



Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Producto Integrador de Aprendizaje.

Manual de Operaciones

066

tópicos de tecnologías

Docente: Lic. Alejandro Muñoz Solorio

Alumnos: Jesús Alfonso Hernández Topetes

1808373

Eduardo Alonso Gaytan Valadez

1796603

Ciudad Universitaria, San Nicolás de los Garza, Nuevo León a 21 de noviembre de 2024

Introducción

Este documento tiene como objetivo explicar detalladamente el funcionamiento de la aplicación descentralizada (dApp) "Tópicos de Tecnologías - Gestor de Tareas". La DApp permite a los usuarios crear y gestionar tareas en la blockchain de Ethereum, proporcionando una solución descentralizada y transparente para el manejo de tareas personales o colaborativas.

Arquitectura General de la DApp

La dApp está compuesta por dos componentes principales:

Contrato Inteligente (Smart Contract): Escrito en Solidity, el lenguaje de programación para contratos inteligentes en Ethereum. Este contrato define la lógica de negocio, maneja el almacenamiento de datos y ejecuta las operaciones en la blockchain.

Interfaz de Usuario (Frontend): Construida con HTML, CSS y JavaScript, utilizando Web3.js para interactuar con el contrato inteligente. El diseño está enfocado en ofrecer una experiencia de usuario atractiva y moderna, acorde con aplicaciones de una startup tecnológico.

Funcionamiento del Contrato Inteligente

Estructura del Contrato

El contrato inteligente GestorTareas se encarga de:

1. Almacenar las tareas creadas por los usuarios.
2. Permitir la creación y actualización del estado de las tareas.
3. Garantizar que solo el creador de una tarea pueda marcarla como completada.
4. Componentes Clave

Variables de Estado:

contadorTareas: Lleva la cuenta del número total de tareas creadas.

tareas: Un mapping que relaciona el ID de cada tarea con su estructura correspondiente.

Estructura Tarea:

id:	Identificador único de la tarea.
titulo:	Título de la tarea.
descripcion:	Descripción detallada de la tarea.
completada:	Indica si la tarea ha sido completada (true o false).
creador:	Dirección de Ethereum del usuario que creó la tarea.

Eventos:

TareaCreada:	Se emite cuando se crea una nueva tarea.
TareaCompletada:	Se emite cuando una tarea es marcada como completada.

Funciones Principales**1.- crearTarea(string _titulo, string _descripcion)**

Descripción: Permite a un usuario crear una nueva tarea proporcionando un título y una descripción.

Funcionamiento:

1. Incrementa contadorTareas para asignar un nuevo ID.
2. Crea una nueva instancia de Tarea y la almacena en el mapping tareas.
3. Establece completada como false y registra la dirección del creador.
4. Emite el evento TareaCreada.

2.- completarTarea(uint256 _id)

Descripción: Permite al creador de una tarea marcarla como completada.

Funcionamiento:

1. Verifica que el ID de la tarea exista.
2. Confirma que el solicitante es el creador de la tarea.

3. Verifica que la tarea no esté ya completada.
4. Actualiza el estado de completada a true.

5. Emite el evento TareaCompletada.

3.- obtenerTarea(uint256 _id)

Descripción: Proporciona los detalles de una tarea específica.

Funcionamiento:

1. Verifica que el ID de la tarea exista.
2. Retorna los atributos de la tarea solicitada.
3. Seguridad y Validaciones

Control de Acceso: Solo el creador de una tarea puede marcarla como completada, evitando modificaciones no autorizadas.

Validaciones de Existencia: Las funciones verifican que las tareas solicitadas existan antes de operar sobre ellas, previniendo errores y posibles ataques.

Funcionamiento del Frontend

Lenguajes y utilidades empleados.

HTML y CSS: Para la estructura y el diseño de la interfaz de usuario.

JavaScript: Para la lógica del frontend y la interacción con el contrato inteligente.

Web3.js: Biblioteca que permite la comunicación entre el frontend y la blockchain de Ethereum.

MetaMask: Extensión de navegador que funciona como billetera digital, facilitando la gestión de cuentas y la firma de transacciones.

Componentes de la Interfaz

Formulario de Creación de Tareas:

Campos de Entrada:

1. Título de la Tarea: Campo de texto para ingresar el título.

2. Descripción: Área de texto para ingresar la descripción.

Botón "Crear Tarea":

Al hacer clic, invoca la función `crearTarea` del contrato inteligente.

Botón "Obtener Tareas":

Actualiza y muestra la lista de tareas existentes en la blockchain.

Lista de Tareas:

Muestra las tareas con sus detalles:

1. Título
2. Descripción
3. Estado: Indica si la tarea está pendiente o completada.
4. Botón "Marcar como Completada": Disponible solo para tareas pendientes creadas por el usuario actual. Al hacer clic, invoca la función `completarTarea` del contrato.

Interacción con el Contrato Inteligente

Conexión a MetaMask:

Al cargar la DApp, se solicita al usuario que conecte su cuenta de MetaMask.

La DApp utiliza la cuenta conectada para interactuar con la blockchain.

Configuración de Web3.js:

Se crea una instancia de Web3 utilizando `window.ethereum` proporcionado por MetaMask.

Se inicializa el contrato inteligente usando la dirección del contrato y su ABI.

Funciones JavaScript Principales:

`crearTarea()`:

Captura los valores de los campos de entrada.

