

## WP4-A3. Especificaciones funcionales.



Esta obra está licenciada bajo una [Licencia Internacional Creative Commons  
Atribución-CompartirIgual 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

*"Financiado por la Unión Europea. Las opiniones y puntos de vista expresados solo comprometen a su(s) autor(es) y no reflejan necesariamente los de la Unión Europea o los de la Agencia Ejecutiva Europea de Educación y Cultura (EACEA). Ni la Unión Europea ni la EACEA pueden ser considerados responsables de ellos."*



## Índice

1. INTRODUCCIÓN .....	4
2. RESUMEN DEL SISTEMA Y FUNCIONES .....	5
2.1. Concepto de RockChain y escenarios principales de entrenamiento .....	5
2.2. Usuarios objetivo y roles .....	5
Aprendices (jugadores) .....	5
Formadores / facilitadores .....	6
Administradores técnicos.....	6
2.3. Entorno de despliegue y tiempo de ejecución .....	7
Dispositivos .....	7
Conectividad.....	7
Escala y estructura de la sesión.....	7
3. ARQUITECTURA FUNCIONAL Y MÓDULOS .....	8
3.1. Arquitectura cliente-servidor de alto nivel .....	8
3.2. Módulos frontales .....	9
3.3. Módulos de backend .....	11
4. REQUISITOS FUNCIONALES DETALLADOS .....	14
4.1. Gestión y autenticación de usuarios .....	14
4.2. Creación de juegos, unión y gestión de salas .....	14
4.3. Ciclo de vida del juego y de las rondas.....	15
4.4. Lógicas de mercado, recursos y economía circular .....	15
4.5. Desafíos de minería y validación de soluciones .....	16
4.6. Puntuación, recompensas y selección de sectores .....	17
4.7. Seguimiento, registro y datos necesarios para la evaluación de WP5.....	18
5. FLUJOS DE JUEGO Y TRANSICIONES DE ESTADO.....	19
5.1. Ciclo de vida general del juego .....	19
5.2. Ciclo de vida de la ronda .....	20
5.3. Flujos de jugadores .....	22
5.4. Reglas de sincronización entre clientes .....	23
6. REQUISITOS NO FUNCIONALES .....	26
6.1. Rendimiento y escalabilidad en entornos típicos de VET/ADU .....	26



---

6.2. Fiabilidad y tolerancia a fallos en condiciones de aula.....	26
6.3. Seguridad, privacidad y protección de datos .....	27
6.4. Usabilidad y accesibilidad para aprendices adultos con habilidades digitales heterogéneas .....	28
7. CONCLUSIONES Y PRÓXIMOS PASOS .....	29

## 1. INTRODUCCIÓN

Este documento presenta los resultados de la actividad WP4-A3: Especificaciones funcionales de la herramienta de e-learning RockChain. Basado en la base de datos y la arquitectura backend entregada en WP4-A1 y en la versión refinada y lista para piloto de la herramienta producida en WP4-A2, esta actividad se centra en describir de forma concisa y estructurada lo que se espera que haga RockChain desde un punto de vista funcional. El énfasis está en el comportamiento de la herramienta tal y como lo experimentan los aprendices, formadores y facilitadores durante sesiones de formación reales, más que en la implementación a nivel de código o en detalles técnicos de bajo nivel.

El objetivo principal de WP4-A3 es proporcionar una descripción clara de las funciones, módulos y flujos que conforman la Herramienta de e-Learning RockChain tras la fase de refinamiento. Esto incluye cómo se autentican y gestionan los usuarios, cómo se crean y orquestan las sesiones y rondas de partida, cómo se comportan las mecánicas de mercado y minería, y cómo se generan las puntuaciones, recompensas e indicadores básicos tanto para la jugabilidad como para la evaluación. Las especificaciones capturan cómo interactúan estos elementos en escenarios típicos de EFP y aprendizaje de adultos, con pequeños grupos de participantes, a menudo mayores de 45 años y con habilidades digitales heterogéneas, guiados por un formador que utiliza el juego como punto de partida para debatir sobre economía circular y gestión de residuos.

En términos prácticos, las especificaciones funcionales sirven como puente entre el diseño conceptual y pedagógico de RockChain y su implementación técnica. Traducen los requisitos identificados en los anteriores Paquetes de Trabajo en comportamientos y reglas observables que pueden usarse como referencia compartida para desarrolladores, formadores y responsables de la toma de decisiones: qué debe funcionar siempre, qué información debe estar disponible en pantalla y bajo qué condiciones el sistema debe avanzar de un estado a otro. El documento se refiere a la arquitectura existente basada en un cliente móvil React Native, servicios de Firebase y una capa de comunicación en tiempo real, pero mantiene el foco en los roles, responsabilidades y el comportamiento esperado del sistema. De este modo, WP4-A3 apoya la preparación de directrices para formadores en WP4-A4, el trabajo de empaquetado y despliegue en WP4-A5, y las actividades de evaluación en WP5, asegurando que la herramienta de e-Learning RockChain pueda mantenerse, reutilizarse y adaptarse más allá de la vida útil del proyecto de manera coherente y reproducible.

## 2. RESUMEN DEL SISTEMA Y FUNCIONES

RockChain es un juego serio multijugador diseñado para ayudar a los adultos a explorar cómo los datos, los mercados y los mecanismos de validación pueden apoyar estrategias de economía circular en los sectores de la roca ornamental y la construcción. En cada sesión, un pequeño grupo de estudiantes se une a un juego compartido en sus dispositivos móviles y juega una o más rondas en las que compran y transforman productos, generan y reducen residuos, resuelven desafíos mineros y observan cómo sus decisiones afectan tanto a los resultados económicos como medioambientales.

### 2.1. Concepto de RockChain y escenarios principales de entrenamiento

La herramienta está pensada principalmente para usarse en **sesiones dirigidas por un entrenador** más que como una aplicación independiente. Los escenarios típicos incluyen:

- Bloques interactivos cortos dentro de un curso o taller (por ejemplo, un segmento de 45–90 minutos donde la clase juega uno o dos juegos y luego discute los resultados).
- Sesiones de demostración en las que un formador utiliza RockChain para ilustrar conceptos como cadenas de valor, corrientes de residuos, trazabilidad e incentivos para prácticas circulares.
- Actividades mixtas en las que los alumnos reciben primero aportaciones teóricas (por ejemplo, en materiales de WP3 u otros módulos) y luego utilizan RockChain para experimentar con decisiones y observar el impacto en el desperdicio, los costes y las recompensas.

En todos los casos, RockChain no pretende sustituir la enseñanza, sino servir como un motor práctico para la exploración y la discusión, proporcionando una experiencia compartida que puede analizarse durante el informe informativo.

### 2.2. Usuarios objetivo y roles

Las especificaciones funcionales consideran tres tipos principales de usuarios:

#### Aprendices (jugadores)

- Participantes adultos en formación profesional y educación de adultos, muchos de ellos mayores de 45 años y con habilidades digitales heterogéneas.

- Usa RockChain en sus propios dispositivos Android o iOS para unirse a una partida, tomar decisiones durante las rondas y ver sus resultados.
- Interactúa solo con el cliente móvil: iniciar sesión, unirse a una partida mediante código, navegar entre mercado, minería, reciclaje, estadísticas y pantallas de perfil, y recibir comentarios y recompensas.

