

# Universidad ORT Uruguay

Escuela de Tecnología

## Programación 3 Docente: Luis Dentone

Obligatorio 1

Grupo: N3B

Joaquin Acuña - 213406



[JoaquinAcuna97](#)

## Índice

### Diagrama de casos de uso

#### Descripción Narrativa de los casos de uso

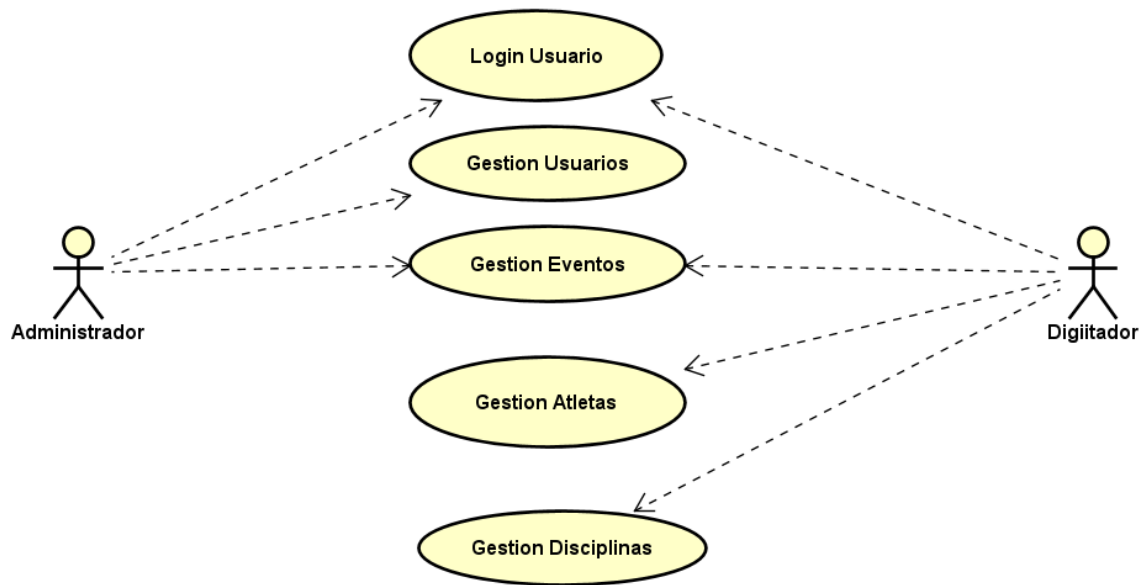
Alta de Atleta

Ingreso de puntaje de atleta

### Diagrama de Clases UML

#### Código fuente de las clases del dominio

## Diagrama de casos de uso



# Descripción Narrativa de los casos de uso

## Alta de Evento

<b>Identificador:</b> CU-01	<b>Nombre:</b> Registrar un Evento
<b>Descripción:</b> Permite registrar un nuevo evento en el sistema, dejándolo disponible para su visionado.	
<b>Actor/es:</b> Usuario registrado con rol administrador o digitador	
<b>Precondiciones:</b> Existen disciplinas cargadas y al menos 3 atletas de esa disciplina	
<b>Pos condiciones:</b> El evento quedó registrado en el catálogo con todos sus datos, se le asignó un código numérico y queda disponible para ser visualizado.	
<b>Flujo Normal:</b> <ol style="list-style-type: none"><li>1. El usuario ingresa a agregar nuevo evento.</li><li>2. El sistema solicita Nombre, fecha de inicio, fecha de fin, Disciplina y atletas.</li><li>3. El actor introduce los datos solicitados y solicita continuar el ingreso.</li><li>4. El sistema verifica que no haya un evento con ese nombre y habilita el ingreso de los demás datos: la lista de atletas, se verifica que sean 3 de la misma disciplina</li><li>5. El sistema comprueba la validez de los datos, le asigna un código autonumérico, y redirecciona a la lista de eventos donde podemos ver el evento ingresado</li></ol>	

**Flujo/s Alternativo/s:**

3a. El usuario cancela el ingreso.

3a - 1. La evento no es guardado.

4a. No se ingresó nombre, fecha inicio o fin, o el rango de fechas es incorrecto

4a - 1. El sistema no solicita los datos adicionales y avisa que hay un error, marcando los campos erróneos con una descripción del error producido, quedando a la espera de su corrección.

4b. Ya existe una evento con ese nombre.

4b - 1. El sistema muestra un error indicando que el nombre esta repetido

5a. Los atletas seleccionados no son de la disciplina del evento

5b - 1. Si hay 3 o más atletas correctos el sistema crea el evento excluyendo a los atletas de otras disciplinas

5b - 2. Si no hay 3 atletas correctos el sistema muestra un error y no crea el evento

6a. El usuario cancela el ingreso.

6a- 1. Evento no es guardado.

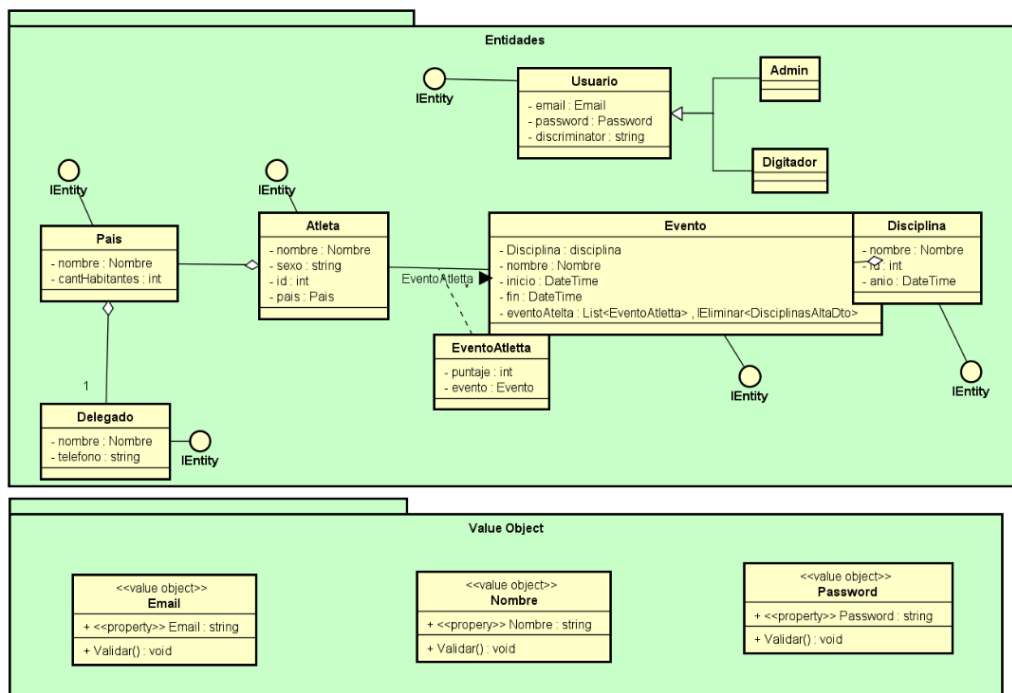
**Flujo/s Excepcionales/s:**

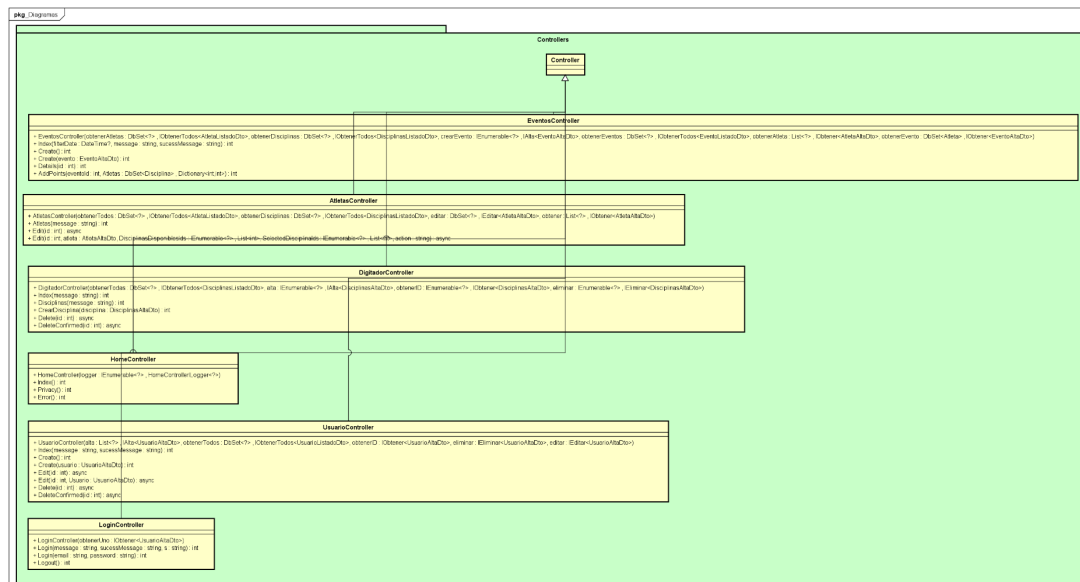
La comunicación con el servidor se corta durante el registro. El evento es descartado en su totalidad.

## Ingreso de puntaje de atleta

<b>Identificador:</b> CU-02	<b>Nombre:</b> Puntaje de atleta
<b>Descripción:</b> Permite ingresar el puntaje del atleta en el sistema	
<b>Actor/es:</b> Usuario registrado con rol administrador o digitador	
<b>Precondiciones:</b> Existe un evento cargado correctamente en el sistema	
<b>Pos condiciones:</b> Los datos del puntaje del atleta son ingresados al sistema.	
<b>Flujo Normal:</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa a lista de eventos, y selecciona agregar puntaje.</li> <li>2. El sistema muestra una lista de los atletas del evento y un input numerico para ingresar el puntaje.</li> <li>3. El usuario introduce los datos solicitados y solicitaguardar puntos.</li> </ol>	
<b>Flujo/s Excepcionales/s:</b>  La comunicación con el servidor se corta durante el registro. La película es descartada en su totalidad.	

# Diagrama de Clases UML







## Código fuente

\*\*\*\*\*

Archivo: ComiteContext.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\EF\ComiteContext.cs

\*\*\*\*\*

```
using ComiteAccesoDatos.EF.Config;
using ComiteLogicaNegocio.Entidades;
using Microsoft.EntityFrameworkCore;
```

```
namespace ComiteAccesoADatos.EF
```

```
{
```

```
    public class ComiteContext : DbContext
```

```
    {
```

```
        public DbSet<Usuario> usuarios { get; set; }
```

```
        public DbSet<Admin> admins { get; set; }
```

```
        public DbSet<Digitador> digitadores { get; set; }
```

```
        public DbSet<Disciplina> disciplinas { get; set; }
```

```
        public DbSet<Pais> paises { get; set; }
```

```
        public DbSet<Atleta> atletas { get; set; }
```

```
        public DbSet<Evento> eventos { get; set; }
```

```
        public DbSet<EventoAtleta> eventosAtletas { get; set; }
```

```
        protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder)
```

```
        {
```

```
            base.OnConfiguring(optionsBuilder);
```

```
            optionsBuilder.UseSqlServer(@"
```

```
                Data Source = (localdb)\MSSQLLocalDB;
```

```
                Initial Catalog = comite;
```

```
        Integrated Security = True;"
    );
}

protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    base.OnModelCreating(modelBuilder);
    modelBuilder.ApplyConfiguration(new UsuarioConfiguration());
}

}
}

*****
Archivo: RepositorioAtleta.cs
Carpeta:
C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\EF\Reposi
torioAtleta.cs
*****
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesRepositorio;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteAccesoADatos.EF
{
    public class RepositorioAtleta : IRepositoryAtleta
    {
        private ComiteContext _context;
        public RepositorioAtleta(ComiteContext context)
        {
            _context = context;
        }

        public void Add(Atleta obj)
```

---

```
{
    if (obj == null)
    {
        throw new ArgumentNullException("No se recibio el atleta válido");
    }
    _context.Add(obj);
    _context.SaveChanges();
}

public IEnumerable<Atleta> GetAll()
{
    return _context.atletas
        .Include(a => a.Pais)
        .Include(a => a.Disciplinas)
        .OrderBy(a => a.Pais.NombrePais) // Asegúrate de que 'Nombre' es la
propiedad correcta en 'Pais'
        .ThenBy(a => a.Nombre) // Suponiendo que 'Apellido' es la propiedad
que contiene el apellido del atleta
        .ToList();
}

public Atleta GetById(int id)
{
    Atleta? a = null;
    a =
        _context.atletas.Include(a=> a.Pais).Include(a => a.Disciplinas)
        .FirstOrDefault(atleta => atleta.ID == id);
    if (a == null)
    {
        throw new Exception($"No se encontro la el atleta con id {id}");
    }
    return a;
}

public void Edit(Atleta obj)
{
    Atleta a = GetById(obj.ID);
    a.Nombre = obj.Nombre;
    a.Sexo = obj.Sexo;
    foreach(int i in obj.DisciplinasIds){
        Disciplina? d = null;
        d =
```

```
        _context.disciplinas
            .FirstOrDefault(disciplina => disciplina.ID == i);
        a.Disciplinas.Add(d);
        a.DisciplinasIds.Add(i);
    }
    _context.atletas.Update(a);
    _context.SaveChanges();
}
}
```

\*\*\*\*\*

Archivo: RepositorioDisciplina.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\EF\RepositorioDisciplina.cs

\*\*\*\*\*

```
using ComiteAccesoADatos.Excepciones;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.SqlServer.Query.Internal;
```

```
namespace ComiteAccesoADatos.EF
```

```
{
    public class RepositorioDisciplina : IRepositoryDisciplina
    {
```

```
        private ComiteContext _context;
```

```
        public RepositorioDisciplina(ComiteContext context)
```

```
        {
            _context = context;
        }
```

```
        public void Add(Disciplina obj)
```

```
        {
            if (obj == null)
            {
                throw new DisciplinaException("No se recibio una disciplina valida");
            }
        }
```

