , elija Compilar solución.

Creación de un origen de datos a partir de la base de datos

En este paso se crea un origen de datos utilizando el Asistente para configuración de orígenes de datos basado en las tablas Customers y Orders de la base de datos de ejemplo Northwind.

Para abrir la ventana Orígenes de datos, en el menú Datos , haga clic en Mostrar orígenes de datos.

En la ventana Orígenes de datos, seleccione Agregar nuevo origen de datos para iniciar el Asistente para configuración del origen de datos.

Seleccione Base de datos en la página Elegir un tipo de datos de origen y luego haga clic en Siguiente.

En la página Elegir la conexión de datos realice una de las siguientes operaciones:

Si una conexión de datos a la base de datos de ejemplo Northwind está disponible en la lista desplegable, selecciónela.

Seleccione Nueva conexión para iniciar el cuadro de diálogo Agregar o modificar conexión.

Si su base de datos requiere una contraseña, seleccione la opción para incluir datos confidenciales y haga clic en Siguiente.

Haga clic en Siguiente en la página Guardar la cadena de conexión en el archivo de configuración de la aplicación.

Expanda el nodo Tablas en la página Elija los objetos de base de datos.

Seleccione las tablas Customers y Orders y después haga clic en Finalizar.

NorthwindDataSet se agrega al proyecto y las tablas Customers y Orders aparecen en la ventana Orígenes de datos.

Establecer la columna CustomerID de la tabla Orders para usar el control LookupBox

Dentro de la ventana Orígenes de datos puede establecer el control que se va a crear antes de arrastrar elementos a un formulario.

Abra Form1 en el diseñador.

Expanda el nodo Customers en la ventana Orígenes de datos.

Expanda el nodo Orders (incluido en el nodo Customers debajo de la columna Fax).

Haga clic en la flecha de lista desplegable en el nodo Orders y elija Detalles de la lista de control.

Haga clic en la flecha de lista desplegable en la columna CustomerID (del nodo Orders) y elija Personalizar.

Seleccione LookupBox de la lista de Controles asociados en el cuadro de diálogo Opciones de personalización de la interfaz de usuario de datos.

Haga clic en OK.

Haga clic en la flecha de lista desplegable en la columna CustomerID y elija LookupBox.

Adición de controles al formulario

Puede crear los controles enlazados a datos arrastrando elementos desde la ventana Orígenes de datos al Form1.

Para crear controles enlazados a datos en Windows Form, arrastre el nodo Orders desde la ventana Orígenes de datos al formulario Windows Form y compruebe que el control LookupBox se usa para mostrar los datos de la CustomerID columna.

Enlazar el control para buscar CompanyName en la tabla Customers

Para configurar los enlaces de búsqueda, seleccione el nodo Clientes principal en la ventana Orígenes de datos y arrástrelo al cuadro combinado de CustomerIDLookupBox en Form1.

Así configura el enlace de datos para mostrar el valor CompanyName de la tabla Customers a la vez que mantiene el valor CustomerID de la tabla Orders.

Ejecución de la aplicación

Presione F5 para ejecutar la aplicación.

Navegue por algunos registros y compruebe que CompanyName aparece en el control LookupBox.