



EDUCAÇÃO PRESENCIAL E A DISTÂNCIA

Engenharia de Requisitos

Edson Orivaldo Lessa Junior

Engenharia de Requisitos

- O que é?

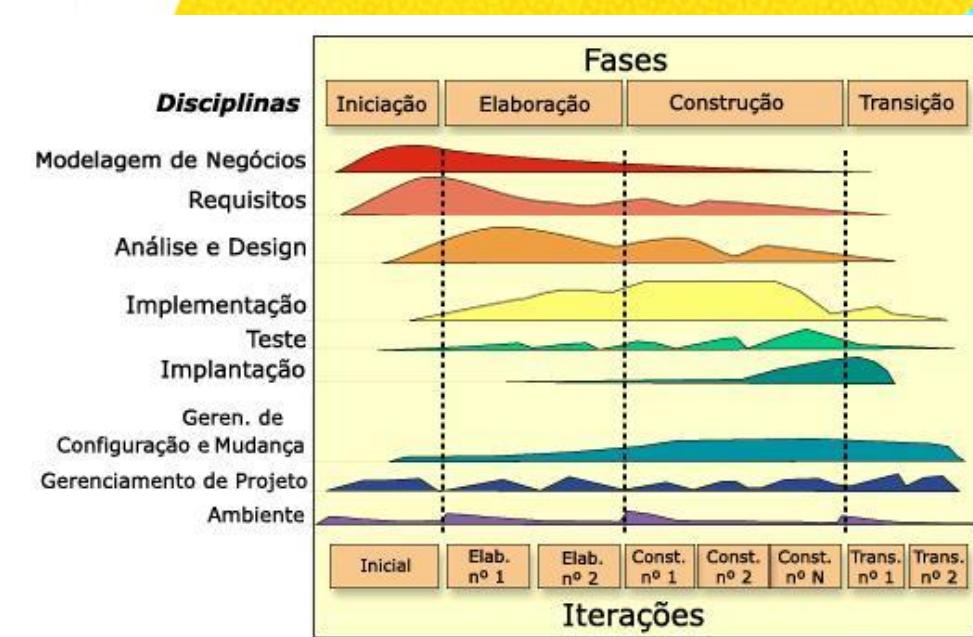
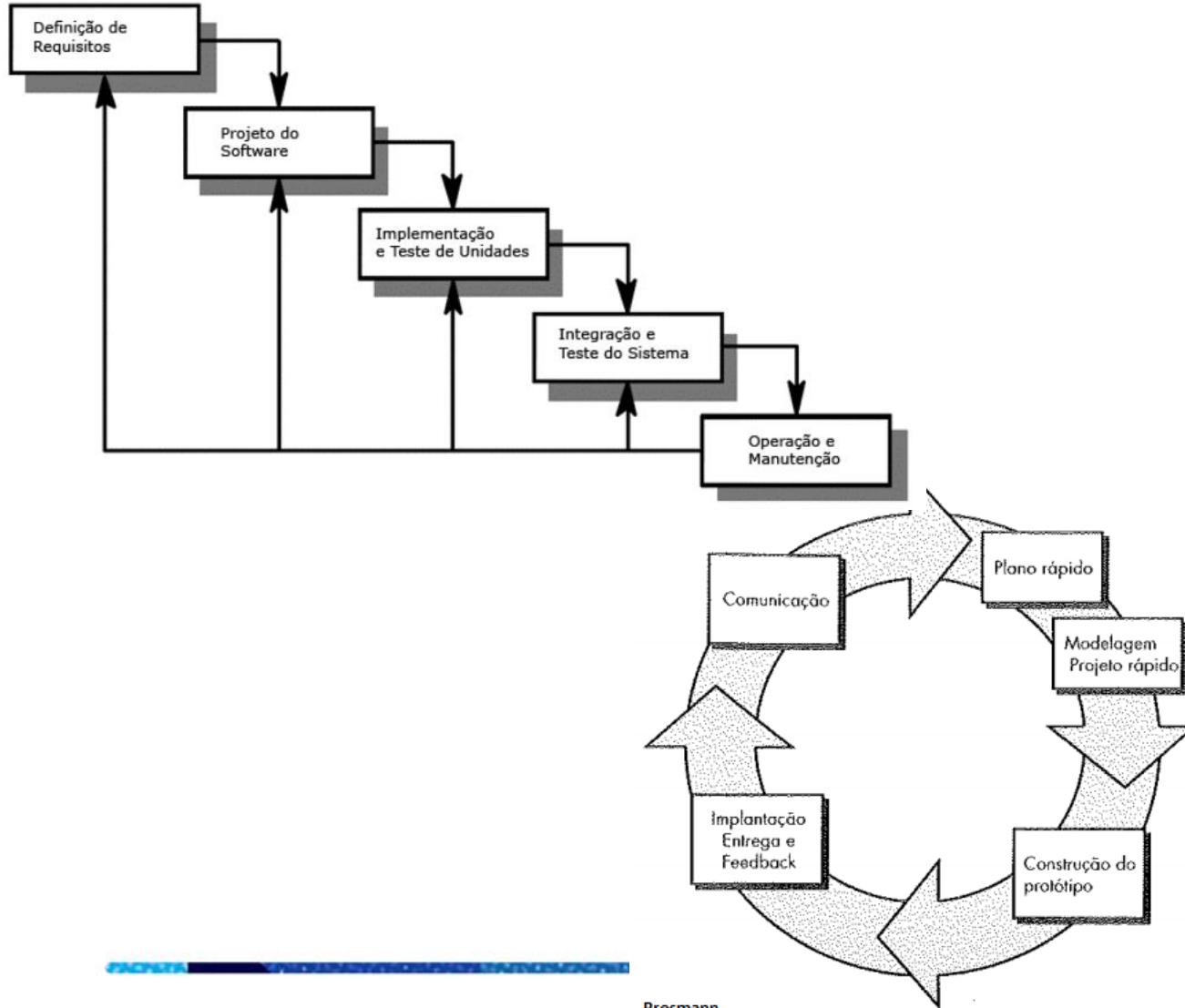




