

Capítulo

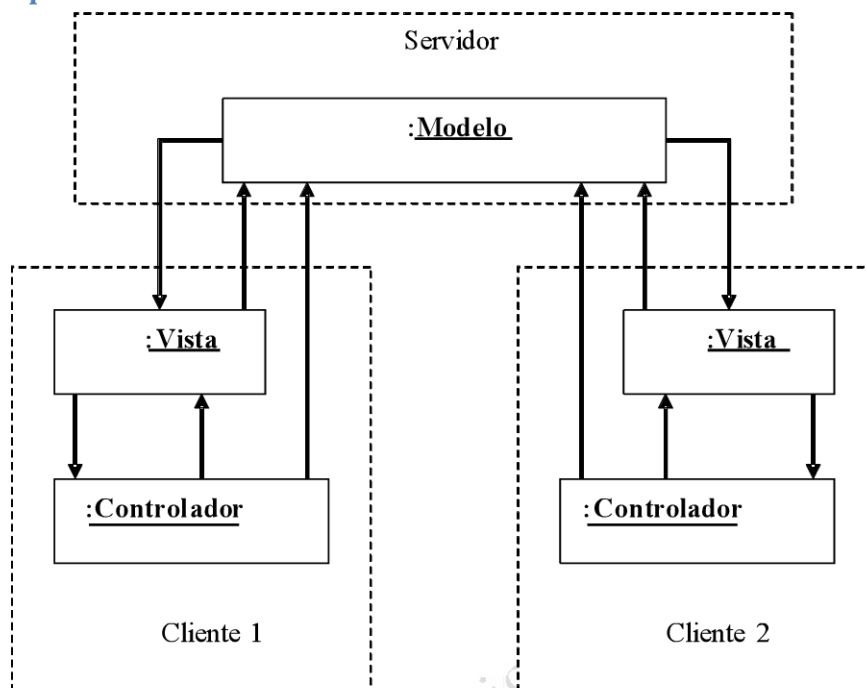
5

DIPLOMATURA EN PROGRAMACION EN .NET

Proyecto Ventas

Creando Software para los Clientes MVC

Componentes de Software de los Clientes MVC

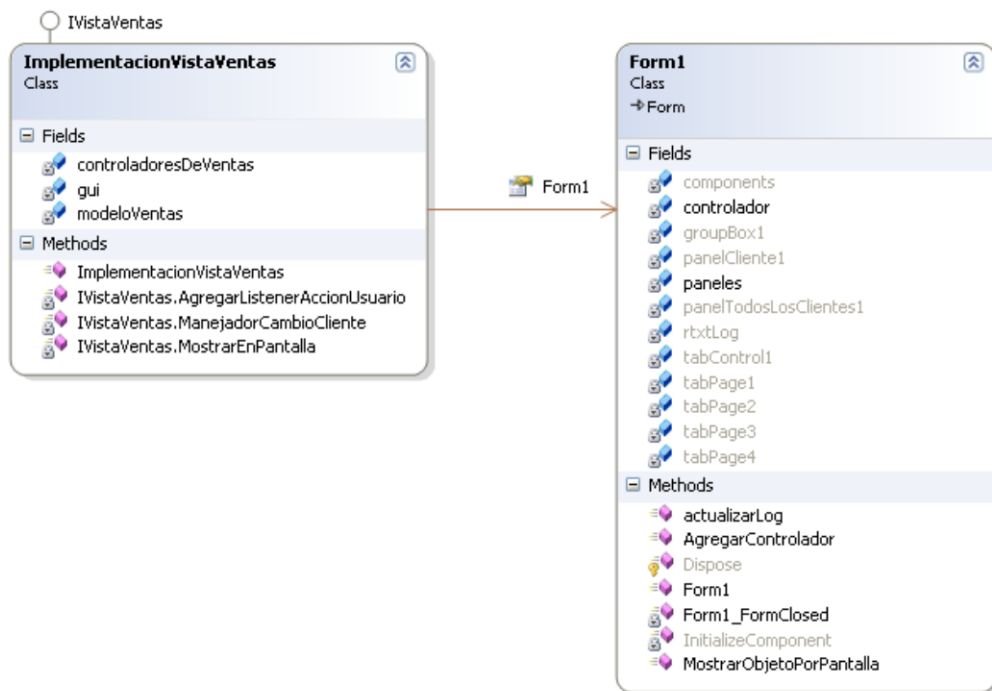


Cada cliente requiere una instancia dedicada de la clase vista y la clase controlador. Estas instancias dedicadas de la vista y el controlador actúan como una única entidad para atender los requerimientos ingresados a través de la interfaz gráfica asociada con la vista

El diseño de la vista del sistema Ventas

En este módulo se crea una implementación de la interfaz IVistaVentas. La construcción en curso de la interfaz IVistaVentas sólo define los aspectos de la vista del cliente del sistema Ventas. En la construcción final, se deberán incluir las vistas para las órdenes de compra y el stock.

