

Implementación MPSoC FPGA de Algoritmos de *Machine Learning* para Aplicaciones Clínicas usando Metodologías de diseño de Alto Nivel

Estudiante: Mario Daniel Guanche Hernández <mario.guanche101@alu.ulpgc.es>

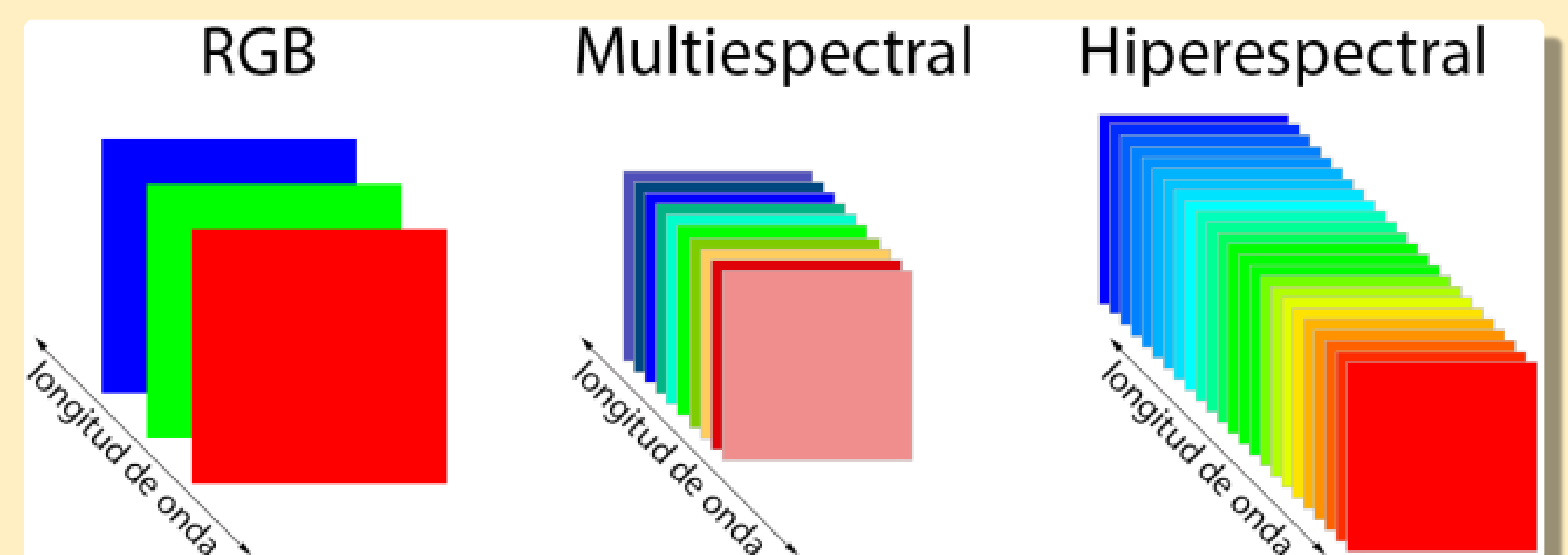
Tutores: Dr. Pedro Pérez Carballo <carballo@iuma.ulpgc.es> Sonia Raquel León Martín <slmartin@iuma.ulpgc.es>

