



- Desenvolvimento
- Referências

Sistemas Embarcados: (ELF74)

Prof: DaLuz

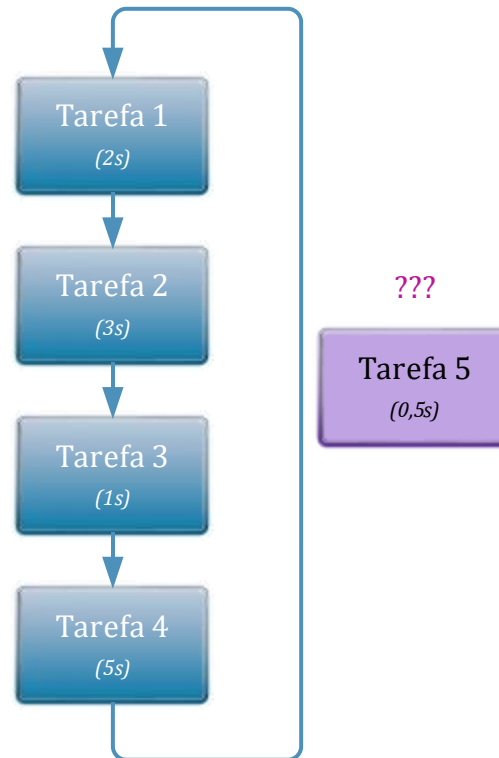


- Desenvolvimento

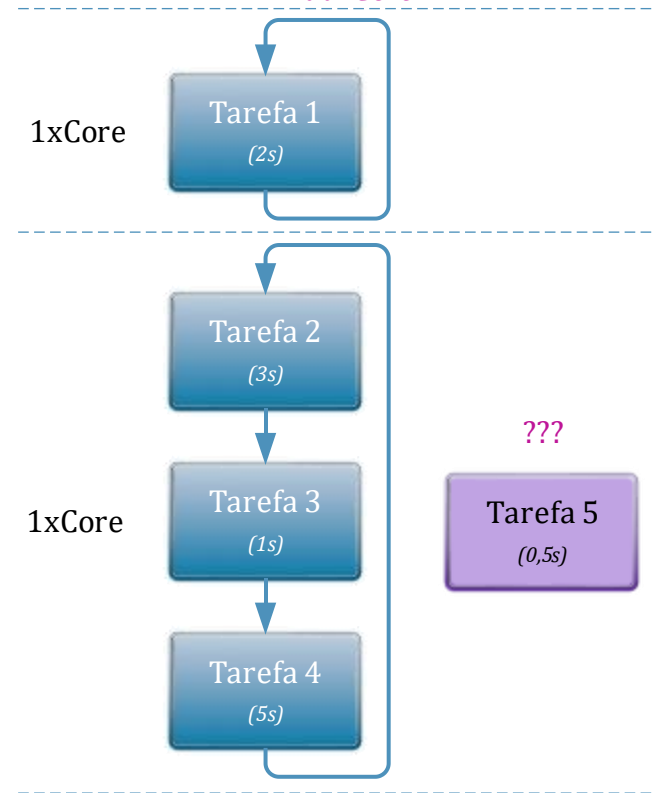
- Referências

Desenvolvimento:

Fluxo de um Programa
"Single Core"



Fluxo de um Programa
"Dual Core"

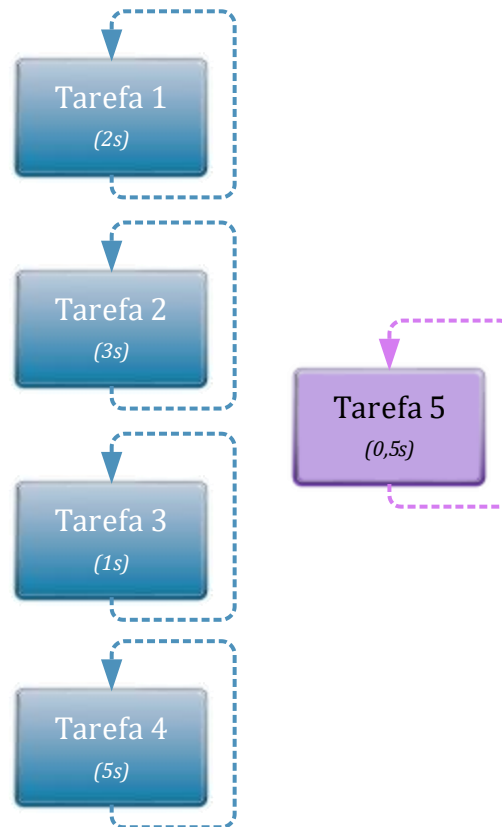




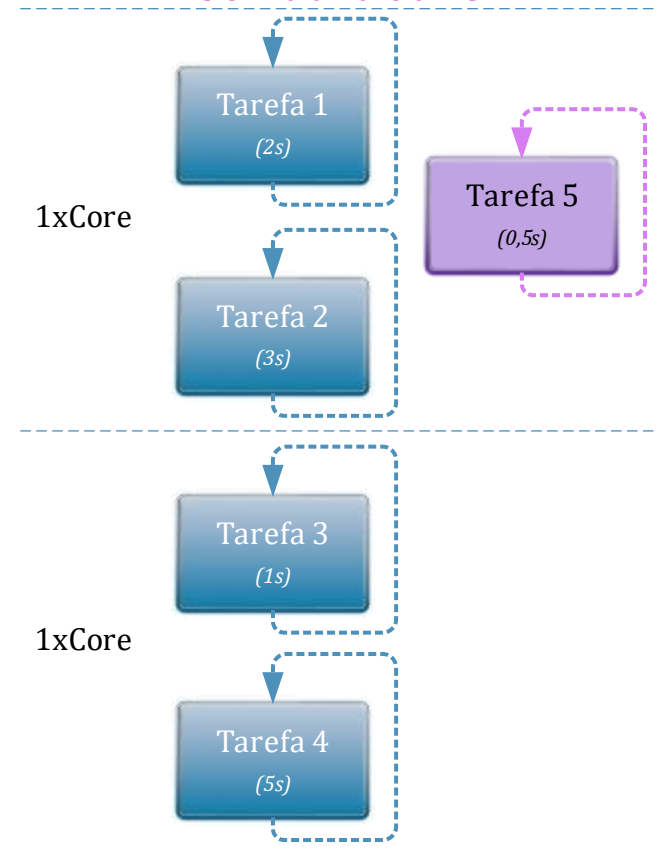
- Desenvolvimento
- Referências

Desenvolvimento:

Fluxo de um Programa
"S.O. Multi-Tarefa | S.C"



Fluxo de um Programa
"S.O. Multi-Tarefa D.C."



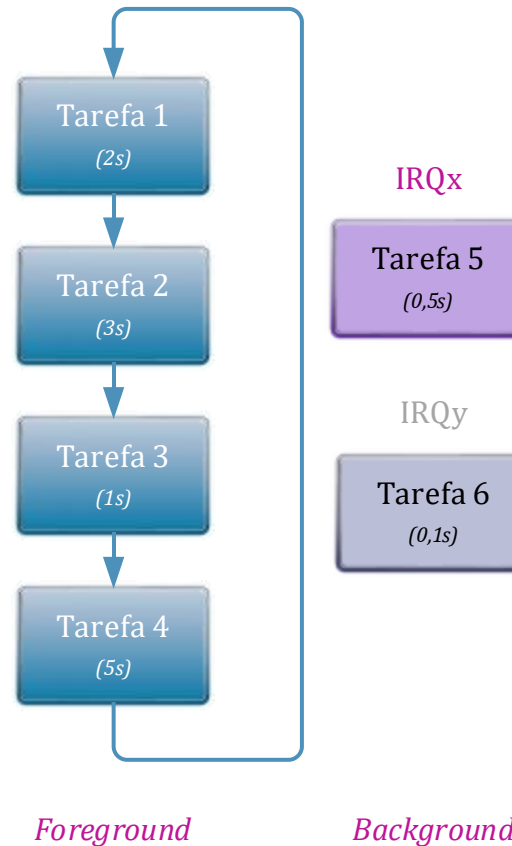


- Desenvolvimento

- Referências

Desenvolvimento:

Microcontrolados:
S.C. "Super Loop"





- Desenvolvimento

- Referências

Desenvolvimento:

Super Loop (Foreground / Background):

Variáveis **Globais**: “Permitidas e necessárias”

Espelhamento de *Hardware* em variáveis de memória **RAM**.

