

HEAVY DUTY CAMP V4

ACERCA
DEL
BOOTCAMP

CLASE
#1



DE QUÉ VA EL HEAVY DUTY CAMP:

- Objetivo: Proporcionar herramientas y conocimientos básicos sobre el desarrollo de programas en Solana utilizando Rust y Anchor.
- Duración: 4 semanas (del 20/01/25 al 14/02/25).
- Modalidad: clases teóricas y prácticas, diversos retos semanales y un proyecto final.

METODOLOGÍA DE TRABAJO

- 3 clases semanales (lunes, miércoles y viernes 19:00 UTC)
- Actividades prácticas para cada clase vista.
- Un proyecto final.
- Quizzes interactivos durante las clases.
- Leaderboard basado en puntos obtenidos al realizar las actividades prácticas y quizzes.
- NFT regalo para quienes finalicen el proyecto final.

CRONOGRAMA

Pensado para construir las bases sobre el desarrollo de programas en Solana.

Semana #1	Semana #2	Semana #3	Semana #4
Entendiendo el modelo de programación de Solana	Escribiendo y modificando datos en Solana	Desarrollando un programa en Solana	Proyecto final

HERRAMIENTAS A UTILIZAR

A lo largo de estas 4 semanas utilizaremos herramientas web que no requieren configuración de un entorno local:

- Solana Playground (<https://beta.solpg.io/>)
- Solana Explorer (<https://explorer.solana.com/>)
- Solana Faucet (<https://faucet.solana.com/>)
- Circle Faucet (<https://faucet.circle.com/>)

DOCUMENTACIÓN RECOMENDADA

- Documentación oficial de Solana (<https://solana.com/es/docs>)
- Guías de desarrollo (<https://solana.com/es/developers/guides>)
- Documentación de Anchor (<https://www.anchor-lang.com>)
- Libro de Rust (<https://book.rustlang-es.org/>)

HEAVY DUTY CAMP V4

CONCEPTOS
BÁSICOS

CLASE
#1



LA BLOCKCHAIN

- En español, cadena de bloques, es una red de almacenamiento distribuido que permite mantener una lista creciente e inmutable de registros, llamados bloques, de manera segura y descentralizada.

IMPORTANCIA DE LA BLOCKCHAIN

- Ha transformado la forma en que se almacenan, gestionan y comparten los datos.
- Todas las transacciones y datos almacenados en la blockchain están accesibles para todos los participantes de la red.
- Es immutable, lo que significa que una vez que los datos se registran en un bloque y este es añadido a la cadena, no pueden ser modificados ni eliminados.

EL VALOR DE LA BLOCKCHAIN

- El verdadero valor de la blockchain radica en su capacidad para proporcionar un sistema de almacenamiento de datos seguro, transparente, y descentralizado que elimina la necesidad de intermediarios, reduce costos, y aumenta la eficiencia. Su capacidad para garantizar la integridad de los datos y permitir la trazabilidad en tiempo real la convierte en una herramienta poderosa en una variedad de industrias.

¿CÓMO FUNCIONA?

- Los datos se almacenan en bloques, cada bloque contiene una serie de transacciones, un sello de tiempo y un enlace al bloque anterior en la cadena.