### Formadores / facilitadores

- Profesores, formadores o consultores responsables de impartir la actividad de aprendizaje.
- Utiliza RockChain para organizar y gestionar sesiones: crear o organizar juegos, ayudar a los estudiantes a unirse, decidir cuándo empezar y terminar las rondas y dirigir la discusión según los resultados.
- También puede revisar indicadores básicos (puntuaciones, niveles de desperdicio, industrias ganadoras, estrategias seleccionadas) para apoyar la sesión informativa y la evaluación.
- No necesitas habilidades técnicas profundas; Su interacción se realiza principalmente a través de la misma interfaz de cliente, posiblemente con una vista proyectada o una pantalla compartida.

### Administradores técnicos

- Personal de instituciones asociadas o centros técnicos que despliegan y mantienen el backend (proyecto Firebase, servidor en tiempo real, versiones de la app móvil).
- Asegúrate de que la autenticación, la base de datos, la configuración de red y las actualizaciones estén correctamente configuradas para las instituciones que utilizan RockChain.
- Normalmente, no participan en las sesiones de formación diarias, pero proporcionan soporte de segunda línea cuando se lanzan nuevas versiones o cuando se necesitan cambios en la infraestructura.

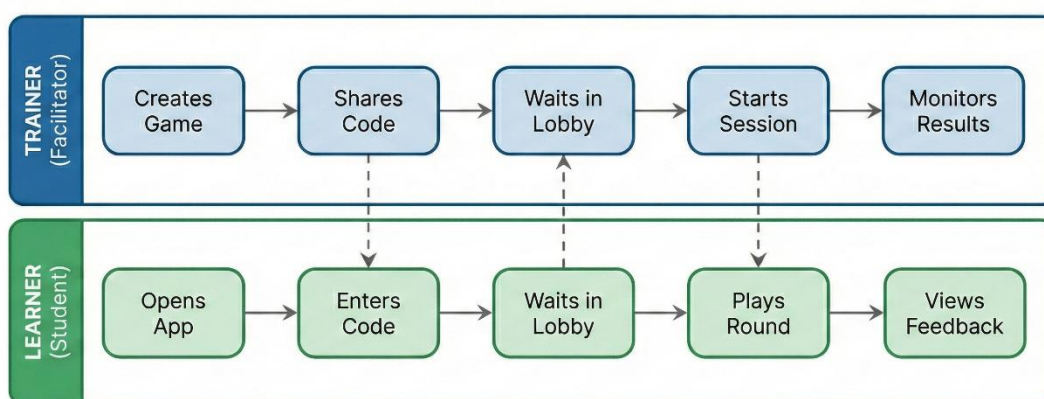


Figura 1: Sesión guiada por entrenador.

El comportamiento funcional especificado en este documento está orientado principalmente a estudiantes y formadores, mientras que los administradores técnicos actúan como facilitadores del entorno en el que se ejecuta la herramienta.

## 2.3. Entorno de despliegue y tiempo de ejecución

Funcionalmente, RockChain asume un patrón de despliegue común que refleje condiciones realistas de formación profesional y aprendizaje adulto:

### Dispositivos

- Los alumnos utilizan smartphones o tabletas que usan la aplicación móvil RockChain (Android o iOS).
- Los formadores también pueden usar un dispositivo móvil, una tableta o un emulador en un portátil, y opcionalmente proyectar su pantalla en la pantalla del aula para mostrar vistas compartidas (por ejemplo, sala de espera, resultados de final de ronda).

### Conectividad

- Se espera que las sesiones se desarrollen con acceso estable a internet: normalmente, Wi-Fi en el aula o redes institucionales, pero también es posible que se permitan los datos móviles.
- El sistema debe tolerar interrupciones breves (por ejemplo, fallos del Wi-Fi, suspensión del dispositivo) y permitir que los clientes se reconecten y reincorporen al juego actual sin perder su posición.
- No se soporta el uso totalmente offline, ya que RockChain depende de un backend alojado en la nube.

### Escala y estructura de la sesión

- Los juegos están diseñados para grupos pequeños (por ejemplo, de 3 a 10 alumnos por partida), con la posibilidad de disputar varias partidas en paralelo en una sola clase si es necesario.
- Los entrenadores planifican las sesiones para que haya tiempo suficiente para: iniciar sesión e incorporación, jugar uno o más juegos, y una fase de debriefing donde los resultados dentro del juego se vinculan al contenido del campo.

Estas suposiciones sobre dispositivos, conectividad y uso en el aula enmarcan los requisitos funcionales descritos en las siguientes secciones: determinan qué debe hacer el sistema para soportar sesiones fluidas dirigidas por formadores en entornos típicos de formación profesional y adultos.



### 3. ARQUITECTURA FUNCIONAL Y MÓDULOS

La herramienta de aprendizaje electrónico RockChain sigue una arquitectura cliente-servidor en la que una aplicación móvil proporciona la interfaz de usuario para aprendices y formadores, mientras que un conjunto de servicios backend gestiona la autenticación, lógica de juegos, persistencia y sincronización en tiempo real. WP4-A1 ya describió esta arquitectura en profundidad técnica; WP4-A3 reutiliza la misma estructura y se centra en las responsabilidades funcionales de cada parte: qué debe hacer el cliente, qué debe garantizar el backend y cómo ambos colaboran para ofrecer una experiencia de aprendizaje coherente.

#### 3.1. Arquitectura cliente-servidor de alto nivel

A un nivel general, RockChain consta de tres capas cooperantes:

- Un cliente móvil React Native (construido con Expo) que funciona en dispositivos Android o iOS. Este cliente es el único componente utilizado directamente por los estudiantes y formadores. Es responsable de la interacción con el usuario, la gestión local del estado durante una sesión y la presentación visual de los estados, retroalimentación y resultados del juego.
- Un backend de Firebase que proporciona autenticación, almacenamiento persistente y lógica de negocio controlada:
  - o La autenticación de Firebase identifica a los usuarios y emite las credenciales.
  - o Cloud Firestore almacena juegos, usuarios, instantáneas de mercado, eventos de minería y datos relacionados con el aprendizaje que deben sobrevivir a través de las sesiones.
  - o Firebase Functions implementa operaciones en el lado del servidor que requieren validación y acceso confiable (por ejemplo, crear juegos, unirse a juegos, actualizar el estado de las rondas, calcular recompensas o actualizar el progreso del usuario).
- Un servidor en tiempo real autorizado que coordina rondas e interacciones rápidas entre jugadores usando WebSockets. Este servidor mantiene una vista en memoria de cada partida activa (jugadores, inventarios, temporizadores, problemas de minería) y transmite eventos autorizados (inicio de ronda, final de ronda, actualizaciones de mercado, problemas de minería) a todos los clientes conectados.



Funcionalmente, el cliente móvil nunca intenta decidir por sí solo cuál debe ser el estado del juego. Reacciona a la información autorizada que proviene del backend y del servidor en tiempo real y envía acciones de usuario como peticiones (vía HTTPS o WebSockets) que el servidor valida y aplica antes de que se realice cualquier cambio persistente.

