

# LOS GAMERS



## calendarización de tareas

*Proyecto:* **CRIPTONET**

*Fecha:* **22 de mayo de 24**

*Cliente:* **Ninguno**

*Patrocinador:* **Ninguno**

*Gerente del Proyecto:* **Arturo Badillo**

## 1. OBJETIVO

Asignar tareas especificadas en el WBS a los recursos identificados en la matriz RACI del proyecto.

## 2. INSTRUCCIONES

En este informe de práctica de laboratorio, coloca la hoja de cálculo con las asignaciones de tareas a recursos, así como una primera estimación de las tareas.

Actividad en equipos (según proyecto). Elabora la planificación de actividades tomando como referencia el WBS y el diagrama de contexto (diseñalo con respecto al producto a desarrollar).

¿Cuál es el esfuerzo estimado en horas?

¿Estimando sólo tareas acumulativas, cuál es el esfuerzo estimado en hombres-mes? (asumiendo sólo la duración del proyecto considerando cinco recursos)

Actualización del documento del plan de gestión de proyectos, dentro de la línea base de alcance.

## 3. EQUIPOS Y MATERIALES

- Computadoras
- WBS del proyecto
- Diagrama de contexto del software a desarrollar
- Herramientas de hojas de cálculo
- Equipo humano
- Conexión a internet

## 4. ACTIVIDADES POR DESARROLLAR

ID	Descripción	N° integrantes	Edison	Nigell	Jesus	Bryan	Arturo	Total horas equipo	Horas acumuladas
1.	Criponet								
1.1.	Planificación del Proyecto								
1.1.1.	Definir el alcance, objetivos y requisitos del proyecto CriptoNet	2	5	5				10	10
1.1.2.	Crear la estructura de desglose de trabajo (WBS) y el cronograma detallado	2			5	5		10	20
1.1.3.	Estudio de factibilidad	2				5	5	10	30
1.1.4.	Acta constitutiva	2	5	5				10	40
1.1.5.	Plan de gestión del proyecto	2			5		5	10	50
1.1.7.	Desarrollar un plan de gestión de riesgos	2	5	5				10	60
1.2.	Seguimiento y Control								60
1.2.1.	Planificar horarios de reuniones semanales de estado con el equipo de proyecto	1			5			5	65
1.2.2.	Monitorear el progreso del proyecto y compararlo con el cronograma establecido	1	5					5	70
1.2.3.	Identificar y mitigar riesgos a medida que surjan durante el desarrollo	1				5		5	75
1.2.4.	Gestionar y controlar las solicitudes de cambios y los cambios aprobados	1					5	5	80
1.3.	Gestión de Comunicaciones								80
1.3.1.	Establecer canales de comunicación eficientes entre los miembros del equipo y los stakeholders	1		5				5	85
1.3.3.	Gestionar y mantener actualizada la documentación del proyecto	2				4	4	8	93
2.	Análisis y Diseño								93
2.1.	Definición de Requisitos								93
2.1.1.	Requisitos funcionales								93
2.1.1.1.	Especificación caso de uso "Iniciar sesión"	2	6	6				12	105
2.1.1.2.	Especificación caso de uso "Registrarse"	2							105
2.1.1.3.	Especificación caso de uso "Administrar publicaciones"	2			6	6		12	117
2.1.1.4.	Especificación caso de uso "Administrar criptomonedas"	2	6				6	12	129
2.1.1.5.	Especificación caso de uso "Administrar recompensas"	2		6	6			12	141
2.1.2.	Requisitos no funcionales							0	141
2.1.2.1.	Requisitos de rendimiento	3	5	5	5			15	156
2.1.2.2.	Requisitos de seguridad	3			5	5	5	15	171
2.1.2.3.	Requisitos de escalabilidad	3		5	5	5		15	186
2.1.2.4.	Requisitos de usabilidad	3	5	5			5	15	201
2.1.3.	Priorizar y validar requisitos								201
2.1.3.1.	Realizar talleres con stakeholders	2				6	6	12	213
2.1.3.2.	Revisar y aprobar requisitos	1	6					6	219
2.2.	Diseño de Arquitectura								219
2.2.1.	Diseñar arquitectura de software								219
2.2.1.1.	Especificación diagrama de clases "Iniciar sesión"	2	5	5				10	229
2.2.1.2.	Especificación diagrama de clases "Registrarse"	2			5	5		10	239
2.2.1.3.	Especificación diagrama de clases "Administrar publicaciones"	2				5	5	10	249
2.2.1.4.	Especificación diagrama de clases "Administrar criptomonedas"	2		5	5			10	259
2.2.1.5.	Especificación diagrama de clases "Administrar recompensas"	2	5				5	10	269
2.2.1.6.	Definir patrones de diseño	1			4	4		8	277
2.2.1.7.	Diseñar arquitectura de capas	1	4					4	281

