

#### INTERACTUANDO CON SOLANA

- La forma en la que nos comunicamos con Solana es enviando transacciones a la red.
- Se puede pensar en una transacción como una solicitud para procesar una serie de operaciones, donde su éxito o fracaso es indivisible: o se ejecuta todo o no se ejecuta nada.

#### TRANSACCIONES EN SOLANA

• Las transacciones consisten en un conjunto de instrucciones que interactúan con varios programas (contratos inteligentes) en Solana.

• Estas instrucciones se procesan en el orden en que se agregan a la transacción.

#### TRANSACCIONES EN SOLANA

- Las transacciones son atómicas, lo que significa que todas las instrucciones deben procesarse correctamente, de lo contrario toda la transacción fallará.
- Las transacciones no pueden exceder el tamaño máximo de 1232 bytes.

#### APROBANDO TRANSACCIONES

• Para que una transacción sea aprobada, debe seguir un proceso que incluye la construcción de la transacción, la firma por parte de los participantes relevantes, la propagación a través de la red, y finalmente, su validación y confirmación por parte de los validadores.

#### ESTRUCTURA DE UNA TRANSACCIÓN

• Una transacción de Solana consiste principalmente en un arreglo de firmas y un mensaje que contiene toda la información de las instrucciones asociadas a la transacción.

TRANSACCIÓN

ARREGLO DE FIRMAS (SY BYTES X # FIRMAS) MENSAJE (METADATA + CUENTAS)

MAX 1232 BYTES

# ESTRUCTURA DE UNA TRANSACCIÓN: FIRMAS (SIGNATURES)

- Una lista de firmas de las claves públicas que autorizan la transacción. Cada transacción debe ser firmada por las cuentas que son marcadas como firmantes en la cabecera.
- La firma asegura que la transacción es válida y que proviene de la persona o entidad que controla las cuentas involucradas.
- Cada firma ocupa 64 bytes del tamaño total de la transacción.

# ESTRUCTURA DE UNA TRANSACCIÓN: MENSAJE (MESSAGE)

- Es un conjunto de datos que incluye el contenido de las cuentas, instrucciones y demás metadatos de la transacción, y es lo que realmente se firma para autorizar la transacción. El mensaje es lo que se verifica y procesa en la red.
- Cada cuenta incluida en el mensaje de la transacción, requiere 32 bytes del tamaño total.

# ESTRUCTURA DE UNA TRANSACCIÓN: MENSAJE (MESSAGE)

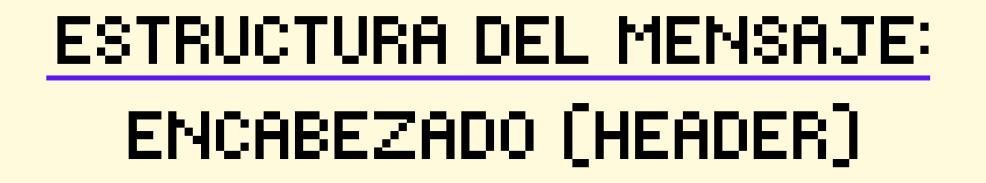
MENSAJE

ENCABEZADO (3 BYTES)

ARREGLO DE CUENTAS (32 BYTES X # CUENTAS) BLOQUE DE REFERENCIA (32 BYTES) ARREGLO DE INSTRUCCIONES

# ESTRUCTURA DEL MENSAJE: ENCABEZADO (HEADER)

- Número de firmantes requeridos: Indica cuántas firmas se necesitan para validar la transacción.
- Número de cuentas solo lectura: Define cuántas de las cuentas involucradas solo se leerán, sin que sus datos sean modificados.



ENCABEZADO (3 BYTES)

# DE FIRMAS REQUERIDAS (U8) # DE CUENTAS SOLO LECTURA QUE REQUIEREN FIRMAS (U8)

# DE CUENTAS SOLO LECTURA QUE NO REQUIEREN FIRMAS (U8)

# ESTRUCTURA DEL MENSAJE: CUENTAS (ACCOUNTS)

- Una lista de direcciones de cuentas involucradas en la transacción. Las cuentas pueden ser: Cuentas de datos o Cuentas de Programas.
- Cada cuenta tiene propiedades específicas, como si es de solo lectura o si necesita ser firmada.

# ESTRUCTURA DEL MENSAJE: CUENTAS (ACCOUNTS)

ARREGLO DE CUENTAS (32 BYTES X #CUENTAS)

CUENTAS QUE REQUIEREN FIRMA (32 BYTES X # CUENTAS QUE REQUIEREN FIRMA)

CUENTAS QUE NO REQUIEREN FIRMA (32 BYTES X # CUENTAS QUE NO REQUIEREN FIRMA)

