



Título do trabalho

AVALIAÇÃO N1 - POO

Nome do professor:

Carlos Veríssimo

Nome do Aluno:

Pedro Paulo Da Silveira Chaves

Nome da disciplina:

PROGRAMAÇÃO A ORIENTADA A OBJETOS

componentes da equipe com a descrição dos papéis/responsabilidades:

Pedro Paulo da Silveira Chaves: desenvolvedor e programador(Responsabilidade: fazer tudo)

Requisitos Funcionais:

1. Registro de Eventos:

- Permitir o cadastro de eventos, incluindo nome, data, hora, local e descrição.

2. Registro de Participantes:

- Possibilitar o registro de participantes, com informações pessoais, como nome, e-mail e número de telefone.

3. Gerenciamento de Inscrições:

- Permitir que os participantes se inscrevam em eventos.
- Verificar a disponibilidade de vagas em tempo real.
- Gerar confirmações de inscrição.

4. Programação de Eventos:

- Criar uma programação de eventos com horários e locais.
- Atualizar automaticamente os horários com base em mudanças.

5. Pagamento e Faturamento:

- Integrar métodos de pagamento para taxas de inscrição.
- Gerar faturas e recibos para participantes.

6. Comunicação:

- Enviar notificações por e-mail ou SMS para participantes, palestrantes e organizadores.
- Facilitar a comunicação entre os participantes.

7. Gerenciamento de Palestrantes:

- Cadastrar palestrantes e atribuir a eles eventos específicos.
- Coletar informações sobre os palestrantes, como biografias e fotos.

8. Avaliações e Feedback:

- Coletar avaliações e feedback dos participantes após os eventos.

- Gerar relatórios com base nas avaliações.

Requisitos Não Funcionais:

1. Desempenho:

- O sistema deve ser capaz de lidar com um grande número de participantes e eventos simultaneamente.
- Tempo de resposta rápido para garantir uma experiência do usuário eficiente.

2. Segurança:

- Proteção de dados pessoais e informações financeiras dos participantes.
- Autenticação e autorização adequadas para garantir que apenas pessoas autorizadas tenham acesso.

3. Confiabilidade:

- Alta disponibilidade para que os eventos ocorram sem interrupções.
- Recuperação de dados em caso de falhas.

4. Usabilidade:

- Interface amigável e de fácil navegação para organizadores e participantes.
- Suporte a dispositivos móveis e acessibilidade.

5. Escalabilidade:

- Capacidade de crescer e lidar com um aumento no número de eventos e participantes.

6. Integração:

- Integração com sistemas de pagamento, como PayPal ou cartões de crédito.
- Integração com serviços de e-mail e mídias sociais para comunicação.

7. Manutenção:

- Facilidade de atualização e manutenção do sistema.

8. Regulamentações:

- Conformidade com regulamentos de proteção de dados, como o GDPR.

Cenário:

O sistema de gerenciamento de eventos é uma aplicação projetada para facilitar a organização e administração de eventos, tais como conferências, festas, workshops e outros. Ele permite aos organizadores criar eventos, gerenciar participantes, controlar o cronograma e fornecer informações importantes aos envolvidos.

Planejamento:

18/8 a 26/8: Planejamento

1/9 a 29/9: Modelagem

6/11 a 17/11: Implementação

17/11 a 1/12: conclusão

O problema a ser resolvido:

A situação-problema é a necessidade de automatizar a gestão de eventos para melhorar o processo de organização e melhorar a experiência dos participantes.

Solução de Software:

1. **Cadastro de Eventos:** Permite a criação e configuração de eventos, incluindo detalhes, data e local.
2. **Gerenciamento de Participantes:** Oferece a capacidade de registrar participantes, gerenciar inscrições e informações relacionadas.
3. **Cronograma do Evento:** Facilita a criação e visualização de um cronograma detalhado do evento.
4. **Emissão de Ingressos:** Possibilita a geração e distribuição de ingressos para os participantes.
5. **Notificações e Comunicações:** Envia notificações e mensagens aos participantes sobre atualizações e informações importantes.
6. **Relatórios de Desempenho:** Gera relatórios para avaliar o desempenho e sucesso dos eventos.
7. **Autenticação e Controle de Acesso:** Fornece autenticação segura para os administradores e controle de acesso aos dados.
8. **Integração com Meios de Pagamento:** Permite a integração com sistemas de pagamento para a venda de ingressos.

os pontos do código que implementam o encapsulamento:

- **Classe Evento:** A classe Evento encapsula os dados relacionados a um evento, como nome, data e organizador. Os atributos da classe são privados, e os métodos públicos (getters e setters) são usados para acessar e modificar esses

atributos. O método **getDataFormatada** é usado para obter a data formatada do evento.

