UNIVERSITAT DE BARCELONA

Trabajo de Fin de Grado

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica

Universitat de Barcelona

Implementación de un sistema de encriptación y autenticación basado en blockchain en el entorno RV-FPGA

Autor: Pablo Martinez Abete

Director: Manuel López de Miguel

Realizado en: Departamento de Matemáticas e Informática

Barcelona, 10 de junio de 2025

1. OBJETIVOS....4
1.1 OBJETIVO GENERAL....4
1.2 OBJETIVOS ESPECÍFICOS....6
2. ESTADO DEL ARTE....8
2.1 BLOCKCHAIN....8
2.2 SHA & AES....9
2.2.1 SHA (Secure Hash Algorithm)....9
2.2.2 AES (Advanced Encryption Standard)....10
2.3 NEXYS A7 (FPGA)....12
2.3.1 Comparativa con otras plataformas FPGA....13
2.3.3 Implementaciones existentes en Nexys A7....13
3. INGENIERÍA DE CONCEPCIÓN....15
3.1 Análisis de requisitos....15
3.2 Decisiones de diseño principales....16
3.2.1 Elección de Visual Studio Code como entorno de desarrollo....16
3.2.2 Elección de Nexys A7 como plataforma de desarrollo....17
3.2.3 Elección de SHA y AES como algoritmos criptográficos....17
3.2.4 Implementación de la lógica del programa en alto y bajo nivel....17
3.2.5 Elección del puente UART-USB para la comunicación....18
3.3 Arquitectura del sistema....18
3.4 Diseño específico....19
3.5 Exploración de alternativas....20
4. INGENIERÍA DE DETALLE....22
4.1 Implementación de algoritmos criptográficos....22
4.1.1 Implementación de SHA....22
4.1.2 Implementación de AES....24
4.1.3 Validación....26
4.2 Desarrollo del sistema blockchain a alto nivel....26
4.2.1 Implementación del blockchain....27
4.2.2 Implementación de la herramienta de verificación de cadena....28
4.2.3 Validación de la lógica blockchain....29
4.3 Migración a bajo nivel....30
4.3.1 Implementación del blockchain....30
4.3.2 Implementación de la interfaz de usuario....31
4.3.3 Validación....32
4.4 Implementación de comunicación entre nodos....32
4.4.1 Desarrollo de la comunicación entre nodos e integración de capas de diferente nivel....32
5. Cronograma....34

5.1 Fases principales.....34
5.1.1 Fase de investigación.....35
5.1.2 Fase de desarrollo de algoritmos criptográficos.....36
5.1.3 Fase de desarrollo del sistema blockchain a alto nivel.....36
5.1.4 Fase de migración a bajo nivel.....37
5.1.5 Fase de pruebas, validación y documentación.....38
5.2 Diagrama de Gantt.....39
6. COSTE ECONÓMICO.....40
6.1 Consideraciones adicionales.....41
7. CONCLUSIONES.....42
7.1 Ampliaciones y futuras mejoras.....44
8. REFERENCIAS.....47
9. ANEXO.....50
9.1 Nexys A7.....50

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Blockchain es una tecnología de registro distribuido que permite almacenar información de forma segura, inmutable y verificable sin necesidad de una autoridad central. Estas propiedades la convierten en una herramienta particularmente eficaz para la certificación de datos, procesos y activos digitales.

En un sistema de certificación basado en blockchain, cada transacción o dato relevante (como un certificado académico, un resultado de laboratorio, una imagen capturada por un sensor o una evidencia de cumplimiento) se registra como un bloque dentro de una cadena criptográficamente enlazada. Una vez validado por la red, este bloque no puede ser alterado sin comprometer la integridad de toda la cadena, lo que garantiza la autenticidad, la trazabilidad y la resistencia a la manipulación.

Además, gracias al uso de hashes criptográficos, firmas digitales y mecanismos de consenso, el blockchain permite comprobar la validez de la información en cualquier momento y desde cualquier parte del mundo, sin revelar necesariamente el contenido original, lo que también protege la privacidad.

Por estas razones, blockchain se está adoptando como infraestructura de confianza en sectores tan diversos como la educación, la cadena de suministro, la sanidad, el registro de propiedad y los sistemas de certificación en entornos IoT o industriales, donde se requiere una alta fiabilidad de los datos y una validación transparente entre múltiples partes.

El objetivo principal de este trabajo de fin de grado es realizar y evaluar la implementación de un sistema básico de blockchain en un entorno dedicado con recursos limitados. Este estudio permitirá cuantificar la viabilidad de la utilización de este tipo de dispositivos en ámbitos académicos y profesionales.

A su vez cuestionará su posible uso de manera extendida, planteando así la posibilidad de ampliar el uso y la adopción de sistemas blockchain para aumentar la seguridad y mantener la integridad de los datos de diferentes sectores populares, incluso siendo éstos una versión básica o simplificada de las ya conocidas implementaciones complejas utilizadas, como por ejemplo la utilizada en el mundo de las Criptomonedas.

El sistema blockchain a implementar y estudiar estará fundamentado en un programa que asegure la integridad de los datos contenidos mediante el uso de los algoritmos criptográficos SHA (Secure Hash Algorithm) y AES (Advanced Encryption Standard). El algoritmo AES permitirá el

cifrado de los datos de manera reversible para garantizar la seguridad de los datos, mientras que el algoritmo SHA se utilizará para verificar la integridad de los bloques de datos y protegerlos de modificaciones y vulneraciones.

La implementación del sistema mencionado se realizará en una arquitectura integrada en una placa de hardware Nexys A7, un dispositivo de hardware basado en FPGA (Field- Programmable Gate Array). Este dispositivo se está utilizando actualmente en diversas asignaturas de los grados en Ingeniería electrónica y de Ingeniería informática, por lo que el presente proyecto tiene también un carácter formativo, ya que estas implementaciones pueden aplicarse posteriormente a diversas asignaturas de estos grados. Además la realización de este proyecto puede resultar en una base consolidada para futuras aplicaciones y/o desarrollos.

1.2 OBJETIVOS ESPECÍFICOS

Para alcanzar el objetivo general planteado, se han definido los siguientes objetivos específicos, los cuales en su conjunto conformarán un sistema que cumplimente los requerimientos básicos planteados:

- **Diseñar e implementar el algoritmo SHA en lenguaje C**:
  Desarrollar una implementación eficiente del algoritmo de hash seguro SHA-256 optimizada para ejecutarse en el microprocesador de la placa FPGA, permitiendo el cálculo de hashes necesarios para la verificación de la integridad de los bloques.

- **Implementar el algoritmo AES para cifrado y descifrado de datos:**
  Desarrollar una implementación del algoritmo AES que permita cifrar información sensible antes de su almacenamiento en la cadena de bloques y descifrarla cuando sea necesario acceder a ella.

- **Implementar un sistema de almacenamiento estructurado en bloques encadenados:**
  Crear una estructura de datos adecuada para representar los bloques y sus relaciones, guardando su contenido en la memoria de la FPGA, definiendo claramente el formato y contenido de cada bloque.

- **Crear funciones de verificación de integridad:**
  Desarrollar algoritmos que permitan comprobar la integridad de la cadena completa, verificando que cada bloque mantiene la relación correcta con sus bloques adyacentes mediante la validación de los hashes almacenados.

- **Desarrollar la comunicación entre diferentes nodos de la red blockchain:**
  Idear y desarrollar la comunicación entre diferentes nodos con la lógica blockchain integrada, los cuales permitan reproducir el comportamiento de una auténtica red blockchain con nodos interoperativos.

- **Desarrollar una interfaz de usuario mínima:**
  Implementar una interfaz de usuario mínima que permita la interacción con el sistema blockchain objetivo y que muestre información de los procesos en curso y el estado de los mismos en tiempo de ejecución

La consecución de estos objetivos específicos en su totalidad permitirá demostrar la capacidad y eficacia de las FPGAs para ejecutar algoritmos criptográficos adaptados, además de demostrar la capacidad de integrar sistemas blockchain básicos, abriendo posibilidades para aplicaciones en entornos donde la integridad y seguridad de la información contenida sea crucial. Esta integración puede ser una opción viable en entornos donde los recursos computacionales son limitados o donde se requiere un alto nivel de seguridad y un control total sobre los recursos disponibles.

2. ESTADO DEL ARTE

2.1 BLOCKCHAIN

Blockchain es una tecnología de registro distribuido que permite almacenar información de manera segura, transparente e inmutable mediante una estructura de datos encadenada criptográficamente.

Un blockchain consiste en una cadena de bloques enlazados criptográficamente, donde cada bloque contiene un conjunto de transacciones o datos, un timestamp, y una referencia hash al bloque anterior. Esta estructura crea una cadena inmutable donde cualquier modificación en un bloque histórico requeriría recalcular y modificar todos los bloques posteriores.

Dada su arquitectura diseñada con la descentralización como objetivo, no existe una autoridad central que controle la red; en su lugar, múltiples nodos participan en la validación y verificación de las transacciones, lo que aumenta la resistencia a fraudes y ataques. Además, la inmutabilidad de la información registrada garantiza que una vez que un bloque es añadido a la cadena, no puede ser modificado ni eliminado sin el consenso de la mayoría de los participantes de la red.

Actualmente, la tecnología blockchain se ha implementado en diversos sectores más allá de las criptomonedas, que fue su primera aplicación popularizada con Bitcoin.

El ámbito financiero ha sido otro de los sectores pioneros en adoptar esta tecnología, utilizándose para sistemas de pagos internacionales, contratos inteligentes y tokenización de activos.

En la cadena de suministro, blockchain permite la trazabilidad de productos desde su origen hasta el consumidor final, proporcionando transparencia y confianza en todo el proceso.

El sector salud también se beneficia de esta tecnología para la gestión segura de historiales médicos y la verificación de medicamentos, mientras que en el ámbito gubernamental se explora su uso para sistemas de votación electrónica y gestión de identidades digitales.

La protección de la propiedad intelectual mediante el registro y protección de derechos de autor es otra aplicación relevante que demuestra la versatilidad de esta tecnología.

La mayoría de las implementaciones blockchain actuales se basan en software ejecutándose en servidores o dispositivos de propósito general. Sin embargo, existen significativas diferencias entre las implementaciones basadas en software y las basadas en hardware. Las primeras ofrecen una alta flexibilidad, siendo fáciles de actualizar y modificar, pero siendo a su vez vulnerables a ataques de software.

Por otro lado, las implementaciones en hardware, aunque más limitadas en términos de flexibilidad y con un coste de desarrollo generalmente más alto, proporcionan mayor resistencia a

ataques de software y potencialmente un mejor rendimiento para operaciones específicas, además de ser más eficientes energéticamente.

En el mercado actual existen algunas implementaciones de sistemas blockchain en dispositivos de bajo nivel, aunque son significativamente menos comunes que las implementaciones en software.

Dispositivos como Ledger y Trezor[1], aunque principalmente son carteras de hardware para criptomonedas, utilizan principios de blockchain en dispositivos dedicados.

Sin embargo, la implementación de un sistema blockchain completo directamente en una FPGA, como se propone en este proyecto, representa un enfoque innovador que difiere de las soluciones existentes en varios aspectos. Al implementar el sistema directamente en la FPGA, se tiene control total sobre los recursos de hardware utilizados, lo que permite optimizaciones específicas para las operaciones criptográficas requeridas. Además, al prescindir de sistemas operativos y capas de software intermedias, se reduce la superficie potencial de ataque, aumentando la seguridad del sistema. Esta aproximación también puede resultar más eficiente energéticamente que las implementaciones en software, un factor crítico en entornos con restricciones energéticas como lo son los sistemas embebidos.

2.2 SHA & AES

2.2.1 SHA (Secure Hash Algorithm)

SHA (Secure Hash Algorithm) es una familia de funciones hash criptográficas desarrolladas por la Agencia de Seguridad Nacional de Estados Unidos (NSA) y publicadas por el Instituto Nacional de Estándares y Tecnología (NIST). Estas funciones generan un valor hash de longitud fija a partir de datos de entrada de longitud variable, siendo fundamentales en aplicaciones de seguridad informática.

Las funciones SHA se caracterizan por ser unidireccionales, lo que significa que es computacionalmente inviable obtener los datos originales a partir del hash generado. Son resistentes a colisiones, es decir, es extremadamente difícil encontrar dos entradas diferentes que produzcan el mismo hash. Además, presentan el efecto avalancha, donde un pequeño cambio en la entrada produce un cambio drástico en la salida.

