

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA CARRERA DE SOFTWARE

Equipo:

NOVA

Nombres de los integrantes:

Darwin Bayas – 7400

Jonathan Chamorro - 7167

Keyla Muñoz - 7191

Stefano Falvo - 6872

Marco Gabilanes - 7171

Materia:

Gestión de proyectos de software

Nivel:

Séptimo 1

Tema:

Estimación de esfuerzo (puntos de función)

Objetivo

Estimar el esfuerzo requerido (hombres-mes, duración y recursos), utilizando la técnica por puntos de función de tu proyecto (toma como referencia las funcionalidades del diagrama de casos de uso).

Desarrollo

Interfaces

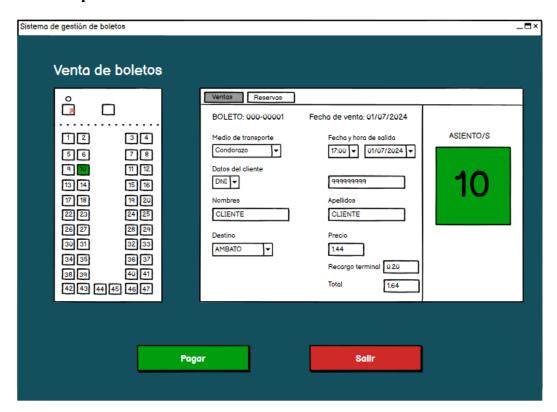
Registro



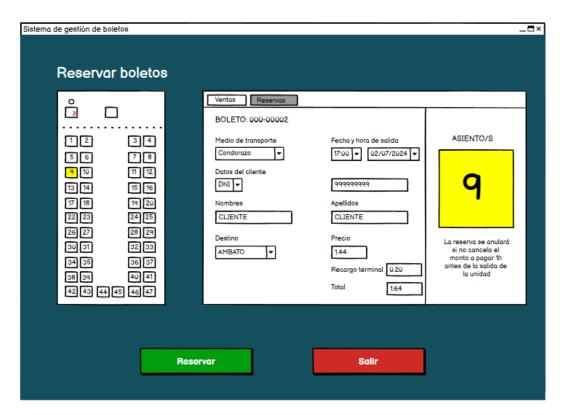
Iniciar sesión



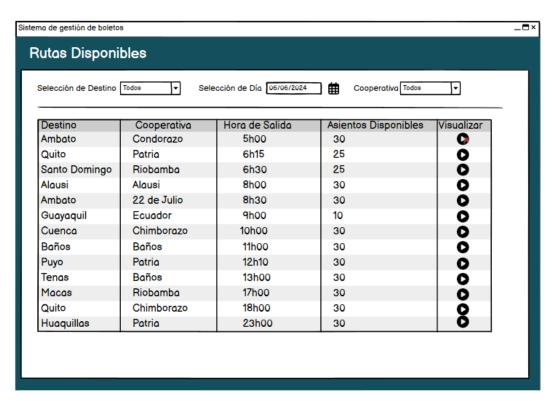
Caso de uso comprar boleto

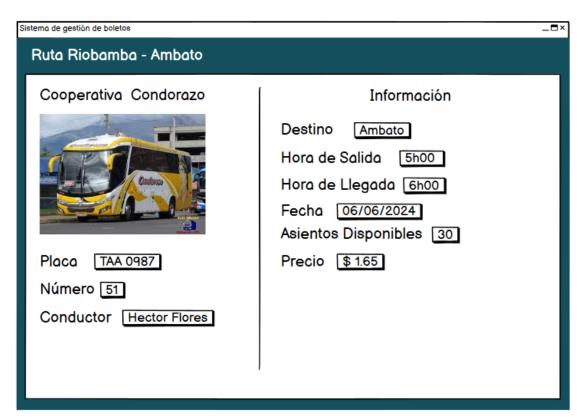


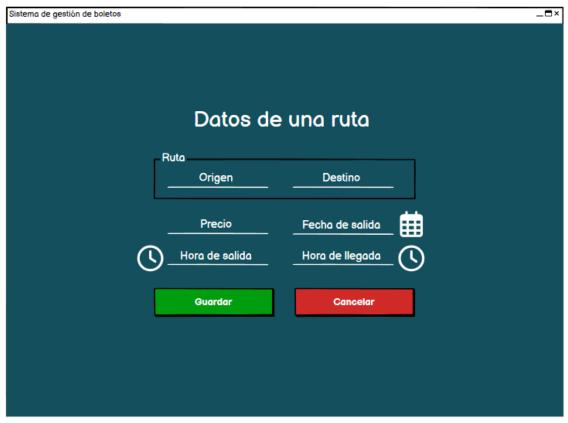
Caso de uso reservar boleto



Caso de uso visualizar rutas







Caso de uso gestionar venta de boletos





Caso de uso gestionar unidades de transporte

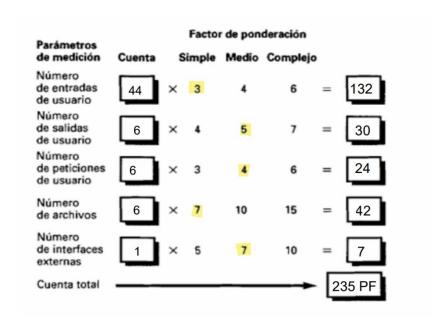






Cálculo del esfuerzo

Puntos de función del sistema



Valor de factor de ajuste:

Características generales del producto software	Valor (0: sin influencia – 5 absolutamente esencial)
1. ¿Requiere el sistema copias de seguridad y	4
de recuperación fiables?	Justificación: La confiabilidad en las copias de seguridad y la recuperación es crucial para asegurar la integridad y disponibilidad de los
2 6 ' ' 1 1 . 0	datos transaccionales de venta de boletos.
2. ¿Se requiere comunicación de datos?	Justificación: El sistema necesita comunicación de datos tanto para las transacciones de boletos como para la sincronización de información en tiempo real entre usuarios y operadores.
3. ¿Existen funciones de procesamiento distribuido?	Justificación: Aunque el sistema puede beneficiarse del procesamiento distribuido, este no es un requisito absolutamente esencial, pero es importante para la escalabilidad.
4. ¿Es crítico el rendimiento?	Justificación: El rendimiento es crítico para asegurar tiempos de respuesta rápidos y una experiencia de usuario fluida, especialmente en períodos de alta demanda.
5. ¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?	Justificación: El sistema se integrará con los procesos existentes del terminal de buses, pero no necesariamente en un entorno extremadamente cargado.