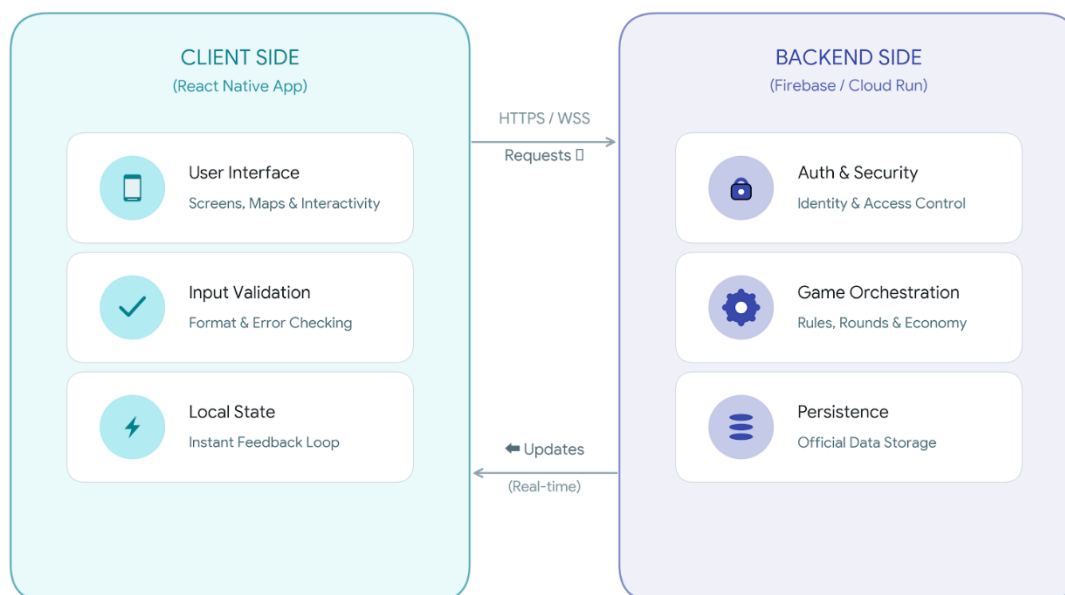


Figura 2: Diagrama de arquitectura

### 3.2. Módulos frontales

En el lado del cliente, el comportamiento se organiza en un conjunto de módulos y pantallas, todos ellos soportados por contextos de estado compartidos (para datos de juegos, minería y progreso de aprendizaje):

**Autenticación e incorporación (Inicio de sesión / Registro):** Gestiona el inicio de sesión y registro del usuario mediante la autenticación de Firebase, la creación básica del perfil y las comprobaciones iniciales (por ejemplo, si el usuario está alojando un juego o uniéndose a uno existente). Desde una perspectiva funcional, este módulo siempre debe llevar al usuario a crear un nuevo juego o a introducir un código válido.

**Sala de espera (Vestíbulo):** Tras crear o unirse a una partida, todos los jugadores son llevados a la sala de espera.

- Muestra quién se ha unido y su estado (esperando/listo)
- indica quién actúa como presentador o entrenador
- permite al anfitrión iniciar la partida cuando el grupo está completo

- muestra cualquier cuenta atrás o mensaje relacionado con el inicio de la primera ronda. Su función principal es mantener al grupo alineado antes de que empiece la partida

**Contenedor y cabecera compartida de GameTabs:** Una vez que el juego ha comenzado, el usuario accede al área principal del juego, estructurada como un conjunto de pestañas (Mercado, Minería, Reciclaje, Estadísticas, Perfil, Chat) envueltas en un encabezado compartido. El encabezado muestra los indicadores principales (número de rondas, tiempo restante, recursos clave) de forma consistente en todas las pestañas. Las especificaciones funcionales requieren que este encabezado refleje siempre el estado autoritario que viene del backend para que los estudiantes puedan confiar en él para sus decisiones.

**MarketScreen:** Muestra los productos disponibles en la ronda actual, sus precios, implicaciones en el desperdicio y otros atributos. Permite a los jugadores comprar o vender objetos, sujeto a las reglas impuestas por el backend. Funcionalmente, esta pantalla debe:

- siempre reflejan la configuración de mercado de la ronda actual
- prevenir operaciones inválidas (por ejemplo, gastar más RockCoins de los disponibles)
- Actualiza la vista local rápidamente cuando las transacciones tengan éxito

**MiningScreen:** Presenta problemas de minería (tareas simples de aritmética o lógica) vinculados a eventos específicos del juego y permite a los jugadores enviar respuestas bajo presión de tiempo. La pantalla debe:

- mostrar problemas activos y sus límites de tiempo
- Desactiva la envío cuando un problema deje de estar activo
- Indica claramente cuándo se ha determinado un ganador para un bloque

**Pantalla de reciclaje:** Permite a los jugadores transformar partes de su inventario o residuos en estados más valiosos o reducir los residuos acumulados según las reglas del juego. Su función funcional es hacer visibles y accionables las estrategias circulares, y alimentar los valores actualizados de residuos y recursos en el estado general del juego.

**StatsScreen:** Ofrece una visión general del rendimiento: puntuaciones, niveles de desperdicio, clasificaciones y otros indicadores por ronda o por partida. Es un módulo clave para el informe durante la sesión, ayudando tanto a los alumnos como a los formadores a interpretar lo ocurrido y qué estrategias fueron más efectivas.

**PerfilPantalla y ajustes:** Permite a los usuarios ver y ajustar su perfil (nombre, avatar, preferencias de idioma) y ver un resumen de sus actividades o logros pasados en el juego actual. También actúa como punto de entrada para opciones básicas de accesibilidad y localización.

**ChatScreen y componentes auxiliares:** Proporciona un canal de chat sencillo para ayudar a los jugadores a coordinarse durante el juego. Los componentes auxiliares (como un banner de estado de red o superposiciones de tutoriales) informan a los usuarios sobre problemas de conexión, proporcionan ayuda contextual y facilitan la incorporación más fluida.

En conjunto, estos módulos definen lo que los alumnos y formadores pueden ver y hacer en cada momento. Las especificaciones funcionales aseguran que cada pantalla reaccione de forma predecible a los cambios en el estado del backend y a las acciones del usuario.



Figura 3: Diagrama de módulos frontend

### 3.3. Módulos de backend

En el backend, la funcionalidad se agrupa en varios módulos lógicos, implementados entre Firebase Functions, las reglas Firestore y el servidor en tiempo real:

- Gestión de usuarios y perfiles: Se encarga del registro, autenticación y almacenamiento de información básica del usuario en Firestore. Funcionalmente, este módulo debe garantizar que:
  - Cada usuario autenticado tiene un identificador único y estable.

