

Evidencia 3 - Reporte

Bases de datos

Profesora: Iris Gabriela Arrona Cardoza

Alumna: Ana Patricia Diosdado Trejo (2980620)

Código:

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

public class Casa
{
    public string Nombre { get; set; }
    public string Fundador { get; set; }
    public string Animal { get; set; }
    public string Colores { get; set; }
    public string JefeCasa { get; set; }
    public int PuntajeCopa { get; set; }
}

public class Alumno
{
    public string Nombre { get; set; }
    public string Apellido { get; set; }
    public DateTime FechaNacimiento { get; set; }
    public DateTime FechaIngreso { get; set; }
    public int AñoActual { get; set; }
    public string Prefecto { get; set; }
    public Casa Casa { get; set; }
    public int PuntoCopa { get; set; }
}

class Program
{
    private static List<Alumno> listaAlumnos = new List<Alumno>();
    private static List<Casa> listaCasas = new List<Casa>();
```

```
static void Main(string[] args)
{
    AgregarDatosDeEjemplo();

    while (true)
    {
        Console.WriteLine("Seleccione una opción:");
        Console.WriteLine("1. Buscar información por colección");
        Console.WriteLine("2. Buscar información modelando relación");
        Console.WriteLine("3. Actualizar información");
        Console.WriteLine("4. Eliminar información");
        Console.WriteLine("5. Salir");

        int opcion = int.Parse(Console.ReadLine());

        switch (opcion)
        {
            case 1:
                BuscarInformacionPorColeccion();
                break;
            case 2:
                BuscarInformacionPorModeloDeRelacion();
                break;
            case 3:
                ActualizarInformacion();
                break;
            case 4:
                EliminarInformacion();
                break;
            case 5:
                return;
            default:
                Console.WriteLine("Opción inválida. Intente de nuevo.");
                break;
        }
    }
}
```

```
}

static void AgregarDatosDeEjemplo()
{
    Casa gryffindor = new Casa
    {
        Nombre = "Gryffindor",
        Fundador = "Godric Gryffindor",
        Animal = "León",
        Colores = "Rojo y Dorado",
        JefeCasa = "Minerva McGonagall",
        PuntajeCopa = 190
    };
    listaCasas.Add(gryffindor);

    Casa slytherin = new Casa
    {
        Nombre = "Slytherin",
        Fundador = "Salazar Slytherin",
        Animal = "Serpiente",
        Colores = "Verde y Plateado",
        JefeCasa = "Severus Snape",
        PuntajeCopa = 170
    };
    listaCasas.Add(slytherin);

    Casa ravenclaw = new Casa
    {
        Nombre = "Ravenclaw",
        Fundador = "Rowena Ravenclaw",
        Animal = "Águila",
        Colores = "Azul y Bronce",
        JefeCasa = "Filius Flitwick",
        PuntajeCopa = 180
    };
    listaCasas.Add(ravenclaw);
}
```

```
Casa hufflepuff = new Casa
{
    Nombre = "Hufflepuff",
    Fundador = "Helga Hufflepuff",
    Animal = "Tejón",
    Colores = "Amarillo y Negro",
    JefeCasa = "Pomona Sprout",
    PuntajeCopa = 160
};
listaCasas.Add(hufflepuff);
```

```
Alumno harryPotter = new Alumno
{
    Nombre = "Harry",
    Apellido = "Potter",
    FechaNacimiento = new DateTime(1980, 7, 31),
    FechaIngreso = new DateTime(1991, 9, 1),
    AñoActual = 1,
    Prefecto = "Gryffindor",
    Casa = gryffindor,
    PuntoCopa = 100
};
listaAlumnos.Add(harryPotter);
```

```
Alumno hermioneGranger = new Alumno
{
    Nombre = "Hermione",
    Apellido = "Granger",
    FechaNacimiento = new DateTime(1979, 9, 19),
    FechaIngreso = new DateTime(1991, 9, 1),
    AñoActual = 1,
    Prefecto = "Gryffindor",
    Casa = gryffindor,
    PuntoCopa = 90
};
listaAlumnos.Add(hermioneGranger);
```

