

# YOCTO

Mg. Ing. Gonzalo E. Sanchez  
MSE - 2022

**Implementación de Sistemas Operativos II**

# YOCTO

- Introducción

# Introducción

# YOCTO

- El slogan del proyecto YOCTO es:

**“ THE YOCTO PROJECT IS NOT AN EMBEDDED LINUX DISTRIBUTION, IT CREATES A CUSTOM ONE FOR YOU. ”**

- YOCTO es un proyecto open source que ayuda a los desarrolladores a construir sistemas basados en linux.

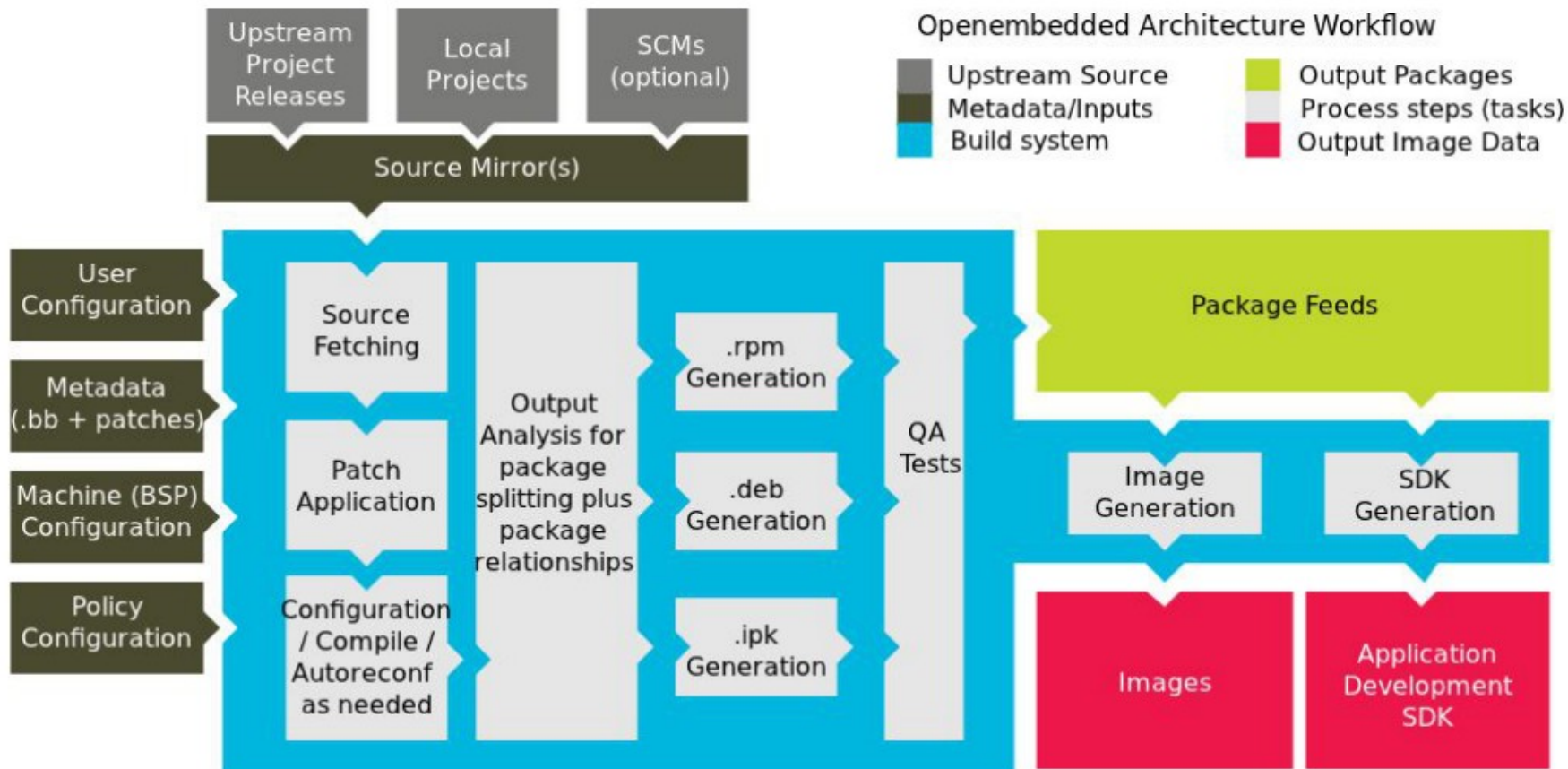
# YOCTO

- Diseñado para soportar diferentes plataformas de hardware y stacks de software de forma mantenible y escalable.
- Basado en *bitbake*, el cual es un sistema de ejecución para automatizar la creación del bootloader, kernel y rootfs.
- La creación de los archivos se hace a través de **recipes** (archivos .bb y .patch)
- Estos especifican toda la información de los paquetes y configuraciones correspondientes.

# YOCTO

- Las recetas se basan en un modelo de “capas”.
- Se define toda la configuración de hardware y software dentro de un BSP.
- Al compilar *bitbake* se encarga de obtener todas las dependencias y herramientas. para generar las imágenes.
- Al finalizar la compilación corre tests automatizados de QA sobre las unidades constitutivas.
- Esto garantiza que el resultado sea tal como fue estipulado en las recetas.

# YOCTO



# YOCTO

- **NOTA:** YOCTO es storage-intensive, se recomienda tener libres 50GB.
- Bitbake se diferencia de make en varios aspectos:
  - Lista de tareas globales para una compilación de alto nivel, leyendo todo el conjunto de archivos relacionados.
  - Make solo procesa un archivo a la vez, solucionando esto con inclusiones anidadas.
  - Bitbake admite mecanismos de herencia, permitiendo definiciones de operaciones comunes que luego se personalizan (custom).



# YOCTO

- Bitbake se diferencia de make en varios aspectos: (cont.)
  - Se considera que *bitbake* tiene una mayor flexibilidad que *make*, y es de más alto nivel.
  - Restricción: Mucha parte de *bitbake* está escrita en python.
- Algo fundamental en el proyecto YOCTO son los **metadatos**.

# YOCTO

- Los *metadatos* son una conjunción de datos que incluyen:
  - Instrucciones de compilacion (compiler, linker, archiver).
  - Comandos y datos indicando versiones de software a utilizar.
  - Patches y archivos auxiliares si hiciesen falta.
- La idea de los metadatos es que sean reutilizables en proyectos similares.

# BuildRoot

## HANDS ON

1. Descargar BuildRoot
2. Generar un sistema mínimo con BuildRoot



Gracias.

