**Primera entrega trabajo final**

**Seminario .NET**

Blanco Valentín (24549/6)

Cabe Pablo (23056/4)

Isla Federico (26241/6)

Manese Martín (23105/5)

Mediante este texto se detallará la funcionalidad del trabajo desarrollado desde Program.cs.

**Aclaraciones**

Los archivos de texto referidos a la persistencia de datos y los archivos de texto de IDs se crearán en la ruta **CentroEventos.Consola\bin\Debug\net8.0**

Se decidió utilizar archivos de texto para persistir los IDs. Además, se aplicó una clase de utilidad para el manejo de ellos (**IdManager.cs**). Esta misma emplea dos métodos:

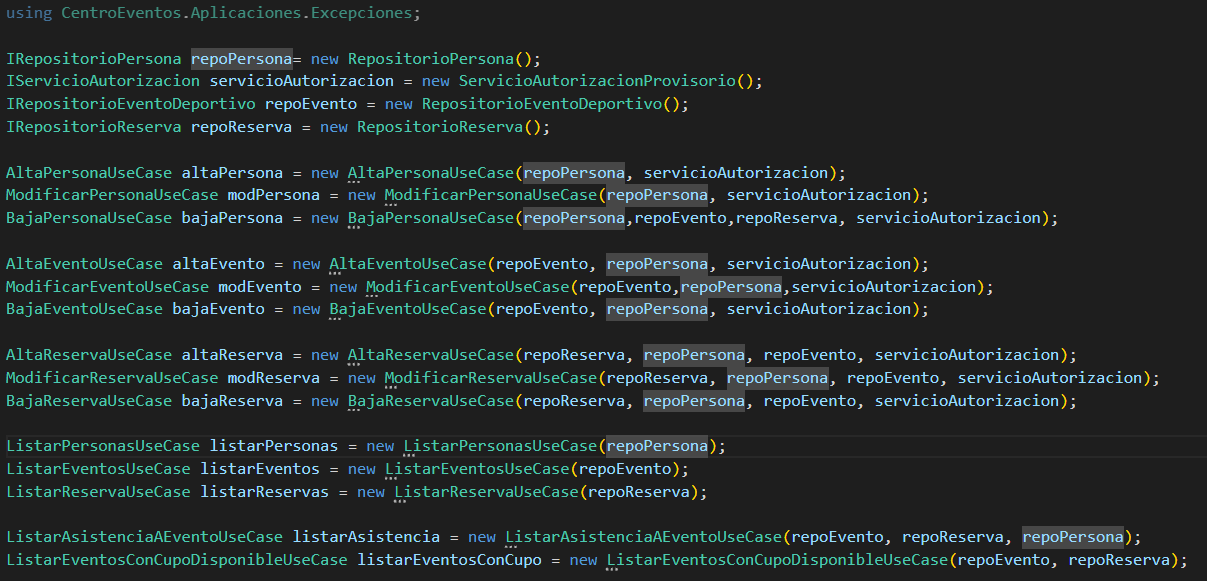
* BuscarUltimoId: si el archivo no existe retorna -1. En caso contrario devuelve el último id.
* ActualizarArchivoId: actualiza el último archivo o lanza la excepción si el archivo no existe o no es posible escribirlo.

Cada repositorio adquirirá la responsabilidad de autoincrementar los respectivos IDs.

**Funcionalidad**

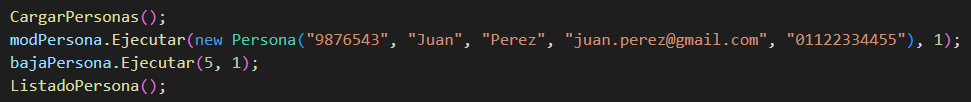
En primer lugar, se instanciarán los **repositorios** ubicados en CentroEventos.Repositorios y los **casos de uso**, situados en CentroEventos.Aplicacion.CasosDeUso.

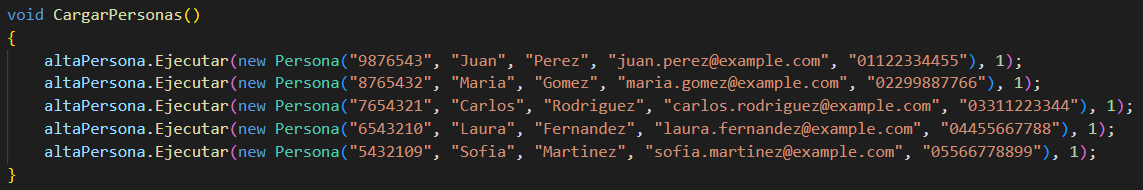
Cabe destacar que los repositorios utilizan las **interfaces** creadas en el directorio CentroEventos.Aplicacion.Interfaces, con la intención de aplicar la inyección de dependencias. Esta resultará útil de cara al futuro para la implementación de otro servicio de persistencia.