LECTURA/ESCRITURA

SOLO LECTURA

LECTURA/ESCRITURA

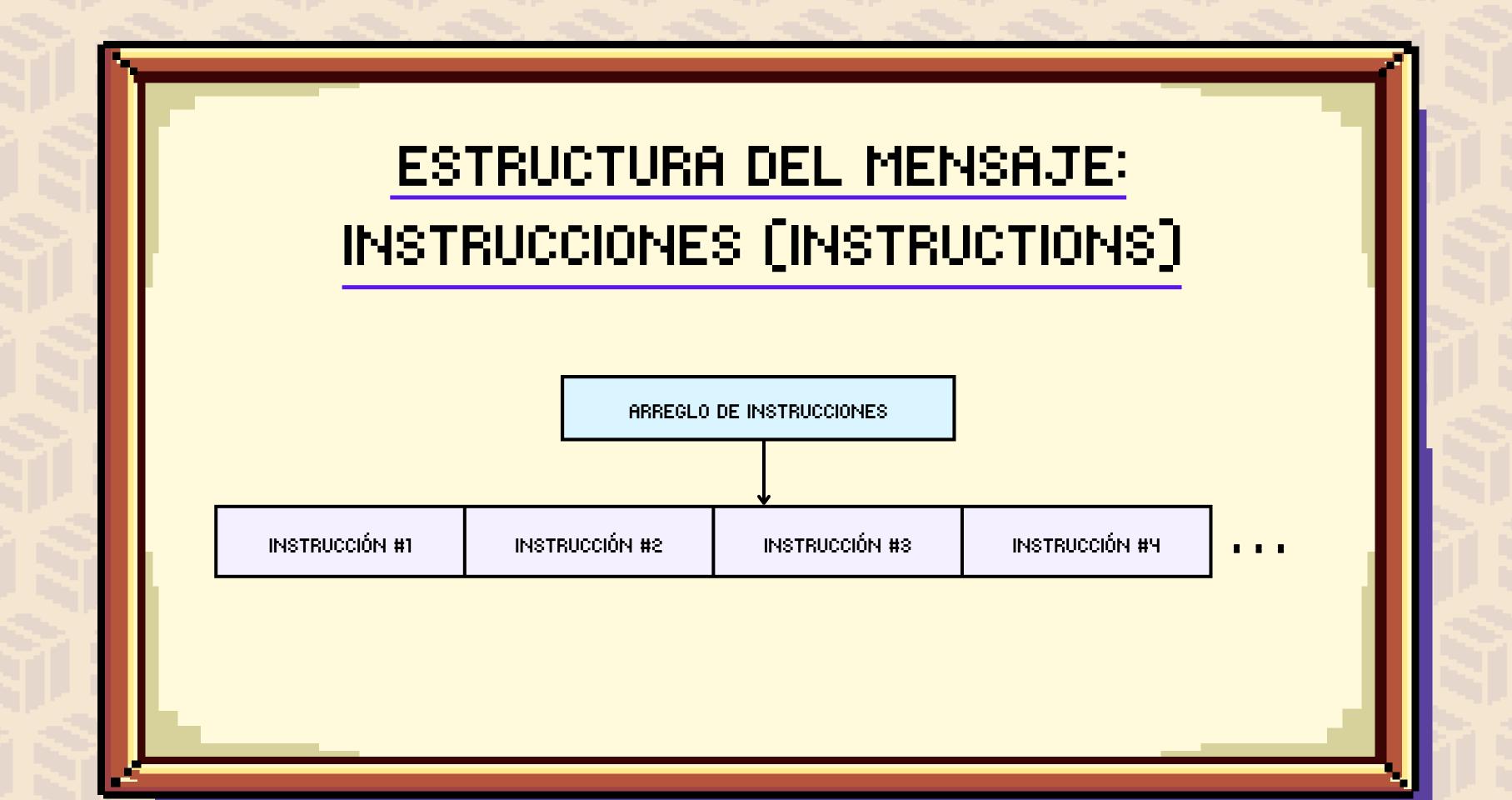
SOLO LECTURA

# ESTRUCTURA DEL MENSAJE: BLOQUE DE REFERENCIA (RECENT BLOCKHASH)

• Incluye el hash (identificador criptográfico) de un bloque reciente de la red para evitar que la transacción sea válida indefinidamente. Esto asegura que la transacción sea procesada dentro de un cierto periodo de tiempo.

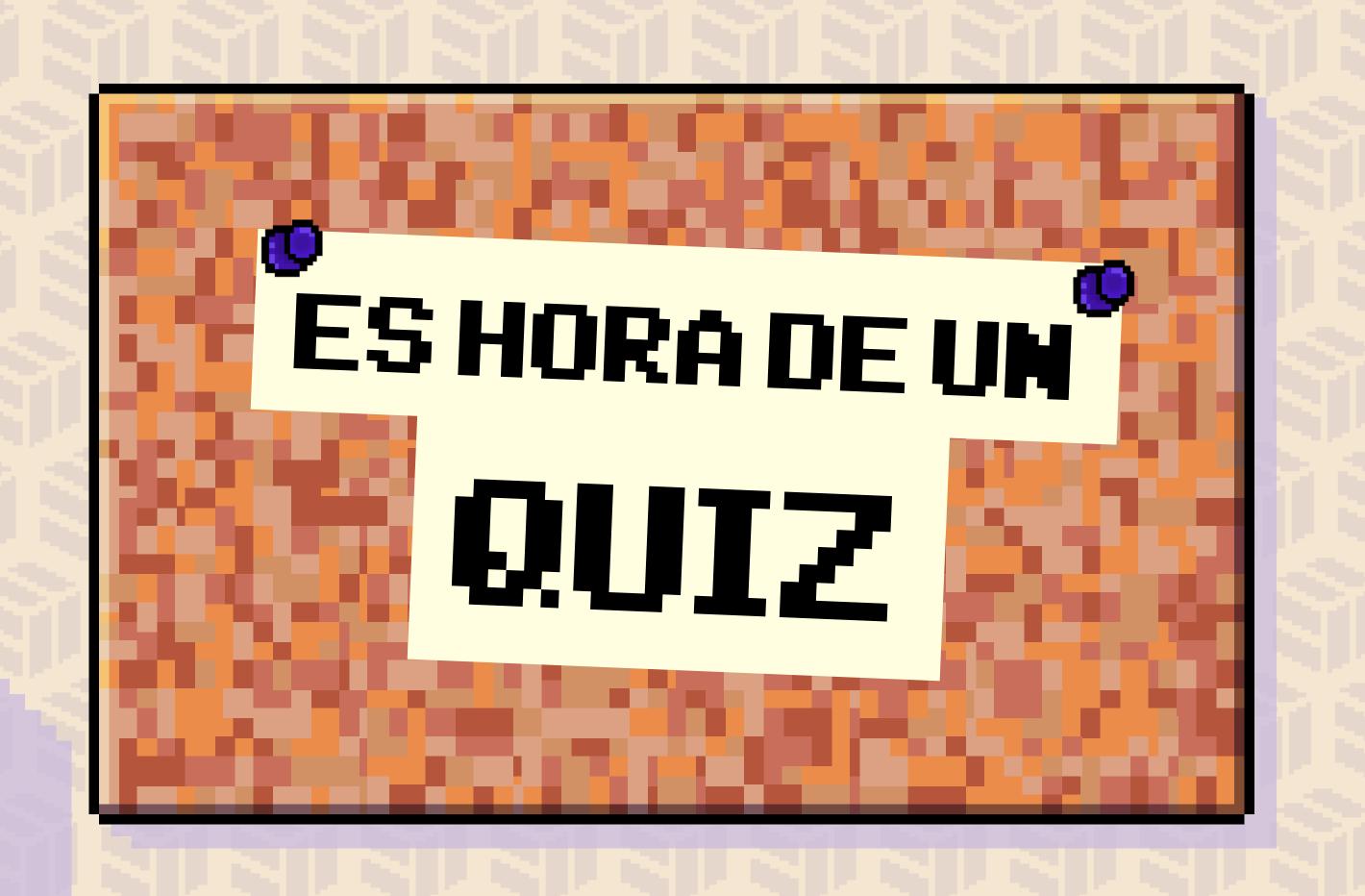
# ESTRUCTURA DEL MENSAJE: INSTRUCCIONES (INSTRUCTIONS)

- Un arreglo de instrucciones que define las operaciones que la transacción pretende realizar.
- Cada instrucción incluye la información necesaria para ejecutar una operación específica en la red.



