 <p>UNIVERSIDAD NACIONAL DE JUJUY FACULTAD DE INGENIERÍA ANALISTA PROGRAMADOR UNIVERSITARIO (PLAN 2008)</p>	<p><b>LABORATORIO DE PROGRAMACIÓN ORIENTADA A OBJETOS II</b></p>	<p>TP N°</p> <p><b>3</b></p>
		<p><b>2023</b></p>


<b>ENLACE A DATOS</b>
-----------------------

Partiendo de la aplicación del Trabajo Práctico anterior, extienda su funcionalidad de acuerdo a los siguientes requerimientos:

- 1) Crear una base de datos denominada “playa.mdf” desde el Explorador de Servidores de Visual Studio, en base al diagrama de clases.
- 2) Mostrar en una grilla los datos de los Tipos de Vehículo:
  - a. En el proyecto ClasesBase, agregar una clase llamada TrabajarTiposVehiculo que contenga un método llamado TraerTiposVehiculo y devuelva el Código, Descripción y Tarifa de la tabla TipoVehiculo de la base de datos.
  - b. Mostrar los datos en una grilla. La misma puede ser visualizada en una nueva ventana.
- 3) Enlazar los datos de un archivo XML para llenar un ComboBox. Al seleccionar un ítem del ComboBox se debe cambiar el valor de la propiedad Fill de un objeto Rectangle. La solución a este ejercicio deberá realizarse usando enlace de datos:
  - a. Agregar un archivo XML llamado “Tiempos.xml”. Definir los datos XML para representar una lista de tiempos de estadía en un sector de la playa de estacionamiento. Las duraciones posibles (en minutos) pueden ser:
    - 0
    - 15
    - 30
    - 60
    - 90
    - 120

Nota: los valores mayores a 0 (cero) indican que el sector está ocupado, mientras que el valor 0 (cero) indica que el sector está disponible.

- b. Crear una ventana “EstadoSector”, agregar un ComboBox y enlazar la lista de duración en minutos del archivo XML.
- c. Crear una clase “ConversorDeEstados” para controlar la conversión de datos. La clase debe implementar la interfase “IValueConverter”. La conversión deberá ser programada de manera que reciba un dato de tipo numérico que represente el tiempo en minutos de estadía de un vehículo en un sector y devuelva un tipo de dato Brush, que es el tipo de dato que soporta la propiedad Fill. Según el estado, el color de relleno del objeto Rectangle será de acuerdo a las siguientes condiciones:
  - Si Duración en minutos = 0: Rellenar con Verde (sector libre).
  - Si Duración en minutos > 0 y <=30: Rellenar con Rojo muy claro (sector ocupado).
  - Si Duración en minutos > 30 y <= 60: Rellenar con Rojo claro (sector ocupado).
  - Si Duración en minutos > 60: Rellenar con Rojo oscuro (sector ocupado).

 UNIVERSIDAD NACIONAL DE JUJUY FACULTAD DE INGENIERÍA ANALISTA PROGRAMADOR UNIVERSITARIO (PLAN 2008)	LABORATORIO DE PROGRAMACIÓN ORIENTADA A OBJETOS II	TP N°
		3
		2023

ENLACE A DATOS
----------------

### ENLACE A DATOS

- d. Agregar un objeto Rectangle y asociar su propiedad Fill al ComboBox creado en el punto b). El enlace debe realizarse con el ítem seleccionado y el color de relleno en base a la clase “ConversorDeEstados”.
- 4) Enlazar elementos de tipo TextBox a los datos de la tabla Cliente y validar los campos:
    - a. Agregar una ventana “ValidarCliente” al proyecto WPF.
    - b. Modificar la clase Cliente declarada en el Proyecto “ClasesBase”, de manera de implementarle la interfase IDataErrorInfo y escribir el código para validar los datos: ClienteDNI (obligatorio), Apellido (obligatorio), Nombre (obligatorio), Telefono (obligatorio).
    - c. Crear una clase llamada TrabajarClientes que contenga un método TraerCliente con un parámetro que represente el dni del Cliente (ClienteDNI). Este método debe devolver los datos apellido, nombre y telefono de la tabla Cliente de la base de datos Playa en un objeto de tipo Cliente (por ejemplo “oCliente”).
    - d. Para el Alta y Modificación de Cliente realizar lo siguiente:
      - Asociar la validación armada dentro de la clase Cliente.
      - Mostrar los errores usando plantillas y estilos.
    - e. Sólo para la Modificación de Cliente:
      - Los datos se deben mostrar en cada uno de los TextBox, y al cambiar el código (dni) se debe actualizar el apellido, nombre y teléfono (es decir, el valor del TextBox para el código funciona como parámetro del método TraerCliente).

### Consideraciones:

- a) El trabajo podrá ser desarrollado en forma individual o grupal con un máximo de 7 integrantes por grupo para la comisión SSJ y de 4 integrantes por grupo para la comisión SP.
- b) El presente trabajo práctico **deberá ser entregado por un integrante de cada grupo en la fecha establecida como fecha de entrega** o de lo **contrario** el mismo se considerará como **desaprobado**.
- c) La entrega del trabajo deberá ser con ejercicios completos que compilen y ejecuten sin errores de compilación. Todo tipo de error que presente el trabajo deberá ser corregido previo a la fecha de entrega. Aquellos trabajos que se entreguen en forma incompleta o con errores se considerarán como **desaprobados**.
- d) La entrega del presente trabajo práctico deberá ser en un archivo .rar que contenga el proyecto desarrollado en Visual Studio. Dicho archivo se deberá enviar al correo del docente responsable de la práctica de las comisiones de SP o SSJ respectivamente. Nota: Si el archivo no puede ser enviado, modificar la extensión del mismo, por ejemplo con un guion bajo: “.rar\_”.