

# GRID MÓVIL

## PARA PROCESAR IMÁGENES MÉDICAS

David Calle • Alfredo Santamaría • David Suárez

DIRECTORA

Ing. Mariela Curiel PhD.

MODALIDAD

Investigación



# OPORTUNIDAD

- Gran cantidad de dispositivos ociosos.
- Características de dispositivos móviles cada vez mejores.

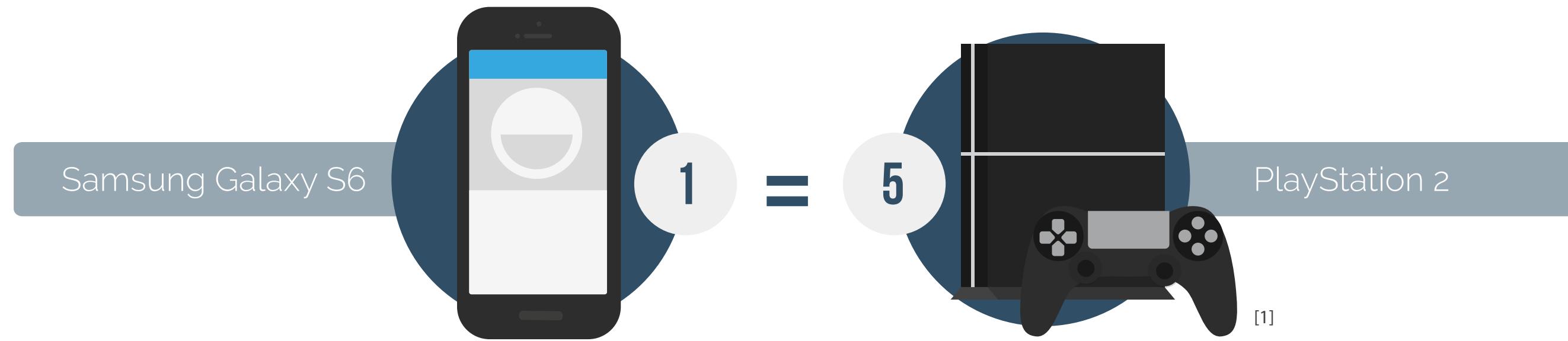
Conectividad

CPU

Almacenamiento

- La mayor parte de este tiempo están encendidos y utilizando únicamente una pequeña fracción del poder de cómputo.
- Ubicuidad, pocas restricciones geográficas.

# OPORTUNIDAD



Samsung Galaxy S6

1

=

5

PlayStation 2

[1]

34.8

GFLOPS

6.2 (GPU)

1.5 GHz quad-core  
2.1 GHz quad-core

CPU Speed

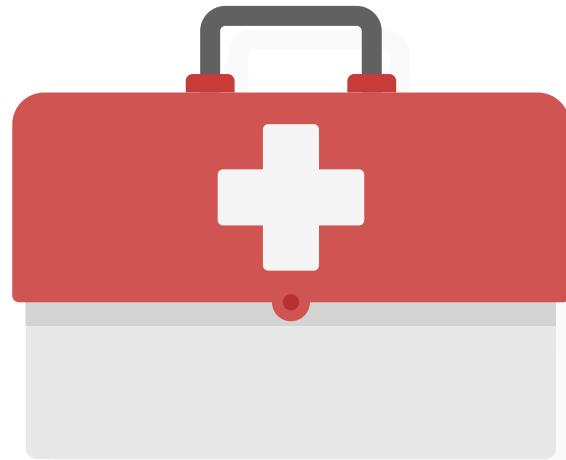
0.3 GHz single-core

3 GB

RAM

32 MB

# PROBLEMÁTICA IMÁGENES MÉDICAS



Toma de decisiones y  
diagnóstico.



Requieren gran cantidad  
de recursos.

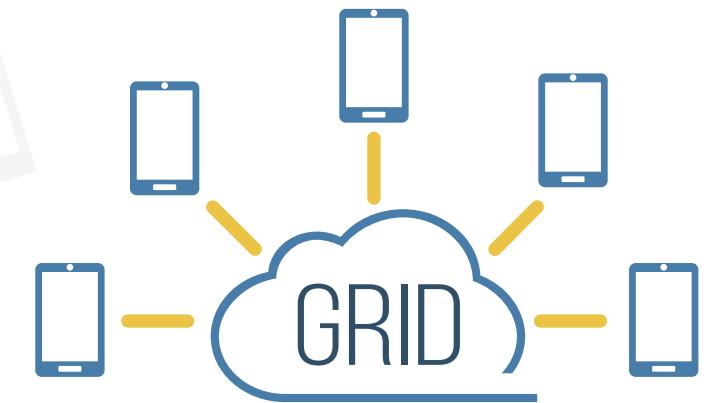


Sobre Imágenes de  
gran tamaño.



ITK  
C++

# GRID MÓVIL



- Arquitectura de hardware y software.
- Integra máquinas conectadas por red.
- Sistema distribuido para uso compartido de recursos.



- Plataforma de código abierto.
- Desarrollada por U. C. Berkeley.
- Permite crear proyectos que operan con recursos de computación públicos.

# PROBLEMÁTICA



¿Es posible procesar imágenes médicas en dispositivos móviles utilizando Boinc?

Ejecución de algoritmos sin modificar

ITK en dispositivos móviles

# DESCRIPCIÓN DEL PROYECTO



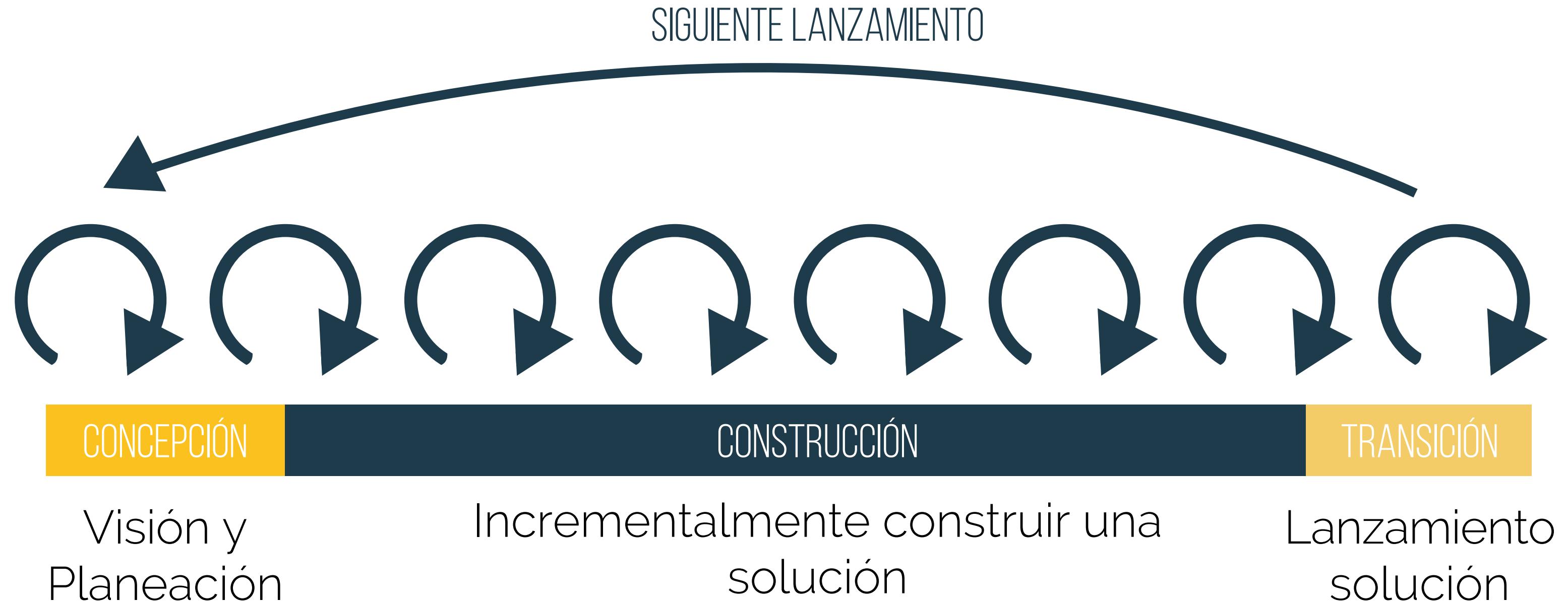
## OBJETIVO GENERAL

Extender Boinc para ejecutar algoritmos, previamente implementados, que procesan imágenes médicas basados en la estrategia dividir y conquistar.