Evitar ao máximo o uso de **DELAY**: “Proibitivo”

Utilizar **Flags** de sinalização de condições no **P.P.**

Fluxo **contínuo** no **P.P.** ao máximo possível

Exemplo: [MyBlink3](#)



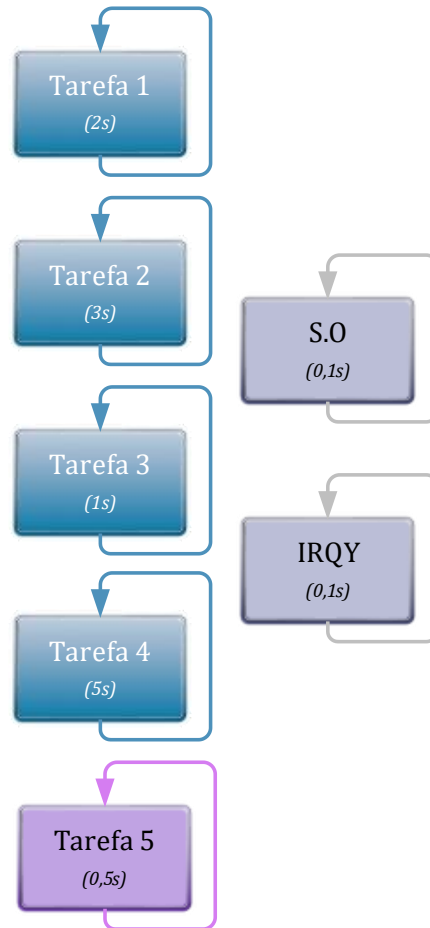


- Desenvolvimento

- Referências

Desenvolvimento:

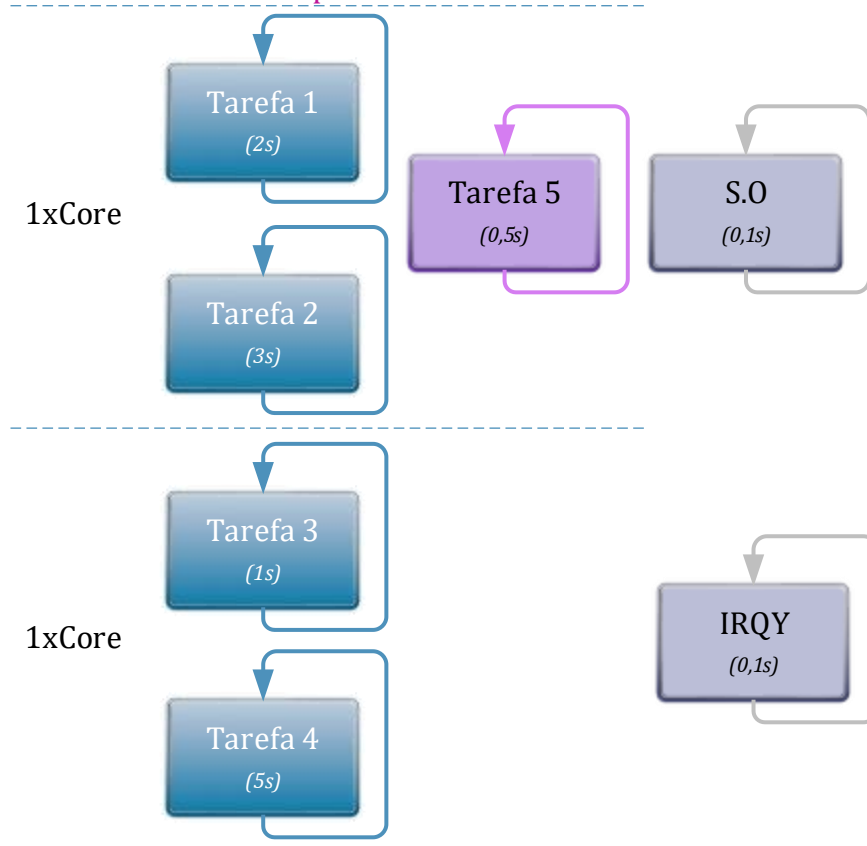
Fluxo de um Programa
"S.O. Tempo Real | S.C"



Foreground

Background

Fluxo de um Programa
"S.O. Tempo Real D.C."



Foreground

Background



- Desenvolvimento

- Referências

Desenvolvimento:

Doxygen:



Linguagem estilo “**Markdown**” para geração de documentação de código através dos comentários.

Comentários formatados em **Doxygen**, podem gerar páginas **HTML** e **PDFs** com facilidade.

Conjunto de regras de marcação de texto.

Ex1: Biblioteca BCM2835:

<https://www.airspayce.com/mikem/bcm2835/>

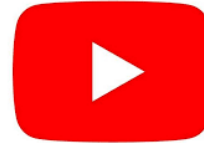


- Desenvolvimento

- Referências

Desenvolvimento:

Git:



É um sistema de **controle** de **versões** distribuído.

Criado por “Linus Benedict Torvalds” para gerenciar o desenvolvimento do **Kernel Linux**.

Foi criado para ser uma versão **livre** de sistemas de versionamento como: Darcs, **BitKeeper**, Mercurial, SVK, Bazaar, **Monotone**, etc.

GitHub.com: Atualmente é propriedade da **MicroSoft**, e é um repositório *online* muito conhecido.

*Ref. **



- Desenvolvimento
- Referências

Desenvolvimento:

Super Loop:



+ - 75%

- **Dificuldade** de **expansão** de código.
- **Dificuldade** de **múltiplos** programadores no projeto.
- **Limitações** no atendimento temporal de eventos fora das IRQs / RSIs.
- Atendem até um certo nível de complexidade no sistema.

Sis. Embarcados:



+ - 25%

- S.O. de Tempo Real.
- Facilidade** de **expansão** de código.
- Facilidade** de termos **múltiplos** programadores no projeto.
- Projetado para melhor **atender** as questões **temporais** no **P.P.**
- Suporte de **Drivers** / **Bibliotecas**



- Desenvolvimento

- Referências

Desenvolvimento:

Objetivos:

Estudar as **principais etapas** do processo de desenvolvimento de **sistemas embarcados**:

- Modelo de desenvolvimento em V
- Engenharia de requisitos
- Projeto do sistema
- Implementação e testes
- Diagramas UML úteis



- Desenvolvimento

- Referências

Desenvolvimento:

Etapas do Desenvolvimento:

- Fase de projeto:
 1. Concepção do produto
 2. Engenharia de requisitos e de sistemas
 3. Projeto arquitetural e detalhado
- Fase de implementação:
 4. Desenvolvimento de *hardware*
 5. Desenvolvimento de *software*
 6. Desenvolvimento da mecânica



- Desenvolvimento

- Referências

Desenvolvimento:

Etapas do Processo:

- Fase de testes:
 7. Integração do sistema
 8. Testes em laboratório
 9. Testes em campo
- Fase de produção:
 10. Documentação de produto + produção
 11. Empacotamento do produto
 12. Descarte do produto

