



# Laurel

12/07/2021

---

## Integrantes

Tomás Herceg

Javiera Naranjo

Scarlett Ojeda

Álvaro Navarro

## Puntos de caso de uso:

$$UCP = UUCP * TCF * ECF * PF$$

Donde se sabe que **UUCP** es Puntos de Caso de Uso sin ajustar, **TCF** el Factor de complejidad técnica, **ECF** el Factor de Complejidad del Medio Ambiente y **PF** el Factor de productividad.

## Puntos de caso de uso sin ajustar:

Primero se calcula **UUCP** -> **UUCP = UAW + UUCW**

En nuestro caso se van a tener:

- 2 casos de uso de complejidad simple
- 2 casos de uso de complejidad media
- 3 caso de uso de complejidad alta
- 4 actores de complejidad media

Por lo tanto: **UUCW = 2\*5+2\*10+3\*15 => 75**

Y **UAW = 4\*2 = 8**

**UUCP = 8+75 => 83**

## Factor de complejidad técnica:

Factor técnico	Peso	Impacto percibido	Factor calculado
T1	2	3	6
T2	1	5	5
T3	1	5	5
T4	1	1	1
T5	1	0	0
T6	0.5	0	0
T7	0.5	5	2.5
T8	2	0	0

T9	1	2	2
T10	1	0	0
T11	1	0	0
T12	1	0	0
T13	1	0	0
		Factor total	21.5

$$TCF = 0.6 + (0.01 * 21.5) \Rightarrow 0.815$$

Factor de complejidad ambiental:

Factor ambiental	Peso	Impacto percibido	Factor calculado
E1	1.5	2	3
E2	-1	3	-3
E3	0.5	2	1
E4	0.5	2	1
E5	1	4	4
E6	1	5	5
E7	-1	2	-2
E8	2	3	6
		Factor ambiental	15

$$ECF = 1.4 + (-0.03 * 15) \Rightarrow 0.95$$

Factor de productividad:

PF=20

Realizados todos los cálculos anteriores tenemos que:

$$\text{UCP} = 83 * 0.815 * 0.95 * 20 \Rightarrow 1285.25$$

El UCP resultó con 1285.25 horas, suponiendo 40 horas de trabajo a la semana, el sistema se desarrollará en 32.13 semanas (aproximadamente 32 semanas).

### Puntos de Función y COCOMO:

Se considero que nuestro software contiene:

- 2 Entradas de complejidad simple, 2 de complejidad media y 3 de complejidad alta
- 7 Salidas complejidad media
- 1 archivo logico interno complejidad baja, 2 archivos lógicos internos complejidad media y 1 archivo lógico interno de complejidad alta
- 4 consultas complejidad simple
- 2 archivos de interfaces complejidad simple, 1 archivos de interfaces complejidad media

En la siguiente tabla se muestran los puntos de función y el lenguaje de programación utilizado (PASCAL) en el proyecto de software:

## Estimación del esfuerzo del proyecto de desarrollo de software 4

SLOC Input Dialog - <sample>

Sizing Method:  
☐ SLOC  
☒ Function Points  
☐ Adaptation and Reuse

Breakage  
 % of code thrown away due to requirements evolution and volatility  
 REVL: 0.00

Module Size in Function Points  
 Language: JAVA Change Multiplier: 53

Ratio Type: ☒ Jones ☐ David

Calculation Method: ☒ Using Table ☐ Input Calculated Function Point

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	2	2	3	32
Outputs	0	7	0	35
Files	1	2	1	42
Interfaces	2	1	0	17
Queries	4	0	0	12
Total Unadjusted Function Points				138
Equivalent Total in SLOC				7314

OK Cancel Help

USC-COCOMO II.2000.3 - Untitled


File Edit View Parameters Calibrate Phase Maintenance Help

Project Name: Laurel Scale Factor: 18.97 Schedule

Project Notes Development Model: Post Architecture

X	Module Name	Module Size	LABOR Rate (\$/month)	EMF	Language	NCM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	<sample>	F:7314	0.00	1.00	JAVA	26.2	26.2	278.9	0.00	0.0	2.5	0.0

	Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Total Lines of Code:	7314							
Hours/PM:	152.00							
Optimistic	21.0	5.7	348.7	0.00	0.0	2.2		
Most Likely	26.2	10.4	278.9	0.00	0.0	2.5	0.0	
Pessimistic	32.8	11.1	223.1	0.00	0.0	2.9		



Como se puede observar usando el software COCOMO II, eligiendo el lenguaje de programación Java, el staff dio como resultado un personal de 2.5 personas, es decir, 3 personas, un esfuerzo promedio de 26.2 persona/ mes y un tiempo de 10.4 meses.