

Business Model Canvas

Designed for:

Trabajo final

Designed by:

Date:

Version:

1

Key Partners	Key Activities	Value Propositions	Customer Relationships	Customer Segments
<ul style="list-style-type: none"> - Proveedores de faucet/testnet (Circle faucet para USDC, testnet Sepolia) - Mediadores para resolución de disputas - Expertos legales y de seguridad informática - Expertos en contratos inteligentes (Solidity) - Proveedores de infraestructura en la nube (AWS) - Plataformas de conversión cripto-fiat (en futuro mainnet) - Comunidades de desarrolladores Web3 y blockchain 	<ul style="list-style-type: none"> - Desarrollo y pruebas en testnet Sepolia - Uso de token USDC de testnet (obtenido vía faucet) - Implementación del Escrow Pattern (fondos retenidos hasta finalización) - Implementación de state machine en smart contract (uso de enums para estados WorkStatus) - Control de acceso (modificadores para limitar funciones al cliente, freelance o mediador) - Emisión de event logs para trazabilidad y transparencia en la blockchain - Desarrollo y despliegue de contratos inteligentes en Sepolia - Integración del frontend (React) con MetaMask y testnet - Simulación y testing de casos de uso completos 	<ul style="list-style-type: none"> - Garantía de cumplimiento y seguridad vía escrow en contratos inteligentes (testnet primero) - Flujo de negocio transparente y auditible (eventos en blockchain) - Pagos con tokens de testnet (USDC) para pruebas seguras antes de producción real - Acceso global y democratizado para contratación de servicios digitales - Validación segura y simple entre partes sin riesgo para el usuario durante pruebas 	<ul style="list-style-type: none"> - Onboarding con wallet MetaMask en modo testnet - Automatización de contratos y comunicaciones entre cliente y freelance - Flujo transparente del estado del trabajo mediante event logs - Validación de entregas y liberación de fondos siguiendo el flujo sugerido: <ol style="list-style-type: none"> 1. Cliente crea trabajo y deposita fondos → WorkStatus.Created 2. Trabajador acepta → WorkStatus.InProgress 3. Trabajador entrega trabajo → WorkStatus.Submitted 4. Cliente aprueba → Work Status.Completed + liberación de fondos 	<ul style="list-style-type: none"> - Freelancers y creadores de contenido digital (en entorno de pruebas) - Empresas y particulares que contraten digital services
Key Resources			Channels	

	<ul style="list-style-type: none"> - Contrato inteligente en Solidity (testnet compliant) - MetaMask integrado para conexión con Sepolia testnet - Token USDC de testnet - Infraestructura en la nube (AWS) - Frontend React integrado con Web3.js/Ethers.js y testnet - Repositorio de código y gestión de versiones (GitHub) - Pipeline CI/CD para despliegues seguros en testnet 		<ul style="list-style-type: none"> - Aplicación Web y Mobile (conexión testnet Sepolia) - Grupos en Discord, Telegram, foros Web3 - Workshops y eventos de prueba en universidades y meetups tech
<p>Cost Structure</p> <ul style="list-style-type: none"> - Desarrollo de contratos inteligentes y auditoría de seguridad en testnet - Desarrollo web y mobile - Costos de infraestructura en la nube (AWS) - Soporte al usuario y capacitación - Costos legales para cumplimiento regulatorio futuro 		<p>Revenue Streams</p> <ul style="list-style-type: none"> - En testnet → No ingresos (ambiente de pruebas) - Modelo proyectado a futuro: comisión por transacción, servicios premium, publicidad externa, integraciones 	