La siguiente figura presenta un diseño flexible para la implementación de la interfaz IVistaVentas. Separa la vista de la interfaz gráfica. Esta separación permite múltiples alternativas de interfaces gráficas para la implementación, como HTML, páginas ASP y aplicaciones clientes que pueden ser utilizadas para la construcción de la interfaz gráfica



La clase ImplementacionVistaVentas

La figura anterior muestra como la clase **ImplementacionVistaVentas** implementa todos los métodos de la interfaz. De manera adicional, define clases anidadas anónimas del tipo **ActionListener**. Cada clase anidada está dedicada a atender el evento generado por uno de los botones de la interfaz gráfica del usuario **GUIVentas**.

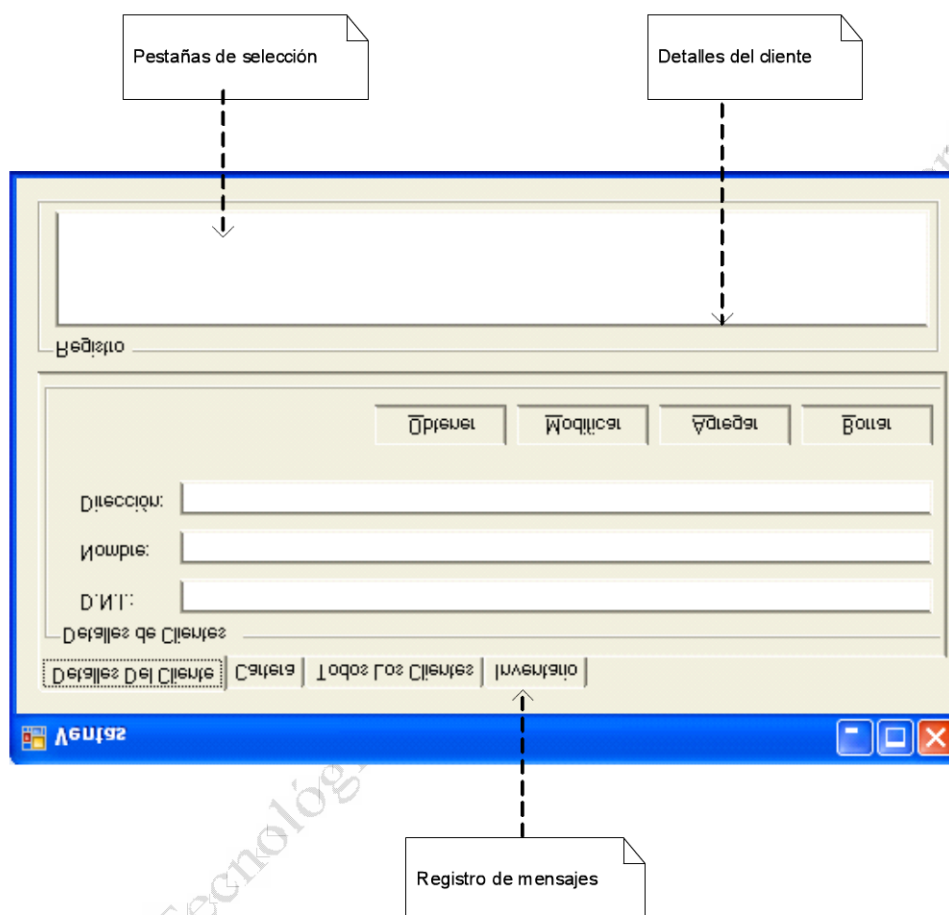
El consultor creó una versión inicial de **ImplementacionVistaVentas** que se tendrá que completar en la construcción 3.

