

## Padrões do Sistema de Gestão

Gabriel Alves dos Santos

Link do projeto: [https://github.com/GabrielAlSant/Gestao\\_De\\_Tarefas](https://github.com/GabrielAlSant/Gestao_De_Tarefas)

### Padrão Observer

O Observer é um padrão de design que permite que um objeto (sujeito) notifique automaticamente um grupo de observadores sobre mudanças de estado. Esse padrão é útil para manter diferentes partes do sistema atualizadas quando ocorre um evento específico.

No meu projeto o padrão observer garante que todos os usuários que comentaram, incluindo o dono, sejam avisados caso tenham algum novo comentário.

```
1  const { tarefa, comentario } = require('../models');
2  const { getIO } = require('../bin/socket');
3
4  class Observer {
5    constructor(tarefaId, newCommentUserId) {
6      this.tarefaId = tarefaId;
7      this.newCommentUserId = newCommentUserId;
8    }
9    async notifyUsers() {
10     const io = getIO();
11     try {
12       const tarefaDados = await tarefa.findByPk(this.tarefaId);
13       if (!tarefaDados) throw new Error('Tarefa não encontrada.');
```

A classe acima é responsável por avisar os usuários, tendo uma instância criada a cada tarefa gerada, é chamada toda vez que um novo comentário é adicionado, tornando os usuários sempre informados.

## Padrão Event Sourcing

O **Event Sourcing** é um padrão que armazena todas as mudanças de estado de um sistema como uma sequência de eventos. Em vez de armazenar apenas o estado final de um dado no banco de dados, ele registra cada alteração, permitindo reconstruir o histórico completo de mudanças.

No meu projeto foi incluído devido a necessidade de registrar os eventos referente aos status de uma tarefa, como ela está progredindo ou regredindo, incluindo a data de alteração/inclusão e seu status antigo.

```
class EventSourcing {
  constructor(logModel) {
    this.logModel = logModel;
  }

  async createEvent(eventData) {
    try {
      await this.logModel.create(eventData);
    } catch (error) {
      console.error('Erro ao registrar evento:', error);
      throw new Error('Falha ao registrar o evento');
    }
  }
}

module.exports = { EventSourcing };
```

A classe acima é responsável por criar um registro na tabela log, isso torna o site visualmente informativo referente às datas e a visualização de como a tarefa progrediu ao decorrer do tempo.