AGREGANDO BLOQUES

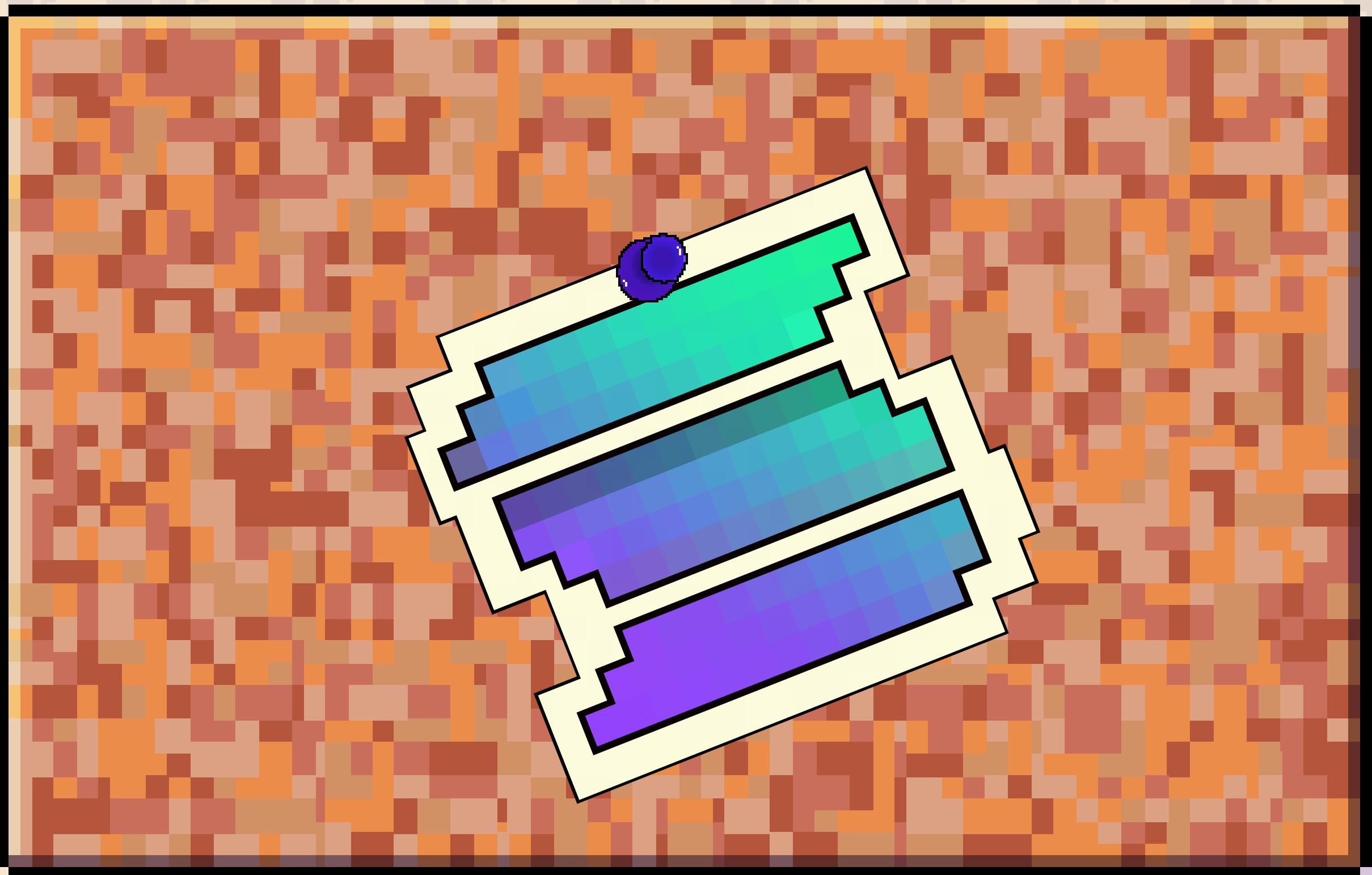
- Agregar un nuevo bloque a una blockchain implica un proceso estructurado que asegura que todas las transacciones contenidas en el bloque sean válidas y que el consenso de la red se mantenga.
- Este trabajo lo realizan los nodos de la red, que compiten por ser seleccionados para argegar nuevos bloques a la cadena.

COMO SE AGREGA UN NUEVO BLOQUE

1. Los usuarios crean y envian transacciones a la red.
2. Los nodos seleccionan las transacciones pendientes y las agrupan en un bloque candidato.
3. El nodo verifica que cada transacción en el bloque candidato sea válida y propone el nodo a la red.
4. Un bloque es seleccionado y enviado para su verificación.
5. El nuevo bloque es verificado por los demás nodos y agregado a la cadena de bloques.
6. El nodo creador del bloque seleccionado recibe una recompensa.

NA

SAB	DOM
4	5
11	12
18	19
25	26



LA BLOCKCHAIN DE SOLANA

- Alta velocidad y rendimiento: Conocida por su capacidad de procesar hasta 65,000 transacciones por segundo (TPS).
- Proof of History (PoH): Crea un registro histórico que prueba que un evento ha ocurrido en un momento específico en el tiempo.
- Bajas comisiones: Gracias a su eficiencia, las comisiones por transacción en Solana son extremadamente bajas, lo que la hace atractiva para desarrolladores y usuarios.

MODELO DE PROGRAMACIÓN DE SOLANA

- El componente básico del modelo de programación de Solana son las cuentas.
- La forma en que se organizan los datos en Solana se asemeja a una base de datos clave-valor, donde cada entrada en la base de datos es una cuenta.
- Las cuentas almacenan el estado, pudiendo contener todo tipo de datos, desde programas hasta saldos de tokens.

ORGANIZACIÓN DE DATOS EN SOLANA

Cuentas

Cuenta
individual [

Clave	Valor
Dirección de la cuenta	Información de la cuenta
Dirección de la cuenta	Información de la cuenta
Dirección de la cuenta	Información de la cuenta

CUENTAS EN SOLANA

- Cada cuenta es identificable por su dirección única (Public Key).
- Solo el dueño de la cuenta de datos puede modificar sus datos ydebitar saldo.
- Cualquiera puede enviar saldo a una cuenta de datos.

ESTRUCTURA DE UNA CUENTA EN SOLANA

Public Key:

"HBYDJrvqm...NNnsvgfrH8c7"

Address

Account

AccountInfo

lamports

owner

executable

data

rent_epoch

ESTRUCTURA DE UNA CUENTA EN SOLANA

Campo	Descripción
lamports	El número de lamports que tiene la cuenta
owner	El programa dueño de esa cuenta
executable	Indica si la cuenta es de tipo ejecutable
data	El array de bytes de datos que tiene la cuenta
rent_epoch	La siguiente época que esta cuenta deberá renta

CUENTAS EN SOLANA

- Las cuentas pueden almacenar hasta 10 MB de datos, que pueden consistir en código de programa ejecutable o estado del programa (datos).
- Para calcular el tamaño de una cuenta es necesario conocer la estructura de los datos que almacena.