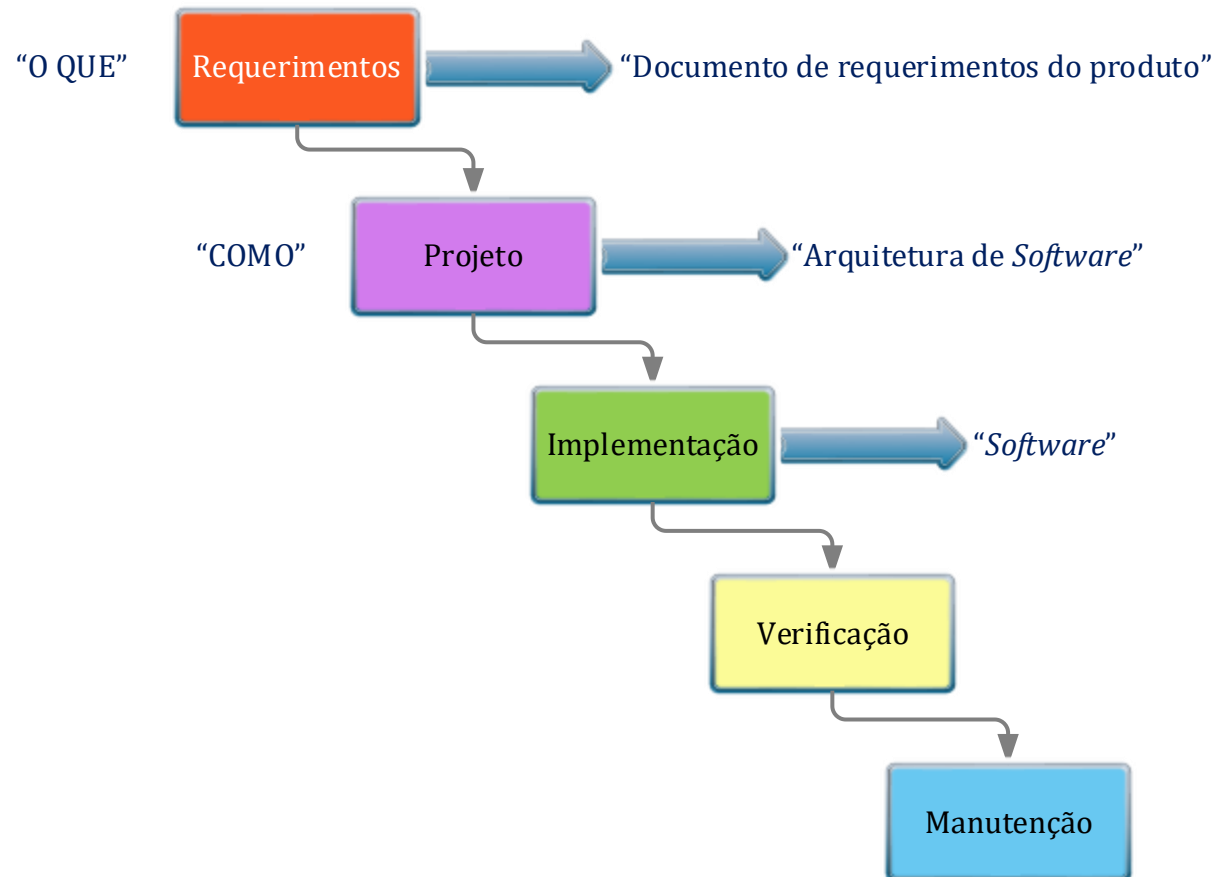


- Desenvolvimento

- Referências

Desenvolvimento:

Método Cascata (WhaterFall):

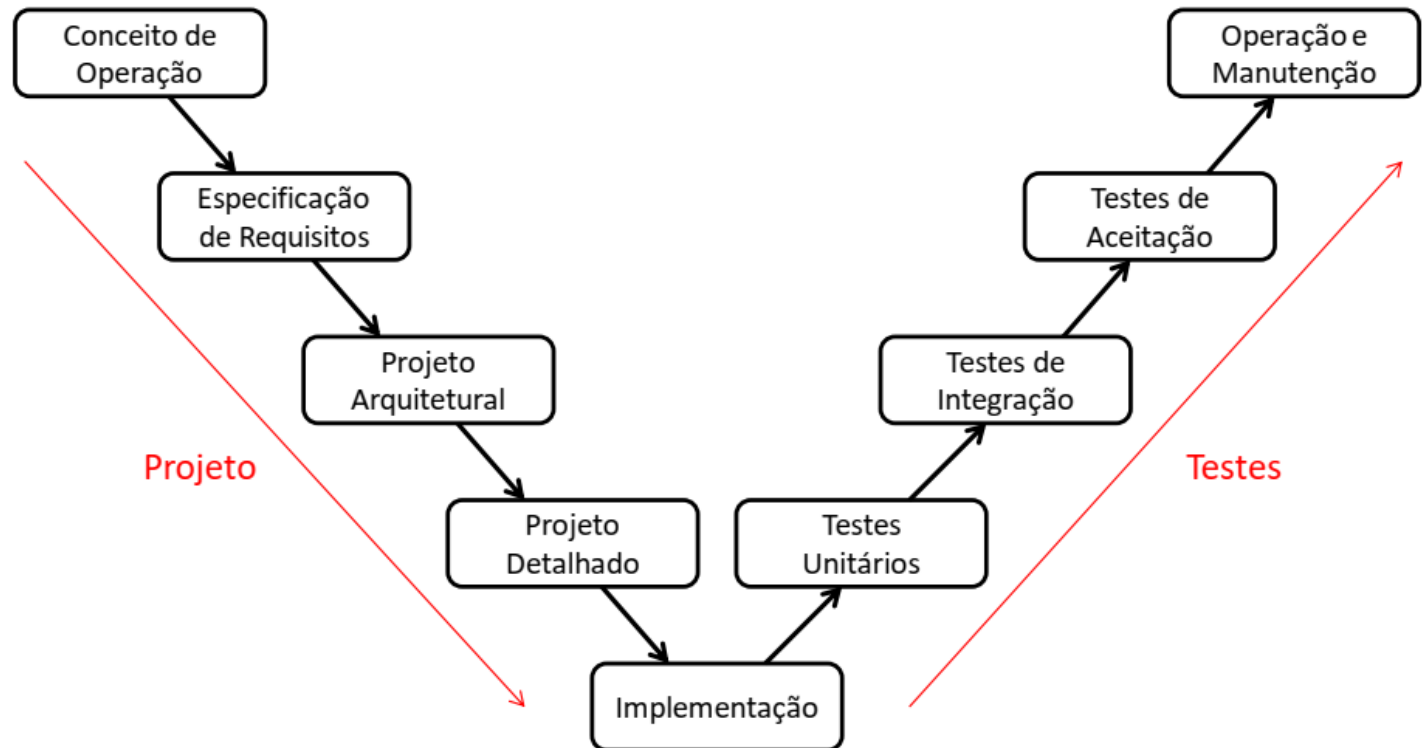




- Desenvolvimento
- Referências

Desenvolvimento:

Modelo de Desenvolvimento V:

