

Resenha do Artigo: “The Big Ball of Mud”

Flávio de Souza Ferreira Junior

16/08/2025

Acabei de ler um artigo de 1997 chamado “The Big Ball of Mud” (Bola de Lama Gigante), de Brian Foote e Joseph Yoder, e nossa, parece que foi escrito ontem. Para quem, como eu, está começando na carreira de desenvolvimento, essa leitura foi quase um alívio.

O artigo fala sobre o tipo de arquitetura de software que a gente mais encontra por aí, mas que ninguém fala nos cursos ou nos livros: o famoso “código macarrão” ou, como eles chamam, a “Grande Bola de Lama”. É basicamente aquele sistema que a gente pega para trabalhar que é uma zona, todo bagunçado, onde tudo parece estar conectado com tudo de forma caótica e sem um pinga de planejamento. A primeira reação é pensar “quem foi o maluco que fez isso?”, mas o artigo mostra que a realidade é bem mais complexa.

O mais legal é que os autores não tratam isso como se a gente fosse programador ruim por fazer isso. Pelo contrário, eles defendem que, se essa é a arquitetura mais popular, alguma coisa ela deve estar fazendo certo. E aí eles listam os motivos que levam a gente a criar essas “bolas de lama”, e eu me identifiquei com vários:

- **Pressão do tempo e custo:** Sabe quando o chefe chega e fala “precisamos disso pra ontem”? Pois é, nessas horas, a arquitetura bonita e pensada fica para depois. O importante é entregar e fazer funcionar.
- **Falta de experiência:** Às vezes, a gente simplesmente não conhece o problema o suficiente para criar uma arquitetura genial de primeira. A gente vai aprendendo enquanto faz, e o sistema vai crescendo meio torto.
- **Mudanças constantes:** O cliente pede uma coisa, depois muda de ideia, aí surge um requisito novo que ninguém esperava. Manter a arquitetura intacta com tudo isso acontecendo é quase impossível.

Isso me lembrou muito uma experiência que tive no meu estágio. Peguei um sistema onde a tarefa parecia simples: “só adiciona um campo novo nesse relatório”. Moleza, pensei. Fui lá, adicionei o campo na tela e na geração do PDF. Quando subi a alteração, o módulo de autenticação de usuários parou de funcionar. Eu entrei em pânico. Não

fazia o menor sentido uma coisa quebrar a outra. Depois de quase 3 dias quebrando a cabeça sozinho, eu finalmente descobri que uma variável de configuração do relatório, que era global, era reutilizada no fluxo de login por algum motivo histórico bizarro. Quando li a descrição no artigo de que, numa “bola de lama”, “informação é compartilhada promiscuamente entre elementos distantes do sistema”, a ficha caiu na hora. Ali eu entendi que o problema não era eu, era a estrutura do sistema que tinha se tornado um campo minado.

Uma parte que me pegou muito no texto foi a do **“Throwaway Code” (Código Descartável)**. É aquela famosa “gambiarra” que a gente faz só para resolver um problema rápido ou para uma demonstração, prometendo que depois vai arrumar. O problema é que esse código provisório acaba funcionando “bem o suficiente” e nunca mais é jogado fora, se tornando a base para uma futura bola de lama.

Outro ponto que faz muito sentido é o **“Piecemeal Growth” (Crescimento Gradual)**. O sistema vai crescendo de forma descontrolada, uma coisinha aqui, outra ali, e quando a gente vê, a estrutura original já se perdeu completamente, levando à erosão da arquitetura.

No fim, o artigo não dá uma fórmula mágica para consertar tudo, mas me deu um alívio por mostrar que esses problemas são normais e fazem parte do desenvolvimento de software no mundo real. Eles até falam de algumas estratégias, como “varrer a sujeira para debaixo do tapete” (isolar a parte bagunçada criando uma interface mais limpa na frente dela) ou, em casos extremos, a “reconstrução” (jogar tudo fora e começar de novo).

Recomendo demais a leitura, principalmente para quem está começando e às vezes sente meio perdido na bagunça do dia a dia. É bom saber que não estamos sozinhos e que até a “bagunça” tem seus motivos para existir. Ajuda a gente a ter um olhar mais crítico e, quem sabe, a tentar fazer um pouco melhor no próximo projeto, mesmo com toda a pressão.