

SÚMARIO

- 1. [Equipe](#)
- 2. [Objetivo Estratégico do Projeto](#)
- 3. [Resumo do Projeto](#)
- 4. [Imersão](#)
 - i. [Problemas que o Projeto Visa Resolver](#)
 - ii. [Pessoas que o Projeto Visa Ajudar](#)
 - iii. [Benefícios do Projeto](#)
 - iv. [Produtos Semelhantes](#)
 - v. [Soluções Encontradas](#)
 - vi. [Diferencial do Projeto](#)
- 5. [Backlog do Projeto](#)
- 6. [Construção do Projeto](#)
 - i. [Tecnologia Utilizada](#)
- 7. [Diagramas](#)
 - i. [Diagrama de Projeto](#)
 - ii. [Diagrama de Classes](#)
- 8. [Estruturas de Pastas MVC](#)

EQUIPE

Nome	Papel
Eduardo Pitanga Loureiro	Desenvolvedor
Theo Mischiatti Gomes	Desenvolvedor

OBJETIVO ESTRATÉGICO DO PROJETO

O objetivo estratégico do projeto é definido pela seguinte frase:

"Nosso objetivo é fornecer uma plataforma intuitiva e eficiente que permita a criação, organização e gestão de eventos de forma simplificada, proporcionando uma experiência otimizada para organizadores e participantes, com foco em aumentar a produtividade, reduzir custos e melhorar a interação nos eventos."

RESUMO DO PROJETO

O EventMaster é uma plataforma digital desenvolvida para facilitar a criação e a gestão de eventos, além de permitir o controle e a organização de participantes, palestrantes e organizadores. O sistema é estruturado de forma a atender diferentes tipos de usuários, sendo eles: participantes, palestrantes e organizadores, cada um com suas características e responsabilidades dentro da plataforma. Todos esses usuários possuem informações comuns, como nome, CPF e data de nascimento, mas também têm atributos específicos que os definem em suas funções.

Os participantes possuem informações adicionais, como o tipo de ingresso adquirido e o status da inscrição. Já os palestrantes têm uma biografia e uma especialidade descritas, além de estarem associados às palestras que irão ministrar durante os eventos. Por fim, os organizadores são responsáveis por gerenciar e coordenar os eventos.

Um evento dentro da plataforma é composto por várias informações, como o nome, descrição, data, horário, local, o organizador responsável. Além disso, um evento possui palestrantes envolvidos e participantes que estão inscritos. O evento precisa estar associado a um organizador e a um local previamente cadastrados no sistema, sendo que a associação de palestrantes e participantes pode ser feita posteriormente, conforme a evolução do planejamento do evento.

O local de um evento é composto por informações como nome, logradouro, número, UF (Unidade Federativa), cidade e bairro, permitindo que os eventos sejam mapeados para um espaço específico. Para a criação de um evento, é obrigatório associar um organizador e um local que já estejam cadastrados no sistema.

Além disso, o sistema oferece a capacidade de inserir e remover tanto participantes quanto palestrantes de um evento. Isso é necessário porque ambos podem, eventualmente, cancelar sua presença.

A plataforma também permite apagar e atualizar qualquer uma das entidades cadastradas no sistema, sejam eventos, pessoas (participantes, palestrantes ou organizadores) ou locais. No caso de exclusão de um evento, é preciso desassociar todos os participantes, palestrantes e organizadores vinculados ao evento. A mesma lógica se aplica para a exclusão de uma pessoa ou um local. Essa regra não se aplica ao organizador, que não pode ser excluído se estiver vinculado a um evento ativo. Isso garante que os eventos sempre tenham um organizador designado.

IMERSÃO

A Imersão tem como objetivo entender o problema e, assim, determinar o Produto Mínimo e Viável (MVP, do inglês, Minimum Viable Product). Essa atividade ajuda no desenvolvimento do planejamento do projeto. Ao fim dessa atividade, a equipe sabe as funcionalidades mínimas que devem ser produzidas para atender as necessidades do público alvo do produto

PROBLEMAS QUE O PROJETO VISA RESOLVER

O projeto visa resolver principalmente o problema encontrado para se gerenciar eventos, principalmente eventos em que é necessário ter um controle de participantes inscritos, devido a limitação de no número de vagas disponíveis. Gerenciar eventos pode ser uma tarefa desafiadora, mas que é facilitada por meio da utilização do sistema proposto.