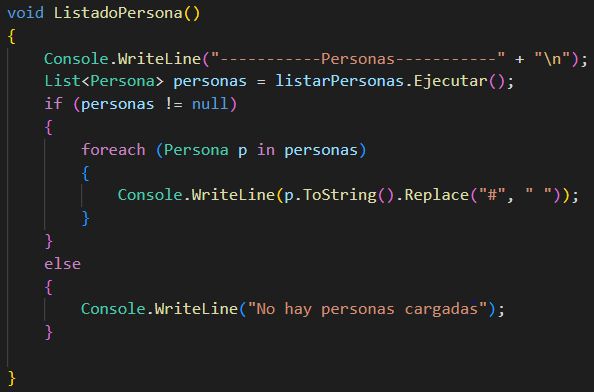


Una vez instanciados los casos de uso y los repositorios, se emplea un try catch para probar la persistencia de datos de las entidades correspondientes. En este paso, se cargan las personas, los eventos y las reservas.

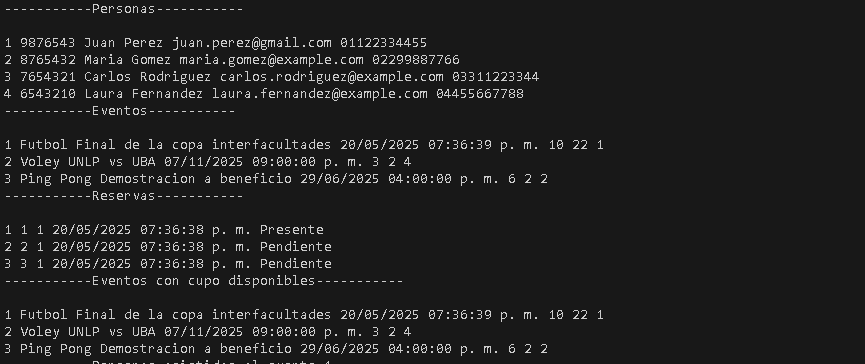
**Código de ejemplo**





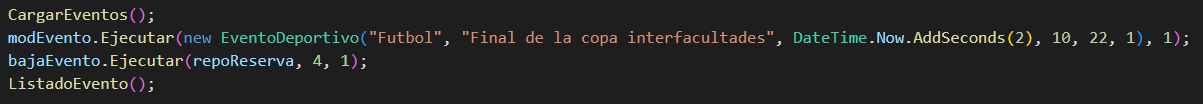


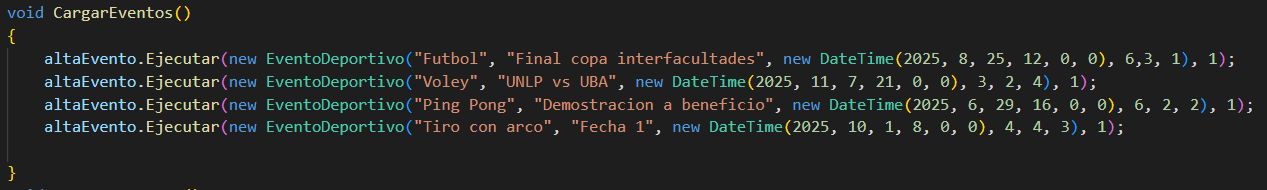
En primer lugar, se cargarán las personas mediante el caso de uso AltaPersona.Ejecutar. Luego se modifica la persona y por último se da de baja la persona N°5 (Sofía), por lo cual se imprimirá el listado sin ella, tal como se observa a continuación:

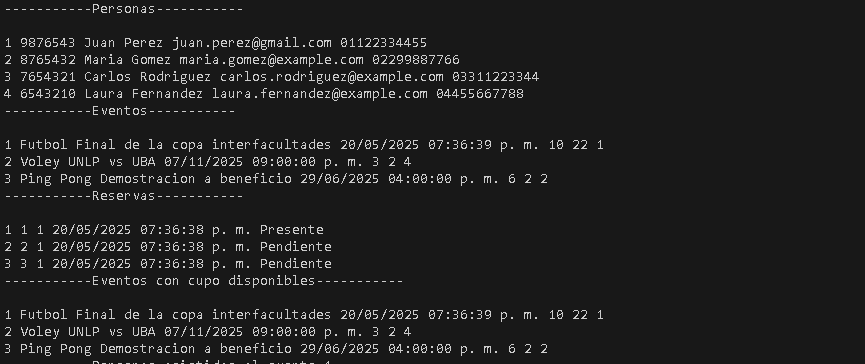


En segundo paso, cargamos los eventos a través del caso de uso AltaEvento.Ejecutar. Cabe destacar que la entidad EventoDeportivo cuenta con la propiedad FechaHoraInicio, la cual al asignarla es necesario agregar dos segundos a la fecha actual para que sea posible listar las personas de los eventos vencidos.