TAMAÑO DE UNA CUENTA EN SOLANA

- Conocer el tamaño de una cuenta es importante a la hora de desarrollar en Solana.
- Para calcular el tamaño de una cuenta en Solana, se suma el tamaño (en bytes) de cada uno de los elementos que contiene dicha cuenta.
- Las cuentas almacenan un mínimo de 128 bytes en datos adicionales que la red utiliza para gestionar y mantener la cuenta.

TAMAÑO DE UNA CUENTA EN SOLANA

Tipo de Dato	Tamaño
u8 / i8	1 byte
u16 / i16	2 bytes
u32 / i32	4 bytes
u64 / i64	8 bytes
Public key	32 bytes

Tipo de Dato	Tamaño
f32	4 bytes
f64	8 bytes
bool	1 byte
String de tamaño n	n + 4 bytes

EJEMPLO CALCULO DEL TAMAÑO DE UNA CUENTA

Suponiendo un cuenta que almacena:

- Un entero de 8 bits
- Una dirección de Solana (Public Key) de 32 bytes
- Un flotante de 64 bits

$$\text{Tamaño} = \frac{1 \text{ byte}}{(8 \text{ bits})} + \frac{32 \text{ bytes}}{\text{bytes}} + \frac{8 \text{ bytes}}{(64 \text{ bits})} + \frac{128 \text{ bytes}}{\text{bytes}} = 169 \text{ bytes}$$