2.2.2.	Diseñar arquitectura de criptomonedas								281
2.2.2.1.	Seleccionar protocolos de criptomonedas	1					6	6	287
2.2.2.2.	Diseñar integración con protocolos	1				6		6	293
2.2.2.3.	Diseñar módulos de transacciones	1			6			6	299
2.2.3.	Definir protocolos y tecnologías a utilizar								299
2.2.3.1.	Seleccionar tecnologías frontend	1	5					5	304
2.2.3.2.	Seleccionar tecnologías backend	1		5				5	309
2.2.3.3.	Seleccionar proveedores de nube	1			5			5	314
2.3.	Diseño de Bases de Datos								314
2.3.1.	Diseñar modelo de datos								314
2.3.1.1.	Identificar entidades y relaciones	1	4					4	318
2.3.1.2.	Normalizar modelo de datos	1		4				4	322
2.3.2.	Diseñar esquemas de bases de datos								322
2.3.2.1.	Diseñar esquemas relacionales	1				6		6	328
2.3.2.2.	Diseñar esquemas no relacionales	1					6	6	334
2.3.3.	Definir estructuras de datos y relaciones								334
2.3.3.1.	Definir tablas y campos	1			5			5	339
2.3.3.2.	Definir índices y claves primarias	1			5			5	344
2.3.3.3.	Definir relaciones y llaves foráneas	1			5			5	349
2.4.	Diseño de Interfaces de Usuario								349
2.4.1.	Diseñar wireframes y mockups								349
2.4.1.1.	Diseñar wireframes de baja fidelidad	3	5	5	5			15	364
2.4.1.2.	Diseñar mockups de alta fidelidad	3			5	5	5	15	379
2.4.2.	Definir estándares de diseño								379
2.4.2.1.	Definir guías de estilo	2	5	5				10	389
2.4.2.2.	Definir patrones de diseño UI/UX	2				5	5	10	399
2.4.3.	Diseñar flujos de navegación								399
2.4.3.1.	Mapear flujos de usuario	2		4	4			8	407
3.	Desarrollo								415
3.1.	Frontend								415
3.1.1.	Implementar interfaz de usuario								415
3.1.1.1.	Implementación caso de uso "Iniciar sesión"	2	5	5				10	425
3.1.1.1.	Implementación caso de uso "Registrarse"	2			5	5		10	435
3.1.1.1.	Implementación caso de uso "Administrar publicaciones"	2				5	5	10	445
3.1.1.1.	Implementación caso de uso "Administrar criptomonedas"	2		5	5			10	455
3.1.1.1.	Implementación caso de uso "Administrar recompensas"	2	5				5	10	465
3.1.2.	Integrar con frameworks y librerías	2		6	6			12	477
3.2.	Backend								477
3.2.1.	Implementar lógica de negocio								477
3.2.1.1.	Desarrollar servicios y controladores	3	6	6	6			18	495
3.2.1.2.	Implementar reglas de negocio	3			6	6	6	18	513
3.2.2.	Integración con Base de Datos								513
3.2.2.1.	Desarrollar capa de acceso a datos	2	5	5				10	523
3.2.2.2.	Implementar consultas y operaciones	2				5	5	10	533

3.2.3.	Integración con Criptomonedas							533
3.2.3.1.	Implementar protocolos de criptomonedas	2		6	6		12	545
3.2.3.2.	Desarrollar módulos de transacciones	2	6		6		12	557
3.2.4.	Implementación caso de uso "Iniciar sesión"	2	5	5			10	567
3.2.5.	Implementación caso de uso "Registrarse"	2			5	5	10	577
3.2.6.	Implementación caso de uso "Administrar publicaciones"	2			5	5	10	587
3.2.7.	Implementación caso de uso "Administrar criptomonedas"	2	5		5		10	597
3.2.8.	Implementación caso de uso "Administrar recompensas"	2		5		5	10	607
3.3.	Desarrollo de Billetera Digital							607
3.3.1.	Implementar funcionalidades de billetera	1	8				8	615
3.3.2.	Integrar con protocolos de criptomonedas	1			6		6	621
3.3.3.	Implementar medidas de seguridad	1				8	8	629
4.	Integración y Pruebas							629
4.1.	Integración de Componentes							629
4.1.1.	Integrar frontend y backend	2		10	10		20	649
4.1.2.	Integrar con bases de datos	2	8			8	16	665
4.1.3.	Integrar con servicios externos - Configurar conexiones con servicios de criptomonedas	1				10	10	675
4.3.	Pruebas unitarias							675
4.3.2.	Prueba unitaria caso de uso "Iniciar sesión"	2	5	5			10	685
4.3.3.	Prueba unitaria caso de uso "Registrarse"	2			5	5	10	695
4.3.4.	Prueba unitaria caso de uso "Administrar publicaciones"	2				5	5	705
4.3.5.	Prueba unitaria caso de uso "Administrar criptomonedas"	2		5	5		10	715
4.3.6.	Prueba unitaria caso de uso "Administrar recompensas"	2	5			5	10	725
4.2.	Pruebas de Integración							725
4.2.1.	Desarrollar casos de prueba de integración	1	6				6	731
4.2.2.	Ejecutar pruebas de integración	1		6			6	737
4.2.3.	Corregir defectos identificados	1			6		6	743
4.3.	Pruebas de Aceptación							743
4.3.1.	Desarrollar casos de prueba de aceptación	2				6	6	755
4.3.2.	Ejecutar pruebas de aceptación	2			6	6	12	767
4.3.3.	Corregir defectos identificados	2	6	6			12	779
4.4.	Pruebas de Rendimiento							779
4.4.1.	Definir escenarios y métricas de rendimiento	2			6		6	791
4.4.2.	Ejecutar pruebas de rendimiento	2	6	6			12	803
4.4.3.	Analizar resultados y optimizar	2				6	6	815
4.5.	Pruebas de Seguridad							815
4.5.1.	Definir alcance y plan de pruebas de seguridad	1	5				5	820
4.5.2.	Ejecutar pruebas de seguridad	2		6	6		12	832
4.5.3.	Analizar resultados y corregir vulnerabilidades	2	6			6	12	844
5.	Implementación							844
5.1.	Preparación del Entorno de Producción							844
5.1.1.	Aprovisionar infraestructura de producción	2	5	5			10	854
5.1.2.	Configurar entorno de producción	2			5	5	10	864
5.1.3.	Implementar monitoreo y alertas	2				5	5	874