---

```
        if (DisciplinaExist(obj.Nombre)) {
            throw new DisciplinaException("La disciplina ya existe");
        }
        if (obj.Nombre.Length < 10)
        {
            throw new DisciplinaException("El nombre de disciplina debe tener 10
caracteres o mas");
        }
        if (obj.Nombre.Length > 50)
        {
            throw new DisciplinaException("El nombre de disciplina debe tener menos
de 50 caracteres");
        }
        if (obj.Year > 2024)
        {
            throw new DisciplinaException("El año no puede ser mayor al actual");
        }
        try
        {
            _context.Add(obj);
            _context.SaveChanges();
        }
        catch (Exception)
        {

            throw new DisciplinaException("No se pudo agregar la disciplina");
        }

    }

    public void Delete(Disciplina obj)
    {
        try
        {
            Disciplina d = GetById(obj.ID);
            _context.disciplinas.Remove(d);
            _context.SaveChanges();
        }
        catch (Exception)
        {
```

---

```
        throw new DisciplinaException("Error al eliminar disciplina");
    }

}

public IEnumerable<Disciplina> GetAll()
{
    return _context.disciplinas
        .Include(d => d.Atletas)
        .OrderBy(d => d.Nombre)
        .ToList();
}

public Disciplina GetById(int id)
{
    Disciplina? d = null;
    d =
        _context.disciplinas
            .FirstOrDefault(disciplina => disciplina.ID == id);
    if (d == null)
    {
        throw new DisciplinaException($"No se encontro la disciplina con id {id}");
    }
    return d;
}

public Disciplina GetByName(string nombre)
{
    Disciplina? d = null;
    d =
        _context.disciplinas
            .AsEnumerable()
            .FirstOrDefault(d => d.Nombre == nombre);
    if (d == null)
    {
        throw new Exception($"No se encontro la disciplina con nombre
{nombre}");
    }
    return d;
}

public bool DisciplinaExist(string nombre) {
    Disciplina? d = null;
```

```
        d =
            _context.disciplinas
                .AsEnumerable()
                .FirstOrDefault(d => d.Nombre == nombre);
        return d != null;
    }
}
```

\*\*\*\*\*

Archivo: RepositorioEvento.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\EF\RepositorioEvento.cs

\*\*\*\*\*

```
using ComiteAccesoADatos.Excepciones;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesRepositorio;
using Microsoft.EntityFrameworkCore;
```

```
namespace ComiteAccesoADatos.EF
```

```
{
    public class RepositorioEvento : IRepositoryEvento
    {
        private ComiteContext _context;

        public RepositorioEvento(ComiteContext context)
        {
            _context = context;
        }

        public void Add(Evento obj)
        {
            if (obj == null)
            {
                throw new ArgumentNullException("No se recibio evento válido");
            }
            if (EventoExist(obj.Nombre))
            {
                throw new EventoException("El evento ya existe");
            }
        }
    }
}
```

---

```
    }
    if (obj.Inicio > obj.Fin)
    {
        throw new EventoException("El inicio no puede ser antes que el fin");
    }
    try
    {
        _context.Add(obj);
        _context.SaveChanges();
    }
    catch (Exception)
    {
        throw new EventoException("No se pudo agregar el evento");
    }
}

public IEnumerable<Evento> GetAll()
{
    return _context.eventos.Include(a => a.Disciplina).ToList();
}

public Evento GetByld(int id)
{
    Evento? e = null;
    e =
        _context.eventos.Include(e=>e.atletas)
        .AsEnumerable()
        .FirstOrDefault(e => e.ID == id);
    if (e == null)
    {
        throw new Exception($"No se encontro el evento con id {id}");
    }
    return e;
}

public Evento GetByName(string nombre)
{
    Evento? e = null;
    e =
        _context.eventos
```



```
        .AsEnumerable()
        .FirstOrDefault(e => e.Nombre == nombre);
    if (e == null)
    {
        throw new Exception($"No se encontro la disciplina con nombre
{nombre}");
    }
    return e;
}

public bool EventoExist(string nombre)
{
    Evento? e = null;
    e =
        _context.eventos
        .AsEnumerable()
        .FirstOrDefault(e => e.Nombre == nombre);
    return e != null;
}

public void AgregarAtletaAlEvento(Atleta atleta)
{
    throw new NotImplementedException();
}
}
}
```

\*\*\*\*\*

Archivo: RepositorioPais.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\EF\RepositorioPais.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteAccesoADatos.EF
{
    public class RepositorioPais : IRepositoryPais
    {
        private ComiteContext _context;

        public RepositorioPais(ComiteContext context)
        {
            _context = context;
        }
        public void Add(Pais obj)
        {
            if (obj == null)
            {
                throw new ArgumentNullException("No se recibio pais válido");
            }
            _context.Add(obj);
            _context.SaveChanges();
        }

        public IEnumerable<Pais> GetAll()
        {
            return _context.países;
        }

        public Pais GetById(int id)
        {
            Pais? p = null;
            p =
                _context.países
                .FirstOrDefault(pais => pais.ID == id);
            if (p == null)
            {
                throw new Exception($"No se encontro la el pais con id {id}");
            }
            return p;
        }
    }
}
```

\*\*\*\*\*

Archivo: RepositorioUsuario.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\EF\RepositorioUsuario.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.Excepciones.Usuario;
using ComiteLogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore.SqlServer.Query.Internal;
```

```
namespace ComiteAccesoADatos.EF
```

```
{
    public class RepositorioUsuario : IRepositoryUsuario
    {
        private ComiteContext _context;

        public RepositorioUsuario(ComiteContext context)
        {
            _context = context;
        }

        public void Add(Usuario obj)
        {
            if (obj == null)
            {
                throw new UsuarioException("No se recibio el usuario válido");
            }
            if (UsuarioExiste(obj.Email.Value))
            {
                throw new UsuarioException("El usuario ya existe");
            }
            _context.Add(obj);
            _context.SaveChanges();
        }

        public IEnumerable<Usuario> GetAll()
        {
            return _context.usuarios;
        }

        public Usuario GetById(int id)
```

---

```
{
    Usuario? unUsuario = null;
    unUsuario =
        _context.usuarios
            .FirstOrDefault(usuario => usuario.ID == id);
    if (unUsuario == null)
    {
        throw new UsuarioException($"No se encontro la el usuario con id {id}");
    }
    return unUsuario;
}

public Usuario GetByEmail(string email)
{
    Usuario? unUsuario = null;
    unUsuario =
        _context.usuarios
            .AsEnumerable()
            .FirstOrDefault(usuario => usuario.Email.Value == email);
    if (unUsuario == null)
    {
        throw new UsuarioException($"No se encontro la el usuario con email
{email}");
    }
    return unUsuario;
}

public bool UsuarioExiste(string email)
{
    Usuario? unUsuario = null;
    unUsuario =
        _context.usuarios
            .AsEnumerable()
            .FirstOrDefault(usuario => usuario.Email.Value == email);
    return unUsuario != null;
}

public void Delete(Usuario obj)
{
    try
    {
        Usuario u = GetByEmail(obj.Email.Value);
```

---

```
        if (u == null)
        {
            throw new UsuarioException("No se recibio el usuario válido");
        }
        _context.usuarios.Remove(u);
        _context.SaveChanges();
    }
    catch (Exception)
    {
        throw new UsuarioException("Error al eliminar usuario");
    }
}

public void Edit(Usuario obj)
{
    Usuario u = GetById(obj.ID);
    if (u == null)
    {
        throw new UsuarioException("No se recibio el usuario válido");
    }
    if (UsuarioExiste(obj.Email.Value)) {
        throw new UsuarioException("El usuario ya existe");
    }
    try
    {
        u.Email = obj.Email;
        u.Password = obj.Password;
        _context.usuarios.Update(u);
        _context.SaveChanges();
    }
    catch (Exception)
    {
        throw new UsuarioException("Error al editar usuario");
    }
}
}
```

\*\*\*\*\*

Archivo: DisciplinaException.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Excepciones\DisciplinaException.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteAccesoADatos.Excepciones
```

```
{
    public class DisciplinaException : Exception
    {
        public DisciplinaException()
        {
        }

        public DisciplinaException(string? message) : base(message)
        {
        }

        public DisciplinaException(string? message, Exception? innerException) :
base(message, innerException)
        {
        }

        protected DisciplinaException(SerializationInfo info, StreamingContext context)
: base(info, context)
        {
        }
    }
}
```

\*\*\*\*\*

Archivo: EventoException.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Excepciones\EventoException.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteAccesoADatos.Excepciones
```

```
{
    public class EventoException : Exception
    {
        public EventoException()
        {
        }

        public EventoException(string? message) : base(message)
        {
        }

        public EventoException(string? message, Exception? innerException) :
base(message, innerException)
        {
        }

        protected EventoException(SerializationInfo info, StreamingContext context) :
base(info, context)
        {
        }
    }
}
```

\*\*\*\*\*

Archivo: UsuarioException.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Excepciones\UsuarioException.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.Text;
using System.Threading.Tasks;

namespace ComiteAccesoADatos.Excepciones.Usuario
{
    public class UsuarioException : Exception
    {
        public UsuarioException()
        {
        }

        public UsuarioException(string? message) : base(message)
        {
        }

        public UsuarioException(string? message, Exception? innerException) :
base(message, innerException)
        {
        }

        protected UsuarioException(SerializationInfo info, StreamingContext context) :
base(info, context)
        {
        }
    }
}
```

\*\*\*\*\*

Archivo: 20241015164546\_init.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Migrations\  
20241015164546\_init.cs

\*\*\*\*\*

```
using System;
using Microsoft.EntityFrameworkCore.Migrations;
```

```
#nullable disable
```



---

```
namespace ComiteAccesoADatos.Migrations
{
    /// <inheritdoc />
    public partial class init : Migration
    {
        /// <inheritdoc />
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "disciplinas",
                columns: table => new
                {
                    ID = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Year = table.Column<int>(type: "int", nullable: false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_disciplinas", x => x.ID);
                });

            migrationBuilder.CreateTable(
                name: "paises",
                columns: table => new
                {
                    ID = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    NombrePais = table.Column<string>(type: "nvarchar(max)", nullable:
false),
                    CantidadHabitantes = table.Column<int>(type: "int", nullable: false),
                    NombreContacto = table.Column<string>(type: "nvarchar(max)",
nullable: false),
                    TelefonoContacto = table.Column<string>(type: "nvarchar(max)",
nullable: false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_paises", x => x.ID);
                });
        }
    }
}
```

```
migrationBuilder.CreateTable(
    name: "usuarios",
    columns: table => new
    {
        ID = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Email = table.Column<string>(type: "nvarchar(450)", nullable: false),
        Password = table.Column<string>(type: "nvarchar(max)", nullable:
false),
        Discriminator = table.Column<string>(type: "nvarchar(13)", maxLength:
13, nullable: false),
        fecRegistro = table.Column<DateTime>(type: "datetime2", nullable:
false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_usuarios", x => x.ID);
    });
```

```
migrationBuilder.CreateTable(
    name: "eventos",
    columns: table => new
    {
        ID = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Disciplinald = table.Column<int>(type: "int", nullable: false),
        Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false),
        Inicio = table.Column<DateTime>(type: "datetime2", nullable: false),
        Fin = table.Column<DateTime>(type: "datetime2", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_eventos", x => x.ID);
        table.ForeignKey(
            name: "FK_eventos_disciplinas_Disciplinald",
            column: x => x.Disciplinald,
            principalTable: "disciplinas",
            principalColumn: "ID",
```

```
        onDelete: ReferentialAction.Cascade);
    });

migrationBuilder.CreateTable(
    name: "atletas",
    columns: table => new
    {
        ID = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false),
        Sexo = table.Column<string>(type: "nvarchar(max)", nullable: false),
        PaisId = table.Column<int>(type: "int", nullable: false),
        DisciplinasIds = table.Column<string>(type: "nvarchar(max)", nullable:
false, defaultValue: ""),
        EventoID = table.Column<int>(type: "int", nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_atletas", x => x.ID);
        table.ForeignKey(
            name: "FK_atletas_eventos_EventoID",
            column: x => x.EventoID,
            principalTable: "eventos",
            principalColumn: "ID");
        table.ForeignKey(
            name: "FK_atletas_paises_PaisId",
            column: x => x.PaisId,
            principalTable: "paises",
            principalColumn: "ID",
            onDelete: ReferentialAction.Cascade);
    });

migrationBuilder.CreateTable(
    name: "AtletaDisciplina",
    columns: table => new
    {
        AtletasID = table.Column<int>(type: "int", nullable: false),
        DisciplinasID = table.Column<int>(type: "int", nullable: false)
    },
    constraints: table =>
    {
```

```
        table.PrimaryKey("PK_AtletaDisciplina", x => new { x.AtletasID,
x.DisciplinasID });
        table.ForeignKey(
            name: "FK_AtletaDisciplina_atletas_AtletasID",
            column: x => x.AtletasID,
            principalTable: "atletas",
            principalColumn: "ID",
            onDelete: ReferentialAction.Cascade);
        table.ForeignKey(
            name: "FK_AtletaDisciplina_disciplinas_DisciplinasID",
            column: x => x.DisciplinasID,
            principalTable: "disciplinas",
            principalColumn: "ID",
            onDelete: ReferentialAction.Cascade);
    });
```

```
migrationBuilder.CreateTable(
    name: "eventosAtletas",
    columns: table => new
    {
        ID = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Eventold = table.Column<int>(type: "int", nullable: false),
        Atletald = table.Column<int>(type: "int", nullable: false),
        Puntaje = table.Column<decimal>(type: "decimal(18,2)", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_eventosAtletas", x => x.ID);
        table.ForeignKey(
            name: "FK_eventosAtletas_atletas_Atletald",
            column: x => x.Atletald,
            principalTable: "atletas",
            principalColumn: "ID",
            onDelete: ReferentialAction.Cascade);
        table.ForeignKey(
            name: "FK_eventosAtletas_eventos_Eventold",
            column: x => x.Eventold,
            principalTable: "eventos",
            principalColumn: "ID",
```

---

