TEMARIO CURSO DE LIGHTNING

Módulos de aprendizaje	Temáticas
	I. Understanding the Lightning Network
Introduction	Lightning Network Basic Concepts
	Trust in Decentralized Networks
	Fairness Without Central Authority
	Trusted Protocols Without Intermediaries
	A Fairness Protocol in Action
	Security Primitives as Building Blocks
	Example of the Fairness Protocol
	Motivation for the Lightning Network
	Scaling Blockchains
	The Lightning Network's Defining Features
	Lightning Network Use Cases, Users, and Their Stories
	Conclusion
2. Getting Started	Alice's First Lightning Wallet
	Lightning Nodes
	Lightning Explorers
	Lightning Wallets
	Testnet Bitcoin
	Balancing Complexity and Control
	Downloading and Installing a Lightning Wallet
	Creating a New Wallet
	Responsibility with Key Custody
	Mnemonic Words
	Storing the Mnemonic Safely
	Loading Bitcoin onto the Wallet
	Acquiring Bitcoin
	Receiving Bitcoin
	From Bitcoin to Lightning Network
	Lightning Network Channels
	Opening a Lightning Channel
	Buying a Cup of Coffee Using the Lightning Network

	Commonality of Bitcoin and Lightning
	Monetary Unit
	Irreversibility and Finality of Payments
	Trust and Counterparty Risk
	Permissionless Operation
	Open Source and Open System
	Conclusion
4. Lightning Node Software	**Lightning Development Environment**
	Using the Command Line
	Downloading the Book Repository
	Docker Containers
	Bitcoin Core and Regtest
	Building the Bitcoin Core Container
	The c-lightning Lightning Node Project
	Building c-lightning as a Docker Container
	Setting Up a Docker Network
	Running the bitcoind and c-lightning Containers
	Installing c-lightning from Source Code
	Installing Prerequisite Libraries and Packages
	Copying the c-lightning Source Code
	Compiling the c-lightning Source Code
	The Lightning Network Daemon Node Project
	The LND Docker Container
	Running the bitcoind and LND Containers
	Installing LND from Source Code
	Copying the LND Source Code
	Compiling the LND Source Code
	The Eclair Lightning Node Project
	The Eclair Docker Container
	Running the bitcoind and Eclair Containers
	Installing Eclair from Source Code
	Copying the Eclair Source Code
	Compiling the Eclair Source Code
	Building a Complete Network of Diverse Lightning Nodes
	Using docker-compose to Orchestrate Docker Containers
	docker-compose Configuration
	Starting the Example Lightning Network

	Opening Channels and Routing a Payment
	Conclusion
5. Operating a Lightning Network Node	**Choosing Your Platform**
	Why Is Reliability Important for Running a Lightning Node?
	Types of Hardware Lightning Nodes
	Running in the "Cloud"
	Running a Node at Home
	What Hardware Is Required to Run a Lightning Node?
	Switching Server Configuration in the Cloud
	Using an Installer or Helper
	RaspiBlitz
	Mynode
	Umbrel
	BTCPay Server
	Bitcoin Node or Lightweight Lightning
	Operating System Choice
	Choose Your Lightning Node Implementation
	Installing a Bitcoin or Lightning Node
	Background Services
	Process Isolation
	Node Startup
	Node Configuration
	Network Configuration
	Security of Your Node
	Operating System Security
	Node Access
	Node and Channel Backups
	Hot Wallet Risk
	Sweeping Funds
	Lightning Node Uptime and Availability
	Tolerate Faults and Automate
	Monitoring Node Availability
	Watchtowers
	Channel Management
	Opening Outbound Channels
	Getting Inbound Liquidity
	Closing Channels

