

Apellido y Nombre:

DNI: \_\_\_\_\_

**PRÁCTICA****Enunciado del Parcial - Gestión de Reservas de viajes:****1) Modelado de base de datos:**a) Crear en **SQL Server** la base de datos **ReservasViajes** y las siguientes tablas:i) **Reserva**: con los campos

(IdReserva INTEGER, CantidadPasajeros INTEGER, FechaInicio DATE, FechaFin DATE, IdDestino INTEGER, Eliminado BIT).

ii) **Destino**: con los campos

(IdDestino INTEGER, Nombre NVARCHAR(50)).

Relación:

- Una **Reserva** está asociada a un único **Destino**.
- Un **Destino** puede estar asociado a múltiples Reservas.

Por ejemplo, una reserva puede incluir un viaje para "5 pasajeros" al destino "París", con fechas de inicio y fin especificadas. El campo **Eliminado** será utilizado para implementar *soft delete* y marcar registros como eliminados sin borrarlos físicamente.

Configurar las claves primarias como **IDENTITY** y establecer la relación 1 a N entre **Destino** y **Reserva** mediante el campo **IdDestino**.

**2) Proyecto MVC**

- Crear una solución **GestionReservas.sln**, proyecto MVC llamado **GestionReservas.Web**, proyectos **GestionReservas.Data** y **GestionReservas.Lógica** para gestionar clientes, destinos y reservas.
- Utilizando Entity Framework (*Database First*), generar el modelo conceptual en una carpeta llamada EF.

**3) Funcionalidad ASP.NET:**

- Crear una página **/Reservas/Agregar.cshtml** para agregar nuevas Reservas, que contenga:
  - Un campo numérico para ingresar la **Cantidad de Pasajeros**.
  - Dos campos de selección de fecha para el **Inicio y Fin del viaje**.
  - Un combo que permita seleccionar un **Destino** (ordenado alfabéticamente).

- b) Un botón "**Registrar Reserva**".
- b) Generar el código necesario para almacenar la nueva Reserva en la base de datos con el valor por defecto de **Eliminado = 0**.

#### 4) Visualización de Reservas (/Reservas/Visualizar)

- a) Luego de agregar una Reserva, redirigir a una Vista **/Reservas/Visualizar.cshtml** que muestre todas las Reservas activas (donde **Eliminado = 0**), incluyendo:
  - i) Cantidad de Pasajeros.
  - ii) Nombre del Destino.
  - iii) Fecha de Inicio y Fecha de Fin.
- b) Crear una funcionalidad de **filtrado** que permita listar Reservas para un **Destino** específico, utilizando un combo que cargue los Destinos disponibles.

#### 5) Eliminación Lógica de Reservas

- a) En la lista de Reservas, agregar un botón "Eliminar" junto a cada registro. Al hacer clic en este botón, no se debe eliminar físicamente el registro, sino que se debe actualizar el campo **Eliminado** a **1** para marcarlo como eliminado.
- b) Luego de eliminar una Reserva, se debe refrescar la lista para que ya no aparezca en la visualización.

#### Aclaraciones:

- Se debe utilizar EF y LINQ o expresiones Lambda sin excepción.
- No es obligatorio controlar las validaciones de ingreso de información.
- Eliminar aquellas vistas que no tienen que ver con el enunciado, como por ejemplo Privacy y demás.
- Entregar un .zip con el formato **Apellido-Nombre-Parcial-2.zip**, recordar incluir script de sql creando la base de datos y tablas. Recordar excluir del .zip las carpetas /bin y /obj.
- Validar con el profesor que la entrega llegó.
- Tener abierta la solución y estar preparado para compartir pantalla para mostrar la resolución del ejercicio en caso de que el profesor así lo indique.