```
        onDelete: ReferentialAction.Cascade);
    });

    migrationBuilder.CreateIndex(
        name: "IX_atletas_EventoID",
        table: "atletas",
        column: "EventoID");

    migrationBuilder.CreateIndex(
        name: "IX_eventosAtletas_AtletaID",
        table: "eventosAtletas",
        column: "AtletaID");

    migrationBuilder.CreateIndex(
        name: "IX_eventosAtletas_EventoID",
        table: "eventosAtletas",
        column: "EventoID");

    migrationBuilder.CreateIndex(
        name: "IX_AtletaDisciplina_DisciplinasID",
        table: "AtletaDisciplina",
        column: "DisciplinasID");

    migrationBuilder.CreateIndex(
        name: "IX_atletas_PaisID",
        table: "atletas",
        column: "PaisID");

    migrationBuilder.CreateIndex(
        name: "IX_usuarios_Email",
        table: "usuarios",
        column: "Email",
        unique: true);
}

/// <inheritdoc />
protected override void Down(MigrationBuilder migrationBuilder)
{
    migrationBuilder.DropTable(
        name: "AtletaDisciplina");
}
```

```
migrationBuilder.DropTable(
    name: "eventosAtletas");

migrationBuilder.DropTable(
    name: "usuarios");

migrationBuilder.DropTable(
    name: "atletas");

migrationBuilder.DropTable(
    name: "eventos");

migrationBuilder.DropTable(
    name: "paises");

migrationBuilder.DropTable(
    name: "disciplinas");
}
}
}
```

\*\*\*\*\*

Archivo: 20241015164546\_init.Designer.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Migrations\  
20241015164546\_init.Designer.cs

\*\*\*\*\*

// <auto-generated />

```
using System;
using ComiteAccesoADatos.EF;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Migrations;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
```

#nullable disable

```
namespace ComiteAccesoADatos.Migrations
{
```

---

```
[DbContext(typeof(ComiteContext))]
[Migration("20241015164546_init")]
partial class init
{
    /// <inheritdoc />
    protected override void BuildTargetModel(ModelBuilder modelBuilder)
    {
#pragma warning disable 612, 618
        modelBuilder
            .HasAnnotation("ProductVersion", "8.0.8")
            .HasAnnotation("Relational:MaxIdentifierLength", 128);

        SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

        modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
        {
            b.Property<int>("ID")
                .ValueGeneratedOnAdd()
                .HasColumnType("int");

            SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));

            b.Property<string>("DisciplinasIds")
                .IsRequired()
                .HasColumnType("nvarchar(max)");

            b.Property<string>("Nombre")
                .IsRequired()
                .HasColumnType("nvarchar(max)");

            b.Property<int>("PaisId")
                .HasColumnType("int");

            b.Property<string>("Sexo")
                .IsRequired()
                .HasColumnType("nvarchar(max)");

            b.HasKey("ID");

            b.HasIndex("PaisId");
```

---

```
        b.ToTable("atletas");
    });

    modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Disciplina", b =>
    {
        b.Property<int>("ID")
            .ValueGeneratedOnAdd()
            .HasColumnType("int");

        SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));

        b.Property<int?>("AtletalD")
            .HasColumnType("int");

        b.Property<string>("Nombre")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.Property<int>("Year")
            .HasColumnType("int");

        b.HasKey("ID");

        b.HasIndex("AtletalD");

        b.ToTable("disciplinas");
    });

    modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Pais", b =>
    {
        b.Property<int>("ID")
            .ValueGeneratedOnAdd()
            .HasColumnType("int");

        SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));

        b.Property<int>("CantidadHabitantes")
            .HasColumnType("int");
```



---

```
b.Property<string>("NombreContacto")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

b.Property<string>("NombrePais")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

b.Property<string>("TelefonoContacto")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

b.HasKey("ID");

b.ToTable("paises");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Usuario", b =>
{
    b.Property<int>("ID")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));

b.Property<string>("Discriminator")
    .IsRequired()
    .HasMaxLength(13)
    .HasColumnType("nvarchar(13)");

b.Property<string>("Email")
    .IsRequired()
    .HasColumnType("nvarchar(450)");

b.Property<string>("Password")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

b.Property<DateTime>("fecRegistro")
```

---

```
        .HasColumnType("datetime2");

        b.HasKey("ID");

        b.HasIndex("Email")
            .IsUnique();

        b.ToTable("usuarios");

        b.HasDiscriminator().HasValue("Usuario");

        b.UseTphMappingStrategy();
    });

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Admin", b =>
{
    b.HasBaseType("ComiteLogicaNegocio.Entidades.Usuario");

    b.HasDiscriminator().HasValue("Admin");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Digitador", b =>
{
    b.HasBaseType("ComiteLogicaNegocio.Entidades.Usuario");

    b.HasDiscriminator().HasValue("Digitador");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
{
    b.HasOne("ComiteLogicaNegocio.Entidades.Pais", "Pais")
        .WithMany()
        .HasForeignKey("PaisId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Pais");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Disciplina", b =>
{
```

```
        b.HasOne("ComiteLogicaNegocio.Entidades.Atleta", null)
            .WithMany("Disciplinas")
            .HasForeignKey("AtletaID");
    });

    modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
    {
        b.Navigation("Disciplinas");
    });
#pragma warning restore 612, 618
    }
}
}

*****

Archivo: 20241015164708_SeedUsuarios.cs
Carpeta:
C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Migrations\
20241015164708_SeedUsuarios.cs
*****

using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace ComiteAccesoADatos.Migrations
{
    /// <inheritdoc />
    public partial class SeedUsuarios : Migration
    {
        /// <inheritdoc />
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.InsertData(
                "usuarios",
                new[] { "ID", "Email", "Password", "Discriminator", "fecRegistro" },
                new object[] { 1, "admin@admin.com", "Admin.123", "Admin",
DateTime.Now }
            );
            migrationBuilder.InsertData(
                "usuarios",
                new[] { "ID", "Email", "Password", "Discriminator", "fecRegistro" },
```

```
        new object[] { 2, "digitador@digitador.com", "Digitador.123", "Digitador",  
        DateTime.Now }  
    );  
  
}
```

```
    /// <inheritdoc />  
    protected override void Down(MigrationBuilder migrationBuilder)  
    {  
  
    }  
}
```

\*\*\*\*\*

Archivo: 20241015164708\_SeedUsuarios.Designer.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Migrations\  
20241015164708\_SeedUsuarios.Designer.cs

\*\*\*\*\*

```
// <auto-generated />  
using System;  
using ComiteAccesoADatos.EF;  
using Microsoft.EntityFrameworkCore;  
using Microsoft.EntityFrameworkCore.Infrastructure;  
using Microsoft.EntityFrameworkCore.Metadata;  
using Microsoft.EntityFrameworkCore.Migrations;  
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
```

#nullable disable

namespace ComiteAccesoADatos.Migrations

```
{  
    [DbContext(typeof(ComiteContext))]  
    [Migration("20241015164708_SeedUsuarios")]  
    partial class SeedUsuarios  
    {  
        /// <inheritdoc />  
        protected override void BuildTargetModel(ModelBuilder modelBuilder)  
        {
```

---

```
#pragma warning disable 612, 618
modelBuilder
    .HasAnnotation("ProductVersion", "8.0.8")
    .HasAnnotation("Relational:MaxIdentifierLength", 128);

SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
{
    b.Property<int>("ID")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

    b.Property<string>("DisciplinasIds")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b.Property<string>("Nombre")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b.Property<int>("PaisId")
        .HasColumnType("int");

    b.Property<string>("Sexo")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b.HasKey("ID");

    b.HasIndex("PaisId");

    b.ToTable("atletas");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Disciplina", b =>
{
    b.Property<int>("ID")
```

```
.ValueGeneratedOnAdd()  
.HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));
```

```
b.Property<int?>("AtletaID")  
.HasColumnType("int");  
  
b.Property<string>("Nombre")  
.IsRequired()  
.HasColumnType("nvarchar(max)");
```

```
b.Property<int>("Year")  
.HasColumnType("int");
```

```
b.HasKey("ID");
```

```
b.HasIndex("AtletaID");
```

```
b.ToTable("disciplinas");  
});
```

```
modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Pais", b =>  
{  
    b.Property<int>("ID")  
        .ValueGeneratedOnAdd()  
        .HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));
```

```
b.Property<int>("CantidadHabitantes")  
.HasColumnType("int");
```

```
b.Property<string>("NombreContacto")  
.IsRequired()  
.HasColumnType("nvarchar(max)");
```

```
b.Property<string>("NombrePais")  
.IsRequired()
```

---

```
        .HasColumnType("nvarchar(max)");

        b.Property<string>("TelefonoContacto")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.HasKey("ID");

        b.ToTable("paises");
    });

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Usuario", b =>
{
    b.Property<int>("ID")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));

    b.Property<string>("Discriminator")
        .IsRequired()
        .HasMaxLength(13)
        .HasColumnType("nvarchar(13)");

    b.Property<string>("Email")
        .IsRequired()
        .HasColumnType("nvarchar(450)");

    b.Property<string>("Password")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b.Property<DateTime>("fecRegistro")
        .HasColumnType("datetime2");

    b.HasKey("ID");

    b.HasIndex("Email")
        .IsUnique();
```

---

```
b.ToTable("usuarios");

b.HasDiscriminator().HasValue("Usuario");

b.UseTphMappingStrategy();
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Admin", b =>
{
    b.HasBaseType("ComiteLogicaNegocio.Entidades.Usuario");

    b.HasDiscriminator().HasValue("Admin");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Digitador", b =>
{
    b.HasBaseType("ComiteLogicaNegocio.Entidades.Usuario");

    b.HasDiscriminator().HasValue("Digitador");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
{
    b.HasOne("ComiteLogicaNegocio.Entidades.Pais", "Pais")
        .WithMany()
        .HasForeignKey("PaisId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Pais");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Disciplina", b =>
{
    b.HasOne("ComiteLogicaNegocio.Entidades.Atleta", null)
        .WithMany("Disciplinas")
        .HasForeignKey("AtletaID");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
{
```



```
        b.Navigation("Disciplinas");
    });
#pragma warning restore 612, 618
    }
}
}
```

\*\*\*\*\*

Archivo: 20241015164725\_SeedDisciplinas.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Migrations\  
20241015164725\_SeedDisciplinas.cs

\*\*\*\*\*

```
using Microsoft.EntityFrameworkCore.Migrations;
```

```
#nullable disable
```

```
namespace ComiteAccesoADatos.Migrations
```

```
{
    /// <inheritdoc />
    public partial class SeedDisciplinas : Migration
    {
        /// <inheritdoc />
```

```
        protected override void Up(MigrationBuilder migrationBuilder)
        {
```

```
            migrationBuilder.InsertData(
                "disciplinas",
                new[] { "ID", "Nombre", "Year" },
                new object[] { 1, "Balonmano", 94 }
            );
```

```
            migrationBuilder.InsertData(
                "disciplinas",
                new[] { "ID", "Nombre", "Year" },
                new object[] { 2, "Surf", 94 }
            );
```

```
            migrationBuilder.InsertData(
                "disciplinas",
                new[] { "ID", "Nombre", "Year" },
                new object[] { 3, "Taekwondo", 94 }
            );
```

```
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 4, "Tenis", 94 }
);
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 5, "Tenis de mesa", 94 }
);
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 6, "Tiro", 94 }
);
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 7, "Tiro con arco", 94 }
);
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 8, "Triatlón", 94 }
);
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 9, "Vela", 94 }
);

migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 10, "Atletismo", 94 }
);
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 11, "Ciclismo", 94 }
);
```

```
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 12, "Natación", 94 }
);
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 13, "Gimnasia", 94 }
);
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 14, "Boxeo", 94 }
);
migrationBuilder.InsertData(
    "disciplinas",
    new[] { "ID", "Nombre", "Year" },
    new object[] { 15, "Esgrima", 94 }
);

}

/// <inheritdoc />
protected override void Down(MigrationBuilder migrationBuilder)
{
}
}
```

\*\*\*\*\*

Archivo: 20241015164725\_SeedDisciplinas.Designer.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Migrations\  
20241015164725\_SeedDisciplinas.Designer.cs

\*\*\*\*\*

// <auto-generated />

```
using System;
using ComiteAccesoADatos.EF;
using Microsoft.EntityFrameworkCore;
```

---

```
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Migrations;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

#nullable disable

namespace ComiteAccesoADatos.Migrations
{
    [DbContext(typeof(ComiteContext))]
    [Migration("20241015164725_SeedDisciplinas")]
    partial class SeedDisciplinas
    {
        /// <inheritdoc />
        protected override void BuildTargetModel(ModelBuilder modelBuilder)
        {
#pragma warning disable 612, 618
            modelBuilder
                .HasAnnotation("ProductVersion", "8.0.8")
                .HasAnnotation("Relational:MaxIdentifierLength", 128);

            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

            modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
            {
                b.Property<int>("ID")
                    .ValueGeneratedOnAdd()
                    .HasColumnType("int");

                b.Property<string>("DisciplinasIds")
                    .IsRequired()
                    .HasColumnType("nvarchar(max)");

                b.Property<string>("Nombre")
                    .IsRequired()
                    .HasColumnType("nvarchar(max)");

                b.Property<int>("PaisId")
```

---

```
        .HasColumnType("int");

        b.Property<string>("Sexo")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.HasKey("ID");

        b.HasIndex("PaisId");

        b.ToTable("atletas");
    });

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Disciplina", b =>
{
    b.Property<int>("ID")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));

    b.Property<int?>("AtletaID")
        .HasColumnType("int");

    b.Property<string>("Nombre")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b.Property<int>("Year")
        .HasColumnType("int");

    b.HasKey("ID");

    b.HasIndex("AtletaID");

    b.ToTable("disciplinas");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Pais", b =>
{
```

```
b.Property<int>("ID")
    .ValueGeneratedOnAdd()
    .HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));
```

```
b.Property<int>("CantidadHabitantes")
    .HasColumnType("int");
```

```
b.Property<string>("NombreContacto")
    .IsRequired()
    .HasColumnType("nvarchar(max)");
```

```
b.Property<string>("NombrePais")
    .IsRequired()
    .HasColumnType("nvarchar(max)");
```

