# **Sistema de Eventos Corporativos 🎉**

Este projeto é um **sistema de gestão de eventos corporativos**, desenvolvido em **.NET 6** com **Entity Framework Core**.  
 Ele permite gerenciar **eventos, participantes, fornecedores e endereços**, além das relações entre eles.

## **🚀 Como rodar o projeto**

### **1. Clonar o repositório**

git clone https://github.com/seu-usuario/seu-repositorio.git

cd seu-repositorio

### **2. Compilar o projeto**

No diretório do projeto, rode:

dotnet build

### **3. Instalar as ferramentas do Entity Framework Core**

Caso ainda não tenha instalado, rode:

dotnet tool install --global dotnet-ef

Se já tiver instalado, apenas atualize:

dotnet tool update --global dotnet-ef

### **4. Criar o banco de dados**

Existem duas formas de criar o banco:

#### **Opção A - Usando Migrations**

No diretório do projeto, rode:

dotnet ef database update

Isso criará o banco de dados **EventosDB** (conforme configurado no AppDbContext).

#### **Opção B - Usando script SQL**

Se preferir, você pode gerar o script SQL da migration e executar manualmente no SQL Server:

dotnet ef migrations script -o script.sql

Depois, abra o **SQL Server Management Studio** ou outro cliente e rode o script no servidor.

## **📂 Estrutura do Projeto**

* **Core/** → Entidades e modelos do sistema
* **DATA/** → Contexto do Entity Framework (AppDbContext)
* **Migrations/** → Histórico de migrations do banco
* **UI/** → (se aplicável) Interface do usuário

## **🛠️ Tecnologias Utilizadas**

* .NET 6
* Entity Framework Core
* [SQL Server](https://www.microsoft.com/sql-server)

## **💡 Observações**

O banco está configurado para rodar em **localhost**:  
  
 optionsBuilder.UseSqlServer("Server=localhost;Database=EventosDB;Trusted\_Connection=True;TrustServerCertificate=True;");

* Se necessário, altere a connection string no AppDbContext.cs.

Na parte de “Server=localhost”, mude para o nome do SEU servidor.

Após isso, o programa deveria rodar tranquilamente.