



# **Sobrevivendo ao Squeak**

MAC 5714 – Programação Orientada a  
Objetos

---



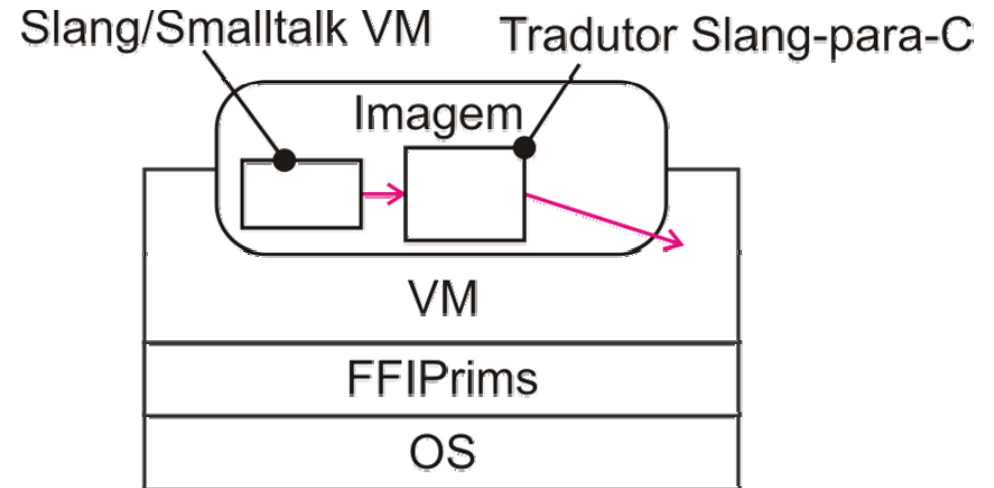
# Squeak

- Ambiente Smalltalk

- Portável;
- compacto;
- livre;
- escrito em Smalltalk;
- em menos de 6 meses.

- Ambiente para pesquisa

- Filosofia smalltalk:
  - Tudo deve ser inspecionável e modificável.





## Squeak (cont.)

---

- Legal...
  - ... mas eu acho o Squeak bizarro.
- Smalltalk predata:
  - modelo de negócios atual para software;
  - Microsoft Windows, Microsoft Office.
- Alan Kay
  - computador como ele próprio;
  - supermídia, superpapel;
  - ferramenta de aprendizado.



# Squeak (cont.)

---

- Organização:
  - 1995 – Apple computing;
  - Disney Imagineering (SqC);
    - Pequeno grupo;
    - Squeak em aplicações comerciais.
  - Squeak Foundation (SqF).
  - ViewPoints Research Institute.
    - Modelo do RedHat;
    - Dan Ingalls está na Sun Microsystems;
    - Alan Kay está na HP;
    - parece que não foi muito adiante.



# Squeak Foundation

---

- “Comunidade Squeak”;
- conflitos de organização;
- processo de desenvolvimento imaturo;
  - contribuições são desperdiçadas;
- contribuições relevantes;
  - poucas pessoas em posição de dá-las.
- Resultado:
  - avanço lento e desorganizado;
  - documentação precária.



# Squeak Foundation (cont.)

---

- Mas...
  - há bons projetos usáveis;
  - há projetos “quentes” (wxSqueak, Exupery, Seaside);
  - dicas;
  - curiosidade.
- Pauta:
  - Recuperando-se das falhas catastróficas;
  - shout;
  - refactoring browser;
  - monticello;
  - SUnit;
  - mais sobre projetos quentes.



# Falhas Catastróficas

---

- Morphic
  - idéia fantástica, em Self;
  - Squeak: bloat bloat bloat.
- *Threads Squeak*
  - Níveis de prioridade;
  - maior prioridade primeiro;
  - no mesmo nível;
    - *multithreading* cooperativo.
  - UI pode cair com facilidade.
  - ALT + .
    - suspende processo no controle da interface;
    - cria um processo novo.



# Falhas Catastróficas (cont.)

---

- ALT + . nem sempre funciona
  - processos de alta prioridade rodando;
  - erros originados na máquina virtual (raros).
- Cena:
  - EP para dia X;
  - no dia X às 23:30, a VM cai;
  - na hora em que você estava salvando;
  - no processo, sua imagem foi para o buraco;
  - E agora, quem poderá nos defender?
- Change Sets são seus amigos. Use-os.





# Falhas Catastróficas (cont.)

---

- Change Sets
  - Log de certas operações.
    - representação otimizada para escrita;
    - controle de versões;
    - *snapshotting + logging = reliable storage*;
  - Às vezes, dá trabalho;
  - menos trabalho que começar de novo.



# Shout

---

- Squeak *syntax highlighting*:
  - durante visualização.
- *Shout*
  - durante edição;
  - reduz a tentativa-e-erro.
- Bons acompanhantes:
  - Shout workspace;
  - shout monticello;
  - eCompletion.



# Refactoring Browser

---

- Browser + Ferramentas de Refatoração
  - Básico:
    - Renomear métodos, classes e categorias;
    - remover métodos e classes com segurança;
    - criação automática de *accessors*.
  - Intermediários:
    - Adicionar parâmetros;
    - *tile editing*;
    - in-lining de métodos.
  - Avançado:
    - *SmallLint*;
    - refatoração por linguagem de expressões.



# Refactoring Browser (cont.)

**Rewrite rule editor**

Search for:

```
``@Dictionary
  at: ``@key
  ifAbsent:
    [| ``@Temps |
     ``@.Statements.
     ``@Dictionary at: ``@key put: ``@object]
```

Replace with:

```
``@Dictionary
  at: ``@key
  ifAbsentPut:
    [| ``@Temps |
     ``@.Statements.
     ``@object]
```

Legend:

- ` = meta var
- @ = list
- ` = recurse into
- . = statement
- # = literal

Method

Methods...

Find...

Replace...



# Monticello

---

- Monticello
  - Empacotamento/distribuição;
  - controle de versões (das distribuições);
  - convenções.
- Formato para a entrega dos exercícios-programa:
  - **SAR** – **S**queak **AR**chive;
  - pacote monticello;
  - *preamble/postscript*;
  - bibliotecas adicionais.



# SUnit

---

- Testes de unidade automatizados para Smalltalk.
  - subclasses de TestCase;
  - overriding de métodos especiais.
- Resultados dos testes:
  - Smalltalk é reflexiva;
  - pilhas preservadas;
  - voltar e repetir casos de teste.
- Nos exercícios-programa:
  - Testes na categoria Foo-Testes.
- Bons testes = mais pontos. 😊



# Projetos Quentes

---

- Projetos mais importantes;
- “*Squeak killer apps*”;
- Seaside;
  - Arcabouço Web, vimos na aula passada.
- wxSqueak.
  - Widgets nativos para o Squeak.
- Exupery;
  - Compilador JIT, torna o Squeak 5x mais rápido.



# Outros pacotes interessantes

---

- PostgreSQL/MySQL;
- Regular Expression Plugin;
- BTree;
- Emacs keybindings;
- GOODS (General Object-Oriented Database System).





# Conclusão

---

- Squeak
  - interessante;
  - pode ser produtivo;
  - requer algum investimento;
  - vocês estão aqui prá isso! 😊



# Dúvidas?

---

