Práctica primer trimestre

Normativa para la práctica:

- ☐ El nombre de la solución y del archivo comprimido ha de ser: Apellido1_Nombre_VENTASDAM
- ☐ Documentar cada uno de los diferentes ficheros de nuestro proyecto de la siguiente forma:
  ///<author> Antonio Rodríguez</author>
- ☐ La nomenclatura de las variables tiene que ser camelCase y el nombre de las mismas debe ser significativo en cuanto a su contenido.
- ☐ La nomenclatura de los atributos xxxxx siendo el nombre camelCase y el nombre de las mismas debe ser significativo en cuanto a su contenido.
- ☐ La nomenclatura de las propiedades y métodos deben seguir la nomenclatura PascalCase y en el caso de las propiedades el nombre igual que el atributo al que expone.
- ☐ Las soluciones deben basarse en P.O.O.
- ☐ La aplicación deberá realizarse con .Net 6.0

Descripción de la aplicación

La aplicación **VentasDAM** gestionará los datos relacionados con la empresa Northwind, que tendrá pedidos, clientes proveedores, etc. La aplicación se deberá desarrollar basándose en un patrón multicapa con entidades y programación orientada a objetos. La base de datos elegida para este proyecto es **SQL Server**, proporcionándose los scripts de creación de dicha base de datos y de una capa de datos deberá estar basada en Entity Framework Core 6.0, realizada en la práctica anterior.

Funcionamiento de la aplicación

La aplicación constará de un formulario principal y nada más iniciar la aplicación deberá aparecer un formulario de selección de usuario. En esa selección de usuario deberán aparecer todos los empleados y una vez seleccionado uno, nos pedirá que confirmemos la selección introduciendo su *EmployeeId*. (Como no tenemos contraseña usamos este campo). El formulario principal tendrá implementados las diferentes opciones mediante formularios MDI.

A continuación, se explicarán las funcionalidades que se deben desarrollar en el formulario principal.

Formulario PRINCIPAL.

El formulario principal, que se ejecutará con una resolución de 1280x900, sin posibilidad de redimensionar. (Ojo con el dpi de monitores de gran resolución). En parte superior se mostrará una barra de menús que contendrá acceso a todas las funcionalidades de la aplicación, así como una barra de herramientas con acceso a las opciones de Insertar empleado, Nuevo Pedido e Imprimir una factura. También deberá mostrar en todo momento el usuario que ha iniciado sesión en la parte derecha y su fotografía. La barra de menús que contendrá las siguientes opciones:

- **Empleados**
  - **Insertar**
  - **Modificar**

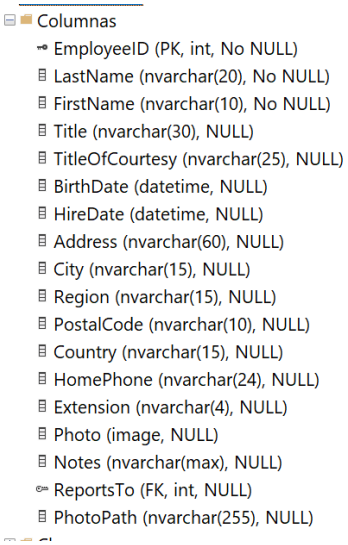
- **Eliminar**
- **Productos**
  - **Modificar Precio**
- **Pedidos**
  - **Nueva**
  - **Modificar**
  - **Eliminar**
- **Estadísticas**
  - **Total de Pedidos por cliente**
  - **Productos por categoría**
- **Informes**
  - **Factura**
- **Acerca de …**
- **Salir**

Formulario Empleado. (Insertar, actualizar y borrar)

Este formulario nos permitirá gestionar los diferentes empleados que gestionan la aplicación. Desde las diferentes opciones podremos dar de alta, de baja o modificar los diferentes usuarios. Las opciones de insertar y actualizar utilizarán el mismo formulario, pero con distinta funcionalidad.