La clase Form1

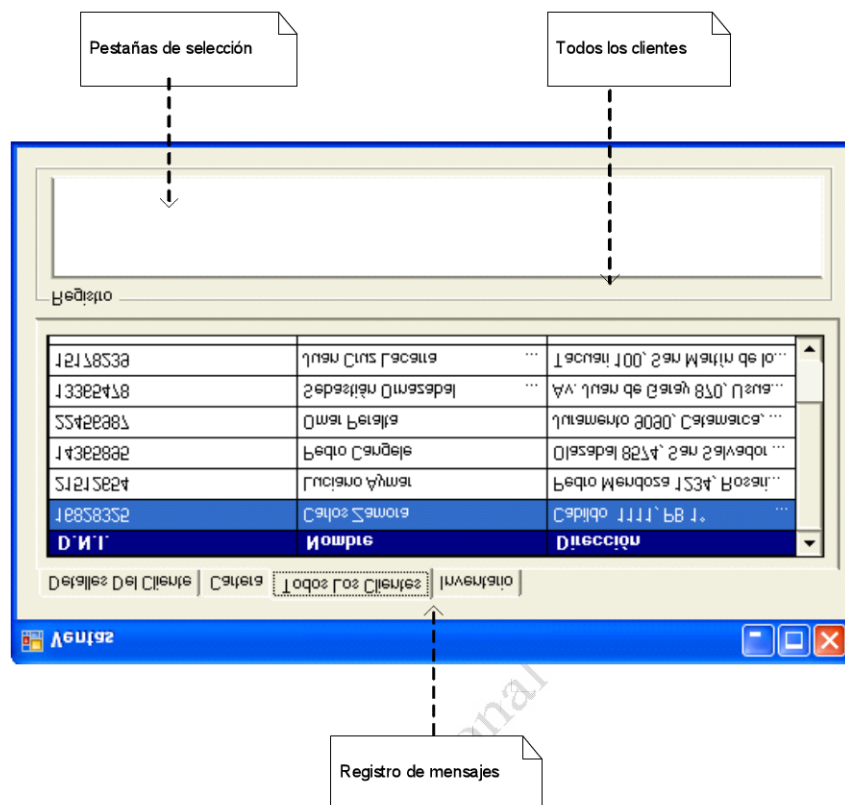
La clase **Form1** es un prototipo de la interfaz de usuario desarrollada para la aplicación **Ventas**. Consiste en un panel de mensajes de registro de mensajes permanente, un panel de selección de función para el vendedor y uno de los siguientes paneles seleccionables para el usuario:

- Detalles del Cliente
- Cartera
- Todos los Clientes
- Inventario

La siguiente figura muestra el Form1 con las pestañas de selección, el panel de detalles de cliente y el panel de registro de mensajes.



La siguiente figura muestra las pestañas de selección, el panel de todos los clientes y el registro de mensajes.