Mayo 2023

Motivación

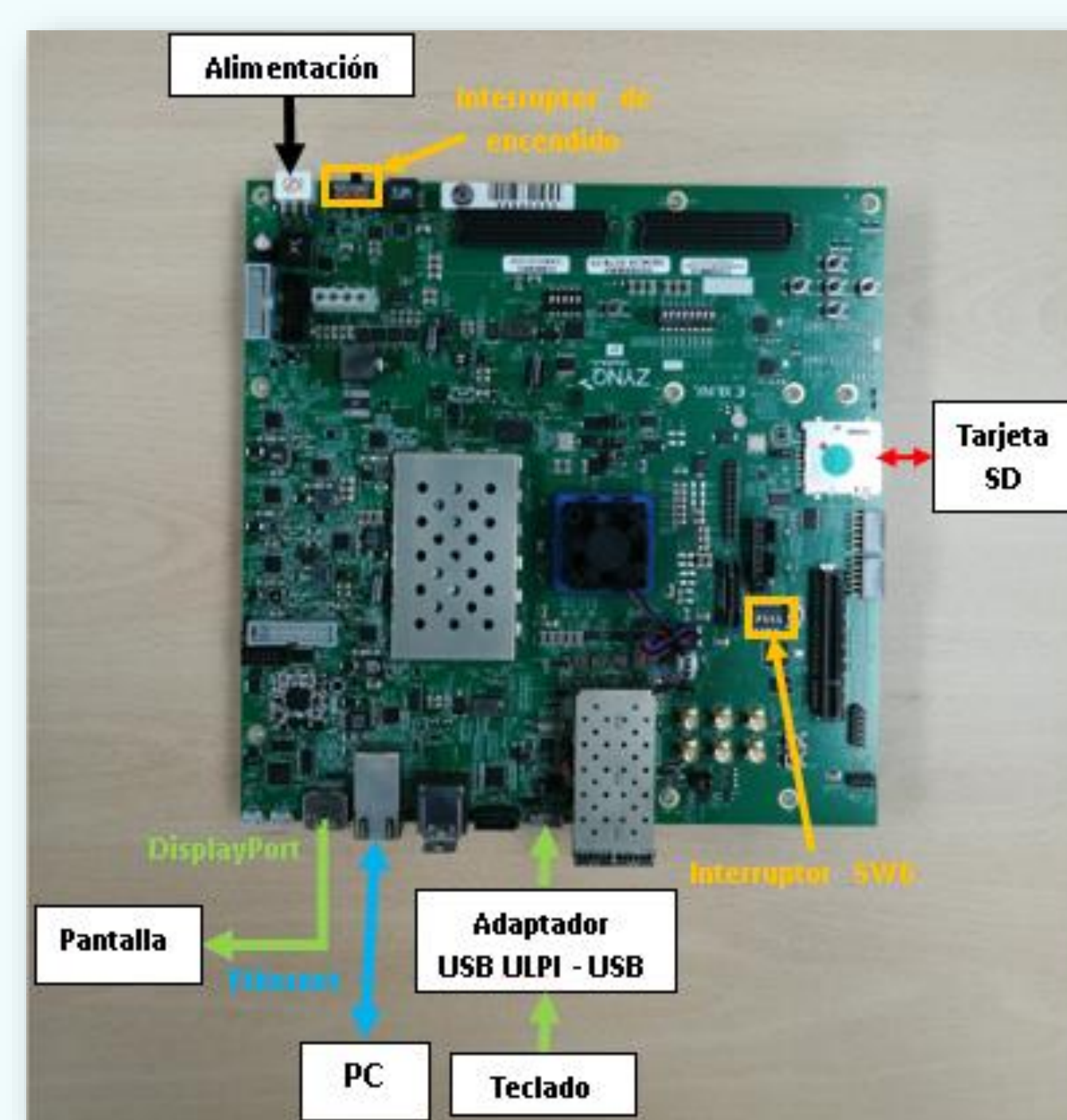
Aplicación para **detectar cáncer de piel** con **imágenes hiperespectrales**:

- Se identifica afección y distribución.
- Relación entre patología y coeficientes de absorción y reflexión espectrales.
- No invasiva y no ionizante.
- Análisis automatizable mediante *machine learning* (k-means).
- Muchos datos, tiempo y energía: necesidad de aceleración *hardware*.



