

## Teoria

### Arquivos

- Armazena informação (bytes)
- É possível armazenar qualquer informação, dado o espaço
- Usado para persistência

### Formas de acesso:

- Acesso Sequencial
- Acesso Aleatório

### Abstrações em Java

- Fluxos (java.io)
  - Tráfego de uma origem (Input) para o programa
  - Tráfego do programa para um destino (Output)
  - Síncrona (o programa espera pra ler)
  - Tipos
    - Objetos (ObjectInputStream / ObjectOutputStream)
    - Bytes (ByteArrayInputStream / ByteArrayOutputStream)
    - Caracteres (Writer/Reader)
    - Dados (DataInputStream / DataOutputStream)
  - Fluxos podem ser compostos (composição)
    - Ex.: ARQUIVO ----bytes--> BUFFERIZADOR ---bytes--> PROGRAMA
    - Exemplos: Buffer, Inflate, Piped, Checked, Cipher, Digest
- Canais (java.nio)
  - Assíncrona (o programa pede pra ler e continua)
  - Sistema preenche um buffer após a leitura
  - Melhor desempenho sem esforço
  - java.nio.Files

### Armazenamento e Transferência de Objetos

- Serializáveis
  - Todo objeto que pode ser transmitido
    - DADOS SÃO TRANSMITIDOS
    - REFERÊNCIAS SÃO PERDIDAS
  - Versionamento
    - Mapeamento Objeto <-> Classe através de ID

### Sistema Mensagens

- Como usuário, quero entrar no sistema (LOGIN matheusgr)
- Como usuário, quero adicionar lembretes (LEMBRETE Comprar pão!)

- Como usuário, quero poder listar meus lembretes (LISTAR)
- Como usuário, quero poder armazenar lembretes em disco (SALVAR)
- Como usuário, quero poder sair do sistema (SAIR)
- Como usuário, quero poder carregar lembretes (CARREGAR)
- Como usuário, quero salvar o histórico das ações num arquivo (log.txt)

### Arquitetura proposta

```
Main----> FacadeUser
      |      \----> ControladorDeLembretes ---> Persistencia
      \----> Logger
```

### Situações de erro

- CARREGAR SEM SALVAR
  - FileNotFoundException
- PROBLEMA AO CARREGAR
  - Serialização
  - Sem acesso
  - Problema nos dados armazenados