



Linux embebido con Yocto  
y algunas buenas prácticas.

¿Por qué esta charla?

# Sobre mí

Programador C amante del software libre

[<alvaropg@gmail.com>](mailto:alvaropg@gmail.com)

<https://fosstodon.org/@alvaropg>

<https://alvaropg.org/>

Encuesta

# Qué es Yocto?

- Un proyecto libre con un conjunto de herramientas para la creación de sistemas Linux independientemente del hardware
- <https://www.yoctoproject.org/>

# Lo bonito de usar Yocto

- Es FOSS, hay comunidad
- Hay mucha documentación
- Hay mucho soporte de software y hardware
- Alguien habrá hecho eso antes, seguramente

# Partes/conceptos de Yocto

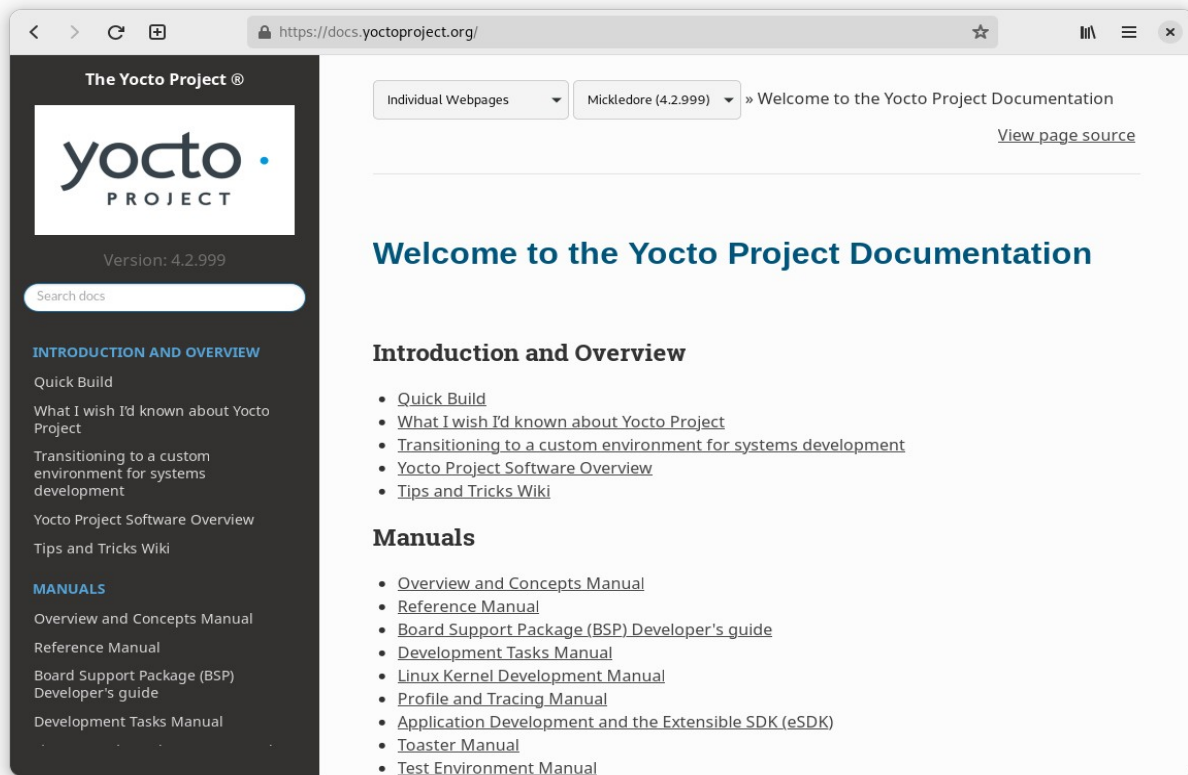
- Aplicaciones: bitbake, devtool
- Capas
- Recetas
- Machine
- Features
- Distro
- Cache

# Capas

- Hay muchas existentes  
<https://layers.openembedded.org/layerindex/branch/master/layers/>
- Separar soporte hardware y software o funcionalidades
- Son gratis...



# ¿Por dónde empezamos?



The screenshot shows a web browser window with the URL `https://docs.yoctoproject.org/`. The page features a dark sidebar on the left with the Yocto Project logo and navigation links. The main content area is white and contains a header with version information, a search bar, and a list of manuals.