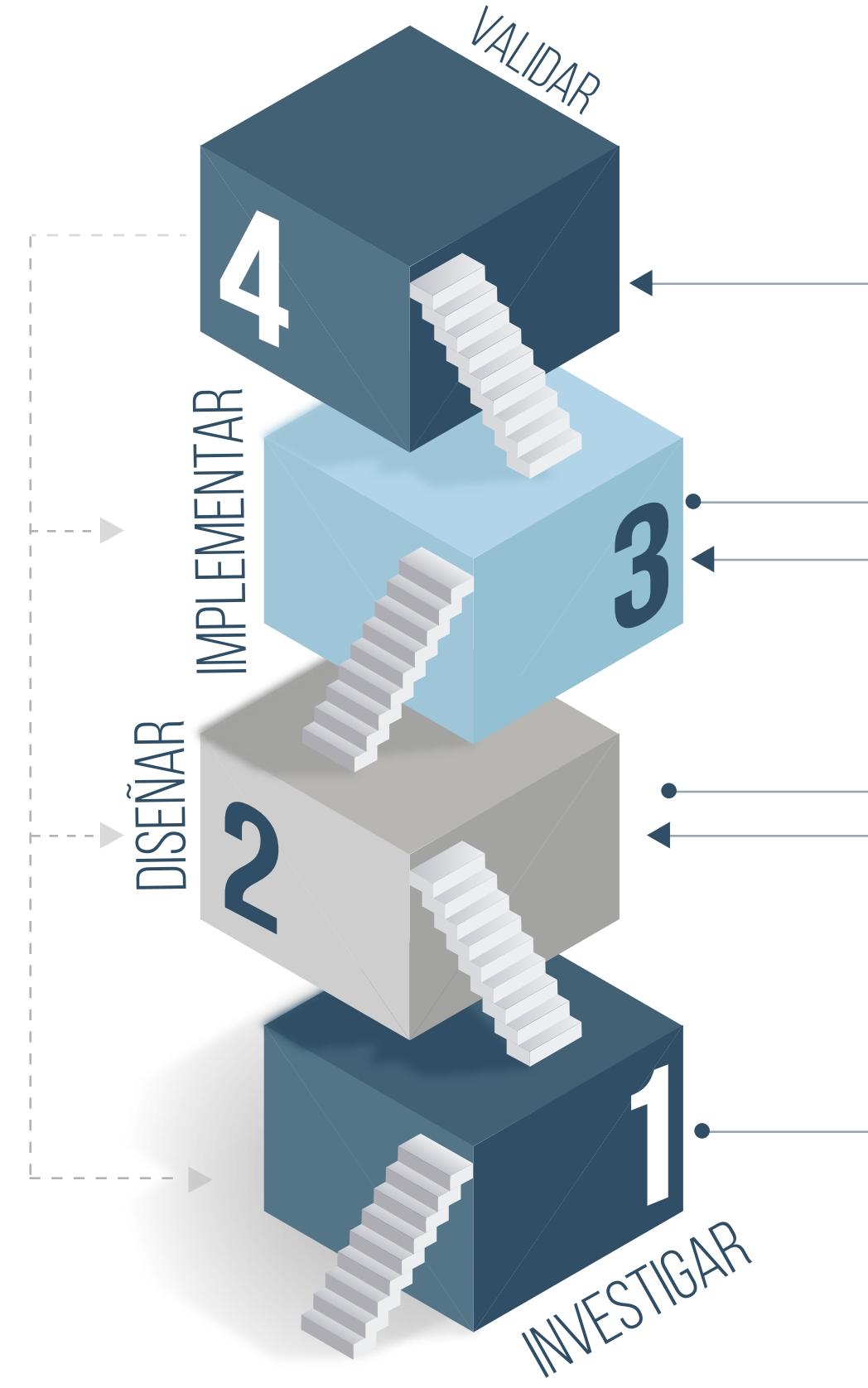
## OBJETIVOS ESPECÍFICOS

- Familiarizarse con el uso y la configuración de Boinc.
- Diseñar una solución para extender Boinc permitiendo adaptar algoritmos con características específicas.
- Implementar la solución diseñada para extender Boinc.
- Ejecutar en Boinc algoritmos que utilizan librerías de procesamiento de imágenes médicas.
- Validar la solución propuesta.

# CICLO DE VIDA-DAD

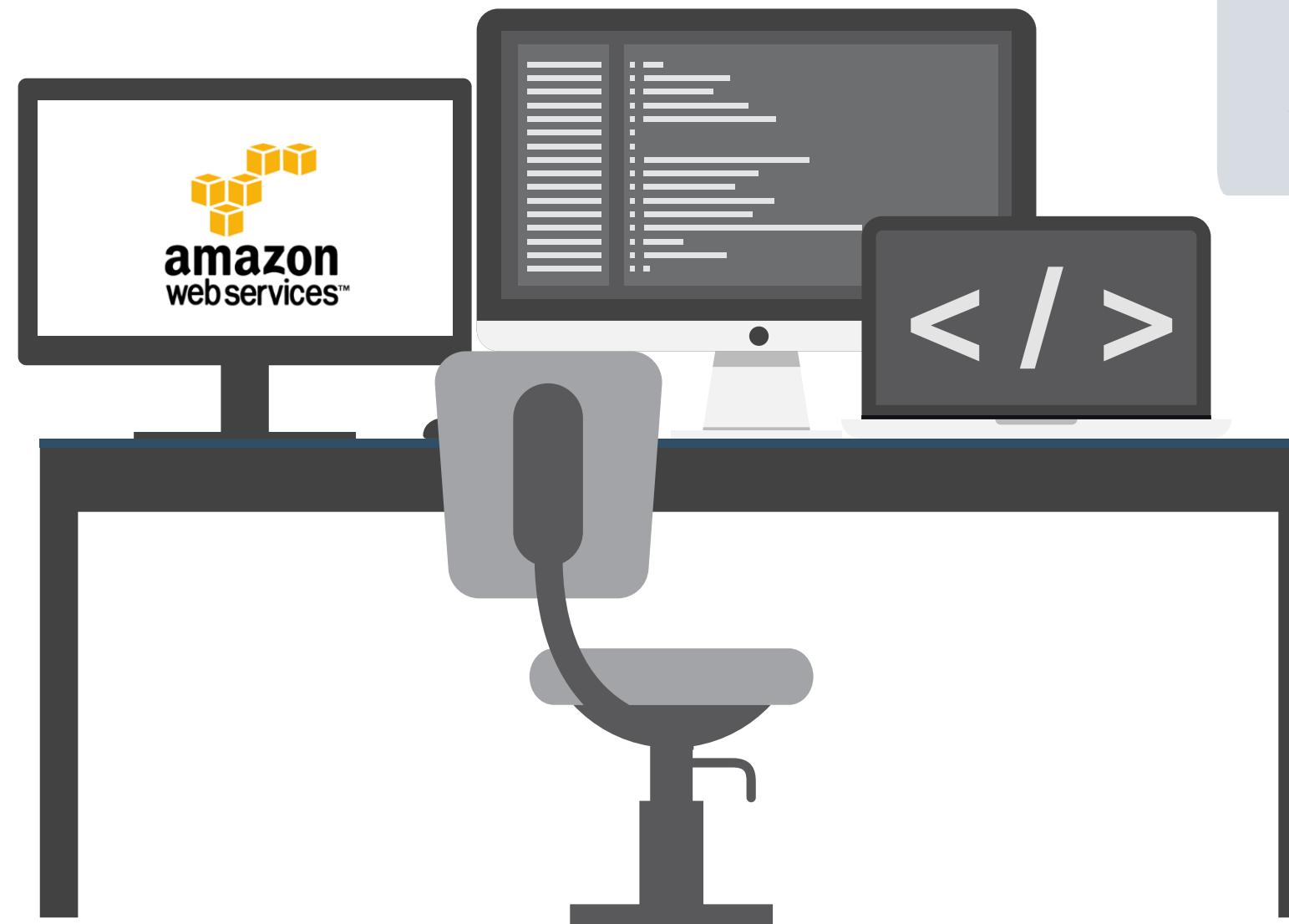


# DAD ADAPTADO



# FAMILIARIZACIÓN CON BOINC

Creación de una aplicación  
Hola Mundo para Linux,  
usando el wrapper compilado  
que ofrece la plataforma.