Tecnología

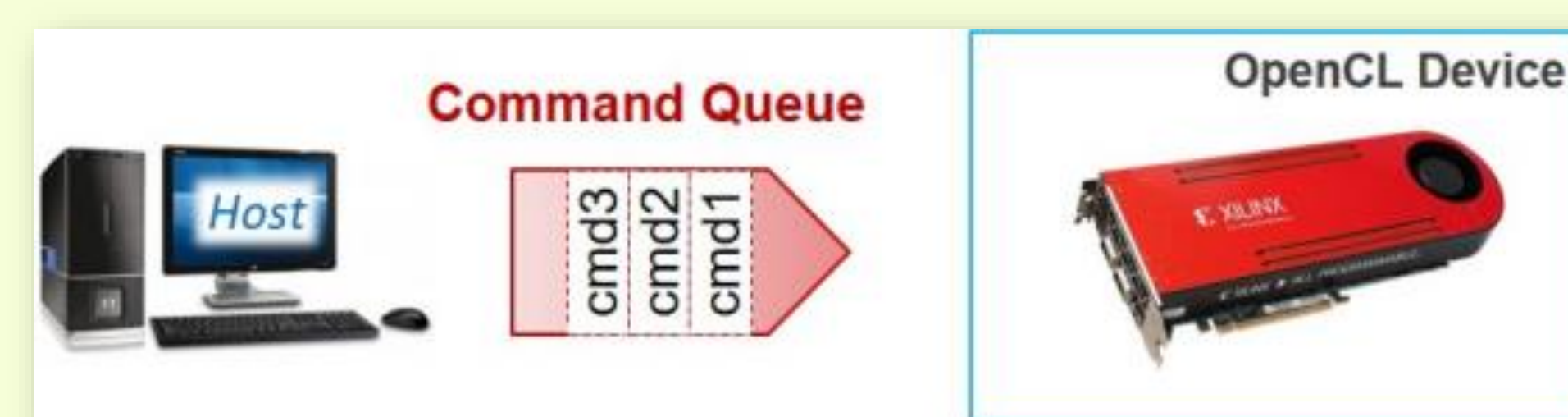
- Tipos de **aceleración hardware** contempladas.
 - GPUs (referencia).
 - FPGAs (implementada).
- Plataforma **Zynq Ultrascale+ MPSoC ZCU102**.
 - Implementación heterogénea
 - Sistema empotrado** MPSoC
 - Microprocesador **APU** como *host* o elemento central.
 - FPGA** como acelerador *hardware*.
 - Arranque mediante tarjeta SD



