

LOS GAMERS



WBS

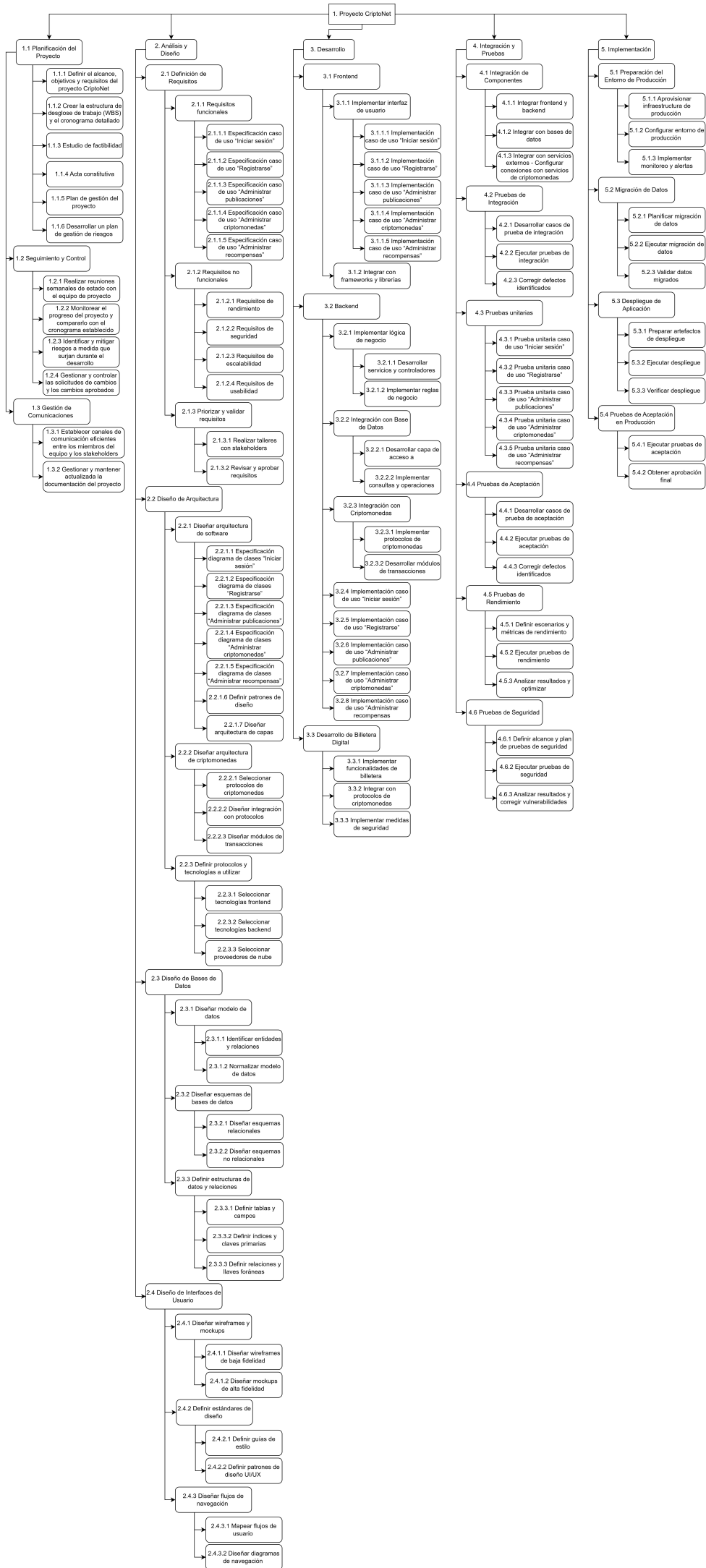
Proyecto: *CRIPTONET*

Fecha: *12 de MAYO de 24*

Ciente: *Ninguno*

Patrocinador: *Ninguno*

Gerente del Proyecto: *Arturo Badillo*



REPRESENTACIÓN TABULAR WBS

| ID | Descripción |
|----------|--|
| 1. | Criponet |
| 1.1. | Planificación del Proyecto |
| 1.1.1. | Definir el alcance, objetivos y requisitos del proyecto CriptoNet |
| 1.1.2. | Crear la estructura de desglose de trabajo (WBS) y el cronograma detallado |
| 1.1.3. | Estudio de factibilidad |
| 1.1.4. | Acta constitutiva |
| 1.1.5. | Plan de gestión del proyecto |
| 1.1.7. | Desarrollar un plan de gestión de riesgos |
| 1.2. | Seguimiento y Control |
| 1.2.1. | Realizar reuniones semanales de estado con el equipo de proyecto |
| 1.2.2. | Monitorear el progreso del proyecto y compararlo con el cronograma establecido |
| 1.2.3. | Identificar y mitigar riesgos a medida que surjan durante el desarrollo |
| 1.2.4. | Gestionar y controlar las solicitudes de cambios y los cambios aprobados |
| 1.3. | Gestión de Comunicaciones |
| 1.3.1. | Establecer canales de comunicación eficientes entre los miembros del equipo y los stakeholders |
| 1.3.3. | Gestionar y mantener actualizada la documentación del proyecto |
| 2. | Análisis y Diseño |
| 2.1. | Definición de Requisitos |
| 2.1.1. | Requisitos funcionales |
| 2.1.1.1. | Especificación caso de uso “Iniciar sesión” |
| 2.1.1.2. | Especificación caso de uso “Registrarse” |
| 2.1.1.3. | Especificación caso de uso “Administrar publicaciones” |
| 2.1.1.4. | Especificación caso de uso “Administrar criptomonedas” |
| 2.1.1.5. | Especificación caso de uso “Administrar recompensas” |
| 2.1.2. | Requisitos no funcionales |
| 2.1.2.1. | Requisitos de rendimiento |
| 2.1.2.2. | Requisitos de seguridad |
| 2.1.2.3. | Requisitos de escalabilidad |
| 2.1.2.4. | Requisitos de usabilidad |
