

**FACULDADE DE TECNOLOGIA SENAC FLORIANÓPOLIS**  
**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E**  
**DESENVOLVIMENTO DE SISTEMAS**

**DISCIPLINA: Arquitetura de Sistemas de Software**  
**e Gestão da Informação**

Aluno 1: **Guilherme Valmir de Andrade.**

**Cada dupla deverá desenvolver o modelo lógico de banco de dados definido em sala de aula.**

Avaliação de Ferramenta Case		Notas	
Heurística	Comentários	Lucid Chart	Draw.IO
1. Visibilidade do estado do sistema	O Lucid Chart é mais intuitivo.	X	
2. Equivalência entre o sistema e o mundo real	Ambos têm equivalência zero com SGBDs com os quais trabalhamos, mas o Lucid Chart dá uma ideia de que você está trabalhando num SGBD, mesmo que na prática não seja isso.	X	
3. Liberdade e controle do usuário	Por ter um sistema mais intuitivo, o Lucid Chart te dá mais liberdade pra trabalhar.	X	
4. Consistência e padrões	Como dito anteriormente, o Lucid Chart lembra um SGBD, o que dá a ele uma cara mais profissional.	X	
5. Prevenção de erro	Ambos não avisam nada em relação a erro. São “paints” com expressões UML prontas.	X	
6. Reconhecer ao invés de relembrar	O Lucid Chart tem um sistema mais intuitivo. É mais fácil de reaprender a usar. O Draw IO não é tão trivial.	X	
7. Flexibilidade e eficiência de uso	Por ser praticamente um Paint, o Draw IO acaba te dando total liberdade pra colocar na tela o que quiser, mesmo que isso te leve a um processo errado.		X
8. Estética e design minimalista	Por ter um sistema mais intuitivo, o Lucid Chart é melhor nesse quesito.	X	
9. Auxiliar usuários a reconhecer, diagnosticar e recuperar ações erradas	Nenhum dos dois programas têm essa funcionalidade.	X	X
10. Ajuda e documentação	Ambos têm um padrão semelhante de documentação e uma quantidade de material parecido quando se busca sobre eles na internet.	X	X
Total de Pontos		7	1
Média Final		8	

Fonte: Adaptado do livro *Usability Engineering* de Jakob Nielsen (1993).

### **Considerações Finais Case 1**

**O Lucid Chart, assim como o seu concorrente, não passa de um Paint com algumas expressões prontas. O lado positivo para esse tipo de programa é poder fazer tabelas UML na fase de construção do banco de dados de um programa. As tabelas desse tipo de programa também servem para mapeamento de processos de qualquer tipo. Mas a proposta desses programas é bem diferente em relação a um SGBD. Quando o desenvolvedor estiver construindo o banco de dados de fato, ele pode no máximo se basear em uma tabela gerada por esses programas, mas a nível de testes eles são irrelevantes.**

### **Considerações Finais Case 2**

**Idem ao primeiro comentário, com exceção que o Draw IO tem um sistema menos trivial e menos controle quando se está construindo tabelas.**