6. ¿Requiere el sistema entrada de datos interactiva?	Justificación: La interacción del usuario para la compra de boletos y la selección de asientos es fundamental para el sistema, incluyendo así entradas interactivas como sugerencias de entradas, autocompletado, cambio del flujo de entrada de datos de acuerdo a las selecciones realizadas.
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	Justificación: La entrada de datos puede requerir varias pantallas para la selección de rutas, asientos y pago, pero esto es manejable.
8. ¿Se actualizan los archivos maestros de forma interactiva?	Justificación: los archivos maestros, como los de rutas, horarios y disponibilidad de asientos,

	deben actualizarse dinámicamente con la			
	información de venta de boletos			
9. ¿Son complejas las entradas, las salidas, los	3			
archivos o las peticiones?	Justificación: la complejidad es moderada,			
	con múltiples rutas y opciones de boletos, pero			
	no extremadamente compleja.			
10. ¿Es complejo el procesamiento interno?	4			
	Justificación: el procesamiento interno,			
	incluyendo la validación de boletos y la			
	sincronización de los datos, es			
	significativamente complejo.			
11. ¿Se ha diseñado el código para ser	2			
reutilizable?	Justificación: aunque la reutilización de			
	código es beneficiosa, no es un objetivo			
	primordial en este sistema.			
12. ¿Están incluidas en el diseño la conversión	2			
y la instalación?	Justificación: a pesar de que el sistema se			
	integrará con ciertos procesos existentes del			
	terminal de buses, al ser un sistema web nuevo,			
	es importante pero no crucial la conversión de			
	datos e instalación.			
13. ¿Se ha diseñado el sistema para soportar	1			
múltiples instalaciones en	Justificación: el sistema está diseñado			
diferentes organizaciones?	principalmente para una única instalación			
	inicial en el terminal de Riobamba, aunque			
	podría ampliarse en el futuro.			
14. ¿Se ha diseñado la aplicación para facilitar	4			
los cambios y para ser	Justificación: la facilidad de uso y la			
fácilmente utilizada por el usuario?	capacidad de adaptación a cambios futuros son			
	cruciales para la aceptación y sostenibilidad			
	del sistema.			