SHA-256 es una variante específica que produce un hash de 256 bits (32 bytes). El algoritmo opera procesando los datos en bloques de 512 bits y utiliza 64 rondas de operaciones matemáticas complejas. El algoritmo utiliza ocho valores de inicialización derivados de las raíces cuadradas de los primeros ocho números primos. Durante cada ronda, se aplican funciones no lineales que incluyen operaciones de rotación, XOR, AND, OR y adiciones modulares. Estas operaciones garantizan que el resultado final sea altamente sensible a cualquier cambio en la entrada.

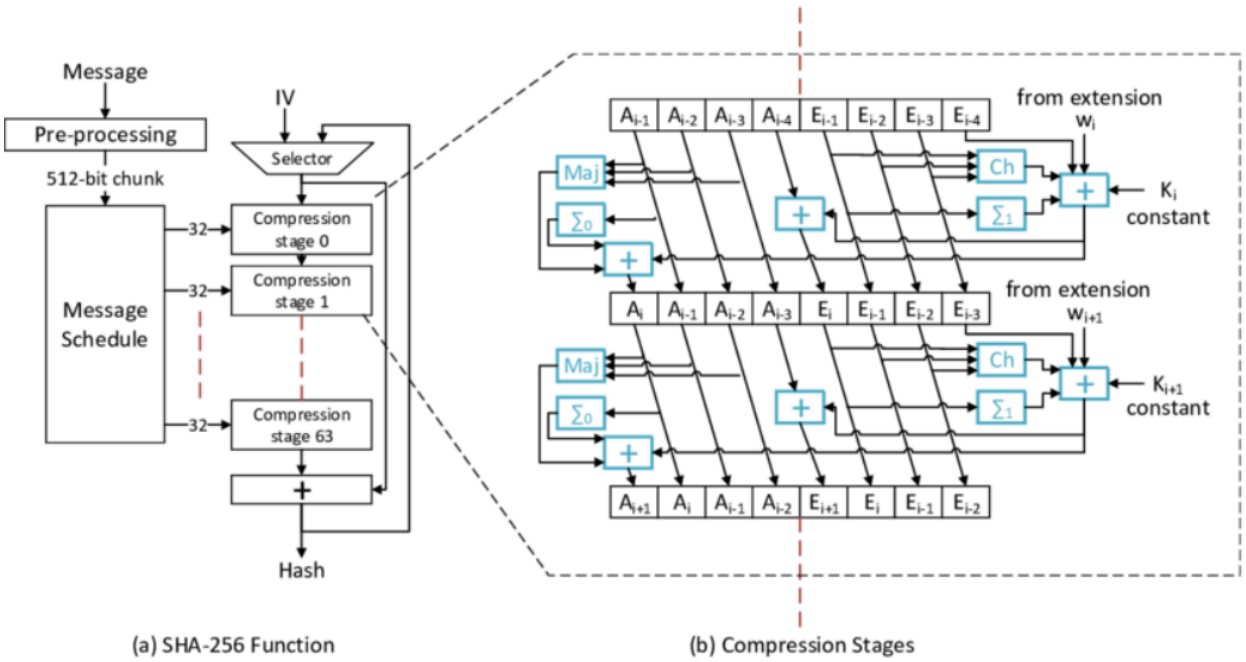


Figura 1. Diagrama de bloque de algoritmo SHA-256. (a) Flujo de ejecución del algoritmo (b) Diagrama detallado de las fases de compresión.

SHA-256 es ampliamente utilizado en aplicaciones críticas de seguridad, incluyendo sistemas blockchain como Bitcoin, certificados digitales, firmas electrónicas y verificación de integridad de archivos. Su resistencia criptográfica lo hace adecuado para aplicaciones que requieren garantías de seguridad a largo plazo, y su efecto avalancha resulta útil en aplicaciones que trabajan con volúmenes de datos extensos, donde debido a la gran cantidad de datos procesados, una pequeña modificación de estos puede ser imperceptible y pasar desapercibida si no se implementan mecanismos de seguridad como este.

Para este proyecto, se ha seleccionado específicamente SHA-256, que produce un hash de 256 bits (32 bytes) y es ampliamente utilizado en aplicaciones blockchain y de diferente índole por su equilibrio entre seguridad y rendimiento.

2.2.2 AES (Advanced Encryption Standard)

AES (Advanced Encryption Standard) es un algoritmo de cifrado simétrico adoptado como estándar por el gobierno de Estados Unidos en 2001, tras un riguroso proceso de selección que duró varios años. Reemplazó al algoritmo DES (Data Encryption Standard) y se ha convertido en el algoritmo de cifrado más utilizado mundialmente.

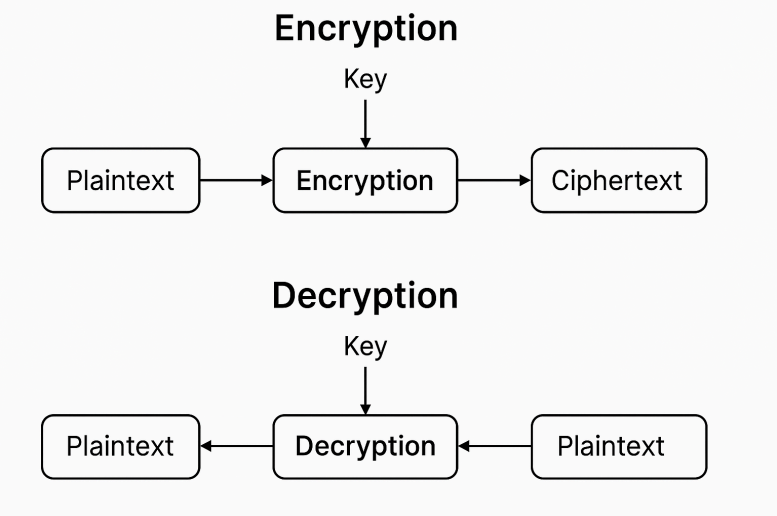


Figura 2: Diagrama operacional de AES.

AES puede operar en varios modos para cifrar datos de longitud mayor a 128 bits. Los modos más comunes incluyen ECB (Electronic Codebook), que cifra cada bloque independientemente; CBC (Cipher Block Chaining), que encadena los bloques usando el resultado del bloque anterior; CFB (Cipher Feedback) y OFB (Output Feedback), que convierten AES en un cifrador de flujo.

AES ha resistido extensos análisis criptográficos durante más de dos décadas. Los ataques más efectivos conocidos son marginalmente mejores que la fuerza bruta, pero siguen siendo impracticables. Su estructura se basa en una serie de transformaciones matemáticas que incluyen sustituciones, permutaciones y mezclas, lo que garantiza que incluso un pequeño cambio en la entrada o en la clave produzca un resultado completamente diferente en el texto cifrado.

Debido a ello, AES es resistente a ataques criptográficos, como el análisis de frecuencia y los ataques de fuerza bruta, pero permite el descifrado de los datos si se conoce la clave de cifrado, por lo cual es un algoritmo reversible.

La utilización del algoritmo AES es altamente popular y está muy extendida en aplicaciones de todo tipo, debido a que como bien su nombre indica, es un estándar de seguridad adoptado por la gran mayoría de desarrolladores de software contemporáneos, dada su utilidad en la protección de datos sensibles y su capacidad de preparación de datos para su posterior envío.

En este proyecto, la implementación de AES complementa al algoritmo SHA-256, proporcionando capacidades de cifrado reversible que permiten no solo verificar la integridad de los datos sino también proteger su confidencialidad. Mientras SHA-256 se utiliza principalmente para la verificación de integridad en la cadena de bloques, AES se emplea para cifrar datos sensibles antes de su almacenamiento en la cadena.

2.3 NEXYS A7 (FPGA)

La placa Nexys A7 es una plataforma de desarrollo basada en FPGA que utiliza el chip Artix-7 de Xilinx. Una FPGA (Field Programmable Gate Array) es un dispositivo semiconductor que puede ser configurado por el usuario para implementar funciones lógicas específicas después de la fabricación. Representa una solución intermedia entre los circuitos integrados de aplicación específica (ASIC[2]) y los procesadores de propósito general.

Las FPGAs modernas incluyen bloques especializados como bloques de memoria RAM distribuida y de bloque, multiplicadores y unidades de procesamiento digital de señales (DSP), gestores de reloj (CMT), y bloques de entrada/salida (IOB) que proporcionan interfaces con el mundo exterior. La configuración de una FPGA se realiza mediante un archivo binario llamado bitstream[3] que define la funcionalidad de cada elemento configurable. Las principales ventajas de las FPGAs incluyen su flexibilidad para implementar diseños personalizados, la capacidad de prototipado rápido, el paralelismo inherente que permite acelerar algoritmos específicos y la posibilidad de reconfiguración dinámica.

Las FPGAs se utilizan ampliamente en telecomunicaciones para el procesamiento de señales, en sistemas de defensa y aeroespaciales para aplicaciones críticas, en computación de alto rendimiento para acelerar algoritmos específicos, en prototipado de sistemas digitales, y en aplicaciones de procesamiento de imágenes y video en tiempo real.

La Nexys A7 presenta características técnicas que la hacen altamente adecuada para este proyecto. Está equipada con un FPGA Artix-7[4] modelo XC7A100T-1CSG324C que cuenta con 15,850 slices lógicos, 4,860 Kbits de memoria de bloque y 240 DSP slices[4], proporcionando recursos suficientes para implementaciones moderadamente complejas.

La conectividad es otro punto fuerte: con puerto Ethernet 10/100, USB-UART, USB HID host para ratones y teclados, y puerto MicroSD. Para la interacción con el usuario, cuenta con 16 interruptores, 16 LEDs, display de 7 segmentos, botones, y un acelerómetro de 3 ejes. Además, incluye conectores PMOD para módulos de expansión y puede ser alimentada por USB o por una fuente externa.

Comparada con otras plataformas FPGA disponibles en el mercado, la Nexys A7 representa un equilibrio óptimo entre recursos disponibles, coste y facilidad de uso para este proyecto. A continuación se presenta una tabla comparativa entre diferentes placas FPGA popularmente utilizadas en proyectos académicos y profesionales.

2.3.1 Comparativa con otras plataformas FPGA

Plataforma	Ventajas	Desventajas	Idoneidad para blockchain
Nexys A7 (Artix-7)	Buen equilibrio entre recursos y precio, memoria flash integrada	Recursos moderados	Alta - Ideal para prototipos
Basys 3	Más económica, menos consumo	Menos recursos, sin memoria flash integrada	Media - Limitada para implementaciones complejas
Zybo Z7 (Zynq-7000)	Incluye procesador ARM, mayor flexibilidad	Mayor coste, mayor complejidad	Alta - Permite combinar software y hardware
DE10-Nano (Cyclone V)	Incluye procesador ARM, coste bajo	Ecosistema Intel/Altera	Media - Requiere adaptación significativa
UltraScale+	Muchos más recursos, mayor rendimiento	Coste muy elevado, gran consumo	Baja - Sobredimensionada para prototipos

Figura 3: Tabla comparativa entre diferentes modelos de placas FPGA

2.3.3 Implementaciones existentes en Nexys A7

Actualmente existen diferentes implementaciones de algoritmos criptográficos y diferentes proyectos con temática lindante a la temática blockchain desarrollados en la placa Nexys A7, como por ejemplo:

- **Encryption Algorythm Benchmarking with the Nexys A7**[5]**:**
  Un análisis comparativo de diferentes algoritmos de cifrado, implementados todos en la placa FPGA Nexys A7

- **Implementation of RSA in Verilog[6]:**
  Una implementación completa del algoritmo RSA en Verilog[7] realizada sobre la plataforma Nexys A7.

- **AES Implementation in Verilog[8]:**
  Una implementación del algoritmo cirptográfico AES en Verilog, sobre la placa Nexys A7.

A pesar de la existencia de implementaciones de algoritmos particulares, no existe ninguna implementación completa publicada de un sistema blockchain funcional desarrollado puramente en el entorno FPGA Nexys A7.

3. INGENIERÍA DE CONCEPCIÓN

La ingeniería de concepción representa la fase donde se definieron los conceptos fundamentales y se tomaron las decisiones de diseño principales del proyecto. En esta etapa se analizaron los requisitos específicos del sistema, se evaluaron diferentes alternativas de implementación y se estableció la arquitectura general que guiaría todo el desarrollo posterior. El objetivo de esta fase fue transformar la idea inicial en un diseño conceptual sólido y bien fundamentado.

3.1 Análisis de requisitos

El primer paso en la ingeniería de concepción fue la identificación y análisis de los requisitos del sistema blockchain a implementar. Este análisis permitió establecer claramente los objetivos y limitaciones del proyecto, los requisitos mínimos planteados fueron los siguientes:

- **Almacenamiento seguro de datos:**
  El sistema debe permitir almacenar información en bloques enlazados criptográficamente, de manera ordenada y realizando una verificación y validación previa antes de añadir un bloque a la cadena.

