

# Como instanciar un Repositorio

**Repositorio X:** Para crear el repositorio o para instanciarlo, se debe pasar como parámetro el Path(**string**) de el archivo en donde se guardaran las entidades (**Persona**, **EventoDeportivo** o **Reservas**) y como segundo parámetro se debe pasar el Path(**string**) en donde se encuentra el ultimo ID registrado.

```
RepositorioPersona(Path,PathIDs);  
RepositorioEventoDeportivo(Path,PathIDs);  
RepositorioReserva(Path,PathIDs);
```

C# Program.cs X

```
1  using CentroEventos.Aplicacion;  
2  using CentroEventos.Repositorios;  
3  
4  //instancian los Repositorios(Path en donde se almacenan las entidades y los ids de cada repositorio)  
5  var repoPer = new RepositorioPersona(@"c:\Archivos\Personas.txt",@"c:\Archivos\idPersona.txt");  
6  var repoEve = new RepositorioEventoDeportivo(@"c:\Archivos\Eventos.txt",@"c:\Archivos\idEvento.txt");  
7  var repoRes = new RepositorioReserva(@"c:\Archivos\Reservas.txt", @"c:\Archivos\idPreserva.txt");  
8
```

Path Archivo de entidades

Path de Archivo con el ultimo ID

# Como instanciar Los Validadores

**Validador X:** Se encargara de validar los datos de las entidades que agregues, Para cada entidad existe un validador especifico y cada una requiere Repositorios especificos para poder validar:

`New ValidacionPersona(RepositorioPersona);`

`New ValidacionReserva(RepositorioPersona,RepositorioEventoDeportivo,RepositorioReserva);`

`New ValidacionEventoDeportivo(RepositorioPersona);`

y existe uno que solo se le asigna a los eventos y reservas que es un validador de permisos:

`New ServicioAutorizacionProvisorio();`

```
4  string personaPath = @"c:\Archivos\Personas.txt";
5  string idPersonasPath = @"c:\Archivos\IDsPersonas.txt";
6  string eventoPath = @"c:\Archivos\Eventos.txt";
7  string idEventosPath = @"c:\Archivos\IDsEventos.txt";
8  string reservasPath = @"c:\Archivos\Reservas.txt";
9  string idReservasPath = @"c:\Archivos\IDsReservas.txt";
10
11 // Instanciar repositorios y validaciones
12 var repoPersonas = new RepositorioPersona(personaPath,idPersonasPath);
13 var repoEventos = new RepositorioEventoDeportivo(eventoPath,idEventosPath);
14 var repoReservas = new RepositorioReserva(reservasPath, idReservasPath);
15
16
17 var servicioAutorizacion = new ServicioAutorizacionProvisorio();
18
19 var validacionPersona = new ValidacionPersona(repoPersonas);
20
21 var validacionEvento = new ValidacionEventoDeportivo(repoPersonas);
22
23 var validacionReserva = new ValidacionReserva(repoPersonas, repoEventos, repoReservas);
24
```

# Como Agregar Una Entidad

**AgregarPersonaUseCase()**: Este UseCase nos permitirá agregar a una persona a el repositorio de personas, para esto se debe instanciar el caso de uso y se debe pasar por parámetro el repositorio de Persona y un **validador de Persona** al que también se le debe pasar el repositorio por parámetro por ejemplo:

```
var validador = new ValidacionPersona(RepositorioPersona);  
var agregarPersona = New AgregarPersonaUseCase(RepositorioPersona,Validador);
```

**agregarPersona.Ejecutar(Persona)**: Ejecutara el UseCase,para eso Tendrá que recibir una **Persona** como parámetro para poder agregarla al repositorio, por lo que se tiene que instanciar antes una **Persona** o directamente instanciarla al llamar al método