TDI (Total Degree of Influence) = 45

$$VAF = (TDI \times 0.01) + 0.65$$

$$VAF = (45 \times 0.01) + 0.65$$

VAF = 1.1

Puntos de función ajustados

PFs ajustados = PFs sin ajustar x VAF

PFs ajustados = 235×1.1

PFs ajustados = 258.5

Transformación de PFs ajustados a LOCs

De acuerdo con la siguiente tabla tomada de QSM (Quantitative Software Management), en JavaScript que es el lenguaje elegido para el desarrollo del software, un PF corresponde en promedio a 47 LOCs

Language	Avg	Median	Low	High	
JavaScript *	47	53	31	63	

Por lo tanto: 258.5 PFs ajustados x 47 LOCs = 12149.5 LOCs

Aplicación de COCOMO para la estimación del esfuerzo

KLOC = (PF ajustados x LOC leng)/1000

KLOC = (12149.5)/1000

KLOC = 12.1495 = 12 KLOCs

Se identifica así como un modelo orgánico por ello se usarán los siguientes valores:

	c1	c2	с3
Orgánico	2.4	1.05	0.38

Esfuerzo en hombres-mes

 $H-M = c1 \times KLOC^{c2}$

 $H-M = 2.4 \text{ x } (12)^{1.05}$

 $H-M = 32.61 \approx 33 \text{ Hombres - Mes}$

Duración estimada en meses

 $Tdev = 2.5 \text{ x H-M}^{c3}$

Tdev = $2.5 \times (32.61)^{0.38} = 9.39$ meses

Programadores que pueden conformar el equipo

No. Prog = H-M / Tdev

No. Prog = 33 / 9.39

No. Prog = 3.51 programadores ≈ 4 programadores

Conclusiones

- La técnica de puntos de función utilizada en el proyecto permite una estimación detallada y precisa del esfuerzo requerido para el proyecto del software. Al ajustar los puntos de función de acuerdo con el Valor de Factor de Ajuste (VAF) y convertir estos puntos a líneas de código (LOC) específicas del lenguaje de programación elegido (JavaScript), se obtiene una cuantificación clara del esfuerzo en términos de hombres-mes. Esto facilita la planificación y asignación de recursos, mejorando la gestión del proyecto.
- El análisis de las características generales del producto, tales como la necesidad de copias de seguridad confiables, la comunicación de datos, el procesamiento distribuido, y la interacción de datos, destaca la importancia de evaluar los atributos específicos del sistema para ajustar correctamente los puntos de función. Este enfoque asegura que todas las particularidades y requisitos críticos del sistema se consideren en la estimación del esfuerzo, lo que contribuye a un desarrollo más alineado con las necesidades del usuario y los objetivos del proyecto.
- Se puede observar finalmente, que los resultados obtenidos al aplicar COCOMO haciendo uso de los Puntos de Función varían de los obtenidos al no haberlos aplicado. Por ejemplo en una previa aplicación de COCOMO sin puntos de función se obtuvo un aproximado de 50 Hombres Mes, sin embargo ahora al aplicar los Puntos de Función se ha obtenido un estimado de 33 Hombres Mes. Esto puede deberse a una estimación más detallada y minuciosa al haber aplicado dichos puntos de función.