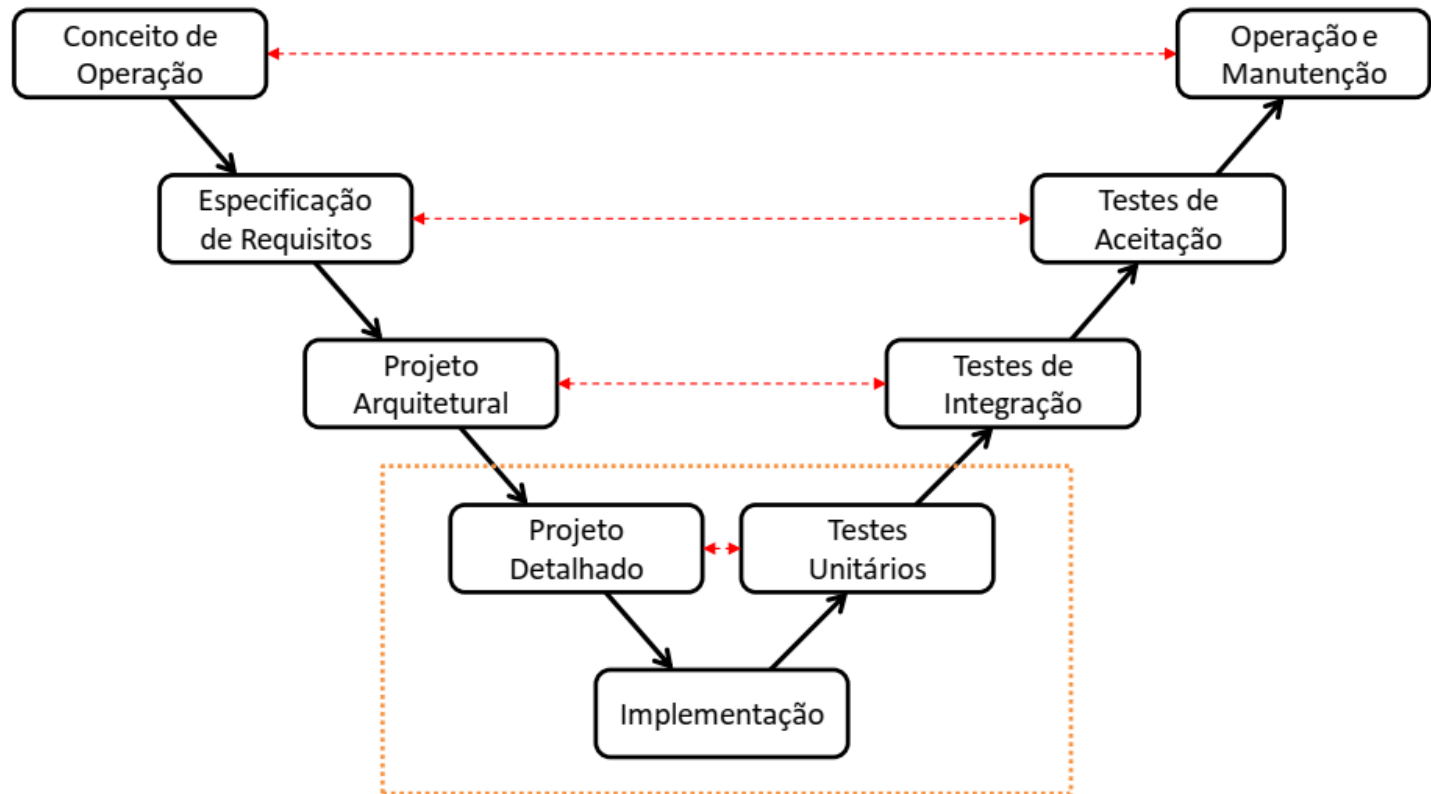




- Desenvolvimento
- Referências

Desenvolvimento:

Modelo de Desenvolvimento V:





- Desenvolvimento

- Referências

Desenvolvimento:

Engenharia de Requisitos:

- Entrada: requisitos (informais, de cliente, de normas regulatórias)
- Objetivo: entender o problema
- Meios: modelagem, prototipação
- Saída: especificação de requisitos (formal, clara, precisa, consistente, completa)
- ✓ Funcional: o que o sistema faz, como se usa
- ✓ Não-funcional: desempenho, robustez, aspectos do desenvolvimento
- **O QUE**



- Desenvolvimento

- Referências

Desenvolvimento:

Projeto do Sistema:

- Entrada: especificação de requisitos
- Objetivo: planejar a solução
- Meios: modelagem, prototipação
- Saída: documentação de projeto (descrição da solução planejada)
- ✓ Estudo da plataforma de *hardware*, ferramentas e bibliotecas disponíveis
- ✓ Projeto da arquitetura do sistema, projeto detalhado de cada bloco
- **COMO**



- Desenvolvimento

- Referências

Desenvolvimento:

Implementação e Testes:

- Entrada: documentação de projeto
- Objetivo: gerar produto
- Meios: montagem de *hardware*, codificação de *software*, integração, testes
- Saída: protótipo de produção, documentação de produção
- **AÇÃO**



- Desenvolvimento

- Referências

Desenvolvimento:

Etapas Resumidas:

- Funcionalidades do Dispositivo
- Domínio do Problema
- Análise das Interfaces
- Requisitos Funcionais
- Requisitos Não-funcionais
- Estudo da Plataforma



- Desenvolvimento

- Referências

Desenvolvimento:

Projeto Arquitetural:

- Arquitetura funcional (abstrata):
 - Diagrama em blocos
 - Cada bloco representa uma função do sistema
- Arquitetura física:
 - Bloco de *hardware*: unidade funcional
 - Bloco de *software*: tarefa, função, estrutura
 - *Utiliza-se diagramas de classes/objetos (UML) com a notação de classe/objeto ativo*



- Desenvolvimento

- Referências

Desenvolvimento:

Projeto Detalhado:

- *Hardware:*
 - Diagrama esquemático de cada unidade funcional
- *Software:*
 - Modelos dinâmicos de cada tarefa ou função
 - Diagramas de Estados (UML)
 - Diagramas de Atividades (UML)



- Desenvolvimento

- Referências

Desenvolvimento:

Comparativo Simples (Relativo):

Característica	Modelo Cascata	Modelo V
Especificações de Requisitos	No Início	No Início
Custo	Baixa	Alto
Garantia de Sucesso	Baixa	Alta
Simplicidade	Simples	Média
Flexibilidade	Rígido	Pouco Flexível
Reusabilidade	Limitada	Até certo ponto
Envolvimento do Usuário	Somente no Início	Mais no Início
Alterações	Difícil	Difícil
Tempo	Rápido	Longo



- Desenvolvimento

- Referências

Desenvolvimento:

Um bom projeto de software deve:

- 1) Planejamento das fases de desenvolvimento.
- 2) Definição do problema a ser resolvido.
- 3) Especificação da solução.
- 4) Escolha da plataforma de *Hardware*.
- 5) Planejamento das estruturas de dados.
- 6) Projeto da solução.
- 7) Iteração de dados (parâmetros).
- 8) Algoritmos necessários para as tarefas.
- 9) Configuração da IDE.
- 10) Construção do código para a tarefa.
- 11) Testes e depuração do código.



- Desenvolvimento

- Referências

Desenvolvimento:

Projeto de Software (UML):

	Estrutural (Estático)	Comportamental (Dinâmico)
Alto Nível (Arquitetural)	<ul style="list-style-type: none">• Diagramas de Classes/Objetos (geral)	<ul style="list-style-type: none">• Diagramas de Estados• Diagramas de Sequência
Baixo Nível (Detalhado)	<ul style="list-style-type: none">• Diagramas de Classes/Objetos (cada tarefa)	<ul style="list-style-type: none">• Diagramas de Estados• Diagramas de Atividades