- **Verificación de integridad:**
  El sistema debe contar con la funcionalidad de evaluar la cadena en su totalidad para asegurar la integridad de sus datos y detectar modificaciones maliciosas en bloques ya integrados en la cadena.

- **Cifrado de datos:**
  El sistema debe proporcionar mecanismos para cifrar información sensible antes de su almacenamiento en la cadena y de su transmisión entre los diferentes nodos de la red.

- **Operación autónoma:**
  El sistema debe funcionar directamente en el entorno en el que se ejecute, sin depender de sistemas externos para sus operaciones básicas.

- **Intercomunicación:**
  El sistema debe de constar de diferentes nodos que conformen la red, con mecanismos de comunicación y validación que tengan en cuenta la decisión consensuada de los nodos partícipes.

- **Distribución de la información:**
  La información de la cadena de bloques ha de estar distribuida; cada nodo debe albergar una copia de la cadena de bloques idéntica a las copias del resto de nodos.

- **Interfaz de usuario:**
  El programa implementado debe contar con una interfaz mínima de comunicación con el mundo exterior, la cual permita al usuario interactuar con la cadena de bloques y recibir información acerca del estado de esta.

3.2 Decisiones de diseño principales

Una vez se habían establecido claramente los requisitos principales del proyecto, y una vez evaluados los recursos disponibles, se realizó un diseño estructural para todos los componentes del sistema blockchain. Las decisiones tomadas y los motivos que llevaron a dicha toma de decisiones se relata de manera individual a continuación.

3.2.1 Elección de Visual Studio Code como entorno de desarrollo

Se escogió VSC (Visual Studio Code) como entorno de desarrollo en base a las ventajas que este nos iba a brindar.

VSC es un entorno de desarrollo gratuito que permite ser instalado en diferentes sistemas operativos (en mi caso Linux) y soporta múltiples lenguajes de programación de manera nativa, además de contar con la ampliación de dicho catálogo mediante extensiones instalables dentro del mismo programa.

Adicionalmente cuenta con herramientas de depuración de código y cuenta con una extensión que integra PlatformIO[9] dentro del entorno, lo cual hizo que VSC fuera el candidato predilecto en lo que a programas de desarrollo de código respecta.

Finalmente se decidió que todo el sistema se programaría en lenguaje C por su naturaleza de lenguaje de nivel intermedio-bajo, la cual le proporciona estructuras de control, funciones y tipos

de datos que son características de los lenguajes de alto nivel, lo que facilita la programación y la legibilidad del código; pero a su vez permite la interacción y manipulación directa de memoria.

3.2.2 Elección de Nexys A7 como plataforma de desarrollo

La placa Nexys A7 se estableció como la plataforma de desarrollo a utilizar debido a que es una placa ampliamente utilizada por la Facultad de Física de la Universidad de Barcelona, por lo cual en caso de necesitar algún tipo de soporte, sería de fácil búsqueda y acceso. Esta FPGA también cuenta con una amplia variedad de datasheets y documentos técnicos disponibles en internet[10].

Además, la comparación mostrada en apartados anteriores demuestra que es una placa realmente adecuada para el contexto de este proyecto, pues sus capacidades y características tales como la potencia de computación y los puertos de comunicación inter-placa la hacen ideal para desarrollar un sistema blockchain en ellas.

Otra razón adicional para la inclinación hacia este modelo de placa fue que PlatformIO cuenta con soporte para esta plataforma concreta.

3.2.3 Elección de SHA y AES como algoritmos criptográficos

La elección de Secure Hash Algorithm y Advanced Secure Standard como algoritmos criptográficos se basó en la robustez y popularidad de dichos algoritmos.

La robustez y seguridad indiscutible de ambos algoritmos y el hecho de que mundialmente sean los algoritmos criptográficos básicos por antonomasia en contextos de verificación y protección/cifrado de datos, hizo que la decisión de utilizarlos fuera prácticamente inmediata.

Optar por la implementación 256 de ambos algoritmos estuvo fundamentado en el equilibrio entre eficacia y rendimiento para el entorno donde se iban a desarrollar[11].

3.2.4 Implementación de la lógica del programa en alto y bajo nivel

Debido a que contaba con una única placa Nexys A7 pero también contaba con el ordenador donde se iba a realizar el desarrollo, se decidió que la red de nodos que implementase mi sistema blockchain iba a estar formada por una placa FPGA Nexys A7 (bajo nivel) y un ordenador (alto nivel). Crear la lógica del sistema blockchain tanto en alto nivel como en bajo nivel me iba a permitir darle diferentes enfoques a cada opción y estudiar las diferentes ventajas y desventajas de cada una de ellas, junto con el reto de integrarlas posteriormente.

De manera adicional, crear primero la lógica a alto nivel, me permitiría simplificar y facilitar todo

el proceso de concepción, permitiendo que la lógica a bajo nivel se pudiese realizar mediante una migración hacia bajo nivel desde la implementación original.

Finalmente se decidió que la implementación a alto nivel se realizaría utilizando ficheros .txt como bloques de la cadena, debido a la facilidad que proporcionaba este enfoque a la hora de depurar el código o de visualizar el contenido de los bloques de la cadena.

La implementación a bajo nivel se realizaría definiendo una estructura en C y guardándola directamente en la memoria de la placa, debido a la falta de un sistema de archivos de la Nexys A7.

3.2.5 Elección del puente UART-USB para la comunicación

La elección del puente UART-USB para la comunicación entre nodos vino condicionada por la decisión tomada previamente. Puesto que la comunicación entre nodos consistiría en una comunicación FPGA-PC, optar por utilizar la interfaz UART-USB fue la mejor decisión, ya que es un puente de comunicación ya integrado en el SoC[12] de la placa y fácil de manipular.

3.3 Arquitectura del sistema

Tras todo el proceso de selección de recursos y metodologías, la arquitectura ideada fue la siguiente:

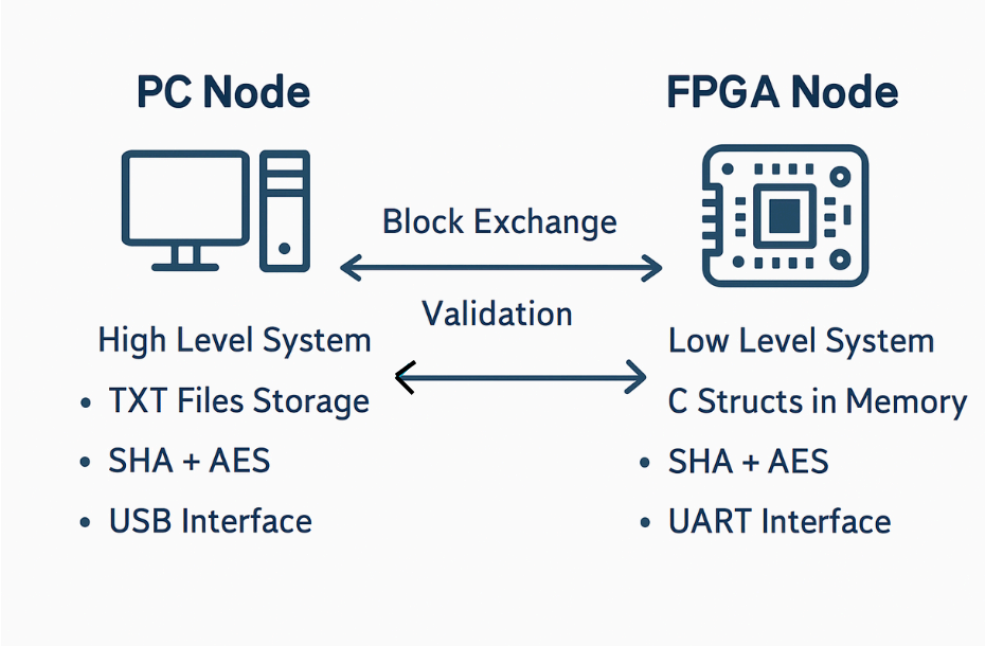


Figura 4: Ilustración de la arquitectura del sistema global ideado

- Nodo PC:
    - Implementación a alto nivel de la lógica de blockchain
    - Almacenado de los bloques de la cadena como ficheros .txt en el sistema de archivos del ordenador
    - Comunicación a través de USB

- Nodo FPGA:
    - Implementación a bajo nivel de la lógica blockchain
    - Almacenado de los bloques de la cadena como structs guardados directamente en memoria
    - Comunicación a través de UART

3.4 Diseño específico

El diseño de la lógica blockchain que se implementaría en cada nodo de la red se estructuró siguiendo una arquitectura modular de tres capas claramente diferenciadas, cada una con responsabilidades específicas y bien definidas. Esta separación permite un desarrollo ordenado, facilita el mantenimiento del código y facilita la reutilización de componentes.

La capa más básica y elemental está constituida por el módulo criptográfico, responsable de implementar los algoritmos SHA-256 y AES-256. Esta capa se encarga exclusivamente de las operaciones de cifrado, descifrado y generación de hashes. Su diseño independiente permite que las funciones criptográficas puedan ser utilizadas por cualquier otro componente sin dependencias externas.

Sobre la capa criptográfica se construye el módulo de lógica blockchain, que implementa la funcionalidad específica de la cadena de bloques. Esta capa se encarga de la creación de bloques, la gestión de la estructura de la cadena, y la verificación de integridad tanto de bloques individuales como de la cadena completa. Para realizar estas operaciones, utiliza los servicios proporcionados por la capa criptográfica, manteniendo una separación clara de responsabilidades.

Finalmente, la capa superior implementa la interfaz de usuario y actúa como coordinador general del sistema. Esta capa unifica las funcionalidades de las capas inferiores, proporcionando los menús de interacción y gestionando el flujo general de operaciones del sistema blockchain. Su diseño permite que el usuario acceda a todas las funcionalidades del sistema de manera intuitiva y organizada. El diseño de capas de aplicación resultante es el siguiente:

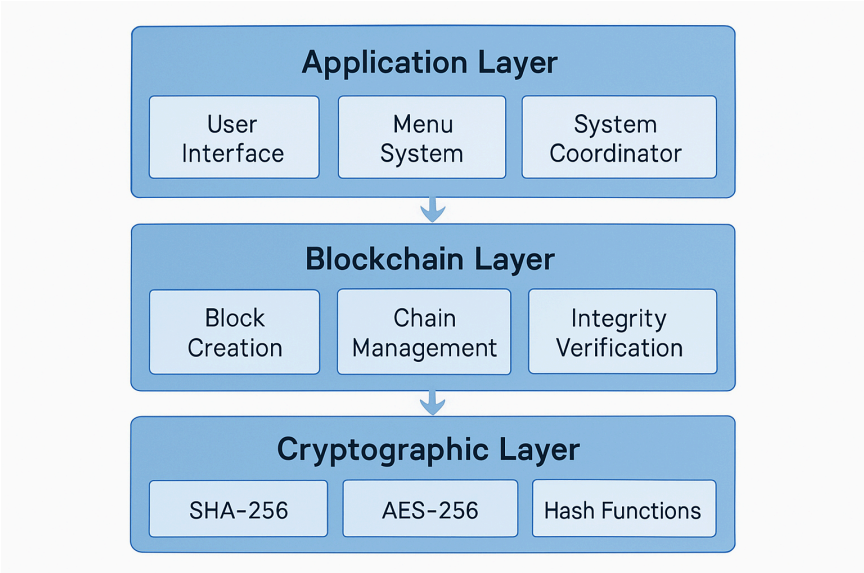


Figura 5: Diagrama de la arquitectura específica de la lógica del programa implementado en cada nodo

3.5 Exploración de alternativas

Durante las fases iniciales y las fases de desarrollo se contemplaron diferentes maneras de implementar el sistema blockchain, evaluando con detenimiento sus ventajas e inconvenientes.

Las alternativas presentadas pero no escogidas fueron las siguientes:

- **Implementación completa en VHDL/Verilog:**
  Esta alternativa implicaría programar directamente la lógica de la FPGA para implementar los algoritmos criptográficos y el sistema blockchain en VHDL/Verilog. Ofrecería el mejor rendimiento pero requeriría un tiempo de desarrollo significativamente mayor y una curva de aprendizaje extremadamente pronunciada, dado que nunca he trabajado con dichos lenguajes.