PESSOAS QUE O PROJETO VISA AJUDAR

O projeto tem como objetivo atender os seguintes tipos de pessoa:

1. Organizadores, que são as pessoas que irão organizar eventos e associar palestrantes;
2. Palestrantes, que poderão visualizar eventos que irão ministrar;
3. Participantes, que irão escolher eventos para se inscrever.

BENEFÍCIOS DO PROJETO

- Facilidade de gerenciamento de eventos
- Facilidade para inscrição de pessoas
- Facilidade para criação de locais
- Facilidade para criação de eventos

PRODUTOS SEMELHANTES

- [Gerenciador de Eventos da Meta](#)

SOLUÇÕES ENCONTRADAS

Para o desenvolvimento do nosso projeto, optamos por utilizar a linguagem C#. Essa escolha foi motivada por dois fatores principais: primeiro, a oportunidade de aprendizado, já que se trata de uma linguagem nova para os integrantes do grupo, permitindo-nos explorar e aprofundar nossos conhecimentos. Segundo, a facilidade oferecida pela linguagem para a criação de entidades em um banco de dados InMemory, característica que se mostrou bastante útil no contexto do projeto. Além disso, o C# foi abordado em sala de aula, com foco em aplicações web, o que reforçou sua relevância para o projeto.

Adotamos o modelo de desenvolvimento MVC, que promove uma estrutura modular e segue boas práticas de programação. Esse modelo permite a criação de:

- Model, incluindo DTOs e entidades;
- View, responsável pela interface com o usuário (nesse caso não implementado pela falta de interface gráfica);
- Controller, que gerencia as interações entre a interface e a lógica de negócio.

Para garantir uma arquitetura bem estruturada, implementamos também um repositório com interfaces e suas respectivas implementações.

Entre os pacotes instalados para o projeto, destacam-se o Microsoft.EntityFrameworkCore e o Microsoft.EntityFrameworkCore.InMemory, que foram fundamentais para a construção e manipulação do banco de dados utilizado.

DIFERENCIAL DO PROJETO

Nosso diferencial é promover um projeto que seja funcional e simples ao mesmo tempo, que possa atender a alguém que não tenha um conhecimento tecnológico muito aprofundado.

Apesar de simples, o projeto oferece um ótimo gerenciamento de eventos.

BACKLOG DO PROJETO

ID	História do Usuário	MoSCoW	Importância	RoadMap
1	Como participante, quero me inscrever em eventos para participar de palestras e workshops.	Must Have	Alta	Sprint 1
2	Como organizador, quero gerenciar a lista de palestrantes para garantir que as informações estejam atualizadas.	Should Have	Média	Sprint 2
3	Como administrador, quero remover participantes que	Must Have	Alta	Sprint 1

ID	História do Usuário	MoSCoW	Importância	RoadMap
	cancelaram a inscrição para manter a lista atualizada.			
4	Como palestrante, quero editar minha biografia e especialidades para informar melhor os participantes.	Should Have	Baixa	Sprint 3
5	Como organizador, quero visualizar relatórios de eventos para avaliar o sucesso e engajamento.	Must Have	Alta	Sprint 2