**The Yocto Project**  
yocto PROJECT  
Version: 4.2.999  
Search docs

Individual Webpages Mickledore (4.2.999) » Welcome to the Yocto Project Documentation  
[View page source](#)

## Welcome to the Yocto Project Documentation

### Introduction and Overview

- [Quick Build](#)
- [What I wish I'd known about Yocto Project](#)
- [Transitioning to a custom environment for systems development](#)
- [Yocto Project Software Overview](#)
- [Tips and Tricks Wiki](#)

### Manuals

- [Overview and Concepts Manual](#)
- [Reference Manual](#)
- [Board Support Package \(BSP\) Developer's guide](#)
- [Development Tasks Manual](#)
- [Linux Kernel Development Manual](#)
- [Profile and Tracing Manual](#)
- [Application Development and the Extensible SDK \(eSDK\)](#)
- [Toaster Manual](#)
- [Test Environment Manual](#)

# KAS

- Mecanismo para generar un entorno definido y reproducible que:
  - Crear un contenedor
  - Despliegue capas
  - Configure entorno
  - Construir objetivos
- <https://kas.readthedocs.io/en/latest/index.html>

# KAS

```
alvaropg ~ sudo pip3 install kas
Collecting kas
  Downloading kas-3.2.3.tar.gz (33 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting PyYAML<6,>=3.0
  Downloading PyYAML-5.4.1.tar.gz (175 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 175.1/175.1 kB 4.5 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: distro >=1.0.0 in /usr/lib/python3.11/site-packages (from k
```

# KAS

```
1 header:
2   version: 11
3
4 repos:
5   # Mickledore (4.2)
6   poky:
7     url: https://git.yoctoproject.org/git/poky
8     refspec: 30d015c6362bce2bf6bbb2e60890431d8cd72403
9     layers:
10      meta:
11      meta-poky:
12      meta-yocto-bsp:
13
14 bblayers_conf_header:
15   meta-eslibre: |
16     POKY_BBLAYERS_CONF_VERSION = "2"
17     BBPATH = "${TOPDIR}"
18     BBFILES ?= ""
19
20 local_conf_header:
21   meta-eslibre: |
22     CONF_VERSION = "2"
23     PACKAGE_CLASSES ?= "package_rpm"
24     PATCHRESOLVE = "noop"
25     DL_DIR ?= "${TOPDIR}/downloads"
26     SSTATE_DIR ?= "${TOPDIR}/sstate-cache"
27     BB_HASHSERVE_UPSTREAM = "hashserv.yocto.io:8687"
28     SSTATE_MIRRORS ?= "file://.* http://sstate.yoctoproject.org/all/PATH;downloadfilename=PATH"
29     BB_DISKMON_DIRS ??= "\
30     STOPTASKS,${TMPDIR},1G,100K \
31     STOPTASKS,${DL_DIR},1G,100K \
32     STOPTASKS,${SSTATE_DIR},1G,100K \
33     STOPTASKS,/tmp,100M,100K \
34     HALT,${TMPDIR},100M,1K \
35     HALT,${DL_DIR},100M,1K \
36     HALT,${SSTATE_DIR},100M,1K \
37     HALT,/tmp,10M,1K"
38
39 machine: qemux86-64
40 distro: poky
41 target: core-image-minimal
```

# KAS

```
alvaropg ~ > work > eslibre > kas-container build meta-eslibre/kas.yml  
Trying to pull ghcr.io/siemens/kas/kas:3.2.3
```

