

POO 2

Profa. Fabíola S. F. Pereira

fabiola.pereira@ufu.br

www.facom.ufu.br/~fabiolapereira

Ficha técnica da disciplina

- Profa. Dra. Fabíola S. F. Pereira
- Contato: fabiola.pereira@ufu.br
- BSI-MC - 4o período
- Horários das aulas:
 - Seg: 13:10 às 14:50
 - Ter: 13:10 às 14:50
- Horário de atendimento:
 - Presencial: segunda-feira, 12:00 às 13:00 (agendar com a professora)
 - Online: via MS Teams ou email

Sejam muito bem-vindos(as)!

- Profa Fabíola Souza Fernandes Pereira
- Linha de pesquisa PPGCO: Ciência de Dados
 - Temporal Social Networks
 - Evolving Networks
 - Opinion and Preference Mining
 - Network-centric recommendation
 - Natural Language Processing Techniques
 - Business Applications of AI

<https://www.facom.ufu.br/~fabiolapereira>

Ficha técnica da disciplina

- Team @ MS Teams: **POO2 2024/1**

bit.ly/poo220241



- **Todas as atualizações serão feitas diretamente no site da disciplina**

Avaliação

- **Prova P1 (20 pontos)**, no dia **17-09**.
- **Prova P2 (25 pontos)**, no dia **29-10**.
- **Trabalho final (35 pontos)**, no dia **11-11**.
- **Exercícios práticos (20 pontos)**, a serem entregues individualmente, com prazo de entrega máximo de 7 dias a contar da data de disponibilização da atividade. O valor de cada atividade será distribuído uniformemente em relação ao número de atividades aplicadas, totalizando 20 pontos.
- **Atividade avaliativa substitutiva:** prova de substituição da P1 ou P2, no dia **18-11**

Controle de Frequência

- A presença será cobrada através de “chamadas”
- Alunos com mais de 25% de faltas serão automaticamente reprovados por falta, independente da nota final

Recuperação de aprendizagem

- Prova sub
 - Discente com nota final < 60 E frequência $\geq 75\%$
 - Discente escolhe substituir a nota da P1 ou P2
- Data: 18-11

Bibliografia básica

DEITEL, H. M. Java: como programar, 8. ed. São Paulo: Prentice Hall, 2010.

FREEMAN, Eric. FREEMAN, Elisabeth. Use a cabeça! padrões de projeto. Rio de Janeiro: Atlas Books, 2005.

JACOBSON, Ivar. Aspect-oriented software development with use cases. Upper Saddle River: Addison-Wesley, 2004.

Bibliografia complementar

- GAMMA, Erich et al. Design Patterns. Upper Saddle River: Addison Wesley, 1995.
- LADDAD, Ramnivas. Aspect in action: practical aspect oriented programming. Greenwich Mannign, 2003.
- KRUCHTEN, Philippe. The rational unified process: an introduction. 3rd ed. Boston: Addison Wesley, 2003.
- KRUCHTEN, Philippe. The rational unified process made easy: practitioner's guide to the RUP. Boston: Addison Wesley, 2003.
- LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos. Porto Alegre: Bookman, 2000.
- SOMMERVILLE, I. Engenharia de software. 8th ed. Harlow: Addison Wesley, 2007.
- FOWLER, Martin. Refactoring: Improving the design of existing code. Addison Wesley, 2000.
- MENDES, Antonio. Arquitetura de Software, Ed. Campus, 2002.
- SZYPERSKI, C. Component Software: Beyond Object Oriented Programming. Addison Wesley, 1998.
- LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos. Porto Alegre: Bookman, 2001

Bibliografia complementar

- Marco Tulio Valente. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade, Editora: Independente, 2020.

<https://engsoftmoderna.info/>



Aulas práticas

- Lab8
- Aguarde a professora avisar que a aula será no laboratório
- **Se nada for avisado, aula em SALA**

Dúvidas??

Envolva-se em atividades de pesquisa!

