

# yocto

## PROJECT

*Templates, tools and methods to create custom linux images*

# Sobre

- Fundado em 2010
  - Projeto colaborativo envolvendo diversos fabricantes de hardware, desenvolvedores de sistemas operacionais e fabricantes de eletrônicos.
  - Visa trazer ordem ao caos do desenvolvimento de sistemas operacionais Linux embarcados.
- O que é:
  - Projeto colaborativo de código aberto
  - Provê *templates*, ferramentas e métodos para o desenvolvimento de distribuições customizadas baseadas em Linux para sistemas embarcados.

# Pré-requisitos

- Placa Intel Galileo
- PC com Linux moderno 64-bits
- Conexão com a internet
- Aproximadamente 20GB de espaço livre em disco

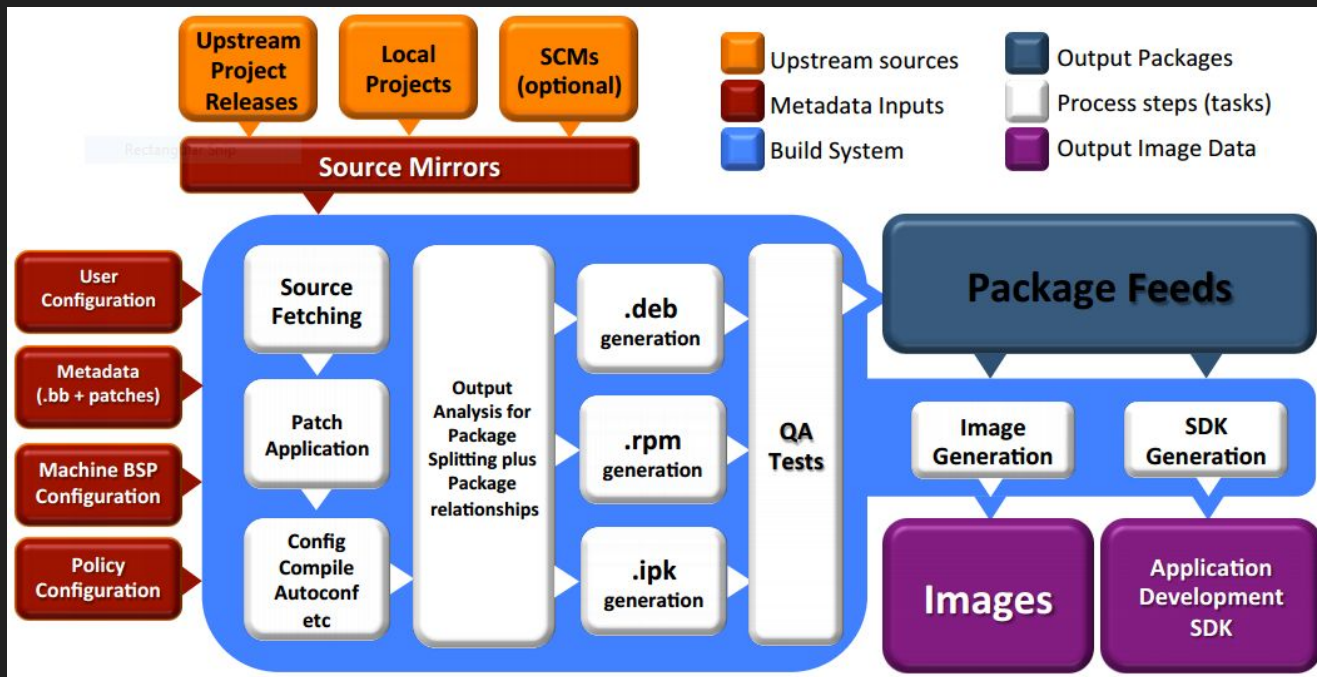
# Walk Through

1. Baixar código fonte da Yocto Project
2. Baixar pacotes adicionais à distribuição.
3. Configurar o ambiente de desenvolvimento para imagem a ser gerada
4. “Assar” os seus bits.
5. Empacotar a imagem em um arquivo para cartão uSD bootável

Passo-a-passo para construir a IoTdevkit:

<https://software.intel.com/en-us/blogs/2015/03/04/creating-a-yocto-image-for-the-intel-galileo-board-using-split-layers>

# Forno para Bits



# Board Support Package

- Um BSP é uma coleção de informações que definem como dar suporte a um *hardware* de um dispositivo particular, conjunto deles ou até uma plataforma.
- Inclui informação sobre funcionalidades presentes no *hardware* e informações de configuração do *kernel*, juntamente com *drivers* requisitados.

# Board Support Package

- BSP padrão que a Intel provê suporte é a Intel® IoT *Developer Kit*
- Possibilidade de customização através de "receitas" adicionadas aos pacotes padrões do BSP
  - Criar um arquivo *.bbappend*
  - Manter estrutura dos arquivos dentro do BSP *layer* (para máquina alvo)
  - Colocar os arquivos dentro de uma pasta com nome referente à máquina alvo (i.e. *recipes-bsp*, *recipes-graphics*, *recipes-core* e outras)
  - Colocar arquivos fonte do BSP nas pastas correspondentes

# *Open Embedded*

- *Framework* para a construção de Linux embarcado
- Ambiente de *Cross-Compilation*
- Adotado como sistema padrão do Yocto Project desde março de 2011
- Suporte a diversas arquiteturas
- Fácil de se customizar



# libc

- Desenvolver uma imagem utilizando outra libc além da uclibc (mais comum)

Passo a passo (não testado): <http://www.malinov.com/Home/sergey-s-blog/intelgalileo-buildinglinuximage>

Dúvidas?