5.2.	Migración de Datos								874
5.2.1.	Planificar migración de datos	1	6					6	880
5.2.2.	Ejecutar migración de datos	2		6	6			12	892
5.2.3.	Validar datos migrados	1				5		5	897
5.3.	Despliegue de Aplicación								897
5.3.1.	Preparar artefactos de despliegue	1					6	6	903
5.3.2.	Ejecutar despliegue	2		5	5			10	913
5.3.3.	Verificar despliegue	1	5					5	918
5.4.	Pruebas de Aceptación en Producción								918
5.4.1.	Ejecutar pruebas de aceptación	2				5	5	10	928
5.4.3.	Obtener aprobación final	2	5	5				10	938
		<b>Totales</b>	<b>194</b>	<b>188</b>	<b>206</b>	<b>180</b>	<b>170</b>	<b>938</b>	<b>938</b>

## 5. RESULTADOS OBTENIDOS

Cálculo de los hombres-mes (H-M) para nuestro escenario con 938 horas y 5 personas en el proyecto

Esfuerzo total del proyecto: 938 horas

Jornada semanal: 40 horas

Horas por mes: 160 horas en promedio 40 horas/semana x 4 semanas/mes tomando en cuenta que son horarios fijos laborables

Tiempo de desarrollo a meses

$$T_{dev} = 938/160=5.86 \text{ Meses}$$

Número de personas: 5

$$\text{Hombre-Mes} = 5.86 \text{ meses} * 5 = 29.3 \text{ Hombre-Mes}$$

## 6. CONCLUSIONES

- La diferencia significativa entre la planificación inicial y las nuevas estimaciones sugiere que se debe reevaluar la planificación original para identificar posibles errores o sobreestimaciones.
- Con una duración mucho más corta de lo esperado, se puede optimizar el uso de recursos, posiblemente reduciendo

costos y mejorando la eficiencia del proyecto.

- La distribución uniforme y eficiente del trabajo es clave para cumplir con las nuevas estimaciones. Se debe prestar atención a la gestión del equipo y a la asignación de tareas para mantener esta eficiencia.
- Dado que las estimaciones pueden variar, es crucial implementar un sistema de monitoreo continuo para ajustar la planificación y los recursos según sea necesario durante el desarrollo del proyecto.
- La reducción en la duración del proyecto puede impactar positivamente en los plazos de entrega, permitiendo una entrega más temprana de los resultados o productos, lo que podría beneficiar a los interesados y al mercado.
- Es posible que las metodologías y herramientas utilizadas en la nueva estimación sean más precisas o adecuadas para el tipo de proyecto en cuestión. Considerar la adopción de estas metodologías para futuros proyectos podría ser beneficioso.

## 7. RECOMENDACIONES

- Es fundamental identificar los riesgos potenciales que podrían afectar el proyecto y desarrollar planes de contingencia. Esto incluye riesgos técnicos, de recursos, de plazos y cualquier otro factor que pueda impactar el desarrollo del proyecto.
- Asegurarse de que todos los involucrados, desde el equipo hasta los stakeholders, tengan expectativas realistas sobre el tiempo y los recursos necesarios para completar el proyecto, esto ayuda a evitar desilusiones y permite una mejor gestión de los plazos y recursos.
- Mantener una comunicación clara y constante con todos los miembros del equipo y stakeholders es esencial para alinear las expectativas y asegurarse de que todos entienden los riesgos y las estimaciones.