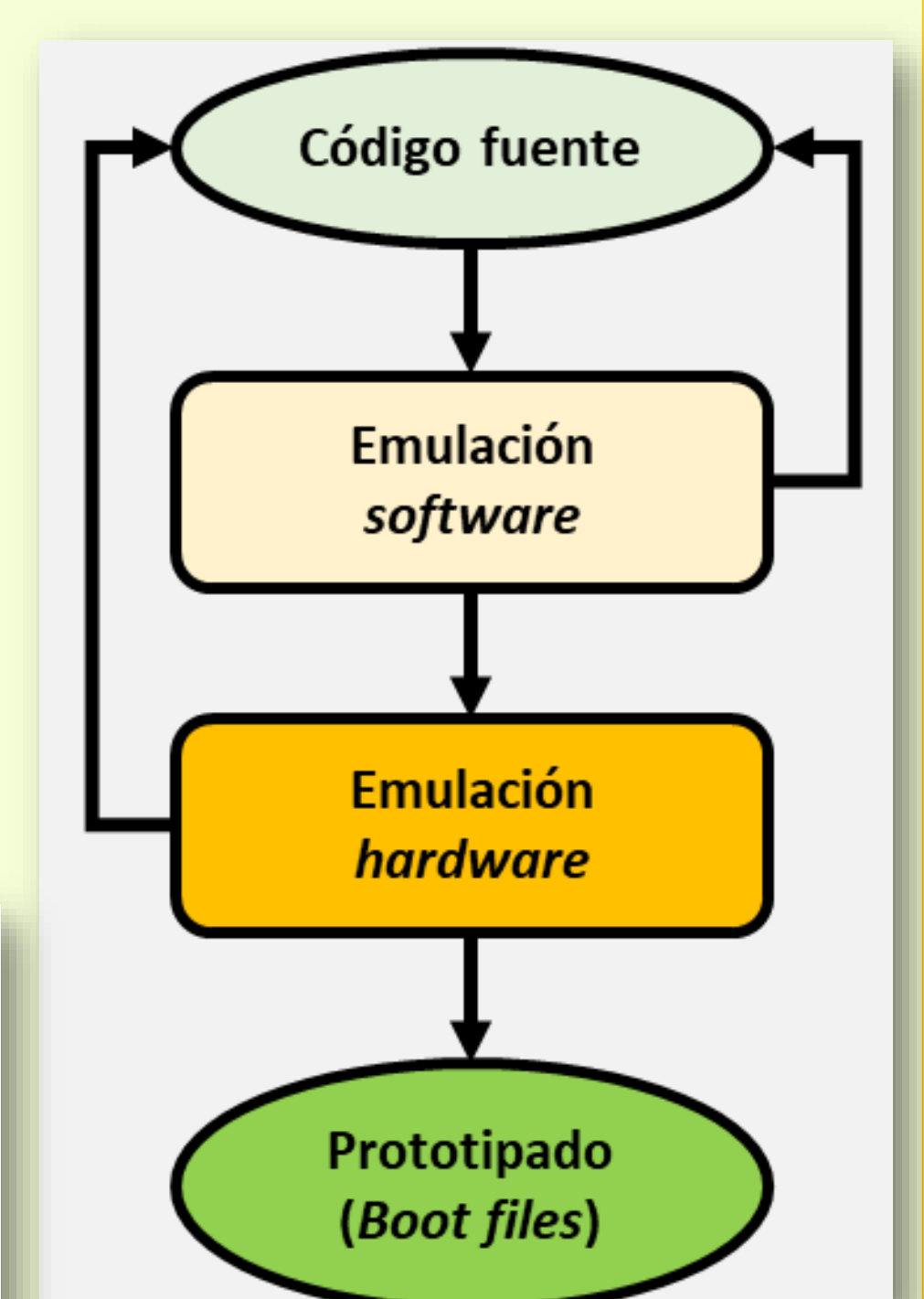
Metodología

Programación

- Host* en C/C++ y SO Linux.
- Kernels* FPGA en **Vitis HLS** (C/C++ y directivas `#pragma`).
- Interacción *host-kernels* mediante **OpenCL**



Flujo de diseño.