Verifica que los campos no estén vacíos.

Llama a `contract.methods.crearTarea` y envía una transacción desde la cuenta del usuario.

Muestra una alerta al completar la operación.

completarTarea(id):

Llama a `contract.methods.completarTarea` con el ID de la tarea.

Envía una transacción desde la cuenta del usuario.

Actualiza la lista de tareas tras la confirmación.

obtenerTareas():

1. Obtiene el contador de tareas desde el contrato.
2. Itera desde el número total de tareas hasta 1 para mostrar las tareas en orden descendente (las más recientes primero).
3. Para cada tarea, llama a `contract.methods.obtenerTarea` y crea elementos HTML para mostrar la información.
4. Verifica si el usuario actual es el creador de la tarea para mostrar el botón "Marcar como Completada" cuando corresponda.

Diseño y Experiencia de Usuario

Estética Moderna:

Uso de colores oscuros con acentos vibrantes para un aspecto profesional.

Tipografía elegante y legible con la fuente Montserrat.

Interactividad:

Animaciones y transiciones suaves al interactuar con elementos como botones y tarjetas de tareas.

Feedback inmediato al usuario mediante alertas y actualizaciones en la interfaz.

Responsividad:

Diseño adaptativo que asegura una experiencia óptima en dispositivos móviles y de escritorio.

Flujo de Uso de la DApp

Acceso a la DApp:

El usuario abre la DApp en su navegador y es recibido por una interfaz atractiva y centrada.

Funcionamiento de la dApp.

Se solicita al usuario que conecte su cuenta de MetaMask para interactuar con la blockchain.

Creación de una Nueva Tarea:

El usuario ingresa el título y descripción de la tarea.

Al hacer clic en "Crear Tarea", se envía una transacción para registrar la tarea en la blockchain.

MetaMask solicita la confirmación de la transacción.

Tras la confirmación, la tarea aparece en la lista de tareas.

Visualización de Tareas:

El usuario puede ver todas las tareas existentes, incluyendo aquellas creadas por otros usuarios.

Cada tarea muestra su estado actual (Pendiente o Completada).

Marcado de Tareas como Completadas:

Si el usuario es el creador de una tarea pendiente, puede marcarla como completada.

Al hacer clic en "Marcar como Completada", se envía una transacción para actualizar el estado de la tarea.

MetaMask solicita la confirmación de la transacción.

Tras la confirmación, el estado de la tarea se actualiza en la interfaz.

Consideraciones Técnicas

Gas y Costos de Transacción:

Cada interacción que modifica el estado de la blockchain (crear o completar tareas) requiere una transacción y, por tanto, el pago de gas.

El gas es una medida del costo computacional que se paga en Ether (ETH).

Seguridad:

Al usar MetaMask, las claves privadas del usuario nunca se exponen a la DApp, manteniendo la seguridad de las cuentas.

El contrato inteligente implementa controles de acceso para asegurar que solo los creadores puedan modificar sus tareas.

Persistencia de Datos:

Los datos se almacenan en la blockchain, lo que garantiza su inmutabilidad y disponibilidad.

Incluso si la DApp deja de existir, los datos permanecerán en la blockchain.

Ventajas de Utilizar una DApp

Descentralización:

No depende de un servidor centralizado; la lógica y los datos residen en la blockchain.

No hay punto único de falla, aumentando la resiliencia de la aplicación.

Transparencia:

Las transacciones y estados son públicos y verificables en la blockchain.

Los usuarios pueden confiar en que las reglas del contrato inteligente se aplican de manera consistente.

Propiedad de Datos:

Los usuarios tienen control sobre sus datos y activos digitales.

Las interacciones requieren el consentimiento explícito del usuario a través de MetaMask.

Limitaciones y Consideraciones

Escalabilidad y Velocidad:

Las transacciones en la blockchain pueden tardar tiempo en confirmarse, lo que afecta la inmediatez de las operaciones.

La congestión de la red puede aumentar los tiempos de espera y los costos de gas.

Experiencia de Usuario:

Los usuarios necesitan conocimientos básicos sobre criptomonedas y el uso de billeteras digitales.

La necesidad de pagar gas por transacciones puede ser una barrera para algunos usuarios.

Posibles Mejoras y Extensiones**1.- Implementación en Redes de Capa 2:**

Migrar la DApp a soluciones de escalado como Polygon o Optimism para reducir costos y tiempos de transacción.

2.- Autenticación y Perfil de Usuarios:

Ampliar el contrato para permitir perfiles de usuario más detallados y funcionalidades sociales.

3.- Notificaciones y Actualizaciones en Tiempo Real:

Integrar servicios que permitan notificaciones push cuando una tarea es completada o asignada.

4.- Interfaz Multilingüe:

Adaptar la DApp para soportar múltiples idiomas, ampliando su alcance internacional.

Conclusión

La DApp "Tópicos de Tecnologías - Gestor de Tareas" es una aplicación que demuestra el poder y la versatilidad de las tecnologías blockchain y las aplicaciones descentralizadas. Proporciona una

plataforma transparente y segura para la gestión de tareas, con un diseño moderno y atractivo que realza la experiencia del usuario.

Al aprovechar las características de Ethereum y las herramientas de desarrollo modernas, esta DApp ofrece una solución innovadora que puede ser ampliada y personalizada para satisfacer necesidades más complejas en el futuro.