```
b.Property<string>("TelefonoContacto")
    .IsRequired()
    .HasColumnType("nvarchar(max)");
```

```
b.HasKey("ID");
```

```
b.ToTable("paises");
});
```

```
modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Usuario", b =>
{
    b.Property<int>("ID")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));
```

```
b.Property<string>("Discriminator")
    .IsRequired()
    .HasMaxLength(13)
    .HasColumnType("nvarchar(13)");
```

---

```
b.Property<string>("Email")
    .IsRequired()
    .HasColumnType("nvarchar(450)");

b.Property<string>("Password")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

b.Property<DateTime>("fecRegistro")
    .HasColumnType("datetime2");

b.HasKey("ID");

b.HasIndex("Email")
    .IsUnique();

b.ToTable("usuarios");

b.HasDiscriminator().HasValue("Usuario");

b.UseTphMappingStrategy();
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Admin", b =>
{
    b.HasBaseType("ComiteLogicaNegocio.Entidades.Usuario");

    b.HasDiscriminator().HasValue("Admin");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Digitador", b =>
{
    b.HasBaseType("ComiteLogicaNegocio.Entidades.Usuario");

    b.HasDiscriminator().HasValue("Digitador");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
{
    b.HasOne("ComiteLogicaNegocio.Entidades.Pais", "Pais")
        .WithMany()
```

```
.HasForeignKey("PaisId")
.OnDelete(DeleteBehavior.Cascade)
.IsRequired();

b.Navigation("Pais");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Disciplina", b =>
{
    b.HasOne("ComiteLogicaNegocio.Entidades.Atleta", null)
        .WithMany("Disciplinas")
        .HasForeignKey("AtletaID");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
{
    b.Navigation("Disciplinas");
});
#pragma warning restore 612, 618
}
}
}

*****

Archivo: 20241015164742_SeedPaisesAtletas.cs
Carpeta:
C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Migrations\
20241015164742_SeedPaisesAtletas.cs
*****

using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace ComiteAccesoADatos.Migrations
{
    /// <inheritdoc />
    public partial class SeedPaisesAtletas : Migration
    {
        /// <inheritdoc />

        protected override void Up(MigrationBuilder migrationBuilder)
```



```
{
    migrationBuilder.InsertData(
        "países",
        new[] { "ID", "NombrePaís", "NombreContacto", "CantidadHabitantes",
"TelefonoContacto" },
        new object[,]
        {
            { 1, "Uruguay", "Milton Winan", 3000000, "+5465497898321" },
            { 2, "Argentina", "Lucia Gomez", 45000000, "+5491123456789" },
            { 3, "Brazil", "Carlos Silva", 211000000, "+5511987654321" },
            { 4, "Chile", "Sofia Torres", 19000000, "+56612345678" },
            { 5, "Colombia", "Juan Martinez", 50000000, "+573012345678" },
            { 6, "Mexico", "Ana Ruiz", 126000000, "+5215551234567" },
            { 7, "Peru", "Pedro Lopez", 32000000, "+519876543210" },
            { 8, "Paraguay", "Marta Fernandez", 7000000, "+595981234567" },
            { 9, "Venezuela", "Luis Rodriguez", 28000000, "+582121234567" },
            { 10, "Ecuador", "Julia Castillo", 17000000, "+593987654321" }
        }
    );
    migrationBuilder.InsertData(
        "atletas",
        new[] { "ID", "Nombre", "Sexo", "PaísId", "DisciplinasIds" },
        new object[,]
        {
            { 1, "Milton Winan", "M", 1, "[1,2]" },
            { 2, "Sofia Rodriguez", "F", 1, "[1,3]" },
            { 3, "Lucia Gomez", "F", 2, "[1,4,5]" },
            { 4, "Carlos Silva", "M", 2, "[1,6]" },
            { 5, "Ana Ruiz", "F", 3, "[1,7,8]" },
            { 6, "Pedro Martinez", "M", 3, "[1,9]" },
            { 7, "Sofia Torres", "F", 4, "[1,10]" },
            { 8, "Juan Herrera", "M", 4, "[1]" },
            { 9, "Maria Fernandez", "F", 5, "[2]" },
            { 10, "Luis Ramirez", "M", 5, "[3]" },
            { 11, "Clara Sanchez", "F", 6, "[4]" },
            { 12, "Miguel Alvarez", "M", 6, "[5]" },
            { 13, "Julia Castillo", "F", 7, "[6]" },
            { 14, "Roberto Perez", "M", 7, "[7]" },
            { 15, "Natalia Gomez", "F", 8, "[8]" },
            { 16, "Diego Lopez", "M", 8, "[9]" },
            { 17, "Isabella Torres", "F", 9, "[10]" },
        }
    );
}
```

```
        { 18, "Andres Jimenez", "M", 9, "[1]" },
        { 19, "Valentina Rojas", "F", 10, "[2]" },
        { 20, "Fernando Vasquez", "M", 10, "[3]" },
        { 21, "Carlos López", "M", 1, "[1,3]" }
    }
);

}

/// <inheritdoc />
protected override void Down(MigrationBuilder migrationBuilder)
{
}
}
}

*****
Archivo: 20241015164742_SeedPaisesAtletas.Designer.cs
Carpeta:
C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\Migrations\
20241015164742_SeedPaisesAtletas.Designer.cs
*****

// <auto-generated />
using System;
using ComiteAccesoADatos.EF;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Migrations;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

#nullable disable

namespace ComiteAccesoADatos.Migrations
{
    [DbContext(typeof(ComiteContext))]
    [Migration("20241015164742_SeedPaisesAtletas")]
    partial class SeedPaisesAtletas
    {
    }
}
```

---

```
/// <inheritdoc />
protected override void BuildTargetModel(ModelBuilder modelBuilder)
{
#pragma warning disable 612, 618
    modelBuilder
        .HasAnnotation("ProductVersion", "8.0.8")
        .HasAnnotation("Relational:MaxIdentifierLength", 128);

    SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

    modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
    {
        b.Property<int>("ID")
            .ValueGeneratedOnAdd()
            .HasColumnType("int");

        SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));

        b.Property<string>("DisciplinasIds")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.Property<string>("Nombre")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.Property<int>("PaisId")
            .HasColumnType("int");

        b.Property<string>("Sexo")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.HasKey("ID");

        b.HasIndex("PaisId");

        b.ToTable("atletas");
    });
```

---

```
modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Disciplina", b =>
{
    b.Property<int>("ID")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));
```

```
    b.Property<int?>("AtletaID")
        .HasColumnType("int");

    b.Property<string>("Nombre")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b.Property<int>("Year")
        .HasColumnType("int");

    b.HasKey("ID");

    b.HasIndex("AtletaID");

    b.ToTable("disciplinas");
});
```

```
modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Pais", b =>
{
    b.Property<int>("ID")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));
```

```
    b.Property<int>("CantidadHabitantes")
        .HasColumnType("int");

    b.Property<string>("NombreContacto")
        .IsRequired()
        .HasColumnType("nvarchar(max)");
```

---

```
b.Property<string>("NombrePais")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

b.Property<string>("TelefonoContacto")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

b.HasKey("ID");

b.ToTable("paises");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Usuario", b =>
{
    b.Property<int>("ID")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ID"));

b.Property<string>("Discriminator")
    .IsRequired()
    .HasMaxLength(13)
    .HasColumnType("nvarchar(13)");

b.Property<string>("Email")
    .IsRequired()
    .HasColumnType("nvarchar(450)");

b.Property<string>("Password")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

b.Property<DateTime>("fecRegistro")
    .HasColumnType("datetime2");

b.HasKey("ID");
```

---

```
b.HasIndex("Email")
    .IsUnique();

b.ToTable("usuarios");

b.HasDiscriminator().HasValue("Usuario");

b.UseTphMappingStrategy();
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Admin", b =>
{
    b.HasBaseType("ComiteLogicaNegocio.Entidades.Usuario");

    b.HasDiscriminator().HasValue("Admin");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Digitador", b =>
{
    b.HasBaseType("ComiteLogicaNegocio.Entidades.Usuario");

    b.HasDiscriminator().HasValue("Digitador");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
{
    b.HasOne("ComiteLogicaNegocio.Entidades.Pais", "Pais")
        .WithMany()
        .HasForeignKey("PaisId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Pais");
});

modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Disciplina", b =>
{
    b.HasOne("ComiteLogicaNegocio.Entidades.Atleta", null)
        .WithMany("Disciplinas")
        .HasForeignKey("AtletaID");
});
```

```
        modelBuilder.Entity("ComiteLogicaNegocio.Entidades.Atleta", b =>
        {
            b.Navigation("Disciplinas");
        });
#pragma warning restore 612, 618
    }
}
```

\*\*\*\*\*

Archivo: AdminController.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteApp\Controllers\AdminController.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.InterfacesCasoUso;
using Microsoft.AspNetCore.Mvc;
```

```
namespace Libreria.WebMvc.Controllers
```

```
{
    public class AdminController : Controller
    {
```

```
        IAlta<UsuarioAltaDto> _alta;
        IObtenerTodos<UsuarioListadoDto> _obtenerTodos;
```

```
        public AdminController(
            IAlta<UsuarioAltaDto> alta,
            IObtenerTodos<UsuarioListadoDto> obtenerTodos
        )
        {
            _alta = alta;
            _obtenerTodos = obtenerTodos;
        }
```

```
[HttpGet]
```

```
public IActionResult Index()
{
    try {
        var todos = _obtenerTodos.Ejecutar();
        return View(todos);
    } catch (Exception e) {
        ViewBag.Message = e.Message;
        return View();
    }
}
```

\*\*\*\*\*

Archivo: AtletasController.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteApp\Controllers\AtletasController.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Atletas;
using ComiteCompartido.Dtos.Disciplinas;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesCasoUso;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;
```

```
namespace ComiteApp.Controllers
```

```
{
    public class AtletasController : Controller
    {
        IObtenerTodos<AtletaListadoDto> _obtenerTodos;
        IObtenerTodos<DisciplinasListadoDto> _obtenerTodasDisciplinas;
        IEditar<AtletaAltaDto> _editar;
        IObtener<AtletaAltaDto> _obtenerID;
        public AtletasController(
            IObtenerTodos<AtletaListadoDto> obtenerTodos,
            IObtenerTodos<DisciplinasListadoDto> obtenerDisciplinas,
            IEditar<AtletaAltaDto> editar,
            IObtener<AtletaAltaDto> obtener
```



---

```
    ) {
        _obtenerTodos = obtenerTodos;
        _obtenerTodasDisciplinas = obtenerDisciplinas;
        _editar = editar;
        _obtenerID = obtener;
    }

[HttpGet]
public IActionResult Atletas(string message)
{
    ViewBag.Message = message;
    return View(_obtenerTodos.Ejecutar());
}

// GET: Atletas/Edit/5
public async Task<IActionResult> Edit(int id)
{
    if (id == null)
    {
        return NotFound();
    }

    var Atleta = _obtenerID.Ejecutar(id);
    if (Atleta == null)
    {
        return NotFound();
    }

    List<DisciplinasListadoDto> disciplinas =
_obtenerTodasDisciplinas.Ejecutar().ToList();
    List<DisciplinasListadoDto> disciplinasAtleta = new
List<DisciplinasListadoDto>();
    foreach (int i in Atleta.DisciplinasIds) {
        DisciplinasListadoDto? disciplina = disciplinas.FirstOrDefault(d => d.Id ==
i);
        disciplinasAtleta.Add(disciplina);
    }

    ViewBag.Disciplinas = disciplinas;
```

---

```
        ViewBag.DisciplinasAtleta = discilinasAtleta;
        return View(Atleta);
    }

    // POST: Altetas/Edit/5

    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Edit(int id, AtletaAltaDto atleta, List<int>
DisciplinasDisponiblesIds, List<int> SelectedDisciplinalIds, string action)
    {
        if (id != atleta.id)
        {
            return NotFound();
        }
        try
        {
            if (action == "add")
            {
                // Get selected disciplines from available list
                var selectedIds = DisciplinasDisponiblesIds.Where(id =>
!SelectedDisciplinalIds.Contains(id)).ToList();
                atleta.DisciplinasIds.AddRange(selectedIds);
            }
            else if (action == "remove")
            {
                // Remove selected disciplines
                var idsToRemove = SelectedDisciplinalIds.Where(id =>
DisciplinasDisponiblesIds.Contains(id)).ToList();
                foreach (var i in idsToRemove)
                {
                    atleta.DisciplinasIds.Remove(i);
                }
            }
            _editar.Ejecutar(atleta);
        }
        catch (DbUpdateConcurrencyException)
        {
            return NotFound();
        }
        return RedirectToAction("Atletas");
    }
```

```
}  
}  
}
```

\*\*\*\*\*

Archivo: DigitadorController.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteApp\Controllers\Digitador  
Controller.cs

\*\*\*\*\*

```
using ComiteLogicaAplicacion.CasoUso.Usuarios;  
using ComiteLogicaNegocio.Entidades;  
using ComiteLogicaNegocio.Excepciones;  
using ComiteLogicaNegocio.Excepciones.Usuario;  
using ComiteLogicaNegocio.InterfacesCasoUso;  
using ComiteApp.Models;  
using Microsoft.AspNetCore.Mvc;  
using System.Diagnostics;  
using ComiteCompartido.Dtos.Usuarios;  
using Microsoft.EntityFrameworkCore;  
using ComiteCompartido.Dtos.Disciplinas;
```

```
namespace ComiteApp.Controllers
```

```
{  
    public class DigitadorController : Controller  
    {  
        IObtenerTodos<DisciplinasListadoDto> _obtenerTodas;  
        IAlta<DisciplinasAltaDto> _alta;  
        IObtener<DisciplinasAltaDto> _obtenerID;  
        IEliminar<DisciplinasAltaDto> _eliminar;  
        public DigitadorController(  
            IObtenerTodos<DisciplinasListadoDto> obtenerTodas,  
            IAlta<DisciplinasAltaDto> alta,  
            IObtener<DisciplinasAltaDto> obtenerID,  
            IEliminar<DisciplinasAltaDto> eliminar)  
        {  
            _obtenerTodas = obtenerTodas;  
            _alta = alta;  
            _obtenerID = obtenerID;  
            _eliminar = eliminar;  
        }  
    }  
}
```