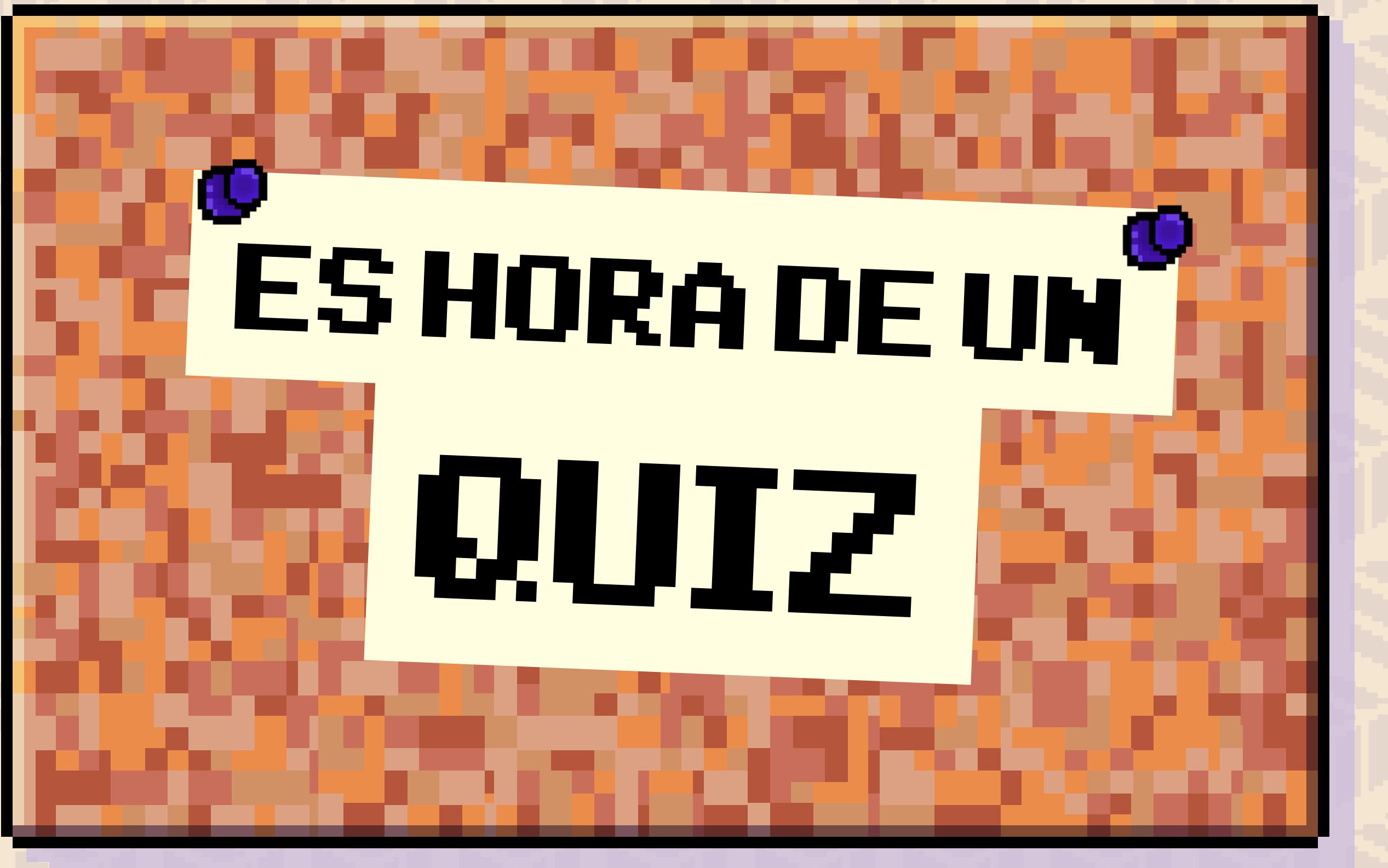
CUENTAS EN SOLANA

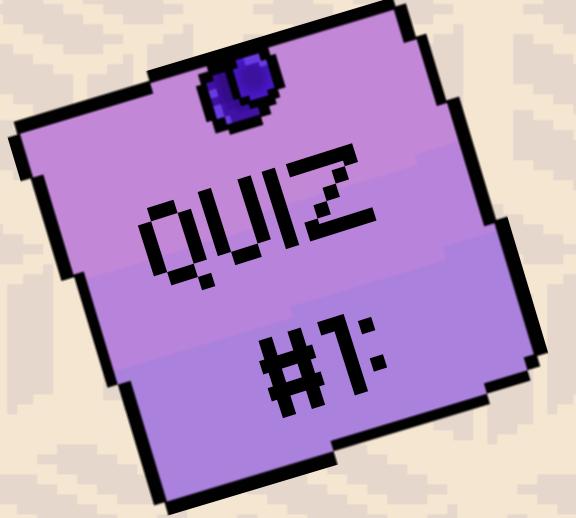
- Para mantener las cuentas en Solana se debe pagar una renta en $\$SOL$ que es proporcional al espacio que ocupa la cuenta en la red; de lo contrario, la cuenta será eliminada junto con los datos que almacena.



19

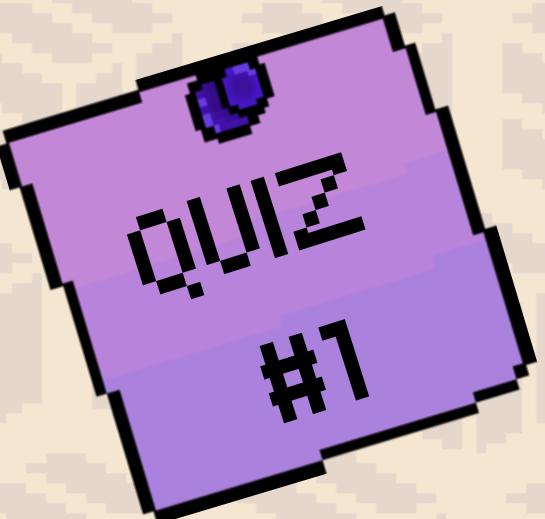
LUN	MAR	M
6	7	
13	14	
20	21	22
27	28	29





¿Qué elemento es esencial en la estructura de una cuenta en Solana?:

- A. PublicKey (Clave pública)
- B. Clave privada
- C. Contrato inteligente
- D. Token asociado



¿Qué elemento es esencial en la

estructura de una cue

- A. PublicKey (Clave p
- B. Clave privada
- C. Contrato inteligente
- D. Token asociado

RESPUESTA
CORRECTA:

A. PUBLICKEY
(CLAVE PÚBLICA)



RENTA

- Mecanismo diseñado para gestionar a largo plazo el almacenamiento de datos en Solana.
- Evita que las cuentas ocupen espacio en la blockchain de forma indefinida sin estar siendo utilizadas.

RENTA

- Cada cuenta debe tener una cantidad de \$SOL proporcional al tamaño de los datos almacenados en ella.
- Se puede pensar en la renta como una especie de depósito, ya que los \$SOL se pueden recuperar en su totalidad cuando se cierra dicha cuenta.

RENTA

- La renta se retiene en el balance normal de cada cuenta y puede reclamarse cuando se cierra la cuenta.
- La renta en Solana se fija para toda la red, basándose principalmente en un valor llamado "lamports por byte por año".

CALCULANDO LA RENTA

La renta en Solana se calcula en función del espacio que ocupa la cuenta. El costo de almacenamiento por byte por año es fijo (0.00000348 \$SOL), y el cálculo general es el siguiente:

$$\text{Renta} = \frac{\text{Costo por byte}}{\text{por año}} \times \text{Tamaño de la cuenta* (en bytes)}$$

*ADEMÁS DE LOS DATOS ALMACENADOS, EL TAMAÑO DE LA CUENTA DEBE INCLUIR EL ENCABEZADO QUE TIENE UN TAMAÑO FIJO DE 128 BYTES.

EJEMPLO CÁLCULO DE LA RENTA

Supongamos una cuenta de 1K bytes:

- Tamaño de los datos: 872 bytes
- Tamaño del encabezado: 128 bytes
- Costo por byte por año: 0.00000348 SOL

$$\text{Renta} = 0.00000348 \times 1,000 = 0.00348 \text{ \$SOL}$$

RENT-EXEMPTION

En lugar de pagar una tarifa constante, la renta se puede "prepagar" depositando la cantidad de $\$SOL$ que cubra los costos de almacenamiento de la cuenta por un periodo de 2 años. Esto hace que la cuenta esté exenta de pagar la renta siempre que contenga esa cantidad de $\$SOL$ y nunca menos.