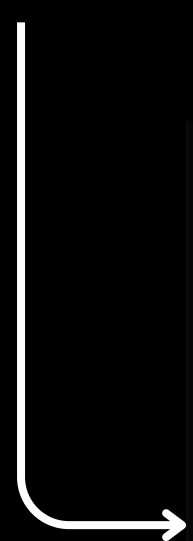
```
C# Program.cs •  
1  using CentroEventos.Aplicacion;  
2  using CentroEventos.Repositorios;  
3  
4  //instancian los Repositorios(Path en donde se almacenan las entidades y los ids de cada repositorio)  
5  var repoPer = new RepositorioPersona(@"c:\Archivos\Personas.txt",@"c:\Archivos\idPersona.txt");  
6  var repoEve = new RepositorioEventoDeportivo(@"c:\Archivos\Eventos.txt",@"c:\Archivos\idEvento.txt");  
7  var repoRes = new RespositorioReserva(@"c:\Archivos\Reservas.txt", @"c:\Archivos\idPreserva.txt");  
8  
9  
10 var agregarPersona = new AgregarPersonaUseCase(repoPer, new ValidacionPersona(repoPer));  
11  
12  
13 var persona = new Persona() { Nombre = "Sofía", Apellido = "Ramírez", DNI = "33445566", Email = "sofia.ramirez@email.com", Telefono = "1122334455" };  
14  
15  
16 agregarPersona.Ejecutar(persona );  
17  
18
```

Para Agregar **Eventos Deportivos** y Agregar **Reservas** es el mismo proceso, se instancia el caso de uso pero con diferentes repositorios pasados por parámetros y a diferencia de **AgregarPersona** a estos también se les pasa el servicio de autorización y por ultimo su respectivo validador

```
new AgregarReservaUseCase(RepositorioReserva, ServicioDeAutorizacion, ValidarReserva);  
    .Ejecutar(Entidad , idUsuario);  
new AgregarEventoDeportivoUseCase(RepositorioEventoDeportivo, ServicioDeAutorizacion, ValidarEvento);
```

# Luego de agregar A Sofia al repositorio de Personas

Se Agrego a la Persona con nombre Sofia y se le genero el ID 5



```
~/Documents/GitHub/Proyecto.NET/CentroEventos/CentroEventos.Consola git:(main)±1 (1.925s)
dotnet run

ID: 1, DNI: 12131311, Nombre: Alex, Apellido: Javier, Email: alex.javier@email.com, Telefono: 221212311
ID: 2, DNI: 30123456, Nombre: Ana, Apellido: García, Email: ana.garcia@email.com, Telefono: 123456789
ID: 3, DNI: 28987654, Nombre: Luis, Apellido: Martínez, Email: luis.martinez@email.com, Telefono: 987654321
ID: 4, DNI: 31234567, Nombre: María, Apellido: López, Email: maria.lopez@email.com, Telefono: 456123789
ID: 5, DNI: 33445566, Nombre: Sofía, Apellido: Ramírez, Email: sofia.ramirez@email.com, Telefono: 1122334455
```



# Como Eliminar una Entidad

**EliminarPersonaUseCase()**: Este UseCase nos permitirá eliminar a una persona del repositorio de personas, para esto se debe instanciar el caso de uso y se debe pasar por parámetro el repositorio de Persona, el Repositorio de Eventos y el Repositorio de Reservas para que pueda verificar que la entidad no tiene reservas en algún evento:

**new EliminarPersonaUseCase(RepositorioPersona,RepositorioEventoDeportivo,RepositorioReserva);**

**.Ejecutar(idPersona)**: Ejecutara el UseCase,para eso Tendrá que recibir el ID(**Int**) por parametro de la persona que desea eliminar del repositorio.

```
C# Program.cs X
1  using CentroEventos.Aplicacion;
2  using CentroEventos.Repositorios;
3
4  //instancian los Repositorios(Path en donde se almacenan las entidades y los ids de cada repositorio)
5  var repoPer = new RepositorioPersona(@"c:\Archivos\Personas.txt",@"c:\Archivos\idPersona.txt");
6  var repoEve = new RepositorioEventoDeportivo(@"c:\Archivos\Eventos.txt",@"c:\Archivos\idEvento.txt");
7  var repoRes = new RepositorioReserva(@"c:\Archivos\Reservas.txt", @"c:\Archivos\idPReserva.txt");
8
9
10 var eliminarPersona = new EliminarPersonaUseCase(repoPer, repoEve,repoRes);
11
12
13 eliminarPersona.Ejecutar(1);
14
```

Para Eliminar **Eventos Deportivos** y **Reservas** es el mismo proceso lo único que varia son los parámetros que se tienen que pasar al UseCase de cada uno y que al momento de ejecutar también se pasa el id de usuario

**new EliminarReservaUseCase(RepositorioReserva,ServicioDeAutorizacion)**

**.Ejecutar(idEntidad , idUsuario);**

**new EliminarEventoDeportivoUseCase(RepositorioEventoDeportivo,ServicioDeAutorizacion,RepositorioReserva)**

# luego de Eliminar a ID = 1 e imprimir el repositorio

~/Documents/GitHub/Proyecto.NET/CentroEventos/CentroEventos.Consola git:(main)±1 (2.06s)