- **Implementación completa en bajo nivel:**
  Esta alternativa implicaría programas la lógica del sistema blockchain directamente en bajo nivel en ambos nodos de la red (FPGA y PC). Ofrecería una menor dificultad de integración entre implementaciones (nula, ya que serían la misma implementación), pero requeriría de diseñar y utilizar un emulador o un sistema de virtualización de la Nexys A7 dentro del PC. Además de resultar en un enfoque menos enriquecedor en términos de diversificación de implementaciones de un sistema blockchain.

4. INGENIERÍA DE DETALLE

Una vez establecidas las bases conceptuales y tomadas las decisiones fundamentales de diseño en la fase de estructuración, las cuales están detalladamente explicadas en el apartado de ingeniería de concepción, se procedió a la implementación técnica detallada del sistema blockchain. Esta fase se caracterizó por el desarrollo específico de cada componente del sistema, desde la configuración inicial del entorno hasta la integración final de todos los elementos en un sistema funcional.

El proceso de implementación se estructuró siguiendo una metodología incremental, comenzando con el desarrollo y validación de los componentes criptográficos fundamentales, continuando con la implementación de un prototipo a alto nivel para validar la lógica del sistema, y con la migración completa a bajo nivel para operar directamente sobre la placa; finalmente se desarrolló un protocolo de comunicación entre los 2 nodos de la red blockchain, el cual integra las implementaciones pertenecientes a diferentes niveles.

4.1 Implementación de algoritmos criptográficos

Al ser la base de la aplicación y las futuras librerías esenciales del programa, comencé implementando los algoritmos criptográficos SHA y AES respectivamente.

4.1.1 Implementación de SHA

La implementación de SHA presente en este proyecto es una versión genérica de dicho algoritmo, pero con ciertas peculiaridades.

Durante la fase de investigación y la fase de desarrollo del estado del arte, me topé con múltiples implementaciones de este algoritmo en diferentes lenguajes de programación, y las más adecuadas para mi proyecto no estaban capacitadas para ejecutarse en la placa Nexys A7. El problema de dichas implementaciones, es que trabajaban sobre archivos y se utilizaban características de ejecución propias de un ordenador, y la placa Nexys A7 al no contar con dichas características ni con un sistema de gestión de archivos no podría soportar dicha implementación.

Debido a esto, se realizó una implementación propia adaptada al entorno FPGA:

- **Adaptación:**
  En lugar de realizar el cálculo de hash sobre archivos, se recibe por parámetro un array con la información a hashear contenida dentro. Esto permite que la placa FPGA pueda soportar la implementación y utilizarla de manera autocontenida.

- **Modularización:**

En lugar de contar con un único archivo de código que realizase la función de aplicación e implementase la lógica SHA de manera conjunta, se crearon un archivo .h (cabecera) y un .c (programa en C) cuya única responsabilidad es calcular el resultado hash del array que reciben como parámetro. Este enfoque permitió contar con una librería independiente reutilizable y ejecutable en prácticamente cualquier entorno con unos recursos mínimos.

El resto del desarrollo de la función criptográfica sigue el esquema estándar:

- **Inicialización de constantes:**
  Se establecen valores iniciales fijos para las variables de trabajo y las constantes del algoritmo

- **Padding del mensaje:**
  Se añade relleno al mensaje en caso de necesitarlo, para que el mensaje tenga una longitud específica requerida por el algoritmo (múltiplo de 512 bits)

- **División en bloques:**
  Los datos se dividen en bloques de 512 bits y se realizan las rondas de operaciones matemáticas y lógicas pertinentes (bitshifts, rotaciones, etc.)

- **Generación de hash:**
  Finalmente se combinan los resultados del procesado de cada bloque y así se obtiene el hash resultante

```c
// SHA-256 constants
static const uint32_t K[64] = {
    0x428a2f98, 0x71374491, 0xb5c0fbcf, 0xe9b5dba5, 0x3956c25b, 0x59f111f1, 0x923f82a4, 0xab1c5ed5,
    0xd807aa98, 0x12835b01, 0x243185be, 0x550c7dc3, 0x72be5d74, 0x80deb1fe, 0x9bdc06a7, 0xc19bf174,
    0xe49b69c1, 0xefbe4786, 0x0fc19dc6, 0x240ca1cc, 0x2de92c6f, 0x4a7484aa, 0x5cb0a9dc, 0x76f988da,
    0x983e5152, 0xa831c66d, 0xb00327c8, 0xbf597fc7, 0xc6e00bf3, 0xd5a79147, 0x06ca6351, 0x14292967,
    0x27b70a85, 0x2e1b2138, 0x4d2c6dfc, 0x53380d13, 0x650a7354, 0x766a0abb, 0x81c2c92e, 0x92722c85,
    0xa2bfe8a1, 0xa81a664b, 0xc24b8b70, 0xc76c51a3, 0xd192e819, 0xd6990624, 0xf40e3585, 0x106aa070,
    0x19a4c116, 0x1e376c08, 0x2748774c, 0x34b0bcb5, 0x391c0cb3, 0x4ed8aa4a, 0x5b9cca4f, 0x682e6ff3,
    0x748f82ee, 0x78a5636f, 0x84c87814, 0x8cc70208, 0x90befffa, 0xa4506ceb, 0xbef9a3f7, 0xc67178f2
};
```

Figura 6: Inicialización de las constantes del algoritmo en mi implementación. Estas constantes se obtienen de la derivación decimal de los K primeros números primos

4.1.2 Implementación de AES

La implementación del algoritmo AES siguió un desarrollo similar al presentado en el caso de SHA; durante la fase de investigación se encontraron implementaciones interesantes existentes, pero todas contaban con los mismos problemas encontrados en el caso anterior: se realizaban en diferentes lenguajes y operaban sobre archivos específicos, lo cual una vez más imposibilitaban su integración con la placa Nexys A7.

Por lo tanto se desarrolló nuevamente una implementación adaptada basada en las aproximaciones estándar:

- **Adaptación:**
  Se adaptó el programa para recibir por parámetro un array con la información a cifrar contenida dentro. Esto permite que la placa FPGA pueda soportar la implementación y utilizarla de manera autocontenida.

- **Modularización:**
  En lugar de contar con un único archivo de código que realizase la función de aplicación e implementase la lógica AES de manera conjunta, se crearon un archivo .h (cabecera) y un .c (programa en C) cuya única responsabilidad es cifrar el contenido del array que reciben como parámetro. Este enfoque permitió contar con una librería independiente reutilizable y ejecutable en prácticamente cualquier entorno con unos recursos mínimos.

El resto del desarrollo de la función criptográfica sigue el esquema estándar:

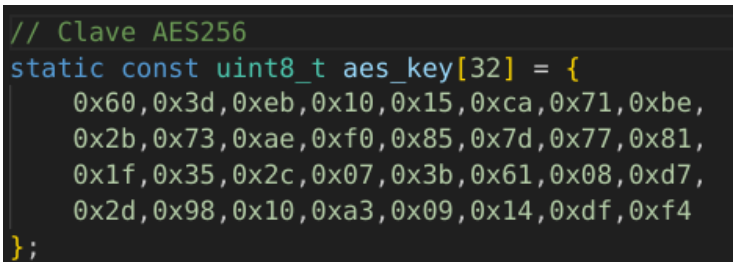
- **Expansión de clave:**
  A partir de la clave original de 256 bits se generan múltiples claves derivadas para usar en cada ronda.

- **Padding del mensaje:**
  Se añade relleno al mensaje en caso de necesitarlo, para que el mensaje tenga una longitúd específica requerida por el algoritmo (múltiplo de 128 bits en el caso de AES)

- **División en bloques:**
  Los datos se dividen en bloques de 128 bits y se realizan las rondas de operaciones matemáticas y lógicas pertinentes (bitshifts, rotaciones, etc.)

- **Generación del resultado cifrado:**
  Finalmente se combinan los resultados del procesado de cada bloque y así se obtiene el una consecución que conforma la información cifrada en su totalidad.

El desarrollo de AES se realizó implementándolo en modo ECB (Electronic CodeBook), lo cual implica que cada bloque se cifra de manera independiente, a diferencia de modos como por ejemplo CBC (Cipher Block Chaining) que combina cada bloque con el anterior antes de cifrarlo (modos como CBC mejoran la seguridad del cifrado debido a que eliminan patrones detectables, pero a costa de sacrificar mucho rendimiento).



```
// Clave AES256
static const uint8_t aes_key[32] = {
    0x60,0x3d,0xeb,0x10,0x15,0xca,0x71,0xbe,
    0x2b,0x73,0xae,0xf0,0x85,0x7d,0x77,0x81,
    0x1f,0x35,0x2c,0x07,0x3b,0x61,0x08,0xd7,
    0x2d,0x98,0x10,0xa3,0x09,0x14,0xdf,0xf4
};
```

Figura 7: Inicialización de la clave original de cifrado en mi implementación del algoritmo.

4.1.3 Validación

Con ambas implementaciones realizadas, finalmente se sometieron a evaluación para garantizar la validación de los resultados. Para ello se creó un programa main.c de prueba que utilizando vectores de prueba estándar proporcionados por el NIST[13] (National Institute of Standards and Technology), calcula hashes y a su vez cifra y descifra datos, comparando el contenido original con el contenido descifrado.

A continuación se muestra el resultado de la ejecución de dicho programa:

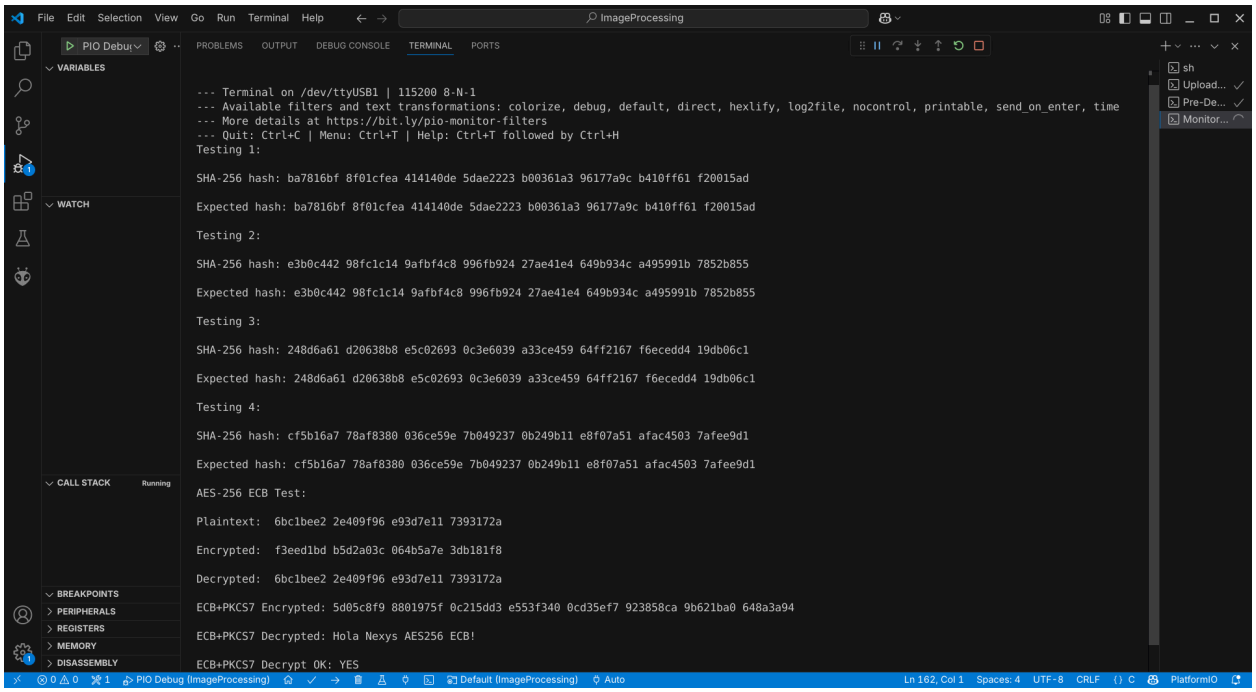


Figura 8: Resultado de ejecución del programa validador de los algoritmos SHA y AES.

4.2 Desarrollo del sistema blockchain a alto nivel

Posterior a la validación de las librerías criptográficas creadas, se desarrolló el programa que contiene la lógica de un sistema blockchain a alto nivel, el cual se ejecuta en un ordenador (el nodo de alto nivel de la red).

Este programa también se modularizó mediante su separación en ficheros .code y .header, lo cual le otorga la característica de ser reutilizable y poder implementar esta lógica en diferentes programas o interfaces de usuario.

4.2.1 Implementación del blockchain

Para comenzar con la explicación acerca de la lógica que se desarrolla en el programa blockchain, es conveniente explicar la estructura ideada y utilizada en los bloques de la cadena de bloques:

Los bloques están representados por ficheros .txt. Cada BlockX.txt es un bloque de nuestra cadena de bloques, y los datos contenidos se distribuyen en las 3 primeras líneas del fichero/documento.

