

**Parcial PC – Tema A (Usuario)**

Resolver en Visual Studio .Net las siguientes consignas construyendo una solución .Net organizada en capas. Serán considerados como parte de los criterios de evaluación el correcto funcionamiento (compilación sin errores), uso de estándares y buenas prácticas.

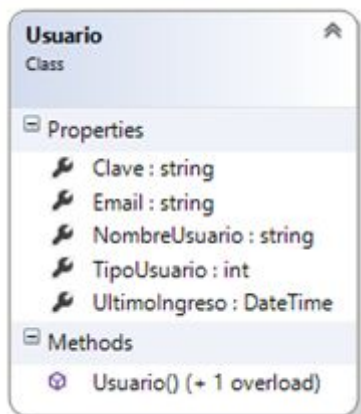
**NOTA IMPORTANTE:** crear carpeta de nombre **ApellidoNombre** y dentro de ella todo lo que se solicita. Configurar cada proyecto (y sus clases) dentro de namespace con formato [Apellido].[NombreProyecto] (Ejemplos: Perez.Entidades y Perez.Datos)

**BASE A DATOS DE UTILIZAR:** [ver tutorial](#)

**Consignas****1. Creación de la Solución y proyecto Entidades**

PARA REGULARIZAR (1 PUNTO)

- Crear una solución cuyo nombre sea su **Apellido** en la carpeta propuesta, conteniendo un proyecto de *Biblioteca de clases* llamado **Entidades**. Configurar a nivel de *proyecto* el *namespace* como se propone en la Nota inicial, en este caso [Apellido].Entidades.
- Dentro del proyecto **Entidades**, crear la clase de la imagen con las *propiedades* públicas que se muestran en el diagrama adjunto, respetando *tipo de datos*.
- Agregue a la clase dos (2) constructores, uno sin *parámetros* y otro con los *parámetros* necesarios que permita completar cada *propiedad*.



PARA APROBACIÓN DIRECTA (1 PUNTO)

- Agregar la propiedad **DiasUltimoIngreso** como propiedad de solo lectura y que devuelva la cantidad de días que pasaron desde el último ingreso.

**2. Creación de la capa de Acceso a Datos y conexión a la Base de Datos**

PARA REGULARIZAR (3 PUNTOS)

- Agregar un proyecto del tipo “Biblioteca de clases” denominado **Datos** y agregar referencia/s necesaria/s. Explicitar namespace a nivel de proyecto en base a Nota Inicial como [Apellido].Datos.
- Dentro del proyecto, crear la clase llamada **Base** que contenga una propiedad pública del tipo **xxxConnection** (del mismo proveedor de la base de datos). Al instanciarse debe crear la conexión a la base de datos llamada **NetGenerica**, utilizando el string de conexión acorde (deseable que el connection string se recupere de forma centralizada) y mantenerla de modo tal que las clases que la hereden puedan usar dicha conexión (la apertura y cierre de la conexión puede resolverse con métodos dentro de esta clase o ejecutando desde la clase derivada)

- c. Crear clase pública **Usuario** que contengan los siguientes métodos que se conecten Base de Datos utilizando la conexión de la clase Base y recuperen/persistan datos de/en la tabla homónima. Considerar que el/los objeto/s retornado/s y argumentos de los métodos tomen el objeto de Entidad o colección de este tipo (Ej. List<Entidades.Usuario>)
  - i. Método RecuperarTodos que obtenga todos los registros de la tabla.
  - ii. Método Agregar que reciba la entidad e inserte un nuevo registro en la tabla.

PARA APROBACIÓN DIRECTA (2 PUNTOS)

- d. Agregar un método RecuperarUno que reciba NombreUsuario de Parámetro y devuelva la Entidad Usuario que corresponda. Implementar esta funcionalidad usando LINQ sobre el resultado del método RecuperarTodos del punto c.  
**Observación:** Si este punto no se resuelve usando LINQ y se decide resolverlo usando ADO.NET tiene el 50% del valor en la corrección.

3. Creación de la capa de Negocio.

PARA REGULARIZAR (1 PUNTO)

- a. Agregar proyecto del tipo “Biblioteca de clases” denominado Negocio. Agregar en el proyecto referencias necesarias y configurar namespace como [Apellido].Negocio.
- b. Crear la clase pública **Usuario** dentro de Negocio, y crear los métodos estáticos necesarios que llamen a los métodos creados en la capa de Datos tomando el mismo nombre que estos.

PARA APROBACIÓN DIRECTA (2 PUNTOS)

- c. Crear la clase pública Validaciones dentro de Negocio, y crear un método estático EsMailValido que reciba como argumento un string (que representa el mail a validar) y retorne un boolean indicando si es válido o no el argumento que debe ser analizado con una expresión regular.
- d. Crear el método estático que llame al método Recuperar Uno de la capa de Datos.
- e. Si el parámetro NombreUsuario del método RecuperarUno es nulo o vacío (null or empty) disparar una Exception que informe de este problema a la clase que lo está invocando.

4. Creación del proyecto Escritorio, formularios Principal y Alta

PARA REGULARIZAR (3 PUNTOS)

- a. Agregar un proyecto de tipo Windows Forms llamado Escritorio. Configurar namespace como [Apellido].Escritorio. Crear formulario frmPrincipal que contenga un control Menu que incluya menú Principal con la opción Alta. Usar convención de nombres para formulario y controles.
- b. Agregue un formulario Windows Forms denominado “Alta” el cual contiene Labels, TextBoxes que permitan ingresar los datos de un nuevo registro de la Entidad del punto 1 y los siguientes botones:
  - i. Botón Cancelar: cierra el formulario.
  - ii. Botón Aceptar: graba los datos de los campos en la Base de Datos y muestra mensaje confirmando que la operación fue satisfactoria.

PARA APROBACIÓN DIRECTA (2 PUNTOS)

- c. En el botón Aceptar agregar validación de campos requeridos y correo válido (usando método de la clase Validaciones del Proyecto Negocio) que en caso de fallar algún informe de los errores antes de persistir los datos.

5. Creación del proyecto Web y formulario ASP.Net

PARA REGULARIZAR (2 PUNTOS)

- a. Agregar un proyecto de tipo Aplicación Web Form ASP. NET llamado Web.
- b. Crear un Web Form Listado.aspx que contenga un control grilla que muestre todos los registros de la tabla utilizando el método adecuado de Negocio.

PARA APROBACIÓN DIRECTA (3 PUNTOS)

- c. Crear un Web Form Usuario.aspx que contenga un TextBox en el cual se pueda ingresar el NombreUsuario y un botón Recuperar que usando ese nombre de usuario recupere el resto de la Entidad y la muestre en Labels en el mismo Web Form.
- d. Agregar código que capture la Exception que es disparada por la Capa de Negocios cuando el nombre de usuario es vacío y muestre un mensaje apropiado al usuario.