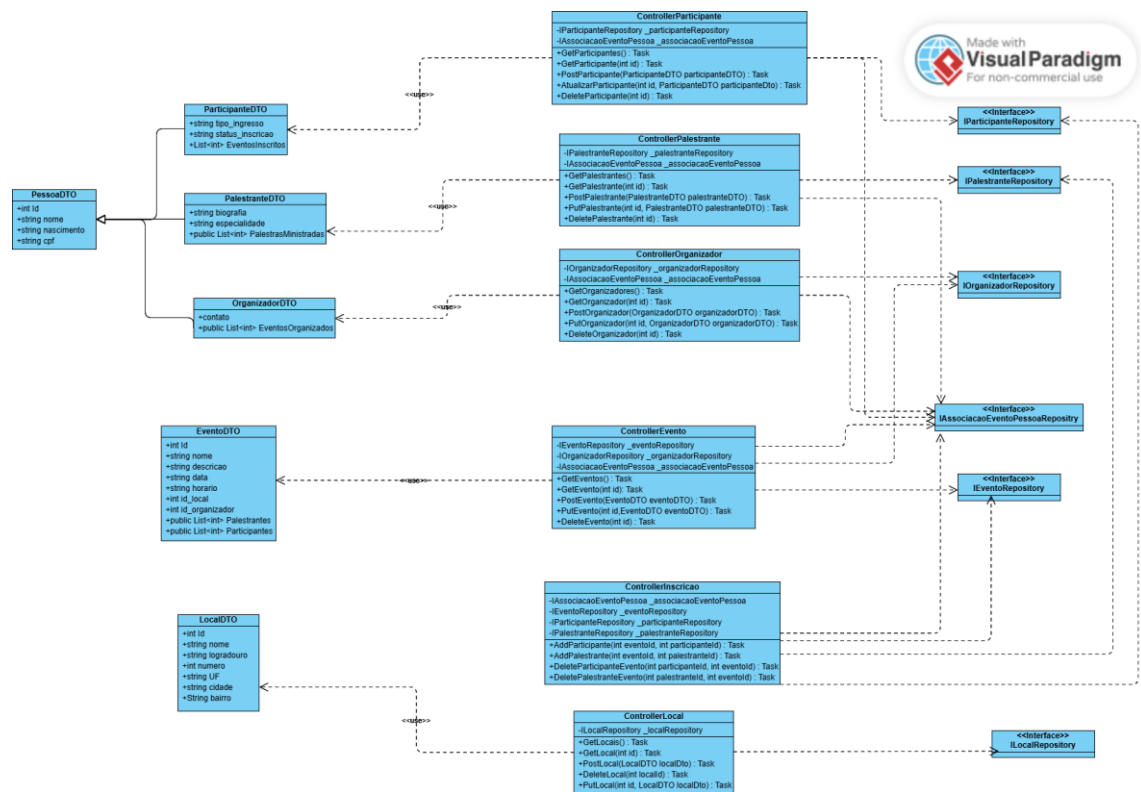
CONSTRUÇÃO DO PROJETO

TECNOLOGIA UTILIZADA

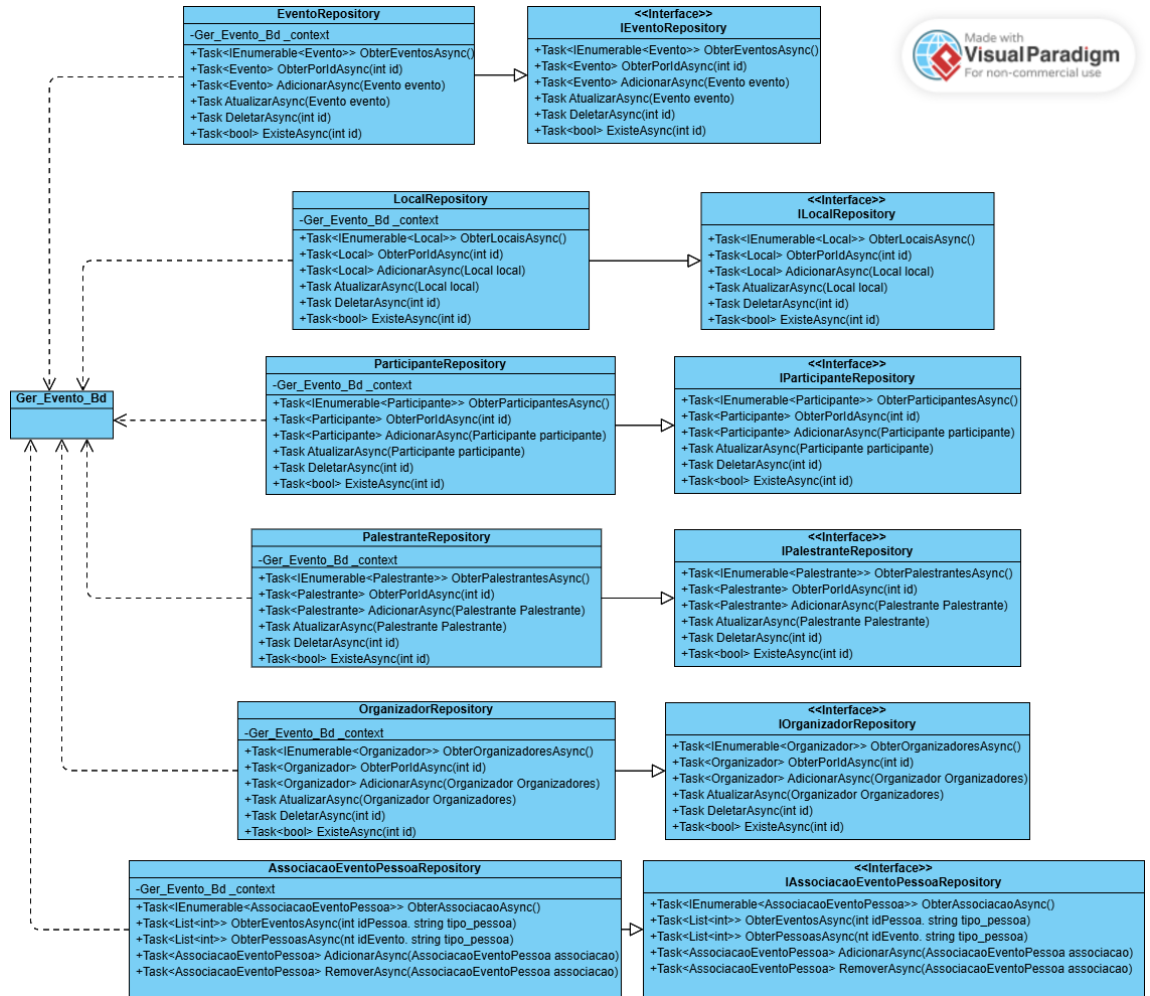
Para nosso projeto, usamos a linguagem C# da Microsoft, por ser uma oportunidade de aprendermos uma nova linguagem e por ser semelhante ao Java, que já conhecíamos. Além disso, a linguagem C# trouxe algumas facilidades como a criação de um banco InMemory e da criação de Tabelas por meio de Classes.