- **Línea 1:**
  En esta primera línea está el hash calculado del bloque anterior, lo cuál permite enlazar los bloques entre ellos mediante una verificación de identidad y contenido.

- **Línea 2:**
  En esta línea está la información que se busca almacenar esencialmente, esta línea es la más importante del bloque pues es la que contiene los datos que se buscan proteger.

- **Línea 3:**
  En esta última línea el bloque presenta un hash de sí mismo, lo cual permite asegurar la no-modificación de los datos y actúa como referencia para el siguiente bloque de la cadena (el cuál en su primera línea tendrá escrito un hash idéntico al de esta línea)

El número de bloques y el índice del último bloque de la cadena se guarda en otro fichero Counter.txt, esta información permite al sistema saber dónde ha de añadir el bloque más reciente cuando se le solicite.

Siguiendo esta estructura, el flujo de operación del sistema blockchain es el siguiente:

- **Comienzo a partir del bloque génesis:**
  El primer bloque de una cadena siempre es especial, ya que no tiene bloque anterior, por lo que no puede contener su hash, además el bloque génesis suele contener datos arbitrarios insignificantes. En su última línea sí que contiene el hash de sí mismo, ya que es a partir de aquí donde realmente “empezará” la blockchain.

- **Adición del primer bloque (y de los n siguientes, consecutivamente):**
  Cuando el programa ya cuenta con la información que quiere guardar, procede a construir un bloque y añadirlo a la cadena de la siguiente manera:

    - Obtiene el último bloque añadido anteriormente a la cadena y calcula su hash; verifica que el hash calculado es el mismo que ese bloque refleja en su tercera línea y si coinciden, escribe dicho hash en su primera línea.

    - Cifra la información que se quiere proteger y escribe la información cifrada en su línea 2.

    - Concatena el hash del bloque anterior con la información que se quiere proteger SIN CIFRAR y calcula el hash, después escribe este mismo hash en su tercera línea, la cuál servirá como referencia para bloques futuros.

- **Actualización del fichero Counter.txt:**
  Si toda la operativa previa ha sido exitosa, el bloque se consolida en la cadena y se actualiza el fichero Counter.txt, aumentando en 1 el número de bloques de la cadena y actualizando el índice del último bloque añadido.

4.2.2 Implementación de la herramienta de verificación de cadena

Para evitar vulneraciones y modificaciones maliciosas en bloques anteriores o añadidos previamente a la cadena, los cuales ya han pasado la fase de verificación del hash y la información, se desarrolló una herramienta capaz de recorrer todos los bloques de la cadena, descifrar los datos contenidos en cada bloque y calcular y comparar los hashes entre bloques consecutivos, para así verificar los siguientes aspectos:

- Integridad de la información:
  Asegurar que dentro de un mismo bloque, el hash calculado es el que dicho bloque refleja, confirmando de esta manera que la información no ha sido adulterada.

- Consecutividad de bloques:
  Asegurar que los hashes presentes en la línea 3 de los bloques coinciden con los hashes presentes en la primera línea del bloque siguiente; demostrando así que no se ha modificado el orden de la cadena ni se ha introducido/removido ningún bloque sin autenticación ni consentimiento.

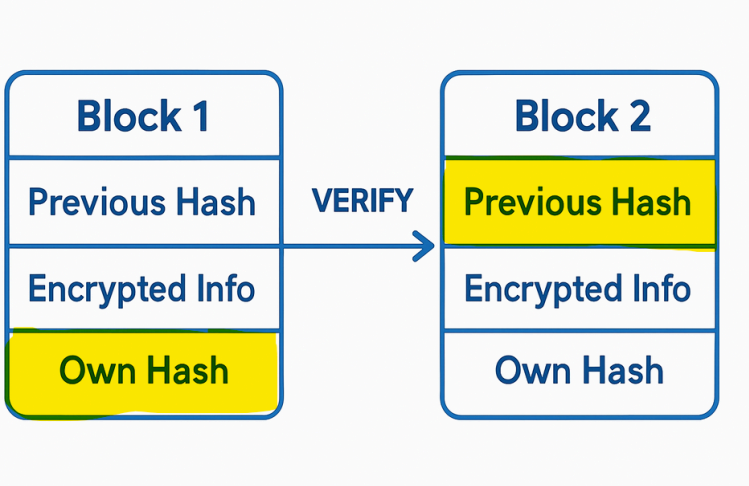


Figura 9: Comparación de los hashes de enlace entre 2 bloques consecutivos.

4.2.3 Validación de la lógica blockchain

Siguiendo la metodología aplicada con los algoritmos criptográficos, se desarrolló un programa main.c, el cual utilizando la librería blockchain recién implementada desarrolla acciones como la adición repetida de bloques a una cadena y verifica que la funcionalidad sea correcta y óptima.

A continuación se muestra el resultado de la ejecución de dicho programa:

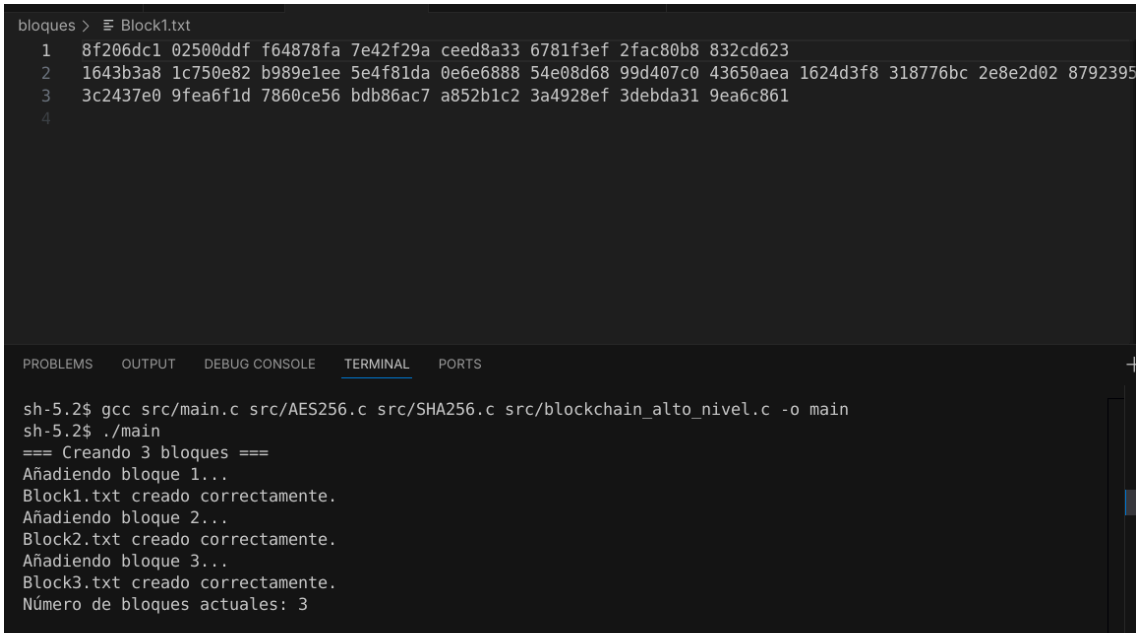


Figura 11: Resultado de ejecución del programa validador del sistema blockchain

4.3 Migración a bajo nivel

Con el sistema blockchain ideado y planteado a alto nivel, cuyo funcionamiento se había comprobado y verificado, el siguiente paso del proyecto era migrar toda la lógica del programa a bajo nivel, a un nivel donde la FPGA Nexys A7 es capaz de trabajar; sin operaciones específicas características de entornos complejos ni sistemas de archivos; trabajando directamente sobre los recursos hardware.

Esta implementación también se modularizó mediante su separación en ficheros .code y .header, lo cual le otorga la característica de ser reutilizable y poder implementar esta lógica en diferentes programas.

4.3.1 Implementación del blockchain

La implementación del sistema blockchain a bajo nivel se desarrolló concienzudamente para seguir la misma lógica que la implementación a alto nivel, las diferencias presentes entre ambas se deben a los diferentes entornos donde han de ser ejecutadas y son las siguientes:

- **Escritura directa sobre memoria:**
  En lugar de almacenar los bloques como ficheros .txt mediante el sistema de archivos del sistema operativo, como lo hace la implementación adaptada al ordenador, el blockchain a bajo nivel trabaja guardando los datos directamente sobre la memoria de la placa.

- **Estructura de bloques:**
  En lugar de distribuir el contenido de los bloques en 3 líneas de un fichero de texto, el blockchain a bajo nivel define un struct con 3 arrays que distribuye la información de igual manera, pero que permite ser almacenada en la memoria de la FPGA Nexys A7.

- **Counter:**
  En lugar de tener un .txt con el recuento de bloques de la cadena y con el índice del último bloque añadido, dicha información se almacena también en memoria.

Ambas variantes de la librería blockchain se pueden representar mediante el siguiente diagrama:

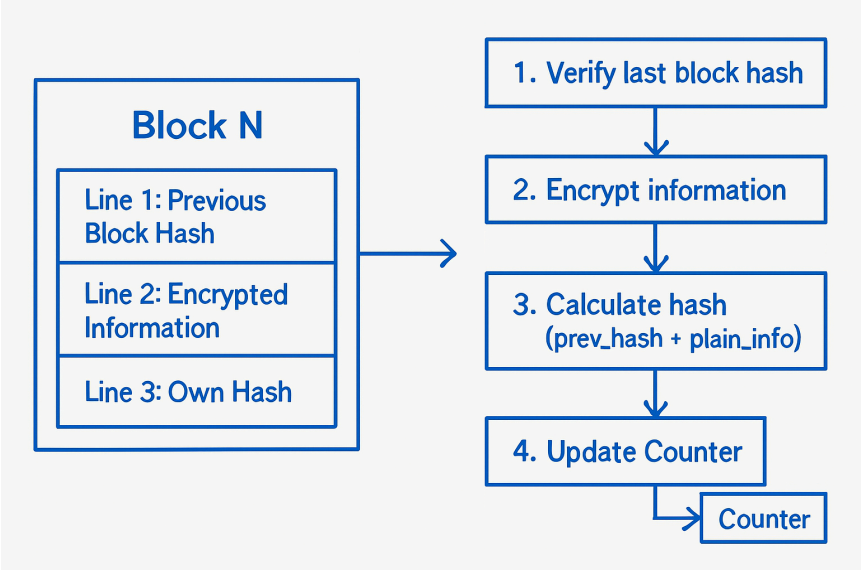


Figura 12a: Diagrama explicativo sobre la lógica/flujo de ejecución de los sistemas blockchain desarrollados.

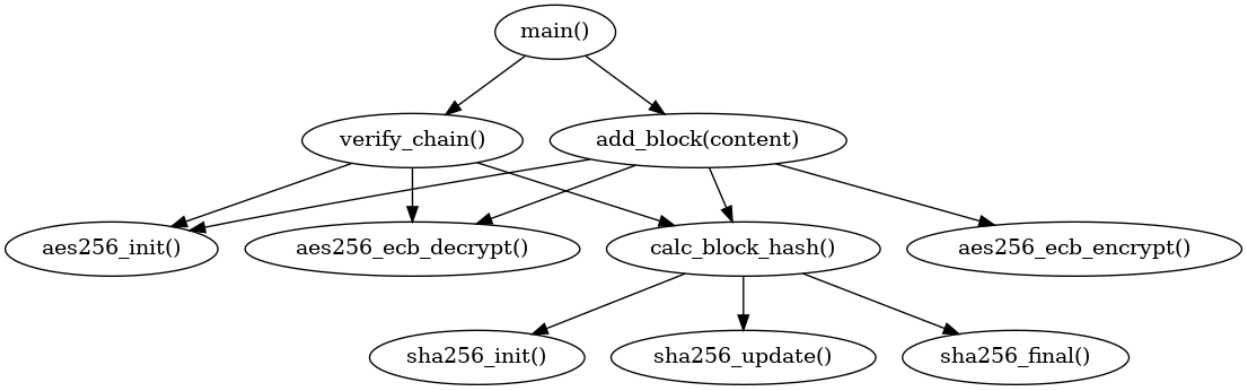


Figura 12.b: Funciones utilizadas en la implementación del sistema blockchain.

4.3.2 Implementación de la interfaz de usuario

Debido a la opacidad de trabajar con una placa como tal, se diseñó una interfaz de usuario simple que permite al usuario interactuar con la aplicación blockchain cargada en la placa a través de el puerto USB del ordenador y la UART de la Nexys A7.