Despliegue de un  
servidor Boinc

Procesamiento de  
imagen con ITK,  
wrapper y Boinc.

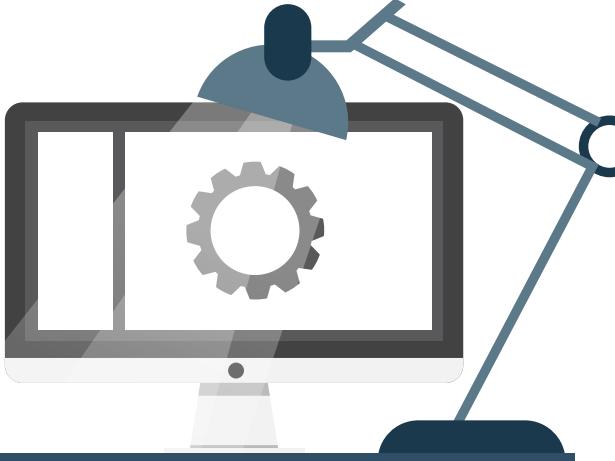


# WRAPPER

Permite que dos componentes se comuniquen entre sí sin que ninguno conozca los detalles de implementación del otro.

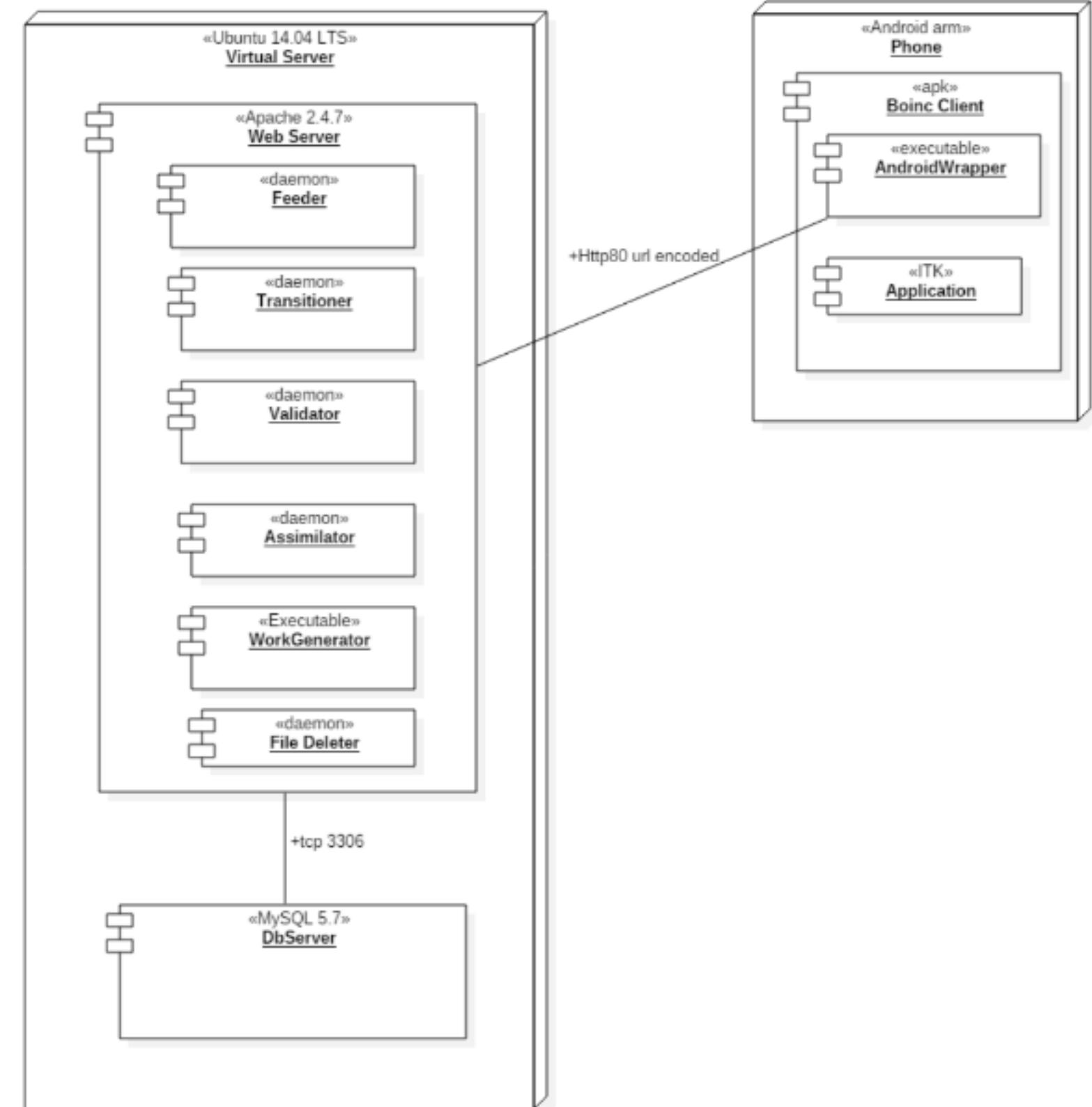


Se puede entender como el patrón adaptador.



# ARQUITECTURA BOINC

- Feeder
- Transitioner
- Validator
- Assimilator
- WorkGenerator
- File Deleter
- Wrapper



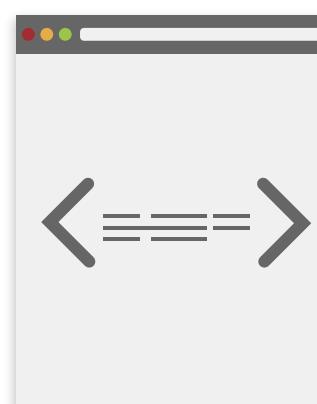
# FAMILIARIZACIÓN CON BOINC

## APRENDIZAJE



Despliegue y administración  
de un servidor Boinc  
complicado

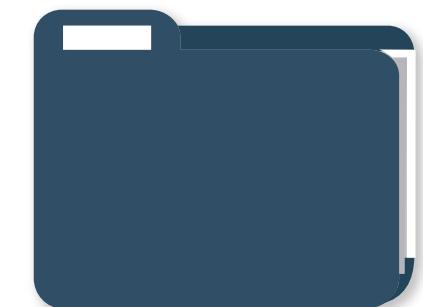
- 1 mes de un administrador de servidores experimentado
- 1 mes de un desarrollador web
- 1 mes de un programador



El wrapper permite integrar  
algoritmos previamente  
implementados sin modificar  
su código fuente  
Es necesario crear un XML  
adicional.



Boinc soporta múltiples  
plataformas  
Condición: Se debe proveer  
la versión compilada  
para la plataforma destino.



Es necesario compilar las  
aplicaciones sin  
carga dinámica  
de librerías.

# WRAPPER



PROBLEMÁTICA

Ejecución de algoritmos  
sin modificar

ITK en dispositivos  
móviles

# MATRIZ DAR

CARACTERÍSTICAS	CONSTRUIR WRAPPER	MODIFICAR WRAPPER EXISTENTE	GENERADOR DE CÓDIGO
Facilidad de uso	3	3	4
Curva de aprendizaje	1	2	2
Modificación de Algoritmo	4	4	1
Duración de implementación	1	2	1
TOTAL	9	11	8



# WRAPPER PARA ANDROID

Boinc provee versiones  
compiladas del wrapper para varias  
plataformas  
  
No incluye ARM

- Compilación para ARM
  - Proceso de construcción
  - Compilación dependencias
  - Compilación Estática, PIE
  - Modificación del código fuente

Validación  
Ejecución de Hola Mundo  
BOINC  
Wrapper

# Fixed wrapper execution time error in android #1671

[Edit](#)**Merged**

davidpanderson merged 1 commit into BOINC:master from DAD-Grid:android-wrapper 28 minutes ago

Conversation 1

Commits 1

Files changed 1

+3 -2



davidfcalle commented on Oct 9



Android wrapper had an execution time error because de env variable \$PWD was not found. Changing getenv("\$PWD") for getcwd() fixes the issue.

Commits 1 Fixed execution time error in android

f6001b3

Commits 1 davidfcalle changed the title from **Fixed execution time error in android** to **Fixed execution time error in android wrapper** 10 days ago

Commits 1 davidfcalle changed the title from **Fixed execution time error in android wrapper** to **Fixed wrapper execution time error in android** 10 days ago

Commits 1 davidpanderson merged commit **48f1720** into BOINC:master 28 minutes ago [View details](#) [Revert](#)  
1 check passed



davidpanderson commented 28 minutes ago

Collaborator



Thanks.