#### EJECUTANDO UNA TRANSACCIÓN

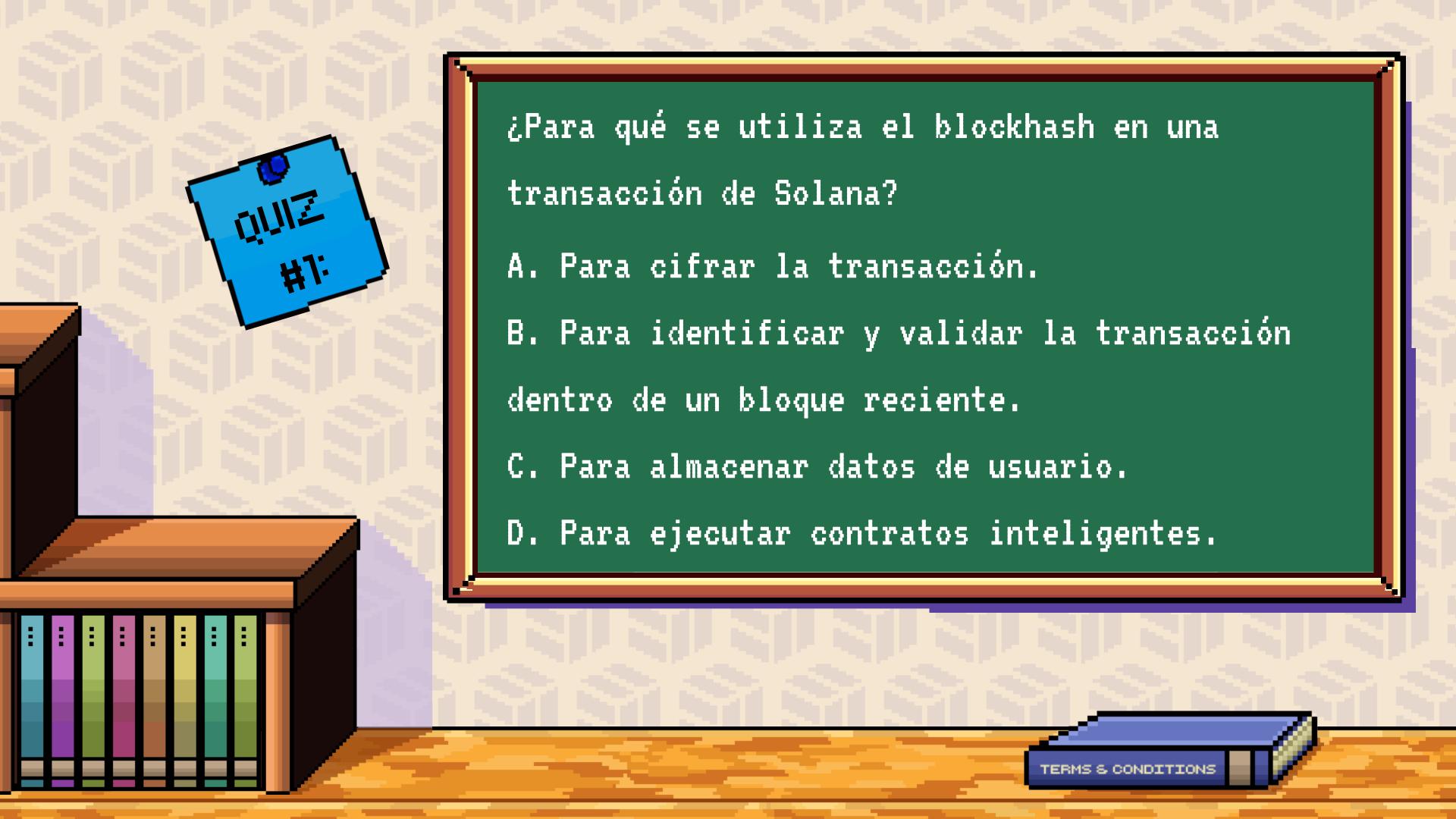
- Verificación de firmas: Las firmas son verificadas contra las claves públicas de las cuentas firmantes.
- Ejecución de instrucciones: Las instrucciones son ejecutadas en el orden especificado.
- Actualización del Estado: Los cambios resultantes se aplican a las cuentas involucradas.

