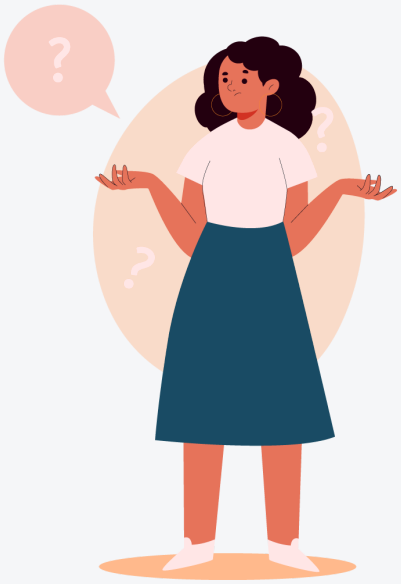
Ingreso al sistema y codificación de los Sub_Sistemas (Parte II)

Sitio:	Agencia de Habilidades para el Futuro	Imprimido por:	Eduardo Moreno
Curso:	Desarrollo de Sistemas Orientado a Objetos 1° D	Día:	lunes, 19 de mayo de 2025, 00:48
Libro:	Ingreso al sistema y codificación de los Sub_Sistemas (Parte II)		

Tabla de contenidos

Preguntas orientadoras

- ¿Cómo vinculo una consulta simple de MySQL con C#?
- ¿Es posible guardar un documento generado por el sistema?
- ¿Podemos ocultar objetos del formulario?
- ¿Un DataGridView puede contener datos de la base de datos?

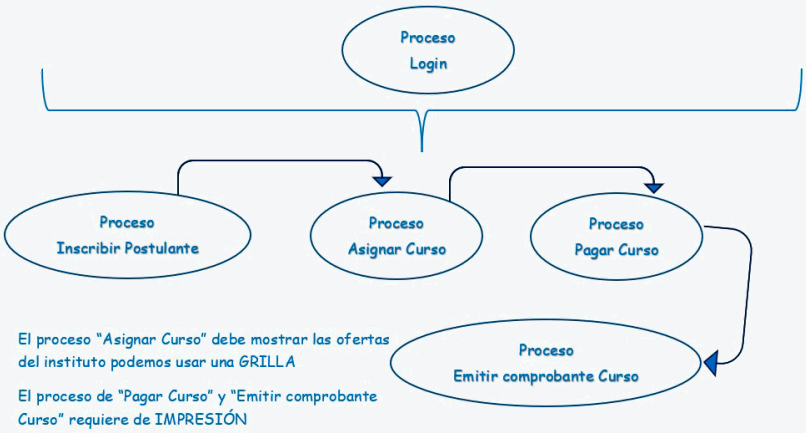


Introducción

En este módulo, continuaremos lo iniciado en la semana anterior, ingresando al sistema y codificando los subsistemas.

Ya tenemos el formulario Principal y el código del primer proceso, ahora, pasemos a dar prioridades a los restantes en función de las entradas y salidas.

Podemos bocetar a los procesos según la siguiente gráfica.



¡Importante!

Las flechas nos indican la prioridad en la programación.

¿Qué nos falta aprender? Cargar una grilla desde el entorno de datos e imprimir el diseño de un documento (acciones de los dos últimos procesos).

Vamos por la carga de la grilla ...

Carga de grillas desde el entorno de datos

Pasos para la carga

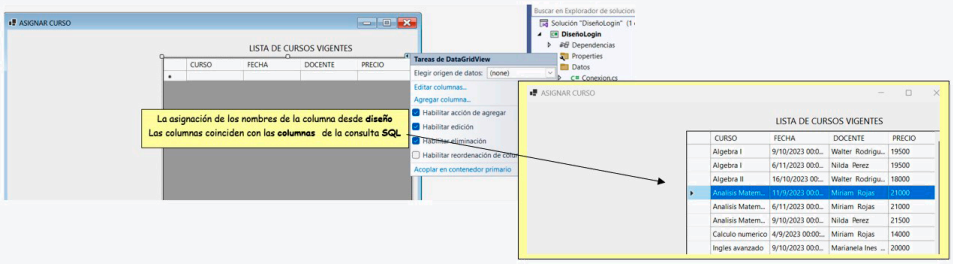
1. En primer lugar, podemos vincular a las columnas de una grilla (DataGridView) los atributos que resultan de la ejecución de una consulta SQL.
2. En este caso nos vinculamos a la base con una query y no con un store procedure.
3. La instancia a MySqlCommand utiliza en la ejecución un DataReader.
4. El DataReader tiene almacenado el resultado de la consulta SQL, cada una de las columnas se referencian con un número comenzando de cero; y las filas se recorren una a una mediante su lectura.
5. En el proceso de asignación de curso, bocetamos la pantalla con el uso de una grilla para que muestre la información relevante del curso y el docente a cargo.
6. Queremos que la grilla tenga la información cuando llamamos al formulario entonces, la codificación debe colocarse en el **evento de carga del formulario**, ese evento es el **Load()**.

Para tener en cuenta

Para ordenar las líneas de código colocamos la lógica en un procedimiento llamado **CargaGrilla()** y desde el load simplemente lo llamamos.

Para barrer todas las líneas de resultados se usa un ciclo While cuya condición es que el **read** tenga datos para leer.

El nombre de las columnas se asigna desde el diseño (¡volvé a la semana 4 para repasarlo!).



La codificación completa está en el documento **S10_AsignarGrilla.pdf.**

Continuamos con la programación y el sistema tiene como salida un documento que, además de quedar almacenado en la base de datos, debe permitir la salida para el usuario.

