

Tecnología ECO

Practica.

Los siguientes puntos te guían paso a paso, para crear una aplicación utilizando tecnología ECO. Partimos de un diseño de nuestro modelo con cinco clases, la clase Persona que se declara como **abstract**, y de la cual deriva la clase Cliente tal como muestra la figura. Declaramos la clase Pedido y establecemos una relación uno a muchos entre las clases Cliente y Pedido. Procedemos de igual forma con las clases Detalle del pedido y con la clase Producto.

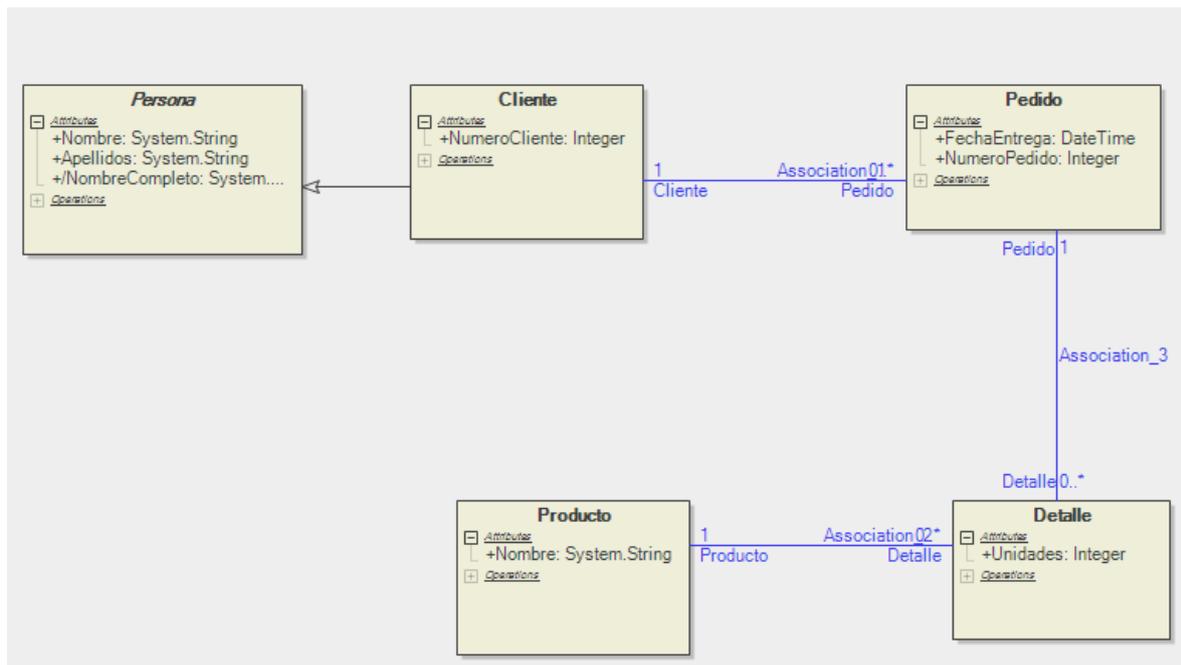


Fig. 1- Diagrama de clases

Se pretende crear una aplicación que gestione todas las clases declaradas, mostrando en la ventana principal de la aplicación, una relación maestro-detalle entre Pedidos y Detalle mediante dos controles DataGridView. Además hay que añadir dos botones cada uno de los cuales accederá a un WinForm. Uno para mostrar los clientes y el otro WinForm los productos. En estos dos últimos formularios se insertará un DataGridView para visualizar la información (Clientes y Productos). En la ventana principal se incluirá otro botón para actualizar la base de datos con los últimos cambios que hayamos realizado.

Desarrollo del Diagrama de clases

1- Iniciar un nuevo proyecto Eco WinForms

(File --> New → Other → Delphi for .NET Project → ECO WinForm Applications)

2- Ir al panel ModelView → Desplegar paquete CoreClasses → pinchar CoreClasses → click botón derecho → Open Diagram

3- Añadir clase Persona:

Inspector de objetos: Propiedad **Abstract: True**

Añadir atributos a la clase: Nombre, Apellidos de tipo String
Añadir atributo NombreCompleto de tipo String y cambiar las propiedades en el Inspector de Objetos: **derived: true, DerivedOcl: Nombre + ' ' + Apellidos**

4- Añadir clase Cliente:

Atributos: NumeroCliente de tipo String

5- Añadir Generalization/Implementation desde cliente a persona.

6- Añadir clase Producto:

Atributos: Nombre de tipo String

7- Añadir clase Pedido:

Atributos: NumeroPedido de tipo Integer, FechaEntrega de tipo DateTime

8- Establecer Association desde Cliente a Pedido, configurar el inspector de objetos cambiando las propiedades **End1: 1, End2: 0..*** del componente Association

9- Añadir clase Detalle:

Atributos: Unidades de tipo integer

10- Establecer Association desde Producto a Detalle, cambiar las propiedades en el inspector de objetos del Association **End1: 1, End2: 0..***

11- Establecer Association desde Pedido a Detalle, cambiar las propiedades en el inspector de objetos de Association **End1: 1, End2: 0..***

Configuración de la Persistencia ECOSpace.