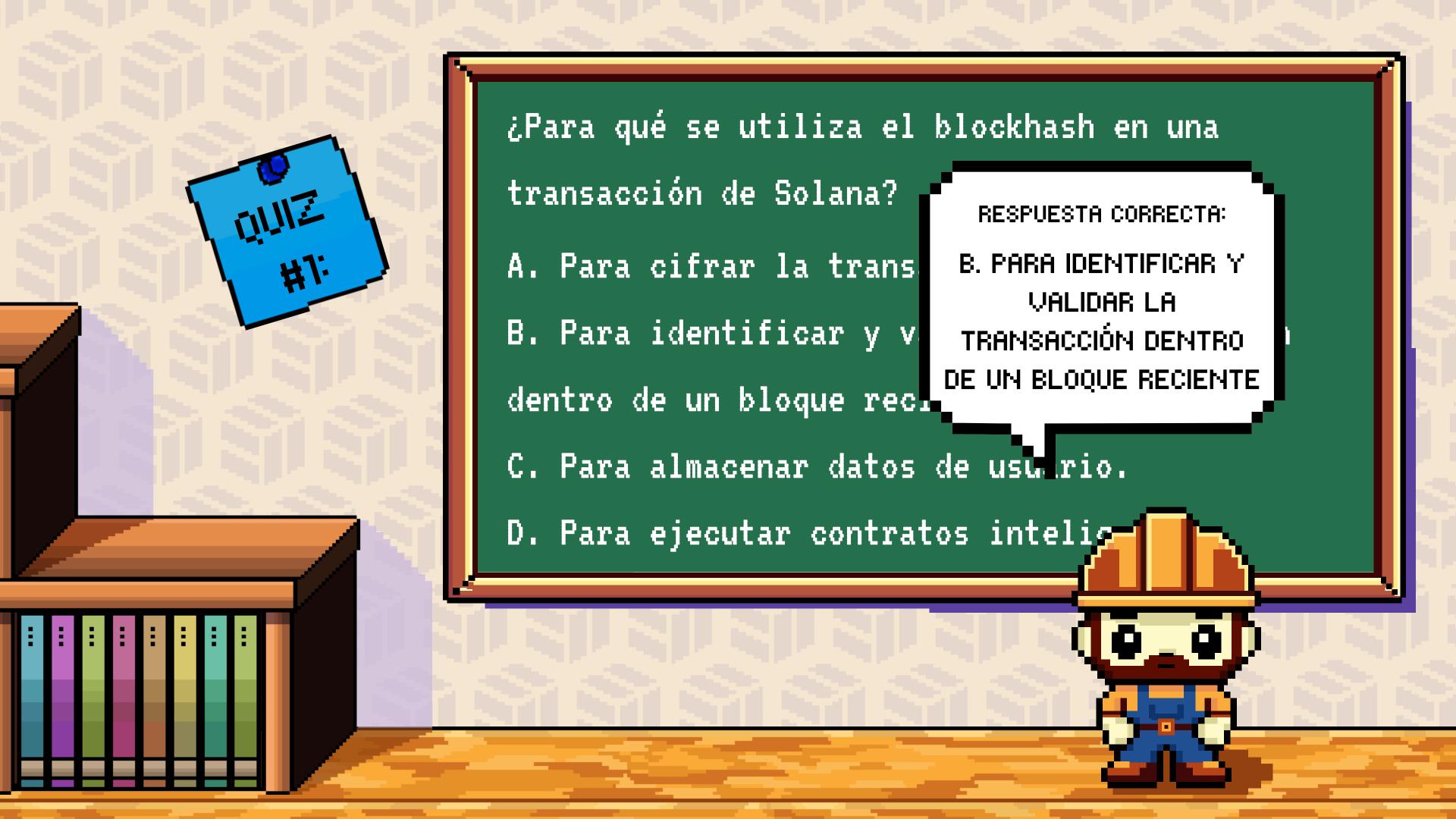




E

LUN	MAR	t
Ø	7	
13	14	
<b>20</b>	21	

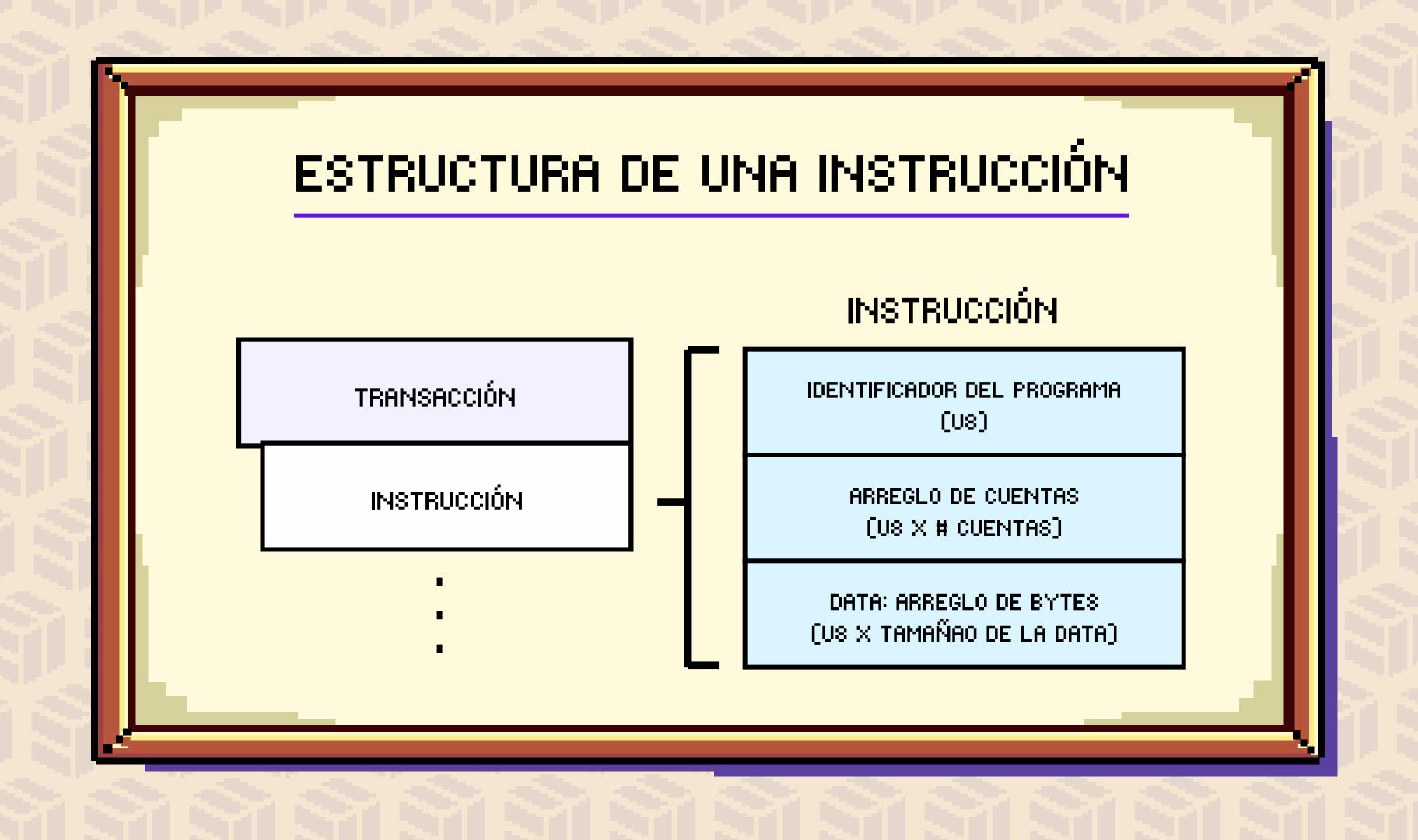




# ¿QUÉ SON LAS INSTRUCCIONES?

• Es la unidad más pequeña de lógica de ejecución en un programa en Solana.

