

Flujo de Depósito de HBAR (WHBAR) en Bonzo Finance

Contexto: Bonzo Finance y el uso de WHBAR

Bonzo Finance es un protocolo de préstamos sobre Hedera basado en Aave v2, adaptado para funcionar con la EVM de Hedera y su **Hedera Token Service (HTS)** ¹. En Bonzo (al igual que en Aave), para depositar HBAR se utiliza un token envuelto llamado **WHBAR** (Wrapped HBAR). WHBAR es un token ERC-20 que representa HBAR 1:1; esto permite que HBAR (que normalmente no es un ERC-20) pueda integrarse en contratos inteligentes como cualquier otro token ². En esencia, al depositar HBAR en el protocolo, éste se convierte (se “envuelve”) en WHBAR, y luego el usuario recibe un **aToken** (aWHBAR) que representa su depósito dentro de la plataforma. A continuación, se detalla el flujo de trabajo completo al hacer un “supply” (depósito) de HBAR en Bonzo Finance.

Contrato LendingPool: Proxy e Implementación

El contrato **LendingPool** de Bonzo en mainnet es un contrato proxy ubicado en la dirección `0x236897c518996163E7b313aD21D1C9fCC7BA1afc` ³. Esto significa que esta dirección no contiene la lógica directamente, sino que delega las llamadas a un contrato de implementación (LendingPoolImpl). HashScan muestra que `0x2368...` es un `InitializableImmutableAdminUpgradeabilityProxy` (un proxy upgradeable); por eso solo expone funciones como `admin()`, `upgradeTo()` etc., y no las funciones de lectura/escritura del pool en sí. La lógica real de `deposit()` reside en el contrato de implementación asociado (por ejemplo, la dirección de LendingPoolImpl en mainnet es `0x5290b075d737606fccccA2f745D7337E0fCe633B` según la documentación ⁴). Al interactuar con el LendingPool (por ejemplo, llamando a `deposit`), en realidad el proxy forwarda la llamada a la implementación donde ocurre la ejecución real de la función.

Asociación del token WHBAR antes del depósito

Antes de realizar un depósito de HBAR/WHBAR, es necesario **asociar el token WHBAR** a la cuenta del usuario en la red Hedera. Hedera requiere que las cuentas “acepten” (associate) explícitamente cada token HTS que vayan a poseer. En este caso, WHBAR es un token HTS (ID `0.0.1456986` en Hedera) ⁵, por lo que la interfaz de Bonzo primero le solicita al usuario una **transacción de asociación**. Esta asociación permite que la cuenta del usuario pueda recibir y poseer WHBAR o aWHBAR sin que las transferencias fallen. (Nota: El aToken aWHBAR de Bonzo está implementado como un contrato ERC-20 inteligente, no como token nativo HTS ⁵, por lo que probablemente no requiere asociación. Sin embargo, la asociación de WHBAR sí es obligatoria para interactuar con HBAR envuelto.) Una vez que el usuario firma y envía la asociación de WHBAR, ya está listo para depositar HBAR en el protocolo.

deposit()

Cuando el usuario realiza un **“supply”** de HBAR en Bonzo (por ejemplo, 1.01 HBAR), internamente se siguen estos pasos:

1. **Llamada a deposit desde el front-end:** El usuario inicia la transacción de depósito a través de la UI o herramienta, la cual invoca `LendingPool.deposit(address asset, uint256 amount, address onBehalfOf, uint16 referralCode)`. Para el caso de HBAR, el **asset** indicado será la dirección de WHBAR (por ejemplo, `0x00000000000000000000000000000000163b5a` corresponde al token WHBAR ⁵) y el **amount** será la cantidad en *tinybars* (HBAR tiene 8 decimales en HTS). El parámetro **onBehalfOf** suele ser la dirección del mismo usuario (a menos que esté depositando en favor de otra cuenta), y **referralCode** típicamente es 0. Importante: como HBAR es la moneda nativa, el usuario **adjunta HBAR como valor (msg.value) en la transacción** por el monto especificado. Por ejemplo, para depositar `1.01 HBAR`, `amount = 101,000,000` (ya que 1 HBAR = 100,000,000 tinybars) y se envían 1.01 *h* como value en la llamada al contrato.
2. **Envolviendo HBAR a WHBAR (mint de WHBAR):** Al recibirse la llamada, la implementación de `deposit()` detecta que el activo depositado es WHBAR (el wrapper de HBAR). En consecuencia, el contrato procede a **convertir la HBAR enviada en tokens WHBAR equivalentes**. Básicamente, el protocolo utiliza el contrato especial de WHBAR para acuñar la cantidad correspondiente de WHBAR a partir de los HBAR recibidos ². Cada HBAR depositado resulta en 1 WHBAR acuñado, manteniendo una paridad 1:1. Internamente, esto se logra mediante las precompiladas de HTS: el contrato `LendingPool/WHBAR` invoca la función de **mint** del token WHBAR (para la cual tiene permisos de supply) creando, por ejemplo, 1.01 WHBAR si el usuario envió 1.01 HBAR.