- Oportunidades de Iniciação Científica - falar com a professora



CENTERIS

Home

Conference

Submissions

Registration

Program

CENTERIS 2024

The stunning views over the ocean, close proximity to the city's centre, and its unique prize-awarded architecture stands out as one of its main features

1

2

3

4











5

6

7

8

Envolva-se na MARATONA DE PROGRAMAÇÃO

→   maratona.sbc.org.br/index.html         Up



Maratona SBC de Programação

[Sobre](#) ▾

[Participe!](#) ▾

[Histórico](#)

[Contato](#)



Entrando no clima de POO 2

Na minha máquina funciona

Só eu entendo meu código

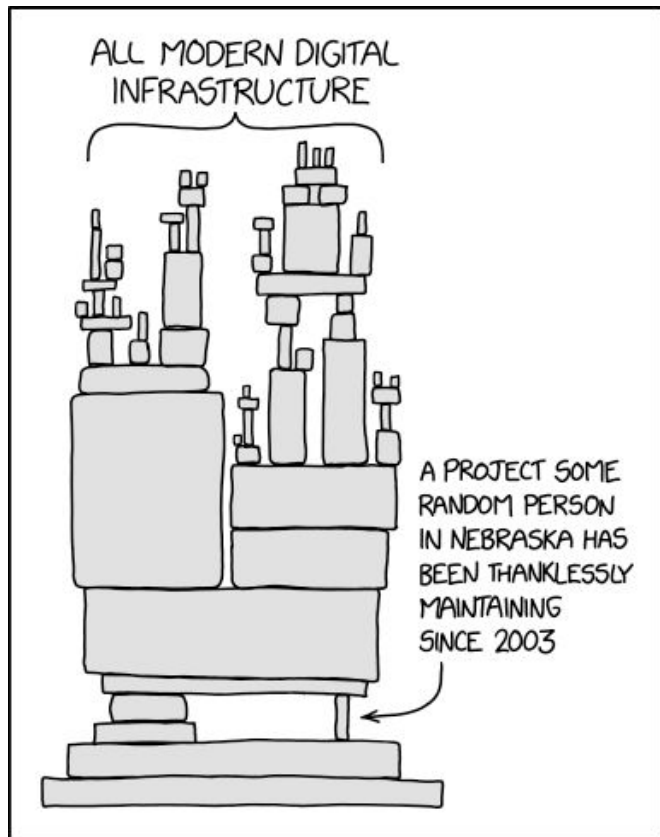
Eu que fiz há 1 mês atrás e não estou entendendo nada. Esqueci tudo.

Vou colocar “uns prints” pra entender!

O código está bem escrito?

```
float sin_or_cos(double x, int op) {  
    if (op == 1)  
        "calcula e retorna seno de x"  
    else  
        "calcula e retorna cosseno de x"  
}
```

Riscos de Acoplamento



Principalmente, quando envolve dependências para bibliotecas de terceiros

A palavra-chave é **REUSO**

Difícilmente, na indústria, você irá começar um código “**do zero**”

Difícilmente, na indústria, você irá codificar **sozinho** um projeto inteiro

Difícilmente, na indústria, você irá **ignorar** a figura do arquiteto de software

Expectativas??