 Activan programas específicos que, al ser ejecutados, implementan los cambios deseados en la red.

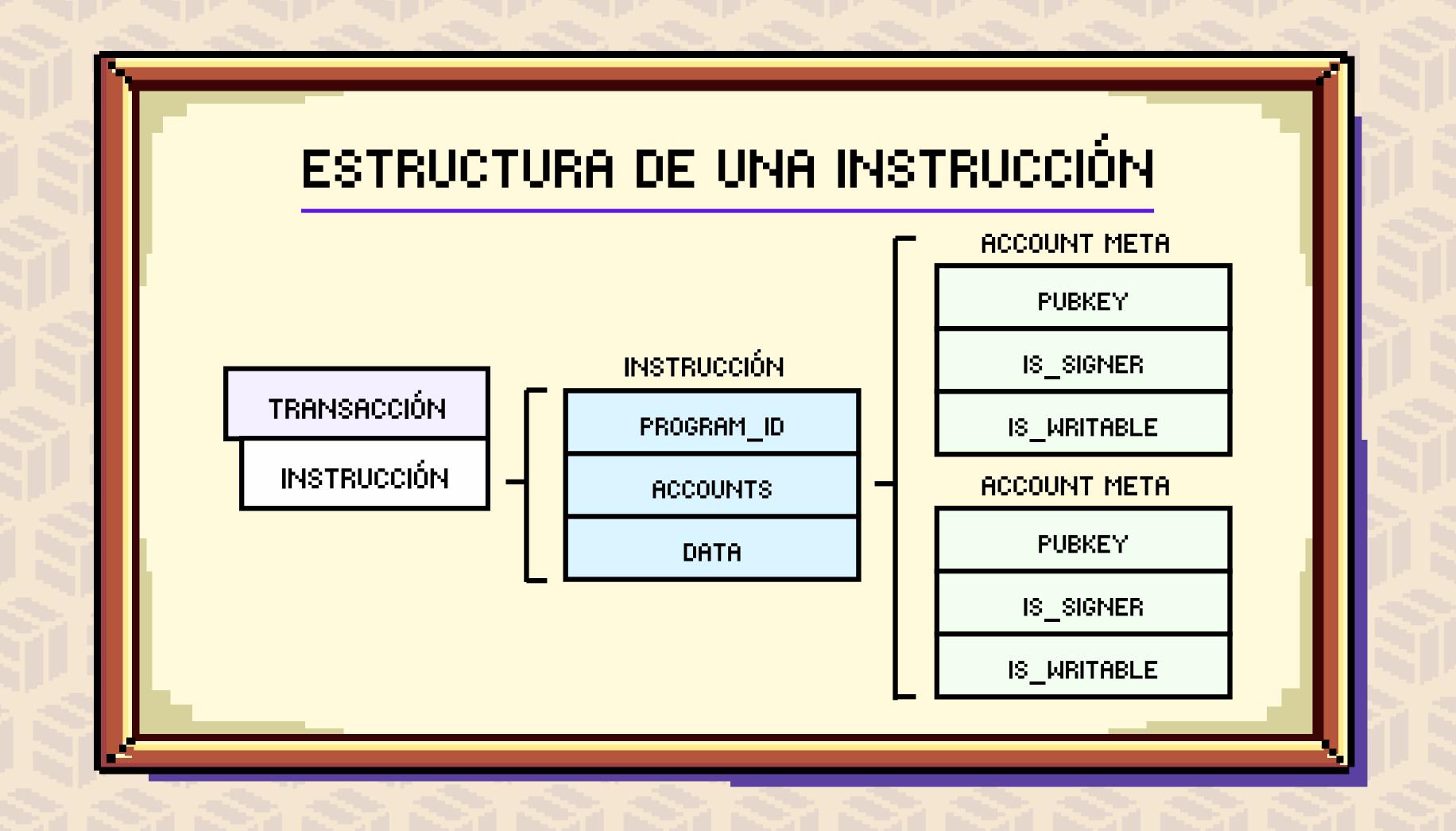


# ESTRUCTURA DE UNA INSTRUCCIÓN

CAMPO	DESCRIPCIÓN
program_id: identificador del programa	Dirección del programa que será invocado
accounts: arreglo de cuentas	Lista de cuentas cuentas necesarias para la instrucción, incluyendo otros programas.
data	Código serializado que dice al programa sobre qué acción debe realizar.

#### LISTA DE CUENTAS DE UNA INSTRUCCIÓN

• Dentro de una instrucción las cuentas requeridas para su ejecución se representan en una estructura llamada AccountMeta que contiene información específica sobre la cuenta y los permisos de lectura/escritura asociados, además de establecer si la cuenta es una cuenta firmate.



### ESTRUCTURA DE UNA INSTRUCCIÓN

CAMPO	DESCRIPCIÓN
pubkey	La dirección en la cadena de bloques de una cuenta
is_signer	Especifica si la cuenta es requerida como un firmante en la transacción
is_writable	Especifica si los datos de la cuenta serán modificados

#### PROCESANDO INSTRUCCIONES

- Al especificar todas las cuentas requeridas por una instrucción, y si cada cuenta es escribible, las transacciones se pueden procesar en paralelo.
- Por ejemplo, dos transacciones que no incluyen ninguna cuenta que escriba en el mismo estado pueden ejecutarse al mismo tiempo.