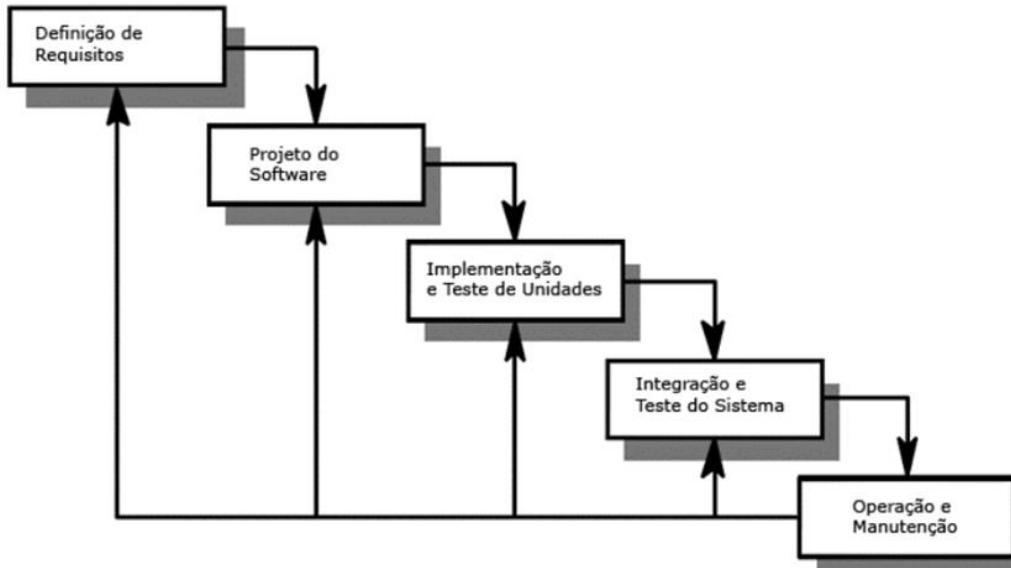
é o processo pelo qual os requisitos de um produto de software são coletados, analisados, documentados e gerenciados ao longo de todo o ciclo de vida do software.