CÁLCULO DE RENT EXEMPTION

Calcular la rent-exemption es similar a calcular la renta, con la diferencia de que se multiplica la renta por la cantidad de años mínimos para que una cuenta sea exenta de renta, actualmente ese periodo de tiempo es de 2 años.

$$\text{Rent exemption} = \frac{\text{Costo por byte por año}}{\text{Tamaño total de la cuenta (en bytes)}} \times 2 \text{ años}$$

EJEMPLO DE CÁLCULO DE RENT EXEMPTION

Supongamos una cuenta de 1K bytes:

- Tamaño TOTAL de la cuenta: 1,000 bytes
- Costo por byte por epoch: 0.00000348 SOL
- Período mínimo: 2 años

$$\text{Rent}_{\text{Exemption}} = 0.00000348 \times 1,000 \times 2 = 0.006 \text{ \$SOL}$$

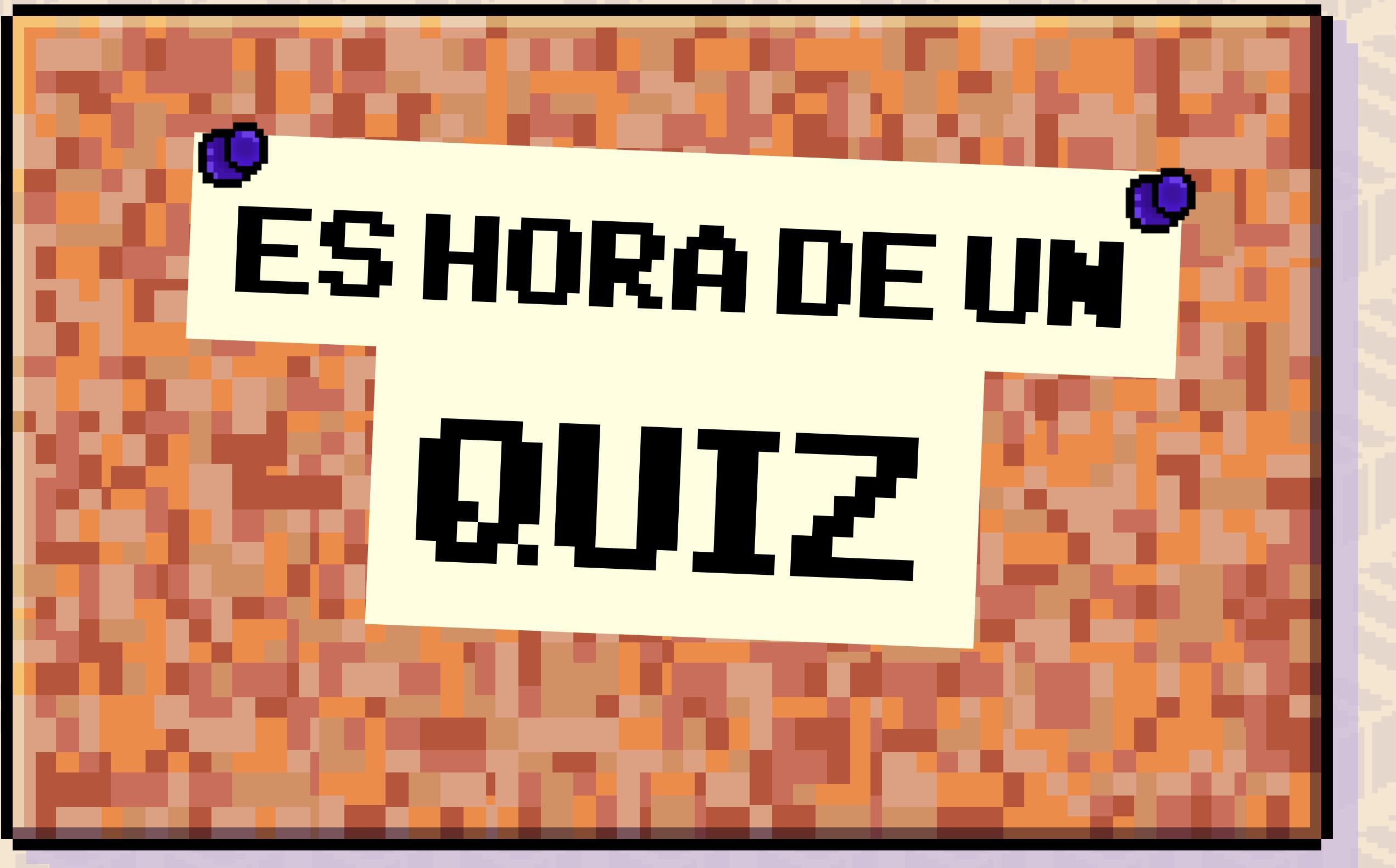
RENTA

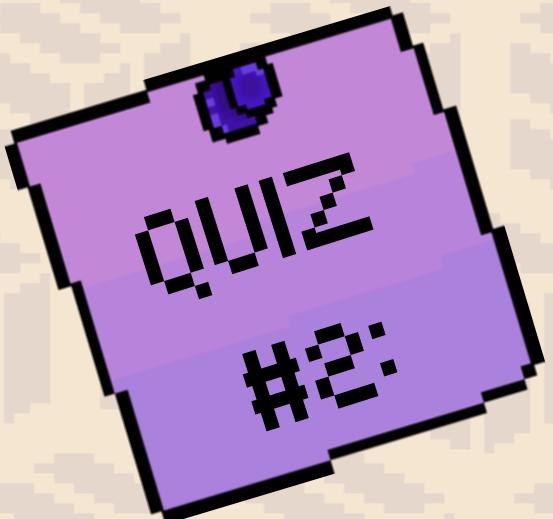
- Al crear cuentas en Solana, se requiere que estas se inicialicen con un balance en lamports lo suficientemente alto para ser exento de renta y permanecer en la red. Cualquier transacción que intente reducir el balance de una cuenta por debajo de su respectivo balance mínimo para ser exento de renta fallará (a menos que el balance se reduzca exactamente a cero).



19

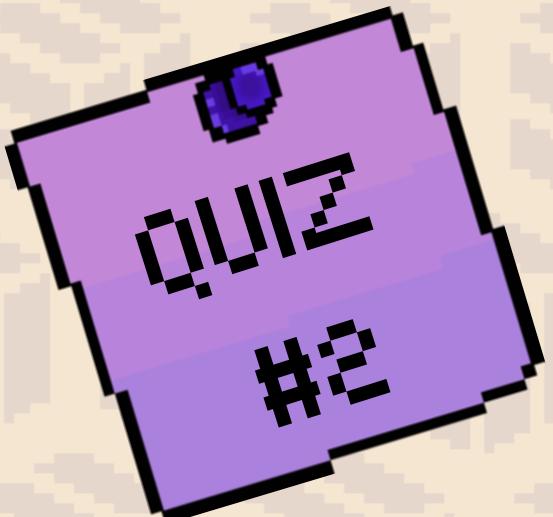
LUN	MAR	M
6	7	
13	14	
20	21	22
27	28	29





¿Qué factor determina la cantidad de SOL requerida para que una cuenta sea rent-exempt?:

- A. El número de transacciones realizadas por la cuenta.
- B. El tamaño de la cuenta en bytes
- C. La frecuencia de las interacciones con el programa.
- D. El saldo USDC en la cuenta durante el último mes.



¿Qué factor determina la cantidad de SOL requerida para que una cuenta sea rentable?