Projects

None yet

Labels

None yet

Milestone

No milestone

Assignees

No one assigned

2 participants



Notifications

[Unsubscribe](#)

You're receiving notifications because you authored the thread.

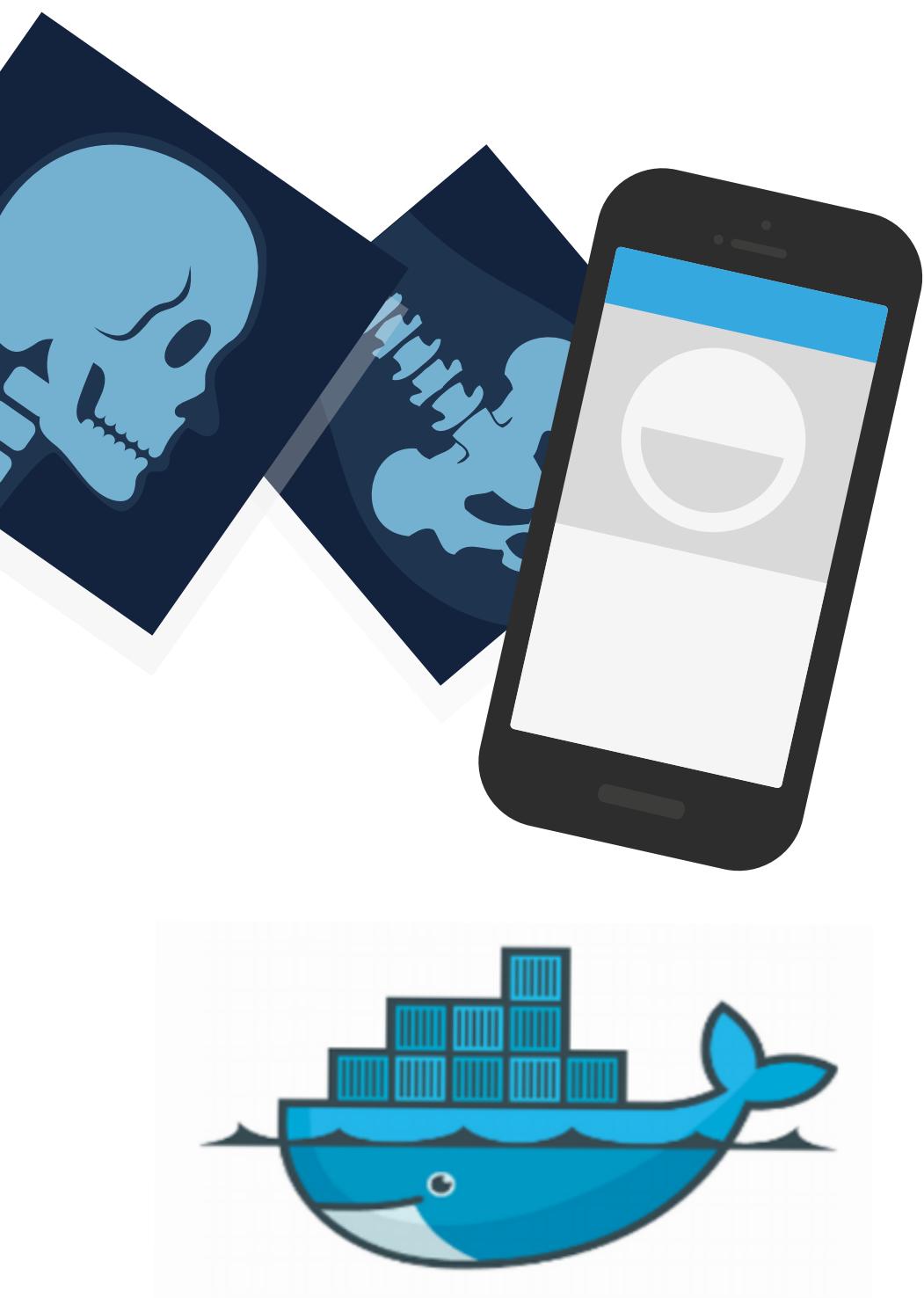
# PROCESAMIENTO IMÁGENES MÉDICAS



## PROBLEMÁTICA

Ejecución de algoritmos  
sin modificar

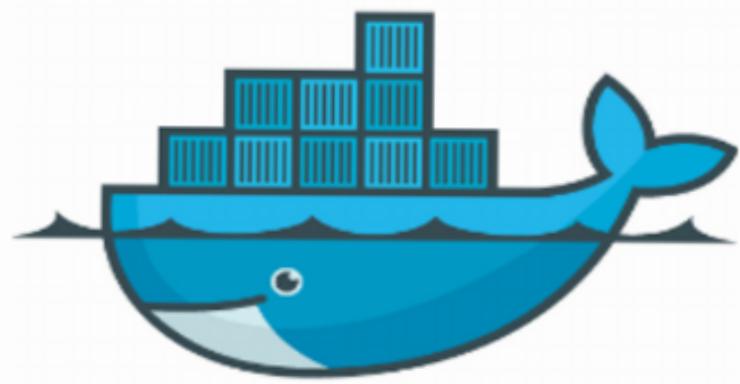
ITK en dispositivos  
móviles



# PROCESAMIENTO DE IMÁGENES EN CELULARES

## ITK

- Soporte multiplataforma (CMake)
- Está dividida por módulos (Core, Filtering, Registration, etc).



## Compilación cruzada para ARM

- Error de construcción (validar variables en plataforma destino).
- Algunos módulos necesitan generación de ejecutables intermedios.
- Se decidió construir los módulos Core y Filtering.
- No deben cargar dinámicamente la librería.

# PROCESAMIENTO DE IMÁGENES EN CELULARES

## Elección algoritmo

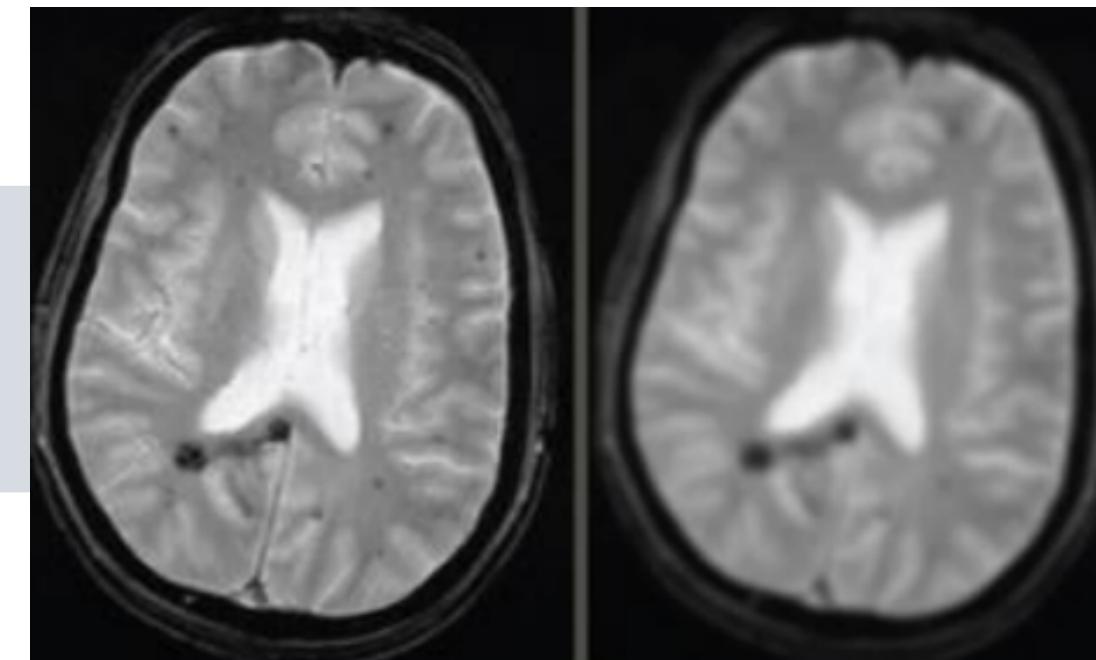
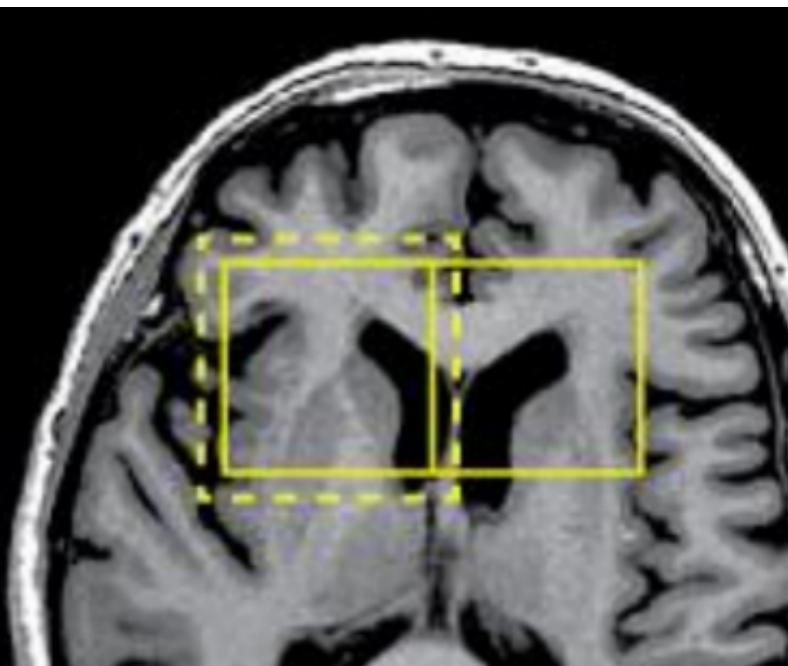
Filtrado (atenuar o resaltar características) / Dominio Frecuencia o Espacio