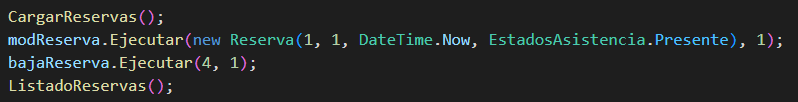
Aclaración: es necesario ejecutar el Program.cs y esperar dos segundos para volver a ejecutarlo debido a la validación VerFecha, la cual determina si la fecha a crear/modificar es mayor o igual a la fecha actual. En otras palabras, es imposible ejecutarlo sin esperar el tiempo vencido. Asimismo, para ejecutar el método se deben comentar los métodos que modifiquen a los archivos.

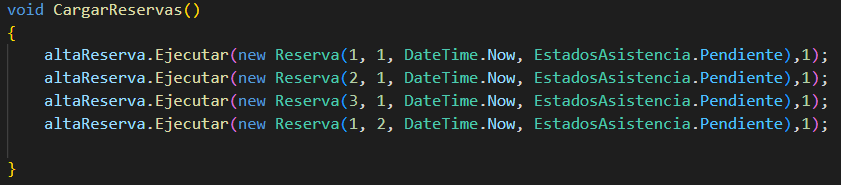


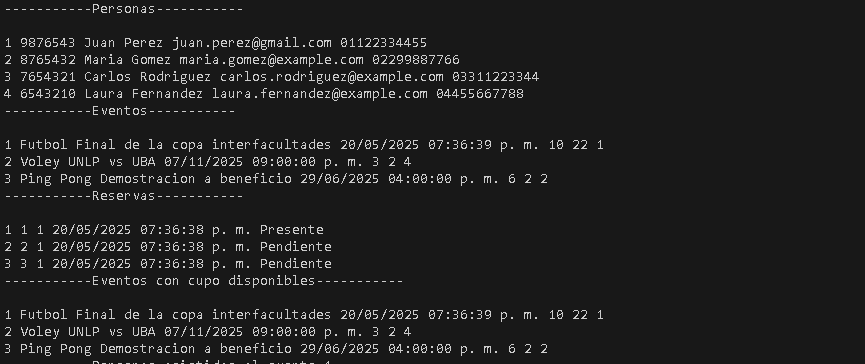




En tercer lugar, con respecto a las reservas, el funcionamiento es similar al de cargarPersona. A continuación se realiza la baja de la reserva con ID 4.





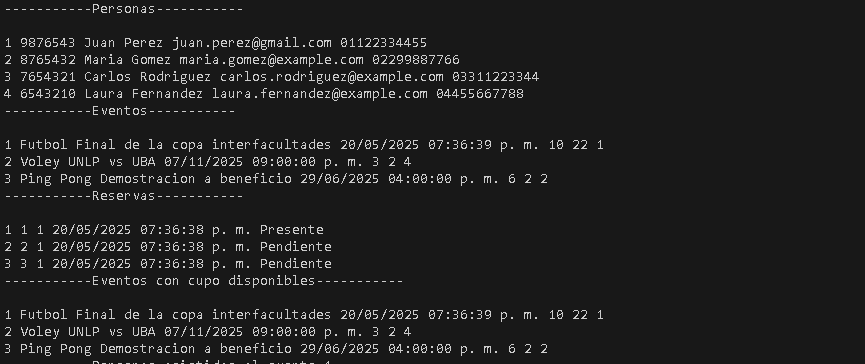


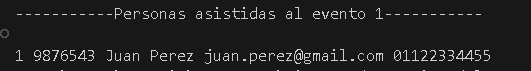
ListadoEventosDisponibles: en este se implementa el caso de uso ListarEventosConCupoDisponible, el cual muestra los eventos que poseen disponibilidad de cupo.

ListarPersonasAsistidas: se muestran las personas presentes en el evento con ID = 1.



**Salida esperada por consola**





Esta es la salida del método que se debe ejecutar pasados dos segundos.