- Las actualizaciones de perfil son validadas y son consistentes.
- Y solo los usuarios autorizados pueden acceder o modificar sus propios datos.
- Orquestación de partidas y rondas: Cubre el ciclo de vida de los juegos y rondas: crear y eliminar partidas, unirse y salir, iniciar y terminar rondas, y determinar cuándo termina la partida. Este módulo define las transiciones de estado válidas y garantiza que:
  - Los códigos de juego son únicos y resolubles.
  - No más del número permitido de jugadores que se unen a una partida.
  - Los cambios de estado (espera, en curso, final de ronda, finalizado) se aplican de forma constante a todos los jugadores.
- Lógica de mercado y recursos: Gestiona la generación de configuraciones de mercado por ronda, las reglas de precios y desperdicios, y la validación de las transacciones. Desde una perspectiva funcional, este módulo asegura que:
  - El mercado de cada ronda cumple reglas predefinidas y restricciones de aleatoriedad.
  - Los inventarios y balances de los jugadores se actualizan de forma atómica y constante.
  - Los datos resultantes pueden rastrearse para su análisis (por ejemplo, mediante instantáneas o registros).
- Minería y lógica de validación: Genera problemas de minería, registra respuestas y determina ganadores según las reglas del juego. Este módulo debe:
  - Asegúrate de que los problemas estén asociados a un contexto específico del juego.
  - Registrar los intentos de una manera que pueda ser auditada.
  - Aplica criterios justos (por ejemplo, primera respuesta correcta) al asignar recompensas.
- Capa de coordinación en tiempo real: Implementa el servidor basado en WebSocket que mantiene el estado activo del juego en memoria y emite eventos autorizados. Sus responsabilidades funcionales incluyen:
  - Manteniendo a todos los jugadores conectados sincronizados respecto al estado de las rondas, temporizadores y actualizaciones clave.

- Gestionar desconexiones y reconexiones para que los jugadores puedan reincorporarse a las rondas en curso cuando sea posible.
- Interactuar con firestore y funciones en puntos bien definidos (por ejemplo, cuando persisten resultados de final de ronda).
- Progreso del aprendizaje y registro: Registra eventos relevantes y resúmenes necesarios para el seguimiento educativo y la posterior evaluación en WP5. Este módulo define qué acciones y resultados (por ejemplo, bloques minados, residuos reducidos, elecciones de industria) se almacenan como indicadores de aprendizaje y cómo pueden recuperarse y agregarse a nivel de usuario, juego o cohorte.

Al asignar claramente estas responsabilidades a módulos front-end y backend, la arquitectura funcional garantiza que RockChain se comporte de forma predecible, comprobable y extensible. La siguiente sección se basa en esta estructura para definir con más detalle los requisitos funcionales fundamentales.

## 4. REQUISITOS FUNCIONALES DETALLADOS

Los siguientes requisitos describen lo que debe hacer la Herramienta de Aprendizaje en Línea RockChain desde el punto de vista de los alumnos, formadores y el contexto formativo. Se basan en la arquitectura descrita anteriormente, pero se expresan en términos de comportamiento observable y reglas del sistema más que de detalles de implementación.

### 4.1. Gestión y autenticación de usuarios

El sistema debe identificar a los usuarios de forma consistente entre sesiones y mantener su información básica y su participación en el juego en una forma estable.

Funcionalmente, esto significa que el sistema deberá:

- Exigir a los usuarios autenticarse antes de acceder a cualquier funcionalidad relacionada con el juego.
- Crea y mantén un identificador de usuario único y estable y un perfil básico (por ejemplo, nombre de visualización, preferencia de idioma) para cada usuario autenticado.
- Permite a los usuarios actualizar campos de perfil seleccionados mientras conservan los enlaces a partidas anteriores y los datos de aprendizaje.
- Restringir el acceso para que cada usuario solo pueda leer o modificar su propio perfil y estado dentro del juego, salvo la información compartida explícitamente dentro del juego (por ejemplo, clasificaciones públicas).

### 4.2. Creación de juegos, unión y gestión de salas

El sistema debe permitir que entrenadores o anfitriones designados creen juegos y que otros usuarios se unan fácilmente, asegurando que todos los participantes estén correctamente registrados antes de que comience el juego.

El sistema deberá:

- Permite que un usuario cree un nuevo juego y asigne automáticamente un código y un identificador únicos.
- Asegúrate de que una partida recién creada se inicialice con parámetros predeterminados (conteo de rondas, temporizadores, número máximo de jugadores, recursos iniciales).

- Permite a los usuarios unirse a un juego existente introduciendo un código válido, con retroalimentación clara si el código es incorrecto o si el juego no es accesible.
- Impone límites en el número de jugadores por partida y evita las incorporaciones tardías una vez que la partida haya avanzado más allá de un punto acordado.
- Proporciona una vista de sala de espera (lobby) donde se vean todos los jugadores unidos y su estado de preparación, y donde el anfitrión pueda iniciar la partida cuando se cumplan las condiciones.

### 4.3. Ciclo de vida del juego y de las rondas

Una vez creado un juego, el sistema debe gestionar su ciclo de vida de forma controlada, moviéndose entre estados bien definidos que se reflejan de forma consistente en todos los clientes.

El sistema deberá:

- Mantén un estado explícito de la partida (por ejemplo, Esperando, en progreso, terminado) y un estado de ronda (por ejemplo, No iniciado, activo, resuelto, completado) para cada partida.
- Permitir solo transiciones válidas entre estos estados, activadas por acciones autorizadas (por ejemplo, inicio del juego por parte del anfitrión, expiración del temporizador de la ronda, final explícito del juego).
- Comienza cada ronda transmitiendo la información necesaria (número de ronda, tiempo restante, mercado inicial y configuración de minería) a todos los jugadores.
- Termina cada ronda según condiciones autorizadas (temporizador, finalización de los pasos requeridos) y asegúrate de que todos los clientes reciban una indicación clara de que el juego ha terminado.
- Marca el juego como terminado cuando se haya completado el número configurado de rondas o cuando el anfitrión termine explícitamente la sesión e evita que se hagan más acciones en esa partida después.

### 4.4. Lógicas de mercado, recursos y economía circular

La mecánica de mercado y recursos de RockChain debe reflejar el enfoque de aprendizaje previsto en el desperdicio, el valor y las estrategias circulares.



El sistema deberá:

- Generar, para cada ronda, un conjunto de productos disponibles con atributos asociados (tipo, material, calidad, precio, implicaciones de residuos) según reglas predefinidas.
- Presenta a cada jugador una visión de mercado que refleje el estado autoritario de esa ronda y actualízala cuando se compren o vendan productos.
- Valida todas las transacciones con los recursos actuales del jugador (por ejemplo, Rockcoins, inventario) y evita acciones que violen restricciones.
- Actualiza los inventarios, saldos y niveles de desperdicio de los jugadores de forma constante después de cada transacción válida.
- Asegúrese de que la configuración del mercado y los estados resultantes puedan reconstruirse posteriormente para su análisis (por ejemplo, mediante instantáneas almacenadas o datos derivados).