Para manejar la complejidad del diseño visual, se decidió dividir la interfaz en componentes de la siguiente manera:

Panel	Propósito	Diseño
Registro de mensajes	Mostrar en pantalla mensajes y errores	Formulario principal
Pestañas de selección	Seleccionar el panel que actualmente se va a mostrar en pantalla	Control tab en formulario principal
Detalles del cliente	Soporta los casos de uso del cliente	Panel independiente creado como un control definido por el usuario
Cartera	Soporta los casos de uso de órdenes de compra en cartera	Panel independiente creado como un control definido por el usuario
Todos los clientes	Muestra a todos los clientes	Panel independiente creado como un control definido por el usuario
Inventario	Soporta los casos de uso de inventario	Panel independiente creado como un control definido por el usuario

Aunque la implementación actual de la clase Form1 es funcional, no todos sus paneles han sido contruidos. La siguiente tabla muestra el estado de completitud de cada parte.

Panel	Propósito	Comentario
Registro de mensajes	Mostrar en pantalla mensajes y errores	Completo
Pestañas de selección	Seleccionar el panel que actualmente se va a mostrar en pantalla	Completo
Detalles del cliente	Soporta los casos de uso del cliente	Completo
Cartera	Soporta los casos de uso de órdenes de compra en cartera	No se comenzó
Todos los clientes	Muestra a todos los clientes	Incompleto
Inventario	Soporta los casos de uso de inventario	No se comenzó

Diseño de un control definido por el usuario

Los controles de Microsoft ®. NET son todos heredados de una clase base común que se llama UserControl. Esta clase tiene toda la funcionalidad básica para un control gráfico que será utilizado en un formulario de Windows. Todos los controles internos de .NET heredan de esta misma clase base. De la misma manera, se puede heredar de esta clase base y crear controles propios. Cuando se hereda de la clase UserControl, se suministran automáticamente ciertas propiedades y eventos.

Crear un control de usuario

Para crear cualquier control de usuario, se deben seguir una serie de pasos. Los principales pasos son los siguientes.

1. Crear un nuevo proyecto del tipo Windows Control Library
2. Dibujar la interfaz de usuario mediante la selección de otros controles de la caja de herramientas que definen cómo se desea que el control funcione.
3. Crear las propiedades adicionales
4. Crear cualquier evento que se desee
5. Escribir procedimientos de eventos para los controles constituyentes
6. Crear cualquier método que se desee
7. Agregar algunos atributos si es necesario
8. Construir el proyecto del control
9. Añadir un nuevo proyecto Windows Application así se puede probar el control

Examen del diseño de la interfaz gráfica

Para construir la interfaz gráfica como se analizó, se necesitan implementar una clase con el formulario principal que componga los distintos paneles que se mostrarán cada vez que se seleccione una pestaña. Este formato hace pensar en utilizar un patrón de diseño llamado

Composite para la implementación del código puesto que los paneles están “compuestos” en el formulario principal. Este patrón de diseño consiste en los siguientes elementos:

- Componente
- Implementaciones de los componentes
- Composición

Componente

Un componente define el tipo de los objetos contenidos por la composición. Si bien esta definición de tipo se puede hacer mediante una clase, la mayoría de las implementaciones utilizan tanto una clase abstracta como una interfaz. En particular, el proyecto utiliza la interfaz `IPanelVentas` como el tipo que define a los componentes de manera que contiene los servicios que tendrán cada uno de ellos.

