**DOCUMENTACION TECNICA: API REST PROYECTO CIBER**

**Alumno: Gonzalo Tomas Godoy Com:13**

**Actividades Integradas:**

* Arquitectura en capas
* Principios SOLID
* API REST con ASP.NET Core
* WinForms
* EF Core para acceso a datos
* Delegados y eventos (si aplica)
* Despliegue local (SQL Server Express)

**ProyectoCiber:** Es una aplicación .NET API Rest de gestión de turnos y PC/CONSOLAS de un Ciber.

**ProyectoCiber/**

| **Carpeta** | **Descripción** | **Archivos** |
| --- | --- | --- |
| **Controllers/** | Controladores API que gestionan las rutas HTTP | PCController.cs, TurnoController.cs, WeatherForecastController.cs |
| **Data/** | Contiene la clase del contexto de EF Core para acceso a base de datos | TurnoDBContext.cs |
| **Interfaces/** | Declaración de interfaces que definen contratos | IPCServicio.cs, ITurnoServicio.cs |
| **Models/** | Entidades que representan las tablas de la base de datos | PC.cs, Turno.cs |
| **Servicio/** | Lógica de negocio intermedia entre controller y base de datos | PCServicio.cs, TurnoServicio.cs |
| **Program.cs** | Configura la aplicación web ASP.NET Core |  |
| **appsettings.json** | Configuración general de la conexión a base de datos |  |

**Endpoints**

| **Acción** | **Método HTTP** | **Ruta** | **Descripción** |
| --- | --- | --- | --- |
| Obtener todos los Turnos | GET | /api/Turno | Devuelve la lista completa |
| Obtener Turno por ID | GET | /api/Turno/{id} | Devuelve un turno específico |
| Crear Turno | POST | /api/Turno | Agrega un nuevo turno |
| Actualizar Turno | PUT | /api/Turno/{id} | Modifica datos existentes |
| Eliminar Turno | DELETE | /api/Turno/{id} | Elimina un turno |

**Ejemplo: Turno/POST/Swagger**

**idTurnos “1”**

**id”16”**

**Pcid”3”**

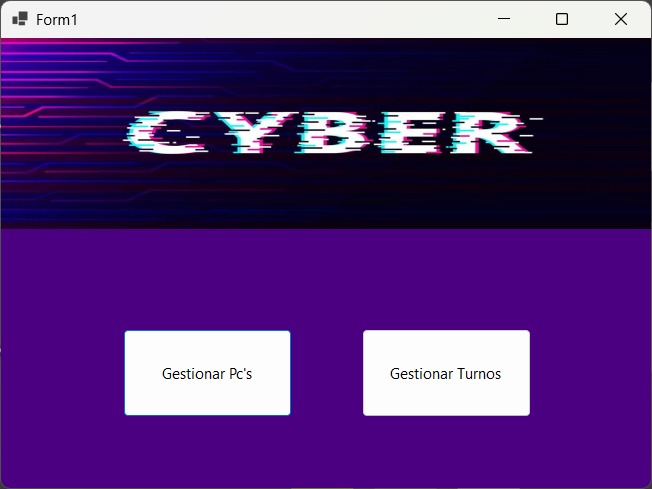
**HorariodeInicio “2025-06-18 15:45:37”**

**PRINCIPIOS SOLID APLICADOS**

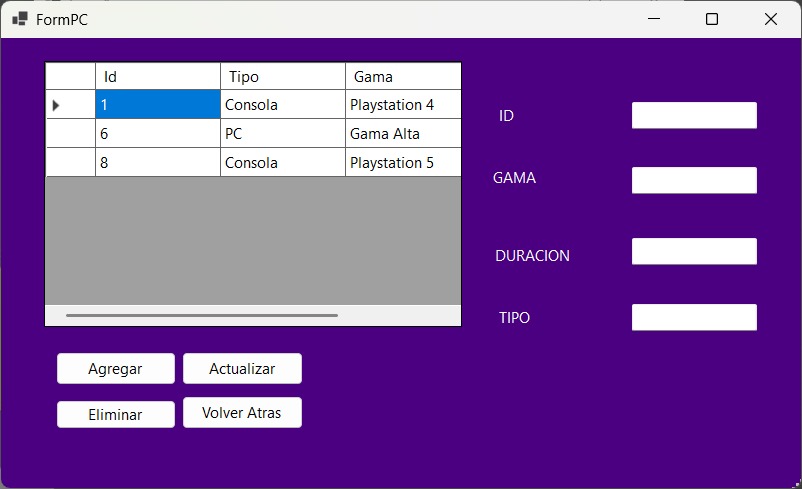
* **SRP:** cada clase tiene una única tarea. Ej: **PCServicio** se encargarsolo de la lógica de turnos.
* **OCP:** extiende servicios sin tocar el código base. Ej: **PCServicio** agrega validaciones sin tocar el código base.
* **LSP:** La implementación de **PCServico** puede remplazar interfaces con **IPCServicio** sin alterar nada.
* **ISP:** La interfaz **IPCServicio** está bien definida y no obliga a implementar métodos innecesarios.
* **DIP:** El controlador depende de la abstracción ITurnoServicio y no de una clase en específico.