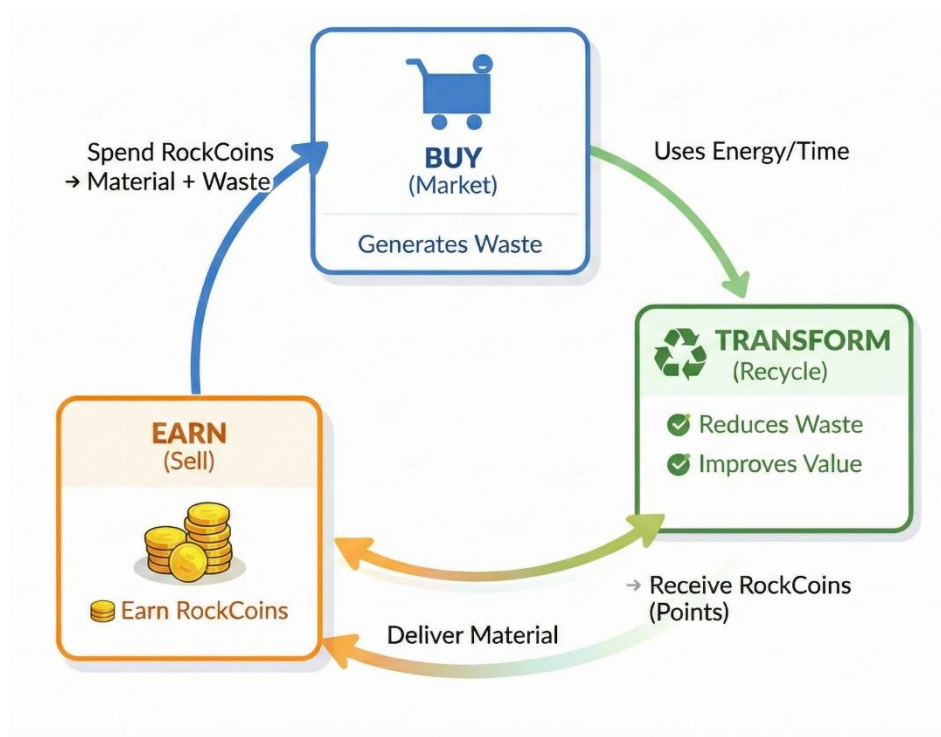


Figura 4: Economía circular en Rockchain.

#### 4.5. Desafíos de minería y validación de soluciones

Los eventos de minería ofrecen desafíos con límite de tiempo que conectan las decisiones y los mecanismos de validación con la narrativa inspirada en blockchain del juego.

El sistema deberá:

- Genera problemas de minería claramente asociados con un contexto de juego y ronda y hazlos disponibles solo mientras estén activos.
- Notifica a todos los jugadores relevantes cuando surja un nuevo problema de minería y indica cuánto tiempo permanecerá abierto.
- Acepta y graba las respuestas del reproductor mientras el problema está activo, rechazando envíos tardíos una vez cerrado o resuelto.
- Determinar los ganadores según reglas claramente definidas (por ejemplo, primera respuesta correcta) y asegurarse de que solo se registre un ganador por problema.
- Actualiza el estado del juego (por ejemplo, recompensas de minería, historial de problemas resueltos) e informa a todos los jugadores del resultado de forma transparente.

#### 4.6. Puntuación, recompensas y selección de sectores

La puntuación y las recompensas deben ayudar a los alumnos a comprender las consecuencias de sus decisiones tanto en los resultados económicos como en los indicadores de desperdicio/circularidad.

El sistema deberá:

- Mantener, para cada jugador, una puntuación actual y indicadores clave como el desperdicio acumulado y los recursos.
- Permite a los jugadores seleccionar o confirmar una industria o estrategia cuando sea relevante y utiliza esta información en el cálculo de recompensas al final de cada ronda.
- Calcular las recompensas por ronda basándose en reglas transparentes que combinan comportamiento de mercado, resultados de minería y elecciones de economía circular.
- Aplica estas recompensas de forma atómica a los estados de los jugadores (actualizando puntuaciones, niveles de desperdicio e inventarios) y almacena un resumen de los resultados de las rondas por jugador.
- Presenta resúmenes de final y final de partida que muestren claramente cómo se obtuvieron las puntuaciones e indicadores, apoyando el informe y la discusión.

## 4.7. Seguimiento, registro y datos necesarios para la evaluación de WP5

Por último, el sistema debe registrar suficiente información para respaldar la evaluación educativa en WP5 y análisis futuros, respetando los principios de protección de datos.

El sistema deberá:

- Registrar eventos y estados clave relevantes para el análisis de aprendizaje, como productos comprados y vendidos, residuos generados y reducidos, intentos y éxitos mineros, y elecciones de industria o estrategia.
- Asocia estos registros con identificadores de usuario y de juego seudónimos para que los patrones puedan analizarse a nivel de jugador, juego y cohorte.
- Proporcionar una forma de extraer o agregar esta información (por ejemplo, mediante colecciones dedicadas o instantáneas) para los socios responsables de la evaluación y la elaboración de informes.
- Registrar eventos técnicos (por ejemplo, desconexiones, reconexiones, errores) de forma que ayude a interpretar anomalías en la jugabilidad y favorezca la mejora continua.
- Respetar las políticas de minimización y retención de datos acordadas en el proyecto, asegurando que los registros y los datos de aprendizaje se almacenen solo el tiempo necesario para su evaluación e informes.

Estos requisitos funcionales fundamentales definen lo que RockChain debe hacer de forma fiable en cada despliegue, sirviendo como referencia compartida para los flujos más detallados y las restricciones no funcionales descritas en las siguientes secciones.

## 5. FLUJOS DE JUEGO Y TRANSICIONES DE ESTADO

El comportamiento funcional de RockChain se organiza en torno a un pequeño número de flujos recurrentes que se repiten en cada sesión. Estos flujos determinan cómo empieza una partida, cómo avanzan las rondas, qué ocurre cuando se acaba el tiempo y cómo reacciona el sistema si los jugadores se desconectan y vuelven a conectar. Son la columna vertebral que experimentan los formadores y aprendices durante el uso, y deben mantenerse predecibles a lo largo de los despliegues.

### 5.1. Ciclo de vida general del juego

Desde un punto de vista funcional, cada juego sigue un ciclo de vida común:

#### **Configuración del juego:**

- Se crea un nuevo juego al iniciar sesión, recibiendo un código de juego.
- El juego se inicializa con parámetros predeterminados (máximo de jugadores, número de rondas, duración de la ronda, recursos iniciales) y se marca como en espera.

#### **Entrada y fase de lobby:**

- Los estudiantes se autentican e introducen el código del juego para unirse a la misma sesión.
- El sistema registra a cada jugador en el juego, los muestra en la sala de espera y registra su estado de preparación.
- El anfitrión supervisa quién se ha unido y decide cuándo el grupo está completo.

#### **Juego en curso:**

- Cuando el presentador inicia el juego, el estado cambia a en curso y se inicializa la primera ronda.
- Los jugadores son llevados a la zona del juego (pestañas para mercado, minería, reciclaje, estadísticas, etc.) mientras un encabezado compartido muestra la ronda actual y el tiempo restante.
- El juego avanza a través de un número fijo de rondas, cada una siguiendo el ciclo de vida de las rondas descrito a continuación.

#### **Final del juego y informe:**

- Tras el número configurado de rondas, o si el presentador decide terminar la partida antes de tiempo, el sistema marca la partida como terminada.

- Una vista resumida final presenta puntuaciones, indicadores de desperdicio y resultados clave para todos los jugadores.
- Los formadores pueden utilizar este resumen, junto con los datos almacenados, como base para la discusión y la reflexión.

Una vez terminada una partida, no se permiten más acciones en esa sesión; Los alumnos pueden abandonar o unirse a una nueva partida si el entrenador decide dirigir otra.