- A. El número de transacciones realizadas en la cuenta.
- B. El tamaño de la cuenta en bytes.
- C. La frecuencia de las interacciones con el programa.
- D. El saldo USDC en la cuenta durante el último mes.



TIPOS DE CUENTAS EN SOLANA

Cuentas Ejecutables

Almacenan programas
de usuarios

Almacenan programas
nativos

Cuentas No Ejecutables

Almacenan datos

PROGRAMAS EN SOLANA

- En el ecosistema de Solana, los "contratos inteligentes" se denominan programas.
- Un programa es una cuenta que contiene código ejecutable.
- No tienen estado; todos los datos que necesitan para operar se almacenan en cuentas separadas.

PROGRAMAS EN SOLANA

- Cuenta de un programa: La cuenta principal que representa un programa. Almacena la dirección de una cuenta de datos ejecutables y la autoridad de actualización del programa.
- Cuenta de datos ejecutables de un programa: Una cuenta que contiene el código ejecutable en bytes del programa.
- Cuenta de Buffer: Una cuenta temporal que almacena código en bytes mientras un programa está siendo desplegado o actualizado activamente.

ESTRUCTURA DE UN PROGRAMA EN SOLANA

Public Key:

"HBYM...H8c7"

Address (Program ID)

Program

AccountInfo

data

owner: BPF Loader

executable: true

lamports

Account Data

Program Code

TIPOS DE PROGRAMAS EN SOLANA

Programas On-Chain

- Programas desarrollados por los usuarios.
- Pueden ser actualizados únicamente por la cuenta autorizada.

Programas Nativos

- Programas integrados en el núcleo de Solana.
- Solo se pueden actualizar mediante actualizaciones de la red.