- **Métodos Privados:** Os métodos privados como **criarEvento**, **listarEventos**, **editarEvento** e **removerEvento** são encapsulados para realizar a lógica específica relacionada à criação, listagem, edição e remoção de eventos. Eles ajudam a manter o código organizado e modular.
- **Uso de Scanner:** O Scanner é utilizado para interagir com o usuário e obter entradas. O código encapsula a leitura e limpeza do buffer de entrada com **scanner.nextLine()** após a leitura de inteiros.
- **DateFormat:** A formatação de datas é encapsulada no objeto **SimpleDateFormat** para garantir que as datas sejam tratadas corretamente.
- **Lista de Eventos:** A lista de eventos é uma coleção que encapsula múltiplos objetos do tipo **Evento**.
- **Menu de Opções:** A interação com o usuário é encapsulada no menu de opções, onde o usuário pode escolher entre criar, listar, editar, remover eventos ou sair do programa.
- **Validação de Entradas:** O código encapsula a validação das entradas do usuário, como a verificação de formato de data válido.

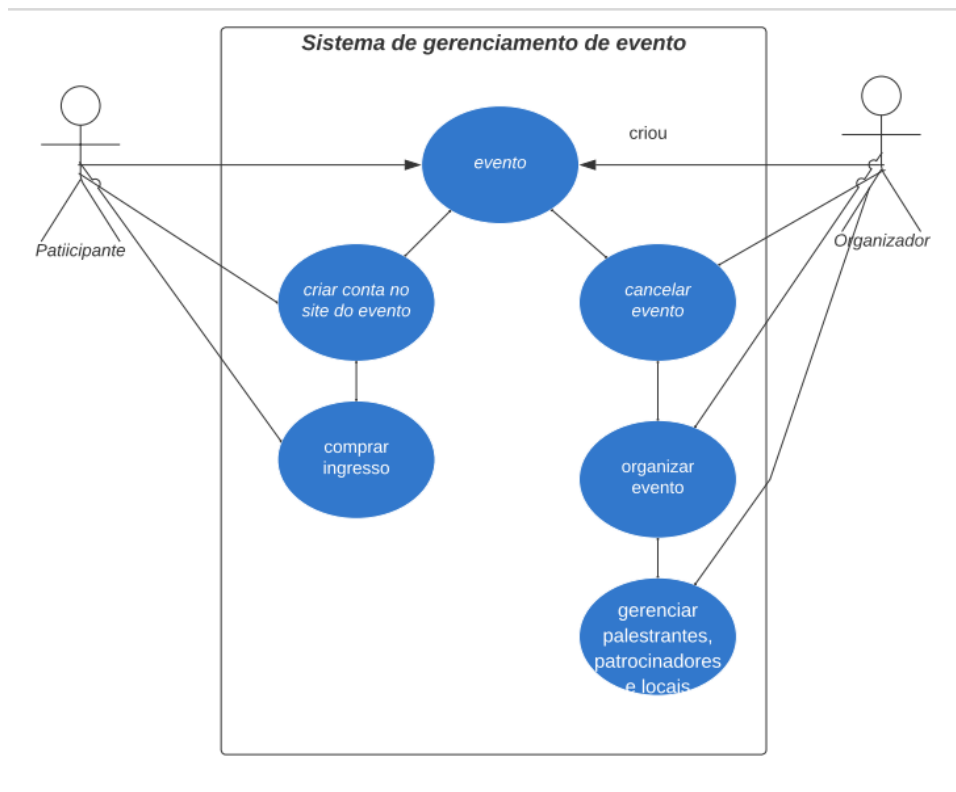
Pontos do código que atendem ao baixo acoplamento das classes:

- **Métodos Privados:** Os métodos privados no **SistemaGerenciamentoEventos** (por exemplo, **criarEvento**, **listarEventos**, **editarEvento**, **removerEvento**) realizam operações específicas em eventos, mas são encapsulados. Eles não expõem detalhes de implementação para outras partes do código. Isso ajuda a manter uma interface mais simples e reduz o acoplamento com outras partes do programa.
- **Classe Evento Independente:** A classe **Evento** é independente e encapsula todas as informações relacionadas a um evento (nome, data, organizador). Isso significa que outras partes do programa não precisam conhecer os detalhes internos de como um evento é representado. O baixo acoplamento é alcançado porque outras partes do código não dependem de detalhes de implementação da classe **Evento**.
- **Uso de Parâmetros:** Os métodos que manipulam eventos passam os objetos **Evento** como parâmetros, em vez de depender de variáveis globais ou estados compartilhados. Isso reduz o acoplamento, uma vez que os métodos não precisam saber de onde vêm os objetos **Evento**.
- **Mensagens de Erro e Exceções:** O código lida com exceções e mensagens de erro de forma apropriada, como quando ocorre um erro ao analisar a data. Isso permite que as partes do código que chamam esses métodos

saibam que algo deu errado, mas não precisam saber os detalhes do tratamento do erro.