Programa da Disciplina

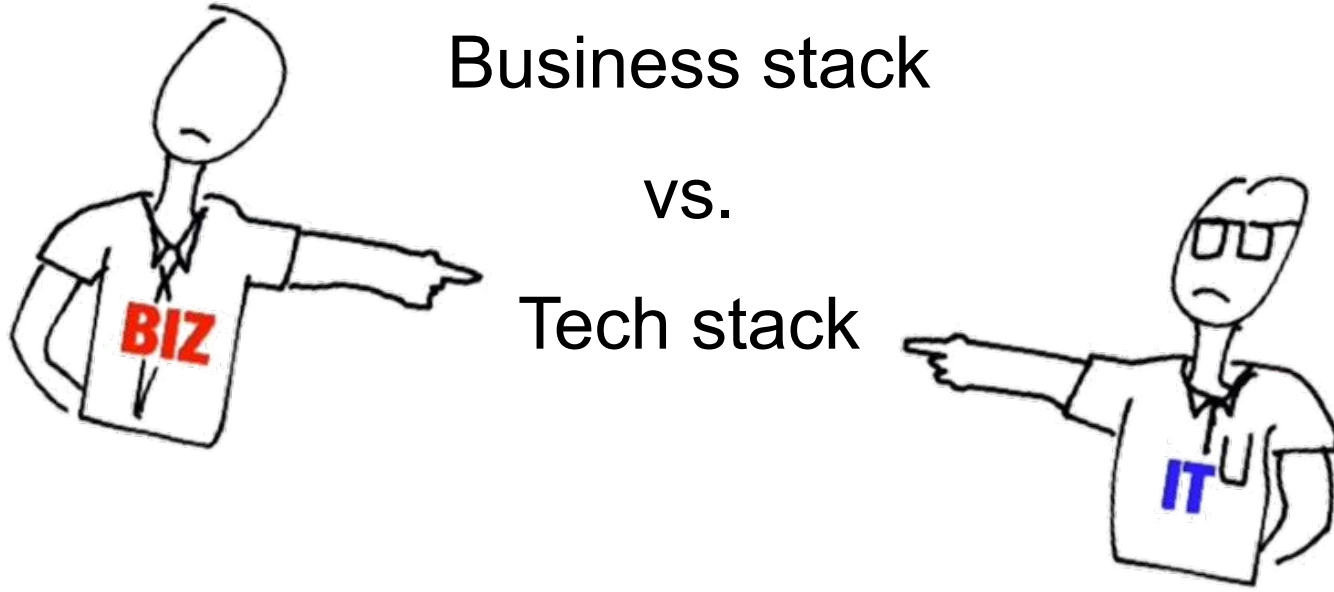
1. Princípios de Projeto

2. Padrões de Projeto

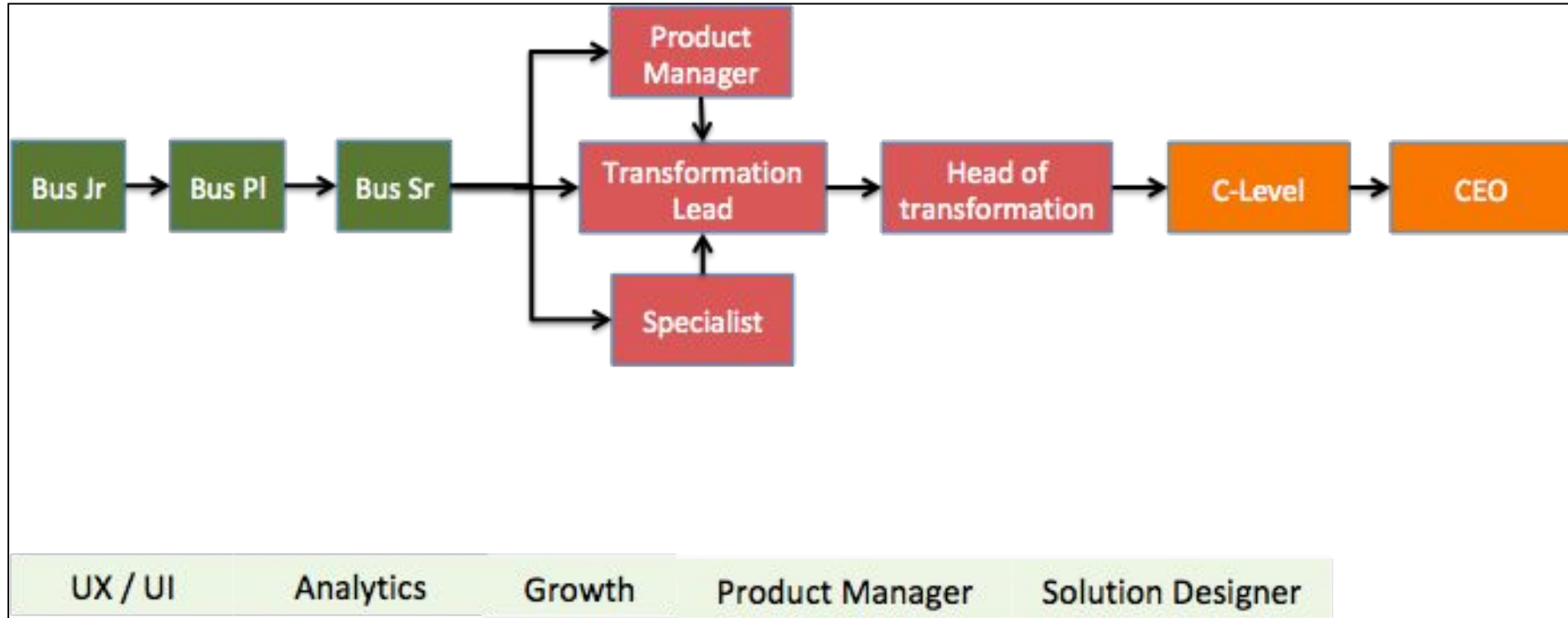
3. Arquitetura

4. Frameworks de Persistência

Pilhas de carreira no mercado de TI, **atualmente**

