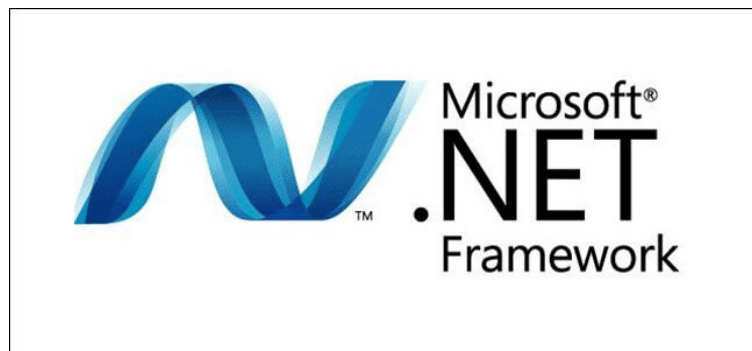




## Laboratorio de Programación y Lenguajes 2024

### Trabajo Práctico Obligatorio Lenguaje de programación C#



*Facultad de Ingeniería  
Universidad Nacional de la Patagonia San Juan Bosco*

### ***Especificación de los trabajos finales de lenguajes de programación unificados por el uso de una base de datos PostgreSQL.***

Los trabajos finales de lenguajes utilizan una base relacional PostgreSQL, de esta forma se deberá interactuar de una forma ordenada y organizada para lograr la ejecución correcta de las consultas, ya que el motor procesa en forma de consultas las interacciones que se programen para dar solución a lo que indique en los requerimientos del sistema.

### **Sistema de gestión de Turística**

Sistema de gestión La secretaría de Turismo de la Provincia quiere administrar los datos de las operaciones turísticas de las distintas agencias que están inscriptas como proveedoras de servicios turísticos en todo el territorio de la provincia. Para ello los directivos de la misma han participado de un relevamiento con nuestra consultora. Del relevamiento se desprende que existen distintas agencias que ofrecen paquetes turísticos para observar, fotografiar y en algunos casos hasta interactuar con algunos animales marinos y/o terrestres como así también visitar y recorrer diferentes lugares naturales con una variada y atractiva fauna y vegetación. Cada agencia se encuentra radicada en distintas localidades y posee un código identificadorio, nombre, dirección(Calle,nro/piso/dpto) y al menos 2 teléfonos. Se encargan de ofrecer distintos paquetes turísticos que incluyen avistajes, ya sea de ballenas, delfines, pingüinos, lobos marinos, elefantes marinos, guanacos, avestruces, etc. También caminatas (trekking) en distintos parajes con variado nivel de dificultad y duración. Los costos de los paquetes se conforman del importe de las actividades que componen al mismo. Dentro de la información del paquete se puede especificar el nivel o dificultad que a su vez incluye uno o varias actividades vinculadas a ese nivel indicado. Las actividades pueden contar con un transporte o no, según sea el caso el importe del transporte se suma al valor de la actividad. Los datos de los transportes son la identificación o dominio, y una descripción que registra el tipo de transporte, capacidad y detalles. Cada agencia ofrece un conjunto de paquetes turísticos predeterminados. Por último cada agencia permite la creación de un paquete a medida dependiendo de las actividades que puede proveer, dando lugar a una adaptación única por turista que lo arma a su medida. Luego de tener determinado el paquete que el turista desea contratar, se genera para dicho paquete una factura, en el detalle figura el paquete. Un turista podrá pagar más de un paquete en una misma factura.

### **Funcionalidad necesaria del sistema(aspectos generales).**

A nivel funcional el sistema deberá realizar registro de todas las entidades indicadas, actualización de información, poder verificar si un Turista ya existe al ingresarlo. Validando el Dni.

Se va a requerir que cada ingreso de información o interacción con el usuario sea validado, por ejemplo no ingresar nombres de localidades sin texto, turistas, transportes, con datos completos, si se debe leer un dato numérico o fecha se deben validar, .. etc.