	Rebalancing Channels
	Routing Fees
	Node Management
	Ride The Lightning
	lndmon
	ThunderHub
	Conclusion
6. Lightning Network Architecture	**The Lightning Network Protocol Suite**
	Lightning in Detail
7. Payment Channels	**A Different Way of Using the Bitcoin System**
	Bitcoin Ownership and Control
	Diversity of (Independent) Ownership and Multisig
	Joint Ownership Without Independent Control
	Preventing "Locked" and Un-Spendable Bitcoin
	Constructing a Payment Channel
	Node Private and Public Keys
	Node Network Address
	Node Identifiers
	Connecting Nodes as Direct Peers
	Constructing the Channel
	Peer Protocol for Channel Management
	Channel Establishment Message Flow
	The Funding Transaction
	Generating a Multisignature Address
	Constructing the Funding Transaction
	Holding Signed Transactions Without Broadcasting
	Refund Before Funding
	Constructing the Presigned Refund Transaction
	Chaining Transactions Without Broadcasting
	Solving Malleability (Segregated Witness)
	Broadcasting the Funding Transaction
	Sending Payments Across the Channel
	Splitting the Balance
	Competing Commitments
	Cheating with Old Commitment Transactions
	Revoking Old Commitment Transactions
	Asymmetric Commitment Transactions

	Delayed (Timelocked) Spending to_self
	Revocation Keys
	The Commitment Transaction
	Advancing the Channel State
	The commitment_signed Message
	The revoke_and_ack Message
	Revoking and Recommitting
	Cheating and Penalty in Practice
	The Channel Reserve: Ensuring Skin in the Game
	Closing the Channel (Cooperative Close)
	The Shutdown Message
	The closing_signed Message
	The Cooperative Close Transaction
	Conclusion
8. Routing on a Network of Payment Channels	**Routing a Payment**
	Routing Versus Pathfinding
	Creating a Network of Payment Channels
	A Physical Example of "Routing"
	Fairness Protocol
	Implementing Atomic Trustless Multihop Payments
	Revisiting the Tipping Example
	On-Chain Versus Off-Chain Settlement of HTLCs
	Hash Time-Locked Contracts
	HTLCs in Bitcoin Script
	Payment Preimage and Hash Verification
	Extending HTLCs from Alice to Dina
	Back-Propagating the Secret
	Signature Binding: Preventing Theft of HTLCs
	Hash Optimization
	HTLC Cooperative and Timeout Failure
	Decrementing Timelocks
	Conclusion
9. Channel Operation and Payment Forwarding	**Local (Single Channel) Versus Routed (Multiple Channels)**
	Forwarding Payments and Updating Commitments with HTLCs
	HTLC and Commitment Message Flow
	Forwarding Payments with HTLCs
	Adding an HTLC

	The update_add_HTLC Message
	HTLC in Commitment Transactions
	New Commitment with HTLC Output
	Alice Commits
	Bob Acknowledges New Commitment and Revokes Old One
	Bob Commits
	Multiple HTLCs
	HTLC Fulfillment
	HTLC Propagation
	Dina Fulfills the HTLC with Chan
	Bob Settles the HTLC with Alice
	Removing an HTLC Due to Error or Expiry
	Making a Local Payment
	Conclusion
10. Onion Routing	**A Physical Example Illustrating Onion Routing**
	Selecting a Path
	Building the Layers
	Peeling the Layers
	Introduction to Onion Routing of HTLCs
	Alice Selects the Path
	Alice Constructs the Payloads
	Key Generation
	Wrapping the Onion Layers
	Fixed-Length Onions
	Wrapping the Onion (Outlined)
	Wrapping Dina's Hop Payload
	Wrapping Chan's Hop Payload
	Wrapping Bob's Hop Payload
	The Final Onion Packet
	Sending the Onion
	The update_add_htlc Message
	Alice Sends the Onion to Bob
	Bob Checks the Onion
	Bob Generates Filler
	Bob De-Obfuscates His Hop Payload
	Bob Extracts the Outer HMAC for the Next Hop
	Bob Removes His Payload and Left-Shifts the Onion