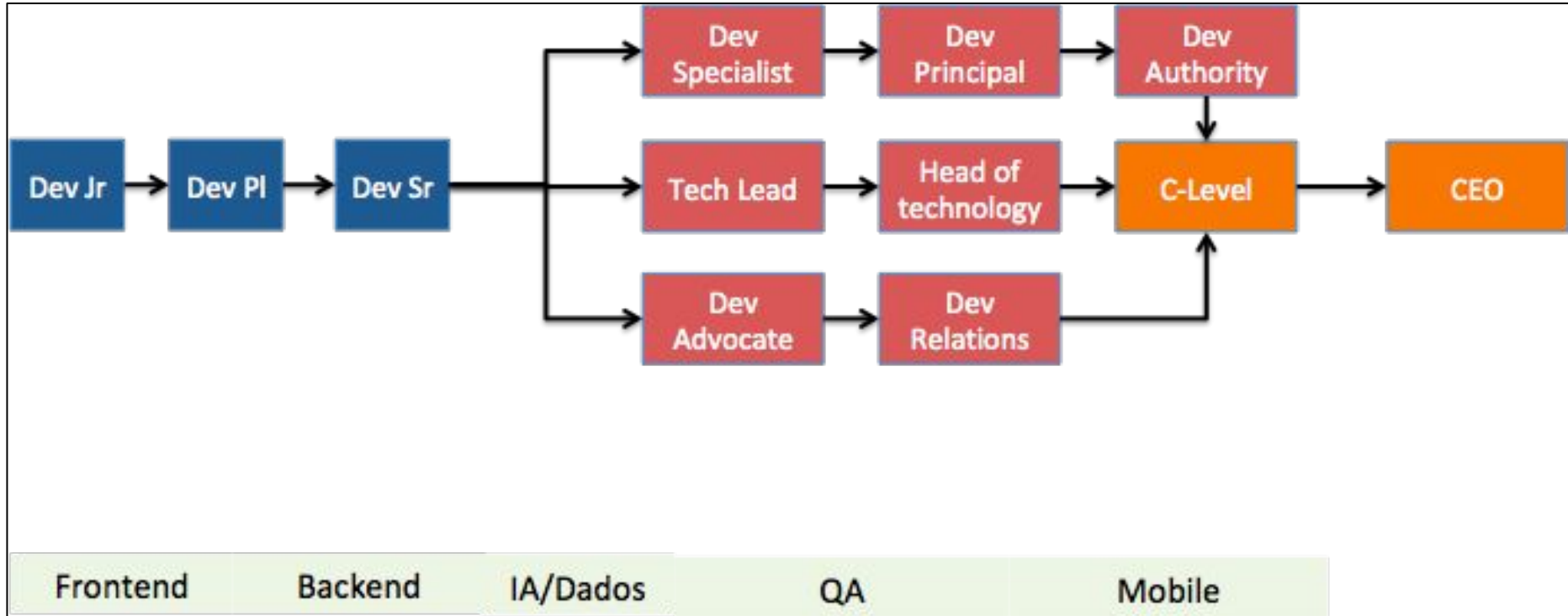


Pilhas de carreira no mercado de TI, **atualmente** (cont.)



Exemplo de uma *business stack*

Pilhas de carreira no mercado de TI, **atualmente** (cont.)



Exemplo de uma *tech stack*

Fim