### COMISIONES (FEES) DE TRANSACCIÓN

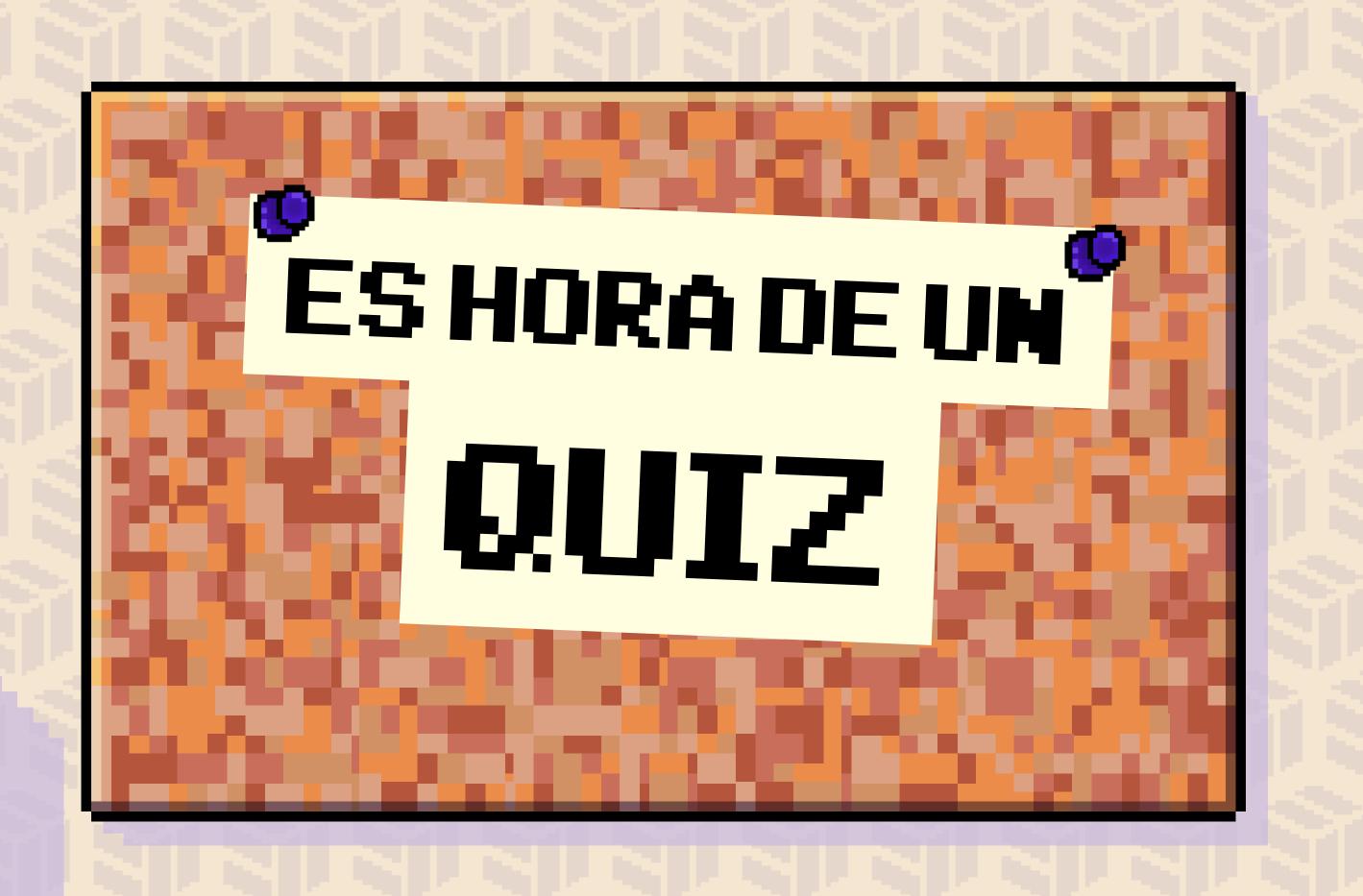
- Cada transacción en Solana tiene una tarifa base mínima que se paga a los validadores por incluir la transacción en un bloque.
- La tarifa base está diseñada para cubrir los costos computacionales y de almacenamiento asociados con la validación y ejecución de las transacciones.
- Actualmente, la comisión base de transacción de Solana se establece en un valor fijo de 0,000005 \$SOL.

#### COBRO DE COMISIONES

- Las transacciones requieren al menos una cuenta que haya firmado la transacción y que sea mutable.
- Estas cuentas firmantes mutables se serializan primero en la lista de cuentas y la primera de ellas siempre se utiliza como la cuenta pagadora de comisiones (fee payer).
- Antes de procesar cualquier instrucción de la transacción, se deducirá del balance de la cuenta del "fee payer" para pagar las comisiones de transacción.