	Bob Constructs the New Onion Packet
	Bob Verifies the HTLC Details
	Bob Sends the update_add_htlc to Chan
	Chan Forwards the Onion
	Dina Receives the Final Payload
	Returning Errors
	Failure Messages
	Keysend Spontaneous Payments
	Custom Onion TLV Records
	Sending and Receiving Keysend Payments
	Keysend and Custom Records in Lightning Applications
	Conclusion
11. Gossip and the Channel zGraph	**Peer Discovery**
	P2P Bootstrapping
	DNS Bootstrapping
	SRV Query Options
	The Channel Graph
	A Directed Graph
	Gossip Protocol Messages
	The node_announcement Message
	The channel_announcement Message
	The channel_update Message
	Ongoing Channel Graph Maintenance
	Conclusion
12. Pathfinding and Payment Delivery	**Pathfinding in the Lightning Protocol Suite**
	Where Is the BOLT?
	Pathfinding: What Problem Are We Solving?
	Selecting the Best Path
	Pathfinding in Math and Computer Science
	Capacity, Balance, Liquidity
	Uncertainty of Balances
	Pathfinding Complexity
	Keeping It Simple
	Pathfinding and Payment Delivery Process
	Channel Graph Construction
	Liquidity Uncertainty and Probability
	Fees and Other Channel Metrics

	Finding Candidate Paths
	Payment Delivery (Trial-and-Error Loop)
	First Attempt (Path #1)
	Second Attempt (Path #4)
	Multipart Payments
	Using MPP
	Trial and Error over Multiple "Rounds"
	Conclusion
13. Wire Protocol: Framing and Extensibility	**Messaging Layer in the Lightning Protocol Suite**
	Wire Framing
	High-Level Wire Framing
	Type Encoding
	Type-Length-Value Message Extensions
	The Protocol Buffers Message Format
	Forward and Backward Compatibility
	Type-Length-Value Format
	BigSize Integer Encoding
	TLV Encoding Constraints
	TLV Canonical Encoding
	Feature Bits and Protocol Extensibility
	Feature Bits as an Upgrade Discoverability Mechanism
	TLV for Forward and Backward Compatibility
	A Taxonomy of Upgrade Mechanisms
	Channel Construction-Level Updates
	Conclusion
14. Lightning's Encrypted Message Transport	**Encrypted Transport in the Lightning Protocol Suite**
	Introduction
	The Channel Graph as Decentralized Public Key Infrastructure
	Why Not TLS?
	The Noise Protocol Framework
	Lightning Encrypted Transport in Detail
	Noise_XK: Lightning Network's Noise Handshake
	Handshake Notation and Protocol Flow
	High-Level Overview
	Handshake in Three Acts
	Conclusion
	Invoices in the Lightning Protocol Suite

15. Lightning Payment Requests	**Introduction**
	Lightning Payment Requests Versus Bitcoin Addresses
	BOLT #11: Lightning Payment Request Serialization and Interpretation
	Payment Request Encoding in Practice
	The Human-Readable Prefix
	bech32 and the Data Segment
	Conclusion
16. Security and Privacy of the Lightning Network	**Why Is Privacy Important?**
	Definitions of Privacy
	Process to Evaluate Privacy
	Anonymity Set
	Differences Between the Lightning Network and Bitcoin in Terms of Privacy
	Attacks on Lightning
	Observing Payment Amounts
	Linking Senders and Receivers
	Revealing Channel Balances (Probing)
	Denial of Service
	Commitment Jamming
	Channel Liquidity Lockup
	Cross-Layer De-Anonymization
	On-Chain Bitcoin Entity Clustering
	Off-Chain Lightning Node Clustering
	Cross-Layer Linking: Lightning Nodes and Bitcoin Entities
	Lightning Graph
	How Does the Lightning Graph Look in Reality?
	Centralization in the Lightning Network
	Economic Incentives and Graph Structure
	Practical Advice for Users to Protect Their Privacy
	Unannounced Channels
	Routing Considerations
	Accepting Channels
	Conclusión
	References and Further Reading
17. Conclusión	**Decentralized and Asynchronous Innovation**
	Bitcoin Protocol and Bitcoin Script Innovation
	Lightning Protocol Innovation
	TLV Extensibility