

WUOLAH



LosCocos

www.wuolah.com/student/LosCocos



Relación Tema 3.pdf

Relación Problemas Resueltos Tema 3



2º Sistemas Operativos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



Exámenes, preguntas, apuntes.



RELACIÓN EJERCICIOS TEMA 3

3) La única forma de implementar el swapping sería cargar las páginas en las mismas posiciones de memoria en que se cargaron originalmente. Sería posible que el cargador recalculara las direcciones en base al desplazamiento respecto a la posición donde se cargó la página por i / n veces, (PILA) cualquier referencia a una posición de memoria, p.ej un puntero, quedaría invalidado por eso, debido a que este conoce las direcciones y puede recalcularlas, pero no su contenido, que podría ser una referencia a memoria que quedaría sin recalcular.

4) Direcciones lógicas de 32 bits = 4B
 2^{20} páginas

Memoria física de 32 MBytes

¿Cuánta memoria requiere un proceso de 453 KB + tabla páginas → entradas 32 bits

Entrada Tabla Páginas 32 bits

20	12
----	----

4p $2^{12}B = 4KB$

$$\frac{453KB}{4KB} = 113.25 \approx 114 \text{ páginas}$$

$$TAM = 114 \cdot 2^{12} + 456$$
$$\approx 456.5 \text{ MB}$$

$$TP \rightarrow 4B/\text{dirección} \times 114 \text{ entradas} = 456B$$

6) No, aunque el tamaño del ejecutable sea muy grande, es posible que el proceso solo necesite acceder a una parte del segmento de datos con regularidad y no al segmento completo, por lo que el conjunto de trabajo sería más reducido que el ejecutable.

7) Esto es debido a que las direcciones lógicas son relativas a cada proceso por lo que no se corresponden con la misma dirección en procesos distintos. Sin embargo, las direcciones físicas son fijas, se refieren a una única posición de memoria.

8) Espacio virtual de 65.536 bytes dividido en págs de 4096 'B'

$$\text{Región de texto} = 32768 \text{ B} \rightarrow \frac{32768 \text{ B}}{4096 \text{ B/pág}} = 8 \text{ págs}$$

$$\text{Región de datos} = 16386 \text{ B} \rightarrow \frac{16386 \text{ B}}{4096 \text{ B/pág}} = 4,000488... 5 \text{ págs}$$

$$\text{Pila} = 15878 \text{ B} \rightarrow \frac{15878 \text{ B}}{4096 \text{ B/pág}} \rightarrow 3,876... 4 \text{ págs}$$

$$\text{Espacio Virtual } \frac{65536 \text{ B}}{4096 \text{ B/pág}} = 16 \text{ páginas}$$

Programa $8 + 5 + 4 = 17$ páginas \rightarrow El proceso No cabe en el E.V

- La solución sería disminuir el tamaño de página hasta que cupiera el proceso

9) El programa puede necesitar una página no cargada en memoria, que se haya modificado después de comenzar la ejecución, por lo que la sección de texto puede no coincidir.
 \rightarrow Esto no ocurriría si las páginas no cargadas estuvieran en la zona de intercambio, ya que al recompilar no se actualizarían las páginas almacenadas allí, y al volver a cargarlas si estamos cargando la versión del programa que exista antes de recompilarla.

- 10)
- a) Número de marco \rightarrow Lo escribe el SO al asignar memoria física a una página
 - b) Bit de presencia \rightarrow SO escribe al asignar memoria física a una página
 \rightarrow Lo consulta el MMU en cada acceso
 - c) Bit de protección \rightarrow SO, lo consulta el MMU para validarlo
 - d) Bit de modificación \rightarrow MMU activa, SO detecta, SO lee
cuando detecta acceso cuando acaba de escribirse en MS
 - e) Bit de referencia \rightarrow MMU lo pone a 1 cuando detecta acceso a una página
SO consulta y pone a cero si el algoritmo de reemplazo lo requiere.

11

a) 999 $999 / 1024 = 0$ resto = 999

Nº pág 0, bit de presencia 0, se producirá una falta de página

b) 2121 $2121 / 1024 = 2$ resto = 73

marco 1, desplazamiento 73

DF = $1 * 1024 + 73 = 1097$

c) 5400 $5400 / 1024 = 5$ resto = 280

Nº pág 5, marco 0, desplazamiento = 280

DF = $0 * 1024 + 280 = 280$

13

Una de las propiedades de la localidad de un proceso es la espacial. Si se han referenciado ciertas posiciones de memoria es probable que las adyacentes también sean referenciadas. Como en la segmentación los bloques no tienen por qué estar continuos en memoria, no tenemos en cuenta su localidad.

