

Estimación y planificación de desarrollo

Integrantes: Edgar Alarcón
Jonathan Gonzalez
Mónica Higuera

PARTE 1 : ESTIMACIÓN DE TRABAJO

Para obtener los valores de la estimación de trabajo se hará uso del software COCOMO:

SLOC

Para obtener el número total de líneas de códigos sin comentarios , hacemos uso de los valores de “*Function Type*” obtenido en el documento de *Estimación KSLOC_1.0.pdf*. Donde nos da un total de **3127 SLOC**

SLOC Input Dialog - Suralumapp

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	0	0	6
Outputs	1	2	0	14
Files	0	2	0	20
Interfaces	2	0	0	10
Queries	3	0	0	9
Total Unadjusted Function Points				59
Equivalent Total in SLOC				3127

OK Cancel Help

Figura 1:SLOC - COCOMO

FACTORES DE ESCALA

- **Precedentes:**NORM- Se comprende a cabalidad los objetivos del producto,hemos trabajado con sistemas de software relacionados pero no de manera considerable.
- **Flexibilidad de desarrollo:** NOM - Los requisitos preestablecidos son básicos y simples, pero deben cumplirse.
- **Arquitectura/resolución de riesgos:** NOM-Se identificaron los riesgos, y aunque son menores se planificaron soluciones a estos.
- **Cohesión del grupo:**HIGH- Ya hemos trabajado como grupo en proyectos anteriores, es fácil compartir ideas y distribuir tareas.
- **Madurez del proceso:** NORM-El proyecto a pesar de ser simple en términos de trabajo y desarrollo de componentes, requiere apegarse a estándares de calidad, aunque no requiere de trabajo con otros grupos.

Factor	base	Incr%
Precedentedness	NOM	0%
Development Flexibility	NOM	0%
Architecture / risk resolution	NOM	0%
Team cohesion	HI	0%
Process maturity	NOM	0%

Scale Factor : 17.87

OK Cancel Help

Figura 2: Scale Factor- COCOMO

MULTIPLICADORES DE ESFUERZO

- Product base:
 - **RELY:** LOW - Riesgo del software en caso de que llegue a fallar es bajo.
 - **DATA:** NOM - Tamaño de base de datos ($10 < \text{Bytes/SLOC} < 100$).
 - **DOCU:** NOM - La documentación se adecúa a las necesidades del ciclo de vida del proyecto.
 - **CPLX:** NOM - Las operaciones realizadas por la aplicación serán básicas, pero dependerá de otros dispositivos para queries de entrada y salida.
 - **RUSE:** HIGH - Usaremos reutilización de código dentro de la programación de la aplicación.
- Platform base:
 - **TIME:** NOM - Se usa menos del 50% del tiempo disponible por ejecución.
 - **STOR:** NOM - Uso menor al 50% del espacio disponible.
 - **PVOL:** LOW - No van a haber grandes cambios en la aplicación en el tiempo.
- Personnel base:
 - **ACAP:** HIGH - Alta habilidades para diseñar y en diseño detallado.
 - **PCAP:** LOW - Poseemos habilidades de comunicación y cooperación, pero decaemos en eficiencia y habilidad.
 - **PCON:** VHIGH - No existe rotación de personal.
 - **APEX:** VLOW - Tenemos experiencia menor o igual a 2 meses en esta clase de aplicaciones.
 - **LTEX:** NOM - Aproximadamente un año de conocimiento de lenguaje java.
 - **PLEX:** NOM - Tenemos experiencia utilizando android studio.
- Project base:
 - **TOOL:** NOM - Son herramientas de ciclo de vida básicas, moderadamente integradas.
 - **SITE:** VHIGH - Hacemos uso de todos los medios de comunicación disponibles.

base + Incr % = rating

Product:	RELY	DATA	DOCU	CPLX	RUSE
base	LO	LO	NOM	NOM	HI
Incr%	0%	0%	0%	0%	0%

Platform:	TIME	STOR	PVOL
base	NOM	NOM	LO
Incr%	0%	0%	0%

Personnel:	ACAP	PCAP	PCON	APEX	LTEX	PLEX
base	HI	NOM	VHI	VLO	NOM	NOM
Incr%	0%	0%	0%	0%	0%	0%

Project:	TOOL	SITE
base	NOM	VHI
Incr%	0%	0%

User:	USR1	USR2
base	NOM	NOM
Incr%	0%	0%

EAF is also affected by Schedule

EAF: 0.56

OK Cancel Help

Figura 3: Factores de esfuerzo- COCOMO

Si el modelo es de *Early Design*:

Total Lines of Code:	3127
Hours/PM:	152.00

Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	6.8	6.7	458.8	0.00	0.0	1.0	
Most Likely	10.2	7.6	307.4	0.00	0.0	1.3	0.0
Pessimistic	15.3	8.7	204.9	0.00	0.0	1.8	

Figura 4: resultados early design - COCOMO

Si el modelo es de *Post Architecture*:

Total Lines of Code:	3127
Hours/PM:	152.00

Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	4.5	5.9	690.2	0.00	0.0	0.8	
Most Likely	5.7	6.3	552.1	0.00	0.0	0.9	1.7
Pessimistic	7.1	6.8	441.7	0.00	0.0	1.0	

Figura 5: resultados post architecture - COCOMO

ANÁLISIS DE RESULTADOS

Como nuestro proyecto corresponde al modelo de post arquitectura, de manera optimista los resultados entregados por COCOMO son que el proyecto total durará 6 semanas aprox , se requiere solo una persona para su desarrollo y el factor de esfuerzo es igual a 4.5.