```
}

[HttpGet]
public IActionResult Index(string message)
{
    ViewBag.Message = message;
    try
    {
        return View(_obtenerTodas.Ejecutar());
    }
    catch (Exception e)
    {
        ViewBag.Message = e.Message;
        return View();
    }
}

[HttpGet]
public IActionResult Disciplinas(string message)
{
    ViewBag.Message = message;
    try
    {
        return View(_obtenerTodas.Ejecutar());
    }
    catch (Exception e)
    {
        ViewBag.Message = e.Message;
        return View();
    }
}

[HttpPost]
public IActionResult CrearDisciplina(DisciplinasAltaDto disciplina)
{
    try
    {
        _alta.Ejecutar(disciplina);
        return RedirectToAction("Disciplinas");
    }
    catch (Exception e)
```

---

```
{
    ViewBag.Message = e.Message;
    return RedirectToAction("Disciplinas", new { message = e.Message });
}

}
// GET: Digitador/Delete/5
public async Task<IActionResult> Delete(int id)
{
    try
    {
        if (id == null)
        {
            return NotFound();
        }

        var Usuario = _obtenerID.Ejecutar(id);
        if (Usuario == null)
        {
            return NotFound();
        }

        return View(Usuario);
    }
    catch (Exception e)
    {
        ViewBag.Message = e.Message;
        return View();
    }
}

// POST: Usuarios/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    try
    {
        var Usuario = _obtenerID.Ejecutar(id);
        if (Usuario != null)
```

```
        {
            _eliminar.Ejecutar(Usuario);
        }
        return RedirectToAction("Disciplinas");
    }
    catch (Exception e)
    {
        ViewBag.Message = e.Message;
        return View();
    }
}
}
```

\*\*\*\*\*

Archivo: EventosController.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteApp\Controllers\EventosController.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Atletas;
using ComiteCompartido.Dtos.Disciplinas;
using ComiteCompartido.Dtos.Eventos;
using ComiteLogicaAplicacion.CasoUso.CasoUsoAtleta;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesCasoUso;
using Microsoft.AspNetCore.Mvc;
```

namespace ComiteApp.Controllers

```
{
    public class EventosController : Controller
    {
        IObtenerTodos<AtletaListadoDto> _obtenerAtletas;
        IObtenerTodos<DisciplinasListadoDto> _obtenerDisciplinas;
        IAlta<EventoAltaDto> _crearEvento;
        IObtenerTodos<EventoListadoDto> _obtenerEventos;
        IObtener<AtletaAltaDto> _obtenerAtleta;
        IObtener<EventoAltaDto> _obtenerEvento;
        public EventosController(
```

---

```
        IObtenerTodos<AtletaListadoDto> obtenerAtletas,
        IObtenerTodos<DisciplinasListadoDto> obtenerDisciplinas,
        IAlta<EventoAltaDto> crearEvento,
        IObtenerTodos<EventoListadoDto> obtenerEventos,
        IObtener<AtletaAltaDto> obtenerAtleta,
        IObtener<EventoAltaDto> obtenerEvento
    )
    {
        _obtenerAtletas = obtenerAtletas;
        _obtenerDisciplinas = obtenerDisciplinas;
        _crearEvento = crearEvento;
        _obtenerEventos = obtenerEventos;
        _obtenerAtleta = obtenerAtleta;
        _obtenerEvento = obtenerEvento;
    }

    public IActionResult Index(DateTime? filterDate, string message, string
sucessMessage)
    {
        ViewBag.Message = message;
        ViewBag.sucessMessage = sucessMessage;

        // Get all events
        var eventos = _obtenerEventos.Ejecutar();

        // Filter by date if filterDate is provided
        if (filterDate.HasValue)
        {
            eventos = eventos.Where(e => e.Inicio.Date ==
filterDate.Value.Date).ToList();
        }

        return View(eventos);
    }

    [HttpGet]
    public IActionResult Create()
    {
        try
        {
```

---

```
        List<DisciplinasListadoDto> disciplinas =
        _obtenerDisciplinas.Ejecutar().ToList();
        ViewBag.Disciplinas = disciplinas;
        List<AtletaListadoDto> atletas = _obtenerAtletas.Ejecutar().ToList();
        ViewBag.Atletas = atletas;

        return View();
    }
    catch (Exception e)
    {
        ViewBag.Message = e.Message;
        return View();
    }
}

[HttpPost]
public IActionResult Create(EventoAltaDto evento)
{
    try
    {
        _crearEvento.Ejecutar(evento);
        return RedirectToAction("Index", new { successMessage = "Evento creado
con exito" });
    }
    catch (Exception e)
    {
        List<DisciplinasListadoDto> disciplinas =
        _obtenerDisciplinas.Ejecutar().ToList();
        ViewBag.Disciplinas = disciplinas;
        List<AtletaListadoDto> atletas = _obtenerAtletas.Ejecutar().ToList();
        ViewBag.Atletas = atletas;
        ViewBag.Message = e.Message;
        return View();
    }
}

[HttpGet]
public IActionResult Details(int id)
```



---

```
{

    try
    {
        List<AtletaAltaDto> atletas = new List<AtletaAltaDto>();
        EventoAltaDto evento = _obtenerEvento.Ejecutar(id);
        foreach (int i in evento.AtletasIds)
        {
            atletas.Add(_obtenerAtleta.Ejecutar(i));
        }
        List<DisciplinasListadoDto> disciplinas =
        _obtenerDisciplinas.Ejecutar().ToList();
        string d = disciplinas.FirstOrDefault(d => d.Id ==
evento.Disciplinald)?.Nombre;
        ViewBag.Disciplina = d;
        ViewBag.Atletas = atletas;
        return View(evento);
    }
    catch (Exception e)
    {
        ViewBag.Message = e.Message;
        return View();
    }
}

[HttpPost]
public IActionResult AddPoints(int eventold, Dictionary<int, int> Atletas)
{
    // Logic to save points for athletes
    // For example:
    foreach (var atleta in Atletas)
    {
        // Save points logic here
        // Example: _repository.SavePoints(eventold, atleta.Key, atleta.Value);
    }

    return RedirectToAction("Index", new { sucessMessage = "Puntos guardados
exitosamente." });
}
}
```

\*\*\*\*\*

Archivo: HomeController.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteApp\Controllers\HomeController.cs

\*\*\*\*\*

using ComiteApp.Models;

using Microsoft.AspNetCore.Mvc;

using System.Diagnostics;

namespace ComiteApp.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

return View();

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None,  
NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??  
HttpContext.TraceIdentifier });

}

}

}

\*\*\*\*\*

Archivo: LoginController.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteApp\Controllers>LoginController.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.InterfacesCasoUso;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
```

```
namespace ComiteApp.Controllers
```

```
{
```

```
    public class LoginController : Controller
```

```
    {
```

```
        IObtener<UsuarioAltaDto> _obtenerUno;
```

```
        public LoginController(
```

```
            IObtener<UsuarioAltaDto> obtenerUno
```

```
        )
```

```
        {
```

```
            _obtenerUno = obtenerUno;
```

```
        }
```

```
        [HttpGet]
```

```
        public IActionResult Login(string message, string successMessage, string s)
```

```
        {
```

```
            ViewBag.Message = message;
```

```
            ViewBag.successMessage = successMessage;
```

```
            return View();
```

```
        }
```

```
        [HttpPost]
```

```
        public IActionResult Login(string email, string password)
```

```
        {
```

```
            // ir con el caso de uso
```

```
            UsuarioAltaDto usuario = null;
```

```
            try
```

```
{
    usuario = _obtenerUno.Ejecutar(email);
}
catch (Exception e)
{
    return RedirectToAction("Login", new { message= e.Message});
}

HttpContext.Session.SetString("mail", usuario.Email);

if (usuario.Discriminator == "Admin")
{
    HttpContext.Session.SetString("rol", "admin");
    return Redirect("/usuario/index");
}
else
{
    HttpContext.Session.SetString("rol", "digitador");
    return Redirect("/digitador/index");
}
}

[HttpGet]
public IActionResult Logout()
{
    HttpContext.Session.Clear();
    return RedirectToAction("Login");
}
}
}
```

\*\*\*\*\*

Archivo: UsuarioController.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteApp\Controllers\UsuarioController.cs

\*\*\*\*\*

```
using ComiteLogicaAplicacion.CasoUso.Usuarios;
using ComiteLogicaNegocio.Entidades;
```

```
using ComiteLogicaNegocio.Excepciones;
using ComiteLogicaNegocio.Excepciones.Usuario;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteApp.Models;
using Microsoft.AspNetCore.Mvc;
using System.Diagnostics;
using ComiteCompartido.Dtos.Usuarios;
using Microsoft.EntityFrameworkCore;

namespace ComiteApp.Controllers
{
    public class UsuarioController : Controller
    {
        IAlta<UsuarioAltaDto> _alta;
        IObtenerTodos<UsuarioListadoDto> _obtenerTodos;
        IObtener<UsuarioAltaDto> _obtenerID;
        IEliminar<UsuarioAltaDto> _eliminar;
        IEditar<UsuarioAltaDto> _editar;

        public UsuarioController(
            IAlta<UsuarioAltaDto> alta,
            IObtenerTodos<UsuarioListadoDto> obtenerTodos,
            IObtener<UsuarioAltaDto> obtenerID,
            IEliminar<UsuarioAltaDto> eliminar,
            IEditar<UsuarioAltaDto> editar
        )
        {
            _alta = alta;
            _obtenerTodos = obtenerTodos;
            _obtenerID = obtenerID;
            _eliminar = eliminar;
            _editar = editar;
        }

        [HttpGet]
        public IActionResult Index(string message, string successMessage)
        {
            try
            {
                ViewBag.Message = message;
            }
        }
    }
}
```

---

```
        ViewBag.sucessMessage = sucessMessage;
        return View(_obtenerTodos.Ejecutar());

    } catch (Exception e) {
        ViewBag.Message = e.Message;
        return View();
    }
}

[HttpGet]
public IActionResult Create()
{
    return View();
}

[HttpPost]
public IActionResult Create(UsuarioAltaDto usuario)
{
    try
    {
        _alta.Ejecutar(usuario);
        return RedirectToAction("Index", new { sucessMessage = "Usuario creado
con exito" });
    }
    catch (Exception e)
    {
        ViewBag.Message = e.Message;
    }
    return View(usuario);
}

// GET: Usuarios/Edit/5
public async Task<IActionResult> Edit(int id)
{
    if (id == null)
    {
        return NotFound();
    }
    try
    {
```

---

```
        var Usuario = _obtenerID.Ejecutar(id);
        if (Usuario == null)
        {
            return NotFound();
        }
        return View(Usuario);

    } catch (Exception e) {
        ViewBag.Message = e.Message;
        return View();
    }
}

// POST: Usuarios/Edit/5
// To protect from overposting attacks, enable the specific properties you want to
bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Edit(int id, UsuarioAltaDto Usuario)
{
    if (id != Usuario.Id)
    {
        return NotFound();
    }
    try
    {
        _editar.Ejecutar(Usuario);
    }
    catch (Exception e) {
        ViewBag.Message = e.Message;
        return View();
    }
    return RedirectToAction("Index", new { sucesoMessage = "Usuario editado
con exito" });
}

// GET: Usuarios/Delete/5
public async Task<IActionResult> Delete(int id)
{
```

```
        if (id == null)
        {
            return NotFound();
        }
        try
        {
            var Usuario = _obtenerID.Ejecutar(id);
            if (Usuario == null)
            {
                return NotFound();
            }

            return View(Usuario);
        }
        catch (Exception e)
        {
            ViewBag.Message = e.Message;
            return View();
        }
    }

    // POST: Usuarios/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult> DeleteConfirmed(int id)
    {
        try
        {
            var Usuario = _obtenerID.Ejecutar(id);
            if (Usuario != null)
            {
                _eliminar.Ejecutar(Usuario);
            }
            return RedirectToAction("Index");
        }
        catch (Exception e)
        {
            ViewBag.Message = e.Message;
            return View();
        }
    }
```



```
}
```

```
}  
}
```

```
*****
```

Archivo: ErrorViewModel.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteApp\Models>ErrorViewMo  
del.cs

```
*****
```

```
namespace ComiteApp.Models
```

```
{
```

```
    public class ErrorViewModel
```

```
    {
```

```
        public string? RequestId { get; set; }
```

```
        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
```

```
    }
```

```
}
```

```
*****
```

Archivo: VMUsuario.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteApp\Models\VMUsuario.c  
s

```
*****
```

```
namespace ComiteApp.Models
```

```
{
```

```
    public class VMUsuario
```

```
    {
```

```
        public string Email { get; set; }
```

```
        public string Password { get; set; }
```

```
    }
```

```
}
```

```
*****
```

Archivo: Admin.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Entidades\  
Admin.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Vo.Generic;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace ComiteLogicaNegocio.Entidades
```

```
{  
    public class Admin : Usuario  
    {  
        protected Admin()  
        {  
        }  
  
        public Admin(int id, string email, string password) : base(id, new Email(email),  
new Password(password))  
        {  
        }  
    }  
}
```

\*\*\*\*\*

Archivo: Atleta.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Entidades\  
Atleta.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Vo.Generic;  
using Microsoft.Win32;  
using System;  
using System.Collections;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using static System.Runtime.InteropServices.JavaScript.JSType;
```

```
namespace ComitelogicaNegocio.Entidades
{
    public class Atleta
    {
        public int ID { get; set; }
        public string Nombre { get; set; }
        public string Sexo { get; set; }

        public Pais Pais { get; set; }

        public int PaisId { get; set; }

        public List<Disciplina> Disciplinas { get; set; }
        public List<int> DisciplinasIds { get; set; }

        public Atleta() { }

        public Atleta(int iD, string nombre, string sexo, int paisId, List<int>
disciplinasIds)
        {
            ID = iD;
            Nombre = nombre;
            Sexo = sexo;
            PaisId = paisId;
            DisciplinasIds = disciplinasIds;
            Disciplinas = new List<Disciplina>();
        }
    }
}
```

\*\*\*\*\*

Archivo: Delegado.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Entidades\Delegado.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteLogicaNegocio.Entidades
{
    public class Delegado
    {
    }
}
```