Veamos de una manera simple como imprimir un diseño de documentación

Diseño de documentos e impresión

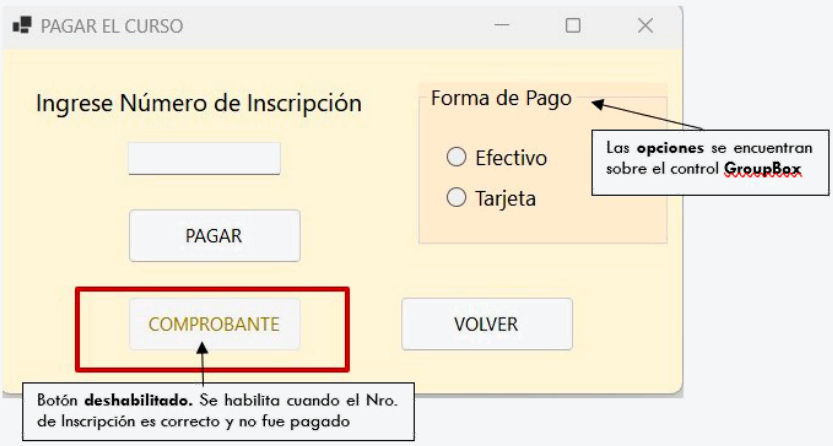
En el sistema de ejemplo que estamos desarrollando vamos a simular el comprobante del pago del curso.

Vamos a diseñar en el formulario nuestro comprobante utilizando los controles de la barra de herramientas teniendo en cuenta que los textos para los títulos de los datos y para el valor propiamente dicho de los mismos, son etiquetas que por sus características **no se pueden modificar** desde el exterior.

Una vez terminado el diseño, colocamos en un extremo un control Button con la leyenda **Imprimir** para que desencadene el código correspondiente. Se debe ocultar el botón antes de la impresión para que no aparezca en la salida del documento. La impresión convierte el documento en un pdf para ser almacenado.

El documento está plasmado en un formulario distinto al que lo genera.

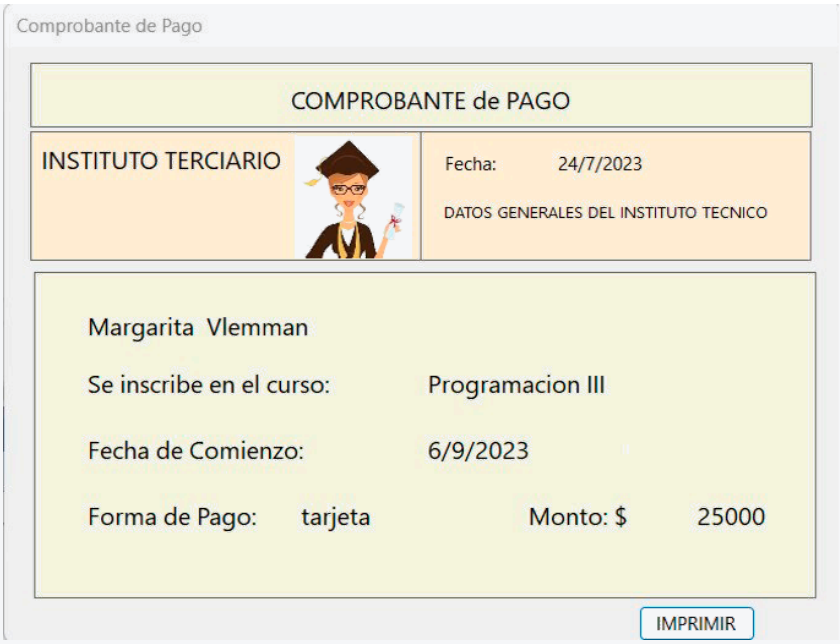
La ventana que desencadena el proceso es:



Se ingresa el Nro. De Inscripción y la forma de pago al hacer click en **Pagar** se habilita **Comprobante** siempre que el número exista y no se haya pagado la inscripción. La propiedad que habilita y deshabilita un objeto es **Enabled** y sus posibles valores son **True** o **False**.

Comprobante permite pasar a la ventana del diseño propiamente dicho con todos los datos significativos.

El diseño es el siguiente, y los datos los traemos con una consulta SQL desde la base de datos en el evento **load** del formulario que invoca el diseño. Usamos variables para colocarlos en las etiquetas.



Los marcos que se encuentran en el diseño se obtienen con el control **Panel**, el prefijo es **pnl**

Recorda que el diseño de las ventanas queda a tu criterio.

En el pdf **S10_Pagar** se encuentra el código de la ventana que solicita los datos para generar el documento y en el pdf **S10_Asignacion_al_diseño** está el código que inserta los datos en el diseño del documento y genera la impresión.

En resumen ...

En resumen

El manejo de la Grilla de datos la vimos en la semana 4, en esa ocasión su contenido lo generamos mediante clases cuyos datos se ingresaron desde el teclado. Esta semana esa carga de datos es a través de una consulta SQL desde una base de datos. El diseño que puede tener la grilla lo determinas usando la barra de propiedades del IDE de programación.

La creación de los documentos que genera el sistema depende de las características de la regla de negocios, y de los requerimientos del usuario. En el caso del Proyecto Integrador está el **carnet de socio** y el **comprobante de pago** de la cuota.

Estos documentos son salida (están en las postcondiciones de las plantillas de caso de uso), y para que se concrete esa acción es que usamos el conjunto se instrucciones vinculadas al **print**.

El Script que contiene la base de datos del sistema que estamos usando para la explicación debe ayudarte para crear la del sistema integrador. Nuevamente aparece en esta semana porque se fueron agregando elementos, y se adjuntó al final los procedimientos almacenados.

Acordate que si por alguna razón se almacenaron datos de manera incorrecta se puede borrar definitivamente la base con la sentencia **drop** y volver a cargarla en el sistema gestor.