

# Seminário de Tecnologias Futuras

Python, sua agilidade e sua robustez

# O que é o Python?

- 100% OOP (de verdade)
- Dinâmica/fortemente tipada
- Compilada para bytecode e interpretada
- Multi-plataforma
- Apoiada oficialmente pela Microsoft (IronPython) e Google (Guido von Rossum)
- Mantida pela Python Software Foundation

# Filosofia (the zen of python)

- KISS (Keep it simple, stupid)
- DRY (Don't repeat yourself)
- Código explícito
- Semântica

# O que é o Django?

- Framework ágil para web
- Compatível com SGBDs: MySQL, Postgres, SQLite, Oracle e SQL Server
- Oferece ORM para acesso a dados orientado a objetos
- Criado pela Lawrence Journal-World e disponibilizado como software livre
- Extremamente simples e rápido

# Período pré-WWW

- 60's ~ 80's
  - Processo de consolidação da área
  - Computador ainda não popularizado
  - Superação da linguagem de máquina
- 90's
  - Pequenas empresas
  - Internet
  - Massificação do computador
  - Software Livre

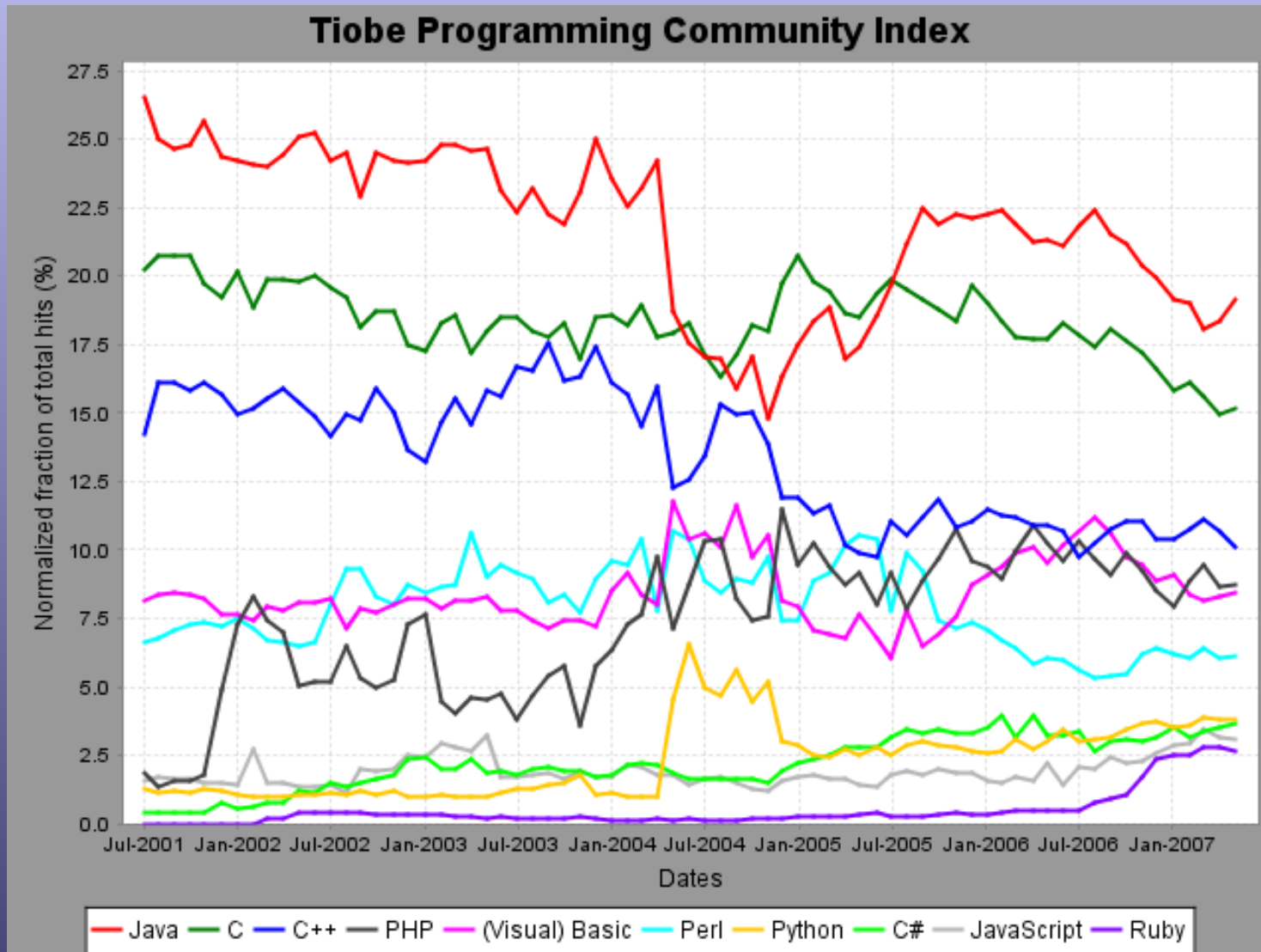
# Nova realidade

- 2000~2004
  - Popularização da OO
  - Multi-plataforma
  - Integração
  - Dispositivos móveis
- 2005~?
  - Web2.0
  - Agilidade
  - SOA
  - TV Digital