**dotnet run**

ID: 2, DNI: 30123456, Nombre: Ana, Apellido: García, Email: ana.garcia@email.com, Telefono: 123456789  
ID: 3, DNI: 28987654, Nombre: Luis, Apellido: Martínez, Email: luis.martinez@email.com, Telefono: 987654321  
ID: 4, DNI: 31234567, Nombre: María, Apellido: López, Email: maria.lopez@email.com, Telefono: 456123789  
ID: 5, DNI: 33445566, Nombre: Sofía, Apellido: Ramírez, Email: sofia.ramirez@email.com, Telefono: 1122334455

# Como Modificar una Entidad

**ModificarPersonaUseCase()**: Este UseCase nos permitirá modificar a una persona del repositorio de personas, para esto se debe instanciar el caso de uso y pasar por parámetro el repositorio de Persona:

**new ModificarPersonaUseCase(RepositorioPersona);**

**.Ejecutar(idPersona,Persona)**: Ejecutara el UseCase, para eso Tendrá que recibir el ID(Int) por parámetro de la persona que desea modificar del repositorio y una **Persona** instanciada con los datos modificados.

```
C# Program.cs 1 X
1  using CentroEventos.Aplicacion;
2  using CentroEventos.Repositorios;
3
4  int idUsuario = 1;  The variable 'idUsuario' is assigned but its value is never used
5  string personaPath = @"c:\Archivos\Personas.txt";
6  string idPersonasPath = @"c:\Archivos\IDsPersonas.txt";
7
8  // Instanciar repositorios y validaciones
9  var repoPersonas = new RepositorioPersona(personaPath,idPersonasPath);
10
11
12  var modificarPersonaUC = new ModificarPersonaUseCase(repoPersonas);
13
14  //Cambio Email Y nro de Telefono
15  var persona = new Persona() { Nombre = "Leandro", Apellido = "Paredes", Email = "Leandro1213@email.com", DNI = "35672323", Telefono = "2215543434" };
16
17
18  modificarPersonaUC.Ejecutar(6, persona);
19
```

Para la modificación de **Eventos Deportivos** y **Reservas** es el mismo proceso lo único que varia es el **repositorio** que se envía por parámetros, y que ademas se envía el **servicio de Autorizacion** para el **IDdeUsuario**

**new MoficarReservaUseCase(RepositorioReserva, ServicioDeAutorizacion)**

**.Ejecutar(idEntidad , Entidad);**

**new EliminarEventoDeportivoUseCase(RepositorioEventoDeportivo, ServicioDeAutorizacion)**

# Lista del repositorio Personas luego de Modificar

```
~/Documents/GitHub/Proyecto.NET/CentroEventos/CentroEventos.Consola git:(main)±1 (1.927s)
```

```
dotnet run
```

```
ID: 1, DNI: 35678901, Nombre: Lucia, Apellido: Fernández, Email: fernandez@email.com, Telefono: 3344556677  
ID: 2, DNI: 28987654, Nombre: Luis, Apellido: Martínez, Email: luis.martinez@email.com, Telefono: 987654321  
ID: 3, DNI: 33445566, Nombre: Sofía, Apellido: Ramírez, Email: sofia.ramirez@email.com, Telefono: 1122334455  
ID: 4, DNI: 34561234, Nombre: Javier, Apellido: Suárez, Email: javier.suarez@email.com, Telefono: 2211334455  
ID: 6, DNI: 35672323, Nombre: Leandro, Apellido: Paredes, Email: L.Paredes@email.com, Telefono: 334455211
```

cambio de numero y Email

```
~/Documents/GitHub/Proyecto.NET/CentroEventos/CentroEventos.Consola git:(main)±1 (2.171s)
```

```
dotnet run
```

```
C:\Users\Dylan\Documents\GitHub\Proyecto.NET\CentroEventos\CentroEventos.Consola\Program.cs(4,5): warning CS0219: La variable 'idUsuario' está asignada pero su valor nunca se usa  
ID: 1, DNI: 35678901, Nombre: Lucia, Apellido: Fernández, Email: fernandez@email.com, Telefono: 3344556677  
ID: 2, DNI: 28987654, Nombre: Luis, Apellido: Martínez, Email: luis.martinez@email.com, Telefono: 987654321  
ID: 3, DNI: 33445566, Nombre: Sofía, Apellido: Ramírez, Email: sofia.ramirez@email.com, Telefono: 1122334455  
ID: 4, DNI: 34561234, Nombre: Javier, Apellido: Suárez, Email: javier.suarez@email.com, Telefono: 2211334455  
ID: 6, DNI: 35672323, Nombre: Leandro, Apellido: Paredes, Email: Leandro1213@email.com, Telefono: 2215543434
```