12

a) FIFO, 3 marcos

Marcos	1	2	3	4	1	2	5	1	2	3	4	5
0	1	1	1	4	4	4	5	5	5	5	5	5
1		2	2	2	1	1	1	1	1	3	3	3
2			3	3	3	2	2	2	2	2	4	4
	*	*	*	*	*	*	*			*	*	

→ 9 Fallos de página

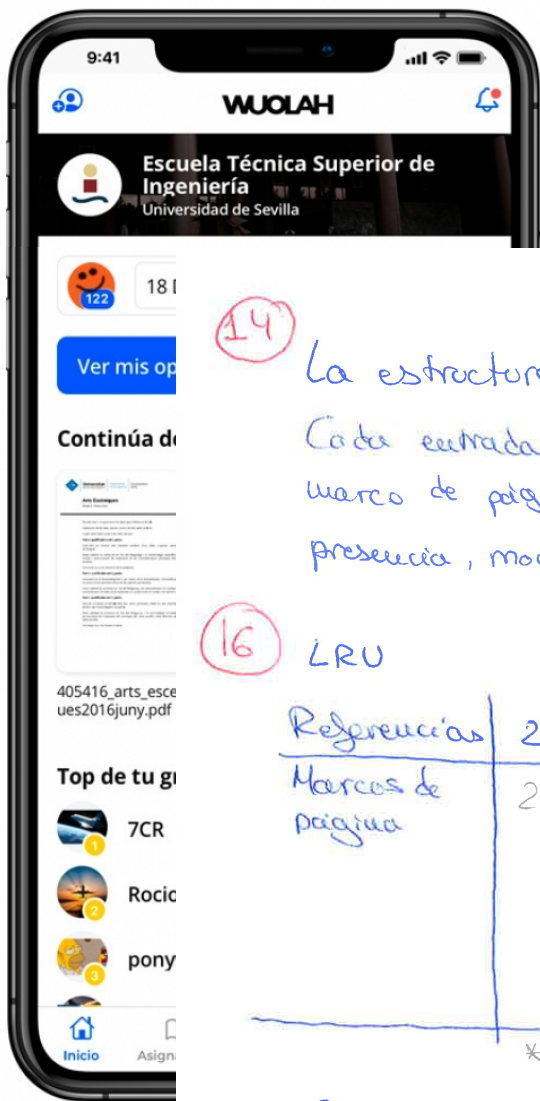
b) FIFO, 4 marcos

Marcos	1	2	3	4	1	2	5	1	2	3	4	5
0	1	1	1	1	1	1	5	5	5	5	4	4
1		2	2	2	2	2	2	1	1	1	1	5
2			3	3	3	3	3	3	2	2	2	2
3				4	4	4	4	4	4	4	3	3
	*	*	*	*			*	*	*	*	*	*

→ 10 Fallos de página

2 No se corresponde, acortala

Descarga la app de Wuolah desde tu store favorita



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



14

La estructura es la tabla de páginas. Cada proceso tiene una. Cada entrada de esta tabla contiene información sobre el marco de página donde se almacena en memoria, bits de presencia, modificación y de protección.

16 LRU

Referencias	2	1	3	4	3	5	6	4	5	7	4	2
Marcos de página	2	2	2	2	2	2	6	6	6	6	6	6
		1	1	1	1	1	1	1	1	1	1	2
			3	3	3	3	3	3	3	7	7	7
				4	4	4	4	4	4	4	4	4
						5	5	5	5	5	5	5
	*	*	*	*	*	*	*	*	*	*	*	*

→ Se producen 8 faltas de página.

17

- En caso de que se produzcan muchas faltas de página, el algoritmo FFP realiza más comprobaciones sobre las páginas que están o no y aprovecha más la memoria, aumenta el conjunto residente, mientras que el algoritmo WS realiza las comprobaciones en intervalos fijos.
- Si hay pocas faltas de página, el algoritmo FFP realizará menos comprobaciones y podremos tener páginas que no se referencian durante mucho tiempo en memoria.