# Cronologia das tecnologias

- 60's ~ 80's
  - COBOL, Fortran, Dataflex, MUMPS, Pascal, C++, dBase
- 90's
  - Clipper, Fox Pro, Visual Basic, Delphi
- 2000~2004
  - Java, C#, PHP, ASP
- 2005~?
  - Python, Ruby, Lisp, Scheme, Lua, JavaScript

# Gráfico de “calor” das linguagens



# Agilidade

- Na criação de sistemas (desenvolvimento)
- Na evolução de sistemas (manutenção)
- No treinamento de novos profissionais
- Na performance do lado Servidor
- Na performance do lado Cliente
- Na implantação
- Nas mudanças de tecnologia
- Nas integrações

# Agilidade no desenvolvimento

- A framework Django cria o esquema da base de dados
- DRY - Don't Repeat Yourself
- Grande quantidade de frameworks e bibliotecas prontas

# Agilidade na manutenção

- Linguagem semântica e indentada
- Código de fonte pequeno e explícito - fácil de manter
- Boas práticas das PEP's facilitam manutenção
- Python é tão simples, que dispensa ferramentas RAD

# Agilidade para novos profissionais

- Apesar do mercado oferecer mais profissionais que se dizem preparados em Java, Delphi ou C#, a maioria deles não possui noções básicas como Lógica de Programação ou OOP
- Curva de aprendizado é muito pequena
- Mais rápido e barato treinar um profissional com Python do que “estagiar” um profissional de linguagens mais complexas
- Não possui certificação porque é muito simples

# Agilidade no servidor

- Robôs de busca do Google, Yahoo e YouTube usam Python por causa da performance
- Grande parte das ferramentas de servidores Linux para indexação e manutenção são escritas em Python

# Agilidade no cliente

- Adotada oficialmente pela maioria das distros Linux para ferramentas GUI
- Adotada por aplicações de renderização de imagem e 3D

# Agilidade na implantação

- Faça o teste: implante um site em J2EE e um em Django num servidor de hospedagem
- Python Eggs possibilita implantação modular extremamente simples
- Várias máquinas virtuais, com realidades completamente distintas - de forma fácil

# Agilidade na mudança

- Python é adaptável a praticamente tudo
- Como a curva de aprendizado é baixa, mudança tem custo baixo
- MVC possibilita mudança parcial
- Banco de dados e templates são reutilizáveis

# Integrações

- Jython -> Java
- IronPython -> .Net
- WebServices, XML e JSON de maneira simples
- Bibliotecas gráficas MFC, GTK+, Qt, wxWindows
- Os mais variados SGBDs são suportados por Python
- Próximas versões do Visual Studio e Silverlight terão Python (e Ruby)

# Robustez e escalabilidade

- Ferramentas pesadas do Google, Yahoo, Washington Post, Yahoo e Bittorrents são construídas em Python
- Sistemas de missão crítica da NASA são construídos em Python
- Zope/Plone são mais conhecidos que o Python nos meios governamentais para portais e intranets
- Cálculos de terabytes que renderam relatório sobre aquecimento global foram construídos em Python (PyDAC)
- E quem disse que pra ser robusto e escalável, precisa ter milhares de linhas de código e dezenas de bibliotecas?

# Flamewars: qual é a melhor?

- Todas as linguagens, especialmente as mais populares “do futuro” (Java, C#, Python, Ruby e PHP) são excelentes
- Produtividade e performance se obtém com: profissionais qualificados, framework e processo bem definido
- Porém, existem pontos fortes e fracos de cada uma dessas linguagens

# Pontos “pró”

- Python na web é praticamente imbatível:
  - possui muitas bibliotecas práticas
  - oferece um código limpo e fácil de manter
  - oferece frameworks ágeis, como Django, TurboGears e Zope/Plone
- Python só fica atrás do Java em compatibilidade com dispositivos móveis
- Python é prático e ágil para GUI
- Tempo curto para preparo de novos profissionais

# Pontos “contra”

- Em móveis e TV digital, Java vence
- Em assunto de pacotes de ferramentas, Visual Studio.Net vence
- Em assunto de migração no ambiente da Tron, Delphi vence
- Em assunto de mercado, PHP, Java e Delphi oferecem mais profissionais (em quantidade)
- Em assunto de frameworks, o Ruby on Rails é o mais maduro da atualidade

# Sugestão do autor

- Adoção do Python (especialmente Django) ou Ruby (especialmente Rails) para projetos web e ferramentas ágeis

# Perguntas?

**Marinho Brandão**  
Equipe de Desenvolvimento  
[josemario@tron.com.br](mailto:josemario@tron.com.br)