| 2.1.3. | Priorizar y validar requisitos |
| 2.1.3.1. | Realizar talleres con stakeholders |
| 2.1.3.2. | Revisar y aprobar requisitos |
| 2.2. | Diseño de Arquitectura |
| 2.2.1. | Diseñar arquitectura de software |
| 2.2.1.1. | Especificación diagrama de clases “Iniciar sesión” |
| 2.2.1.2. | Especificación diagrama de clases “Registrarse” |
| 2.2.1.3. | Especificación diagrama de clases “Administrar publicaciones” |
| 2.2.1.4. | Especificación diagrama de clases “Administrar criptomonedas” |
| 2.2.1.5. | Especificación diagrama de clases “Administrar recompensas” |
| 2.2.1.6. | Definir patrones de diseño |
| 2.2.1.7. | Diseñar arquitectura de capas |
| 2.2.2. | Diseñar arquitectura de criptomonedas |
| 2.2.2.1. | Seleccionar protocolos de criptomonedas |
| 2.2.2.2. | Diseñar integración con protocolos |
| 2.2.2.3. | Diseñar módulos de transacciones |

- 2.2.3. Definir protocolos y tecnologías a utilizar
- 2.2.3.1. Seleccionar tecnologías frontend
- 2.2.3.2. Seleccionar tecnologías backend
- 2.2.3.3. Seleccionar proveedores de nube
- 2.3. Diseño de Bases de Datos
- 2.3.1. Diseñar modelo de datos
- 2.3.1.1. Identificar entidades y relaciones
- 2.3.1.2. Normalizar modelo de datos
- 2.3.2. Diseñar esquemas de bases de datos
- 2.3.2.1. Diseñar esquemas relacionales
- 2.3.2.2. Diseñar esquemas no relacionales
- 2.3.3. Definir estructuras de datos y relaciones
- 2.3.3.1. Definir tablas y campos
- 2.3.3.2. Definir índices y claves primarias
- 2.3.3.3. Definir relaciones y llaves foráneas
- 2.4. Diseño de Interfaces de Usuario
- 2.4.1. Diseñar wireframes y mockups
- 2.4.1.1. Diseñar wireframes de baja fidelidad
- 2.4.1.2. Diseñar mockups de alta fidelidad
- 2.4.2. Definir estándares de diseño
- 2.4.2.1. Definir guías de estilo
- 2.4.2.2. Definir patrones de diseño UI/UX
- 2.4.3. Diseñar flujos de navegación
- 2.4.3.1. Mapear flujos de usuario
- 2.4.3.2. Diseñar diagramas de navegación

