

```
using System;
```

```
namespace OrganizadordeEvento
```

```
{
```

```
    internal class Program
```

```
    {
```

```
        //verifica se a resposta é válida
```

```
        static string ChecagemOrganizacao(string resposta)
```

```
        {
```

```
            while (true)
```

```
            {
```

```
                // verifica se a resposta não é válida
```

```
                if (resposta == null || (resposta != "1" && resposta != "2" && resposta != "3" && resposta  
!= "4" && resposta != "5"))
```

```
                {
```

```
                    Console.WriteLine("Resposta inválida! Por favor, digite uma opção válida.");
```

```
                    resposta = Console.ReadLine(); // recebe nova entrada do usuário
```

```
                }
```

```
            else
```

```
            {
```

```
                break; // sai do loop quando a resposta é válida
```

```
            }
```

```
        }
```

```
        return resposta; // retorna a resposta
```

```
    }
```

```

// chama Ordenar para ordenar a lista de eventos

static void Ordenar(Evento[] lista)
{
    int N = lista.Length; // pega o tamanho da lista

    // mostra as opções de ordenação

    Console.WriteLine("\nQual o tipo de ordenação que você deseja?\n1: Ordem alfabética\n2:
Ordem De ID\n3: Ordem por preço (Menor para Maior) \n4: Ordem por preço(Maior para Menor)
\n5: Ordem por Data (Mais proximo da Data Atual para Mais distante)");

    string resposta = Console.ReadLine();

    resposta = ChecagemOrganizacao(resposta); // verifica se a resposta é válida

    // ordena a lista baseado na opção escolhida
    for (int i = N - 1; i >= 0; i--)
    {
        for (int j = 0; j < i; j++)
        {
            if (lista[j].Maior(lista[j + 1], resposta)) // compara eventos
            {
                Evento aux = lista[j]; // troca os eventos
                lista[j] = lista[j + 1];
                lista[j + 1] = aux;
            }
        }
    }
}

```

```
// imprimir os eventos na tela
```

```
static void printEventos(Evento[] lista)
{
    for (int i = 0; i < lista.Length; i++)
    {
        // imprime cada evento desta seguinte maneira
        Console.Write(
            $"ID      :{lista[i].GetID()}      | |" +
            $"Nome    :{lista[i].GetNome()}      | |" +
            $"Local   :{lista[i].GetLocal()}      | |" +
            $"Data    :{lista[i].GetData():dd/MM/yyyy}| |" +
            $"Horario :{lista[i].GetHorario()}      | |" +
            $"Preço   :R$ {lista[i].GetPreco()}      | |\n");
    }
}
```

```
//método main
```

```
static void Main(string[] args)
{
    // cria um array de eventos
    Evento[] evento = new Evento[5];

    // inicializa os eventos com seus respectivos dados
    evento[0] = new Evento(10, "Invasão pirata", "Terra do Nunca",
        DateTime.Parse("25/10/2026"), "22:25", 650);

    evento[1] = new Evento(8, "Dominação de terras", "Brazil", DateTime.Parse("29/09/2026"),
        "21:10", 1100);
}
```

```
evento[2] = new Evento(7, "Levar Madeira até Jua", "Perto de Tupí",  
DateTime.Parse("09/10/2025"), "08:07", 50);
```

```
evento[3] = new Evento(6, "Enfrentar Baianinho de Mauá", "Mauá",  
DateTime.Parse("08/05/2025"), "15:25", 0);
```

```
evento[4] = new Evento(5, "Abrir a caixa de Pandora", "Olimpo",  
DateTime.Parse("01/05/2027"), "06:00", 0);
```

```
// imprime os eventos antes de ordenar
```

```
printEventos(evento);
```

```
// ordena a lista de eventos
```

```
Ordenar(evento);
```

```
// imprime a lista após a ordenação
```

```
Console.WriteLine("\n Lista pós ordenação \n");
```

```
printEventos(evento);
```

```
}
```

```
}
```

```
}
```