\*\*\*\*\*

Archivo: Digitador.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Entidades\  
Digitador.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Vo.Generic;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteLogicaNegocio.Entidades
{
    public class Digitador : Usuario
    {
        // public Admin adminRegistro { get; set; }
        protected Digitador()
        {
        }
    }
}
```

```
        public Digitador(int id, string email, string password) : base(id, new
Email(email), new Password(password))
        {
        }
    }
}
```

\*\*\*\*\*

Archivo: Disciplina.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Entidades\  
Disciplina.cs

\*\*\*\*\*

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace ComiteLogicaNegocio.Entidades
```

```
{  
    public class Disciplina  
    {  
        public int ID { get; set; }  
        public string Nombre { get; set; }  
        public int Year { get; set; }  
  
        public List<Atleta> Atletas { get; set; }  
  
        public Disciplina() { }  
  
        public Disciplina(int id, string nombre, int year)  
        {  
            ID = id;  
            Nombre = nombre;  
            Year = year;  
        }  
    }  
}
```

\*\*\*\*\*

Archivo: Evento.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Entidades\  
Evento.cs

\*\*\*\*\*

```
using System;  
using System.Collections.Generic;  
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;

namespace ComiteLogicaNegocio.Entidades
{
    public class Evento
    {
        public int ID { get; set; }
        public Disciplina Disciplina { get; set; }

        public int Disciplinald { get; set; }
        public string Nombre { get; set; }
        public DateTime Inicio { get; set; }
        public DateTime Fin { get; set; }

        public List<Atleta> atletas { get; set; }

        protected Evento() { }

        public Evento(int iD, int disciplina, string nombre, DateTime inicio, DateTime fin)
        {
            ID = iD;
            Disciplinald = disciplina;
            Nombre = nombre;
            Inicio = inicio;
            Fin = fin;
            atletas = new List<Atleta>();
        }
    }
}
```

\*\*\*\*\*

Archivo: EventoAtleta.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Entidades\EventoAtleta.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace ComiteLogicaNegocio.Entidades
```

```
{  
    public class EventoAtleta  
    {  
        public int ID { get; set; }  
        public Evento Evento { get; set; }  
        public int Eventold { get; set; }  
        public Atleta Atleta { get; set; }  
        public int Atletald { get; set; }  
        public decimal Puntaje { get; set; }  
  
        protected EventoAtleta() { }  
  
        public EventoAtleta(int iD, int evento, int atleta, decimal puntaje)  
        {  
            ID = iD;  
            Eventold = evento;  
            Atletald = atleta;  
            Puntaje = puntaje;  
        }  
    }  
}
```

```
*****
```

Archivo: Pais.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Entidades\  
Pais.cs

```
*****
```

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace ComiteLogicaNegocio.Entidades
```

```
{  
    public class Pais  
    {
```

```
public int ID { get; set; }
public string NombrePais { get; set; }

public int CantidadHabitantes { get; set; }

public string NombreContacto { get; set; }
public string TelefonoContacto { get; set; }

public Pais() { }
public Pais(int iD, string nombrePais, int cantidadHabitantes, string
nombreContacto, string telefonoContacto)
{
    ID = iD;
    NombrePais = nombrePais;
    CantidadHabitantes = cantidadHabitantes;
    NombreContacto = nombreContacto;
    TelefonoContacto = telefonoContacto;
}
}
```

\*\*\*\*\*

Archivo: Usuario.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Entidades\
Usuario.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Excepciones.Usuario;
using ComiteLogicaNegocio.Vo.Generic;
```

```
namespace ComiteLogicaNegocio.Entidades
```

```
{
    public abstract class Usuario
    {
        private string email;
        private string password;

        public int ID { get; set; }
        public Email Email { get; set; }
        public Password Password { get; set; }
        public string Discriminator { get; set; }
    }
}
```



```
public DateTime fecRegistro { get; set; }
protected Usuario() { }
public Usuario(int id,
    Email email,
    Password password)
{
    ID = id;
    Email = email;
    Password = password;
    fecRegistro = DateTime.Now;
    Validar();
}
private void Validar()
{
}
}
```

\*\*\*\*\*

Archivo: LogicaNegocioException.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Excepciones\LogicaNegocioException.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.Text;
using System.Threading.Tasks;

namespace ComiteLogicaNegocio.Excepciones
{
    public class LogicaNegocioException : Exception
    {
        public LogicaNegocioException()
        {
        }

        public LogicaNegocioException(string? message) : base(message)
        {
        }
    }
}
```

```
    }

    public LogicaNegocioException(string? message, Exception? innerException) :
base(message, innerException)
    {
    }

    protected LogicaNegocioException(SerializationInfo info, StreamingContext
context) : base(info, context)
    {
    }
}
}
```

\*\*\*\*\*

Archivo: IAlta.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
CasoUso\IAlta.cs

\*\*\*\*\*

```
namespace ComiteLogicaNegocio.InterfacesCasoUso
{
    public interface IAlta <T>
    {
        void Ejecutar(T obj);
    }
}
```

\*\*\*\*\*

Archivo: IEditor.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
CasoUso\IEditor.cs

\*\*\*\*\*

```
namespace ComiteLogicaNegocio.InterfacesCasoUso
{
    public interface IEditor <T>
    {
        void Ejecutar(T obj);
    }
}
```

```
}
}
```

\*\*\*\*\*

Archivo: IEliminar.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
CasoUso\IEliminar.cs

\*\*\*\*\*

```
namespace ComiteLogicaNegocio.InterfacesCasoUso
```

```
{
    public interface IEliminar <T>
    {
        void Ejecutar(T obj);
    }
}
```

\*\*\*\*\*

Archivo: IObtener.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
CasoUso\IObtener.cs

\*\*\*\*\*

```
namespace ComiteLogicaNegocio.InterfacesCasoUso
```

```
{
    public interface IObtener <T>
    {
        T Ejecutar(int id);
        T Ejecutar(string s);
    }
}
```

\*\*\*\*\*

Archivo: IObtenerTodos.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
CasoUso\IObtenerTodos.cs

\*\*\*\*\*

```
namespace ComiteLogicaNegocio.InterfacesCasoUso
{
    public interface IObtenerTodos <T>
    {
        IEnumerable<T> Ejecutar();
    }
}
```

\*\*\*\*\*

Archivo: IRepository.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
Repositorio\IRepositorio.cs

\*\*\*\*\*

```
namespace ComiteLogicaNegocio.InterfacesRepositorios
{
    public interface IRepository <T>
    {
        void Add(T obj);
        T GetById(int id);
        IEnumerable<T> GetAll();

    }
}
```

\*\*\*\*\*

Archivo: IRepositoryAtleta.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
Repositorio\IRepositorioAtleta.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteLogicaNegocio.InterfacesRepositorio
{
    public interface IRepositoryAtleta : IRepository<Atleta>
    {
        void Edit(Atleta obj);
    }
}
```

\*\*\*\*\*

Archivo: IRepositoryDisciplina.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
Repositorio\IRepositoryDisciplina.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Entidades;
```

```
namespace ComiteLogicaNegocio.InterfacesRepositorios
{
    public interface IRepositoryDisciplina : IRepository<Disciplina>
    {
        Disciplina GetByName(string nombre);
        void Delete(Disciplina obj);
    }
}
```

\*\*\*\*\*

Archivo: IRepositoryEvento.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
Repositorio\IRepositoryEvento.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteLogicaNegocio.InterfacesRepositorio
```

```
{
    public interface IRepositoryEvento : IRepository<Evento>
    {
        Evento GetByName(string nombre);
        public void AgregarAtletaAlEvento(Atleta atleta);
    }
}
```

\*\*\*\*\*

Archivo: IRepositoryPais.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
Repositorio\IRepositoryPais.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.InterfacesRepositorios;
using ComiteLogicaNegocio.Entidades;
namespace ComiteLogicaNegocio.InterfacesRepositorio
{
    public interface IRepositoryPais : IRepository<Pais>
    {
    }
}
```

\*\*\*\*\*

Archivo: IRepositoryUsuario.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Interfaces  
Repositorio\IRepositoryUsuario.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Entidades;
```

```
namespace ComiteLogicaNegocio.InterfacesRepositorios
{
    public interface IRepositoryUsuario : IRepository<Usuario>
    {
        void Delete(Usuario obj);
        void Edit(Usuario obj);
        Usuario GetByEmail(string email);
    }
}
```

\*\*\*\*\*

Archivo: DisciplinaConfiguration.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\EF\Config\  
DisciplinaConfiguration.cs

\*\*\*\*\*

```
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Microsoft.EntityFrameworkCore.Metadata.Builders;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.Vo.Generic;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
using System.Reflection.Emit;
```

```
namespace ComiteAccesoDatos.EF.Config
```

```
{
    public class DisciplinaConfiguration : IEntityTypeConfiguration<Disciplina>
    {
        public void Configure(EntityTypeBuilder<Disciplina> builder)
        {
            builder.HasIndex(a => a.Nombre).IsUnique();
        }
    }
}
```

\*\*\*\*\*

Archivo: UsuarioConfiguration.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteAccesoADatos\EF\Config\  
UsuarioConfiguration.cs

\*\*\*\*\*

```
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;
using Microsoft.EntityFrameworkCore.Metadata.Builders;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.Vo.Generic;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
using System.Reflection.Emit;

namespace ComiteAccesoDatos.EF.Config
{
    public class UsuarioConfiguration : IEntityTypeConfiguration<Usuario>
    {
        public void Configure(EntityTypeBuilder<Usuario> builder)
        {
            // indico a EF que nombre es un VO de 1 (UN) solo atributo
            // configura el vo de nombre en este caso con hasconversion porque luego
            // vamos a crear un indice unico
            var emailConvert = new ValueConverter<Email, string>(
                v => v.Value,
                v => new Email(v)
            );
            builder.Property(a => a.Email).HasConversion(emailConvert);
            builder.HasIndex(a => a.Email).IsUnique();

            // indico a EF que nombre es un VO de 1 (UN) solo atributo
            // configura el vo de nombre en este caso con hasconversion porque luego
            // vamos a crear un indice unico
            var passwordConvert = new ValueConverter<Password, string>(
                v => v.Value,
                v => new Password(v)
            );
            builder.Property(a => a.Password).HasConversion(passwordConvert);
        }
    }
}
```

\*\*\*\*\*

Archivo: AtletaAltaDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Atleta\AtletaAltaDto.cs



\*\*\*\*\*

```
using ComiteLogicaNegocio.Entidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteCompartido.Dtos.Atletas
{
    public record AtletaAltaDto(
        int id,
        string Nombre,
        string Sexo,
        int paisId,
        List<int> DisciplinasIds
    )
    {
    }
}
```

\*\*\*\*\*

Archivo: AtletaListadoDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Atleta\AtletaListadoDto.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteCompartido.Dtos.Atletas
{
    public record AtletaListadoDto(
        int id,
        string Nombre,
        string Sexo,
        int paisId,
        string NombrePais,
```

```
        List<String> Disciplinas,
        List<int> DisciplinasIds)
    {
    }
}
```

\*\*\*\*\*

Archivo: DisciplinasAltaDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Disciplinas\DisciplinasAltaDto.cs

\*\*\*\*\*

```
namespace ComiteCompartido.Dtos.Disciplinas
{
    public record DisciplinasAltaDto (
        int Id,
        string Nombre,
        int Year
    )
    {
    }
}
```

\*\*\*\*\*

Archivo: DisciplinasListadoDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Disciplinas\DisciplinasListadoDto.cs

\*\*\*\*\*

```
namespace ComiteCompartido.Dtos.Disciplinas
{
    public record DisciplinasListadoDto(
        int Id,
        string Nombre,
        int Year
    )
    {
    }
}
```

\*\*\*\*\*

Archivo: EventoAltaDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Eventos\  
EventoAltaDto.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Entidades;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace ComiteCompartido.Dtos.Eventos
```

```
{  
    public record EventoAltaDto(  
        int id,  
        int DisciplinalId,  
        string Nombre,  
        DateTime Inicio,  
        DateTime Fin,  
        List<int> AtletasIds  
    )  
    {  
    }  
}
```

\*\*\*\*\*

Archivo: EventoListadoDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Eventos\  
EventoListadoDto.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Entidades;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace ComiteCompartido.Dtos.Eventos
{
    public record EventoListadoDto(
        int id,
        string NombreDisciplina,
        string Nombre,
        DateTime Inicio,
        DateTime Fin
    )
    {
    }
}
```

\*\*\*\*\*

Archivo: AtletaMapper.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Mappers  
\AtletaMapper.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Atletas;
using ComiteCompartido.Dtos.Disciplinas;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.Vo.Generic;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteCompartido.Dtos.Mappers
{
    public class AtletaMapper
    {
        public static Atleta FromDto(AtletaAltaDto item)
        {
            return new Atleta(item.id,
                item.Nombre,
                item.Sexo,
                item.paisId,
                item.DisciplinasIds);
        }
    }
}
```

```
    }

    public static AtletaAltaDto ToDto(Atleta item)
    {
        return new AtletaAltaDto(
            item.ID,
            item.Nombre,
            item.Sexo,
            item.PaisId,
            item.DisciplinasIds
        );
    }

    public static IEnumerable<AtletaListadoDto> ToListaDto(IEnumerable<Atleta>
items)
    {