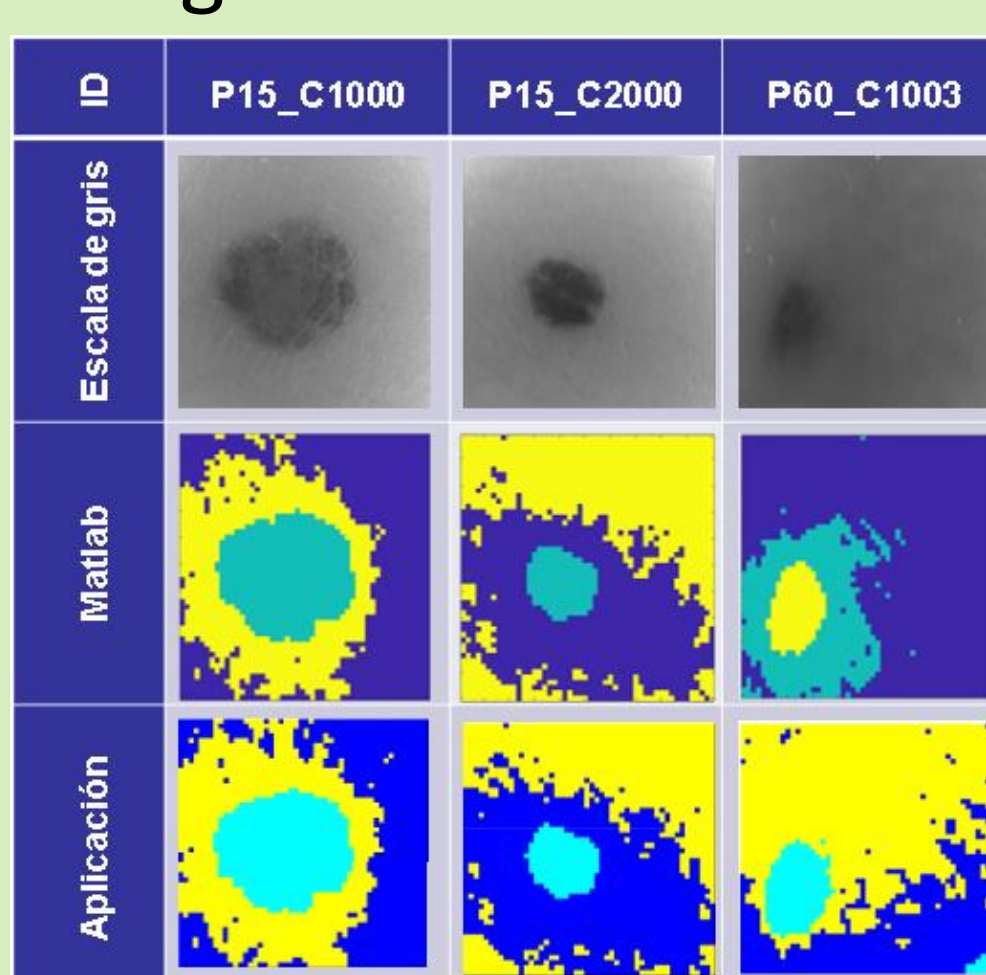
Diseño

- Solución **hardware/software** diseñada con especificaciones de **alto nivel: aproximación algorítmica**.
- Aplicación **host**: sincronización y transferencia de datos **OpenCL**.
- Aceleradores hardware**: diseño mediante Vitis HLS implementados como **kernels** FPGA
 - Filtrado de la imagen (multFilter).
 - Normalizado de la imagen (normalizers).
 - Clusterización de la imagen (top_kmeans).
 - Procesamiento en *streaming* de los *kernels* FPGA.
 - Configuración estática.
 - Tamaño de los datos (1, 2, 4 o 8 bytes).
 - Paralelismo de los *kernels* FPGA.
- Aplicación** invocada desde ventana de comandos **Linux**. Se especifica.
 - Criterio de ordenación (BSQ, BIL o BIP).
 - Dimensiones (ancho x largo x espectro).