Figura: Diagrama ilustrativo de cómo funciona el wrapping (envolver) de HBAR a WHBAR y su posterior unwrapping. Al depositar HBAR en el contrato de WHBAR, se acuña la misma cantidad de tokens WHBAR; al retirar, se queman WHBAR y se libera la HBAR subyacente ².

En el contexto de Bonzo, este paso ocurre dentro de la ejecución de `deposit()`. La transacción en Hedera muestra un evento **Token Mint** del token WHBAR (ID `0.0.1456986`) por el monto depositado (p. ej. 1.01000000 WHBAR) ², lo que confirma que se crearon nuevos tokens WHBAR equivalentes a los HBAR entregados.

1. **Transferencia de WHBAR al pool (reserva):** Tras acuñar los WHBAR, el protocolo necesita almacenar estos tokens como liquidez del pool. En Aave/Bonzo, los depósitos se custodian en el contrato del **aToken**, que actúa como reserva del activo. Por ello, el siguiente sub-paso es transferir los WHBAR recién acuñados hacia la dirección del **aWHBAR** (el aToken correspondiente a HBAR). En el registro de la transacción, esto aparece como un **Crypto Transfer** de WHBAR: por ejemplo, se observó una transferencia de **1.01 WHBAR** desde la cuenta emisora (la tesorería o contrato WHBAR) hacia la cuenta **0.0.7308509**, que corresponde al contrato aWHBAR en mainnet ⁵. En otras palabras, los 1.01 WHBAR generados se movieron al contrato que mantiene la liquidez de WHBAR en el protocolo (es decir, el aToken actúa como “caja fuerte” del subyacente). Después de este paso, esos WHBAR quedan bajo control del protocolo (reserva del pool), no del usuario.

2. **Mint de aToken (aWHBAR) para el usuario:** Ahora que el protocolo tiene 1.01 WHBAR adicionales en su reserva, emite al usuario sus recibos de depósito en forma de **aWHBAR**. El aWHBAR es un token ERC-20 interest-bearing (con rendimiento) que representa la participación del usuario en el pool de WHBAR. El contrato aWHBAR acuña exactamente la cantidad equivalente al depósito (1.01 aWHBAR) y se la asigna al usuario. En la transacción, esto se refleja en eventos del contrato aWHBAR, típicamente un evento **Transfer** (ERC20) desde la dirección `0x0` (indicando mint) hacia la dirección del usuario, por el valor 1.01 aWHBAR, y un evento interno de **Mint** en el aToken que registra la acción. De esta forma, el usuario pasa a ser titular de tokens aWHBAR, mientras que el pool mantiene los WHBAR subyacentes. (Cabe destacar que en la documentación de Bonzo se listan tanto el token WHBAR como su aToken correspondiente: por ejemplo, WHBAR tiene dirección `0x...163b5a` y su aToken es `0x6e96...15af32`, que coincide con la cuenta `0.0.7308509` donde fueron depositados los WHBAR ⁵).