Estrategia dividir y conquistar (SIMD)

Información de pixeles vecinos

SmoothingRecursiveGaussianImageFilter / Difuminar

## Validación



# INTEGRACIÓN BOINC



## PROBLEMÁTICA

Ejecución de algoritmos  
sin modificar

ITK en dispositivos  
móviles

Procesamiento paralelo de  
imágenes médicas

# INTEGRACIÓN CON BOINC



## Problemática

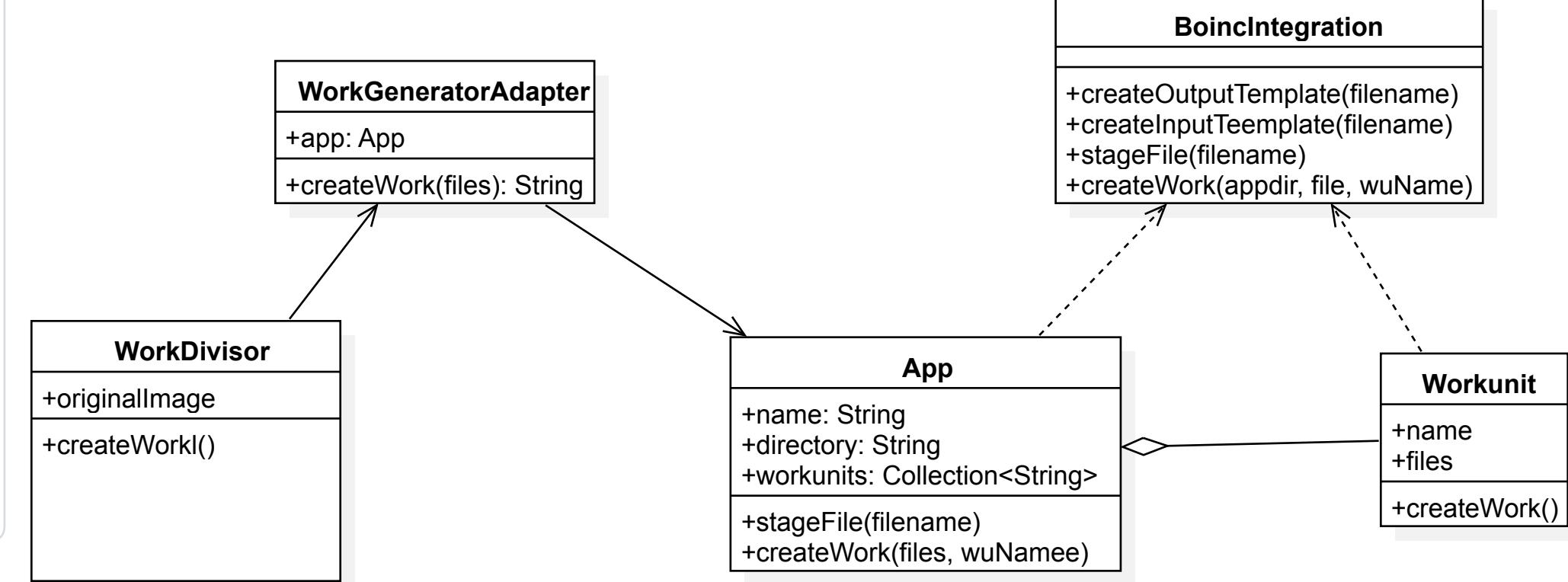
- Boinc por sí solo no provee mecanismos para generar trabajos automáticamente, tampoco para unir los resultados.

## Solución

- Simplificar el proceso de creación de trabajos.
- Recopilar los trabajos para generar el resultado final.

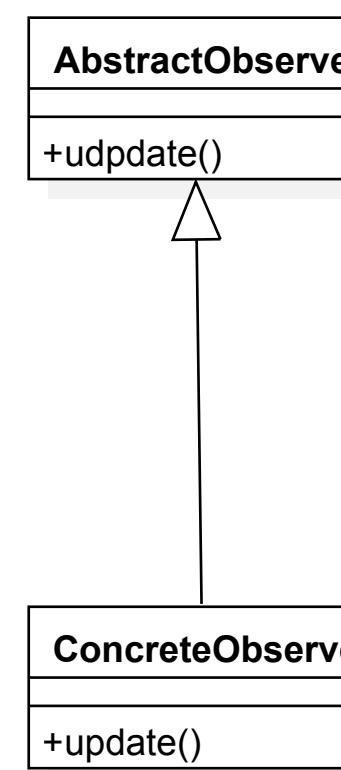
# GENERADOR DE TRABAJOS

- Hecho en Python.
- No es necesario incluir la librería del servidor de Boinc.
- Permite a un programador con pocos conocimientos crear trabajos dado un algoritmo ya creado.



# ASIMILADOR

- Escrito en Python
- Extensión del componente proveído en Boinc
- Implementa el patrón observador



BaseAssimilator
+config +STOP_TRIGGER_FILENAME +caught_sig_int +log +pass_count +update_db +noinsert +wu_id_mod +wu_id_remainder +one_pass +one_pass_N_WU +appname +sleep_interval  +notify() +check_stop_trigger() +sigint_handler(sig, stack) +filename_hash(self, name, hash_fanout) +get_file_path(self, result) +assimilate_handler(self, wu, results, canonical_result) +report_errors(self, wu) +do_pass(self, app) +parse_args(self, args) +def run(self)

ConcreteAssimilator
+suscribers  +notify() +addObserver(observer)