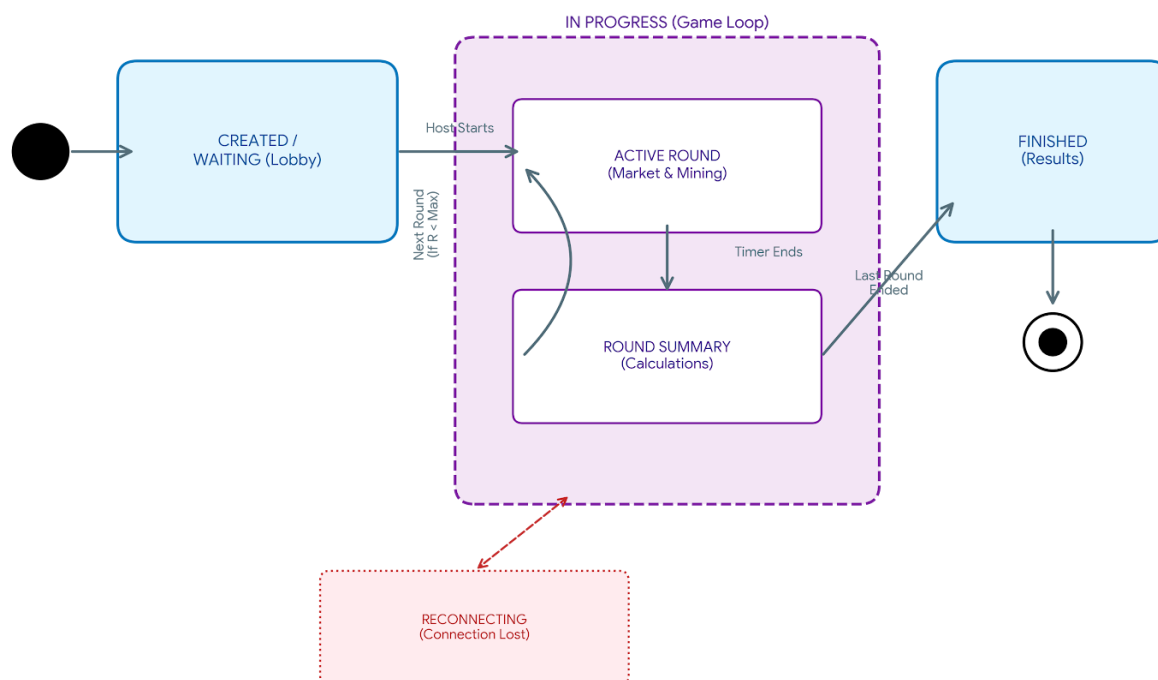


Figura 5: Diagrama de máquina de estados

## 5.2. Ciclo de vida de la ronda

Dentro del juego en general, cada ronda tiene su propio ciclo de vida interno. El sistema gestiona este ciclo de vida para que todos los jugadores experimenten las mismas fases al mismo tiempo:

### Inicialización de ronda:

- Al inicio de una ronda, el backend define el estado de autoridad: número de ronda, hora de finalización, configuración del mercado y cualquier contexto inicial de minería.
- Esta información se transmite a todos los clientes conectados, que actualizan sus opiniones en consecuencia. El encabezado inicia la cuenta atrás.

### Fase de juego active:

- Durante el juego activo, los jugadores pueden:
  - Interactúa con el mercado (comprando/vendiendo productos).
  - Resuelve problemas de minería cuando esté disponible.
  - Utiliza opciones de reciclaje.
  - Consulta estadísticas o habla por chat.
- Todas las acciones se envían como peticiones al backend; solo se aplican operaciones validadas y se reflejan a otros jugadores.

#### **Fase de resolución:**

- Cuando expira el temporizador autoritativo, o cuando el sistema decide que se han cumplido todas las condiciones necesarias, la ronda entra en estado de resolución.
- Las nuevas acciones se bloquean o limitan. El backend calcula:
  - Ganadores y recompensas mineras.
  - Bonos relacionados con la industria.
  - Cambios en puntuaciones, inventarios y niveles de residuos.
- Se escribe un conjunto consistente de resultados para almacenamiento persistente y se emite a todos los clientes.

#### **Retroalimentación al final de la ronda:**

- Los jugadores ven un resumen al final de la ronda: sus recursos actualizados, niveles de desperdicio, recompensas obtenidas y cualquier industria ganadora.
- Los formadores pueden usar esta visión para comentar estrategias y vincularlas a conceptos de economía circular antes de comenzar la siguiente ronda.

#### **Transición a la siguiente ronda:**

- Si quedan más rondas, el anfitrión o el sistema activa la siguiente ronda.
- El ciclo de inicialización, juego activo, resolución y retroalimentación se repite hasta que el juego termina.

Este ciclo de vida de la ronda garantiza que, en cada grupo, todos los alumnos experimenten un ritmo claro: *prepararse* → *actuar* → *ver las consecuencias* → reflexionar.

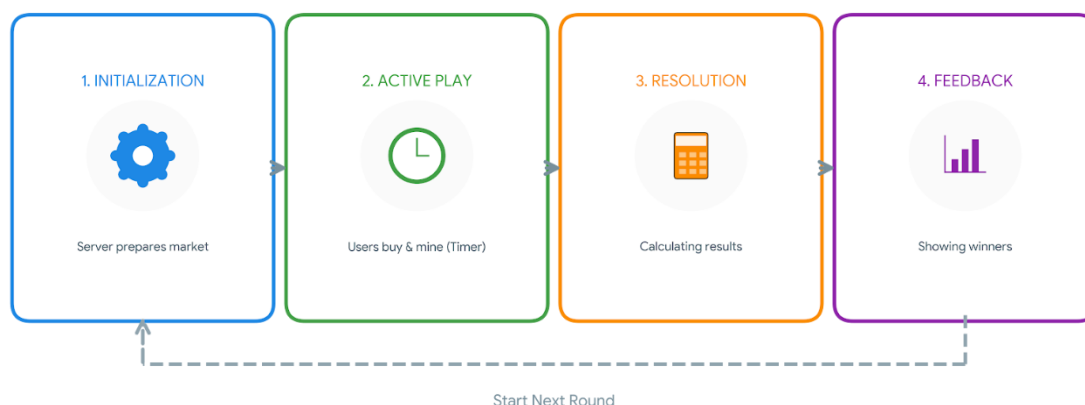


Figura 6: Ciclo de vida de la ronda.

### 5.3. Flujos de jugadores

En sesiones de entrenamiento reales, los participantes pueden llegar tarde, perder temporalmente la conexión o cerrar accidentalmente la aplicación. RockChain define reglas funcionales para manejar estas situaciones de forma fluida:

#### Unirse antes de empezar:

- Los jugadores pueden unirse libremente al juego mientras está en estado de espera.
- El sistema muestra los llegados tardíos en el vestíbulo y actualiza la lista para todos.
- Una vez que el presentador inicia la partida, las nuevas incorporaciones se impiden o se restringen según las reglas acordadas (por ejemplo, no se pueden unir después de que haya comenzado la primera ronda).