PARTE 2 : PLANIFICACIÓN DE DESARROLLO

Para realizar la planificación de desarrollo nos basamos en la metodología **scrum**, aunque con algunas variaciones, en este caso decidimos tener una reunión final dentro de nuestro “Product Backlog” para decidir qué abordaríamos en cada uno de los sprints siguientes y cuántos serían estos. Esto fue decidido, ya que nuestra aplicación es algo bastante sencilla y nos permitía hacer estos pequeños cambios a la metodología.

Haciendo uso del software *Redmine* realizamos la planificación del desarrollo del software, la cual se vería así, siendo esta una carta gantt:

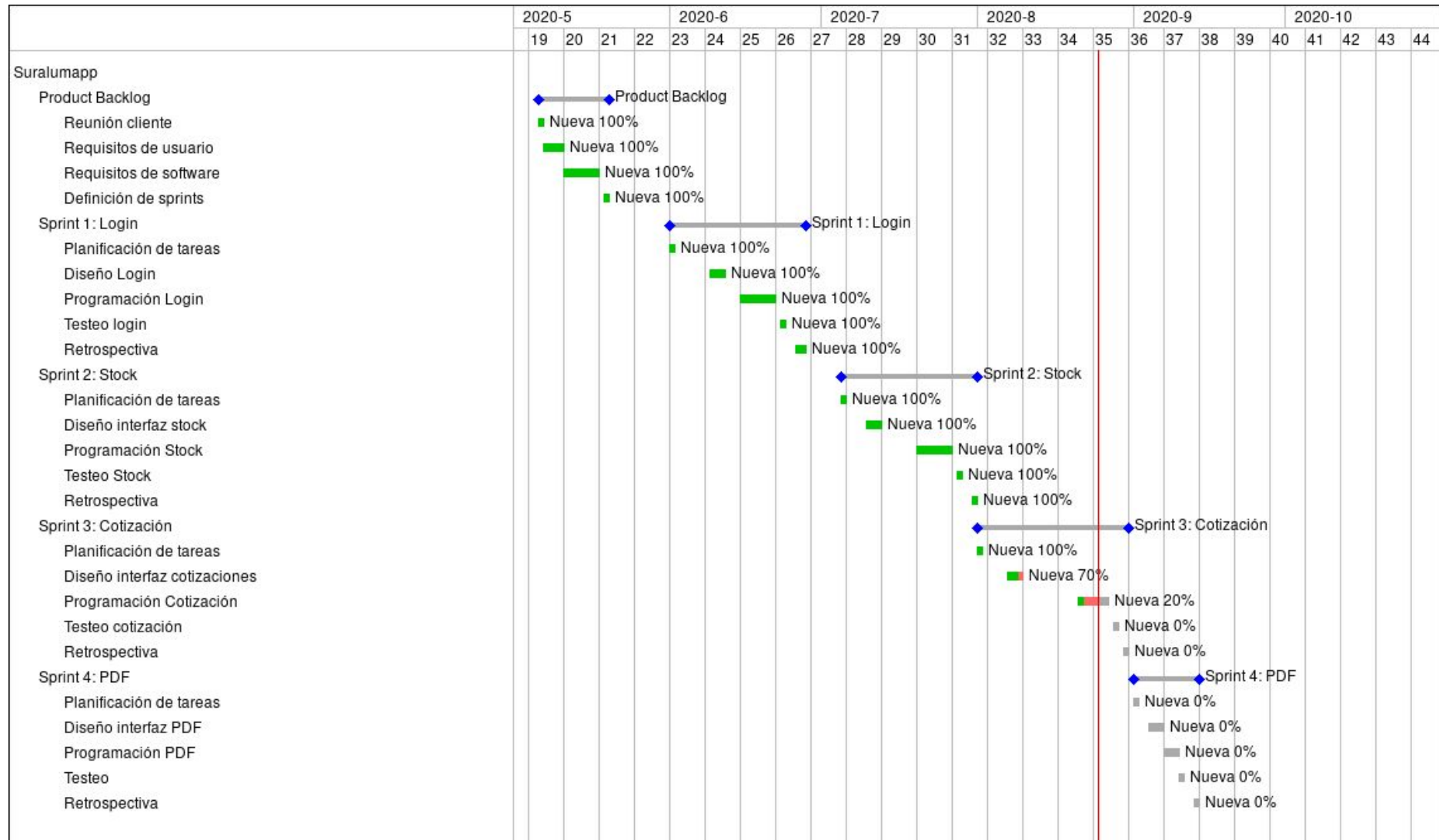


Figura 6: Planificación- Redmine

Referencias :

<https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>

<https://proyectosagiles.org/que-es-scrum/>