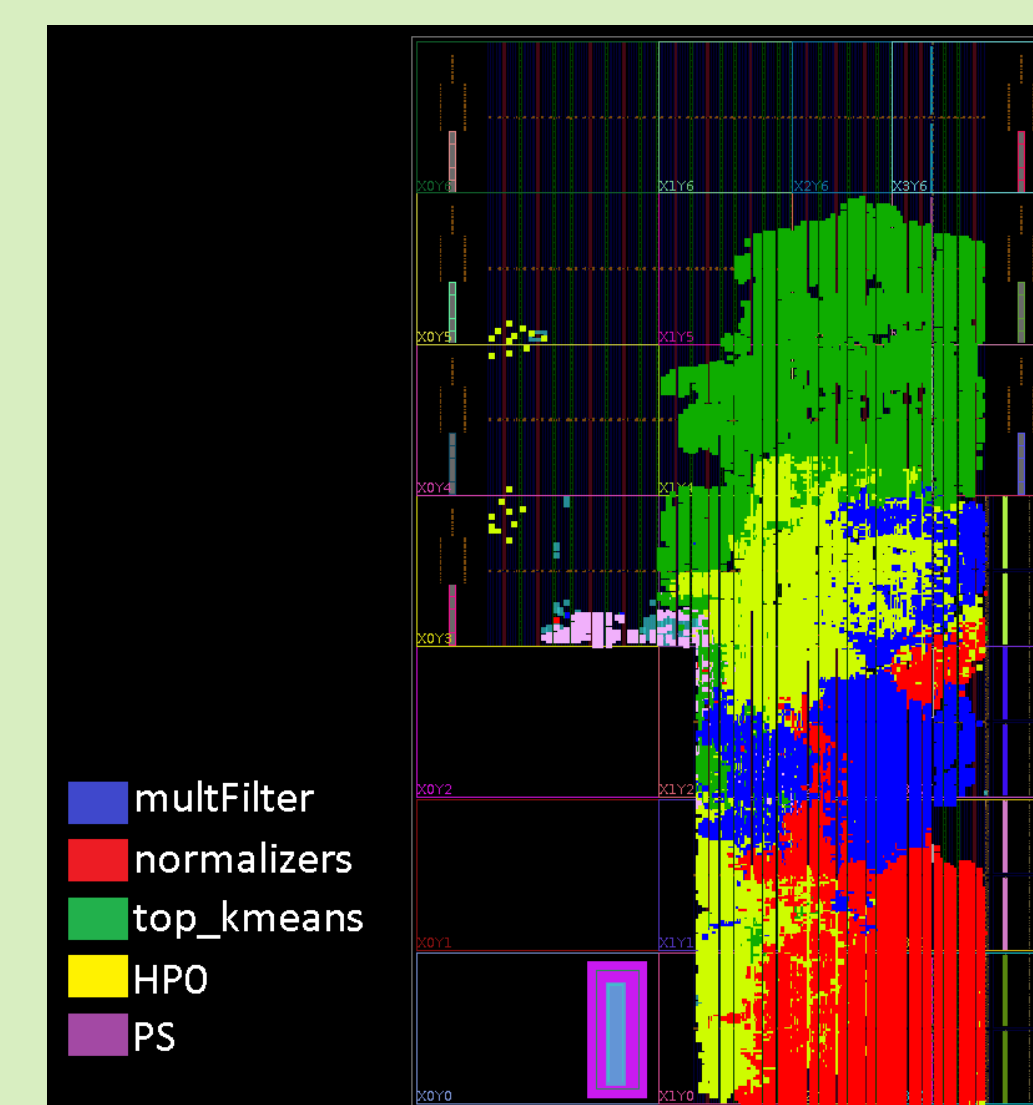


Resultados

Segmentación de las imágenes en 3 *clusters*.



- Utilización de la FPGA.
- Impacto significativo de las interconexiones de los *kernels* (HP0)