C#

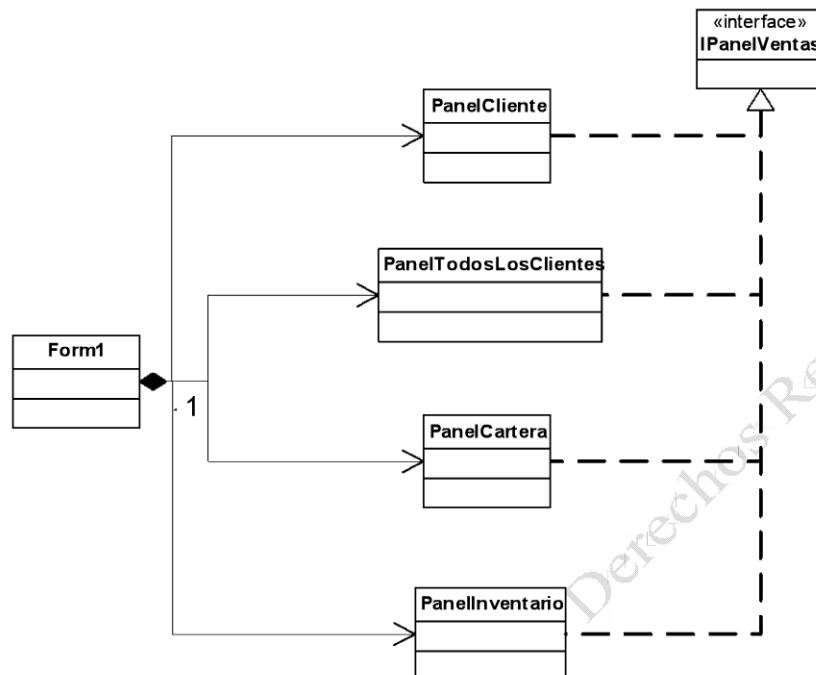
```
public interface IPanelVentas
{
    void RegistrarControlador(IControladorVentas controlador);
    void RegistrarModelo(IModeloVentas modelo);
    void MostrarPorPantalla(Object obj);
    void Refrescar();
}
```

VB

```
Public Interface IPanelVentas
    Sub RegistrarControlador(ByVal controlador As IControladorVentas)
    Sub RegistrarModelo(ByVal modelo As IModeloVentas)
    Sub MostrarPorPantalla(ByVal obj As Object)
    Sub Refrescar()
End Interface
```

Implementación de los componentes

La implementación de los componentes se hace mediante la creación de paneles como controles definidos por el usuario que implementan la interfaz `IPanelVentas` que define a cada componente. El diseño de las clases que definen a los controles de usuario está representado en la siguiente figura:



Composición

El diseño más estricto de este patrón de diseño exige que la clase que contiene a los componentes y conforma la composición debe implementar la interfaz que define al componente por ser ella mismo uno de estos. Sin embargo, para el caso particular del proyecto, esto no brinda ninguna utilidad, por lo tanto se adaptará el patrón de diseño a los fines del proyecto

Beneficios de utilizar el patrón de diseño Composite

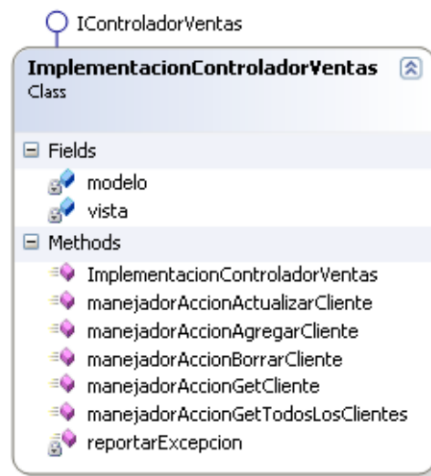
El principal beneficio es su fácil mantenimiento, el cual resulta de la habilidad de los tipos de los componentes para definir los servicios requeridos por las implementaciones de los componentes, lo cual simplifica:

- El agregado de nuevos componentes
- El reemplazo de componentes existentes

El diseño del controlador de Ventas

El controlador de Ventas consiste en una clase que implementa la interfaz IControladorVentas identificada y definida en una previa iteración del ciclo de desarrollo de software.

La siguiente figura muestra el diagrama de clases de ImplementacionControladorVentas. La actual construcción de la interfaz IControladorVentas define sólo los aspectos del controlador para el segmento del cliente de Ventas. En la construcción final, se debe incluir los aspectos del controlador de Cartera e Inventario de Ventas.





Laboratorio 5

Construcción 3

Implementando IVistaVentas

Laboratorio 6

Construcción 4

Implementar el controlador de Ventas

Laboratorio 7

Implementar eventos en la vista de Ventas

Laboratorio 8

Implementar eventos en el controlador de Ventas