Medeiros, Higor - 2003

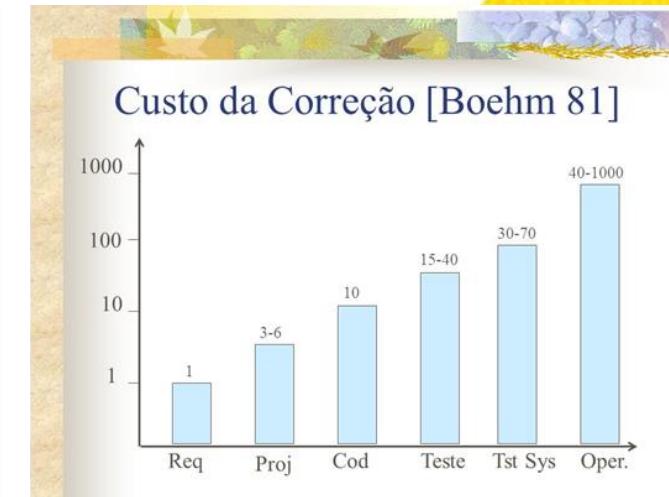
Modelos de processo



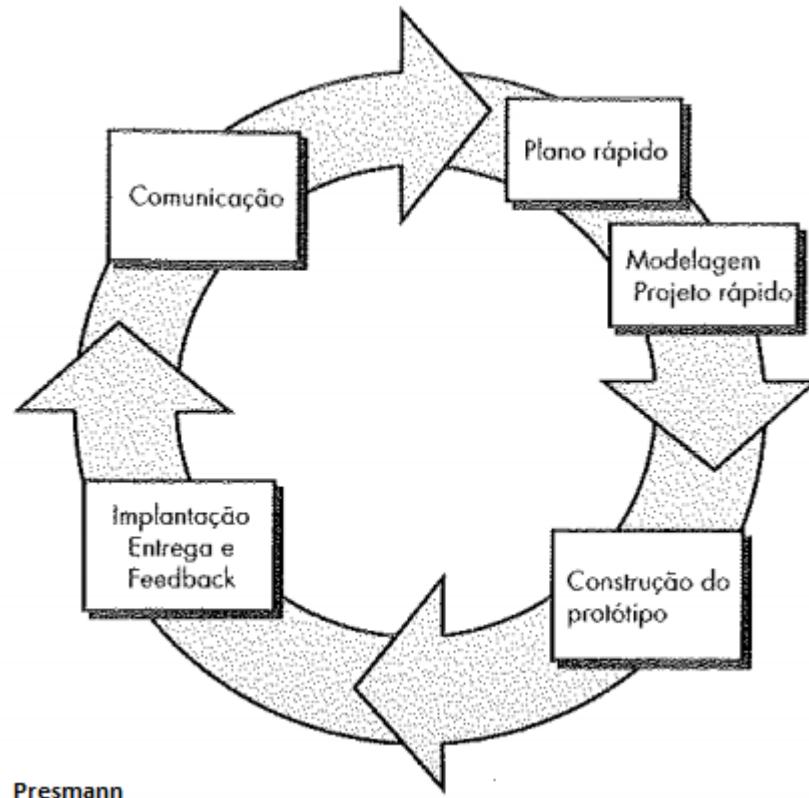
Modelo em cascata



- Demora na entrega do produto
- Acúmulo de riscos
- Inflexível a mudanças nos requisitos
- Porém, muito usado em outros modelos
- Interessante para requisitos estáveis



Prototipagem



- **Evolucionária**
 - Para sistemas pequenos
 - Requisitos bem compreendidos
- **Descartável**
 - Para sistemas maiores
 - Requisitos vagos
- **Usuários sentem o sabor do sistema**
- Facilita validação de requisitos
- Pode gerar arquiteturas precárias