18 FFP $m=3$

	0	3	1	1	1	3	4	4	2	2	4	0	0	0	0	3
0	0	0	0	0	0	0	-	-	-	-	-	0	0	0	0	0
1	-	-	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	-	-	2	2	2	2	2	2	2	2
3	-	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	-	-	-	-	-	-	4	4	4	4	4	4	4	4	4	4
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

6 Faltas más

22

- a) La tabla de marcos de página, que contiene información acerca de la protección del marco (bit de protección)
- b) La página cargada en memoria podría ser usada como un buffer de entrada/salida que debe estar bloqueado durante la transferencia; sin embargo, al no ser referenciado por el proceso, un algoritmo de gestión podría eliminar la página de memoria. Con la función *fix-page (up)* lo solucionamos.
- c) Podría producirse hiperpaginación si se fijan más páginas de las necesarias. Sería conveniente restringir el número de marcos que se pueden fijar.

26

- a) 0,430 → Segmento 0, presencia 0
No se puede traducir, se produciría una falta de página que se debe tratar
- b) 1,10 → Segmento 1, Dir base = 2300, longitud = 14
 $10 < 14 \Rightarrow DF = 2300 + 10 = \underline{2310}$
- c) 3,400 → Segmento 3, presencia 0
No se puede traducir, se produciría una excepción por falta de página
- d) 4,112 → Segmento 4, Dir base = 1952, longitud = 96
 $112 > 96 \Rightarrow$ No podemos obtener la DF, no es válida si el desplazamiento es mayor o igual a la longitud del segmento.

(20) WS $T=3$

	1	4	4	4	2	4	1	1	3	3	5	5	5	5	1	4
2	1	1	1	-	2	2	2	-	3	3	3	3	-	-	1	1
	2	2	-	-	-	-	1	1	1	1	5	5	5	5	5	5
		4	4	4	4	4	4	4	-	-	-	-	-	-	-	4
0	*	*			*		*		*		*		*		*	*
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

→ 8 Falta de página.

(21) Para adaptar el algoritmo WS, ordenamos que cada x referencias la función limpiar- U borre aquellas páginas cuyo bit U este a 0, es decir, que se eliminen las páginas que no hayan sido referenciadas en ese intervalo de tiempo x . Después de usar esta función, reiniciamos el bit U .

(19)

	1	4	2	2	2	4	5	5	3	3	5	1	1	1	1	4
2	1	1	1	1	1	1	5	5	5	5	5	5	5	5	5	5
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	-	-	-	-	-	-	-	-	3	3	3	3	3	3	3	3
0	*	*					*		*		*		*		*	*
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

(*)

~ Se produce un bloqueo, no tenemos los recursos necesarios.

El proceso esperaría hasta tener los recursos necesarios para cumplir su tamaño.