# KAS

```
alvaropg ~ > work > eslibre > kas-container build meta-eslibre/kas.yml
2023-05-12 13:49:03 - INFO - kas 3.2.3 started
2023-05-12 13:49:03 - INFO - /repo$ git rev-parse --show-toplevel
2023-05-12 13:49:03 - INFO - /repo$ hg root
2023-05-12 13:49:03 - INFO - /repo$ git rev-parse --show-toplevel
2023-05-12 13:49:03 - INFO - /repo$ hg root
2023-05-12 13:49:03 - INFO - /work/poky$ git remote set-url origin https://git.yoctoproject.org/git/poky
2023-05-12 13:49:03 - INFO - /work/poky$ git cat-file -t 30d015c6362bce2bf6bbb2e60890431d8cd72403
2023-05-12 13:49:03 - INFO - Repository poky already contains 30d015c6362bce2bf6bbb2e60890431d8cd72403 as commit
2023-05-12 13:49:03 - INFO - /work/poky$ git status -s
2023-05-12 13:49:03 - INFO - /work/poky$ git rev-parse --verify -q origin/30d015c6362bce2bf6bbb2e60890431d8cd72403
2023-05-12 13:49:03 - INFO - /work/poky$ git checkout -q 30d015c6362bce2bf6bbb2e60890431d8cd72403
2023-05-12 13:49:03 - INFO - /work/poky$ /tmp/tmp30zwn4ri/get_bb_env /build
2023-05-12 13:49:03 - INFO - /build$ /work/poky/bitbake/bin/bitbake -c build core-image-minimal
Loading cache: 100% | ETA: --:--:--
Loaded 0 entries from dependency cache.
Parsing recipes: 100% |#####| Time: 0:00:19
Parsing of 899 .bb files complete (0 cached, 899 parsed). 1798 targets, 49 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION = "2.4.0"
BUILD_SYS = "x86_64-linux"
NATIVELSBSTRING = "debian-11"
TARGET_SYS = "x86_64-poky-linux"
MACHINE = "gemux86-64"
DISTRO = "poky"
DISTRO_VERSION = "4.2"
TUNE_FEATURES = "m64 core2"
TARGET_FPU = ""
meta
meta-poky
meta-yocto-bsp = "HEAD:30d015c6362bce2bf6bbb2e60890431d8cd72403"

NOTE: Fetching uninative binary shim http://downloads.yoctoproject.org/releases/uninative/3.9/x86_64-nativesdk-libc-3.9.tar.xz;sha256sum=3dd82c3fbbd59e87bf091c3eef555a05fae528eeda3083828f76cd4deaceca8b (will check PREMIRRORS first)
Initialising tasks: 100% |#####| Time: 0:03:28
Checking sstate mirror object availability: 100% |#####| Time: 0:00:24
Sstate summary: Wanted 1078 Local 0 Mirrors 1073 Missed 5 Current 0 (99% match, 0% complete)
NOTE: Executing Tasks
Setscene tasks: 847 of 1078
Currently 3 running tasks (281 of 2909) 9% |#####|
0: openssl-3.1.0-r0 do_package_write_rpm_setscene - 4m47s (pid 241) 59% |#####| 1.59M/s
1: gcc-12.2.0-r0 do_package_write_rpm_setscene - 4m13s (pid 3131) 41% |#####| 1.18M/s
2: llvm-native-15.0.7-r0 do_populate_sysroot_setscene - 47s (pid 16091) 95% |#####| 1.92M/s
```

# Shared State Cache

- Desde cero: empezamos de algo conocido.

```
NOTE: Fetching archive binary from http://downloads.yoctoproject.org/releases/archive/3.1  
Initialising tasks: 100% |#####  
Sstate summary: Wanted 1081 Local 0 Mirrors 0 Missed 1081 Current 0 (0% match, 0% complete)  
NOTE: Executing Tasks
```

- Con sstate-cache

```
Initialising tasks: 100% |#####  
Checking sstate mirror object availability: 100% |#####  
Sstate summary: Wanted 1081 Local 0 Mirrors 387 Missed 694 Current 0 (35% match, 0% complete)  
NOTE: Executing Tasks
```

# ccache

- `INHERIT += "ccache"`
- `CCACHE_DIR = "${TOPDIR}/../ccache"`



# Distribución o reproducibilidad

BB\_NO\_NETWORK="1"

# Cadena de suministro

- Permite generar la información de la cadena de suministro de nuestras imágenes.
  - Que incluye
  - De donde viene
  - Licencia
  - Es vulnerable
- <https://git.yoctoproject.org/meta-spdxscanner/>
- `$ bitbake --runall=spdx core-image-minimal`

# CVE

- Comprobación de vulnerabilidades
- INHERIT += "cve-check"
- <https://docs.yoctoproject.org/dev/dev-manual/vulnerabilities.html>

# ptests

- `DISTRO_FEATURES_append = " ptest"`
- `IMAGE_INSTALL_append = " ptest-runner"`

# Esto es todo

## Muchas gracias

### Veán “Yocto Live Coding”

[https://www.youtube.com/watch?v=ThTI4FItfMI&list=PLD4M5FoHz-TxMfBFrDKfIS\\_GLY25Qsfyj](https://www.youtube.com/watch?v=ThTI4FItfMI&list=PLD4M5FoHz-TxMfBFrDKfIS_GLY25Qsfyj)