```
Alumno dracoMalfoy = new Alumno
{
    Nombre = "Draco",
    Apellido = "Malfoy",
    FechaNacimiento = new DateTime(1980, 6, 5),
    FechaIngreso = new DateTime(1991, 9, 1),
    AñoActual = 1,
    Prefecto = "",
    Casa = slytherin,
    PuntoCopa = 80
};
listaAlumnos.Add(dracoMalfoy);

Alumno lunaLovegood = new Alumno
{
    Nombre = "Luna",
    Apellido = "Lovegood",
    FechaNacimiento = new DateTime(1981, 2, 13),
    FechaIngreso = new DateTime(1992, 9, 1),
    AñoActual = 1,
    Prefecto = "",
    Casa = ravenclaw,
    PuntoCopa = 75
};
listaAlumnos.Add(lunaLovegood);

Alumno cedricDiggory = new Alumno
{
    Nombre = "Cedric",
    Apellido = "Diggory",
    FechaNacimiento = new DateTime(1977, 10, 1),
    FechaIngreso = new DateTime(1990, 9, 1),
    AñoActual = 2,
    Prefecto = "",
    Casa = hufflepuff,
    PuntoCopa = 85
};
```

```

        listaAlumnos.Add(cedricDiggory);
    }

    static void BuscarInformacionPorColeccion()
    {
        Console.WriteLine("Seleccione una opción:");
        Console.WriteLine("1. Buscar alumno específico");
        Console.WriteLine("2. Mostrar todos los prefectos");

        int opcion = int.Parse(Console.ReadLine());

        switch (opcion)
        {
            case 1:
                Console.Write("Ingrese el nombre del alumno: ");
                string nombreAlumno = Console.ReadLine();

                var alumno = listaAlumnos.Find(a => a.Nombre ==
nombreAlumno);
                if (alumno != null)
                {
                    Console.WriteLine($"Alumno: {alumno.Nombre}
{alumno.Apellido}");
                    Console.WriteLine($"Casa: {alumno.Casa.Nombre}");
                    Console.WriteLine($"Fundador de la casa:
{alumno.Casa.Fundador}");
                    Console.WriteLine($"Animal de la casa:
{alumno.Casa.Animal}");
                    Console.WriteLine($"Colores de la casa:
{alumno.Casa.Colores}");
                    Console.WriteLine($"Jefe de la casa:
{alumno.Casa.JefeCasa}");
                    Console.WriteLine($"Puntaje de Copa de la casa:
{alumno.Casa.PuntajeCopa}");
                }
                else
                {

```

```

        Console.WriteLine("El alumno no se encontró en la base
de datos.");
    }
    break;

    case 2:
        var prefectos = listaAlumnos.Where(a =>
!string.IsNullOrEmpty(a.Prefecto));

        if (prefectos.Any())
        {
            Console.WriteLine("Prefectos:");
            foreach (var prefecto in prefectos)
            {
                Console.WriteLine($"Nombre: {prefecto.Nombre}
{prefecto.Apellido}");
                Console.WriteLine($"Casa:
{prefecto.Casa.Nombre}");
                Console.WriteLine($"Prefecto de:
{prefecto.Prefecto}");
                Console.WriteLine($"Puntos de Copa:
{prefecto.PuntoCopa}");
                Console.WriteLine();
            }
        }
        else
        {
            Console.WriteLine("No se encontraron prefectos en la
base de datos.");
        }
        break;

    default:
        Console.WriteLine("Opción inválida. Intente de nuevo.");
        break;
    }
}

```