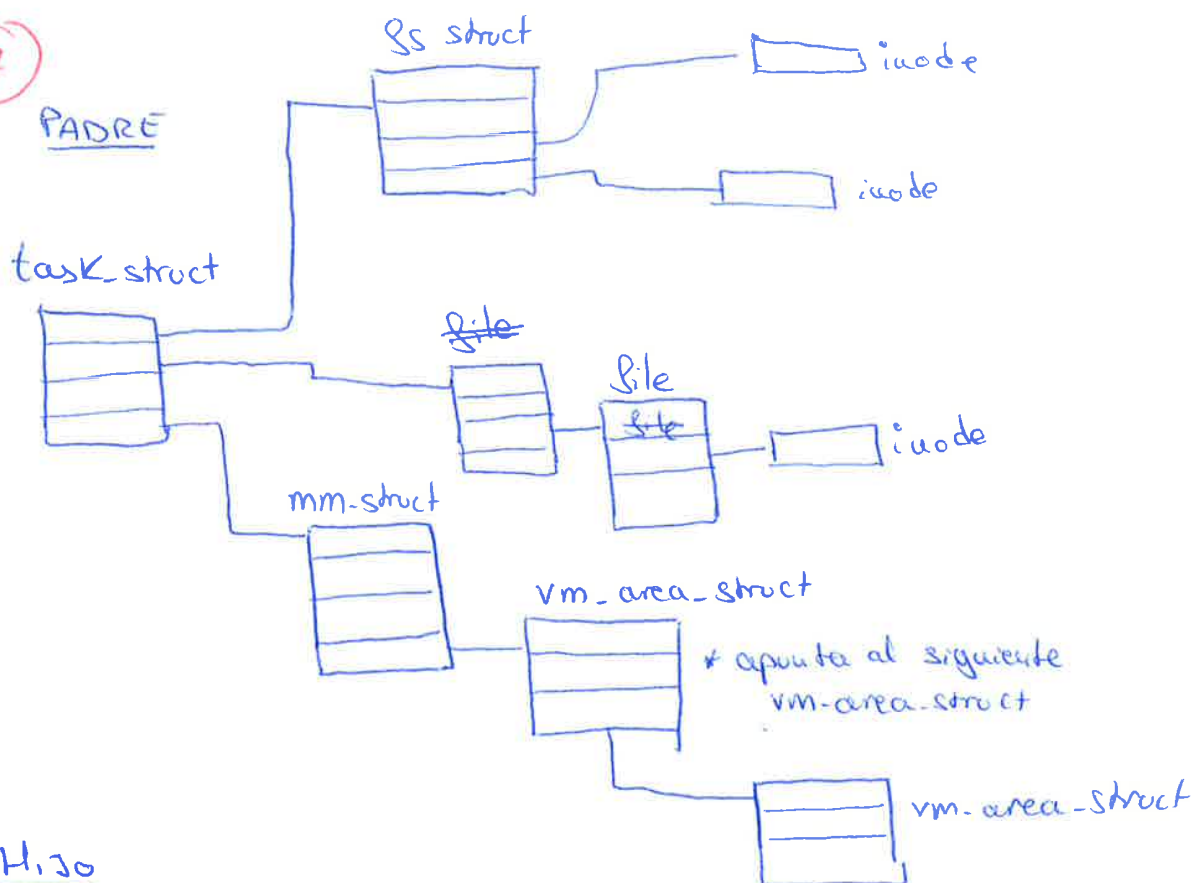
→ 5 Falta de página.

(5)

Después de los finales, ¡RELÁJATE CON GUATAFAC!

WUOLAH

(27)



H.30

Al ejecutar el padre via fork, obtiene una copia de todas sus estructuras pero ninguna es compartida. La única forma de que alguna de estas estructuras sea compartida es usando clone.

(28)

- No comparten el segmento de datos
- No comparten el segmento de código
- No comparten el espacio de direcciones
- Solo comparten las secciones abiertas

(3)

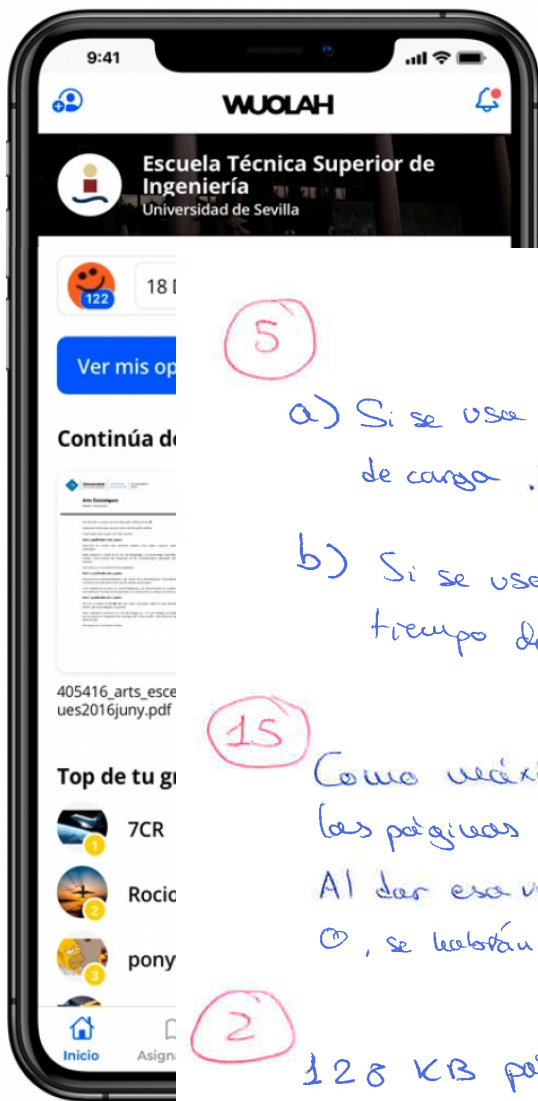
→ 35% utilización 97% disp. paginación

Estado de hiperpaginación, estados bloqueados esperando a que se resuelvan las faltas de página

→ 15% utilización y disp. paginación

El porcentaje es bajo. Hay espacio y podemos ejecutar varios procesos a la vez.