# Referências Bibliográficas

1. YOCTO PROJECT. About. Disponível em: <https://www.yoctoproject.org/about>. Acessado em 30/11/2015
2. INTEL. Intel® Quark SoC X1000: *Board Support Package (BSP) Build Guide*. Disponível em: [http://download.intel.com/support/processors/quark/sb/quark\\_bspbuildguide\\_329687\\_001.pdf](http://download.intel.com/support/processors/quark/sb/quark_bspbuildguide_329687_001.pdf). Acessado em 30/11/2015
3. INTEL. CREATING A YOCTO IMAGE FOR THE INTEL® GALILEO BOARD USING SPLIT LAYERS. Disponível em: <https://software.intel.com/en-us/blogs/2015/03/04/creating-a-yocto-image-for-the-intel-galileo-board-using-split-layers>. Acessado em: 30/11/2015.
4. HALLINAN, C. Create a Custom Embedded Linux Distribution for Any Embedded Device Using the Yocto Project. Disponível em: <https://www.yoctoproject.org/sites/default/files/ypdd-oct2013-into-hands-on-lab.pdf>. Acessado em: 30/11/2015.

# Referências Bibliográficas

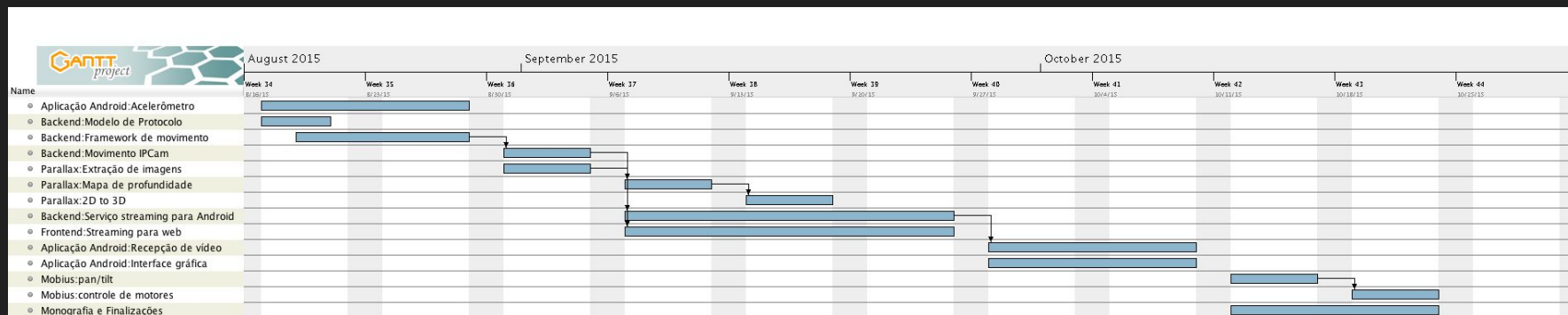
5. INTEL. Just Add Inspiration with the Intel® IoT Developer Kit. Disponível em: <https://software.intel.com/en-us/iot/hardware/devkit>
6. OPEN EMBEDDED. Welcome to OpenEmbedded. Disponível em: [http://www.openembedded.org/wiki/Main\\_Page](http://www.openembedded.org/wiki/Main_Page)

# Projeto - R3V

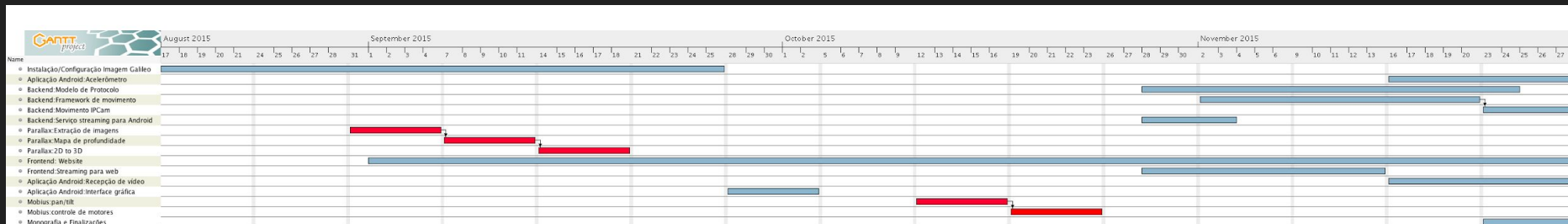
- Consiste de um sistema de visualização 3D baseado no movimento de um usuário conectado remotamente com o dispositivo da câmera



# Cronograma projetado



# Cronograma final



# Referência do projeto



MENDES, D.; RUSA, H. R3V - Remote 3D Viewer. Disponível em: <https://github.com/ddmendes/R3V-Remote3DViewer>