La interfaz permite controlar varias acciones predefinidas del menú de la aplicación a través de los botones incorporados a bordo de la placa. De esta manera se simplifica en gran medida el envío de instrucciones y la visualización de resultados y del proceso de ejecución del programa.

```
 *  Executing task: platformio device monitor

--- Terminal on /dev/ttyUSB1 | 115200 8-N-1
--- Available filters and text transformations: colorize, debug, default, direct, hexlify, log2file, nocontrol, printable, send_on_enter, time
--- More details at https://bit.ly/pio-monitor-filters
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H

--- MENU BLOCKCHAIN ---
BTN0 (Centro): Añadir bloque válido
BTN1: Añadir bloque corrupto
BTN2: Verificar cadena
BTN3: Mostrar hashes
BTN4: Recibir una tecla por UART
Bloque válido añadido
Hash del bloque: 0803a35e2dcd235c0eb11f4c02d03f2d1f54f97fd38af7f3073f9c8566ed106c
Bloque válido añadido
Bloque válido añadido
Hash del bloque: 0803a35e2dcd235c0eb11f4c02d03f2d1f54f97fd38af7f3073f9c8566ed106c
Hash del bloque: 8ea516491f0145f5a342b5a9979a6f6aff08a116ac837f222a0c85812167ba63
Hash del bloque: 140ff39285686b0511d24bfd937b0cd42e602ad11bd2f2b01bc52c6d44abab38
```

Figura 13: Visualización del menú interactivo a través de la UART y resultado de ejecución del blockchain.

4.3.3 Validación

La validación de la implementación a bajo nivel se ha realizado mediante mi interacción con la placa a través de la interfaz explicada y comprobando particularmente las acciones realizadas por el sistema, utilizando la implementación a alto nivel como herramienta comparativa de apoyo y de referencia.

4.4 Implementación de comunicación entre nodos

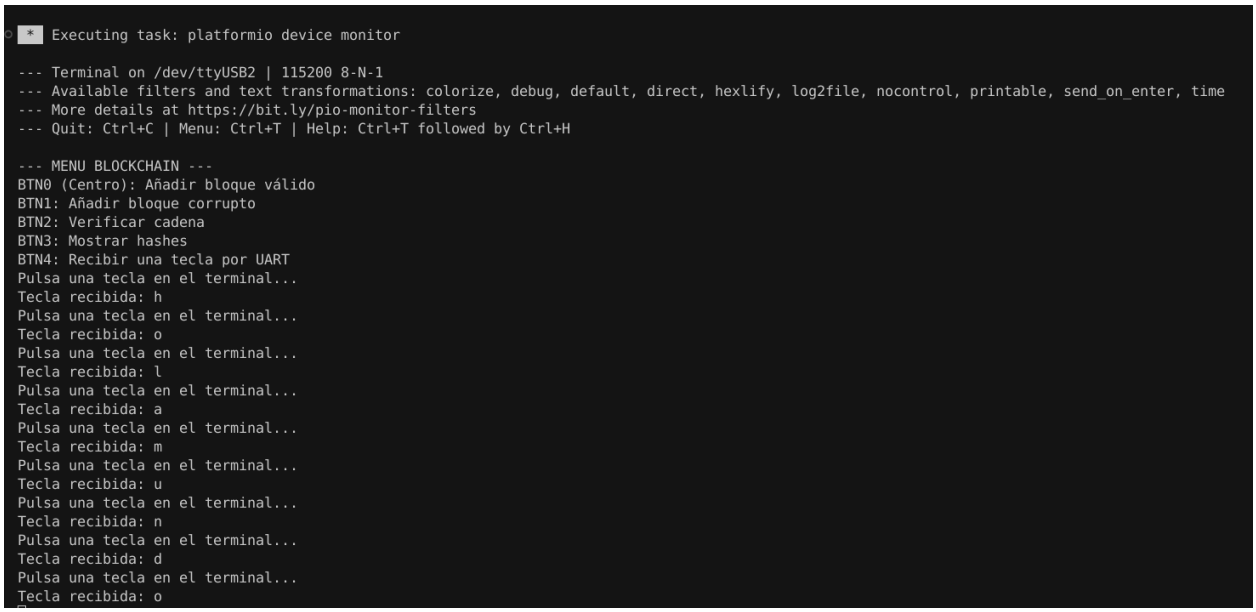
La última fase de desarrollo de este proyecto realizado de manera incremental fue la fase de comunicación entre los nodos que componen la red blockchain.

Hasta el momento, cada nodo (placa y PC) contaban con su implementación propia y trabajaban de manera individual, sin ningún tipo de comunicación ni consenso. El paso final consistió en integrar ambas capas de diferente nivel en una misma red básica de nodos, que permitieran una aproximación más cercana a lo que es, en esencia, una aplicación blockchain operada por diferentes nodos interconectados.

4.4.1 Desarrollo de la comunicación entre nodos e integración de capas de diferente nivel

El desarrollo de la comunicación entre los nodos se basa en la capacidad del PC y de la placa Nexys A7 de comunicarse entre ellos mediante las interfaces USB y UART.

Dicha comunicación se ha ido desarrollando hasta resultar en un protocolo eficiente que permite la comunicación Ordenador-FPGA. A continuación se presenta una imagen de la primera versión de comunicación entre nodos:



Con la aplicación blockchain a bajo nivel ejecutándose en la placa, y la aplicación blockchain a alto nivel ejecutándose en el PC, y con ambos nodos conectados en serie, utilizando un adaptador USB-UART: el FTDI TTL-232R-3V3-2MM Usb-Serial Cable. Se estableció un protocolo de comunicación simple, en el cual cada nodo, cada vez que procesa un bloque lo comunica al nodo homólogo para que esté a su vez también lo procese.

Cuando uno de los nodos está a la escucha y recibe un nuevo bloque para añadir a la cadena por parte del otro nodo, procesa el bloque y comunica su veredicto al resto de la red. Si el proceso resulta exitoso, todos los nodos integran el nuevo bloque en la cadena y la consolidan mediante las herramientas de verificación de cadena descritas anteriormente. Si el proceso resulta fallido, todos los nodos descartan el bloque y por precaución verifican la integridad de la cadena haciendo uso de la misma herramienta.

La integración de las capas de alto y bajo nivel para la comunicación entre nodos se puede visualizar en el siguiente diagrama simple:

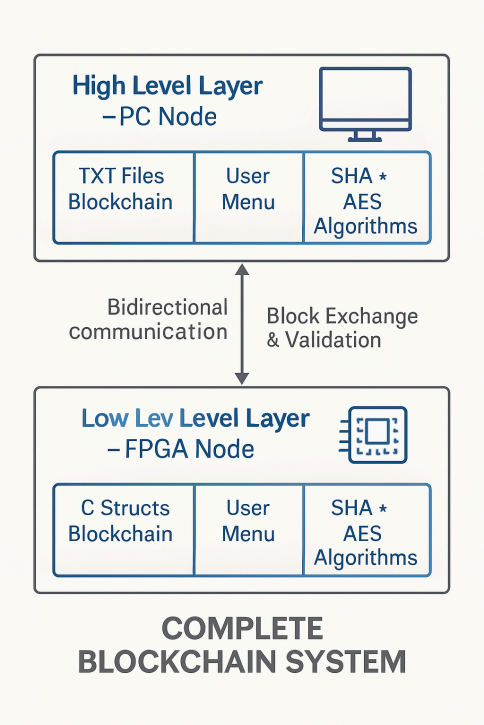


Figura 14: Diagrama de integración entre capas de diferente nivel

5. Cronograma

El desarrollo de este proyecto se ha llevado a cabo siguiendo una planificación estructurada en fases, con objetivos específicos para cada etapa. Esta organización temporal ha permitido abordar de manera sistemática los diferentes aspectos del trabajo, desde la investigación inicial hasta la implementación final del sistema blockchain.

5.1 Fases principales

El proyecto se ha dividido en cinco fases principales, cada una con objetivos específicos y entregables concretos:

- Fase de investigación y formación (Semanas 1-4)
- Fase de desarrollo de algoritmos criptográficos (Semanas 5-8)
- Fase de desarrollo de sistema blockchain a alto nivel (Semanas 9-12)
- Fase de migración a bajo nivel (Semanas 13-16)
- Fase de pruebas, validación y documentación (Semanas 17-20)

5.1.1 Fase de investigación

Durante el primer mes del proyecto, se realizó una inmersión profunda en los fundamentos teóricos necesarios para el desarrollo. Esta fase estableció una base sólida de conocimientos antes de comenzar con la implementación práctica.

Las actividades principales incluyeron:

- **Estudio de la tecnología blockchain:**
  Se investigaron los principios fundamentales, mecanismos de consenso, estructuras de datos y aplicaciones actuales. Este conocimiento fue esencial para diseñar posteriormente un sistema blockchain adaptado a las limitaciones de la placa FPGA.

- **Análisis de algoritmos criptográficos:**
  Se estudiaron en detalle los algoritmos SHA-256 y AES, comprendiendo su funcionamiento interno, propiedades de seguridad y posibles optimizaciones para entornos limitados.

- **Familiarización con la placa Nexys A7:**
  Se revisó la documentación técnica de la placa, incluyendo sus especificaciones, arquitectura y periféricos disponibles. Este conocimiento fue fundamental para entender las capacidades y limitaciones del hardware objetivo.

- **Aprendizaje del material RVFPGA:**
  Se trabajó con el material educativo Xilinx RVFPGA 3.0 de Imagination University, completando los laboratorios de aprendizaje para adquirir experiencia práctica con la placa y su entorno de desarrollo.

5.1.2 Fase de desarrollo de algoritmos criptográficos

Una vez establecidos y consolidados los conocimientos básicos obtenidos en la primera fase, dió inicio la segunda, la cual se centró en la implementación y optimización de los algoritmos criptográficos necesarios para el sistema blockchain. Esta fase se dividió y repartió de la siguiente manera:

- **Implementación de SHA-256:**
  Se desarrolló una versión eficiente del algoritmo en lenguaje C, adaptada para ejecutarse en la placa Nexys A7. Se realizaron pruebas exhaustivas para verificar su corrección utilizando vectores de prueba estándar.

- **Implementación de AES:**
  Se desarrolló el algoritmo de cifrado simétrico, prestando especial atención a la optimización del uso de memoria y rendimiento computacional.

- **Pruebas y validación:**
  Se realizaron pruebas comparativas con implementaciones de referencia para asegurar la corrección de los algoritmos y medir su rendimiento en la placa.

- **Optimización:**
  Se identificaron y aplicaron mejoras para reducir el tiempo de ejecución y el uso de recursos, considerando las limitaciones específicas de la plataforma.

Al finalizar esta fase, se contaba con implementaciones funcionales y optimizadas de ambos algoritmos criptográficos, validadas mediante pruebas estandarizadas y listas para ser integradas en el sistema blockchain.

5.1.3 Fase de desarrollo del sistema blockchain a alto nivel

Durante esta fase, se implementó una versión inicial del sistema blockchain utilizando archivos de texto para simular los bloques de la cadena. Este enfoque permitió desarrollar y probar la lógica del sistema sin las complicaciones adicionales del acceso directo a la memoria. Las actividades principales incluyeron:

- **Diseño de la estructura de bloques:**
  Se definió la estructura de datos para representar los bloques de la cadena, incluyendo el hash del bloque anterior, los datos y el hash del bloque actual.

- **Implementación con archivos de texto:**
  Se desarrolló un sistema que utilizaba archivos (Block0.txt, Block1.txt, etc.) para representar los bloques de la cadena, facilitando la visualización y depuración.

- **Desarrollo de funciones de verificación:**
  Se implementaron algoritmos para verificar la integridad de la cadena, comprobando tanto la consistencia interna de cada bloque como la validez de los enlaces entre bloques consecutivos.

- **Pruebas de concepto:**
  Se realizaron pruebas para verificar el funcionamiento correcto del sistema, incluyendo la implementación de un programa capaz de recorrer la cadena de bloques y diseñado y capacitado para la detección de modificaciones no autorizadas en los bloques.

5.1.4 Fase de migración a bajo nivel

Contando ya con los algoritmos criptográficos funcionales y con una implementación inicial a alto nivel de la logica del sistema blockchain, la cuarta fase se centró en adaptar el sistema blockchain para trabajar directamente con la memoria de la placa Nexys A7, eliminando la dependencia de archivos de texto y sistemas de archivos.

La distribución de las tareas y los desarrollos fue la siguiente:

- **Análisis de la memoria:**
  Se estudió en detalle la estructura y funcionamiento de la memoria de la placa, incluyendo su organización en sectores y protocolo de comunicación SPI.

