**Guía Rápida: Contrato Inteligente de Verificación de Entregables**

**🚀 SETUP INICIAL**

**1. Preparación**

* Abrir **Remix IDE** (remix.ethereum.org)
* Conectar **MetaMask** a red **Sepolia**
* Tener **ETH de testnet** (faucet de Sepolia)

**2. Deploy del Contrato**

* Pegar código en **nuevo archivo .sol**
* **Compilar** con Solidity ^0.8.19
* **Deploy** con parámetro: 24 (deadline 24 horas)

**👥 CONFIGURACIÓN DE PARTICIPANTES**

**3. Registrar Participantes (Solo Owner)**

// Developer (rol 0)

registerParticipant("0xAddressDev", 0, "Juan Developer")

// Tester (rol 1)

registerParticipant("0xAddressTester", 1, "Maria Tester")

// Verifier (rol 2)

registerParticipant("0xAddressVerifier", 2, "Carlos Verifier")

**👨‍💻 ENTREGAS DEL DEVELOPER**

**4. Preparar Programas en GitHub**

* Crear **3 programas**: Fibonacci, Factorial, USD/BOB Converter
* Hacer **commits separados** para cada programa
* Obtener **hash de cada commit** y **timestamp**

**5. Entregar Programas (Cambiar a cuenta Developer)**

// Fibonacci (a tiempo)

submitFibonacci("repoURL", "commitHash1", timestampAntes, "fileHash1")

// Factorial (tarde)

submitFactorial("repoURL", "commitHash2", timestampDespues, "fileHash2")

// Conversor (opcional)

submitConversorUSDTOB("repoURL", "commitHash3", timestamp3, "fileHash3")

**🧪 ENTREGAS DEL TESTER**

**6. Preparar Tests**

* Crear **archivo de tests** para los 3 programas
* **Commit** y **push** a GitHub
* Obtener **hash del commit** y **timestamp**

**7. Entregar Tests (Cambiar a cuenta Tester)**

submitTestFile("testRepoURL", "testCommitHash", timestamp, "testFileHash")

**✅ VERIFICACIÓN FINAL**

**8. Cerrar Contrato (Cambiar a cuenta Verifier)**

// Aprobar proyecto

verifyAndCompleteContract(true)

// O rechazar proyecto

verifyAndCompleteContract(false)

**🔍 VERIFICACIONES IMPORTANTES**

**9. Consultas de Estado**

// Ver participante específico

getParticipant("0xAddress")

// Ver programa entregado

getProgramSubmission(0) // 0=Fibonacci, 1=Factorial, 2=Conversor

// Estado general del contrato

getContractSummary()

// Verificar deadline

checkDeadlineStatus()

**📊 SISTEMA DE PUNTOS**

**Puntos por Acción:**

* **Inicio**: 50 puntos cada participante
* **Entrega a tiempo**: +5 puntos
* **Entrega tardía**: -1 punto
* **No entrega**: -5 puntos (van al otro participante)
* **Verificación completa**: +5 puntos al verifier

**⚠️ PUNTOS CLAVE**

**Controles de Seguridad:**

* ✅ **Hashes únicos** - No reutilización de commits
* ✅ **Un rol por persona** - No duplicados
* ✅ **Timestamps válidos** - No fechas futuras
* ✅ **Entregas individuales** - Por programa separado

**Flujo Obligatorio:**

1. **Owner** registra participantes
2. **Developer** entrega programas (individual)
3. **Tester** entrega tests (después del developer)
4. **Verifier** cierra contrato (UNA SOLA VEZ)

**Estado Final:**

* Una vez **cerrado**, no acepta más entregas
* **Puntos finales** registrados permanentemente
* **Datos inmutables** en blockchain

**🎯 RESULTADOS ESPERADOS**

**Demo Exitosa:**

* Contrato desplegado en **Sepolia**
* **3 participantes** registrados
* **Entregas escalonadas** (a tiempo/tarde)
* **Sistema de puntos** funcionando
* **Cierre definitivo** del contrato

**Evidencias para Entrega:**

* **Screenshots** de transacciones exitosas
* **Dirección del contrato** en Sepolia
* **Video** demostrando flujo completo
* **Código fuente** documentado