

En este módulo, tu desafío es actualizar tu contrato existente **KipuBankV2** hacia una aplicación DeFi más avanzada y real — el **KipuBankV3**.

Tu nueva versión debe aceptar **cualquier token soportado por Uniswap V2**, intercambiarlo a **USDC**, y acreditar el monto resultante en el balance del usuario — todo mientras se respeta el **límite máximo del banco (bank cap)** y se preserva la lógica principal de KipuBankV2.

Este examen consolida tu entendimiento de integración de protocolos, enrutamiento de tokens y composabilidad de smart contracts.

Objetivos:

Al finalizar este examen, tu 'KipuBankV3' debe ser capaz de:

1. **Manejar cualquier token intercambiable en Uniswap V2.** Además de los tokens nativos y USDC, los usuarios deben poder depositar cualquier token soportado por Uniswap V2.
2. **Ejecutar swaps de tokens dentro del smart contract.**
Tu contrato debe intercambiar automáticamente los tokens depositados a USDC usando el router de Uniswap V2.
3. **Preservar la funcionalidad de KipuBankV2.**
Debe mantenerse el soporte a depósitos, retiros y lógica de ownership.
4. **Respetar el límite del banco.**
El valor total en USDC almacenado nunca debe superar el 'bankCap', incluso después de realizar los swaps.

Requisitos Clave:

1. Integración con Uniswap V2

2. Depósitos de tokens generalizados

Los usuarios deben poder depositar:

- * Token nativo (ETH)
- * USDC (almacenado directamente)
- * Cualquier otro token ERC20 que tenga un par directo con USDC en Uniswap V2.

* Si se deposita un token distinto a USDC, debe **intercambiarse automáticamente a USDC** mediante Uniswap V2 y luego acreditarse al balance en USDC.

3. Respetar el Bank Cap

* El balance total nunca debe superar 'bankCap'.

* El resultado del swap (en USDC) debe considerarse antes de actualizar el balance del usuario.

4. Preservar la funcionalidad de KipuBankV2

* Mantener todas las características previas:

* Control del owner

* Mecanismos de depósito/retiro para tokens soportados

Criterios de Evaluación:

* Correctitud

¿La integración realiza correctamente los swaps a USDC y actualiza el balance respetando el límite del banco?

* Seguridad y Gas

¿Se manejan de forma segura las aprobaciones, transferencias, protección contra reentradas y consideraciones de gas?

* Calidad de Código

¿El código es limpio, modular, legible y consistente con las mejores prácticas?

* Dependencias

¿Las librerías y helpers externos son usados de forma adecuada? .

* Aprendizaje

Maneján de forma correcta los temas vistos a lo largo del curso, tanto del gitbook como lo dado en clases.

Entregables:

Debes enviar lo siguiente a través de la plataforma:

1. URL del repositorio en GitHub

Un repo público que contenga:

- * El smart contract actualizado dentro de la carpeta `/src`.
- * Un archivo `README.md` que incluya:
 - * Una explicación de alto nivel de las mejoras implementadas y el porqué.
 - * Instrucciones de despliegue e interacción.
 - * Notas sobre decisiones de diseño o trade-offs.

2. URL al contrato verificado en ethscan, routescan o blockscout.

Este proyecto final de `KipuBankV3` mostrará tu habilidad para **integrar protocolos reales, aplicar buenas prácticas de seguridad, y desarrollar aplicaciones DeFi listas para producción** en el ecosistema Ethereum. Un paso firme para ser desarrollador Web 3.