- **Adaptación del sistema blockchain:**
  Se modificó el sistema para trabajar directamente con la memoria, definiendo una estructura compacta para los bloques y desarrollando funciones para su lectura y escritura.

Al finalizar esta fase, se disponía de un sistema blockchain completamente funcional que operaba directamente en la memoria de la placa, sin necesidad de sistemas de archivos o capas de software intermedias.

5.1.5 Fase de pruebas, validación y documentación

La fase final del proyecto se dedicó a la realización de pruebas exhaustivas del sistema completo, la optimización de aspectos específicos y la documentación detallada del trabajo realizado.

- **Pruebas de integración:**
  Se verificó el funcionamiento conjunto de todos los componentes del sistema, asegurando que la cadena de bloques se creaba y verificaba correctamente.

- **Pruebas de seguridad:**
  Se realizaron intentos controlados de modificación de la cadena para verificar la capacidad del sistema para detectar alteraciones.

- **Documentación técnica:**
  Se elaboró documentación detallada del diseño, implementación y funcionamiento del sistema, incluyendo diagramas, especificaciones y guías de uso.

- **Redacción de la memoria:**
  Se preparó la memoria final del Trabajo de Fin de Grado, documentando todo el proceso de desarrollo, resultados obtenidos y conclusiones.

5.2 Diagrama de Gantt

A continuación se presenta un diagrama de Gantt simplificado que muestra la distribución temporal de las diferentes fases y actividades del proyecto:

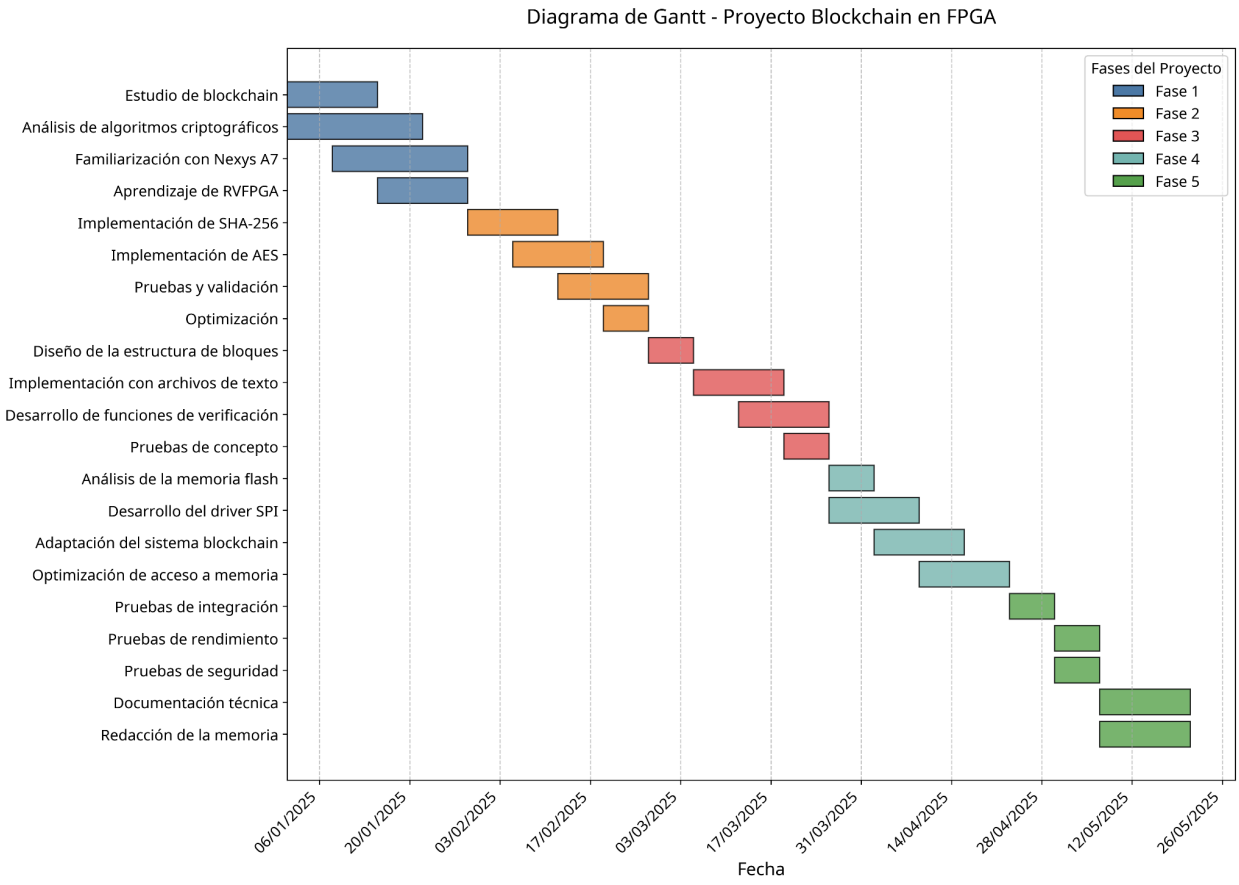


Figura 2: Diagrama de Gantt ilustrativo con las fases y los periodos de desarrollo

6. COSTE ECONÓMICO

En esta sección se presenta un análisis detallado de los costes asociados al desarrollo del proyecto, incluyendo tanto los costes de hardware y software como una estimación de los recursos humanos necesarios.

Componente	**Tipo**	**Descripción**	**Coste (€)**
Placa Nexys A7	Hardware	FPGA Artix-7 de Xilinx con periféricos integrados	299.00
Cable USB	Hardware	Cable ASB-A a micro-USB para conexión con el ordenador	3.00
Xiaomi RedmiBook 16	Hardware	Equipo utilizado durante las fases de programación y depuración	900.00
Total Hardware			**1202.00**
Visual Studio Code	Software	Entorno de desarrollo utilizado durante las fases de programación y depuración	0.00
PlatformIO	Software	Extensión para desarrollos en sistemas embebidos	0.00
Total Software			**0.00**
Coste humano	Humano	Precio/hora de trabajador con rol de Ingeniero Informático	25 x 640 = 16000.00
Total Humano			**16000.000**
Coste Total del Proyecto			**17202.00 €**

Figura 3: Tabla representativa de costes económicos

6.1 Consideraciones adicionales

Es importante destacar que este análisis de costes corresponde a un proyecto de investigación y desarrollo académico. En un entorno comercial o industrial, podrían aplicarse consideraciones adicionales tales como:

- **Licencias comerciales:**
  En un entorno profesional, podría ser necesario adquirir licencias comerciales de algunas herramientas de software, especialmente para versiones completas de Vivado.

- **Hardware adicional:**
  Dependiendo de los requisitos específicos, podrían necesitarse componentes adicionales como analizadores lógicos, osciloscopios o módulos de expansión para la placa FPGA.

- **Costes de mantenimiento:**
  En una implementación a largo plazo, deberían considerarse los costes de mantenimiento y actualización del sistema.

- **Escalabilidad:**
  Para una implementación a mayor escala, sería necesario considerar costes adicionales relacionados con la infraestructura de red, servidores y sistemas de respaldo, etc.

7. CONCLUSIONES

Este Trabajo de Fin de Grado ha explorado la implementación de un sistema blockchain básico directamente en una red formada por una placa FPGA Nexys A7 y un PC, utilizando algoritmos criptográficos para garantizar la integridad y confidencialidad de los datos almacenados.

A lo largo del desarrollo, se han obtenido resultados significativos que demuestran el potencial de esta aproximación, aunque también se han identificado limitaciones que abren camino a futuras investigaciones:

El proyecto ha logrado cumplir con los objetivos planteados inicialmente, demostrando que es viable implementar un sistema blockchain funcional en un dispositivo de hardware con recursos limitados. Se han desarrollado implementaciones eficientes de los algoritmos SHA-256 y AES en lenguaje C, optimizadas para ejecutarse en la placa Nexys A7. Además dichas implementaciones han demostrado ser correctas y eficientes, proporcionando la base criptográfica necesaria para el sistema blockchain ideado para adecuarse al contexto y recursos presentes.

El resultado más relevante ha sido el desarrollo de un sistema blockchain en su totalidad, conformado por varios nodos, capaz de crear y verificar información estructurada en una cadena de bloques y almacenada directamente en la memoria de la placa, sin necesidad de sistemas operativos o capas de software intermedias que la procesen o interactúen con ella.

Este sistema no solo detecta cualquier modificación no autorizada en los bloques de la cadena, garantizando así la integridad de los datos almacenados, sino que también permite cifrar datos sensibles antes de su almacenamiento o su posterior transmisión, proporcionando un nivel adicional de seguridad y robustez.

Estos logros demuestran que las FPGAs pueden ser una plataforma viable para implementaciones de blockchain en entornos donde los recursos computacionales son limitados o donde se requiere un alto nivel de seguridad y se busca tener un control total tanto del hardware utilizado como del software implementado.

Como se ha mencionado, la capacidad de ejecutar algoritmos criptográficos directamente en hardware, sin depender de sistemas operativos vulnerables, ofrece ventajas significativas en términos de seguridad y rendimiento para aplicaciones específicas.

A pesar de los logros alcanzados, el proyecto ha enfrentado varias limitaciones que es importante reconocer. La placa Nexys A7, aunque suficiente para este proyecto, presenta restricciones en términos de capacidad de procesamiento y memoria que limitan la complejidad y el tamaño del sistema blockchain implementado. Las implementaciones en software de SHA-256 y AES, aunque

optimizadas, siguen siendo computacionalmente intensivas y afectan al rendimiento del sistema.

Otro factor limitante ha sido la velocidad de acceso a la memoria, que resulta relativamente lenta y afecta al rendimiento global del sistema, especialmente al guardar, leer o verificar cadenas largas.

Además, el sistema implementado representa una versión simplificada de blockchain, sin incluir características avanzadas como mecanismos de contratos inteligentes, que serían necesarios para aplicaciones reales más complejas.

Estas limitaciones, lejos de restar valor al proyecto, proporcionan valiosas lecciones y abren camino a futuras mejoras y desarrollos. Entre las líneas futuras de trabajo más prometedoras se encuentra la implementación directa de los algoritmos criptográficos en VHDL o Verilog (un nivel de implementación más inferior todavía, lo que los sitúa justo debajo de C, que es considerado un lenguaje intermedio), lo que podría mejorar significativamente el rendimiento del sistema.

También sería interesante optimizar la gestión de la memoria para permitir cadenas más largas y un acceso más eficiente a los bloques.

El desarrollo de mecanismos de consenso distribuido que permitan la participación de múltiples nodos en la validación de transacciones representaría un avance significativo hacia sistemas blockchain más completos.

Todas las consideraciones presentes se tratarán con mayor exhaustividad en los puntos siguientes *(véase 7.1 Ampliaciones y futuras mejoras)*.

En conclusión, este proyecto ha demostrado que es posible implementar tecnologías blockchain directamente en hardware, abriendo nuevas posibilidades para aplicaciones donde la seguridad, la eficiencia energética y la independencia de sistemas operativos son prioritarias. La combinación de FPGAs y blockchain representa un campo prometedor que merece mayor exploración, tanto en entornos académicos como industriales.

7.1 Ampliaciones y futuras mejoras

El sistema blockchain implementado en la placa FPGA Nexys A7 ha demostrado la viabilidad de ejecutar este tipo de tecnología en dispositivos con recursos limitados, logrando un funcionamiento correcto y eficiente dentro de las restricciones existentes. Sin embargo, el desarrollo actual, dada su naturaleza inauguradora, se ha centrado en un único nodo del sistema blockchain, donde la generación y validación de bloques se realiza de manera autónoma en una sola placa. Esta implementación, aunque funcional, representa solo una parte del potencial que ofrece la tecnología blockchain cuando se despliega en un entorno distribuido.

Una de las ampliaciones más prometedoras para este proyecto sería aprovechar el puerto Ethernet integrado en la placa Nexys A7 para establecer comunicación entre múltiples dispositivos idénticos. Esto permitiría crear una verdadera red blockchain distribuida, donde cada placa actuaría como un nodo independiente del sistema. Para lograr esta ampliación, sería necesario desarrollar un protocolo de comunicación específico que permitiera a las placas intercambiar información sobre bloques, verificar transacciones de manera colectiva y alcanzar consenso sobre la validez de nuevos bloques.

La implementación de este sistema distribuido introduciría desafíos adicionales como la sincronización entre nodos, la resolución de conflictos cuando existen versiones divergentes de la cadena, y el desarrollo de mecanismos de consenso adaptados a las capacidades de las placas FPGA. Algoritmos como Proof of Work podrían resultar demasiado intensivos computacionalmente para estos dispositivos, por lo que sería interesante explorar alternativas como Proof of Stake o variantes simplificadas de Byzantine Fault Tolerance que pudieran ejecutarse eficientemente en este entorno.