# Como Listar las Entidades de un repositorio

**ListarPersonasUseCase()**: Este UseCase retorna todas las personas del repositorio de personas a una lista, para poder hacerlo debemos instanciar el caso de uso pasándole por parámetro el repositorio de Personas :

**new ListarPersonasUseCase(RepositorioPersona);**

**.Ejecutar()**: Ejecutara el UseCase que retornara una **Lista<Persona>** por lo que debe ser asignada a una variable ejem:

**var x = listarPersonas.Ejecutar();**

C# Program.cs X

```
1  using CentroEventos.Aplicacion;
2  using CentroEventos.Repositorios;
3
4  //instancian los Repositorios(Path en donde se almacenan las entidades y los ids de cada repositorio)
5  var repoPer = new RepositorioPersona(@"c:\Archivos\Personas.txt",@"c:\Archivos\idPersona.txt");
6  var repoEve = new RepositorioEventoDeportivo(@"c:\Archivos\Eventos.txt",@"c:\Archivos\idEvento.txt");
7  var repoRes = new RespositorioReserva(@"c:\Archivos\Reservas.txt", @"c:\Archivos\idPReserva.txt");
8
9
10 var listarPersonas = new ListarPersonasUseCase(repoPer);
11
12 var lista = listarPersonas.Ejecutar();
```

Para Listar cualquiera de las 3 entidades se hace el mismo procedimiento, 1. se instancia el caso de uso correspondiente según la entidad, 2. se le inyecta el repositorio correspondiente por ejm: **var listarReservas = new ListarReservas(RepositorioReserva);**  
3. Ejecutar el caso de uso : **var List = listarReservas.Ejecutar();**

# Listar Eventos Con Cupos Disponibles

**ListarEventosConCupoDisponibleUseCase()**: Este UseCase retorna todos los eventos que tienen cupos disponibles, para eso se lo debe instanciar pasándole por parámetros el repositorio de Eventos y el Repositorio de Reservas :

**new ListarEventosConCupoDisponibleUseCase(RepositorioEventoDeportivo,RepositorioReserva);**

**.Ejecutar()**: Ejecutara el UseCase que retornara una **Lista<EventoDeportivo>** por lo que debe ser asignada a una variable ejem:

**var x = listarEventosConCupo.Ejecutar();**

```
5  string personaPath = @"c:\Archivos\Personas.txt";
6  string idPersonasPath = @"c:\Archivos\IDsPersonas.txt";
7  string eventoPath = @"c:\Archivos\Eventos.txt";
8  string idEventosPath = @"c:\Archivos\IDsEventos.txt";
9  string reservasPath = @"c:\Archivos\Reservas.txt";
10 string idReservasPath = @"c:\Archivos\IDsReservas.txt";
11
12 // Instanciar repositorios y validaciones
13 var repoPersonas = new RepositorioPersona(personaPath,idPersonasPath);
14 var repoEventos = new RepositorioEventoDeportivo(eventoPath,idEventosPath);
15 var repoReservas = new RepositorioReserva(reservasPath,idReservasPath);
16 var servicioAutorizacion = new ServicioAutorizacionProvisorio();
17
18
19
20
21 var listarEventosConCupos = new ListarEventosConCupoDisponibleUseCase(repoEventos, repoReservas);
22
23 var lista = listarEventosConCupos.Ejecutar();
```