#### Reentrada durante una partida en curso:

- Si un jugador pierde temporalmente la conexión o cierra la aplicación:
  - Al reconectarse e iniciar sesión, el sistema utiliza su identidad para reasociarlos con la partida actual siempre que sea posible.
  - Se llevan de nuevo a la pantalla y ronda correspondientes, aplicándose el estado de autoridad actual.
- Si ya no es posible reconectarse (por ejemplo, si el juego ha terminado), el sistema informa al usuario y ofrece volver al menú principal.

#### Comportamiento durante las desconexiones:



- Mientras está desconectado, el usuario no puede realizar acciones que modifiquen el estado del juego.
- La interfaz puede indicar un estado limitado o de "esperando reconexión"; Una vez restablecida la conectividad, se obtiene el estado más reciente y se actualiza la vista.

#### **Salida voluntaria:**

- Un jugador que abandona el juego deliberadamente (por ejemplo, mediante una opción de salir) está marcado como ausente.
- Sus contribuciones anteriores permanecen en el historial del juego, pero ya no participan en acciones o recompensas posteriores a menos que vuelvan a unirse según las reglas definidas por el entrenador.

Estos flujos están diseñados para reducir las interrupciones en situaciones de aula donde son comunes problemas breves de conectividad o de dispositivos, manteniendo al mismo tiempo el estado del juego coherente y justo para todos los participantes.

## **5.4. Reglas de sincronización entre clientes**

Como RockChain es una herramienta multijugador en tiempo real, un aspecto central de su comportamiento funcional es mantener a todos los clientes alineados alrededor de una visión única y autoritaria del juego. Las principales reglas de sincronización son:

#### **Estado de backend autoritativo:**

- El backend, no los dispositivos individuales, decide:
  - ¿Qué partida y ronda están activas?
  - Cuando empiezan y terminan las rondas.
  - ¿Qué mercado y configuraciones mineras se aplican?
  - Cuáles son las puntuaciones e indicadores actuales.
- Los clientes escuchan a este estado y ajustan sus opiniones locales en consecuencia.

#### **Referencia de temporización compartida:**

- Las cuentas atrás de ronda y los desafíos con tiempo limitado utilizan un tiempo de referencia compartido proporcionado por el backend.

- Los clientes pueden mostrar el tiempo restante con pequeños ajustes locales, pero la decisión de terminar una ronda o cerrar un problema de minería siempre sigue el temporizador autoritativo, no el reloj del dispositivo.

#### Actualizaciones constantes:

- Cuando se realiza una acción válida (compra, venta, éxito de minería, operación de reciclaje), el backend:
  - Actualiza el estado del juego.
  - Persisten los cambios relevantes.
  - Retransmite el estado o evento actualizado a todos los jugadores.
- Los clientes actualizan sus interfaces solo tras recibir la confirmación, asegurándose de que todos vean los mismos resultados.

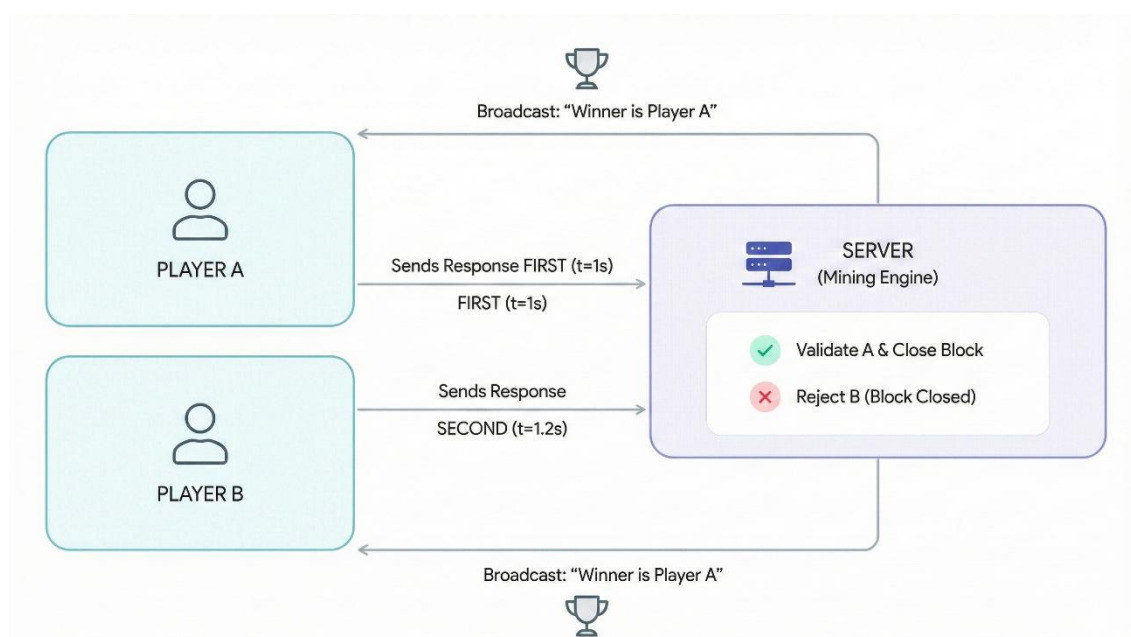


Figura 7: Respuesta de minería y conexión con el servidor.

#### Manejo elegante de acciones tardías o contradictorias:

- Las acciones que llegan después de que una ronda ha terminado o tras la resolución de un problema son rechazadas, con una respuesta clara para el usuario.
- Si dos acciones entraran en conflicto (por ejemplo, dos jugadores compitiendo por la misma recompensa de minería), el backend aplica reglas deterministas (como la "primera respuesta correcta") y notifica a todos los clientes del resultado final.



Estas reglas de sincronización aseguran que, incluso en presencia de variabilidad de la red, RockChain se comporte como un único juego compartido en lugar de una colección de simulaciones locales independientes. Para formadores y estudiantes, esto se traduce en una experiencia coherente donde las rondas comienzan y terminan juntas, los resultados se acuerdan por todos los dispositivos y la discusión tras cada ronda puede depender de una representación estable y fiable de lo ocurrido.

## 6. REQUISITOS NO FUNCIONALES

Más allá de lo que RockChain debe hacer funcionalmente, la herramienta también debe satisfacer un conjunto de requisitos no funcionales para poder usarse de forma fiable en entornos reales de formación profesional y de aprendizaje para adultos. Estos requisitos se refieren al rendimiento, la fiabilidad, la seguridad y la protección de datos, así como la usabilidad de la herramienta para aprendices adultos con habilidades digitales heterogéneas.

### 6.1. Rendimiento y escalabilidad en entornos típicos de VET/ADU

RockChain no está pensado para juegos masivos en línea, sino para grupos pequeños y medianos en aulas o talleres. En este contexto, el sistema deberá:

- Soporta al menos entre 2 y 5 jugadores simultáneos por partida, con la posibilidad de ejecutar varias partidas en paralelo dentro de una misma institución, sin una degradación notable en los tiempos de respuesta.
- Asegúrate de que las interacciones en tiempo real (inicio de rondas, actualizaciones de mercado, eventos de minería) se reflejen en los dispositivos clientes con un retraso corto y predecible (del orden de segundos, no decenas de segundos), bajo la conectividad normal del aula.
- Mantener un rendimiento aceptable cuando se jueguen varias rondas en secuencia, sin necesidad de reinicios ni intervenciones manuales entre partidas.
- Evita un consumo excesivo de recursos del dispositivo (CPU, memoria, batería) para que la app funcione sin problemas en smartphones y tabletas de gama media típicas usadas por adultos que aprenden.