```
static void BuscarInformacionPorModeloDeRelacion()
{
    Console.WriteLine("Ingrese el nombre del alumno: ");
    string nombreAlumno = Console.ReadLine();

    var alumno = listaAlumnos.Find(a => a.Nombre == nombreAlumno);
    if (alumno != null)
    {
        Console.WriteLine($"Alumno: {alumno.Nombre}
{alumno.Apellido}");
        Console.WriteLine($"Casa: {alumno.Casa.Nombre}");
        Console.WriteLine($"Jefe de Casa: {alumno.Casa.JefeCasa}");
    }
    else
    {
        Console.WriteLine("El alumno no se encontró en la base de
datos.");
    }
}

static void ActualizarInformacion()
{
    Console.WriteLine("Ingrese el nombre del alumno: ");
    string nombreAlumno = Console.ReadLine();
    var alumno = listaAlumnos.Find(a => a.Nombre == nombreAlumno);

    if (alumno != null)
    {
        Console.WriteLine($"Alumno: {alumno.Nombre}
{alumno.Apellido}");
        Console.WriteLine($"Casa actual: {alumno.Casa.Nombre}");
        Console.WriteLine($"Puntos actuales de la casa:
{alumno.Casa.PuntajeCopa}");
        Console.WriteLine($"Puntos actuales del alumno:
{alumno.PuntoCopa}");
    }
}
```



```
Console.WriteLine("Seleccione una opción:");
Console.WriteLine("1. Otorgar puntos");
Console.WriteLine("2. Quitar puntos");

int opcionPuntos = int.Parse(Console.ReadLine());

Console.Write("Ingrese la cantidad de puntos: ");
int cantidadPuntos = int.Parse(Console.ReadLine());

switch (opcionPuntos)
{
    case 1:
        alumno.PuntoCopa += cantidadPuntos;
        alumno.Casa.PuntajeCopa += cantidadPuntos;
        Console.WriteLine($"Se otorgaron {cantidadPuntos} puntos a {alumno.Nombre} {alumno.Apellido}.");
        break;
    case 2:
        alumno.PuntoCopa -= cantidadPuntos;
        alumno.Casa.PuntajeCopa -= cantidadPuntos;
        Console.WriteLine($"Se quitaron {cantidadPuntos} puntos a {alumno.Nombre} {alumno.Apellido}.");
        break;
    default:
        Console.WriteLine("Opción inválida.");
        return;
}

Console.WriteLine($"Nuevos puntos de la casa {alumno.Casa.Nombre}: {alumno.Casa.PuntajeCopa}");
Console.WriteLine($"Nuevos puntos de {alumno.Nombre} {alumno.Apellido}: {alumno.PuntoCopa}");
}
else
{
    Console.WriteLine("El alumno no se encontró en la base de datos.");
}
```

```

    }

    static void EliminarInformacion()
    {
        Console.WriteLine("Expulsión de Alumno");
        Console.Write("Ingrese el nombre del alumno a expulsar: ");
        string nombreAlumno = Console.ReadLine();

        var alumno = listaAlumnos.Find(a => a.Nombre == nombreAlumno);

        if (alumno != null)
        {
            var casa = listaCasas.Find(c => c.Nombre ==
alumno.Casa.Nombre);

            if (casa != null)
            {
                casa.PuntajeCopa -= alumno.PuntoCopa;

                listaAlumnos.Remove(alumno);

                Console.WriteLine($"Alumno {alumno.Nombre}
{alumno.Apellido} ha sido expulsado.");
                Console.WriteLine($"Los {alumno.PuntoCopa} puntos han sido
restados de {casa.Nombre}.");
                Console.WriteLine($"Nuevo puntaje de {casa.Nombre}:
{casa.PuntajeCopa}");
            }
            else
            {
                Console.WriteLine("Error: No se encontró la casa del
alumno.");
            }
        }
        else
        {
            Console.WriteLine("El alumno no se encontró en la base de
datos.");
        }
    }
}

```

Capturas:

```
PS C:\Users\anapa\OneDrive\Desktop\ev3bd> dotnet build
Determinando los proyectos que se van a restaurar...
Todos los proyectos están actualizados para la restauración.
ev3bd -> C:\Users\anapa\OneDrive\Desktop\ev3bd\bin\Debug\net8.0\ev3bd.dll

Compilación correcta.
    0 Advertencia(s)
    0 Errores

Tiempo transcurrido 00:00:00.90
PS C:\Users\anapa\OneDrive\Desktop\ev3bd> dotnet run
Seleccione una opción:
1. Buscar información por colección
2. Buscar información modelando relación
3. Actualizar información
4. Eliminar información
5. Salir
```

- Buscar información por colección:

```
Seleccione una opción:
1. Buscar información por colección
2. Buscar información modelando relación
3. Actualizar información
4. Eliminar información
5. Salir
1
Seleccione una opción:
1. Buscar alumno específico
2. Mostrar todos los prefectos
1
Ingrese el nombre del alumno: Luna
Alumno: Luna Lovegood
Casa: Ravenclaw
Fundador de la casa: Rowena Ravenclaw
Animal de la casa: Águila
Colores de la casa: Azul y Bronce
Jefe de la casa: Filius Flitwick
Puntaje de Copa de la casa: 180
```

```
Seleccione una opción:
1. Buscar información por colección
2. Buscar información modelando relación
3. Actualizar información
4. Eliminar información
5. Salir
1
Seleccione una opción:
1. Buscar alumno específico
2. Mostrar todos los prefectos
2
Prefectos:
Nombre: Harry Potter
Casa: Gryffindor
Prefecto de: Gryffindor
Puntos de Copa: 100

Nombre: Hermione Granger
Casa: Gryffindor
Prefecto de: Gryffindor
Puntos de Copa: 90
```

- Buscar información modelando relación:

```

Seleccione una opción:
1. Buscar información por colección
2. Buscar información modelando relación
3. Actualizar información
4. Eliminar información
5. Salir
2
Ingrese el nombre del alumno: Harry
Alumno: Harry Potter
Casa: Gryffindor
Jefe de Casa: Minerva McGonagall
Seleccione una opción:
1. Buscar información por colección
2. Buscar información modelando relación
3. Actualizar información
4. Eliminar información
5. Salir
2
Ingrese el nombre del alumno: Draco
Alumno: Draco Malfoy
Casa: Slytherin
Jefe de Casa: Severus Snape

```

- **Actualizar información:**

<pre> Seleccione una opción: 1. Buscar información por colección 2. Buscar información modelando relación 3. Actualizar información 4. Eliminar información 5. Salir 3 Ingrese el nombre del alumno: Harry Alumno: Harry Potter Casa actual: Gryffindor Puntos actuales de la casa: 190 Puntos actuales del alumno: 100 Seleccione una opción: 1. Otorgar puntos 2. Quitar puntos 1 Ingrese la cantidad de puntos: 10 Se otorgaron 10 puntos a Harry Potter. Nuevos puntos de la casa Gryffindor: 200 Nuevos puntos de Harry Potter: 110 </pre>	<pre> Seleccione una opción: 1. Buscar información por colección 2. Buscar información modelando relación 3. Actualizar información 4. Eliminar información 5. Salir 1 Seleccione una opción: 1. Buscar alumno específico 2. Mostrar todos los prefectos 1 Ingrese el nombre del alumno: Harry Alumno: Harry Potter Casa: Gryffindor Fundador de la casa: Godric Gryffindor Animal de la casa: León Colores de la casa: Rojo y Dorado Jefe de la casa: Minerva McGonagall Puntaje de Copa de la casa: 200 </pre>
---	--

- **Eliminar información:**

```
Seleccione una opción:
1. Buscar información por colección
2. Buscar información modelando relación
3. Actualizar información
4. Eliminar información
5. Salir
4
Expulsión de Alumno
Ingrese el nombre del alumno a expulsar: Hermione
Alumno Hermione Granger ha sido expulsado.
Los 90 puntos han sido restados de Gryffindor.
Nuevo puntaje de Gryffindor: 110
```

```
Seleccione una opción:
1. Buscar información por colección
2. Buscar información modelando relación
3. Actualizar información
4. Eliminar información
5. Salir
1
Seleccione una opción:
1. Buscar alumno específico
2. Mostrar todos los prefectos
2
Prefectos:
Nombre: Harry Potter
Casa: Gryffindor
Prefecto de: Gryffindor
Puntos de Copa: 110
```

- Salir:

```
Seleccione una opción:
1. Buscar información por colección
2. Buscar información modelando relación
3. Actualizar información
4. Eliminar información
5. Salir
5
PS C:\Users\anapa\OneDrive\Desktop\ev3bd> |
```