# VALIDACIÓN

2

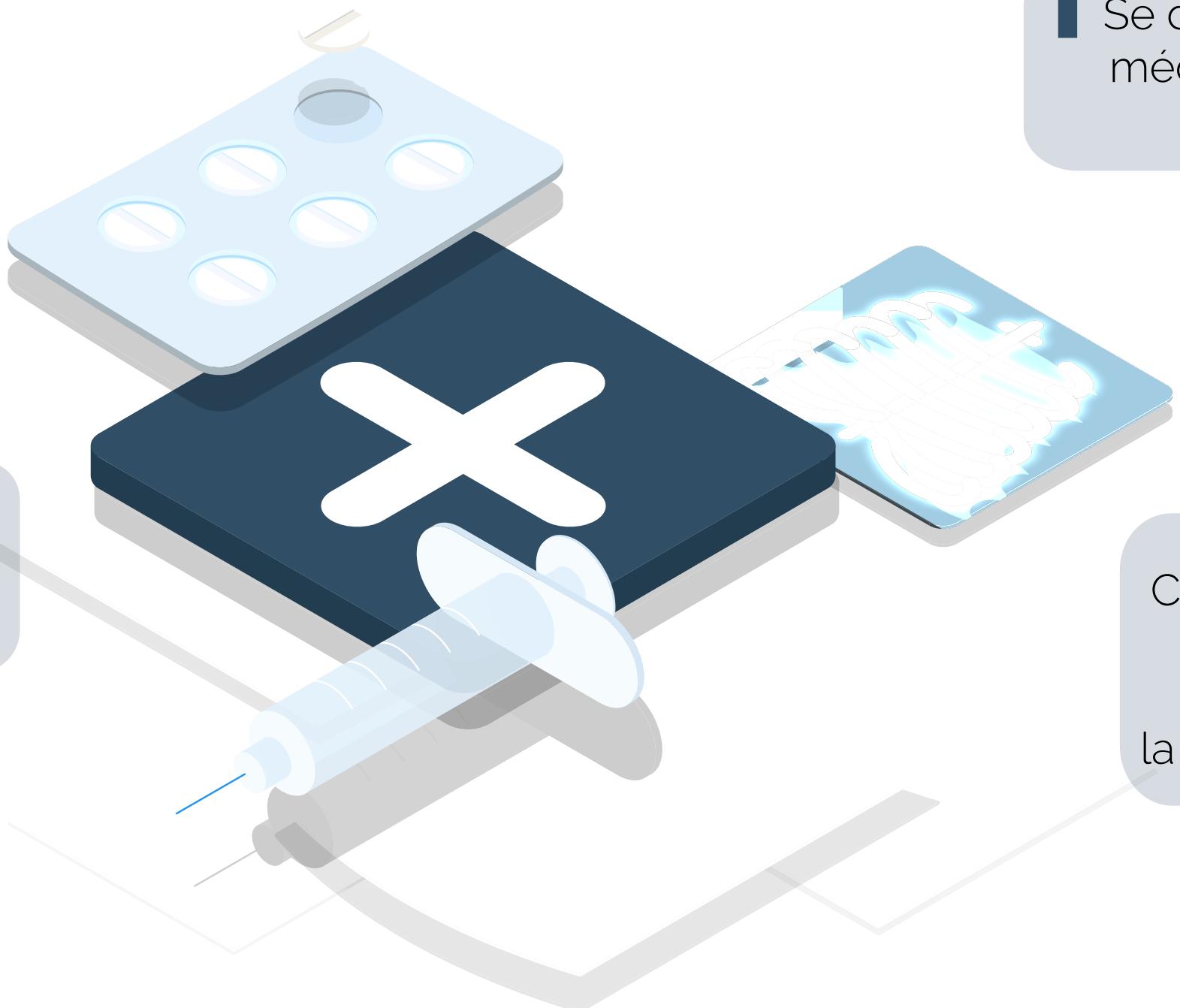
Se guardaron los resultados  
y al obtenerlos se  
unió la imagen.

1

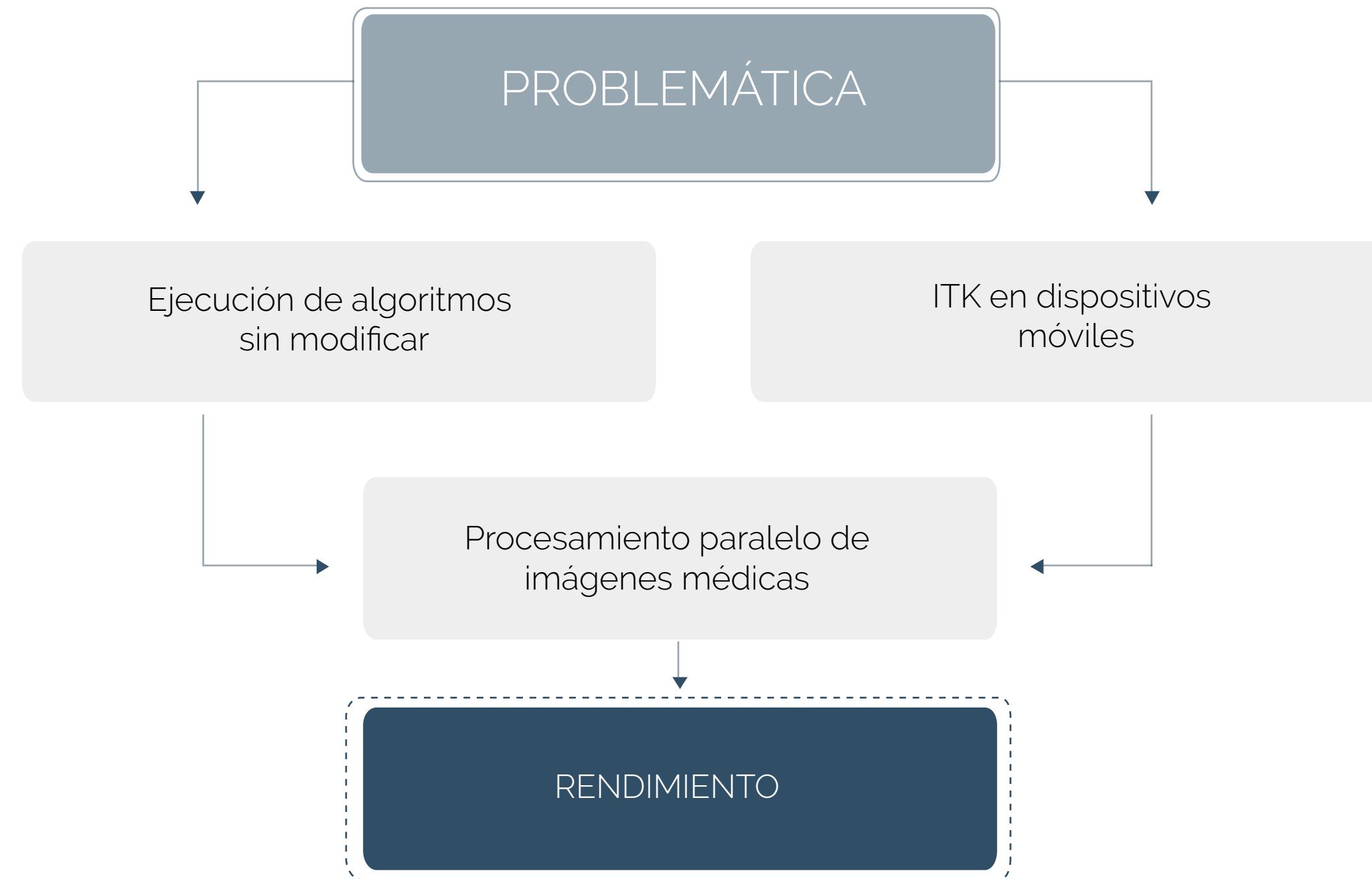
Se dividió una imagen  
médica de 300Mb en  
10000 partes.

3

Comparación a nivel de bits  
del resultado final con  
el resultado de procesar  
la imagen en un computador.



# PRUEBAS



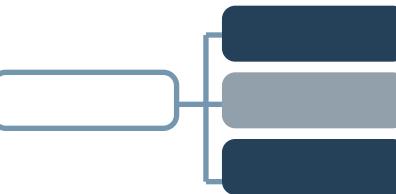
# PRUEBAS DESEMPEÑO

## Prueba Base Diseño Factorial $2^k$

- Determinar efecto de k factores
- Factores



Dispositivos de  
procesamiento



Grado de  
paralelismo



Redundancia



Ubicación  
Servidor

- Métrica  
Tiempo total de ejecución
- Constante  
Algoritmo - Imagen

# FACTORES DEL EXPERIMENTO

FACTOR	NIVEL -1	NIVEL 1
Dispositivos	Computadores Linux x86_64	Dispositivos móviles Android armv7
Grado de Paralelismo	1000	450
Redundancia	1	2
Ubicación Servidor	Local	Remoto

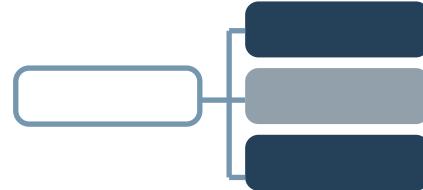
# RESULTADOS

		COMPUTADORES LINUX		DISPOSITIVOS MÓVILES ANDROID	
		1000 TRABAJOS	450 TRABAJOS	1000 TRABAJOS	450 TRABAJOS
LOCAL	Redundancia 1	37m	15m	42m	19m
	Redundancia 2	1h 22m	38m	1h 26m	40m
REMOTO	Redundancia 1	59m	48m	1h 03m	54m
	Redundancia 2	1h 54m	1h 37m	2h 01m	1h 52m