**Manual para el usuario / Ciber**

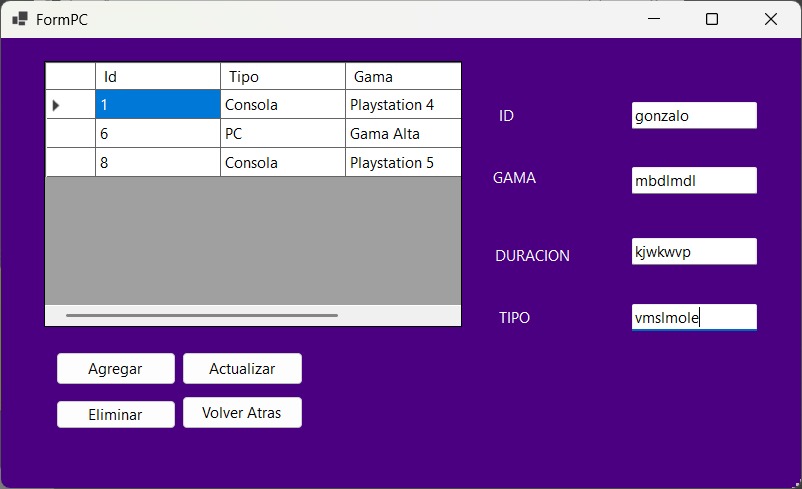
1. En inicio se muestra una ventana con dos botones una para seleccionar la gestión de pc y otra para la gestión de un turno.

****

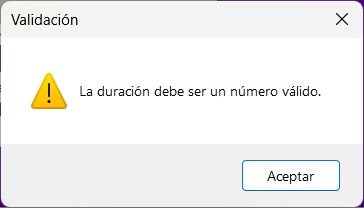
**2.** Si se selecciona **Gestionar Pc’s** se nos desplegara una ventana para agregar, editar yeliminar ya sea Pc o Consolas con sus diferentes tipos/gamas, ademas de un boton para volver a la pantalla anterior. Para Agregar se tiene que llenar todos los campos, para Eliminar solo solicita la Id y para Editar el mismo caso.



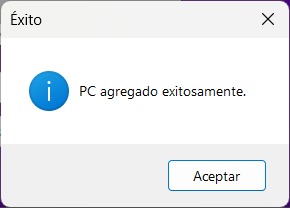
**3.** Si los datos estan erroneamente puestos en los campos que se ven en pantalla.



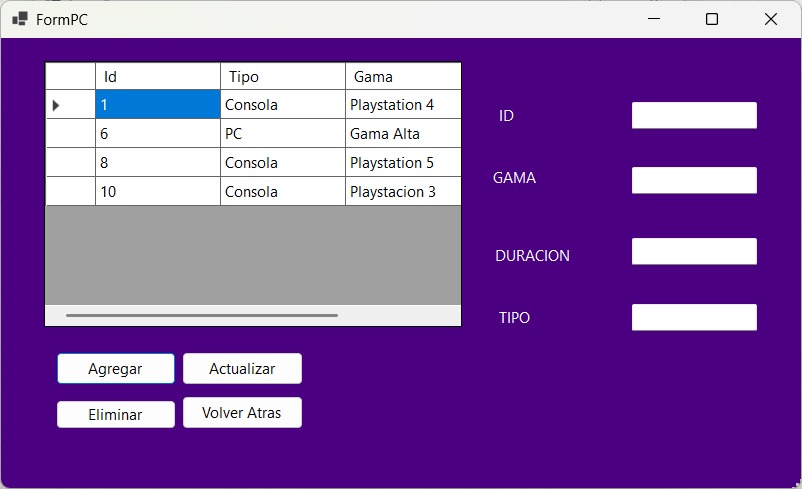
Se desplegará una ventana al seleccionar unos de los botones (agregar, actualizar y eliminar) advirtiendo de los mismos errores. Por ejemplo la próxima imagen:



**4.** En caso de llenar correctamente los campos anteriormente mismo se desplegará una ventana de que se guardó exitosamente la misma.

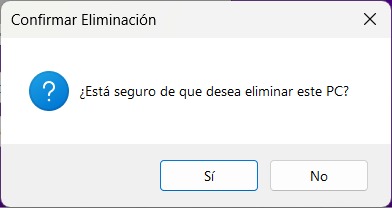


Y se vera en el formulario como se agregó una fila con los datos de los pc anteriormente puestos por uno mismo.