(7)



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



(5)

a) Si se usa FIFO se sustituye la que tiene menor tiempo de carga. (2) con 120.

b) Si se usa LRU se sustituye la que tenga menor tiempo de referencia. (3) con 200.

(15)

Como máximo avanzará una vuelta, en caso de que todas las páginas residentes tengan activo el bit de referencia.

Al dar esa vuelta ya encontrará una página con el bit de referencia a 0, se habrán puesto todos a cero durante la primera vuelta.

(2)

128 KB páginas $\rightarrow 2^{17}$ = 17 direcciones (páginas)

8 KB / página $\rightarrow 2^{13} \rightarrow 13$ bits desplazamiento

MF = 64 MB = 2^{26} $\rightarrow 26$ bits dir física

DL = n° pag + despla = 17 + 13 = 30 bits DC

DF = 26 bits

17 n° pag | 13 desp

12 marca | 13 despl

(23)

a) RAM de 4 KB = 2^{12} KB

1 KB / pag

$2^{12} / 2^{10} / \text{pag} = 2^2 \text{ pag} = 4 \text{ marcos página}$

b)

4 marcos de página = $2^2 = 2$ bits para identificarlos

$$c) 2^{10} \text{ B/pag} = \boxed{1024 \text{ B}}$$

0	KERNEL						
1	Pago P ₁ •	Pago P ₁	Pago P ₁ •	Pago P ₁	Pago P ₁ •	Pago P ₁	Pago P ₁ •
2		Pago P ₂ •	Pago P ₂	Pago P ₂ •	Pago P ₂	Pago P ₂ •	Pago P ₂
3				Pago P ₂ •	Pago P ₂ •	Pago P ₂ •	Pago P ₂ •
	*	*		*	*	*	*

24

1024 páginas

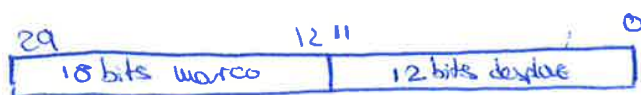
Espacio de direcciones lógicas 4 GB

↳ tamaño pag 4 KB

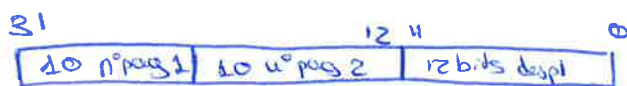
Espacio de dir. físicas hasta 16B = 2^{30} B

Técnica de paginación con tamaño de página 2^{12} B

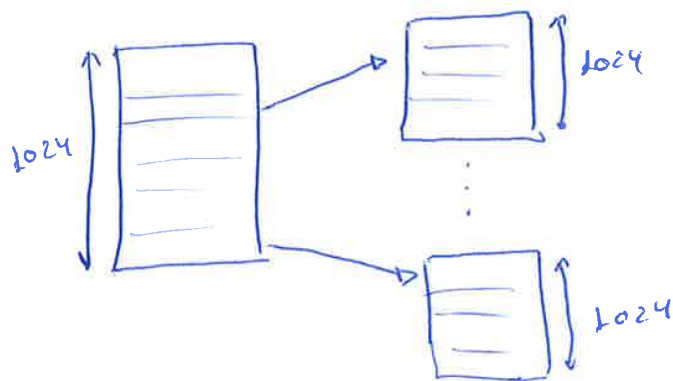
→ La DF tiene 30 Bytes



→ La DL tiene 2^{32} Bytes → 32 Bytes



1024 pag = 2^{10} → 10 bits n° pag 1 y 10 bits n° pag 2



25 Paginación a dos niveles

DIRECCIONES de 8 bits :

2	2	4
1er nivel	2º nivel	desplz.