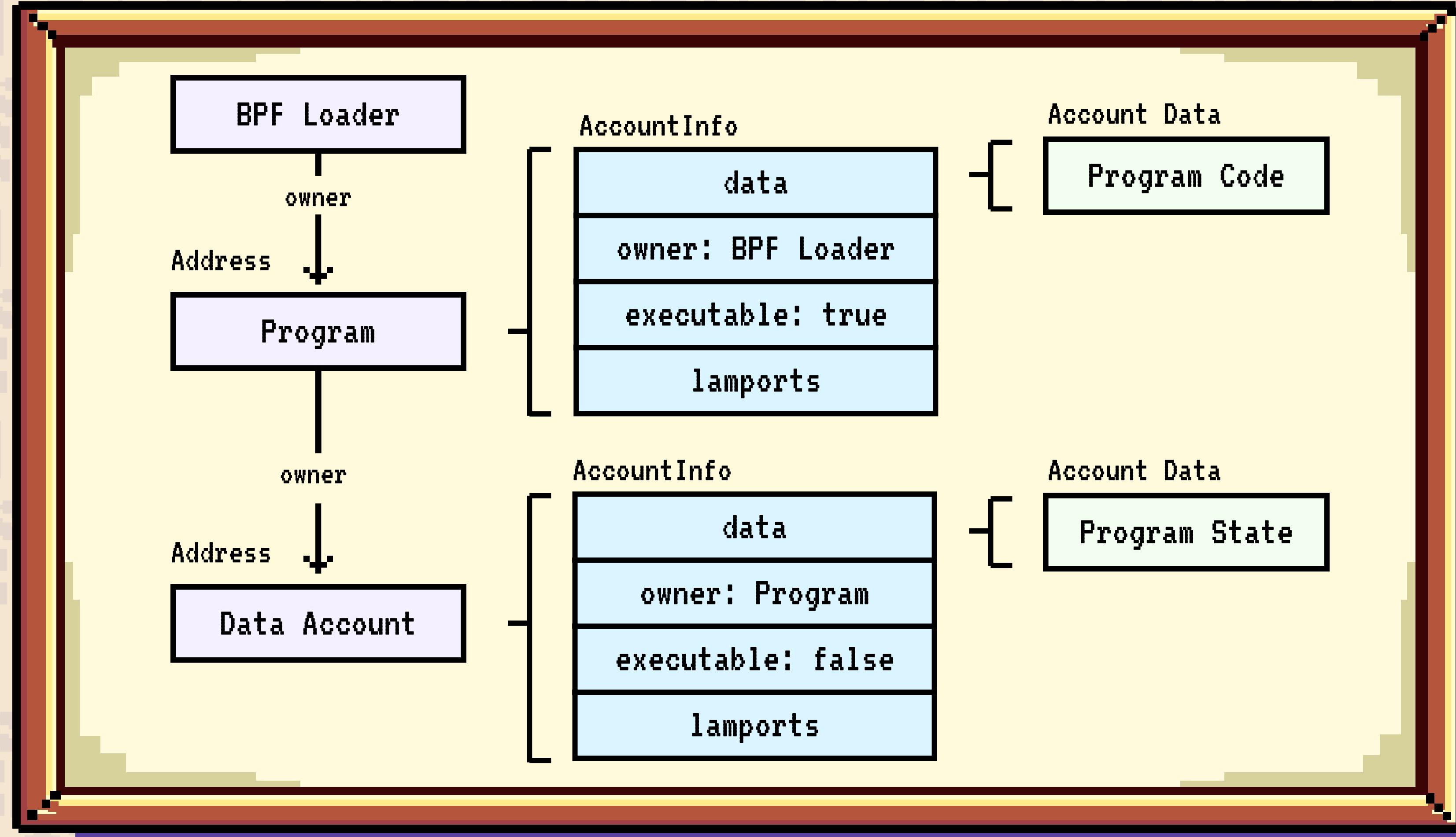
PROGRAMAS NATIVOS

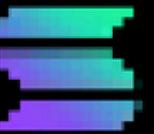
Al desarrollar programas en Solana, normalmente interactúa con dos programas nativos:

- System program: Todas las cuentas nuevas son propiedad del System program, este se encarga de crearlas y gestionarlas.
- BPF Loader: Todos los programas no nativos son propiedad del BPF Loader.