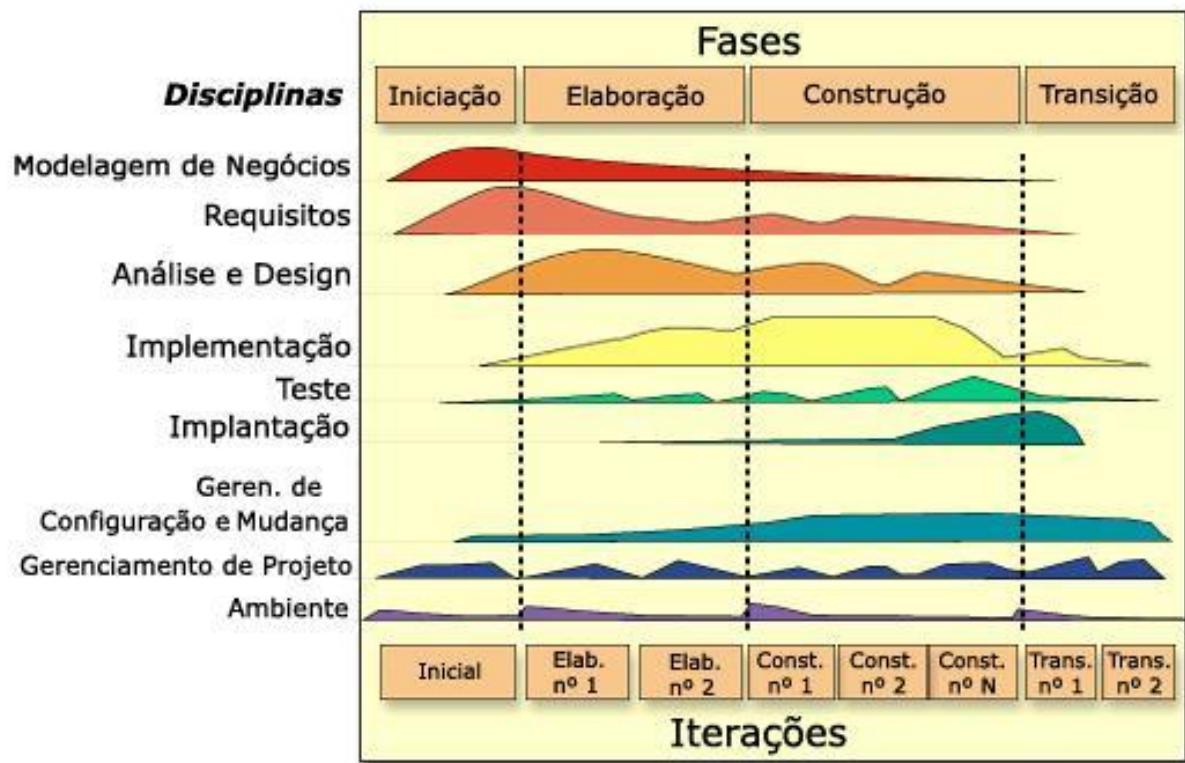
Modelo espiral



Palavra chave: RISCO

- Busca unir o aspecto iterativo da Prototipagem com a sistemática do Cascata.