DIAGRAMA DE PROJETO

- CONTROLLER E DTO



• ENTITY



- REPOSITORY

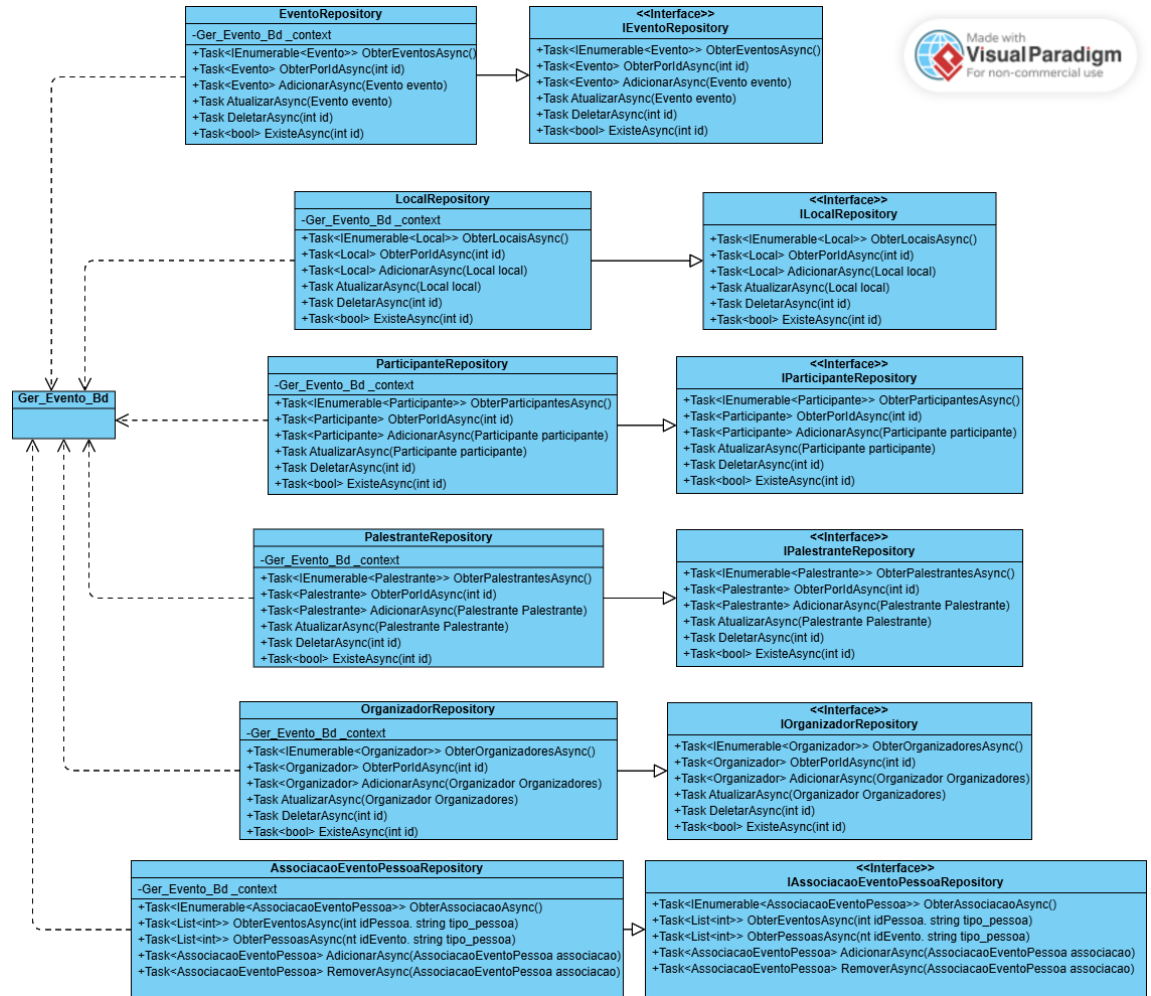


DIAGRAMA DE CLASSES

![Diagrama](Diagrama de Classes.png)

ESTRUTURAS DE PASTAS MVC

eventos_ger

|

|—Application

| Program.cs

|

|—Controller

| ControllerEvento.cs

| ControllerInscricao.cs

| ControllerLocal.cs


```
| ControllerOrganizador.cs
| ControllerPalestrante.cs
| ControllerParticipante.cs
|
| └─ Model
|   | └─ DTOs
|   |   | eventoDTO.cs
|   |   | localDTO.cs
|   |   | organizadorDTO.cs
|   |   | palestranteDTO.cs
|   |   | participanteDTO.cs
|   |   | pessoaDTO.cs
|   |
|   | └─ Entity
|   |   | AssociacaoEventoPessoa.cs
|   |   | Evento.cs
|   |   | Ger_Evento_Bd.cs
|   |   | Local.cs
|   |   | Organizador.cs
|   |   | Palestrante.cs
|   |   | Participante.cs
|   |   | Pessoa.cs
|   |
|   | └─ Repository
|   |   | └─ Implementations
|   |   |   | AssociacaoEventoPessoaRepository.cs
|   |   |   | EventoRepository.cs
|   |   |   | LocalRepository.cs
```

| OrganizadorRepository.cs

| PalestranteRepository.cs

| ParticipanteRepository.cs

|

└─ Interfaces

IAssociacaoEventoPessoa.cs

IEventoRepository.cs

ILocalRepository.cs

IOrganizadorRepository.cs

IPalestranteRepository.cs

IParticipanteRepository.cs