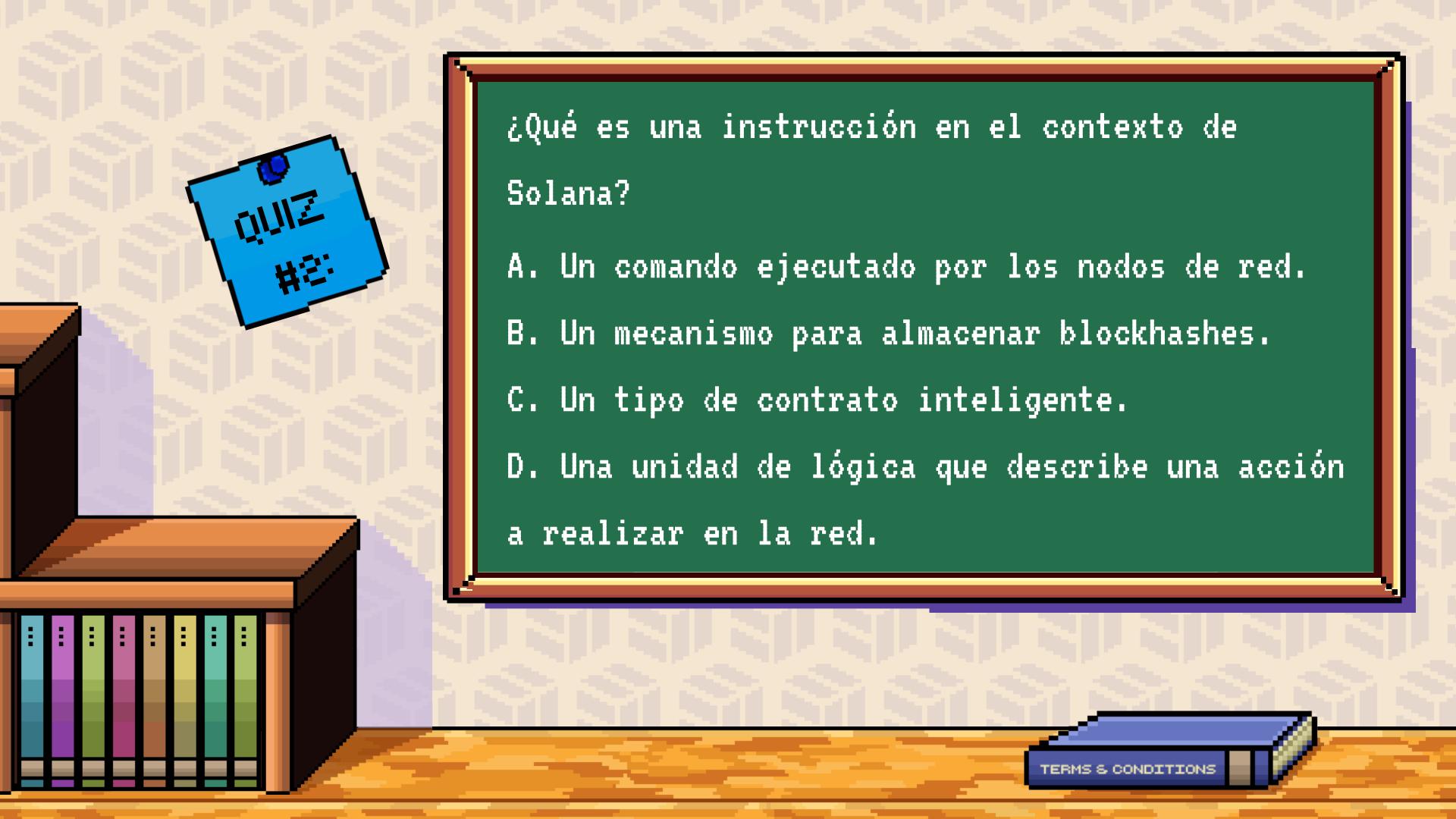
# EJEMPLO PRÁCTICO: TRANFIRIENDO \$SOL

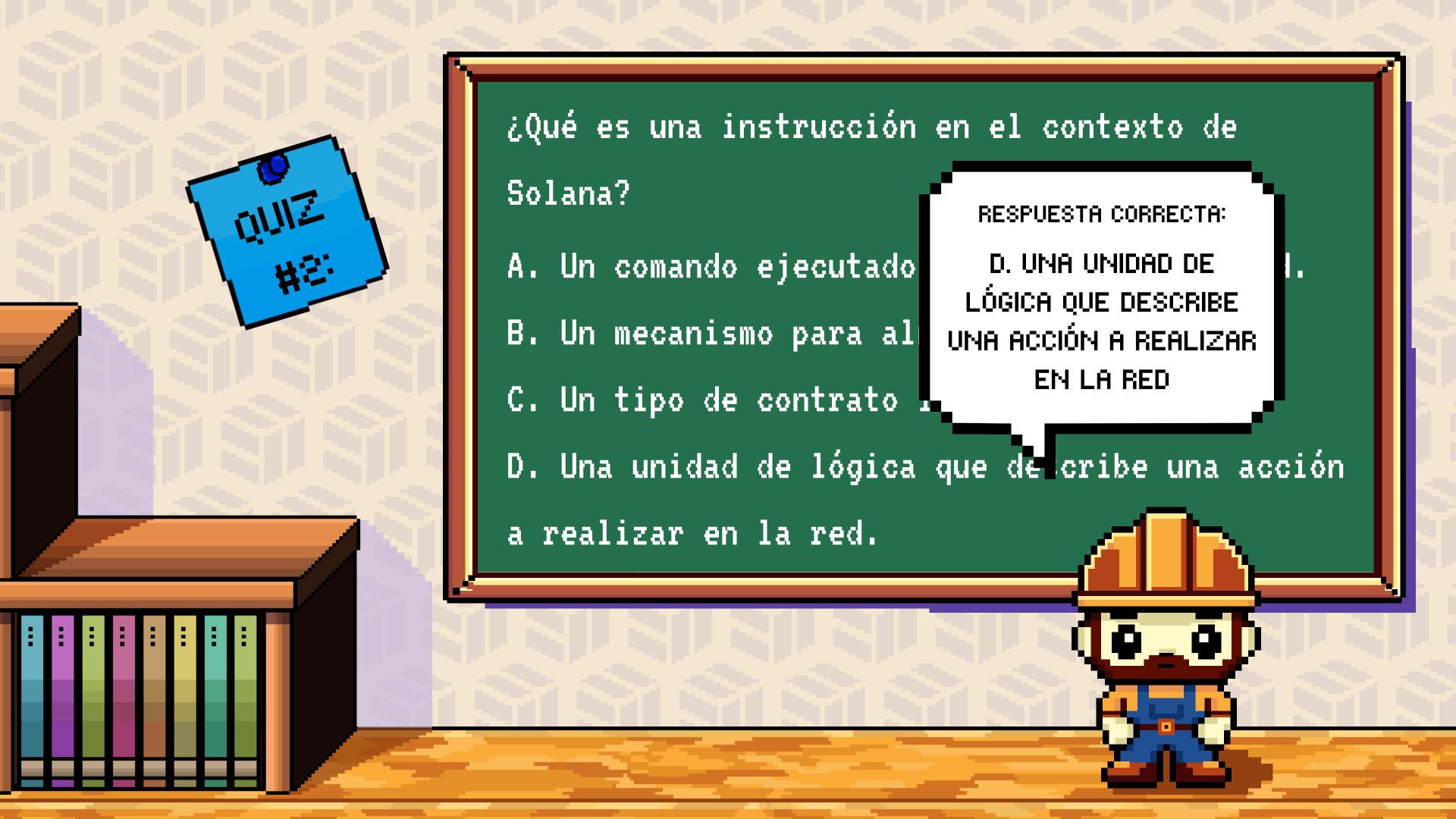
• Podemos transferir saldo \$SOL de una cuenta a otra a través de la herramienta de línea de comandos (CLI) de Solana y confirmar el resultado de la transacción en el explorador de la red.





LUN	MAR	t
٩	7	
13	14	
	21	





# ESCRIBIENDO INSTRUCCIONES EN ANCHOR

 Anchor es un framework para el desarrollo de programas en Solana que simplifica la interacción con contratos inteligentes al proporcionar herramientas y macros que automatizan muchas tareas comunes.

# ESCRIBIENDO INSTRUCCIONES EN ANCHOR

 Escribir instrucciones usando Anchor implica definir las funciones y datos que tu programa manejará, así como la estructura de las cuentas que son requeridas para ejecutar la instrucción; este conjunto de elementos se conoce como contexto de la instrucción.