        List<AtletaListadoDto> dtos = new List<AtletaListadoDto>();

        foreach (Atleta item in items)
        {
            dtos.Add(new AtletaListadoDto(
                item.ID,
                item.Nombre,
                item.Sexo,
                item.PaisId,
                item.Pais.NombrePais,
                item.Disciplinas.Select(d => d.Nombre).ToList(),
                item.DisciplinasIds
            ));
        }

        return dtos;
    }
}
```

\*\*\*\*\*

Archivo: DisciplinaMapper.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Mappers  
\DisciplinaMapper.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Disciplinas;
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesRepositorios;
using ComiteLogicaNegocio.Vo.Generic;

namespace ComiteCompartido.Dtos.MappersDisciplina
{
    public class DisciplinaMapper
    {
        public static Disciplina FromDto(DisciplinasAltaDto item)
        {
            return new Disciplina(item.Id, item.Nombre, item.Year);
        }
        public static DisciplinasAltaDto ToDto(Disciplina? item)
        {
            return new DisciplinasAltaDto(
                item.ID,
                item.Nombre,
                item.Year
            );
        }
        public static IEnumerable<DisciplinasListadoDto>
        ToListaDto(IEnumerable<Disciplina> items)
        {
            List<DisciplinasListadoDto> dtos = new List<DisciplinasListadoDto>();

            foreach (Disciplina item in items)
            {
                dtos.Add(new DisciplinasListadoDto(
                    item.ID,
                    item.Nombre,
                    item.Year
                ));
            }
        }
    }
}
```

```
        return dtos;
    }
}
```

```
}
```

```
}
```

\*\*\*\*\*

Archivo: EventoMapper.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Mappers  
\EventoMapper.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Eventos;
using ComiteLogicaNegocio.Entidades;
```

```
namespace ComiteCompartido.Dtos.Mappers
```

```
{
```

```
    public class EventoMapper
```

```
    {
```

```
        public static Evento FromDto(EventoAltaDto evento) {
```

```
            return new Evento(
```

```
                evento.id,
```

```
                evento.DisciplinalId,
```

```
                evento.Nombre,
```

```
                evento.Inicio,
```

```
                evento.Fin
```

```
            );
```

```
        }
```

```
        public static EventoAltaDto ToDto(Evento evento)
```

```
        {
```

```
            return new EventoAltaDto(
```

```
                evento.ID,
```

```
                evento.DisciplinalId,
```

```
                evento.Nombre,
```

```
                evento.Inicio,
```

```
                evento.Fin,
```

```
                evento.atletas.Select(a=>a.ID).ToList()
```

```
        );
    }
    public static IEnumerable<EventoListadoDto>
    ToListadoDto(IEnumerable<Evento> items)
    {
        List<EventoListadoDto> dtos = new List<EventoListadoDto>();

        foreach (Evento item in items)
        {
            dtos.Add(new EventoListadoDto(
                item.ID,
                item.Disciplina.Nombre,
                item.Nombre,
                item.Inicio,
                item.Fin
            ));
        }

        return dtos;
    }
}
```

\*\*\*\*\*

Archivo: PaisMapper.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Mappers  
\PaisMapper.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Disciplinas;
using ComiteCompartido.Dtos.Paises;
using ComiteLogicaNegocio.Entidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteCompartido.Dtos.Mappers
{
```



---

```
public class PaisMapper
{
    public static Pais FromDto(PaisAltaDto item)
    {
        return new Pais(item.id, item.NombrePais, item.CantidadHabitantes,
            item.NombreContacto,
            item.TelefonoContacto);
    }

    public static PaisAltaDto ToDto(Pais? item)
    {
        return new PaisAltaDto(
            item.ID,
            item.NombrePais,
            item.CantidadHabitantes,
            item.NombreContacto,
            item.TelefonoContacto
        );
    }

    public static IEnumerable<PaisListadoDto> ToListaDto(IEnumerable<Pais>
items)
    {
        List<PaisListadoDto> dtos = new List<PaisListadoDto>();

        foreach (Pais item in items)
        {
            dtos.Add(new PaisListadoDto(
                item.ID,
                item.NombrePais,
                item.CantidadHabitantes,
                item.NombreContacto,
                item.TelefonoContacto
            ));
        }

        return dtos;
    }
}
```

\*\*\*\*\*

Archivo: UsuarioMapper.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Mappers  
\UsuarioMapper.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesRepositorios;
using ComiteLogicaNegocio.Vo.Generic;
```

```
namespace ComiteCompartido.Dtos.Mappers
```

```
{
    public class UsuarioMapper
    {
        public static Digitador FromDto(UsuarioAltaDto usuario)
        {
            return new Digitador(usuario.Id, usuario.Email, usuario.Password);
        }
        public static Admin FromDtoToAdmin(UsuarioAltaDto usuario)
        {
            return new Admin(usuario.Id, usuario.Email, usuario.Password);
        }
        public static UsuarioAltaDto ToDto(Usuario? usuario)
        {
            bool admin = false;
            if (usuario.Discriminator == "Admin")
            {
                admin = true;
            }

            return new UsuarioAltaDto(
                usuario.ID,
                usuario.Email.Value,
                usuario.Password.Value,
                usuario.Discriminator
            );
        }
        else
        {

```

```
// Si es digitador preciso el id del admin que lo creo
return new UsuarioAltaDto(
    usuario.ID,
    usuario.Email.Value,
    usuario.Password.Value,
    usuario.Discriminator
);
}

}

public static IEnumerable<UsuarioListadoDto>
ToListadoDto(IEnumerable<Usuario> usuarios)
{

    List<UsuarioListadoDto> dtos = new List<UsuarioListadoDto>();

    foreach (Usuario item in usuarios)
    {
        dtos.Add(new UsuarioListadoDto(
            item.ID,
            item.Email.Value,
            item.Password.Value,
            item.Discriminator
        ));
    }

    return dtos;
}

}
}
```

\*\*\*\*\*

Archivo: PaisAltaDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Países\PaisAltaDto.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteCompartido.Dtos.Paises
{
    public record PaisAltaDto(
        int id,
        string NombrePais,
        int CantidadHabitantes,
        string NombreContacto,
        string TelefonoContacto)
    {
    }
}
```

\*\*\*\*\*

Archivo: PaisListadoDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Paises\PaisListadoDto.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteCompartido.Dtos.Paises
{
    public record PaisListadoDto(
        int id,
        string NombrePais,
        int CantidadHabitantes,
        string NombreContacto,
        string TelefonoContacto)
    {
    }
}
```

\*\*\*\*\*

Archivo: UsuarioAltaDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Usuarios  
 \UsuarioAltaDto.cs

\*\*\*\*\*

namespace ComiteCompartido.Dtos.Usuarios

```
{
    public record UsuarioAltaDto (
        int Id,
        string Email,
        string Password,
        string Discriminator
    )
    {
    }
}
```

\*\*\*\*\*

Archivo: UsuarioListadoDto.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteCompartido\Dtos\Usuarios  
 \UsuarioListadoDto.cs

\*\*\*\*\*

namespace ComiteCompartido.Dtos.Usuarios

```
{
    public record UsuarioListadoDto(
        int Id,
        string Email,
        string Password,
        string Discriminator
    )
    {
    }
}
```

\*\*\*\*\*

Archivo: EditarAtleta.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUso\CasoUsoAtleta\EditarAtleta.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Atletas;
using ComiteCompartido.Dtos.Mappers;
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorio;
using ComiteLogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteLogicaAplicacion.CasoUso.CasoUsoAtleta
```

```
{
    public class EditarAtleta : IEditar<AtletaAltaDto>
    {
        IRepositoryAtleta _repositorio;

        public EditarAtleta(IRepositoryAtleta repositorio)
        {
            _repositorio = repositorio;
        }

        public void Ejecutar(AtletaAltaDto obj)
        {
            _repositorio.Edit(AtletaMapper.FromDto(obj));
        }
    }
}
```

\*\*\*\*\*

Archivo: ObtenerAtleta.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUso\CasoUsoAtleta\ObtenerAtleta.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Atletas;
```

```
using ComiteCompartido.Dtos.Mappers;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteLogicaAplicacion.CasoUso.CasoUsoAtleta
{
    public class ObtenerAtleta : IObtener<AtletaAltaDto>
    {
        IRepositoryAtleta _repositorio;

        public ObtenerAtleta(IRepositoryAtleta repositorio)
        {
            _repositorio = repositorio;
        }
        public AtletaAltaDto Ejecutar(int id)
        {
            return AtletaMapper.ToDto(_repositorio.GetById(id));
        }

        public AtletaAltaDto Ejecutar(string s)
        {
            throw new NotImplementedException();
        }
    }
}
```

\*\*\*\*\*

Archivo: ObtenerAtletas.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\CasoUsoAtleta\ObtenerAtletas.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Atletas;
using ComiteCompartido.Dtos.Disciplinas;
using ComiteCompartido.Dtos.Mappers;
using ComiteCompartido.Dtos.MappersDisciplina;
```

```
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorio;
using ComiteLogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteLogicaAplicacion.CasoUso.CasoUsoAtleta
{
    public class ObtenerAtletas : IObtenerTodos<AtletaListadoDto>
    {
        IRepositoryAtleta _repositorio;

        public ObtenerAtletas(IRepositoryAtleta repositorio)
        {
            _repositorio = repositorio;
        }
        public IEnumerable<AtletaListadoDto> Ejecutar()
        {
            return AtletaMapper.ToListaDto(_repositorio.GetAll());
        }
    }
}
```

\*\*\*\*\*

Archivo: AltaEvento.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUso\CasoUsoEvento\AltaEvento.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Disciplinas;
using ComiteCompartido.Dtos.Eventos;
using ComiteCompartido.Dtos.Mappers;
using ComiteCompartido.Dtos.MappersDisciplina;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorio;
using ComiteLogicaNegocio.InterfacesRepositorios;
```



---

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteLogicaAplicacion.CasoUso.CasoUsoEvento
{
    public class AltaEvento : IAlta<EventoAltaDto>
    {
        IRepositoryEvento _repositorio;
        IRepositoryAtleta _atletas;
        IRepositoryDisciplina _disciplinas;

        public AltaEvento(IRepositoryEvento repositorio, IRepositoryAtleta atletas,
        IRepositoryDisciplina disciplinas)
        {
            _repositorio = repositorio;
            _atletas = atletas;
            _disciplinas = disciplinas;
        }

        public void Ejecutar(EventoAltaDto obj)
        {
            /*
                De los eventos se conoce la disciplina, el nombre de la prueba (por ejemplo:
                salto largo, 100 metros mariposa,
                gimnasia rítmica), las fechas en que inicia y finaliza (si la duración es de un
                día ambas fechas serán iguales) y
                los atletas participantes. Cada atleta obtiene un puntaje en el evento (por el
                momento, un número decimal).
                Debe controlarse que en un evento solamente participen atletas que se
                hayan registrado para competir en la
                disciplina del evento.
                Se debe validar que no exista otro evento con el mismo nombre de prueba
                previamente
                ingresado.
                o Un evento deberá tener al menos tres atletas registrados.
                o Los atletas registrados deberán estar registrados para participar en la
                disciplina del evento.
            */
        }
    }
}
```

```
Evento e = EventoMapper.FromDto(obj);
Disciplina d = _disciplinas.GetById(obj.Disciplinald);
e.Disciplina = d;
int cantidadAtletas = 0;
foreach (int atletald in obj.AtletasIds)
{
    Atleta a = _atletas.GetById(atletald);
    if (a != null) {
        if (!a.DisciplinasIds.Contains(obj.Disciplinald))
        {
            throw new Exception("El atleta debe pertenecer a la disciplina del
evento");
        }
        cantidadAtletas++;
        e.atletas.Add(a);
    }
}
if (cantidadAtletas >= 3)
{
    _repositorio.Add(e);
}
else {
    throw new Exception("El evento debe tener al menos 3 atletas");
}
}
}
```

\*\*\*\*\*

Archivo: IngresarPuntaje.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\CasoUsoEvento\IngresarPuntaje.cs

\*\*\*\*\*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ComiteLogicaAplicacion.CasoUso.CasoUsoEvento
```

```
{  
    public class IngresarPuntaje  
    {  
    }  
}
```

\*\*\*\*\*

Archivo: ObtenerEvento.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\CasoUsoEvento\ObtenerEvento.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Disciplinas;  
using ComiteCompartido.Dtos.Eventos;  
using ComiteCompartido.Dtos.Mappers;  
using ComiteCompartido.Dtos.MappersDisciplina;  
using ComiteLogicaNegocio.Entidades;  
using ComiteLogicaNegocio.InterfacesCasoUso;  
using ComiteLogicaNegocio.InterfacesRepositorio;  
using ComiteLogicaNegocio.InterfacesRepositorios;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace ComiteLogicaAplicacion.CasoUso.CasoUsoEvento  
{  
    public class ObtenerEvento : IObtener<EventoAltaDto>  
    {  
        IRepositoryEvento _repositorio;  
  
        public ObtenerEvento(IRepositoryEvento repositorio)  
        {  
            _repositorio = repositorio;  
        }  
        public EventoAltaDto Ejecutar(int id)  
        {  
            Evento e = _repositorio.GetById(id);  
            EventoAltaDto edto = EventoMapper.ToDto(e);  
            return edto;  
        }  
    }  
}
```

```
    }  
    public EventoAltaDto Ejecutar(string nombre)  
    {  
        return EventoMapper.ToDto(_repositorio.GetByName(nombre));  
    }  
}  
}
```

\*\*\*\*\*

Archivo: ObtenerEventos.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\CasoUsoEvento\ObtenerEventos.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Disciplinas;  
using ComiteCompartido.Dtos.Eventos;  
using ComiteCompartido.Dtos.Mappers;  
using ComiteCompartido.Dtos.MappersDisciplina;  
using ComiteLogicaNegocio.InterfacesCasoUso;  
using ComiteLogicaNegocio.InterfacesRepositorio;  
using ComiteLogicaNegocio.InterfacesRepositorios;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace ComiteLogicaAplicacion.CasoUso.CasoUsoEvento  
{  
    public class ObtenerEventos : IObtenerTodos<EventoListadoDto>  
    {  
        IRepositoryEvento _repositorio;  
  
        public ObtenerEventos(IRepositoryEvento repositorio)  
        {  
            _repositorio = repositorio;  
        }  
        public IEnumerable<EventoListadoDto> Ejecutar()  
        {  
            return EventoMapper.ToListaDto(_repositorio.GetAll());  
        }  
    }  
}
```