12- Crear base de datos en blanco desde nuestro gestor de base de datos. Para esta práctica se ha utilizado SQL Server. Se ha llamado a la base de datos PruebasECO

13- Abrir la ventana <NombreProyecto>ECOSpace.pas donde <NombreProyecto> es el nombre que le hayamos dado a nuestro proyecto en la primera ventana que abrió el asistente al iniciar el proyecto ECO. Arrastrar desde el panel **Data Explorer** de Delphi, la base de datos PruebasECO (conexion) al interior de la ventana ECOSpace.

14- Añadir componente de la paleta de herramientas PersistenceMapperBDP que se encuentra en la categoría **Enterprise Core Objects** y configurar la propiedad connection del inspector de objetos a la conexión que hemos añadido al ECOSpace. Hacer click botón drcho sobre PersistenceMapperBDP y seleccionar "SQL Server Setup".

15- Botón "Generate Schema" de la barra de menús del EcoSpace.

Creación de la capa presentación

16- Añadimos dos nuevos formularios ECO

(File → New → Other → Delphi for .NET Project → New ECO Files → ECO Enabled WinForms)

17- Añadimos en WinForm botones con el **Text** productos, clientes, cerrar;

18- Añadir referencia a las dos nuevos formularios: File → Use Unit → WinForm1 y WinForm2.

19- Escribir el siguiente código en el evento click de los botones Productos y Clientes:

```
With TWinForm1.Create(FEcoSpace) do ShowDialog;
```

20- Escribir el siguiente código en el evento click del botón cerrar `Application.Exit;`

En WinForm1

20- En el componente RHRoot seleccionamos ECOSpaceType.

21- Añadimos un ExpressionHandler1, configurar las propiedades en el inspector de objetos RootHandle: RHRoot, Expression: Producto.AllInstances.

22- Añadimos un DataGrid1 configurar en el inspector de objetos DataSource:ExpressionHandler1

23- Añadimos un botón, y configurar en el inspector de objetos ECOListAction: Add, RootHandler: ExpressionHandler1

En WinForm2 (repetimos el mismo proceso que en WinForm1)

24- RHRoot seleccionamos ECOSpaceType.

25- Añadimos un ExpressionHandler1, RootHandle: RHRoot, Expression: Cliente.AllInstances.

26- Añadimos un DataGrid1 DataSource:ExpressionHandler1

27- Añadimos un botón, ECOListAction: Add, RootHandler: ExpressionHandler1

En WinForm

28- Añadimos botón para actualizar la base de datos y configuramos el inspector de objetos las propiedades RootHandler: RHRoot, ECOAction: UpdateDataBase

29- RHRoot seleccionamos ECOSpaceType.

30- Añadimos un ExpressionHandler1, configurar las propiedades en el inspector de objetos RootHandle: RHRoot, Expression: Clientes.AllInstances.

31- Añadimos un ListBox y cambiar sus propiedades del inspector de objetos DataSource:ExpressionHandler1, DisplayMember: NombreCompleto

32- Añadimos un CurrencyManagerHandle1, y cambiamos sus propiedades en el inspector de objetos RootHandler: ExpressionHandler1, BindingContext: ListBox1.

33- Añadimos un ExpressionHandler2, y cambiamos las propiedades en el inspector de objetos RootHandle: CurrencyManagerHandle1, Expression: Self.Pedidos

34- Añadimos un DataGrid1, cambiar propiedades DataSource: ExpressionHandler2

35- Añadimos un botón para añadir datos y cambiar propiedades ECOListAction: Add, RootHandler: ExpressionHandler2

36- Añadimos un CurrencyManagerHandle2, cambiar propiedades en el inspector de objetos RootHandler: ExpressionHandler2, BindingContext: DataGrid1.

37- Añadimos un ExpressionHandler3, cambiar propiedades RootHandle: CurrencyManagerHandle2, Expression: Self.Detalle

38- Añadimos un DataGrid2, cambiar propiedades DataSource: ExpressionHandler3

39- Añadimos un botón, cambiar propiedades ECOListAction: Add, RootHandler: ExpressionHandler3

40- Añadimos un ExpressionHandler4, cambiar propiedades RootHandle: RHRoot, Expression: Producto.AllInstances.

41- Añadimos un ComboBox1, cambiar propiedades DataSource: ExpressionHandler4, DisplayMember: Nombre

42- Añadimos un CurrencyManagerHandle3, cambiar propiedades RootHandler: ExpressionHandler3, BindingContext: DataGrid2.

43- Añadimos un CurrencyManagerHandle4, cambiar propiedades RootHandler: ExpressionHandler4, BindingContext: ComboBox1.

44- En el gestor de evento "SelectedIndexChanged" del componente ComboBox1 programamos las siguientes líneas de código:

```
var
    SelectedDetalle: Detalle;
begin
    if CurrencyManagerHandle3.Element.AsObject is Detalle then
        begin
            SelectedDetalle := Detalle(CurrencyManagerHandle3.Element.AsObject);
            SelectedDetalle.Producto := Producto(CurrencyManagerHandle4.Element.AsObject)
        end;
end;
```

45- En el ExpressionHandler3, Inspector de objetos propiedad Column añadimos una columna Name: ProductoNombre, Expression: Self.Producto, Nexted: True, NestedName:Producto

Fin de la práctica

Esperamos que esta práctica le ayude a iniciarse en la tecnología ECO, si desea ampliar esta información le rogamos contacte con los servicios profesionales Danysoft en el 902 123146, o en info@danysoft.com