

## Tarea 03.01- GestionTareas - Login

## 1. Objetivo

Vamos a crear a lo largo de este curso una aplicación de gestión de tareas. Estas tareas formarán parte de proyectos y los usuarios tendrán tareas asignadas dentro de los proyectos a los que pertenezcan. Una tarea solo puede pertenecer a un proyecto. Y para simplificar, una tarea solo puede pertenecer a un usuario. Los proyectos podrán tener muchas tareas y muchos usuarios.

## 2. Primera entrega

En esta primera entrega, nos vamos a centrar en la gestión del acceso a la aplicación. Hay que crear un formulario de Login, que nos dé acceso al formulario principal de la aplicación. Una vez realizado el diseño, deberéis implementar el evento click del botón que compruebe si el usuario y la contraseña existen en la base de datos y si coinciden con los introducidos por el usuario. Si es correcto, accederemos al formulario pantalla principal de la aplicación. El formulario principal para esta práctica será un formulario vacío.

Para el diseño del login, debéis seguir las instrucciones de este vídeo <https://www.youtube.com/watch?v=ebRTeKId4Nc>.

Propiedades de la clase Usuario

- Id (int, Clave Primaria)
- Usuario (String, Obligatorio, Maxlength 30)
- PasswordHash (String, Obligatorio)
- NombreCompleto (String, Maxlength, 100)
- CorreoElectronico (String, Obligatorio, Maxlength, 100)
- Activo (int, 0-1)
- FechaCreacion (DateTime)
- FechaBaja (DateTime)

**Tareas a realizar:**

1. Crear el proyecto incluyendo las referencias necesarias para que el proyecto funcione adecuadamente.
2. Crear la clase Usuario, con DataAnnotations, para gestionar el acceso a la bb.dd. mediante EntityFramework.
3. Crear la clase ServiceUsuario con los métodos necesarios para el acceso a la BD.
4. Ejecutar los comandos para crear la persistencia de datos.
5. Crear el formulario de login siguiendo el tutorial.
6. Crear las funciones necesarias para consultar el usuario y el password con los datos que hay en la bb.dd.
7. Crear el formulario principal y «enlazar» con el resultado correcto del formulario de Login.
8. Insertar un usuario administrador, con la contraseña 1234 (cifrada).
9. Ejecutar y probar la aplicación.

Algunos atributos pueden ser nulos. Para que una variable o atributo en C# admita nulos debemos añadir el símbolo “?” después del tipo.

**int? Es un tipo int que puede contener el valor null.**

Mas información en <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/builtin-types/nullable-value-types>

**Consideraciones finales**

El nombre de la base de datos deberá ser **Tareas\_TUSINICIALES**, por ejemplo, en mi caso sería **TAREAS\_JLF**. Si esta restricción no se cumple la práctica se dará por suspendida, ya que no podré ejecutarla al solaparse con otras bases de datos existentes.

Cuando reciba la práctica por primera vez ejecutaré un **Update-DataBase** para actualizar la base de datos a la última actualización.

## Entrega

La entrega ha de cumplir las siguientes condiciones:

- El nombre de la solución y del archivo comprimido ha de ser:  
**Apellido1\_Nombre\_GestionTareas03.01**
- Documentar cada uno de los diferentes ficheros de nuestro proyecto de la siguiente forma:  

```
///<author> Jose Llaurado</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 ha de seguir el método camelCase y el nombre de las mismas debe ser significativo en cuanto a su contenido.
- La nomenclatura de las propiedades PascalCase y nombre igual que el atributo al que expone, pero en mayúscula.
- Las soluciones deben basarse en P.O.O.