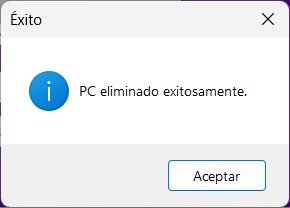


**5.** En caso de querer eliminar unas de las filas vistas en pantalla solamente tendrá que poner el id correspondiente a la fila a eliminar y dar click sobre el mismo botón **“Eliminar”.**

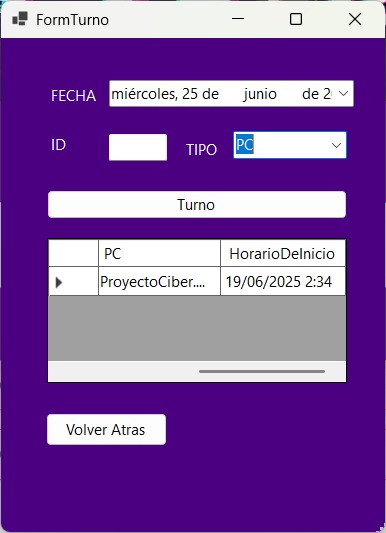
Después de hacer estas acciones se desplegará una ventana preguntando **“¿Estás seguro de que desea eliminar este PC?”**



Al presionar la opción **No** se volverá a la anterior ventana y pero al darle **Si** hay que esperar que se muestre la ventana de “**PC exitosamente eliminada**” para confirmar.

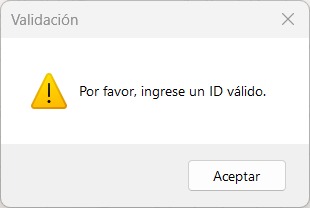


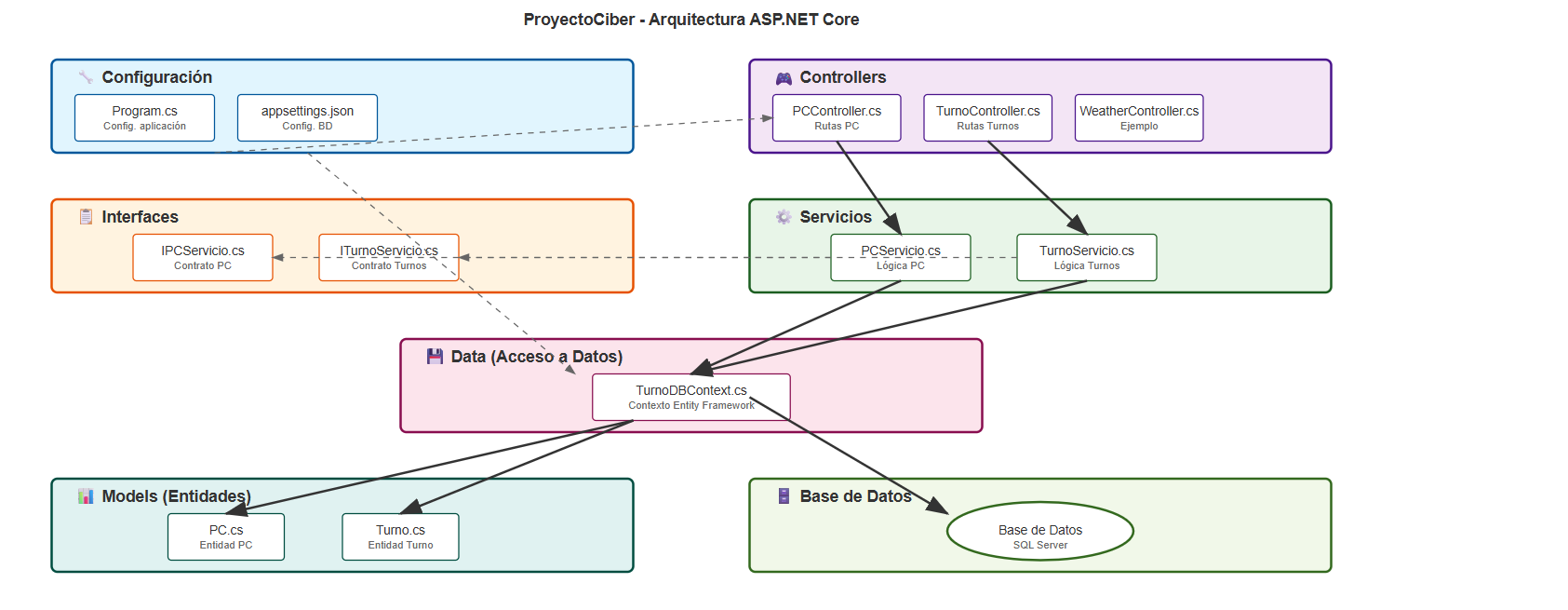
**7.** Si se selecciona el botón de “Gestionar turnos” se mostrara la ventana de para gestionar un turno con la id, el tipo (Consola o Pc) y la fecha.



La ventana mostrara una lista para elegir la fecha, seleccionar un id de la PC y el tipo (PC o Consola).

**6.** En caso de completar mal los campos aparecerá la siguiente ventana:







* **Configuración** - Program.cs y appsettings.json configuran la aplicación
* **Controllers** - Manejan las peticiones HTTP y actúan como punto de entrada
* **Servicios** - Contienen la lógica de negocio y actúan como intermediarios
* **Interfaces** - Definen contratos para la inyección de dependencias
* **Data** - Contexto de Entity Framework para acceso a datos
* **Models** - Entidades que representan las tablas de la base de datos
* **Base de Datos** - Almacenamiento de datos.