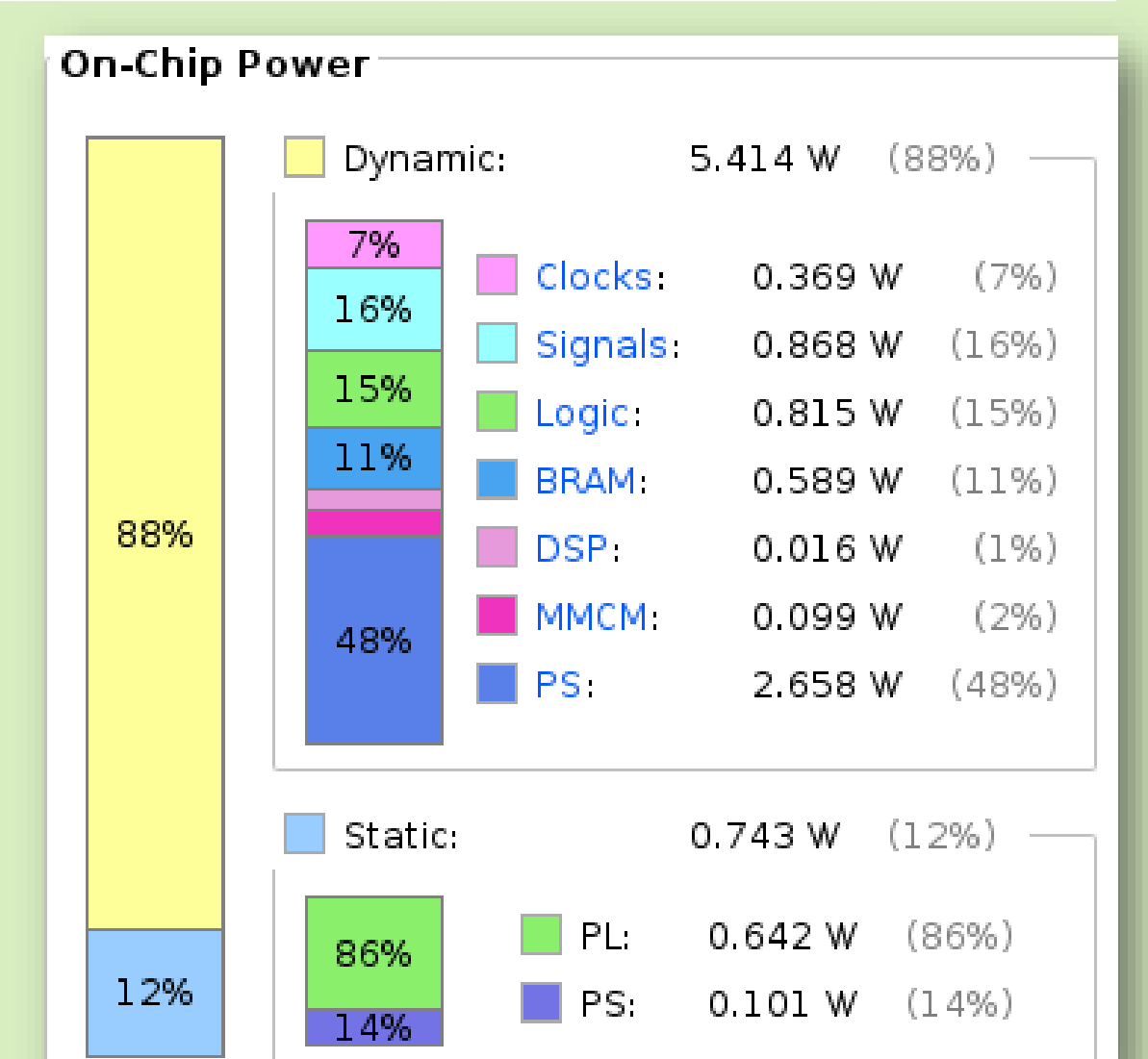


Mejora del **throughput**.

	multFilter (1 iter) (μs)	normalizers (1 iter) (μs)	top_kmeans (1 iter) (μs)	Total (s)
sw_emu	217190,00	220520,00	180500,00	48885,83
hw	95,56	97,32	90,43	22,77
Mejora (hw/sw_emu)	2272,8	2265,9	1996,0	2146,9
NVIDIA RTX 2080				0,8
NVIDIA Tesla K40				0,33

Consumo de potencia.

- 6,16 W.
- Mayoritariamente potencia dinámica
- Mejor relación entre carga computacional y energía en FPGA que en el PS (*host*)



Conclusiones

- Se **disciernen correctamente** las regiones según su grado de afección.
- La **aceleración hardware**.
 - ↑ **Velocidad** de cómputo (factor superior a 1000).
 - ↓ **Consumo energético**.

"EL FUTURO SE CREA
CON CADA PASO,
NO LO SUEÑES,
ALCANZA TU META"