- 3. Desarrollo
- 3.1. Frontend
- 3.1.1. Implementar interfaz de usuario
- 3.1.1.1. Implementación caso de uso “Iniciar sesión”
- 3.1.1.1. Implementación caso de uso “Registrarse”
- 3.1.1.1. Implementación caso de uso “Administrar publicaciones”
- 3.1.1.1. Implementación caso de uso “Administrar criptomonedas”
- 3.1.1.1. Implementación caso de uso “Administrar recompensas”
- 3.1.2. Integrar con frameworks y librerías
- 3.2. Backend
- 3.2.1. Implementar lógica de negocio
- 3.2.1.1. Desarrollar servicios y controladores
- 3.2.1.2. Implementar reglas de negocio
- 3.2.2. Integración con Base de Datos
- 3.2.2.1. Desarrollar capa de acceso a datos
- 3.2.2.2. Implementar consultas y operaciones
- 3.2.3. Integración con Criptomonedas
- 3.2.3.1. Implementar protocolos de criptomonedas
- 3.2.3.2. Desarrollar módulos de transacciones
- 3.2.4. Implementación caso de uso “Iniciar sesión”
- 3.2.5. Implementación caso de uso “Registrarse”
- 3.2.6. Implementación caso de uso “Administrar publicaciones”
- 3.2.7. Implementación caso de uso “Administrar criptomonedas”
- 3.2.8. Implementación caso de uso “Administrar recompensas”
- 3.3. Desarrollo de Billetera Digital

- 3.3.1. Implementar funcionalidades de billetera
- 3.3.2. Integrar con protocolos de criptomonedas
- 3.3.3. Implementar medidas de seguridad

- 4. Integración y Pruebas
 - 4.1. Integración de Componentes
 - 4.1.1. Integrar frontend y backend
 - 4.1.2. Integrar con bases de datos
 - 4.1.3. Integrar con servicios externos - Configurar conexiones con servicios de criptomonedas
 - 4.3. Pruebas unitarias
 - 4.3.2. Prueba unitaria caso de uso “Iniciar sesión”
 - 4.3.3. Prueba unitaria caso de uso “Registrarse”
 - 4.3.4. Prueba unitaria caso de uso “Administrar publicaciones”
 - 4.3.5. Prueba unitaria caso de uso “Administrar criptomonedas”
 - 4.3.6. Prueba unitaria caso de uso “Administrar recompensas”
 - 4.2. Pruebas de Integración
 - 4.2.1. Desarrollar casos de prueba de integración
 - 4.2.2. Ejecutar pruebas de integración
 - 4.2.3. Corregir defectos identificados
 - 4.3. Pruebas de Aceptación
 - 4.3.1. Desarrollar casos de prueba de aceptación
 - 4.3.2. Ejecutar pruebas de aceptación
 - 4.3.3. Corregir defectos identificados
 - 4.4. Pruebas de Rendimiento
 - 4.4.1. Definir escenarios y métricas de rendimiento
 - 4.4.2. Ejecutar pruebas de rendimiento
 - 4.4.3. Analizar resultados y optimizar
 - 4.5. Pruebas de Seguridad
 - 4.5.1. Definir alcance y plan de pruebas de seguridad
 - 4.5.2. Ejecutar pruebas de seguridad
 - 4.5.3. Analizar resultados y corregir vulnerabilidades

- 5. Implementación
 - 5.1. Preparación del Entorno de Producción
 - 5.1.1. Aprovisionar infraestructura de producción
 - 5.1.2. Configurar entorno de producción
 - 5.1.3. Implementar monitoreo y alertas
 - 5.2. Migración de Datos
 - 5.2.1. Planificar migración de datos
 - 5.2.2. Ejecutar migración de datos
 - 5.2.3. Validar datos migrados
 - 5.3. Despliegue de Aplicación
 - 5.3.1. Preparar artefactos de despliegue
 - 5.3.2. Ejecutar despliegue
 - 5.3.3. Verificar despliegue
 - 5.4. Pruebas de Aceptación en Producción
 - 5.4.1. Ejecutar pruebas de aceptación
 - 5.4.3. Obtener aprobación final