# PORCENTAJE DE VARIACIÓN TOTAL POR FACTOR



Dispositivos



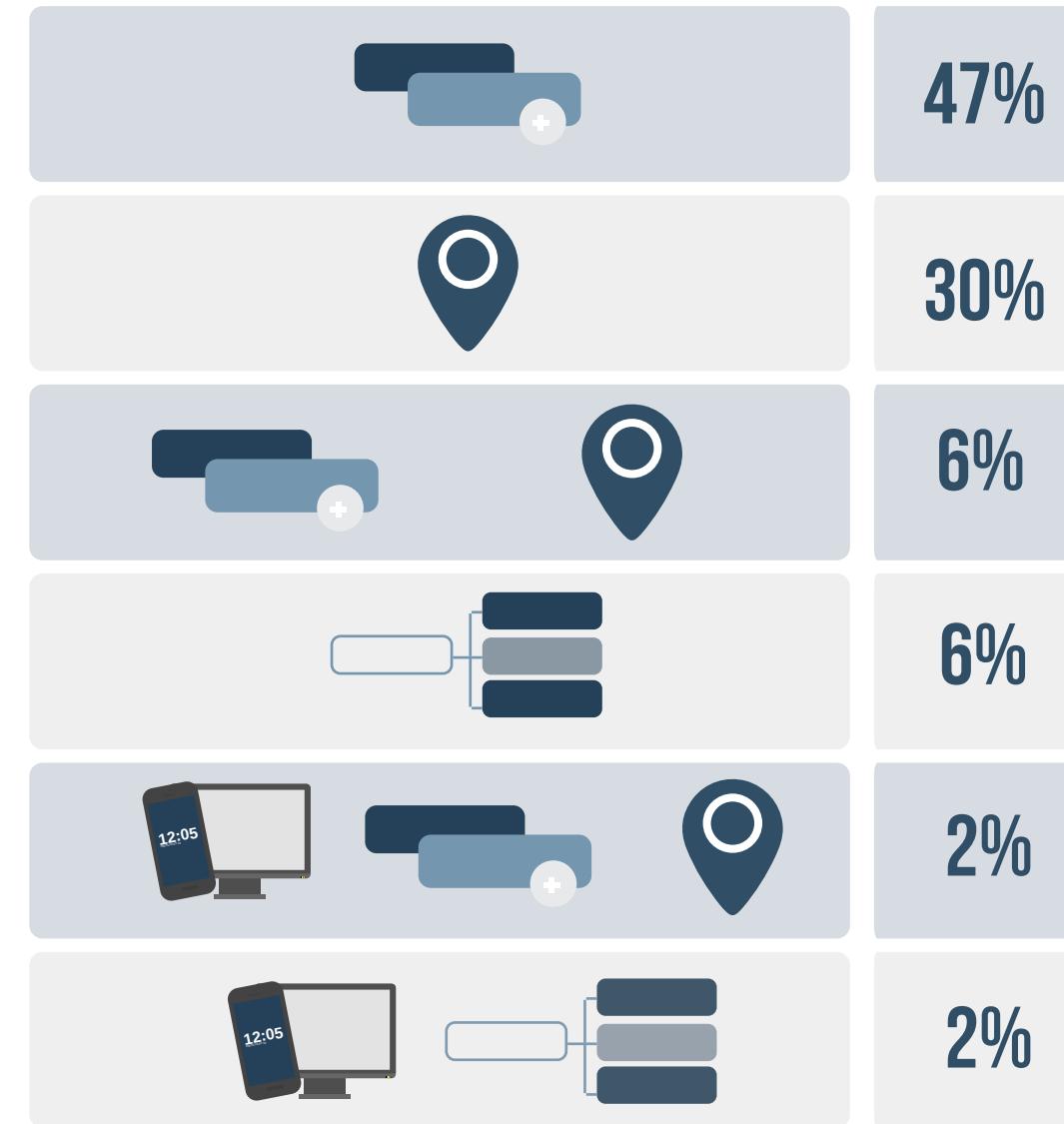
Grado de  
paralelismo



Redundancia

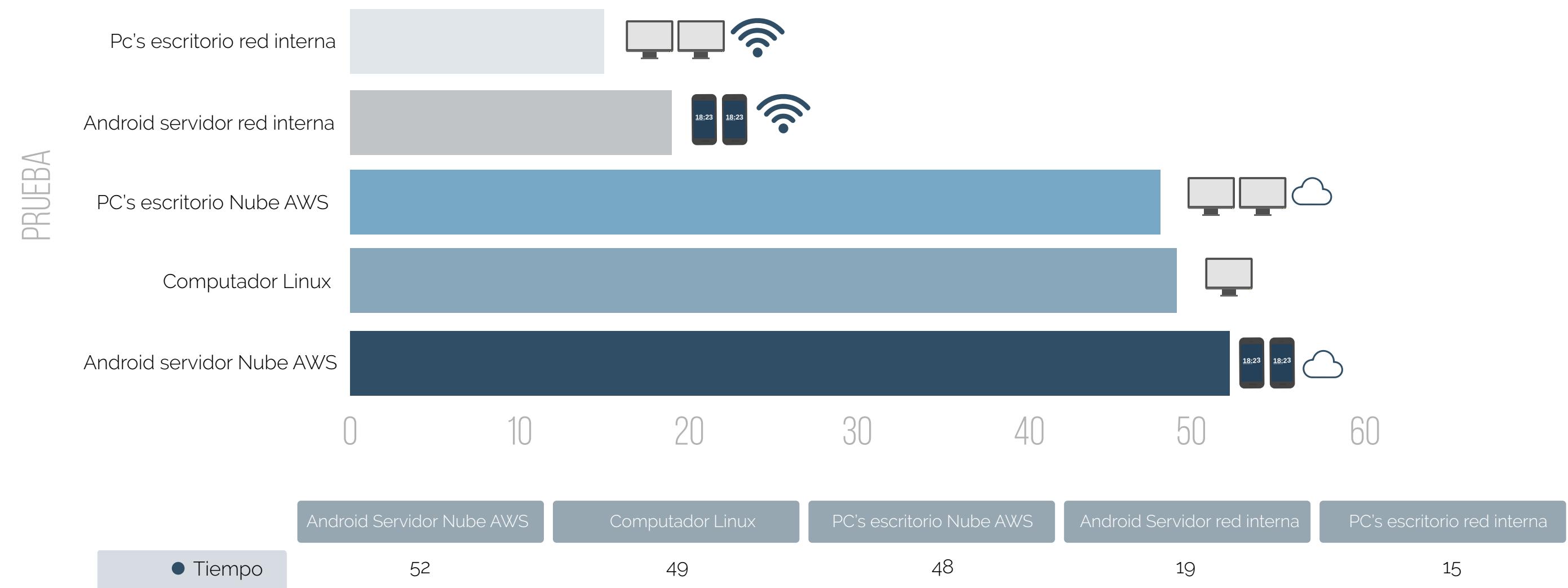


Ubicación  
Servidor



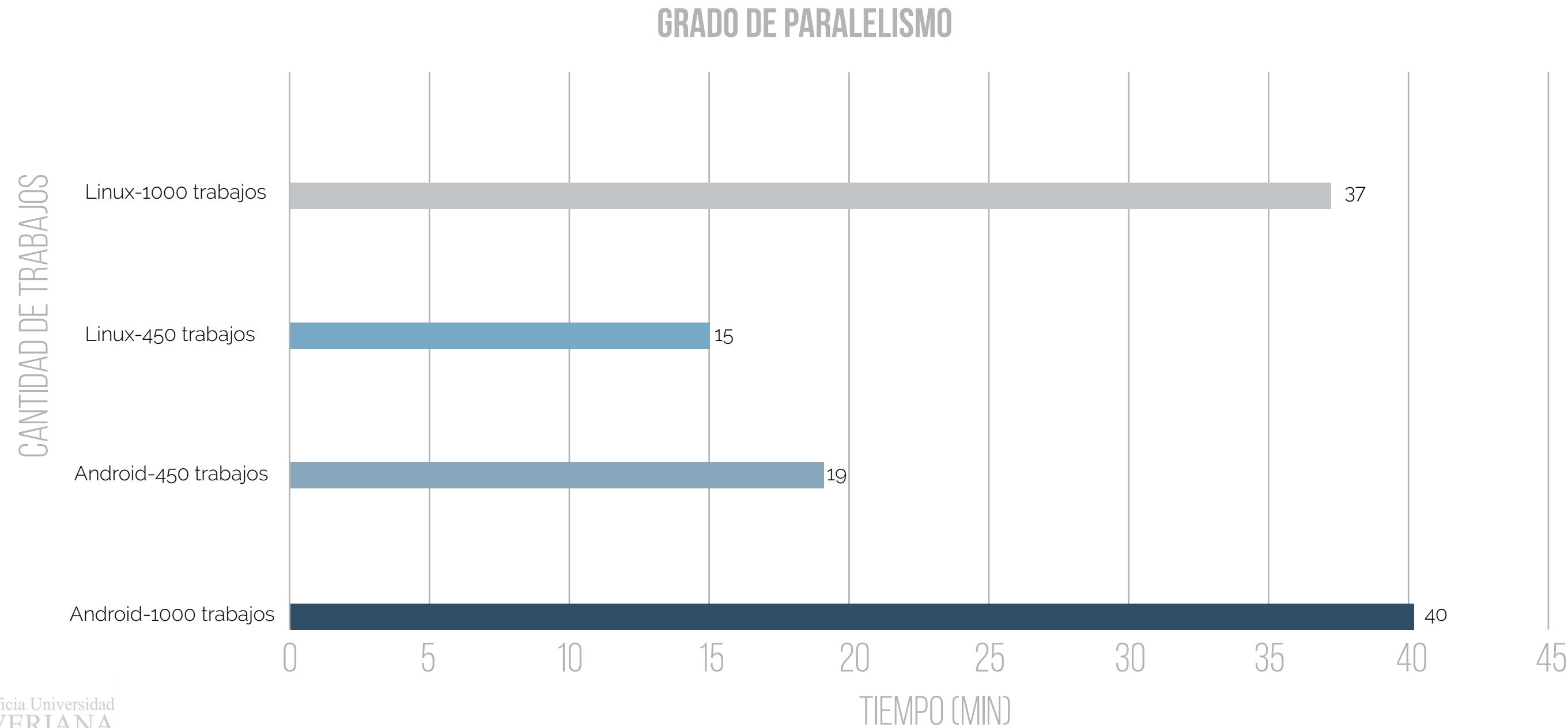
# RESULTADOS

## UBICACIÓN SERVIDOR



# RESULTADOS

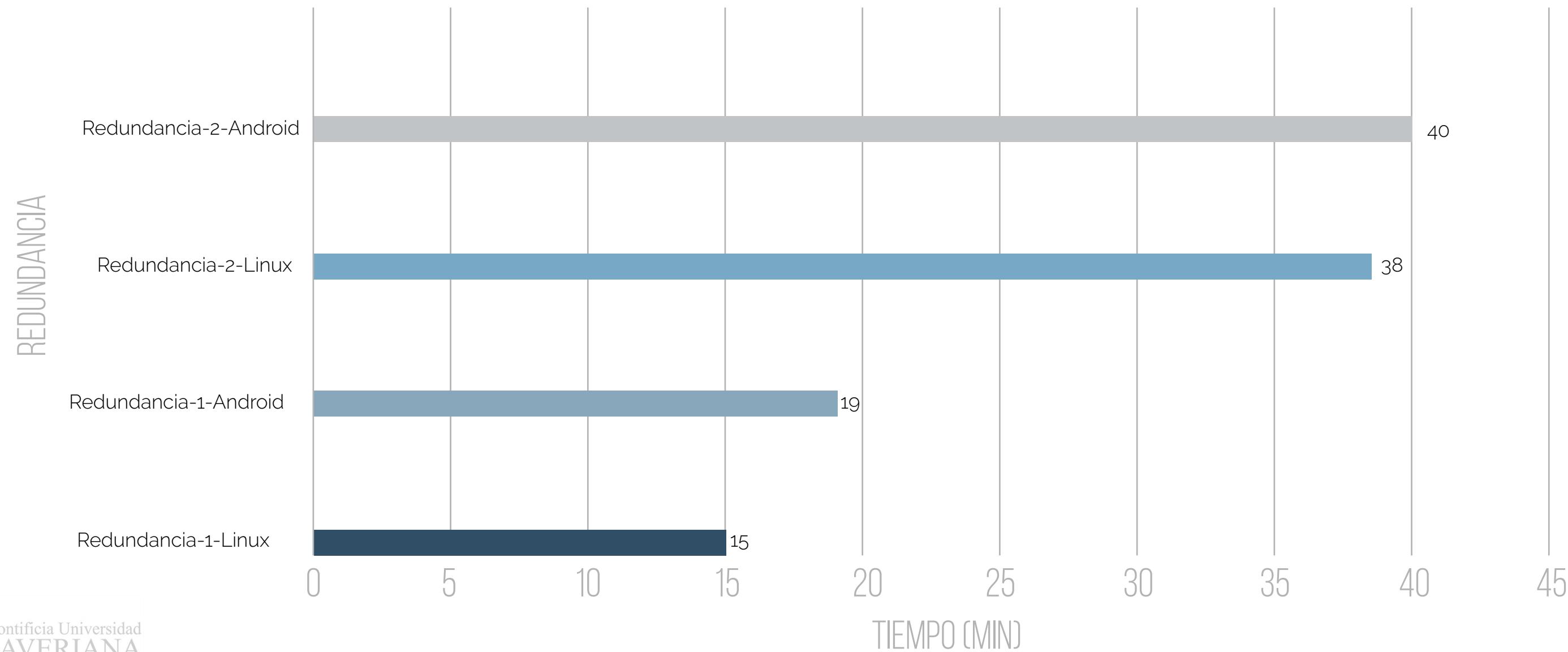
## GRADO DE PARALELISMO



# RESULTADOS

## REDUNDANCIA

### GRID VS CÓMPUTO COLABORATIVO



- Es posible crear una grid móvil para procesar imágenes médicas.