Otra mejora significativa sería aprovechar el slot para tarjetas SD que incorpora la placa Nexys A7. Actualmente, el sistema utiliza la memoria flash integrada de 16MB para almacenar la cadena de bloques, lo que limita considerablemente el tamaño máximo que puede alcanzar. Al desarrollar un driver para escribir directamente en tarjetas SD, se podría ampliar drásticamente la capacidad de almacenamiento, permitiendo cadenas mucho más largas y con bloques de mayor tamaño. Las tarjetas SD modernas ofrecen capacidades de varios gigabytes o incluso terabytes, lo que representaría un salto cualitativo en las posibilidades del sistema.

Además de la ventaja obvia en términos de capacidad, el uso de tarjetas SD podría potencialmente mejorar el rendimiento del sistema. La memoria flash integrada, accesible a través del protocolo SPI, presenta limitaciones en cuanto a velocidad de transferencia. Un estudio comparativo del rendimiento entre la memoria flash y las tarjetas SD en este contexto específico resultaría valioso para determinar la mejor estrategia de almacenamiento para futuras implementaciones. Este análisis debería considerar no solo la velocidad bruta de lectura y escritura, sino también aspectos como la latencia, el consumo energético y la durabilidad ante ciclos repetidos de escritura.

Más allá de estas dos mejoras principales, existen otras líneas de investigación que podrían enriquecer significativamente el proyecto. Una de ellas sería la implementación directa de los algoritmos criptográficos (SHA-256 y AES) en hardware mediante lenguajes como VHDL o Verilog, en lugar de utilizar implementaciones en C ejecutándose sobre un microprocesador soft-core. Esta aproximación podría mejorar drásticamente el rendimiento de las operaciones criptográficas, permitiendo procesar bloques más rápidamente y reduciendo el consumo energético.

Otra dirección interesante sería la incorporación de capacidades de contratos inteligentes simples que pudieran ejecutarse directamente en la FPGA. Aunque las limitaciones de recursos impedirían implementar un sistema completo, sería factible desarrollar un subconjunto reducido de funcionalidades que permitiera ejecutar lógica programable asociada a los bloques. Esto abriría la puerta a aplicaciones más sofisticadas como sistemas de votación, registros de propiedad o mecanismos de certificación autónomos.

La interfaz de usuario también representa un área con amplio margen de mejora. El desarrollo de una aplicación de escritorio o web que se comunicara con la placa FPGA facilitaría enormemente la interacción con el sistema blockchain, permitiendo visualizar la cadena de bloques, monitorizar su estado y ejecutar operaciones de manera intuitiva. Esta interfaz podría aprovechar la comunicación serial o Ethernet ya disponible en la placa.

Finalmente, sería valioso explorar aplicaciones específicas donde esta implementación de blockchain en FPGA ofrezca ventajas competitivas frente a soluciones software tradicionales. Ámbitos como el Internet de las Cosas (IoT) seguro, sistemas embebidos para entornos industriales o dispositivos médicos podrían beneficiarse de las características de seguridad, autonomía y eficiencia energética que ofrece esta aproximación. Un estudio de caso en alguno de estos dominios permitiría validar la utilidad práctica del sistema y orientar su evolución futura.

8. REFERENCIAS

[1] Ledger y Trezor

Ledger y Trezor son carteras de hardware utilizadas para almacenar criptomonedas de forma segura. Utilizan algoritmos criptográficos como AES (Advanced Encryption Standard) y SHA-256 (Secure Hash Algorithm 256) para asegurar las claves privadas y las transacciones de criptomonedas. Ledger emplea AES para cifrar datos sensibles y SHA-256 para generar hashes de transacciones, garantizando la integridad de los datos. Por su parte, Trezor también utiliza SHA-256 para la verificación de transacciones y AES para proteger la información almacenada en sus dispositivos

[2] ASIC

ASIC, o "Application-Specific Integrated Circuit", es un tipo de circuito integrado diseñado para realizar una tarea específica o un conjunto limitado de funciones. A diferencia de los circuitos integrados de propósito general, como los microprocesadores, los ASIC son optimizados para un uso particular, lo que les permite ser más eficientes en términos de rendimiento y consumo de energía. Se utilizan comúnmente en aplicaciones como minería de criptomonedas, procesamiento de señales, y en dispositivos electrónicos como teléfonos móviles y equipos de red.

[3] Bitstream

Un bitstream es una secuencia de bits que representa datos digitales. En el contexto de la electrónica y la informática, se utiliza para transmitir información de manera continua, ya sea a través de un medio físico, como cables, o de forma digital en almacenamiento. En el ámbito de las FPGA (Field-Programmable Gate Arrays), un bitstream también se refiere a la configuración que define cómo se debe programar el dispositivo.

[4] Artix-7

Artix-7 es una familia de dispositivos FPGA (Field-Programmable Gate Array) desarrollada por Xilinx. Los FPGAs Artix-7 son especialmente adecuados para aplicaciones que requieren procesamiento de señales, control industrial, comunicaciones y sistemas embebidos.

Los "slices" son unidades básicas de lógica en los dispositivos FPGA, como los Artix-7 de Xilinx. Cada slice contiene una combinación de elementos de lógica, como LUTs (Look-Up Tables), flip-flops y multiplexores, que permiten implementar funciones lógicas y almacenar datos.

En un FPGA, los slices se agrupan para formar bloques de lógica más grandes, lo que permite la

implementación de circuitos complejos. La cantidad de slices en un FPGA determina su capacidad para realizar operaciones lógicas y manejar múltiples señales simultáneamente. En resumen, los slices son componentes fundamentales que permiten la flexibilidad y la personalización en el diseño de circuitos digitales dentro de un FPGA.

[5] Encryption Algorythm Benchmarking with the Nexys A7

https://scholarworks.calstate.edu/concern/projects/028715250

[6] Implementation of RSA in Verilog

https://scholarworks.calstate.edu/concern/projects/76537633k

[7] VHDL/Verilog

VHDL (VHSIC Hardware Description Language) y Verilog son lenguajes de descripción de hardware utilizados para modelar y diseñar circuitos digitales y sistemas electrónicos. Ambos lenguajes permiten a los ingenieros describir el comportamiento y la estructura de los circuitos de manera textual, facilitando la simulación, síntesis y verificación de diseños.

VHDL: Es un lenguaje más formal y estructurado, que se originó en la década de 1980 para documentar el comportamiento de circuitos integrados. Su sintaxis es similar a Ada y permite una descripción detallada de sistemas complejos. VHDL es conocido por su capacidad para manejar diseños grandes y su fuerte tipado, lo que ayuda a prevenir errores.

Verilog: Es un lenguaje más sencillo y menos formal que VHDL, con una sintaxis que se asemeja a C. Se desarrolló en la misma época y es popular por su facilidad de uso y rapidez en la escritura de código. Verilog es ampliamente utilizado en la industria para el diseño y verificación de circuitos digitales.

Ambos lenguajes son fundamentales en el diseño de FPGAs y ASICs, y la elección entre ellos a menudo depende de las preferencias del equipo de diseño y los requisitos del proyecto.

[8] AES Implementation in Verilog

https://stackoverflow.com/questions/69996151/is-there-a-way-to-map-output-of-aes-algorithm-to-hex-display-on-nexys-a7

[9] PlatformIO

PlatformIO es un entorno de desarrollo integrado (IDE) para la programación de dispositivos embebidos que permite a los desarrolladores trabajar con diversas plataformas de hardware y frameworks de software. Ofrece herramientas para la creación, compilación y carga de proyectos de firmware, y es compatible con una amplia gama de placas y microcontroladores, como Arduino, ESP32 y Nexys. Además, se integra con editores de código populares, facilitando la gestión de bibliotecas y simplificando el proceso de desarrollo. Su enfoque en la flexibilidad y la facilidad de uso lo hace ideal para proyectos de Internet de las Cosas (IoT) y sistemas embebidos.

[10] Nexys A7 datasheet

https://digilent.com/reference/_media/reference/programmable-logic/nexys-a7/nexys-a7_rm.pdf

[11] Estudio de rendimiento para SHA

https://sslinsights.com/sha1-vs-sha2-vs-sha256-vs-sha512/

[12] SoC

Un SoC, o "System on Chip", es un circuito integrado que combina todos los componentes necesarios de un sistema completo en un solo chip. Esto incluye procesadores, memoria, controladores, interfaces de comunicación y, a veces, componentes adicionales como unidades de procesamiento gráfico (GPU) y módulos de conectividad. Los SoC son utilizados en una variedad de dispositivos, como teléfonos inteligentes, tabletas, dispositivos IoT y sistemas embebidos, debido a su capacidad para ofrecer un alto rendimiento y eficiencia energética en un tamaño reducido. Al integrar múltiples funciones en un solo chip, los SoC permiten la miniaturización de dispositivos y reducen los costes de fabricación.

[13] NIST

NIST, o el Instituto Nacional de Estándares y Tecnología (National Institute of Standards and Technology), es una agencia del Departamento de Comercio de los Estados Unidos. Su misión principal es promover la innovación y la competitividad industrial mediante el desarrollo de estándares, directrices y tecnologías en diversas áreas, incluyendo la metrología, la seguridad cibernética, la fabricación y la tecnología de la información. NIST también juega un papel crucial en la investigación y el desarrollo de estándares para asegurar la calidad y la interoperabilidad de productos y servicios, contribuyendo así a la confianza en el comercio y la protección del consumidor.

9. ANEXO

9.1 Nexys A7

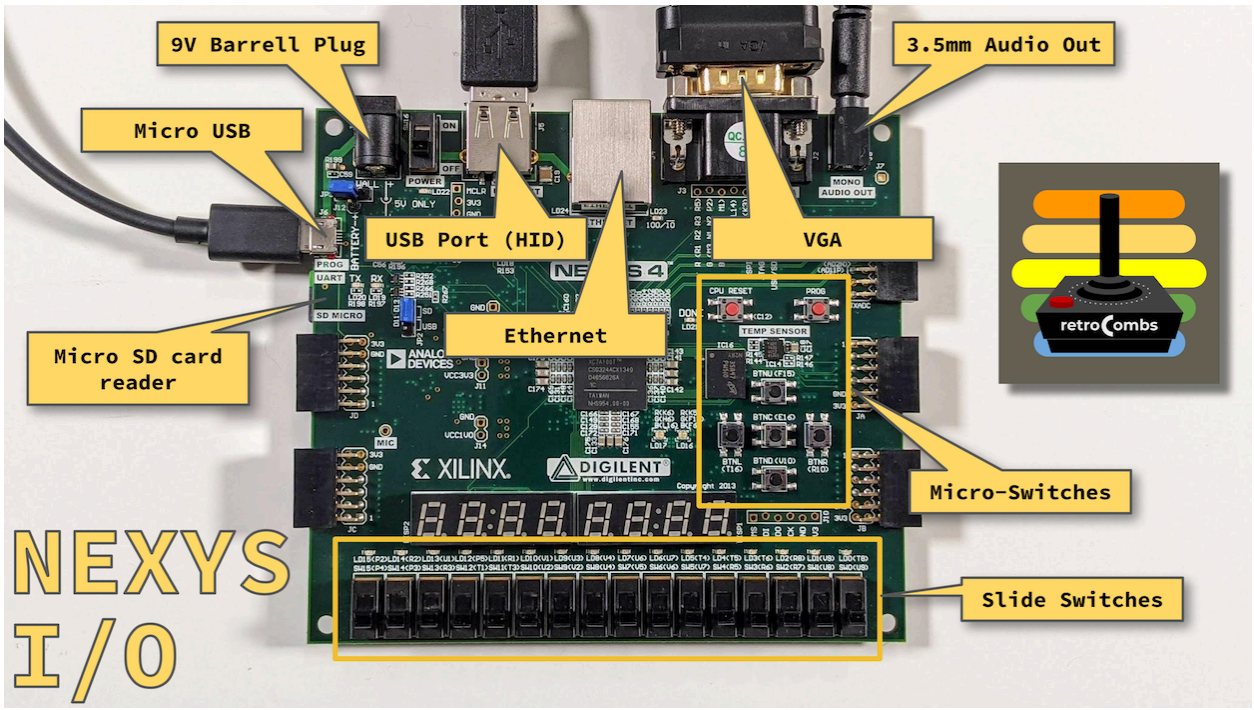


Figura anexo 1: Placa Nexys y sus componentes de a bordo