1^{er} Nivel $\rightarrow 2^2 = 4$ entradas

2^o Nivel $\rightarrow 2^2 = 4$ entradas

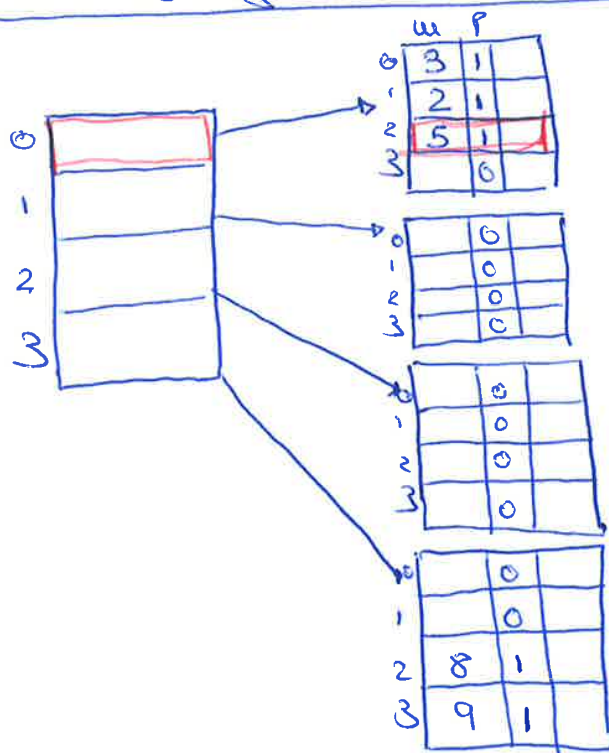
$2^4 \rightarrow 16$ B/pag

Espacio virtual

Texto	16 B
Datos	48 B
Pila	224 B
	256 B

$$\frac{256 \text{ B}}{16 \text{ B/pag}} = 16 \text{ páginas}$$

Tabla de Páginas a dos niveles



MP de 160 B

$$\frac{160 \text{ B}}{16 \text{ B/pag}} = 10 \text{ entradas (marcos)}$$

0
1
2
3
4
5
6
7
8
9

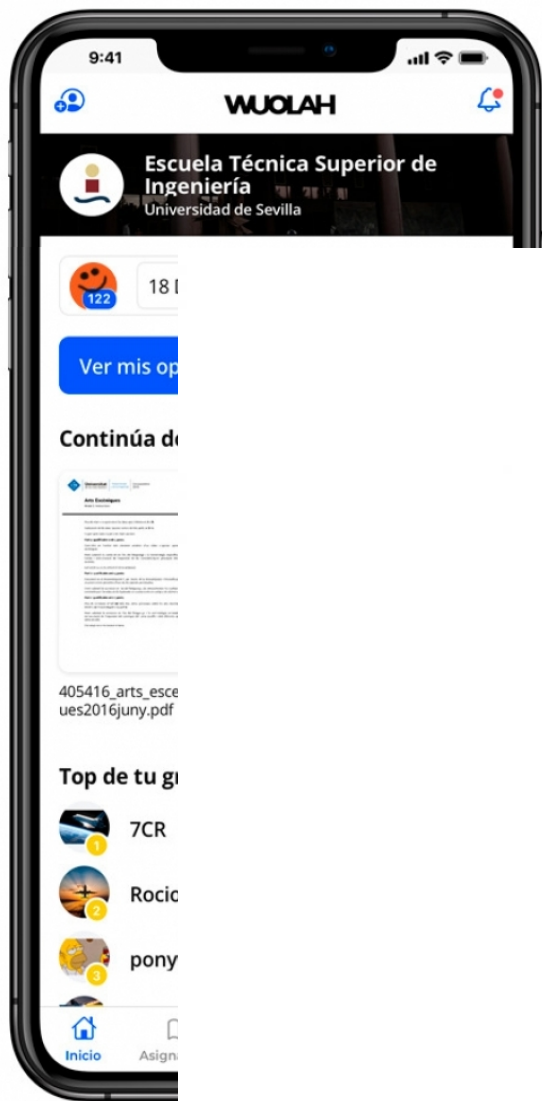
Traducción DV 47

$$1^{\text{er}} \text{ Nivel} = \frac{47}{4 \times 16} = 0 \text{ (marco 1er Nivel)}$$

$$2^{\text{o}} \text{ Nivel} = 47 \% (4 \times 16) = 2 \text{ (marco 2º Nivel)}$$

$$47 \% 16 = 15 \text{ desplazamiento} \Rightarrow \boxed{DF = 5 \times 16 + 15 = 95}$$





Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