PROGRAMAS DE USUARIOS

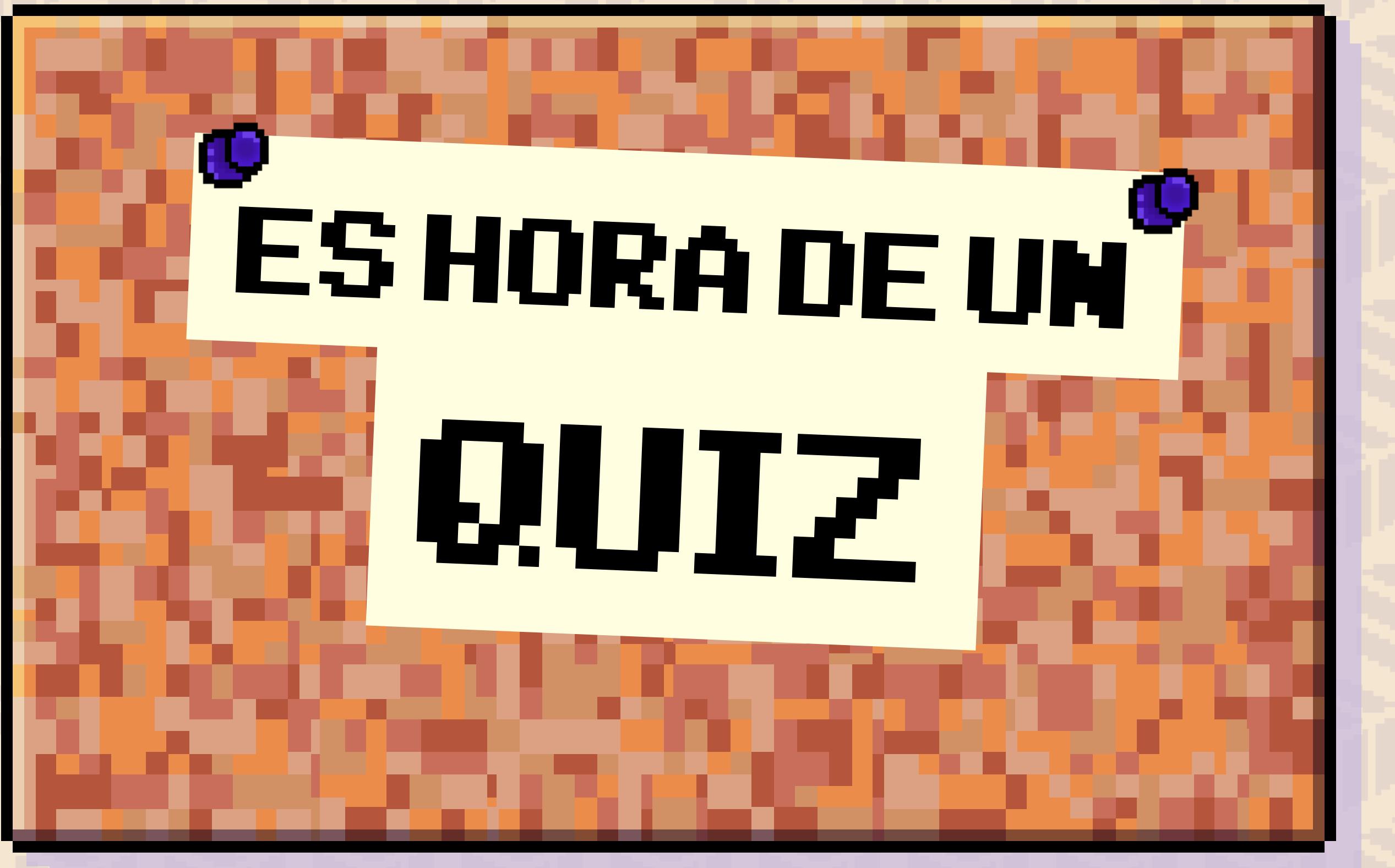
- Son creados y desplegados por desarrolladores externos o proyectos que construyen sobre Solana.
- Pueden realizar una amplia gama de funciones.
- Son fácilmente actualizables por los desarrolladores.

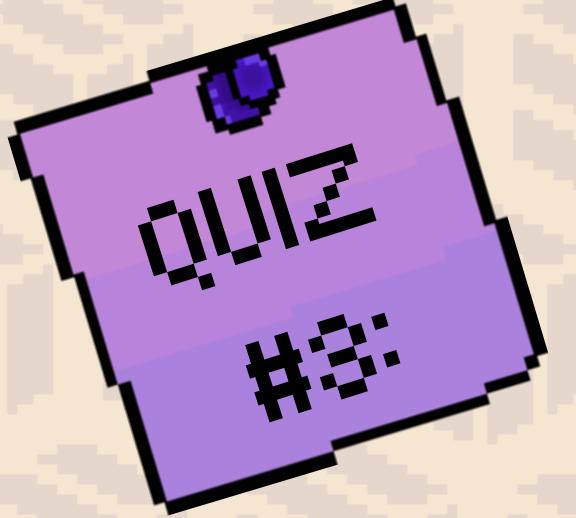




19

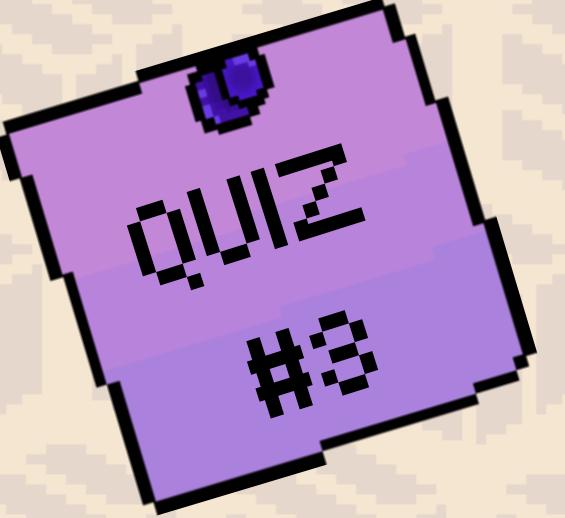
LUN	MAR	M
6	7	
13	14	
20	21	22
27	28	29





¿Qué programa nativo de Solana se utiliza para crear y gestionar cuentas?

- A. Token Program
- B. Stake Program
- C. Vote Program
- D. System Program



¿Qué programa nativo de Solana se utiliza para crear y gestionar cuentas?

- A. Token Program
- B. Stake Program
- C. Vote Program
- D. System Program

RESPUESTA
CORRECTA:
**D. SYSTEM
PROGRAM**



HEAVY DUTY CAMP V4

NOS VEMOS
EN LA
PROXIMA
CLASE!

CLASE
#1