#### DEFINIENDO UNA INSTRUCCIÓN: CUENTAS

- En Anchor, se utiliza la macro #[account] para identificar la estructura de los datos de una cuenta.
- Principalmente #[account] permite identificar las cuentas, incluso dentro del contexto de una instrucción.

## DEFINIENDO EL CONTEXTO DE UNA INSTRUCCIÓN: CUENTAS

- Dentro del contexto de una instrucción, la macro #[account] para definir una cuenta y para validar o acceder cada una de las cuentas dentro del contexto de la instrucción.
- Esta macro contiene información relevante a la hora de crear o modificar una cuenta.

# DEFINIENDO EL CONTEXTO DE UNA INSTRUCCIÓN: #[ACCOUNT]

- Mutabilidad: #[account(mut)] indica que la cuenta puede ser modificada.
- Inicialización: #[account(init)] se usa para inicializar una nueva cuenta.
- Requisitos: Se pueden incluir condiciones adicionales como seeds o bump para cuentas PDA (Program Derived Accounts).

## DEFINIENDO EL CONTEXTO DE UNA INSTRUCCIÓN: DATOS

- Se utiliza la macro #[derive(Account)] para permitir la serialización y deserialización de datos personalizados que serán enviados o recibidos por las instrucciones.
- #[derive(Accounts)] informa a Anchor que una estructura contiene las cuentas necesarias para la instrucción.

# DEFINIENDO UNA INSTRUCCIÓN: FUNCIONES

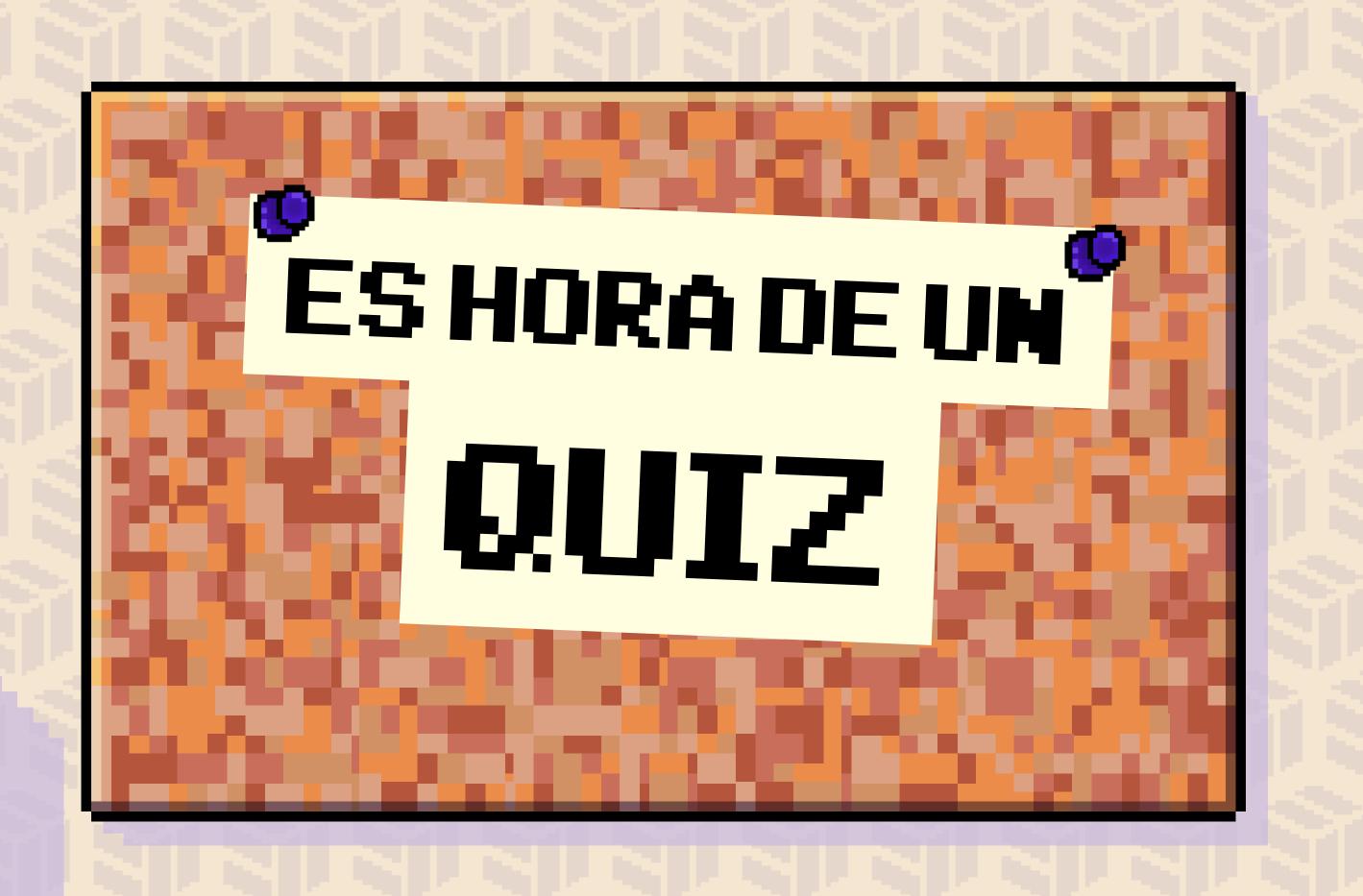
- La macro #[program] se utiliza para definir el módulo principal de un programa de Solana.
- Esta macro convierte las funciones definidas dentro del módulo en puntos de entrada de las instrucciones del programa.
- El punto de entrada de una instrucción es la función que se invoca cuando una transacción llama a dicha instrucción.

# <u>DEFINIENDO UNA INSTRUCCIÓN:</u> FUNCIONES

• El punto de entrada es donde se ejecuta la lógica de la instrucción, utilizando el contexto que incluye las cuentas necesarias y los datos de entrada específicos que son proporcionados por la transacción que invoca a la instrucción.

# EJEMPLO PRÁCTICO: ESCRIBIENDO UNA INSTRUCCIÓN EN ANCHOR

 Podemos almacenar una serie de datos en la red definiendo una instrucción que cree una cuenta que contenga la estructura de datos que se desea almacenar.





LUN	MAR	t
٩	7	
13	14	
	21	