3. **Eventos de depósito y confirmación:** Finalmente, el contrato LendingPool emite un evento `Deposit` para registrar la operación. Este evento incluye información como el activo depositado (WHBAR), el **user** que realizó el depósito (o `onBehalfOf` si aplica), el monto y el referral code. En los logs provistos por HashScan, se puede ver un evento asociado a la dirección del LendingPool (proxy) con un topic que corresponde al hash del evento `Deposit`. Dicho evento muestra como parámetro indexado la dirección del token WHBAR (`...163b5a`), indicando que se trató de un depósito de WHBAR, y muy probablemente incluye la dirección del usuario y la cantidad depositada en los campos de datos (no indexados). Junto con esto, la transacción muestra un resultado **SUCCESS**, confirmando que el depósito se realizó correctamente. A partir de este punto, el usuario tiene **aWHBAR** en su cuenta (que crecerán en valor a medida que genere intereses), y puede eventualmente retirar su depósito solicitando la operación inversa (quemando aWHBAR para recuperar HBAR, lo cual implicará *unwrapping* de WHBAR de vuelta a HBAR por el protocolo).

Cómo replicar el depósito en otro front-end o aplicación

Para replicar este flujo en una herramienta propia (por ejemplo, un agente de IA o script), se deben seguir los pasos clave descritos arriba:

- **Asociación del token:** Asegurarse de que la cuenta desde la cual se depositará tenga asociado el token WHBAR (0.0.1456986). Esto implica enviar una transacción de asociación de token a la red Hedera antes del primer depósito. Sin esta asociación, cualquier intento de transferencia o recepción de WHBAR fallará por reglas de la red.
- **Llamada al contrato de LendingPool:** Invocar la función `deposit` del LendingPool proxy en Hedera Mainnet (0x236897c518996163E7b313aD21D1C9fCC7BA1afc³). Los parámetros deben ser:
 - `asset`: la dirección del token a depositar. Para HBAR, usar la dirección ERC20 de WHBAR 0x00163b5a (que representa el ID 0.0.1456986)⁵.
 - `amount`: la cantidad a depositar en la unidad mínima del token. Para HBAR, calcular los tinybars necesarios (ej.: 1 HBAR = 100,000,000, por lo que 1.01 HBAR = 101,000,000).
 - `onBehalfOf`: la dirección del beneficiario del depósito. Usualmente la misma dirección del usuario que deposita, a menos que se quiera depositar en la cuenta de otro.

- `referralCode`: un código de referidor (en Bonzo normalmente **0** si no hay programa de referidos activo).
- **Enviar HBAR junto a la transacción:** Al ejecutar la llamada, incluir la cantidad de HBAR equivalente al `amount` como valor nativo de la transacción (por ejemplo, en SDKs de Hedera EVM, esto se especifica como `payableAmount`). Esto es esencial ya que el contrato tomará esos HBAR y los convertirá en WHBAR internamente. Si no se envía HBAR (o si se envía una cantidad distinta al `amount` especificado), la transacción podría revertirse por no tener fondos para acuñar los WHBAR.
- **Procesamiento interno:** El contrato se encargará de los pasos internos (mint de WHBAR, transfer a reserva, mint de aToken, etc.) automáticamente. No es necesario que la aplicación los maneje manualmente, solo debe manejar la llamada correctamente. Tras la confirmación de éxito, la aplicación puede verificar que el **balance de aWHBAR del usuario aumentó** en la cantidad depositada y/o que el evento `Deposit` fue emitido.

En resumen, el flujo de un depósito de HBAR en Bonzo involucra convertir HBAR a WHBAR bajo el capó y otorgar al usuario tokens aWHBAR como recibo de su aporte de liquidez. La plataforma maneja la lógica de wrapping y emisión de aTokens de forma transparente ², por lo que al replicarlo en otro frontend es crucial preparar la transacción correctamente (asociaciones previas y parámetros adecuados) para que el contrato LendingPool pueda ejecutar el depósito tal como se observa en la plataforma oficial. Con estos pasos, un agente externo puede interactuar con el protocolo Bonzo en Hedera y realizar depósitos de manera correcta, obteniendo los aTokens correspondientes a cada suministro realizado.

Sources: Bonzo Finance Docs and Hedera Docs ³ ⁵ ¹ ² (análisis del flujo basado en documentación oficial y registros de transacciones en HashScan).

¹ Overview | Bonzo Finance Documentation

<https://docs.bonzo.finance/hub/>

² Wrapped HBAR (WHBAR) | Hedera

<https://docs.hedera.com/hedera/core-concepts/smart-contracts/wrapped-hbar-whbar>

³ ⁴ ⁵ Protocol Contracts | Bonzo Finance Documentation

<https://docs.bonzo.finance/hub/developer/protocol-contracts>