# Eventos Con Cupo Disponible

## Lista Completa

```
~/Documents/GitHub/Proyecto.NET/CentroEventos/CentroEventos.Consola git:(main)±2 (2.145s)
dotnet run

ID: 1, Nombre: Torneo de Fútbol, Descripción: Competencia anual de fútbol, Fecha de inicio: 30/5/2024 17:20:36, Cupo máximo: 2, ID del responsable: 1
ID: 2, Nombre: Copa de Básquet, Descripción: Torneo regional de básquet, Fecha de inicio: 27/5/2025 17:20:36, Cupo máximo: 1, ID del responsable: 2
ID: 3, Nombre: Carrera de Atletismo, Descripción: Torneo intercolegial de atletismo, Fecha de inicio: 28/5/2025 09:00:00, Cupo máximo: 25, ID del responsable: 3
ID: 4, Nombre: Torneo de Ajedrez, Descripción: Competencia anual de ajedrez, Fecha de inicio: 29/5/2025 15:30:45, Cupo máximo: 2, ID del responsable: 1
ID: 5, Nombre: Maratón Solidaria, Descripción: Evento deportivo benéfico, Fecha de inicio: 30/5/2025 07:45:10, Cupo máximo: 5, ID del responsable: 4
ID: 6, Nombre: Partido de Vóley, Descripción: Torneo mixto de vóley, Fecha de inicio: 31/5/2025 18:10:20, Cupo máximo: 1, ID del responsable: 2
```



```
~/Documents/GitHub/Proyecto.NET/CentroEventos/CentroEventos.Consola git:(main)±2 (2.129s)
dotnet run

ID: 3, Nombre: Carrera de Atletismo, Descripción: Torneo intercolegial de atletismo, Fecha de inicio: 28/5/2025 09:00:00, Cupo máximo: 25, ID del responsable: 3
ID: 4, Nombre: Torneo de Ajedrez, Descripción: Competencia anual de ajedrez, Fecha de inicio: 29/5/2025 15:30:45, Cupo máximo: 2, ID del responsable: 1
ID: 5, Nombre: Maratón Solidaria, Descripción: Evento deportivo benéfico, Fecha de inicio: 30/5/2025 07:45:10, Cupo máximo: 5, ID del responsable: 4
ID: 6, Nombre: Partido de Vóley, Descripción: Torneo mixto de vóley, Fecha de inicio: 31/5/2025 18:10:20, Cupo máximo: 1, ID del responsable: 2
```

## Lista Con Cupos

# Listar Asistentes a Eventos Pasados

**ListarAsistenciaAEventoUseCase()**: Este UseCase retorna todas las personas que asistieron a algun evento pasado, para eso se lo debe instanciar pasándole por parámetros el repositorio de Eventos , el Repositorio de Reservas y el repositorio de Personas :

**new**

**ListarAsistenciaAEventoUseCase(RepositorioEventoDeportivo,RepositorioReserva,RepositorioPersona);**

**.Ejecutar()**: Ejecutara el UseCase que retornara una **Lista<Persona>** por lo que debe ser asignada a una variable ejem:

**var x = listarAsistentesAEvento.Ejecutar();**

```
10  string idReservasPath = @"c:\Archivos\IDsReservas.txt";
11
12  // Instanciar repositorios y validaciones
13  var repoPersonas = new RepositorioPersona(personaPath,idPersonasPath);
14  var repoEventos = new RepositorioEventoDeportivo(eventoPath,idEventosPath);
15  var repoReservas = new RepositorioReserva(reservasPath,idReservasPath);
16  var servicioAutorizacion = new ServicioAutorizacionProvisorio();
17
18
19
20
21  var asistentesAEvento = new ListarAsistenciaAEventoUseCase(repoEventos, repoReservas, repoPersonas);
22
23  var lista = asistentesAEvento.Ejecutar();
24
```



# Personas que Asistieron a un evento pasado

## Lista de Reservas

```
~/Documents/GitHub/Proyecto.NET/CentroEventos/CentroEventos.Consola git:(main)±2 (4.805s)
dotnet run

ID: 1, Persona ID: 3, Evento deportivo ID: 2, Fecha de alta de reserva: 20/5/2025 17:53:39, Estado de asistencia: Pendiente
ID: 2, Persona ID: 6, Evento deportivo ID: 1, Fecha de alta de reserva: 20/5/2025 18:53:39, Estado de asistencia: Presente
ID: 3, Persona ID: 3, Evento deportivo ID: 1, Fecha de alta de reserva: 20/5/2025 19:53:39, Estado de asistencia: Presente
```

## Lista de las Personas que asistieron a un evento pasado

```
~/Documents/GitHub/Proyecto.NET/CentroEventos/CentroEventos.Consola git:(main)±2 (2.099s)
dotnet run

ID: 6, DNI: 35672323, Nombre: Leandro, Apellido: Paredes, Email: Leandro1213@email.com, Telefono: 22155434
34
ID: 3, DNI: 33445566, Nombre: Sofía, Apellido: Ramírez, Email: sofia.ramirez@email.com, Telefono: 11223344
55
```