- **Menu de Opções:** O menu de opções no **SistemaGerenciamentoEventos** permite que o usuário escolha o que deseja fazer, tornando o programa interativo, mas também desacoplado das operações específicas realizadas em eventos. Cada opção do menu chama métodos especializados para realizar tarefas, mantendo o acoplamento entre as partes do código baixo.
- **Uso de Interfaces Padrão:** O código utiliza interfaces padrão, como **List** e **SimpleDateFormat**, que são bem definidas e não precisam de alterações no código do usuário. Isso promove o baixo acoplamento, uma vez que as implementações concretas dessas interfaces podem ser trocadas sem afetar o código do usuário.

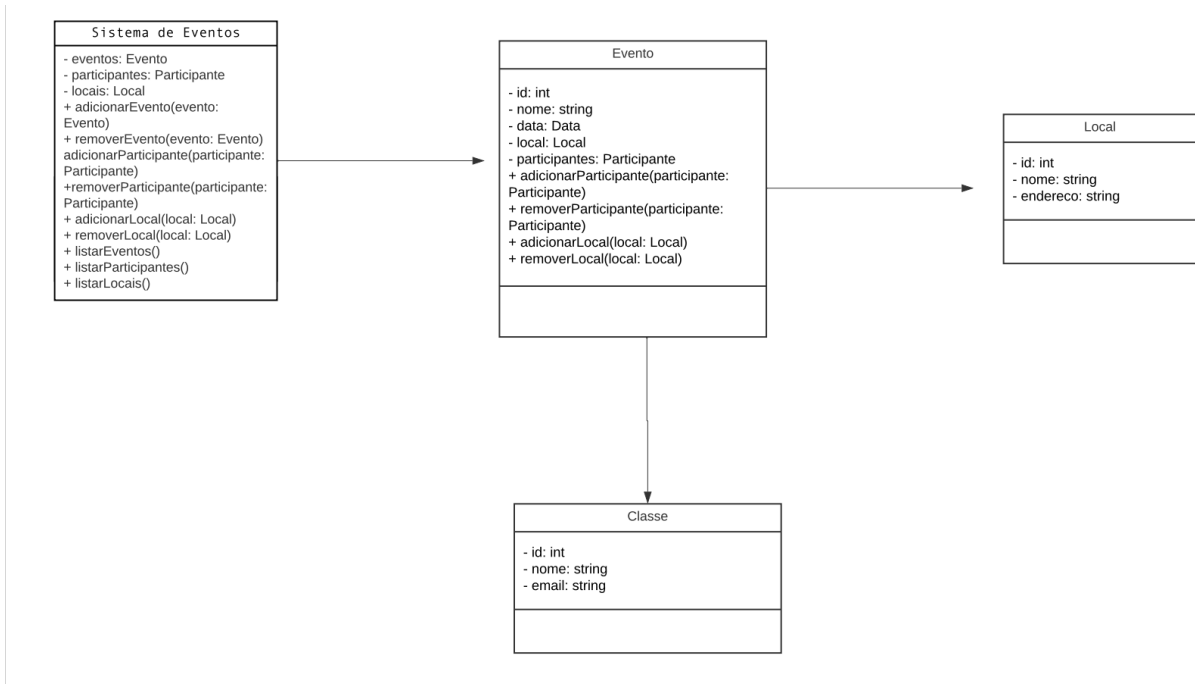
Diagrama de caso de uso:



Atores:

- **Organizador de Eventos:** Este ator é responsável por criar, modificar e gerenciar eventos. Eles também podem gerenciar palestrantes, patrocinadores e locais.
- **Participante:** Os participantes podem se inscrever, comprar ingressos, visualizar eventos e fornecer feedback.

Diagrama de classes:



Conclusão:

Em suma, o sistema de gerenciamento de eventos é uma solução eficaz para simplificar a organização de eventos, melhorar a experiência dos participantes e aumentar a eficiência na administração. Ele oferece ferramentas para criar eventos, gerenciar participantes, configurar cronogramas, emitir ingressos, enviar notificações, gerar relatórios, garantir autenticação segura e integrar sistemas de pagamento. A implementação desse sistema proporciona maior eficiência, satisfação dos participantes e sucesso global dos eventos.