```
}  
}
```

\*\*\*\*\*

Archivo: AltaDisciplina.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Disciplinas\AltaDisciplina.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Mappers;  
using ComiteCompartido.Dtos.Disciplinas;  
using ComiteLogicaNegocio.Entidades;  
using ComiteLogicaNegocio.InterfacesCasoUso;  
using ComiteLogicaNegocio.InterfacesRepositorios;  
using ComiteCompartido.Dtos.MappersDisciplina;  
  
namespace ComiteLogicaAplicacion.CasoUso.Disciplinas  
{  
    public class AltaDisciplina : IAlta<DisciplinasAltaDto>  
    {  
        IRepositoryDisciplina _repositorio;  
  
        public AltaDisciplina(IRepositoryDisciplina repositorio)  
        {  
            _repositorio = repositorio;  
        }  
  
        public void Ejecutar(DisciplinasAltaDto obj)  
        {  
            _repositorio.Add(DisciplinaMapper.FromDto(obj));  
        }  
    }  
}
```

\*\*\*\*\*

Archivo: EliminarDisciplina.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Disciplinas\EliminarDisciplina.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Mappers;
using ComiteCompartido.Dtos.Disciplinas;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorios;
using ComiteCompartido.Dtos.MappersDisciplina;

namespace ComiteLogicaAplicacion.CasoUso.Disciplinas
{
    public class EliminarDisciplina : IEliminar<DisciplinasAltaDto>
    {
        IRepositoryDisciplina _repositorio;

        public EliminarDisciplina(IRepositoryDisciplina repositorio)
        {
            _repositorio = repositorio;
        }

        public void Ejecutar(DisciplinasAltaDto obj)
        {
            try
            {
                _repositorio.Delete(DisciplinaMapper.FromDto(obj));
            }
            catch (Exception ex)
            {
                throw new Exception("Error al eliminar el disciplina");
            }
        }
    }
}
```

\*\*\*\*\*

Archivo: ObtenerDisciplina.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Disciplinas\ObtenerDisciplina.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Disciplinas;
```

```
using ComiteCompartido.Dtos.Mappers;
using ComiteCompartido.Dtos.MappersDisciplina;
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteLogicaNegocio.CasoUso.Disciplinas
{
    public class ObtenerDisciplina : IObtener<DisciplinasAltaDto>
    {
        IRepositoryDisciplina _repositorio;

        public ObtenerDisciplina(IRepositoryDisciplina repositorio)
        {
            _repositorio = repositorio;
        }
        public DisciplinasAltaDto Ejecutar(int id)
        {
            return DisciplinaMapper.ToDto(_repositorio.GetById(id));
        }
        public DisciplinasAltaDto Ejecutar(string nombre)
        {
            return DisciplinaMapper.ToDto(_repositorio.GetByName(nombre));
        }
    }
}
```

\*\*\*\*\*

Archivo: ObtenerDisciplinas.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Disciplinas\ObtenerDisciplinas.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Disciplinas;
using ComiteCompartido.Dtos.Mappers;
using ComiteCompartido.Dtos.MappersDisciplina;
```

```
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorios;

namespace ComiteLogicaAplicacion.CasoUso.Disciplinas
{
    public class ObtenerDisciplinas : IObtenerTodos<DisciplinasListadoDto>
    {
        IRepositoryDisciplina _repositorio;

        public ObtenerDisciplinas(IRepositoryDisciplina repositorio)
        {
            _repositorio = repositorio;
        }
        public IEnumerable<DisciplinasListadoDto> Ejecutar()
        {
            return DisciplinaMapper.ToListaDto(_repositorio.GetAll());
        }
    }
}
```

\*\*\*\*\*

Archivo: ObtenerPaises.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Pais\ObtenerPaises.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Paises;
using ComiteCompartido.Dtos.MappersDisciplina;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ComiteLogicaNegocio.InterfacesRepositorio;
using ComiteCompartido.Dtos.Mappers;
```

```
namespace ComiteLogicaAplicacion.CasoUso.Pais
```



```
{
    public class ObtenerPaises : IObtenerTodos<PaisListadoDto>
    {
        IRepositoryPais _repositorio;

        public ObtenerPaises(IRepositoryPais repositorio)
        {
            _repositorio = repositorio;
        }
        public IEnumerable<PaisListadoDto> Ejecutar()
        {
            return PaisMapper.ToListaDto(_repositorio.GetAll());
        }
    }
}
```

\*\*\*\*\*

Archivo: AltaUsuario.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Usuarios\AltaUsuario.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Mappers;
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorios;

namespace ComiteLogicaAplicacion.CasoUso.Usuarios
{
    public class AltaUsuario : IAlta<UsuarioAltaDto>
    {
        IRepositoryUsuario _repositorio;

        public AltaUsuario(IRepositoryUsuario repositorio)
        {
            _repositorio = repositorio;
        }

        public void Ejecutar(UsuarioAltaDto obj)
```

```
{
    if (obj.Discriminator == "Admin")
    {
        _repositorio.Add(UsuarioMapper.FromDtoToAdmin(obj));
    }
    _repositorio.Add(UsuarioMapper.FromDto(obj));
}
}
```

\*\*\*\*\*

Archivo: EditarUsuario.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Usuarios\EditarUsuario.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Mappers;
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorios;

namespace ComiteLogicaAplicacion.CasoUso.Usuarios
{
    public class EditarUsuario : IEditar<UsuarioAltaDto>
    {
        IRepositoryUsuario _repositorio;

        public EditarUsuario(IRepositoryUsuario repositorio)
        {
            _repositorio = repositorio;
        }

        public void Ejecutar(UsuarioAltaDto obj)
        {
            _repositorio.Edit(UsuarioMapper.FromDto(obj));
        }
    }
}
```

\*\*\*\*\*

Archivo: EliminarUsuario.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Usuarios\EliminarUsuario.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Mappers;  
using ComiteCompartido.Dtos.Usuarios;  
using ComiteLogicaNegocio.Entidades;  
using ComiteLogicaNegocio.InterfacesCasoUso;  
using ComiteLogicaNegocio.InterfacesRepositorios;
```

```
namespace ComiteLogicaAplicacion.CasoUso.Usuarios  
{  
    public class EliminarUsuario : IEliminar<UsuarioAltaDto>  
    {  
        IRepositoryUsuario _repositorio;  
  
        public EliminarUsuario(IRepositoryUsuario repositorio)  
        {  
            _repositorio = repositorio;  
        }  
  
        public void Ejecutar(UsuarioAltaDto obj)  
        {  
            _repositorio.Delete(UsuarioMapper.FromDto(obj));  
        }  
    }  
}
```

\*\*\*\*\*

Archivo: ObtenerUsuario.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Usuarios\ObtenerUsuario.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Mappers;  
using ComiteCompartido.Dtos.Usuarios;  
using ComiteLogicaNegocio.InterfacesCasoUso;
```

```
using ComiteLogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ComiteLogicaNegocio.CasoUso.Usuarios
{
    public class ObtenerUsuario : IObtener<UsuarioAltaDto>
    {
        IRepositoryUsuario _repositorio;

        public ObtenerUsuario(IRepositoryUsuario repositorio)
        {
            _repositorio = repositorio;
        }
        public UsuarioAltaDto Ejecutar(int id)
        {
            return UsuarioMapper.ToDto(_repositorio.GetById(id));
        }
        public UsuarioAltaDto Ejecutar(string email)
        {
            return UsuarioMapper.ToDto(_repositorio.GetByEmail(email));
        }
    }
}
```

\*\*\*\*\*

Archivo: ObtenerUsuarios.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaAplicacion\CasoUs  
o\Usuarios\ObtenerUsuarios.cs

\*\*\*\*\*

```
using ComiteCompartido.Dtos.Mappers;
using ComiteCompartido.Dtos.Usuarios;
using ComiteLogicaNegocio.Entidades;
using ComiteLogicaNegocio.InterfacesCasoUso;
using ComiteLogicaNegocio.InterfacesRepositorios;
```

```
namespace ComiteLogicaAplicacion.CasoUso.Usuarios
```

```
{
    public class ObtenerUsuarios : IObtenerTodos<UsuarioListadoDto>
    {
        IRepositoryUsuario _repositorio;

        public ObtenerUsuarios(IRepositoryUsuario repositorio)
        {
            _repositorio = repositorio;
        }
        public IEnumerable<UsuarioListadoDto> Ejecutar()
        {
            return UsuarioMapper.ToListaDto(_repositorio.GetAll());
        }
    }
}
```

\*\*\*\*\*

Archivo: EmailUsuarioInvalidoException.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Excepciones\Usuario\EmailUsuarioInvalidoException.cs

\*\*\*\*\*

namespace ComiteLogicaNegocio.Excepciones.Usuario

```
{
    public class EmailUsuarioInvalidoException : UsuarioException
    {
        public EmailUsuarioInvalidoException()
        {
        }

        public EmailUsuarioInvalidoException(string? message) : base(message)
        {
        }

        public EmailUsuarioInvalidoException(string? message, Exception?
innerException) : base(message, innerException)
        {
        }
    }
}
```

\*\*\*\*\*

Archivo: PasswordUsuarioInvalidoException .cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Excepciones\Usuario\PasswordUsuarioInvalidoException .cs

\*\*\*\*\*

namespace ComiteLogicaNegocio.Excepciones.Usuario

{

public class PasswordUsuarioInvalidoException : UsuarioException

{

public PasswordUsuarioInvalidoException()

{

}

public PasswordUsuarioInvalidoException(string? message) : base(message)

{

}

public PasswordUsuarioInvalidoException(string? message, Exception? innerException) : base(message, innerException)

{

}

}

}

\*\*\*\*\*

Archivo: UsuarioException.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Excepciones\Usuario\UsuarioException.cs

\*\*\*\*\*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

namespace ComiteLogicaNegocio.Excepciones.Usuario

{

public class UsuarioException : LogicaNegocioException

```
{
    public UsuarioException()
    {
    }

    public UsuarioException(string? message) : base(message)
    {
    }

    public UsuarioException(string? message, Exception? innerException) :
base(message, innerException)
    {
    }

    protected UsuarioException(SerializationInfo info, StreamingContext context) :
base(info, context)
    {
    }
}
```

\*\*\*\*\*

Archivo: EmailException.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Vo\Excepciones\EmailException.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Excepciones;
using System;
```

```
namespace ComiteLogicaNegocio.Vo.Excepciones
{
    public class EmailException : LogicaNegocioException
    {
        public EmailException()
        {
        }

        public EmailException(string? message) : base(message)
        {
        }
    }
}
```

```
        public EmailException(string? message, System.Exception? innerException) :  
base(message, innerException)  
        {  
        }  
    }  
}
```

\*\*\*\*\*

Archivo: NombreException.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Vo\Excepciones\NombreException.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Excepciones;  
using System;
```

```
namespace ComiteLogicaNegocio.Vo.Excepciones
```

```
{  
    public class NombreException : LogicaNegocioException  
    {  
        public NombreException()  
        {  
        }  
  
        public NombreException(string? message) : base(message)  
        {  
        }  
  
        public NombreException(string? message, System.Exception? innerException)  
: base(message, innerException)  
        {  
        }  
    }  
}
```

\*\*\*\*\*

Archivo: PasswordException.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Vo\Excepciones\PasswordException.cs



\*\*\*\*\*

```
using ComitelogicaNegocio.Excepciones;
using System;

namespace ComitelogicaNegocio.Vo.Excepciones
{
    public class PasswordException : LogicaNegocioException
    {
        public PasswordException()
        {
        }

        public PasswordException(string? message) : base(message)
        {
        }

        public PasswordException(string? message, System.Exception?
innerException) : base(message, innerException)
        {
        }
    }
}
```

\*\*\*\*\*

Archivo: Email.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Vo\Genera  
les\Email.cs

\*\*\*\*\*

```
using ComitelogicaNegocio.Vo.Excepciones;
```

```
namespace ComitelogicaNegocio.Vo.Generic
{
    public record Email
    {
        public string Value { get; set; }

        public Email(string value)
```

```
{
    if (string.IsNullOrEmpty(value) )
    {
        throw new EmailException("El email no puede ser vacio");
    }
    if (value.IndexOf('@') == -1)
    {
        throw new EmailException("Su email debe tener @");
    }
    if (value.IndexOf('@') == 0)
    {
        throw new EmailException("Su email no puede empezar con @");
    }
    if (value.IndexOf('@') == value.Length - 1)
    {
        throw new EmailException("Su email no puede terminar con @");
    }
    if (value.IndexOf('@') != value.LastIndexOf('@'))
    {
        throw new EmailException("Su email no puede tener mas de un @");
    }

    this.Value = value;
}

}
```

\*\*\*\*\*

Archivo: Nombre.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Vo\Generales\Nombre.cs

\*\*\*\*\*

using ComiteLogicaNegocio.Vo.Excepciones;

namespace ComiteLogicaNegocio.Vo.Generic

{

```
public record Nombre
{
    public string Value { get; set; }

    public Nombre(string value)
    {
        if (string.IsNullOrEmpty(value) )
        {
            throw new NombreException("El nombre no puede ser vacio");
        }
        this.Value = value;
    }
}
```

\*\*\*\*\*

Archivo: Password.cs

Carpeta:

C:\Users\joaco\source\repos\Obligatorio1-P3-2024\ComiteLogicaNegocio\Vo\Generales\Password.cs

\*\*\*\*\*

```
using ComiteLogicaNegocio.Vo.Excepciones;
```

```
namespace ComiteLogicaNegocio.Vo.Generic
```

```
{
    public record Password
    {
        public string Value { get; set; }

        public Password(string value)
        {
            if (string.IsNullOrEmpty(value) )
            {
                throw new PasswordException("La contraseña no puede ser vacia");
            }
            if (value.Length < 6)
            {

```

```
        throw new PasswordException("La contraseña no puede ser menor a 6
caracteres");
    }
    if (value.Any(char.IsUpper) == false)
    {
        throw new PasswordException("La contraseña debe contener una
mayuscula");
    }
    if (value.Any(char.IsLower) == false)
    {
        throw new PasswordException("La contraseña debe contener una
minuscula");
    }
    if (value.Any(char.IsDigit) == false)
    {
        throw new PasswordException("La contraseña debe contener un digito
especial");
    }
    if (value.Any(char.IsPunctuation) == false)
    {
        throw new PasswordException("La contraseña debe contener un caracter
de puntuacion");
    }
    this.Value = value;
}

}
}
```