## Desarrollo en lenguaje C #

### Objetivos

Desarrollo de una aplicación interactiva que sea robusta y valide las diferentes acciones dadas por el usuario, que sea amigable y fácil de utilizar.

### **Requerimientos**

- Completar el modelo, según los ejemplos de referencia (modelo, relaciones, configuraciones).
- Desarrollar pantallas con validaciones para ingresar y actualizar información de las siguientes entidades:
  - Turistas (Casi Completo. en proyecto plantilla)
  - Agencias (En proceso, falta listado y posibilidad de exportar)
  - Configuraciones de Paquetes y tipos de paquetes, en cada paquete y sus actividades
  - Factura de turistas y sus detalles
  - Transportes
  - Configuraciones de Actividades y tipos de actividades
  - Localidades
  - Forma de Pago.
  - Países
  - Destinos que se vinculan a los paquetes.
- Gestionar en el sistema para producir listados
  - Para el caso de los listados pantallas deberán permitir opciones de ordenamiento y sentido. Por ejemplo, para listado de turistas puede ser por Nombre, Dni o país.
- Posibilitar la exportación a archivos la información del sistema, desde los listados (el usuario deberá indicar el nombre del archivo). Considerar las opciones de ordenamiento.
- Listados Resúmenes (se podrá exportar a archivo)
  - Listar destinos más concurridos.
  - Listar los países que mas turistas han visitado la zona.

Configuración para acceso a la base de datos PostgreSQL, en fuente **App.config** encontraran para configurar datos de servidor, usuario y Password, en la cadena de conexión, para conectarse.

### Se proveerá

- Proyecto de referencia para iniciar el desarrollo de la aplicación WinForms.
  - En el proyecto las clases para gestión ORM
  - Formularios para interacción
  - Código base de validación
  - Código base de inicialización de controles

## **Formularios**

Primero todos los fomularios deben estar en el espacio de nombre

`namespace TurApp.Views`

Tanto el archivo cs como el `cs.designer`, sino dará error de compilación.

Además, cada formulario de la solución deberá heredar de `FormBase`

Y deben incluir el espacio de nombres `using TurApp.db;`

Para tener acceso al modelo de datos.

También deben contar con botones `GuardarBtn` y `CancelarBtn`, uso normal de todo Formulario de actualización de datos.

De los formularios provistos se contará con:

- Formulario para realizar operación de Alta y Modificación integrado en un solo fuente, dependerá de la operación de las validaciones y que se permite actualizar, lo mismo la lógica para guardar nueva información o actualizar datos en la base de datos.
- Formulario par listar información de una entidad, con la aplicación de filtros
- Formulario para la búsqueda en una entidad.

Los formularios deben contar con métodos, para ingreso, modificación o consulta dependiendo del permiso del usuario, por ejemplo en el formulario `FrmTuristaAM.cs`

Existen `ShowModificarTurista` y `ShowIngresoTurista`

Estos métodos son la interfaz publica del formulario para dar soporte a esa funcionalidad, internamente se tiene una propiedad del Formulario que permite configurar el `caption`(título de la ventana) y controles a mostrar y habilitar.

## **Programación en los formularios**

Cada formulario de ingreso de información cuenta con lógica para la validación de los datos, con lo cual se programan conjuntos de eventos para que el usuario interactúe de forma ordenada en la operación de ingreso de datos.

Puede que el punto de acceso para el ingreso de la información esté en el Formulario principal o desde un botón en un formulario de segundo nivel.

Las actualizaciones de datos se deberán acceder desde la búsqueda y consulta de un listado y desde la grilla desplegar el formulario en modo actualización.

Las grillas deberán mostrar información relacionada, por ejemplo, para el caso del turista el país de origen, se debe ver el nombre de ese país y no un código. También deberán poder gestionar el ordenamiento al clicar la columna. (Ver ejemplo en resultado de turistas)

El formulario de búsqueda, deberá tener algunos filtros programados, si no se incluyen, listar todo.

## **Ajuste en la base de datos**

Para poder registrar los destinos, se agregó en la información de paquete el destino, para tener una referencia de un lugar donde aplica el paquete y después sacar un listado relacionado.