- Los resultados obtenidos son muy prometedores.
- No es necesario modificar el código fuente de los algoritmos ya existentes.
- Los componentes desarrollados simplifican el uso de Boinc.



# CONCLUSIONES

# TRABAJO FUTURO

Enviar tareas utilizando un sistema push.

Automatización de la creación y configuración del servidor Boinc.

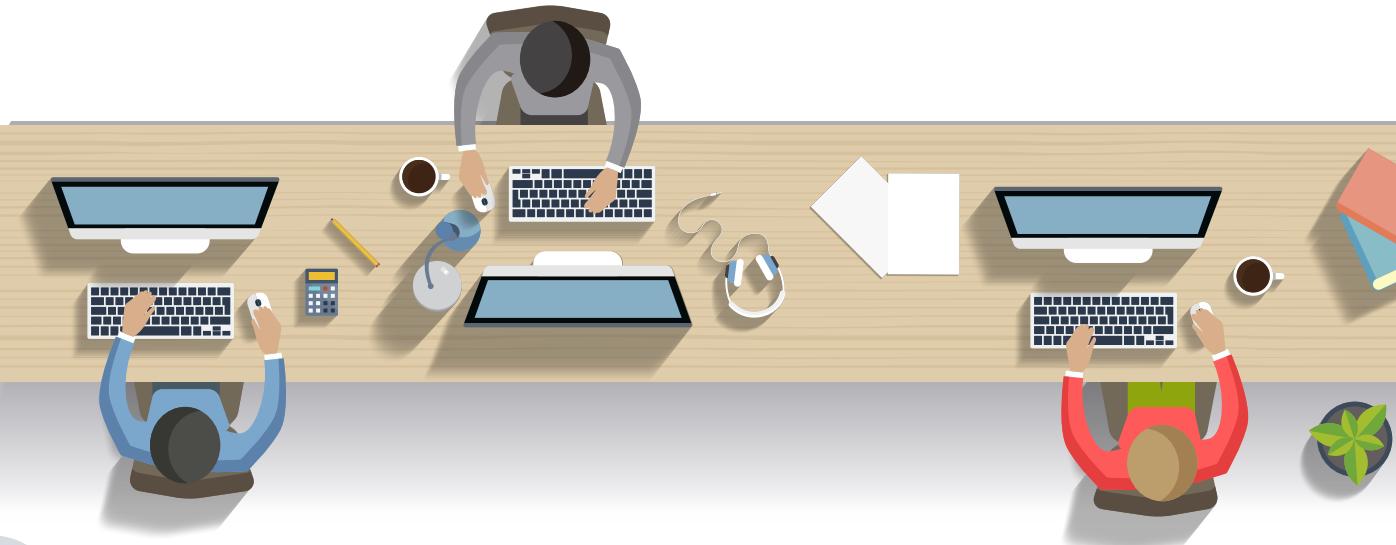
Pruebas Adicionales.

Interfaz Gráfica de Administración.

Soporte a más plataformas.

Modificación del Código de ITK.

Servidor Boinc en celulares.





**iMUCHAS  
GRACIAS!**