Modelos iterativos - RUP

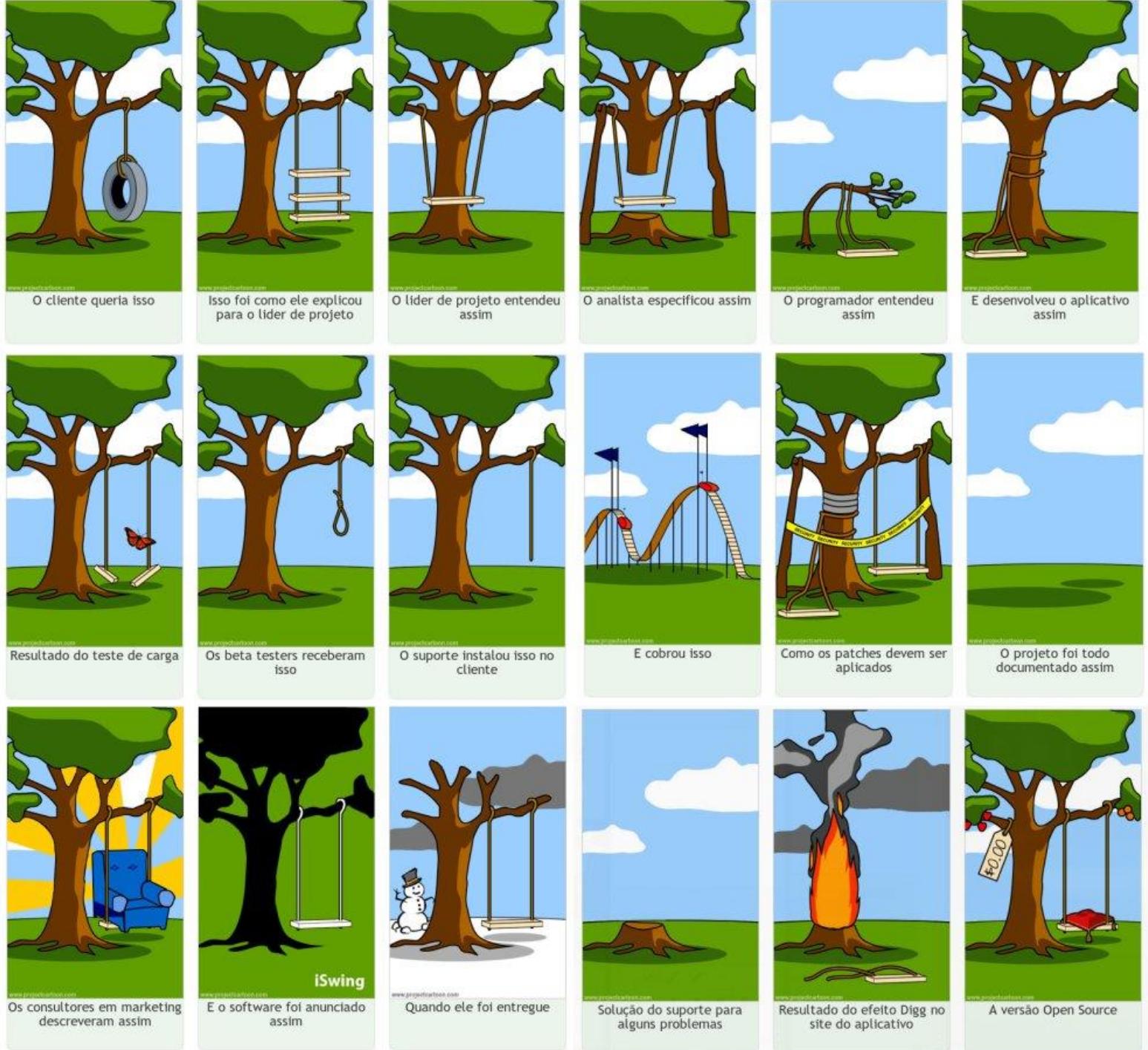


- Palavra chave: UNIFICADO
- Fases com atividades sobrepostas
- Forte integração com a UML
- A implantação passa a fazer parte do processo

RUP – boas práticas

- Desenvolver software iterativamente
- Gerenciar requisitos
- Usar arquitetura baseada em componentes
- Modelar software visualmente
- Verificar a qualidade do software
- Controlar as mudanças do software

Processo de Software



Métodos Ágeis



Título da disciplina

Quem criou o Manifesto Ágil?

- **Robert C. Martin**, o “Uncle Bob”
- **Ken Schwaber**, co-criador do Scrum.
- **Jeff Sutherland**, o inventor do Scrum.
- **Kent Back**, co-criador da eXtreme Programming (XP).
- **Ron Jeffries**, co-criador da eXtreme Programming (XP).
- **Mike Beedle**, co-autor de Desenvolvimento Ágil de Software com Scrum.
- **Arie van Bennekum**, da Integrated Agile.
- **Alistair Cockburn**, criador da Metodologia Ágil Crystal.
- **Ward Cunningham**, criador do conceito wiki.
- **Martin Fowler**, desenvolvedor parceiro da Thoughtworks.
- **James Grenning**, autor de Test Driven Development.
- **Jim Highsmith**, criador do Adaptive Software Development (ASD).
- **Andrew Hunt**, co-autor de O Programador Pragmático.
- **Jon Kern**, atuante até os dias de hoje em assuntos de agilidade.
- **Brian Marick**, cientista da computação e autor de vários livros sobre programação.
- **Steve Mellor**, cientista da computação e um dos idealizadores da Análise de Sistema Orientado a Objetos (OOSA).
- **Dave Thomas**, programador e co-autor de The Pragmatic Programmer.

Valores do Manifesto Ágil

- Indivíduos e interações mais que processos e ferramentas
- Software em funcionamento mais que documentação abrangente
- Colaboração com o cliente mais que negociação de contratos
- Responder a mudanças mais que seguir um plano

Os 12 princípios ágeis



1. Priorize o Cliente



2. Mudanças são bem-vindas



3. Entregas que geram valor



4. Cooperação entre squads e stakeholders



5. Confie e apoie



6. Conversa face a face



7. Priorize o que funciona



8. Desenvolvimento incremental



9. Excelência técnica



10. Mantenha a simplicidade



11. Times autônomos e auto-gerenciáveis



12. Retrospectiva e planejamento

Especificação



Título da disciplina

Projeto



Título da disciplina

Validação



Evolução



Aonde você quer chegar?

Vai com a