La clave del detalle de la factura se ajustó para poder tener más renglones de datos.

Un nuevo campo para agregar observaciones al turista.

Se provee un archivo `Script` para poder generar una serie de cambios en el sistema

## Definición de las clases y utilización del ORM

Las clases del dominio de aplicación se implementan en la carpeta **db** allí se proveen dos fuentes uno estático con las propiedades, de la clase y las relaciones y la metadata, por otro lado la implementación los métodos de la interface **IAccessDB** para implementar el acceso a la base de datos.

Métodos estáticos de las clases:

- **FindAllStatic**: Obtiene listado de instancias de un tipo de clase
- **FindByKeyStatic**: Obtiene una instancia por la clave

Metodos de las instancias.

- **FindAll**: Obtiene listado de instancias dado un criterio o todo.
- **FindByKey**: Obtiene una instancia según la clave.
- **SaveObj**: Ingresa o actualiza información de una instancia.

La clase **ORMDB** es la clase genérica que implementa los métodos antes mencionados, se vale de la metadata de las clases y obtiene desde las propiedades el mapeo a la tabla y columnas, que están en la base Postgres.

## El Modelo

A continuación, se detalla el conjunto de entidades:

|  |   |
|--|---|
| <ul style="list-style-type: none"> <li>- <b>localidad</b> <ul style="list-style-type: none"> <li>- cod_postal</li> <li>- nombre</li> </ul> </li> <li>- <b>destino</b> <ul style="list-style-type: none"> <li>- codigo</li> <li>- nombre</li> <li>- descripcion</li> </ul> </li> <li>- <b>pais</b> <ul style="list-style-type: none"> <li>- codigo</li> <li>- nombre</li> </ul> </li> <li>- <b>forma de pago</b> <ul style="list-style-type: none"> <li>- codigo</li> <li>- forma</li> </ul> </li> <li>- <b>tipos de actividad</b> <ul style="list-style-type: none"> <li>- codigo</li> <li>- nombre</li> <li>- descripcion</li> <li>- duracion</li> <li>- nivel</li> </ul> </li> <li>- <b>tipo paquete tipo actividad</b> <ul style="list-style-type: none"> <li>- legajo</li> <li>- dni</li> <li>- apellido</li> <li>- nombres</li> <li>- domicilio</li> <li>- telefono</li> </ul> </li> <li>- <b>factura</b> <ul style="list-style-type: none"> <li>- codigo</li> <li>- cod_tipo_act</li> <li>- anio</li> <li>- mes</li> <li>- importe</li> </ul> </li> <li>- <b>detalle factura</b> <ul style="list-style-type: none"> <li>- codigo</li> <li>- cod_act_socio</li> <li>- anio</li> <li>- mes</li> <li>- estado</li> <li>- importe</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>- <b>turistas</b> <ul style="list-style-type: none"> <li>- dni</li> <li>- apellido</li> <li>- nombres</li> <li>- domicilio</li> <li>- telefono</li> <li>- cod_paisl -- relación con Pais</li> </ul> </li> <li>- <b>actividades</b> <ul style="list-style-type: none"> <li>- codigo</li> <li>- cod_tipo_act</li> <li>- cod_transporte</li> <li>- importe</li> <li>- nivel</li> </ul> </li> <li>- <b>paquete actividad</b> <ul style="list-style-type: none"> <li>- codigo paquete</li> <li>- codigo actividad</li> <li>- importe</li> <li>- fecha hora desde</li> <li>- fecha hora hasta</li> <li>- detalle</li> </ul> </li> <li>- <b>tipo paquete</b> <ul style="list-style-type: none"> <li>- codigo</li> <li>- nombre</li> <li>- descripcion</li> <li>- duracion</li> <li>- nivel</li> </ul> </li> <li>- <b>paquete</b> <ul style="list-style-type: none"> <li>- codigo</li> <li>- cod tipo paquete</li> <li>- cod_agencia</li> <li>- fecha</li> <li>- dni_turista</li> <li>- nivel</li> </ul> </li> </ul> |
|--|---|