A la hora de realizar cualquiera de estas operaciones habrá que realizar las siguientes validaciones:

1. Validar todos los campos obligatorios (aquellos no puedan ser nulos)
2. Longitud máxima de todos los campos
3. Comprobar que el campo PhotoPath está bien formado, mediante una expresión regular. (Ruta de Windows no permite los siguientes caracteres \ / : * ? " < > | para los nombres de las carpetas)
4. *HomePhone* solo contenga números mediante una máscara de entrada. La longitud del número de teléfono oscila entre 9 y 12.
5. *TitleOfCourtesy* solo puede contener los valores: Ms, Dr., Mrs., Mr.
6. *BirthDate* tiene que ser menor que la fecha de hoy.
7. Para seleccionar la foto [*Photo*] usaremos un *OpenFileDialog*



Una vez todos los campos estén validados se mostrará un mensaje informando que todo ha salido correctamente, en caso contrario se deberá indicar de manera clara e inequívoca dónde se ha producido un error y porqué. Se deja al alumno elegir cuál es el método y el momento más adecuado para realizar las validaciones anteriores.

Hay que tener en cuenta, que el campo *EmployeeID* lo proporciona la base de datos, ya que el campo es un campo autonumérico. (Serial ID)

Nota: Para validar un formulario existen dos momentos:

1. En tiempo de **diseño.**
   En tiempo de diseño podemos validar los formularios mediante las propiedades de los mismos, como son MaxLength, Max, Min, etc.
2. En tiempo de **ejecución**
   2.1. Al pulsar el botón Aceptar del formulario
   2.2. Al salir de control (Evento Leave)
   2.3. Mientras se escribe (KeyPress)
   2.4. Eventos Validating/validate

Insertar
Desde la opción insertar accederemos al formulario, con el modo insertar, donde estarán todos los campos vacíos y rellenaremos los datos necesarios para dar un empleado de alta. Antes de darlo de alta se deberán validar todos los campos según las indicaciones anteriores.
El código de empleado lo proporciona la base de datos por lo que no hay que introducirlo.

Actualizar
Antes de modificar un empleado deberemos seleccionar uno de entre todos los disponibles. Para esto crearemos un formulario de búsqueda que nos permitirá buscar empleados por *Firstname* y *LastName* o por *EmployeeID*. Estas búsquedas de empleados se harán de forma incremental, es decir mientras escribimos e irán actualizando los resultados de búsqueda. **Para realizar esto utilizaremos Dataviews asociados a un DataTable.**
El diseño del formulario de búsqueda constará de dos partes una para escribir los criterios de búsqueda y otra con los resultados, siguiendo un esquema similar al buscador Google.

Cuando seleccionemos un empleado cargaremos el mismo formulario que el de Insertar Empleado, pero en el modo Editar. En este modo se cargarán todos los datos del usuario y se podrán modificar todos los datos excepto el código de cliente. También hay que validar los campos antes de realizar la modificación.

Eliminar
En la opción de eliminar cliente, se reutilizará el buscador anterior, pero cuando lo seleccionemos, se nos preguntará si estamos seguros de querer eliminarlo, mostrando los datos más relevantes del mismo.

Formulario PRODUCTOS.

En esta opción visualizaremos la información de los productos existentes y solo podremos modificar sus precios. Para seleccionar un producto, mostraremos en la parte superior del formulario un control de tipo *DataGridView* que muestre las categorías y permita filtrar por nombre.
Una vez seleccionado una categoría, en la parte media del formulario se mostrarán todos los productos, quedando la parte inferior para mostrar todos sus datos. **No se puede usar para implementar el buscador el objeto Dataview**, hay que implementarlo de una forma diferente. Para modificar el precio requerirá la confirmación del usuario (volver a pedir el *EmployeeId*)

Formulario PEDIDOS.

Nuevo.
Permitirá la creación de un nuevo pedido con sus productos, precios, cantidad y descuentos correspondientes.