Estas expectativas de rendimiento reflejan un uso realista en los cursos de VET/ADU y aseguran que las limitaciones técnicas no eclipsen la experiencia de aprendizaje.

### 6.2. Fiabilidad y tolerancia a fallos en condiciones de aula

En sesiones reales, son comunes fallos temporales de red, suspensión de dispositivos o desconexiones breves. RockChain debe ser lo suficientemente resistente para manejar estas situaciones sin descarrilar la actividad. El sistema deberá:

- Tolera interrupciones cortas de conectividad permitiendo que los clientes se reconecten y resincronizen con la partida y la ronda actuales, siempre que sea técnicamente posible.

- Evitar la corrupción de datos o estados inconsistentes cuando se producen desconexiones durante operaciones críticas (por ejemplo, cálculos al final de la ronda, asignación de recompensas).
- Evitar fallos o errores irreversibles en el cliente móvil en uso normal, proporcionando mensajes claros si ocurre un error crítico.
- Asegúrate de que los entrenadores puedan completar una sesión planificada (una o más partidas) sin tener que reiniciar la infraestructura, incluso si los jugadores experimentan problemas técnicos pasajeros.

El objetivo es que las preocupaciones de fiabilidad no se conviertan en una barrera para los formadores, que pueden disponer de poco tiempo y apoyo técnico durante sus clases.

### 6.3. Seguridad, privacidad y protección de datos

Los requisitos de seguridad, privacidad y protección de datos complementan las salvaguardas técnicas ya definidas para el backend y la base de datos en WP4. A1. Desde un punto de vista funcional, el sistema deberá:

- Asegúrate de que solo los usuarios autenticados puedan acceder a las sesiones del juego y que los códigos de juego no sean adivinables en la práctica.
- Restringir el acceso a datos personales y relacionados con el juego para que cada usuario solo pueda ver:
  - o Su propio perfil y estado dentro del juego.
  - o Información que se comparte explícitamente dentro de un partido (por ejemplo, puntuaciones y clasificaciones visibles para todos los jugadores en ese encuentro).
- Almacenar y procesar solo los datos personales mínimos necesarios para ejecutar sesiones y evaluar los resultados de aprendizaje, utilizando identificadores seudónimos siempre que sea posible.
- Apoyar las obligaciones del proyecto bajo la normativa de protección de datos (por ejemplo, RGPD), incluyendo:
  - o La posibilidad de atender solicitudes de acceso, corrección o eliminación de datos personales.
  - o La aplicación de periodos de retención acordados para registros y datos de aprendizaje.

- Manejo seguro de credenciales y secretos de configuración.
- Proteger la comunicación entre cliente y backend (por ejemplo, usando canales cifrados) y aplicar salvaguardas adecuadas para evitar accesos o manipulaciones no autorizadas de estados del juego.

Estos requisitos garantizan que RockChain pueda implementarse en instituciones educativas sin comprometer la privacidad de los alumnos ni las políticas institucionales.

## 6.4. Usabilidad y accesibilidad para aprendices adultos con habilidades digitales heterogéneas

Por último, la herramienta debe ser utilizable por estudiantes adultos, muchos de ellos mayores de 45 años, que pueden no estar familiarizados con juegos móviles o interfaces complejas. El sistema deberá:

- Presenta una estructura de navegación sencilla y consistente, con etiquetas claras y pasos mínimos entre iniciar sesión, unirse a una partida y empezar a jugar.
- Utiliza lenguaje y elementos visuales fáciles de entender, evitando jerga técnica innecesaria y proporcionando explicaciones breves cuando sea necesario (por ejemplo, para temporizadores, recompensas o elecciones del sector).
- Mantén la información clave (tiempo restante, número redondo, recursos principales) visible en una ubicación estable en pantalla, para que los alumnos no tengan que buscarla al tomar decisiones.
- Apoya el uso multilingüe permitiendo que formadores y alumnos elijan su idioma preferido entre los proporcionados en el proyecto, y asegurando que los mensajes y comentarios esenciales estén disponibles en esos idiomas.
- Comportate de forma predecible cuando los alumnos cometen errores o tardan más en actuar, proporcionando una retroalimentación clara y evitando cambios repentinos que puedan causar confusión o ansiedad.

Al cumplir con estos requisitos de usabilidad y accesibilidad, RockChain puede integrarse en cursos con públicos adultos diversos, reduciendo la necesidad de asistencia técnica constante y permitiendo que los formadores se centren en la facilitación y la discusión en lugar de en la resolución de problemas de la herramienta.

## 7. CONCLUSIONES Y PRÓXIMOS PASOS

WP4-A3 ha consolidado la herramienta de e-Learning RockChain en un sistema funcional claramente especificado, construido directamente sobre la capa de datos y la arquitectura backend entregada en WP4-A1 y los refinamientos realizados en WP4-A2. Mientras que las actividades anteriores se centraban en hacer que RockChain fuera técnicamente robusto y pedagógicamente utilizable, estos entregables capturan lo que se espera que haga la herramienta en la práctica: cómo se autentican y organizan los usuarios en los juegos, cómo se desarrollan las rondas, cómo se comportan los mercados y los desafíos de minería, cómo se producen las puntuaciones e indicadores, y qué eventos deben registrarse para apoyar la evaluación. Al hacerlo, traduce los objetivos de aprendizaje del proyecto en un conjunto de comportamientos y reglas observables que pueden implementarse, probarse y reutilizarse de forma consistente.

Estas especificaciones funcionales sirven ahora como punto de referencia compartido tanto para socios técnicos como pedagógicos. Para WP4-A4, proporcionan la base para las directrices y materiales de facilitación para entrenadores: descripciones paso a paso de los flujos, estados y pantallas del juego que los entrenadores verán al preparar y dirigir las sesiones. Para WP4-A5, reducen la incertidumbre en el empaquetado final y el trabajo de despliegue, ya que el comportamiento esperado de la herramienta es estable y está documentado, permitiendo a los socios centrarse en completar activos, traducciones y configuraciones de compilación sin cuestionar la mecánica central. Las mismas especificaciones también aclaran qué datos e indicadores están disponibles para su análisis, informando directamente cómo el WP5 puede planificar sus instrumentos de evaluación y estrategias de recogida de datos.

Mirando hacia adelante, el comportamiento definido en el WP4-A3 está pensado para ser lo suficientemente robusto para uso en piloto y lo suficientemente flexible para futuras adaptaciones. Los pilotos en WP5 probarán estas especificaciones en condiciones reales en diferentes países, instituciones y perfiles de aprendices, aportando evidencia sobre hasta qué punto el diseño funcional apoya el aprendizaje sobre economía circular y residuos en el sector de la roca ornamental. La retroalimentación de esos pilotos puede conducir a ajustes incrementales, pero se espera que los flujos y requisitos centrales descritos aquí sigan siendo el marco de referencia. De este modo, WP4-A3 marca una transición desde el diseño y perfeccionamiento de RockChain hasta su despliegue y uso sistemático, asegurando que la herramienta pueda mantenerse, replicarse y, si es necesario, extenderse más allá de la vida útil del proyecto.