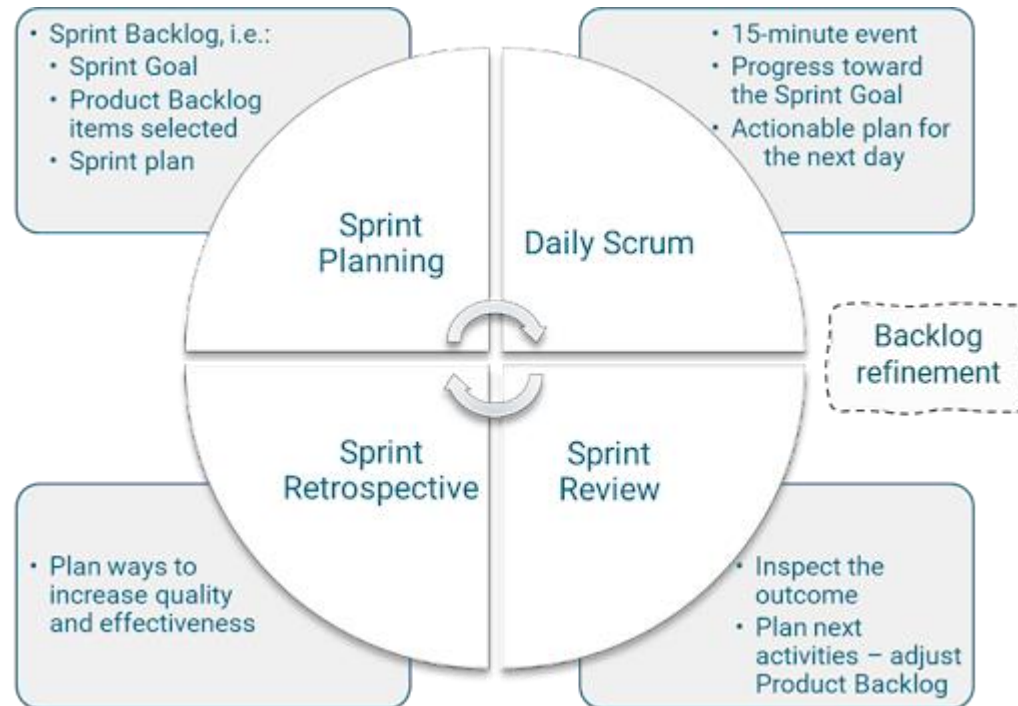


- Desenvolvimento

- Referências

Desenvolvimento:

Scrum: Desenvolvimento de Software



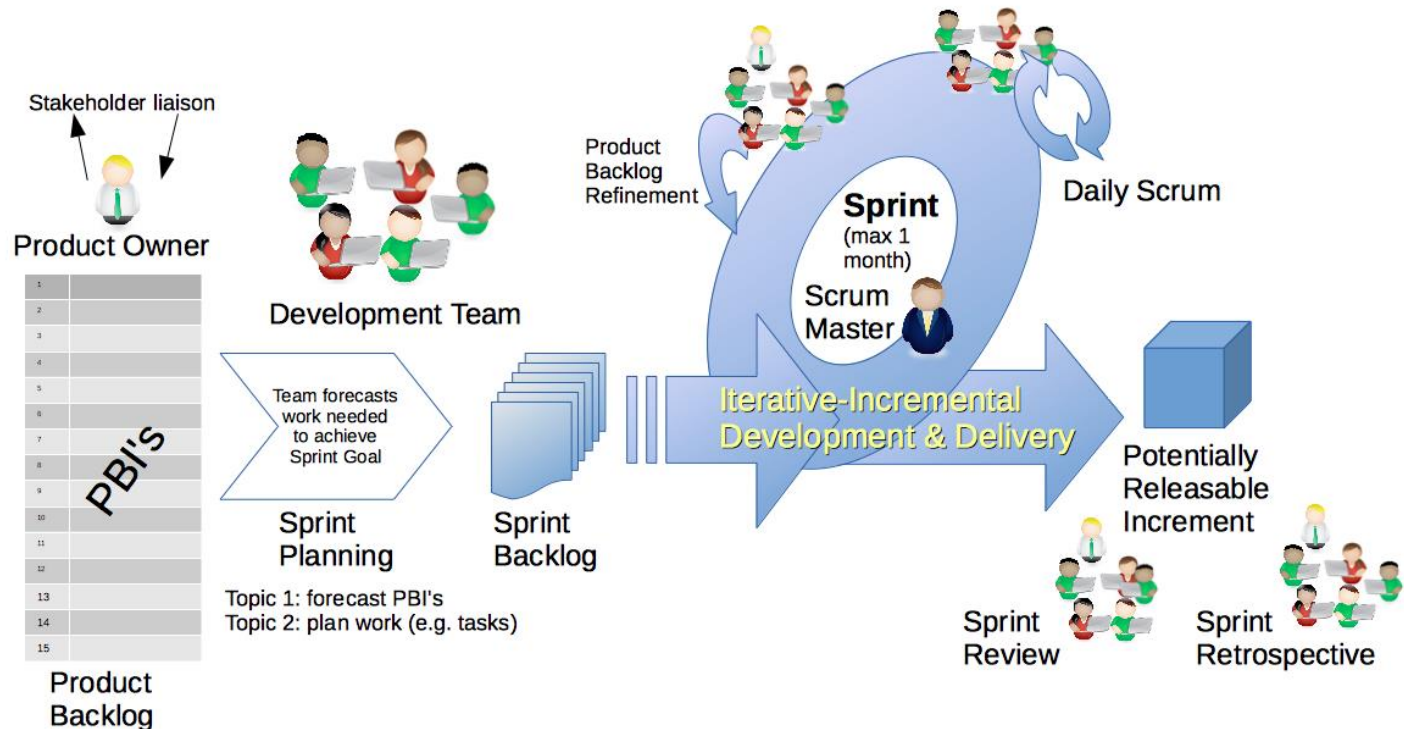
[https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))



- Desenvolvimento
- Referências

Desenvolvimento:

Scrum: Framework



Ref. *

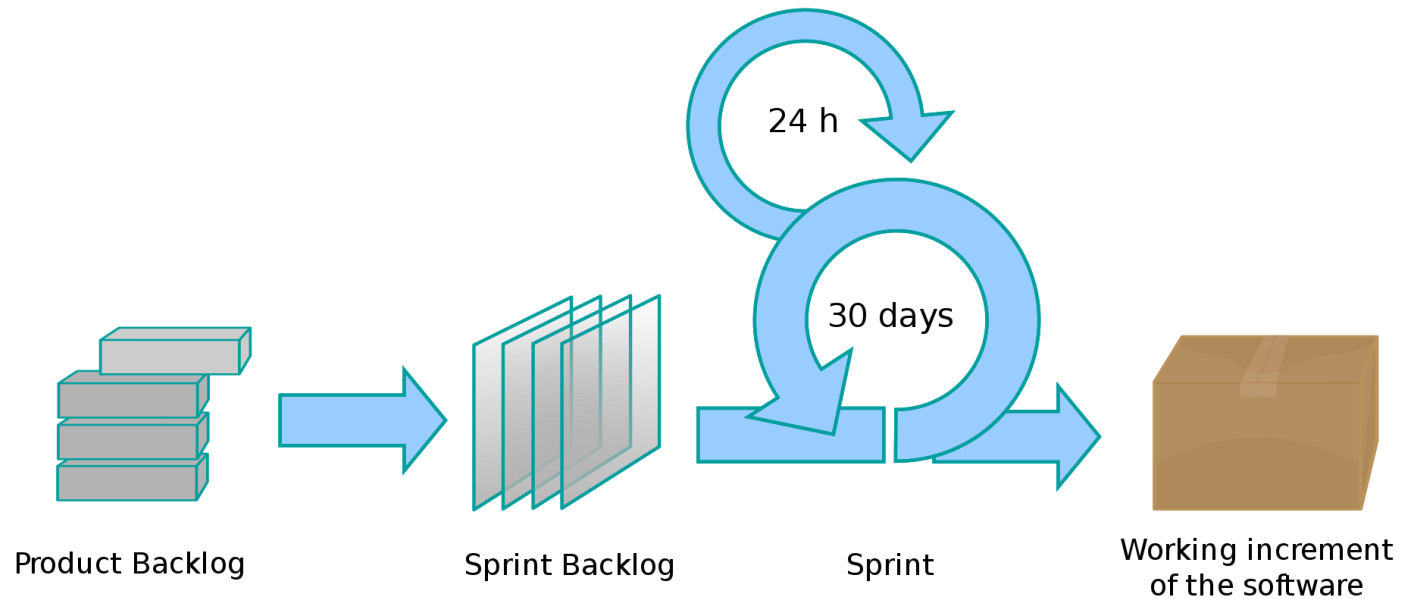


- Desenvolvimento

- Referências

Desenvolvimento:

Processo Scrum:



Ref. *



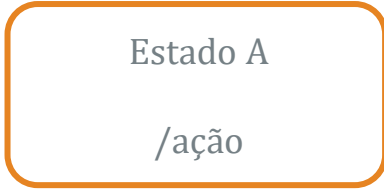




- Desenvolvimento

- Referências

Desenvolvimento:

Principais Entidades:

- Estado inicial: 
- Estado final: 
- Estado: 

Estado A
/ação
- Transição: 
- Evento e Condição: 

Evento [condição] /ação

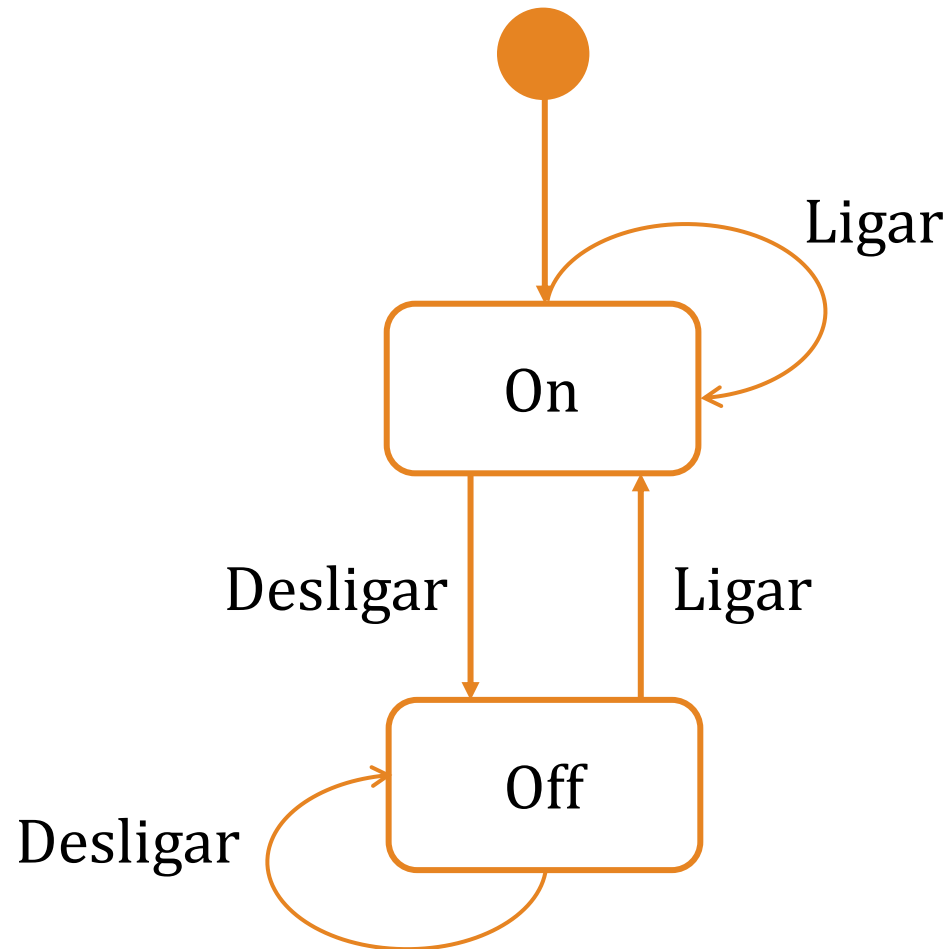


- Desenvolvimento

- Referências

Desenvolvimento:

Exemplo: On/Off (Switch)





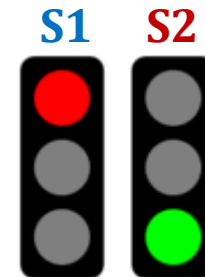
- Desenvolvimento

- Referências

Desenvolvimento:

Exemplo: Semáforo

- Cruzamento de duas vias de mão única com dois semáforos **S1** e **S2**: (**sem o estado do pisca alerta**)
 - $T_{\text{verde}} = 20 \text{ seg.}$
 - $T_{\text{amarelo}} = 4 \text{ seg.}$
 - $T_{\text{vermelho}} = 28 \text{ seg.}$
 - $T_{\text{sobreposição}} = 2 \text{ seg.}$

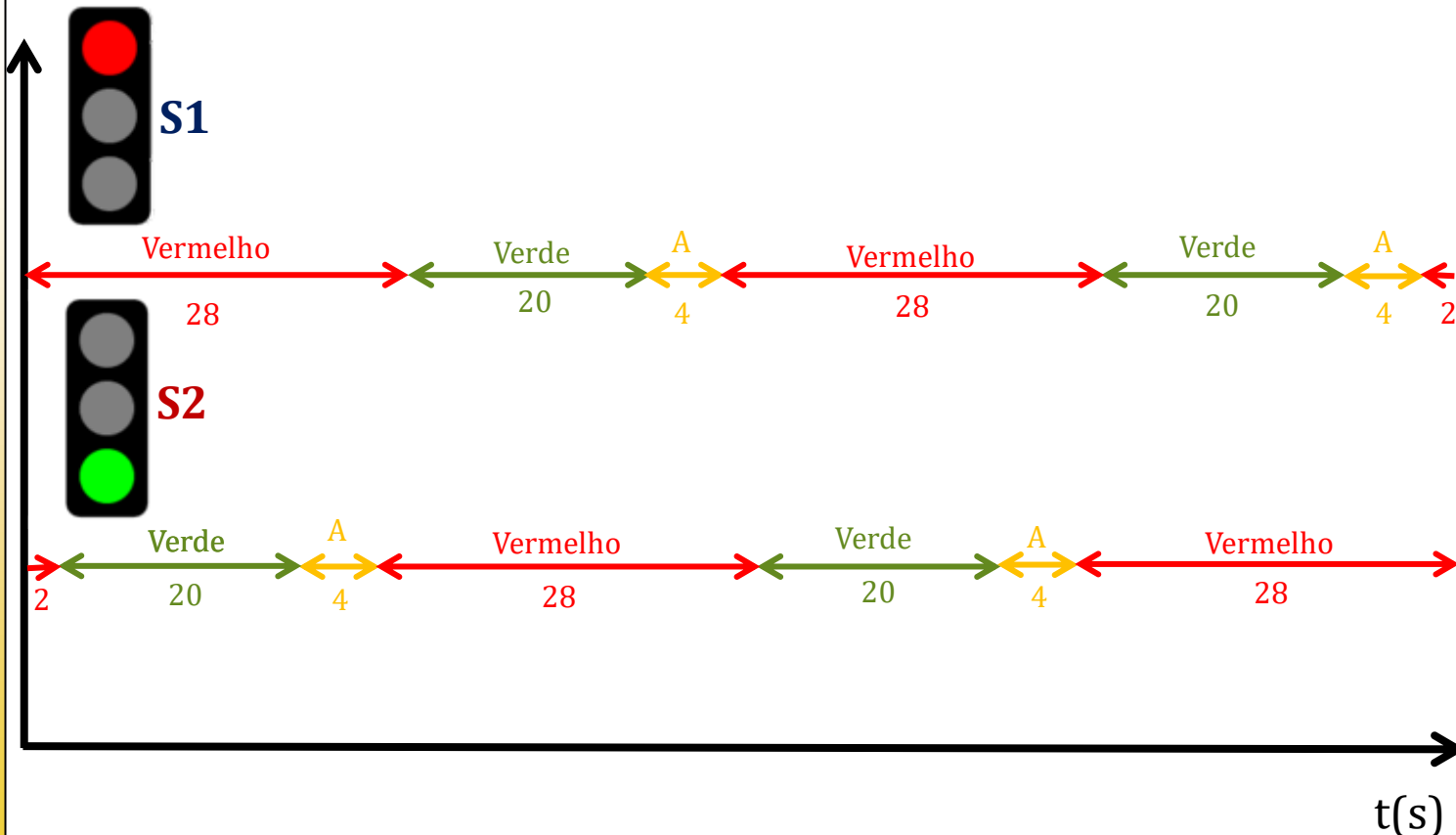




- Desenvolvimento
- Referências

Desenvolvimento:

Exemplo: Semáforo



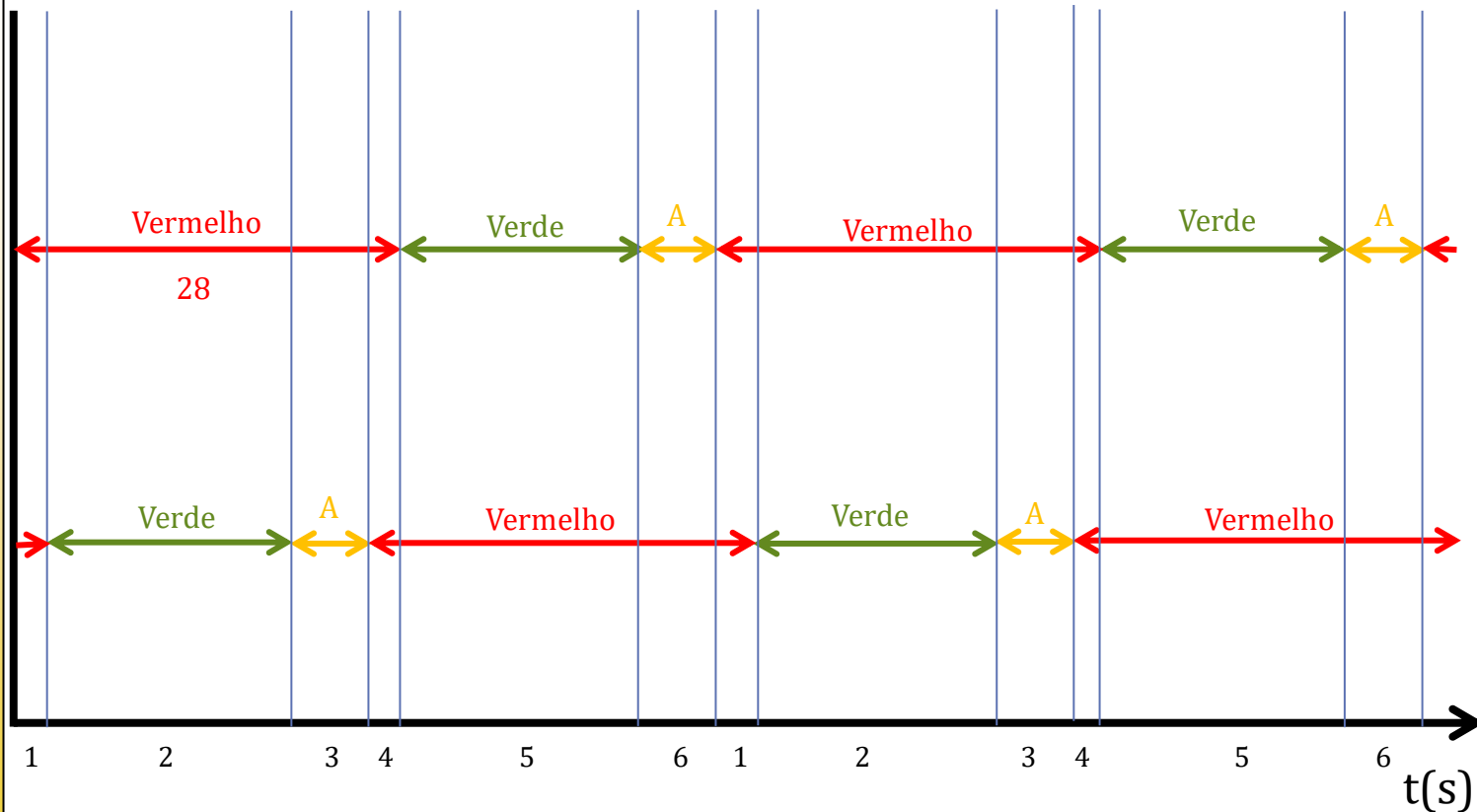


- Desenvolvimento

- Referências

Desenvolvimento:

Exemplo: Estados

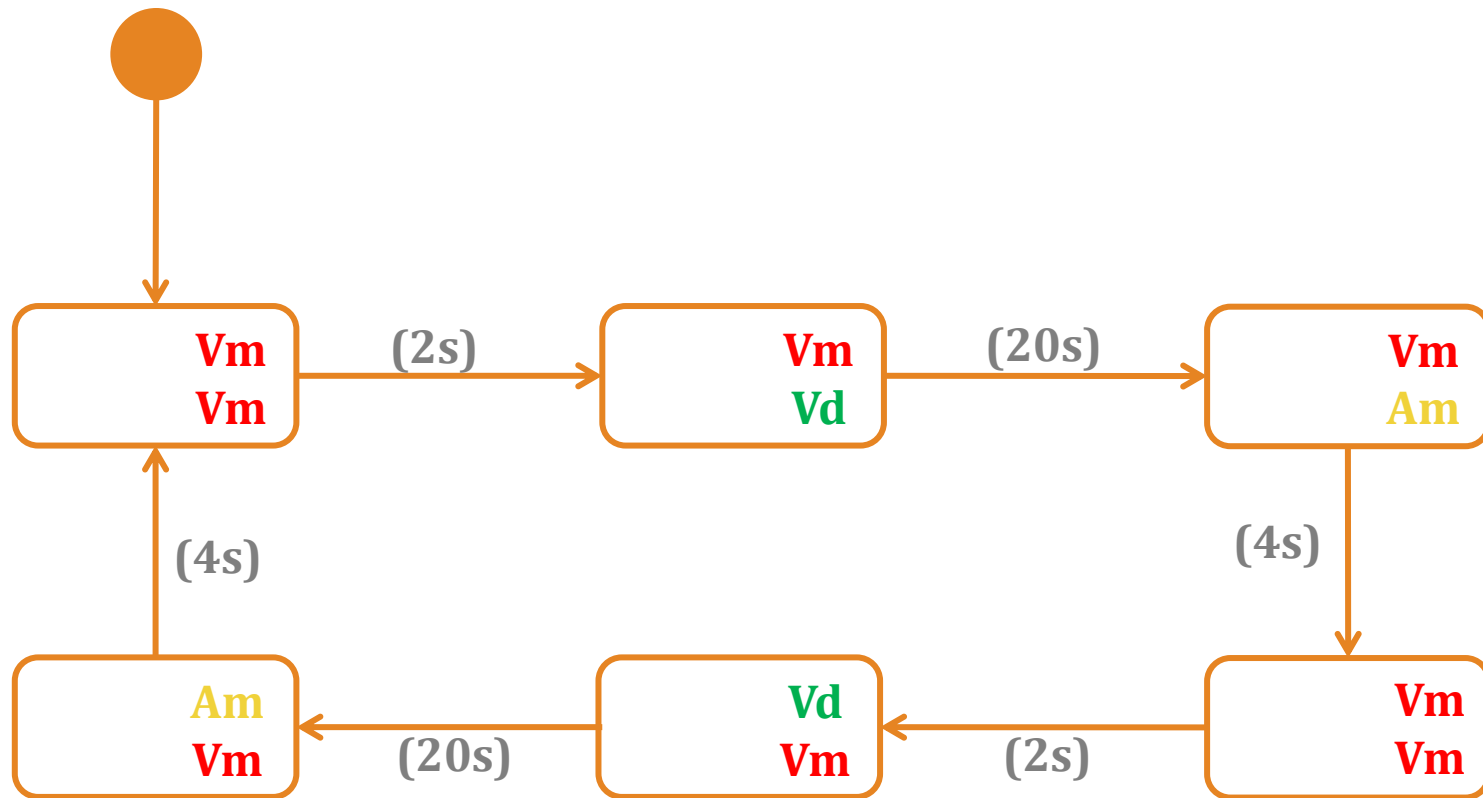




- Desenvolvimento
- Referências

Desenvolvimento:

Exemplo: Semáforo





- Desenvolvimento

- Referências

Desenvolvimento:

Extras:

- **Draw.io** (ferramenta de desenho na nuvem)
 - <https://www.draw.io/>
- **Yakindu Statechart Tools** (desenho + simulação)
 - <http://statecharts.org/>

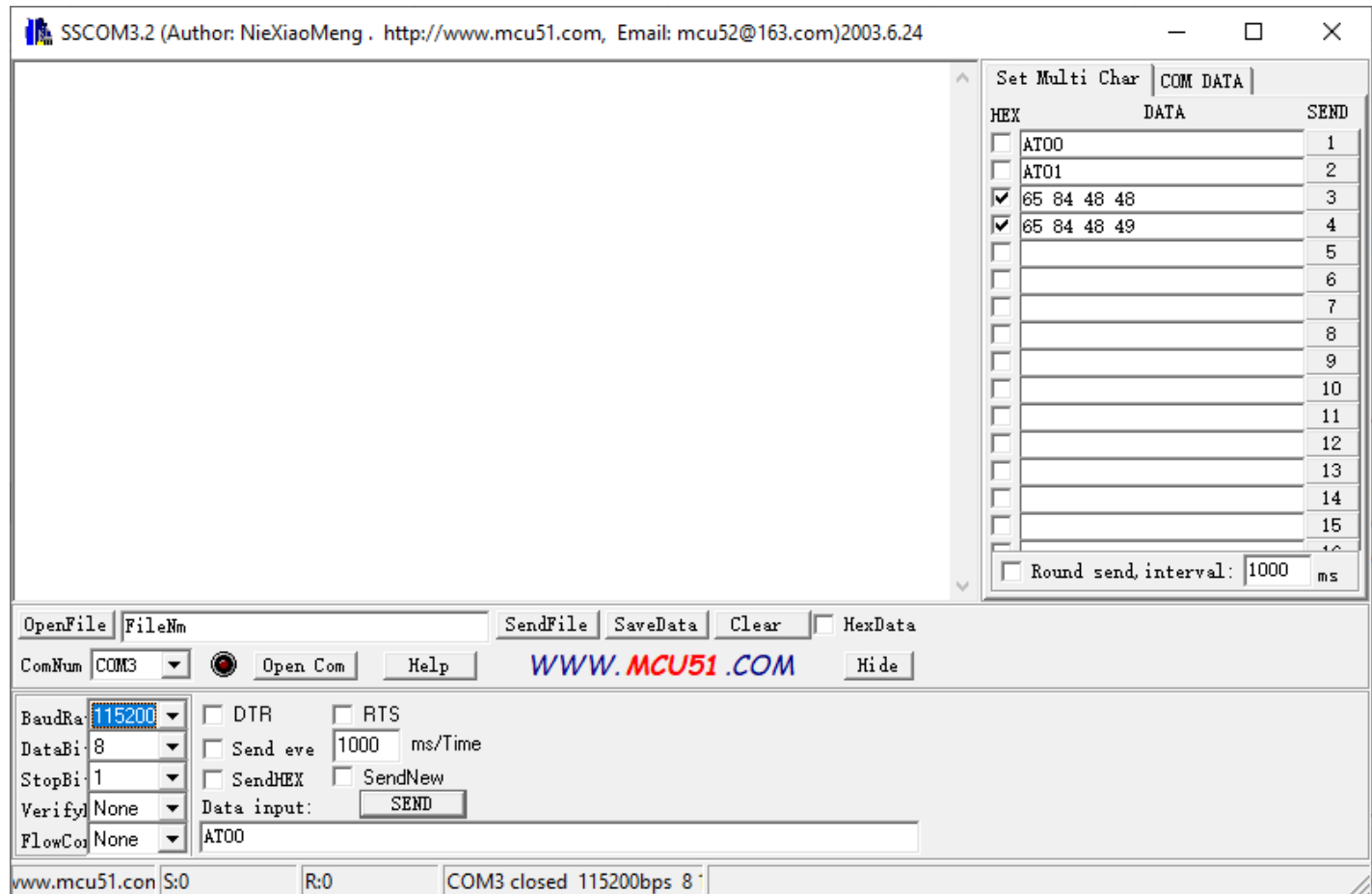




- Desenvolvimento
- Referências

Desenvolvimento:

SSCOM32 (Uart):

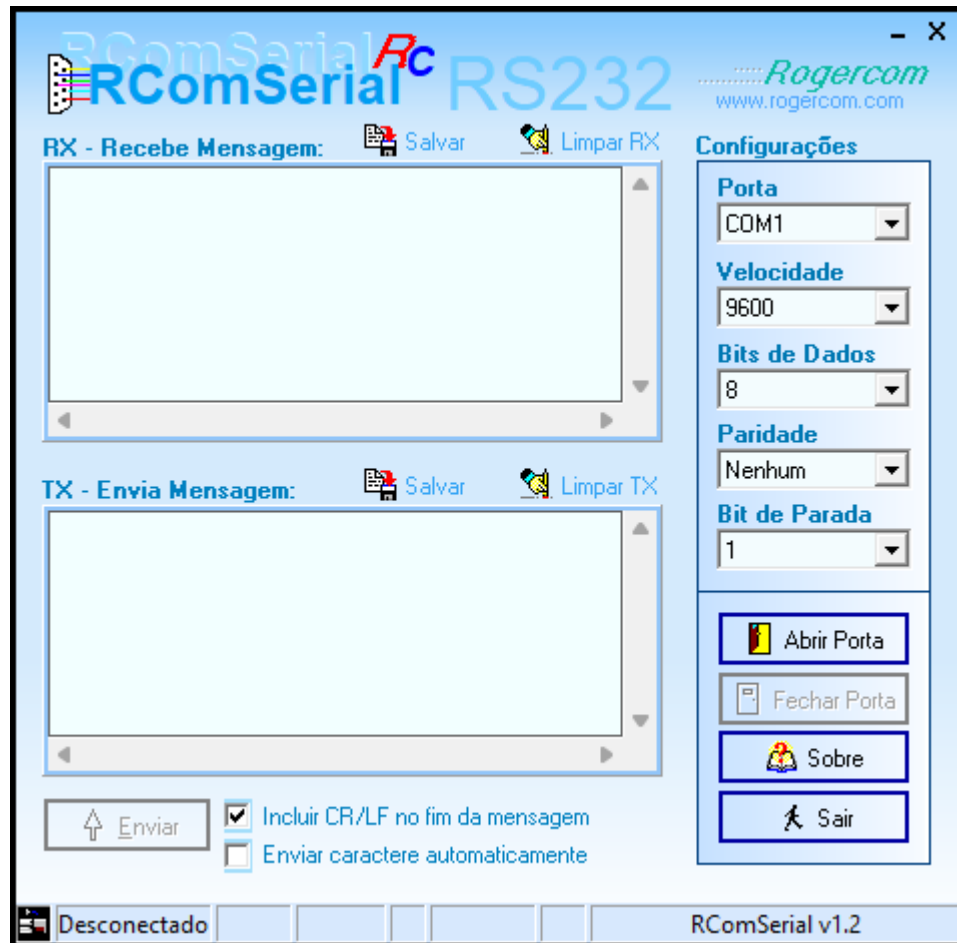




- Desenvolvimento
- Referências

Desenvolvimento:

RcomSerial (Uart):





- Desenvolvimento

- Referências

Referências:

Extras:

Laboratório 02:

http://www.elf74.daeln.com.br/Labs/Keil_Lab2.pdf

* Refs ↔ Renesas.com, Pixabay.com, wikimedia.org, flickr, community.arm.com, Undergraduated course Renesas / CWS71-Sistemas Embarcados (Prof. Douglas P. B. Renaux e Robson Linhares), ytchannel Gustavo W. Dernardin, *ARMv7-M Architecture Reference Manual*, CSW40-Sistemas Microcontrolados – Prof. Peron, toshiba.semicon-storage.com, microncontrollerslab.com, lfelectronics.com.br