El formulario constará de tres partes, por un lado, los datos generales del pedido en la parte superior, tendremos una cabecera dónde se mostrarán los datos de la tabla *Orders*, recodad que, si hay una clave ajena, no se debe mostrar el código si no los datos más importantes del registro referenciando.

En la segunda parte del formulario, seleccionaremos los productos mediante un buscador de productos. Los diferentes valores se insertarán mediante un *DataGridview. Estos registros se podrán modificar, añadir nuevos e incluso eliminar.*

En la tercera parte del formulario, tendremos un resumen de la factura, como el total a pagar, el total sin IVA, y el IVA. Estos datos se calcularán en la capa de Negocio.

Búsqueda y modificación.
Se permitirá la búsqueda de pedidos, por cliente **y/o** fecha. Reutilizaremos el buscador de *Customers* para seleccionar el cliente que realiza el pedido.
Una vez seleccionado el pedido deseado, se mostrarán todos sus datos y podremos modificarlo eliminando o añadiendo artículos, o modificando los datos de la cabecera.

Formulario ESTADÍSTICAS.

Esta opción nos permitirá de manera gráfica y elaborada ver distintos tipos de información. Aunque el diseño es libre, se recomienda usar un control de tipo *tabControl* para mostrar las distintas estadísticas.

- **Total de Pedidos por cliente:** (Gráfico de barras)
- **Total de Productos por categoría.** (Gráfico circular)

Formulario FACTURAS.

Desde esta opción podremos visualizar e imprimir el siguiente informe:
Factura.
Esta opción nos permitirá buscar un pedido y generaré una factura de este. Deberemos seleccionar. un pedido El buscador funcionará igual que el usado en la opción de consulta de pedidos. El diseño del informe debe mostrar todos los datos de la factura, incluidas las líneas de detalle (*OrderDetails*). Todo en una única hoja.

Formulario ACERCA DE.

Ventana incluida en Visual Studio, que nos mostrará información acerca del programa realizado.

Opción de menú SALIR.

En la opción de salir, deberemos pedir confirmación al usuario antes de realizar la opción de salida. ¿Está usted seguro de que quiere salir?

Recordatorio de normas *importantes*:

- El nombre de la solución ha de ser: Apellido1_Nombre_VENTASDAM.
- La nomenclatura de las variables tiene que ser *lowerCamelCase* para las variables y atributos y *UpperCamelCase* para métodos y clases. Los nombres de las mismas significativos en cuanto a su contenido.
- Respetar los nombres de las clases, los atributos, las propiedades y los métodos definidos en el enunciado.
- La representación de los formularios ha de ser lo más aproximada posible.
- Deberá ser un proyecto empleando el patrón de desarrollo multicapa con entidades.
- Para realizar el proyecto se utilizará obligatoriamente WindowsForms y C#.
- Todos los formularios deben cumplir las normas de diseño y de usabilidad tratadas en clase.
- Se valorará el diseño de los formularios del proyecto.
- Se realizará un examen sobre el proyecto, en el cual el alumno deberá explicar al profesor el funcionamiento del proyecto. El profesor podrá solicitar modificaciones "in-situ" sobre el programa. En caso de que el alumno no domine el funcionamiento de su propio programa o no sepa hacer modificaciones, será suspendido.
- Si el programa no compila o alguna de las opciones no funciona o presenta fallos de ejecución la puntuación en ese apartado será 0.

Criterios de evaluación de la entrega

A la hora de evaluar esta entrega de la práctica se tendrán en cuenta los siguientes aspectos:

- Diseño de los formularios siguiendo criterios de usabilidad.
- Correcta implementación de las clases.
- Creación de atributos y propiedades ajustados a la base de datos
- Creación de los métodos y formularios adecuadamente ajustándose al enunciado y a las funcionalidades pedidas.
- Validación del dominio de los tipos de datos.
- Documentación del código.